The Design of Intelligent Decomposed LMS with Improved Ganglia Agent

Fu-Chien Kao, Ting-Hao Huang, Wen-Yu Chang

Department of Computer Science and Information Engineering, DAYEH University, Taiwan

Summary
This paper proposes an intelligent learning management system with load-balancing function based on Grid structure that allows for the replication of learning materials and balance of network traffic. The system consists of the Learning Management System (LMS) for processing basic data, learning data and learning records of learners, and the Learning Content Management Systems (LCMS) based on the Grid structure for managing and storing the related teaching materials. The Web Service cross-platform distribution configuration of this research provides common communications between systems and enhances the capability of integrating learning resources. The Grid architecture of the system ensures a load-balancing functionality for LCMS of different domains. The connection program embedded in the PC of the learner via LMS connects to the Ganglia agent for access to required teaching materials from each LCMS. The LCMS with the minimum load is then selected from suitable LCMS as the source of the teaching materials. The proposed intelligent learning environment based on Grid structure is designed to improve the overload and the expansibility problems existing in the traditional decomposed LMS system.

Key words:
Decomposed LMS, Grid, Ganglia Agent

1. Introduction

With the popularization of the Internet, e-learning and related technologies are increasingly utilized by both the academic community and industry. To integrate learning resources of different platforms and share teaching materials easily using these platforms, designers of teaching materials and developers of platforms must observe the same standards from the earliest stage of the design and development. Hence, the SCORM standard is created and all teaching materials conforming to this standard can be easily transferred to any platforms developed using the same standard. Similarly, different SCORM teaching materials can be imported from platforms based on the same standard. The reusability, accessibility, durability, interoperability, adaptability, affordability, and manageability of teaching materials are thus ensured and users may access more resources under this structure [1].

Standardization of format solves the problem of teaching material transferal, but courses cannot be shared amongst individual learning management systems. Most current research focuses upon the development of functions and teaching materials based on a single learning platform [2, 3, 4], rather than the integration of multiple platforms for users to access course resources from a single portal [5].

Even though there is a good dependency between teaching materials and platforms, the user who needs to transfer resources between platforms must import data from individual courses in each platform. Therefore, sharing efficiency in the circumstances is significantly affected. To solve this problem, a common standard is required for the integration of different platforms and teaching materials to ensure sharing of resources and optimization of their utilization [5].

Besides, the course transfer between platforms should be implemented in a real-time and efficient manner. However, this is inefficient if the importation of teaching materials and transfer of files are executed when the learner cannot find required teaching resources on the platform in use and identifies the resources on another platform via the common interface. It takes even more time for the transfer of huge teaching material files when audio/video multi-media are used. In addition to the common interface between platforms, the architecture of the plant must also be improved.

The proposed system based on a decomposition structure consists of two sub-systems: The Learning Management System (LMS) for processing basic data, learning data and learning records of students, and the Learning Content Management System (LCMS) based on Grid structure for managing and storing course resources [6]. To resolve the rear-end server's overload problems existing in conventional one-to-one decomposed LMS structure, the research presented an intelligent decomposed LMS based on Grid technologies. The proposed system structure adopts grid-related technology to integrate and disperse the rear-end computer's storage space, and to effectively balance the server load. The subsequent sections include SCORM standard, Grid, system structure and a conclusion.

Manuscript received June 5, 2009
Manuscript revised June 20, 2009

2. SCORM Standard and Grid Structure

This research builds a distributing learning system in conformity to the SCORM standard under the distributed Web Service structure. Related standards and structure are described as follows.

2.1 SCORM standard

To shorten the time-to-market and cost of teaching materials and ensure their reuse and circulation for different LMS by building the "Teaching Material Reuse and Share Mechanism", the Whitehouse Technology Office and Department of Defense (DoD) of the United States initialized the ADL (Advanced Distributed Learning Initiative) in late 1997. Teaching material developers, users, and institutions responsible for proliferating IMS2, AICC (Aviation Industry CBT Committee) and IEEE standards participated in this program to develop an associated set of technology guidelines – SCORM based on existing achievements in the creation of relevant standards [7]. The standard mainly defines two system structures: "Content Aggregation Model" (CAM) and "Run-time Environment" (RTE). These structures are briefly described as follows [7, 8].

2.1.1 Content Aggregation Model (CAM)

In past e-Learning environments, the transmission model between teaching materials and platforms differs depending on the structure of individual platforms. As a result, teaching materials can only be executed on platforms developed by the same manufacturer and copying original data to other platforms is costly. To solve the problem of teaching materials that cannot be reused by platforms due to different data formats, SCORM provided the Content Aggregation Model as a standard for the development of teaching materials [6].

CAM defines a picture or music file in the teaching material as an Asset. A collection of Assets is referred to as a SCO (Sharable Content Object), such as a web page containing texts, pictures and music files. Metadata files describe the component (Asset, SCO) of the teaching material in XML format for further management of course resources. Content Packaging uses Manifest files, with the same file name imsmanifest.xml, for packing teaching materials and describes members of the teaching material and course structures in XML format [6, 9, 10, 11]. As shown in Fig. 1, it shows a structure that conforms to the SCORM standard. Once a course is exported as Content Package, the LMS that conforms to SCORM is able to analyze the Manifest file (imsmanifest.xml) of the Content Packaging and import the course in the platform for sharing [6, 7, 12].

2.1.2 Run-Time Environment (RTE)

Most existing e-Learning systems are developed with different languages and the definition of delivered parameters is different, indicating a lack of common communication between platforms and teaching materials. Run-Time Environment uses Java Script for the integration of platforms and teaching materials and creates communications between them via APIWrapper.js and SCOFunctions.js; the former records the current status of courses, while the latter records the performance of each learner. The platform will store this data in the database and the integration of platforms and teaching materials is implemented using the same applications.

Fig. 1 is the schematic drawing of the Run-Time Environment. The user connects to the LMS from his or her PC and embeds the API Adapter downloaded from the LMS in the browser of the PC. The API Adapter is driven by two Java Script applications contained in each course. It is a medium for communication between the LMS and teaching materials and responsible for the transmission and reception of data between the client and server [6].

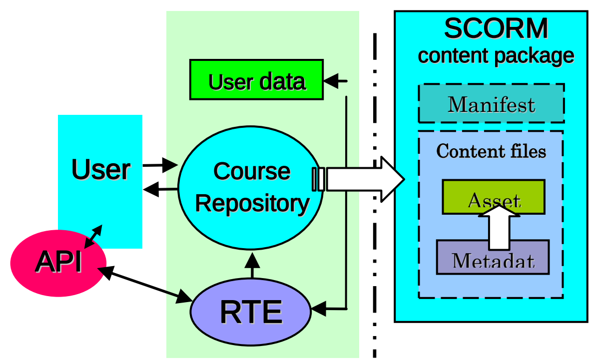


Fig. 1 SCORM Standard

2.2 Grid

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.

There are organizations around the world that utilize the grid technology to set up large-scale computational systems, such as earth's satellite imagery observation, biomedicine DNA computational simulations, weather simulations, military simulations and such that all turn to using the grid technology to process dispersed computation and manage enormous figurative data. While currently a most commonly used intermediary software for launching these large-scale grids has been Globus Toolkit [15].

2.2.1 Grid framework

Developing a comprehensive grid system calls for an infinite number of gird protocols, services and package software developed. Its development objectives and focus center on the protocols, allowing a virtual organization's users to store/retrieve and share the resources. Fig. 5 depicts a five-tier grid framework [15, 16], and each layer's framework and services are described as follows,
(1) Fabric Layer: Grid resources refer to all physical data that can be utilized in a grid, including computational resource, storage resource, networking resource and the like.
(2) Connectivity Layer: Under the grid concept, users in different environments are able to utilize the resources in different environments. Thus a grid needs to define a mechanism that allows intercommunication between the resources and the resources, and between the resources and the user; it also needs to ascertain the issues that the user has undergo the virtual organization's certification or the storage and retrieval of resources are conducted securely.

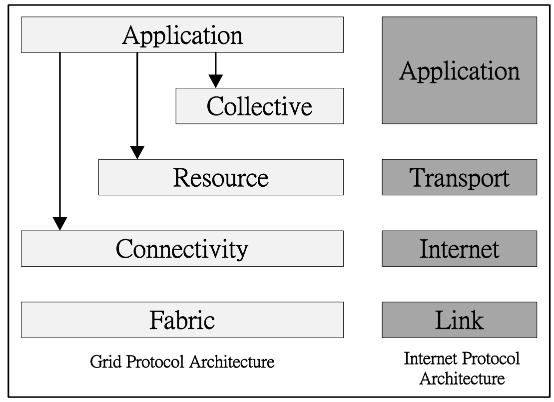


Fig. 2 Five-Tier Grid framework

In a grid, the user's and resources' communication is built upon a secured communications mechanism; the connectivity layer defines the communications protocols and certification mechanism that enable resources on the fabric layer to exchange data securely, while communications protocols are built upon the current networking protocols, i.e. TCP, UDP, DNS and the like, rather than an original networking communication protocol.
(3) Resource Layer: The resource layer defines the resources' independent operations and specifications, rather than specify the cooperation between resources and resources. The resource layer is built under the connectivity layer's communications protocols and certification mechanism, and the defined protocols including the resources' initialization, execution and the resources' shared operations, etc.
(4) Collective Layer: It defines, on the resource layer, each resource's attribute and method. The collective layer defines the resources' mutually collaborated storage and retrieval methods and strategies. The collective layer's defined protocols and services enable the user to store and retrieve multiple resources and work together to complete a process.
(5) Applications Layer: The applications layer offers the user with applications software, grid portal and tools. The user is able to access all resources in a grid using these software tools. From the viewpoint of the applications program, resources in a virtual organization are accessed and utilized to achieve the objective of a mutually collaborated computation of the entire grid's resources.

2.2.2 Grid middleware

Globus Toolkit is currently a most influential grid computing software worldwide; Globus Toolkit offers relevant component service functions in resource management, signal service and figure management, ad is a grid-launching software tool for various platforms, helping and developing the establishment of large-scale gird experiment platforms. The critical components of Globus Toolkit are described as follows:

(a) Grid Security Infrastructure (GSI): GSI [17,18] grids' security certification and encrypted communication offer single sign-on, identity authentication, and transmission encryption functions, while all services and protocols of a grid have been built upon a grid security infrastructure.

(b) Resource Management: Grid resource management serves to collectively manage a variety of resources dispersed in a grid, enabling all resource utilization requests are able to access the resources they are looking for. GRAM [19, 20] offers standardized interface protocols and API that allow the users to access remote resources.

(c) Information Services: MDS [21, 22] offers a set of standard mechanisms to gather the resources' status and adjusted information. The grid information service center offers a grid environment's information registration inquiry and update processes, allowing a real-time response on the state of a grid implementation.

(d) Data Management: GridFTP [23, 24] serves to provide high-performance data transmission, supporting the three-way transmission, parallel transmission functions, and only enables transmission under a safe GSI authentication. As system file backup is often executed in data grids, prompting a multiple number of identical files in the grid system, the grid presents a mechanism and solves the identical file problem.

3. The Intelligent decomposed LMS based on Grid structure

This research provides a Web Service distributing learning management system based on Grid structure to balance the load between the rear-end servers with teaching materials. Relevant system structures are described as the following sections.

3.1 Traditional LMS / LCMS structure

To improve service efficiency, the learning system is split into two independent sub-systems LMS and LCMS as a realizable solution, as shown in Fig 3. LMS operates with processes as its basis, and is mainly used to manage student learning information and their learning progress. Then adjust for the material sequence accordingly. LCMS are mainly used to store and manage course information for student use. This structure will achieve the target of system load spreading, but with the same disadvantage of existent expansion flexibility as single servers. Therefore, this traditional decomposed structure cannot resolve network congestion when massive users are online resulting in lowered performance or even interrupted services of the LCMS.

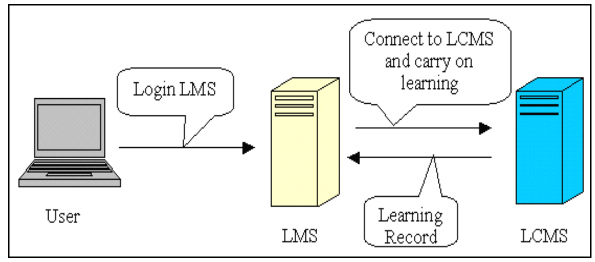


Fig. 3 Traditional one-to-one decomposed LMS structure

3.2 Load-balancing capability

In general, the LMS only needs to interact with the API Adapter of the client and store required data in the system, while the LCMS must provide all teaching materials to the client. The LCMS, therefore, needs more bandwidth and quality hardware for operation than the LMS.

The decomposition structure based on Grid structure allows subsystems to be added only to that part of the entire system that has higher network traffic. The load of the LCMS increases when more users access teaching materials from the system at the same time. Under these circumstances, the performance of the learning platform would be improved by adding another LCMS with the same teaching materials through the replicating technology of Grid RFT component and transferring some of the users to the new LCMS servers for service.

3.3 Monitoring of teaching materials and the performance of LCMS

When users cannot find required teaching materials from the original LCMS, the system will help them to search resources from other LCMS and select the LCMS with the lowest load from suitable LCMS to be the source of the teaching materials. The monitoring function is performed by the proposed improved Ganglia agent. Users may search information on teaching materials from various LCMS on the Internet via this agent server and the integrated Web Service interface.

3.4 Ganglia Agent

A distributed surveillance tool, Ganglia is often used to monitor high-performance clustering systems. Ganglia is able to survey the system status; these constant programs exchange and communicate the data using UDP broadcast, and upload them to Gmetad using a tree structure to sort all of the surveillance data. And the communication between clustering computers and clustering computers has the data exchange executed through TCP using the XML file format.

Ganglia, in the clustering computer structure, is an equivalent of the tree structure, and the nodes at the non-leaf nodes are in charge of gathering the leaf node's surveillance data. Ganglia codes with five cores, as depicted in Fig 4, and a relevant description is provided as follows,

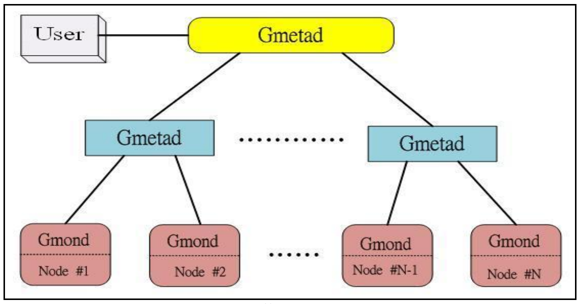


Fig. 4 Ganglia's core structure

(1) Ganglia Monitoring Daemon (gmond)
A multiple-thread residence program, Ganglia monitors system status at the nodes, and deciphers system status at the other nodes using the UDP broadcast method. It also transmits the system status monitored to Gmetad via the XML format.
(2) Ganglia Meta Daemon (gmetad)
Gmetad will routinely request for surveillance information from the nodes, and interpret XML files collected. It transmits sorted surveillance data to the user via the XML file format.
(3)Gmetric
A user-based customized tool, Ganglia enables the user to define desired surveillance data through the command mode integrating Gmond.
(4)Gstat
Gstat serves as a command-sequencing tool on surveillance figures derived from all nodes. The information it offers can be information of the CPU execution count and CPU load of a clustering of computers.
(5) Ganglia PHP Web Frontend
Ganglia offer the user with a graphic-based monitoring data. The user is able to monitor the nodes and load of all clustering systems through an interactive Web page, and prompt data charts Gmetad uploads in a Web page format using the XML mode, as depicted in Fig 5. By modifying the source program of Ganglia surveillance service, the research presents a improved Ganglia Agent that offers the service functions of parsing XML files, and locating the lightest load rear-end server to support the front-end users servers' resource monitoring.

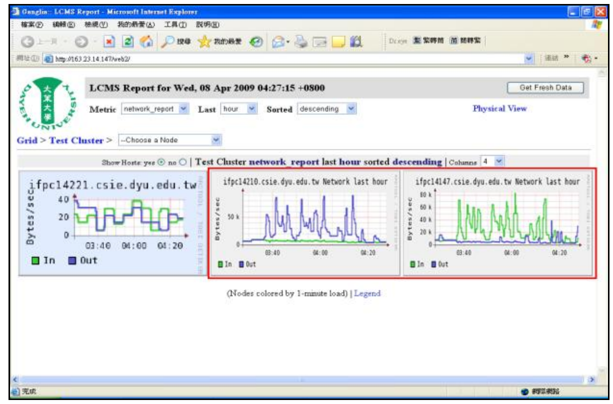
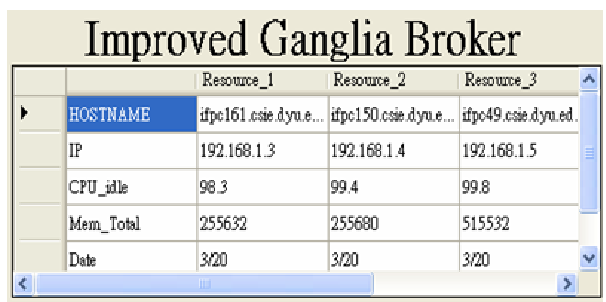


Fig. 5 System resource monitoring

3.4.1 Improved Ganglia Agent

To ensure a real-time learning effect, and ensure that all learners can proceed the learning from LCMS; therefore, the research designed a improved Ganglia Agent 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 improved 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, as depicted in Fig 6.


Improved Ganglia Broker

	Resource_1	Resource_2	Resource_3
HOSTNAME	ifpc161.csie.dyu.e...	ifpc150.csie.dyu.e...	ifpc49.csie.dyu.ed...
IP	192.168.1.3	192.168.1.4	192.168.1.5
CPU_idle	98.3	99.4	99.8
Mem_Total	255632	255680	515532
Date	3/20	3/20	3/20

Fig. 6 Ganglia agent's monitoring screen

3.5 Proposed intelligent system structure

The system framework's actual implementation is as depicted in Fig 7, and relevant launching software includes, Linux CentOS5, Globus Toolkit, Ganglia, Java and Java Cog Kit.

The system replicates all relevant SCORM teaching materials scattered on the rear-end server using the Globus's RFT component. The research develops the teaching resource replicating program using Java Cot Kit.

The improved structure of this research no longer uses LMS JAVA Applet as the API Adapter. Instead, the API Adapter is contained in the LCMS for data transmission and embedded in the LMS web page via dynamic hyperlinks. When the user finds the most suitable LCMS via the improved Ganglia Agent, the LCMS becomes a proxy server under the support of our URL Rewrite technology. The user requests connection to the LMS via the selected LCMS proxy server and implements learning via the LMS learning interface page in conjunction with the LCMS teaching materials. No connection is required from the client to the LMS in the circumstances and, for the browser, LCMS JAVA Script is used by users to drive LCMS API Adapter. With the URL Rewrite technology, the API Adapter contained in the LCMS and embedded in the LMS web page via hyperlinks can be accessed by the LMS. The system structure of this research provides an effective solution for the cross-domain scripting issue. The system operates in five procedures as follows,

(1) The user connects to the LMS from his or her PC, logging into the learning platform and embeds the connection program downloaded from the LMS to the PC.

(2) The user connects the PC to the improved Ganglia Agent via the connection program to explore required teaching materials from different LCMS. The LCMS with the lowest load is then selected from suitable LCMS as the source of the teaching materials.

(3) The learning is implemented by direct connection from the client to the selected LCMS. The user requests connection to the LMS via the selected LCMS as the proxy server and implements learning via the LMS learning interface page in conjunction with the LCMS teaching materials.

(4) All LCMS register relevant information in the Ganglia Agent via Web service in a real-time manner.

(5) When the users become more, then the system would start to implement the replication of teaching material through the Grid RFT component.

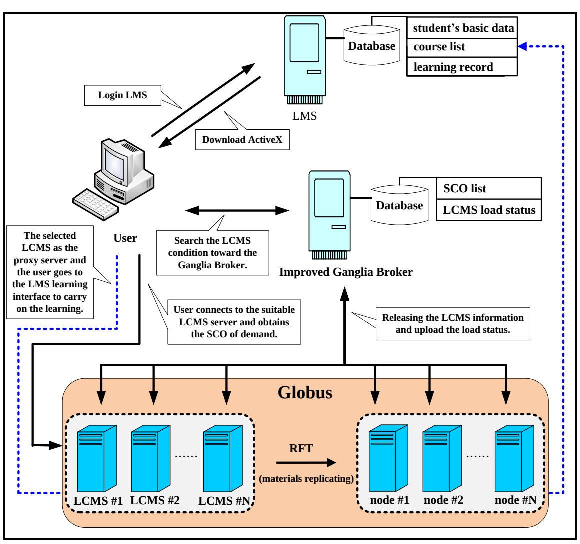


Fig. 7 System structure

4. Performance tests

In this section, the performance tests of traditional system structure with single LCMS server and decomposed system with improved Ganglia agent would be explored respectively. All response time tests in the research are conducted by analyzing the packet response time, acquired by using Ethereal, a professional program for packet capturing analysis. In order to retrieve the packets communicated between improved Ganglia Agent and each learner's PC for analysis of data transmission, Ethereal is utilized (Network packet analyzer, version 0.10.10). In Fig 8, the x-axis stands for the number of learners while the average response time for a traditional single LCMS server system is indicated by the y-axis. While 40 learners are using the system simultaneously, the maximum averaged response time is over 567 ms. For a traditional decomposed structure (with one LMS and one LCMS), the averaged response time appear as a rising trend when the learners gradually increase, as shown in Fig 8. This means the load of LCMS could not be balanced under the traditional decomposed system structure.

This paper provides a load-balancing structure with a Ganglia Agent that balances LCMS network traffic effectively. The structure is capable of creating an integrated SCORM learning environment by collaboration of two servers in different domains to improve the resource integration capability of the SCORM and the response efficiency of the system. The user may log in the LMS to acquire teaching materials and information on the LCMS server, which is selected by the Ganglia agent. According to the dynamical recording method, the designed Ganglia agent selects an appropriate LCMS Server to serve the user. Under Ganglia agent Architecture, the experimental results for the system's response time against increasing the number of learners are shown in Fig 9. When the users continuously connect to the Ganglia agent to request the IP of most suitable LCMS, the maximum averaged waiting time for the user is about 309 ms, as shown in Fig 9. This means that the user only needs to spend about 309 ms at most to connect the appropriate LCMS and begin the learning.

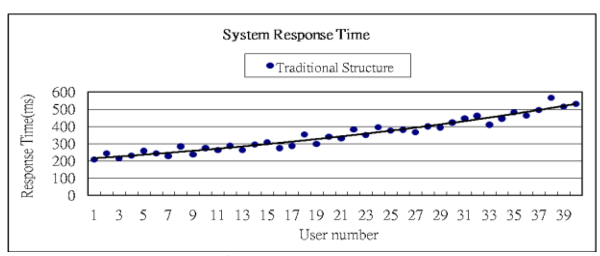


Fig. 8 Response time of traditional decomposed system structure

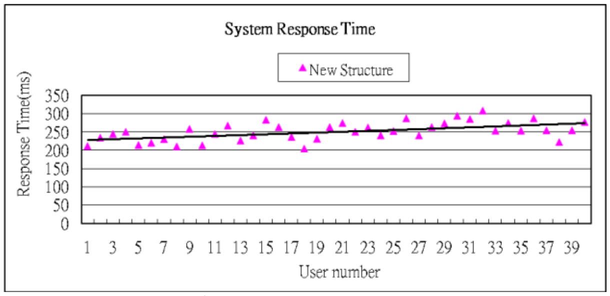


Fig. 9 Response time of proposed system structure with Ganglia Agent

5. Conclusion

This research provides a learning system structure based on Grid that shares LCMS network traffic effectively. The structure is capable of creating an integrated SCORM learning environment by collaboration of multiple servers in different domains to improve the resource integration capability of the SCORM materials and the response efficiency of the system. The system of the research consists of the Learning Management System (LMS) and Learning Content Management System (LCMS) based on Grid structure. This decomposition structure is helpful for load balancing of LCMS in multiple domains. The user may log in the LMS, as shown in Fig 7, to acquire teaching materials and load status information on servers via the improved Ganglia Agent. When the user select a required course and enters its list, the system will dynamically assign the most appropriate LCMS for service. The proposed distributed storage mechanism with load-balancing function utilizes the improved Ganglia agent to balance and disperse the load and storage space of rear-end servers with teaching materials, and operates Globus RFT components to complete the replication of teaching materials. As opposed to the one-to-one conventional decomposed LMS structure that can only support a small number of users to learn as hindered by its storage space and server performance, the proposed system framework is more suitable for distributed, multi-user learning, and can be dynamically incorporated with rear-end servers based on Grid structure to offer an effective, large-scale learning system application.

Acknowledgments

This paper has been supported in part by NSC95-2520-S-212-001-MY3 from the National Science Council of Taiwan.

References

[1] W.J. Hsu, P.C. Lin, "Learning Technology Standards and SCORM", http://andrew.csie.ncyu.edu.tw.
[2] T. Tresso, "An automatic evaluation system for technical education at the University level", IEEE Transactions on Education, Aug, 2002, pp.268-275.
[3] P. Liu, L.H. Hsu and A. Chakroborty, "Towards Automating the Generation of SCORM-Based Multimedia Product Training Manuals", ICME'2002 conference, 2002, pp. 397-400.
[4] T.H. Tung, "Apply Knowledge Map to Develop Physics Problem-Solving System", Thesis of Master, Dep. Information and Computer Engineering, Chung Yuan Chrisian University, Chung-Li, Taiwan, Jan, 2002.
[5] Canadian Department of National Defense, "SCORM Dynamic Appearance model", http://www.online-learning.com/papers/SCORMModel.pdf, 2002.
[6] T.H. Chien, F.C. Kao, "The Design of Load-Balancing LMS Based on Decomposition Structure", 5th IEEE International Conference on Advanced Learning Technologies, 2005, pp. 783 – 787.
[7] P. Dodds, et al. (Eds.), "SCORM Content Aggregation Model Version1.2", Advanced Distributed Learning Initiative, http://www.adlnet.org/ , 2001.
[8] P. Dodds, et al. (Eds.), "SCORM Run Time Environment Version1.2", Advanced Distributed Learning Initiative, http://www.adlnet.org/, 2001.
[9] X.F. Liu, Abdulmotaleb El Saddik, Nicolas D. Georgana, "AN IMPLEMENTABLE ARCHITECTURE OF AN E-LEARNING SYSTEM", CCECE, vol. 2, 2003, pp. 717-720.
[10] T. K. Shih, W.C. Chang, N. H. Lin, L. H. Lin, H.H. Hsu, C.T. Hsieh, "Using SOAP and .NET Web Service to Build SCORM RTE and LMS", 17th international Conference on Advanced information Networking and Applications, 2003, pp. 408-418.
[11] J.T. D. Yang, C.Y. Tsai, T. Lin and J. C. Lin, "A SCORM-compliant Content Repository for Sharable Learning Objects", WISCS 2003, 2003, pp. 25-30.
[12] C.P. Chu, C.P. Chang, C.W. Yeh, Y.F. Yeh, "A Web-Service Oriented Framework for building SCORM Compatible Learning Management Systems", International Conference on Information Technology: Coding and Computing (ITCC'04), vol. 1, 2004, pp. 156-161.
[13] Z.H. Ji, "The Design of Vehicle Surveillance System with Intelligent Image Sensor", Thesis of Da-Yeh University, Taiwan, 2007.
[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.
[16] B. Sotomayor, "Globus Toolkit 4 PROGRAMMING JAVA Services", pp.7-10.
[17] Grid Security Infrastructure （GSI）, Globus Project-Globus Toolkit, 2005, http://www-unix.globus.org/toolkit/docs/4.0/security/
[18] IBM Red Book, "Introduction to Grid Computing with Globus", pp.51-78.
[19] Globus GRAM Architecture, http://www.globus.org/toolkit/docs/4.0/execution/prewsgram
[20] IBM Red Book, "Introduction to Grid Computing with Globus", pp.135-138.
[21] Globus MDS Architecture,

http://www.globus.org/toolkit/docs/4.0/info/
[22] IBM Red Book, "Introduction to Grid Computing with Globus", pp.138-140.
[23] B. Sotomayor, "Globus Toolkit 4 PROGRAMMING JAVA Services", pp. 39-41.
[24] IBM Red Book "Introduction to Grid Computing with Globus", pp. 140-142.