The Design of Ubiquitous Learning System with Embedded Ganglia Agent

Fu-Chien Kao, Chia-Chun Hung and Ting-Hao Huang

Dept. of Computer Science and Information Engineering, DAYEH University
Changhua, 51591, Taiwan

Dept. of Computer Science and Information Engineering, DAYEH University
Changhua, 51591, Taiwan

Dept. of Computer Science and Information Engineering, DAYEH University
Changhua, 51591, Taiwan

Abstract

This research proposes a context-aware computing ubiquitous learning system architecture design. The system integrates data grid, the ability to perform context-awareness computing, and embedded Ganglia Agent design, structuring an architecture that is able to perform context awareness mobile network, creating a ubiquitous learning environment. The embedded Ganglia Agent could provide context information on system network traffic, the CPU load of the content server, and hard disk capacity, and utilize the information to balance the load of back-end content server, providing a flexible expandability mechanism for the back-end content server. The framework of the proposed ubiquitous learning system that has context-awareness computing ability is consisted of 3 major parts: Learning Management System (LMS), Learning Content Management System (LCMS) and the embedded Ganglia Agent (EGA). LMS is responsible for managing the learners' basic personal information and studying records, LCMS is responsible for the management and storage of back-end learning contents, and EGA is responsible for the management network traffic, CPU load and hard disk capacity. With the three, the load of the back-end content server could be balanced, offering a flexible mechanism for the expansion of the server.

Keywords: *EGA, Data Grid, Context-aware, Learning Management System.*

1. Introduction

As information and internet technology advances, the popularization of Mobile Devices has made communication between people more multi and real time, also making the relationship between man and electronic devices more intimate. In the recent years, multi-media learning (e-Learning), with computers as the medium, has become an indispensable supporting method of learning. From the early periods of personal stand-alone learning, to learning using wired or wireless communication methods. The studying contents have evolved from simple text, figure, animation, to three-dimensional virtual reality. The changes of digital learning methods and its' contents have once again proved the diversity and flexibility of digital technology when applied to learning methods.

With the widespread of wireless LAN and basic construction of mobile network, learners are now able to use their mobile devices as convenient devices of learning, without the restrictions of time and space. This kind of learning is called Mobile Learning. Mobile learning utilizes mobile devices that are highly mobile and convenient as supporting tools for learning, enabling learners to acquire the wanted information anytime, anywhere, therefore accomplishing their learning goals. With the increasing diversity of mobile learning devices (including Tablet PC, Pocket PC, PDA or mobile phones), the concept of ubiquitous computing is proposed. The technology of ubiquitous computing implements wireless network technology, enable people to acquire information without limits. Meanwhile, the differences comparing to mobile computing, is that it is context aware, which can decide the most effective and appropriate environment for learning according to the user's position, information about the surroundings, and personal studying conditions. Due to the fact that the sizes of mobile devices are getting smaller and smaller while becoming more powerful, the mobile learning method combined with context awareness is rapidly spreading. This kind of learning method that implements ubiquitous computing is called Ubiquitous Learning.

Harris points out that ubiquitous learning is where mobile technology and digital learning intersects, producing a learning experience that can be had anywhere, with the help of PDA or mobile phones [1]. Lehner& Nosekabel thinks that ubiquitous learning is to provide digital information and contents without the limits of time and space, to help the learner acquire information [2]. With ubiquitous computing, the mobile device used for learning doesn't only display learning contents, but also detects the learner's surroundings, and apply it to display related information and interacts more with the learner. Ubiquitous computing integrates new types of computing such as Mobile Computing, Context Awareness, and Pervasive Computing, while hiding the technology and making it meld into our daily lives, so we could acquire all sorts of service and information without even noticing, without the help of traditional desktop computers. Ubiquitous learning has 5 features: (1) learning content (including studying content, useful resources, interaction with other learners etc); (2)The learning interface is no longer mouse and computer screen, but is provided with conversation and interaction in the surroundings; (3)The computer system that supports learning is more intricate, so the learner will not notice it, avoiding conflict with the process of studying; (4)Communication between learners and learning systems are made with mobile devices that has little burden on carrying and controlling; (5)The communication between man and machine will further more advance. In the future, a learner may not notice that he is using a computer device [3].
In order to construct an ubiquitous learning environment, the following 3 elements are required [4]:

(1) Wireless communication devices: With wireless communication technology, we can achieve ubiquitous learning that can avoid the limits of time and space, achieving interactive learning with teachers, peers and the system.
(2) Ubiquitous learning device: A ubiquitous learning device is an important element of ubiquitous learning. The device must have wireless network, small, lightweight and portable. Hand writing input is also a must for interaction.
(3) Learning activity mode: Other than wireless network and a ubiquitous device, the other important element is learning activity mode. With learning activity mode, the learner can complete a series of learning and establish his own learning style.

The reason why this research uses mobile phones as supporting tools is because not only are mobile phones light and convenient to carry. Mobile phones support wireless network abilities such as GPRS and WiFi and is also much more affordable than laptop computers so that almost everyone has one, making it suitable to be promoted to schools and users for ubiquitous learning.

2. Context-Aware Learning

Context means any information about a person, an event, time, location, things and all entities. The interaction of user, program, and environment is composed of the above mentioned elements, and is applied on the basics of context-awareness [5]. The concept of context-awareness was first proposed by Schilit and Theimer in 1994. Due to the rise of Mobile Distributed Computing, users are made able to interact with other mobile devices and computers using mobile devices, and mobile computing is no longer limited to a restricted position or environment, it can be spread to the office, home, air ports, bus stops etc [6]. For example, as a learner enters a classroom, library or meeting room, his surroundings, time, location, activity, and equipment he uses are all variables to the mobile computing device. The device has to be implemented with context-awareness in order to provide adequate service to the learner. Other than this, the mobility of the device makes the information of the current location very important, since with information of the location, the user can check out the devices and services in the area. The department of defense of USA proposed context-awareness in 1970 by utilizing GPS to acquire information about coordinates, traveling speed, time and etc. This accelerated the concept of Context-Aware Computing [7], which context-awareness program adjusts its' services according to the people, computers, the status of accessing equipment in the surroundings and adjust the services in real-time. This kind of application requires the help from sensors or mobile devices. Context-Awareness must fit the following 3 points: (1) Provide information or service to the user (2) Automatically executes the service for the user (3) Adds context information on the initial data, to enable advanced inquiries by the contents of the labels of context information. The types of context information can be sorted into Computing Context, User Context, Physical Context, and Time Context. Computing Context gathers information on network access, bandwidth and hardware; User Context gathers information on the users preference, location, people in the surrounding, the current social environment, or the user's current position or action etc. Physical Context takes care of the environment information such as lighting, sound, traffic, temperature. Time Context gathers information on time such as day, week, month, and season.

2.1 Architecture of Context-Awareness

Most of the early context-aware systems provide specific functions according to information based on the position. However, the development of context aware application is complicated work, leading to recent studies on Context-Aware Frameworks, providing all sorts of context-awareness development platforms [10]. The architecture can be divided into 5 layers, as shown in figure 1. Sensor Layer is composed of many different sensors. It not only captures the actual hardware but also capture all available information from different information sources. The types of sensors can be sorted as entity sensor, virtual sensor, and logic sensor. Raw data retrieval Layer can capture raw data of the sensor. Preprocessing Layer is not equipped in all context-awareness systems, but when the raw data is too rough, it could provide useful data. The preprocessing layer is used to interpret and inference the context data. Storage/Management Layer is to gather data in an organized way, and provide an open interface for the client to use. The client can choose to access the data with or without sync. Application Layer provides programmers a flexible and friendly way to manage and program for the needs of various contexts. The application layer can also provide tools for clients, or them to understand the condition of the sensors. An example of this is the application of context-aware outdoor environmental tutorial system [11], which utilize the GPS function of the PDA, and provide the adequate learning content in real-time by connecting to the content database after identifying the location of the client. The U-Go Net learning system [12] is a PDA that has RFID function, which can identify the client's location by wireless network technology, and provide related information of the context in real-time to the client.

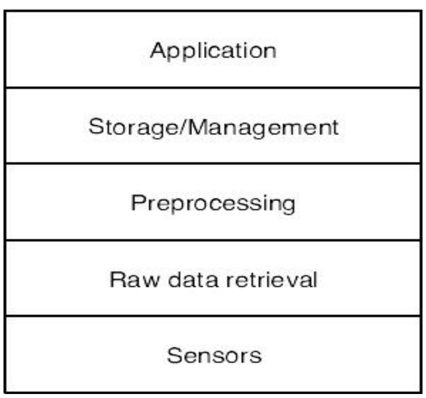

Fig. 1 Context-Aware Framework.

3. The learning architecture of ubiquitous learning with Embedded Ganglia Agent

Grid computation refers to integrating shares resources on the World Wide Web Internet by systematically regrouping such resources to achieve resource sharing and a collaborated computation. In 2002, Ian Foster presented that grids need to satisfy three concepts [13],

(1) Grids are able to integrate a variety of resources and coordinate a host of users enabling users of different environment to access resources in different environment. With that, under the grid environment, there is a need to take into account pertinent security, certification, and authorization issues.
(2) Grids need to utilize standard, open, universal protocols, and these protocols must solve the certification, resource sharing, and resource storage and retrieval problems.
(3) Grids allow the resources to be accessed and utilized to satisfy the user's varied utilization needs.

Under the foresaid concept, rules have been developed to define a virtual organization [14]. Under such virtual organization, any resource can satisfy the foresaid concept and development. In recent years, many organizations around the world have presented many types of grids, and there are two groupings when classified by the nature of problems the grids are designed to solve: (1) Computational grids and (2) Data grids, while in fact a majority of grid computing environments are a combination of the two.

(1) Computational Grids: As computational grids emphasizes on computing resource integration, thus computing the nodes in a grid has been done through clustering computers or with high-performance computers.

(2) Data Grids: Data grids integrate a virtual organization's file system, enabling the user to conduct cross-virtual organizational file storage and retrieval [15].

The main objective of data grid computing is to divide huge amounts of calculations, storage or analyzing procedures, into smaller sections, and appoint them on to idle computers on the internet, achieving super-computer class processing speed with existing hardware, meanwhile also sharing data. All this can be done by releasing the processing capability and storage capability when the computer is idle. You could also say that grid computing is integrating scattered processing resources by implementing the correct routine, network bandwidth, and storage ability, to create a powerful virtual computer system. Grid computing can be seen as requesting resources to many computers in the network at the same time, using it to process problems that require a significant amount of time or data. Many organizations around the work have proposed many kinds of grids. They can be assorted into two categories, one is Computational Grid, and the other is Data Grid [13]. Computational grid places its focus on sharing the calculation resources to achieve fast data processing. Data grid focuses on sharing data, storage of data, quicker access to data and the transferring of scattered data.

Of the many organizations researching on grid, the Globus Alliance's Globus ToolKit is the more mature one. Globus was developed by the Argonne National Research Lab near Chicago, and there are 12 universities and research facilities participating in this project. Globus has fulfilled the theories and concepts of planning a grid, and researches on source management, data management, information management and grid security. The outcomes are now widely applied on many large scale grid platforms. This research utilizes RFT (Reliable File Transfer) for data management. The RFT works under the grid environment of GT4.0. It is set up on the GridFTP transfer mechanism, and therefore has the features of GridFTP, combined with GSI's accountable secure transfer, and uses the HTTP protocol and Third Party Transfer via standard SOAP information, to provide data about TCP buffer size, parallel data flow, confirmation of data routine and other parameter setting interface; The Third Party Transfer system can control two remote computers to proceed with data duplication or deletion, used with the data base to achieve monitoring of the process of transfer, as shown in figure 2 [15].

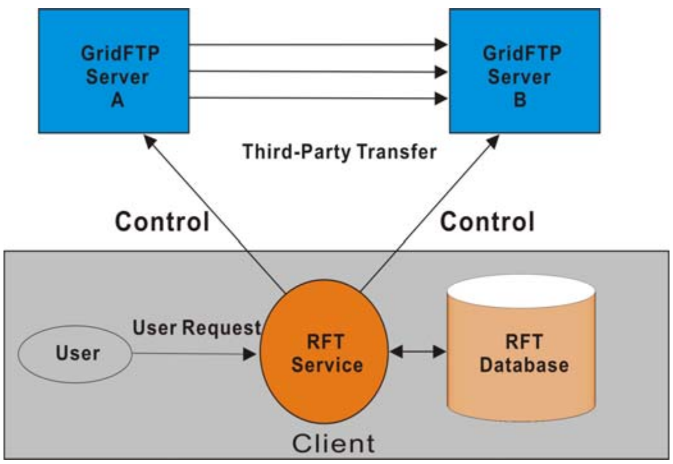

Fig. 2 RFT Supports Third Party Transfer and combines data base storage transfer condition.

3.1 Ganglia Agent

Ganglia is an operation system with dispersion type context-awareness function. The information includes CPU, Memory, hard disk usage, I/O load and net traffic etc; This context-awareness software is often used with grid architectures, to monitor high spec cluster systems. Ganglia executes Gmond Daemon at every node, and Gmond will monitor the condition of the system. These daemons execute data exchange and communication with UDP broadcasting, and send all data via a tree map structure for Gmetad to organize all the monitored data. Cluster computers communicate with each other by XML via TCP. As figure 3 indicates, the whole architecture of Ganglia includes 5 cores that are Ganglia Monitoring Daemon (Gmond)、Ganglia Meta Daemon (Gmetad)、Gmetric、Gstat and Ganglia PHP Web Frontend. Gmond is a multi-threading daemon that monitors system data of nodes, while broadcasts by UDP to gather system data of other nodes, and sends all system status information in XML format to Gmetad; Gmetad will regularly require its' child nodes to provide context monitoring data, and analyzes the XML files collected, then send the organized context data to the client in XML format; Gmetric is a tool customized for clients. The client can input commands to request Gmond to combine the requested monitoring data; Gstat is a command line tool provided by Ganglia to acquire all the node's context data. The context data provided are those such as the number of CPUs of the cluster computer and its CPU load etc; Ganglia PHP Web Frontend provides a graphical monitoring interface. The client can monitor the contexts of all the cluster system's nodes. The XML monitoring data collected by this research is transformed in to a data chart that can be shown on a web page, as shown in figure 4.

This research integrates RFT components with embedded Ganglia Agent (EGA), and proposes a ubiquitous learning system architecture that is context-aware. The proposed system architecture is as shown in figure 5. The EGA provided could analyze the XML files the Gmetad has received, and acquire the network load context information of the LCMS server, to find the LCMS server with the lightest load, providing a convenient mobile learning environment anytime, anywhere, automatically.

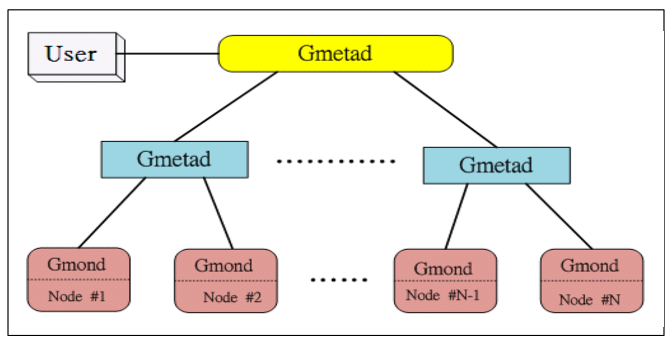

Fig. 3 Ganglia System Architecture Diagram.

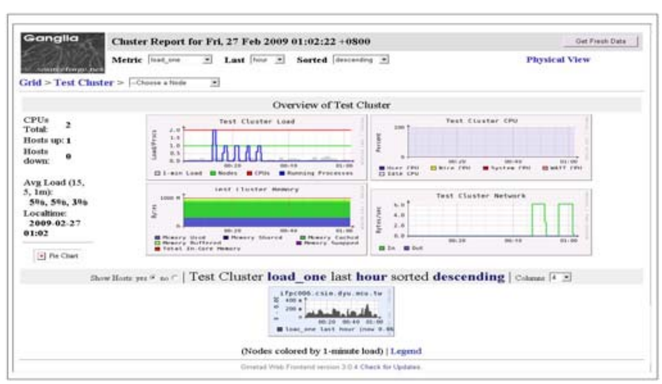

Fig. 4 Ganglia Sensor system information.

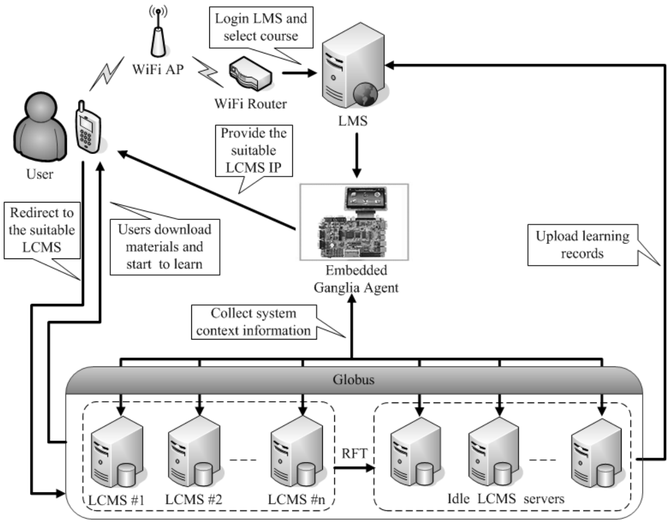

Fig. 5 Ubiquitous learning system architecture with Embedded Ganglia Agent.

3.2 Proposed Embedded Ganglia Agent

To ensure a real-time learning effect, and ensure that all learners can proceed the learning from LCMS; therefore, the research designed an embedded Ganglia Agent, which is designed by the Xscale DMA-270 development board[13], to interpret the surveillance data gathered at the rear-end LCMS. The Ganglia component– Gmetad program records system information on the rear-end's nodes, such as the CPU utilization rate, memory cache size, networking flow volume and so forth, and produce these information into XML files. The proposed embedded Ganglia Agent will download and interpret the XML files, locate a rear-end LCMS server with the lightest load and make a decision to the replication of teaching materials. In order for many learners to smoothly access the LCMS contents, this research proposes the embedded Ganglia Agent design that collects and analyzes related back-end server context information. By analyzing context data, EGA will decide the adequate order of LCMS servers providing contents. When the monitored LCMS server is overloaded, the system activates the data transfer function of the RFT component of the grid, and automatically duplicates the content to an idle LCMS server within the grid, meanwhile guiding the learner to the LCMS server to proceed with his learning. The Embedded Ganglia Agent has Gmetad daemon implemented, and is capable of collecting back-end LCMS servers' context data (such as CPU load, memory space, hard disk capacity, network bandwidth etc). The Gmetad collects the system context data provided by Gmond in all back-end LCMS servers, and automatically organizes the data into XML format, as shown in figure 6. The system is set that every 15 seconds Gmond will automatically transfer LCMS context data to the EGA server, and when the LCMS server's CPU load goes up to 40%, the grid RFT file transfer service will be activated, duplicating learning content to idle servers in the grid, effectively balancing the network load, achieving the flexible expansion mechanism of the LCMS. The following are explanations of executing grid RFT component services:

(1) When the RFT service in inactivated, the LCMS network load stays below 25Bytes, as shown in figure 7, its CPU load below 15%, as shown in figure 8.
(2) With the increase in clients, the LCMS's CPU load increases. When the CPU load goes over 40%, as shown in figure 9, the system will activate RFT file duplication function to copy the learning content on to an idle server in the grid, as shown in figure 10, and the idle LCMS will begin receiving the content, as shown in figure 11.
(3) After flexibly expanding a LCMS, as shown in figure 12, the LCMS network load will be relieved, and the CPU load will decrease, as shown in figure 13.


```
<GANGLIA_XML VERSION="3.0.4" SOURCE="gmond">
<CLUSTER NAME="Test Cluster" LOCALTIME="1236698934" OWNER="stwang" LATLONG="unspecified" URL="unspecified">
<HOST NAME="ifpc14221.csie.dyu.edu.tw" IP="192.168.1.3" REPORTED="1236698933" TN="0" TMAX="20" DMAX="0" LOCATION="unspecified" 
<METRIC NAME="disk_total" VAL="38.230" TYPE="double" UNITS="GB" TN="60" TMAX="1200" DMAX="0" SLOPE="both" SOURCE="gmond"/>
<METRIC NAME="cpu_speed" VAL="1796" TYPE="uint32" UNITS="MHz" TN="60" TMAX="1200" DMAX="0" SLOPE="zero" SOURCE="gmond"/>
<METRIC NAME="part_max_used" VAL="28.3" TYPE="float" UNITS="" TN="20" TMAX="180" DMAX="0" SLOPE="both" SOURCE="gmond"/>
<METRIC NAME="swap_total" VAL="524280" TYPE="uint32" UNITS="KB" TN="60" TMAX="1200" DMAX="0" SLOPE="zero" SOURCE="gmond"/>
<METRIC NAME="os_name" VAL="Linux" TYPE="string" UNITS="" TN="60" TMAX="1200" DMAX="0" SLOPE="zero" SOURCE="gmond"/>
<METRIC NAME="cpu_user" VAL="56.1" TYPE="float" UNITS="%" TN="0" TMAX="90" DMAX="0" SLOPE="both" SOURCE="gmond"/>
<METRIC NAME="cpu_system" VAL="6.3" TYPE="float" UNITS="%" TN="0" TMAX="90" DMAX="0" SLOPE="both" SOURCE="gmond"/>
<METRIC NAME="cpu_aidle" VAL="87.7" TYPE="float" UNITS="%" TN="0" TMAX="3800" DMAX="0" SLOPE="both" SOURCE="gmond"/>
<METRIC NAME="load_five" VAL="1.53" TYPE="float" UNITS="" TN="60" TMAX="325" DMAX="0" SLOPE="both" SOURCE="gmond"/>
<METRIC NAME="proc_run" VAL="2" TYPE="uint32" UNITS="" TN="60" TMAX="950" DMAX="0" SLOPE="both" SOURCE="gmond"/>
<METRIC NAME="mem_free" VAL="3124" TYPE="uint32" UNITS="KB" TN="0" TMAX="180" DMAX="0" SLOPE="both" SOURCE="gmond"/>
<METRIC NAME="mem_buffers" VAL="172" TYPE="uint32" UNITS="KB" TN="0" TMAX="180" DMAX="0" SLOPE="both" SOURCE="gmond"/>
<METRIC NAME="swap_free" VAL="439948" TYPE="uint32" UNITS="KB" TN="0" TMAX="180" DMAX="0" SLOPE="both" SOURCE="gmond"/>
<METRIC NAME="bytes_in" VAL="416.75" TYPE="float" UNITS="bytes/sec" TN="180" TMAX="300" DMAX="0" SLOPE="both" SOURCE="gmond"/>
<METRIC NAME="pkts_out" VAL="14.65" TYPE="float" UNITS="packets/sec" TN="180" TMAX="300" DMAX="0" SLOPE="both" SOURCE="gmond"/>
<METRIC NAME="cpu_num" VAL="1" TYPE="uint16" UNITS="CPUs" TN="60" TMAX="1200" DMAX="0" SLOPE="zero" SOURCE="gmond"/>
<METRIC NAME="disk_free" VAL="27.423" TYPE="double" UNITS="GB" TN="20" TMAX="180" DMAX="0" SLOPE="both" SOURCE="gmond"/>
<METRIC NAME="mem_total" VAL="255604" TYPE="uint32" UNITS="KB" TN="60" TMAX="1200" DMAX="0" SLOPE="zero" SOURCE="gmond"/>
<METRIC NAME="cpu_wio" VAL="14.2" TYPE="float" UNITS="%" TN="0" TMAX="90" DMAX="0" SLOPE="both" SOURCE="gmond"/>
<METRIC NAME="boottime" VAL="1236654019" TYPE="uint32" UNITS="s" TN="60" TMAX="1200" DMAX="0" SLOPE="zero" SOURCE="gmond"/>
<METRIC NAME="machine_type" VAL="x86" TYPE="string" UNITS="" TN="60" TMAX="1200" DMAX="0" SLOPE="zero" SOURCE="gmond"/>
<METRIC NAME="os_release" VAL="2.6.18-92.el5" TYPE="string" UNITS="" TN="60" TMAX="1200" DMAX="0" SLOPE="zero" SOURCE="gmond"/>
<METRIC NAME="cpu_nice" VAL="0.0" TYPE="float" UNITS="%" TN="0" TMAX="90" DMAX="0" SLOPE="both" SOURCE="gmond"/>
<METRIC NAME="cpu_idle" VAL="23.3" TYPE="float" UNITS="%" TN="0" TMAX="90" DMAX="0" SLOPE="both" SOURCE="gmond"/>
<METRIC NAME="load_one" VAL="2.45" TYPE="float" UNITS="" TN="60" TMAX="70" DMAX="0" SLOPE="both" SOURCE="gmond"/>
<METRIC NAME="load_fifteen" VAL="0.66" TYPE="float" UNITS="" TN="60" TMAX="950" DMAX="0" SLOPE="both" SOURCE="gmond"/>
<METRIC NAME="proc_total" VAL="216" TYPE="uint32" UNITS="" TN="60" TMAX="950" DMAX="0" SLOPE="both" SOURCE="gmond"/>
<METRIC NAME="mem_shared" VAL="0" TYPE="uint32" UNITS="KB" TN="0" TMAX="180" DMAX="0" SLOPE="both" SOURCE="gmond"/>
<METRIC NAME="mem_cached" VAL="87156" TYPE="uint32" UNITS="KB" TN="0" TMAX="180" DMAX="0" SLOPE="both" SOURCE="gmond"/>
<METRIC NAME="gexec" VAL="OFF" TYPE="string" UNITS="" TN="60" TMAX="300" DMAX="0" SLOPE="zero" SOURCE="gmond"/>
<METRIC NAME="bytes_out" VAL="2514.25" TYPE="float" UNITS="bytes/sec" TN="180" TMAX="300" DMAX="0" SLOPE="both" SOURCE="gmond"/>
<METRIC NAME="pkts_in" VAL="6.85" TYPE="float" UNITS="packets/sec" TN="180" TMAX="300" DMAX="0" SLOPE="both" SOURCE="gmond"/>
</HOST>
```


Fig. 6 Context data collection of the embedded Ganglia Agent

Fig. 7 Network load of LCMS (192.168.1.5) without activating RFT service.

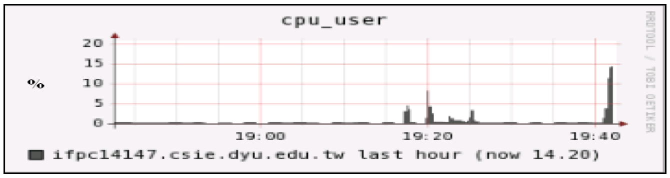

Fig. 8 CPU load of LCMS (192.168.1.5) without activating RFT service.

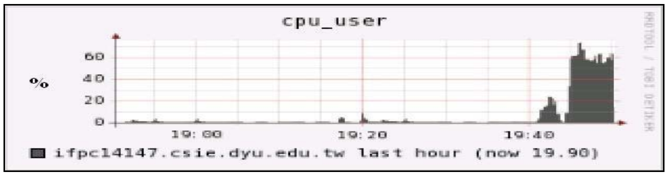

Fig. 9 The CPU load of LCMS(192.168.1.5) going over 40%.

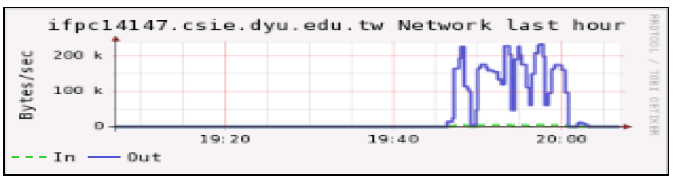

Fig. 10 Activating RFT function of the grid to duplicate content to idle LCMS(192.168.1.6).

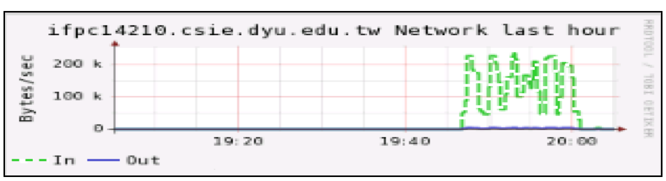

Fig. 11 Idle LCMS(192.168.1.6) starting to receive content.

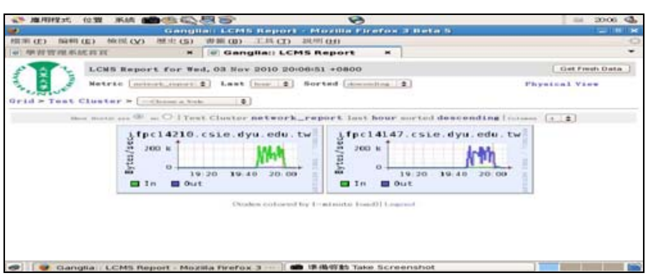

Fig. 12 Network load after flexibly expanding a LCMS(192.168.1.6).

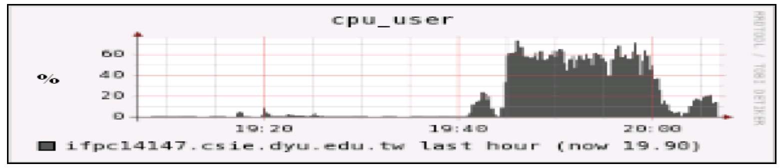

Fig. 13 CPU load of the LCMS(192.168.1.5) decreasing.

4. System Implementation

The system architecture proposed by this research mainly includes LMS, embedded Ganglia agent with context-awareness, and back-end learning content management system (LCMS) with flexible expandability. LMS is responsible for the management of students' personal information, learning log, and provides lesson catalogues for the student to choose from. LCMS is responsible for the management and storage of back-end learning contents. EGA is responsible for not only the sensory data of back-end LCMS server's network bandwidth, CPU load, hard disk capacity, but also integrates the grid RFT components for it to provide LCMS server with more effective flexible expandability, and by this information offers the clients the IP address of the LCMS servers with lower loads, thus effectively balancing back-end LCMS server's load; the actual implementations are as shown in figure 14.

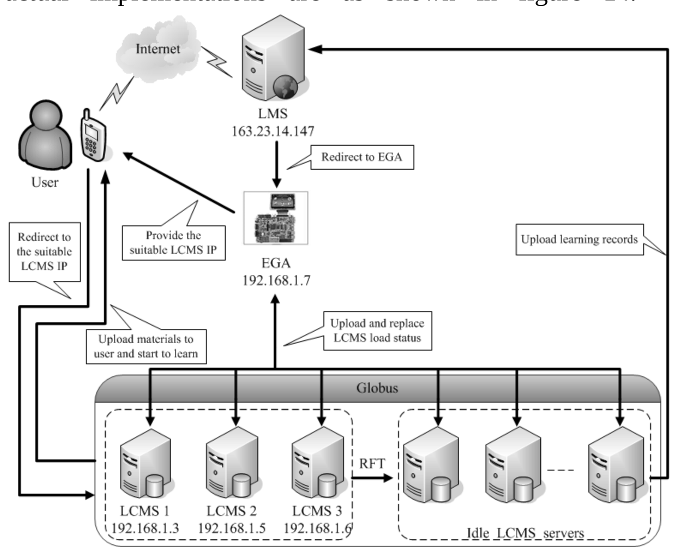

Fig. 14 Implementing EGA's ubiquitous learning system architecture.

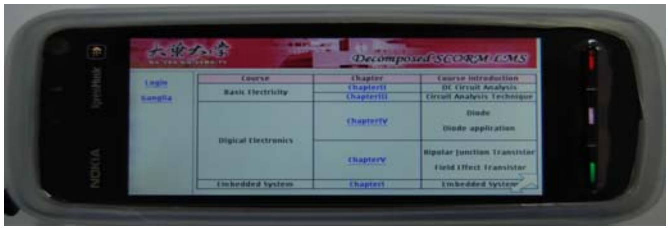

Fig. 15 Lesson option.

Fig. 16 Computing context data

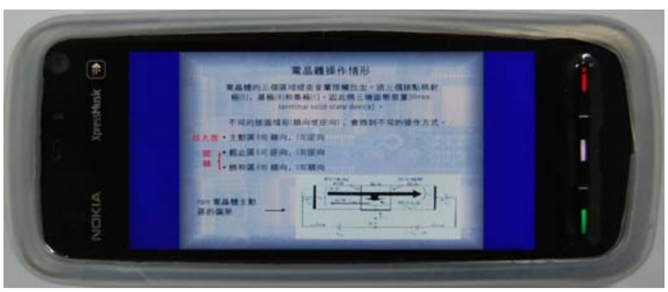

Fig. 17 Using a mobile phone to learn the contents(1).

Fig. 18 Using a mobile phone to learn the contents(2)

4.1 Performance analysis of EGA

In Figure 19 and 20, the x-axis stands for the number of connection while the average response time for the users requesting the LCMS IP from Ganglia agent (unit: ms, millisecond) is indicated by the y-axis. While 100 connections are using the system simultaneously, the response time for each learner will be different because the received status data calculated by averaging the sum of each learner's response time are orderly updated via Ganglia agent into each learner's PC. Under PC Ganglia agent and embedded Ganglia agent, the experimental results for the system's response time against increasing the number of learners are respectively listed below:

(1) Under PC Ganglia agent, while learning is done by grouping, the average response times for Group 1, 2, 3, and 4 are respectively "0.3ms," "0.4ms," "0.43ms," and "0.45~1.15ms," as illustrated in Figure 19.

(2) Under embedded Ganglia agent, while learning is done by grouping, the average response times for Group 1, 2, 3, and 4 are respectively "0.5ms," "0.6ms," "0.61ms," and "0.62~1.3ms," as illustrated in Figure 20.

Under embedded Ganglia agent Architecture, when the users continuously connect to the embedded Ganglia agent to request the IP of most suitable LCMS, the maximum averaged waiting time for the user is about 1.3 ms, as shown in Figure 20. This means that the user only needs to spend about 1.3 ms at most to connect the appropriate LCMS and begin the learning.

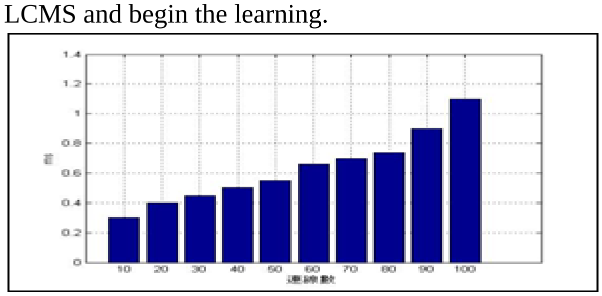

Fig. 19 Response time of PC Ganglia agent.

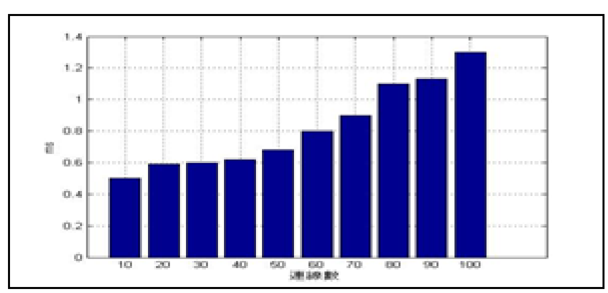

Fig.20 Response time of embedded Ganglia agent.

5. Conclusion

Most digital learning systems nowadays combine learning management system and content management system on a single high-end server. This kind of architecture is easier to design and implement, and is suited for a teaching environment for a less amount of learners. If there were to be a huge number of clients logging on the system to learn, the limits of network bandwidth and data access will become an obstacle to learning. This research proposes ubiquitous learning system architecture with embedded Ganglia Agent. With an agent that has the capability to integrate load balancing, implementing many LCMS to provide learning services could be achieved. After the client logs on the learning management system and chooses a lesson service, the EGA will connect to a LCMS with lower load, and proceed with the lesson. This system utilizes dispersed low-cost hardware equipment to build a high-cost server system architecture. Not only does it effectively decrease the total cost of the system, but by the RFT component's file duplication function in the grid, the back-end LCMS server's flexible expandability mechanism could be achieved, thus effectively improving the total performance of the system.

References

[1] Harris, P. Go in Mobile, Retrieved May 5, 2006, from http://www.learningcircuits.org/2001/jul2001/harris.html.

[2] F. Lehner, H. Nosekabel, "The Role of Mobile Devices in E-learning - First Experience with a E-learning Environment", In M. Milrad, H.U. Hoppe and Kinshuk (Eds.), IEEE International Workshop on Wireless and Mobile Technologies in Education (pp.103-106). Los Alamitos, USA: IEEE Computer Society, 2002.

[3] K. Sakamura, N. Koshizuka, "Ubiquitous Computing Technologies for Ubiquitous Learning", IEEE International Workshop on Wireless and Mobile Technologies in Education, 2005.

[4] G. Zhang, Q. Jin, et al, "A Framework of Social Interaction Support for Ubiquitous Learning", International Conference on Advanced Information Networking and Application, 2005.

[5] Lionel M. Ni, Y.H. Liu, Y.C Lau, Abhishek P. Patil., "LANDMARC: Indoor Location Sensing Using Active RFID.", Wireless Networks, Vol.10, No.6, p.701-710, 2004.

[6] Bill Schilit & Marvin Theimer, "Disseminating active map information to mobile hosts" ,IEEE Network,Vol.8,No.5, pp.22-32, 1994.

[7] Bill Schilit & Norman Adams & Roy Want, "Context-Aware Computing Applications" ,IEEE Workshop on Mobile Computing Systems and Applications. Santa Cruz, CA, US, IEEE Computer Society Press, 1994.

[8] Guanling Chen & David Kotz, "A survey of Context-aware mobile computing research" ,Dartmouth Computer Science Technical Report TR2000-381, 2000.

[9] Schiller & Voisard, "Location-based services", San Francisco, CA: Morgan Kaufmann, 2004.

[10] M. Baldauf et. al, "A Survey on Context-Aware Systems", International Journal of Ad- Hoc and Ubiquitous Computing,Vol.2,No.4, 2007.

[11] C.H. Yin, "Ubiquitous-Learning System for the Japanese Polite Expressions", IEEE International Workshop on Wireless and Mobile Technologies in Education, 2005.

[12] S.J.H.Yang, "Context Aware Ubiquitous Learning Environments for Peer-to-Peer Collaborative Learning", Educational Technology & Society, Vol.9, No.1, pp.188-201, 2006.

[13] F.C. Kao, Y.S. Tsai, A.S. Chiou, S.R. Wang, C.W. Liu, " The Design of Intelligent Sensor Network Based on Grid Structure", IEEE Asia-Pacific Services Computing Conference, pp.623-630, 2008.

[14] I. Foster, C. Kesselman and S. Tuecke, "The Anatomy of the Grid Enabling Scalable Virtual Organizations", Supercomputer Application, 2001, pp. 2 – 6.

[15] I. Foster, C. Kesselman and S. Tuecke, "The Anatomy of the Grid Enabling Scalable Virtual Organizations" Supercomputer Application, 2001, pp. 6-14.