Access Control of Cloud Computing Using Rapid Face and Fingerprint Identification

Bao Rong Chang
Dept. of Computer Science and Information Engineering
National University of Kaohsiung, Taiwan
brchang@nuk.edu.tw

Hsiu-Fen Tsai
Dept. of Marketing Management
Sue-Te University, Taiwan
soenfen@mail.stu.edu.tw

Chi-Ming Chen
Dept. of Computer Science and Information Engineering
National University of Kaohsiung, Taiwan
gcsul0725@gmail.com

Chien-Feng Huang
Dept. of Computer Science and Information Engineering
National University of Kaohsiung, Taiwan
cfhuang15@nuk.edu.tw

***Abstract*—In this paper, a fast access control applied to Hadoop cloud computing using the fingerprint identification and face recognition is introduced. This study employed a way of low-capacity Linux embedded platform linked to Hadoop via Ethernet, WiFi or 3G. At client side, JamVM virtual machine is utilized to form the J2ME environment, and GNU Classpath is viewed as Java Class Libraries. In order to verify effectiveness and efficiency in access control for cloud computing, the rapid face and fingerprint identification has been accomplished successfully within 2.2 seconds to exactly perform the subject identity under the cross-examination. The experimental results show that the one we proposed outperforms two alternative methods.**

Keywords- Hadoop cloud computing; Access control; Fingerprint identification; Face recognition; Linux Embedded Platform; JamVM Virtual Machine.

I. INTRODUCTION

Cloud computing is an emerging and increasingly popular computing paradigm, which provides the users massive computing, storage, and software resources on demand. How to program the efficient distributed parallel applications is complex and difficult. How to dispatch a large scale task executed in cloud computing environments is challenging as well. This is because applications running in cloud computing environment need to conquer some problems such as the network bandwidth, the faults tolerance, and the heterogeneity. Different solutions to programming and tasking in cloud computing environments have been proposed by several independent software vendors (ISV), each has its own strengths and weakness. Cloud computing service providers also have their own programming model and APIs to be used by users. Here we will briefly introduce the different cloud computing perspectives and show our approach in this study. Cloud computing currently under development consists of three areas, massive computing, connectivity, and smart terminal. The aim of cloud computing is towards low-cost (saving you money), green energy (energy efficiency), and the ubiquitous (at any time, any place, any device to access any of the services).

Nowadays cloud computing become a popular term with high-tech concept. In fact, these technologies are not entirely new, probably inherited from the nature of "distributed computing" and "grid computing". That is, we divide a large work into small pieces because it is of the incompetence in a single computer, and then these pieces are carried out by a number of computers. After that, compiling their findings to complete the work is done. In addition, we have devoted to connect a variety of different platforms, different architectures, different levels of the computer through the network such that all of computers are cooperated with each other or network makes the computer to do services more far and wide in the cyber space, but the difference is that "cloud computing" has emphasized, even existing the limited resources in a local context, to make use of the Internet to access remote computing resources.

Cloud computing is divided into two categories, namely "cloud services" and "cloud technology". "Cloud services" is achieved through the network connection to the remote service. Such services provide users installation and use a variety of operating systems, for example Amazon Web Services (CE2 and S3) services. This type of cloud computing can be viewed as the concepts: "Infrastructure as a Service" (IaaS) "Storage as a service" (StaaS), respectively. Both of them are derived from the concept of "Software as a Service" (SaaS) that is the biggest area for cloud services in demand, while "Platform as a Service" (PaaS) concept is an alternative for cloud computing service. Using these services, users can even simply to rely on a cell phone or thin client to do many of things that can only be done on a personal computer in the past, which means that cloud computing is universal. The "cloud technology" is aimed at the use of virtualization and automation technologies to create and spread computer in a variety of computing resources. This type can be considered as traditional data centers (Data Center) extension; it does not require external resources provided by third parties and can be utilized throughout the company's internal systems, indicating that cloud computing also has the specific expertise. Currently on the market the most popular cloud computing services are divided into public clouds, private clouds, community/open clouds, and hybrid clouds, where Goggle App Eng [1], Amazon Web Services [2], Microsoft Azure [3] - the public cloud; IBM Blue Cloud [4] - the private cloud; Open Nebula [5], Eucalyptus [6], Yahoo Hadoop [7] and the NCDM Sector/Sphere [8] - open cloud; IBM Blue Cloud [4] - hybrid cloud.

II. MOTIVATION AND BACKGROUND

From the green energy point of view, cloud computing means the following four characteristics: large amounts of data, low cost, efficiency and reliability. The main purpose

978-0-7695-4606-3/11 $26.00 © 2011 IEEE
DOI 10.1109/IBICA.2011.49

of this study is to build a Hadoop cloud computing [9]. In order to verify the cloud system effectiveness and efficiency in access control, the fingerprint identification and face recognition by using rapid identification have achieved in this study.

In many applications, embedded devices often require huge computing power and storage space, just the opposite of the hardware of embedded devices. Thus the only way to achieve this goal is that it must be structured in the "cloud computing" and operated in "cloud services". The idea is how to use the limited resources of embedded devices to achieve the "cloud computing", in addition to using the wired Ethernet connection, and further use of wireless mobile devices IEEE802.3b / g or 3G to connect, as shown in Fig. 1.

First, we use the standard J2ME [10] environment for embedded devices, where JamVM [11] virtual machine is employed to achieve J2ME environment and GNU Classpath [12] is used as the Java Class Libraries. In order to reduce the amount of data transmission, the acquisition of information processed is done slightly at the front-end embedded devices and then processed data through the network is uploaded to the back-end, private small-cloud computing. After the processing at the back-end is completed, the results sent back to the front-end embedded devices. As shown in Fig.2, an open source Hadoop packages is utilized to establish the private cloud computing easily; in such a way that we can focus on installing the back-end cluster controllers and cloud controller in order to build a private small-cloud computing.

An embedded platform in conjunction with a cloud computing environment is applied to testing the capabilities of fingerprint identification and facial recognition as the access control system. The basic structure of Hadoop cloud computing is developed and has been deployed as well. We will then test the performance of the embedded platform operating in cloud computing to check whether or not it can achieve immediate and effective response to required functions. Meanwhile, we continue to monitor the online operation and evaluate system performance in statistics, such as the number of files, file size, the total process of MB, the number of tasks on each node, and throughput. In a cluster implementation of cloud computing, the statistical assessment by the size of each node is listed. According to the analysis of the results, we will adjust the system functions if changes are required.

III. ACCESS CONTROL IN HADOOP

A. *Deploying Hadoop Cloud Computing*

Once a Hadoop cloud computing server has been established in server site, we have to test the functionality of cloud computing in Hadoop system. As shown in Fig. 3, we first test the Hadoop Administration Interface at the website http://hadoop1:50030. Next, Task Tracker Status at http://hadoop1:50060 will be examined as shown in Fig. 4. After that we move on to check Hadoop Distributed File System (HDFS) Status, as shown in Fig. 5, at http://hadoop1:50070. In order to setup a programming environment for Python or Java, an Eclipse IDE [13] is applied to develop the application program (AP) at local site as shown in Fig. 6. It is noted that please remember to install Java JDK [14] before you setup an Eclipse IDE in local site. If AP has been done and is waiting for dispatch on Hadoop cloud computing server, we deploy this AP via the path of LAN or WiFi. Finally, as shown in Fig. 7, we take a look at HBase in Hadoop server to make sure that the cloud computing is ready for the task.

B. *Establishing Thin Client*

In terms of thin client, JamVM is treated as the framework of programming development; however the virtual machine JamVM has no way to perform the drawing even through their core directly, and thus it must call other graphics library to achieve the drawing performance. Here some options we have are available, for example, GTK+DirectFB, GTK+X11, QT/Embedded, and so on, as shown in Figs. 8, 9, and 10 below. The problem we encountered is that GTK needs a few packages to work together required many steps for installation, compiling different packages to build system is also difficult, and it is often time-consuming for the integration of a few packages no guarantee to complete the work. Therefore this study has chosen QT/Embedded framework instead of GTK series, in such a way that achieves GUI interface functions. In Fig. 11, no matter SWT or AWT in JamVM they apply Java Native Interface (JNI) to communicate C- written graphics library. Afterward QT/Embedded gets through the kernel driver to achieve graphic function as shown in Fig. 12. According to the pictures shown in Fig. 11 and Fig. 12, we can string them together to be the structure of a node device as shown in Fig. 10. This part will adopt a low-cost, low-power embedded platform to realize a thin client.

C. *Installing Access control System*

When establishing a Hadoop cloud computing is done, we will test the cloud employing an embedded platform in a cloud environment to perform fingerprint identification [15] and face recognition [16] to fulfill the access control system [17]. Meanwhile the development of basic structure and deployment for Hadoop cloud computing are valid and even more we test the service performance for an embedded platform collaborated with cloud computing, checking an immediate and effective response to client. The access control system is shown in Fig. 13.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

In order to verify the cloud system effectiveness and efficiency in access control, the experiment on fingerprint identification and face recognition by using rapid identification in Hadoop cloud computing has been done successfully within 2.2 seconds in average access identification to exactly cross-examine the subject identity. As a result the proposed Hadoop cloud computing has been performed very well when it has deployed in local area. Steps are as follows: (a) the operation for face recognition is quickly to open the video camera for the first, and then press the capture button, the program will execute binarization automatically as shown in Fig. 14; (b) the rapid fingerprint identification is first to turn on device, then press the deal button for feature extraction that reduces the amount of information as shown in Fig. 15; (c) at first the terminal device test the connection if Internet is working properly, and then we press the identify button and information sent to the cloud, and at last the cloud will return the identification results as shown in Fig. 16. Two remarkable benchmarks, FACE ID2 [18] and ZKS-F20 [19], for the performance comparison of access control are employed where Equal Error Rate (EER), for both facial recognition and fingerprint verification, and response time are two most critical measures in access control. In TABLE I, the comparison of performance with three models, FACE ID2, ZKS-F20, and CLC-IHU, is consequently shown that the method we proposed here outperforms the other alternatives due to fast response and low misclassification rate in access control.

V. CONCLUSIONS

In this study a way of low-capacity Linux embedded platform linked to Hadoop via Ethernet, WiFi or 3G. At client side, JamVM virtual machine is utilized to form the J2ME environment, and GNU Classpath is viewed as Java Class Libraries. In order to verify the cloud system effectiveness and efficiency in access control, the rapid face and fingerprint identification in Hadoop has been accomplished successfully within 2.2 seconds to exactly perform a subject identity under the cross-examination. The experimental results show that the one we proposed outperforms two alternative methods.

ACKNOWLEDGMENT

This work is fully supported by the National Science Council, Taiwan, Republic of China, under grant number: NSC99-2218--E-390-002.

REFERENCES

[1] Google App Engine, 2010. http://groups.google.com/group/google-appengine

[2] Amazon Web Services (AWS), 2010. http://aws.amazon.com/

[3] Windows Azure- A Microsoft Solution to Cloud, 2010. http://tech.cipper.com/index.php/archives/332

[4] IBM Cloud Comppputing, 2010. http://www.ibm.com/ibm/cloud/

[5] OpenNebula, 2010. http://www.opennebula.org/

[6] Eucalyptus, 2010. http://open.eucalyptus.com/

[7] Welcome to Apache Hadoop, 2010. http://hadoop.apache.org/

[8] Sector/Sphere, National Center for Data Mining, 2009. http://sector.sourceforge.net/

[9] Welcome to Apache Hadoop, 2010. http://hadoop.apache.org/

[10] Java 2 Platform, Micro Edition (J2ME), 2010. http://www.java.com/zh_TW/download/faq/whatis_j2me.xml

[11] JamVM -- A compact Java Virtual Machine, 2010. http://jamvm.sourceforge.net/

[12] GNU Classpath , GNU Classpath, Essential Libraries for Java, in 2010. http://www.gnu.org/software/classpath/

[13] Eclipse Summit, 2011. http://www.eclipse.org/dowload/

[14] Java JDK, 2011. http://www.oracle.com/technetwork/java/javase/downloads/jdk6-jsp-136632.html

[15] VeriFinger SDK, Neuro Technology, 2010. http://www.neurotechnology.com/verifinger.html

[16] VeriLook SDK, Neuro Technology, 2010. http://www.neurotechnology.com/verilook.html

[17] opencv (open source), 2010. http://www.opencv.org.cn/index.php?title=%E9%A6%96%E9%A1%B5&variant=zh-tw

[18] FACE ID2, fingertecusa, 2011. http://fingertecusa.com/face-recognition-model-c-53/face-id-2-p-92

[19] ZKS-iClock 7 & ZKS-F20, ZKS Group, 2011. http://lancyzks.bossgoo.com/product-Video-Door-Phone/ZKS-iClock-7Fingerprint-Time-Attendance-Access-Control-933894.html; http://lancyzks.bossgoo.com/product-Video-Door-Phone/ZKS-F20-STANDALONE-FACE-RECOGNITION-ACCESS-SYSTEM-933839.html

TABLE I
The Performance Comparison of Access control

Performance	FACE ID2	ZKS F20	CLC-IHU
Equal Error Rate (EER)	<0.1%	<0.01%	<0.01%
Face/Fingerprint Image Capture	<1 sec	<1 sec	<0.5 sec
Data Transmission	<1 sec	<0.5 sec	<0.5 sec
Subject Identity	<3 sec	<1.7 sec	<0.7 sec
Result Reply	<0.5 sec	<0.5sec	<0.5 sec
ResponseTime at Thin Client	<5.5 sec	<3.7 sec	<2.2 sec

Fig. 1. Cloud computing server connected with mobile device, PC, and notebook.

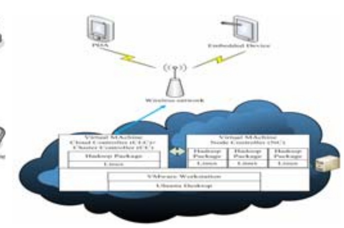


Figure 2. Hadoop server linked to mobile devices over WiFi.

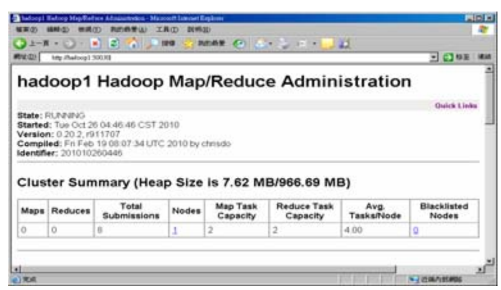


Fig. 3. Testing Hadoop Administration IF at http://hadoop1:50030.

Fig. 4. Testing Task Tracker Status at http://hadoop1:50060.

Fig. 5. Testing HDFS Status at http://hadoop1:50070.

Fig. 6. A Snapshot of Eclipse IDE.

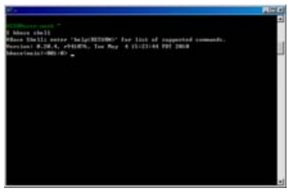


Fig. 7. Key in commend "hbase shell" under Linux Environment.

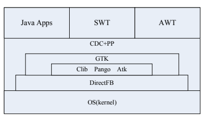


Fig. 8. Terminal node with GTK + DirectFB.

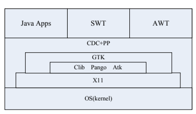


Fig. 9. Terminal node with GTK + X11.

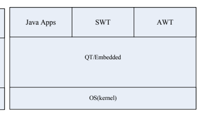


Fig. 10. Terminal node with QT / Embedded.

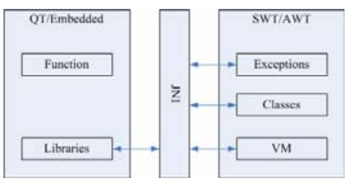


Fig. 11. Communication between SWT/AWT and QT/Embedded.

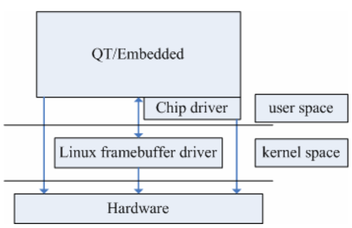
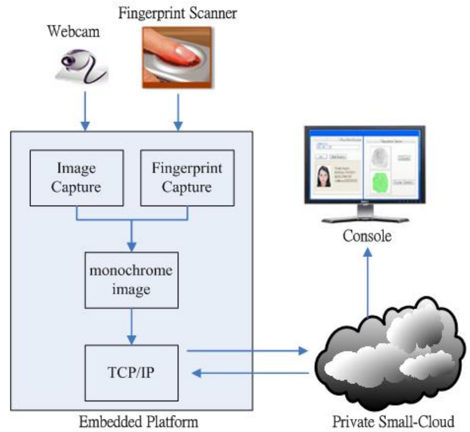


Fig. 12. QT/embedded communicates with the Linux Framebuffer.

(a)



(b)

Fig. 13. System architecture.

Fig. 14. Binarization processing automatically running in program.

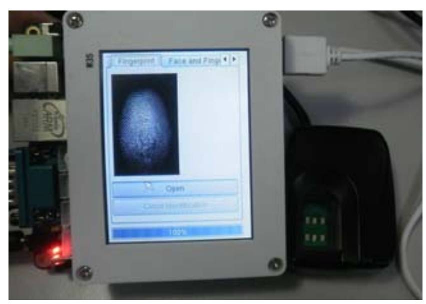


Fig. 15. Processing fingerprint features to reduce the amount of information.

Fig. 16. Information sent to the cloud and cloud returns the results of recognition to the consol.