



Efficient Cloud Platform Providing an Omnipresent Healthcare Services

Kalaiprasath.R¹, R. Udayakumar*², R. Elankavi³

Research Scholar, Bharath University, Chennai, India¹

Associate Professor, Department of Information Technology, Bharath University, Chennai, India²

Research Scholar, Bharath University, Chennai, India³

*Corresponding Author & Research Supervisor

ABSTRACT: Health care is the diagnosis, treatment, and prevention of disease, illness, injury, and other physical and mental impairments in humans. Healthcare Services refers to the work of health care professionals who act as a first point of consultation for all patients within the health care system. Cloud computing owns the pervasive and on-demand service-oriented natures, which can fit the characteristics of health-care services. But in the case of general cloud, there arise few challenges which are providing persistent personalized service by keeping high concurrent online analysis to the public, and the abilities in dealing with multimodal, heterogeneous, and non-stationary physiological signals. In this paper, we proposed a private cloud platform architecture that consists of six layers according to the specific requirements with plug-in algorithm. This platform utilizes message queue as a cloud engine, and each layer thereby achieves relative independence by this loosely coupled means of communications with publish/subscribe mechanism. Also the massive semi-structure or unstructured medical data are accessed adaptively by this cloud architecture. This proposed cloud platform can be robust, stable, and satisfy the high concurrent requests from omnipresent healthcare services.

KEYWORDS: Cloud computing, healthcare services, plug-in algorithm, ubiquitous.

I. INTRODUCTION

According to the World Health Organization report, the incidence of the suboptimal health status (SHS), which is also called “the third state” (between health and disease), accounts for 75% of the world population. Only in China, there are estimated 900 million people in such status. Some of them would pay close attention to their health hoping to get preventive health examination. Moreover, with the aging population growing, we also have to take measures to launch chronic disease surveillance for elder people. Therefore, to satisfy the demands of “SHS” groups and aging population, the healthcare system is proposed and becoming more and more popular, which can provide self-monitoring of healthy status, early warning of disease, and even an instant report of the physiological signal analysis for individuals. Until now, such health monitoring systems have been widely implemented with traditional management information system (MIS) mode, which could no longer meet all the requirements of the healthcare system. For instance, Leel et al. propose a service-oriented architecture

(SOA)-based system to integrate the data from a personal health device and customize services for users, but somewhat lack concurrency and persistent service capabilities; Kulkarni and Ozturk developed a system named *mPHASiS*, which claims an end-to-end solution. However, it relies on the particular programming language, Java, which will result in less efficient of system due to the performance of Java virtual machine (JVM). Nowadays, as an emerging state-of-the-art technology, cloud computing can provide pervasive, economical, and on-demand services for customers. Many pieces of research on medical information have been involved into the cloud recently. It can fulfil the flexibility and scalability of service-orientated systems, just like the healthcare system infrastructure should have, which is very different from the transaction-oriented MIS mode. Cloud computing inherited the features of high-performance parallel computing, distributed computing, and grid computing, as well as the further development of these techniques to achieve location transparency to enhance the user experiences over the Internet. More and more applications are migrated onto the cloud platform, which can provide services for various application scenarios. Such a platform generally comprises a number of diverse groups of components, no matter they are open source or not. Consequently, how to design a model with optional components to meet the needs of flexibility and scalability is one of the critical issue when constructing a cloud platform. In the case of the solution for this healthcare system, we presented a six-layer deployment architecture and proposed a message queue (MQ) as cloud engine as well as fundamental cloud storage based on the open-source components, which can be obtained freely and timely from Internet communities. Also, a



plug-in algorithm framework was developed with publish/subscribe mechanism to implement this healthcare system for the community people. The expandable feature of the publish/subscribe communication model can meet the needs of a health cloud platform in high concurrent performance. Details about the implementation are depicted in Section III after the analysis of the practical requirements of healthcare in Section II. The results of performance test of the system are given in Section IV, and the Section V draws a conclusion and foresees the future work.

II. SYSTEM REQUIREMENTS ANALYSIS

A. Application Scenarios

This system is planned to serve hundreds of families residing in the city for health monitoring via Internet, and will support thousands of potential customers including physicians and whoever care about health of themselves in the near future. So, higher concurrent transactions and shorter response time are the requirements to our system. Certainly, visualization of the results is also compute-intensive tasks usually with large-scale graphic drawing, which need to be stored and accessed in an efficient way to provide instant services to individuals. The multimodal and non-stationary characteristics of special medical signals, such as electrocardiogram (ECG), photoplethysmograph (PPG), and electroencephalograph (EEG), are very trivial for storage and may lead to a synchronous problem on GlusterFS. Hence, these large numbers of semi-structured or unstructured health records, as well as accompanied with trivial files, are all adapted to NoSQL databases instead of traditional relational ones. These fundamental characteristics are very different from the features of grid computing which aimed at special applications and difficult to operate for unprofessional users (e.g., scientific exploration projects). Furthermore, the method of service delivery is another important factor which affects the usability of the system. In addition to a mobile phone as an appropriate front-end device, TV set is another friendly interface for aging people. So, with the assistance of set-top box, services would be delivered through a TV cable ubiquitously.

B. Overview of the Data Flow

There are two main activities of users: upload the raw data and browse the results of diagnosis. Upload activities are divided into two steps: the first is to transmit physiological data, which are collected by sensors, to the gateway through short-distance transmission by a wired USB or a wireless Bluetooth. The second is to deliver the data from the gateway to the cloud servers through IP networks. Set-top box (for USB) and mobile phone (for Bluetooth) play the important role of gateway to relay data, respectively. We have developed three types of interfaces to users including TV cable, Wi-Fi/Ethernet, and cellular networks (2G/3G). All of these forms are interacted with cloud servers via Internet at the end. Browsing the results on a TV or a mobile phone is very straightforward. As aforementioned, all of the physiological data should be collected in a distributed database cluster as the reservoir for data mining. Although the data may possibly be lost by accident (e.g., power failure) during operation, it must be restored from the databases in cloud. Therefore, to avoid a single point of failure, we utilize a primary/secondary scheme provided by a database to maintain enough replicas of data. Keep-alive mechanism is also adopted in this scheme and switch back and forth if the heart-beat signal is missing between primary ones and secondary ones. Fig. 1 shows the overview of our ubiquitous healthcare service system.

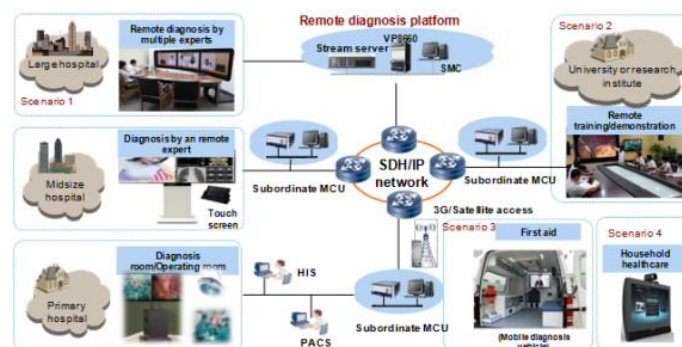


Fig. 1. System overview of ubiquitous healthcare service.

III. SYSTEM ARCHITECTURE AND IMPLEMENTATION

Considering the application requirements and the data of the inconsistent format, we improved our previous work and proposed a hybrid model which simplified the one derived from the National Institute of Standards and Technology (NIST) and Jericho models. Being everything can be treated as a service (XaaS), we can also choose component as a service to take very efficient way to develop this specific private cloud architecture. The philosophy of this private healthcare cloud inherited the software as a service (SaaS) of the NIST model and introduced the insource/outsource concept into our cloud platform development. Outsource products include open-source components and commercial software, whereas the core data mining models are developed by ourselves as an insource product. The whole cloud platform comprised of several building blocks according to functionality requirements. Fig. 2 shows the six-layer architecture of the healthcare cloud platform while each layer's content and function are interpreted as follows.

- 1) Service interaction is a top layer where users can interact with the terminals such as 3G mobile phone and set-top box, and browser on a computer, etc., to collect and upload original physiological data, as well as download the analysis results from the cloud servers.
- 2) Service presentation can be regarded as an interface with various kinds of services such as wireless application protocol (WAP), Web, image provider, etc. In addition, the node balancer also works on this layer, which can provide high concurrency capability. Some run-time information is also reserved at this level.
- 3) Session cache stores the user sessions on one hand, which maintains the authentication and certification status at the first time when the user accesses the services, i.e., service interaction through the presentation layer. On the other hand, memory cache is adopted to expedite the result data access. In fact, this session's architecture is a big hashed key/value table.
- 4) Cloud engine is a dispatcher to cooperate the components with each other to make the cloud running by using message-driven mode via MQ. The functionalities of the management for the queue are provided by this layer, which is regarded as a critical core for scheduling tasks.

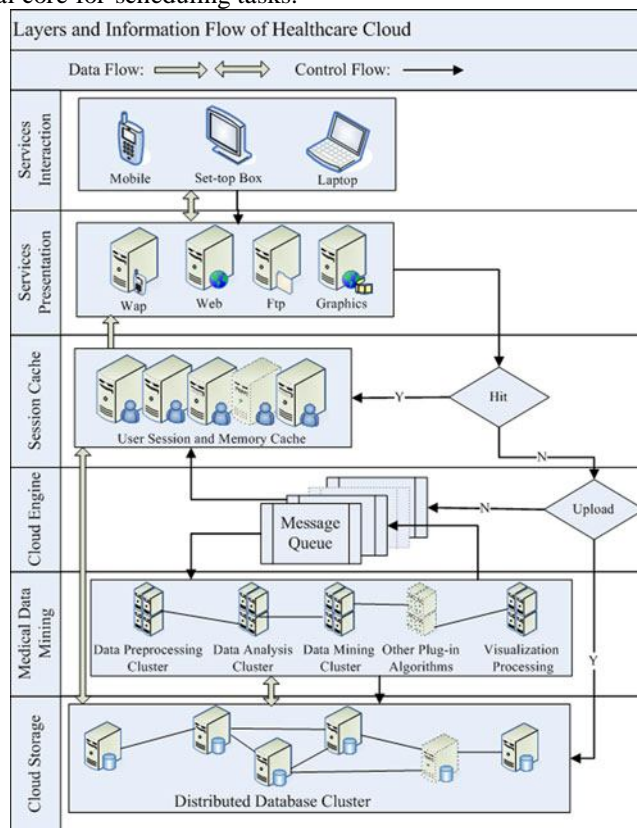


Fig. 2. Six layers of a healthcare cloud platform.

- 5) Medical data mining is clusters of algorithms, including data pre-processing, data analysis, mining algorithms, and visualization processing. This layer can handle the data transmitted from the front end and generate the results back to database. Other algorithms can also be easily plugged in if needed.



6) Cloud storage provides data resources for the entire health cloud platform, including user information, vital signs, health records, and graphic data for processing. Physiological data collected from body area networks and massive graphic data for distributed processing of such data-intensive tasks are the primary contents. Using the distributed file storage system, like Hadoop Distributed File System with semistructural database, can provide more extensible and efficient access. Moreover, sharding is another major characteristic of this distributed database to gain higher availability. Redundancy among these pieces of shards and different views of the same data provide the consistency to a large extent. This mechanism can guarantee the integration of global data and transparency to users.

There are two key components addressed here: the first one is MQ, which is a kind of component that provides asynchronous communication protocol to achieve the independence between message sender and receiver. Queue is utilized for storing the

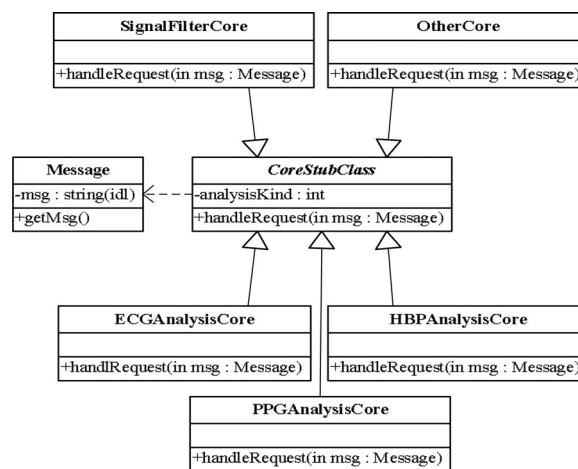


Fig. 3. Plug-in algorithm framework.

event message generated by a sender (publisher), and all of the listeners (subscriber) who are interested in this kind of event can fetch this message to process a predefined routine. All communication parties should follow the same protocol (e.g., advanced message queue protocol), and utilize available MQ application program interface to operate the queue. It is not necessary that two parties must know the existence of each other. After a message is inserted into a queue, it will be preserved and not deleted in the MQ until the corresponding subscriber reconnects to the system. Messages can be exchanged on the level of process, application, or even intercloud. Queue resides in cloud just as an engine to drive the system. Addition, different message head identifies the analysis algorithm and indicates the subsequent behaviours of cloud. The second one is the plug-in algorithm framework developed with respect to the extensibility of various services, which is based on the publish/subscribe mechanism to provide customized functions conveniently, not only for healthcare but also for other services. The whole system can reduce the module coupling by adopting this algorithm. For instance, various data mining algorithms, such as analysis of peripheral vascular function, instantaneous heart rate, and chaotic characteristics of power spectral density, etc., should be adopted to perform automatically according to different analyzed signals. Every different function can subscribe the different theme of message, which is classified by a message head. Accordingly, we designed an abstract core class named *CoreStubClass*, including a private attribute *analysisKind*, and an abstract method *handleRequest(int)*. This class communicates with *Message* via MQ, and the other kinds of concrete implementation classes extended the *CoreInterface*, e.g., *SignalFilterCore*, *ECGAnalysisCore*, and other data mining classes. Fig. 3 shows the class framework of the plug-in algorithm.

IV. TEST RESULTS AND ANALYSIS

As a result, Fig. 4 shows the upload interface on a TV while collecting user's psychological signals for 2 min. Results of



Fig. 4. Presentation of a user's upload interface on a TV.

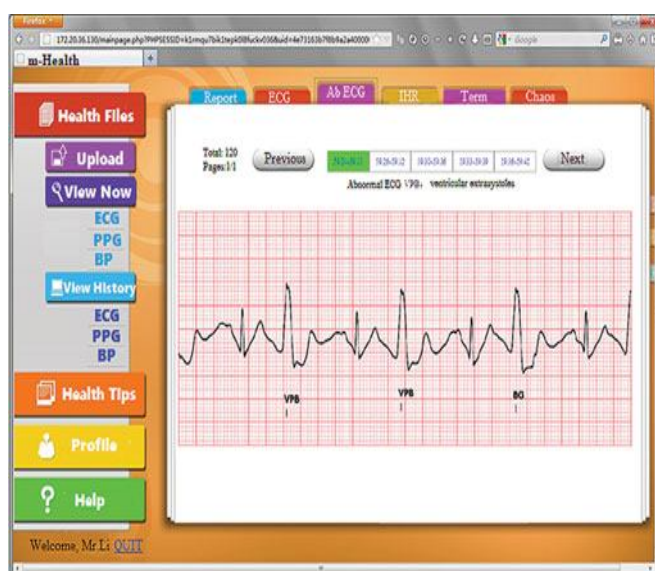


Fig. 5. Presentation of a user's abnormal ECG on a browser.

TABLE I
OVERALL STATISTICS OF UPLOADING AND BROWSING ACTIVITIES

Activity	Name	Highest Rate (Bytes/sec)	Mean Time (sec)	Count
Uploading	connect	118.9	0.12	32225
	page	243.6	0.40	64771
	request	604.7	0.16	163011
	session	103.6	1.22	28170
Browsing	connect	121.9	0.30	23633
	page	896.4	0.25	206381
	request	2611.1	0.08	573683
	session	98.4	14.98	23233

Note. *Connect*: duration of the connection establishment; *page*: response time for each set of requests; *request*: response time for each request; *session*: duration of a user's session.

analysis are viewed by a Web browser as shown in Fig. 5, which can also be shown on a TV. To perform stress and load tests on our system, we designed two test cases to verify the capabilities of cloud platform: simulating initial 100 users for *uploading* activity lasting for 10 min, as well as 300 users for *browsing* lasting for 30 min. Table I shows the overall



of the testing statistics. Use four servers with the same configuration (2.93-GHz Intel Xeon CPU with 4G RAM) as front-end clusters to provide visualization services under the *Tsung* testing environment. *Tsung* can generate many sessions and make concurrent requests to a server automatically. It starts with the initial user count and reports the performance in detail when testing is over. A new

TABLE II
STATISTICS OF NETWORK TRAFFIC AND PERFORMANCE

Activity	Receive/Sent	Bandwidth (KB/sec)	Total (MBytes)	Users_count
<i>Uploading</i>	size_rcv	835.84	224.58	28170
	size_sent	2999.04	803.37	
<i>Browsing</i>	size_rcv	26277.12	6113.28	23235
	size_sent	522.24	108.54	

Note. Size_rcv: size of responses; size_sent: size of requests.

TABLE III
STATISTICS FOR READING PERFORMANCE OF AMAZON S3 [19]

Page Size (KB)	Response. Time (sec)	Bandwidth (KB/sec)
10	0.14	71.4
100	0.45	222.2
1000	3.87	258.4

HTTP *request* generated within a given interval of 0.02 s represents that a new user connection happened, and a *session* will be created according to the probability presented in the testing configuration file. The mean response time and count (for page, request, etc.) for the whole test are computed, which generated 16 3011 and 57 3683 concurrent requests for the two activities, respectively, as shown in *Mean Time* and *Count* columns. Response time less than 1s is tolerable for user experience. The actual number of concurrent users is nearly 30 000 during the experiment as the *Users_count* column of Table II shows. It is acceptable to us. By comparison, Table III shows the reading performance of Amazon's simple storage service (S3) with different page sizes (writing data to S3 takes about three times as long as reading data). Ignoring the slight differences of hardware, the usability of our system is satisfied in terms of *Bandwidth*, which is at least double of that in S3 (compare the last row of Table II with that of Table III). *Response Time* is also better obviously. Overall, it demonstrates that this document-oriented cloud storage architecture is more appropriate for the environment of health-care services with a large number of trivial files, rather than nondocument-oriented ones such as S3. Furthermore, due to platform capabilities of linear extendibility, we simply increase the number of servers if there are more potential users' requests in the future.

V. CONCLUSIONS

In this paper, we propose a six-layer architecture of the private cloud platform associated with the technologies such as MQ, load balance, plug-in algorithm, and cloud storage. This platform can manage the semistructure, unstructured, and heterogeneous physiological signal data very well. Testing results show that it can cope with the issues about high concurrent requests in ubiquitous healthcare service and dispose of tasks for massive physiological signals analysis rapidly, as well as owning robust, instant, and efficient features. Particularly, it demands not much for hardware, and most of those integrated components such as NoSQL databases and MQ are open-sourced and free, whereas the commercial portfolio of software is always very expensive. So, it is also an appropriate low-cost solution with respect to the effectiveness. In the future, we will further explore the security issues and add novel trustworthy module to protect user's privacy.

REFERENCES

- [1] H. Ding, J. He, and W. Wang, "The sub-health evaluation based on the modern diagnostic technique of traditional chinese medicine," in *Proc. Workshop Edu. Technol. Comput. Sci.*, Shanghai, China, Mar. 7–8, 2009, pp. 269–273.
- [2] S.-H. Leel, J. H. Song, J.-H. Ye, H. J. Lee, B.-K. Yi, and I. K. Kim, "SOA-based integrated pervasive personal health management system using PHDs," in *Proc. 4th Int. Conf. Pervasive Comput. Technol. Healthcare*, Munich, Germany, Mar. 2010, pp. 1–4.
- [3] P. Kulkarni and Y. Ozturk, "mPHASIS—Mobile patient healthcare and sensor information system," *J. Netw. Comput. Appl.*, vol. 34, pp. 402–417, 2011.
- [4] N. Botts, B. Thoms, A. Noamani, and T. A. Horan, "Cloud computing architectures for the underserved: Public health cyberinfrastructures through



- a network of HealthATMs,” in *Proc. 43rd Syst. Sci. Conf.*, HI, Jan. 2010, pp. 1–10.
- [5] Z. Rashid, U. Farooq, J.-K. Jang, and S.-H. Park, “Cloud computing aware ubiquitous health care system,” in *Proc. 3rd E-Health Bioeng. Conf.*, Iasi, Romania, Nov. 2011, pp. 1–4.
- [6] C. O. Rolim, F. L. Koch, C. B. Westphall, J. Werner, A. Fracalossi, and G. S. Salvador, “A cloud computing solution for patient’s data collection in health care institutions,” in *Proc. 2nd eHealth, Telemed., Social Med. Conf.*, St. Maarten, The Netherlands, Feb. 2010, pp. 95–99.
- [7] B. Jit, J. Maniyeri, K. Gopalakrishnan, S. Louis, P. J. Eugene, H. N. Palit, Y. S. Foo, L. S. Lau, and X. Li, “Processing of wearable sensor data on the cloud—A step towards scaling of continuous monitoring of health and well-being,” in *Proc 32nd Conf. Eng. Med. Biol. Soc. (EMBS)*, Buenos Aires, Argentina, Aug./Sep. 2010, pp. 3860–3863.
- [8] M. Stonebraker, “SQL databases v. NoSQL databases,” *Commun. ACM*, vol. 53, no. 4, pp. 10–11, Apr. 2010.
- [9] F. Miao, X. L. Miao, W. H. Shangguan, and Y. Li, “MobiHealthcare system: Body sensor network based m-health system for healthcare application,” *E-Health Telecommun. Syst. and Netw.*, vol. 1, no. 1, pp. 12–18, 2012.
- [10] A. Samba, “Logical data models for cloud computing architectures,” *IEEE IT Prof.*, vol. 14, no. 1, pp. 19–26, Jan./Feb. 2012.
- [11] MQ API Guide. (Mar. 2012). [Online]. Available: <http://www.rabbitmq.com/api-guide.html>
- [12] Tsung. (Apr. 2012). [Online]. Available: http://tsung.erlang-projects.org/user_manual.html