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 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 and multi-sensing fusion technology and signal and information processing in mobile cloud computing in great detail.**

***Keywords*-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

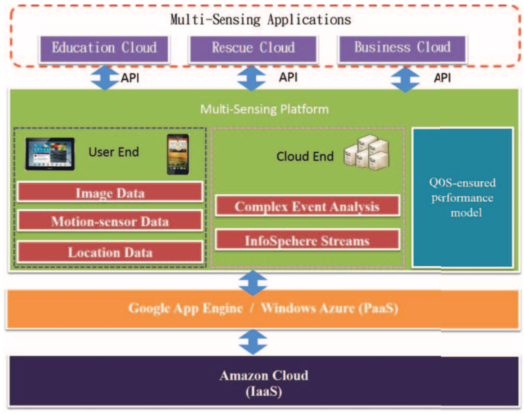

Figure 1. System architecture of mobile and multi-sensing fusion platform.

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.

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 unstructured, intensive and streaming sensing data is the first challenge. If we send all the sensing data to cloud to analyze, we will cost a lot of bandwidths and time to transfer data. Thus, multi-sensing application partitioning between clients and cloud servers is the second challenge. The third challenge is data fault tolerance technologies of multi-sensing applications. Analyzing sensing data without storing technology is proposed to increase the performance of sensing data analysis. However, if the computation node is 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

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

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.

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 behaviors 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 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 of the information relevant to them to decide whether there is a event relevant to users.

B. Cloud Servers

Cloud servers executr 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

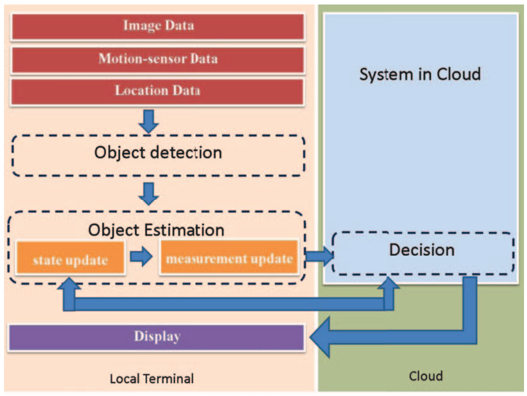

Figure 3. Complex event processing flow.

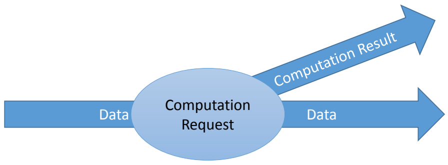

Figure 4. InfoSphere Streams computation model.

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 InfoSphere 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 send data to next operator. In this example, the next operator is split operator. Data is 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.

C. QoS-ensured Performance Model

Applications on mobile and multi-sensing fusion platform are interactive with users and have to response 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. There are single- and two-class traffic loads. We denote P_1, P_2, and P_3 as the

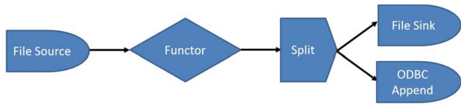

Figure 5. Example of InfoShpere Streams.

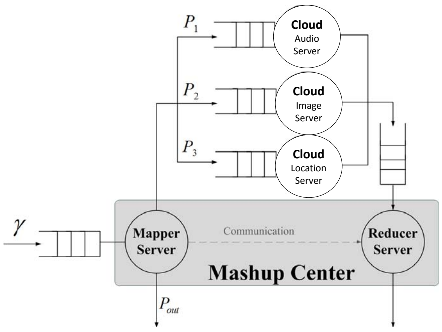

Figure 6. The example of QoS-ensured performance model.

probabilities of the request to three different sensing data processing servers, such as audio processing server, image processing and location processing server. The work of QoS-ensured performance model has published in [10].

In the traditional system analysis, we usually build system first and then we design the experiment to analyze the system. However, the computing resource of cloud platform is dynamic. Therefore, designing a queueing model for the system can estimate and manage system computing resource accurately during system analysis. After building a queueing model, we can develop system according to the estimation result of queueing model and to build verify the estimation result of queueing model and adjusting computing resource dynamically.

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. In education cloud, we implement augmented reality (AR) technology on a book to interactive with users as shown in Fig. 7. If the sensors on the client devices detect the specific image, eagle will be shown on the screen as shown in Fig. 7(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. 7(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. 7(c)(d). We develop feature detection, feature description and feature points matching to catch image as well as hand motion and create the augmented reality interactive learning environment [11]–[13]. In education cloud, the complex event processing is executed on the clients to give response in real-time, so clients' workload is much heavier.

The second application is rescue cloud [14]. Owing to

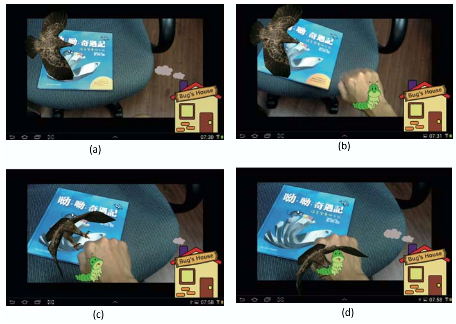

Figure 7. The AR interaction of education cloud.

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. In rescue cloud, a motion sensor which can transfer motion data into skeleton points effectively is chosen as the sensor devices for detecting falls as shown in Fig. 8(a). Once the skeleton points are detected, the skeleton points estimation step is executed to update, compress the skeleton points and send to the cloud servers. 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. 8(b). 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 of rescue 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 intensively. Thus, we design a personal consuming recommendation system as shown in Fig. 9 for customers by mining location data, social data and consuming data of customers. 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 of 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.

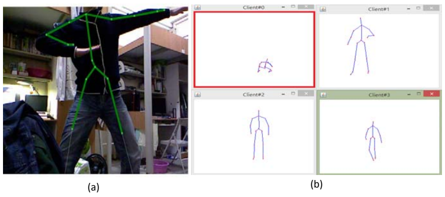

Figure 8. The rescue cloud.

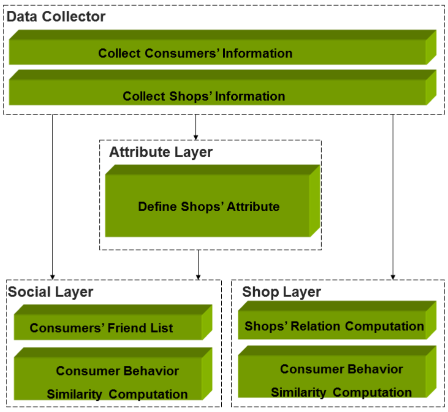

Figure 9. The system architecture of business cloud.

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 become the most challenge issue in the multi-sensing environment.

B. Challenge 2

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. [16] proposed a framework to support dynamic computation partitioning as well as execution of the application and a genetic algorithm for optimizing the computation partition. [17] migrated entire state including the existing stack as well as all reachable heap objects to offload the full process. In [18], 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. [19] which was based on [17] proposed a compiler code analysis to only transfer the essential heap objects and the stack frames actually be necessary by the server. [19] 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.

C. Challenge 3

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. [20]–[22] 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, [23]–[26] proposed partial fault tolerance (PFT) technology. PFT executed partial data duplication and accept some data loss. [24] 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 bursty 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. [27] 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, partition as well as offloading execution process between clients and cloud servers and data fault tolerance as three 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.

VI. Acknowledgement

This work was jointly sponsored by MOST103-2221-E-009-015-MY2, and ITRI-103-EC-17-A-21-0885. The authors of this paper are thankful for Dr. Jian-Ren Chen of Industrial Technology Research Institute for his insightful discussions and comments.

References

[1] D. Robins, "Complex event processing," in *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.

[3] P. Zikopoulos, C. Eaton *et al.*, *Understanding big data: Analytics for enterprise class hadoop and streaming data*. McGraw-Hill Osborne Media.

[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, M. Mendell, H. Nasgaard, S. Schneider, R. Soulé *et al.*, "Ibm streams processing language: Analyzing big data in motion," *IBM Journal of Research and Development*, vol. 57, no. 3/4, pp. 7:1–7:11, 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," http://aws.amazon.com/cn/ec2/.

[9] A. Henrysson and M. Ollila, "Umar: Ubiquitous mobile augmented reality," in *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 *IEEE Wireless Communications and Networking Conference (WCNC)*, 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 *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 *IEEE Wireless Communication and Networking Conference (WCNC)*, 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 *Proceeding of 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 *IEEE Vehicular Technology Conference*, 2014.

[16] 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.

[17] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: elastic execution between mobile device and cloud," in *ACM Conference on Computer Systems*, 2011.

[18] 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 *ACM International Conference on Mobile Systems, Applications, and Services*, 2010.

[19] 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*, 2014.

[20] 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 *ACM/IFIP/USENIX International Conference on Middleware*, 2009.

[21] J.-H. Hwang, U. Cetintemel, and S. Zdonik, "Fast and highly-available stream processing over wide area networks," in *IEEE International Conference on Data Engineering*, 2008.

[22] 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.

[23] N. Bansal, R. Bhagwan, N. Jain, Y. Park, D. Turaga, and C. Venkatramani, "Towards optimal resource allocation in partial-fault tolerant applications," in *IEEE International Conference on Computer Communications*, 2008.

[24] G. Jacques-Silva, B. Gedik, H. Andrade, and K.-L. Wu, "Language level checkpointing support for stream processing applications," in *IEEE/IFIP International Conference on Dependable Systems & Networks*, 2009.

[25] R. N. Murty and M. Welsh, "Towards a dependable architecture for internet-scale sensing," in *Second Workshop on Hot Topics in System Dependability*, 2006.

[26] Q. Zhu, L. Chen, and G. Agrawal, "Supporting fault-tolerance in streaming grid applications," in *IEEE International Parallel and Distributed Processing Symposium*, 2008.

[27] 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 *ACM International Conference on Distributed Event-based System*, 2011.

[28] E. Walker, "Benchmarking amazon ec2 for high-performance scientific computing," *Usenix Login*, vol. 33, no. 5, pp. 18–23, 2008.