Signal and Information Processing in Mobile Cloud Computing: Trends and Challenges

Chia-Yu Lin and Li-Chun Wang
Department of Electrical and Computer Engineering National Chiao Tung University, Taiwan
Email: sallylin0121@gmail.com; lichun@cc.nctu.edu.tw

Abstract —Mobile devices become popular with the help of hardware improvements and new functions supported by many sensors. In this paper, we propose a mobile and multi-sensing fusion platform to integrate the unstructured streaming sensing data collecting as well as processing technology and build a QoS (quality of service) performance model to estimate the computing resource of the platform. We also demonstrate three mobile and multi-sensing fusion applications as the examples on the platform. Besides, we discuss the trend and challenges of combining the mobile as well as multi-sensing fusion technology and signal and information processing in mobile cloud computing in great detail.

Index Terms—Mobile cloud computing, multi-sensing fusion technology

I. Introduction

Mobile devices consist of many sensors, which collect huge amount of data from each person. Sensors on mobile devices such as Global Positioning System (GPS) and camera sensors generate unstructured sensing data continuously. To analyze multiple source streaming sensing data, complex event processing technology which combines data from multiple sources to decide whether there are events or patterns is proposed [1]. However, streaming sensing data collecting and analyzing technology have not been integrated and implemented on a platform in the literature. Thus, we propose mobile and multi-sensing fusion platform which continuously collects multiple unstructured streaming sensing data, preprocesses data to specific programming type and integrates complex event processing technology to analyze streaming sensing data in real-time.

Fig. 1 shows the mobile and multi-sensing fusion platform architecture. We divide platform into three parts. The first part is clients such as smartphones and tablets which include many sensors. Clients generate continuously sensor data such as location data, preprocess, compress and send the data to the second part. The second part is cloud servers which use multi-sensing fusion technology to analyze the sensor data. Since sensor data are generated continuously and have to be analyzed in real-time, InfoSphere Streams computing platform [2]–[5] is adopted to analyze streaming sensor data on cloud servers. Furthermore, most of applications on mobile and multi-sensing fusion platform are real-time applications, so we design a quality of service ensured (QoS-ensured) performance model as the third part to analyze the performance and allocate the computing resource. Besides, Google App Engine [6] as well as Windows Azure [7] provide the computing platform as a service (PaaS) and Amazon Elastic Compute Cloud (Amazon EC2) [8] provide the computing infrastructure as a service (IaaS) for mobile and multi-sensing fusion platform. We also give three different workload multi-sensing applications includes education cloud, rescue cloud and business cloud to be the examples on the mobile and sensing fusion platform.

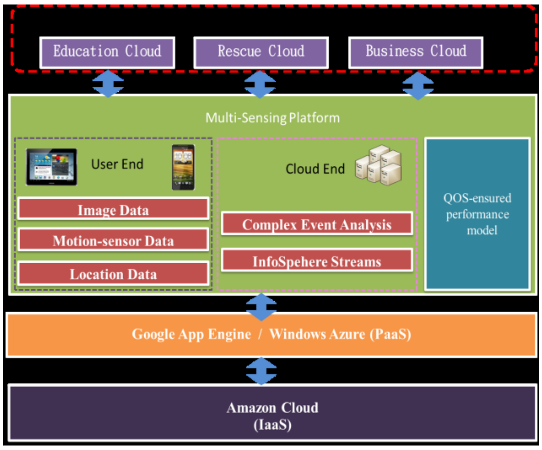

Fig. 1. System architecture of mobile and multi-sensing fusion platform.

In this paper, we not only propose mobile and multi-sensing fusion platform, but also point out the key challenges of mobile and multi-sensing research. Owing to sensing data are generated continuously, how to manage and process unstructured, intensive and streaming sensing data in real-time is the first challenge. The second challenge is QoS issue of streaming data analysis system. Streaming data analysis result should respond to users in real-time. Therefore, designing a queueing model to predict and adjust computing resource becomes critical. If we sent all the sensing data to cloud servers to analyze, we would cost a lot of bandwidths and time to transfer data. Thus, multi-sensing application partitioning between clients and cloud servers is the third challenge. The fourth challenge is data fault tolerance technologies of multi-sensing applications. Analyzing

Manuscript received March, 2015; revised September 3, 2015.
This work was supported Ministry of Science and Technology under Grant No. MOST 103-2221-E-009-015-MY2
Corresponding author email: lichun@g2.nctu.edu.tw
doi:10.12720/jcm.10.9.659-665

sensing data without storing technology is proposed to increase the performance of sensing data analysis. However, if the computation nodes are broken or failed, the sensing data could be lose. Therefore, how to replay continuously sensing data is important for multi-sensing applications.

The paper is organized as follows. Section II introduces mobile and multi-sensing fusion platform. Section III details mobile and multi-sensing fusion applications. Section IV discusses the challenges of multi-sensing fusion research. We conclude the paper in Section V.

II. Mobile and Multi-Sensing Fusion Platform

Clients, cloud servers and QOS-ensured performance model are three parts of mobile and multi-sensing fusion platform. Following are the details of every part.

Fig. 2. Technology overview of mobile and multi-sensing fusion platform.

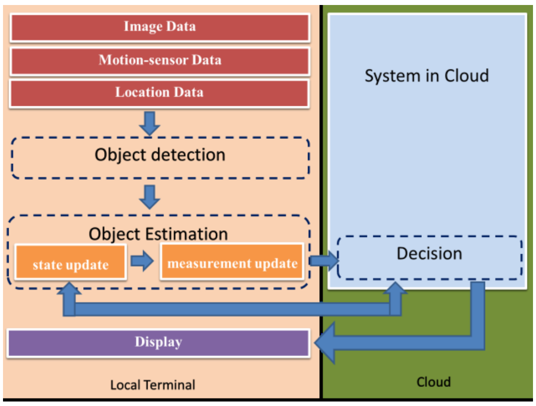

Fig. 3. Complex event processing flow.

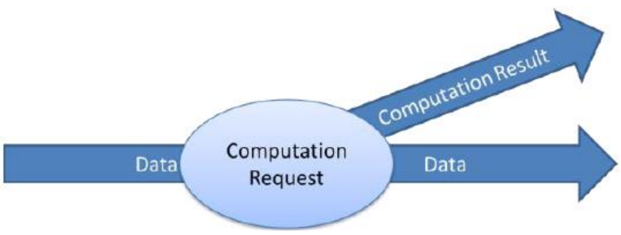

Fig. 4. InfoSphere streams computation model.

A. Clients

Mobile phones consist of many sensors, which can collect huge amount of data from each person. After sensing data are processed and analyzed, valuable information can be created for customers. Fig. 2 shows that visual, audio, motion, location sensors, etc. can identify local environments and the behavior as well as location of users, thereby providing information such as augmented reality or 3D maps. Sensing data are continuously generated by smart devices "seeing" and "interacting" with users. This is the object detection step of complex event processing as shown in Fig. 3. The second step of complex event processing is object estimation. In this step, sensing data are rotated and translated by Kalman Filter or Particle Filter. After Kalman Filter or Particle Filter filters the sensing data, complex event processing step estimates whether updating the new event and discovers the things relevant to users. Object detection and estimation are executed on clients' devices, but the decision step as the third step of complex event processing is executed in cloud servers. The estimating and discovering result are sent to cloud servers. Cloud servers execute learning technology which finds the information inciting users' interests and knowing technology which notifies users about the information relevant to them.

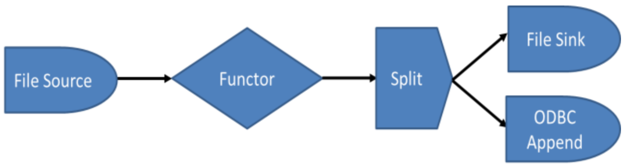

Fig. 5. Example of InfoShpere streams.

B. Cloud Servers

Cloud servers execute the decision step of complex event processing. Since sensing data are generated continuously, InfoSphere Streams which can analyze data in real time with micro-latency [2]–[5] is adopted in cloud servers. InfoSphere Streams is a parallel and high performance stream processing software platform that can scale over a range of hardware environments and also can automatically deploy stream processing applications on configured hardware and extend stream processing application without restarting. In Info Sphere Streams, data flows through operators which manipulate the data stream, and execute in-flight analysis on the data as shown in Fig. 4. Fig. 5 is an example of InfoSphere Streams. Functor operator transforms incoming data in some programmatic manner and sends data to next operator. In this example, the next operator is split operator. Data are classified to either a file sink or a database in split operator. From the example, we can find that InfoSphere Streams can analyze data and get result without storing data. All in all, InfoSphere Streams can intensively enhance performance of analyzing data in motion and achieve the Service Level Agreement (SLA) of real time multi-sensing applications on mobile and multi-sensing fusion platform.

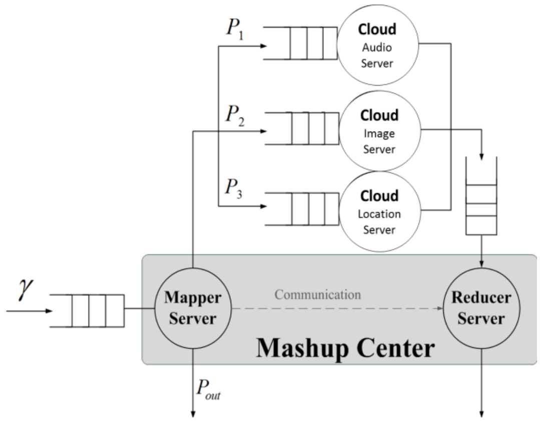

Fig. 6. The example of QoS-ensured performance model.

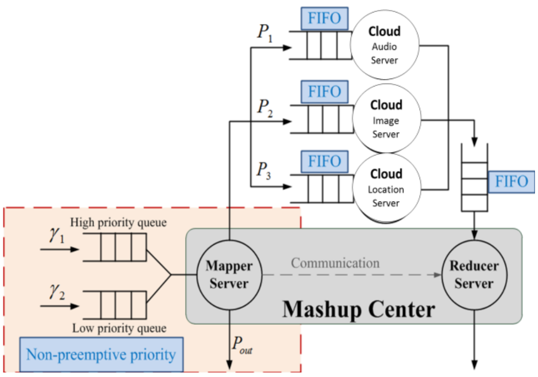

Fig. 7. The example of QoS-ensured performance model.

C. *QoS-Ensured Performance Model*

Applications on mobile and multi-sensing fusion platform are interactive with users and have to respond to users in real-time. Besides, the applications are mashup services which integrate the outcomes from multiple sensing data. For example, a location-based augmented reality application combines the location data and the camera data to create an augmented reality environment [9]. Thus, we propose a QoS-ensured performance model for mashup sensing applications. The QoS-ensured performance model analyzes the computation performance and allocates the computing resource to avoid violating the QoS of applications. We consider a mapper and reducer cloud model and adopt the FIFO policy as well as priority queueing model as shown in Fig. 6. The mashup center is composed of mapper server and reducer sever. When a request enters the mapper server, the service request will be forwarded to three different sensing data processing servers, such as audio processing server, image processing and location processing server with probability P_1, P_2, and P_3, respectively. The traffic will leave the mashup system with a probability of P_{out} if the service does not require the cloud server. The reducer server will integrate the outcomes of three different sensing servers and reply the integrated results to the users. There are single- and two-class traffic loads.

In single class traffic, as shown in Fig. 6, the mapper server and reducer server are modeled as the M/M/c queue, and three different sensing data processing servers are modeled as an M/M/1 queue. All the servers, including mapper, reducer and three different sensing data processing servers adopt First in First Out (FIFO) queue discipline, Poisson distributed arrival process, and the exponential distributed service time.

In two class traffic, the users are classified into payers and free users. The payers have higher priority to be served than the free users. When a payer asks for mashup service, his request will be placed in front of free users' requests, and follow the FIFO queueing discipline within the same class users. When a free user asks for the service, he will line up at the end of the queue if other users are waiting for serving. As mentioned before, the service request will enter into the cloud servers or leave the system with the probability P_1, P_2, P_3 and P_{out}. The queueing model is changed from Fig. 6 to Fig. 7. The external arrivals are divided into two classes. The high priority users are payers. The low priority users are free users. We also assume mapper and reducer are M/M/c queues and three different sensing data processing servers are M/M/1 queue. The service of mapper server discipline is non-preemptive priority and the other discipline is FIFO. The arrival rates for the high priority request and the low priority request is Poisson distribution. The service time at the servers are all exponentially distributed. The work of QoS-ensured performance model has published in [10].

III. Mobile and Multi-Sensing Fusion Applications

We design three multi-sensing applications includes education cloud, rescue cloud and business cloud on mobile and multi-sensing fusion platform. Three applications can represent clients' workload is heavier, clients' and servers' workload is equivalent and servers' workload is heavier.

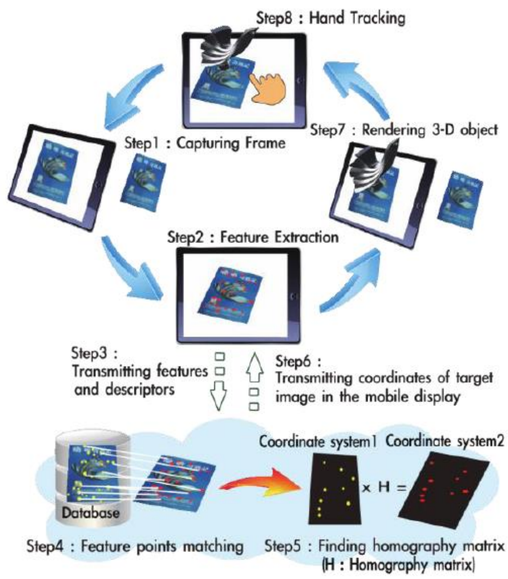

Fig. 8. The computation steps of education cloud.

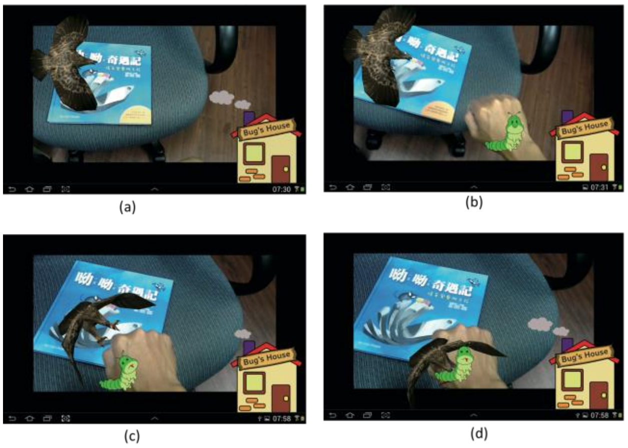

Fig. 9. The AR interaction of education cloud.

A. Education Cloud

In education cloud, we implement augmented reality (AR) technology on a book to interactive with users. If the sensors on the client devices detect the specific image, eagle will be shown on the screen as shown in Fig. 9(a). If the sensors detect users' hands, the caterpillar which lives in the house in the corner of the screen can be controlled by users' hands as shown in Fig. 9(b). Once the object is detected, the object estimation step is executed to update the hands' state. If the location of hands is too close to the eagle, the eagle will attack the caterpillar as shown in Fig. 9(c)(d).

TABLE I: COMPARISON OF MARAIS AND MARAIS-L

	MARAIS	MARAIS-L
Initial Time (s)	1.3	2.6
Use Heap (Detection Time (ms)	87.829	149
Description Time (ms)	139.5	128.6
Matching Time (ms)	10	116.39
Finding Homography (ms)	105	354.85
Processing Delay (ms) (Client Device)	227.329	644.089
Processing Delay (ms) (Cloud Server)	115	-
Upload Time (ms)	18	-
Download Time (ms)	0.011	-

In education cloud, we propose Mobile Augmented Reality Interactive System (MARAIS) to detect feature, describe feature and match feature points as shown in Fig. 8. Clients' devices are responsible for feature detection, feature description and rendering 3-D object onto the camera view to interactive with users. If we also execute feature points matching in local environments (MARAISL), the complex event processing delay is long as shown in Table I. Therefore, we divide MARAIS into MARAIS at mobile device (MARAIS-D) and MARAIS at cloud servers (MARAIS-C) to separate working load. Clients' devices capture color frames by camera and execute feature detection as well as description to find feature points. These feature points and descriptors are sent to MARAIS-C. MARAIS-C matches feature between captured image and target image in database and send the result back to MARAIS-D. After receiving information from MARAIS-C, MARAIS-D renders the corresponding 3-D object onto the camera view and execute particle-based hand tracking to provide user interaction with the 3-D object. From Table I, we can see that executing feature matching on cloud servers decrease process delay intensively. The detail of education cloud is described in [11]–[13]. In education cloud, clients' devices collect image data and build 3-D objects, so clients' workload is much heavier.

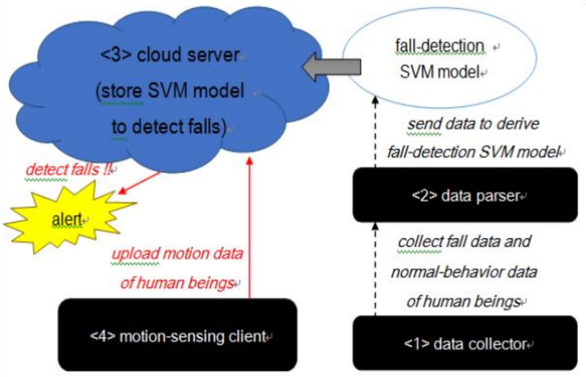

Fig. 10. The system architecture of rescue cloud.

B. Rescue Cloud

The second application is rescue cloud [14]. Owing to the increasing of elderly people falling down, we design a rescue cloud which can monitor the state of elderly people. If elderly people fall down, the rescue cloud call 911 and notify their family immediately. Fig. 10 shows the architecture of rescue cloud. A motion sensor which can transfer motion data into skeleton points effectively is chosen to detect falls as shown in Fig. 11(a).

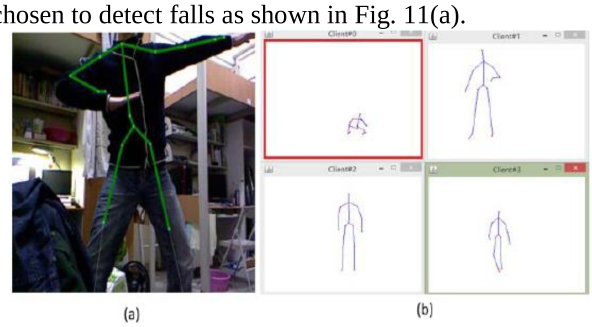

Fig. 11. The rescue cloud.

TABLE II: FALL DETECTION ACCURACY AND TIME

Kernel function	fall-detection accuracy(%) update period of input data			fall-detection accuracy(%) update period of input data		
	0.5 second	1.0 second	1.5 second	0.5 second	1.0 second	1.5 second
linear	99.89	99.82	99.79	1.06	0.94	0.86
polynomial	99.90	99.84	99.81	1.00	0.91	0.87
radial basis	99.27	98.55	97.84	89.52	47.03	32.09
sigmoid	99.27	98.55	97.84	3.02	2.96	3.06

This is the process of data collector. Once the skeleton points are detected, the skeleton points estimation step is executed to update, compress and send the skeleton points to the cloud servers. This is the process of data parser. In the cloud servers, support vector machine-based (SVM-based) intelligent clustering algorithm is designed and implemented in parallel to decide whether users are falling down as shown in Fig. 11(b). Table II shows falling detection accuracy and time by different kernel model of SVM. According to Table II, we adopt polynomial as kernel model and 0.5 second as input data update period in rescue cloud since polynomial has highest accuracy and shorter falling detection time. In rescue cloud, clients collect motion sensor data continuously and compress the sensing data to capture the feature of sensing data. Cloud servers execute complex event processing algorithm in parallel to decide whether there is an event. Therefore, the clients' workload and cloud servers' workload is equivalent in rescue cloud.

C. Business Cloud

The third application is business cloud [15]. Advertisements and coupons forwarding among mobile devices and social platforms to increase the consuming benefit become popular in recent years. However, sending same advertisements and coupons to every user cannot increase business benefit effectively. Thus, we design a personal consuming recommendation system as shown in Fig. 12 for customers by mining location data, social data and consuming data of customers. The input data includes consumers' information and shops' information will be collected by data collector. Attribute layer defines the attributes of every shop. The attributes of shops will be the input of social layer. Social layer shows the consumers' friend lists and compare the similarity of consumer behavior. Shop layer computes the relation between shops and use shop information to find the similarity of consumer behavior.

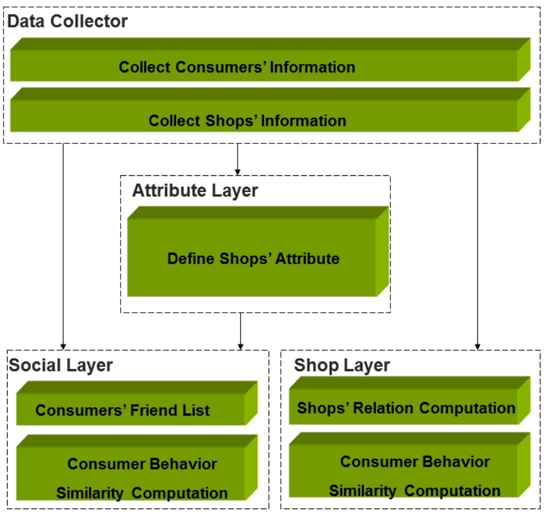

Fig. 12. The system architecture of business cloud.

We implement business cloud in the advertising system of shopping malls to find potential customers. The algorithm is called interest-aware opportunistic advertising by mining social and consuming information (ISC). We compare ISC with an interest aware PeopleRank (IPeR) algorithm, a fully distributed interest aware social-based algorithm to enable soft real-time opportunistic advertisement delivery in mobile networks [16]. In Fig. 13 and Fig. 14, we send one advertisement to one target user and ten advertisements to ten target users as two cases. In Fig. 13, hop count represents the number of intermediate nodes of forwarding advertisements to target customers. We can see that the hop counts of ISC are larger than IPeR. That is, ISC makes more interested customers receive the advertisements. The effect of advertisements is larger. Fig. 14 shows that the transmission delay of ISC is much shorter than IPeR even if the advertisements in ISC algorithm are delivered by many customers. All in all, ISC can achieve increasing total quantity of sales and making the advertisement receive by the target users in the limited time. In business cloud, clients execute object detection step to collect location data, social data and consuming data and send data to cloud servers. Cloud servers execute complex event processing to analyze the sensing data and decide a real-time recommendation result for customers. Therefore, the workload of cloud servers is much heavier than clients.

Three applications represent different workload between clients and cloud servers. Thus, realizing the pattern of different workload of clients and cloud servers and combining the result of QoS-ensured performance model to adjust the platform computing resource can decrease the probability of violating the QoS of different applications on mobile and multi-sensing fusion platform.

IV. Challenge

In this section, we will discuss the challenges and the future trend of mobile and sensing fusion platform.

A. Challenge 1

The mobile and sensing fusion platform we propose can be extended to internet of things (IOT). In the IOT environment, there are many sensors instead of single sensor generate complex sensing data. The sensing data includes longitude as well as latitude, image data, audio data, accelerometer sensor data, etc. The data are unstructured data instead of structured data. Besides, sensing data are generated continuously and have to be analyzed in real-time. Therefore, how to design a real-time complex event processing flow for streaming sensing data becomes the most challenge issue in the multi-sensing environment.

B. Challenge 2

From previous challenge, we can know that sensing data has to be analyzed in real-time. The QoS issue becomes critical during designing real-time complex event processing flow for streaming data. In the traditional QoS analysis, we usually build system first and then we design the experiment to analyze system performance and QoS. The computing resource is allocated and static in the system. However, the computing resource of cloud platform is dynamic. Therefore, using a queueing model can estimate how many computing resource is needed before building the system in the application and adjust the computing resource dynamically during executing jobs to avoid violating QoS. In [10], we proposed a queueing model to analyze QoS of mashup service. [17] proposed a M/G/m/m queueing model to analyze the usage of storage space, memory and computation resource in the physiological streaming data collection platform, Artemis. From the work, we can see that queueing model can not only use for analyzing static data system, but also for dynamic streaming data system. Therefore, modeling an appropriate queueing model for your applications can intensively decrease the chance to violate QoS and improve system performance.

C. Challenge 3

Since streaming data are generated continuously on the mobile and multi-sensing fusion platform, sending all streaming data to cloud occupies lots of bandwidth and increases the response time. To achieve high throughput of processing the streaming data, partitioning solution for mobile data stream applications and execution offloading schemes to migrate a process between machines are proposed. [18] proposed a framework to support dynamic computation partitioning as well as execution of the application and a genetic algorithm for optimizing the computation partition. [19] migrated entire state including the existing stack as well as all reachable heap objects to offload the full process. In [20], the stack was set to run remotely and be invoked by other servers instead of migrating to other servers. Therefore, the usual amount of state transferred was the main factor to decide the migration efficiency. [21] which was based on [19] proposed a compiler code analysis to only transfer the essential heap objects and the stack frames actually be necessary by the server. [21] could maintain the state of the process and reduce the transferred data size intensively. Therefore, how to partition and migrate the sensing data analysis process between clients and cloud servers to increase the analysis performance of streaming sensing data still be a big issue.

D. Challenge 4

In order to decrease the execution time of continuously streaming sensing data, the streaming data is analyzed while data flow into the operators without storing as shown in Fig. 4. Without storing huge streaming data can decrease the access time of I/O to improve the efficiency of data analysis. However, without storing data may result in massive data loss if an operator is broken during the computation. Therefore, data fault tolerance technology is a necessary technology in streaming data processing. [22]-[24] proposed data duplication methods to guarantee that no data is lost or any inconsistency exists. But data duplication occupies many storage space and cause significant performance degradation. Thus, [25]–[28] proposed Partial Fault Tolerance (PFT) technology. PFT executed partial data duplication and accept some data loss. [26] designed a specialized state serialization methods based on a stream operator checkpoint mechanism. When an operator failed, its upstream operators did not send the data to the operator until the operator was fixed and could generate correct result. This method caused much data loss of input stream on the operator. In order to make PFT is viable on stream application, understanding of the impact of faults on the quality of the application output is important. [29] injected faults into application running on the streaming processing platform, which called System S. They deigned a valuing mechanism to assess the application output quality and propose four metrics to evaluate the impact of faults in different stream operators of applications. According to the result, the developer could choose the operator to execute Partial Fault Tolerance technology (PFT).

From the previous discussion, we find that data fault tolerance technology affects the streaming applications performance intensively. Thus, designing efficient fault tolerance methods and execute the methods on right operators are two important issues of streaming applications.

V. Conclusion

In this paper, we proposed mobile and multi-sensing fusion platform which includes clients, cloud servers and QoS-ensured performance model. Clients detected, estimated objects and sent data to cloud servers. Cloud servers analyzed data and decided whether there was an event. QoS-ensured performance model evaluated the computing resource of multi-sensing fusion platform. We also showed three different workload multi-sensing fusion applications as the examples on the platform. Besides, we discussed unstructured streaming data analysis, QoS issue on streaming system, partition as well as offloading execution process between clients and cloud servers and data fault tolerance as four challenges of mobile and multi-sensing fusion research. In mobile cloud computing, this paper described the trend and challenges of signal and information processing in great detail.

References

[1] D. Robins, "Complex event processing," in *Proc. Second International Workshop on Education Technology and Computer Science*, 2010.

[2] H. Andrade, B. Gedik, and D. Turaga, *Fundamentals of Stream Processing: Application Design, Systems, and Analytics*, Cambridge University Press, 2014.

[3] D. DeRoos, C. Eaton, G. Lapis, P. Zikopoulos, and T. Deutsch, *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data*, McGraw-Hill Osborne Media, 2011.

[4] B. Gedik and H. Andrade, "A model-based framework for building extensible, high performance stream processing middleware and programming language for ibm infosphere streams," *Software: Practice and Experience*, vol. 42, no. 11, pp. 1363–1391, 2012.
[5] M. Hirzel, H. Andrade, B. Gedik, G. Jacques-Silva, R. Khandekar, V. Kumar, *et al*., "IBM streams processing language: Analyzing big data in motion," *IBM Journal of Research and Development*, vol. 57, no. 3-4, 2013.
[6] E. Ciurana, "Google app engine," *Developing with Google App Engine*, 2009.
[7] H. Li, *Introduction to Windows Azure*, Springer, 2009.
[8] Amazon ec2. [Online]. Available: http://aws.amazon.com/cn/ec2/
[9] A. Henrysson and M. Ollila, "Umar: Ubiquitous mobile augmented reality," in *Proc. ACM International Conference on Mobile and Ubiquitous Multimedia*, 2004.
[10] W. P. Yang, L. C. Wang, and H. P. Wen, "A queueing analytical model for service mashup in mobile cloud computing," in *Proc. IEEE Wireless Communications and Networking Conference*, 2013.
[11] K. T. Feng, P. H. Tseng, P. S. Chiu, J. L. Yang, and C. J. Chiu, "3d interactive augmented reality-enhanced digital learning systems for mobile devices," in *The Engineering Reality of Virtual Reality*, 2013.
[12] P. H. Chiu, K. T. Feng, and P. H. Tseng, "Particle-based augmented reality interactive system," in *Proc. IEEE International Conference on Multimedia and Expo Workshops*, 2013.
[13] P. H. Chiu, P. H. Tseng, and K. T. Feng, "Cloud computing based mobile augmented reality interactive system," in *Proc. IEEE Wireless Communication and Networking Conference*, 2014.
[14] C. H. C. Liao, K. W. Lee, T. H. Chen, C. C. Chang, and C. H. P. Wen, "Fall detection by a svm-based cloud system with motion sensors," in *Proc. International Conference on Human-Centric Computing*, 2013.
[15] C. Y. Lin, Z. F. Jiang, L. C. Wang, and B. S. P. Lin, "An effective algorithm for interest aware opportunistic advertising by mining social and consuming information," in *Proc. IEEE Vehicular Technology Conference*, 2014.
[16] S. Al Ayyat, K. A. Harras, and S. G. Aly, "Interest aware peoplerank: Towards effective social-based opportunistic advertising," in *Proc. IEEE Wireless Communications and Networking Conference*, 2013.
[17] H. Khazaei, C. McGregor, M. Eklund, K. El-Khatib, and A. Thommandram, "Toward a big data healthcare analytics system: A mathematical modeling perspective," in *Proc. IEEE World Congress on Services*, 2014.
[18] L. Yang, J. Cao, Y. Yuan, T. Li, A. Han, and A. Chan, "A framework for partitioning and execution of data stream applications in mobile cloud computing," *ACM SIGMETRICS Performance Evaluation Review*, vol. 40, no. 4, pp. 23–32, 2013.
[19] B. G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: Elastic execution between mobile device and cloud," in *Proc. ACM Conference on Computer Systems*, 2011.
[20] E. Cuervo, A. Balasubramanian, D. K. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "Maui: Making smartphones last longer with code offload," in *Proc. ACM International Conference on Mobile Systems, Applications, and Services*, 2010.
[21] S. Yang, D. Kwon, H. Yi, Y. Cho, Y. Kwon, and Y. Paek, "Techniques to minimize state transfer costs for dynamic execution offloading in mobile cloud computing," *IEEE Transactions on Mobile Computing*, vol. 13, no. 11, pp. 2648–2660, 2014.
[22] Y. Gu, Z. Zhang, F. Ye, H. Yang, M. Kim, H. Lei, and Z. Liu, "An empirical study of high availability in stream processing systems," in *Proc. ACM/IFIP/USENIX International Conference on Middleware*, 2009.
[23] J. H. Hwang, U. Cetintemel, and S. Zdonik, "Fast and highlyavailable stream processing over wide area networks," in *Proc. IEEE International Conference on Data Engineering*, 2008.
[24] Y. Kwon, M. Balazinska, and A. Greenberg, "Fault-tolerant stream processing using a distributed, replicated file system," *Proceedings of the VLDB Endowment*, vol. 1, no. 1, pp. 574–585, 2008.
[25] N. Bansal, R. Bhagwan, N. Jain, Y. Park, D. Turaga, and C. Venkatramani, "Towards optimal resource allocation in partial-fault tolerant applications," in *Proc. IEEE International Conference on Computer Communications*, 2008.
[26] G. Jacques-Silva, B. Gedik, H. Andrade, and K. L. Wu, "Language level checkpointing support for stream processing applications," in *Proc. IEEE/IFIP International Conference on Dependable Systems & Networks*, 2009.
[27] R. N. Murty and M. Welsh, "Towards a dependable architecture for internet-scale sensing," in *Proc. Second Workshop on Hot Topics in System Dependability*, 2006.
[28] Q. Zhu, L. Chen, and G. Agrawal, "Supporting fault-tolerance in streaming grid applications," in *Proc. IEEE International Parallel and Distributed Processing Symposium*, 2008.
[29] G. Jacques-Silva, B. Gedik, H. Andrade, K. L. Wu, and R. K. Iyer, "Fault injection-based assessment of partial fault tolerance in stream processing applications," in *Proc. ACM International Conference on Distributed Event-Based System*, 2011.
[30] E. Walker, "Benchmarking amazon ec2 for high-performance scientific computing," *Usenix Login*, vol. 33, no. 5, pp. 18–23, 2008.

Chia-Yu Lin received the B.S. and M.S. degree from National Chiao Tung University, Taiwan in 2010, and 2012, respectively, all in computer science. She is currently working toward the Ph.D. degree with the Institute of Communications Engineering, National Chiao Tung University, Taiwan. Her current research interests include VM resource estimation and load balancing in cloud data centers.

Li-Chun Wang received the B.S. degree from National Chiao Tung University , Taiwan, R. O. C. in 1986, the M.S. degree from National Taiwan University in 1988, and the Ms. Sci. and Ph. D. degrees from the Georgia Institute of Technology , Atlanta, in 1995, and 1996, respectively, all in electrical engineering. From 1990 to 1992, he was with the Telecommunications Laboratories of the Ministry of Transportations and Communications in Taiwan (currently the Telecom Labs of Chunghwa Telecom Co.). Since August 2000, he has been a Professor in the Department of Communication Engineering of National Chiao Tung University in Taiwan. His current research interests are in the areas of adaptive/cognitive wireless networks, radio network resource management, cross layer optimization, and cooperative wireless communication networks.