

A Reformative System Architecture of Cloud Service Based on IEEE 1851

Mingxing Jiang¹, Zhongwen Guo², and Chao Liu²

¹ Department of Basic Education Center

² Department of Information Science and Engineering

Ocean University of China, Qingdao, China

{jamison, guozhw}@ouc.edu.cn, liuchao.ouc@gmail.com

Abstract. A system framework is proposed in IEEE 1851; however it is not organized in cloud computing mode. Actually, since information resource in enterprise can be integrated by web service interfaces in IEEE 1851, architecture of cloud service is preferred for better resource sharing. This paper mainly proposes a reformative architecture of cloud service system based on IEEE 1851, analyzes the mechanism on the cloud, and evaluates the implementation results, etc. Also we discuss the interior in the cloud: a common XML schema is put forward for data interaction; memory scheduling and access control are respectively distributed to gain a high, reliable performance. Analysis with experimental data from the novel architecture shows the applicability of the proposed system architecture.

Keywords: IEEE 1851, home appliance test, cloud computing, web service, memory scheduling.

1 Introduction

Through providing computation, data access, storage, and even platform services, today cloud computing is dynamically revolutionizing information technology and impacting the modern Internet computing and businesses in every aspect [1, 2, 3]. In practice, its impact seems more profound for those large enterprises whose information and software resource are in urgent need of centralization and integration for high efficiency [4, 5, 6].

In IEEE 1851 [7], a system framework oriented SOA is proposed to integrate existing resource (e.g. software and data). However, when a close coupling architecture is implemented or security issue is considered, cloud computing mode is practically preferred as its high resource-centralization and reliable resource-access [8, 9].

The architecture in this paper is proposed in a cloud-computing way: the data source, services on the cloud as system key parts, the clients as the end user and the Internet to tie them together. In this architecture users are more convenient to obtain data from different places. Besides usability and convenience, special mechanisms ensure every client a reliable and timely service.

The rest of this paper is organized as follows. In the next section, we begin with a brief overview of the architecture. Section 3 proposes two practical mechanisms on the cloud. In section 4, we analyze the caching mechanism and illustrate access control issue with a few interface diagrams. The last section summarizes our contribution and lists the limitations of this research.

2 Architecture Overview

The proposed architecture incorporating data sources up to IEEE 1851 can be built using three types of components: user authorization module (UAM), virtual resource manager (VRM), and distributed data access module (DDAM). The design of this architecture is generic enough to integrate data sources in form of IEEE 1851 and put them on the cloud side under multilevel security [10].

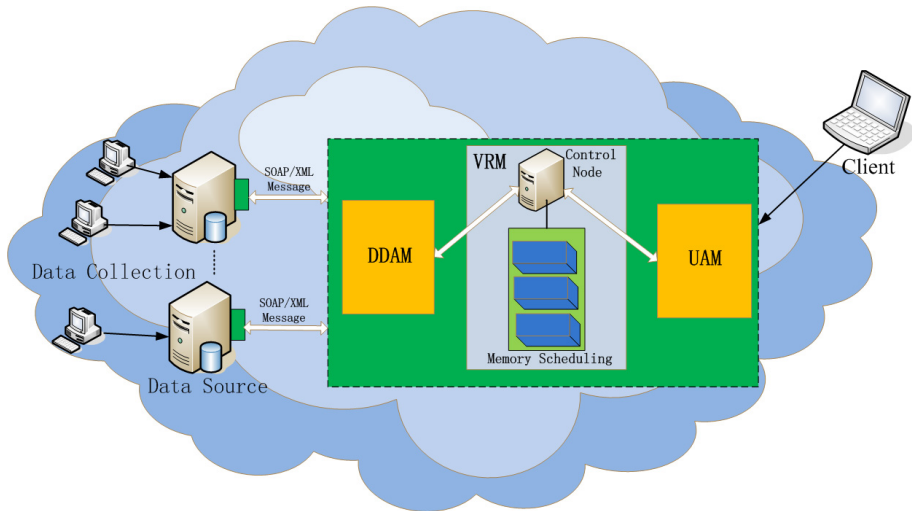


Fig. 1. Cloud Architecture

As Figure 1 shows, in this architecture, VRM is responsible for answering all requests from accredited clients [11]. Once a request comes, firstly UAM will authorize it while expelling invalid ones, then VRM will analyze the request and distribute relevant resource depending on user credentials. Besides, DDAM will take back required data from data source through web service interfaces standardized in IEEE 1851. These data will be loaded into memory for high-speed access.

3 Interior on the Cloud

3.1 Format for Data Interaction

In home appliance test, mass data generates since a large amount of sensors deployed on each test unit are collecting data all the time. We call these data test data, which

occupied most of data interaction time from clients to servers in traditional system architecture. Most applications for home appliance test are designed to use a Picture-Box to display these test data and need lots of client-server interactions concerning large volume of data transmission.

Due to this fact, we propose a novel format for data interaction which enables high-efficiency data transmission. Instead of total test data, a result in form of picture will be sent to clients, which markedly reduce volume of data transmission. To be specific, we design a common XML schema for test data interaction. An image that reflects computing result on the cloud is designed to be embedded into an XML document using the Base-64 encoding approach.

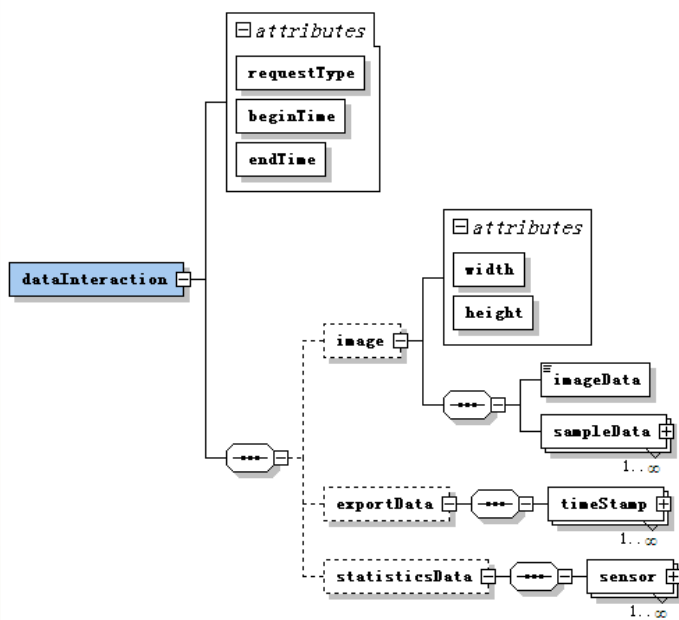


Fig. 2. Structure of dataInteraction.xsd

Fig 2 shows the structure of the proposed XML schema. An instance file of the schema will be shared between clients and the cloud. Every request contains 3 basic parameters: beginTime, endTime, and requestType. Since requestType element is an enumeration including Image, Export, Statistics, responses are supposed to fill corresponding element (image, exportData, and statisticsData) with computational results. Three elements are listed below:

- **image:** This element is used to response to the “Image” request. XML instance includes an encoded binary data array about the image. Also is contained is the image size (width and height) and sampleData element.

- **exportData:** This element is used to response to the “Export” request. When user from client side needs test data of a certain period, this type of request will be triggered.
- **statisticsData:** This element is used to response to the “Statistics” request. When users want to view statistics of a certain period, the cloud will fill this element with statistics results.

3.2 Cache Mechanism

Because there are no data available in data center memory at beginning of each client request, required data will be fetched through DDAM by implementing the web service interfaces. Even though those mentioned components are deployed on the cloud, it remains inefficient in data transmission between web service interfaces and DDMA, due to the inherent weakness of web service. Based on this, we proposed a caching mechanism on the cloud to deal with efficiency problem.

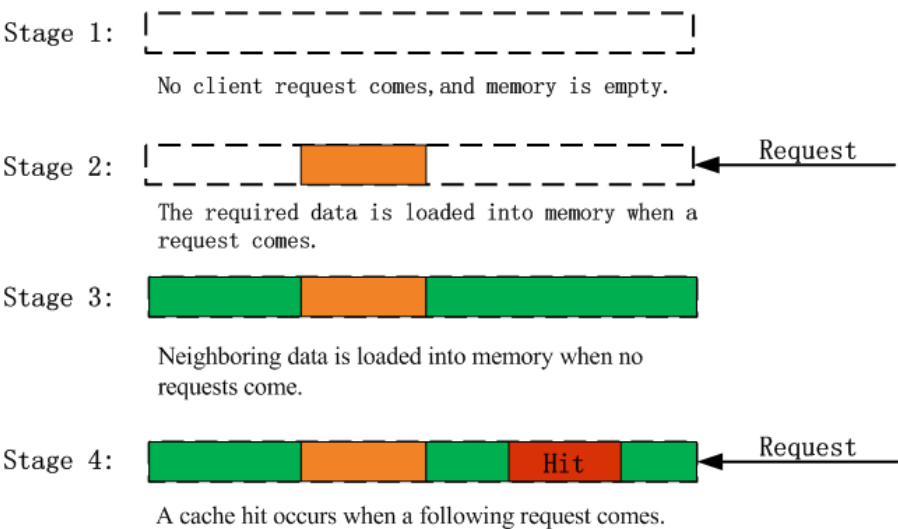


Fig. 3. Sequence diagram of memory scheduling

As Fig. 3 shows, upon receiving request from accredited client, URM will first get the required data from DDAM. After responding the coming request, URM will obtain data related to the latest request and load them into memory. When another request from the same client comes later, the required data will be probably loaded in main memory, because in practice requests from one client usually concentrate on neighboring data.

Based on the above consideration, a cache mechanism is proposed in this paper. The URM will not only allocate storage space for the coming request, but also for probable requests later. Certainly, storage space loaded in main memory is designed to be released when the session between clients and servers ends. In home appliance test, a data set usually contains test data from beginning to end (or till now), so when the proposed mechanism is applied to home appliance test, URM mainly maintains a complete data set of one test in the memory on the cloud.

3.3 Security Mechanism

As is well-known, it is challenging to implement effective security in web service framework. That means exposing web service interfaces on the Internet is a big security flaw. In this proposed architecture, web service interfaces are hidden on the cloud, clients will never be authorized direct access to these interfaces. So, adopting this architecture is part of the security mechanism.

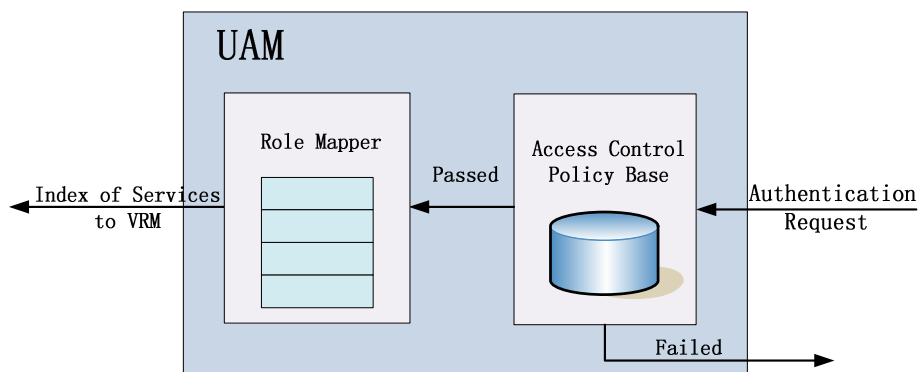


Fig. 4. User Authorization Module (UMA) Architecture

Further, a restrictive user access is adopted to ensure security of this cloud computing architecture. As Fig. 4 illustrates, every request from clients is supposed to turn to UAM for a certification with a pair of username and matched password. What's more, UAM not only responses with a "passed" or "failed" result to client request, but also generates a detailed privilege level if authentication is passed. Then depending on different levels, URM will present different index of services. In this way, we keep web service interfaces only open to certain authenticated users of sufficient authority.

4 Case Study (Implementation Results and Analysis)

In this section, a prototype application was constructed for validating the practicability and security of the proposed architecture. Additionally, we also analyze experimental data and draw some conclusions from it.

Before access to the cloud service, the authentication process is necessary. We use the traditional username and password way to realize this function. The login user interface is shown in Fig.5. After authentication, we could use the service smoothly to display data result in web browser. Fig.6 shows a user interface of household appliance test application. At the top of the frame, several different functions are provided, such as time profile, user profile and so on. The right frame shows the authority information of this user, which is listed in an expandable tree view.

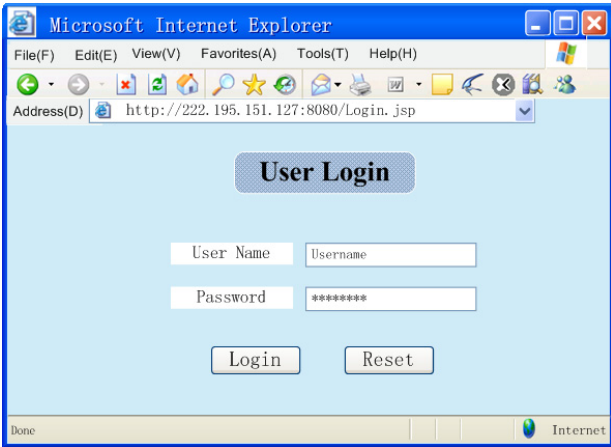


Fig. 5. Login Interface

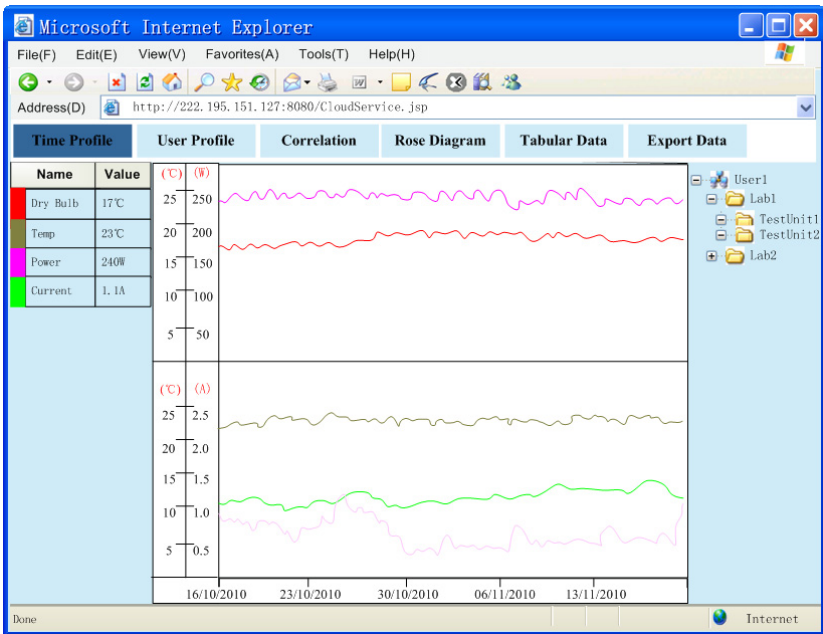


Fig. 6. User Interface

An experiment is done to test and prove that clients can gain a higher efficiency in getting data from server with all request data loaded in memory than from server without it. We define these two methods as cache method and non-cache method. Fig.7 specifies the detailed comparison diagram. Data volume is measured by the amount of time because data generates periodically in home appliance test. Obviously, cache method appears more efficient as its lower time cost. Table 1 indicates the time cost ratio between cache method and non-cache. With the growth of data volume, time cost ratio is stable at nearly 17%. Assume that we would require data covers 1000 hours one time and the cache hit rate is 50%. The system performance would be improved by 55.6%.

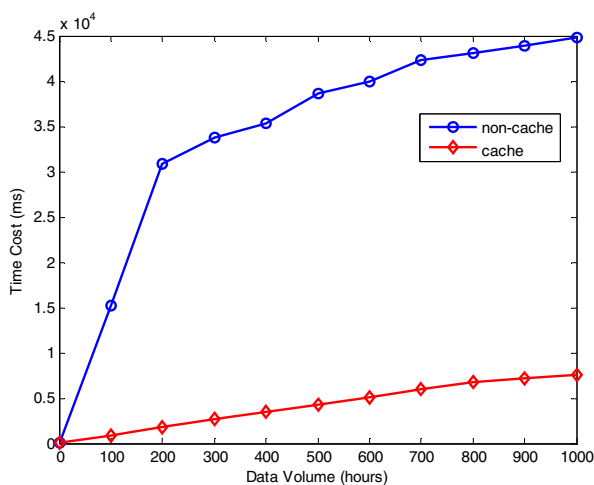


Fig. 7. Performance of cache method and non-cache method

Table 1. Time cost ratio

Data Volume	200	400	600	800	1000
Ratio	5.9%	10.02%	12.74	15.58%	16.9%

5 Conclusion and Future Work

Finally, as the object of study, a reformative architecture based on IEEE 1851 Standard is proposed. Existing resource in an enterprise can be integrated well by adopting this architecture. Furthermore it presents an opportunity for providing service over the Internet to improve resource utilization. We have discussed the challenges and opportunities for maintaining a sustained and stable service in cloud environments. Three important strategies are described to make the proposed architecture more efficient and secure. Implementation result shows practicability of the architecture and effectiveness of these mechanisms.

In this paper, we just implement system security mechanism on a small scale. When the number of users expands, many problems unexpected will have to be dealt with. Besides, according to our cumulative experience, the conclusion that individual user requests usually focus on neighboring data is drawn, and a minimum hit ratio is proposed for reference when a caching mechanism is designed. If more practical data are provided, we can obtain some other instructive results.

References

1. Gerber, C.: Computing in the clouds. *Military Information Technology* 12(10), 15–17 (2008)
2. Abdulrahman, A., Sarfraz, M.I., Basalamah, S.: A Distributed Access Control Architecture for Cloud Computing. *IEEE Software*, 36–44 (March/April 2012)
3. Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I.: Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility. *Future Generation Computer Systems* 25(6), 599–616 (2009)
4. Napper, J., Bientinesi, P.: Can Cloud Computing Reach the Top 500? In: UCHPC-MAW 2009, Ischia, Italy, May 18–20 (2009)
5. Daponte, P., Grimaldi, D., Nigro, L., Pupo, F.: Distributed measurement systems: An object-oriented architecture and a case study. *Comput. Stand. Inter.* 18(5), 383–395 (1997)
6. Song, E., Lee, K.: Service-oriented Sensor Data Interoperability for IEEE 1451 Smart Transducers. In: *Proc. I2MTC*, Singapore, May 5–7, pp. 1043–1048 (2009)
7. IEEE 1851, Standard for Design Criteria of Integrated Sensor-based Test Applications for Household Appliances. IEEE Standards Association (2012)
8. Takabi, H., Joshi, J.B.D., Ahn, G.-J.: Security and Privacy Challenges in Cloud Computing Environments. *IEEE Security & Privacy* 8(6), 24–31 (2010)
9. Amit, G., Heinz, S., David, G.: Formal Models of Virtual Enterprise Architecture: Motivations and Approaches. In: *PACIS 2010 Proceeding* (2010)
10. Patterson, D.A.: Technical perspective: the Data Center is the Computer. *Communications of the ACM* 51(1), 105 (2008)
11. Buyya, R., Yeo, C.S., Venugopal, S.: Market-oriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities. In: *Proc. 10th IEEE Int. Conference on High Performance Computing and Communications, HPCC 2008*, Dalian, China (September 2008)