Journal of Information Hiding and Multimedia Signal Processing ©2012 ISSN 2073-4212
Ubiquitous International Volume **3**, Number **2**, April **2012**

Rapid Access Control on Ubuntu Cloud Computing with Facial Recognition and Fingerprint Identification

Bao Rong Chang

Department of Computer Science and Information Engineering
National University of Kaohsiung
700, Kaohsiung University Rd., Nanzih District, Kaohsiung, Taiwan 811
brchang@nuk.edu.tw

Chien-Feng Huang

Department of Computer Science and Information Engineering
National University of Kaohsiung
700, Kaohsiung University Rd., Nanzih District, Kaohsiung, Taiwan 811
cfhuang15@nuk.edu.tw

Hsiu-Fen Tsai

Department of Marketing Management
Shu-Te University
59, Hun Shang Rd., Yen Chao, Kaohsiung County, Taiwan 824
soenfen@mail.stu.edu.tw

Zih-Yao Lin

Department of Computer Science and Information Engineering
National University of Kaohsiung
700, Kaohsiung University Rd., Nanzih District, Kaohsiung, Taiwan 811
qazwsxee3118@xuite.net

Received July 2011; revised April 2012

Abstract. *In this paper we have employed an open source Ubuntu Enterprise Cloud to establish a Ubuntu Cloud Computing, where a cloud controller (CLC) can been attached to a number of cluster controllers (CC), upon which we can initiate a couple of cloud services, for example, SaaS, PaaS, and/or IaaS. A cloud controller (CLC) will setup a connection to mobile devices or thin clients via wired Ethernet or wireless WiFi or 3G.network. Mobile device or thin client is designated to be a low-capacity embedded platform with Linux system in which JamVM virtual machine is used to develop the J2ME environment and GNU Classpath acts as a sort of Java Class Libraries. Finally, the rapid facial recognition and fingerprint identification accomplishes fast access control in Ubuntu Cloud Computing for preventing illegal incursions outside the cloud computing system. It takes below 2.2 seconds to finish the authentication and therefore our proposed approach outperforms two alternatives benchmarks.*
Keywords: Ubuntu Cloud Computing (UCLC), Access Control, Facial Recognition, Fingerprint Identification, JamVM Virtual Machine, GNU Classpath

---

1. **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 [1] 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" [2]. "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 (including both 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 [3] that can only be done on a personal computer in the past, which means that cloud computing is universal, especially in big data processing [4]. 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 [5]. A number of remarkable cloud computing related firm, as we know, are indicated as follows: (a) public cloud like Goggle App Engine, Amazon Web Services, and Microsoft Azure; (b) private cloud like Microsoft MCloud, IBM Blue Cloud, and SalesForce.com; (c) open cloud: Open Nebula, Eucalyptus, Apache Hadoop, and NCDM Sector/Sphere; (d) hybrid cloud like IBM Blue Cloud. Access control has been designed to realize authentication, authorization, and accounting (AAA) [6] security in cloud computing system.

Even though we got amazing amounts of benefits from cloud computing that helps companies accomplish more by breaking the physical bonds between an IT infrastructure and its users, security assurance over IaaS, PaaS, and/or SaaS is still challenging the cloud computing environment. Regard to the aspect of intruder detection, the use of Access Control systems to prevent illegal incursions outside the cloud is taken into account in this paper.

2. **Motivation And Background.** The main purpose of this study is to build a Private Small-Cloud Computing (PSCC). The idea of private small-cloud computing is based on three concepts: small clusters, virtualization, and general graphics processor [7]. This cloud system will include the use of virtualization technology by Xen [8], planning to use general-purpose graphics processors by VMGL [9], management of cluster structure by Open Nebula [10], and implementation of the cloud controller by Eucalyptus [11]. 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.

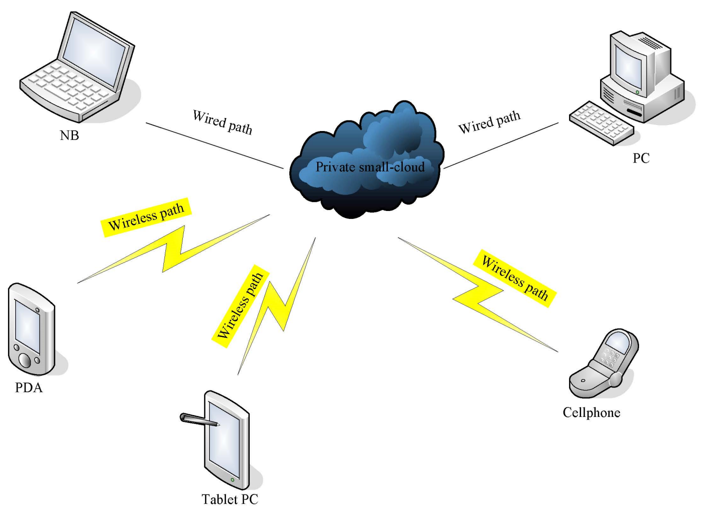


FIGURE 1. Cloud computing server connected with mobile device, PC, and notebook.

First, we use the standard J2ME [12] environment for embedded devices, where JamVM [13] virtual machine is employed to achieve J2ME environment and GNU Classpath [14] 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 package, Ubuntu Enterprise Server [15] & Ubuntu Enterprise Cloud [16], is utilized to establish the private cloud computing easily and it is called Ubuntu Cloud Computing (UCLC); 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

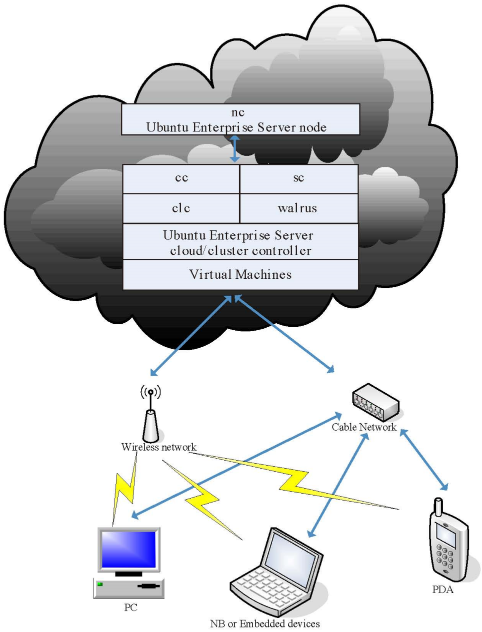


Figure 2. PSCC server linked to mobile devices over WiFi.

fingerprint identification and facial recognition served as the Access Control system. The basic structure of Ubuntu Cloud Computing (UCLC) 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.

3. ACCESS CONTROL ON CLOUD COMPUTING.

3.1. **Deploying Ubuntu Cloud Computing (UCLC).** A private small-cloud computing (PSCC) is built by packages Xen, OpenNebula, Eucalyptus, Euca2ools [17] and so on. For the purpose of simplicity, Ubuntu Enterprise Server includes all of packages we need to install a private small-cloud computing. Cloud Controller (CLC) structure as shown in Fig. 3 in which each Cluster Controller (CC) has its own OpenNebula, Zen, VMGL, Lustre [18] and hardware resources, through an unified CLC to manage all of CC. Node Controller (nc) structure as shown in Fig. 4. nc hardware resources will determine the cloud service capabilities; the more powerful nc hardware (more CPU core and more memory), the more virtual machine resources. Furthermore, according to the above Fig. 3 and Fig. 4 they provide a way to establish CLC and nc, and we can completely describe the structure of private small-cloud computing, as shown in Fig. 5. The storage server in conjunction to Eucalyptus is Walrus [19] that is a compatible storage interface like Amazon S3 storage system and can be managed through the web interface to modify it. In addition, a control unit managing the storage server is called the storage controller (sc) [20] as shown in Fig. 6. In Fig. 7, PSCC can still link to the remote cloud through the Internet such that node device gets remote cloud services via private small-cloud, such as Goggle App Eng, Amazon Web Services, Yahoo Hadoop, Microsoft Azure, IBM Blue Cloud, and NCDM Sector / Sphere.

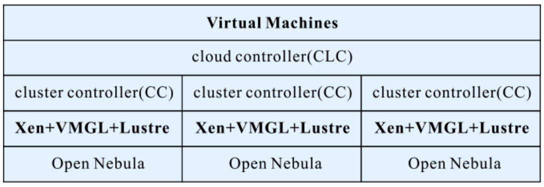


Figure 3. Cloud controller (CLC) architecture.

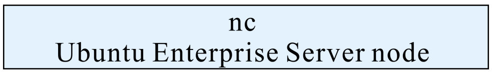


Figure 4. Node controller (nc) structure.

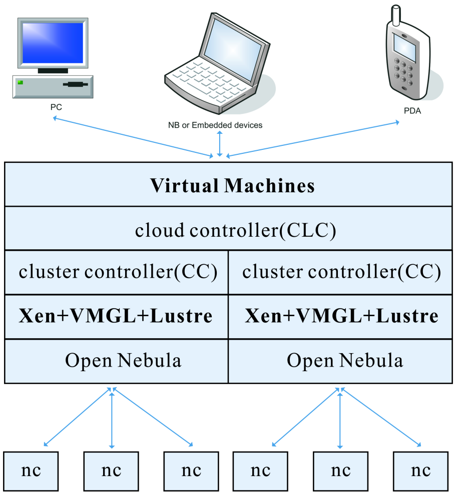


Figure 5. A complete structure of CLC + nc shows a private small-cloud computing.

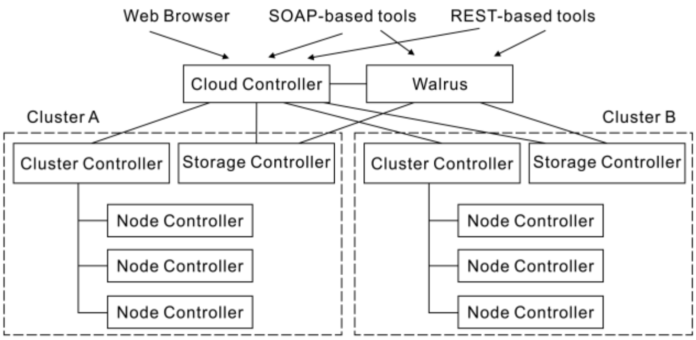


Figure 6. Advanced Eucalyptus setup along with Walrus.

3.2. **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

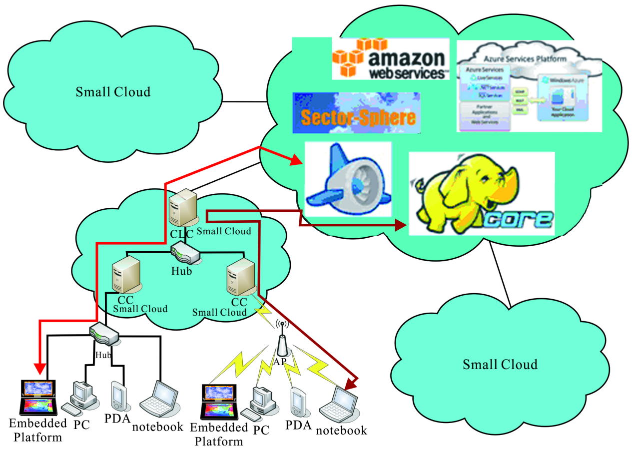


Figure 7. Various cloud services in PSCC and remote cloud.

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 act as a thin client.

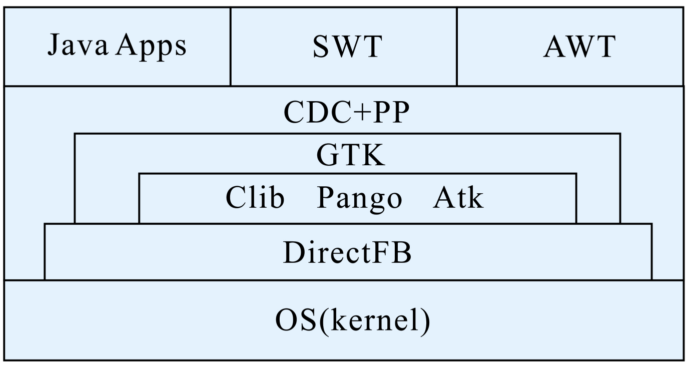


Figure 8. Terminal node with GTK + DirectFB.

subsectionInstalling Access Control SystemFor AAA security, multibiometrics-based verification systems use two or more classifiers pertaining to the same biometric modality or different biometric modalities [21]. The investigation in [22] has designed a serial fusion scheme for combining face and fingerprint classifiers and achieved significant reduction in verification time and the required degree of user cooperation. According to the above investigation, cloud computing has installed a multi-biometrics verification for access control that can performs rapid fingerprint identification [23] and face recognition [24] simultaneously. After we have validated the basic AAA security of cloud computing system, we will deploy the services to it from a PC. Finally, a thin client (or a handheld device) launches a request call to cloud computing for the test of system performance. Thereafter we check whether or not a quick response can be achieved. Fig. 13 illustrates the access control system in cloud computing.

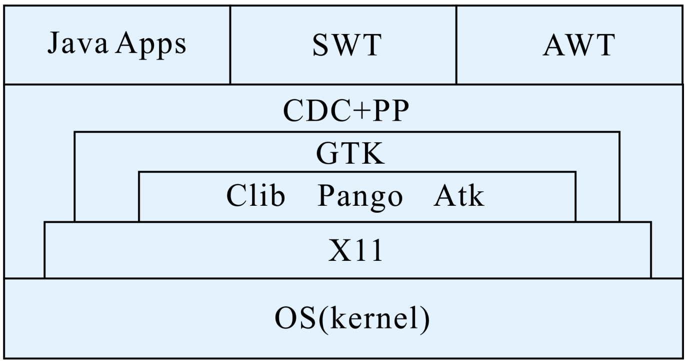


Figure 9. Terminal node with GTK + X11.

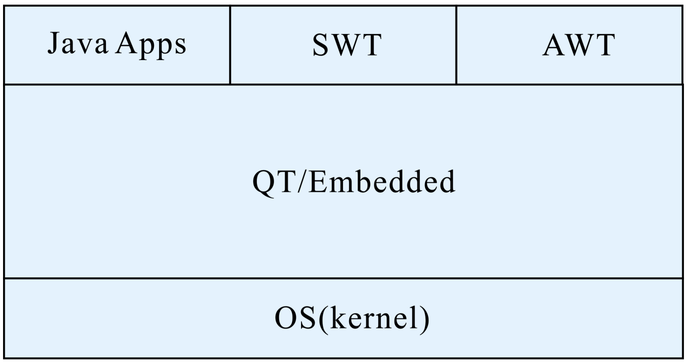


Figure 10. Terminal node with QT / Embedded.

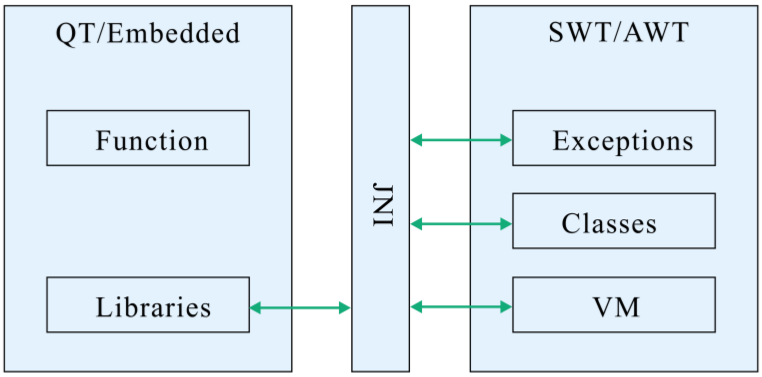


Figure 11. Communication between SWT/AWT and QT/Embedded.

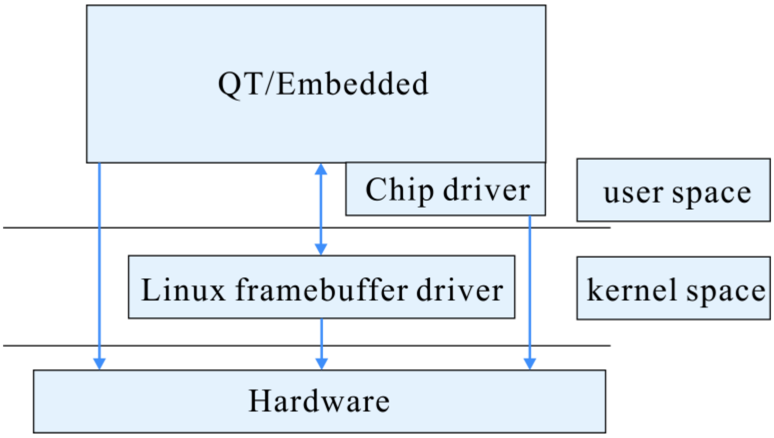
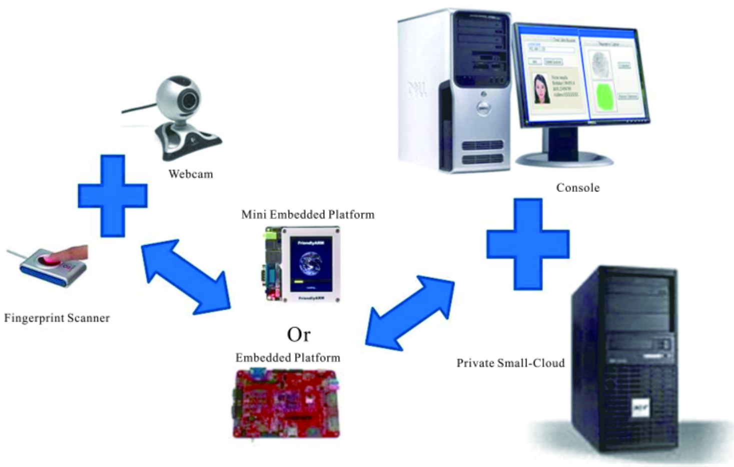
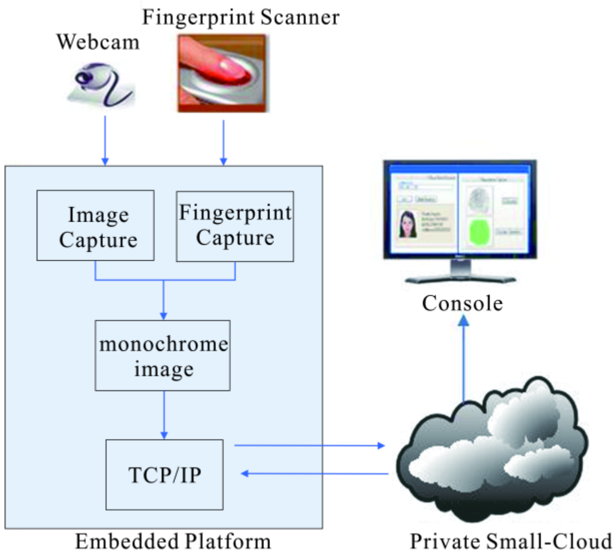


FIGURE 12. QT/embedded communicates with the Linux Framebuffer.



(a)



(b)

FIGURE 13. System architecture.

4. **Experimental Results and Discussions.** In order to deploy a minimum of cloud structure, we need at least two dedicated systems. One will be used as a cloud controller (clc), and contains the entire back-end cluster controller (cc), storage server Walrus, and the storage controller (sc). This host needs fast disks and a few fast processors to match those disks. Another one is a node controller (nc), used to perform many of the cloud entity. The host takes a lot of capacity with CPU virtualization technologies (VT) [25], a large number of CPU computing power, large memory and fast disk. Constructing a cloud computing in the following steps:

1. Installing virtual machine
   In this study we adopt the VMware-Workstation 7 to install virtual machines because VMware currently can only fully support the latest version of Ubuntu Linux and is a paid software.
2. Deploying cloud computing architecture
   To deploy cloud structure will generally need the software with Xen, OpenNebula, Eucalyptus, and Euca2ools. Since system installation needs many steps, manipulation often encounter some errors and the configuration is not easy, this study employs an open source, Ubuntu Enterprise Server Edition, because this version of the Ubuntu has included all of the above packages that are used to deploy cloud structure rapidly and easily. Ubuntu Enterprise Server Edition used to install the cloud/ cluster controller can be visited at the following webpage. ISO file booting system with the English language choice is recommended in order to avoid unusual characters input because command line style is only one input mode for the node controller as shown in Fig. 14. There is an illustration for selecting the installation type Cluster as shown in Fig. 15.
3. Installing node controller
   Before installing node controller, ISO file booting system with the English language choice is recommended in order to avoid unusual characters input because command line style is only one input mode for the node controller as shown in Fig. 16 shows. There is an illustration for selecting the installation type Node as shown in Fig. 17.
4. Setting cloud controller
   Back to the cloud controller, and executing commands to find the node controller and examining a link to node controller you created earlier as shown in Fig. 18.
5. Setting cloud user through web interface
   Before the user at client side uses the clouds, the client are required to do some of the settings in cloud controller through web interface. Login Management Interface: The default account is admin and password admin as shown in Fig. 19. Web managers can do some settings as shown in Fig. 20. Applying an account for a new user is found as shown in Fig. 21.
6. Setting embedded platform
   As the PC Java Virtual Machine (Linux Jre, X86 Jre) can not run in the embedded platform, the virtual machine for embedded platforms must be downloaded and the most useful ones are in a wide range of KVM, CVM, JamVM ... etc. So this study adopts JamVM and the following is related links in cite13.

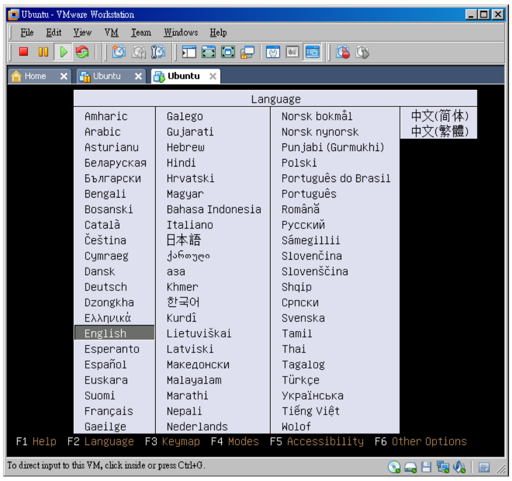


FIGURE 14. Selecting language English.

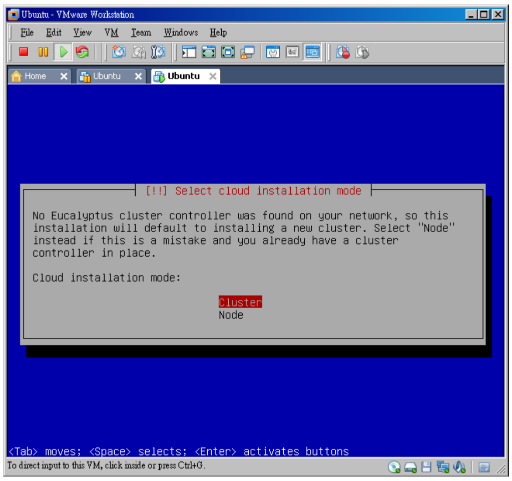


FIGURE 15. Selecting Installation Type Cluster.

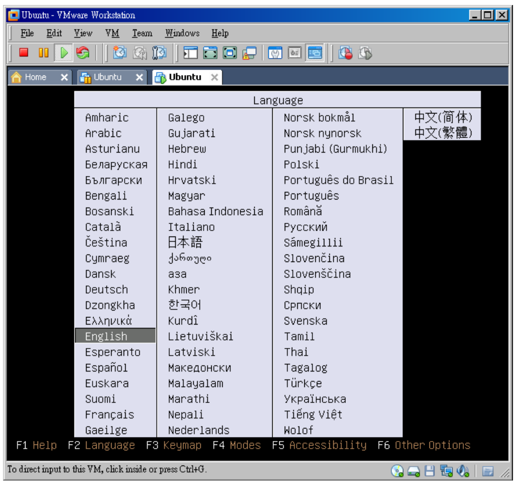


FIGURE 16. Selecting language English.

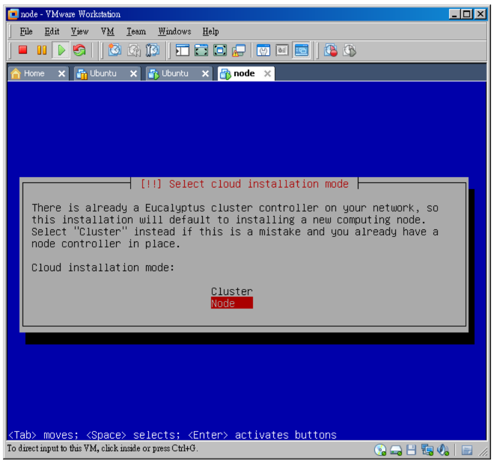


FIGURE 17. Selecingt the installation type Node.

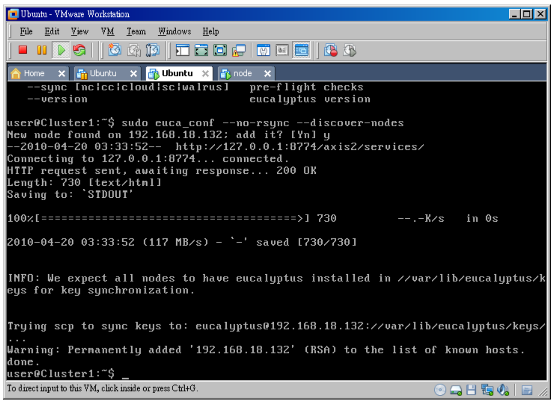


FIGURE 18. A link from cloud controller to node controller.

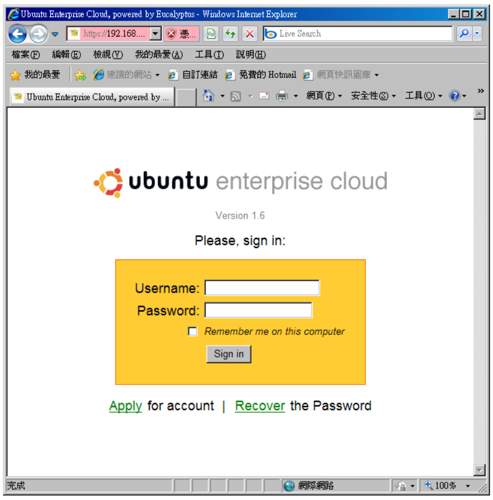


FIGURE 19. Cloud web interface.

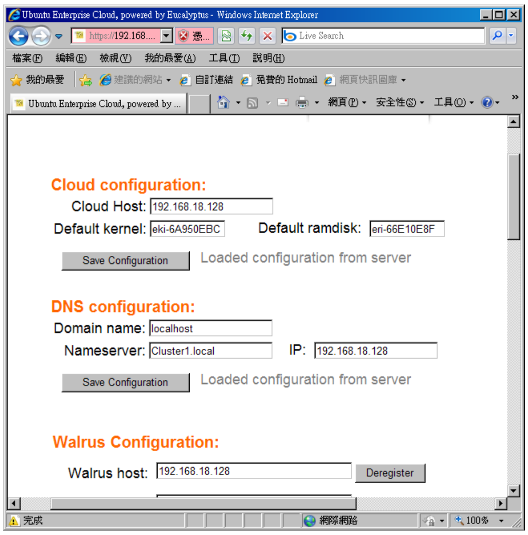


Figure 20. Setting by web.

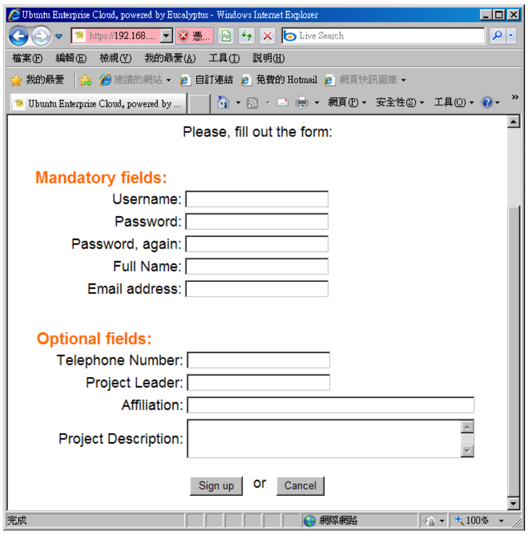


Figure 21. Applying a new account.

In order to verify the cloud system effectiveness and efficiency in Access Control for preventing illegal incursions outside the cloud, the experiment first on fingerprint identification and face recognition by using rapid identification in Ubuntu Cloud Computing (UCLC) has been done successfully to exactly authenticate the registered user less than 2.2 seconds. As a result the proposed Ubuntu Cloud Computing has been performed very well when it compared with the other alternatives. 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. 22; (b) the rapid fingerprint identification is first to turn on terminal device [26], then press the deal button for feature extraction that reduces the amount of information as shown in

Fig. 23; (c) at first the terminal device test the connection if Internet works properly, and then we press the identify button and information sent to the cloud, and at last the cloud will return the identification results to the consol and thin client as shown in Fig. 24. Biometrical Error rates are quantitative metrics of the accuracy of biometrical systems. Two accuracy errors, False Match Rate (FMR) and False Non-Match Rate (FNMR) error rates, arise from a biometrical systems. Equal Error Rate (EER) is the value where FMR and FNMR error rates are equal. So the EER is the best single description of the error rate of a biometric system.

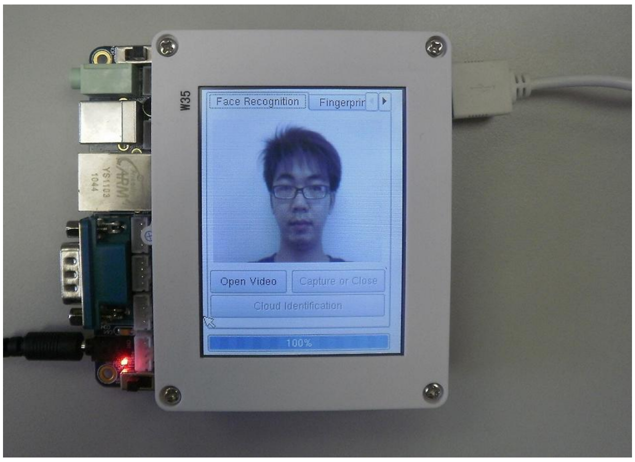


Figure 22. Cloud web interface.

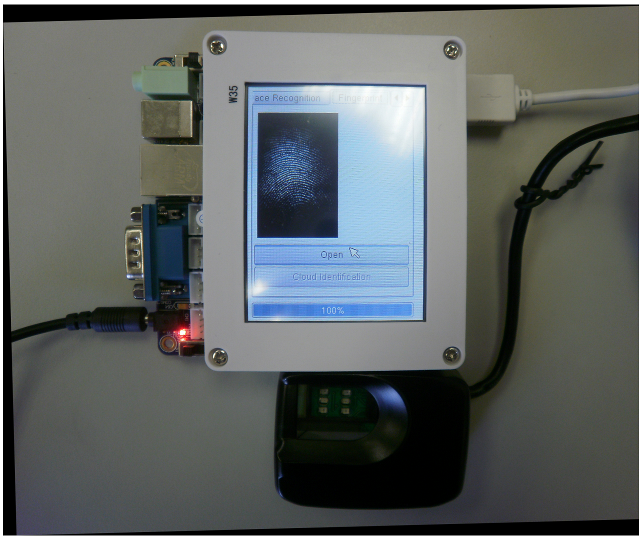


Figure 23. Setting by web.

Two remarkable benchmarks for the performance comparison of Access Control are revealed in FACE ID2 [27] and ZKS-F20 [28] where Equal Error Rate (EER), for both processes on facial recognition and fingerprint identification, and Response Time are two most concerned measures in the control of Access Control. As listed in Table 1, the comparison of performance with three models, FACE ID2, ZKS-F20, and UCLC, is consequently shown that the method we proposed here outperforms the others due to fast response and low misclassification rate in access control.

Figure 24. Applying a new account.

Table 1. The Performance Comparison of Access Control

Performance	FACE ID2	ZKS F20	UCLC
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
Authentication	< 3 sec	< 1.7 sec	< 0.7 sec
Result Reply	< 0.5 sec	< 0.5 sec	< 0.5 sec
Response Time	< 5.5 sec	< 3.7 sec	< 2.2 sec

5. **Conclusions.** In this paper we have employed Ubuntu Enterprise Cloud to establish an Ubuntu Cloud Computing upon which we can initiate a couple of cloud services, for example, SaaS, PaaS, and/or IaaS. Furthermore, the low-capacity embedded platforms with Linux system are connected to Private Small-Cloud Computing via wired Ethernet or wireless WiFi or 3G.network. Finally, the rapid facial recognition and fingerprint identification, which outperforms two alternatives benchmarks, accomplishes fast access control in Ubuntu Cloud Computing for preventing illegal incursions outside the cloud computing system.

Acknowledgment. This work is fully supported by the National Science Council, Taiwan, Republic of China, under grant number: NSC 100-2221-E-390 -011 -MY3

REFERENCES

[1] I. Foster, Y. Zhao, I. Raicu, and S. Lu, Cloud computing and grid Computing 360-degree compared, *Proc. of Workshop on Grid Computing Environments*, pp. 1-10, 2008.
[2] R. Buyyaa, C. S. Yeo, S. Venugopala, J. Broberg, and I. Brandic, Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility, *Journal of Future Generation Computer Systems*, vol. 25, no. 6, pp. 599-616, 2009.
[3] A. Kimball, S. Michels-Slettvet, and C. Bisciglia, Cluster computing for web-ccale data processing, *Proc. of the 39th SIGCSE Technical Symposium on Computer Science Education*, vol. 40, no. 1, pp. 116-120, 2008.

[4] B. R. Chang, H. F. Tsai, C. P. Young, and R. Y. Fang, Embedded system for inter-vehicle heterogeneous-wireless-based real-time multimedia streaming and video/voice over IP, *Proc. of 4th International Conference on Innovative Computing, Information and Control*, pp. 365-368, 2009.
[5] B. R. Chang, C. P. Young, H. F. Tsai, and R. Y. Fang, Timed PR-SCTP for fast voice/video over IP in wired/wireless environments, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 2, no.4 , pp. 320-331, 2011.
[6] C. Metz, Aaa protocols: authentication, authorization, and accounting for the internet, *Journal of IEEE Internet Computing*, vol. 3, no. 6, pp. 75-79, 1999.
[7] General-Purpose Computation on Graphics Processing Units, `http://gpgpu.org/`.
[8] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauery, I. Pratt, and A. Warfield, Xen and the art of virtualization *Proc. of the 9th ACM Symposium on Operating Systems Principles*, vol. 37, pp. 164-177, 2003.
[9] VMGL: VMM-Independent Graphic Acceleration, `http://www.cs.toronto.edu/~andreslc/publications/slides/Xen-Summit-2007/vmgl.pdf`.
[10] OpenNebula, `http://www.opennebula.org/`.
[11] Eucalyptus, `http://open.eucalyptus.com/`.
[12] Java 2 Platform, Micro Edition (J2ME), `http://www.java.com/zh_TW/download/faq/whatis_j2me.xml`.
[13] JamVM – A compact Java Virtual Machine, `http://jamvm.sourceforge.net/`.
[14] GNU Classpath, GNU Classpath, Essential Libraries for Java, `http://www.gnu.org/software/classpath/`.
[15] Ubuntu Enterprise Server, `http://docs.sun.com/app/docs/doc/821-1045/ggfrh?l=zh_TW&a=view`.
[16] Ubuntu Enterprise Cloud (UEC), `https://help.ubuntu.com/community/UEC`.
[17] Euca2ools User Guide, `http://open.eucalyptus.com/wiki/Euca2oolsGuide_v1.1`.
[18] Lustre a Network Clustering FS, `http://wiki.lustre.org/index.php/Main_Page`.
[19] Walrus/Eucalyptus, `http://open.eucalyptus.com/wiki/EucalyptusStorage_v1.4`.
[20] SC/Walrus/Eucalyptus, `http://open.eucalyptus.com/wiki/EucalyptusAdvanced_v1.6`.
[21] H. S. Bhatt, S. Bharadwaj, M. Vatsa, R. Singh, A. Ross, and A. Noore, A framework for quality-based biometric classifier selection, *Proc. of International Joint Conference on Biometrics*, pp. 1-7, 2011.
[22] G. Marcialis, F. Roli, and L. Didaci, Personal identity verification by serial fusion of fingerprint and face matchers, *Journal of Pattern Recognition*, vol. 42, no. 11, pp. 2807-2817, 2009.
[23] Griaule fingerprint recognition software, Griaule Biometrics, `http://www.griaulebiometrics.com/en-us`.
[24] Luxand FaceSDK, Luxand Inc., `http://www.luxand.com/index.php`.
[25] V. Chaudhary, C. Minsuk, J. P. Walters, S. Guercio, and S. Gallo, A comparison of virtualization technologies for HPC, *Proc. of 22nd International Conference on Advanced Information Networking and Applications*, pp. 861-868, 2008.
[26] Mini2440 — S3C2440 ARM9 Board, FriendlyARM, `http://www.friendlyarm.net/products/mini2440`.
[27] FACE ID2, fingertecusa, `http://fingertecusa.com/face-recognition-model-c-53/face-id-2-p-92`.
[28] ZKS–iClock 7 & ZKS–F20, ZKS Group,
`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`.