Volume 39 Number 4 December 2015 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics

Special Issue:

Advances in Secure Data Streaming Systems

Guest Editors:

Fatos Xhafa
Jin Li
Vladi Kolici

1977

Editorial Boards

Informatica is a journal primarily covering intelligent systems in the European computer science, informatics and cognitive community; scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor from the Editorial Board can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the list of referees. Each paper bears the name of the editor who appointed the referees. Each editor can propose new members for the Editorial Board or referees. Editors and referees inactive for a longer period can be automatically replaced. Changes in the Editorial Board are confirmed by the Executive Editors.

The coordination necessary is made through the Executive Editors who examine the reviews, sort the accepted articles and maintain appropriate international distribution. The Executive Board is appointed by the Society Informatika. Informatica is partially supported by the Slovenian Ministry of Higher Education, Science and Technology.

Each author is guaranteed to receive the reviews of his article. When accepted, publication in Informatica is guaranteed in less than one year after the Executive Editors receive the corrected version of the article.

Editors' Introduction to the Special Issue on "Advances in Secure Data Streaming Systems"

With the fast development in networking and cloud computing technologies, data streaming is becoming central to many modern systems. Indeed, on the one hand, the streaming capabilities of networking systems, and especially of mobile ones, have significantly increased. On the other hand, there is a variety of data streaming types such as ones based on sensor networks. The number of challenges raised in data streaming systems is increasing beyond the traditional *QoS* requirements, including optimization and allocation for live streaming, streaming in combined Mobile, P2P and Cloud based systems as well as security and privacy in multimedia and data streaming.

This special issue follows the 8th 3PGCIC-2014, the 8th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 8-10th November, 2014, Guangzhou, China. The special issue comprises 5 papers carefully selected after a two round review process. The papers in the special issue are arranged as follows.

Zhang *et al.* in the first paper "*A Novel Scheme for Improving Quality of Service of Live Streaming*" propose a novel streaming scheme based on a guarantee mechanism of contingency resource (GMCR), which can improve the quality of service (QoS) of live streaming by deploying a contingency server. The results of theoretical analysis and simulation experiment present the feasibility and validity of GMCR scheme.

The second paper by Chen and Lv "*Adaptive Bandwidth Allocation Strategy under Cloud Platform*" present a hybrid file sharing system that combines P2P mode and cloud serving mode aiming to provide both peer-assisted acceleration and cloud-assisted acceleration to download processes. An adaptive cloud bandwidth rental and allocation strategy is then proposed. The experimental results show that the system with this strategy not only ensures the quality of service but also slashed cloud bandwidth consumption.

Kawakami *et al.* in the third paper "*A Churn Resilience Technique on P2P Sensor Data Stream Delivery System Using Distributed Hashing*" investigate research issues arising in sensor data stream delivery. The authors propose an approach to distribute communication loads by relay nodes in the case of delivering the sensor data streams that have different data delivery cycles. A churn resilience technique is therefore proposed that enhances the robustness of delivery system. Through simulations it was confirmed in that the proposed technique improves the reliability of the delivery system.

The fourth paper "*An Experimental Approach to Examine a Multi-Channel Multi-Hop Wireless Backbone Network*" by Taenaka et al., present an experimental deployment of a multi-channel multi-hop wireless backbone network (WBN) with an OpenFlow-based traffic management method. The experimental results show that the proposed WBN can increase the network capacity in accordance with the number of channels, thereby providing significant throughput performance for various applications.

Wang et al. in the fifth paper "*Privacy-preserving Cloud-based Personal Health Record System Using Attribute-based Encryption and Anonymous Multi-Receiver Identity-based Encryption*" present a cloud-based personal health record (CB-PHR) system to securely store their health data on the semi-trusted cloud service providers, and to selectively share their health data with a wide range of PHR users. Extensive analytical and experimental showed that the proposed CB-PHR system is secure, privacy-protected, scalable and efficient.

Fatos Xhafa
Jin Li
Vladi Kolici

A Novel Scheme for Improving Quality of Service of Live Streaming

Guomin Zhang, Chao Hu and Changyou Xing
Department of Network Engineering, College of Command Information System
PLA Univ. of Sci. and Tech. Nanjing, China
E-mail: zhang_gmwn@163.com

Na Wang
Department of Electronic Technology, College of Communication Engineering
PLA Univ. of Sci. and Tech. Nanjing, China

Xianglin Wei
3. PLA University of Science and Technology, Nanjing, China
E-mail: wei_xianglin@163.com

Keywords: live streaming system, peer-to-peer network, quality of service, queuing model

Received: July 16, 2015

P2P live streaming system is one of the most popular Internet applications which developed rapidly in the past decade. However, some common problems, such as long startup delay and unsmooth playback, seriously restrict user's experience on live streaming. In this paper, we propose a novel but simple scheme, namely guarantee mechanism of contingency resource (GMCR), which can improve the quality of service (QoS) of live streaming by deploying a contingency server to provide contingency service for those chunks whose playback deadlines are urgent. Then we establish a queuing model to analyze the quantitative relation between the amount of contingency server resources and the level of user's QoS. Finally, we simulate our scheme in a P2P live streaming simulation platform, and obtain the optimal value of some critical parameters. The results of theoretical analysis and simulation experiment present the feasibility and validity of GMCR scheme.

Povzetek: Predstavljen je nov mehanizem za bolj kvalitetno predvajanje video posnetkov v živo preko spleta.

1 Introduction

Recently, transmitting TV programs over the Internet has become an increasingly popular type of network application. This type of application provides users with abundant, convenient, highly interactive multimedia service, and has engendered a large-scale industry [1]. Live streaming is such a type of application that delivers live program over the Internet, and involves a camera for the media, an encoder to digitize the content, a media publisher, and a content delivery network to distribute and deliver the content [2].

According to the transmission mode in the Internet, the content delivery network of live streaming can be divided into client/server (C/S) paradigm and peer-to-peer (P2P) paradigm. C/S paradigm provides live program from servers entirely, but this transmission mode brings many problems, such as poor scalability and high costs, which makes it difficult to implement large-scale deployment. On the contrary, P2P paradigm exploits the idle resources in end users effectively, who can share and exchange their video chunks, thus improving the operation efficiency and decreasing the costs. There exist many live streaming systems, such as Coolstreaming [6], PPLive, PRIME [13]etc. However, despite P2P can meet the demands of file sharing application, it can't provide QoS guarantee for time-sensitive and bandwidth-sensitive applications, e.g. live streaming. Bo Li et al. measured Coolstreaming and discovered that only about 95% of the chunks could reach user's buffer before being played [5], while Yan Huang analyzed the data collected from PPLive and found the buffering time of almost 20% of the users occupied more than 80% of the total time [6]. Due to these missing chunks, the screen will be frozen, and the media players have to wait until these chunks arrive at the user's buffer, which seriously degrades the quality of experience. An intuitive solution is to increase the amount of streaming servers to cut down the loss rate of the chunks, but in another aspect, too many servers deployed in the Internet will raise the costs and result in the waste of resources.

Therefore, studying the relation between the amount of server resources and user's QoS level, and finding out a scheme to achieve the balance between them have great theoretical significance and practical importance. However, the instinct of IP network is *best effort*, and it's difficult to provide rigorous QoS for users, but the statistical analysis still has important reference to our research. In this paper, from the perspective of improving the QoS of live streaming, we propose a novel but simple scheme, namely guarantee mechanism of contingency resource (GMCR for short), which deploys a contingency server to provide necessary countermeasure to those potential missing chunks to promote user's QoS. Subsequently, we establish a queuing model to analyze the quantitative relation between the resources of contingency server and user's QoS.

The rest of this paper is organized as follows. Related work is introduced in Section 2. GMCR is proposed in Section 3. The quantitative relation of contingency server resources and user's QoS is analyzed by queuing model in Section 4. The performance of GMCR and the results of theoretical model are evaluated by simulation experiment in Section 5. Finally, our work is concluded in Section 6.

2 Related work

Because P2P can alleviate the pressure of the streaming servers and make use of the peers' resources, P2P technology was first introduced to a practical live streaming system in [6]. A data-driven based overlay network, namely DONet, was constructed to implement better transmission and dissemination of the live streaming data. Moreover, AnySee and PRIME [13] also used P2P for the deployment and operation of live streaming systems, and the use of the peers' resources effectively decreased the cost of the servers. However, the intrinsic characteristics of the peers of P2P networks, including dynamic and peer churn, as well as the impact of firewall and network address translation, make it impossible to provide high QoS for the users if a live streaming system entirely relies on the resources from the peers. In [3], CDN and P2P hybrid architecture was proposed to disseminate video streaming. Under this circumstance, there are more server resources, which can lead to better QoS. In [4], a peer-assisted content delivery network was proposed, in which there are two layers, and CDN distribution layer lies in the core network while P2P distribution layer is deployed in the access network. In [14], a radically different cross-channel P2P streaming framework, namely View-Upload Decoupling (VUD), was proposed, which decouples the peer downloading from uploading, bringing stability to multi-channel systems and enabling cross-channel resource sharing to make sure the resource provision is sufficient for user demand in each channel.

Besides, improving resource utility is another way to improve user experience. In [11], a mixed strategy to schedule video chunk was proposed, and it remarkably increased the rate of data arriving at the users' buffer in time. In [12], Yan Yang et al. introduced a deadline-aware scheduling approach, which avoided the waste of resources to a certain extent by considering the data request deadline. In [9], random network coding was employed to P2P live streaming, and it polished up the system performance. In [10], a new P2P streaming algorithm that incorporated the network coding seamlessly with the scalable video coding was designed, and the experiments demonstrated the feasibility and better performance of the approach.

Actual video streams carry highly organized information, part of which is more important than others, and with high variability in the generated bitrate. Chunk loss probability and delivery delay provide therefore only a partial view of the actual performance of a P2P-TV system, the user Quality of Experience (QoE) being the paramount index [15]. In the multimedia and signal processing communities, indeed, the evaluation of the QoE is considered mandatory, see [16], [17] for notable examples. In [15], a realistic simulative model of the system was proposed, which represented the effects of access bandwidth heterogeneity, latencies, peculiar characteristics of the video, while still guaranteeing good scalability properties. Otherwise, a new latency/ bandwidth-aware overlay topology design strategy was proposed, which improved application layer performance while reducing the underlying transport network stress. Reference [15] investigated the impact of chunk scheduling algorithms that explicitly exploit properties of encoded video.

In [18], Hu et al. studied the chunk dissemination of P2P live streaming, and introduce a discrete and slotted mathematical model to analyze chunk selection algorithms, including rarest first algorithm and greedy algorithm. Moreover, Xing et al. presented a performance metric to evaluate chunk selection algorithms, as well as the optimization function for the exploration of chunk dissemination strategies. Reference [19] pointed out the causes of poor performance of these algorithms, and propose a service request randomization mechanism to promote the use of peer resources, which can prevent chunk requests from rendezvous on a few of peers. Simultaneously, they employ weight assignment strategies to avoid excessive requests for rare chunks. Besides, an enhanced model was presented, which adds node degree constraint.

Simultaneously, analyzing P2P live streaming with mathematical model is also a hot spot. In [11], a discrete and slotted model was adopted to study the chunk selection strategy of P2P live streaming. Kumar et al employed stochastic fluid theory to model P2P streaming systems, and exposed the fundamental characteristics and limitations in [20].

When some popular programs start, many users will access this channel during a short time, and if these requests couldn't be handled appropriately, severe performance problem will emerge. Another crucial issue is peer churn, and robust live streaming systems must have the capability against peer dynamics. Practical chunk scheduling mechanisms must be able to deal with these issues. In [21], a radically different cross-channel P2P streaming framework, namely View-Upload Decoupling (VUD), was proposed, which decouples the peer downloading from uploading, bringing stability to multi-channel systems and enabling cross-channel resource sharing to make sure the resource provision is sufficient for user demand in each channel. Kumar et al employed stochastic fluid theory to model P2P streaming systems, and exposed the fundamental characteristics and limitations in [22]. Liu et al. theoretically studied chunk-based P2P video streaming in [23], and showed the delay bound to distribute video chunks to all peers. Furthermore, a conceptual snowball streaming algorithm was proposed to approach the minimum delay bound in dynamic P2P network environment. In [25], the authors presented a novel metric, called the Content Propagation Metric (CPM), to quantitatively evaluate the marginal benefit of available bandwidth, and CPM could guide a global allocation of bandwidth to maximize the aggregate download bandwidth of consumers.

3 Guarantee Mechanism of Contingency Resource

3.1 Basic idea

In P2P live streaming system, source server codes the live television signal, and periodically generates new video chunks, and then distributes these chunks to peers in P2P network. Subsequently, peers share and exchange their possessed chunks to take charge of the partial uploading assignment for source server, which enables source server without powerful upload capacity to provide live service for a great number of users, and improve the scalability of live streaming system.

However, in contrast to the dedicated server, the upload capacity of ordinary peers is limited. Especially, there are plenty of peers locating behind firewall or network address translation, and those devices restrict the resources usage of P2P network, where network resources includes processing resources and bandwidth resources etc., but the performance bottleneck of live streaming system generally lies on uplink bandwidth (In order to prevent terminological confusion, we don’t distinguish resource and uplink bandwidth in the following content). Furthermore, streaming application is different from file-sharing application in time sensitivity. If a chunk doesn’t arrive at peer before being played, it is equivalent to chunk miss even though the chunk reaches user buffer afterwards. Consequently, it is necessary to ensure the amount of resources provided by server or peers must be adequate for the resource requirements of peers. Assume a live channel has n peers and source server’s uplink bandwidth is u_s, the uplink bandwidth of peer i is u_i, and the playback rate is r, then the precondition that all peers can watch live program smoothly is:

$$\frac{u_s+\sum_{i=1}^{n}u_i}{nr}\geq 1 \tag{1}$$

For a single peer, it also needs enough resources for continuous playing. Assume the instantaneous bandwidth peer i derives from source server is u_{is}, and the value is u_{ij} from peer j, where $u_{ii}=0$, and then the instantaneous total bandwidth that peer i receives is $u_{is}+\sum_{j=1}^{n}u_{ij}$. For each peer, it can receive sufficient in instantaneous state when equation (2) is satisfied.

$$\begin{cases} u_{is}+\sum_{j=1}^{n}u_{ij}\geq r,\ for\ all\ i=1,2,...,n \\ \sum_{i=1}^{n}u_{is}\leq u_s \\ \sum_{i=1}^{n}u_{ij}\leq u_j,\ for\ all\ j=1,2,...,n \end{cases} \tag{2}$$

Equation (1) ensures that there are enough resources for users, while it also makes equation (2) have feasible solutions of resource allocation. However, because of the uncertainty of resource scheduling and delay sensitivity of live streaming, it’s difficult to guarantee that every peer can receive sufficient resources to download video chunks for smooth playing every time despite the total resources exceeds users’ requirements from the macroscopic angle.

Every video chunk stored in peer buffer has a deadline apart from being played. Assume the playback deadline is t_p, and this deadline will decrease with the time’s lapse. When chunk’s deadline is less than a certain threshold, and the chunk hasn’t received yet, it is in the risk of missing. Suppose the threshold value is T_u, and if the playback deadline of unpossessed chunk satisfies $t_p \leqslant T_u$, then this type of chunk is called ***urgent chunk***, and the other chunk is ***non-urgent chunk***.

In contrast with those non-urgent chunks, urgent chunk should be served firstly, because the loss probability is much larger than that of non-urgent chunk. Nevertheless, the program progresses of different users are diverse in practical live streaming system, and the resource owners can’t distinguish which chunk is nearer to be played, and they can’t easily find out the more urgent chunk requests and serve them. Consequently, if there is a dedicated server for those urgent chunks to provide contingency service, it’s likely to reduce the loss probability with fewer resources, and improve user’s QoS.

3.2 The model of Guarantee Mechanism of Contingency Resource

Based on the above idea, we propose a scheme, namely guarantee mechanism of contingency resource (GMCR), to promote system performance. GMCR is a scheme to provide contingency service for urgent chunks in order to make sure they can arrive at user’s buffer in time. In practical P2P live streaming system, we can deploy a server that accomplishes GMCR to serve those urgent chunks, and this server is called ***contingency server***. It is a viable solution to the improvement of live streaming QoS to resort to the coordination of contingency server and P2P network.

In order to implement GMCR, we need to adjust chunk scheduling mechanism appropriately. Firstly, peer should partition all the unreceived chunks into urgent chunks and non-urgent chunks according to the value of t_p and T_u. For non-urgent chunks, peer requests them from other peers or source server based on P2P paradigm. Once a non-urgent chunk becomes urgent due to the decrease of t_p, the peer sends chunk request to contingency server immediately, and contingency server responses to this request promptly to provide this urgent chunk. Figure 1 shows the model of GMCR.

In Figure 1, source server and peers constitute the typical P2P live streaming system, and they share and exchange video chunks with each other, while contingency server is dedicated to provide contingency service for urgent chunk request. Peers ought to decide the resource requested object based on the deadline of chunk.

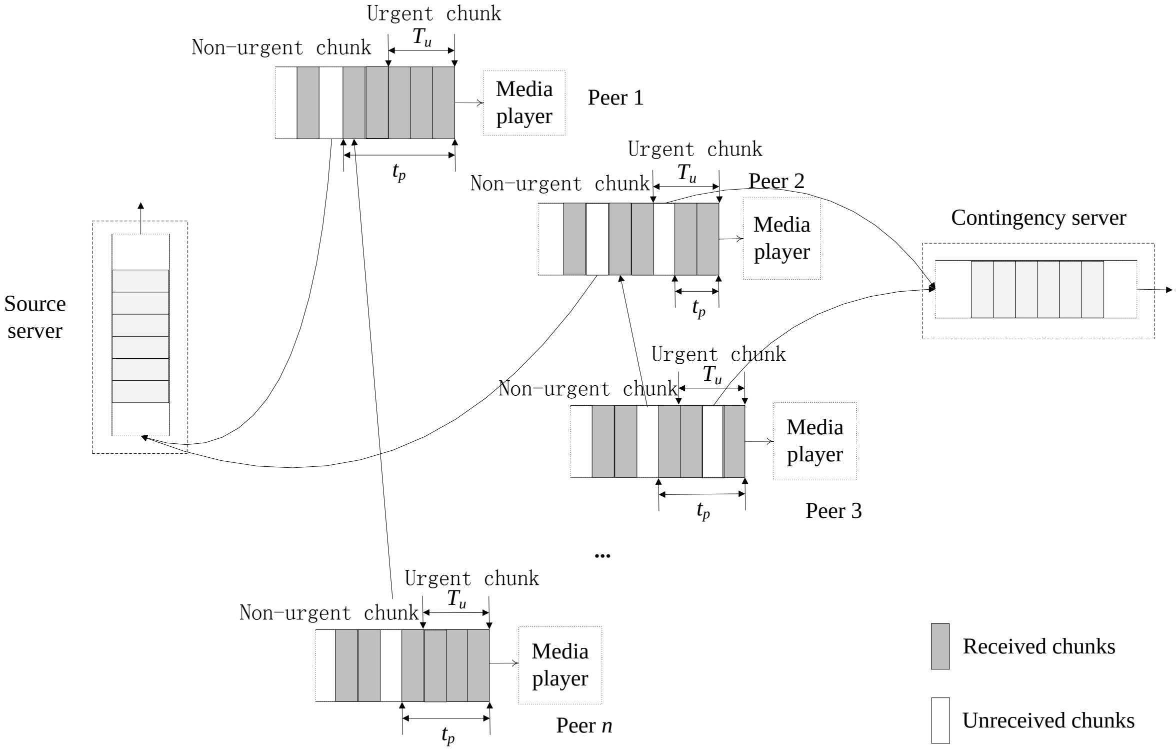


Figure 1: The model of GMCR.

4 Queuing model analysis of Guarantee Mechanism of Contingency Resource

In this section, we analyze the GMCR model quantitatively with queuing theory to discover the relation between the resource amount of contingency server and users' QoS.

4.1 Model and notations

According to GMCR, if the playback deadline of a chunk in peer's buffer t_p is less than T_u, the peer will send request to contingency server immediately to prevent chunk missing. Subsequently, contingency server is going to insert these urgent chunk requests into service queue, and provide contingency service for them to keep the chunk from missing caused by time out. Figure 2 shows the queuing model depicting the above work flow.

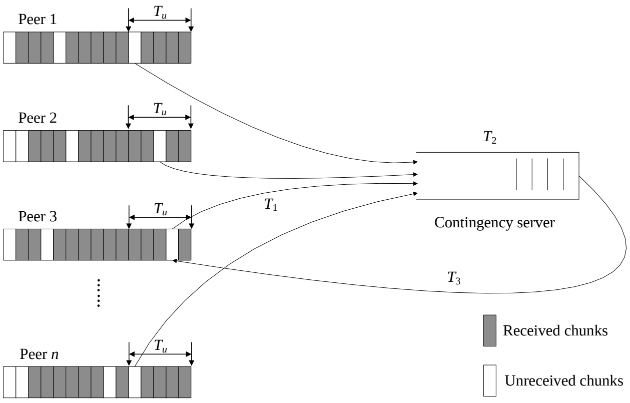


Figure 2: The queuing model for contingency.

To describe the model, we define the system parameters and notations in Table I.

Table 1: Notation and definition of GMCR queuing model.

Notation	Definition
T_u	Playback deadline threshold of urgent chunks
t_p	Playback deadline of chunk away from being played
U	Uplink bandwidth of contingency server
N	The number of peers in live streaming system
R	Playback rate of live program
L	Size of each chunk in live streaming
B	Queue length of contingency server to store urgent chunk requests
a	The probability of chunk that doesn't arrive at peer buffer when its playback deadline reduces to T_u
T_1	The delay of peer sending urgent chunk request to contingency server
T_2	The queuing delay of urgent chunk request in contingency server
T_3	The delay of contingency server sending urgent chunk to peer

4.2 Performance analysis

To analyze the model, we divide time into slots firstly, and the size of slot is equal to the period that source server generates a new video chunk, so the size of each slot is

$$T = L / R \tag{3}$$

When a slot passes, at most one non-urgent chunk will turn into urgent chunk in every peer, and it is going to send an urgent chunk request to contingency server. As defined, *a* is the probability of the chunk that doesn't arrive at peer buffer when deadline reduces to T_u, which means the probability every peer will send request in each slot is *a*. Because the peer's number is *N* in system, the probability that contingency server will receive *k* urgent chunk requests in a slot is

$$a_k = \binom{N}{k} a^k b^{N-k} \tag{4}$$

Where *b*=1-*a*.

On the other hand, the most amount of urgent chunk that contingency server can upload in a slot is

$$M = \left\lfloor \frac{U}{R} \right\rfloor \tag{5}$$

When the arrival ratio of urgent chunk request is less than the capacity of contingency server, the server can handle all the requests in one slot. But if the arrival ratio exceeds the server's capacity, part of the requests have to stay in contingency server's queue.

Assume the state probability that there are *k* urgent chunk requests in contingency server queue to wait for service is s_k, while the queue length is *B*, so all of the state probabilities constitute a state vector as follows.

$$S = [s_0\ s_1\ s_2 \ldots s_B] \tag{6}$$

According to the arrival ratio of chunk request in each slot and contingency server's capacity, we can obtain the state transition diagram in Figure 3.

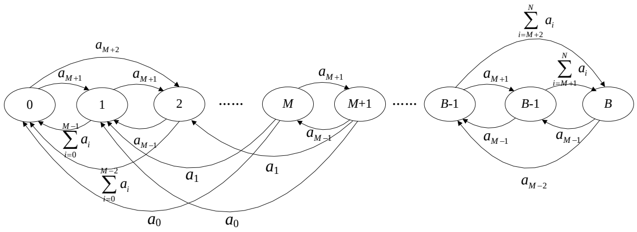


Figure 3: State transition diagram of contingency server's queue.

Therefore, the state transition matrix is denoted as *P*, and the element p_{ij} in the matrix means the probability of the number of urgent chunk request in contingency server's queue transiting from *i*-1 to *j*-1.

$$P = \begin{bmatrix} \sum_{i=0}^{M} a_i & a_{M+1} & a_{M+2} & \ldots & a_N & 0 & \ldots & 0 \\ \sum_{i=0}^{M-1} a_i & a_M & a_{M+1} & \ldots & a_{N-1} & a_N & \ldots & 0 \\ \sum_{i=0}^{M-2} a_i & a_{M-1} & a_M & \ldots & a_{N-2} & a_{N-1} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ a_0 & a_1 & a_2 & \ldots & a_{N-M} & a_{N-M+1} & \ldots & \sum_{i=B-M}^{N} a_i \\ 0 & a_0 & a_1 & \ldots & a_{N-M-1} & a_{N-M} & \ldots & \sum_{i=B-M-1}^{N} a_i \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & \ldots & \ldots & \ldots & a_M \end{bmatrix} \tag{7}$$

At equilibrium, the input probability is equal to the output probability for every state, so the equilibrium equation is given as

$$SP = S \tag{8}$$

Meanwhile, the queue length of contingency server is *B*, so we can obtain the following condition

$$\sum_{i=0}^{B} s_i = 1 \tag{9}$$

Substitute equation (6) and (7) for equation (8), and combine equation (9), we can compute the state probability of contingency server's queue.

For every urgent chunk, if peer can receive this chunk from contingency server in T_u, this video chunk can be played on schedule, otherwise it will be lost. When a peer wants to get urgent chunk, it must suffer three phase latencies, and they are T_1, T_2, and T_3, respectively. Consequently, the condition that ensures urgent chunk arrives in time is T_1+ T_2+ $T_3 \leqslant T_u$, and the longest waiting time of urgent chunk request in contingency server is T_u- T_1- T_3. Because contingency server can deal with $\left\lfloor \frac{U}{R} \right\rfloor$ requests in one slot, the chunk's number that an urgent chunk request can wait for is given by

$$W_m = \left\lfloor \frac{(T_u - T_1 - T_3)}{T} \times \frac{U}{R} \right\rfloor = \left\lfloor \frac{(T_u - T_1 - T_3)U}{L} \right\rfloor \tag{10}$$

Therefore, urgent chunk request can arrive at peer buffer in time when the number of chunk request in contingency server's queue doesn't exceed W_m, and the loss probability p_{loss} is approximately equal to

$$p_{loss} \approx 1 - \sum_{i=0}^{W_m} s_i \tag{11}$$

For instance, if there are 1,000 peers in live streaming system, and the playback rate is 400kbps, the size of chunk is 64kB, and the arrival ratio of urgent chunk is 5%, while contingency server's uplink bandwidth is 21Mbps, and its buffer can store 1,000 urgent chunk requests, the value of T_u, T_1 and T_3 is 2s, 100ms and 100ms, respectively, then we can figure out the loss probability is 0.015% according to equation (11). In fact, considering the quasi-synchronous characteristic of live streaming, all the operation to offer the contingency service can be implemented in the memory of server, and a server can provide service for more users with lower loss probability.

5 Simulation experiment

To validate the feasibility and potential performance of GMCR, we test this mechanism in a P2P streaming simulator, namely P2PStrmSim [24], and compare GMCR to P2P-only mechanism. The purpose of our simulation is to check the dependability of GMCR queuing model, and we contrast the simulation results to theoretical results.

5.1 Experiment scenario and metrics

P2PStrmSim is an event-driven P2P live streaming simulator, and it can simulate the packet-level data exchange process. All the events, including control packet exchange and data packet transmission etc., are stored in event queue; meanwhile, they are managed and executed by event engine. When an event is executed, the corresponding operation will be called. The system framework of P2PStrmSim is depicted as Figure 4.

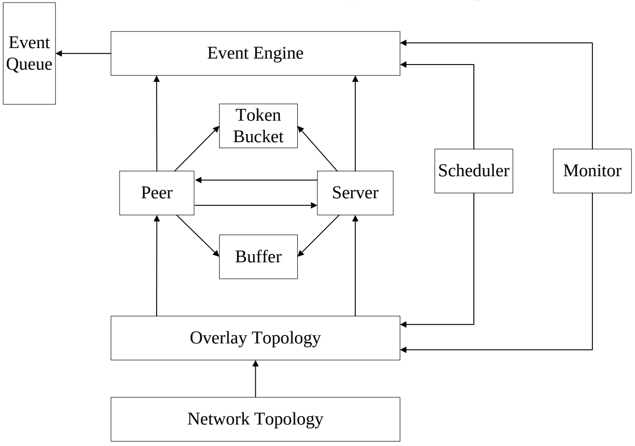


Figure 4: The system framework of P2PStrmSim.

In P2PStrmSim, network topology is based on the measurement results of Internet delay, and it ignores the impact of packet queuing in routers. Peers construct overlay topology on network topology, and every peer has 15 neighbors. The uplink bandwidth of server and peers are implemented by token bucket, and all video chunks are stored in buffer. The function of scheduler is to schedule peers' arrival and departure, while monitor measures the system parameters and performance metrics. Event queue stores all the events of live streaming system generated in the simulation process, including peers' join/leave system, buffer map exchanging between peers to be aware of which chunk other peers have received, as well as sending and reception of the packet. Event engine inserts newly generated event into event queue, and execute the event in the front of event queue. By configuring system parameters briefly, P2PStrmSim can directly simulate the whole process of P2P live streaming. In order to simulate GMCR, we modify the original simulator and append correlative module to implement contingency server. Contingency server takes charge of providing service to urgent chunks, and we introduce three types of delay to denote T_1, T_2 and T_3. In peer module, we add the function of classifying urgent and non-urgent chunks based on T_u, and non-urgent chunks are requested by typical P2P paradigm, while urgent chunk requests are sent to contingency server. The simplicity of developing GMCR also indirectly demonstrates the feasibility of this mechanism.

The scheduler can adjust communication paradigm according to the lowest quality requirement of various paradigm after obtaining performance information. There are mainly two parts in adaptive communication mechanism: network performance awareness module and adaptive paradigm adjustment module. The former is devoted to obtain end to end performance information through measurement and inference technology; and the latter intends to adopt the optimum communication paradigm to fulfill user requirement.

In practical experiment scenario, the uplink bandwidth of source server is 10Mbps, and it generates video chunk with the rate of 400kbps, contingency server can store 1,000 urgent chunk requests. Simultaneously, some peers in the system download the video chunks according to the fixed scheduling mechanism. Moreover, the request window is 30s, and the size of the video chunk is 64kB. All peers access the network by asymmetrical digital subscriber line (ADSL), and the downlink bandwidth exceeds the uplink bandwidth. In order to simulate the heterogeneity of the peer's access bandwidth, we introduce three types of ADSL, whose uplink bandwidths are 1Mbps, 384kbps and 128kbps, respectively, and their proportions are given in Table II.

Table 2: The proportion of the three type of uplink bandwidth in the simulation experiment.

Uplink bandwidth	Proportion in experiment
1 Mbps	0.2
384 kbps	0.45
128 kbps	0.35

We evaluate the performance of GMCR by monitoring the continuity of live program. Once a video chunk doesn't arrive at peer's buffer before being played, it will lead to program pause or screen frozen, so we employ chunk arrival ratio (CAR), which is equal to the ratio of chunks arriving in time and the total chunks, as the metric to evaluate user QoS. Obviously, the higher CAR means users have received higher QoS. In our simulation, all the results are the average value of 10 experiment results tested with different seeds.

5.2 Experiment results and analysis

5.2.1 Performance of Guarantee Mechanism of Contingency Resource

Figure 5 shows the CAR's cumulative distribution function (CDF) of GMCR and P2P based live streaming in different scale channels, where N is the peer's number, the uplink bandwidth of contingency server is 2Mbps, and T_u is 2s.

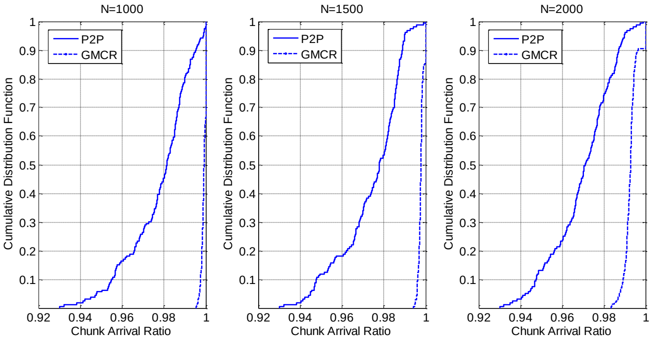


Figure 5: The CDF of CAR in GMCR and P2P based live streaming system.

From Figure 4 we can find GMCR can provide better QoS for users in all different scale channels, and almost all the peers have a chunk arrival ratio above 99%. But in typical P2P paradigm, there are more than 15% of the peers whose loss probabilities exceed 4%. Though the uplink bandwidth of source server is not high, contingency server is deployed in live streaming system and it can provide contingency service to the peers when they have urgent chunk, which greatly reduces the amount of chunk missing and improves CAR. Furthermore, the CARs of GMCR and P2P based system decreases in different degree with the increment of peers. This trend of performance degradation is due to the augment of user demand on the bandwidth resource, and the total amount resource provision of source server and contingency server is invariable.

5.2.2 The impact of T_u

According to the analysis in Section 3, the length of urgent chunk's deadline threshold T_u will affect system performance, because T_u decides the probability of urgent chunk generated in peer buffer and the length of queuing time that these urgent chunk requests can wait in contingency server. If the value of T_u is set to be larger, the probability of non-urgent chunk turning into urgent chunk will increase, and then more urgent chunk requests reach the contingency server, thus augmenting its burden. But if the value of T_u is set to be too small, the chunk request that can stay in contingency server will shorten, thus also increasing the probability of urgent chunks that cannot be played in time. Hence, there exists an appropriate value of T_u that could achieve the optimal system performance. In our experiment, we adjust T_u to get a serial of CDF curves of chunk arrival ratio. Figure 6 shows the CDF of chunk arrival ratio with different T_u.

From the movement trend of Figure 6, we can analyze the rough impact of T_u. Obviously, the curves shows that the value of T_u has significant influence on system performance. When T_u changes from 4 seconds to 2 seconds, chunk arrival ratio increases clearly. But when T_u is set to be 1 second, chunk arrival ratio drops down. This result indicates that GMCR can provide fine contingency service and avoid too many urgent chunk requests when T_u is set to be about 2 seconds.

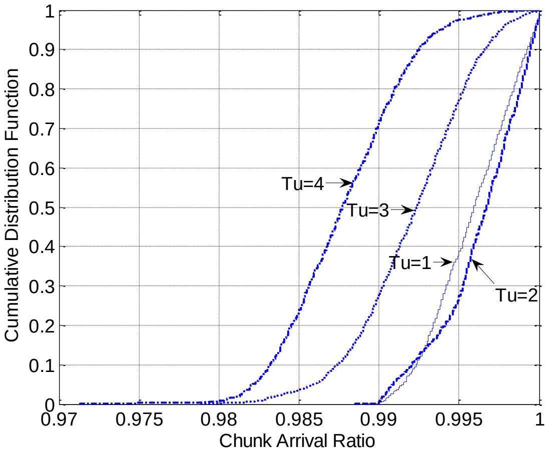


Figure 6: The influence of T_u on system performance.

5.2.3 The impact of contingency server resource

Subsequently, we analyze how to deploy contingency server resource in GMCR based live streaming. Theoretically, many urgent chunk requests are unable to be served without sufficient resource, and the user QoS will deteriorate. But if too much resource of contingency server is deployed in the system, it will cause the waste of resource. Therefore, it is necessary to supply appropriate contingency server resource according to the practical requirement. Figure 7 shows the movement curves of loss probability in contingency server under different scale channel when the uplink bandwidth of contingency server changes from 1.95Mbps to 2.1Mbps. At the same time, Figure 7 also shows the theoretical value to validate the dependability of GMCR queuing model established in Section 4.

According to Figure 7, we can obtain three conclusions. Firstly, in three different scale channels, the theoretical results are very close to the experimental results, which demonstrates GMCR queuing model can describe the quantitative relation between contingency server resource and user QoS, and this queuing model has great dependability. Secondly, when contingency server has insufficient resource, the loss probability is very high, but if the amount of resource exceeds the demand of urgent chunk request, user QoS will improve significantly. Hence, we should deploy a few more contingency server resources. Thirdly, the requirement of contingency server resources will increase with the augment of peer's number, because this situation will incur the insufficiency of resource provision and result in the increment of urgent chunk request, and more contingency server resources are needed for contingency service. But the requirement increment of contingency server resource is not notable, where the resource amount only increases from 1.95Mbps to 2.1Mbps.

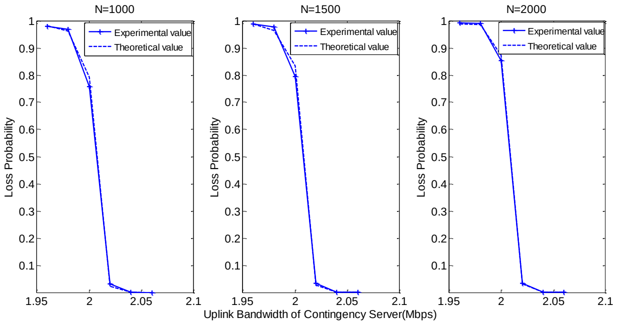


Figure 7: The loss probability of theoretical value and experimental value in contingency server under different scale channel.

6 Conclusion

In this paper, we propose a novel but simple scheme, namely guarantee mechanism of contingency resource, which greatly improves the QoS of live streaming system by deploying contingency server to provide service for urgent chunk request in time. We also establish GMCR queuing model to analyze the quantitative relation between the amount of contingency server resource and user QoS. Finally, we simulate our scheme in simulation experiment, and obtain some conclusions. The experimental results and the theoretical analysis show the dependability and validity of GMCR and this queuing model. Our work sheds light of a new approach for the QoS elevation of live streaming system. In the future, we will focus on the technological approach to construct mathematical model on the different requirement of system resource in distinct phase to improve the QoS of live streaming system.

Acknowledgement

This work is supported by the National Basic Research Program of China under Grant No. 2012CB315806, the National Natural Science Foundation of China under Grant No. 61402521, 61070173 and No. 61103225, and Jiangsu Province Natural Science Foundation of China under Grant No. BK20140070 and BK20140068.

References

[1] Multimedia Research Group Inc. [EB/OL]. Available: http://www.mrgco.com/iptv/ gf1210.html.

[2] Live streaming. Available [EB/OL]: http://en.wikipedia.org/wiki/Streaming_media.

[3] Karl Skevik, Vera Goebel, Thomas Plagemann. Design of a hybrid CDN [C]. Second International Workshop on Multimedia Interactive Protocols and Systems, Grenoble, France, 2004: 206-217.

[4] Yu Liu, Yin Hao, Guangxi Zhu, et al. Peer-assisted content delivery network for live streaming: architecture and practice [C]. International Conference on Networking, Architecture, and Storage, Chongqing, China, 2008: 149-150.

[5] Bo Li, Susu Xie, Yang Qu, et al. Inside the new coolstreaming: principles, measurements and performance implications [C]. INFOCOM, Phoenix, USA, 2008: 1031-1039.

[6] Yan Huang, Tom Z. J. Fu, Dah-Ming Chiu, et al. Challenges, design and analysis of a large-scale P2P-VoD system [C]. ACM SIGCOMM, Seattle, Washington, USA, 2008: 375-388.

[7] Xinyan Zhang, Jiangchuan Liu, Bo Li, et al. CoolStreaming/DONet: a data-driven overlay network for efficient live media streaming [C]. INFOCOM, Miami, USA, 2005: 2102-2111.

[8] Xiaoqun Yuan, Geyong Min, Yi Ding, et al. Adaptive resource management for P2P live streaming systems[J], Future Generation Computer Systems, Vol.29, No.6, 2013.08: 1573－1582.

[9] Mea Wang, Baochun Li. R2: random push with random network coding in live peer-to-peer streaming [J]. Journal on Selected Areas in Communications, 2007, 25(9): 1655-1666.

[10] Anh Tuan Nguyen, Baochun Li, Frank Eliassen. Chameleon: adaptive peer-to-peer streaming with network coding [C]. IEEE INFOCOM, San Diego, CA, USA, 2010: 1-9.

[11] Yipeng Zhou, Dah-Ming Chiu, John C.S. Lui. A simple model for chunk-scheduling strategies in P2P streaming [J]. IEEE/ACM Transactions on Networking, 2011, 19(1): 42-54.

[12] Yan Yang, Alix L.H. Chow, Leana Golubchik, et al. Improving QoS in bitTorrent-like VoD systems [C]. INFOCOM, San Diego, CA, USA, 2010: 2061-2069.

[13] Nazanin Magharei, Reza Rejaie. PRIME: peer-to-peer receiver-driven mesh-based streaming [J]. Transactions on networking, 2009, 17(4): 1052-1065.

[14] Di Wu, Chao Liang, Yong, Liu, et al. View-Upload Decoupling: A Redesign of Multi-Channel P2P Video Systems [C]. INFOCOM, Janeiro, Brazil, 2009: 2726-2730.

[15] Fortuna, R., Leonardi, E., Mellia, M., Meo, M., Traverso S. QoE in Pull Based P2P-TV Systems: Overlay Topology Design Tradeoffs[C]. IEEE P2P 2010

[16] Z. Shen and R. Zimmermann, ISP-friendly peer selection in p2p networks[C]. in ACM Multimedia, Beijing, China, October 2009.

[17] E. Setton, J. Noh, and B. Girod, Low latency video streaming over peer-to-peer networks[C]. in IEEE ICME, Toronto, Canada, July 2006.

[18] HU Chao, CHEN Ming, XING Changyou. Towards Efficient Video Chunk Dissemination in Peer-to-Peer Live Streaming[J]. Computer Networks, Vol. 57, issue 15, pp.3009-3024, 2013.

[19] XING Changyou, CHEN Ming, HU Chao. Capacity aware Scalable Video Coding in P2P on Demand Streaming Systems[J]. KSII Transactions on Internet and Information Systems, Vol. 7, issue 9, pp. 2268-2283, 2013.

[20] Rakesh Kumar, Yong Liu, Keith Ross. Stochastic Fluid Theory for P2P Streaming Systems. IEEE INFOCOM, Anchorage, Alaska, USA, 2007: 919-927.

[21] R. S. Peterson, B. Wong, E. G. Sirer, A content propagation metric for efficient content distribution[C], in: ACM SIGCOMM 2011, New York, NY, USA, 2011, pp. 326-337.

[22] A. P. C. da Silva, E. Leonardi, M. Mellia, M. Meo, S. Traverso, A bandwidth-aware scheduling strategy for P2P-TV systems[C], in: 8th International Conference on Peer-to-Peer Computing, Aachen, Germany, 2008, pp. 279-288.

[23] Z. Liu, C. Wu, B. Li, S. Zhao, UUSee: large-scale operational on-demand streaming with random network coding[C], in: IEEE INFOCOM 2010, San Diego, CA, USA, 2010, pp. 1-9.

[24] Peer-to-Peer Streaming Simulator. http://media.cs.tsinghua.edu.cn/~zhangm/download/. 2012

[25] R. S. Peterson, B. Wong, E. G. Sirer, A content propagation metric for efficient content distribution, in: ACM SIGCOMM 2011, New York, NY, USA, 2011, pp. 326-337.

Adaptive Bandwidth Allocation Strategy under Cloud Platform

Hong Li Chen and Shan Guo Lv
Software School, East China Jiaotong University, Nanchang, China
E-mail: chl@ecjtu.jx.cn

Keywords: cloud computing; cloud bandwidth allocation; cloud collaboration synergy

Received: July 16, 2015

With the rapid development of cloud-computing technologies, more and more Internet applications appear with cloud platform. In this paper, cloud computing is introduced. Renting cloud platform which provides computing, storage, bandwidth resources can improve the performance of file sharing systems. The hybrid file sharing system combines P2P mode and cloud serving mode. This system provides both peer-assisted acceleration and cloud-assisted acceleration to download processes. Cloud bandwidth is scalable in the cloud-assisted file sharing system. In order to save cost while meeting QoS requirement, author conducts measurement and analysis on the QQ offline downloading system to find key factors which impact the cloud bandwidth consumption of download process. An adaptive cloud bandwidth rental and allocation strategy is proposed. The experimental results show that the system with this strategy not only ensures the quality of service but also slashed cloud bandwidth consumption.

Povzetek: Predlagana je izboljšava računalništva v oblaku.

1 Introduction

With the rapid development of cloud-computing technologies, more and more Internet applications appear with cloud platform. As popular services with a large amount of users, P2P (peer-to-peer) file distribution systems are also evoluting toward cloud, and then turning into the cloud-assisted file sharing systems. Cloud computing is dynamic and extensible. It usually provides the virtual computing model of resources through the internet. Its major features are capable of rapid deployment of resources or access to services, scalability on-demand and use. It provides service through the internet.

In order to enhance the quality of file sharing and save system cost, we study on real system to solve these questions. Network measurement is considered to be one of the important means to understand the internet and its applications. It is also often used to find problems in network applications and provides the basis for system optimization. P2P file-sharing system which is the most popular web application has become the main source of Internet traffic. P2P file-sharing system has attracted many scholars to start measurement researching. Some researchers found that there are two obvious flaws in this system. One is the availability of file resources can not be guaranteed, the other is the huge download speed difference of users [1]. On the one hand, the availability of documents can not be guaranteed because there are no central servers saved the actual contents of the file in P2P file-sharing system. File data is completely provided by the user. But user may be online or offline, the dynamic behavior can not guarantee to provide a complete copy of all documents at any time. Therefore system can not guarantee the availability of documents. On the other hand, there are two factors for the difference of users download speed. One is the difference between supply and demand in different files, the other is the difference of connectivity between different network nodes. User may even suffer from extremely low download speed. Studies have shown that node connectivity between each other is very good in the situation of campus network environment, but every day there will be a large number of download task stop or restart due to access not any file data in P2P file sharing system. Therefore the most fundamental reason leading to above two defects is that P2P file sharing system service model, rather than the node network connectivity. In order to compensate for the lack of pure P2P model, some scholars have put forward the hybrid file-sharing publishing system, this hybrid system combines the CDN (content distribution network) and P2P system [2,3]. In this hybrid system some popular shared files will be uploaded to the CDN server, users can download the popular file data from the server. But there is almost no effect on the improvement of non-popular file download experience.

With the improvement of cloud technology and cloud infrastructure, renting cloud platform provides computing, storage, bandwidth resources. It can improve the performance of P2P file sharing and distribution system. People apply the idea to practice, forming a cloud-assisted P2P file sharing application. For example, QQ offline download is also known as cloud download [4], which provides a support for the appointment file download service. After receiving the user's download request, the system takes over the user's download task through renting computing resources from the cloud platform. User does not need to online wait after user submits a download request. The renting computing resources serve as download machines to finish the task.

The download file will be uploaded to the storage space rented from a cloud platform [5-7]. This storage space is called cache cloud. The system immediately notifies the user can download the booking documents in the next period of time (e.g. seven days).

At present the measurement research of the cloud assist P2P file sharing system is lacking. In this paper, using QQ cyclones offline download system for measuring object, we research the file availability, user's download experience and system cost. First the offline download system architecture and working principle are introduced. Second, through the actual measurement we analyze the characteristics of the offline download system, understand how to solve the defects of the file availability and user download speed in the cloud assist service mode and analyze the load of the offline download system and cloud bandwidth consumption. Finally, an adaptive cloud bandwidth rental and allocation strategy is proposed. The strategy is applicable to the cloud assist P2P file sharing system. It can maximize the synergy between the node acceleration and cloud acceleration and guarantee the quality of service and save the cloud bandwidth cost.

The rest of this paper is organized into five sections. In Section 2, the offline download system architecture and working principle are introduced. In Section 3, we measure the actual system and analyze the effectiveness of cloud collaboration service model. An adaptive cloud bandwidth rental and allocation strategy is proposed in section 4, discuss how to guarantee the quality of service and save the cloud bandwidth cost. Experiments and discussions are detailed in Section 5. This paper is concluded in section 6.

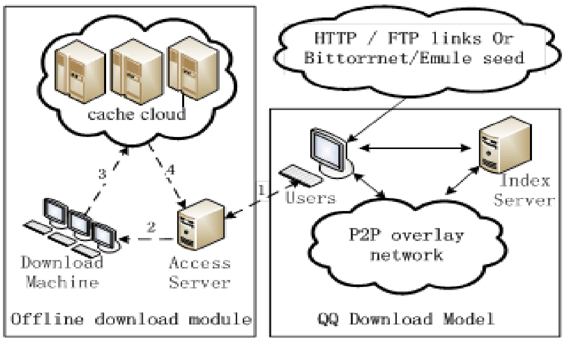


Figure 1: Offline Download System Architecture.

2 Offline download system architecture and operating principle

In order to improve the service quality of the existing cloud –assisted file sharing application system, we collected the actual operational data of QQ offline download system for 60 days. In this section, we first describe the offline download system architecture. Then describe its working principle.

2.1 System architecture

QQ offline download service is built based on the original QQ Tornado download acceleration system architecture. In the original QQ cyclone system, the source that users download can be ordinary HTTP/FTP links and Bittoirent/eMule. After the client has obtained the Hash of content about the target download file, it will communicate with QQ index server to get other online nodes information. These online nodes are sharing user's target file. Therefore, in the process of the download we can simultaneously acquire data from the source and QQ P2P networks. In Figure 1, the right part is the architecture of the conventional QQ download system. The left part is the new modules.

QQ offline download system is composed of the following components.

(1) Caching cloud: The general term lease from the cloud platform to store and upload bandwidth resources. Cache cloud is composed of many cloud storage servers which are distributed in different network locations. Cache cloud is used to cache the files which users appoint offline download, and accelerate the retrieve user's files process.

(2) Access server: It holds the information of all files currently stored in the caching cloud. Such as the file original source link, content Hash, file size and so on. In addition, if the file that user requests to download is not yet stored in the caching cloud, the access server will send the relevant information of the target file to download machine and assign download machine to take over the download task.

(3) Download machine: Download machine is composed of a large number of virtual machines. The physical location of the virtual machine may be distributed in a number of different areas. According the information received from the access server, download machine plays online download node to help users downloading the target file, and upload the successful download files to the caching cloud.

2.2 Operating principle

When users use offline download, the download process can be divided into two steps: add tasks and retrieve files.

(1) Add task: When the user requests a file offline download, QQ client will send the file information including content Hash and source link to access server of offline download system. It is shown as the arrow 1 in Figure 1. If the target file is not stored in the cache cloud, the access server will send the received file information to download machines, and assign one of the virtual machines to take over the user's download task. It's shown as the arrow 2 in Figure 1. The download progress will be informed the client in real time. In this case, users do not need to wait for download process online. After the file is completely downloaded, download machine will upload the file to caching cloud. It's shown as the arrow 3 in Figure 1. Then the caching cloud will generate a URL for the received file and send

information to the access server. It's shown as the arrow 4 in Figure 1. Access server will store cache cloud URL, file Hash and file size in the database. As long as the requested file already stored in the cache cloud, the client will receive the URL that access server sends. The user is informed immediately that file retrieval operation can be carried out at any time within a certain period of validity. The process of add a task completes. Of course, if the user initiate offline download request, the target file already stored in the caching cloud, then the process of adding task complete immediately. Access server can immediately notify user to start high speed retrieve the file.

(2) Retrieve file: In the offline download application, users download the request file to a local machine. The process is called retrieve file. In this process, the data may come from a source link of the file, the QQ node that is sharing the file and a cloud that has been stored the file. It is shown as the broken line arrows in Figure 1. The primary task of the cache cloud URL is to ensure availability of the files. In addition, the initial system design, it provides consistent upload speed limit for each retrieved file.

3 Offline download system measurement and analysis

By actual measurement, we expect to get the following features of the offline download system: cloud service model can help solve the defects of original P2P file-sharing file system; in the user file retrieve process we monitor cloud data transfer rate, system load and the corresponding cloud bandwidth consumption; find the major factors which affect the bandwidth consumption of cloud in the file retrieval process. According to the above measurement target, we set the data information obtained. First the measurement data set is described. Subsequently, display and analyze the measurement results.

3.1 Measurement methods and data set description

According to the measurement target, we need obtain the following data.

(1) The requested file has been stored in the cache cloud. There is at least one complete copy in the system. Therefore, the proportion of the request reflects the ability of guaranteeing file availability. We can get the proportion from the requests stored in access server.

(2) Study the effect of cloud assist in improving the download speed, we need to compare the offline download file retrieval process and the data transfer rate of ordinary QQ download process. The duration and the amount of total download data are required in each acquisition process.

(3) The proportion of various types of data transfer rates in the file retrieval process, we need to get statistical task duration, the amount of data download from cache cloud, P2P networks and the source link respectively.

(4) Get the start time and the stop time of each file retrieval process. From the two times we can calculate the number of concurrent tasks in the system at any time. It reflects the system load. According to each file download speed from cache cloud, we can approximately get the cloud bandwidth consumption of offline download system at any moment.

(5) In each file retrieval process we extract the identity of the download target file, the download user ID and the identity of the user's network connection type. It can help us to reveal whether the different users have different download experience when they retrieval different documents.

The above data, the total amount of offline download request and the amount of request completed immediately can be obtained from offline download access server, the rest information needs to extract from the user QQ client records.

3.2 The effectiveness of cloud collaboration service model

On the basis of measurement results, we analyze the cloud collaboration service model. Whether the existing cloud collaboration service model can solve the availability of documents and users download speed in P2P file sharing system.

(1) Availability of download file

We have a statistics of the users' retrieval files through offline download. The amount of their respective downloads is rare. In all the offline download files, more than 50% of the resources are only downloaded once within the 10-day period. 91.7% of the average daily downloads is less than one time, while the proportion of documents more than 10 times only is 0.44% in the average daily downloads. These results indicate that the files which user requests by offline download service are almost non-popular documents. The file retrieval process can start, namely, the target file already stored in the cache cloud, and the cache cloud can provide high-speed upload for users. Therefore, file availability is guaranteed. The resource of user request have stored cache cloud, user can get the best download experience, but the resource is not stored cache cloud, user need to wait the download machine to finish the download task. The proportion of cache cloud files is bigger, the user download experience better. During the measurement period of two months, we found that the proportion every day is more than 94%. This results show that users can immediately begin high-speed download after users send a offline download request. So cloud assist offline download service is good to guarantee the availability of download files.

(2) Download speed

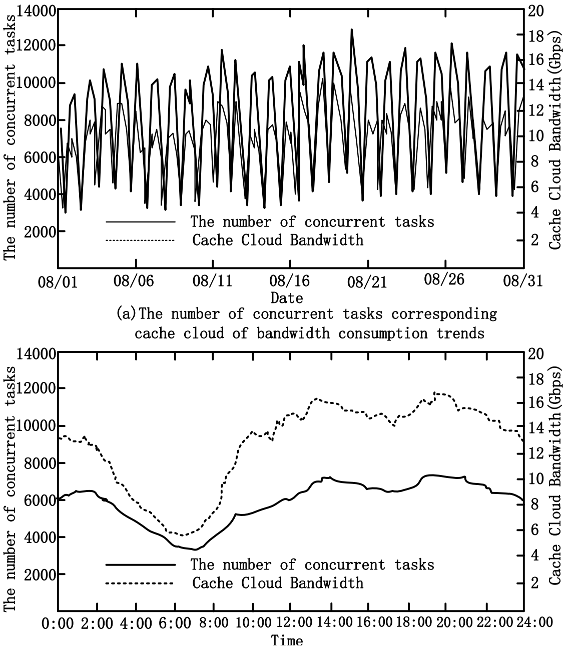
Download speed is generally considered the most important performance indicators in the file sharing system. Now we analyze the measurement results of the data transfer rate in the process of offline download files retrieving. Compared offline download process with general QQ download process, the average download speed can evaluate the effectiveness of improving download speed of cloud collaboration.

The traditional QQ download system mainly depends on the node accessibility to accelerate the download process. For the user, offline download system provides the cloud collaboration and synergy auxiliary node acceleration services. Cloud acts as a significant role in assisting acceleration mode, but in the process of offline download file retrieval download speed obtained from QQ P2P network is very limited, with an average of only 28 KB/S. Node auxiliary acceleration effect on offline download file retrieval process is not as good as on traditional QQ download task. This maybe that most offline download requests are unpopular files, also maybe that the stable data transmission rate provided by cache cloud seizes the user download bandwidth.

Cloud Collaboration offline download service significantly improves the performance of P2P file sharing system. The system leases computing resources from cloud platform to act as download machines which take over the download tasks. The system rents large capacity cloud storage resources to store user request download files. It greatly improves the usability of the unpopular file. The system also rents the cloud bandwidth to provide cloud collaboration acceleration for user file retrieval. The model can ensure the user high-speed downloads. Even if the request download file is unpopular, it can not accelerate from a P2P network, but it can still get very high download speeds. With the rapid expansion of the user group, file retrieval process over reliance on cloud acceleration method is bound to the offline download system brings a huge cloud bandwidth consumption.

(3) Offline download system load

Actual measurements reflect the bandwidth consumption and the load condition in the cloud offline download system. Figure 2 (a) depicts the number of concurrent tasks and the corresponding cache cloud bandwidth consumption in the offline download system during the first 4 weeks of the measurement cycle. It shows that the consumption of cloud bandwidth in the offline download system changes as the parallel number of the file retrieval process. Bandwidth consumption reaches a peak value of 18Gbps. Through the characterization of system load changes in a single day, we can observe more clear cache cloud bandwidth consumption and the relationship of system parallel tasks. Figure 2 (b) depicts the system load and upload bandwidth evolution of cache cloud in a single day. The figure shows the number of running tasks in the system is relatively less from 5:00 to 7:00 in the morning. In this period of time the cache cloud bandwidth consumption remained at the lowest level of the day. From 8:00 to 13:00 in the morning file retrieval process running in the offline download system steady growth. Upload bandwidth consumption of cache cloud will also rise. Both reached a peak in the vicinity of 20:00 in the day. In the case of node assisted download acceleration, it still shows strong positive correlation between cache cloud bandwidth consumption and the number of parallel tasks. This will cause a great deal of system operation cost. It is essential to reasonable cache cloud bandwidth allocation strategy.



(a)The number of concurrent tasks corresponding cache cloud of bandwidth consumption trends

(b)System load evolution in a day

Figure 2: QQ offline download system cache cloud bandwidth consumption trend.

4 Cache cloud bandwidth strategy

If the potential of the node acceleration mode can be fully played out, we can not only ensure the user download speed but also reduce the data transmission speed which is provided for file retrieval process by cache cloud. This can reduce the system cloud bandwidth consumption, save the operating cost of the system. Therefore, this section an adaptive cache cloud strategy is proposed. It applies to the mixed model of cloud collaboration and P2P file sharing system. The strategies maximize the synergistic effect on node Auxiliary Acceleration and cloud assist acceleration. This can guarantee the quality of service and save the bandwidth overhead of cloud.

The following is the design idea of the adaptive cache cloud bandwidth strategy. We first get the download speed of the file retrieval process obtained from P2P networks, calculate cloud acceleration rate with QoS indicators in each file retrieval process, and then get cache cloud bandwidth rented from cloud platform.

Assuming at time t there are N(t) files are in offline download file retrieval process， these files can be expressed as $I=\{1,2,\ldots,N(t)\}$, any file $i\in I$,useful information extracted from the index server are summarized as follows: $D(i,t)$ is the number of users downloading the file i at time t ; $D_{NAT}(i,t)$ is the number of NAT users downloading the file i at time t ; $D_d(i,t)$ is the number of directly connected users downloading files i at time t ; $S_{NAT}(i,t)$ is the number of NAT users as a seed file i at time t ; $S_d(i,\mathrm{t})$ is the number of directly connected users as a seed file i at time t ; $D^o_{NAT}(i,t)$ is the number of NAT users using offline download to retrieve files i at time t ; $D^o_d(i,t)$ is the number of directly connected users using offline download to retrieve files i at time t .

NAT Users upload bandwidth is expressed as u_{NAT} , directly connected users upload bandwidth is expressed as u_d . P2P connections can only be initiated by the NAT user itself, initiated by an external node connection requests will be masked the NAT [8]. For the file i , all users can provide at time t available upload bandwidth, which is calculated as (1) shown below:

$$U_{NAT}(i,t)=u_{NAT}\times D_{NAT}(i,t) \qquad (1)$$

All directly connected users who is sharing file i can provide at time t available upload bandwidth, which is calculated as (2) shown below:

$$U_d(i,t)=u_d\times[D_d(i,t)+S_d(i,t)] \qquad (2)$$

NAT users who are Downloading file i can obtain the upper limit of the average speed of P2P, the upper limit is calculated as (3) shown below:

$$\begin{aligned} r_{P2P,d}(i,t) &= \frac{U_d(i,t)}{D(i,t)}+\frac{U_{NAT}(i,t)}{D_d(i,t)} \\ &= \frac{u_d[D_d(i,t)+S_d(i,t)]}{D_d(i,t)+D_{NAT}(i,t)}+\frac{u_{NAT}D_{NAT}(i,t)}{D_d(i,t)} \end{aligned} \qquad （3）$$

Collaboration in the cloud service model, cloud bandwidth leased in accordance with the cycle, the system can adjust the amount of bandwidth leased before the arrival of each lease period. in every lease cycle (assuming each lease period length T) within a cloud platform provides a fixed maximum upload bandwidth offline download system, adaptive caching cloud strategy we propose is that the bandwidth of the decision before the arrival of each lease period Cloud Bandwidth the amount of rent and hire cycles in the cloud cache bandwidth reallocation necessary. Strategies are as follows:

Assuming a cloud bandwidth leasing period beginning from the time t， cache cloud provide bandwidth for offline download file retrieval process of file i ,the bandwidth is calculated as (4) shown below:

$$\begin{aligned} &J_{req}(i,t)=K_{NAT}(i,t)[R_{thres}(t)-r_{P2P,NAT}(i,t)]\times \\ &D^o_{NAT}(i,t)+K_d(i,t)[R_{thres}(t)-r_{P2P,d}(i,t)]D^o_d(i,t) \end{aligned}$$
(4)

$R_{thres}(t)$ represents cache cloud provide acceleration threshold value at time t , $K_{NAT}(i,t)$ represents NAT users cloud acceleration switch, the values is calculated as (5) shown below:

$$K_{NAT}(i,t)=\begin{cases}1 & r_{P2P,NAT}(i,t)<R_{thres}(t)\\ 0 & r_{P2P,NAT}(i,t)\geq R_{thres}(t)\end{cases} \qquad (5)$$

$K_d(i,t)$ represents directly connected users cloud acceleration switch, the values is calculated as (6) shown below:

$$K_d(i,t)=\begin{cases}1 & r_{P2P,d}(i,t)<R_{thres}(t)\\ 0 & r_{P2P,d}(i,t)\geq R_{thres}(t)\end{cases} \qquad (6)$$

At time t, all running the file retrieval process of the total demand for bandwidth cache cloud, the total demand is calculated as (7) shown below:

$$J_{total}(t)=\sum_{i=1}^{N(t)}J_{req}(i,t) \qquad (7)$$

Therefore, from time t to $t+T$ within the cloud bandwidth rental period, system leased from the cloud platform cache cloud bandwidth $B=J_{total}(t)$.

If the lease period, no other file $j\notin I$ retrieval process is started, for any file $i\in I$, cache cloud provides a total bandwidth of all the file retrieval process within the renting cycle always $J_{req}(i,t)$.

If at time $t'\in(t,t+T)$, start the file retrieval process of file $j\notin I$, then the system cache cloud reallocation of bandwidth leased. first, the update the file retrieval process is in the collection of all the files, the updated collection is $I=\{1,2,\ldots,N(t')\}$, then obtains the number of users corresponding to each file $i\in I$ and the user type information of network connection, according to the equation (4) calculate the cache bandwidth $J_{req}(i,t')$ it needs to provide the cloud, Finally, calculate all the tasks needed to retrieve the cache files total cloud bandwidth $J_{total}(t)$.

If $J_{total}(t')\leq B$, the system first assigned to the file i size of $J_{req}(i,t')$ cache cloud Bandwidth, the remaining part of the cache cloud of bandwidth $B-J_{total}(t')$ will be allocated equally to each the file retrieval process is running.

If $J_{total}(t')>B$, to be fair, caching cloud assign cache cloud bandwidth to all the file retrieval processes of the files i .the bandwidth is $B\times J_{req}(i,t')\,/\,J_{total}(t')$.

In the lease cycle, whenever the file retrieval process of a new file start, more than the bandwidth of the cache cloud rented allocation strategy executed once. Rent amount until the next cycle before the next arrival of renting, adaptive bandwidth strategy according to formula (7) obtained in one period are adjusted cloud cache bandwidth.

Considering the system dynamic, the cloud bandwidth strategy can adaptively adjust the amount of leased cloud bandwidth and the accelerate efforts to process each file download. This strategy not only considers the influence of file polularity, but also provides the same service for different file download process.

5 The experimental results

Through simulation of the actual system operational data, analysis of the feasibility of bandwidth adaptive caching cloud strategy and cloud saving system bandwidth overhead effects. In order to save the system cloud bandwidth overhead, we calculate the theoretical maximum rate of P2P in the file retrieval process. Then the acceleration ability of P2P network to the file retrieval process is evaluated. We improve the node assisted acceleration as far as possible. After the P2P rate in offline download file retrieval process, the bandwidth strategy will decide whether to provide cloud acceleration according to the threshold rate. The sum of cloud acceleration rate provided to all file retrieve process aiming at one file is cache cloud bandwidth applied for this file. The cache bandwidth sum of cloud which is assigned to all being downloaded files is system rented cloud bandwidth in the next cycle.

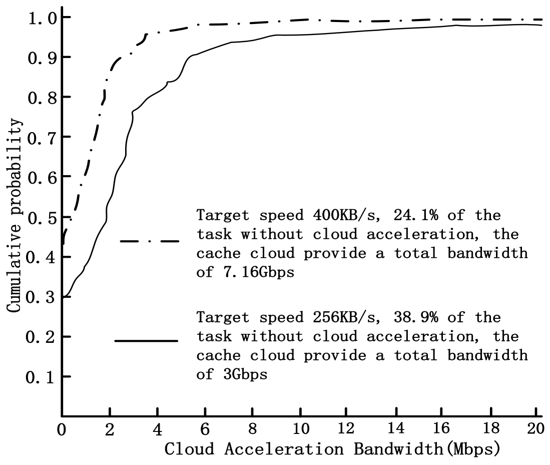


Figure 3: Cumulative probability distribution of the accelerate bandwidth that allocate each files by the cache cloud bandwidth strategy at idle time.

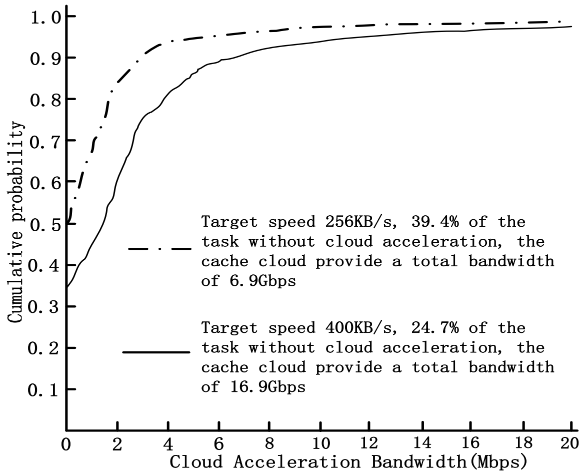


Figure 4: Cumulative probability distribution of the accelerate bandwidth that allocate each files by the cache cloud bandwidth strategy at peak hours.

In the idle period (6:00) and peak hours (20:00), after the implementation of the strategy of cloud bandwidth allocation, the distribution of the accumulated probability of the cache bandwidth provided to each file is as Figure 3 and 4. From Figure 3, when the threshold speed is set to 256KB / S, over 46% of 1,686 files being downloaded can't get accelerate bandwidth provided by the cache cloud. Most files which need cache cloud to provide acceleration bandwidth required bandwidth not high. When the threshold rate was 256KB/S, the gain-bandwidth cloud acceleration values greater than 4Mbps file only about 5 percent of the total file was being requested. When the threshold rate is 400 KB/S, the ratio of the value is only about 23%. In addition, even if the threshold value is set to be as high as 400 KB/S, as many as 30% of the file does not need to get acceleration bandwidth from the cache cloud. Finally, in the two different rates, only a very small amount of files need to get the acceleration bandwidth over 20 Mbps from the cache cloud.

In the hot time, the online node is the most and the ability to accelerate the process of file retrieval is more powerful, so the long tail of cache cloud acceleration bandwidth distribution is obvious. Figure 4 shows that the threshold rate is 256KB/S, over 50% of 2,742 files being downloaded do not need accelerate bandwidth provided by the cache cloud. When the threshold rate is 400 KB/S, the ratio is still more than 30%. The two values are higher than the corresponding value at 6:00. This indicates that the offline download file retrieval process may be obtained a greater intensity of node assisted acceleration from the P2P network. As the same as the idle period, most of the downloaded file does not need to consume too much bandwidth of cloud cache during the hot time. While the number of concurrent offline download files is even greater, there are still only a very small number of files to need a larger acceleration bandwidth by cache cloud.

Analysis of the results of the strategy implementation, the strategy has the ability to significantly relieve the pressure on the cache cloud bandwidth in the condition of ensuring the user download experience

Conclusion

With the development of cloud technology and cloud infrastructure improvement, the cloud platform provide computing, storage and bandwidth resources. Cloud assist service model hire a large number of cloud storage space to cache user appointment downloaded file. The system can guarantee download speed of the appointment documents through renting cloud bandwidth. The system can adjust the rental amount of cloud bandwidth every cycle. We propose a cloud bandwidth rental and allocation strategy. Based on the actual operational data, we simulate the adaptive cache cloud bandwidth strategy implementation process. The strategies maximize the synergistic effect on node auxiliary acceleration and cloud assist acceleration. This can guarantee the quality of service and save the bandwidth overhead of cloud.

References

[1] Johan Pouwelse, Pawel Garbacki,Dick Epema, Henk Sips. The Bittorrent P2P File-Sharing System [J]. Measurements and Analysis Lecture Notes in Computer Science, 2005， Vol.36 (40), pp: 205-216.

[2] [2] Yin H., Liu X., Zhan T., et al. Design and deployment of a hybrid CDN-P2P system for live video streaming experiences with Live Sky[C]. In Proceedings of ACM Multimedia (MM), 2009.

[3] Xu D,Kulkami S., Rosenberg C, Chai H.. Analysis of a CDN-P2P hybrid architecture for cost-effective streaming media distribution [J]. Multimedia Systems, 2006.

[4] Huang Y., Li Z., Liu G., et al. CloudDownload: Using Cloud Utilities to Achieve High-quality Content Distribution for Unpopular Videos[C]. Proceedings of ACM Multimedia (MM'11), 2011.

[5] Ao N.X., Xu Y.Y.,Chen C.J. et al. Offline Downloading: A Non-Traditional Cloud-Accelerated and Peer-Assisted Content Distribution Service[C]. Proceeding of 2012 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC， 12).

[6] Qiu X.J, Li H.X., Wu C. Cost-Minimizing Dynamic Migration of Content Distribution Services into Hybrid Clouds[C]. Proceedings of the 31st IEEE International conference on computers and communications (ICC'12), 2012.

[7] Li X, Yang Y. Trusted Data Acquisition Mechanism for Cloud Resource Scheduling Based on Distributed Agents [J]. China Communications, 2011, Vol.8 (6), pp: 108-116.

[8] Liu Y., Pan J. The impact of NAT on BitTorrent-like P2P systems[C]. Proceeding of 9th IEEE Peer-to-Peer Computing (P2P’09), 2009.

[9] Jin Li, Xinyi Huang, Jingwei Li, Xiaofeng Chen, Yang Xiang. Securely Outsourcing Attribute-based Encryption with Checkability. IEEE Transactions on Parallel and Distributed Systems, vol. 25(8), pp: 2201-2210, 2014.

[10] Jin Li, Xiaofeng Chen, Mingqiang Li, Jingwei Li, Patrick Lee, Wenjing Lou. Secure Deduplication with Efficient and Reliable Convergent Key Management. IEEE Transactions on Parallel and Distributed Systems, vol. 25(6), pp: 1615-1625, 2014.

A Churn Resilience Technique on P2P Sensor Data Stream Delivery System Using Distributed Hashing

Tomoya Kawakami
Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma, Nara, Japan
E-mail: kawakami@is.naist.jp

Tomoya Kawakami, Yoshimasa Ishi, Tomoki Yoshihisa and Yuuichi Teranishi
Cybermedia Center, Osaka University, Ibaraki, Osaka, Japan
E-mail: ishi.yoshimasa@ais.cmc.osaka-u.ac.jp, yoshihisa@cmc.osaka-u.ac.jp

Yuuichi Teranishi
National Institute of Information and Communications Technology, Koganei, Tokyo, Japan
E-mail: teranisi@cmc.osaka-u.ac.jp

Keywords: sensor data, data stream, delivery cycle, distributed processing, churn resilience

Received: July 16, 2015

Recently, sensor data stream delivery system that collects sensor data periodically and delivers successively has been attracting great attention. As for this sensor data stream delivery, receivers are possible to require the same sensor data stream with different delivery cycles. Our research team proposed methods to distribute communication loads by relay nodes in the case of delivering the sensor data streams that have different data delivery cycles. However, in the previous methods, since the specific node builds delivery paths and notifies related nodes, the assigned node is required to be updated when the related nodes churn. Therefore, in this paper, we propose a churn resilience technique that enhances the robustness of delivery system. We confirmed in simulations that the proposed technique improves the reliability of the delivery system.

Povzetek: Razvita in stestirana je nova metoda za asinhrono pošiljanje in zbiranje toka podatkov.

1 Introduction

In recent years, various types of applications such as video delivery and environmental monitoring have been possible, and therefore, sensor data stream delivery where sensor data is periodically collected and delivered successively has been attracting great attention. As for this sensor data stream delivery, it is possible for the same sensor data stream to have different collection periods depending on the receivers. In the case where a live video of a solar eclipse taken from a camera is delivered, for example, the video is delivered at 30 fps to personal computers connected to the Internet through a wire and is delivered at 10 fps to mobile computers connected to the Internet through a 3G channel while moving.

It is general in sensor data stream delivery that sensor data gained by one sensor is shared by a large number of users. Currently, various P2P-based techniques for dispersing the communication load of the deliverer (source) have been proposed in the data streaming [1–10]. In these researches, the same sensor data stream is delivered to a number of terminals (destinations), the communication load of the source is dispersed by sending the received data to other destinations. When the delivery cycle is different, the sensor data stream whose delivery cycle is a common divisor of required cycles can be delivered to all of the destinations if the delivery cycles are in a multiple relationship or can be approximated as having a multiple relationship. However, the destinations receive redundant data which are not included to the times of each required cycle.

We have proposed techniques for sensor data stream delivery system in P2P model [11, 12]. In our previous techniques, destinations having a long delivery cycle transmit the sensor data stream to other destinations so that the load of the source is dispersed. However, in the previous methods, since the specific node builds delivery paths and notifies related nodes, the assigned node is required to be updated when the related nodes churn.

Therefore, in this paper, we propose a churn resilience technique that enhances the robustness of delivery system. The proposed technique uses a successor list used in Chord [13]. We confirmed in simulations that the proposed technique improves the reliability of the delivery system.

2 Addressed problems

2.1 Assumptions

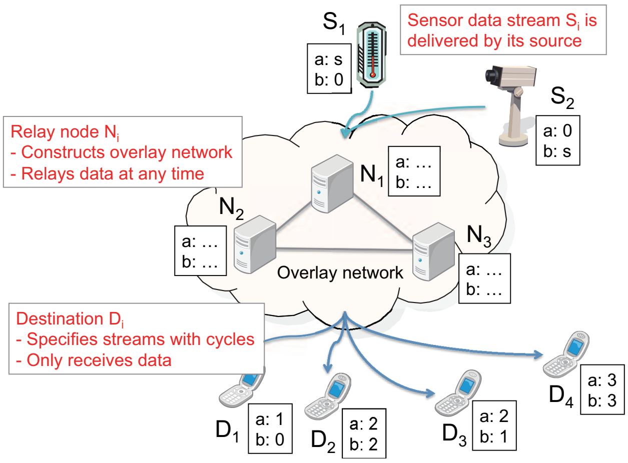
We assume that computers (nodes) to relay sensor data streams constructs P2P overlay network in the sensor data stream delivery system. The sensor data stream delivery system distributes the delivery loads to the nodes and keeps high scalability in an environment where there are a huge number of sensor data streams and destinations. Sensor data streams are periodically sent from their sources through the Internet and delivered to destinations by the hops among nodes. Destinations request sensor data streams with those delivery cycles to a specific node also through the Internet. We assume that selectable delivery cycles for each sensor data stream are given. Nodes are able to send sensor data to other nodes anytime, and sensor data are distributed for each sensor data stream and time.

The sensor data streams are denoted by S_i ($i = 1, \cdots, l$), destinations are denoted by D_i ($i = 1, \cdots, m$), and nodes are N_i ($i = 1, \cdots, n$). Figure 1 shows a model of delivery system. In Figure 1, the number of sensor data streams is $l = 2$, the number of destinations is $m = 4$ and the number of nodes is $n = 3$. The 'a' represents the sensor data stream S_1, and the 'b' represents the sensor data stream S_2. The delivery cycles are shown near sources, nodes and destinations in Figure 1. The 's' represents the source of sensor data stream, and the numbers near destinations are requested delivery cycles from each destination. In the case where the delivery cycle is 0, it means that the destination does not request the sensor data stream. This corresponds to case where a live camera acquires an image once every second, and D_1 does not view the image, D_2 and D_3 view the image once every second, and D_4 views the image once every three seconds, for example. In this paper, we assume that selectable delivery cycles for each sensor data stream are given and are represented by C_i ($i = 1, 2, \cdots$). The sensor data delivery system assigns delivery cycles or times to relay sensor data streams to nodes. The nodes send and receive various sensor data each other on specific times.

2.2 Objective function

In sensor data stream delivery, it is important to avoid concentrating of processes and loads to a specific computer or network because the processing time affects and accumulates a delay of delivery. Therefore, in this paper, we aim to distribute communication loads to relay nodes on sensor data stream delivery system.

The communication load of the node N_i is given as the total of the load due to the reception of the sensor data stream and the load due to the transmission. The communication load due to the reception is referred to as a reception load, and the reception load of N_i is I_i. The communication load due to the transmission is referred to as transmission load, and the transmission load of N_i is O_i.



Sensor data stream S_i is delivered by its source

Relay node N_i
- Constructs overlay network
- Relays data at any time

Destination D_i
- Specifies streams with cycles
- Only receives data

Figure 1: System model.

In many cases, the reception load and the transmission load are proportional to the number of sensor data pieces per unit hour of the sensor data stream to be sent and received. The number of pieces of sensor data per unit hour of the sensor data stream that is to be received by N_i from N_j or S_k ($i \neq j$; $i, j = 1, \cdots, n$; $k = 1, \cdots, l$) is $R(N_j, N_i)$ or $R(S_k, N_i)$. In addition, the number of pieces of sensor data per unit hour of the sensor data stream that is to be sent by N_i to N_j or D_k ($i \neq j$; $i, j = 1, \cdots, n$; $k = 1, \cdots, m$) is $R(N_i, N_j)$ or $R(N_i, D_k)$. The loads L_i, I_i and O_i of N_i are given in the following equations:

$$L_i = I_i + O_i \tag{1}$$

$$I_i = \alpha \sum_{j=1}^{n} R(N_j, N_i) + \alpha \sum_{k=1}^{l} R(S_k, N_i) \tag{2}$$

$$O_i = \beta \sum_{j=1}^{n} R(N_i, N_j) + \beta \sum_{k=1}^{m} R(N_i, D_k) \tag{3}$$

where α and β are loads with which one piece of sensor data is received and sent, respectively.

The communication load SL of the entirety of the system is given by the following equation:

$$SL = \sum_{i=1}^{n} L_i \tag{4}$$

In addition, the following fairness index (FI) is often used as an index for load dispersion:

$$FI = \frac{\left(\sum_{i=1}^{n} L_i\right)^2}{n \sum_{i=1}^{n} L_i^2} \tag{5}$$

where $0 \leq FI \leq 1$, and when $FI = 1$, $L_0 = \cdots = L_n$. It is indicated that the closer FI is to 1, the more the load is dispersed. Another purpose of this study is to disperse the communication load to the destination nodes while suppressing the communication load of the entirety of the system. Therefore, the objective function is SL and $1 - FI$, and the delivery path is determined to make these values minimum.

3 Robustness enhancement technique

3.1 Previous methods

We have proposed techniques for a P2P model where such an environment that a number of destinations collect the sensor data stream during different cycle is assumed so that the load of the source is dispersed [11]. In addition, we have proposed a method which determines relay nodes based on distributed hashing and constructs delivery paths autonomously by each node [12].

The previous method using distributed hashing devides nodes into groups represented by the combination of sensor data stream and delivery cycle. In addition, the previous method assigns delivery times to nodes for each group of delivery cycle. The previous method distributes processes by assigning a node based on delivery time. In addition, the previous method avoids concentrating loads to a specific node and time by assigning a node for each group of delivery cycle. However, in the previous methods, since the specific node builds delivery paths and notifies to related nodes, the assigned node is required to be updated when the related nodes churn.

3.2 Redundancy of node assignment by successor list

In the sensor data stream delivery, the number of data to send/receive varies among different delivery cycles. The shorter the delivery cycle is, the larger the number of data and the load are. Therefore, the previous method using distributed hashing first generates circular hash spaces for each sensor data stream and puts nodes on hash spaces based on the distributed hashing of the combination of sensor data stream and node ID. After that, the previous method divides each hash space into partial hash spaces as groups for each delivery cycle. By this process, partial hash spaces of the shorter cycle have the more nodes. The size of each partial hash space is determined based on its cycle. For example, in the case where the selectable delivery cycles are $C_i = i$ $(i = 1, 2, 3)$, the ratio of the sizes of partial hash spaces is $1/C_1 : 1/C_2 : 1/C_3 = 1/1 : 1/2 : 1/3 = 6 : 3 : 2$. The previous method treats each partial hash space as circular and assigns related times for each cycle to nodes on its partial hash space. In the case where there are no nodes on the partial hash space, the previous method assigns the partial hash space to the nearest neighbor node on the next partial hash space. In addition, the previous method determines the root node on the partial hash space of the shortest cycle based on distributed hashing such as the least common multiple of cycles. The root node first receives data from the source of sensor data stream.

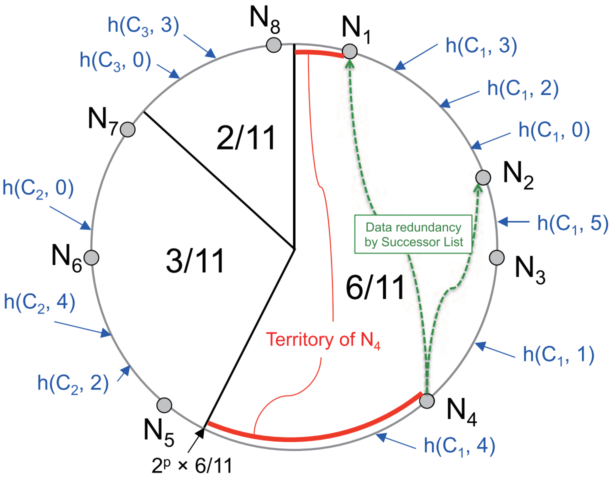


Figure 2: Assignment to a group of cycle.

```
Require:
    $cycles$: Arrangement of delivery cycles of the nodes sorted in ascending order
    (cycle of source node is $-1$ and at index 0)
    $ownId$: An identification of own (node)
    $assignedCycleIndex$: An index of own assigned delivery cycle in $cycles$
    $succCount$: The length of a successor list

$cycleLcm \leftarrow calculateLCM(cycles)$;
if $assignedCycleIndex \neq 0$
or $searchNode(0, cycleLcm, 0) \neq ownId$ then
    $time \leftarrow 0$;
    while $time < cycleLcm$ do
        $assignedNode$
        $\leftarrow searchNode(assignedCycleIndex, time, 0)$;
        if $assignedNode = ownId$ then
            $longCycleIndex$
            $\leftarrow calculateLongestCycleIndex(cycles, time, 0)$;
            $relayNode$;
            if $longCycleIndex = assignedCycleIndex$ then
                $relayNode \leftarrow searchNode(0, cycleLcm, 0)$;
                $succList$;
                $succNode \leftarrow ownId$;
                for $i \leftarrow 0$ to $succCount$ do
                    $succNode \leftarrow successor(succNode)$;
                    $succList.add(succNode)$;
                end for
            else
                $succNodeIndex$
                $\leftarrow random(0, succCount + 1)$;
                $relayNode \leftarrow searchNode($
                $longCycleIndex, time, succNodeIndex)$;
            end if
            $requestToSend(relayNode, ownId, time)$;
        end if
        $time \leftarrow time + cycles[assignedCycleIndex]$;
    end while
end if
```

Figure 3: Pseudocode to construct delivery paths by nodes.

In this paper, we propose a churn resilience technique that enhances the robustness of delivery system by a successor list used in Chord [13]. Figure 2 shows an example of the case where the number of nodes is $n = 8$, cycles are $C_i = i$ $(i = 1, 2, 3)$, the size of a hash space is 2^p, and the length of the successor list is 2. In Figure 2, the beginning values of each partial hash space are $2^p \times 0/11$, $2^p \times 6/11$, and $2^p \times 9/11$.

To construct delivery paths, nodes first calculate the least common multiple of selectable delivery cycles for each sensor data stream. After that, the nodes search the sender nodes for each related time that the same time is assigned to in the other cycle groups. The nodes determine the cycle groups to search for each time based on the approach such as the LCF method [11], and the node that belongs

```
Require:
    $cycles$: Arrangement of delivery cycles of the nodes sorted in ascending order
    (cycle of source node is $-1$ and at index 0)
    $ownId$: An identification of own (destination)
    $requestCycleIndex$: An index of request delivery cycle in $cycles$
    $succCount$: The length of a successor list
1: $cycleLcm \leftarrow calculateLCM(cycles)$;
2: $time \leftarrow 0$;
3: while $time < cycleLcm$ do
4:     $targetCycleIndex$
       $\leftarrow getRandomCycleIndex(cycles, time)$;
5:     $succNodeIndex$
       $\leftarrow random(0, succCount + 1)$;
6:     $relayNode \leftarrow searchNode($
       $targetCycleIndex, time, succNodeIndex)$;
7:     $requestToSend(relayNode, ownId, time)$;
8:     $time \leftarrow time + cycles[requestCycleIndex]$;
9: end while
```

Figure 4: Pseudocode to construct delivery paths by destinations.

to the longest cycle on each time receives data from root node and sends to the nodes that belong to the other cycle groups. The node that does not belong to the longest cycle group on each time searches the node on the longest cycle group and requests to send data. On the other hand, the node that belongs to the longest cycle group on each time searches the root node of the sensor data stream and requests to send data. Figure 3 shows the pseudocode to construct delivery paths by a node.

In the the pseudocode on the Figure 3, the least common multiple of the selectable delivery cycles is calculated in the line 1. The case shown in the line 2 is a case where this node does not belong to the shortest cycle group or is not the root node. The case shown in the line 6 is a case where *time* in the cycle group is assigned to this node, and the case shown in the line 9 is a case where the cycle group of this node is the longest cycle at *time*. In the line 10, the root node is searched as a sender to this node at *time*. Successor node is searched in the line 14, and the searched node is added to the successor list in the line 15. Successor node that receives data is selected at random in the line 18, and the node that belongs to the longest cycle group and *succNodeIndex* is searched as a sender to this node at *time* in the line 19. Finally, delivery path from the relay node at *time* is constructed in the line 21.

Similarly destinations first calculate the least common multiple of selectable delivery cycles for each sensor data stream. After that, the destinations determine the cycle group for each time at random among the related cycles. The destinations search the senders in the determined cycle groups for each time and request to send data. Figure 4 shows the pseudocode to construct delivery paths by a destination.

In the the pseudocode on the Figure 4, the least common multiple of the selectable delivery cycles is calculated in the line 1. The cycle group able to send at *time* and successor node that receives data are selected at random in the line 5. After that, the node that belongs to the selected cycle group and *succNodeIndex* is searched as a sender to this destination at *time* in the line 6. Finally, delivery path from the relay node at *time* is constructed in the line 7.

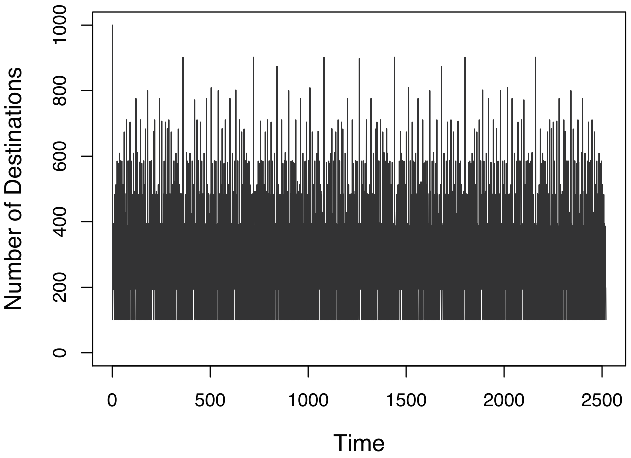


Figure 5: The number of desitinations for the longest cycle group at each time.

4 Evaluation

4.1 Simulation environment

In this section, we evaluate the proposed technique using distributed hashing in Section 3 by simulation. In the simulation environment, the number of nodes is $n = 2^7 = 128$, the number of sensor data streams is $l = 2^7 = 128$, and the number of destinations is $m = 1000$. The delivery cycles that destinations request are $C_i = i\ (i = 1, \cdots, 10)$ and determined at random from 1 to 10 for each sensor data stream. In this environment, the maximum of the least common multiple of delivery cycles is 2520, and then the timetable for delivery is from time 0 to time 2519.

As an evaluated value, we calculated the load of each node, system total loads (*SL*), and fairness index (*FI*) among the time of the least common multiple of selectable delivery cycles. We executed this simulation 10 times for each method and environments described later. We calculated the average of the results.

4.2 Number of influenced destinations

Figure 5 shows the number of destinations that cannot receive the data stream in the case where the assigned node of the longest cycle group on each time churns.

In the proposed technique, the assigned node of the longest cycle group on each time relays to the assigned nodes of the other cycle groups on that time. Since a specific node of the longest cycle group is assigned each time, nodes of the other cycle groups and their destinations cannot receive the data stream in the case where the assigned node of the longest cycle group churns. In Figure 5, the assigned nodes of all cycle groups deliver to their destinations at time 0. The longest cycle group at time 0 is 10. Therefore, all of the one thousand destinations cannot receive the

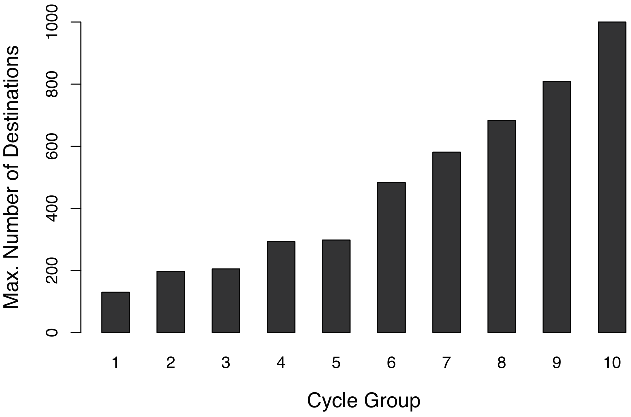


Figure 6: Maximum instantaneous number of destinations for each cycle group.

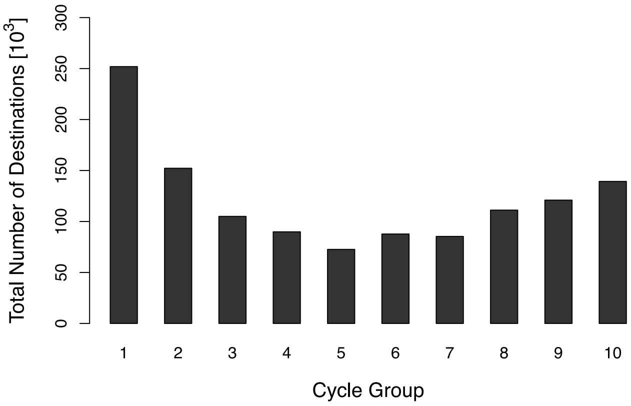


Figure 7: Total number of destinations for each cycle group.

data stream in the case where the assigned node of the cycle group 10 at time 0 churns. Similarly, the longest cycle group and the number of destinations are different among times, and the number of influenced destinations changes as shown in Figure 5. However, the successor list in our proposal makes this influence of node changes reduced.

Figure 6 shows the maximum instantaneous number of destinations for each cycle group and time. In addition, Figure 7 shows the total number of destinations for each cycle group.

In our assumptions, the longer cycle groups such as over 6 have the higher probability to become the longest cycle for each time. Therefore, the longer cycle groups have the larger number of influenced destinations per unit time shown in Figure 6. On the other hand, the shorter cycle groups such as 1 deliver to destinations for more times. Therefore, the shorter cycle groups have the larger number of the total influenced destinations between time 0 to 2519 as shown in the Figure 7.

4.3 Results by the length of successor list

Figure 8, Figure 9 and Figure 10 show the minimum of the instantaneous system reliability in the proposed tech-

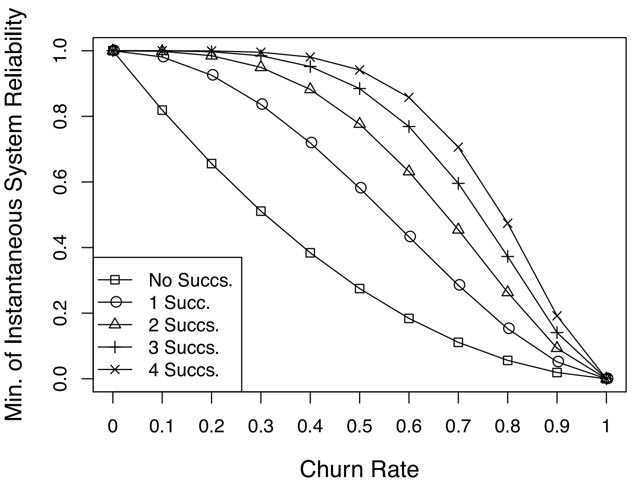


Figure 8: The minimum of instantaneous system reliability in the constant scenario.

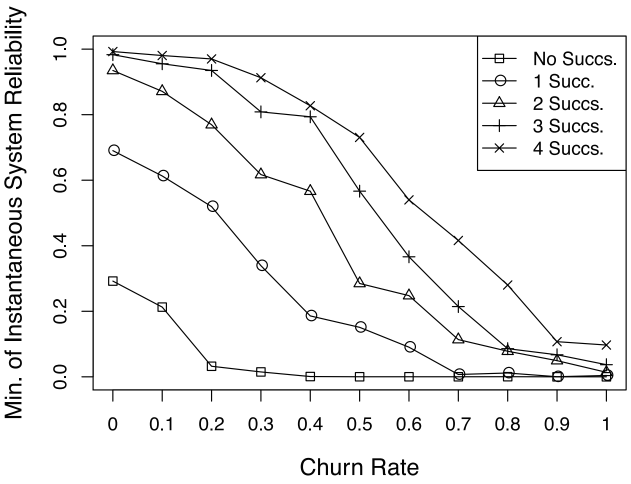


Figure 9: The minimum of instantaneous system reliability in the Gaussian scenario.

nique and an environment where the number of successors is $0, \cdots, 4$. The instantaneous system reliability shows the rate of destinations each time that receive data successfully. Figure 8 is the result in the case where the churn rate of nodes is constant to the value on the lateral axis. Figure 9 is the result in the case where the churn rate of nodes is individually determined based on the Gaussian distribution with the mean being the value on the lateral axis and the dispersion being 1. Figure 10 is the result in the case where the churn rate of nodes is individually determined at random from 0 to 1. The longitudinal axis is the minimum of the instantaneous system reliability.

Figure 11, Figure 12 and Figure 13 show the average of the instantaneous system reliability in the proposed technique and same environment. The longitudinal axis is the average of the instantaneous system reliability.

In these results, the instantaneous system reliability basically becomes higher by the number of successors. The increment from the case that has no successors is especially

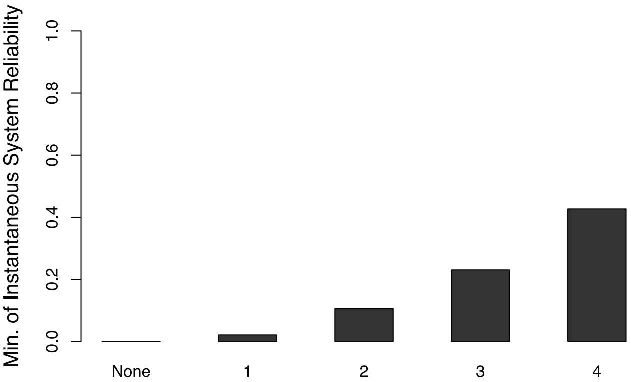


Figure 10: The minimum of instantaneous system reliability in the random scenario.

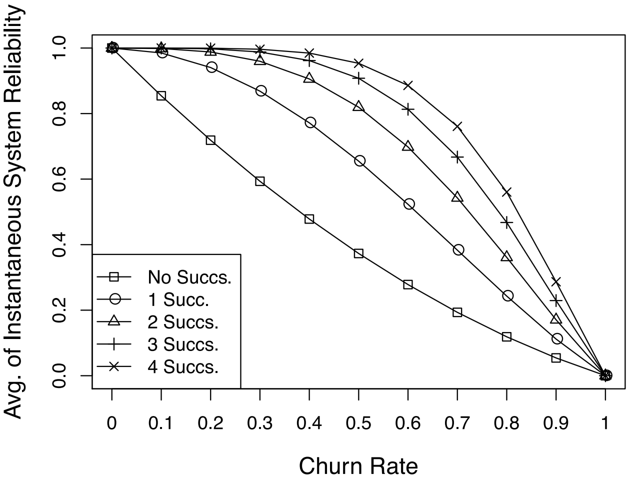


Figure 11: The average of instantaneous system reliability in the constant scenario.

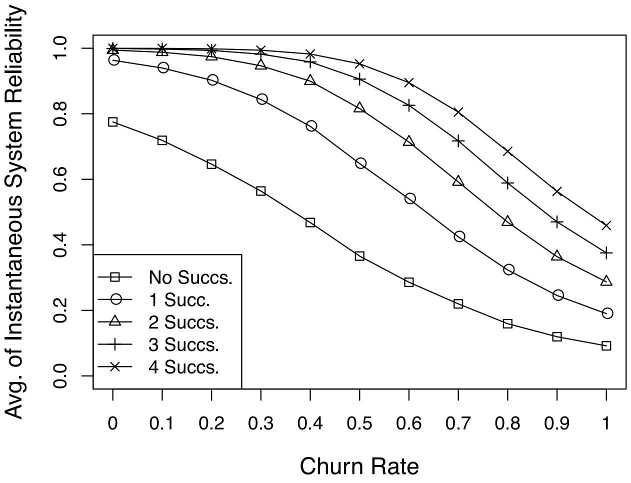


Figure 12: The average of instantaneous system reliability in the Gaussian scenario.

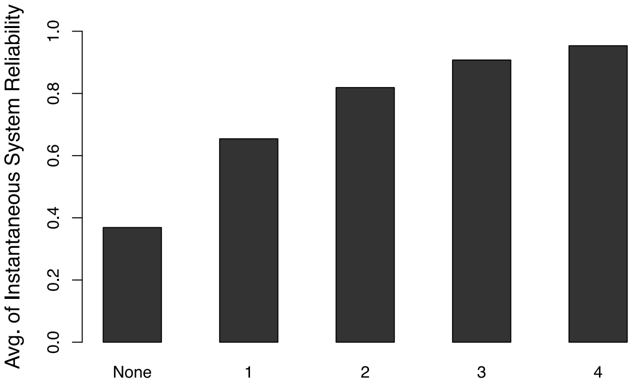


Figure 13: The average of instantaneous system reliability in the random scenario.

large even if there are a few successors. Therefore, the successor list is effective to enhance the robustness and reliability of the delivery system.

Figure 14 shows the maximum instantaneous load in an environment where the number of successors is $0, \cdots, 4$. The maximum instantaneous load is the maximum load for each node and time. The longitudinal axis is the maximum instantaneous load, and the lateral axis is the number of successors. In Figure 14, the difference of the maximum instantaneous load is small in this simulation environment. In the same environment, Figure 15 shows the maximum load of node, Figure 16 shows SL, and Figure 17 shows FI. Also the differences in those results are small in this simulation environment. Therefore, the maintenance cost of the successors is not influenced largely in the case where the number of destinations is especially higer than the number nodes.

Figure 18 shows the rate of the number of hops to each node at time 0 in an environment where the number of successors is $0, \cdots, 4$. The lateral axis is the number of hops, and the longitudinal axis is the rate of the nodes that receive data under the number of hops. The nodes shown as "N/A" do not receive data at time 0 because other nodes in the same cycle group are assigned to time 0. In Figure 18, the nodes shown as "N/A" are reduced in the longer successor list because the nodes that receive data at time 0 are increased. Although the rate of the number of the higher hops are increased in the longer successor list, all of the nodes receive data under four hops.

Figure 19 shows the rate of the number of hops to each destination at time 0 in the same environment. The longitudinal axis is the rate of the destinations that receive data under the number of hops. Although the rate of the number of the higher hops are increased in the longer successor list, all of the nodes receive data within five hops.

5 Related work

Various techniques for dispersing the communication load in the delivery of streams have been proposed [14].

A P2P stream delivery technique using a P2P technology for sending and receiving data has been proposed in order to disperse the communication load among the terminals [1–5]. The P2P stream delivery technique is di-

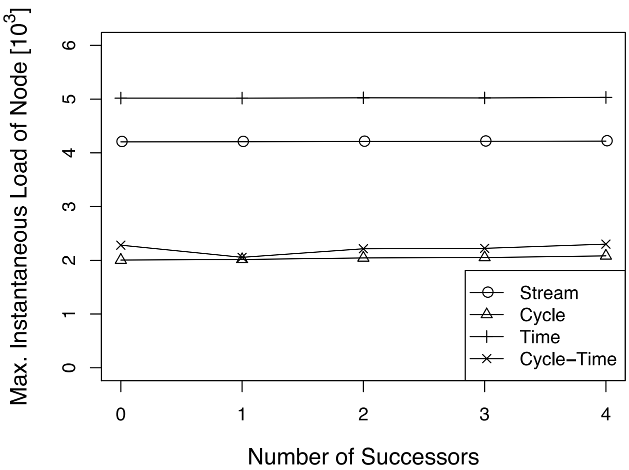


Figure 14: Maximum instantaneous load by the number of successors.

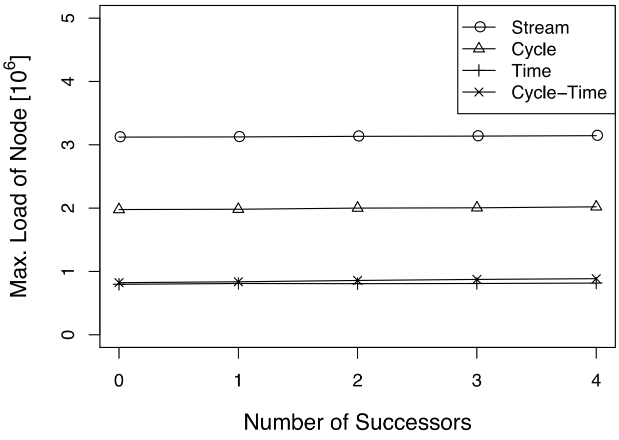


Figure 15: Maximum load by the number of successors.

vided into a pull type technique and a push type technique. In a pull type technique, such as PPLive[1], DONet [1], and SopCast[2], the reception terminal that receives data requests data from another terminal and acquires it. Although the reception terminal find terminals that have not yet been received the requested data, no redundant communications are carried out. In a push type technique, such as AnySee, data is sent from the transmission terminal that sends data to another terminal [2]. Although the transmission terminal find terminals that have not yet received the requested data, no such redundant communications are carried out. A technique combining a pull type and a push type, such as PRIME, has been proposed [3].

In P2P stream delivery techniques, a case where the same data stream is delivered to a number of terminals is assumed. In the delivery of the sensor data streams, however, there are cases where a data stream of the same sensor having different delivery cycles is delivered. In this case,

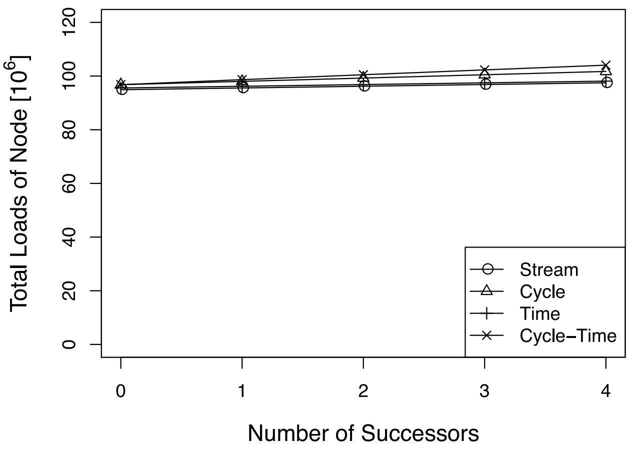


Figure 16: Total sytem loads by the number of successors.

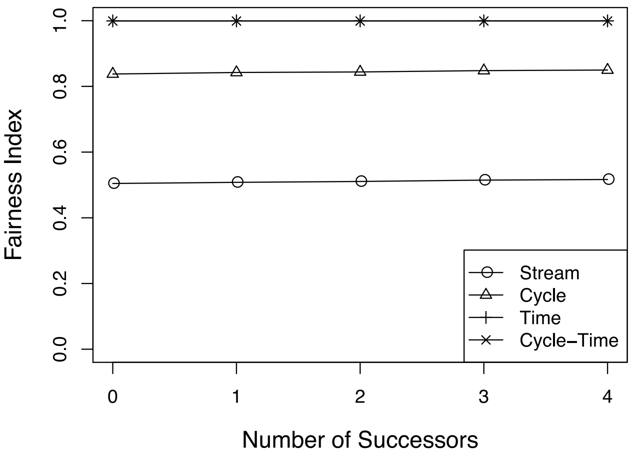


Figure 17: Load balance by the number of successors.

sensor data streams having different delivery cycles are delivered as different data streams.

Several techniques for preventing the communication load from being concentrated on a particular terminal by constructing a data delivery path, which is referred to as a multicast tree, in advance so that a data stream is delivered have been proposed [6–10]. In the ZIGZAG method, a multicast tree is constructed by clusters that are collections of terminals [7]. The number of clusters included in each depth of a multicast tree is made the same, and thus, the load is dispersed. Multicast trees are constructed only of information gained in the application layer, and it is not necessary to understand the physical network structure.

In the MSMT/MBST method, the communication load can be prevented from concentrating on a particular terminal as compared to ZIGZAG by taking the communication delay between terminals into consideration in the case where the physical network structure can be understood [8]. The implementability of the MSMT/MBST method was poor because it is necessary to understand all the network structures between the terminals concerning stream delivery. In LAC (locality aware clustering), a load dispersion higher than that in ZIGZAG is achieved by taking into con-

[1] `http://www.pplive.com/`

[2] `http://www.sopcast.com/`

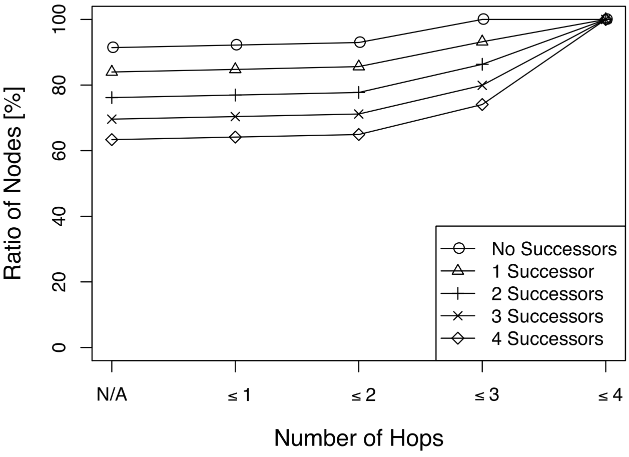


Figure 18: Rate of nodes by the number of hops to delivery.

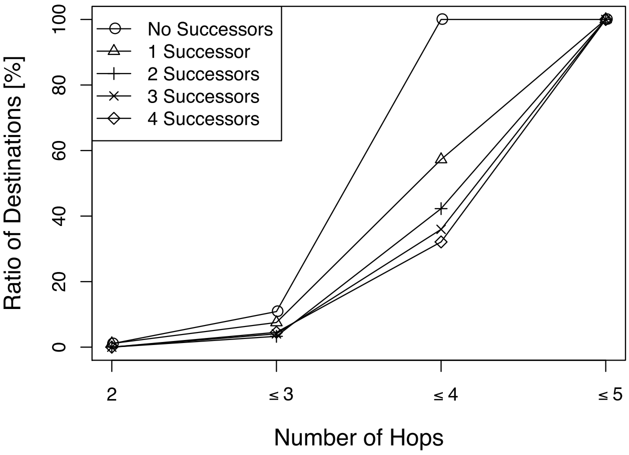


Figure 19: Rate of destinations by the number of hops to delivery.

sideration the communication delay between only some terminals, though the physical network structure cannot be understood [9].

P2P sensor data stream delivery systems must be robust and resilient against node churn because P2P networks have high flexibility and always variable. Currently, techniques related to the churn resilience have been proposed [15–17], and also an analytical framework that allows to model retrieval times has been proposed [18]. In this paper, we applied a successor list to P2P sensor data stream delivery system to accommodate heterogeneous cycles, and we can also apply these related works, e.g., a logic layer named Dechurn that uses the complementary nature of node joining and leaving [16].

6 Conclusion

In this paper, we proposed a churn resilience technique that enhances the robustness of delivery system by a successor list in Chord. Validated through evaluation, the reliability of the delivery system is improved.

In the future, we will study an algorithm to calculate the appropriate length of the successor list each time because the required system reliability changes by applications, situations, and so on.

Acknowledgement

This research was partly supported by the collaborative research of National Institute of Information and Communications Technology (NICT) and Osaka University (Research on high functional network platform technology for large-scale distributed computing). This research was partly supported by the Strategic Information and Communications R&D Promotion Programme (SCOPE) of the Ministry of Internal Affairs and Communications.

References

[1] X. Zhang, J. Liu, B. Li, and T.-S. P. Yum, “CoolStreaming/DONet: A data-driven overlay network for peer-to-peer live media streaming,” in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005)*, Mar. 2005, pp. 2102–2111.

[2] X. Liao, H. Jin, Y. Liu, L. M. Ni, and D. Deng, “Anysee: Peer-to-peer live streaming,” in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM 2006)*, Apr. 2006, pp. 1–10.

[3] N. Magharei and R. Rejaie, “PRIME: Peer-to-peer receiver-driven mesh-based streaming,” in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM 2007)*, May 2007, pp. 1415–1423.

[4] L. Yu, X. Liao, H. Jin, and W. Jiang, “Integrated buffering schemes for P2P VoD services,” *Peer-to-Peer Networking and Applications*, vol. 4, no. 1, pp. 63–74, 2011.

[5] S. Sakashita, T. Yoshihisa, T. Hara, and S. Nishio, “A data reception method to reduce interruption time in P2P streaming environments,” in *Proceedings of the 13th International Conference on Network-Based Information Systems (NBiS)*, Sep. 2010, pp. 166–172.

[6] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, “Scalable application layer multicast,” in *Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM 2002)*, Aug. 2002, pp. 205–217.

[7] D. A. Tran, K. A. Hua, and T. Do, “ZIGZAG: An efficient peer-to-peer scheme for media streaming,” in *Proceedings of the 22nd Annual Joint Conference*

of the IEEE Computer and Communications Societies (INFOCOM 2003), vol. 2, Mar. 2003, pp. 1283–1292.

[8] X. Jin, W.-P. K. Yiu, S.-H. G. Chan, and Y. Wang, "On maximizing tree bandwidth for topology-aware peer-to-peer streaming," *IEEE Transactions on Multimedia*, vol. 9, no. 8, pp. 1580–1592, Dec. 2007.

[9] K. Silawarawet and N. Nupairoj, "Locality-aware clustering application level multicast for live streaming services on the Internet," *Journal of Information Science and Engineering*, vol. 27, no. 1, pp. 319–336, 2011.

[10] T. A. Le and H. Nguyen, "Application-aware cost function and its performance evaluation over scalable video conferencing services on heterogeneous networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference: Mobile and Wireless Networks (WCNC 2012 Track 3 Mobile and Wireless)*, Apr. 2012, pp. 2185–2190.

[11] T. Kawakami, Y. Ishi, T. Yoshihisa, and Y. Teranishi, "A P2P-based sensor data stream delivery method to accommodate heterogeneous cycles," *Journal of Information Processing (JIP)*, vol. 22, no. 3, pp. 455–463, Jul. 2014.

[12] ——, "A load distribution method based on distributed hashing for P2P sensor data stream delivery system," in *Proceedings of the 3rd IEEE International Workshop on Modeling and Verifying of Distributed Applications (MVDA 2014) in Conjunction with the 38th Annual International Computer, Software and Applications Conference (COMPSAC 2014)*, Jul. 2014, pp. 716–721.

[13] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup protocol for internet applications," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 17–32, Feb. 2003.

[14] Z. Shen, J. Luo, R. Zimmermann, and A. V. Vasilakos, "Peer-to-peer media streaming: Insights and new developments," *Proceedings of the IEEE*, vol. 99, no. 12, pp. 2089–2109, Oct. 2011.

[15] S. Legtchenko, S. Monnet, P. Sens, and G. Muller, "RelaxDHT: A churn-resilient replication strategy for peer-to-peer distributed hash-tables," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 7, no. 2, Article 28, Jul. 2012.

[16] X. Meng, X. Chen, and Y. Ding, "Using the complementary nature of node joining and leaving to handle churn problem in P2P networks," *Computers and Electrical Engineering*, vol. 39, no. 2, pp. 326–337, Feb. 2013.

[17] C. Hu, M. Chen, C. Xing, and G. Zhang, "Exploring the optimal substream scheduling and distribution mechanism for data-driven P2P media streaming," vol. 44, pp. 14–25, May 2014.

[18] L. Pamies-Juarez, M. Sanchez-Artigas, P. García-López, R. Mondéjar, and R. Chaabouni, "On the interplay between data redundancy and retrieval times in P2P storage systems," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 59, pp. 1–16, Feb. 2014.

An Experimental Approach to Examine a Multi-Channel Multi-Hop Wireless Backbone Network

Yuzo Taenaka[1], Masaki Tagawa[1,2] and Kazuya Tsukamoto[3]
E-mail: taenaka@nc.u-tokyo.ac.jp
[1] Information Technology Center, The University of Tokyo, Japan
[2] Graduate School of Information Science, Nara Institute of Science and Technology, Japan
[3] Department of Computer Science and Electronics, Kyushu Institute of Technology, Japan

Keywords: traffic management, channel utilization, experiment, multiple channels, OpenFlow, wireless backbone network, multi-hop network, wireless LAN

Received: July 16, 2015

This paper presents an experimental deployment of a multi-channel multi-hop wireless backbone network (WBN) with an OpenFlow-based traffic management method. Specifically, a set of APs, each of which uses a single but different channel, is connected by Ethernet and thus constructs a Virtual AP (VAP), thereby achieving a WBN with multiple channels. To flexibly control traffic flows transmitted over a multi-channel multi-hop WBN, we propose a simple traffic management method based on the OpenFlow control. In the performance evaluation, we first conduct a preliminary experiment as a lab scale and then deploy a 6-hop WBN enabling to provide the Internet access service in a conference (from proof-of-concept to a practical environment). Since the control messages are inherently transmitted with the introduction of OpenFlow, the way of isolation between control plane and data plane will become a critical issue to actually deploy the proposed system for the Internet service. We additionally employ a wireless control network for the conference experiment. The experimental results show that the proposed WBN can increase the network capacity in accordance with the number of channels, thereby providing significant throughput performance for various applications.

Povzetek: Predstavljena je eksperimentalna analiza več-kanalnega več-skokovnega brezžičnega ogrodja za mreže.

1 Introduction

The use of various mobile devices such as smartphones and tablets is rapidly widespread and is becoming increasingly essential to our daily life. A large amount of data is exchanged for various purposes such as video communication and streaming, and thus it is expected to increase by about 11 times between 2013 and 2018 [1].

Since this growth of the mobile traffic is faster than the increase of the network capacity in the advanced cellular technology such as LTE/4G, cellular carriers are rapidly deploying public WLAN access points (APs) to offload the mobile traffic. However, each WLAN coverage is relatively small so that APs tend to be densely placed within specific locations such as shops and a part of public space, thereby suffering radio interference of APs due to highly overlapped coverage. From these considerations, the way of extending WLAN coverage is essential.

To achieve this, a wireless mesh network (WMN) that extends WLAN coverage attracts much attention. A WMN mainly consists of two sorts of APs: gateway AP (called Internet gateway (IGW)) providing the Internet reachability to other APs and other APs constructing a multi-hop wireless backbone network (WBN) to reach the IGW (i.e., the Internet). Since a WMN can extend its coverage due to the ease of WBN extension, WMN has been already deployed in relative wide area (e.g., a shopping mall and a city).

However, the existing WMN always suffers a limited network capacity due to the nature of multi-hop transmission on multi-hop network. To offload the mobile traffic, the network capacity of a WBN has to be increased so that it can accommodate the amount of the increasing traffic reliably. In particular, since the mobile traffic basically passes through the IGW (i.e., from/to the Internet), the network capacity on a single route toward the IGW should be increased.

Since the channel capacity is physically limited and an existing WBN has been generally designed to consistently use the same channel even for multi-hop transmission, the network capacity is drastically decreased by the competition of channel access and radio interference of nearby APs along with the increase of traffic and the number of hops. Therefore, the effective use of multiple channels on a WBN is absolutely necessary to expand the network capacity. In existing researches, routing protocols, channel assignments, and MAC protocols handling multiple channels are mainly studied [2–6]. These studies cannot simultaneously use multiple paths (channels) between neighboring two APs due to the limitation of the conventional routing schemes. Also, since an existing AP hardware only has a few wireless interfaces (IFs) (two IFs generally), the number of channels that the AP can use simultaneously is less than the number of IFs that the AP equips with even when many channels (more than the number of IFs) are vacant.

To physically increase the network capacity, IEEE 802.11n/ac have a function of the channel bonding [7, 8]. In the channel bonding, consecutive channels have to be vacant. However, it is quite difficult to monopolize the use of all vacant channels because there are multiple but not consecutive vacant channels scattered in the 2.4 GHz and 5 GHz bands in a real environment. In preliminary experiments, we observed that there are many vacant channels in the 5 GHz band, which means that the available resources cannot be used effectively. Thus, the flexible way of integrating the scattered channels becomes crucial.

From the distinctive points, two problems should be addressed to increase the network capacity of a WBN. One is the limitation on the number of channels that APs can use simultaneously. Another is under-utilization of the channels in use. To address these problems, we first propose an efficient framework, which is potentially capable of handling the unlimited number of channels in parallel. A multi-channel management framework by exploiting OpenFlow is also presented. To the best of our knowledge, there is no literature that not only focuses on these problems but also applies the OpenFlow based approach to the WBN. We then implement these frameworks in a real AP hardware/software and experimentally deploy two sorts of WBNs (a lab scale proof-of-concept and a 6-hop WBN providing the Internet service). In the experiment, we examine the feasibility of our WBN implementation/deployment and also evaluate the effectiveness of increasing the network capacity.

2 Related work

Many studies attempted to increase the network capacity of a WMN by using multiple channels [2–6]. The majority of them focuses on routing protocols, channel assignments, and MAC protocols [3]. Most studies propose a multi-channel routing protocol [4, 9], which jointly works with channel assignment [5]. The use of multiple channels potentially enables to simultaneously transmit packets, thereby being effective to increase the network capacity. However, the number of channels APs can simultaneously use is limited because a practical AP hardware equips with generally two radios and three radios in maximum as far as we know. Therefore, many studies assume that each AP has only two IFs [5]. Since multiple vacant channels more than two are often available, the number of channels each AP simultaneously uses should be flexibly increased in accordance with the number of vacant channels.

The routing studies basically switch channels (routes) to avoid radio interference. However, these studies cannot simultaneously use multiple links (channels) between two neighboring APs because the routing table contains only a single link to reach a neighbor AP. That is, the network capacity between two APs cannot be increased, thereby being a bottleneck. To potentially increase the capacity, packets have to be transmitted through different channels even if all packets are destined to the same destination address. Thus, a traffic management framework that can use multiple channels in parallel is essential.

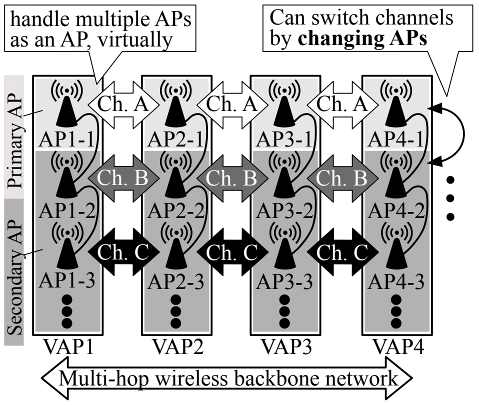

handle multiple APs as an AP, virtually
Can switch channels by **changing APs**
Primary AP
Secondary AP
AP1-1
AP2-1
AP3-1
AP4-1
AP1-2
AP2-2
AP3-2
AP4-2
AP1-3
AP2-3
AP3-3
AP4-3
Ch. A
Ch. B
Ch. C
VAP1
VAP2
VAP3
VAP4
Multi-hop wireless backbone network

Figure 1: Virtual AP based multi-channel WBN.

The studies focusing on MAC protocol try to schedule frame transmissions for collision avoidance between nearby APs [6, 10]. Reference [11] proposes a new MAC protocol transmitting packets on multiple channels based on channel hopping on a single radio (practical hardware equipment). Although it can handle multiple channels, the switching delay in the order of several milliseconds [12] is inherently necessary at every channel switching. It cannot effectively use multiple channels due to the delay and the time-allocated manner. Then, the multiple channels have to be used simultaneously to completely exploit their available resources.

From the viewpoint of OpenFlow, there are few studies employing OpenFlow in WMN [13, 14]. They use OpenFlow functions to provide user mobility in WMN. They mention about the utilization of multiple channels but the structure of WMN is the same with the existing WMN, which does not use multiple channels in parallel. That is, the channel utilization of multiple channel on each hop is not considered.

3 Multi-channel WBN enabling to increase network capacity

3.1 Handling multiple channels

To increase the number of channels that each AP can handle, we employ a virtual AP (VAP) shown in Figure 1. A set of APs (sub-APs), each of which uses a single but different channel for the WBN, is connected by Ethernet of daisy chain and constructs a VAP handling multiple channels. In the WBN constructed by VAPs, since each VAP has multiple wireless links of different channels with neighboring VAPs simultaneously, switching forwarding path of a flow is the same meaning with switching channels of the flow.

As shown in Figure 1, one AP in each VAP is treated as a primary AP (e.g., AP1-1) and the others are secondary APs. Both the primary and secondary APs construct WBNs on respective channels, while only the primary APs provide the Internet access to client terminals. From this architecture, the number of channels used in parallel can be flexibly increased by adding APs for VAP. Note that, to easily identify each VAP, a VAP is indicated with a number, called VAPID, (=X) such like VAP-*X*. In the same way, the sub-APs is denoted as AP-*X*-*Y*, where *Y* is a sequence number of APs in a VAP, called APID.

3.2 Channel utilization of WBN

In WBN, we assume that a routing protocol identifies a route toward IGW (e.g., VAP4 $\to$ VAP3 $\to$ VAP2 $\to$ VAP1 in Figure 1)[1] and thus all traffic are forwarded along with the identified route. To use multiple channels on the route, we propose a traffic management framework based on software defined network (i.e., OpenFlow) technology, which enables us to flexibly select a path (channel) for each *flow* at each hop. A flow is defined based on various identifications from layer 1 to layer 4. In this study, we use a 4-tuple (source/destination IP address and port number) as the flow identification.

OpenFlow consists of one OpenFlow Controller (OFC) and some OpenFlow Switches (OFSs). The OFC establishes a TCP connection with each of OFSs for control message exchange. Then, the OFC determines control rules for each flow (called flow entries) and registers them to OFSs. An OFS (an AP in this study) actually manages packets of each flow by following the registered flow entries stored in the local database (called flow table). Thus, the OFC essentially controls all flow by registering flow entries to OFSs.

A flow entry consists of a flow identification and a corresponding action. In the study, flow entries are registered (a) when an OFS initially connects with the OFC, and (b) when the OFC receives a *packet_in*, which is sent by an OFS whenever the OFS receives an unknown packet (i.e., the packet does not match any flow entries in the flow table). A set of flow entries registered at (a) is fixed rule and is called as *base entry*. Also, the process that the OFC creates/registers a flow entry to an OFS is called as *flow_mod*.

3.3 Traffic management framework

We propose a traffic management framework based on OpenFlow technology. Since a single OFC can collaboratively work with all OFSs and can totally manage all flows through the connected OFSs, the OFC has a potential to dynamically manage flows in response to the variations of the traffic condition in whole WBN. Thus, the channel utilization can be optimized by exploiting the OpenFlow based WBN. To evaluate the performance of the OpenFlow based WBN, we here propose a simple channel utilization method that can multiplex a conventional (single-channel) multi-hop transmission as an example method.

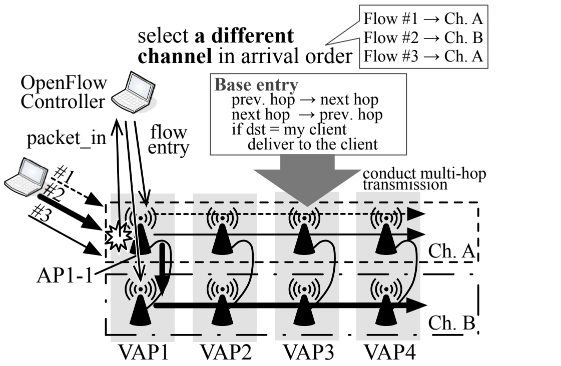


Figure 2: An example of traffic management on multi-channel WBN.

In this method, the OFC allocates a channel for each flow in order of the arrival of *packet_in* and the selected channel is persistently used for multi-hop transmission in the WBN. Figure 2 shows the mechanism of the channel utilization method. The OFC initially registers flow entries (base entry) to all OFSs when establishing a control TCP connection with each OFS. The base entry consists of rules that make all APs forward flows from the previous hop to the next hop and from the next hop to the previous hop. Note that, if the destination of the flow is a client device connecting to its VAP, the AP directly delivers the flow to the destination. Thus, all APs have the base entry conducting multi-hop transmission in advance.

When a client terminal starts a new communication (flow), a first packet arrives at one of APs (e.g., AP 1-1 in Figure 2) and the AP sends a *packet_in* to the OFC. The OFC determines the identification of the flow based on the combination of source/destination pairs of <IP address, port number>, selects a channel persistently used on all hops for the flow, and then registers flow entries, which transmits packets of the flow via the selected channel. In the next hop, the flow is transmitted by the base entry in accordance with the transmission direction. In this way, the combination of the base entry and packet_in driven flow entry achieves the utilization of multiple channels.

To determine a channel for each flow, the OFC selects a different channel one by one in arrival order of *packet_in*. In Figure 2, as three flows are arriving at VAP1, three respective *packet_in*s reporting those flows are sent to the OFC. The OFC selects a channel for each flow in the arrival order of *packet_in* (i.e., channel A for flow 1, channel B for flow 2, and channel A for flow 3 again) and performs *flow_mod* for arrival flow. Then, each VAP forwards all packets of the flow by using the selected channel. The details of this implementation are presented in the reference [15].

[1] Routing is beyond the scope of this paper.

3.4 Overhead introduced by OpenFlow

The delay due to packet_in/flow_mod message exchange and the management traffic coming from this message exchange inherently become overhead. When an AP receiving a new flow (a first packet of new flow), the AP sends a packet_in to the OFC unless a flow entry matching the flow is registered in the AP. The OFC then selects a channel for the flow and registers new flow entry (flow_mod) indicating the selected channel to transmit the flow. After that, the AP starts to forward the flow based on the informed flow entry. Therefore, the delay for registration of flow entry, which highly depends on round trip time (RTT) between the AP and the OFC, is necessary only when receiving a first packet of each flow (i.e., only once for each flow). Moreover, the processing delay for channel selection is also necessary, but it limits to a quite short period. So, the communication interruption period almost becomes RTT between the AP and the OFC.

On the other hand, OpenFlow does not require extra delay to process packets except a first packet of each flow. The OFS process works as a Linux kernel module (kernel process) in each AP and handles packets instead of the TCP/IP stack on the Linux kernel. That is, the OFS process refers to the flow table to forward packets. This kind of process is almost same with the process of the TCP/IP stack, which refers the forwarding information base to forward packets. The paper [13] actually examined the delay by using a userland OFS software and showed that the throughput performance does not depend on the number of flow entries if simple rules (match on port numbers only) are used. Since our method also use only simple rules for all flow entries, the introduction of OpenFlow does not impact on the performance in our study.

From the viewpoint of the amount of management traffic, the amount of traffic by packet_in/flow_mod is necessary. A packet_in message consists of the OpenFlow header (18 bytes) and the entire frame that triggers a new packet_in, while a flow_mod contains some flow entries. That is, the amount of each packet_in depends on the data frame size but the amount of a flow_mod is always same size (150 bytes per frame) in our method. Actually, since we employ UDP with 1500-byte packets (=1514-byte frame), the size of packet_in becomes 1598 bytes (= Ethernet/IP/TCP header (66 bytes) + OpenFlow header (18 bytes) + entire frame (1514 bytes)). Also, since our method triggers packet_in once when receiving a new flow, the total amount of extra traffic will be proportional to the number of the arrivals in new flows and thus the extra management traffic can be limited to a considerable little value.

4 Lab scale proof-of-concept

To evaluate the proposed multi-channel multi-hop WBN, we first construct a lab scale WBN and demonstrate the feasibility of the proposed WBN.

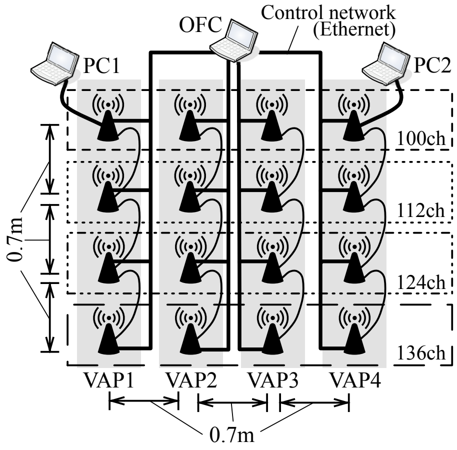


Figure 3: AP placement in the lab scale testbed.

4.1 Facilities and environment

We prepare Buffalo WZR-HP-AG300H as AP hardware. We also install OpenWrt [16] (linux-based OS) with Open vSwitch [17] to all APs. Open vSwitch is a kernel-land OFS software that controls packet transmissions in APs. For the OFC software, we employ Trema [18].

Figure 3 shows the AP placement in the lab scale testbed, which is a 3-hop WBN. Each AP of a VAP is placed to 0.7 m apart from each other and the distance between two neighboring VAPs is 0.7 m. To enable OpenFlow based control, all OFSs (APs) directly connect to an OFC by using dedicated wired network to establish a control connection with the OFC.

In the wireless settings, IEEE802.11a is used to construct the WBN and also four channels (100, 112, 124, and 136 channels) are selectively assigned for the experiment. Note that there is no radio interference on these channels with any other nearby WLANs. To generate traffic, we prepare two PCs, PC1 and PC2. These PCs are connected to AP1-1 and AP4-1 by Ethernet, respectively, because we here focus on the multi-hop data transmission over the WBN.

4.2 Performance measurement in case of UDP traffic

We investigate the maximum network capacity in accordance with the number of channels used in parallel. Note that we define the network capacity as the total throughput at the end host. To keep same traffic rate independent to the network condition, PC1 generates UDP traffic of constant bitrate by using iperf. Specifically, PC1 sends 40 UDP flows (1,500 byte packets) with fixed 1 Mbps to PC2 one by one at 5 seconds interval (totally, 40 Mbps). During the experiment, we measure the total amount of traffic received by PC2 and average the total throughput for 30 seconds after 5 seconds since the last flow starts. This experiment is performed nine times.

Table 1 indicates the summary of measurement results.

Table 1: Total throughput in UDP (Mbps).

	# of channels used in parallel			
	1 ch.	2 ch.	3 ch.	4ch.
Maximum	9.73	19.14	29.03	38.47
Median	9.71	19.05	28.91	38.16
Minimum	9.58	18.96	28.80	38.13

Table 2: Total throughput in TCP (Mbps).

	# of channels used in parallel			
	1 ch.	2 ch.	3 ch.	4 ch.
Maximum	7.67	15.08	22.75	30.14
Median	7.53	15.05	22.66	30.06
Minimum	7.44	14.88	22.60	29.98

In the results, although the total data rate of UDP flows is 40 Mbps, the obtained throughput cannot reach it. Since the traffic is equally distributed in this experiment, all channels on the WBN are exhaustive filled by the traffic. That is, the maximum UDP throughput means the maximum network capacity of the WBN. Then, we can see from the results that the network capacity with a single-channel 3-hop WBN is about 10 Mbps. According to the increase of the number of channels used in parallel, the network capacity also increases. Indeed, it becomes twice in 2 channels and three times in 3 channels. Therefore, our WBN can linearly increase the maximum network capacity in response to the increase of channels.

4.3 Performance measurement in case of TCP traffic

In the Internet access network, there are various flows in which TCP is a dominant. In this experiment, we measure the TCP performance on our WBN while increasing the number of channels used in parallel. TCP has a function that dynamically controls transmission rate depending on the network condition so that traffic congestion could be avoided. Due to the latency of congestion control, we can expect that the network capacity in case of TCP is lower than that in case of UDP. We then examine the (effective) network capacity in the realistic traffic environment.

In this experiment, PC1 performs some TCP data transmissions. Specifically, PC1 establishes a TCP connection with PC2 and transmits data by using iperf through the connection. The number of this TCP flows is increased from one up to 40 one by one at every 5 seconds. As for the capacity, we average the total throughput for 30 seconds after 5 seconds since the last flow starts. This experiment is performed nine times.

Table 2 shows the summary of experimental results. As with Table 1, Table 2 shows that the total throughput is linearly increased in accordance with the number of channels used in parallel. However, the obtained throughput is clearly less than the results in Table 1 due of the congestion control mechanism, as stated previously.

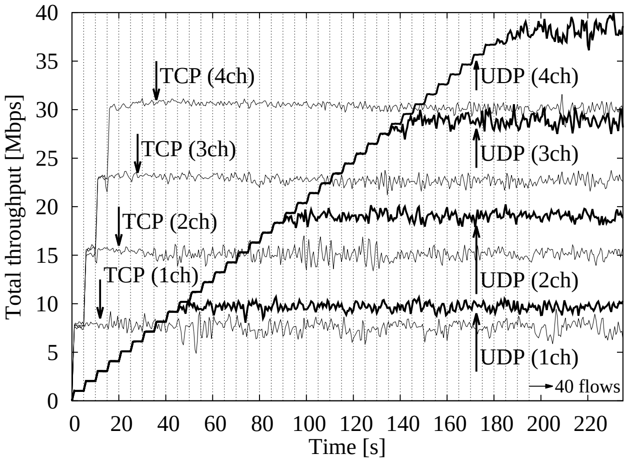
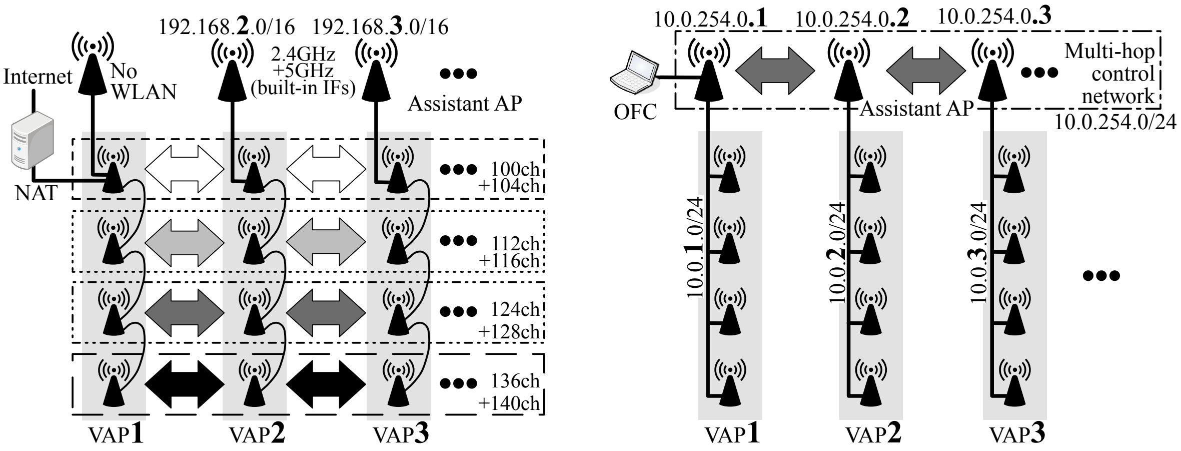


Figure 4: Time series variation on median results.

We next compare TCP with UDP in the time series shown in Figure 4. To illustrate this figure, we select the median results in Tables 1 and 2. Since the number of UDP flows is gradually increased up to 40 flows at every 5 second, throughput also grows along with the increase of flows until the maximum network capacity, which is determined by the number of channels used in parallel. On the other hand, in TCP experiment, once at least one flow is transmitted on every channel, the total throughput is kept in almost same value irrespective of the number of TCP flows (before the number of flows reaches to 40). Moreover, after all flows start to be transmitted, the TCP throughput is about 2Mbps lower on each channel than that of UDP. That is, it is totally 8 Mbps lower in case of the 4-channel WBN. From these results, we can say that our WBN can potentially increase the network capacity in accordance with the number of channels used in parallel but the effective capacity is different depending on the characteristics of traffic.

5 WBN providing the Internet access

We deploy our WBN to provide the Internet access for conference attendees. The conference is held at a single floor of a hotel and our WBN is deployed in a part of the floor. The number of conference attendees is 133 but only a part of them uses our Internet access service because other Internet services are also available and some attendees uses them. The installation environment with wireless configuration is described in Section 5.1. In this experiment, we employ an additional AP (namely assistant AP), whose roles are described in Section 5.2. The design of AP placement is explained in Section 5.3. Finally, we show the results of performance measurement in Section 5.4.



(a) Internet access network.

(b) OpenFlow control network.

Figure 5: Construction of an experimental network.

5.1 Installation environment

As shown in the lower part of Figure 5(a), we employ four APs for each VAP (i.e., four channels available on WBN). These four APs are connected by Ethernet of daisy chain and compose a VAP. Each AP uses the channels of 100, 112, 124, and 136 for WBN, respectively.

In the WLAN settings on all APs, IEEE802.11n is used and an adjacent channel is bonded (i.e., 40 MHz channel bonding is activated). For example, the channel 104 is bonded with the channel 100. It should be noted that our proposed WBN framework can adapt to change in the wireless technology such as 802.11ac and its channel bonding technology. We then manage user traffic through these four links by our proposed channel utilization method described in Section 3.3. Note that we confirmed that there is no radio interference on these channels with any other nearby WLANs in the hotel.

5.2 Assistant APs

We have investigated the basic performance of a lab scale WBN in Section 4. Since our focus was the way of constructing the WBN and its performance, we simplified the experimental environment, as described in Section 4.1.

Each VAP must provide WLAN access for the Internet service. Also, to deploy our proposed WBN in a large scale network, the control traffic arising from OpenFlow should be conveyed on wireless network. Thus, we additionally prepare a Buffalo WZR-HP-AG300H equipped with a USB wireless IF (i.e., this AP has two built-in IFs and one USB IF) for each VAP, called assistant AP. The assistant AP allows us not only (1) to provide WLAN access to client terminals but also (2) to construct the control network, as shown in Figure 5.

5.2.1 Providing the WLAN access

To achieve (1), an assistant AP provides WLAN access in both 2.4 and 5 GHz bands by using two built-in IFs (Figure 5(a)). These WLANs are configured as IEEE802.11n with 20 MHz channel width (i.e., the channel bonding is not used for client user access).

These two IFs and one Ethernet port of the assistant AP are bridged in layer 2 based on the Linux bridge module in OpenWrt and the Ethernet port is connected to the primary AP (i.e., AP-*X*-1) by Ethernet. The primary AP then acts as a DHCP server and dynamically allocates an IP address for each client terminals in accordance with VAPID. For example, when a client terminal associates with the assistant AP of VAP2 on the 2.4GHz or 5GHz bands, an IP address in 192.168.*2*.0/16 is assigned for the client terminal by AP2-1. From this structure, the primary AP can handle user traffic of all client terminals associating with the assistant AP through the Ethernet port.

5.2.2 Control network conveying OpenFlow message

To carry the control messages between an OFC and OFSs, the IP reachability between an OFC and OFSs have to be guaranteed in advance. However, OFS's IFs controlled by OpenFlow cannot receive/transmit any packets until the connection between the OFC and the OFS is established because the OFS does not have any flow entries at first. Thus, the OFS cannot reach the OFC through our WBN. To solve the problem, the control network must be constructed separately from the user network (i.e., WBN). We then prepare the dedicated control network constructed by assistant APs, which are not managed by OpenFlow.

Figure 5(b) illustrates the dedicated control network. The assistant AP connects with all sub-APs by Ethernet cables. To guarantee the IP reachability between the OFC and all OFSs, the assistant APs construct a multi-hop wireless network based on static routing by using the USB wire-

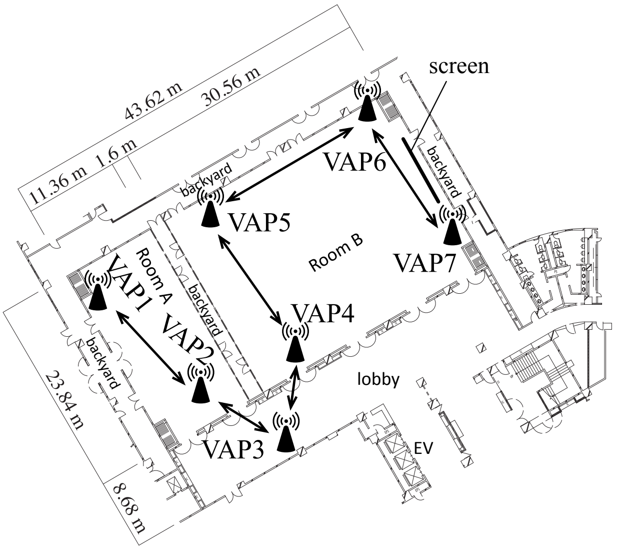


Figure 6: Map of VAP placement in the experimental deployment.

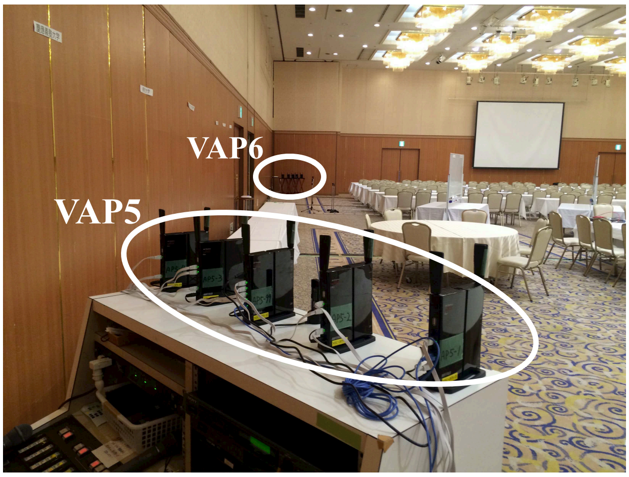


Figure 7: Actual placement of VAP5 and VAP6.

less IF. Since the multi-hop control network is not managed by OpenFlow but a conventional IP routing based network, each OFS can communicate with the OFC in advance. It should be noted that, since this paper focuses on how to use multiple channels on a large scale WBN, the feasibility of the control network will be addressed in the future work.

5.3 AP placement

Figure 6 shows a map of a part of the conference venue with the AP placement. We place seven VAPs in two rooms (Room A and B). An example of the VAP deployment (VAP5 and VAP6) is shown in Figure 7. As seen in the figure, we adjust the location of VAP so as to have line-of-sight between two neighboring VAPs. That is, doors between VAP2 and VAP3 and between VAP3 and VAP4 are always opened.

VAP1 located in Room A is the IGW and directly connects with the OFC. That is, user traffic and control traffic of OpenFlow must pass through VAP1. To ensure that user traffic is forwarded in multi-hop, VAP1 (i.e., the assistant AP of VAP1) does not provide the WLAN access to client terminals as shown in Figure 5(a). Instead, VAP1 (the primary AP of VAP1) connects to a router, which performs NAT of 192.168.0.0/16, so that all users can access the Internet.

Table 3: Throughput for 30 seconds (Mbps).

Maximum	Median	Minimum
30.61	28.94	27.31

Since our proposed WBN can construct only a chain topology with static routing at this time, we design a 6-hop chain topology with a pre-defined path, which is shown by arrows in Figure 6. Thus, all user traffic is forwarded on this multi-hop WBN along with the path.

In the conference, Room A is used for a meeting and Room B is for a plenary event. Since the conference attendees concentrate in Room B when a plenary event is held, we place VAP6 and VAP7 around the plenary event area. As shown in Figure 7, the plenary event area is right side of Room B (in front of the screen) and thus almost attendees may associate with VAP6 or VAP7.

5.4 Performance measurement

We measure the network capacity by obtaining the sum of throughput of all flows on the WBN. In Section 5.4.1, we investigate the maximum network capacity when no users utilize the WBN. Section 5.4.2 evaluates the practical network capacity when many users simultaneously access the Internet by using various applications.

5.4.1 Maximum network capacity

We obtain the maximum performance of the WBN. In this experiment, two PCs connect to VAP1 and VAP7 by Ethernet, respectively, and UDP traffic is transmitted from VAP7 to VAP1 to obtain the maximum network capacity of a 6-hop WBN. During this experiment, there is no other traffic than the experimental UDP traffic.

In the preliminary experiment, we investigated the UDP transmission rate that meets the maximum channel capacity of a 6-hop WBN with a single channel (Figure 6). The measurement is conducted by increasing transmission rate of UDP with 1,500-byte packet by iperf. Since we find that 8 Mbps is the maximum capacity of a channel (packet loss rate is less than 1%) in advance, we here use this traffic rate to measure the network capacity of our WBN.

In the experiment, we generate four 8-Mbps UDP flows with 1,500-byte packets at 10 seconds interval. The throughput is averaged for 30 seconds after 5 seconds since the last flow starts, and this experiment is performed nine times. Table 3 indicates the measurement results. From

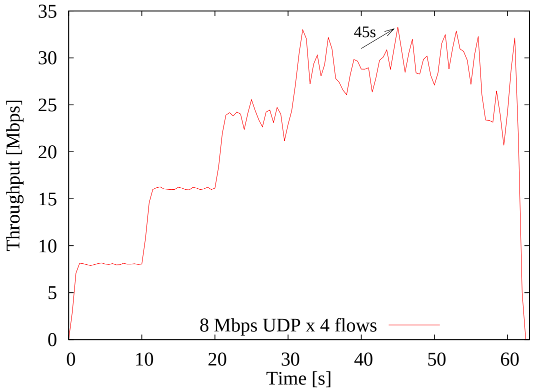


Figure 8: Time series throughput of the average result on the median result.

the table, we can see that the proposed WBN can provide around 29 Mbps of the network capacity.

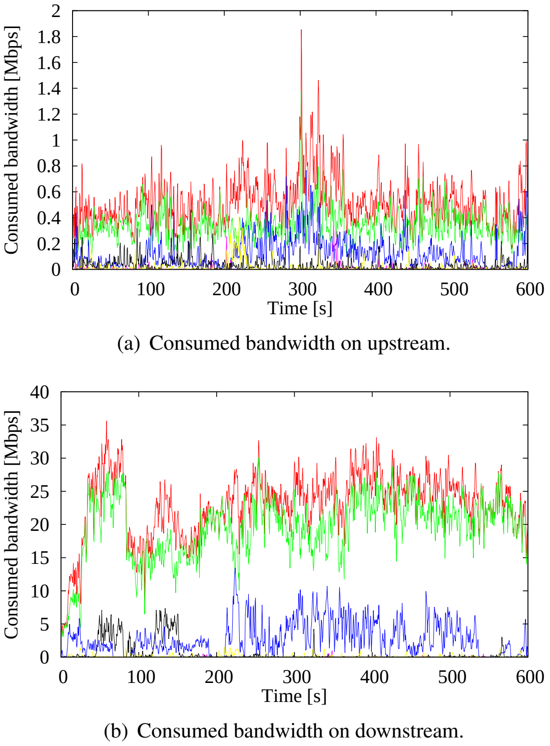
Figure 8 shows the time series throughput of the median result in Table 3. In the figure, we can see that the throughput dynamically fluctuates between 25 and 30 Mbps. Also, the maximum instantaneous throughput is 33.28 Mbps at 45 seconds. From these results, we can say that the proposed WBN potentially has 29 Mbps of the network capacity.

5.4.2 Practical network capacity

We next treat a case where clients (conference attendees) use the WBN to reach the Internet. To obtain the practical network capacity, we measure the total throughput at VAP1 during a plenary event. As a measurement result, we intentionally select the period of 10 minutes during the conference when the largest amount of traffic is observed. Figure 9 shows the actually measured traffic forwarded from/to each VAP during the 10 minutes in time series. From Figures 9(a) and 9(b), we can see that the amount of the downstream traffic is extremely larger than that of the upstream traffic, i.e., about 0.5 Mbps upstream traffic and about 25 Mbps downstream traffic. That is, the asymmetric nature of the Internet traffic can be seen.

From Figure 10, we can see that almost traffic is from/to VAP6 and VAP7. When focusing on VAP7, the throughput measured at 254 seconds is 30.42 Mbps, which is the maximum instantaneous traffic during 10 minutes but a little less than that of UDP experiments (Section 5.4.1). We also show that the stable throughput on VAP7 is around 26 Mbps. Since the WBN of this experiment provides the WLAN Internet access to client terminals, TCP traffic is likely to be dominant in the Internet. As evaluated in Section 4, TCP cannot completely utilize the network capacity due to its congestion control mechanism. Therefore, the network capacity utilized by TCP becomes a little less than the results in Section 5.4.1.

From above results, the OpenFlow based WBN can aggregate the capacity of multiple channels efficiently, thereby providing the Internet access with the large network capacity. In summary, we can conclude that the WBN can effectively extend WLAN coverage while maintaining the large network capacity.



(a) Consumed bandwidth on upstream.

(b) Consumed bandwidth on downstream.

Figure 9: Consumed bandwidth for 10 minutes during a plenary event.

6 Discussion

In this paper, we prepared the experimental environment with no difference/variation of wireless link quality and then conducted experiments to perform the proof-of-concept of the proposed architecture handling multiple channels efficiently. To apply it to the real deployment, we still have some concerns including the difference/variation on wireless condition, the difference/variation on communication, reliability of control network, scalability, and deployment.

In a real deployment, APs may be distributed in wide area so that the radio range of a AP is adjusted to reach only nearby APs. In such a deployment, wireless condition differs in every hop and changes dynamically (e.g., due to radio interference). Since we assume same and stable (no variation of wireless condition) environment in this study, how to apply the proposed method in a real deployment should be further considered. Indeed, when the OFC receives packet_in, the channel selection should be conducted in accordance with the channel condition. While the WBN is transmitting a flow, the OFC should effectively control the channel utilization (switches channels of the

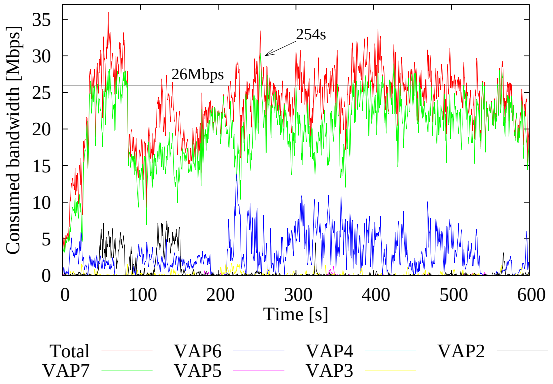


Figure 10: Total bandwidth during the event.

flow) along with the change of channel condition.

To adaptably utilize channels, the OFC has to quickly obtain the difference and variation of wireless condition and then flexibly control the channel utilization. However, since OpenFlow is originally designed for point-to-point wired network, it is not capable of treating point-to-multipoint wireless network, i.e., the OFC cannot obtain any wireless information. To overcome this, the OFC may estimate the wireless link condition based on OpenFlow technology. Actually, since the OFC is capable to collect the amount of sent/received traffic in each AP, the OFC may be aware of the difference or degradation of wireless link quality by comparing the amount of sent traffic in an AP with the amount of the received traffic on the next hop. In this way, we also need to develop the way of obtaining the wireless link condition and accordingly controlling the channel utilization.

Since various sorts of communication are conducted in a real WMN, the channel utilization method has to handle them to effectively utilize multiple channels. Since the flow arrival timing, flow length, and transmission rate are various, the optimization of channel utilization is extremely difficult. That is, some channels may be saturated even if other channels are not full yet. Although the OFC selects a different channel in the order of packet_in arrival in this paper, the amount traffic loaded on each channel may also be additionally considered to utilize multiple channels.

The reliability of OFC and control network should be considered to employ OpenFlow. Since the focus of this paper is proof-of-concept, we actually have not ensured the reliability yet. For improving the OFC reliability, a simple way that prepares multiple OFCs in the WBN may be effective. On the other hand, to keep the control network reliable, it must have redundancy because there is a single point of failure in the current control network. Specifically, since a multi-hop network with a single channel is dedicated for the control network in the current WBN, the WBN causes communication failures if one of wireless links in the control network is disturbed or disconnected. To avoid it, the control network should be construct by multiple channels. Since the channel resources are limited, the control traffic may be coexistent with the data traffic in our WBN.

The scalability issue of the control network should be addressed. As described in Section 3.4, the delay of packet_in/flow_mod delay may increase in accordance with the WBN size. Moreover, the saturation of the control network should be considered. A dedicated channel is used for a multi-hop control network in our WBN. On one hand, the number of clients (flows) tends to increase in accordance with the WBN size (i.e., the coverage area). The control traffic may be dropped due to the saturation of control network in the extensive WBN. These concerns should be further investigated.

Finally, to actually deploy the WMN based on our WBN, we have to consider the configuration and deployment way. In the current WBN, we configure all APs in advance so that all wireless connections including the control network are certainly established. For the deployment, it is necessary to establish the wireless connection and then detect the topology (neighboring APs). Since as we described above the OpenFlow has no functions for wireless network, we need to further develop a way to handle wireless information in the OpenFlow technology. Actually, monitoring beacon frames may be useful to obtain the availability of the channels and collect the information of neighboring APs.

7 Conclusion

To increase the network capacity, this paper introduces the OpenFlow based WBN and a channel utilization method. The WBN is examined in two sorts of testbed. First, we evaluate the basic characteristics of a lab scale proof-of-concept. In the testbed, since we simplified the experimental environment (i.e., client nodes are connected to AP by an Ethernet cable and also the control messages of OpenFlow are conveyed through the wired network), we need to provide WLAN access to client terminals and to construct the wireless network for transmitting control messages.

To solve these problems, we employ an additional AP and then deploy the WBN in a large-scale testbed providing the Internet access to conference attendees. In the measurement, we can demonstrate that the WBN can bring the large amount of network capacity in 6-hops with 4 channels. From the results of two experiments, we can conclude that our WBN can extend WLAN coverage while linearly increasing the network capacity in accordance with the number of channels used in parallel. Since this study focuses on the proof-of-concept of our WBN, we next plan to solve remaining concerns including the difference/variation on wireless condition, the difference/variation on communication, reliability of control network, scalability, and deployment.

Acknowledgment: This work was partly supported by the Telecommunications Advancement Foundation, Japan.

References

[1] Cisco Visual Networking Index. Global Mobile Data Traffic Forecast Update, 2013–2018. `http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.pdf`.

[2] P. H. Pathak and R. Dutta. A Survey of Network Design Problems and Joint Design Approaches in Wireless Mesh Networks. *IEEE Communications Surveys and Tutorials*, 13(3):396–428, 2011.

[3] D. Benyamina, A. Hafid, and M. Gendreau. Wireless Mesh Networks Design - A Survey. *IEEE Communications Surveys and Tutorials*, 14(2):299–310, 2012.

[4] E. Alotaibi and B. Mukherjee. Survey Paper: A Survey on Routing Algorithms for Wireless Ad-Hoc and Mesh Networks. *Computer Networks*, 56(2):940–965, February 2012.

[5] W. Si, S. Selvakennedy, and A. Y. Zomaya. An Overview of Channel Assignment Methods for Multi-radio Multi-channel Wireless Mesh Networks. *Journal of Parallel and Distributed Computing*, 70(5):505–524, May 2010.

[6] K. S. Vijayalayan, A. Harwood, and S. Karunasekera. Distributed Scheduling Schemes for Wireless Mesh Networks: A Survey. *ACM Computing Survey*, 46(1):14:1–14:34, July 2013.

[7] IEEE Std 802.11n-2009.

[8] IEEE Std 802.11ac-2013.

[9] A. Raniwala, K. Gopalan, and T. Chiueh. Centralized Channel Assignment and Routing Algorithms for Multi-channel Wireless Mesh Networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 8(2):50–65, 2004.

[10] P. Kyasanur, J. So, C. Chereddi, and N. F. Vaidya. Multichannel mesh networks: challenges and protocols. *IEEE Wireless Communications*, 13(2):30–36, May 2006.

[11] J. So and N. H. Vaidya. Multi-channel Mac for Ad Hoc Networks: Handling Multi-channel Hidden Terminals Using a Single Transceiver. In *MobiHoc*, pages 222–233, 2004.

[12] A. Raniwala and T. Chiueh. Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. In *INFOCOM*, pages 2223–2234, 2005.

[13] P. Dely, A. Kassler, and N. Bayer. OpenFlow for Wireless Mesh Networks. In *IEEE ICCCN*, pages 1–6, 2011.

[14] A. Detti, C. Pisa, S. Salsano, and N. Blefari-Melazzi. Wireless Mesh Software Defined Networks (wmSDN). In *2nd International Workshop on Community Networks and Bottom-up-Broadband*, pages 89–95, 2013.

[15] M. Tagawa, Y. Wada, Y. Taenaka, and K. Tsukamoto. Network Capacity Expansion Methods based on Efficient Channel Utilization for Multi-Channel Wireless Backbone Network. In *the 2014 International Workshop on Smart Complex Engineered Networks (SCENE)*, August 2014.

[16] OpenWrt. `https://openwrt.org/`.

[17] Open vSwitch. `http://openvswitch.org/`.

[18] Trema. Full-Stack OpenFlow Framework in Ruby and C. `http://trema.github.io/trema/`.

Privacy-preserving Cloud-based Personal Health Record System Using Attribute-based Encryption and Anonymous Multi-Receiver Identity-based Encryption

Changji Wang
Cisco School of Informatics, Guangdong University of Foreign Studies, Guangzhou 510006, China
E-mail: wchangji@gmail.com

Xilei Xu, Dongyuan Shi and Jian Fang
School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510275, China

Keywords: personal health record, cloud computing, ciphertext-policy attribute-based encryption, anonymous multi-receiver identity-based encryption

Received: July 16, 2015

As an emerging patient-centric model of health information exchange, cloud-based personal health record (CB-PHR) system holds great promise for empowering patients and ensuring more effective delivery of health care. In this paper, we design a novel CB-PHR system. It allows PHR owners to securely store their health data on the semi-trusted cloud service providers, and to selectively share their health data with a wide range of PHR users. To reduce the key management complexity, we divide PHR users into two security domains named public domain and personal domain. PHR owners encrypt their health data for the public domain using ciphertext-policy attribute-based encryption scheme, while encrypt their health data for the personal domain using anonymous multi-receiver identity-based encryption scheme. Only authorized users whose credentials satisfy the specified ciphertext-policy or whose identities belong to dedicated identities can decrypt the encrypted health data. Extensive analytical and experimental results are presented which show that our CB-PHR system is secure, privacy-protected, scalable and efficient.

Povzetek: Predstavljen je sistem CB-PHR, tj. sistem za oblačne zdravstvene kartone.

1 Introduction

In recent years, personal health record system has emerged as a patient-centric model of health information exchange. It enables the patient to create and control their health data in a centralized place through web-based application from anywhere and at any time, which has made the storage, retrieval, and sharing of the health data more efficient. Due to the high cost of building and maintaining specialized data centers, as well as vigorous development of cloud computing in recent years, many PHR services are outsourced to third-party cloud service providers (CSPs), for example, Microsoft Health Vault, Google Health, Indivo and MyPHR.

Although cloud-assisted PHR services could offer a great opportunity to improve the quality of health care services and potentially reduce health care costs, there have been wide privacy concerns as personal health information could be exposed to those semi-trusted CSPs and to unauthorized parties. Health data can reveal very sensitive information, including fertility, surgical procedures, emotional and psychological disorders and diseases, etc. There exist health care regulations such as HIPAA which is recently amended to incorporate business associates, but CSPs are usually not covered entities. Moreover, due to the high value of health data, CSPs are often the targets of various malicious behaviors which may lead to exposure of health data. In addition, CSPs have significant commercial interest in collecting and sharing patients' health data with either pharmacy companies, research institutions or insurance companies.

To keep sensitive health data confidential against those semi-trusted CSPs and unauthorized parties in a CB-PHR system, a natural way is to store only the encrypted data in the cloud. While it is important to allow patients to selectively share their health data with a wide range of users, including staffs from health care providers and medical research institutions, and family members or friends, thus it is essential to provide fine-grained data access control mechanisms that work with semi-trusted CSPs.

1.1 Related work

Anonymous Multi-Receiver Identity-Based Encryption: Boneh and Franklin [1] proposed the first practical and secure identity-based encryption (IBE) scheme from bilinear pairings. Since then, IBE has attracted a lot of attention and a large number of IBE schemes and related systems have been proposed.

Considering a situation where a sender would like to encrypt a message for t receivers, the sender must encrypt the message t time using conventional IBE schemes. To improve the performance, Baek et al. [2] first introduced the notion of multi-receiver IBE scheme, and proposed an efficient provably secure multi-receiver IBE scheme from bilinear pairings. Next, Boyen and Waters [3] proposed an anonymous IBE scheme to guarantee receiver's privacy, where the ciphertext does not leak the identity of the recipient. Later, Fan et al. [4] introduced the concept of anonymous multi-receiver IBE (AMRIBE) scheme, and proposed an AMRIBE scheme from bilinear parings. Fan et al. claimed that their AMRIBE scheme makes it impossible for an attacker or any other receiver to derive the identity of a message receiver such that the privacy of every receiver can be guaranteed. Unfortunately, Chien [5] showed that in Fan et al.'s AMRIBE scheme any selected receiver may extract the identities of the other selected receivers, and presented an improved AMRIBE scheme. However, only heuristic arguments for security proofs are presented. Recently, Tseng et al. [6] proposed an efficient AMRIBE scheme with complete receiver anonymity and proved that the scheme is semantically secure against adaptively chosen-ciphertext attacks.

Attribute-Based Encryption: In some scenarios, the recipient of the ciphertext is not yet known at the time of the encryption or there are more than one recipient who should be able to decrypt the ciphertext. To preserve data confidentiality and enforce fine-grained access control simultaneously, Sahai and Waters [7] first introduced the concept of attribute-based encryption (ABE), which is envisioned as an important tool for addressing the problem of secure and fine-grained data sharing and access control.

ABE has attracted lots of attention from both academia and industry in recent years, various ABE schemes have been proposed, such as [8–13]. There are two main types of ABE schemes in the literatures: Key-Policy ABE (KP-ABE) and Ciphertext-Policy ABE (CP-ABE).

In a KP-ABE system, ciphertexts are labeled by the sender with a set of descriptive attributes, and users' private keys are issued by the trusted attribute authority are associated with access structures that specify which type of ciphertexts the key can decrypt. Goyal et al. [8] proposed the first KP-ABE scheme, which was very expressive in that it allowed the access policies to be expressed by any monotonic formula over encrypted data. While in a CP-ABE system, when a sender encrypts a message, they specify a specific access policy in terms of access structure over attributes in the ciphertext, stating what kind of receivers will be able to decrypt the ciphertext. Users possess sets of attributes and obtain corresponding secret attribute keys from the attribute authority, such a user can decrypt a ciphertext if his/her attributes satisfy the access policy associated with the ciphertext. Bethencourt et al. [9] constructed the first CP-ABE scheme, but its security was proved in the generic group model. Later, Waters [10] proposed an efficient CP-ABE scheme with expressive access policy described in general linear secret sharing scheme.

Several CB-PHR systems using ABE schemes have been developed in recent years. Ibraimi et al. [14] proposed a secure PHR management system using Bethencourt et al.'s CP-ABE scheme, which allows PHR owners to encrypt their health data according to an access policy over a set of attributes issued by two trusted authorities. Later, Li et al. [15] proposed a secure and scalable PHR sharing framework on semi-trusted storage servers under multi-owner settings by leveraging both KP-ABE and CP-ABE techniques.

1.2 Our contributions

As we all know, semantically secure against adaptive chosen-ciphertext attacks (IND-CCA) is the de facto level of security required for asymmetric encryption schemes used in practice. Access policy supported by Waters's CP-ABE scheme [10] is expressive. However, it is only proved to be semantically secure against chosen-plaintext attack (IND-CPA). Okamoto and Pointcheval [16] proposed a method named rapid enhanced-security asymmetric cryptosystems transform (REACT) for any asymmetric encryption schemes to achieve IND-CCA secure from IND-CPA secure. In this paper, we first apply REACT technique for Waters' CP-ABE scheme [10] to obtain an IND-CCA secure CP-ABE scheme in the random oracle model.

Tseng et al. [6] extended Boneh and Franklin's IBE scheme [1] to multiple recipients scenario and proposed an efficient AMRIBE scheme. To achieve IND-CCA secure, they adopted the Fujisaki-Okamoto transformation [17] for any asymmetric encryption schemes to achieve IND-CCA secure from one-way secure in the random oracle model. We note that k can play the same role as σ in the Fujisaki-Okamoto transformation of Tseng et al.'s AMRIBE scheme [6]. In this paper, we further improve Tseng et al.'s AMRIBE scheme without compromising security.

Finally, we propose a new CB-PHR system, which allows patients to securely store their health data on semi-trusted CSPs, and selectively share their health data with a wide range of users, including health care professionals like doctors and nurses, family members or friends. To reduce the key management complexity for PHR owners and PHR users, we divide the system into public domain (PUD) and personal domain (PSD). The PUD consists of users who make access based on their professional roles, such as doctors, nurses and medical researchers. The PSD consists of users who are familiar to the PHR owner, such as family members or close friends. PHR owners encrypt their health data for the PUD user using CP-ABE scheme, while they encrypt their health data for the PSD using AMRIBE scheme. Only authorized users whose credentials satisfy the specified ciphertext-policy or whose identities belong to dedicated identities can decrypt the encrypted health data, where ciphertext-policy or dedicated identities are embedded in the encrypted health data.

1.3 Paper organization

This paper is structured as follows. We review some necessary preliminary work in Section 2. Next, we describe our proposed CB-PHR system in Section 3. Then, we give security and efficiency analysis in Section 4. Finally, we conclude our paper and discuss our future work in Section 5.

2 Preliminaries

A prime order bilinear group generator $\mathcal{G}$ is an algorithm that takes as input a security parameter κ and outputs a bilinear group $(p, \mathbf{G}_1, \mathbf{G}_2, \hat{e}, g)$, where p is a prime of size 2^κ, $\mathbf{G}_1$ and $\mathbf{G}_2$ are p order cyclic groups, g is a generator of $\mathbf{G}_1$, and $\hat{e} : \mathbf{G}_1 \times \mathbf{G}_1 \rightarrow \mathbf{G}_2$ is a bilinear map with the following properties:

- Bilinearity: $\hat{e}(g^a, g^b) = \hat{e}(g, g)^{ab}$ for $a, b \overset{\$}{\leftarrow} \mathbf{Z}_p^*$. Here $x \overset{\$}{\leftarrow} \mathbf{S}$ is denoted by picking an element a uniformly at random from the set $\mathbf{S}$.
- Non-degeneracy: $\hat{e}(g, g)$ is a generator of $\mathbf{G}_2$.
- Computability: There is an efficient algorithm to compute $\hat{e}(g_1, g_2)$ for $g_1, g_2 \overset{\$}{\leftarrow} \mathbf{G}_1$.

The *bilinear Diffie-Hellman (BDH)* assumption in a prime order bilinear group $(p, \mathbf{G}_1, \mathbf{G}_2, \hat{e}, g)$ is that if a tuple (g, g^a, g^b, g^c) is given for unknown $a, b, c \overset{\$}{\leftarrow} \mathbf{Z}_p^*$, there is no probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ can compute $\hat{e}(g, g)^{abc}$ with non-negligible advantage.

The *decisional bilinear Diffie-Hellman (DBDH)* assumption in a prime order bilinear group $(p, \mathbf{G}_1, \mathbf{G}_2, \hat{e}, g)$ is that if a tuple (g, g^a, g^b, g^c, T) is given for unknown $a, b, c \overset{\$}{\leftarrow} \mathbf{Z}_p^*$ and $T \overset{\$}{\leftarrow} \mathbf{G}_2$, there is no PPT adversary $\mathcal{A}$ can decide whether $T = \hat{e}(g, g)^{abc}$ with non-negligible advantage.

The *gap bilinear Diffie-Hellman (GBDH)* assumption in a prime order bilinear group $(p, \mathbf{G}_1, \mathbf{G}_2, \hat{e}, g)$ is that if a tuple (g, g^a, g^b, g^c) is given for unknown $a, b, c \overset{\$}{\leftarrow} \mathbf{Z}_p^*$, there is no PPT adversary $\mathcal{A}$ can compute $\hat{e}(g, g)^{abc}$ with the help of the DBDH oracle with non-negligible advantage. The DBDH oracle means that given a tuple (g, g^a, g^b, g^c, T), outputs 1 if $T = \hat{e}(g, g)^{abc}$ and 0 otherwise.

The *decisional q-parallel bilinear Diffie-Hellman exponent (q-DBDHE)* assumption is that if $X \overset{\$}{\leftarrow} \mathbf{G}_2$ and $\vec{y} =$

$$\begin{aligned}&(g, g^s, g^a, \ldots, g^{(a^q)}, g^{(a^{q+2})}, \ldots, g^{(a^{2q})},\\ &g^{s \cdot b_j}, g^{a/b_j}, \ldots, g^{(a^q/b_j)}, g^{(a^{q+2}/b_j)}, \ldots, g^{(a^{2q}/b_j)},\\ &g^{a \cdot s \cdot b_k/b_j}, \ldots, g^{(a^q \cdot s \cdot b_k/b_j)}).\end{aligned}$$

are given for unknown $a, s, b_1, \ldots, b_q \overset{\$}{\leftarrow} \mathbf{Z}_p^*$, where $1 \leq j \leq q$, $1 \leq k \leq q$ and $k \neq j$, there is no PPT adversary $\mathcal{A}$ can decide whether $X = \hat{e}(g, g)^{a^{q+1}s}$ with non-negligible advantage.

Let $\mathbf{\Omega} = \{\mathsf{attr}_1, \mathsf{attr}_2, \ldots, \mathsf{attr}_n\}$ be a set of attributes. A collection $\mathbb{A} \subseteq 2^{\mathbf{\Omega}}$ is monotone if for any set of attributes $\vec{\eta}$ and $\vec{\vartheta}$, we have that if $\vec{\eta} \in \mathbb{A}$ and $\vec{\eta} \subseteq \vec{\vartheta}$ then $\vec{\vartheta} \in \mathbb{A}$. An *access structure* (respectively, *monotone access structure*) is a collection (respectively, monotone collection) $\mathbb{A} \subseteq 2^{\mathbf{\Omega}} \setminus \{\emptyset\}$. The sets in $\mathbb{A}$ are called the authorized sets of attributes, and the sets not in $\mathbb{A}$ are called the unauthorized sets of attributes.

If a set of attributes $\vec{\omega}$ satisfies an access structure $\mathbb{A}$, we denote it as $\mathbb{A}(\vec{\omega}) = 1$. In this paper, we restrict our attention to monotone access structures. As stated in [18], any monotone access structure can be represented by a linear secret sharing scheme (LSSS). A secret sharing scheme Π for an access structure $\mathbb{A}$ over a set of attributes $\mathbf{\Omega}$ is called linear over $\mathbf{Z}_p$ if

- The shares for each attribute form a vector over $\mathbf{Z}_p$.
- There exists a matrix $\mathbf{M}_{\ell \times n}$ called the share generating matrix for Π. For all $i = 1, 2, \ldots, \ell$, we let the function ρ defined the attribute labeling row i of $\mathbf{M}_{\ell \times n}$ as $\rho(i)$. When we consider the column vector $\vec{v} = (s, r_2, \ldots, r_n)^\intercal$, where $s \in \mathbf{Z}_p$ is the secret to be shared, and $r_2, \ldots, r_n \overset{\$}{\leftarrow} \mathbf{Z}_p$, then $\vec{\alpha} = \mathbf{M}_{\ell \times n}\vec{v}$ is the vector of ℓ shares of the secret s according to Π. The share $\alpha_i = (\mathbf{M}_{\ell \times n}\vec{v})_i$ belongs to attribute $\rho(i)$.

Beimel [18] showed that every LSSS enjoys the linear reconstruction property: Suppose that Π is a LSSS for the access structure $\mathbb{A}$. Let $\vec{\omega} \in \mathbb{A}$ be any authorized set, and define $\mathbf{I} = \{i | \rho(i) \in \vec{\omega}\} \subset \{1, 2, \ldots, \ell\}$. If $\{\alpha_i\}$ are valid shares of any secret s according to Π, then there exist constants $\{\beta_i\}$ for $i \in \mathbf{I}$ such that $\sum_{i \in \mathbf{I}} \alpha_i \beta_i = s$, and these constants $\{\beta_i\}$ can be found in time polynomial in the size of $\mathbf{M}_{\ell \times n}$. For unauthorized sets, no such constants $\{\beta_i\}$ exist.

3 Our CB-PHR system

There are four participants involved in our CB-PHR system.

- A trusted authority (TA), who acts as the root of trust and is responsible for generating system parameters, issuing attribute-based private keys or identity-based private keys for PHR owners and PHR users.
- A semi-trusted CSP, who manages a cloud to provide data storage service. It is important to assume that CSP is semi-trusted, which means CSP will try to find out as much secret information in the stored health data as possible, but it will honestly follow the protocol in general.
- Multiple PHR users, who belong to PUD or PSD. PHR users in PUD make access based on their professional roles, such as doctors, nurses, and medical researchers, while PHR users in PSD make access based

on their identities, such as patients' family members or close friends.

- Multiple PHR owners (patients), who encrypt and outsource their sensitive health data to CSP. Specifically, PHR owners encrypt their health data for PUD users using improved Waters' CP-ABE scheme, while they encrypt their health data for PSD users using improved Tseng et al.'s AMRIBE scheme.

Fig.1 illustrates the system architecture and workflow of our CB-PHR system, which is explained as follows.

3.1 Setup

TA first defines the universe $\mathbf{\Omega}$ of attributes, runs $\mathcal{G}(1^\kappa) \to (p, \mathbf{G}_1, \mathbf{G}_2, \hat{e}, g)$, chooses $x, y \xleftarrow{\$} \mathbf{Z}_p^*$, $h_i \xleftarrow{\$} \mathbf{G}_1$ for $1 \leq i \leq n$. Next, TA computes $h = g^x$ and $Y = \hat{e}(g,g)^y$, picks a semantically secure symmetric encryption scheme Γ with key space $\mathbf{K}$, encryption algorithm Enc and decryption algorithm Dec, respectively. TA then chooses a cryptographically secure message authentication code $\mathrm{MAC} : \mathbf{K} \times \{0,1\}^* \to \mathbf{Z}_p^*$, three cryptographically secure hash functions: $H_1 : \{0,1\}^* \to \mathbf{G}_1$, $H_2 : \mathbf{G}_2 \to \mathbf{K}$ and $H_3 : \mathbf{G}_2 \to \mathbf{Z}_p^*$. Finally, TA sets the master secret key $msk = \langle x, g^y \rangle$, and the system parameters $mpk = \langle \mathbf{\Omega}, p, \mathbf{G}_1, \mathbf{G}_2, \hat{e}, g, h, Y, \{h_i\}_{i=1}^n, \{H_i\}_{i=1}^3, \mathrm{MAC}, \Gamma \rangle$.

3.2 KeyGen

Given a user's identity ID, and a set $\vec{\omega} \subseteq \mathbf{\Omega}$ of attributes belonging to the user, TA chooses $z \xleftarrow{\$} \mathbf{Z}_p^*$, computes $g_{\mathsf{ID}} = H_1(\mathsf{ID})$, $D_{\mathsf{ID}} = g_{\mathsf{ID}}^x$, $K = g^{xz}g^y$, $L = g^z$, $K_i = h_i^z$ for all $\mathsf{attr}_i \in \vec{\omega}$. TA then sets user's private key $sk_{\mathsf{ID},\vec{\omega}} = \langle D_{\mathsf{ID}}, K, L, \{K_i\}_{\mathsf{attr}_i \in \vec{\omega}} \rangle$, and sends $sk_{\mathsf{ID},\vec{\omega}}$ to the user via a secure channel.

Note: If a user requests identity-based private key corresponding to an identity ID, then TA only needs to compute $sk_{\mathsf{ID}} = D_{\mathsf{ID}}$. If a user requests attribute-based private key corresponding to a set $\vec{\omega}$ of attribute, then TA only needs to compute $sk_{\vec{\omega}} = \langle K, L, \{K_i\}_{\mathsf{attr}_i \in \vec{\omega}} \rangle$.

3.3 Encrypt

Given an original health data m to be encrypted, a LSSS access structure $\mathbb{A} = (\mathbf{M}_{\ell \times n}, \rho)$ and a list of identities $\mathbf{ID}_R = \{\mathsf{ID}_i\}_{i=1}^t$, PHR owner performs the following steps.

1. Choose $s \xleftarrow{\$} \mathbf{Z}_p^*$, $u_1, \ldots, u_n, r_1, \ldots, r_\ell \xleftarrow{\$} \mathbf{Z}_p$, $U \xleftarrow{\$} \mathbf{G}_2$, and set $\vec{u} = (s, u_2, \ldots, u_n)^\mathsf{T}$.

2. Compute $k_1 = H_2(U)$, $E_1 = \mathrm{Enc}(k_1, m)$, $C' = g^s$, $C_1' = U \cdot \hat{e}(g,g)^{sy}$, $\alpha_i = (\mathbf{M}_{\ell \times n}\vec{u})_i$, $C_i = g^{x\alpha_i} h_{\rho(i)}^{-r_i}$, and $D_i = g^{r_i}$ for $1 \leq i \leq \ell$, $\lambda_1 = \mathrm{MAC}(k_1, m, E_1, C', C_1', C_1, D_1, \ldots, C_\ell, D_\ell)$.

3. Choose $k_2 \xleftarrow{\$} \mathbf{K}$, compute $E_2 = \mathrm{Enc}(k_2, m)$, $g_{\mathsf{ID}_i} = H_1(\mathsf{ID}_i)$ and $v_i = H_3(\hat{e}(g_{\mathsf{ID}_i}, h)^s)$ for $\mathsf{ID}_i \in \mathbf{ID}_R$.

4. Construct the polynomial $f(x) = \prod_{i=1}^t (x - v_i) + k_2 = c_0 + c_1 x + \cdots + c_{t-1}x^{t-1} + x^t \bmod p$, compute $\lambda_2 = \mathrm{MAC}(k_2, m, E_2, C', c_0, c_1, \ldots, c_{t-1})$.

5. Set the ciphertext $CT = \langle C', C_1', \{C_i, D_i\}_{i=1}^\ell, \{c_i\}_{i=0}^{t-1}, E_1, E_2, \lambda_1, \lambda_2 \rangle$.

6. Finally, PHR owner uploads the ciphertext to CSP along with a description of access policy $(\mathbf{M}_{\ell \times n}, \rho)$ and a set of identities of designated recipients $\mathbf{ID}_R$.

Note: If a PHR owner wants to share his/her health data with PHR users from the PUD, then the PHR owner only needs to perform step 1 and step 2. If a PHR owner wants to share his/her health data with PHR users from the PSD, then the PHR owner only needs to perform step 3, step 4 and compute $C' = g^s$.

3.4 Decrypt

Given a ciphertext CT along with a description of access policy $\mathbb{A} = (\mathbf{M}_{\ell \times n}, \rho)$ and a set $\mathbf{ID}_R$ of identities, a PHR user performs different steps depending on whether the PHR user is from the PUD or from the PSD.

- If the PHR user is from the PUD, and he owns credentials corresponding to a set $\vec{\omega}$ of attributes such that $\mathbb{A}(\vec{\omega}) = 1$, then the PHR user computes

$$\begin{aligned}
\widetilde{U} &= C_1' \cdot \frac{\prod_{i \in \mathbf{I}} (\hat{e}(C_i, L)\hat{e}(D_i, K_{\rho(i)}))^{\beta_i}}{\hat{e}(C', K)} \\
\widetilde{k}_1 &= H_2(\widetilde{U}) \\
\widetilde{m} &= \mathrm{Dec}(\widetilde{k}_1, E_1) \\
\widetilde{\lambda}_1 &= \mathrm{MAC}(\widetilde{k}_1, \widetilde{m}, E_1, C', C_1', \{C_i, D_i\}_{i=1}^\ell)
\end{aligned}$$

  where $\rho(i)$, β_i and $\mathbf{I}$ are defined in Section 2. Finally, PHR user tests whether $\widetilde{\lambda}_1 = \lambda_1$ or not. If it holds, PHR user accepts the message $\widetilde{m} = m$ and outputs $\perp$ otherwise.

- If the PHR user is from the PSD, and his identity ID_i belongs to the set $\mathbf{ID}_R$ of identities of designated recipients, then the PHR user computes

$$\begin{aligned}
\widehat{v}_i &= H_2(\hat{e}(D_{\mathsf{ID}_i}, C')) \\
\widehat{k}_2 &= f(\widetilde{v}_i) \\
&= c_0 + c_1\widetilde{v}_i + \ldots + c_{t-1}\widetilde{v}_i^{t-1} + \widetilde{v}_i^t \bmod p \\
\widehat{m} &= \mathrm{Dec}(\widehat{k}_2, E_2), \\
\widehat{\lambda}_2 &= \mathrm{MAC}(\widehat{k}_2, \widehat{m}, E_2, C', c_0, c_1, \ldots, c_{t-1})
\end{aligned}$$

  Finally, PHR user tests whether $\widehat{\lambda}_2 = \lambda_2$ or not. If it holds, PHR user accepts the message $\widehat{m} = m$ and outputs $\perp$ otherwise.

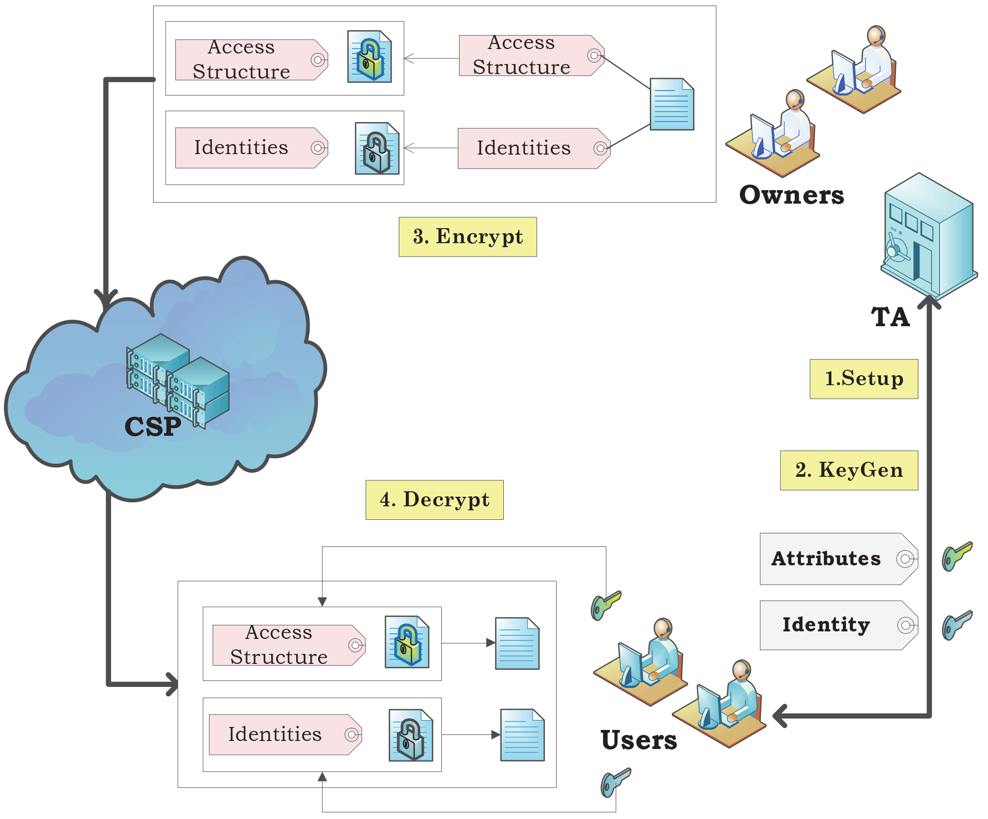


Figure 1: Architecture and workflow of our CB-PHR system.

4 Security proofs and efficiency analysis

Theorem 1. *Our CB-PHR system is correct.*

Proof. The correctness can be verified as follows.

$$\begin{aligned}
&\frac{\hat{e}(C',K)}{\prod_{i\in\mathbf{I}}(\hat{e}(C_i,L)\hat{e}(D_i,K_{\rho(i)}))^{\beta_i}}\\
&\quad=\frac{\hat{e}(g^s,g^{xz}g^y)}{\prod_{i\in\mathbf{I}}[\hat{e}(g^{x\alpha_i}h_{\rho(i)}^{-r_i},g^z)\hat{e}(g^{r_i},h_{\rho(i)}^z)]^{\beta_i}}\\
&\quad=\frac{\hat{e}(g,g)^{sy}\hat{e}(g,g)^{sxz}}{\prod_{i\in\mathbf{I}}\hat{e}(g,g)^{xz\alpha_i\beta_i}}=\frac{\hat{e}(g,g)^{sy}\hat{e}(g,g)^{sxz}}{\hat{e}(g,g)^{xz\sum_{i\in\mathbf{I}}\alpha_i\beta_i}}\\
&\quad=\frac{\hat{e}(g,g)^{sy}\hat{e}(g,g)^{sxz}}{\hat{e}(g,g)^{sxz}}=\hat{e}(g,g)^{sy}\\
&H_2(\hat{e}(D_{\mathsf{ID}_i},C'))=H_2(\hat{e}(g^x_{\mathsf{ID}_i},g^s))\\
&\quad=H_2(\hat{e}(g_{\mathsf{ID}_i},h)^s)=v_i\\
&f(x)=c_0+c_1x+\cdots+c_{t-1}x^{t-1}+x^t\\
&\quad=\prod_{i=1}^{t}(x-v_i)+k_2 \bmod p\\
&\quad=(x-v_i)F(x)+k_2 \bmod p\\
&\quad\Rightarrow f(v_i)=c_0+c_1v_i+\ldots+c_{t-1}v_i^{t-1}+v_i^t\\
&\quad=(v_i-v_i)F(v_i)+k_2 \bmod p=k_2
\end{aligned}$$

This completes the proof. □

Theorem 2. *Our CB-PHR system satisfies receiver anonymity in the random oracle model under the GBDH assumption.*

Proof. PHR owners encrypt their health data for receivers in the PUD using an improved Waters's CP-ABE scheme, where REACT technique [16] is applied to achieve IND-CCA secure. Intended receivers are specified through attributes owned by receivers instead of receivers' identities, and these attributes are potentially able to be shared by unlimited number of PHR users. Thus receiver anonymity is satisfied for PHR users in the PUD.

PHR owners encrypt their health data for PHR users in the PSD using an improved Tseng et al.'s AMRIBE scheme [6]. We improved Tseng et al.'s AMRIBE scheme [6] without compromising security by removing σ and related operations, because k plays the same role as σ in the Fujisaki-Okamoto transformation of Tseng et al.'s AMRIBE scheme [6]. Tseng et al.'s AMRIBE scheme is proved to satisfy receiver anonymity in the random oracle model under the GBDH assumption, thus receiver anonymity is satisfied for PHR users in the PSD. □

Table 1: Efficiency analysis of our CB-PHR system

	Private key size	Encrypt cost	Decrypt cost
PHR Owner	×	$N_{\mathrm{R}}t_p + (2\ell+1)t_m + t_e + 2t_E + N_{\mathrm{R}}t_H$	×
A PUD User	$(N_{\mathrm{A}}+2)\lvert\mathbf{G}_1\rvert$	×	$(2+N_{\mathbf{I}})t_p + N_{\mathbf{I}}t_e + t_D$
A PSD User	$\lvert\mathbf{G}_1\rvert$	×	$t_p + t_D$

Theorem 3. *Our CB-PHR system is IND-CCA secure in the selective model under the* q*-DBDHE assumption and GBDH assumption.*

Proof. PHR owners encrypt their health data for PHR users in the PUD using our improved IND-CCA secure CP-ABE scheme, which is obtained by applying REACT transformation for Waters' CP-ABE scheme [10]. Waters' CP-ABE scheme is proved to be IND-CPA secure in the selective model under the q-DBDHE assumption, and REACT transformation is a generic method for any asymmetric encryption schemes to achieve IND-CCA secure from IND-CPA secure, thus our improved CP-ABE scheme is IND-CCA secure in the selective model under the q-DBDHE assumption. For detailed proofs, we recommend you refer to [10] and [16].

PHR owners encrypt their health data in the PSD using our improved Tseng et al.'s AMRIBE scheme. We improved Tseng et al.'s AMRIBE scheme [6] without compromising security by removing σ and related operations, because k plays the same role as σ in the Fujisaki-Okamoto transformation of Tseng et al.'s AMRIBE scheme [6]. Tseng et al.'s AMRIBE scheme is proved to be IND-CCA secure in the selective model under the GBDH assumption, thus our improved AMRIBE scheme is IND-CCA secure in the selective model under the GBDH assumption. For detailed proofs, we recommend you refer to [6].

In summary, our CB-PHR system is IND-CCA secure in the selective model under the q-DBDHE assumption and GBDH assumption. □

Table 1 shows the computational cost of each participant in our CB-PHR system. Denote by t_p, t_m, t_e, t_H, t_E, t_D, the computation cost of a bilinear pairing in $(\mathbf{G}_1, \mathbf{G}_2)$, a multiplication in $\mathbf{G}_1$, an exponentiation in $\mathbf{G}_2$, a map-to-point hash function H_1, an encryption and a decryption in Γ, respectively. Other operations are omitted in the following analysis since their computation cost is trivial. Denote by N_{R}, N_{A}, $N_{\mathbf{I}}$, $|m|$, $|\mathbf{G}_1|$ and $|\mathbf{Z}_q^*|$ the number of receivers in the PSD, the number of attributes owned by a user in the PUD, the number of attributes in the set $\mathbf{I}$, the bit-length of a plaintext, an element in group $\mathbf{G}_1$, and an element in group $\mathbf{Z}_q^*$, respectively.

In order to evaluate the performance of our CB-PHR system, we implement the corresponding algorithms in our CB-PHR system based on Charm Crypto Framework (version 0.42) [19] and pairing-based crypto (PBC) library [22]. Figure 2 shows the performance of our CB-PHR system, where times are measured in seconds (averaged over 30 iterations) and were computed on an Intel processor with 2GB RAM and hosted on 2.40GHz.

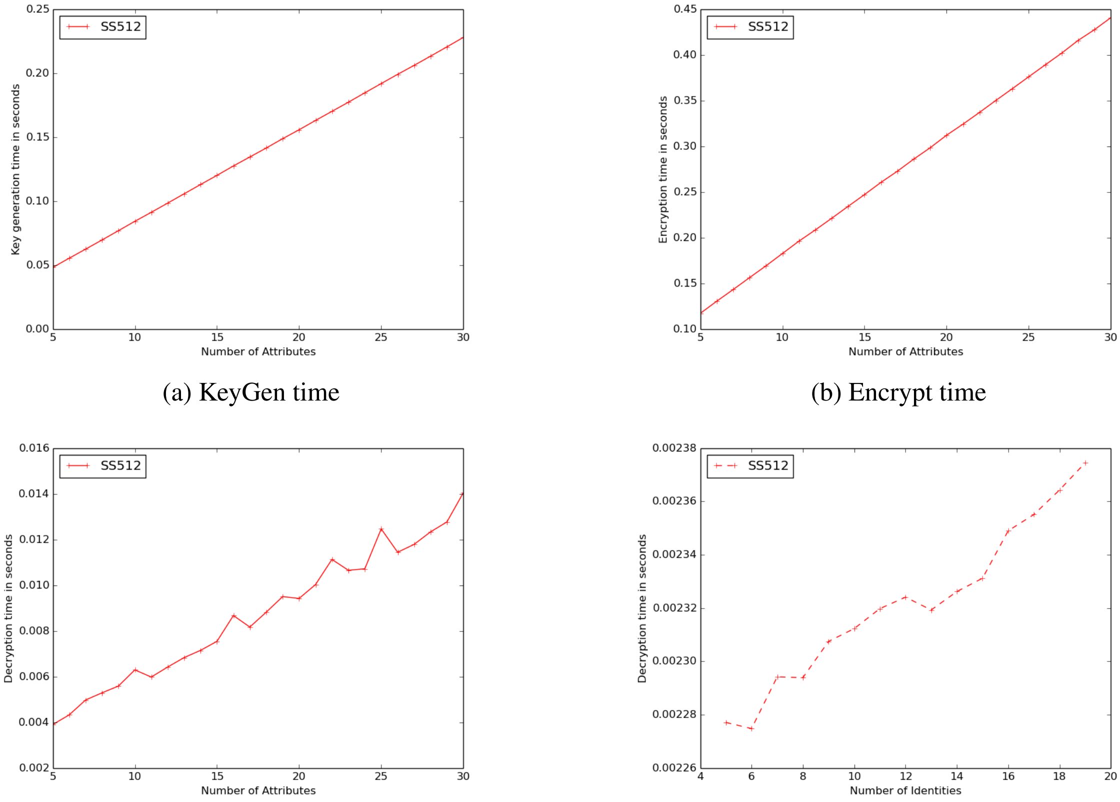
We test on SS512-type elliptic curves with symmetric bilinear pairings, 512 bytes plaintext, AES-256 symmetric encryption algorithm, and the number of attributes and identities are chosen from 5 to 30 and from 5 to 15, respectively. Figure 2(a) illustrates the relationship between the running time for attribute-based private key generation and the number of attributes. Figure 2(b) illustrates the relationship between the running time for encryption and the number of attributes, where we fix the number of receivers 15. Figure 2(c) illustrates the relationship between the running time for decryption for a PHR user in the PUD and the number of attributes. Figure 2(d) illustrates the relationship between the running time for decryption for a user in the PSD and the number of designated receivers.

5 Conclusion

In this paper, we propose a novel patient-centric framework for secure sharing of personal health records in cloud computing. It allows patients to securely store their health data on the semi-trusted cloud service providers, and to selectively share their health data with a wide range of users, including health care professionals such as doctors and nurses, family members or friends. To reduce the key management complexity for patients and users, we divide the users into public domain and personal domain. Different from existing cloud-based personal health record system, patients encrypt their health data for the public domain using ciphertext-policy attribute-based encryption scheme, and encrypt their health data for the personal domain using anonymous multi-receiver identity-based encryption scheme in our cloud-based personal health record system. Extensive analytical and experimental results show that our cloud-based personal health record system is secure, privacy-protected, scalable and efficient. In future work we will design cloud-based personal health record system supporting efficient data utilization services, such as data retrieval and data statistics.

Acknowledgement

This paper is jointly supported by the National Natural Science Foundation of China (Grant No. 61173189), the



(a) KeyGen time

(b) Encrypt time

(c) Decrypt time for PUD users

(d) Decrypt time for PSD users

Figure 2: Performance test of CB-PHR system.

Foundation for Innovative Research Team of Yunnan University, Guangdong Province Information Security Key Laboratory Project, Yunnan Province Software Engineering Key Laboratory Project (Grant No. 2015SE203).

References

[1] D. Boneh and M. Franklin (2001) Identity-based encryption from the Weil pairing, *CRYPTO 2001*, LNCS 2139, Springer Berlin Heidelberg, pp. 213–229.

[2] J. Baek, R. Safavi-Naini and W. Susilo (2005) Efficient Multi-receiver Identity-Based Encryption and Its Application to Broadcast Encryption, *PKC 2005*, LNCS 3386, Springer Berlin Heidelberg, pp.380–397.

[3] X. Boyen and B. Waters (2006) Anonymous hierarchical identity-based encryption (without random oracles), *CRYPTO 2006*, LNCS 4117, Springer Berlin Heidelberg, pp. 290–307.

[4] C.I. Fan, L.Y. Huang and P.H. Ho (2010) Anonymous multireceiver identity-based encryption, *IEEE Transactions on Computers*, Vol. 59, No. 9, pp. 1239–1249.

[5] H.Y. Chien (2012) Improved anonymous multi-receiver identity-based encryption, *The Computer Journal*, Vol. 55, No. 4, pp. 439–445.

[6] Y.M. Tseng, Y.H. Huang and H.J. Chang (2012) CCA-secure anonymous multi-receiver ID-based encryption, *26th International Conference on Advanced Information Networking and Applications Workshops*, IEEE, pp. 177–182.

[7] A. Sahai and b. Waters (2005) Fuzzy identity-based encryption, *EUROCRYPT 2005*, LNCS 3494, Springer Berlin Heidelberg, pp. 457–473.

[8] V. Goyal, O. Pandey, A. Sahai and B. Waters (2006) Attribute-based encryption for fine-grained access control of encrypted data, *CCS 2006*, ACM, New York, pp. 89–98.

[9] J. Bethencourt, A. Sahai and B. Waters (2007) Ciphertext-policy attribute-based encryption, *IEEE Symposium on Security and Privacy*, IEEE, pp. 321–334.

[10] B. Waters (2011) Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization, *PKC 2011*, LNCS 6571, Springer Berlin Heidelberg, pp. 53–70.

[11] J. Li, Q. Wang, C. Wang and R. Kui (2011) Enhancing attribute-based encryption with attribute hierarchy, *Mobile Network Application*, Vol. 16, No. 5, pp. 553–561.

[12] C.J. Wang and J.F. Luo (2013) An efficient key-policy attribute-based encryption scheme with constant ciphertext length, *Mathematical Problems in Engineering*, Hindawi, Vol. 2013, pp. 1–7.

[13] J. Li, X.Y. Huang, J.W. Li, X.F. Chen and Y. Xiang (2014) Securely outsourcing attribute-based encryption with checkability, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 25, No. 8, pp. 2201–2210.

[14] L. Ibraimi, M. Asim and M. Petkovic (2009) Secure management of personal health records by applying attribute-based encryption, *6th International Workshop on Wearable Micro and Nano Technologies for Personalized Health (pHealth)*, IEEE, pp. 71–74.

[15] M. Li, S.C. Yu, Y. Zheng, K. Ren and W.J. Lou (2013) Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 24, No. 1, pp. 131–143.

[16] T. Okamoto and D. Pointcheval (2001) REACT: rapid enhanced-security asymmetric cryptosystem transform, *CT-RSA 2001*, LNCS 2020, Springer Berlin Heidelberg, pp. 159–174.

[17] E. Fujisaki and T. Okamoto (2011) Secure integration of asymmetric and symmetric encryption schemes, *Journal of Cryptology*, Vol. 26, No. 1, pp. 80–101.

[18] A. Beimel (1996) Secure schemes for secret sharing and key distribution, *PhD Thesis*, Israel Institute of Technology, Technion, Haifa, Israel.

[19] J.A. Akinyele, et al. (2013) Charm: a framework for rapidly prototyping cryptosystems, *Journal of Cryptographic Engineering*, Vol. 3, No. 2, pp. 111-128.

[20] M. Green and J.A. Akinyele (2014) The functional encryption library, *Online, accessed 18-July-2014*, `http://code.google.com/p/libfenc/`.

[21] E. Young and T. Hudson (2014) The openssl project, *Online, accessed 18-July-2014*, `http://www.openssl.org/`.

[22] B.Lynn (2014) The pairing-based cryptography library, *Online, accessed 18-July-2014*, `http://crypto.stanford.edu/pbc/`.

Advances in the Field of Automated Essay Evaluation

Kaja Zupanc and Zoran Bosnić
University of Ljubljana, Faculty of Computer and Information Science, Ljubljana, Slovenia
E-mail: {kaja.zupanc, zoran.bosnic}@fri.uni-lj.si

Overview paper

Keywords: automated essay evaluation, automated scoring, natural language processing

Received: June 16, 2015

Automated essay evaluation represents a practical solution to a time-consuming activity of manual grading of students' essays. During the last 50 years, many challenges have arisen in the field, including seeking ways to evaluate the semantic content, providing automated feedback, determining validity and reliability of grades and others. In this paper we provide comparison of 21 state-of-the-art approaches for automated essay evaluation and highlight their weaknesses and open challenges in the field. We conclude with the findings that the field has developed to the point where the systems provide meaningful feedback on students' writing and represent a useful complement (not replacement) to human scoring.

Povzetek: Avtomatsko ocenjevanje esejev predstavlja praktično rešitev za časovno potratno ročno ocenjevanje študentskih esejev. V zadnjih petdesetih letih so se na področju avtomatskega ocenjevanja esejev pojavili mnogi izzivi, med drugim ocenjevanje semantike besedila, zagotavljanje avtomatske povratne informacije, ugotavljanje veljavnosti in zanesljivosti ocen in ostale. V članku primerjamo 21 aktualnih sistemov za avtomatsko ocenjevanje esejev in izpostavimo njihove slabosti ter odprte probleme na tem področju. Zaključimo z ugotovitvijo, da se je področje razvilo do te mere, da sistemi ponujajo smiselno povratno informacijo in predstavljajo koristno dopolnilo (in ne zamenjavo) k človeškemu ocenjevanju.

1 Introduction

Essays are a short literary composition on a particular theme or subject, usually in prose and generally analytic, speculative, or interpretative. Researchers consider essays as the most useful tool to assess learning outcomes. Essays give students an opportunity to demonstrate their range of skills and knowledge, including higher-order thinking skills, such as synthesis and analysis [62]. However, grading students' essays is a time-consuming, labor-intensive and expensive activity for educational institutions. Since teachers are burdened with hours of grading of written assignments, they assign less essays, thereby limiting the needed experience to reach the writing goals. This contradicts the aim to make students better writers, for which they need to rehearse their skill by writing as much as possible [44].

A practical solution to many problems associated with manual grading is to have an automated system for essay evaluation. Shermis and Burstein [53] define an automated essay evaluation (AEE) task as *the process of evaluating and scoring the written prose via computer programs*. AEE is a multi-disciplinary field that incorporates research from computer science, cognitive psychology, educational measurement, linguistics, and writing research [54]. Researchers from all these fields are contributing to the development of the field: computer scientists are developing attributes and are implementing AEE systems, writing scientists and teachers are providing constructive criticisms to the development, and cognitive psychologists expert opinion is considered when modeling the attributes. Psychometric evaluations provide crucial information about the reliability and validity of the systems, as well.

In Figure 1 we illustrate the procedure of automated essay evaluation. As shown in the figure, most of the existing systems use a substantially large set of *prompt*-specific essays (i.e. set of essays on the same topic). Expert human graders score these essays with scores e.g. from 1 to 6, to construct the learning set. This set is used to develop the scoring model of the AEE system and attune it. Using this scoring model (which is shown as the black box in Figure 1), the AEE system assigns scores to new, ungraded essays. The performance of the scoring model is typically validated by calculating how well the scoring model "replicated" the scores assigned by the human expert graders [18].

Automated essay evaluation has been a real and viable alternative, as well as a complement to human scoring, in the last 50 years. The widespread development of the Internet, word processing software, and natural language processing (NLP) stimulated the later development of AEE systems. Motivation for the research in the field of automated evaluation was first focused on time and cost savings, but in the recent years the focus of the research has moved to development of attributes addressing the *writing construct* (i.e. various aspects of writing describing "what"

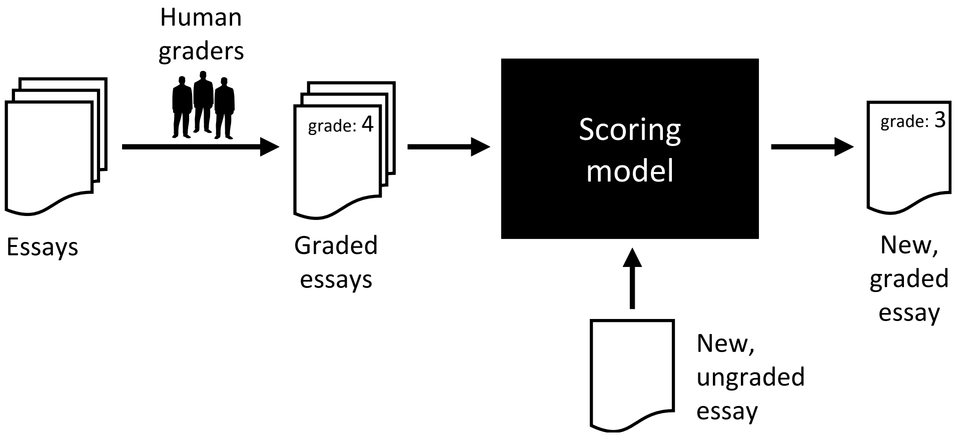


Figure 1: Illustration of the automated essay evaluation: A set of essays is pre-scored by human graders and used to develop the scoring model. This scoring model is used to assign the scores to new, ungraded essays.

and "how" the students are writing). Researchers are also focusing on providing comprehensive feedback to students, evaluating the semantic content, developing AEE systems for other languages (in addition to English), and increasing the validity and reliability of AEE systems.

In this survey we make a comprehensive overview of the latest development in the field. In Section 2 we first describe the reasons and progress of the field development in the last 50 years. Then we present advantages and disadvantages of AEE systems and provide an overview of open problems in the field in Section 3. Section 4 briefly describes the field of NLP and then overview the existing commercial and publicly-available AEE systems. This is followed by a comparison of those approaches. Section 5 concludes the paper.

2 History

Throughout the development of the field, several different names have been used for it interchangeably. The names automated essay scoring (AES) and automated essay grading (AEG) were slowly replaced with the term automated writing evaluation (AWE) or automated essay evaluation (AEE). The term *evaluation* within the name (AEE) came to use because the systems are expected also to provide feedback about linguistic properties that are related to writing quality, interaction, and altogether wider range of possibilities for software.

The first AEE system was proposed almost 50 years ago. In 1966, the former high school English teacher E. Page [44] proposed machine scoring technology and initiated the development of the field. In 1973 [1] he had enough hardware and software at his disposal to implement the first AEE system under the name Project Essay Grade. The first results were characterized as remarkable as the system's performance had more steady correlation with human graders than the performance of two trained human graders. Despite its impressive success at predicting teachers' essay ratings, the early version of the system received only limited acceptance in writing and education community.

By the 1990s, with the widespread of the Internet, natural language processing tools, e-learning systems, and statistical methods, the AEE became a support technology in education. Nowadays, the AEE systems are used in combination with human graders in different high-stakes assessments such as the Graduate Record Examination (GRE), Test of English as a Foreign Language (TOEFL), Graduate Management Admissions Test (GMAT), SAT, American College Testing (ACT), Test of English for International Communication (TOEIC), Analytic Writing Assessment (AWA), No Child Left Behind (NCLB) and Pearson Test of English (PTE). Furthermore, some of them also act as a sole grader.

E-rater was the first system to be deployed in a high-stakes assessment in 1999 [49]. It provided one of two scores for essays on the writing section of the Graduate Management Admissions Test (GMAT). The second score for each essay was provided by an expert human grader. The term "*blended*" scoring model [35, 36] for the use of both human and machine scoring for a single assessment program, came to use at the time.

3 Challenges in the field of automated essay evaluation

In addition to savings of time and money, AEE systems provide higher degree of feedback tractability and score logic for a specific response Feedback for each specific response provides information on quality of different aspects of writing, as partial score as well as descriptive feedback. Their constant availability for scoring gives a possibility to students to repetitively practice their writing at any time. AEE systems are reliable and consistent as they predict the same score for a single essay each time that essay is input to the system. This is important since the scoring consistency between prompts turned out to be one of the most difficult psychometric issues in human scoring [3].

In the following, we provide an overview of open problems and challenges in the field of AEE.

3.1 Validation, validity, and reliability

The terms validation and validity have two related yet distinct meanings in the measurement of student learning [30]. Validation is defined as the *accumulation of evidence* to support interpretation and use of the proposed test score, and validity is a *teacher's judgement that the validation evidence is sufficient* to warrant the proposed interpretation and use [48]. Reliability, on the other hand, is concerned with consistency of test scores and is based on the idea that the observed score on a test is only one possible result that might have been obtained under different situations [3]. Reliability contributes to the validity argument because it provides evidence on the repeatability of scores across different measurement conditions.

Authors of existing AEE systems demonstrate the validity of their systems by correlating their output with expert human scores. Many authors have proven that AEE systems produce reliable and valid essay scores when compared with expert human graders [4, 55], but as stated in [30,47,66] this is only a part of what would be required for an overall validity argument.

A recent work of Attali [3] emphasized two difficulties regarding the validation of AEE systems that derive from the fact that AEE has always been conceived as a simulation of human grading. Thus it is necessary to show that machine scores measure the same construct as human ratings. The contradiction comes from the fact that AEE should replace the human graders but at the same time cannot truly understand an essay.

When taking expert human scores as the "resolved score" (final score acquired from more than one human score), researchers in the field of AEE often appear to operate under the assumption that humans do not make mistakes. In reality, human graders are inconsistent and unreliable, biased scoring is thought to be due to various aspects of reader characteristics (teaching, rating and content experiences), reader psychology (factors that occur internally to the reader), and rating environment (including pressure) [8]. Systematic human errors introduce construct-irrelevant variance into scores and therefore impact their validity [35]. The solution lies in redefining the purpose of AEE, which would be rather to serve as a complement (instead of replacement) to human scoring [3].

Attali [3] proposed a list of the following steps for validation of AEE systems:

1. Validating attributes - establishing the elements of writing construct that an AEE system measures.

2. Analysing implications of different aggregation approaches on the meaning of essay scores when combining attributes into essay scores.

3. Considering combining human and machine scores - incorporation of AEE scores into assessment program.

With these 3 steps Attali [3] proposed AES validation that proceeds first by clarifying the construct it can measure independently of what humans measure, and only then evaluate the similarity in the measured constructs.

3.2 Evaluation of semantic content

Existing AEE systems use a variety of attributes to describe essay features including grammar, mechanics (e.g. spellchecking errors, capitalization errors, punctuation errors), content, lexical sophistication, style, organization, and development of content. However, lack of consideration of text semantics is one of their main weaknesses. The systems evaluate content by comparing the vocabulary of the new essay with already scored essays, and by evaluating the discourse elements (e.g. title, introductory material, thesis, main idea, supporting idea, conclusion) using specifically designed attributes. Some systems use latent semantic analysis (LSA) [31], latent Dirichlet allocation (LDA) [28], and content vector analysis (CVA) [2] to evaluate the semantic of the essay.

The major limitation of LSA is that it only retains the frequency of words by disregarding the word sequence, and the syntactic and semantic structure of texts. An improvement of LSA that considers semantics by means of the syntactic and shallow semantic tree kernels was proposed in 2013 [13]. Experiments suggest that syntactic and semantic structural information can significantly improve the performance of the models for automated essay evaluation. However, only two existing systems [6, 19] use approaches that partially check if the statements in the essays are correct. Despite the efforts, these systems are not automatic, as they require manual interventions from the user. None of the existing systems is therefore capable of assessing the correctness of the given common sense facts.

3.3 Evaluation methodology

For evaluation of performance of essay grading systems, a variety of common metrics is being used, such as Pearson's and Spearman's correlation, exact and adjacent degree of agreement, precision, recall, F-measure, and the kappa metric. Since there are no specialised metrics in the field of AEE, Shermis and Hammer [57] observed several of them in their works:

- correspondence in mean and standard deviations of the distributions of human scores to that of automated scores,

- agreement (reliability) statistics measured by correlation, weighted kappa and percent agreement (exact and exact + adjacent), and

- degree of difference (delta) between human-human agreement and automated-human agreement by the same agreement metrics as above.

In the public competition on Kaggle (see Section 4.3), the **quadratic weighted kappa** was used, which also became the prevalent evaluation measure used for AEE systems. Quadratic weighted kappa is an error metric that measures the degree of agreement between two graders (in case of AEE this is an agreement between the automated scores and the resolved human scores) and is an analogy to the correlation coefficient. This metric typically ranges from 0 (random agreement between graders) to 1 (complete agreement between graders). In case that there is less agreement between the graders than expected by chance, this metric may go below 0. Assuming that a set of essay responses E has S possible ratings, $1, 2, \ldots, S$, and that each essay response e is characterized by a tuple (e_A, e_B) which corresponds to its scores by grader A and grader B, the metric is calculated as follows:

$$\kappa = 1 - \frac{\sum_{i,j} w_{i,j} O_{i,j}}{\sum_{i,j} w_{i,j} E_{i,j}} \tag{1}$$

where w are weights, O is a matrix of observed ratings and E is a matrix of expected ratings. Matrix of weights w_{ij} is an S-by-S matrix that is calculated based on the difference between graders' scores, such that

$$w_{i,j} = \frac{(i-j)^2}{(S-1)^2}, \tag{2}$$

O is an S-by-S histogram (agreement) matrix of *observed* ratings, which is constructed over the essay ratings, such that $O_{i,j}$ corresponds to the number of essays that received a rating i by grader A and a rating j by grader B; analogously, E is an S-by-S histogram matrix of *expected* ratings:

$$E_{i,j} = \frac{H_{Ai} \cdot H_{Bj}}{N} \tag{3}$$

where $H_{Ai}, i = 1, \ldots, S$ denotes the number of essays that grader A scored with score i, and N is a number of gradings or essays. E is normalized with N such that E and O have the same sum.

3.4 Unavailability of data sets

The public availability of the experimental data sets would accelerate the progress in the AEE field. This would allow the researchers and organizations to compare their systems with others on the same data sets with the same evaluation methodology. Currently, the only publicly available data set is the Kaggle competition data set [57] (see Section 4.3).

3.5 Language dependency

Although most of the AEE research has been conducted for English, researchers have applied the technology also to some other languages:

- French - Assistant for Preparing EXams (Apex) [33],
- Hebrew [(Vantage Learning, 2001) as cited in [54]],
- Bahasa Malay [(Vantage Learning, 2002) as cited in [54]],
- Maleysian [61],
- Japanese - Japanese Essay Scoring System (JESS) [24, 25],
- German [64],
- Finnish - Automatic Essay Assessor (AEA) [28, 29],
- Chinese [14, 16],
- Spanish and Basque [12],
- Arabic [42] and
- Swedish [43].

The requirements for AEE systems are the same for other languages. However, the major problem is lack of the NLP tools for different languages which are the main component of AEE systems. The complexity of development of such tools is associated with the complexity of individual languages. Another reason for slower development is also the much bigger number of people and researchers using and speaking English than other languages.

3.6 Tricking the system

The state-of-the-art systems include detection of advisories that point out the inappropriate and unorthodox essays, for example if an essay is has problems with discourse structure, includes a large number of grammatical errors, contains text with no lexical content, consists of copied text, or is off-topic and does not respond to the essay question. Detecting such essays is important from the perspective of validity.

Powers et al. [46] studied tricking the e-rater system. The best score from the set of inappropriate essays received an essay that repeated the same paragraph 37 times. Later Herrington and Moran [22] as well as McGee [41] tested the accuracy of e-rater and Intelligent Essay Assessor, respectively. They submitted multiple drafts and were able to make such revisions to essays that the systems would assign them high scores. They were quickly able to figure out how the systems "read" the essays and submitted essays that satisfied these criteria.

With the development attributes that address a wide range of aspects of writing, tricking the system became a non-trivial process.

3.7 Automated feedback

AEE systems can recognize certain types of errors, including syntactic errors, and offer automated feedback on correcting these errors. In addition, the systems can also provide global feedback on content and development. Automated feedback reduces teachers' load and helps students become more autonomous. In addition to numerical score such feedback provides a meaningful explanation by suggesting improvements. Systems with feedback can be an aid, not a replacement, for classroom instruction. Advantages of automated feedback are its anonymity, instantaneousness, and encouragement for repetitive improvements by giving students more practice for writing essays [63].

The current limitation of the feedback is that its content is limited to the completeness or correctness of the syntactic aspect of the essay. Some attempts have been made [6, 19] to include also semantic evaluation, but these approaches are not automatic and work only partially.

4 Automated essay evaluation systems

This section provides an overview of the state-of-the-art AEE systems. First we briefly describe the field of NLP that has influenced the growing development of the AEE systems in the last 20 years the most. This is followed by the presentation of proprietary AEE systems developed by commercial organizations as well as two publicly-available systems and approaches proposed by the academic community. We conclude this section with a comparison of described systems.

4.1 Natural language processing

Natural language processing is a computer-based approach for analyzing language in text. In [34] it is defined as "*a range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of task applications*". This complex definition can be fractionated for better understanding: "The range of computational techniques" relates to the numerous approaches and methods used for each type of language analysis; and "Naturally occurring texts" describes the diversity of texts, i.e. different languages, genres, etc. The primary requirement of all NLP approaches is that the text is in a human understandable language.

Research in the field started in the 1940s [27]. As many other fields of computer science, the NLP field began growing rapidly in the 1990 along with the increased availability of electronic text, computers with high speed and high memory capabilities, and the Internet. New statistical and rule-based methods allowed researchers to carry out various types of language analyse, including analyses of syntax (sentence structure), morphology (word structure), and semantics (meaning) [11]. The state-of-the-art approaches include automated grammatical error detection in word processing software, Internet search engines, machine translation, automated summarization, and sentiment analysis.

As already mentioned, NLP methods played the crucial role in the development of AEE technologies, such as: part of speech tagging (POS), syntactic parsing, sentence fragmentation, discourse segmentation, named entity recognition, and content vector analysis (CVA).

4.2 AEE systems

Until recently, one of the main obstacles to achieve progress in the field of AEE has been lack of open-source AEE systems, which would provide insight into their grading methodology. Namely, most of the AEE research has been conducted by commercial organizations that have protected their investments by restricting access to technological details. In the last couple of years there were several attempts to make the field more "exposed" including recently published Handbook of Automated Essay Evaluation [56].

In this section we describe the majority of systems and approaches. We overview the systems that have predominance in this field - and are consequently more complex and have attracted greater publicity. All systems work by extracting a set of attributes (system-specific) and using some machine learning algorithm to model and predict the final score.

- **Project Essay Grade (PEG)**
  PEG is a proprietary AES system developed at Measurement Inc. [44]. It was first proposed in 1966, and in 1998 a web interface was added [58]. The system scores essays through measuring *trins* and *proxes*. A *trin* is defined as an intrinsic higher-level variable, such as punctuation, fluency, diction, grammar etc., which as such cannot be measured directly and has to be approximated by means of other measures, called *proxes*. For example, the trin *punctuation* is measured through the proxes *number of punctuation errors* and *number of different punctuations used*. The system uses regression analysis to score new essays based on a training set of 100 to 400 essays [45].

- **e-rater**
  E-rater is a proprietary automated essay evaluation and scoring system developed at the Educational Testing Service (ETS) in 1998 [10]. E-rater identifies and extracts several attribute classes using statistical and rule-based NLP methods. Each attribute class may represent an aggregate of multiple attributes. The attribute classes include the following [4, 9]: (1) grammatical errors (e.g. subject-verb agreement errors), (2) word usage errors (e.g. their versus there), (3) errors in writing mechanics (e.g. spelling), (4) presence of essay-based discourse elements (e.g. thesis statement, main points, supporting details, and conclusions), (5) development of essay-based discourse elements, (6) style weaknesses (e.g. overly repetitious words), (7) two content vector analysis (CVA)-based

attributes to evaluate topical word usage, (8) an alternative, differential word use content measure, based on the relative frequency of a word in high scoring versus low-scoring essays, (9) two attributes to assess the relative sophistication and register of essay words, and (10) an attribute that considers correct usage of prepositions and collocations (e.g., powerful computer vs. strong computer), and variety in terms of sentence structure formation. The set of ten attribute classes represent positive attributes, rather than number of errors. The system uses regression modeling to assign a final score to the essay [11]. E-rater also includes detection of essay similarity and advisories that point out if an essay is off topic, has problems with discourse structure, or includes large number of grammatical errors [23].

- **Intelligent Essay Assessor (IEA)**
  In 1998 the Pearson Knowledge Technologies (PKT) developed Intelligent Essay Assessor (IEA). The system is based on the Latent Semantic Analysis (LSA), a machine-learning method that acquires and represents knowledge about meaning of words and documents by analyzing large bodies of natural text [32]. IEA uses LSA to derive attributes describing content, organization, and development-based attributes of writing. Along with LSA, IEA also uses NLP-based measures to extract attributes measuring lexical sophistication, grammatical, mechanical, stylistic, and organizational aspects of essays. The system uses approximately 60 attributes to measure above aspects within essays: content (e.g. LSA essay semantic similarity, vector length), lexical sophistication (e.g. word maturity, word variety, and confusable words), grammar (e.g. n-gram attributes, grammatical errors, and grammar error types), mechanics (e.g. spelling, capitalization, and punctuation), style, organization, and development (e.g. sentence-sentence coherence, overall essay coherence, and topic development). IEA requires a training with a representative sample (between 200 and 500) of human-scored essays.

- **IntelliMetric**
  IntelliMetric was designed and first released in 1999 by Vantage Learning as a proprietary system for scoring essay-type, constructed response questions [51]. The system analyzes more than 400 semantic-, syntactic-, and discourse-level attributes to form a composite sense of *meaning*. These attributes can be divided into two major categories: content (discourse/rhetorical and content/concept attributes) and structure (syntactic/structural and mechanics attributes). The content attributes evaluate the topic covered, the breadth of content, and support for advanced concepts, cohesiveness and consistency in purpose and main idea, and logic of discourse. Whereas structure attributes evaluate grammar, spelling, capitalization, sentence completeness, punctuation, syntactic variety, sentence complexity, usage, readability, and subject-verb agreement [51]. The system uses multiple predictions (called judgements) based on multiple mathematical models, including linear analysis, Bayesian, and LSA to predict the final score and combines the models into a single final essay score [49]. Training Intellimetric requires a sample of at least 300 human-scored essays. IntelliMetric uses Legitimatch technology to identify responses that appear off topic, are too short, do not conform to the expectations for edited American English, or are otherwise inappropriate [51].

- **Bookette**
  Bookette [48] was designed by California Testing Bureau (CTB) and became operational in classroom settings in 2005 and in large-scale testing settings in 2009. Bookette uses NLP to derive about 90 attributes describing student-produced text. Combinations of these attributes describe traits of effective writing: organization, development, sentence structure, word choice/grammar usage, and mechanics. The system uses neural networks to model expert human grader scores. Bookette can build prompt-specific models as well as generic models that can be very useful in classrooms for formative purposes. Training Bookette requires a set (from 250 to 500) of human-scored essays. Bookette is used in CTB's solution Writing Roadmap 2.0, in West Virginia's summative writing assessment known as Online Writing Assessment (OWA) program and in their formative writing assessment West Virginia Writes. The system provides feedback on students writing performance that includes both holistic feedback and feedback at the trait level including comments on the grammar, spelling, and writing conventions at the sentence level [48].

- **CRASE**
  Pacific Metrics proprietary automated scoring engine, CRASE [35], moves through three phases of the scoring process: identifying inappropriate attempts, attribute extraction, and scoring. The attribute extraction step is organized around six traits of writing: ideas, sentence fluency, organization, voice, word choice, conventions, and written presentation. The system analyzes a sample of already-scored student responses to produce a model of the graders' scoring behaviour. CRASE is a Java-based application that runs as a web service. The system is customizable with respect to the configurations used to build machine learning models as well as the blending of human and machine scoring (i.e., deriving hybrid models) [35]. Application also produces text-based and numeric-based feedback that can be used to improve the essays.

- **AutoScore**
  AutoScore is a proprietary AEE system designed by

the American Institute for Research (AIR). The system analyzes measures based on concepts that discriminate between high- and low- scored papers, measures that indicate the coherence of concepts within and across paragraphs, and a range of word-use and syntactic measures. Details about the system were never published, however, the system was evaluated in [57].

– **Lexile Writing Analyzer**
  The Lexile Writing Analyzer is a part of The Lexile Framework for Writing [59] developed by MetaMetrics. The system is score-, genre-, prompt-, and punctuation-independent and utilizes the Lexile writer measure, which is an estimate of student's ability to express language in writing, based on factors related to semantic complexity (the level of words used) and syntactic sophistication (how the words are written into sentences). The system uses a small number of attributes that represent approximations for writing ability. Lexile perceives writing ability as an underlying individual trait. Training phase is not needed since a vertical scale is employed to measure student essays [60].

– **SAGrader**
  SAGrader is an online proprietary AEE system developed by IdeaWorks, Inc. [7]. The system was first known under the name Qualrus. SAGrader blends a number of linguistic, statistical, and artificial intelligence approaches to automatically score the essay. The operation of the SAGrader is as follows: The instructor first specifies a task in a prompt. Then the instructor creates a rubric identifying the "desired features" – key elements of knowledge (set of facts) that should be included in a good response, along with relationships among those elements – using a semantic network (SN). Fuzzy logic (FL) permits the program to detect the features in the students' essays and compare them to desired ones. Finally, an expert system scores student essays based on the similarities between the desired and observed features [6]. Students receive immediate feedback indicating their scores along with the detailed comments indicating what they did well and what needs further work.

– **OBIE based AEE System**
  The AEE system proposed by Gutierrez et al. [20,21] provides both, scores and meaningful feedback, using ontology-based information extraction (OBIE). The system uses logic reasoning to detect errors in a statement from an essay. The system first transforms text into a set of logic clauses using open information extraction (OIE) methodology and incorporates them into domain ontology. The system determines if these statements contradict the ontology and consequently the domain knowledge. This method considers incorrectness as inconsistency with respect to the domain. Logic reasoning is based on the description logic (DL) and ontology debugging [19].

– **Bayesian Essay Test Scoring sYstem (BETSY)**
  The first scoring engine to be made available publicly was Rudner's Bayesian Essay Test Scoring sYstem (BETSY) [50]. BETSY uses multinomial or Bernoulli Naïve Bayes models to classify texts into different classes (e.g. pass/fail, scores A-F) based on content and style attributes such as word unigrams and bigrams, sentence length, number of verbs, noun–verb pairs etc. Classification is based on assumption that each attribute is independent of another. Conditional probabilities are updated after examining each attribute. BETSY worked well only as a demonstration tool for a Bayesian approach to scoring essays. It remained a preliminary investigation as the authors never continued with their work.

– **LightSIDE**
  Mayfield and Rosé released LightSIDE [38], an easy-to-use automated evaluation engine. LightSIDE made very important contribution to the field of AEE by publicly providing compiled and source code. This program is designed as a tool for non-experts to quickly utilize text mining technology for a variety of purposes, including essay assessment. It allows choosing what set of attributes is best suited to represent the text. LightSIDE offers a number of algorithms to perform learning mappings between attributes and the final score (e.g. linear regression, Naïve Bayes, linear support vector machines) [39].

– **Semantic Automated Grader for Essays (SAGE)**
  SAGE, proposed by Zupanc and Bosnić [67], evaluates coherence of student essays. The system extracts linguistic attributes using statistical and rule-based NLP methods, and content attributes. The novelty of the system is a set of semantic coherence attributes measuring changes between sequential essay parts from three different perspectives: semantic distance (e.g. distance between consecutive parts of an essay, maximum distance between any two parts), central spatial tendency/dispersion, and spatial autocorrelation in semantic space. These attributes allow better evaluation of local and global essay coherence. Using the random forests and extremely randomized trees the system builds regression models and grades unseen essays. The system achieves better prediction accuracy than 9 state-of-the-art systems evaluated in [57].

– **Use of Syntactic and Shallow Semantic Tree Kernels for AEE**
  Chali and Hasan [13] exposed the major limitation of LSA - it only retains the frequency of words by disregarding the word sequence and the syntactic and semantic structure of texts. They proposed the use of

syntactic and shallow semantic tree kernels for grading essays as a substitute to LSA. The system calculates the syntactic similarity between two sentences by parsing the corresponding sentences into syntactic trees and measuring the similarity between the trees. Shallow Semantic Tree Kernel (SSTK) method allows to match portions of a semantic trees. The SSTK function yields the similarity score between a pair of sentences based on their semantic structures.

- **A Ranked-based Approach to AEE**
  Chen et al. [15] consider the problem of AEE as a ranking problem instead of classification or regression problem. Ranking algorithms are a family of supervised learning algorithms that automatically construct a ranking model of the retrieved essays. They consider the following three groups of attributes: term usage, sentence quality, and content fluency and richness. Authors showed that in AES learning to rank outperforms other classical machine learning techniques.

- **OzEgrader**
  OzEgrader is an Australian AES system proposed by Fazal et al. [18]. Grading process considers different aspects of content and style: audience, text structure, character and setting, paragraphing, vocabulary, sentence structure, punctuation, spelling, cohesion and ideas. Techniques such as POS tagging, named entity recognition, artificial neural networks, and fuzzy regression are employed in order to model linear or non-linear relationships between attributes and the final score. The system also includes the methodology for noise reduction in the essay dataset.

- **AEE using Generalized LSA**
  Islam and Hoque [26] developed an AEE system using Generalized Latent Semantic Analysis (GLSA) which makes *n-gram by document* matrix instead of *word by document* matrix as used in LSA. The system uses the following steps in grading procedure: preprocessing the training essays, stopword removal, word stemming, selecting the n-gram index terms, *n-gram by document* matrix creation, computation of the singular value decomposition (SVD) of *n-gram by document* matrix, dimensionality reduction of the SVD matrices, and computation of the similarity score. The main advantage of GLSA is observance of word order in sentences.

- **AEE using Multi-classifier Fusion**
  Bin and Jian-Min [5] proposed an approach to AEE using multi-classifier Fusion. The system first represents each essay by the vector space model and removes stopwords. Then it extracts the attributes describing content and linguistic from the essays in the form of attribute vector where each vector is expressed by corresponding weight. Three approaches including document frequency (DF), information gain (IG) and chi-square statistic (CHI) are used to select attributes by some predetermined thresholds. The system classifies an essay to an appropriate category using different classifiers, such as naive Bayes, k-nearest neighbors and support vector machine. Finally, the ensemble classifier is combined by those component classifiers.

- **Markit**
  Markit [65] is a proprietary AEE system developed by Blue Wren Software Pty Ltd. The system is capable of running on typical desktop PC platforms. It requires comprehensive knowledge in a form of one model (exemplary) answer against which the student essays are compared. A student essay is processed using a combination of NLP techniques to build the corresponding propriety knowledge representation. Pattern matching techniques (PMT) are then employed to ascertain the proportion of the model answer knowledge that is present in the student's answer, and a score assigned accordingly.

- **PS-ME**
  The Paperless School proprietary AEE system was designed primarily for day-to-day, low stakes testing of essays. The student essay is compared against each relevant master text to derive a number of parameters which reflect knowledge and understanding as exhibited by the student. When multiple master texts are involved in the comparison, each result from an individual comparison gets a weight, which could be negative in the case of a master text containing common mistakes. The individual parameters computed during the analysis phase are then combined in a numerical expression to yield the assignments' score and used to select relevant feedback comments from a comment bank [37].

- **Schema Extract Analyse and Report (SEAR)**
  Christie [17] proposed a software system Schema Extract Analyse and Report (SEAR), which provides the assessment both of style and content. The methodology adopted to assess style is based on a set of common metrics as used by other AES systems. For content assessment the system uses two measures: usage and coverage. Using content schema system measures how much of each essay is included in schema (usage) and how much of schema is used by the essay (coverage).

4.3 Comparison

In the previous section we described many existing AEE systems. In Table 1 we now summarize their key features. We can see that although these systems perform a similar task, each of them uses a different combination of methodologies for attribute extraction and model building. The prevailing methodology used in described systems is NLP. This is consistent with our argument that NLP strongly influenced the development of AEE systems in the last 20

Table 1: Comparison of the key features of AEE systems.

Systems	Developer	Methodology♯	Main focus	Feedback application	# essays required for training	Prediction model	Rank and average accuracy⋆
PEG	Measurement Inc.	Statistical	Style	N/A	100-400	multiple linear regression	2 0.79
e-rater	ETS	NLP	Style and content	Criterion	250	linear regression	3 0.77
IEA	PKT	LSA, NLP	Content	WriteToLearn	200-500	machine learning	9 0.73
IntelliMetric	Vantage Learning	NLP	Style and content	MyAccess!	300	multiple mathematical models	4 0.76
Bookette	CTB	NLP	Style and content	Writing Roadmap 2.0	250-500	neural networks	10 0.70
CRASE	Pacific Metrics	NLP	Style and content	Writing Power	100 per score point	machine learning	6 0.75
AutoScore	AIR	NLP	Style and content	N/A	?	statistical model	8 0.73
Lexile	MetaMetrics	NLP	Style and content	N/A	0	Lexile measure	11 0.63
SAGrader	IdeaWorks	FL, SN	Semantic	SAGrader	0	rule-based expert systems	N/A
OBIE based AEE	University of Oregon	OIE, DL	Semantic	Without name	0	logic reasoner	N/A
BETSY	University of Maryland	Statistical	Style and content	N/A	460*	Bayesian networks	N/A
LightSIDE	Carnegie Mellon University	Statistical	Content	N/A	300	machine learning	7 0.75
SAGE	University of Ljubljana	NLP	Style and content	N/A	800*	random forest	1 0.83
Semantic tree based AEE	University of Lethbridge	LSA, tree kernel functions	Content	N/A	0	cosine similarity	N/A
Ranked-based AEE	GUCAS	NLP	Style and content	N/A	800*	learning to rank	5 0.75
OzEgrader	Curtin University	NLP	Style and content	N/A	?	neural networks	N/A
GLSA based AEE	Bangladesh University	GLSA	Content	N/A	960*	cosine similarity	N/A
Multi-classifier Fusion AEE	Soochow University	NLP, DF, IG, CHI	Style and content	N/A	200*	ensemble classifiers	N/A
Markit	Blue Wren Software Pty Ltd	NLP, PMT	Content	N/A	1 model essay	linear regression	N/A
PS-ME	Paperless School	NLP	Style	N/A	30*	linear regression	N/A
SEAR	Robert Gordon University	Statistical	Style and content	N/A	?	linear regression	N/A

♯ For explanation of abbreviations see Section 4.2.
∗ Data is not available, the number represents the smallest data set in the reported experiments.
⋆ Ranking is based on average accuracy measured with quadratic weighted kappa and reported in [15, 57, 67]

years. The main focus of the systems in the recent years, in addition to evaluation of style, also includes evaluation of content. Many systems consider both and thus provide a holistic score and possibly also feedback for an essay. But most systems could still be characterized as AES systems, only few provide automated feedback and could thus be labelled as AEE systems. Variety of different approaches is used for model building, however machine learning with regression is the prevailing model. The required set of essays for training varies around a few hundreds and is also dependent on prediction model.

In the past, there was a lack of independent studies of AEE systems that have simultaneously compared more than one system (e.g. [40] compared two systems); furthermore, none have included more than three systems. The main cause for this is certainly the commercial origin of many AEE systems. In the end of 2012, the Automated Student Assessment Prize (with funding from the William and Flora Hewlett Foundation) concluded the first larger independent study that involved nine AEE systems (eight commercial vendors and LightSIDE scoring engine) and 8 different data sets [57]. The study included nearly the entire commercial market for automated scoring of essays in the United States [52] and offered a closer look and better understanding of the state-of-the-art approaches. In addition there was a public competition on Kaggle[1] using the same data sets to complement private vendor demonstra-

[1] http://www.kaggle.com/c/ASAP-AES

tion.

Shermis and Hammer [57] reported that two human scores (as measured by quadratic weighted kappas) ranged in rates of agreement from 0.61 to 0.85 and machine scores ranged from 0.60 to 0.84 in their agreement with the human scores. Results of the study for specific system can be seen in Table 1. Two other systems [15, 67] also used the same data set and reported on the prediction accuracy. Unfortunately we were not able to test the rest of the systems on the same data set or use independent data set to compare all of the system, since majority of the systems are proprietary and not publicly available.

5 Conclusion

Development of the automated essay evaluation is important since it enables teachers and educational institutions to save time and money. Moreover it allows students to practice their writing skills and gives them an opportunity to become better writers. From the beginning of the development of the field the unstandardized evaluation process and lack of attributes for describing the writing construct have been emphasized as disadvantages. In the last years, the advancement in the field became faster by the rising number of papers describing publicly available systems that achieve comparable results with other state-of-the-art systems [38].

In this survey we made an overview of the field of automated essay evaluation. It seems that one of the current challenges concerns the meaningful feedback that instructional applications offer to a student. AEE systems can recognize certain types of errors including syntactic errors, provide global feedback on content and development, and offer automated feedback on correcting these errors. Researchers are currently trying to provide a meaningful feedback also about the completeness and correctness of the semantic of the essay. This is closely related to the evaluation of semantic content of student essays, more specifically with the analysis of correctness of the statements in the essays. Another problem concerning the AEE community is the unification of evaluation methodology.

The fact that more and more classical educational approaches has been automatized using computers raises concerns. Massive open online courses (MOOC) have become part of the educational systems and are replacing the traditional teacher - student relation and call into question the educational process in the classrooms. While computer grading of multiple choice tests has been used for years, computer scoring of more subjective material like essays is now moving into the academic world. Automated essay evaluation is playing one of the key roles in the current development of the automated educational systems, including MOOC. All these leaves many open questions regarding the replacement of human teachers with computer, which should be taken into consideration in the future and be answered with the further development of the field.

As a summary of our review, we would like to encourage all the researchers from the field to publish their work as an open-source resources, thereby allow others to compare results. This would contribute to faster development of the field and would consequently lead to novel solutions to the above described challenges.

References

[1] H. B. Ajay, P. I. Tillet, and E. B. Page, “Analysis of essays by computer (AEC-II),” U.S. Department of Health, Education, and Welfare, Office of Education, National Center for Educational Research and Development, Washington, D.C., Tech. Rep., 1973.

[2] Y. Attali, “A Differential Word Use Measure for Content Analysis in Automated Essay Scoring,” *ETS Research Report Series*, vol. 36, 2011.

[3] ——, “Validity and Reliability of Automated Essay Scoring,” in *Handbook of Automated Essay Evaluation: Current Applications and New Directions*, M. D. Shermis and J. C. Burstein, Eds. New York: Routledge, 2013, ch. 11, pp. 181–198.

[4] Y. Attali and J. Burstein, “Automated Essay Scoring With e-rater V . 2,” *The Journal of Technology, Learning and Assessment*, vol. 4, no. 3, pp. 3–29, 2006.

[5] L. Bin and Y. Jian-Min, “Automated Essay Scoring Using Multi-classifier Fusion,” *Communications in Computer and Information Science*, vol. 233, pp. 151–157, 2011.

[6] E. Brent, C. Atkisson, and N. Green, “Time-shifted Collaboration: Creating Teachable Moments through Automated Grading,” in *Monitoring and Assessment in Online Collaborative Environments: Emergent Computational Technologies for E-learning Support*, A. Juan, T. Daradournis, and S. Caballe, Eds. IGI Global, 2010, pp. 55–73.

[7] E. Brent and M. Townsend, “Automated essay grading in the sociology classroom,” in *Machine Scoring of Student Essays: Truth and Consequences?*, P. Freitag Ericsson and R. H. Haswell, Eds. Utah State University Press, 2006, ch. 13, pp. 177–198.

[8] B. Bridgeman, “Human Ratings and Automated Essay Evaluation,” in *Handbook of Automated Essay Evaluation: Current Applications and New Directions*, M. D. Shermis and J. C. Burstein, Eds. New York: Routledge, 2013, ch. 13, pp. 221–232.

[9] J. Burstein, M. Chodorow, and C. Leacock, “Automated Essay Evaluation: The Criterion Online Writing Service,” *AI Magazine*, vol. 25, no. 3, pp. 27–36, 2004.

[10] J. Burstein, K. Kukich, S. Wolff, C. Lu, and M. Chodorow, "Computer Analysis of Essays," in *Proceedings of the NCME Symposium on Automated Scoring*, no. April, Montreal, 1998, pp. 1–13.

[11] J. Burstein, J. Tetreault, and N. Madnani, "The E-rater® Automated Essay Scoring System," in *Handbook of Automated Essay Evaluation: Current Applications and New Directions*, M. D. Shermis and J. Burstein, Eds. New York: Routledge, 2013, ch. 4, pp. 55–67.

[12] D. Castro-Castro, R. Lannes-Losada, M. Maritxalar, I. Niebla, C. Pérez-Marqués, N. C. Álamo Suárez, and A. Pons-Porrata, "A multilingual application for automated essay scoring," in *Advances in Artificial Intelligence – 11th Ibero-American Conference on AI.* Lisbon, Portugal: Springer, 2008, pp. 243–251.

[13] Y. Chali and S. A. Hasan, "On the Effectiveness of Using Syntactic and Shallow Semantic Tree Kernels for Automatic Assessment of Essays," in *Proceedings of the International Joint Conference on Natural Language Processing*, no. October, Nagoya, Japan, 2013, pp. 767–773.

[14] T. H. Chang, C. H. Lee, P. Y. Tsai, and H. P. Tam, "Automated essay scoring using set of literary sememes," in *Proceedings of International Conference on Natural Language Processing and Knowledge Engineering, NLP-KE 2008.* Beijing, China: IEEE, 2008, pp. 1–5.

[15] H. Chen, B. He, T. Luo, and B. Li, "A Ranked-Based Learning Approach to Automated Essay Scoring," in *Proceedings of the Second International Conference on Cloud and Green Computing.* Ieee, Nov. 2012, pp. 448–455.

[16] Y. Chen, C. Liu, C. Lee, and T. Chang, "An Unsupervised Automated Essay- Scoring System," *IEEE Intelligent systems*, vol. 25, no. 5, pp. 61–67, 2010.

[17] J. R. Christie, "Automated Essay Marking – for both Style and Content," in *Proceedings of the Third Annual Computer Assisted Assessment Conference*, 1999.

[18] A. Fazal, T. Dillon, and E. Chang, "Noise Reduction in Essay Datasets for Automated Essay Grading," *Lecture Notes in Computer Science*, vol. 7046, pp. 484–493, 2011.

[19] F. Gutiererz, D. Dou, S. Fickas, and G. Griffiths, "Online Reasoning for Ontology-Based Error Detection in Text," *On the Move to Meaningful Internet Systems: OTM 2014 Conferences Lecture Notes in Computer Science*, vol. 8841, pp. 562–579, 2014.

[20] F. Gutierrez, D. Dou, S. Fickas, and G. Griffiths, "Providing grades and feedback for student summaries by ontology-based information extraction," in *Proceedings of the 21st ACM international conference on Information and knowledge management - CIKM '12*, 2012, pp. 1722–1726.

[21] F. Gutierrez, D. Dou, A. Martini, S. Fickas, and H. Zong, "Hybrid Ontology-based Information Extraction for Automated Text Grading," in *Proceedings of 12th International Conference on Machine Learning and Applications*, 2013, pp. 359–364.

[22] A. Herrington, "Writing to a Machine is Not Writing At All," in *Writing assessment in the 21st century: Essays in honor of Edward M. White*, N. Elliot and L. Perelman, Eds. New York: Hampton Press, 2012, pp. 219–232.

[23] D. Higgins, J. Burstein, and Y. Attali, "Identifying off-topic student essays without topic-specific training data," *Natural Language Engineering*, vol. 12, no. 02, pp. 145–159, May 2006.

[24] T. Ishioka and M. Kameda, "Automated Japanese essay scoring system:jess," *Proceedings. 15th International Workshop on Database and Expert Systems Applications, 2004.*, pp. 4–8, 2004.

[25] T. Ishioka, "Automated Japanese Essay Scoring System based on Articles Written by Experts," in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, no. July, Sydney, 2006, pp. 233–240.

[26] M. M. Islam and A. S. M. L. Hoque, "Automated essay scoring using Generalized Latent Semantic Analysis," *Journal of Computers*, vol. 7, no. 3, pp. 616–626, 2012.

[27] K. S. Jones, "Natural language processing: a historical review," *Linguistica Computazionale*, vol. 9, pp. 3–16, 1994.

[28] T. Kakkonen, N. Myller, E. Sutinen, and J. Timonen, "Comparison of Dimension Reduction Methods for Automated Essay Grading," *Educational Technology & Society*, vol. 11, no. 3, pp. 275–288, 2008.

[29] T. Kakkonen, N. Myller, J. Timonen, and E. Sutinen, "Automatic Essay Grading with Probabilistic Latent Semantic Analysis," in *Proceedings of the second workshop on Building Educational Applications Using NLP*, no. June, 2005, pp. 29–36.

[30] M. T. Kane, "Validation," in *Educational Measurement*, 4th ed., R. L. Brennan, Ed. Westport, CT: Praeger Publishers, 2006, pp. 17–64.

[31] T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse Processes*, vol. 25, no. 2-3, pp. 259–284, Jan. 1998.

[32] T. K. Landauer, D. Laham, and P. W. Foltz, "The Intelligent Essay Assessor," *IEEE Intelligent systems*, vol. 15, no. 5, pp. 27–31, 2000.

[33] B. Lemaire and P. Dessus, "A System to Assess the Semantic Content of Student Essays," *Journal of Educational Computing Research*, vol. 24, no. 3, pp. 305–320, 2001.

[34] E. D. Liddy, "Natural Language Processing," in *Encyclopedia of Library and Information Science*, 2nd ed., M. Decker, Ed. Taylor & Francis, 2001.

[35] S. M. Lottridge, E. M. Schulz, and H. C. Mitzel, "Using Automated Scoring to Monitor Reader Performance and Detect Reader Drift in Essay Scoring." in *Handbook of Automated Essay Evaluation: Current Applications and New Directions*, M. D. Shermis and J. Burstein, Eds. New York: Routledge, 2013, ch. 14, pp. 233–250.

[36] S. M. Lottridge, H. C. Mitzel, and F. Chou, "Blending machine scoring and hand scoring for constructed responses," in *Paper presented at the CCSSO National Conference on Student Assessment*, Los Angeles, California, 2009.

[37] O. Mason and I. Grove-Stephenson, "Automated free text marking with paperless school," in *Proceedings of the Sixth International Computer Assisted Assessment Conference*, 2002, pp. 213–219.

[38] E. Mayfield and C. Penstein-Rosé, "An Interactive Tool for Supporting Error Analysis for Text Mining," in *Proceedings of the NAACL HLT 2010 Demonstration Session*, Los Angeles, CA, 2010, pp. 25–28.

[39] E. Mayfield and C. Rosé, "LightSIDE: Open Source Machine Learning for Text," in *Handbook of Automated Essay Evaluation: Current Applications and New Directions*, M. D. Shermis and J. Burstein, Eds. New York: Routledge, 2013, ch. 8, pp. 124–135.

[40] D. McCurry, "Can machine scoring deal with broad and open writing tests as well as human readers?" *Assessing Writing*, vol. 15, no. 2, pp. 118–129, 2010.

[41] T. McGee, "Taking a Spin on the Intelligent Essay Assessor," in *Machine Scoring of Student Essays: Truth and Consequences?2*, P. Freitag Ericsson and R. H. Haswell, Eds. Logan, UT: Utah State University Press, 2006, ch. 5, pp. 79–92.

[42] K. M. Nahar and I. M. Alsmadi, "The Automatic Grading for Online exams in Arabic with Essay Questions Using Statistical and Computational Linguistics Techniques," *MASAUM Journal of Computing*, vol. 1, no. 2, 2009.

[43] R. Östling, A. Smolentzov, and E. Höglin, "Automated Essay Scoring for Swedish," in *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, vol. 780, Atlanta, Georgia, US., 2013, pp. 42–47.

[44] E. B. Page, "The Imminence of... Grading Essays by Computer," *Phi Delta Kappan*, vol. 47, no. 5, pp. 238–243, 1966.

[45] ——, "Computer Grading of Student Prose , Using Modern Concepts and Software," *Journal of Experimental Education*, vol. 62, no. 2, pp. 127–142, 1994.

[46] D. E. Powers, J. C. Burstein, M. Chodorow, M. E. Fowles, and K. Kukich, "Stumping e-rater:challenging the validity of automated essay scoring," *Computers in Human Behavior*, vol. 18, no. 2, pp. 103–134, Mar. 2002.

[47] C. Ramineni and D. M. Williamson, "Automated essay scoring: Psychometric guidelines and practices," *Assessing Writing*, vol. 18, no. 1, pp. 25–39, 2013.

[48] C. S. Rich, M. C. Schneider, and J. M. D'Brot, "Applications of Automated Essay Evaluation in West Virginia," in *Handbook of Automated Essay Evaluation: Current Applications and New Directions*, M. D. Shermis and J. Burstein, Eds. New York: Routledge, 2013, ch. 7, pp. 99–123.

[49] L. M. Rudner, V. Garcia, and C. Welch, "An Evaluation of the IntelliMetric Essay Scoring System," *The Journal of Technology, Learning and Assessment*, vol. 4, no. 4, pp. 3–20, 2006.

[50] L. M. Rudner and T. Liang, "Automated Essay Scoring Using Bayes ' Theorem," *The Journal of Technology, Learning and Assessment*, vol. 1, no. 2, pp. 3–21, 2002.

[51] M. T. Schultz, "The IntelliMetric Automated Essay Scoring Engine - A Review and an Application to Chinese Essay Scoring," in *Handbook of Automated Essay Evaluation: Current Applications and New Directions*, M. D. Shermis and J. C. Burstein, Eds. New York: Routledge, 2013, ch. 6, pp. 89–98.

[52] M. D. Shermis, "State-of-the-art automated essay scoring: Competition, results, and future directions from a United States demonstration," *Assessing Writing*, vol. 20, pp. 53–76, 2014.

[53] M. D. Shermis and J. Burstein, "Introduction," in *Automated essay scoring: A cross-disciplinary perspective*, M. D. Shermis and J. Burstein, Eds. Manwah, NJ: Lawrence Erlbaum Associates, 2003, pp. xiii–xvi.

[54] M. D. Shermis, J. Burstein, and S. A. Bursky, "Introduction to Automated Essay Evaluation," in *Handbook of Automated Essay Evaluation: Current Applications and New Directions*, M. D. Shermis,

J. Burstein, and S. A. Bursky, Eds. New York: Routledge, 2013, ch. 1, pp. 1–15.

[55] M. D. Shermis, J. Burstein, and K. Zechner, "Automated Essay Scoring: Writing Assessment and Instruction," in *International encyclopedia of education*, 3rd ed., P. Peterson, E. Baker, and B. McGaw, Eds. Oxford, UK: Elsevier, 2010.

[56] M. D. Shermis and J. C. Burstein, Eds., *Handbook of Automated Essay Evaluation: Current Applications and New Directions*. New York: Routledge, 2013.

[57] M. D. Shermis and B. Hamner, "Contrasting State-of-the-Art Automated Scoring of Essays: Analysis," in *Handbook of Automated Essay Evaluation: Current Applications and New Directions*, M. D. Shermis and J. Burstein, Eds. New York: Routledge, 2013, ch. 19, pp. 313–346.

[58] M. D. Shermis, H. R. Mzumara, J. Olson, and S. Harrington, "On-line Grading of Student Essays: PEG goes on the World Wide Web," *Assessment & Evaluation in Higher Education*, vol. 26, no. 3, pp. 247–259, 2001.

[59] M. I. Smith, "The Reading-Writing Connection," MetaMetrics, Tech. Rep., 2009.

[60] M. I. Smith, A. Schiano, and E. Lattanzio, "Beyond the classroom." *Knowledge Quest*, vol. 42, no. 3, pp. 20–29, 2014.

[61] M. Syed, I. Norisma, and A. Rukaini, "Embedding Information Retrieval and Nearest-Neighbour Algorithm into Automated Essay Grading System," in *Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05)*, 2005, pp. 169–172.

[62] S. Valenti, F. Neri, and A. Cucchiarelli, "An Overview of Current Research on Automated Essay Grading," *Journal of Information Technology Education*, vol. 2, pp. 319–330, 2003.

[63] S. C. Weigle, "English as a Second Language Writing and Automated Essay Evaluation," in *Handbook of Automated Essay Evaluation: Current Applications and New Directions*, M. D. Shermis and J. C. Burstein, Eds. New York: Routledge, 2013, ch. 3, pp. 36–54.

[64] F. Wild, C. Stahl, G. Stermsek, Y. Penya, and G. Neumann, "Factors Influencing Effectiveness in Automated Essay Scoring with LSA," in *Proceedings of AIED 2005*, Amsterdam, Netherlands, 2005, pp. 947–949.

[65] R. Williams and H. Dreher, "Automatically Grading Essays with Markit©," *Issues in Informing Science and Information Technology*, vol. 1, pp. 693–700, 2004.

[66] D. M. Williamson, X. Xi, and F. J. Breyer, "A Framework for Evaluation and Use of Automated Scoring," *Educational Measurement: Isues and Practice*, vol. 31, no. 1, pp. 2–13, 2012.

[67] K. Zupanc and Z. Bosnić, "Automated Essay Evaluation Augmented with Semantic Coherence Measures," in *Proceedings of the 14th IEEE International Conference on Data Mining*, Shenzhen, China, 2014, pp. 1133–1138.

Parallel Fuzzy Rough Support Vector Machine for Data Classification in Cloud Environment

Arindam Chaudhuri
Samsung R & D Institute Delhi Noida - 201304 India
E-mail: arindamphdthesis@gmail.com

Keywords: FRSVM, Hadoop, MapReduce, PFRSVM

Received: November 27, 2014

Data classification has been actively used for most effective means of conveying knowledge and information to users. With emergence of huge datasets existing classification techniques fail to produce desirable results where the challenge lies in analyzing characteristics of massive datasets by retrieving useful geometric and statistical patterns. We propose a supervised parallel fuzzy rough support vector machine (PFRSVM) for in-data classification in cloud environment. The fuzzy rough set model takes care of sensitiveness of noisy samples and handles impreciseness in training samples bringing robustness to results. The algorithm is parallelized with a view to reduce training times. The system is built on support vector machine library using Hadoop implementation of MapReduce. The algorithm is tested on large datasets present at the cloud environment available at University of Technology and Management, India to check its feasibility and convergence. It effectively resolves outliers' effects, imbalance and overlapping class problems, normalizes to unseen data and relaxes dependency between features and labels with better average classification accuracy. The experimental results on both synthetic and real datasets clearly demonstrate the superiority of the proposed technique. PFRSVM is scalable and reliable in nature and is characterized by order independence, computational transaction, failure recovery, atomic transactions, fault tolerant and high availability attributes as exhibited through various experiments.

Povzetek: Razvita je nova verzija metode podpornih vektorjev (tj. strojnega učenja) nad velikimi podatki v oblaku, imenovana PFRSVM.

1 Introduction

The volume of business data is always expanding with rapid increase of global competitiveness [1] among the organizations. It is estimated that the volume of business data double within every two years. This fact is evident in both advanced and emerging economies. A common task often performed by the analysts and managers is data classification [2] which categorizes data into different subgroups in which ideas and objects are recognized and understood. In this process relevant and meaningful hidden information is discovered from data. From economic perspective, knowledge obtained from classified data can be applied directly for business application and services. However, as the amount of data increases continuously classification becomes more and more complex [3] where present techniques produce spurious results. This in turn disturbs the integrity of data. The inherent challenge lies in analyzing and interpreting characteristics of huge datasets by extracting significant usage patterns through various machine learning techniques [3].

Knowledge discovery [4] of meaningful information has been a topic of active research since past few years. The ongoing rapid growth of online data generally referred to as big data [1] have created an immense need for effective classification techniques [5]. The process of extracting knowledge from data draws upon the research in pattern classification and optimization to deliver advanced business intelligence [3]. The big data is specified using three characteristics viz. volume, variety and velocity [6]. This means that at some point in time when volume, variety and velocity of data are increased the current techniques may not be able to process the data. Ideally these three characteristics of a dataset increase data complexity and thus the existing techniques function below expectations within given processing time. Many applications such as classification, risk analysis, business forecasting etc. suffer from this problem. These are time sensitive applications and require efficient techniques to tackle the problem on the fly.

Some emerging techniques such as hadoop distributed file systems [7], cloud technology [8] and hive database [9] can be combined to big data classification problem. With this motivation this work entails the development of a supervised classification algorithm in cloud environment incorporating machine intelligence techniques for mining useful information [10]. The classification here is performed by the fuzzy rough [11] version of support vector machine (SVM) [12] which though considered faster than artificial neural network (ANN) [13] for training large datasets but is computationally intensive. Given a large enough dataset the training time can range from days to weeks. This problem is handled by extending the fuzzy rough version of SVM to parallel framework using hadoop implementation of MapReduce.

In this Paper, we propose parallel fuzzy rough support vector machine (PFRSVM) with MapReduce to classify huge data patterns in a cloud environment. Using MapReduce the scalability and parallelism of split dataset training is improved. Fuzzy rough support vector machine (FRSVM) [12], [14] is trained in cloud storage servers that work concurrently and then in every trained cloud node all support vectors are merged. This operation is continued until the classifier converges to an optimal function value in finite iteration size. This is done because it is impossible to train large scale datasets using FRSVM on a single computer. The fuzzy rough model is sensitive to noisy mislabeled samples which brings robustness to classification results. All the experiments are performed on the cloud environment available at University of Technology and Management, India.

The major contributions of this work include: (a) parallel implementation of FRSVM in cloud environment (b) formulation of sensitive fuzzy rough sets for noisy mislabeled samples to bring robustness in classification results (c) training FRSVM with MapReduce (d) identifying relevant support vectors at each computing node and merging with global support vectors (e) development of a scalable and reliable in-stream data classification engine adhering to the fundamental rules of stream processing such that it is maintains order independence in data processing, streamlines computational transaction, recovers from failure, generates atomic transactions and are fault tolerant and highly available in nature. To the best of our knowledge PFRSVM presented in this research work illustrates a robust architecture of in-stream data analytics which is first of its kind. The proposed computational framework has never been studied rigorously prior to this research work [15].

This Paper is organized as follows. The section 2 presents some work related to classification using fuzzy and rough versions of SVM. In section 3 an overview of SVM is presented. This is followed by a brief discussion on fuzzy rough sets. FRSVM is described in section 5. The section 6 illustrates the MapReduce pattern. PFRSVM formulation is highlighted in section 7. The experimental results and discussions are given in section 8. Finally conclusions are given in section 9.

2 Related work

Over the past decade data classification though fuzzy and rough versions of SVM have been rigorously used by researchers in several applications [15]. A brief illustration of few important ones is highlighted here. Mao et al investigated multiclass cancer classification by using fuzzy support vector machine (FSVM) and binary decision tree with gene selection. They proposed two new classifiers with gene selection viz. FSVM and binary classification tree based on SVM tested on three datasets such as breast cancer, round blue cell tumors and acute leukemia data which gave higher prediction accuracy. Abe et al studied multiclass problems using FSVM where they used truncated polyhedral pyramidal membership function for decision functions to train SVM for two different pairs of classes. Huang et al proposed new SVM fuzzy system with high comprehensibility where SVM is used to select significant fuzzy rules directly related to a fuzzy basis function. Analysis and comparative tests about SVM fuzzy system show that it possesses high comprehensibility and satisfactory generalization capability. Thiel et al studied fuzzy input fuzzy output one against all SVM where fuzzy memberships were encoded in fuzzy labels to give fuzzy classification answer to recognise emotions in human speech. Shilton et al proposed an iterative FSVM classification whereby fuzzy membership values are generated iteratively based on positions of training vectors relative to SVM decision surface itself. Li et al studied fault diagnosis problem using FSVM. Pitiranggon et al constructed a fuzzy rule based system from SVM which has the capability of performing superior classification than the traditional SVM. Zhu et al used FSVM control strategy based on sliding mode control to reduce oscillation. Parameters of FSVM controller were optimized by hybrid learning algorithm which combines least square algorithm with improved genetic algorithm to get the optimal control performance for controlled object. Li et al proposed double or rough margin based FSVM algorithm by introducing rough sets into FSVM. First, the degree of fuzzy membership of each training sample is computed and then data with fuzzy memberships were trained to obtain decision hyperplane that maximizing rough margin method. Chen et al extracted a new feature of consonants employing wavelet transformation and difference of similar consonants. Then algorithm classified consonants using multiclass FSVM. Long et al proposed network intrusion detection model based on FSVM. They concentrated on automatic detection of network intrusion behavior using FSVM. The system composed of five modules viz. data source, AAA protocol, FSVM located in local computer, guest computer and terminals. The intrusion detection algorithm based on FSVM is implemented by training and testing process. Jian et al coined FSVM based method to refine searching results of SEQUEST which is a shotgun tandem mass spectrometry based peptide sequencing using programs on a dataset derived from synthetic protein mixtures. Performance comparison on various criteria show that proposed FSVM is a good approach for peptide identification task. Duan et al studied FSVM based on determination of membership. They investigated sensitivity issues relating SVM to outlier and noise points which favours use of FSVM though appropriate fuzzy membership. Shi et al proposed an emotional cellular based multiclass FSVM on product's kansei image extraction. Li et al proposed fuzzy twin SVM algorithm that has computational speed faster than traditional SVM. It takes into account the importance of training samples on learning of decision hyperplane with respect to classification task. Yan et al proposed probability FSVM based on the consideration both for fuzzy clustering and probability distributions. The model is based on consideration that probability distribution among samples exhibits superior performance.

3 Support vector machine

Support vector machine (SVM) [12] is a promising pattern classification tool based on structural risk minimization and statistical learning theory [16]. Many complex problems have been solved by SVMs. It minimizes prediction error and models complexities. SVM formalizes classification boundary by dividing points having different labels so that boundary distance from closest point is maximized. It transforms training vectors into high dimensional feature space labeling each vector by its class. It classifies data through set of support vectors which are members of training input set that outline a hyperplane in feature space [12], [16] as shown in figure 1. Structural risk minimization reduces generalization error. The number of free parameters depends on margin that separates data points. SVM fits hyperplane surface to data using kernel function that allows handling of curse of dimensionality. To recognize support vectors the problem is restructured as following quadratic optimization problem [12] which is convex, guarantees uniqueness and optimality:

$$\min \quad \|w\|^2$$

$$\text{subject to: } z_i\left(w^T Y_i + b\right) \geq 1,\ i = 1, \ldots\ldots, M \quad (1)$$

In equation (1) is weight vector and b is bias term. Slack variables $\xi_i; i \in \{1, ..., M\}$ measures violation of constraints such that the quadratic problem now becomes:

$$\min \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{M}\xi_i$$

$$\text{subject to:}$$
$$\left\{z_i\left(w^T Y_i + b\right) 1 - \xi_i; i = 1, \ldots\ldots, M\ ;\ \xi_i \geq 0 \qquad (2)$$

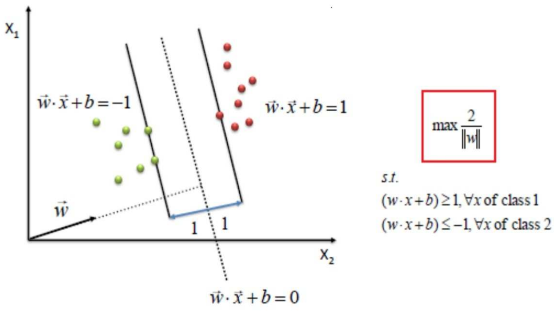

X₁
$\vec{w}\cdot\vec{x}+b=-1$
$\vec{w}\cdot\vec{x}+b=1$
$\vec{w}$
1
1
X₂
$\vec{w}\cdot\vec{x}+b=0$
max $\frac{2}{\|w\|}$
s.t.
$(w\cdot x+b)\geq 1, \forall x$ of class 1
$(w\cdot x+b)\leq -1, \forall x$ of class 2

Figure 1: Separating Hyperplane between Classes leading to different Support Vectors.

In equation (2) regularization parameter C determines constraint violation cost. To classify nonlinear data [17] the mapping transforms classification problem into higher dimensional feature space giving linear separability. This is achieved by transforming Y_i into higher dimensional feature space through $\Phi(Y_i)$ satisfying Mercer's condition [12]. The quadratic problem is solved by scalar product $K(Y_i, Y_j) = (Y_i) \cdot (Y_j)$. By using Lagrange multipliers and kernels the problem becomes:

$$\min \quad \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M} z_i z_j \alpha_i \alpha_j K(Y_i, Y_j) - \sum_{j=1}^{M}\alpha_j$$

$$\text{subject to:}$$
$$\sum_{i=1}^{M} z_i\alpha_i = 0; i = 1, \ldots\ldots, M;\ 0 \leq \alpha_i \leq C \ (3)$$

The commonly used kernels are polynomial and gaussian functions. In training SVMs we need kernel function and its parameters to achieve good results and convergence. When solving two class classification problems each training point is treated equally and assigned to only one class. In many real word problems some training points are corrupted by noise. Some points also are misplaced on wrong side. These points are outliers and belong to two classes with different memberships. SVM training algorithm makes decision boundary to severely deviate from optimal hyperplane as it is sensitive to outliers [12], [16]. This is handled by techniques as illustrated in [17], [18]. [19].

4 Fuzzy rough sets

Let R be an equivalence relation on universal set P. The family of all equivalence classes induced on P by R is denoted by $\frac{P}{R}$. One such equivalence class in $\frac{P}{R}$ contains $p \in P$ is denoted by $[p]_R$. For any output class $A \subseteq P$ lower and upper approximations approaching A closely from inside and outside are defined [11]. Rough set $R(A)$ is a representation of A by lower and upper approximations. When all patterns from equivalence class do not carry same output class label rough ambiguity is generated as manifestation of one-to-many relationship between equivalence class and output class labels. The rough membership function $rm_A(p) : A \rightarrow [0, 1]$ of pattern $p \in P$ for output class A is given by equation (4) in Appendix A.

When equivalence classes are not crisp they form fuzzy classes $\{FC_1, \ldots.., FC_H\}$ generated by fuzzy weak partition [11] of input set P. Fuzzy weak partition means that each $FC_i; i \in \{1, \ldots., H\}$ is normal fuzzy set. Here, $\max_p \mu_{FC_i}(p) = 1$ and $inf_p \max_i \mu_{FC_i}(p) > 0$ while $\underbrace{sup}_{x} \min_{i,j}\left\{\mu_{FC_i}(p), \mu_{FC_j}(p)\right\} < 1 \ \forall i, j \in \{1, 2, \ldots\ldots, H\}$. Here $\mu_{FC_i}(p)$ is fuzzy membership function of pattern p in class FC_i.The output classes $C_c; c = \{1, 2, \ldots\ldots, H\}$ may be fuzzy also. Given a weak fuzzy partition $\{FC_1, FC_2, \ldots\ldots, FC_H\}$ on

P description of any fuzzy set C_c by fuzzy partitions under upper approximation $\overline{C_c}$ is given by equation (5) in Appendix A and lower approximation $\underline{C_c}$ is:

$$\mu_{\underline{C_c}}(FC_i) = \underbrace{sup}_{x \in C_c} min\left\{\mu_{FC_i}(p), \mu_{C_c}(p)\right\} \quad \forall p \ (6)$$

The tuple $\left\langle \underline{C_c}, \overline{C_c} \right\rangle$ is called fuzzy rough set. Here $\mu_{C_c}(p) = \{0, 1\}$ is fuzzy membership of input p to C_c. The fuzzy roughness appears when class contains patterns that belong to different classes.

5 Fuzzy rough support vector machine

Based on SVM and fuzzy rough sets [11] we present FRSVM. To solve misclassification problem in SVM, fuzzy rough membership is introduced to each input point such that different points can make unique contribution to decision surface. The input's membership is reduced so that its contribution to total error term is decreased. FRSVM also treats each input as of opposite class with higher membership. This way fuzzy rough machine makes full use of data and achieves better generalization ability. We consider training sample points as:

$$SP = \{(Y_i, z_i, frm_i(p)); i = 1, \ldots\ldots, M\} \ (7)$$

Here each $Y_i \in \mathrm{R}^N$ is training sample and $z_i \in \{-1, +1\}$ represents its class label; $frm_i(p); i = 1, \ldots\ldots, M$ is fuzzy rough membership function satisfying $s_j \leq frm_i(p) \leq s_i; i, j = 1, \ldots\ldots., M$ with sufficiently small constant $s_j > 0$ and $s_i \leq 1$ considering pattern p. Taking $P = \{Y_i \mid (Y_i, z_i, frm_i(p)) \in SP\}$ containing two classes; one class C^+ with sample point $Y_i(z_i = 1)$ and other class C^- with sample point $Y_i(z_i = -1)$ such that:

$$C^+ = \{Y_i | Y_i \in SP \wedge z_i = 1\} \ (8)$$

$$C^- = \{Y_i | Y_i \in SP \wedge z_i = -1\} \ (9)$$

Here, $P = C^+ \cup C^-$; then classification problem is given by equation (10) in Appendix A.

In equation (10) C is constant. The fuzzy rough membership $frm_i(p)$ governs the behavior of corresponding point Y_i towards one class and ξ_i is error measure in FRSVM. The term $frm_i(p)\xi_i$ is an error measure with different weights. A smaller $frm_i(p)$ reduces the effect of ξ_i in equation (10) such that point Y_i is treated less significant. The quadratic problem can also be solved by their dual alternatives [12]. The kernel function used is hyperbolic tangent kernel $K(Y_i, Y_j) = \tanh[(Y_i) \cdot (Y_j)]$ [13] given in figure 2. It is conditionally positive definite and allows lower computational cost and higher rate of positive eigenvalues of kernel matrix alleviating limitations of other kernels. The sigmoid kernel has been used in several cases with appreciable success [19] motivating its usage in fuzzy rough membership function in proposed machine. The class centre of C^+ and C^- in feature space is defined as Φ_+ and Φ_- respectively.

$$\Phi_+ = \frac{1}{m_+} \sum_{Y_i \in C^+} (Y_i) f_i \ (11)$$

$$\Phi_- = \frac{1}{m_-} \sum_{Y_i \in C^-} (Y_i) f_i \ (12)$$

In equations (11) and (12) m_+ and m_- is number of samples of class C^+ and C^- with f_i frequency of i^{th} sample in feature space (Y_i). The radius of C^+ ($Y_i \in C^+$) and C^- ($Y_i \in C^-$) with $n = \sum_i f_i$:

$$rd_+ = \frac{1}{n} \max \|\Phi_+ - \Phi(Y_i)\| \ (13)$$

$$rd_- = \frac{1}{n} \max \|\Phi_- - \Phi(Y_i)\| \ (14)$$

Then equation (13) can be written as equation (15) which is given in Appendix A. In equation (15) $Y' \in C^+$ and m_+ is number of training samples in C_+. Similarly, we have equation (16) as given in Appendix A.

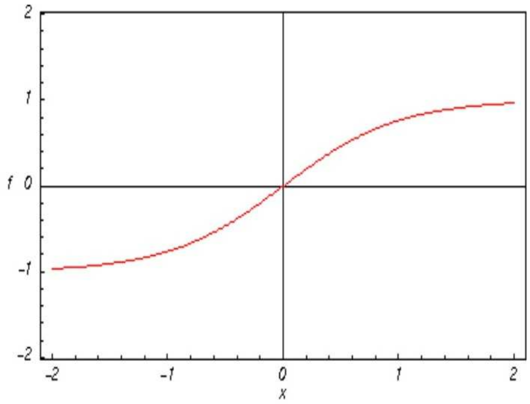


Figure 2: Hyperbolic Tangent Kernel.

In equation (16) $Y' \in C^-$ and m_- is number of training samples in C_-. The square of distance between sample $Y_i \in C^+$and $Y_i \in C^-$ to their class centres in feature space is:

$$dist_{i+}^2 = \|(Y_i) - \Phi_+\|^2 = \Phi^2(Y_i) - 2tanh[\Phi(Y_i) \cdot \Phi_+] + \Phi_+^2$$

$$dist_{i+}^2 = K(Y_i, Y_j) - \frac{2}{m_+} \sum_{Y_j \in C^+} K(Y_i, Y_j) + \frac{1}{m_+^2} \sum_{Y_j \in C^+} \sum_{Y_k \in C^+} K(Y_j, Y_k) \ (17)$$

$$dist_-^2 = K(Y_i, Y_j) - \frac{2}{m_-} \sum_{Y_j \in C^-} K(Y_i, Y_j) +$$

$$\frac{1}{m_-^2} \sum_{Y_j \in C^-} \sum_{Y_k \in C^-} K(Y_j, Y_k) \quad (18)$$

Now, $\forall i; i = 1, \ldots\ldots, M$ fuzzy rough membership function $frm_i(p)$ is defined in equation (19) as given in Appendix A. The term $(\cdot)$ in equation (19) holds when $(\exists i)\, \mu_{FC_i}(p) > 0$ and $\varepsilon > 0$ so that $frm_i(p) \neq 0$. Here, $\tau^i_{C_c} = \frac{\|FC_i \cap C_c\|}{\|FC_i\|}$ and $\frac{1}{\sum_i \mu_{FC_i}(p)}$ normalizes fuzzy rough membership function $\mu_{FC_i}(p)$. The function is a constrained fuzzy rough membership function. The above definition can further be modified as equation (20) which is given in Appendix A.

In equation (20) $\hat{H}$ is number of clusters and p has a non-zero membership. When p does not belong to any cluster then $\hat{H} = 0$ so that $\frac{\sum_{i=1}^{H} \mu_{FC_i}(p) \tau^i_{C_c}}{\hat{H}}$ becomes undefined. This issue is resolved by taking $frm^c_i(p) = 0$ when p does not belong to any cluster. This definition does not normalize fuzzy rough membership values and so the function is a possibilistic fuzzy rough membership function. The equations (19) and (20) expresses the fact that if an input pattern belongs to clusters (all belonging to only one class) with non-zero memberships then no fuzzy roughness are involved. However, in equation (20) it matters to what extent the pattern belongs to clusters. This is evident from property 11. Some of the important properties applicable to equations (19) and (20) are:

Property 1: $0 < frm_i(p) < 1$ and $0 < frm^c_i(p) < 1$

Property 2: $frm_i(p)/frm^c_i(p) = 1$ or 0 iff no uncertainty exists

Property 3: If no uncertainty is with p then $frm_i(p)/frm^c_i(p) = \tau^i_{C_c}$ for some $j \in \{1, 2, \ldots\ldots, H\}$

Property 4: If no uncertainties are with p then $frm_i(p)/frm^c_i(p) = rm_A(p)$

Property 5: When each class is crisp and fine and class memberships are crisp $frm_i(p)/frm^c_i(p)$ is equivalent to fuzzy membership of p in class C_c

Property 6: If a and b are two patterns with $\mu_{FC_j}(a) = \mu_{FC_j}(b)\ \forall j$ and $\mu_{FC_{C_c}}(a) = \mu_{FC_{C_c}}(b)$ then $frm_i(a) = frm_i(b)$ and $frm^c_i(a) = frm^c_i(b)$

Property 7: $rm^c_{P-C_c}(p) = \left(\frac{\sum_{i=1}^{H} \mu_{FC_i}(p) \tau^i_{C_c}}{\hat{H}} \right) - rm^c_{C_c}(p) \wedge rm_{P-C_c}(p) = 1 - rm_{C_c}(p)$

Property 8: $\tau_{C_c \cup V}(p) \geq max\{\tau_{C_c}(p), \tau_V(p)\}$

Property 9: $\tau_{C_c \cap V}(p) \leq min\{\tau_{C_c}(p), \tau_V(p)\}$

Property 10: If W is family of pairwise disjoint crisp subsets of P then $\tau_{\cup W}(p) = \sum_{C_c \in W} \tau_{C_c}(p)$

Property 11: For C class classification problem with crisp classes, possibilistic fuzzy rough functions behave in possibilistic manner and constrained fuzzy rough functions behave otherwise

Property 12: If class is fuzzy then $0 \leq \sum_{c=1}^{C} \tau_{C_c}(p) \leq C$

The fuzzy rough membership values depend on fuzzy classification of input dataset. The fuzziness in classes represents fuzzy linguistic uncertainty present in dataset. The classification can be performed through either (a) unsupervised classification which involves collecting data from all classes and classify them subsequently without considering associated class labels with data or (b) supervised classification where separate datasets are formed for each class and classification is performed on each such dataset to find subgroups present in data from same class. Both classification tasks can be performed by some trivial classification algorithms [17], [18], [19]. However, there are certain problems which are to be taken care of such as: (a) number of classes which have to be fixed apriori or which may not be known (b) it will not work in case number of class is one and (c) generated fuzzy memberships are not possibilistic.

To overcome the first problem evolutionary programming based method may be used [18]. For various classification problems evolutionary methods can automatically determine number of classes. It is worth mentioning that number of classes should be determined as best as possible. Otherwise, calculation of fuzzy linguistic variables will be different and as a result fuzzy rough membership values may also vary. For the second problem if it is known apriori that only one class is present then mean and standard deviation are calculated from input dataset and π fuzzy membership curve is fitted. But while doing so care must be taken to detect possible presence of the outliers in input dataset. To overcome third problem possibilistic fuzzy classification algorithm or any mixed classification algorithm can be used. As of now there is no single classification algorithm which can solve all the problems. If output class is fuzzy then it may be possible to assign fuzzy memberships for output class subjectively. However, if domain specific knowledge is absent then we have to be satisfied with given crisp membership values.

The fuzzy rough ambiguity plays a critical role in many classification problems because of its capability towards modeling non statistical uncertainty. The characterization and quantification of fuzzy roughness are important aspects affecting management of uncertainty in classifier design. Hence measures of fuzzy roughness are essential to estimate average ambiguity in output class. A measure of fuzzy roughness for discrete output class $C_c \subseteq X$ is a mapping $S(X) \to \Re^+$ that quantifies degree of fuzzy roughness present in C_c. Here $S(X)$ is set of all fuzzy rough power sets defined within universal set X. The fuzzy rough ambiguity must be zero when there is no ambiguity in deciding whether an input pattern belongs to it or not. The equivalent classes form fuzzy classes so that each class is fuzzy linguistic variable. The membership is function of center and radius of each class in feature space and is represented with kernel.

In formulation of FRSVM, fuzzy membership reduces outliers' effects [18], [19]. When samples are nonlinear separable fuzzy memberships are calculated in input space but not in feature space. The contribution of each point in hyperplane in feature space cannot be represented properly and fuzzy rough membership function efficiently solves this. Through fuzzy rough membership function the input is mapped into feature space. The fuzzy rough memberships are calculated in feature space. Further using kernel function it is not required to know shape of mapping function. This method represents contribution of each sample point towards separating hyperplane in feature space [19]. Thus, the proposed machine reduces outlier' effects efficiently and has better generalization ability.

The higher value of fuzzy rough membership function implies importance of data point to discriminate between classes. It implies highest value is given by support vectors. These vectors are training points which are not classified with confidence. These are examples whose corresponding $\alpha_i$values are non zero. From representer theorem [20] optimal weight vector w^* is linear combination of support vectors which are essential training points. The number n_{SV} of support vectors also characterizes complexity of learning task. If $n_{SV}$is small then only a few examples are important and rest can be disregarded. If n_{SV} is large then nearly every example is important for accuracy. It has been shown that under general assumptions about loss function and underlying distribution training data $n_{SV} = \Omega(n)$. This suggests that asymptotically all points are critical for training. While this gives $\Omega(n)$ bound on training time this solves FRSVM problem exactly. Further, datasets need not necessarily have $\Theta(n)$ support vectors.

6 MapReduce

MapReduce [21] illustrated in figure 3 is a programming model for processing large datasets with parallel distributed algorithm on cluster. It is composed of map and reduce function combinations derived from functional programming. The users specify map function that processes key value pair to generate a set of intermediate key value pairs and reduce function that merges all intermediate values associated with same intermediate key [21]. MapReduce is divided into two major phases called map and reduce separated by an internal shuffle phase of intermediate results. The framework automatically executes those functions in parallel over n number of processors. MapReduce job executes three basic operations on dataset distributed across many shared nothing cluster nodes. The first task is map function that is processed in parallel manner by each node without transferring any data with other nodes. In next operation processed data by map function is repartitioned across all nodes of cluster. Finally reduce task is executed in parallel by each node with partitioned data.

A file in distributed file system is split into multiple chunks and each chunk is stored on different data nodes. A map function takes key value pair as input from input chunks and produces list of key value pairs as output. The type of output key and value can be different from input values.

$$map\left(key_1, value_1\right) \Rightarrow list\left(key_2, value_2\right) \quad (21)$$

A reduce function takes key and associated value list as input and generates list of new values as output:

$$reduce\left(key_2, list\left(value_2\right)\right) \Rightarrow list\left(value_3\right) \quad (22)$$

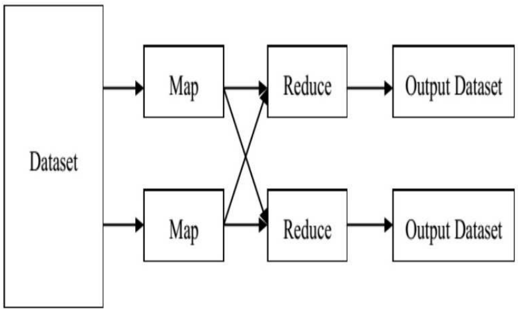


Figure 3: MapReduce System.

Each reduce call produces either value $value_3$ or an empty return, through one call returns more than one value. The return of all calls is collected as desired result list. The main advantage of MapReduce is that it allows distributed processing of submitted job on subset of whole dataset in the network.

7 Experimental framework: Parallel fuzzy rough support vector machine

FRSVM suffers from the scalability problem [22] both in terms of memory and computational time. In order to improve the scalability problem a parallel FRSVM viz. PFRSVM is developed which handles the stated problems through parallel computation. It is executed through multiple commodity computing nodes on cascade FRSVM model parallel in cloud environment. PFRSVM training is realized through FRSVMs where each FRSVM acts as filter. This leads to the process of deriving local optimal solutions which contribute towards the global optimum solution. Through PFRSVM huge scale data optimization problems are divided into independent small optimization problems. The support vectors of the prior FRSVM are used as the input of later FRSVM. FRSVM is aggregated into PFRSVM in hierarchical fashion. The PFRSVM training process is described in the figure 4.

In this architecture, support vectors sets of two FRSVMs are combined together as input to new FRSVM. This process continues until only one vector set remains. Here a single FRSVM never deals with the whole training set. If filters in the first few layers are efficient in extracting more support vectors it leads to maximum optimization. This results in handling fewer support vectors in the later layers. Thus training sets of each of the sub problems are much smaller than that of whole problem where support vectors are a small subset of training vectors. For training FRSVM classifier functions LibSVM [23] with various kernels. The cross validation test is used to find appropriate values of parameters C and γ as discussed in section 8. The entire framework is implemented with Hadoop and streaming Python package.

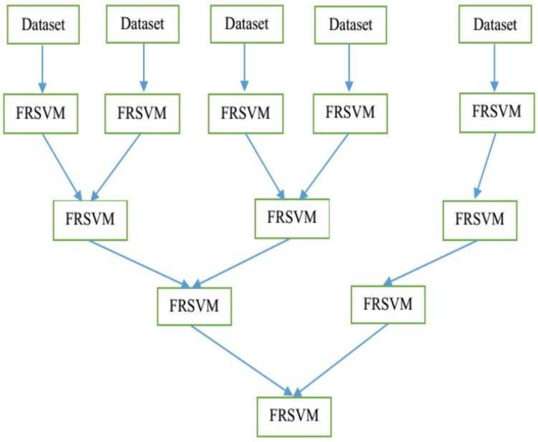


Figure 4: The training flow of PFRSVM.

Given computing nodes in cloud environment the original large scale data SD is partitioned into smaller data sections $\{SD_1, \ldots\ldots, SD_n\}$ uniformly. These small data sets SD_i are placed on the computing nodes. Then the corresponding partition files are created. Based on the available computation environment the job configuration manager [24] configures the computation parameters such as map, reduce, class names combination, number of map and reduce tasks, partition file etc. The driver manager [24] initiates the MapReduce task. The dynamic parameters are transformed to each computing node through an API interface [24].

In each computing node the map tasks are operated. The first layer of figure 4 loads the sample data from local file system according to the partition file. Each node in the layer classifies partitioned dataset SD_i locally through FRSVM from which support vectors are obtained. In following layers training samples are support vectors of the former layer. The local support vectors obtained in earlier layers are merged with global support vectors in the later layers. LibSVM is used to train each FRSVM. In LibSVM sequential minimal optimization [23] is used to select workset in decomposition methods for training FRSVM. FRSVM is trained using [23].

In map job of MapReduce subset of training set is combined with other local support vectors. The trained support vectors are sent to reduce jobs. In reduce job support vectors of all map jobs are collected, evaluated, merged with global support vectors and fed back to the client. Each computer within cloud environment reads global support vectors. Then it merges global support vectors with subsets of local training data and classifies via FRSVM. Finally, all computed support vectors in cloud computers are merged. The algorithm saves global support vectors with new ones. The training process is performed iteratively and stops when all FRSVMs are combined together resulting into PFRSVM. The entire system is schematically shown in figure 5. The steps of algorithm are:

PFRSVM Algorithm

```
1. Initialize global support vector set $\left[i = 0, GV^i = \phi\right]$
2. $i = i + 1$
3. For any computing node $c \in C$
     Read global support vectors
     Merge them with subset of training data
4. Train FRSVM with merged new dataset
5. Find support vectors
6. When all computers in cloud complete training
   Merge all calculated support vectors
   Save to global support vector set
7. If $(t^i = t^{i-1})$
     Stop
   else
     Goto 2
```

The map and reduce functions of PFRSVM are:

Map function of PFRSVM Algorithm

```
GV= $\phi$
While $(t^i \neq t^{i-1})$ do
for $c \in C$ do
        $TS_c^i = TS_c^i \cup GV^i$
   end for
end while
```

Reduce function of PFRSVM Algorithm

```
While $(t^i \neq t^{i-1})$ do
for $c \in C$ do
        $SV_c, t^i = frsvm(TS_c)$
end for
for $c \in C$ do
        $GV = GV \cup SV_C$
end for
end while
```

The architecture of PFRSVM developed is not able to achieve linear speedup when number of machines continues to increase beyond a data size dependent threshold. This happens because of communication and synchronization overheads between the computing nodes. The communication overhead occurs when message passing takes place between machines. The synchronization overhead occurs when master node waits for task completion on slowest machine. The MapReduce compatible algorithm runs with Hadoop cluster [25] which uses identical software versions and hardware configurations through which linear speedup is achieved.

Another aspect which deserves attention is convergence of PFRSVM while performing classification [12], [19]. To consider this let us assume a subset ST of training set TS with $OPT\left(ST\right)$ as optimal objective function over ST. Here H^{*} is global optimal hypothesis with minimal empirical risk $RK_{emp}\left(H^{*}\right)$. The algorithm starts with $GV^{0}=0$ and generates a decreasing sequence of positive set of vectors GV^{i} with following hinge loss function:

$$HL\left(f\left(x\right),y\right)=max\left[0,1-y\cdot f\left(x\right)\right] \quad (23)$$

The empirical risk is computed with approximation:

$$RK_{emp}\left(H\right)=\frac{1}{n}\sum_{i=1}^{n}HL\left(H\left(x_{i}\right),y_{i}\right) \quad (24)$$

According to empirical risk minimization principle learning algorithm chooses hypothesis $\hat{H}$ minimizing risk:

$$\hat{H}=arg\ min_{H\in\mathcal{H}}\ RK_{emp}\left(H\right) \quad (25)$$

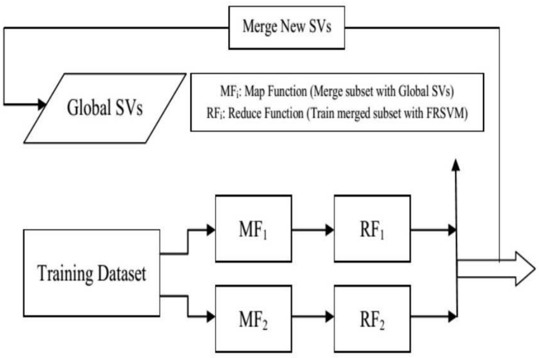

Merge New SVs
Global SVs
MFi: Map Function (Merge subset with Global SVs)
RFi: Reduce Function (Train merged subset with FRSVM)
Training Dataset
MF1
RF1
MF2
RF2

Figure 5: Schematic representation of PFRSVM in Cloud Environment.

A hypothesis exists in every cloud node. Let Y be subset of training data at cloud node j; $H^{i,j}$ is hypothesis at node j with iteration i such that optimization problem in equation (3) and corresponding equation (10) becomes equation (26) given in Appendix A.

In equation (26) KM_{12} and KM_{21} are kernel matrices with respect to $KM_{12}=\left\{K_{j,k}\left(y_{jk},GV^{i}_{(j,k)}\right)|j=1,..,m;k=1,..n\right\}$. Here α_{1} and α_{2} are solutions estimated by node j with dataset Y and GV. The kernel matrix KM is symmetric positive definite on square because of Mercer's condition as a result of which KM_{12} and KM_{21} are equal. At iteration i matrices $KM_{11}=\left\{K_{j,k}\left(y_{jk},y_{jk}\right)|y_{jk}\in Y,j=1,..,m;k=1,..n\right\}$ and

$$KM_{22}=\left\{K_{j,k}\left(GV,GV\right)|j=1,..,m;k=1,..n\right\}.$$

The algorithm terminates when hypothesis' empirical risk is same with previous iteration i.e. $RK_{emp}\left(H^{i}\right)=RK_{emp}\left(H^{i-1}\right)$. The accuracy of decision function of PFRSVM classifier at i^{th} iteration is always greater than or equal to maximum accuracy of decision function of SVM classifier at 1^{st}iteration i.e. $RK_{emp}\left(H^{i}\right)\leq arg\ min_{H\in\mathcal{H}^{i-1}}\ RK_{emp}\left(H\right)$.

Finally we discuss the complexity of proposed algorithm. The time complexity of FRSVM is $O\left(n^{2}\right)$. The bandwidth of network determines the transmission time of data T_{tt}between map and reduce nodes. When training data is divided into p partitions computation cost is calculated in terms of layers of cascade FRSVM as $\log_{2}p$. Considering ratio between number of support vectors and whole training sample as $a\left(0<a<1\right)$ and ratio between support vectors and training sample excluding first layer as $b(1 \quad < \quad b \quad < \quad 2)$, the number of training samples of i^{th} layer is $nab\left(\frac{b}{2}\right)^{\log_{2}p\ -i}$. The computation time is: $O\left(\left(\frac{n}{p}\right)^{2}\right)+\sum_{i}O\left(\left(nab\left(\frac{b}{2}\right)^{\log_{2}p\ -i}\right)^{2}\right)+O\left(\sum_{i}nab\left(\frac{b}{2}\right)^{\log_{2}p\ -i-1}2^{\log_{2}p\ -i-1}\right)+T_{tt}$. The overhead of data transfer includes three parts: (a) the first part is data transfer from map to reduce nodes which are support vectors obtained by map nodes (b) the second part is data transfer from reduce to server node which are support vectors and (c) the third part is data transfer from server nodes to map node which are training samples combined by support vectors. The overhead of data transfer depends on bandwidth of MapReduce cluster.

8 Experimental results and discussions

In this section, the effectiveness of PFRSVM is demonstrated with various experiments. At first a brief discussion on generation of synthetic data is given. Then real datasets used are highlighted. Next we illustrate selection of optimum values of (C,γ) and kernel used in PFRSVM. The classification on synthetic data follows this. Next the outlier generation in real data is given. This is followed by classification on real data. The imbalance and overlapping class classification is presented next. This is followed by generalization to unseen data when size of training and test dataset are varied. The discussion on features and labels relaxation follows this. The comparative classification performance of PFRSVM with other approaches is given next. Finally, some critical issues regarding implementation of

PFRSVM in cloud environment is highlighted.

8.1 Generation of synthetic data

To validate performance of PFRSVM in realistic environments the datasets are generated as: (a) We randomly created S clusters such that for each cluster: (i) centre $cp \in [cp_l, cp_h]$ for each dimension independently; (ii) radius $rd \in [rd_l, rd_h]$ and (iii) number of points $np \in [np_l, np_h]$ in each cluster (b) We labeled clusters based on X-axis value such that cluster T_i is labeled as positive if $cp_i^x < \alpha - rd_i$ and negative if $cp_i^x > \alpha + rd_i$. Here $cp_i^x$is X-axis of centre cp_i and α is threshold between $[cp_l, cp_h]$. We removed clusters not assigned to either of positive or negative which lie across threshold α on X-axis to make them linearly separable. (c) Once characteristics of each cluster are determined points for clusters are generated according to 2-dimensional independent normal distribution with $[cp, rd]$ as mean and standard deviation. The class label of each data is inherited from label of its parent cluster. It is noted that due to normal distribution maximum distance between a point in cluster and centre is unbounded. The points that belong to one cluster but located farther than surface of cluster are treated as outsiders. Due to this dataset does not become completely linearly separable. Figure 6 shows a dataset according to parameters (see Table 1). The data generated from clusters in left and right side are positive ('l') and negative ('-') respectively. Figure 6(b) shows 0.5 % randomly sampled data from original dataset of figure 6(a). From figure 6(b) the random sampling reflects unstable data distribution of original dataset which includes nontrivial amount of unnecessary points. The dashed ellipses on Figure 6(b) indicate densely sampled areas that reflect original data distribution are mostly not very close to boundary. As such areas around boundary are less dense because cluster centers which are very dense are unlikely to cross over boundary of multiple classes. Thus unnecessary data increases training time of PFRSVM without contributing to support vectors of boundary. The random sampling disturbs more when probability distributions of training and testing data are different because random sampling only reflects that distribution of training data could miss significant regions of testing data. This happens as they are collected in different time periods. For fair evaluation testing data is generated using same clusters and radiuses but different probability distributions by randomly reassigning number of points for each cluster.

8.2 Real datasets used

The real datasets from UCI Machine Learning Repository viz. German Credit, Heart Disease, Ionosphere, Semeion Handwritten Digit and Landsat Satellite are used to conduct experiments and illustrate convergence of PFRSVM. The different attributes of datasets are given (see Table 2 in Appendix B). The missing values' problem in datasets is resolved by genetic programming [19]. The nominal values

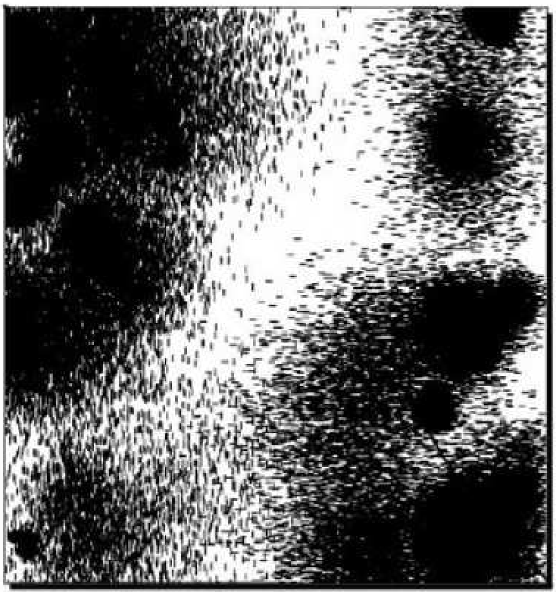


Figure 6: Original dataset $[N = 114996]$.

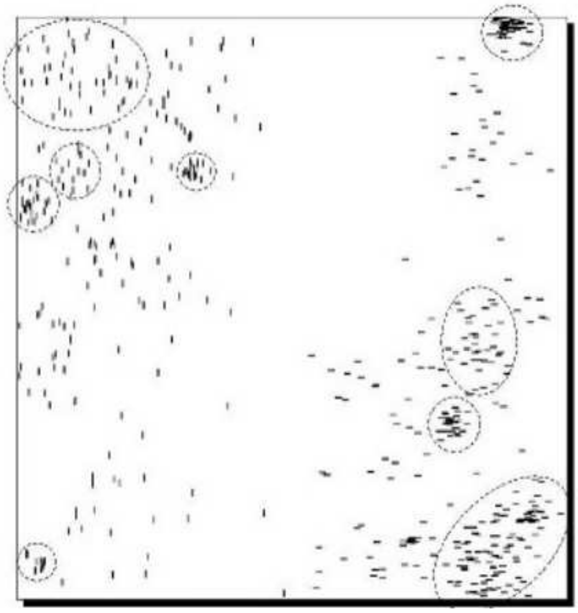


Figure 7: 0.5 % randomly sampled data $[N = 602]$

Figure 8: Synthetic Dataset in 2-Dimensional Space.

are converted into numerical ones during processing. The datasets are divided into training and test sets each consisting of 50 $\%$ samples. The training set is created by randomly selecting samples from whole dataset and remaining samples constitute test set.

Parameter	Values
Number of clusters S	70
Range of cp $[cp_l, cp_h]$	[0.0, 1.0]
Range of rd $[rd_l, rd_h]$	[0.0, 0.1]
Range of np $[np_l, np_h]$	[0, 5000]
α	0.7

Table 1: Data Generated for figure 6.

8.3 Selection of optimum values of (C, γ) and kernel used in PFRSVM

Selection of appropriate PFRSVM parameters plays a vital role in achieving good results. We consider RBF kernel and use cross validation to find best parameters of (C, γ) to train and test whole training set. A common strategy is to separate dataset into known and unknown parts. The prediction accuracy obtained from unknown set precisely reflects classifying performance on an independent dataset. An improved version viz. v fold cross validation is used $(v = 20)$ where training set is divided into v equal subsets. One subset is tested using classifier trained on remaining $(v - 1)$ subsets. Each instance of whole training set is predicted once cross validation accuracy is data percentage correctly classified. This prevents over fitting problem. The grid search also finds (C, γ) using cross validation. Various pairs of (C, γ) values are tried and best cross validation accuracy is selected. We found that exponentially growing sequences of (C, γ) viz. $(C = 2^{-5}, 2^{-3}, \ldots\ldots, 2^{15}; \gamma = 2^{-15}, 2^{-13}, \ldots\ldots, 2^{3})$ give best results. The grid search performs exhaustive parameter search by using heuristics with good complexity. The kernel used here is RBF. It has been used with considerable success so its choice is obvious. It nonlinearly maps samples into higher dimensional space when relation between class labels and attributes is nonlinear. RBF kernel has less hyper parameters which also influences complexity of model selection. Also RBF kernel has fewer numerical difficulties. After scaling the datasets we first use grid search and find average best (C, γ) values as $(2^3, 2^{-7.37})$ with average cross validation rate 83 $\%$. After the best (C, γ) is found whole training set is trained to generate final classifier. The proposed approach works well with thousands or more points.

8.4 Classification on synthetic data

Considering the synthetic data generated in section 8.1 Table 3 in Appendix B shows results on testing dataset. PFRVM accuracy is evaluated for different values of $C \in \{0.5,\ 1,\ 1.5,\ 2,\ 5,\ 10,\ 20\}$. The best value is when $C =$ 20. For larger C PFRSVM accuracy improves and error decreases. The number of false predictions is reported on testing dataset because of data size. PFRSVM outperforms SVM with same number of random samples. The FRSVM training time is almost 0.5 $\%$ of random samples. The sampling time for PFRSVM constructs 572 data points. With the growth of data size random sample gives similar accuracies as PFRSVM. The training time of SVM with random sampling is longer than PFRSVM. It is evident that using standard kernel functions good classification results are produced.

A larger dataset is generated according to parameters to verify PFRSVM performance (see Table 3 in Appendix B). The results of random sampling, MFSVM and PFRSVM on large dataset are also given (see Table 4 in Appendix B). Due to simple linear boundary on large training data random sampling does not increase MFSVM performance when sample size grows. The error rates of MFSVM and PFRSVM are approximately around 15 $\%$ lower than random sampling of highest performance. The total training time of PFRSVM including sampling time is less than MFSVM or random sampling of highest performance. For voluminous datasets MFSVM takes longer time than PFRSVM. MFSVM is executed with $\delta = 7$ starting from one positive and one negative sample and adding seven samples at each round yielding good results. The value of δ is set below 10. If δ is too high, its performance converges slower with larger amount of training data to achieve same accuracy. If δ is too low, MFSVM may need to undergo too many rounds.

8.5 Generation of outliers in real data

The outliers are generated from real datasets using distance based outliers algorithm [19]. Each point is ranked on basis of its distance to k^{th} nearest neighbor and top n points are declared as outliers. Also classical nested loop join and index join partition based algorithms are used for mining outliers. The input dataset are first partitioned into disjoint subsets. The entire partitions are pruned when they cannot contain outliers resulting in substantial savings in computation. The partition based algorithm scales well with respect to both dataset size and dimensionality. The performance results are dependent on number of points, k^{th} nearest neighbor, number of outliers and dimensions [26].

8.6 Classification on real data

Now we consider real datasets and study the comparative performance of PFRSVM with FRSVM. The experimental results using Gaussian RBF and Bayesian kernels are listed [17], [18] (see Tables 5 and 6 in Appendix B). Both training and testing rates are highlighted. For different datasets the values of C considered are 8 and 128. When Gaussian kernel is used PFRSVM achieve highest test rate. When bayesian kernel is used then also PFRSVM have better generation performance for Ionosphere and Semeion

Handwritten Digit datasets. The table also illustrates that PFRSVM has better generalization ability than FRSVM. Finally PFRSVM has better performance than FRSVM on reducing outliers' effects.

8.7 Classification with imbalance and overlapping classes

PFRSVM resolves the class overlapping combined with imbalance problem effectively. It is different from traditional classifiers using crisp decision producing high misclassifications rates. The soft decision of PFRSVM with optimized overlapping region detection addresses this. It provides multiple decision options. The overlapping region detected optimizes performance index that balances classification accuracy of crisp and cost of soft decisions. The optimized overlapping regions divide feature space into two parts with low and high confidence of correct classification. For test data falling into overlapping regions multiple decision options and measures of confidence are produced for analysis while for test data falling into non overlapping regions results in crisp decisions. The training procedure first builds fuzzy rough soft model using training data. The fuzzy rough information of training data $FR_{training}$ is used to search optimal threshold θ^* defining overlapping region. In testing stage incoming data is first processed to find its location. In feature space X overlapping region $R\left(\theta\right)$ is centered at decision boundary with margin at each boundary side. The width of margin is determined by threshold θ as given by Equation(27) in Appendix A. Here $FR\left(\omega_i|X\right)$ is posterior fuzzy rough measure of class i given X. The location of decision boundary $\left(FR\left(\omega_1|X\right) = FR\left(\omega_2|X\right)\right)$ is determined by class distribution of training data. In overlapping region detection the problem is determination of θ. In overlapping region detection two considerations should be taken (i) region should be large enough to cover most potentially misclassified data so that classification in non-overlapping region is highly accurate and (ii) region should be compact so as to avoid making soft decisions to too many patterns as patterns with soft decisions are verified by system and hence increase cost. To implement the stated considerations two criteria i.e. classification accuracy in non-overlapping regions $acc\left(\theta\right)$ and cost $c\left(\theta\right)$ are considered. To find a good tradeoff between $acc\left(\theta\right)$ and $c\left(\theta\right)$ an aggregate performance evaluation criterion is achieved through weighted harmonic mean $HM_\beta\left(\theta\right)$. The weight parameter β is predefined to attend to accuracy in non-overlapping region $\frac{\beta}{1-\beta}$ times as volume of non overlapping region. The default $\beta = 0.86$ since decreasing rate of $c\left(\theta\right)$ is always faster than increasing rate of $acc\left(\theta\right)$. The optimal threshold θ^* maximizes criterion $HM_\beta\left(\theta\right)$. By using this optimization criterion optimal volume of overlapping region is able to adapt to various data distributions and overlapping degrees. This criterion can be extended to multiple overlapping classes.

8.8 Generalization to unseen data when varying the size of training to test dataset

PFRSVM generalizes well to unseen data when size of training to test dataset is varied. It has been observed that dataset sizes have been growing steadily larger over the years. This leads development of training algorithms that scale at worst linearly with number of examples. Supervised learning involves analyzing given set of labeled observations (training set) so as to predict labels of unlabeled future data (test set). Specifically, the goal is to learn some function that describes relationship between observations and their labels. The interest parameter here is size of training set. The learning problem is called large scale if its training set cannot be stored in memory [25]. In large scale learning main computational constraint is time available rather than number of examples. In order for algorithms to be feasible on datasets they scale at worst linearly with number of examples.

The dual quadratic programming method in PFRSVM favors smooth handing of kernels. With focus of problem in large scale setting several methods have shifted back to primal. But dual is amenable to certain optimization techniques that converge quickly. The dual solvers used are special techniques to quickly achieve good solution. There are exponentially many constraints to problem which are expected for structural prediction. It is desirable that there is a single slack variable shared across each of these constraints. This affords some flexibility in solving the problem. The problem is solved through cutting plane method. The problem is approached iteratively by keeping working set W of constraints and restricted to constraints in W. Each element of W is vector $w \in \{0,\ 1\}^n$ and is considered as some combination of training point indices. The working set is updated each iteration to include indices for points that are currently misclassified. The algorithm terminates when it is within optimal primal solution and it achieves with appreciable time. However, training time increases such that it takes longer to reach an approximate solution.

8.9 Relaxation of dependency between features and labels

PFRSVM effectively relaxes dependencies between features of an element and its label. In this direction sequence labeling is used which identifies best assignment to collection of features so that it is consistent with dependencies set. The dependencies constrain output space. The dependencies are modeled with constraints so that it is a constrained assignment problem. To solve this, two-step process [27] is used that relies on constraint satisfaction algorithm viz. relaxation labeling. First PFRSVM classifier affects initial assignment to features without considering dependencies and then relaxation process applies successively to constraints to propagate information and ensure global consistency. It aims at estimating for each feature probability distribution over labels set. To produce these maximum entropy framework is adopted. It models joint distribution of labels and input features. The probability of labeling feature tr with label λ is modeled as exponential distribution:

$$prob\left(\lambda|tr;\theta\right)=\frac{1}{Z_{\theta}\left(tr\right)}exp\left\langle\theta,\phi\left(tr,\lambda\right)\right\rangle \quad (28)$$

Here $\phi\left(tr,\lambda\right)$ is feature vector describing jointly feature tr and label λ; $Z_{\theta}\left(tr\right)$ is normalizing factor and θ is parameter vector. To estimate θ maximum entropy framework advocates among all probability distributions that satisfy constraints imposed by training set the one with highest entropy. Relaxation labeling is an iterative optimization technique that solves efficiently assigning labels problem to features set satisfying constraints set. It reaches an assignment with maximal consensus among labels and feature sets. Denoting $TR=\{tr_1,\ldots..,tr_n\}$ set of n features, Λ is set of m possible labels and λ and ????? two features of Λ. The interactions between labels are denoted by compatibility matrix $CM=\{cm_{ij}\left(\lambda,\mu\right)\}$. The coefficient $cm_{ij}\left(\lambda,\mu\right)$ represents constraint and measures to which extent i^{th} feature is modeled with label λ when label of j^{th} feature is μ. These coefficients are estimated from training set. The algorithm starts from an initial label assignment from classifier. The relaxation iteratively modifies this assignment so that labeling globally satisfies constraints described by compatibility matrix. All labels are updated in parallel using information provided by compatibility matrix and current label assignment. For each feature tr_i and each label λ support function describes compatibility of hypothesis label of tr_i is λ and current label assignment of other features defined by:

$$q_i^{(t)}\left(\lambda;\overline{p}\right)=\sum_{j=1}^{n}\sum_{\mu\in\Lambda}cm_{ij}\left(\lambda,\mu\right)p_j^{(t)}\left(\mu\right) \quad (29)$$

In equation (29) $\overline{p}=\{\overline{p}_1,\ldots..,\overline{p}_n\}$ is weighted label assignment and each $p_j^{(t)}\left(\mu\right)$ in current confidence in hypothesis label of i^{th} feature is λ. The weighted assignment is updated to increase $p_i\left(\lambda\right)$ when $q_i\left(\lambda\right)$ is big and decrease it otherwise. More precisely update of each $p_i\left(\lambda\right)$ is:

$$p_i^{(t+1)}\left(\lambda\right)\leftarrow\frac{p_i^{(t)}\left(\lambda\right)\cdot q_i^{(t)}\left(\lambda,\overline{p}^{(t)}\right)}{\sum_{\mu\in\Lambda}p_i^{(t)}\left(\lambda\right)\cdot q_i^{(t)}\left(\lambda,\overline{p}^{(t)}\right)} \quad (30)$$

The calculation of support and mapping update is iterated until convergence.

8.10 Comparative classification performance of PFRSVM with other approaches

Keeping in view the results on real datasets a comparative average classification performance of PFRSVM is presented here with respect to parameters $C=8,\gamma=2^{-7.37}$ and Gaussian RBF and Bayesian kernels. The results are also highlighted (see Tables 7 and 8 in Appendix B). It is evident that PFRSVM achieves superior average classification accuracy percentage as compared to other SVM versions for both kernels.

8.11 Some critical issues regarding implementation of PFRSVM in cloud environment

PFRSVM implemented here presents a powerful method of huge datasets classification in distributed cloud environment. In this process we have discussed a new architecture of in-stream data analytics [24]. The parallel approach of computation offers a significant processing model for handling discontinuity in data thereby enabling the development of a scalable and reliable system [28]. The application processes real time streams of data which pushes the limits of traditional data processing architectures. This leads to a new class of system software viz. stream processing engine whose attributes are characterized by a core set of the following eight general rules [29]:

1. Always keep the data moving
2. Place stream based query from the database
3. Handle the stream imperfections through delayed, missing and out-of-order data
4. Always generate predictable outcomes
5. Effectively integrate the stored and streaming data
6. Always guarantee data safety and availability
7. Automatically partition and scale the applications
8. Always process and respond instantaneously

The architecture of PFRSVM as shown in figure 7 is developed keeping in view the above eight rules of stream processing [29]. The overall architecture of the system provides guaranteed order independence which is challenging and also vital in building scalable and reliable systems. In what follows we highlight various critical problems solved in order to build an enterprise class stream data classification engine [24]:

1. Architecture of stream processing engine: The architecture of the system is designed by splitting the entire computational workload into parallel phases that relies on partial processing. These partial phases are eventually combined together serially that operates using traditional approach. This entails identification of the portions of computation that are suitable for parallel processing, pushing partial workload processing right to the input manager of the data loading pipeline and hooking up the results of concurrent partial processing to serial processing [30]. The parallel

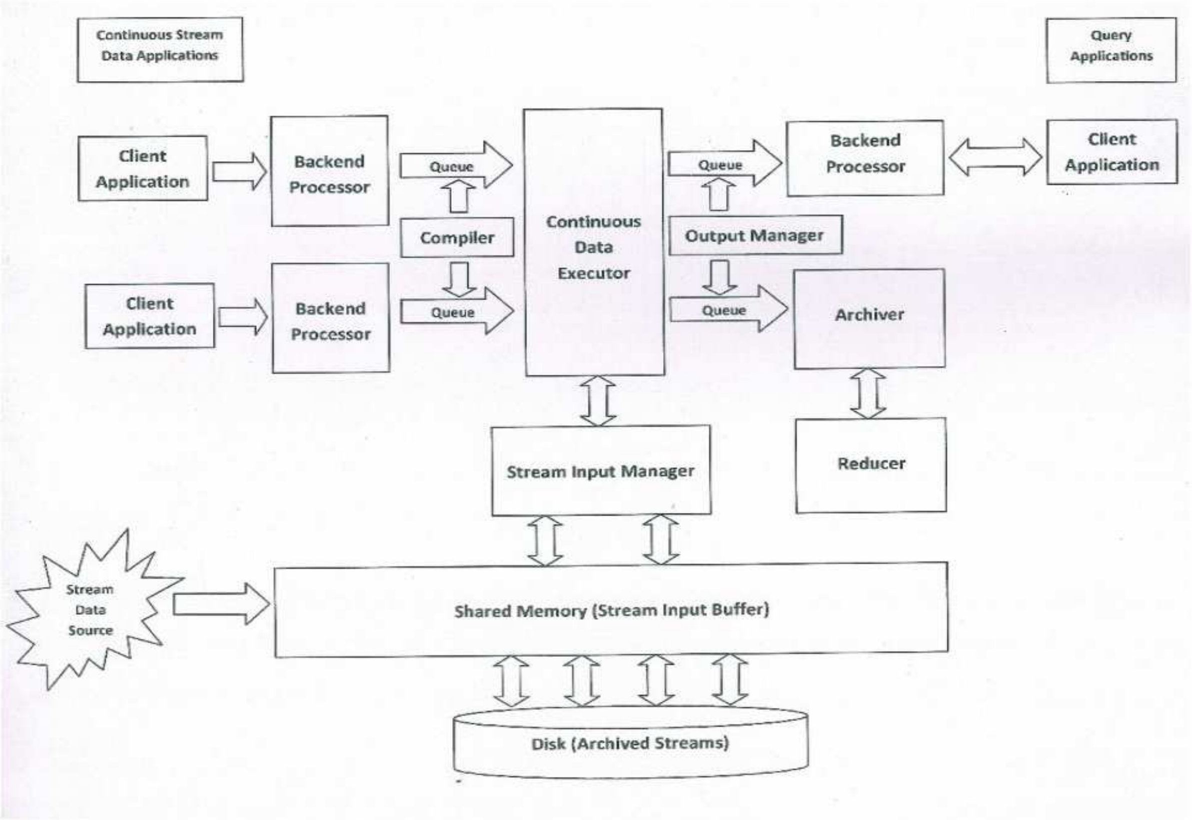


Figure 9: The architecture of PFRSVM.

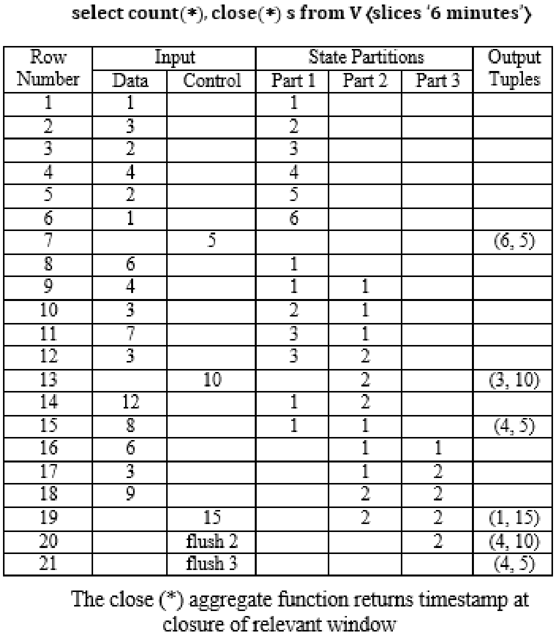
processing streams handle raw and intermediate derived streams earmarked for parallel phase and serial processing streams handle derived streams and other computations that operate in serial phase. The data sources are connected to the system using standard protocols and start pumping data into streams using bulk loading. The system forks an associated thread dedicated to each connection. This thread instantiates a local data flow network of computation operators on appropriate stream on which data arrives. These operators are specifically generated for parallel processing streams that correspond to particular raw stream and produces result runs. The local data flow network is also responsible for archiving raw data into tables it processes as well as corresponding parallel processing results it produces. It also sends parallel processing data to the shared workload executor for use in processing serial processing streams via shared memory queues [24]. The executor receives similar threads that services other data sources. As such the executor is part of a single very long running transaction for its entire existence. The executor thread fetches parallel processing records from input queues and processes them through a data flow network. The executor exploits multiple available cores in the system by splitting operations across multiple threads. The system also includes an archiver thread responsible for writing out windows of data produced for serial processing streams by the executor to tables. There also exists a reducer thread responsible for eagerly combining partial results in background and a repair thread that continually repairs contents of archives of serial processing streams. It is always possible to spawn several instances each executor, archiver, reducer and repair threads.

2. Order independence in data processing: In this computational framework order independence is always achieved for parallel processing streams. For serial processing streams order independence is implemented by processing out of order data by periodically repairing any affected archived serial processing data. The archives are always correct with respect to out of order data on an eventual consistency basis. It involves two activities viz. (i) spooling all data tuples that arrive too late into an auxiliary structure through a system generated corrections table and (ii) a repair process that periodically scan records from auxiliary table and combines them with an appropriate portion of originally arrived tuples in order to recompute and update affected portions of archived serial processing data. This approach is similar to the dynamic revision

of results as illustrated in [31].

3. Streamlining computational transaction: The computations in PFRSVM are defined through transactions which define the unit of work [24]. The transaction is associated with well-known ACID properties viz. atomicity, consistency, isolation and durability. The focus here is basically on atomicity, consistency and durability. Atomicity is vital in order to easily undo the effects of failures in either individual computations or the entire system. Consistency leads to integrity of data processed in the system. Durability is critical in order to recover state of the system after a crash. These properties are key attributors in the way data is loaded into the in-stream analytic system where loading application batches up a dataset and loads it in a single unit of work. This model is vital in connecting a transactional message queue with streaming system such that no records can ever be lost. This depends on the ability to abort a data loading transaction either based on an error condition which may occur in the loader, network or system. On abort it is vital that all modifications to raw and derived stream histories must be rolled back at least eventually. It is very challenging to support the abort requirement here because waiting until data is committed before processing leads to significant extra latency defeats the purpose of the streaming system and processing dirty uncommitted data from multiple transactions makes it hard to unwind the effects of a single transaction. The latter is particularly hard because archiving of serial processing streams is the responsibility of archiver threads that run their own transactions and are independent of thread that manages the unit of work in which data is loaded. The solution is achieved through two stages where we push down the partial processing and archiving results to input manager thread that handles the data loading transaction and we organize the data loading application into several concurrent units of work each of which loads one or more chunks of data. The data chunk is a finite subpart of stream that arises naturally as by product, the way data is collected and sent to a streaming system. Here the individual systems spool data into log files kept locally. These files are bulk loaded through standard interfaces and are often split at convenient boundaries based on number of records, size or time and are sent separately to the stream processing engine. They also serve as natural units for data chunks with clear boundaries.

   Consider an example of order independent partial processing transaction where certain types of operations are implemented that are tolerant to appreciable amounts of disorder in their input [24]. The transaction adheres to all the ACID properties stated earlier. Let us take tumbling or non-overlapping window count query with window of 6 minutes shown in figure 8 which operates over a stream of data with many records arriving out-of-order and input manager provides progress information on heuristic basis. This has resulted in 6 out-of-order tuples being discarded. Here we have 3 columns each representing the state of an individual partition. The first window returns the behavior of an out-of-order processing approach. The second window the arrival of an out-of-order tuple with timestamp 3 (row 10) reduces the system to second partition. When an out-of-order tuple with timestamp 3 arrives during that same window it is handled in second partition as it is still in-order relative to that partition. When tuple with timestamp 6 (in row 16) comes in during third window its timestamp is high enough to cause the open window of second partition to close producing partial result of (2, 5) and processing new tuple in second partition associated with second window ending at time 10. When next two tuples (rows 17 and 18) with timestamps 3 and 9 come in they are too late to be processed in second partition and requires the system to proceed to third partition where they are sent. Next tuple with timestamp 9 (row 18) comes in and is sent to second partition. When system receives a control tuple with timestamp 15 it flushes second and third partitions producing partial results of (2, 10) and (2, 5).



select count(*), close(*) s from V ⟨slices '6 minutes'⟩

Row Number	Input		State Partitions			Output Tuples
	Data	Control	Part 1	Part 2	Part 3	
1	1		1			
2	3		2			
3	2		3			
4	4		4			
5	2		5			
6	1		6			
7		5				(6, 5)
8	6		1			
9	4		1	1		
10	3		2	1		
11	7		3	1		
12	3		3	2		
13		10		2		(3, 10)
14	12		1	2		
15	8		1	1		(4, 5)
16	6			1	1	
17	3			1	2	
18	9			2	2	
19		15		2	2	(1, 15)
20		flush 2			2	(4, 10)
21		flush 3				(4, 5)

The close (*) aggregate function returns timestamp at closure of relevant window

Figure 10: A transaction illustrating the Order Independent Partial Processing adhering ACID properties.

4. Recovery from failure: Once the system fails it is recovered by bringing the system back to a consistent state after a crash when all in flight transactions during the crash are deemed aborted [24]. Here the recovery archives of parallel processing streams are free since all writes of raw and corresponding derived data hap-

pen as part of the same transaction. We benefit both from the robust recovery architecture of the storage subsystem and also from other features such as online backup mechanisms. The recovery operation for serial processing is a more challenging because the large amounts of runtime state managed in main memory structures by operators in executor and the decoupled nature in which durable state is written out originally by the archiver. The crash recovery therefore involves standard database style recovery of all durable state, making all serial processing archives self-consistent with each other as well as the latest committed data from their underlying archive and rebuilding the runtime state of the various operators in executor. The ability to have robust recovery implementation that is capable of quickly recovering from a failure is essential. Furthermore, the longer it takes to recover from a failure the more the amount of pent-up data has gathered and the longer it takes to catch up to the live data.

The recovery from failure in distributed PFRSVM is assessed in terms of delay, process and correct (DPC) protocol which handles crash failures of processing nodes and network failures [24]. Here, the choice is made explicit as the user specifies an availability bound and it attempts to minimize the resulting inconsistency between the node replicas while meeting the given delay threshold. The protocol tolerates occurrence of multiple simultaneous failures that occur during recovery. In DPC each node replica manages its own availability and consistency by implementing state machine as shown in figure 9 that has three states viz. STABLE, UPSTREAM FAILURE (UP FAILURE), and STABILIZATION.

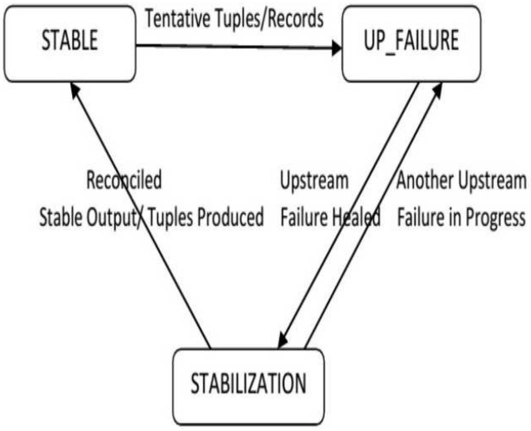


Figure 11: The DPC State Machine.

The DPC provides eventual consistency even when multiple failures overlap in time and with at least two replicas of any processing node it maintains the required availability at all times. In both scenarios client applications eventually receive stable version of all result tuples and there are no duplications. A single processing node with no replica, no upstream and downstream neighbors is executed and its inputs are controlled directly. The node produces complete and correct output stream tuples with sequentially increasing identifiers as shown in figure 10.

First a failure is injected on input stream 1 (Failure 1) and then on input stream 3 (Failure 2). Figure 11(a) shows the output when two failures overlap in time and figure 11(b) shows the output when Failure 2 occurs exactly at the moment when Failure 1 heals and node starts reconciling its state.

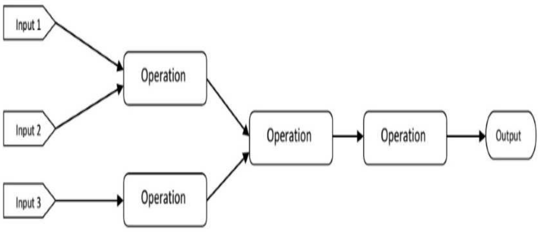


Figure 12: The Query Diagram used in Simultaneous Failures Experiments.

Here the node temporarily loses one or two of its input streams and there are no other replicas that could be reconnected to. When another replica exists for an upstream neighbor the node tries to switch to that replica. When a node switches to a different replica of an upstream neighbor there is a short gap in data it receives. In this prototype implementation we measured that it takes a node approximately 30 milliseconds to switch between upstream neighbors once the node detects failure. Failure detection time depends on the frequency with which downstream node sends keep-alive requests to its upstream neighbors. With a keep-alive period of 100 milliseconds it takes at most 130 milliseconds between the time a failure occurs and the time a downstream node receives data from a different replica of its upstream neighbor. For many application domains it is expected that this value is much smaller than minimum incremental processing latency that the application tolerates. If the application cannot tolerate a short delay the effect of switching upstream neighbors is same as the effect of failure 1 as shown in figure 11(a) but without subsequent failure 2 i.e. the downstream node first suspends and then produces tentative tuples. Once reconnected to new upstream neighbor the downstream node goes back and corrects tentative tuples it produced during the switch. The

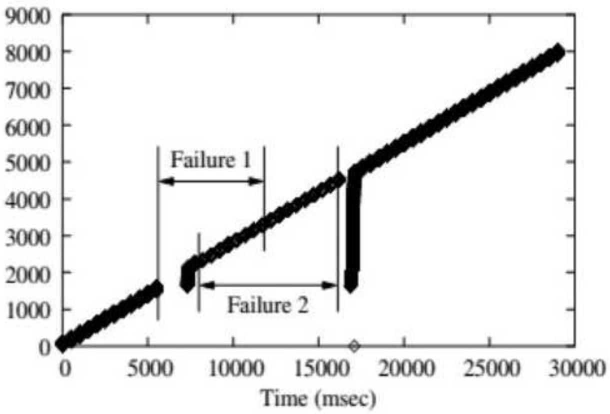


Figure 13: Overlapping Failures.

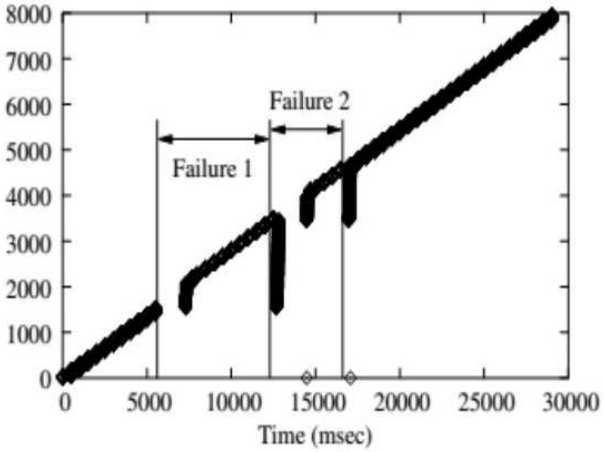


Figure 14: Failure during Recovery

Figure 15: The Outputs with Simultaneous Failures.

maximum incremental latency is set as 3 seconds. Figure 11 shows the maximum gap between new tuples remains below that bound at any time. However node manages to maintain the required availability with sufficiently short reconciliation time.

For longer duration failures reconciliation easily takes longer than the maximum latency bound. The DPC relies on replication to enable a distributed PFRSVM to maintain a low processing latency by ensuring that at least one replica node always processes most recent input data within required bound. To demonstrate this we use the experimental setup as shown in figure 12. We create a failure by temporarily disconnecting one of the input streams without stopping data source. After the failure heals data source replays all missing tuples while continuing to produce new tuples.

The Table 9 in Appendix B shows the maximum processing latency measured at client for failures with

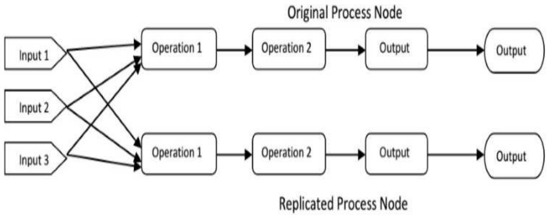


Figure 16: The Experimental Setup for Consistency and Availability tradeoffs for Single Node.

different durations. As the experiment is of deterministic nature each result is an average of three experiments. All values measured are within a few percent of the reported average. The client always receives new data within required 4 second bound. Each node processes input tuples as they arrive without trying to delay them to reduce inconsistency. When the failure first occurs both node replicas suspend for the maximum incremental processing bound and then return to processing tentative tuples as they arrive. After the failure heals DPC ensures that only one replica node at a time reconciles its state while remaining replica nodes continue processing the most recent input data. Once the first replica reconciles its state and catches up with current execution then other replica node reconcile its state in turn. The client application has thus access to the most recent data at all times. The major overhead of DPC lies in buffering tuples during failures in order to replay them during state reconciliation. The overhead is in terms of memory and it does not affect runtime performance. There are however additional sources of overhead that affect runtime performance [24]. To evaluate overhead of these delays the experimental setup is shown in figure 13 which produces appreciable results in terms of failure recovery.

5. Atomicity of transaction: All the data that is generated here through the parallel processing stream archive is automatically consistent with respect to unit of work into the underlying base stream. It is critical that same atomicity property is also supported for archives of serial processing streams in order for recovery and high availability to work correctly. One simple approach to facilitate atomicity is to delay processing of any data until it has been committed. Waiting for commits unfortunately introduces latency within the system. What is really required is the ability to offer a strong guarantee about atomicity and durability of data loaded in the system within a single unit of work without compromising on immediate processing of data. This calls for speculatively processing dirty uncommitted data in a laissez-faire fashion based on the assumption that errors and transaction aborts are few and far between [24]. When a transaction is ac-

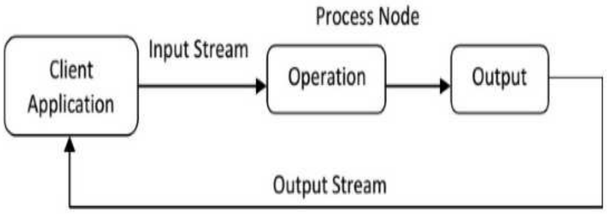


Figure 17: Fault Tolerance Setup.

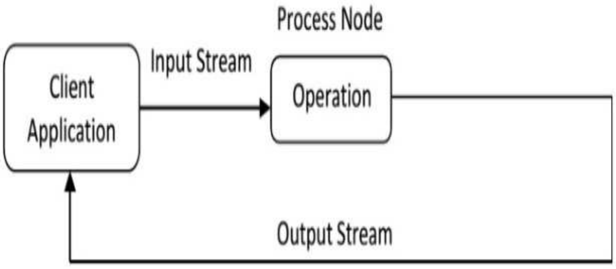


Figure 18: Setup without Fault Tolerance

Figure 19: The Experimental Setup for Fault Tolerance Overhead Experiments.

tually aborted the system asynchronously repairs associated serial processing archives similar to the way repair guarantees order-independence on an eventual consistency basis.

6. Fault tolerance and high availability: The data driven parallel approach used here provides a natural path towards scalability. The next concern is towards enhancing it in order to provide fault tolerance and high availability in the system [32]. The fault tolerance is defined as the ability of system to react well from any kind of extreme or catastrophic error which may happen either from streaming engine itself in the application or in some aspect of hardware or software environment. In particular quick recovery from failure state is critical in realizing fault tolerance. The high availability is characterized as the ability of system to remain up even in the face of any catastrophic error. It is generally realized using additional backup resources that are organized together in either an active-standby or active-active configuration. The unit of work and recovery functionality of the system highlighted earlier serve as key building blocks for fault tolerance and high availability in PFRSVM classification system. This implementation supports a comprehensive fault tolerance and high availability solution by organizing a cluster of FRSVM nodes in the cloud environment in a multi-master active-active configuration. Here same workload are typically running on all nodes of the cluster. Any incoming run of data can be sent to any but only one node in the cluster. It is then the responsibility of a special stream replicator component in each computing node to communicate the complete contents of each run to the peers in the cluster. The runs of data that are populated in a stream by peer are treated just like any other incoming data except that they are not further re-replicated to other nodes. This model has one eventual consistency such that run replication procedure happens on an asynchronous basis. In this way there is very small amount of risk of data loss in the event of any catastrophic media failure between a run getting committed and replicated to peer. The order independent infrastructure plays significantly in realizing simple fault tolerant and high availability architecture. Since each individual node in computing cluster accepts data in any order different nodes stay loosely coupled and implement simple and easy to verify protocols. When a node recovers from failure it immediately starts accepting new transactions and patch up the data it has missed asynchronously. As a node fails in the cluster application layer takes the responsibility in directing all workloads to other nodes. After the failed node is brought back online it captures the entire data it missed while being non-functional. This is accomplished by replicator component using a simple protocol that tracks both replicated and non-replicated runs. PFRSVM executes by parallelizing in-stream dataflow across cluster of workstations in a cost effective way by scaling the high throughput applications. We can imagine having a large number of simultaneous sessions and sources. To keep up with high throughput input rates and low latencies dataflow can be scaled by partitioning it across a cluster. On the cluster in-stream dataflow is a collection of dataflow segments which may be one or more per machine. The individual operations are parallelized by partitioning input and processing across the cluster. When the partitions of an operation need to communicate to nonlocal partitions of the next operation in the chain communication occurs through exchange architecture [24] as shown in figure 14.

In this configuration a scheme that naively applies dataflow pair technique without accounting for cross machine communication within parallel dataflow quickly becomes unreliable. It is shown that a cluster pair approach leads to a mean-time-to-failure (MTTF) that falls off quadratically with number of machines. Also the parallel dataflow must stall during recovery thereby reducing the availability of system. By embedding the coordination and recovery logic within exchange architecture speeds up mean-time-to-recovery (MTTR) thereby improving both availability and MTTF. The improved MTTF falls off linearly with the number of machines. The flux design achieves the desired MTTF [24] as shown in figure

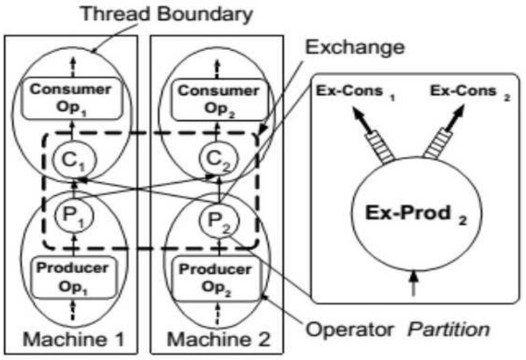


Figure 20: Exchange Architecture.

15.

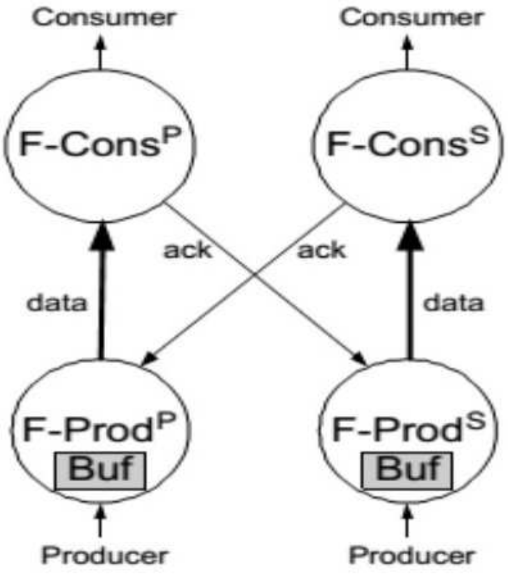


Figure 21: The Flux Design and Normal Case Protocol.

Now we illustrate benefits accrued from the design of PFRSVM by examining performance of parallel implementation of dataflow which highlights its fault tolerant behavior [24]. We implemented streaming group-by aggregation operations, boundary operations and flux within in-stream open source code base. We partition this dataflow across five machines in a cluster and place the operators to perform operations on a separate sixth machine. The operators are supplemented by a duplicate elimination buffer. The flux is inserted after the first group-by operator to repartition its output. Initially we place a partition of each operator on each of the five machines 0 to 4 and replicate them using a chained declustering strategy. Thus each primary partition has its replica on the next machine and last partition has its replica on the first. In this configuration when a single machine fails all five survival scenarios occur in different partitions. At the beginning we introduce a standby machine with operators in their initial state. We did not implemented the controller as it has been implemented by standard cluster management software [24]. We simulate failure by killing the in-stream process on one of those machines which causes connections to that machine to close and raise an error. Each machine is connected to a 100 mbps switch. To approximate workload of high throughput network monitoring the operator generates sequentially numbered session start and end events as fast as possible. With this setup figure 16 shows the total output rate (throughput) and average latency per tuple.

Here the main bottleneck is the network. At t = 20 sec when steady state is reached one of the five machines is killed. The throughput remains steady for some time and then suddenly drops. The drop occurs because during state movement the partition being recovered is stalled and eventually causes all downstream partitions to also stall. Here about 8.9 MB of state was transferred in 941 msec. Once catch up is finished at t = 21 seconds a sudden spike in throughput is observed. This spike occurs because during movement all queues to the unaffected partitions are filled and ready to be processed once catch-up completes. Figure 16 shows an increase in latency because input and in-fight data are buffered during movement. Then the output rate and average latency settle down to normal. During this entire process the input rate stayed at constant 42000 tuples/sec with no data dropped.

This investigation illustrates that with piecemeal recovery and sufficient buffering we can effectively mask the effects of machine failures. To understand the overheads of flux we added enough CPU processing to the lower level group-by to make the bottleneck. Here for single parallel dataflow the input rate was 83000 tuples/sec; for cluster pairs it was 42000 tuples/sec and for flux it was 37000 tuples/sec. Additional processing only reduces flux overhead relative to others.

7. Bootstrapping the system: When a live workload is added to the streaming classification system on an adhoc basis it is easy to see only the data that arrives after the request is submitted. This approach is quite reasonable if the request involves small windows involving few seconds only. In many situations this is not the case as requests may continue for hours and a naive streaming implementation does not produce complete results until steady state is reached. This entails bootstrapping feature requirement from the system [24]. This exploits any available archives of the underly-

ing raw or derived stream that workload is based on by reaching into the archive and replaying history in order to build up runtime state of the request and produce complete results as early as possible. In addition, if a new workload has an associated archive there is often a need for the system to populate this archive with all data already in the system prior to the request.

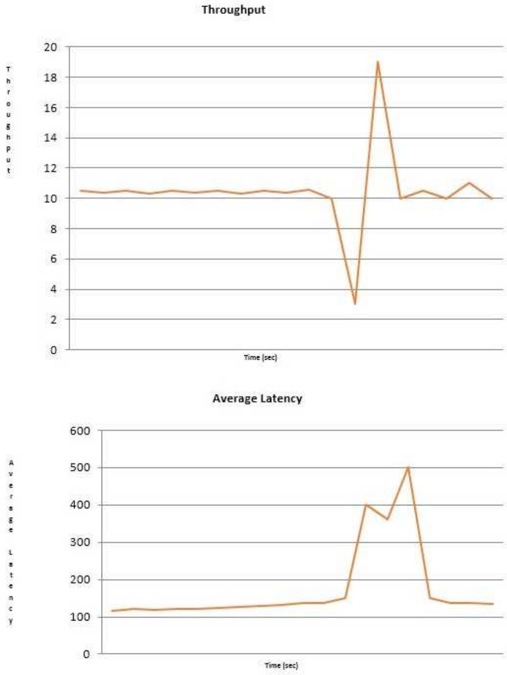


Figure 22: System Performance (Throughput and Average Latency per Tuple).

9 Conclusion

In this work we propose a robust parallel version of fuzzy support vector machine i.e. PFRSVM to classify and analyze useful information from huge datasets present in cloud environment. It is an in-stream data classification engine which adheres to the fundamental rules of stream processing. The classifier is sensitive to noisy data samples. The fuzzy rough membership function brings sensitivity to noisy samples and handles impreciseness in training samples giving robust results. The decision surface sampling is achieved through membership function. The larger membership function value increases the importance of corresponding point. The classification success lies in proper selection of fuzzy rough membership function. PFRSVM success is attributed towards choosing appropriate parameter values. The training samples are either linear or nonlinear separable. In nonlinear training samples, using linear separating input space is mapped into high dimensional feature space to compute separating surface. PFRSVM effectively addresses nonlinear classification and imbalance and overlapping class problems. It generalizes to unseen data and relaxes dependency between features and labels. PFRSVM performance is also assessed in terms of number of support vectors. We use MapReduce technique to improve scalability and parallelism of split dataset training. The research work is performed on cloud environment available at University of Technology and Management, India on real datasets from UCI Machine Learning Repository. The experiments illustrate the convergence of PFRSVM. The performance and generalization of the algorithm are evaluated through Hadoop. The algorithm works on cloud systems dealing with large scale dataset training problems without having any knowledge about the number of computers connected to run in parallel. The experimental results have been reported here for $C = 8$ and 128 using Gaussian RBF and Bayesian kernels. For Gaussian kernel both PFRSVM and FRSVM achieve highest test rate. For Bayesian kernel both algorithms have better generalization performance for Ionosphere and Semeion Handwritten Digit datasets. Further PFRSVM has better generalization ability than FRSVM. PRFSVM also has better performance on reducing the outliers' effects. Empirically prediction variability, generalization performance and risk minimization of PFRSVM is better than existing SVMs. The average classification accuracy of PFRSVM is better than SVM, FSVM and MFSVM for Gaussian RBF and Bayesian kernels. The experimental results on both synthetic and real datasets clearly demonstrate the superiority of proposed technique. The stream classification engine is scalable and reliable and maintains order independence in data processing, streamlines computational transaction, recovers from failure, generates atomic transactions and are fault tolerant and highly available in nature.

Acknowledgement

The author extends his gratitude to the technical staff of Samsung Research and Development Institute Delhi, India for providing support through the entire course of this research work.

References

[1] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh and A. H. Byers, Big Data: The next frontier for innovation, competition, and productivity, Technical Report, McKinsey Global Institute, McKinsey and Company, 2011.

[2] M. B. Miles, M. A. Huberman and J. Saldaña, *Qualitative data analysis*: *A methods sourcebook*. SAGE Publications, 2014.

[3] J. Han, M. Kamber and J. Pei, *Data mining*: *Concepts and techniques*, San Francisco: Morgan Kaufmann Publishers, 2011.

[4] J. Canny and H. Zhao, Big data analytics with small footprint: Squaring the cloud, *In 2013 Proc. Nineteenth ACM SIGKDD Conf. on Knowledge Discovery and Data Mining*, pp. 95–103.

[5] C. Statchuk and D. Rope, Enhancing Enterprise Systems with Big Data, Technical Report, IBM Business Analytics Group, IBM Corporation, 2013.

[6] J. Leskovec, A. Rajaraman and J. D. Ullman, *Mining of Massive Datasets*. Cambridge University Press, 2014.

[7] HDFS (Hadoop Distributed File System) Architecture: http://hadoop.apache.org/common/docs/ current/hdfs_design.html, 2009.

[8] K. Hwang, G. C. Fox and J. J. Dongarra, *Distributed and Cloud Computing*: *From Parallel Processing to Internet of Things*. Morgan Kaufmann, 2011.

[9] E. Capriolo, D. Wampler and J. Rutherglen, *Programming Hive*. O'Reilly Media, 2012.

[10] J. Abonyi, B. Feil and A. Abraham, Computational Intelligence in Data Mining, Informatica, vol. 29, no. 1, pp. 3—12, 2005.

[11] D. Dubois and H. Prade, Putting Rough Sets and Fuzzy Sets together, In R. Slowinski (Editor) *Intelligent Decision Support*, Handbook of Applications and Advances of the Rough Set Theory, pp. 203—232, Kluwer Academic Publishers, 1992.

[12] C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, vol. 2, no. 2, pp. 121—167, 1998.

[13] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer Verlag, 2007.

[14] Q. Hu, S. An, X. Yu and D. Yu, Robust Fuzzy Rough Classifiers, Fuzzy Sets and Systems, vol. 183, no.1, pp. 26—43, 2011.

[15] A. Chaudhuri, *Data Classification through Fuzzy and Rough versions of Support Vector Machines*: *A Survey*. Technical Report, Samsung Research and Development Institute Delhi, 2014.

[16] V.N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[17] A. Chaudhuri, K. De, and D. Chatterjee, A Comparative Study of Kernels for Multi-Class Support Vector Machine, *In 2008 Proc. Fourth Conf. on Natural Computation*, vol. 2, pp. 3—7.

[18] A. Chaudhuri and K. De, Fuzzy Support Vector Machine for Bankruptcy Prediction, Applied Soft Computing, vol. 11, no. 2, pp. 2472—2486, 2011.

[19] A. Chaudhuri, Modified Support Vector Machine for Credit Approval Classification, AI Communications, vol. 27, no. 2, pp. 189—211, 2014.

[20] H. J. Zimmermann, *Fuzzy Set Theory and its Applications*. Boston: Kluwer Academic, 2001.

[21] S. Perera and T. Gunarathne, *Hadoop MapReduce Cookbook*. Packt Publishers, 2013.

[22] R. Bekkerman, M. Bilenko and J. Langford, *Scalable Machine Learning*. Cambridge University Press, 2012.

[23] C. C. Chang and C. J. Lin, LIBSVM: A Library for Support Vector Machines, ACM Transactions on Intelligent Systems and Technology, vol. 2, no. 3, pp. 1—27, 2011.

[24] A. Chaudhuri, *Studies on Parallel SVM based on MapReduce*. Technical Report, Birla Institute of Technology Mesra, Patna Campus, India, 2010.

[25] I. W. Tsang, J. T. Kwok and P. M. Cheung, Core Vector Machines: Fast SVM Training on very Large Datasets, Journal of Machine Learning Research, vol. 6, pp. 363—392, 2005.

[26] S. Ramaswamy, R. Rastogi and K. Shim, Efficient Algorithms for Mining Outliers from Large Datasets, *In 2000 Proc. ACM SIGMOD Conf. on Management of Data*, pp. 427—438.

[27] V. Punyakanok, D. Roth, W. Tau Yih and D. Zimak, Learning and Inference over Constrained Output, *In 2005 Proc. 19th Joint Conf. on Artificial Intelligence*, pp. 1124—1129.

[28] B. Ellis, *Real Time Analytics*: *Techniques to Analyze and Visualize Streaming Data*. John Wiley and Sons, 2014.

[29] M. Stonebraker, U. Cetintemel and S. Zdonik, *The Eight Rules of Real Time Stream Processing*. White Paper, StreamBase Systems, MA, United States, 2010.

[30] J. L. Hennessy and D. A. Patterson, *Computer Architecture – A Quantitative Approach*. 5th Edition, Morgan Kaufmann Publications, Elsevier Inc., 2012.

[31] Borealis: Second Generation Stream Processing Engine: http://nms.lcs.mit.edu/projects/borealis, 2003.

[32] R. Jhawar, V. Piuri and M. Santambrogio, Fault Tolerance Management in Cloud Computing: A System Level Perspective, IEEE Systems Journal, vol. 7, no. 2, pp. 288-–297, 2013.

Appendix A

$rm_A(p) = \frac{\|[p]_R \cap A\|}{\|[p]_R\|}$; $\|\mathrm{A}\|$ is cardinality of set A (4)

$$\mu_{\underline{C_c}}(FC_i) = \underbrace{inf}_{x \in C_c} max\left\{1 - \mu_{FC_i}(p), \mu_{C_c}(p)\right\} \quad \forall p \quad (5)$$

$$\min \quad \frac{1}{2}\|\mathrm{w}\|^2 + C\sum_{i=1}^{M} frm_i(p)\xi_i$$

$$\text{subject to:} \left\{ \begin{array}{c} z_i\left(\mathrm{w}^{\mathrm{T}}\Phi\left(Y_i\right) + b\right) \geq 1 - \xi_i, i = 1, .., M; \\ \xi_i \geq 0 \end{array} \right. \quad (10)$$

$$rd_+^2 = \frac{1}{n}\max\|\Phi(Y') - \Phi_+\|^2 = \frac{1}{n}\max\left[\Phi^2(Y') - 2\Phi(Y')\cdot\Phi_+ + \Phi_+^2\right] =$$

$$\frac{1}{n}\max\left[\begin{array}{c} \Phi^2(Y') - \frac{2}{m_+}\sum_{Y_i \in C^+} tanh\left[\Phi(Y_i)\cdot\Phi(Y')\right] + \\ \frac{1}{m_+^2}\sum_{Y_i \in C^+}\sum_{Y_j \in C^+} tanh\left[\Phi(Y_i)\cdot\Phi(Y_j)\right] \end{array}\right]$$

$$= \frac{1}{n}\max\left[\begin{array}{c} K(Y', Y') - \frac{2}{m_+}\sum_{Y_i \in C^+} K(Y_i, Y') + \\ \frac{1}{m_+^2}\sum_{Y_i \in C^+}\sum_{Y_j \in C^+} K(Y_i, Y_j) \end{array}\right] \quad (15)$$

$$rd_-^2 = \frac{1}{n}\max\left[K(Y', Y') - \frac{2}{m_-}\sum_{Y_i \in C^-} K(Y_i, Y') + \frac{1}{m_-^2}\sum_{Y_i \in C^-}\sum_{Y_j \in C^-} K(Y_i, Y_j)\right] \quad (16)$$

$$frm_i(p) = \left\{ \begin{array}{c} 1 - \left(\frac{\sum_{i=1}^{H}\mu_{FC_i}(p)\tau_{C_c}^i}{\sum_i \mu_{FC_i}(p)}\right)\sqrt{\frac{\left\|dist_{i+}^2\right\| - \left\|dist_{i+}^2\right\|\cdot rd_+^2 + rd_+^2}{\left(\left\|dist_{i+}^2\right\| + \left\|dist_{i+}^2\right\|\cdot rd_+^2 + rd_+^2\right) + \varepsilon}} \quad (z_i = 1) \\ 1 - \left(\frac{\sum_{i=1}^{H}\mu_{FC_i}(p)\tau_{C_c}^i}{\sum_i \mu_{FC_i}(p)}\right)\sqrt{\frac{\left\|dist_{i-}^2\right\| - \left\|dist_{i-}^2\right\|\cdot rd_-^2 + rd_-^2}{\left(\left\|dist_{i-}^2\right\| + \left\|dist_{i-}^2\right\|\cdot rd_-^2 + rd_-^2\right) + \varepsilon}} \quad (z_i = -1) \end{array} \right. \quad (19)$$

$$frm_i^c(p) = \left\{ \begin{array}{c} 1 - \left(\frac{\sum_{i=1}^{H}\mu_{FC_i}(p)\tau_{C_c}^i}{\hat{H}}\right)\sqrt{\frac{\left\|dist_{i+}^2\right\| - \left\|dist_{i+}^2\right\|\cdot rd_+^2 + rd_+^2}{\left(\left\|dist_{i+}^2\right\| + \left\|dist_{i+}^2\right\|\cdot rd_+^2 + rd_+^2\right) + \varepsilon}} \quad (z_i = 1) \\ 0 \qquad (z_i = -1) \end{array} \right. \quad (20)$$

$$\max H^{i,j} = -\frac{1}{2}\left[\begin{array}{c} \alpha_1 \\ \alpha_2 \end{array}\right]^T \left[\begin{array}{cc} KM_{11} & KM_{12} \\ KM_{21} & KM_{22} \end{array}\right]\left[\begin{array}{c} \alpha_1 \\ \alpha_2 \end{array}\right] + \left[\begin{array}{c} 1 \\ 1 \end{array}\right]^T \left[\begin{array}{c} \alpha_1 \\ \alpha_2 \end{array}\right]$$

subject to:

$$0 \leq \alpha_i \leq C \;\; \forall i \;\; \wedge \;\; \sum_{i=1}^{I}\alpha_i z_i = 0 \qquad (26)$$

$$R(\theta) = \{X : |(FR(\omega_1|X) - FR(\omega_2|X))| < \theta\} \qquad (27)$$

Appendix B

Dataset	Characteristics	No. of Instances	No. of Attributes
German Credit	Multivariate, Categorical, Integer	1000	20
Heart Disease	Multivariate, Categorical, Integer	303	75
Ionosphere	Multivariate, Integer	351	34
Semeion Handwritten Digit	Multivariate, Integer	1593	256
Landsat Satellite	Multivariate, Integer	6435	36

Table 2: The UCI Machine Learning Datasets used.

	SVM	PFRSVM	0.5 % samples
Number of data points	114996	572	602
SVM training time (sec)	160.796	0.002	0.002
Sampling time (sec)	0.0	9.569	3.114
Number of false predictions (Number of false positives, Number of false negatives)	75 (50, 25)	69 (58, 11)	237 (219, 18)

Table 3: Experimental Results on Synthetic Dataset (Number of Training Data = 124996, Number of Testing Data = 107096).

Sampling rate (%)	Number of data	Number of errors	Training time	Sampling time
0.0001	24	6423	0.000114	823.00
0.001	230	2399	0.000969	825.00
0.01	2344	1125	0.02	828.00
0.1	23475	1009	6.286	834.25
1	234386	1014	1189.786	838.86
5	1151719	1019	20705.4	842.69
MFSVM	2308	859	2.969	2796.214
PFRSVM	3896	862	1.422	2236.219

Table 4: Experimental Results on Large Synthetic Dataset (Number of Training Data = 24069196, Number of Testing Data = 242896).

PFRSVM with Gaussian RBF Kernel			
Datasets	Support Vectors	Training Rate %	Test Rate %
German Credit	606	85.0	86.0
Heart Disease	137	65.5	73.7
Ionosphere	160	69.0	77.7
Semeion Handwritten Digit	737	89.4	93.5
Landsat Satellite	1384	95.0	97.0
FRSVM with Gaussian RBF Kernel			
German Credit	636	79.7	78.9
Heart Disease	165	60.7	69.0
Ionosphere	180	65.0	74.9
Semeion Handwritten Digit	765	84.5	86.8
Landsat Satellite	1407	93.0	93.7
PFRSVM with Bayesian Kernel			
German Credit	640	89.0	85.7
Heart Disease	145	67.7	69.9
Ionosphere	157	70.7	75.0
Semeion Handwritten Digit	735	89.0	89.7
Landsat Satellite	1396	96.9	96.9
FRSVM with Bayesian Kernel			
German Credit	645	83.0	78.0
Heart Disease	175	64.8	65.8
Ionosphere	177	67.0	70.7
Semeion Handwritten Digit	755	86.8	83.8
Landsat Satellite	1425	96.0	89.8

Table 5: Experimental Results of PFRSVM and FRSVM on different Datasets with $C = 8$.

PFRSVM with Gaussian RBF Kernel			
Datasets	Support Vectors	Training Rate %	Test Rate %
German Credit	596	95.2	96.0
Heart Disease	127	77.5	88.7
Ionosphere	142	82.0	92.7
Semeion Handwritten Digit	727	97.9	94.5
Landsat Satellite	1362	96.0	97.2
FRSVM with Gaussian RBF Kernel			
German Credit	625	94.0	93.5
Heart Disease	154	75.0	84.7
Ionosphere	172	80.9	89.7
Semeion Handwritten Digit	754	97.7	91.5
Landsat Satellite	1392	94.2	94.6
PFRSVM with Bayesian Kernel			
German Credit	632	97.2	95.7
Heart Disease	147	82.7	82.8
Ionosphere	145	86.7	87.2
Semeion Handwritten Digit	725	98.0	98.6
Landsat Satellite	1386	97.0	97.6
FRSVM with Bayesian Kernel			
German Credit	635	96.9	93.7
Heart Disease	165	79.9	80.0
Ionosphere	165	83.0	85.0
Semeion Handwritten Digit	745	97.7	97.0
Landsat Satellite	1402	95.2	95.7

Table 6: Experimental Results of PFRSVM and FRSVM on different Datasets with $C = 128$.

Dataset	SVM	FSVM	MFSVM	PFRSVM
German Credit	78.3	80.5	82.9	94.8
Heart Disease	83.5	86.9	87.0	95.9
Ionosphere	85.7	87.7	89.0	96.5
Semeion Handwritten Digit	82.7	86.0	86.5	95.5
Landsat Satellite	86.8	86.9	87.2	96.2

Table 7: The Average Classification Accuracy Percentage of different versions of SVM with Gaussian RBF Kernel.

Dataset	SVM	FSVM	MFSVM	PFRSVM
German Credit	77.7	78.0	80.7	94.2
Heart Disease	83.4	85.5	86.0	94.9
Ionosphere	85.5	86.0	87.9	95.5
Semeion Handwritten Digit	80.8	85.0	85.8	94.7
Landsat Satellite	86.9	87.2	87.8	96.6

Table 8: The Average Classification Accuracy Percentage of different Versions of SVM with Bayesian Kernel.

Failure Duration (seconds)	Maximum Processing Latency (seconds)
2	2.1
4	2.7
6	2.7
8	2.7
10	2.7
12	2.7
14	2.7
16	2.7
30	2.7
45	2.7
60	2.7

Table 9: The Maximum Processing Latency for different Failure Durations and for Single Node Deployment with One Replica.

Data-intensive Service Mashup Based on Game Theory and Hybrid Fireworks Optimization Algorithm in the Cloud

Wanchun Yang
School of Electronics and Information Engineering, Tongji University, Shanghai 201804, China
School of Sciences, Shandong Jiaotong University, Jinan 250357, China
E-mail:yangwch1982@126.com

Chenxi Zhang* and Bin Mu
School of Software Engineering, Tongji University, Shanghai 201804, China
E-mail: chenxizhang10@126.com, binmu@tongji.edu.cn
*Corresponding author

Keywords: cloud computing, data-intensive, mashup, hybrid fireworks optimization algorithm, game theory, service correlation

Received: June 2, 2015

End users can create kinds of mashups which combine various data-intensive services to form new services. The challenging issue of data-intensive service mashup is how to find service from a great deal of candidate services while satisfying SLAs. In this paper, Service-Level Agreement (SLA) consists of two parts, which are SLA-Q and SLA-T. SLA-Q (SLA-T) indicates the end-to-end QoS (transactional) requirements. SLA-aware service mashup problem is known as NP-hard, which takes a significant amount of time to find optimal solutions. The service correlation also exists in data-intensive service mashup problem. In this paper, the service correlation includes the functional correlation and QoS correlation. For efficiently solving the data-intensive service mashup problem with service correlation, we propose an approach GTHFOA-DSMSC (Data-intensive Service Mashup with Service Correlation based on Game Theory and Hybrid Fireworks Optimization Algorithm) which evolves a set of solutions to the Pareto optimal front. The experimental tests demonstrate the effectiveness of the algorithm.

Povzetek: Razvit je nov algoritem za reševanje NP težkega problema prepletanja storitev v oblaku.

1 Introduction

With the rapid development of cloud computing, the number of data-intensive services has increased dramatically. Data-intensive services are defined as the services whose inputs are large data sets. Users can create various mashups which combine data-intensive services to form new value-added services [1-2]. Data-intensive service mashup has become an important type of application in the field of Big Data. The challenging problem of data-intensive service mashup is how to find service from a large number of candidate services while satisfying SLAs. A service-level agreement (SLA) is defined upon a mashup as its end-to-end requirements [3]. A large number of research works such as [4-6], solve the SLA-aware service selection problem by leveraging linear programming. However, integer linear programming is only suitable for small size problems and suffers from high computational costs.

In order to solve the problem of high computational costs, numerous approaches have been studied. Alrifai et al. [7] proposed a hybrid solution that combines global optimization with local selection. The approach first adopted mixed integer programming to get the optimal decomposition of end-to-end QoS constraints, and then performed efficient local selection to get the best services which satisfy the local QoS constraints. Compared with the integer linear programming, the hybrid solution performs better in the time efficiency while achieving close-to-optimal results. The skyline technique has been used to reduce the number of candidate services. Instead of considering all services of each service class, a large number of efficient algorithms speed up the service selection process and discover the optimal composite service from a reduced solution space [8-11].

A large number of research works leverage heuristic algorithms to solve the service selection problem. In [12], an approach based on genetic algorithm was presented where the solution is encoded as a chromosome. However, there are some shortcomings in classical genetic algorithm, such as premature phenomena. To overcome the flaws of genetic algorithm, a number of improved genetic algorithms have been used to find the sub-optimal solution [13-18]. In [18], an improved genetic algorithm was proposed for SLA-aware service selection problem which results with higher fitness values. The simulated annealing and harmony search were used as mutation operator in the improved algorithm. Compared with genetic algorithm, other heuristic algorithms can also achieve better optimality. Wang et al.[19] combined an approximation approach with artificial bee colony to solve the QoS-aware service selection problem. In [20], an improved immune optimization algorithm based on PSO was presented for QoS-aware service selection with end-to-end QoS constraints. In [21], an effective service selection approach with global QoS constraints based on particle swarm optimization algorithm was proposed.

To our best knowledge, there are only a small number of works which concern SLA from transactional risk. In [22], a survey of SLA assurance was conducted in the cloud. Haddad et al. [23] proposed a series of construction and processing rules to obtain a transactional mashup. However, the approach cannot gain global optimality and cannot support SLA-aware service selection. Wu et al.[24] combined the transaction properties into QoS-aware service selection and presented an approach based on ant colony algorithm.Compared with [23], the approach shows better performance in efficiency.

Many approaches regard the service mashup problem as a single objective optimization problem. To obtain multiple Pareto-optimal solutions, a few researches adopt multi-objective genetic algorithms to solve the problem [25-26]. In [25], a multi-objective optimization framework for SLA-aware service selection was presented. By leveraging multi-objective genetic algorithm, the framework can produce a set of Pareto-optimal solutions effectively. In [26], the authors improved a multi-objective optimization algorithm which applies background knowledge to find QoS-optimized service selection. In [27], an effective multi-objective approach was presented to solve QoS-aware service selection with conflicting objectives and diverse constraints on quality matrices. In [28], a hybrid multi-objective particle swarm optimization algorithm was proposed for SLA-aware service selection problem. Fireworks optimization algorithm (FOA) [29-30] is a relatively new heuristic method inspired by the phenomenon of fireworks explosion. As far as we know, there is still no research on the application of fireworks optimization algorithm in multi-objective service selection problem.

The contributions which distinguish our work from the above researches can be summarized as follows: 1. the problem of SLA-aware data-intensive service mashup with service correlation is formulated; 2. the SLA is divided into two aspects which are SLA-Q and SLA-T; 3. the service correlation is composed of two parts which are functional correlation and QoS correlation; 4. an approach GTHFOA-DSMSC (Data-intensive Service Mashup with Service Correlation based on Game Theory and Hybrid Fireworks Optimization Algorithm) which evolves a set of solutions to the Pareto optimal front is presented.

The remainder of this paper is organized as follows: Section 2 briefly presents the framework for data-intensive service mashup. Section 3 introduces the multi-objective optimization model, while Section 4 presents an approach based on Game Theory and Hybrid Fireworks Optimization Algorithm. The analysis of simulated results is done in Section 5. Finally, Section 6 concludes our work.

2 Framework for data-intensive service mashup

Figure 1 demonstrates the data-intensive service mashup process in the cloud. The framework is composed of three main components: 1) Planner component; 2) Generator component; 3) Execution Engine.

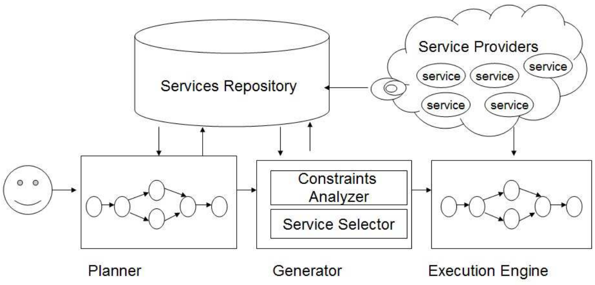


Figure 1: Framework for data-intensive service mashup.

2.1 Planner component

The component receives the request to generate an abstract mashup. An abstract mashup specifies the execution sequences among the activities. Each activity is an abstract service(AS) which corresponds to a candidate services set.

2.2 Generator component

Given a certain abstract mashup, in addition to a set of concrete services that can implement the activities from the cloud provider, the generator component can decide on what concrete services to include in the mashup. This component consists of two sub-components:1) Constraints Analyzer; 2) Service Selector.

Constraints Analyzer component receives the end-to-end constraints and generates a score based on constraints. Service Selector component selects services to construct the concrete mashup. The concrete mashup is composed of a set of concrete services from the candidate services. For an abstract mashup with n abstract services and l candidate services in each abstract service, there are l^n concrete mashups to be evaluated. A concrete service is represented by a tuple denoted as $< N, I, O, T, Q >$. Following is the detail description of the tuple.

- N is the name of a service.
- I=$\{I_1$,...,$I_n\}$ is a set of inputs which are required when performing the service.

- O={O_1,...,O_n} is a set of outputs that will be acquired after completing the service.
- $Transactional\ properties(T)$ guarantee the failure atomicity during execution.
- $QoS(Q)$ presents the quality of service which can be used to assess a service.

2.3 Execution engine

The concrete mashup is sent to execution engine to be executed. The execution engine is responsible for coordinating the execution of the components in the most effective way.

3 Multi-objective optimization model

3.1 QoS attributes

QoS attributes are introduced to describe non-functional properties of data-intensive services. They are used to differentiate the services providing the same functionality during the service selection process. In this paper, four most popular QoS attributes are considered: execution cost(C), response time(T), availability(A) and reliability(R).

- Execution cost : the cost that a service requester has to pay for the service invocation.
- Response time : the time interval between when the service is invoked and when the result is obtained.
- Availability : the probability that the service is accessible.
- Reliability : the probability that a request is correctly responded.

3.2 Normalization of attribute value

Due to the diverse measurement metrics of QoS attributes, attribute values should be normalized. For positive attributes, higher value indicates better quality (e.g. availability and reliability), which are normalized as equation (1). For negative attributes, lower value indicates better quality (e.g. cost and response time), which are normalized as equation (2).Q_i^{max} andQ_i^{min} are the maximal and minimal attribute values among all services, respectively. $Q_i^{'}$ refers to the attribute value of Q_i after normalization.

$$Q_i^{'} = \begin{cases} \frac{Q_i - Q_i^{min}}{Q_i^{max} - Q_i^{min}} & Q_i^{max} - Q_i^{min} > 0 \\ 1 & Q_i^{max} - Q_i^{min} = 0 \end{cases} \tag{1}$$

$$Q_i^{'} = \begin{cases} \frac{Q_i^{max} - Q_i}{Q_i^{max} - Q_i^{min}} & Q_i^{max} - Q_i^{min} > 0 \\ 1 & Q_i^{max} - Q_i^{min} = 0 \end{cases} \tag{2}$$

3.3 QoS computation of mashup

A mashup can be constructed from several services in different structures. There are four basic structures: sequential, parallel, branch, and loop structures. Figure 2 shows the four structures.

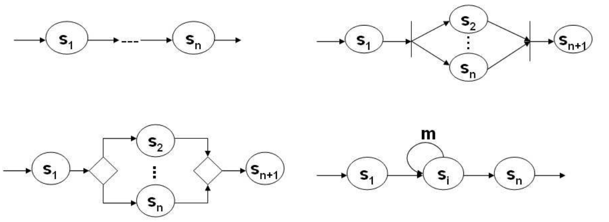


Figure 2: Four basic structures.

This paper computes the QoS of mashup according to the equations in table 1.For an additive property (e.g. cost and response time), we should compute the value through add operation. For a multiplicative property (e.g. availability and reliability), the value should be determined by multiply operation.

	Sequential	Parallel	branch	Loop
C	$\sum_{i=1}^{n} c_i$	$\sum_{i=1}^{n} c_i$	$\sum_{i=1}^{n} p_i c_i$	$k*c$
T	$\sum_{i=1}^{n} t_i$	$\max[t_i]$	$\sum_{i=1}^{n} p_i t_i$	$k*t$
A	$\prod_{i=1}^{n} a_i$	$\prod_{i=1}^{n} a_i$	$\prod_{i=1}^{n} p_i a_i$	a^k
R	$\prod_{i=1}^{n} r_i$	$\prod_{i=1}^{n} r_i$	$\prod_{i=1}^{n} p_i r_i$	r^k

Table 1: QoS aggregation functions.

3.4 Transactional properties

The transactional properties of services that we consider in this paper are pivot, compensatable, retriable and their combination. To obtain a transactional mashup, the rules are proposed in [23]. The transactional property of mashup $Tp(M) \in Tpset$, $Tpset$={p,c,r,cr}.

- $TP(M) = p$. Once all services of mashup execute successfully, the effects cannot be undone.
- $TP(M) = c$. Mashup is able to recover its effects even if it is executed successfully.
- $TP(M) = r$. Mashup will execute repeatedly until it is successful.
- $TP(M) = cr$. Mashup is both compensatable and retriable.

Table 2 and 3 represent the rules, where row heading indicates the transactional property of the first service, column heading indicates the transactional property of the second service. The value in each table cell represents the transactional property of mashup. $''-''$ denotes the mashup does not satisfy atomic consistency.

	p	c	r	cr
p	-	-	p	p
c	p	c	p	c
r	-	-	r	r
cr	p	c	r	cr

Table 2: Transactional rules for sequential construct.

	p	c	r	cr
p	-	-	-	p
c	-	c	-	c
r	-	-	r	r
cr	p	c	r	cr

Table 3: Transactional rules for parallel construct.

3.5 Service level agreement (SLA)

We assume that the end user has more SLA requirements with regard to the QoS values and transactional properties of the requested mashup. For a given abstract mashup, we consider a selection as a feasible selection, if it contains exactly one service for each service class and satisfies the end-to-end QoS (transactional) requirements.

3.6 Service correlation

- Functional Correlation: some concrete services are functional dependent on each other. The functional correlation $FC(S_{im}, S_{jn})$ indicates if the abstract service S_i selects the m^{th} concrete service, then the n^{th} concrete service should be selected for abstract service S_j .
- QoS Correlation: the QoS values delivered by a service in a mashup may vary according to the other services selected. $QC_a(S_{im}, S_{jn})$ indicates a QoS correlation between S_{im} and S_{jn}, regarding a QoS attribute a.

3.7 Multi-objective optimization model

In this paper, we take T, C and R as three objective functions for the sake of simplicity. A model of multi-objective service mashup can be formalized as follows:

Minimum(T(M),C(M),-R(M))
$s.t.$
$(1) A(M) > A_0$
$(2) T(M) < T_0$
$(3) C(M) < C_0$
$(4) Tp(M) \in$ {p,c,r,cr}
(5)Correlation constraints are satisfied.

Where T(M),C(M),A(M) and R(M) represent QoS attributes of mashup. A_0 ,T_0 and C_0 are the constraints to availability, time and cost respectively. The goal is to make the objective functions to be minimized simultaneously.

4 Service mashup based on game theory and hybrid fireworks optimization algorithm

4.1 Game theory

During the game, a problem is divided into several simpler problems according to the number of players. Each player seeks the best strategy in order to improve its objective criterion. As soon as no players can improve its objective value by adjusting its own best strategy, the goal is reached.

4.2 Fireworks optimization algorithm

Fireworks Optimization Algorithm (FOA), a novel heuristic algorithm, is implemented by simulating the fireworks explosion. At each iteration, the algorithm selects some quality locations as fireworks, which generate many sparks to search the local area. The algorithm continues until optimal location is found, or the termination condition is satisfied. The number of sparks and the amplitude generated by each firework are respectively defined such that:

$$s_i = m \times \frac{f_{max} - f(x_i) + \alpha}{\sum_{i=1}^{n}(f_{max} - f(x_i)) + \alpha} \tag{3}$$

$$A_i = A \times \frac{f(x_i) - f_{min} + \alpha}{\sum_{i=1}^{n}(f(x_i) - f_{min}) + \alpha} \tag{4}$$

Where m and A are control parameters, n is the size of the population, f_{max} and f_{min} indicate the maximum and minimum object values among the n fireworks respectively, and α is a small constant. To avoid overwhelming effects of splendid fireworks, bounds are defined for s_i as follows:

$$s_i^{'} = \begin{cases} round(\beta \times m) & \text{if } s_i < \beta m \\ round(\delta \times m) & \text{if } s_i > \delta m \\ round(s_i) & otherwise \end{cases} \tag{5}$$

Where β and δ are constant parameters. The location of each spark x_j generated by x_i can be calculated by equation (6):

$$x_j^d = x_i^d + A_i \times rand(-1, 1) \tag{6}$$

If the obtained location falls out of the search area, we should map it to the search area as follows:

$$x_j^d = x_{min}^d + x_j^d mod(x_{max}^d - x_{min}^d) \tag{7}$$

In the algorithm, the current best location is always selected as a firework of the next explosion iteration. Afterwards, n-1 locations are selected according to their distances to other locations.

4.3 Fitness assignment

In this paper, we adopt the notion of domination value to compute the fitness. A solution x_i is said to dominate a solution x_j in three cases:

- The solution x_i is feasible, and x_j violates some constraints.
- Both x_i and x_j are feasible, and x_i dominates x_j in terms of their object values.
- Both x_i and x_j violate constraints, and x_i dominates x_j in terms of their SLA violations.

In the GTHFOA-DSMSC, the fitness value of x_i is determined:

$$f(x_i) = \sum_{x_j > x_i} s(x_j) + \frac{1}{d(x_i)} \tag{8}$$

Where $d(x_i)$ is the distance from x_i to its nearest solution. The strength value $s(x_j)$ is computed according to the number of other solutions it dominates.

$$s(x_j) = |\{x_k \in P \cup NP | x_j > x_k\}| \tag{9}$$

4.4 Coding strategy

As illustrated in figure 3, the firework is encoded as an integer array of n elements: AS_1,AS_2, ..., AS_n and the value of AS_i ranges from 1 to m . Where n is the number of activities in the abstract mashup and m is the number of candidate services for each of the activity.

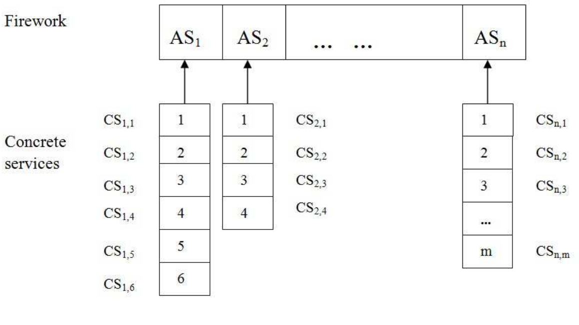


Figure 3: Coding strategy.

4.5 Crossover and mutation operators

The crossover and mutation operators are incorporated into the FOA to improve the performance. If crossover occurs at certain position, solutions in pairs swap their values at that position and the resulting solutions are used as offspring. The mutation operator is done by randomly selecting a position in a parent solution and randomly choosing a new concrete service to replace the one at that position.

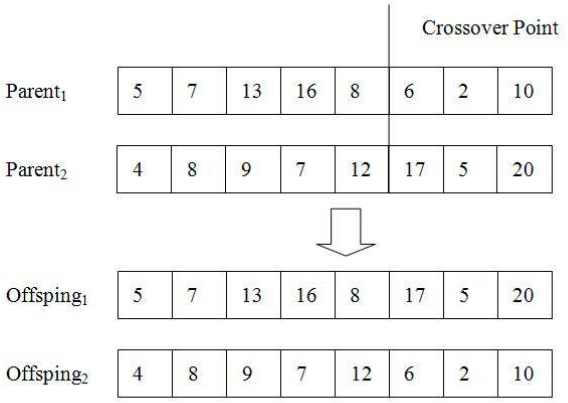


Figure 4: Crossover operator.

4.6 Non-dominated archive controller

The function of the archive controller is to determine whether a solution should be inserted into the external archive. The size of external archive may increase quickly, and thus it is required to limit the size of archive. If the external archive is empty, then the current solution is added into the archive. If the new solution is dominated by an individual within the archive, then the solution is discarded. If none of the individuals included in the archive dominates the new solution, then the solution is inserted in the archive. If there are individuals in the archive which are dominated by the new solution, then the individuals are removed from the external archive. Lastly, when the external archive reaches the size limit, the approach proposed by [31] is invoked.

4.7 Hypervolume (HV)

The HV of a solution set w signifies the hypervolume in the objective space that is dominated by w . In Figure 5, the solution w_5 is dominated by the solution set $\{w_1$, w_2, w_3 ,$w_4\}$. In this paper, we combine the non-dominated fronts of several algorithms into a maximum front w_{max} . The HV ratio of w is calculated by equation (10):

$$HVRatio = \frac{HV(w)}{HV(w_{max})} \tag{10}$$

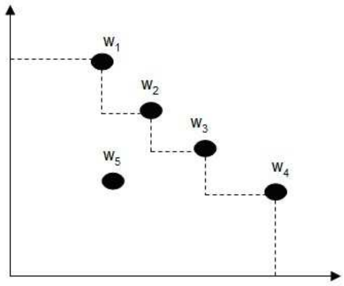


Figure 5: The hypervolume of the solution set $\{w_1, w_2, w_3, w_4\}$.

4.8 Algorithm design of GTHFOA-DSMSC

The algorithm description is as follows:

step1 Randomly generate a population P of n feasible solutions; create the empty external archive, and select non-dominated solutions from P to update the archive. Each player optimizes only his object.

step2 If the termination criteria is satisfied, goto Step5; else continue;

step3 Compute s_i and A_i for each individual x_i in P according to equations (3) (4)and (5); generate sparks of x_i according to equations (6) and (7); compute fitness for all sparks according to equations (8) and (9); select n solutions from the fireworks and sparks, use the crossover and mutation operations on the n solutions. Each player obtains his best solution respectively. Update the archive based on the new solutions.

step4 Update P by including the best solution and other $n-1$ ones selected based on their distance to other locations, goto Step2.

step5 Get the solutions and stop the algorithm.

5 Experiment and analysis

In this section, we conduct experiments to assess the performance of the proposed algorithm on a PC with Pentium 2.0GHz processor, 4.0GB of RAM and Windows7. In the experiment, the QoS attributes of the services are randomly generated expect for the cost, which is partially anti-correlated to the other QoS attributes. The algorithms optimize three objectives and have to meet constraints. The transactional property of each service is selected from $\{p,c,r,cr\}$ randomly. The percentage Θ indicates the strength of end-to-end QoS constraints. For positive QoS attributes, Θ is calculated by equation (11). For negative QoS attributes, Θ is calculated by equation (12).

$$\Theta = \frac{c_i - q_i^{min}}{q_i^{max} - q_i^{min}} \tag{11}$$

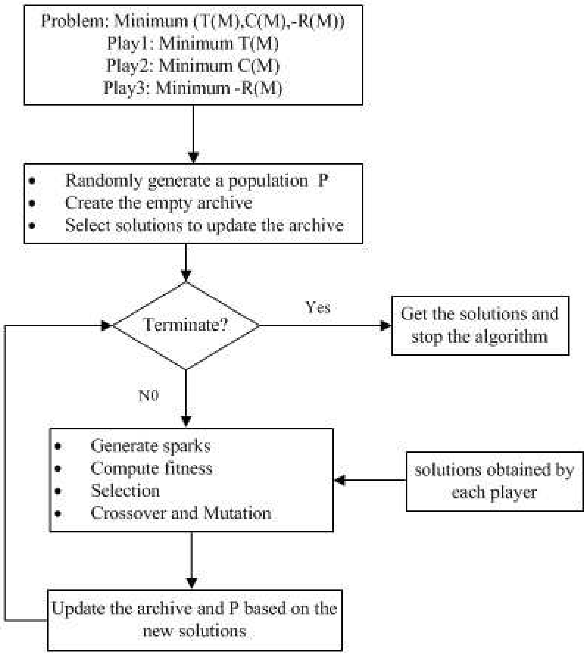

Problem: Minimum (T(M),C(M),-R(M))
Play1: Minimum T(M)
Play2: Minimum C(M)
Play3: Minimum -R(M)
Randomly generate a population P
Create the empty archive
Select solutions to update the archive
Terminate?
Yes
Get the solutions and stop the algorithm
N0
Generate sparks
Compute fitness
Selection
Crossover and Mutation
solutions obtained by each player
Update the archive and P based on the new solutions

Figure 6: The flowchart of GTHFOA-DSMSC.

$$\Theta = \frac{q_i^{max} - c_i}{q_i^{max} - q_i^{min}} \tag{12}$$

Where c_i is the i^{th} QoS constraint, q_i^{max} and q_i^{min} indicate the maximum and minimum aggregated values of the i^{th} QoS attribute of the mashup. For all the algorithms, the upper limit of the archive size is set to 20. The population size is set to 30 for the GTHFOA, 200 for the NSGA-II [32] and GDE [33]. We evaluate every test case 100 times and limit the runtime of each algorithm to 30s.

5.1 Performance vs problem size

We investigate the performance of GTHFOA with the problem size. In figure 7, the number of abstract services is fixed 8. The number of concrete service candidates increases from 40 to 120 with the step 20. In figure 8, the number of abstract services increases from 4 to 12 with the step 2. The number of concrete service candidates is fixed 100. As the figures illustrate, GTHFOA is able to achieve above 90% in average. NSGA-II and GDE have decreasing HV ratio with the increase of the problem size. Through the experiment, we conclude GTHFOA can efficiently escape from the local optima and guide the search towards the Pareto fronts. The exploration strategies used by the other heuristic algorithms are not sufficient for achieving a good approximation of the Pareto-front in the solution space.

5.2 Performance vs strength of constraints

We investigate the performance of GTHFOA with the strength of constraints. In figure9, the strength of con-

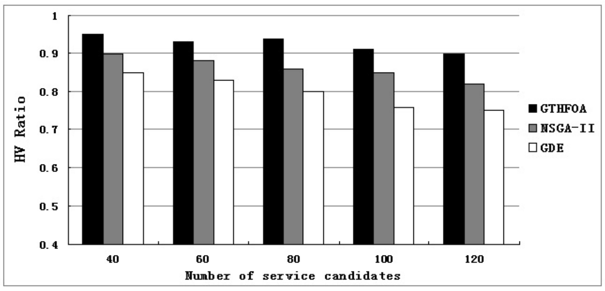


Figure 7: Performance vs service candidates.

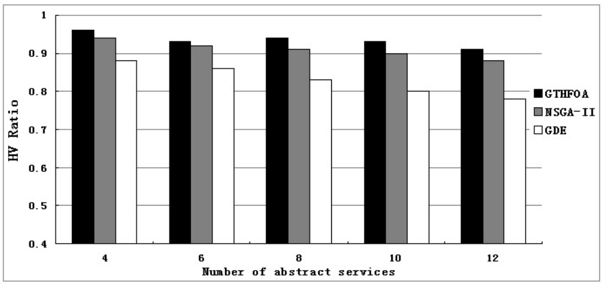


Figure 8: Performance vs abstract services.

straints increases from 0.2 to 0.6 with the step 0.1. With the increase of the strength of constrains, the existence probability for a feasible solution which satisfy all constraints declines. In figure9, GTHFOA has a constant HV ratio above 90%. If the strength of constraints is fixed 0.6, GTHFOA can gain HV ratio 91 %, while NSGA-II 85% and GDE 68%. As can be seen, GTHFOA shows better performance over NSGA-II and GDE. GTHFOA has a higher probability of reaching the Pareto-optimal front than other optimization algorithms.

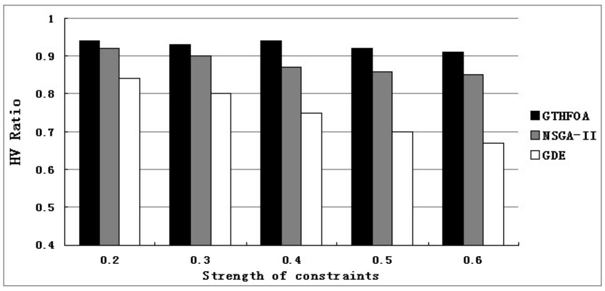


Figure 9: Performance vs strength of constraints.

5.3 Performance vs convergence

This part shows the convergence of GTHFOA. We fix the number of abstract services 8 and the number of service candidates 100. The strength of constraints is fixed 0.4. In figure 10, the number of iterations increases from 0 to 400 with the step 100. As can be seen, the convergence of GTHFOA is approximately 95%, while NSGA-II 90% and GDE 75%. The performance of GTHFOA overwhelms the other two algorithms. The GDE has the lowest performance among the algorithms, which indicates that its search capability is very limited. By comparison, we find GTHFOA converges fast and can converge to a higher value.

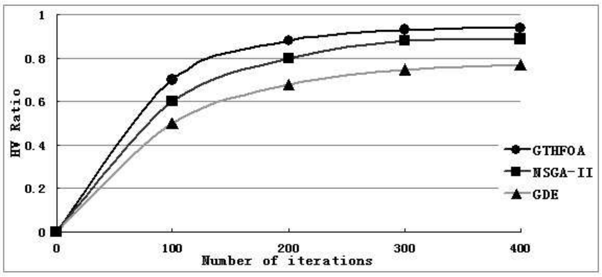


Figure 10: Performance vs convergence.

5.4 Performance vs service correlation

This part shows the performance of GTHFOA with service correlation. The approach GTHFOA-withoutQC does not consider the service correlation. We fix the number of abstract services 8 and the number of service candidates 100. The strength of constraints is fixed 0.4. We evaluate the ratio of the results of the two approaches. The ratio can be calculated by

$$ratio = \frac{GTHFOA - withoutQC}{GTHFOA} \quad (13)$$

In figure 11, the number of service correlation increases from 20 to100 with the step 20. As can be seen, GTHFOA-withoutQC does not consider the correlation, so it cannot reach the Pareto-optimal front. With the number of service correlation increases, the ratio declines. That is because the QoS deviation happened during the service selection.

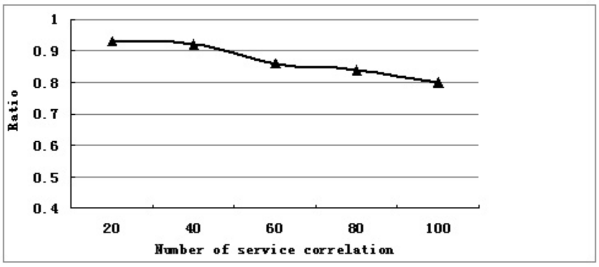


Figure 11: Performance vs service correlation.

6 Conclusion

This paper solves the data-intensive service mashup from QoS and transactional dimensions. The Service-Level

Agreement (SLA) consists of two parts, which are SLA-Q and SLA-T. In this paper, we consider the service correlation which includes the functional correlation and QoS correlation. In order to solve the service mashup problem efficiently, we propose an approach based on game theory and hybrid fireworks optimization algorithm which evolves a set of solutions to the Pareto optimal front. In our future work, we will further improve the performance of multi-objective evolution algorithm to solve the multi-objective service selection problem. The problem of runtime service process reconfiguration is also left for future research.

Acknowledgement

The research is supported by the Fundamental Research Funds for the Central Universities, the Natural Science Foundation of Shandong Jiaotong University (Z201342) and National Key Basic Research Program of China (2010CB328106).

References

[1] A. Bouguettaya, S. Nepal, W. Sherchan, X. Zhou, J. Wu, S. Chen, L. Liu, H. Wang and X. Liu(2010). End-to-End Service Support for Mashups,*IEEE Transactions on Service Computing*, 3(3), pp. 250-263.

[2] A. Ngu, M. Carlson, Q. Sheng and Hye-young Paik(2010). Semantic-Based Mashup of Composite Applications,*IEEE Transactions on Service Computing*, 3(1), pp. 2-15.

[3] F.H. Zulkernine and P. Martin(2011). An Adaptive and Intelligent SLA Negotiation System for Web Services,*IEEE Transactions on Service Computing*, 4(1), pp. 1939-1374.

[4] L. Zeng, B. Benatallah, A. Ngu, M. Dumas, J. Kalagnanam and H. Chang(2004). QoS-aware Middleware for Web Services Composition,*IEEE Transactions on Software Engineering*, 30(5), pp. 311-327.

[5] D. Ardagna and B. Pernici(2007). Adaptive Service Composition in Flexible Processes,*IEEE Transactions on Software Engineering*, 33(6), pp. 369-384.

[6] T.Yu, Y.Zhang and K.J.Lin(2007). Efficient Algorithms for Web service selection with End-to-End QoS Constraints,*ACM Transactions on the Web*, 1(1), article 6.

[7] M. Alrifai, T. Risse and W. Nejdl(2012). A Hybrid Approach for Efficient Web Service Composition with End-to-End QoS Constraints, *ACM Transactions on the Web* , 6(2), article 7.

[8] M. Alrifai, D. Skoutas and T. Risse(2010). Selecting Skyline Services for QoS-based Web Service Composition, *International Conference on World Wide Web*, Raleigh, North Carolina, pp. 11-20.

[9] K. Benouaret, D. Benslimane and A. Hadjali (2011). On the Use of Fuzzy Dominance for Computing Service Skyline Based on QoS,*IEEE International Conference on Web Services*, Washington,DC, pp. 540-547.

[10] K. Benouaret, D. Benslimane and A. Hadjali(2012). WS-Sky: An Efficient and Flexible Framework for QoS-Aware Web Service Selection, *IEEE International Conference on Services Computing*, Honolulu, HI, pp.146-153.

[11] Q. Yu and A. Bouguettaya(2013). Efficient Service Skyline Computation for Composite Service Selection, *IEEE Transactions on Knowledge and Data Engineering*, 25(4), pp. 776-789.

[12] G. Canfora, M. Penta, R.Esposito and M. Villani (2005). An Approach for QoS-aware Service Composition based on Genetic Algorithms,*7th annual conference on Genetic and evolutionary computation*, Washington,DC, pp.1069-1075.

[13] Y. Ma and C. Zhang(2008). Quick Convergence of Genetic Algorithm for QoS-driven Web Service Selection, *Computer Networks*, 52(5), pp. 1093-1104.

[14] Y. Syu, Y. FanJiang, J. Kuo and S. Ma(2012). A Genetic Algorithm with Prioritized Objective Functions for Service Composition, *International Conference on Advanced Information Networking and Applications Workshops*, Fukuoka, pp. 932-937.

[15] M. Chen and S. Ludwig(2012). Fuzzy-guided Genetic Algorithm applied to the Web Service Selection Problem,*IEEE World Congress on Computational Intelligence*, Brisbane, QLD, pp. 1-8.

[16] Y. Yu, H. Ma and M. Zhang(2013). An Adaptive Genetic Programming Approach to QoS-aware Web Services Composition, *IEEE World Congress on Evolutionary Computation*,Cancun,Mexico, pp. 1740-1747.

[17] F. Lecue and N. Mehandjiev(2011). Seeking Quality of Web Service Composition in a Semantic Dimension, *IEEE Transactions on Knowledge and Data Engineering*,23(6), pp. 942-959.

[18] A.E.Yilmaz and P. Karagoz (2014). Improved Genetic Algorithm based Approach for QoS Aware Web Service Composition, *IEEE International Conference on Web Services*,Anchorage, AK , pp. 463-470.

[19] X. Wang, Z. Wang and X. Xu(2013). An Improved Artificial Bee Colony Approach to QoS-Aware Service Selection, *IEEE International Conference on Web Services*, Santa Clara, CA, pp. 395-402.

[20] X. Zhao, B. Song, P. Huang, Z. Wen, J. Weng and Y. Fan(2012). An Improved Discrete Immune Optimization Algorithm based on PSO for QoS-driven Web

Service Composition, *Applied Soft Computing*, 12(8), pp. 2208-2216.

[21] G. Kang, J. Liu, M. Tang and Y. Xu (2012). An Effective Dynamic Web Service Selection Strategy with Global Optimal QoS Based on Particle Swarm Optimization Algorithm, *IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum*, Shanghai, China, pp.2280-2285.

[22] L. Sun, J. Singh and O. Hussain(2012).Service Level Agreement (SLA) Assurance for Cloud Services: A Survey from a Transactional Risk Perspective, *10th International Conference on Advances in Mobile Computing & Multimedia*, Bali, Indonesia, pp. 263-266.

[23] J.E. Haddad and G. Ramirez(2010). TQoS: Transactional and QoS-aware Selection Algorithm for Automatic Web Service Composition, *IEEE Transactions on Service Computing*, 3(1), pp. 73-85.

[24] Q. Wu and Q. Zhu(2013). Transactional and QoS-aware dynamic service composition based on ant colony optimization, *Future Generation Computer Systems*, 29(5), pp. 1112-1119.

[25] H.Wada, J. Suzuki, Y.Yamano and K. Oba(2012). E^3: A Multiobjective Optimization Framework for SLA-Aware Service Composition, *IEEE Transactions on Service Computing*, 5(3), pp. 358-372.

[26] F. Wagner, B. Klopper, F. Ishikawa and S. Honiden(2012). Towards Robust Service Compositions in the Context of Functionally Diverse Services,*International Conference on World Wide Web*, Lyon, France, pp. 969-978.

[27] A. Moustafa and M. Zhang(2013). Multi-Objective Service Composition Using Reinforcement Learning,*International Conference on Service-Oriented Computing*, Berlin, Germany, pp.298-312.

[28] H. Yin, C. Zhang, B. Zhang, R. Sun and T. Liu(2014). A Multi-Objective Discrete Particle Swarm Optimization Algorithm for SLA-Aware Service Composition Problem, *Acta Electronica Sinica*, 42(10), pp. 1983-1990.

[29] Y. Tan and Y. Zhu (2010). Fireworks Algorithm for Optimization, *International Conference on Swarm Intelligence*, Beijing, China, pp.355-364.

[30] Y. Pei, S. Zheng, Y. Tan and H. Takagi(2012). An Empirical Study on Influence of Approximation Approaches on Enhancing Fireworks Algorithm,*IEEE International Conference on Systems, Man, and Cybernetics*,Seoul, Korea, pp. 1322-1327.

[31] J.D.Knowles and D.W.Corne (2000). Approximating the Nondominated Front using the Pareto Archived Evolution Strategy,*Evolutionary Computation*, 8(2), pp. 149-172.

[32] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan (2002). A Fast and Elitist Multiobjective Genetic Algorithm:NSGA-II,*IEEE Transactions on Evolutionary Computation*, 6(2), 182-197.

[33] S. Kukkonen and J. Lampinen(2005). GDE3: The Third Evolution Step of Generalized Differential Evolution, *IEEE Congress on Evolutionary Computation*, Edinburgh, Scotland, pp. 443-450.

Efficient Multimedia Data Storage in Cloud Environment

Prachi Deshpande, S.C. Sharma and Sateesh K. Peddoju
Indian Institute of Technology Roorkee-247667, India
E-mail: psd17dpt,scs60fpt,drpskfec@iitr.ac.in

Ajith Abraham
Machine Intelligence Research Labs (MIR Labs), WA, USA
IT4Innovations-Center of Excellence, VSB - Technical University of Ostrava, Czech Republic
E-mail:ajith.abraham@ieee.org

Keywords: BMA, cloud, compression, data storage, multimedia, SOA, web

Received: June 24, 2014

With the rapid adoption of social media, people are more habituated to utilize the images and video for expressing themselves. Future communication will replace the conventional means of social interaction with the video or images. This, in turn, requires huge data storage and processing power. This paper reports a compression/decompression module for image and video sequences for the cloud computing environment. The reported mechanism acts as a submodule of IaaS layer of the cloud. The compression of the images is achieved using redundancy removal using block matching algorithm. The proposed module had been evaluated with three different video compression algorithms and variable macroblock size. The experimentations has been carried out on a cloud host environment by using VMWarework station platform. Apart from being simple in execution, the proposed module does not incur an additional monetary burden, hardware or manpower to achieve the desired compression of the image data. Experimental analysis has shown a considerable reduction in data storage requirement as well as the processing time.

Povzetek: Predstavljena je izvirna metoda za učinkovito shranjevanje multimedijskih vsebin v oblak.

1 Introduction

With the advent of the concept of cloud computing, the problem of data storage has been solved for a while. The distributed nature of the cloud allowed storage of huge data without any hassle. The cloud computing is an amalgamation of different technologies such as networking infrastructure, service-oriented architecture (SOA), Web 2.0 and virtualization.

The advent of Cloud computing has brought the complexities of underlying networks due to a variety of applications and data formats to generality. A time and location independent services are available for the users in the cloud due to the generalization. Considering these facts, efforts had been initiated to implement an own private cloud for research and analysis purpose [1, 2].

From the last decade or more, the web has emerged as a powerful tool for social interaction. The possibility of use of multimedia content in the communication has revolutionized the social interaction. This had given rise to the intense growth in usage of multimedia enabled mobile appliances. In consequence, a huge volume of data is produced and processed on a daily basis in the forms of content (multimedia), structure (links) and usage (logs) [3].

The demand of information exchange, in particular, in the form of video/pictures had increased in many folds. Recently 'Microsoft' claimed that nearly 11 billion images had been hosted by its cloud storage service [4]. 'Facebook' has also announced a suppression of 220 billion of photos with an increase of 300 million images per day [5]. Categorization of such a huge chunk of data is very costly affair owing to the storage (hard disk) cost. The scenario may become worst when the overheads on the account of power consumption, cooling system and more significantly the skilled manpower recruitment were added up in the storage cost.

In future, although processed via cloud setup, processing of user generated image data may be abstracted by its volume. Hence, an efficient mechanism to compress the image data along with an overall reduction in the storage cost over the cloud is the need of the hour. The compression mechanism also must observe the quality of the reconstructed image without the requirement of an additional hardware setup. A proper compression technique may reduce the burden not only on cloud storage but also on the application devices for processing the image data.

In general, the images are stored in the joint photographic expert group (JPEG)/bitmap (BMP) file format. However, the individual compression achieved with these file formats may not be sufficient when a sequence of images (video) is to be stored. This is because the redundancy between the images is ignored while compressing them.

Fig. 1 depicts an image sequence indicating the interrelation with each other (the motion) and the huge redundancy between them.

A variety of approaches are reported in the literature to achieve the image data compression by redundancy removal approach. Some popular approaches are block matching (BMA), multiple to one prediction methodology, pseudo sequence, and content-based image retrieval (CBIR) [6–8]. In BMA, the redundancy is searched in the immediate next incoming frame. This method may be applied for similar or dissimilar images. Moreover, the search area is restricted to the incoming frame rather than the entire database. The multiple to one compression methodology is based on the hypothesis that for the similar images, the values of their low-frequency components are very close to that of their neighboring pixel in the spatial domain. A low-frequency template is created and used as a prediction for each image to compute its residue. The accuracy of this method is proportional to the similarity of the image data.

Pseudorandom-based image compression exploits the statistical randomness between the subsequent images. This method requires an addition mechanism to extract the statistical characteristics of the image. The similar images are arranged into a tree structure. The compression methodology needs to be applied to each branch. In the context of cloud, this method seems to be very complex as the cloud may have a variety of images rather than similar one. CBIR is used to search digital images in large databases using the image attributes like colors, shapes, textures, or any other information that may be derived from the image itself. In this way, to achieve comprehensive compression of image data in the cloud paradigm, this method needs to search the entire database over the cloud.

Considering the factors like the complex environment of the cloud, ease of searching process and speed of operation, a compression and decompression module for image and video data in the cloud computing environment have been proposed in this paper. Simple BMA is used in the proposed module to achieve the desired compression.

The novelty of the present work lies in the verification of the proposed module with different block size and some fast BMA with an aim to study the quality of the reconstructed images. The simulations are also carried out in a cloud environment to validate the proposed concept. The rest of the paper is organized as: Section 2 presents the state-of-the-art multimedia usage over cloud environment. Further, the proposed approach is described in Section 3, whereas the results are discussed in Section 4. The article is concluded with further scope in Section 5.

2 Related work

2.1 Image compression

It is believed that images and video will be the most preferred mode of communication in the next generation communication. This requires huge data storage. Hence, to cope with the data storage issue, the existing networks must be highly scalable. However, this option is very costly and complex to implement. Video/image compression provides an efficient way to eliminate the redundant information within images, which may lead to the less storage requirement and quick transmission of the images. An in-depth focus on the current and future technologies for video compression had been reported in [9]. JPEG, HEVC and H.264 are the prominent image compression standards available to minimize the superfluous information [10–12].

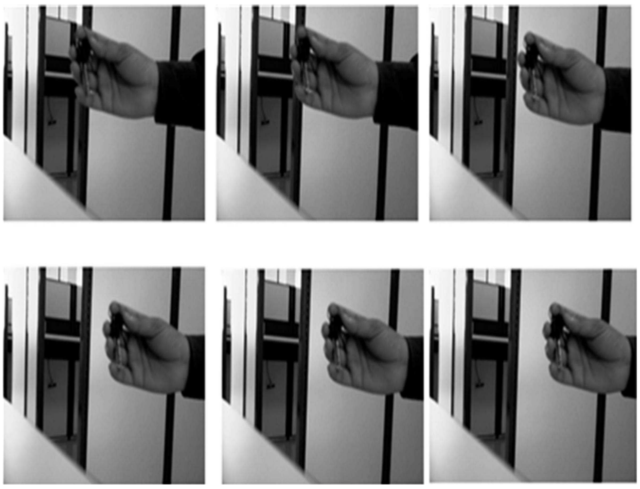


Figure 1: Video Stream.

The pseudo sequence compression method suffers from two main drawbacks. Firstly, it requires highly correlated images for compression and secondly it does not compress beyond the limits of the sequence definition [7]. Hence, the inter-image redundancies may be a problem when used in a cloud scenario. Local feature description approach sounds good regarding quality of reproduced image [13]. It decides the reconstruction of the image by searching the similarity pattern over the entire available data sets. In the cloud scenario, it may encounter a huge database search. Searching and retrieving of the image over the internet may also be carried out by using the description of the images such as outline, semantic contents, segmentation of moving targets, sub-band coding and multiple hypergraph ranking [14–18].

However, all these methods suffer from one or more drawbacks such as large search area/database, the speed of search, quality of reproduced image and the method of removing the redundancies. Intra prediction and transform is another popular approach for image compression [19, 20]. However, this approach suffered by the requirement of a highly correlated encoder and decoder. Hence, there is a need for a new mechanism to deal with the big data arises from future multimedia communication.

BMA is a tool used for the judgment of matching blocks in video frames or images for motion estimation. This finds a matching block from a reference frame i and in some other incoming (reference) frame *j*. BMA provides increased efficiency in interframe compression by identifying the temporal redundancy in the video sequence. This approach utilizes different cost functions to decide whether a given block in frame *j* matches the search block in frame *i* or else. Unlike its counterparts, BMA does not require a huge database search to reconstruct the image. It only searches the resemblance in the next immediate frame. The reconstruction quality of the image is also fair enough as it is based on the motion estimation in the successive images/frames.

Hence, BMA is preferred in the proposed compression module. Table 1 provides a brief summary of the various approaches to the image compression.

2.2 Cloud-based multimedia data storage

In the recent years, multimedia data processing over the cloud had become prominent due to the increasing use of image/video in social media platforms. Significant efforts were reported describing the progress of multimedia data processing over the cloud-based environment. A cloud-based multimedia platform for image and video processing is discussed by Gadea et al. [21]. However, this approach does not emphasize the reduction of storage requirement of multimedia data and its dependence upon the capacity of cloud environment for data storage. An SOA module for medical image processing using cloud computing was proposed by Chiang et al. [22]. However, this effort was also dependent on the ability of the cloud to store and process the multimedia data and never considered the cost incurred towards the data storage.

Zhu et al. [23] had discussed the multimedia cloud computing in detail. They concentrated on the storage and processing of multimedia data and proposed a multimedia cloud. This approach was stuck up by the need of a separate arrangement for storing and processing the multimedia data in the cloud environment. In a server-based multimedia computing, a set of server deals with multimedia computing and the clients are controlled by the servers [24]. However, this method was suffered from the deployment cost. In peer to peer (P2P) multimedia computing, computing task was carried out in a piecewise manner between the peers [25]. This had improved the scalability at the cost the quality of service (QoS). Content delivery network (CDN) reduced the communication overhead. However, this approach was stuck up by the scalability challenge due to limited server capabilities [26, 27].A data middleware is proposed in [28] to overcome the I/O bottleneck issues of big data legacy applications. However, this approach was developed to support the requirements of the document stores.

A dedicated media cloud concept is proposed in [29], wherein the cloud is only meant to process the multimedia data. A scale invariant feature transform was reported in [30] for image compression over the cloud. This approach was based on searching the similarity from all the images stored in the cloud, as the search accuracy was entirely dependent on the number of images available for the search. Recently a framework for multimedia data processing over the heterogeneous network was proposed [33]. All these methodologies never considered the huge storage memory requirement and allied overheads for the on-demand video/image access by the users.

3 Proposed approach

Cloud computing is the best alternative to cope with the huge data storage requirement. The cloud computing may solve the data storage requirements at the cost of a huge monetary and infrastructure overhead. As the cloud services were based on the 'pay-per-use' concept, end users have to pay these overheads. The data storage cost may be minimized by compressing the image data before storage and decompressing it as and when required. This approach may minimize the monetary overheads by a great deal.

A cloud-based compression and decompression mechanism (CDM) will solve this purpose. Hence, in this paper, a CDM has been reported to cater the need of video data storage over the cloud infrastructure. It utilizes the interframe coding to compress the incoming images with minimal burden on cloud resources. In general, a cloud had structured with three main layers of operation such as infrastructure as a service (*IaaS*); Platform as a service (*PaaS*) and Software as a service (*SaaS*). Each of them is meant for providing some specialized services to the cloud users. The *IaaS* layer deals with the storage of data on cloud mechanism. Hence, we suggest a CDM module at each virtual machine (VM) as a software abstraction in IaaS layer of the cloud to store the video data in the compressed form. Table 2 provides a brief comparison of the proposed method with existing approaches for processing multimedia data over the cloud.

In this analysis, an H.264 based interframe predictive coding is used for eliminating the temporal and spatial redundancy in video sequences for effective compression. In typical prophetic coding approach, the distinction between the present frame and the anticipated frame had coded and transmitted. The anticipated frame had been dependent on the previous frame. The transmission bit rate is directly proportional to the prediction of the video frame.

This approach was accurate for a still picture. However, for video sequences with a large motion, a better prediction is possible only with the proper information of the moving objects. Motion compensation (MC) is the phenomenon, which utilizes information of the displacement of an object in consecutive frames for reconstruction of a frame.

The proposed CDM module consists of an encoder and decoder as shown in Fig. 2. At the encoder side the first frame of the image/video sequence is initially considered as the reference frame. The next frame is considered as the incoming frame. The individual image is divided into macroblocks of desired dimensions (i.e.$16 \times 16, 8 \times 8, 4 \times 4$). In

Contribution	**Approach**	**Methodology**	**Disadvantage**	**In cloud scenario**
Zou et al. [7]	Pseudo sequence compression	minimum spanning tree (MST)	Exhaustive search is required for finding the base feature	QoS degradation as the search area is large
Rajurkar and Joshi [8]	CBIR	Attribute based search	Loss of information is very high due to compression	Searching of a specific attribute may be a burden on the cloud
Wallace [10]	JPEG	converts each frame of the video source from the spatial (2D) domain into the frequency domain	Images with strong contrasts donot compress well	Cloud have images with a variety of contrast in its storage
Wiegand et al. [12]	H.264	Block based similarity search	Computationally complex	Interframe similarity search from a sequence
Zhou et al. [13]	STFT	Local feature description	Needs a huge database for correct reconstruction of images	Entire cloud will be the search area.

Table 1: Image compression techniques.

Contribution	Data Processing	Method	Algorithm	Search area
Shi et al. [17]	By separate encoder and decoder in the cloud	Use of internal/ external correlation between target image and images in the cloud	Sub band coding	Entire database over cloud
W. Zhu et al. [23]	Dedicated couldlet servers placed at the edge of a cloud to provide media services	Load balancer and cloud proxy is used for processing	Feature extraction and image matching	Entire database over cloud
Hui et al. [29]	By a dedicated cloud mechanism	Searching by comparing the features of incoming video with database	Attribute based search	Entire database over cloud
Yue et al. [30]	By separate encoder and decoder in the cloud	Searching similarity in the large scale image database available on the cloud	Scale invariant feature transform	Entire database over cloud
Kesavan et at. [31]	By a dedicated private cloud mechanism	The private cloud store, search, multimedia services to user multimedia services to user	—	Entire database over cloud
Hussein and Badr [32]	By using native cloud capacities	Lossy and lossless compression techniques	Huffman encoding	Entire database over cloud
Proposed Approach	Local encoder/decoder as a software abstraction	Use of information in the video/image to be stored	Block matching	Limited to the video/ image to be stored.

Table 2: Comparison of the proposed approach.

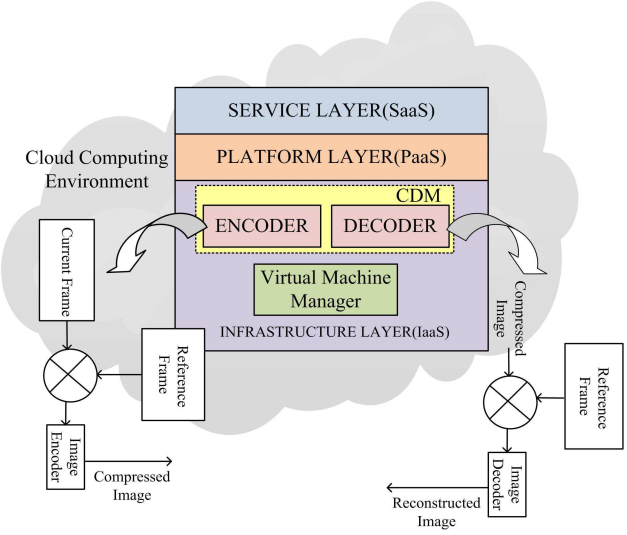


Figure 2: Conceptual diagram of the proposed CDM.

this study, the block-matching algorithm (BMA) is used as an MC tool for interframe predictive coding. BMA works on a block-by-block basis. It finds a suitable match for each block in the current frame from the reference frame.

The comparison eliminates the similar part from the particular block of the current frame and provides a pixel (*pel*) position from which the motion (difference) appears. This pixel position is called as a motion vector (MV) corresponding a particular block of the image. This is a two-bit parameter (x, y) which is searched in either direction of a particular pixel with a search range parameter *d*. The search procedure is repeated for all pixel of a block, and a single motion vector is achieved. Hence, an image consists of MVs proportional to its block size. Thus, an effective compression may be achieved by transmitting only MVs rather than the entire frame along with the reference frame. Computationally lightweight search criteria's (cost functions) such as Peak signal to noise ratio (PSNR), Mean square error (MSE) and Mean absolute error (MAE) is used for the evaluation of a suitable match. Fig. 2 shows the architecture of the proposed concept. At the decoder side, the reference image and the MV information of the compressed image are required for the reconstruction of the image. The corresponding matching block is predicted based on the information of MVs and search range parameter *d*. Thus, the image is reconstructed by computing a block for each MV. The quality of the reconstructed image is validated by its PSNR value. The proposed mechanism may run as a software part with each VM in the cloud. Whenever, image/ video data is to be processed, this module will perform the compression or decompression as per the use requirement. This module will not incur any monitory or tactical burden on the existing mechanism, as no additional hardware is required. In this way, it may be the best alternative for reduction of data storage size and cost for the image/video data.

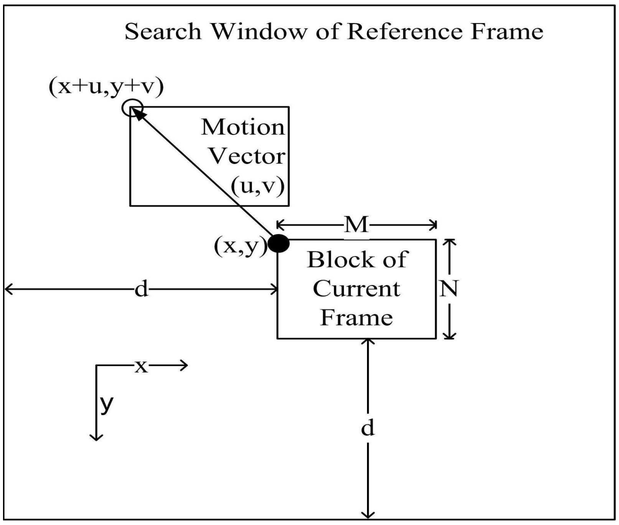


Figure 3: Concept of macroblock matching [34].

4 Result and discussions

Fig. 3 shows the basic concept of BMA. It consists of a macroblock of $M \times N$ size, which is to be searched within a search window of size *d* in all directions of the macro block. The typical block size is of the order of 16×16 pixels. The output of a cost function influences the macroblock matching with each other. A macroblock with the least cost is the most excellent equivalent to current block under search.

Eq. 1 to 3 provides the computationally efficient, cost functions such MAE; MSE; and PSNR. 'CPU TIME', an inbuild function of MatLab, is used to estimate the computational time for the proposed method. In block matching approach, the CDM divides the incoming frame into a matrix of macroblocks. This macroblock are compared with the equivalent block and its neighboring blocks in the reference (previous) frame.

The process forms a two-dimensional vector, which indicates the motion of a macroblock from one position to another in the reference frame. The motion in the incoming frame is anticipated based on the movement of all the macroblocks in a frame. To have an accurate macroblock match, the search range is confined to *d pels* in all four directions of the equivalent macroblock in the reference frame. *d* is the search range restriction and is proportional to the nature of motion.

$$MAE(dx, dy) = \frac{1}{MxN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |C(x,y) - R(x,y)| \tag{1}$$

$$MSE(dx, dy) = \frac{1}{MxN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |C(x,y) - R(x,y)|^2 \tag{2}$$

$$PSNR(dB) = 10log_{10}\left[\frac{255}{\sqrt{MSE}}\right] \tag{3}$$

Here, C(x,y) = the *pels* in the current macroblock and R(x,y) = the *pels* in the reference block. M and N= the size of the macroblock.

A motion compensated image is formed with the knowledge of motion vectors and macroblocks from the reference frame. The cost functions like PSNR and MSE are used to determine the quality of the prediction while CPU TIME was used to estimate the computational complexity. The full search (FS) algorithm involves large computations for block matching purpose since the cost functions were calculated at each possible position in the search range. Hence, BMA provides an excellent match with highest PSNR but at the cost of enormous computation. Fast BMAs like the 3-step search (TSS) and 4-step search (FSS) [35] provide a compatible PSNR with that of FS method with reduced computation. Hence, in the proposed work, the fast BMAs had been evaluated in the cloud environment along with FS method. The different macroblock size is also used to predict the performance of the proposed module. The performance of the cloud setup under these conditions is also evaluated. Fig. 4 shows the software abstraction used for the experimentation of the proposed method. Here the virtualization has been achieved by using the VMware workstation with Windows 7 as the host OS. Two VMs were also created on the host OS with Windows 7 as the guest OS. The performance was analyzed by executing the algorithms on the host OS.

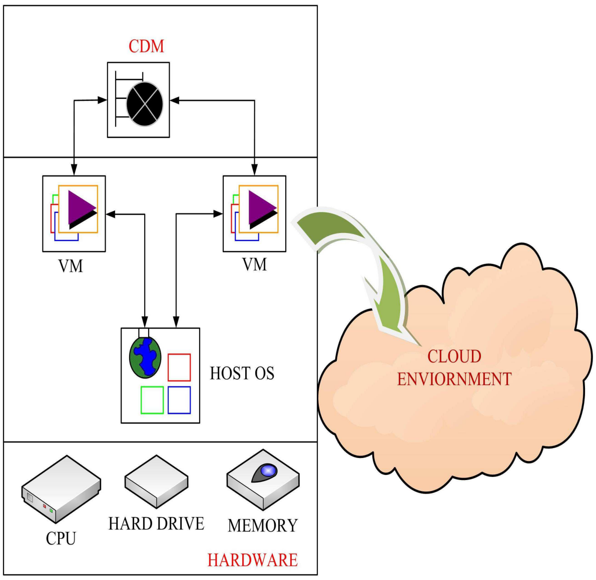


Figure 4: CDM deployment in cloud environment.

The performance of the FS, TSS and FSS was verified with a standard block size of 16×16 *pels*. Further, the evaluation was carried out with a block size of 8×8 and 4×4 *pels* to decide the accuracy of the image reconstruction. Two gray scale images were captured and stored in BMP format with a width of 320 *pels*, the height of 240 *pels* and a depth of 8 bits. Table 3 summarizes the results of BMA algorithms along with different block size. From Table 3, it is evidenced that lower the macroblock size higher is the detection accuracy. The results obtained from choosing 16×16 or 8×8 macroblock size were also compatible with that of 4×4 macroblock size. Hence depending on the need of the application a particular block size may be chosen. The PSNR, in all cases, is well above $30dB$. This is required for the proper reconstruction of the image/video at the receiver side [36]. The CPU time indicates the total time required by the CDM module to complete the encoding as well as decoding operation. By observing the 'CPU time' reading for all the algorithms with different block size, it may be inferred that the proposed module is competent enough to be used in the real-time applications also. The fair PSNR value and the 'CPU time' parameters indicate a competitive QoS regarding the reconstructed images.

The proposed method may be easily adapted for the continuous video streams. In this case, the reference image frame will be exchanged with a particular incoming frame, if its PSNR degrades below $30dB$ after reconstruction. The principle advantage of the CDM module is the reduction in the storage size (memory) for the video data. The storage size of a gray scale image is estimated by using Eq. 4.

$$FS(bytes) = \left[\frac{H_P \times V_P \times B_D}{8}\right] \qquad (4)$$

Where FS=File size; HP= Horizontal *pels*; VP= Vertical *pels*; BD= Bit depth.

In traditional data storage, a $M \times N$ size video frame requires MxN bits of memory. However, in the case of CDM approach, the video frame had been divided into n equal sized macroblocks, and each macroblock is represented by its motion vector(x,y). Each motion vector requires only two bits for its storage. Hence, a video frame with *n* macroblocks requires only $2n$ bits for its storage. Thus, the storage requirement has been reduced significantly. Table 4 summarizes the storage requirement with the proposed approach for an image/frame size of 320×240 *pels* with a 8-bit depth.

The approach proposed in [17] depends on the hypothesis that large-scale images in the cloud were always available for the similarity search. This methodology was tested on the commercially available database 'ZubBud'. The highest PSNR of $33.57dB$ for the reconstructed image was obtained by using this method. The approach proposed in [23], [29] and [31] never commented on the reconstruction quality of the image/video. The approach proposed in [30] used a separate encoder/decoder mechanism for the image data compression and decompression in the cloud. The methodology was verified by using INRIA Holiday data set. This methodology provided the highest PSNR of $29.65dB$ for the reconstructed image.

In the present analysis, an indigenously captured image sequence is used for analysis. The proposed methodology provides a highest PSNR of $39.50dB$ with a TSS. However, for FS and FSS methodology, the resulted PSNR value is higher than the highest value of methods in [17] and [30]. Moreover, the existing methodologies suffer from

Algorithm	**Block Size** $(M \times N)$	**CPU Time** (s)	**PSNR(dB)**	**MSE**	**MAE**
Full Search(FS)	16×16	18.04	33.75	10.90	3.30
	8×8	22.50	36.42	14.81	3.84
	4×4	39.03	37.20	18.48	4.29
Three Step Search(TSS)	16×16	6.67	33.50	33.81	5.38
	8×8	21.93	37.50	11.55	3.34
	4×4	80.40	39.50	7.28	2.69
Four Step Search(FSS)	16×16	8.85	33.41	29.64	5.44
	8×8	28.06	37.81	10.74	3.32
	4×4	105.70	38.90	8.37	2.90

Table 3: BMA analysis with different macroblock.

Approach	**Block size** (*pels*)	**No.of Macroblock**	Memory **Requirement(bytes)**
Traditional	–	–	76800
FS/TSS/FSS	16×16	300	75
	8×8	1200	300
	4×4	4800	600

Table 4: Analysis of storage requirements.

Contribution	**Cloud specific disadvantages**
Shi et al. [17]	Similarity search image for compression is carried out by searching database over the entire cloud.
Zhu et al. [23]	Separate Cloudlet servers are required. Similarity search for image compression is carried out by searching database over the entire Cloud.
Hui et al. [29]	Independent media Cloud is required to process multimedia data.
Yue et al. [30]	Separate encoder and decoder mechanism is required.

Table 5: Cloud specific limitations of the existing methodologies.

Contribution	**Data set used**	**PSNR** $(dB)^*$
Shi et al. [17]	ZubBud [37]	33.57
Zhu et al. [23]	–	–
Hui et al. [29]	–	–
Yue et al. [30]	INRIA Holiday [38]	29.65
Proposed Approach	Original image sequence	39.50

Table 6: QoS analysis of the proposed methodology.

$*$ Highest PSNR Value

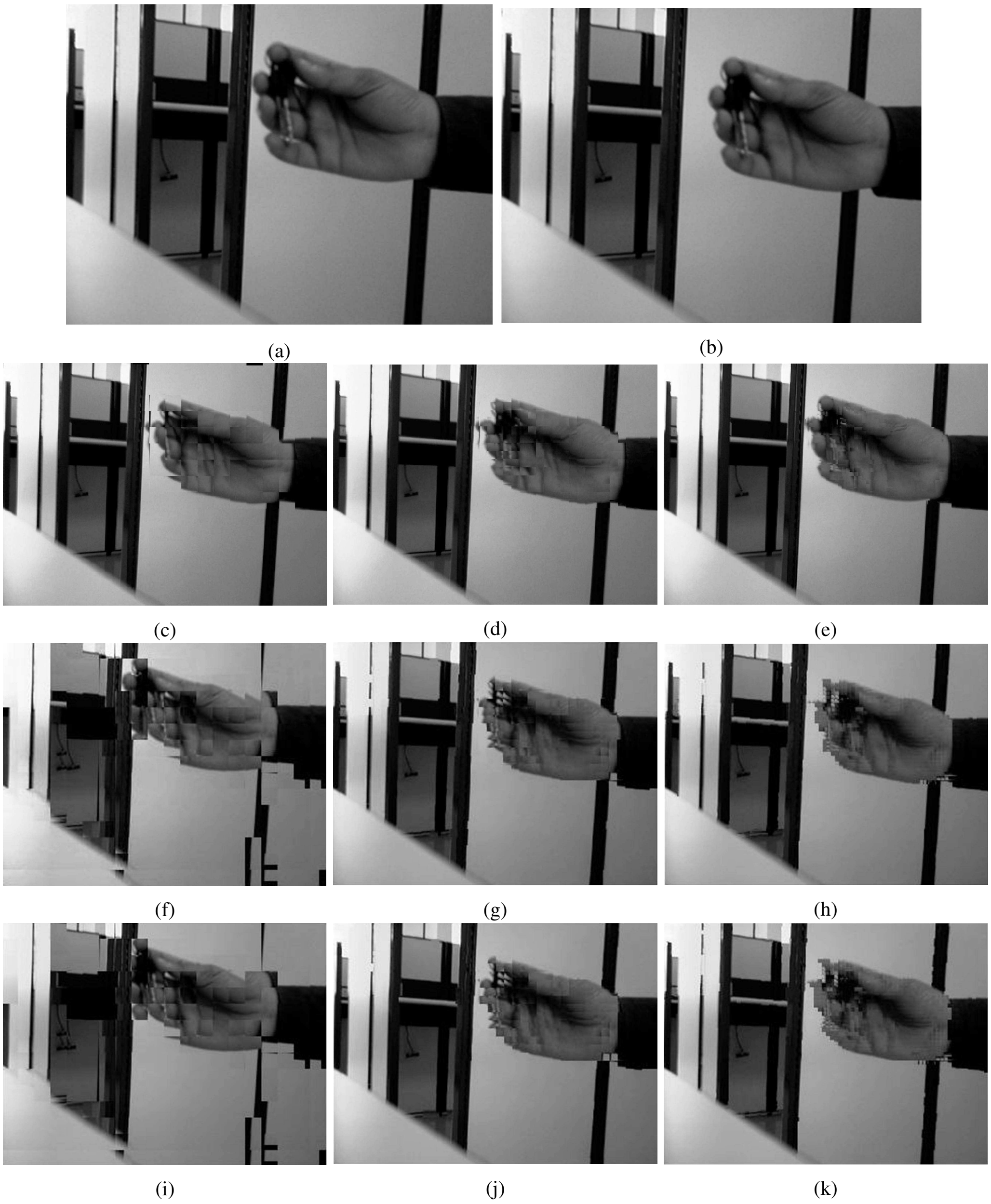


Figure 5: (a) Reference frame,(b) Current frame, (c) FS 16×16 block,(d) FS 8×8 block,(e) FS 4×4 block,(f) TSS 16×16 block, (g) TSS 8×8 block,(h) TSS 4×4 block,(i) FSS 16×16 block, (j) FSS 8×8 block and (k) FSS 4×4 block.

Algorithm	**Macroblock size**	**CPU Usage (%)**	**Memory Usage (%)**
FS	16 ×16 block	7	36
	8 ×8 block	4	36
	4 ×4 block	2	36
TSS	16 ×16 block	5	36
	8 ×8 block	3	35
	4 ×4 block	1	35
FSS	16 ×16 block	4	35
	8 ×8 block	2	36
	4 ×4 block	1	36

Table 7: CDM performance in cloud environment.

the drawback of the vast search area in the cloud or a separate and dedicated cloud arrangement for the multimedia data processing.

However, the proposed module only depends upon the information of the incoming and previous frame for its operation. Hence, the storage requirements and search operation are reduced to a great deal. This module does not require any separate hardware arrangement and will work as a software abstraction in the cloud infrastructure. This is the catch line advantage of the proposed approach over the other reported methods.

Fig. 5 depicts the motion compensated video frames, and it may be concluded that the fast BMA overrules the computational burden with FS BMA, without considerable compromise with the quality of the reconstructed image. The different block size is also an added advantage to achieve a fair quality of the reconstructed image.

Fig. 6 shows the cloud performance for the proposed CDM module with the help of 'resource manager'. The performance has been analyzed regarding CPU and memory usage. Table 7 gives the performance analysis of the proposed setup under cloud environment. It is evident from Table 7 that the proposed module never accessed the network resources to perform its operation.

The physical memory usage is also constant around 35-36% throughout the analysis for different BMA. The CPU usage is varied according to the macroblock size. In all cases, the CPU usage never increased beyond 10% of its maximum capacity. This indicates that the proposed module works without causing an additional burden on the available resources.

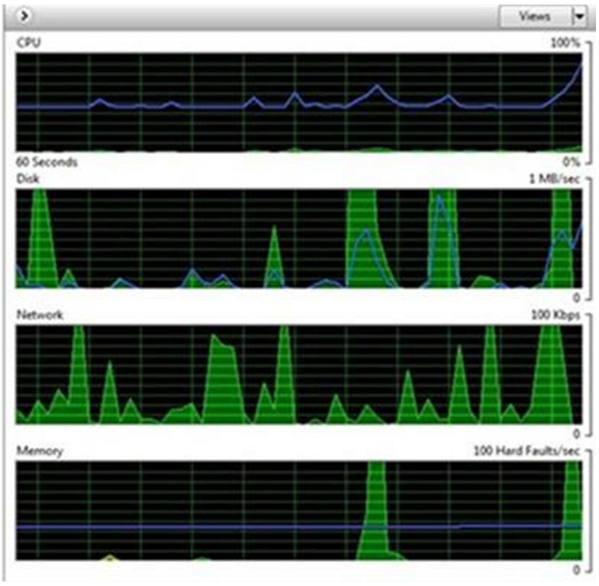


Figure 6: CPU and memory performance for the proposed module for FS (16 × 16 block).

5 Conclusions

The paper reports CDM, a compression and decompression module for cloud-based video data processing and storage. This module may be placed in each VM as a software abstraction at IaaS (SasIaas) layer of the cloud architecture. The novelty of the present work is the analysis of BMA with different block size in a cloud environment. With the proposed module, the requirement of a dedicated cloud for data processing and storage has been overruled. With the deployment of the proposed module, the multimedia data storage requirement is reduced with minimal overheads. The proposed module demonstrated a fair QoS regarding the reconstructed images. Hence, this approach may be the best candidate for the future generation information processing technology. In future, efforts may be initiated to design an adaptive CDM module to minimize the trade-off between a selection of a precise BMA algorithm along with a suitable block size for a specific application.

References

[1] P. Deshpande, S. Sharma and S. K. Peddoju, Deploying a private Cloud: Go through the errors first, *Proceedings of Conference on Advances in Communication and Control Systems*, Deharadun, India, Apr. 2013, pp. 638−641.

[2] P. Deshpande, S. Sharma, S. Peddoju, "Installation of a private cloud: A case study", *Advances in Intelligent Systems and Computing*, vol. 249, no. 2, pp. 635-648, 2014.

[3] Z. Zhang, Z. Zhengyou, R. Jain, "Socially connected multimedia across cultures", *Journal of Zhejiang University-SCIENCE C*, vol. 13, no. 12, pp. 875-880, 2012.

[4] J. Ong, Picture this: Chinese internet giant tencent's Qzone social network now hosts over 150 billion photos 2012 [Available:] http://thenextweb.com/asia/2012/08/09/ picture-this-chinese-internet-giant-tencents -qzone-social-network-now-hosts-over-150 -billion-photos

[5] K. Kniskern, How fast is SkyDrive growing? 2012 [Available:] http://www. liveside.net/2012/10/ 27/how-fast-is-skydrive-growing

[6] C. Yeung, O. Au, K. Tang, Z. Yu, E. Luo, Y. Wu, and S. Tu, "Compressing similar image sets using low frequency template", *Proceedings of IEEE International Conference Multimedia and Expo*, Barcelona, Spain, Jul. 2011, pp. 1−6.

[7] R. Zou, O. Au, G. Zhou, W. Dai, W. Hu, and P. Wan, "Personal photo album compression and management", *Proceedings of IEEE International Symposium on Circuits and Systems*, Beijing, Chaina, May 2013, pp. 1428−1431.

[8] A. Rajurkar and R. Joshi, "Content-based image retrieval in defense application by spatial similarity", *Defence Science Journal*, vol. 52, no. 3, pp. 285−291, Jul. 2002.

[9] L. Yu and J. Wang, "Review of the current and future technologies for video compression", *Journal of Zhejiang University-SCIENCE C*, vol. 11, no. 1, pp. 1−13, 2010.

[10] G. Wallace, "The JPEG still picture compression standard", *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii−xxxiv, Feb. 1992.

[11] JCT-VC, WD6: Working draft 6 of high-efficiency video coding. JCTVC-H1003, JCTVC Meeting, San Jose, Feb. 2012.

[12] T. Wiegand, J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard", *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560−576, Jul. 2003.

[13] W. Zhou, Y. Lu and H. Li, "Spatial coding for large scale partial-duplicate web image search", *Proceedings of the 18th ACM international conference on Multimedia*, Firenze, Italy, pp. 511−520, Oct. 2010.

[14] J. Smith and S. Chang, "Visualseek: A fully automated content based image query system", *Fourth ACM international conference on Multimedia*, Boston MA USA, 1996, pp. 87−98.

[15] M. Lew, N. Sebe, C. Djeraba and R. Jain, "Content-based multimedia information retrieval: State of the art and challenges", *ACM Transaction on Multimedia Computing, Communication Application*, vol. 2, no. 1, pp. 1−19, Feb. 2006.

[16] Y. Zhou, A. Shen, and J. Xu, "Non-interactive automatic video segmentation of moving targets", *Journal of Zhejiang University-SCIENCE C*, vol. 13, no. 10, pp. 736−749, 2012.

[17] Z. Shi, X. Sun and F. Wu, "Cloud-based image compression via subband-based reconstruction", *PCM-2012, Lecture Notes in Computer Science*, vol. 7674, pp. 661−673, 2012.

[18] Y. Han, J. Shao, F. Wu and B. Wei, "Multiple hypergraph ranking for video concept detection", *Journal of Zhejiang University-SCIENCE C*, vol. 11, no. 7, pp. 525−537, 2010.

[19] G. Sullivan and J. Ohm, "Recent developments in standardization of high efficiency video coding (HEVC)", SPIE, vol. 7798, pp.77980 V1–V7, 2010.

[20] R. Song, Y. Wang, Y. Han and Y. Li, "Statistically uniform intra-block refresh algorithm for very low delay video communication", *Journal of Zhejiang University- SCIENCE C*, vol. 14, no. 5, pp. 374-382, 2013.

[21] C. Gadea, B. Solomon, B. Ionescu and D. Ionescu, "A Collaborative Cloud-based multimedia sharing platform for social networking environments", *20th International Conference on Computer Communications and Networks*, Maui, HI, 2011, pp. 1−6,

[22] W. Chaing, H. Lin, T. Wu, and C. Chen, "Building a Cloud service for medical image processing based on service orient architecture", *4th International Conference on Biomedical Engineering and Informatics*, Shanghai, Oct. 2011, pp. 1459−1465.

[23] W. Zhu, C. Luo, J. Wang and S. Li, "Multimedia Cloud computing", *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 59−69, May 2011.

[24] K. Lee, D. Kim, J. Kim, D. Sul and S. Ahn, "Requirements and referential software architecture for home server based inter home multimedia collaboration services", *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 145−150, Feb. 2004.

[25] L. Zhao, J. Luo, and M. Zhang, "Gridmedia: a practical peer-to-peer based live video streaming system", *7th IEEE Workshop on Multimedia Signal Processing*, Shanghai, Nov. 2005, pp. 1−4.

[26] G. Fortino, C. Mastroianni and W. Russo, "Collaborative media streaming services based on CDNs", *Content Delivery Networks, LNEE*, vol. 9, no. 3, pp. 297−316, 2008.

[27] N. Carlsson and D. Eager, "Server selection in large-scale video-on-demand systems", *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 6, no. 1, pp. 1−26, Feb. 2010.

[28] K. Ma and A. Abraham, "Toward lightweight transparent data middleware in support of document stores", *Proceedings of the third World Congress on Information and Communication Technologies (WICT 2013)*, Hanoi, Vietnam, Dec. 2013, pp. 255−259.

[29] W. Hui, C. Lin, and Y. Yang, "Media Cloud: A new paradigm of multimedia computing", *KSII Transactions on Internet and Information Systems*, vol. 6, no. 4, pp. 1153−1170, Apr. 2012.

[30] H. Yue, X. Sun, J. Yang and F. Wu, "Cloud based image coding for mobile devices-towards thousands to one compression", *IEEE Transactions on Multimedia*, vol. 15, no. 4, pp. 845−857, Jan. 2013.

[31] S. Kesavan, J. Anand and J. Ayakumar, "Controlled multimedia cloud architecture and advantages", *Advanced Computing: An International Journal*, vol. 3, no. 2, pp. 29−40, 2012.

[32] S. Hussein and S. Badr, "Healthcare Cloud integration using distributed Cloud storage and hybrid image compression", *International Journal of Computer Applications*, vol. 80, no. 3, pp. 9−15, 2013.

[33] Y. Xu, C. Chow, M. Tham and H. Ishii, "An enhanced framework for providing multimedia broadcast/multicast service over heterogeneous networks", *Journal of Zhejiang University-SCIENCE C*, vol. 15, no. 1, pp. 63−80, Jan. 2014.

[34] A. Barjatya, "Block matching algorithms for motion estimation", *DIP 6620 spring 2004 Final Project Paper*, pp. 1−6, 2004. [Available at:] *http://profesores.fi-b.unam.mx/maixx/Biblioteca/ Librero_Telecom/BlockMatchingAlgorithmsForMotion Estimation.PDF*

[35] L. Po and W. Ma, "A novel four step search algorithm for fast block motion estimation", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 313−317, Jun. 1996.

[36] S. Welstead, "Fractal and wavelet image compression techniques", *SPIE Publication*, pp. 155−156, Dec. 1999, $ISBN$: 9780819435033.

[37] Eth-Zurich, Zurich building image database. [Available at]: *http://www.vision.ee.ethz.ch/showroom/ zubud/index.en.html*.

[38] H. Jegou and M. Douze, "INRIA Holiday Dataset2008". [Available at]: *http://lear. inrialpes. fr/people/jegou/data.php*.

Asynchronous Adaptive Delay Tolerant Index Cache Using In-memory Delta Cell

Kun Ma[1,2] and Bo Yang[1]
[1] Shandong Provincial Key Laboratory of Network Based Intelligent Computing,
University of Jinan, Jinan 250022, China
[2] Shandong Provincial Key Laboratory of Software Engineering,
Shandong University, Jinan 250100, China
E-mail: ise_mak@ujn.edu.cn http://kunma.net

Keywords: index, cache, delta cell, query cache, indexing, relational database

Received: October 8, 2015

Relational database indexes, used to speed up access to data stored in a database, are maintained when data in the source table of the index is modified. Therefore, relational database index management can involve time consuming manual analysis and specialized development efforts, and impose organizational overhead and database usage costs, especially in the context of big data. To address this limitation, this paper proposes an asynchronous adaptive delay tolerant index cache using in-memory delta cell. The contributions of index cache are adaptive management and fine-grained delta cell. Finally, our experimental evaluation shows that this simple index cache has the features such as update efficiency with frequent changes, transparency to developers, and low impact on database performance.

Povzetek: Predstavljena je nova oblika indeksa, ki omogoča boljše delovanje relacijskih baz.

1 Introduction

1.1 Background

Relational databases are organized collections of data using schemas such as tables, records and columns. Information retrieval can be made more efficient by using relational indexes to provide rapid access to data stored in a table [1]. An index is a data structure that is created using one or more columns of a base table using balanced trees, B+ trees, and hashes techniques. Indexes are updated when data in the source table of the index is modified. Therefore, indexes maintenance [2] is performed to provide accurate responses to applications that retrieve data in the presence of frequent changes. Generally, an index is updated immediately when data in its source table is modified [2]. Changes to base tables result from statements to insert, update, or delete records in the base table. Maintaining an index immediately may be inefficient due to frequent changes. For instance, a particular record may be modified several times before it is read when evaluating a query. In this situation, only the latest change to this record before the query is concerned. In addition, index maintenance may occur at peak operating times of the database, especially in the context of big data. Thus, the processing power of the database may be drained due to index maintenance operations. And index maintenance has become the bottleneck of big data access.

1.2 Data access with frequent changes

To address the issue of rapid data access with frequent changes, many approaches and strategies have been proposed. The first solution is distributed cache. A distributed cache may span multiple rapid storage nodes so that it can grow in size and in transactional capacity. It is mainly used to store frequently accessed data residing in database and web session data. This solution is popular due to the cheap hardware such as memory, solid state disk, and disk array. In addition, a distributed cache works well on lower cost machines. Ehcache and Memcached are distributed cache for general purpose caching [3], originally intended for use in speeding up data access by alleviating the database load. They feature memory and disk stores, replicate by copy and invalidate, and cache loaders. However, distributed cache might be suited for the scenario in which the data is read frequently. While in the presence of frequent changes, swapped in and out lead to excessive spending on consistency.

The second solution is cache table. Cache table [4] enables persistent caching of the full or partial contents of the relational table in the distributed environment. The content of a cache table is dynamic, which is either defined in advance at setup time or determined on demand at query time. Although this solution exploits the characteristics of short transactions and simple equality predicates, too massive maintenance of cache table and extra storage spaces are needed in the context of frequent changes.

The third solution is caching query results. TxCache [5] is a transparent caching framework that supports transactions with snapshot isolation. It is designed to cache query results, and extends them to produce invalidation tags in the presence of updates. This works when the workload of an application consists of simple exact-match selection predicates. CacheGenie [6] provides high-level caching abstractions for common query patterns, and generates SQL queries and object instances stored in the cache. It can perform this for a subset of query patterns generated by the ORM. These frameworks are suitable with minor changes of data.

The fourth solution is augmented cache. Cache augmented systems [7] [8] enhance the velocity of simple operations that read and write a small amount of data from big data, which are most suitable for those applications with workloads that exhibit a high read to write ratio. Some query intensive applications augment a database with a middle-tier cache to enhance the performance. In the presence of updates to the normalized tables, invalidation based consistency techniques delete the impacted key-value pairs residing in the cache. A subsequent reference for these key-value pairs observes a cache miss and re-computes the new values. It is difficult to keep consistency in the presence of frequent changes.

The last solution is augmented index. This method improves the traditional index in the presence of updates. Service indexes [9] are created to assist main indexes to record the changes in the presence of updates. They are maintained when there is data manipulation on main indexes. Asynchronous index [10] is a delay index to maintain database indexes or sub-indexes. After the database receives a data manipulation statement to modify particular data, the index associated with this operation is maintained asynchronously until an index maintenance event. In this situation, there are inconsistencies between the delayed index data and actual data. Index maintenance includes delta tables as well as control tables. The challenges of augmented index is how to implement adaptive index management and reduce the cost of maintaining indexes.

1.3 Contributions

The biggest disadvantage of the above five solutions is the bottleneck in the presence of frequent changes. To address this limitation, we attempt to benefit indexes from cache techniques. We call this index cache. Compared with index techniques, we attempt to address index maintenance issue using cache techniques. Unlike cache, index cache is used to speed up read and write at the same time. Thus, index cache is suitable for both high read to write ratio and high write to read ratio. Innovation points of this article lies on the following. First, we provide dynamic management of index cache. Several index management metrics (column access frequency, index maintenance frequency, and deadlock frequency) are collected to compare with the thresholds to determine management actions, such as reorganizing indexes, creating indexes and removing indexes. Proposed actions may be subject to final authorization or may be implemented automatically after the metric threshold values are satisfied. On one hand, the profiler we proposed is general to monitor data query and manipulation statements using JDBC or other middleware. On the other hand, frequency is a corrected metrics. Second, we provide delay tolerant index cache using delta cell. Index maintenance caused by data manipulation associated with this index is delayed within the tolerance. This method is based on an isolation level of a transaction including a query that triggered the index maintenance. In this solution, fine-grained delta cells are used to describe the changes of data. Reset, read, write, and consistency of index cache are also concerned. On one hand, fine-grained delta cells save more storage than delta tables using versioning management. On the other hand, the write of index cache is oriented to cache itself using eventually consistency strategy.

The remainder of this paper is organized as follows. Relevant recent work on dynamic management of index and augmented index is reviewed in Section 2. In Section 3, a description of asynchronous adaptive delay tolerant index cache is presented. First, adaptive dynamic management of index cache is provided to reorganize, create, and remove indexes by the collected metrics. Furthermore, delay tolerant method is proposed to reset, read, write of the index cache to implement the consistency using fine-grained delta cells. Section 4 presents the experimental evaluation of this asynchronous adaptive delay tolerant index cache to illustrate its update efficiency with frequent changes. Brief conclusions and future research directions are outlined in the last section.

2 Related work

2.1 Dynamic management of index

Generally, indexes are created by administrators to speed up data access. In the context of applications with high read to write ratio, indexes are competent to organize data records. Most relational database can provide benefits by controlling index fragmentation and inserting/removing indexes based on database queries [1]. Unfortunately, it involves time consuming manual analysis and specialized development efforts. In some situations, such index management may be performed without an integral management, leading to problems such as the following [11]. First, running query profilers to trace query patterns may cause significant performance overhead on databases. Second, resolving index related issues may impose organizational overhead and slow turnaround time.

Recent researches focus on automatical integral index management for a relational database. For example, dynamic integral index management actions and index management metric thresholds are provided/rectified by administrator. An index metrics collection module automatically collects metric values to determine whether to reorganize or insert/remove indexes [11]. Another case is index monitoring system for selectively maintaining an index [12]. An indication of an index usage criterion associated with each of two or more indexes is provided to efficiently determine exactly what and how indexes are used, and whether the index should be removed and created. Some well-known tuning advisors [13], such as Oracle and Microsoft SQL Server, provide index recommendations for a given work load of queries. Other relational database products also have a separate component that would read a given set of queries and provide the indexing recommendations based on storage, partitioning, and other considerations [14]. Many such products have significant limitations. For instance, the tools are manually controlled. A set of user queries to be analyzed must be captured from production database servers using profilers that can add significant performance. Implementing the index recommendations on the production databases may require IT release cycles, which is often time consuming. Sometimes the metrics are not correct enough to conclude good guiding significance.

2.2 Augmented index

Augmented index is a method to enable indexes to implement the maintenance in the presence of updates. Compared with augmented cache solution, this method is efficient in the context of frequent changes with high write to read ratios. At least a service index [9] is proposed to record the changes caused by main indexes. This is a delayed update method of index maintenance. After data manipulation on main indexes, changes are immediately saved to at least a service index. Maintenance to main indexes is delayed with the help of service index. There are several insufficiencies. First, single table with no more than one index will lead to generate more service indexes. Second, the performance impact on index maintenance is inevitable because main index maintenance is just delayed to update. In the presence of frequent changes, it will also become the bottleneck of data access.

Another augmented index is asynchronous index [10]. Asynchronous indexes may need to be maintained when records of the base table with the index are changed in response to a data manipulation statement. Asynchronously updating an index may improve the efficiency of index maintenance by reducing the number of inputs/outputs needed for index maintenance. This method is particularly efficient for a database table having frequent writes, but infrequent reads. Insufficiencies of this method lies on the following. First, delta tables to store the changes caused by index will occupies huge amounts of required storage spaces. A changes record is stored no matter how many columns have been changed. That indicates that the unchanged column is also stored as long as the record including this column is changed. Second, the merging strategies of massive records in delta tables are not discussed. Without a reasonable merging strategy, the records of delta tables grow fast in the presence of frequent updates.

3 Asynchronous dynamic delay tolerant index cache

3.1 Dynamic management by metrics

We provide a profiler on the read and write statements to regularly monitor the workload (a set of data query and manipulation statements that execute against a database) and control indexes management actions appropriately, to remove unused indexes, to re-organize used indexes, and to create required indexes based on the frequency. We define column access frequency, index maintenance frequency, and deadlock frequency to determine whether to maintain indexes dynamically.

We want to create indexes on frequently accessed column, to remove indexes on frequently index maintenance column, and to re-organize indexes on tables with many deadlocks. The frequency is the broad frequency belonging to one column. We take different frequencies as the metrics of index management.

3.1.1 Column access frequency

Column access frequency reflects the frequency that one column is accessed. When the data in this column are accessed by querying, the column access frequency count is plus an increment. The reason why it is called broad is that the increment is not simply one, depending on the product of the priority weight of this column and the correction factor. When the column access frequency exceeds the threshold, the index on this column should be created to speed up the data access. Column access frequency count f is computed by the query predicates: selection predicates, aggregate predicates, and ordered predicates. To describe the rectification of the column access frequency, we assume the following terminology for a SQL query:

SELECT *target list* **FROM** *table list*
WHERE *qualification list*
ORDER BY *ordered list*

We consider the column access frequency on three predicates: selection predicates (target, exact-match and range selection), aggregate predicates, and ordered predicates. At the beginning, the column access frequency count is zero. Other weights and factors are empirically determined.

We describe the affection of the target and exact-match predicates in turn. Consider the following query with a quantification list consisting of exact-match selection predicates:

SELECT $a_1, a_2, ..., a_n$ **FROM** *table list*
WHERE $a_1 = C_1$ and/or $a_2 = C_2$ ... and/or $a_m = C_m$

The proposed profiler constructs the rectification of the column access frequency count. If the column is located in the target list, the access frequency count f_{ai} of column ai is plus to the product of the weight ws_{ai} of the column and selection correction factor ks, denoted as $f = f + ws_{ai} * ks$. If the column is located in the exact-match list, the access frequency count f_{ai} of column ai is plus to the product of the weight wm_{ai} of the column and exact-match correction factor km, denoted as $f = f + wm_{ai} * km$.

We describe the affection of the range selection predicates. Consider the following query with a qualification list consisting of range selection predicates:
SELECT *target list* **FROM** *table list*
WHERE $(a_1 > C_1 and a_1 < C_2)$ and/or ... and/or $(a_m > C_{2k}$ and $a_1 < C_{2k+1})$

If the column is located in the range list, the access frequency count f_{ai} of column ai is plus to two times the product of the weight wr_{ai} of the column and range correction factor kr, denoted as $f = f + 2 * wr_{ai} * kr$.

We describe the affection of the aggregate selection predicates. Consider the following query with a quantification list consisting of aggregate selection predicates:
SELECT $function_1(a_1), ..., function_m(a_m)$
FROM *table list*
WHERE *quantification list*

If the column is located in the aggregate list, the access frequency count f_{ai} of column ai is plus to the product of the weight wa_{ai} of the column and aggregate correction factor ka, denoted as $f = f + wa_{ai} * ka$.

We describe the affection of the ordered selection predicates. Consider the following query with a quantification list consisting of ordered selection predicates:
SELECT *target list* **FROM** *table list*
WHERE *quantification list*
ORDER BY a_1 $asc/desc, ..., a_m$ $asc/desc$

If the column is located in the ordered list, the access frequency count f_{ai} of column ai is plus to the product of the weight wo_{ai} of the column and ordered correction factor ko, denoted as $f = f + wo_{ai} * ko$.

3.1.2 Index maintenance frequency

Index maintenance frequency reflects the frequency that indexes are maintained due to data manipulation. When the index is rebuilt by the changed column, the index maintenance frequency of this column is plus an increment. The reason why it is called broad is that the increment is not simply one, depending on the product of the priority weight of this column and the correction factor. Index maintenance frequency count g of one column i is denoted as $g = g + w_{bi} * k$, where w_{bi} is the weight, and k is index maintenance correction factor. When the index maintenance frequency exceeds the threshold, the index on this column should be removed to speed up the data access with changes.

3.1.3 Deadlock frequency

Deadlock frequency reflects the times that one table is locked by the index maintenance. Table access is locked when the index maintenance is not complete. We provide lock frequency h to record the deadlock times. When table access is locked, the deadlock frequency is plus one. When the deadlock frequency exceeds the threshold, a manual check is needed to re-organize the indexes.

3.2 Delay tolerant index cache using delta cell

3.2.1 Delta cell

In order to define the architecture and management actions of the delay tolerant index cache, a mathematical representation of the fine-grained model is necessary. In our solution, we split the storage structure of the basic element of index cache into sets of delta cells divided by column. Delta cell is a fine-grained model of frequent changes of a relational database.

First, we define the elements of a delta cell.

- key is the key of a delta cell. It corresponds to the key of relational changed record before it is divided into cells. key is denoted as 2-tuple $key :< keyname, keyvalue >$, where $keyname$ is the key name of the record, and $keyvalue$ is the key value of the record;
- C is key/value of this delta cell. It is denoted as 2-tuple $C :< name, value >$, where $name$ and $value$ are the name and value of this delta cell respectively.
- V, $V \in \mathbb{Z}^+$, is the version number of this delta cell. It is a non-negative integer. The initial version number of a delta cell is one. When the delta cell is removed, the version number is set zero.

Second, we give the definition of a delta cell. The delta cell is a 3-tuple $< key, C, V >$, where key is the key of a delta cell, C is key/value of this delta cell, and V is the version number of this delta cell. We take schema-free key/value stores to save delta cells. The query of delta cells is through SQL-like HiveQL [15].

As mentioned, delta cells are the first-class artifacts to represent frequent changes. These models are typically created and modified by the profiler we design. One of the techniques used to support index cache management activities is version control. Version control is used during delta cell evolution to keep track of different versions of delta cell artifacts produced over time. Version control enables simultaneous transactions to access the delta cells that stores different versions of the data. When a transaction updates the delta cell, it maintains its previous versions. After index cache is reset, all the versions of delta cells are emptied.

3.2.2 Architecture of index cache

Figure 1 shows the architecture of our proposed asynchronous adaptive delay tolerant index cache. Index cache is the supplement of actual data with indexes. When there are data query statements, the query results are from the merging of index cache and actual data with indexes. When

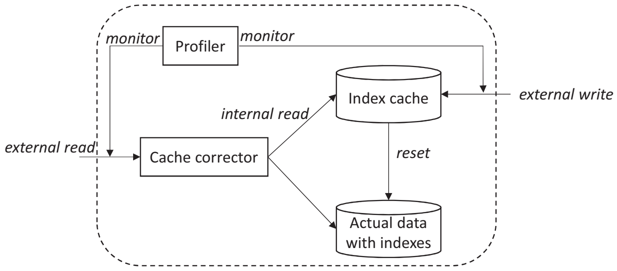


Figure 1: Architecture of asynchronous adaptive delay tolerant index cache.

there are data manipulation statements, all the updates are written to index cache. Index cache is reset triggered by forced update event or idle update event. With this architecture, there is no immediately index maintenance until a forced or idle update event generates. Besides, profiler is used to monitor the external read and write operations to collect the frequencies to adaptively manage indexes.

3.2.3 Reset of index cache

Triggered by a forced or idle update event, the data in index cache is forced to be written to actual data with indexes. When the version number of the delta cell in the index cache is zero, the corresponding record in the actual data with indexes should be removed. When the version number of the delta cell in the index cache is a positive integer, the latest version of this delta cell in the index cache should replace the original record in the actual data with indexes.

3.2.4 Read of index cache

In the architecture of delay tolerant index cache, the query results are from both index cache and actual data with indexes. In order to make the index cache transparent to the user, the query results should be corrected by the inner result corrector in the index cache. When there is a data query statement, the result corrector delivers the query request to both the actual data with indexes and index cache at the same time. In the index cache, only delta cells with a positive integer are to execute the query statement. The query of delta cells is using HiveQL [15]. Afterwards, the results from both index cache and actual data with indexes are merged together. The merging action needs to meet the mergence rules shown below:

- Results with the same key: the query results from index cache replace the results from actual data with indexes.

- Results with different keys: the final results are the union set of both index cache and actual data with indexes.

3.2.5 Write of index cache

With index cache architecture, the write of index cache acts on only itself rather than actual data with indexes. For the creation data manipulation statement, the new record is broken down into several newborn delta cells with version number 1. For the delete data manipulation statement, the removed record is broken down into several destroyed delta cells with version number 0. For the update data manipulation statement, it is divided into two cases. When the changed data is in the index cache, the only thing to do is to update the existing delta cell with the changed data. When the changed data is not in the index cache, the only thing to do is to create a newborn delta cell with version number 1.

In the process of write of index cache, the delta cells are created and updated in order. That is to say that the same delta cell might be updated more than once in a short time. For instance, the data is first inserted, then updated, and deleted at last. Therefore, update merging method is introduced to merge the intermediate result. Afterwards, the delta cells are in no particular order.

Table 1: Merging result of both actions.

Action 1	Action 2	Merging result
Insert	Insert	×
Insert	Update	Insert
Insert	Delete	Ignore
Update	Insert	×
Update	Update	Update
Update	Delete	Delete
Delete	Insert	Insert
Delete	Update	×
Delete	Delete	×

Merging of the delta cells reduces the times of several updates to the final update when data are updated more than once. The mergence rules are shown in Table 1. After continuous two actions of the same delta cell, the final merging result is shown in the third column. The expected merging results might be $impossible$ (×), unchanged ($ignore$).

4 Experiments

We have conducted a set of experiments to evaluate the efficiency and effectiveness of our proposed asynchronous adaptive delay tolerant index cache using delta cell. After a description of the experimental setup, we evaluate three solutions (database without any external index optimizations, augmented index, and index cache).

4.1 Experiment setup

We deploy the experiment architecture with Intel Core(R) i5-2300 @2.80 GHz CPU, 16GB memory. It runs a 64-bit

CentOS Linux OS with a Java 1.6 64-bit server JVM. We use MySQL server 5.6 GA as the relational database. We initialize 1, 000, 000 relational records with 20 columns, 1 primary key, and 4 indexes (each index is on one column).

4.2 Update time with frequent changes

We evaluate three different solutions under three circumstances. The x axis is transactional workload (presented using transactions per second), and the y axis is the average time executing 1, 000 data manipulation statements. To increase comparability of the results, 1, 000 statements include one third of new records, one third of changed records, and one third of deleted records. We take asynchronous index as an example of augmented index.

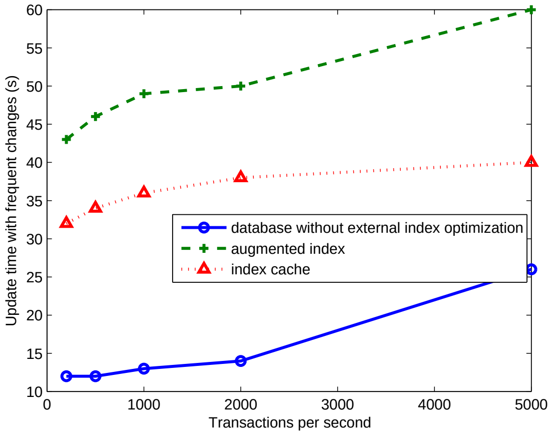


Figure 2: Update time when randomly updating non-index columns.

The first circumstance is randomly updating non-index columns (other 16 columns except 4 index columns). Figure 2 shows the average update time with different transactions per second. Since the frequent changes are not in the index columns, database without external index optimizations solution has the smallest update time with the increasing of transactions per second. Unfortunately, the augmented index works not well due to massive index maintenance. The update time of our proposed index cache is in the middle, because the write of index cache is just in the index cache itself without index maintenance.

The second circumstance is randomly updating 4 index columns. Figure 3 shows the average update time with different transactions per second. Since the frequent changes lie in the index columns, index maintenance issue become the bottleneck of the updates. Database without external index optimizations solution is the worst. When the transactions per second are below 500, the update time of our index cache solution is a little larger than asynchronous index solution. That is due to the reset of index cache in the presence of low frequency. When the transactions per second exceed 500, our index cache is starting to change for the better. That is caused by little reset of index cache in the presence of high frequency.

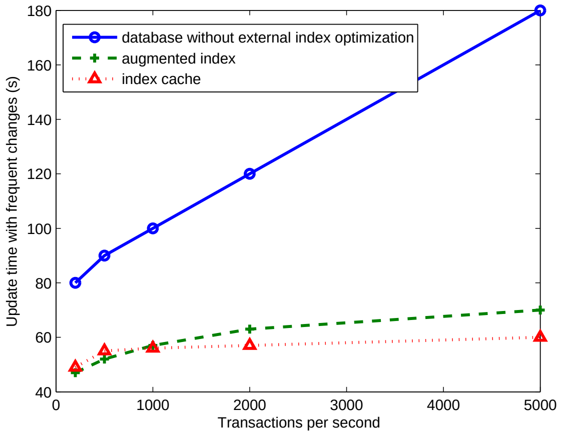


Figure 3: Update time when randomly updating 4 index columns.

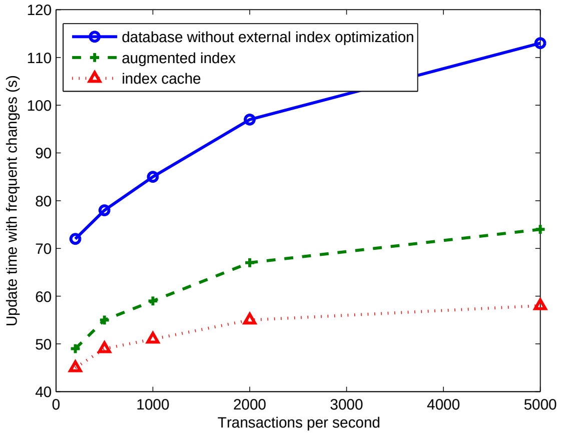


Figure 4: Update time when randomly updating 8 columns.

The third circumstance is randomly updating 8 columns (3 index columns and 5 non-index columns). Figure 4 shows the average update time with different transactions per second. Database without external index optimizations solution is the worst, because index maintenance on 3 index columns takes up the update time. Our index cache works better due to the dynamic index management by metrics. After the experiment, our index cache solution removes 1 index on the frequent updated columns, and creates 2 new indexes on 2 frequest accessed columns.

5 Conclusions

To reduce index maintenance, this paper has propose the asynchronous adaptive delay tolerant index cache using delta cell. This method has some features such as dynamic index management and fine-grained controls. This is a new method to improve the database performance.

Acknowledgement

This work was supported by the Doctoral Fund of University of Jinan (XBS1237), the Shandong Provincial Natural Science Foundation (ZR2014FQ029), the Shandong Provincial Key R&D Program (2015GGX106007), the Teaching Research Project of University of Jinan (J1344), the National Key Technology R&D Program (2012BAF12B07), and the Open Project Funding of Shandong Provincial Key Laboratory of Software Engineering (No. 2015SE03).

References

[1] Radoslaw Boronski and Grzegorz Bocewicz. Relational database index selection algorithm. In *Computer Networks*, pages 338–347. Springer, 2014.

[2] Harumi Kuno and Goetz Graefe. Deferred maintenance of indexes and of materialized views. In *Databases in Networked Information Systems*, pages 312–323. Springer, 2011.

[3] Brad Fitzpatrick. Distributed caching with memcached. *Linux journal*, 2004(124):5, 2004.

[4] Mehmet Altinel, Christof Bornhövd, Sailesh Krishnamurthy, Chandrasekaran Mohan, Hamid Pirahesh, and Berthold Reinwald. Cache tables: Paving the way for an adaptive database cache. In *Proceedings of the 29th international conference on Very large data bases-Volume 29*, pages 718–729. VLDB Endowment, 2003.

[5] Dan RK Ports, Austin T Clements, Irene Zhang, Samuel Madden, and Barbara Liskov. Transactional consistency and automatic management in an application data cache. In *OSDI*, volume 10, pages 1–15, 2010.

[6] Priya Gupta, Nickolai Zeldovich, and Samuel Madden. A trigger-based middleware cache for orms. In *Middleware 2011*, pages 329–349. Springer, 2011.

[7] Shahram Ghandeharizadeh and Jason Yap. Cache augmented database management systems. In *Proceedings of the ACM SIGMOD Workshop on Databases and Social Networks*, pages 31–36. ACM, 2013.

[8] Shahram Ghandeharizadeh and Jason Yap. Gumball: a race condition prevention technique for cache augmented sql database management systems. In *Proceedings of the 2nd ACM SIGMOD Workshop on Databases and Social Networks*, pages 1–6. ACM, 2012.

[9] Ying Ming Gao, Jia Huo, Kai Zhang, and Xian Zou. Database index management, February 13 2012. US Patent App. 13/371,577.

[10] Peter A Carlin, Per-Ake Larson, and Jingren Zhou. Asynchronous database index maintenance, March 20 2012. US Patent 8,140,495.

[11] Meiyalagan Balasubramanian and Rohit Sabharwal. Dynamic integrated database index management, July 16 2013. US Patent 8,489,565.

[12] John Martin Whitehead, Subrahmanyeswar Vadali, and Kalur Sai Kishan. Database index monitoring system, January 7 2014. US Patent 8,626,729.

[13] Sanjay Agrawal, Surajit Chaudhuri, Lubor Kollar, Arun Marathe, Vivek Narasayya, and Manoj Syamala. Database tuning advisor for microsoft sql server 2005: demo. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 930–932. ACM, 2005.

[14] Gary Valentin, Michael Zuliani, Daniel C Zilio, Guy Lohman, and Alan Skelley. Db2 advisor: An optimizer smart enough to recommend its own indexes. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pages 101–101. IEEE Computer Society, 2000.

[15] Ashish Thusoo, Joydeep Sen Sarma, Namit Jain, Zheng Shao, Prasad Chakka, Suresh Anthony, Hao Liu, Pete Wyckoff, and Raghotham Murthy. Hive: a warehousing solution over a map-reduce framework. *Proceedings of the VLDB Endowment*, 2(2):1626–1629, 2009.

Stackelberg Surveillance

Bikramjit Banerjee and Landon Kraemer
School of Computing, The University of Southern Mississippi
118 College Dr. #5106, Hattiesburg, MS 39406, U.S.A
E-mail: Bikramjit.Banerjee@usm.edu,Landon.Kraemer@eagles.usm.edu

Keywords: Stackelberg games, security games, camera surveillance

Received: October 23, 2014

Bayesian Stackelberg game theory has recently been applied for security-resource allocation at ports and airports, transportation, shipping and infrastructure, modeled as security games. *We model the interactions in a camera surveillance problem as a security game, and show that the Stackelberg equilibrium of this game can be formulated as the solution to a non-linear program (NLP). We provide two approximate solutions to this formulation: (a) a linear approximation based on an existing approach (called ASAP), and (b) a hill-climbing based policy search approximation. The first reduces the problem to a single (but difficult) linear program, while the second reduces it to a set of (easier) linear programs. We consider two variants of the problem: one where the camera is visible, and another where it is contained in a tinted enclosure. We show experimental results comparing our approaches to standard NLP solvers.*

Povzetek: V zadnjih letih se v varnostnih nalogah pogosto uporablja metode za iskanje ravnotežja. V prispevku je predstavljena teorija iger na osnovi Bayesa in Stackelberga.

1 Introduction

Bayesian Stackelberg game theory has recently been applied for security-resource allocation at ports and airports, transportation, shipping and infrastructure [7]. Stackelberg games are played by two players: a "leader" and a "follower". In security applications, these players are referred to as "defender" and "attacker" respectively. Typically a defender (leader) acts first by committing to a patrolling/inspection strategy, which is observed by an attacker (follower) of some type. The attacker then plays a best response, such as attack what it predicts to be the weakest/least protected asset, which also determines the defender's payoff besides the attacker's own payoff. The task for the defender is to play its *expected*-payoff-maximizing strategy, knowing the game payoffs (both its and the various types of attackers' payoffs) and the distribution over attacker types. These games are typically non-zero-sum, i.e., one player's loss does not numerically equal the other player's gain. The defender's optimal strategy incorporates randomization because security-resources are typically limited, i.e., not every asset can be simultaneously protected. In the rest of this article, we shall employ the security application terminology and refer to the players as "defender" and "attacker", instead of the general leader-follower game theoretic terminology.

In this article, we formulate a camera surveillance problem as a security game. In a typical camera surveillance scenario, a few fixed cameras are located in strategic spots, each with large coverage and concomitantly low resolution, that often fail to give sufficient details of forensic value after a crime. We consider unmanned surveillance, where no active control of the cameras is possible. For instance, in our university campus there are two cameras in each computer lab, yet articles have been stolen and never have the (fuzzy, grainy) footage enabled post facto identification of any perpetrator. The reliance on short focus setting (i.e., low zoom) aims to balance between the amount of data collected and coverage of the surveilled area. Therefore, attempts to solve the problem by increasing the number of cameras increases the amount of data collected (besides cost), or by increasing the resolution of each camera increases the cost and demand on technology for large surveilled areas.

Our goal is to allow cameras to operate at long focus settings (i.e., high zoom) to capture greater details for forensic value. However, this would lead to reduction in space coverage (unless we deploy a lot of cameras to regain coverage, but this would also blow up the amount of collected data). To address this problem, we allow the cameras to *turn* (i.e., move from one pan/tilt setting to another) at a Stackelberg-randomized schedule, regaining coverage *in time*, without increasing the amount of collected data compared to the fixed cameras scenario.

Figures 1, 2 illustrate the problem and our approach. The target scenario is of unmanned video recordings (from fixed cameras) that may be called for a closer look *post facto*, e.g., for a crime investigation after the crime has occurred. Figure 1 shows a snapshot from such a (real) video recording, where a vehicle is identified as the object of interest. However, zooming in post facto (Figure 2) does not help; not only the license plate but also the make/model are not discernible. The alternative envisaged in this article is a Stackelberg optimized schedule of $(pan,\ tilt)$ settings

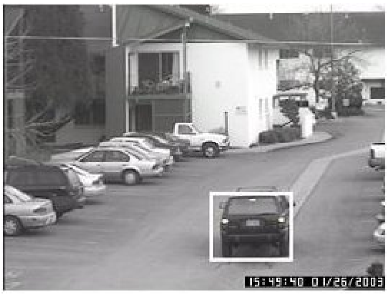


Figure 1: A snapshot from a security video.

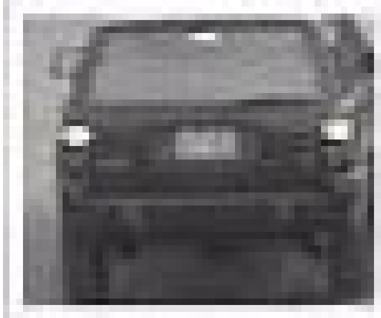


Figure 2: Zooming in to an object of interest in Figure 1 hardly gives any information.

for a camera always operating at a high zoom setting, so that there would be a high likelihood of the camera catching details of the vehicle—thus being of post facto forensic value—even if the vehicle was actively trying to evade it. Instead of actually optimizing likelihoods, we optimize the expected payoff of the camera by assigning differing values to different strategic locations (in an abstract waypoint graph). For instance, in Figure 1, the choke-point (around the bend) could be valued highly, resulting in more frequent coverage of it.

Our formulation of the camera surveillance problem as a security game shows that the Stackelberg equilibrium is given by the solution to a mixed integer non-linear program. We evaluate two first-cut approximate approaches: (a) a linear approximation approach that reduces the mixed integer non-linear program (MI-NLP) to a single (but difficult) mixed integer linear program (MILP), and (b) a policy search approach that generates many mixed integer linear programs that are potentially easier to solve because they have fewer integer variables (and constraints). Our experiments show that indeed the policy search approach is more scalable and produces higher quality solutions, compared not only to (a) but also to some standard NLP solvers.

2 Problem formulation

Henceforth, we will refer to the camera as the "defender" and any target of future interest as the "attacker". We define the camera surveillance problem as a tuple $\langle L, O, T_a, T_d, R_a, R_d \rangle$, where

- L is a set of potential locations of the attacker (i.e., vertices in a waypoint graph),
- O is a set of defender orientations (i.e., discrete pan-tilt settings),
- $T_a(\ell)$ denotes the set of (neighboring) locations that the attacker can reach from location $\ell \in L$ in one step,
- $T_d(o)$ denotes the set of (neighboring) orientations that the defender can reach from orientation $o \in O$ in one step,
- $R_a(\ell, o)$ and $R_d(\ell, o)$ denote the rewards received by the attacker and defender, respectively, when the attacker is in location $\ell \in L$ and the defender orientation is $o \in O$.

We assume discretized time, and that the defender and attacker can change (or not) their current orientation/location simultaneously on each tick. Since the defender wants to cover the current location of the attacker, but the latter wants to evade the defender, $R_d(\ell, o) > 0$ and $R_a(\ell, o) < 0$ iff ℓ is *covered* in orientation o. Otherwise, we assume $R_d = 0$ and $R_a \geq 0$. R_a can also vary according to attacker types. As in general security games, this application is not zero/constant-sum. This is easily seen from the fact that the defender's valuation of assets that it covers may differ from the attacker's, which may further vary by attacker types. For instance, a shoplifter in a superstore may value a flash drive more than a 65" television. On the other hand, a vandal's valuation of a television may be higher than a pricier piece of wood furniture, but negligible for a flash drive. However, it is reasonable to assume that the defender has a fixed valuation for all assets—a departure from traditional security games where the defender's valuations can vary by attacker types. In this article, we assume a single attacker type, but our methodology can be easily extended to multiple attacker types.

Given a problem model $\langle L, O, T_a, T_d, R_a, R_d \rangle$, the goal of the defender is to find a policy that maximizes its expected reward, assuming that the attacker will always play the best response to the defender's policy. Since the defender is unable to observe the attacker's location (it doesn't know who the attacker is), its policy is based on its current orientation only. If it was a deterministic policy that always mapped an orientation to the same neighboring orientation, then an attacker's best response could deterministically allow it to stay out of the defender's view. In fact, as in general security games, this application also presents *resource constraint*, where the resource is the ability to cover waypoints at high enough resolution to be of high forensic value later. Therefore, we use a stochastic policy representation for the defender. It is represented as a real-valued transition function, giving a probability distribution over its next (neighboring) orientation given its current orientation $\zeta : O \times O \mapsto [0, 1]$. The attacker's policy is a Boolean function that gives a mapping from location-orientation pairs to subsequent (neighboring) locations, $\sigma : L \times O \times L \mapsto \{0, 1\}$. Thus we assume that the attacker can observe the defender's current orientation in its decision making.

Although the problem can be considered episodic from the perspective of a particular attacker (with well-defined source and sink vertices in its waypoint graph), from the defender's perspective it is a continuing task, with no horizon, because it never knows the attacker (except after the fact). Since we solve the problem from the defender's perspective with indefinite attacker, we consider the *steady state* of the Markov chain over $L \times O$ occupancies (i.e. the state space) for a given joint policy $\langle \zeta, \sigma \rangle$. Notice that from the defender's perspective the states of the Markov chain are positive recurrent. Now for the particular joint policy $\langle \zeta, \sigma \rangle$, the steady-state probability distribution over $L \times O$ is given by the solution to the following set of recursive equations:

$$P(\ell', o'|\zeta, \sigma) = \sum_{\ell \in L, o \in O} \zeta(o, o')\sigma(\ell, o, \ell')P(\ell, o|\zeta, \sigma). \tag{1}$$

We make a key assumption that the above Markov chain is *irreducible*. This is a reasonable assumption in any surveillance domain because typical "blind spots"—locations that the defender cannot cover (such as restrooms), or the attacker cannot access—can simply be omitted from the attacker's waypoint graph. Thus the system of Equations 1 must have a unique solution.

The attacker's best-response to a particular defender policy ζ, is given by

$$\sigma^{\zeta} = \operatorname*{argmax}_{\sigma} \frac{1}{1-\gamma} \sum_{\ell \in L, o \in O} P(\ell, o|\zeta, \sigma)R_a(\ell, o), \tag{2}$$

where $\gamma \in [0, 1)$ is a discount factor. The defender's goal, then, is to find

$$\zeta^* = \operatorname*{argmax}_{\zeta} \frac{1}{1-\gamma} \sum_{\ell \in L, o \in O} P(\ell, o|\zeta, \sigma^{\zeta})R_d(\ell, o). \tag{3}$$

The above solution based on the steady state is a departure from traditional security games, and the key reason why the resulting program turns out to be non-linear (elaborated at the end of the next section). We formulate the solution program in the next section. In the rest of this article, we will ignore the multiplicative constant involving the discount factor in the objective functions only.

3 Mixed Integer Non-linear Program (MI-NLP)

We use the following variables to formulate the optimization problem for Equation 3:

- $\zeta : O \times O \mapsto [0, 1]$ variables represent the defender's policy, i.e. $\zeta(o, o')$ gives the likelihood that the defender will transition to orientation $o' \in O$ from orientation $o \in O$.
- $\sigma : L \times O \times L \mapsto \{0, 1\}$ represent the attacker's deterministic policy, i.e. if the attacker would transition to location $l' \in L$ from state $(l, o) \in L \times O$, then $\sigma(l, o, l') = 1$.
- $X : L \times O \mapsto [0, 1]$ variables represent the steady state joint occupation probabilities from Equation 1, i.e. $P(\ell, o|\zeta, \sigma)$.
- $v : L \times O \times L \mapsto \Re$ variables that represent the attacker's optimal expected value function for transitioning to location ℓ' from state ℓ, o:

$$v(\ell, o, \ell') = R_a(\ell, o)X(\ell, o) + \gamma \sum_{o' \in T_d(o)} \zeta(o, o') \max_{\ell''} v(\ell', o', \ell'')$$

- $maxV : L \times O \mapsto \Re$ variables represent optimal v,

$$maxV(\ell, o) = \max_{\ell'} v(\ell, o, \ell'). \tag{4}$$

- $maxVE : L \times O \times O \mapsto \Re$ variables represent the products

$$maxVE(\ell, o, o') = \zeta(o, o')maxV(\ell, o'). \tag{5}$$

The linear objective function is

$$\text{Maximize: } \sum_{\ell \in L, o \in O} X(\ell, o)R_d(\ell, o)$$

and the constraints include (in addition to Equation 5)

- the steady state probabilities, $\forall \ell' \in L, o' \in O$

$$X(\ell', o') = \sum_{\substack{\ell \in T_a^{-1}(\ell'),\\ o \in T_d^{-1}(o')}} \zeta(o, o')\sigma(\ell, o, \ell')X(\ell, o) \tag{6}$$

- the linearization of Equation 4, $\forall \ell \in L, o \in O, \ell' \in T_a(\ell)$ and for a large enough M,

$$maxV(\ell, o) \geq v(\ell, o, \ell')$$
$$maxV(\ell, o) \leq v(\ell, o, \ell') + M(1 - \sigma(\ell, o, \ell'))$$

- the resulting linearized v function, $\forall \ell \in L, o \in O, \ell' \in T_a(\ell)$,

$$v(\ell, o, \ell') = R_a(\ell, o)X(\ell, o) + \gamma \sum_{o' \in T_d(o)} maxVE(\ell', o, o')$$

- probability distribution and mutual exclusivity constraints:

$$\forall o \in O, \sum_{o' \in T_d(o)} \zeta(o, o') = 1 \tag{7}$$
$$\sum_{\ell \in L, o \in O} X(\ell, o) = 1$$
$$\forall \ell \in L, o \in O, \sum_{\ell' \in T_a(\ell)} \sigma(\ell, o, \ell') = 1$$

In most security games, the game is assumed to be one-shot, i.e., the game ends for both players when the attacker succeeds or fails. However, in our case the game never really ends for the defender, as the attacker is indefinite. As a consequence, our formulation has a steady state term $X(\ell, o)$, which does not appear in traditional security games. We shall see in the next section that this new term $X(\ell, o)$, particularly its presence in Constraint 6, is the only roadblock to completely linearizing the above MI-NLP. Thus the use of steady state in our solution turns out to be the key reason why our formulation is non-linear.

4 Linear approximation

Note that in the above NLP, all constraints are linear except for Equations 6 and 5. Constraint 6's non-linearity lies in the summands, which are products of three variables $\zeta(o, o')$,$\sigma(\ell, o, \ell')$, and $X(\ell, o)$. In order to focus on the summands individually, we represent each $\zeta(o, o')\sigma(\ell, o, \ell')X(\ell, o)$ summand with a new variable $\beta(\ell, o, \ell', o') \in [0, 1]$ and rewrite constraint 6 as

$$\forall \ell' \in L, o' \in O, X(\ell', o') = \sum_{\substack{\ell \in T_a^{-1}(\ell'), \\ o \in T_d^{-1}(o')}} \beta(\ell, o, \ell', o') \quad (8)$$

and add the constraint

$$\sum_{\ell \in L, o \in O, \ell' \in T_a(\ell), o' \in T_d(o)} \beta(\ell, o, \ell', o') = 1. \quad (9)$$

Now we constrain the $\beta(\ell, o, \ell', o')$ variables without reintroducing non-linearity. First, we note that since $\sigma(\ell, o, \ell')$ is binary, the product $\zeta(o, o')\sigma(\ell, o, \ell')X(\ell, o)$ will be zero when $\sigma(\ell, o, \ell') = 0$, and it will be $\zeta(o, o')X(\ell, o)$ when $\sigma(\ell, o, \ell') = 1$. While the former is relatively simple, the latter is still non-linear.

In determining a practical way to linearize $\zeta(o, o')X(\ell, o)$, it is useful to consider Constraint 5, which also needs to be linearized. Constraint 5 contains the product $\zeta(o, o')maxV(\ell, o')$. Note that if $\zeta(o, o')$ were constant, then both of these expressions would be linearlized. To this end, we invoke the *limited randomization* approach (ASAP) from [4] where the defender's mixed strategy is limited to be integer multiples of $1/k$ for a predetermined integer k. That is, we introduce a set of discrete, constant "snap" points $S = \{\frac{0}{|S|-1}, \frac{1}{|S|-1}, \ldots, \frac{|S|-1}{|S|-1}\}$ and a set of indicator variables $I_S : S \times O \times O \in \{0, 1\}$ constrained as follows: $\forall s \in S, o \in O, o' \in T_d(o)$,

$$\begin{aligned} \zeta(o, o') &\geq s \cdot I_S(s, o, o') \\ \zeta(o, o') &\leq s + M(1 - I_S(s, o, o')) \end{aligned} \quad (10)$$

in addition to: $\forall o \in O, o' \in T_d(o), \sum_{s \in S} I_S(s, o, o') = 1$. Together, these constraints ensure that $\zeta(o, o') \in S$. Note that Constraint 7 is still required to ensure that $\zeta(o, *)$ is a proper distribution.

To demonstrate how this discretization is used, we first address the non-linear Constraint 5. We want to define $maxVE(\ell, o, o')$ to be equal to $s \cdot maxV(\ell, o')$ for the snap point corresponding to $\zeta(o, o')$, i.e. s s.t. $I_S(s, o, o') = 1$. Thus we replace Constraint 5 with the following pair: $\forall s \in S, \ell \in L, o \in O, o' \in T_d(o)$,

$$\begin{aligned} maxVE(\ell, o, o') &\leq s \cdot maxV(\ell, o') + M(1 - I_S(s, o, o')) \\ maxVE(\ell, o, o') &\geq s \cdot maxV(\ell, o') - M(1 - I_S(s, o, o')). \end{aligned}$$

Next we use a similar approach to appropriately constrain the $\beta(\ell, o, \ell', o')$ variables. First, we bound $\beta(\ell, o, \ell', o')$ from above with the following: $\forall \ell \in L, o \in O, \ell' \in T_a(\ell), o' \in T_d(o)$,

$$\begin{aligned} &\beta(\ell, o, \ell', o') \leq \sigma(\ell, o, \ell') \quad (11) \\ &\forall s \in S, \beta(\ell, o, \ell', o') \leq s \cdot X(\ell, o) + M(1 - I_S(s, o, o')). \end{aligned}$$

Then, we bound $\beta(\ell, o, \ell', o')$ from below with: $\forall s \in S, \ell \in L, o \in O, \ell' \in T_a(\ell), o' \in T_d(o)$,

$$\begin{aligned} \beta(\ell, o, \ell', o') \geq s \cdot X(\ell, o) - \\ M[2 - I_S(s, o, o') - \sigma(\ell, o, \ell')] \end{aligned} \quad (12)$$

Finally, while we have related X to a summation over β in Constraint 8, we must add another similar constraint to ensure that $X(\ell, o)$ values are recursively consistent:

$$\forall \ell \in L, o \in O, X(\ell, o) = \sum_{\substack{\ell' \in T_a(\ell), \\ o' \in T_d(o)}} \beta(\ell, o, \ell', o') \quad (13)$$

Thus the MI-NLP reduces to an MILP, whose solution will be an approximation of the exact NLP solution. However, the above ASAP [4] approach, like ASAP itself, incorporates many more integer variables (and constraints) which poses a challenge to linear programming solvers. In the next section, we describe a different approximate approach.

5 Policy search

The above linear approximation of the original NLP requires simultaneous solution of both the defender's and attacker's policies (which are dependent upon each other), which can require significant computation time. In this section, we present a potentially more efficient alternative to this approach.

In a Stackelberg game, it is assumed that the attacker has full knowledge of the defender's policy, and therefore, an attacker's equilibrium policy is an optimal response to the defender's equilibrium policy. On the other hand, the defender's equilibrium policy is optimized with respect to the space of attacker responses. It is important to note that *when the defender's policy is given* (i.e., ζ are constants), the attacker's optimal policy is given by the following (mixed integer) *linear* program:

Algorithm 1 POLICYSEARCH$(restarts, \delta)$

```
1:  $overallMax \leftarrow (-\infty, \emptyset)$
2:  for $r = 1 \dots restarts$ do
3:      $currentMax \leftarrow randomPolicy()$
4:      while True do
5:          $done \leftarrow true$
6:          $neighbors \leftarrow$GETNEIGHBORS$(currentMax, \delta)$
7:          for $n \in neighbors$ do
8:              $solution \leftarrow (evaluate(n), n)$
9:              if $solution_1 > currentMax_1$ then
10:                 $currentMax \leftarrow solution$
11:                 $done \leftarrow false$
12:             end if
13:         end for
14:         if $done$ then
15:             break
16:         end if
17:     end while
18:     if $currentMax_1 > overallMax_1$ then
19:         $overallMax \leftarrow currentMax$
20:     end if
21: end for
22: Return $overallMax$
```

$$\text{Maximize:} \sum_{\ell \in L, o \in O} X(\ell, o) R_a(\ell, o)$$

subject to all constraints in the MI-NLP formulation (including Equation 5 which is now linear because ζ is given), except Equation 6. To replace Equation 6, we use the following set of linear constraints: Equations 8, 9, 11, 13, and the following pair: $\forall \ell \in L, o \in O, \ell' \in T_a(\ell), o' \in T_d(o)$,

$$\beta(\ell, o, \ell', o') \leq \zeta(o, o') X(\ell, o) \quad (14)$$

$$\beta(\ell, o, \ell', o') + M(1 - \sigma(\ell, o, \ell')) \geq \zeta(o, o') X(\ell, o) \quad (15)$$

which are again linear for a given ζ.

To leverage this linearity, we separate the defender's decision problem from the (linear) attacker's problem, and perform a hill-climbing search for the former. This search, with multiple restarts, is given in Algorithm 1.

The variables $overallMax$, $currentMax$, and $solution$ in Algorithm 1 are all tuples, specifically pairs, where the second component is the defender's policy, and the first component is its value. For instance, $overallMax$ is initialized to the empty policy with a value of $-\infty$, in Line 1 of Algorithm 1. Subscripted notations in this algorithm, such as $currentMax_1$ in Line 9, refer to the first (value) component of the tuple. The function $evaluate(\zeta)$ returns the objective value of the defender, $\sum_{\ell,o} X^*(\ell, o) R_d(\ell, o)$, where X^* are computed by solving the attacker's MILP for the given defender's policy ζ, as described above in this section. Algorithm 2 specifies one way to generate a limited set of neighboring distributions of a given ζ, with only one dimension perturbed (by δ) at a time. Thus the parameter $\delta \in (0, 1]$ controls the distance between a given ζ and its neighbors. Note that the evaluation of neighbors can be executed in parallel. The function $randomPolicy()$ returns a random ζ and solution value, as determined by $evaluate(\zeta)$.

Algorithm 2 GETNEIGHBORS(ζ, δ)

```
$neighbors \leftarrow \emptyset$
$indices \leftarrow T_d(1) \times T_d(2) \times \dots \times T_d(|O|)$
for $(i_1, i_2, \dots, i_{|O|}) \in indices$ do
    $\zeta' \leftarrow \zeta$
    for $o \in |O|$ do
        $\zeta'(o, i_o) \leftarrow \zeta'(o, i_o) + \delta$
        for $o' \in |O|$ do
            $\zeta' \leftarrow \frac{\zeta'(o,o')}{1+\delta}$
        end for
    end for
    $neighbors \leftarrow neighbors \cup \{\zeta'\}$
end for
Return $neighbors$
```

6 Tinted enclosure

We also consider a variant of the surveillance problem where the attacker is unable to observe the current orientation of the defender, perhaps because the camera is contained in a tinted enclosure. This requires the redefinition of $v(\ell, o, \ell')$ as $v(\ell, \ell')$ and simplifies its expression as

$$v(\ell, \ell') = \sum_{o \in O} R_a(\ell, o) X(\ell, o) + \gamma \max_{\ell''} v(\ell', \ell'')$$

Furthermore, $maxV(\ell, o)$ is redefined as $maxV(\ell)$, and the attacker's policy as $\sigma(\ell, \ell')$. Despite these, the key constraint 6 remains cubic, and the above approaches are still applicable. However, the handicap of the attacker means that the defender's policy values can be expected to improve in this setting.

7 Experiments

In the first batch of experiments, we evaluated the linear approximation (ASAP) and Policy Search, over 9 sets of random camera surveillance (non-tinted) problems, the sets defined by increasing number of waypoint vertices of attacker locations, $|L|$, ranging from 2 to 10. Each set contains 20 random problem instances defined by a random edge set in the waypoint graph (single component), and random functions R_a, R_d. In all instances, we restricted $|T_a(\ell)|, |T_d(o)|$ to 3.

For comparison, we also evaluated two standard NLP solvers, Bonmin and SNOPT, and used the (unapproximated) MI-NLP formulation given earlier. Bonmin is a full-fledged MI-NLP solver. SNOPT, on the other hand, treats integer variables as continuous variables when solving MI-NLPs. All solvers except for SNOPT were run on cluster nodes with 12 processors (a mix of Intel Xeon X5570, X5670, and AMD Opteron processors with up to 12 cores each) and solvers were allowed to use all cores for threading/multiprocessing; however, Bonmin does not support multithreading. SNOPT was run on the NEOS Server [9], and its memory usage was not available.

The Policy Search solver was configured with $\delta = 0.01$ and $restarts = 4$, and the Linear Approximation solver was configured with 25 snap points (i.e. $|S| = 25$). Both of these approaches require MILP solvers, for which we used IBM's ILOG CPLEX. Since the Linear Approximation solver requires solution of a single MILP, we allocated 12 threads to a single CPLEX instance for this solver. On the other hand, since the Policy Search solver requires solution of numerous (but less difficult) MILPs, we allowed for 6 instances of CPLEX to run simultaneously, using 2 threads each. For each problem instance, we measured the expected value of the defender's policy produced by each solver, the amount of wall time (in seconds) required by each solver to find a solution, and the maximum amount of RAM used (in megabytes).

Figure 3 shows the average defender's objective value, runtimes and memory usage for the first 3 sets, $|L| = 2, 3, 4$. The performances of Bonmin and SNOPT clearly show the inadequacy of general purpose MI-NLP solvers for the camera surveillance problem. While the growth in the time and memory requirements of the Policy Search solver is orders of magnitude smaller than Bonmin, it produces higher quality solutions than Bonmin. SNOPT, on the other hand, is more efficient but it produces extremely poor solutions. The performance of the (ASAP based) Linear Approximation solver is clearly dominated by the Policy Search solver, hence we selected the latter for further evaluation. Furthermore, the time and memory requirements of both the Linear Approximation solver and Bonmin were prohibitively high for $|L| > 4$, so we did not run those experiments. For the problem sets $|L| = 5 \ldots 10$, we only compare Policy Search with SNOPT.

Figures 4 (left and middle) show the defender's value and computation times of Policy Search and SNOPT on the remaining problem sets. It is interesting that the computation times of SNOPT remains fairly constant. However the solutions continue to be extremely poor compared to Policy Search. SNOPT performs a general purpose approximation by treating Boolean variables as real. This, apart from its handling of non-linear constraints, is less suited than the more targetted MILP approximation performed by Policy Search.

The second batch of experiments reported in Figure 4 (right) shows the comparative policy values produced by Policy Search on the original (labeled "PS Visible") and the tinted (labeled "PS Tinted") variants of the surveillance problem. As expected, the average defender's policy value improves in the tinted variant and Policy Search is able to significantly pick up this improvement for $|L| > 2$. For comparison, the plot also shows the defender values when ζ is held fixed at the uniform random policy, verifying that Policy Search does produce non-trivial solutions.

8 Related work

Most solution systems for targetted applications of security games, such as ARMOR (LAX airport security [5]), IRIS (federal air marshal service [8]) and GUARDS (transportation security administration [6]) formulate the resource allocation/scheduling problems as (mixed-integer) linear programming problems. Both the defender and the attacker's decision problems are formulated as linear programs. In the camera surveillance problem, although the attacker's decision problem is indeed a mixed integer linear program, the defender's decision problem becomes a non-linear program. Our problem formulation shares many key characteristics of security games, including multiple types of attackers, non-zero-sum utilities, and randomized optimal strategy for the defender, as discussed before.

The attacker's decision problem in this article, that of *stealth navigation*, has been addressed separately before. For instance, [1] considers the navigation of a robot through a field of dynamic obstacles (e.g., beams of search light) without colliding with any. They present a polynomial time algorithm for the case that the robot can move faster than any obstacle. In our time-discretized setting, however, the defender's view and the attacker's location can vary at every time step, hence at the same speed. Therefore, it is likely that no polynomial solution exists even for our attacker's decision problem. More recently, [3] has looked into stealth navigation of a defender, sneaking up on an invader, in order to be detected by the invader as late as possible, ideally no earlier than capture time. They apply heuristic approaches to determine the defender's plan based on a predetermined roadmap (waypoint graph). While we also use a waypoint graph to represent the decision problems, our objective and methodologies are very different.

The defender's decision problem has often been addressed in a distributed constrained optimization setting, such as a *sensor network*, where multiple fixed sensor nodes must coordinate to track (potentially multiple) targets [2, 10]. The sensors are assumed to have significant computational capabilities. By contrast, we consider a single defender and (mobile) attacker, and our approach is readily implementable on existing pan-tilt-zoom cameras since the computation occurs offline. Furthermore, with multiple defenders (cameras), it may become difficult for the attacker to observe their (joint) policies, therefore the attacker's policies may need to be represented in a potentially less interesting way for this application. However, with an appropriate formulation and improved approximation, it may be possible to extend the camera surveillance problem to multiple cameras (defenders) in the future.

Our take on the camera surveillance problem appears unique, and we are unaware of any existing study on the

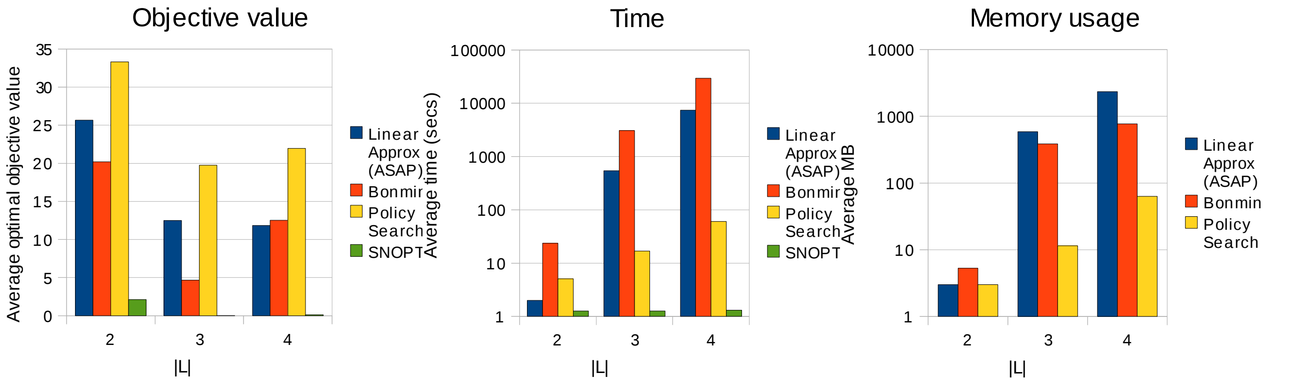


Figure 3: Plots of average (over 20 instances) metrics over the smallest 3 problem sets. (a) shows the defender's objective value only.

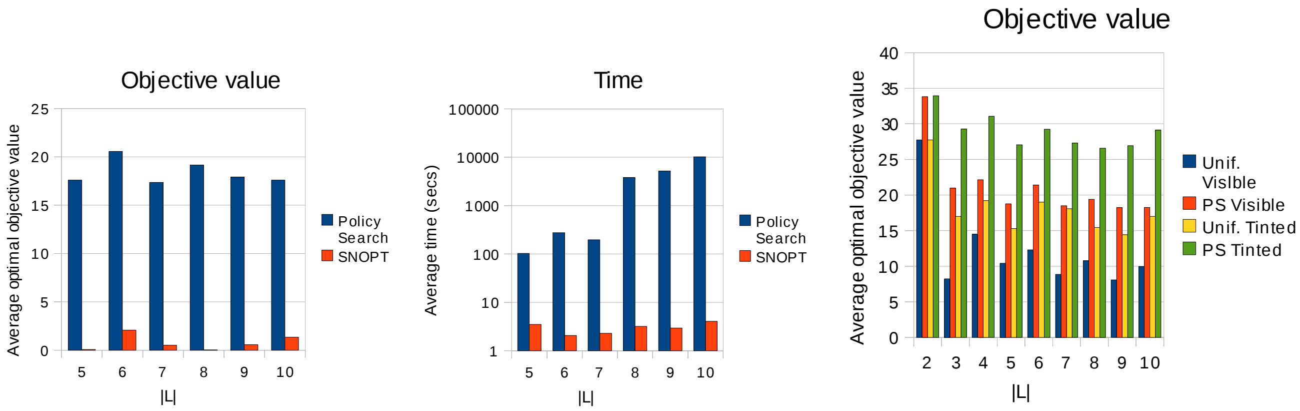


Figure 4: Left & Middle: Comparative solution quality and runtimes of Policy Search and SNOPT. Right: Comparative solution quality of Policy Search on Visible and Tinted formulations, also showing the quality of uniform random defender policy in both formulations.

Stackelberg view of this problem.

9 Conclusions and future work

We have formulated the tradeoff between the amount of collected data and the coverage of a surveillance camera, as a security game between a defender and an (indefinite) attacker. We have shown that this yields a mixed integer non-linear program, on which standard MI-NLP solvers are either prohibitively inefficient or produce poor policies. We have presented two simple approximate solutions: a linear approximation based on an existing aproach (ASAP), and a policy search (hill climbing) approach that leverages the segregation of the defender's decision problem from the attacker's. We have shown experimentally that the policy search approach is more scalable and produces higher quality solutions.

Although we were able to solve instances with up to 10 nodes in the waypoint graph, scaling up to larger and more complex waypoint graphs remains a challenge. One standard technique would be to reduce MILPs into linear programs by assuming that the Boolean variables can be fractional. While this version would be able to scale much better than our current PolicySearch algorithm, it would produce an approximation whose error-bound is unknown at this time. This is a potential avenue for future exploration.

This article lays the foundation for the development of more competent solution approaches in the future, particularly better quality approximations. For instance, a potential avenue for future work is to determine the conditions under which the camera surveillance security game can be posed as a semi-definite program (SDP). It would be interesting to investigate whether standard interior point SDP solvers can produce higher quality solutions more efficiently than Policy Search.

Acknowledgment

The authors thank the anonymous reviewers for helpful comments and suggestions. This work was funded in part by the U.S. Army under grant #W911NF-11-1-0124.

References

[1] K. Fujimura and H. Samet. Planning a time-minimal motion among moving obstacles. *Algorithmica*, 10:41–63, 1993.

[2] V. Lesser, C. Ortiz, and M. Tambe, editors. *Distributed Sensor Networks: A multi-agent perspective*. Kluwer, 2003.

[3] J. Park, J.S. Choi, J. Kim, S. Ji, and B.H. Lee. Long-term stealth navigation in a security zone where the movement of the invader is monitored. *International Journal of Control, Automation and Systems*, 8(3):604–614, 2010.

[4] P. Paruchuri, J.P. Pearce, M. Tambe, F. Ordonez, and S. Kraus. An efficient heuristic approach for security against multiple adversaries. In *Proc. 6th Intl. AAMAS Conference*, 2007.

[5] J. Pita, M. Jain, J. Marecki, F. Ordonez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus. Deployed ARMOR protection: The application of a game-theoretic model for security at the Los Angeles International Airport. In *Proc. 7th Intl. AAMAS Conference*, 2008.

[6] J. Pita, M. Tambe, C. Kiekintveld, S. Cullen, and E. Steigerwald. GUARDS: Game theoretic security allocation on a national scale. In *Proc. 10th Intl. AAMAS Conference*, 2011.

[7] M. Tambe. *Security and Game Theory*. Cambridge University Press, 2012.

[8] J. Tsai, S. Rathi, C. Kiekintveld, F. Ordonez, and M. Tambe. IRIS-A tool for strategic security allocation in transportation networks. In *Proc. 8th Intl. AAMAS Conference*, 2009.

[9] Website. NEOS Server: State-of-the-art solvers for numerical optimization. http://www.neos-server.org/neos/.

[10] R. Zivan, R. Glinton, and K. Sycara. Distributed constraint optimization for large teams of mobile sensing agents. In *Proc. International Conference on Intelligent Agent Technology (IAT)*, pages 347–354, 2009.

Evolutionary Multiobjective Optimization Based on Gaussian Process Modeling

Miha Mlakar
Department of Intelligent Systems, Jožef Stefan Institute
Jamova 39, Ljubljana, Slovenia

Keywords: multiobjective optimization, evolutionary algorithms, surrogate models, relations under uncertainty

Received: October 16, 2015

This paper presents a summary of the doctoral dissertation of the author, which addresses the task of evolutionary multiobjective optimization using surrogate models.

Povzetek: Prispevek predstavlja povzetek doktorske disertacije avtorja, ki obravnava večkriterijsko evolucijsko optimizacijo z uporabo nadomestnih modelov.

1 Introduction

The optimization problems where simultaneous optimization of multiple, often conflicting criteria (or objectives) is needed, are present in everyday life and can be found in various fields. One of the most effective ways of solving multiobjective optimization problems is to use multiobjective evolutionary algorithms (MOEAs) [1]. MOEAs are population-based algorithms that draw inspiration from optimization processes that occur in nature. In order to find a Pareto-optimal set, a lot of different solutions have to be assessed (evaluated) during the optimization process. For some optimization problems these solution evaluations can be computationally expensive and a single solution evaluation takes a lot of time. In order to obtain the results of such an optimization problem more quickly (or obtain them in a reasonable amount of time), we can use surrogate models in the optimization process. A surrogate model is constructed based on modeling the response of the simulator from a limited number of previously exactly evaluated solutions. Then instead of using a time-consuming exact evaluation, a solution can be approximated with the surrogate model. Since an individual solution approximation is (much) faster, the whole optimization process can be accelerated.

However, the use of the surrogate models in optimization can also have a drawback. If the optimization problem is very complex and therefore hard to model, the surrogate models can be imprecise and the solutions approximated with these models can be inaccurate. As a consequence, this can slow the optimization process or even prevent the algorithm from finding the best solutions.

When approximating solutions, some surrogate models, in addition to the approximated values, provide also a confidence interval of the approximation. This confidence interval indicates the region in which the exactly evaluated solution should appear. The confidence interval width indicates the certainty of the approximation. If the confidence interval is narrow, we can be more certain about the approximation and vice versa.

The solutions represented with confidence intervals, where exact objective values are unknown, are called solutions under uncertainty. Since the uncertainty offers additional information, it can be effectively used when comparing solutions. In the doctoral dissertation [2], we defined new relations under uncertainty that consider also the confidence intervals to reduce the possibility of incorrect comparisons due to the inaccurate approximations. The relations under uncertainty are described in more details in Section 2.

Based on the relations under uncertainty, we proposed a new surrogate-model-based multiobjective evolutionary algorithm called Differential Evolution for Multiobjective Optimization based on Gaussian Process modeling (GP-DEMO). The algorithm and its comparison to other algorithms is described in Chapter 3.

2 Relations for comparing solutions under uncertainty

If the solutions are represented with the approximated values and confidence intervals for each approximation, the standard Pareto dominance relations are not suitable and must be adapted to accommodate the uncertainty. In order to be able to compare the solutions represented in this way, the relations between the solutions under uncertainty are defined on the bounding boxes (BBs) of their objective values [3]. From the vectors of the approximated values ($\boldsymbol{z}$) and the confidence intervals ($\boldsymbol{\varepsilon}$), the bounding box is designed as shown in Figure 1.

The relations under uncertainty were defined for constrained and unconstrained optimization problems and they cover all possible cases that can occur when comparing solutions under uncertainty.

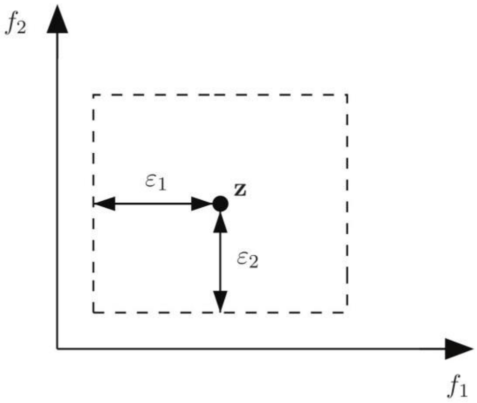


Figure 1: The bounding box of an objective vector.

To check, if the use of the proposed relations under uncertainty reduces the possibility of incorrect comparisons due to the inaccurate approximations, we compare them with Pareto dominance relations. The comparison was performed with various surrogate models and on different benchmark problems. The results shower that relations under uncertainty reduce the possibility of incorrect comparisons.

3 The GP-DEMO algorithm

As shown in the previous chapter, the use of new relations under uncertainty reduces the possibility of incorrect comparisons of inaccurately approximated solutions. So we decided to design a new surrogate-model-based optimization algorithm called GP-DEMO with relations under uncertainty used for comparing solutions [4]. This algorithm is an extension of the Differential Evolution for Multiobjective Optimization (DEMO) algorithm [5], which uses differential evolution to effectively solve numerical multiobjective optimization problems. As a surrogate model Gaussian Process modeling is employed to find approximate solution values together with their confidence intervals. As GP-DEMO works on the same principles as DEMO, the quality of its results is expected to be similar to the results of DEMO, but with fewer exact solution evaluations. To thoroughly test the GP-DEMO algorithm, we compare it with another surrogate-model-based algorithm (GEC) and with DEMO on several benchmark and two real-world optimization problems.

The results showed that GP-DEMO and DEMO obtain similar results, but GP-DEMO needs less exact evaluations and thus has shorter optimization time on real-world problems. In comparison to GEC, GP-DEMO obtained better results. Which algorithm needs less exact evaluations depends on the type of optimization problem.

In order to determine when to use GP-DEMO instead of DEMO, we calculate border times. The border time for a specific optimization problem tells us how long a single exact solution evaluation should last, in order for the optimization times of GP-DEMO and DEMO to be equal. Depending on the appraised complexity of the problem, the border time can be estimated. Therefore, if a single exact solution evaluation takes more than the estimated border time and the quality of the results is important, we can conclude that for this problem GP-DEMO is a more appropriate choice than DEMO.

4 Conclusion

This paper summarized the doctoral dissertation [2] and presents its main ideas and findings. We defined new relations for comparing solutions under uncertainty and confirmed that they reduce the possibility of incorrect comparisons due to the inaccurate approximations.

We then used these relations in the new surrogate-model-based multiobjective evolutionary algorithm GP-DEMO. We thoroughly tested the algorithm and the results show that GP-DEMO in comparison to other MOEAs produces comparable results with fewer exact evaluations of the original objective functions.

References

[1] Deb, K.. Multi-objective optimization using evolutionary algorithms. Wiley, New York, 2001.

[2] Mlakar, M. Evolutionary multiobjective optimization based on Gaussian process modeling, PhD Thesis, IPS Jožef Stefan, Ljubljana, Slovenia, 2015.

[3] Mlakar, M., Tušar, T., Filipič, B. Comparing solutions under uncertainty in multiobjective optimization. Mathematical Problems in Engineering, 2014. doi:10 .1155/2014/817964.

[4] Mlakar, M., Petelin, D., Tušar, T., Filipič, B. GP-DEMO: Differential evolution for multiobjective optimization based on Gaussian process models. European Journal of Operational Research, 2015, 243 (2), 347–361.

[5] Robič T., Filipič B., DEMO: Differential evolution for multiobjective optimization, in Proc. of 3rd Int. Conf. Evol. Multi-Criterion Optimization – EMO 2005, pp. 520–533.

Design and Implementation of Advanced Bayesian Networks with Comparative Probability

Ali Hilal Ali
Faculty of Engineering, University of Kufa, Iraq
E-mail: alih.alathari@uokufa.edu.iq

Keywords: Bayesian networks, comparative probability, aviation safety, decision support systems, MIMIC II database
Received: November 28, 2015

This paper summarizes the major findings, methods, and background theories of the doctoral thesis in [1]. The aim of the thesis has been to enhance the current procedures of designing decision support systems (DSSs) used by decision-makers to comprehend the current situation better in cases where the available amount of information required to make an informed decision is limited. The research resulted in a new innovated theory that combines the philosophical comparative approach to probability, the frequency interpretation of probability, dynamic Bayesian networks and the expected utility theory. It enables engineers to write self-learning algorithms that use example of behaviours to model situations, evaluate and make decisions, diagnose problems, and/or find the most probable consequences in real-time. The new theory was particularly applied to the problems of validating equipment readings in an aircraft, flight data analysis, prediction of passengers behaviours, and real-time monitoring and prediction of patients' states in intensive care units (ICU). The algorithm was able to pinpoint the faulty equipment from between a group of equipment giving false fault indications, an important improvement over the current fault detection procedures. On the ICU application side, the algorithm was able to predict those patients with high mortality risk about 24 hours before they actually deceased.

Povzetek: Predstavljena je disertacija, ki izboljšuje bayesovske mreže s pomočjo primerjalnih verjetnosti.

1 Introduction

We live in an ever-changing world where our convictions about the state of it update with time as we discover new information about our surroundings. As we acknowledge the imperfections of our knowledge repositories regarding the state of the world, we often need to make decisions despite all the missing details and the uncertainty of where our decisions might lead us to. A robot might use its sensory system, for instance, a sonar based sensor, to retrieve cues about its surroundings. Then it might use these cues to decide on which direction is best to turn to. Since the world behind the range of the robot's sensors is unknown, the robot may take a turn that leads to a dead end. Hence, the robot needs to make a decision in an environment where the only available information is that of its immediate surroundings. Even if the robot was in an exceptionally charted environment, its sensors might malfunction or degrade. In this case, the uncertainty arises not from the environment but rather from a lack of trustworthiness of the robot's sensors. In addition, the robot programming may contain bugs, the robot might trip and fall, or its battery may run out of power or be stolen. The list of events that the robot could possibly face in an environment grows infinitely as we consider more details. The problem of specifying all the exceptions a designer needs to consider is called the qualification problem [2]. Probability theory is the main tool used to represent uncertainty arising from laziness and ignorance [1, p 482]. If we consider probability as a measure of how likely an event would be observed in an experiment repeated a certain amount of times, then it could be used as a quantitative representation of our certainty of how likely that event might occur from among all other possible events. In this context, probability is interpreted as a degree of belief rather than a frequency of occurrence.

2 Research Approach

In order to meet the objectives of this thesis, both theoretical and practical approaches were adopted. Firstly, the common approach to decision-making, and in turn knowledge-based decision support systems, is to use probability theory backed by the utility functions to come up with the expected utility of making a decision.

Secondly, as the theory of probability is accepted as the main framework for representing knowledge with uncertainty, many interpretations of probability have been analyzed in order to find the most suitable one that works with as little information available as possible without falling back on a strictly analytical approach or ignorance.

Thirdly, we surveyed the research done in the theory of comparative probability, its axioms, and application to computer science.

3 Results and discussion

We used the Chernoff bounds to come up with a novel approach to updating probability bounds between successive experiment results. Chernoff bounds were used as upper and lower estimates of probability at a given experiment while taking into account all the previous experiment results. As the number of experiments increases, the gap between the upper and lower bounds becomes smaller until it approaches the expectation of the outcome of the experiment. The expectation of an experiment is nothing other than its probability. Hence, a mathematical foundation between Comparative probability (CP) and Kolmogorov probability (KP) was established with a dynamic nature that puts CP as a foreground methodology to evaluate KP.

In addition, we recognized that even with the availability of a simple approach to representing knowledge, the size of the joint probability tables may become too large to process, so we used a Bayesian network to simplify the processing of probabilistic queries and reduced the amount of mathematical backgrounds required to answer them. as probabilistic decision support systems work on averages, it would be unfeasible to attempt to justify the principles of the proposed approach using an example or two. Instead, we adopted two approaches to tackle the issue. Firstly, we used scenario-based validation. Scenarios are ways of generating test data, which can be used to validate system design requirements. The second approach was the ability of the system to predict an output with high accuracy.

Two new enhancements have been suggested to the detection and isolation of faults in aviation and to the optimising the navigation planning. In the first experiment, we proposed a new method for detecting faults that should overcome any limitations that result from using majority vote coming from primary and redundant systems. Whereas, in the second experiment, we proposed a novel application to the BADA database as a DSS for navigation planning. Both experiments where implemented with CP to show the usefulness, admissibility and ascertainability of CP.

Finally, An innovated ICU patient monitoring system was designed. The novel system outperforms all current monitoring systems in terms of its versatility and prediction capabilities. We have shown how it can be used to predict the evolution of patients' physiological parameters over time and how it can predict the mortality risk in patients with a history of cardiac surgery even 24 hours before patients' date of death.

These findings show the development of the reasoning according to which this research was conducted, starting from defining the research question to documenting the results. The research method dictates that a good theory should be able to predict some observations that can be measured and compared to what the theory proposes. In the light of such requirements, it is the belief of the author that the thesis stands on very solid grounds with respect both to meeting the objectives and verifying the soundness of its theory.

4 Final remarks

As the case with any novel proposal, the comparative probability approach proposed in this thesis work summarized here is not yet complete. The best way to show the power of it is through applying it to a wider range of applications and engineering problems while ironing out any issues that arises along the way. While this thesis worked as proof of concept for CP application to DSS and artificial intelligence in general, it is the belief of the author that it has achieved its objectives and still maintaining the de facto interpretation of probability intact. After all, it would not be of benefit to the scientific community to propose the seizure of their very best guide to life.

References

[1] Ali Hilal Ali, "Design and Implementation of Advanced Bayesian Networks with Comparative Probability" Ph.D. dissertation, Lancaster University, 2012.

[2] Russel, S., and Norving, P. "Artificial Intelligence: A Modern Approach". New Jersey: Pearson Education, Inc., 2010.

CONTENTS OF *Informatica* Volume 39 (2015) pp. 1–465

Papers

Ljubešić, N. & , D. Kranjčić. 2015. Discriminating Between Closely Related Languages on Twitter. Informatica 39:1–8.

Ljubešić, N. & , K. Dobrovoljc, D. Fišer. 2015. *MWELex – MWE Lexica of Croatian, Slovene and Serbian Extracted from Parsed Corpora. Informatica 39:293–300.

Ma, K. & , B. Yang. 2015. Asynchronous Adaptive Delay Tolerant Index Cache Using In-memory Delta Cell. Informatica 39:443–449.

Mist, J.J. & , S.J. Gibson, C.J. Solomon. 2015. Comparing Evolutionary Operators, Search Spaces, and Evolutionary Algorithms in the Construction of Facial Composites. Informatica 39:135–145.

Mlakar, M. & . 2015. Evolutionary Multiobjective Optimization Based on Gaussian Process Modeling. Informatica 39:459–460.

Mongus, D. & , B. Žalik. 2015. Detection of Ground in Point-clouds Generated from Stereo-pair Images. Informatica 39:271–275.

Patvardhan, C. & , V.P. Prakash, A. Srivastav. 2015. Fast Heuristics for Large Instances of the Euclidean Bounded Diameter Minimum Spanning Tree Problem. Informatica 39:281–292.

Piltaver, R. & , B. Cvetković, B. Kaluža. 2015. Denoising Human-Motion Trajectories Captured with Ultra-Wideband Real-time Location System. Informatica 39:311–322.

Pucihar, K.Č. & . 2015. Designing Effective Mobile Augmented Reality Interactions. Informatica 39:333–334.

Rogelj, P. & , M. Baraković. 2015. Cervix Cancer Spatial Modelling for Brachytherapy Applicator Analysis. Informatica 39:261–269.

Šilc, J. & , A. Zamuda. 2015. Editors' Introduction to the Special Issue on "Bioinspired Optimization" . Informatica 39:103–104.

Šilc, J. & , K. Taškova, P. Korošec. 2015. Data Mining-Assisted Parameter Tuning of a Search Algoritm. Informatica 39:169–176.

Šnajder, J. & , P. Almić. 2015. Modeling Semantic Compositionality of Croatian Multiword Expressions. Informatica 39:301–310.

Taenaka, Y. & , M. Tagawa, K. Tsukamoto. 2015. An Experimental Approach to Examine a Multi-Channel Multi-Hop Wireless Backbone Network. Informatica 39:365–374.

Tomašič, P. & , G. Papa, M. Žnidaršič. 2015. Using a Genetic Algorithm to Produce Slogans. Informatica 39:125–133.

Verbič, M. & , M. Čok, T. Turk. 2015. An Exact Analytical Grossing-Up Algorithm for Tax-Benefit Models. Informatica 39:23–34.

Wang, C. & , X. Xu, D. Shi, J. Fang. 2015. Privacy-preserving Cloud-based Personal Health Record System Using Attribute-based Encryption and Anonymous Multi-Receiver Identity-based Encryption. Informatica 39:375–382.

Wiesberg, S. & , G. Reinelt. 2015. Relaxations in Practical Clustering and Blockmodeling. Informatica 39:249–256.

Xhafa, F. & , J. Li, V. Kolici. 2015. Editors' Introduction to the Special Issue on "Advances in Secure Data Streaming Systems" . Informatica 39:337–337.

Yang, W. & , C. Zhang, B. Mu. 2015. Data-intensive Service Mashup Based on Game Theory and Hybrid Fireworks Optimization Algorithm in the Cloud. Informatica 39:421–429.

Zamuda, A. & , U. Mlakar. 2015. Differential Evolution Control Parameters Study for Self-Adaptive Triangular Brushstrokes. Informatica 39:105–113.

Zhang, G. & , C. Hu, N. Wang, X. Wei, C. Xing. 2015. A Novel Scheme for Improving Quality of Service of Live Streaming. Informatica 39:339–346.

Zupanc, K. & , Z. Bosnić. 2015. Advances in the Field of Automated Essay Evaluation. Informatica 39:383–395.

JOŽEF STEFAN INSTITUTE

Jožef Stefan (1835-1893) was one of the most prominent physicists of the 19th century. Born to Slovene parents, he obtained his Ph.D. at Vienna University, where he was later Director of the Physics Institute, Vice-President of the Vienna Academy of Sciences and a member of several scientific institutions in Europe. Stefan explored many areas in hydrodynamics, optics, acoustics, electricity, magnetism and the kinetic theory of gases. Among other things, he originated the law that the total radiation from a black body is proportional to the 4th power of its absolute temperature, known as the Stefan–Boltzmann law.

The Jožef Stefan Institute (JSI) is the leading independent scientific research institution in Slovenia, covering a broad spectrum of fundamental and applied research in the fields of physics, chemistry and biochemistry, electronics and information science, nuclear science technology, energy research and environmental science.

The Jožef Stefan Institute (JSI) is a research organisation for pure and applied research in the natural sciences and technology. Both are closely interconnected in research departments composed of different task teams. Emphasis in basic research is given to the development and education of young scientists, while applied research and development serve for the transfer of advanced knowledge, contributing to the development of the national economy and society in general.

At present the Institute, with a total of about 900 staff, has 700 researchers, about 250 of whom are postgraduates, around 500 of whom have doctorates (Ph.D.), and around 200 of whom have permanent professorships or temporary teaching assignments at the Universities.

In view of its activities and status, the JSI plays the role of a national institute, complementing the role of the universities and bridging the gap between basic science and applications.

Research at the JSI includes the following major fields: physics; chemistry; electronics, informatics and computer sciences; biochemistry; ecology; reactor technology; applied mathematics. Most of the activities are more or less closely connected to information sciences, in particular computer sciences, artificial intelligence, language and speech technologies, computer-aided design, computer architectures, biocybernetics and robotics, computer automation and control, professional electronics, digital communications and networks, and applied mathematics.

The Institute is located in Ljubljana, the capital of the independent state of **Slove**nia (or S♡nia). The capital today is considered a crossroad between East, West and Mediterranean Europe, offering excellent productive capabilities and solid business opportunities, with strong international connections. Ljubljana is connected to important centers such as Prague, Budapest, Vienna, Zagreb, Milan, Rome, Monaco, Nice, Bern and Munich, all within a radius of 600 km.

From the Jožef Stefan Institute, the Technology park “Ljubljana” has been proposed as part of the national strategy for technological development to foster synergies between research and industry, to promote joint ventures between university bodies, research institutes and innovative industry, to act as an incubator for high-tech initiatives and to accelerate the development cycle of innovative products.

Part of the Institute was reorganized into several high-tech units supported by and connected within the Technology park at the Jožef Stefan Institute, established as the beginning of a regional Technology park "Ljubljana". The project was developed at a particularly historical moment, characterized by the process of state reorganisation, privatisation and private initiative. The national Technology Park is a shareholding company hosting an independent venture-capital institution.

The promoters and operational entities of the project are the Republic of Slovenia, Ministry of Higher Education, Science and Technology and the Jožef Stefan Institute. The framework of the operation also includes the University of Ljubljana, the National Institute of Chemistry, the Institute for Electronics and Vacuum Technology and the Institute for Materials and Construction Research among others. In addition, the project is supported by the Ministry of the Economy, the National Chamber of Economy and the City of Ljubljana.

Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Tel.:+386 1 4773 900, Fax.:+386 1 251 93 85
WWW: http://www.ijs.si
E-mail: matjaz.gams@ijs.si
Public relations: Polona Strnad

INFORMATICA
AN INTERNATIONAL JOURNAL OF COMPUTING AND INFORMATICS
INVITATION, COOPERATION

Submissions and Refereeing

Please submit a manuscript to: http://www.informatica.si/Editors/PaperUpload.asp. At least two referees outside the author's country will examine it, and they are invited to make as many remarks as possible from typing errors to global philosophical disagreements. The chosen editor will send the author the obtained reviews. If the paper is accepted, the editor will also send an email to the managing editor. The executive board will inform the author that the paper has been accepted, and the author will send the paper to the managing editor. The paper will be published within one year of receipt of email with the text in Informatica MS Word format or Informatica LaTeX format and figures in .eps format. Style and examples of papers can be obtained from http://www.informatica.si. Opinions, news, calls for conferences, calls for papers, etc. should be sent directly to the managing editor.

QUESTIONNAIRE

□ Send Informatica free of charge

□ Yes, we subscribe

Please, complete the order form and send it to Dr. Drago Torkar, Informatica, Institut Jožef Stefan, Jamova 39, 1000 Ljubljana, Slovenia. E-mail: drago.torkar@ijs.si

Since 1977, Informatica has been a major Slovenian scientific journal of computing and informatics, including telecommunications, automation and other related areas. In its 16th year (more than twentyone years ago) it became truly international, although it still remains connected to Central Europe. The basic aim of Informatica is to impose intellectual values (science, engineering) in a distributed organisation.

Informatica is a journal primarily covering intelligent systems in the European computer science, informatics and cognitive community; scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the Refereeing Board.

Informatica is free of charge for major scientific, educational and governmental institutions. Others should subscribe (see the last page of Informatica).

ORDER FORM – INFORMATICA

Name: . . . .

Title and Profession (optional): . . . .

. . . .

Home Address and Telephone (optional): . . . .

. . . .

Office Address and Telephone (optional): . . . .

. . . .

E-mail Address (optional): . . . .

Signature and Date: . . . .

Informatica WWW:

http://www.informatica.si/

Referees from 2008 on:

A. Abraham, S. Abraham, R. Accornero, A. Adhikari, R. Ahmad, G. Alvarez, N. Anciaux, R. Arora, I. Awan, J. Azimi, C. Badica, Z. Balogh, S. Banerjee, G. Barbier, A. Baruzzo, B. Batagelj, T. Beaubouef, N. Beaulieu, M. ter Beek, P. Bellavista, K. Bilal, S. Bishop, J. Bodlaj, M. Bohanec, D. Bolme, Z. Bonikowski, B. Bošković, M. Botta, P. Brazdil, J. Brest, J. Brichau, A. Brodnik, D. Brown, I. Bruha, M. Bruynooghe, W. Buntine, D.D. Burdescu, J. Buys, X. Cai, Y. Cai, J.C. Cano, T. Cao, J.-V. Capella-Hernández, N. Carver, M. Cavazza, R. Ceylan, A. Chebotko, I. Chekalov, J. Chen, L.-M. Cheng, G. Chiola, Y.-C. Chiou, I. Chorbev, S.R. Choudhary, S.S.M. Chow, K.R. Chowdhury, V. Christlein, W. Chu, L. Chung, M. Ciglarič, J.-N. Colin, V. Cortellessa, J. Cui, P. Cui, Z. Cui, D. Cutting, A. Cuzzocrea, V. Cvjetkovic, J. Cypryjanski, L. Čehovin, D. Čerepnalkoski, I. Čosić, G. Daniele, G. Danoy, M. Dash, S. Datt, A. Datta, M.-Y. Day, F. Debili, C.J. Debono, J. Dedič, P. Degano, A. Dekdouk, H. Demirel, B. Demoen, S. Dendamrongvit, T. Deng, A. Derezinska, J. Dezert, G. Dias, I. Dimitrovski, S. Dobrišek, Q. Dou, J. Doumen, E. Dovgan, B. Dragovich, D. Drajic, O. Drbohlav, M. Drole, J. Dujmović, O. Ebers, J. Eder, S. Elaluf-Calderwood, E. Engström, U. riza Erturk, A. Farago, C. Fei, L. Feng, Y.X. Feng, B. Filipič, I. Fister, I. Fister Jr., D. Fišer, A. Flores, V.A. Fomichov, S. Forli, A. Freitas, J. Fridrich, S. Friedman, C. Fu, X. Fu, T. Fujimoto, G. Fung, S. Gabrielli, D. Galindo, A. Gambarara, M. Gams, M. Ganzha, J. Garbajosa, R. Gennari, G. Georgeson, N. Gligorić, S. Goel, G.H. Gonnet, D.S. Goodsell, S. Gordillo, J. Gore, M. Grčar, M. Grgurović, D. Grosse, Z.-H. Guan, D. Gubiani, M. Guid, C. Guo, B. Gupta, M. Gusev, M. Hahsler, Z. Haiping, A. Hameed, C. Hamzaçebi, Q.-L. Han, H. Hanping, T. Härder, J.N. Hatzopoulos, S. Hazelhurst, K. Hempstalk, J.M.G. Hidalgo, J. Hodgson, M. Holbl, M.P. Hong, G. Howells, M. Hu, J. Hyvärinen, D. Ienco, B. Ionescu, R. Irfan, N. Jaisankar, D. Jakobović, K. Jassem, I. Jawhar, Y. Jia, T. Jin, I. Jureta, Đ. Juričić, S. K, S. Kalajdziski, Y. Kalantidis, B. Kaluža, D. Kanellopoulos, R. Kapoor, D. Karapetyan, A. Kassler, D.S. Katz, A. Kaveh, S.U. Khan, M. Khattak, V. Khomenko, E.S. Khorasani, I. Kitanovski, D. Kocev, J. Kocijan, J. Kollár, A. Kontostathis, P. Korošec, A. Koschmider, D. Košir, J. Kovač, A. Krajnc, M. Krevs, J. Krogstie, P. Krsek, M. Kubat, M. Kukar, A. Kulis, A.P.S. Kumar, H. Kwaśnicka, W.K. Lai, C.-S. Laih, K.-Y. Lam, N. Landwehr, J. Lanir, A. Lavrov, M. Layouni, G. Leban, A. Lee, Y.-C. Lee, U. Legat, A. Leonardis, G. Li, G.-Z. Li, J. Li, X. Li, X. Li, Y. Li, Y. Li, S. Lian, L. Liao, C. Lim, J.-C. Lin, H. Liu, J. Liu, P. Liu, X. Liu, X. Liu, F. Logist, S. Loskovska, H. Lu, Z. Lu, X. Luo, M. Luštrek, I.V. Lyustig, S.A. Madani, M. Mahoney, S.U.R. Malik, Y. Marinakis, D. Marinčič, J. Marques-Silva, A. Martin, D. Marwede, M. Matijašević, T. Matsui, L. McMillan, A. McPherson, A. McPherson, Z. Meng, M.C. Mihaescu, V. Milea, N. Min-Allah, E. Minisci, V. Mišić, A.-H. Mogos, P. Mohapatra, D.D. Monica, A. Montanari, A. Moroni, J. Mosegaard, M. Moškon, L. de M. Mourelle, H. Moustafa, M. Možina, M. Mrak, Y. Mu, J. Mula, D. Nagamalai, M. Di Natale, A. Navarra, P. Navrat, N. Nedjah, R. Nejabati, W. Ng, Z. Ni, E.S. Nielsen, O. Nouali, F. Novak, B. Novikov, P. Nurmi, D. Obrul, B. Oliboni, X. Pan, M. Pančur, W. Pang, G. Papa, M. Paprzycki, M. Paralič, B.-K. Park, P. Patel, T.B. Pedersen, Z. Peng, R.G. Pensa, J. Perš, D. Petcu, B. Petelin, M. Petkovšek, D. Pevec, M. Pičulin, R. Piltaver, E. Pirogova, V. Podpečan, M. Polo, V. Pomponiu, E. Popescu, D. Poshyvanyk, B. Potočnik, R.J. Povinelli, S.R.M. Prasanna, K. Pripužić, G. Puppis, H. Qian, Y. Qian, L. Qiao, C. Qin, J. Que, J.-J. Quisquater, C. Rafe, S. Rahimi, V. Rajkovič, D. Raković, J. Ramaekers, J. Ramon, R. Ravnik, Y. Reddy, W. Reimche, H. Rezankova, D. Rispoli, B. Ristevski, B. Robič, J.A. Rodriguez-Aguilar, P. Rohatgi, W. Rossak, I. Rožanc, J. Rupnik, S.B. Sadkhan, K. Saeed, M. Saeki, K.S.M. Sahari, C. Sakharwade, E. Sakkopoulos, P. Sala, M.H. Samadzadeh, J.S. Sandhu, P. Scaglioso, V. Schau, W. Schempp, J. Seberry, A. Senanayake, M. Senobari, T.C. Seong, S. Shamala, c. shi, Z. Shi, L. Shiguo, N. Shilov, Z.-E.H. Slimane, F. Smith, H. Sneed, P. Sokolowski, T. Song, A. Soppera, A. Sorniotti, M. Stajdohar, L. Stanescu, D. Strnad, X. Sun, L. Šajn, R. Šenkeřík, M.R. Šikonja, J. Šilc, I. Škrjanc, T. Štajner, B. Šter, V. Štruc, H. Takizawa, C. Talcott, N. Tomasev, D. Torkar, S. Torrente, M. Trampuš, C. Tranoris, K. Trojacanec, M. Tschierschke, F. De Turck, J. Twycross, N. Tziritas, W. Vanhoof, P. Vateekul, L.A. Vese, A. Visconti, B. Vlaovič, V. Vojisavljević, M. Vozalis, P. Vračar, V. Vranić, C.-H. Wang, H. Wang, H. Wang, H. Wang, S. Wang, X.-F. Wang, X. Wang, Y. Wang, A. Wasilewska, S. Wenzel, V. Wickramasinghe, J. Wong, S. Wrobel, K. Wrona, B. Wu, L. Xiang, Y. Xiang, D. Xiao, F. Xie, L. Xie, Z. Xing, H. Yang, X. Yang, N.Y. Yen, C. Yong-Sheng, J.J. You, G. Yu, X. Zabulis, A. Zainal, A. Zamuda, M. Zand, Z. Zhang, Z. Zhao, D. Zheng, J. Zheng, X. Zheng, Z.-H. Zhou, F. Zhuang, A. Zimmermann, M.J. Zuo, B. Zupan, M. Zuqiang, B. Žalik, J. Žižka,

Informatica

An International Journal of Computing and Informatics

Web edition of Informatica may be accessed at: http://www.informatica.si.

Subscription Information Informatica (ISSN 0350-5596) is published four times a year in Spring, Summer, Autumn, and Winter (4 issues per year) by the Slovene Society Informatika, Litostrojska cesta 54, 1000 Ljubljana, Slovenia.
The subscription rate for 2015 (Volume 39) is
– 60 EUR for institutions,
– 30 EUR for individuals, and
– 15 EUR for students
Claims for missing issues will be honored free of charge within six months after the publication date of the issue.

Typesetting: Borut Žnidar.
Printing: ABO grafika d.o.o., Ob železnici 16, 1000 Ljubljana.

Orders may be placed by email (drago.torkar@ijs.si), telephone (+386 1 477 3900) or fax (+386 1 251 93 85). The payment should be made to our bank account no.: 02083-0013014662 at NLB d.d., 1520 Ljubljana, Trg republike 2, Slovenija, IBAN no.: SI56020830013014662, SWIFT Code: LJBASI2X.

Informatica is published by Slovene Society Informatika (president Niko Schlamberger) in cooperation with the following societies (and contact persons):
Robotics Society of Slovenia (Jadran Lenarčič)
Slovene Society for Pattern Recognition (Janez Perš)
Slovenian Artificial Intelligence Society (Dunja Mladenić)
Cognitive Science Society (Urban Kordeš)
Slovenian Society of Mathematicians, Physicists and Astronomers (Andrej Likar)
Automatic Control Society of Slovenia (Sašo Blažič)
Slovenian Association of Technical and Natural Sciences / Engineering Academy of Slovenia (Vojteh Leskovšek)
ACM Slovenia (Andrej Brodnik)

Informatica is financially supported by the Slovenian research agency from the Call for co-financing of scientific periodical publications.

Informatica is surveyed by: ACM Digital Library, Citeseer, COBISS, Compendex, Computer & Information Systems Abstracts, Computer Database, Computer Science Index, Current Mathematical Publications, DBLP Computer Science Bibliography, Directory of Open Access Journals, InfoTrac OneFile, Inspec, Linguistic and Language Behaviour Abstracts, Mathematical Reviews, MatSciNet, MatSci on SilverPlatter, Scopus, Zentralblatt Math

Volume 39 Number 4 December 2015 **ISSN 0350-5596**

Informatica

An International Journal of Computing and Informatics

End of Special Issue / Start of normal papers

Informatica **39** (2015) Number 4, pp. 337–465