An Efficient Authoring Activities Infrastructure Design through Grid Portal Technology

DAYANG HJH. TIAWA AWANG HJ. HAMID[1], MD. RAJIBUL ISLAM[2], NORMA ALIAS[3] and ABDUL HAFIDZ HAJI OMAR[4]
[1]Department of Multimedia Education, Faculty of Education, University Technolgy Malaysia
[2]Faculty of Information Science and Technology, Multimedia University
[3]Ibnu Sina Institute, Faculty of Science, University Technolgy Malaysia
[4]Department of Therapy and Rehabilitation, Faculty of Biomedical and Health Science Engineering, University Technology Malaysia
MALAYSIA
drdayang@utm.my, md.rajibul.islam05@mmu.edu.my, norma@ibnusina.utm.my, aho@utm.my

Abstract: - An authoring environment allows to built software or digital contents and an educational authoring activities infrastructure allows to create, edit and share artifacts like websites, interactive hypermedia, microworlds, simulations. This study presents an efficient authoring infrastructure dealing with online collaborative tools and collaborative authoring environment based on grid portal technology to solve the unequal distribution of task (among students in a group) and compensation (in terms of performance evaluation i.e. results in grades and positive feedback/comments from their instructors/lecturers) problem. We propose the grid portal technology to support online collaborative efforts among students on distributed parallel computing system to make it faster and efficient in order to enhance authoring support in courseware design. An open source grid portal software has used to design the authoring infrastructure along with PVM, MPI, JAVA on Linux platform. This grid poral software and the grid architecture bring computational clusters together into a grid.

Key-Words: - Grid portal, authoring activities, parallel computing, collaborative learning, online learning, courseware design

1. Introduction

Individual work is vital in any learning course but, students should also learn the collaborative behaviour. Students' contributions are required in group efforts. Group works in designing and authoring a courseware is not an easy task. The key problem in a work group is imbalanced distribution of task and performance evaluation. In order to conquer this difficulty, it is essential to offer the online collaborative tools and collaborative authoring environment. Moreover, the conventional web-based education supports only one server outcome which is very slow in searching, uploading, visualizing output and file saving. Therefore, the design and authoring activities in online group works require efficient and powerful web server in parallel computing platform, which will support online collaborative efforts of students. Grid portal technology with parallel computing platform enhances the authoring support for courseware design.

Some of the activities of authoring environments are as below,

- Programming toolkits, often used together with an Integrated Development Environment (IDE)
- Visual languages to author interactive systems
- Content authoring systems
- Multimedia formats in various form (bitmap graphics, vector graphics, etc.), e.g. tools for formats like Macromedia Flash, Macromedia Director.
- Interactive educational multimedia e.g. HyperCard or Authorware.
- Computer programming code and data formats, e.g. XML editors, Web authoring systems, Text editors, Integrated Development Environments.
- E-learning standards such as most LMSs do include an authoring environment through web-based forms, better systems offer support for standards like IMS Simple Sequencing (and hopefully IMS Learning Design in some near future), activity-based systems like LAMS and CeLS, stand-alone editors like the Reload Editors, eXe or the IMS Learning Design Reload editor.
- Editors for microworlds, e.g. squeak, in particular its visual eToys scripting language, LEGO Mindstorms, ToonTalk, AgentSheets.
- Editors for simulations like STELLA and some microworld tools like SimQuest.
- Editors for drill and practice programs.
- E-learning content editors e.g. eXe.

- Editors for quizzing e.g. IMS QTI tools, Hot Potatoes.
- Authoring tools can be either used by teachers or content designers or by students, typically cognitive tools like microworlds or computer-supported argumentation tools. But in principle, one can organize learning activities with any tool, e.g. let them design quizzes or learning contents.

Parallel computing is becoming a major recent trend. The computational platform is supported by a low cost shared and distributed memory in solving the grand challenge applications. The grid portal technology with parallel computing is supporting web based collaborative group works by improving speedup in terms of searching, supporting the huge memory, high quality of visualization and increasing the computational performances. Several kinds of approaches have proposed by various researchers for authoring activities environments. Among them Begona et al. (2005) have described KADDET which is a cognitive diagnostic environment designed to assess the conceptual and procedural learning activities of students [1]. Dicheva et al. (2002) refined their knowledge classification and indexing approach applied in their system AIMS (Agent-based Information Management System) by introducing ontology-oriented support for collaborative courseware authoring [2]. Within collaborative learning environments knowledge is constructed individually but it is shaped by interactions with peers and instructor(s) [3]. It is also influenced by interactions with course content, traditionally text but increasingly image and sound [4]. Wells [5] cautions that interactions must not consist in simply the sharing of ideas; rather, they must result in transformation of perspective.

We propose the Grid portal technology with open source grid portal software on parallel computing platform that offers an efficient web based collaborative tools and collaborative authoring environment of group efforts. This research will actively engage learners/students in exploring, sharing, visualizing the dynamic resources. In this paper we have presented some activities of authoring environment. In section 2, we have illustrated the development and the implementation of Grid portal and web service technology. In section 3, web service paradigm as well as the process flow, Web service performance evaluations and discussion in section 4 and section 5 will be the conclusion.

2. Grid Portal Technology

A Grid is a collection of independently owned and administered resources which have been joined together by a software and hardware infrastructure that interacts with the resources and the users of the resources to provide coordinated dynamic resource sharing in a dependable and consistent way according to policies that have been agreed to by all parties. Because of the large number of resources available on a Grid at any given time, an individual researcher can always be provided with the best resources available at that point of time for his/her needs, and overall, resource utilization can be distributed for maximum efficiency.

The proposed Grid portal is a web server as well as gateway by which users may access web services, manage data and compose workflows. The portal is used by the administrator to construct the service for others to use and by the users who wish to act together with the service by its automatically generated web interface. Our Grid portal technology offers a framework for supplying single-point access to Grid services, similarly a Web portal such as Yahoo or MSN comprehensive site information, indexes and web pages. A Grid service that is accessible within the portal. A distinctive feature of our grid portal is, a user navigates to the portal page, and afterward, the portal presents the appropriate applications that the user may interact with, derived from their identity and the authorization policies. Like this, a virtual organization may be formed. The Grid is a mixture of network infrastructure and software framework distributing computing services based on distributed hardware and software resources [6].

Users interact with the Grid Portal through an https connection from a web browser. The Grid Portal uses GridSphere to run the portal and Apache Tomcat to run the web-interface. UGP uses MySQL database for the database of information about users, clusters, applications and job status that it needs to run the Portal. The architecture also includes a MyProxy server to store user certificates, a storage server connected to the Grid Portal to provide storage space for poolonly users, and a visualization server to drive the software that is optionally required to provide through-the-web data visualization services to users (see Figure 1).

At the same time as the UGP architecture presents a uniform appearance to users, it provides for a Grid made up of diverse computing environments (hardware, operating systems, job schedulers) and autonomous administrative domains. UGP makes use entirely of open source software: Globus ToolKit, Tomcat, Java, Gridsphere and MySQL. UGP itself is also open source.

In grid portal service, we create schema Web Service Definition Language (WSDL) using tools such as Neatbean IDE, GlassFish as web engine, C compiler and Parallel Virtual Machine (PVM). After finish, clients can access web service portal from a server as called as distributed or grid Computing. Users can access web service portal from a server as called as distributed or grid.

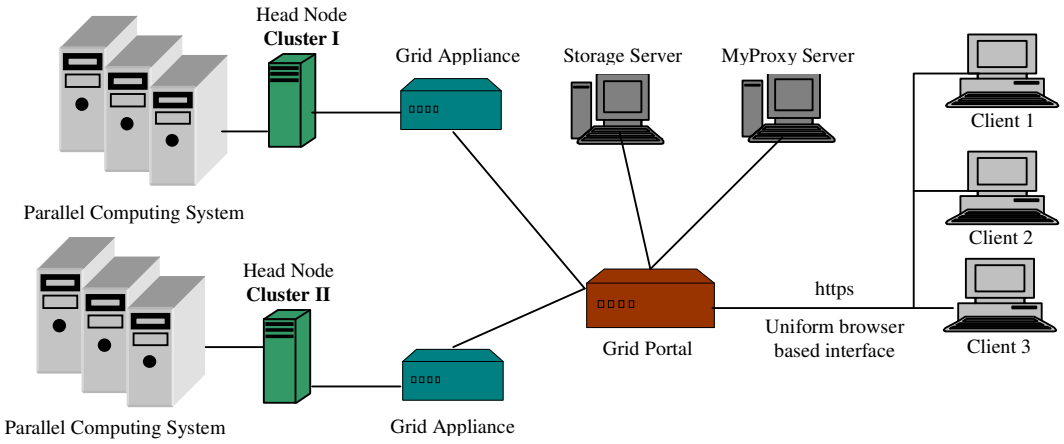


Figure 1: Grid portal technology with parallel computing system

Grid portal supports a framework to provide a web service interface to the existing applications without having to write extra code or modify the existing web services.

2.1 Web Services Technology

Web programming is the design and construction of a program, e.g., an applet, to perform a task on a web page. In the web services development, some concept on GUIs, concurrency; event handling; graphics; network communication; and software engineering techniques and tools are exploited.

For this kind of relationship, we choose service-oriented architecture (SOA) style. Each web services are easily maintainable since there is loose coupling between interacting nodes. The development of this architecture is based on several programming language as it involves algorithm implementation on C, parallelization using Parallel Virtual Machine (PVM) and Java for web services development. The grid computing platform is an open source-based and will be develop under Linux environment. The platform development will increase the acceleration and scaled-out across a virtualized grid. The clusters of processors involved in this platform are developed on increasingly larger computational hardware with inexpensive architecture [7].

Workflow Editing: The workflows can be graphically created at the client machine by the Workflow Editor written as a Java Web-Start application. A simple workflow used in a real-life meteorology application [8] is shown in Figure 2.
https://utmgrid.ibnusina.utm.my:9443/gridsphere/gridsphere

Once press Go / Enter key, the browser will, (1) request *input.htm* from the web server in IIS Server Cluster. The web server will furnish the request by displaying the *input.htm* page. A new user needs to provide several mandatory parameters as requested for the first time in the *input.htm.* The user needs to submit the *input.htm* to the web server via (2) Http Post protocol. The web server will process the request by calling the *PERL-CGI script*. Once called, the script will run as a server background process. After that user will able to sign in and view the list of all clusters as well as list of programmes with different clusters by the provided user name and password. For example, an inserted numerical simulation source code (3) interacts with selected C program in the application directory. The C program will also run as a server background process. The C program process calculates the mathematical problem in the cluster using all the nodes and returns the result to PERL-CGI process. The PERL-CGI process will then terminate the C program process. In another word, the PERL-CGI process spawns C program process and wait for its response. The PERL-CGI process (4) generates a HTML page and embeds the results in it. The generated page will be (5) displayed to the web browser. Files are identified by their logical name and location. It should be also defined if the file is a permanent one or used only temporarily. In the latter case when the job using the temporary file has been finished, the file is automatically removed from the Grid. After creating the workflow on the client machine it should be uploaded to the portal server machine.

3. Service-Oriented Architecture (SOA)

Grid portal services must be located in the bigger framework of Service Oriented Architectures (SOA) that is being used to construct worldwide scalable Grid

systems. SOA offers methods for systems improvement and incorporation where systems group functionality around parallel computing procedures and enclose these as interoperable services. The ideas of SOA are built upon and developing from older perceptions of modular programming and distributed computing [7] [9]. A SOA maintains service amalgamation through published and discoverable interfaces. They can be incorporated in a collection of frameworks as they are message-based relatively attached to an application interface. Our proposed model presents a very flexible programming environment, under the hypothesis that the message semantics do not transform. Administrator and students/ users are open to choose the accomplishment and backend logic and services separately of each other [10].

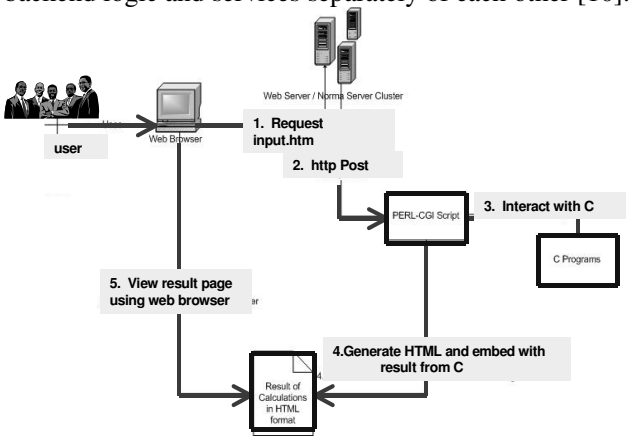


Figure 2: Process Flow of grid portal technology

4. Web Service Paradigm

The web services provided the contents page which will present a synopsis of the selected subject. The user can then either follow a hypertext link to further comprehensive details. It'll provide a parallel programming exercise and the solution can be viewed after the user has completed the exercise. For the pioneer, user can retrieve a solution template. The web services covers most topics excluding domain decomposition technique, data parallelism, concurrency and domain and functional partitioning, message passing paradigm, performance measurements and provide the some numerical libraries in exploiting parallelism for grand challenge applications for the authoring environment.

User will able to perform the following operation through web service paradigm of grid portal,

- Need to share resources among the campus clusters:
  - Better equipment utilization
  - Conserve energy
- Diverse cluster ownership and operation
  - Owners reluctant to give login ids to any but their users
- A number of users have login ids on multiple clusters
  - Need to get to them from one interface/location

Two types of users are supported by UGP:

Cluster Users -- A cluster user has a login id on one or more of the clusters participating in the Grid. A cluster user can get this login id by being a member of a research group that owns one of the clusters. Someone with computational needs can normally also apply for a login id on any cluster that is provided as a campus service.

Cluster users have home directories on each of the clusters they can access. They use their home directories to store files.

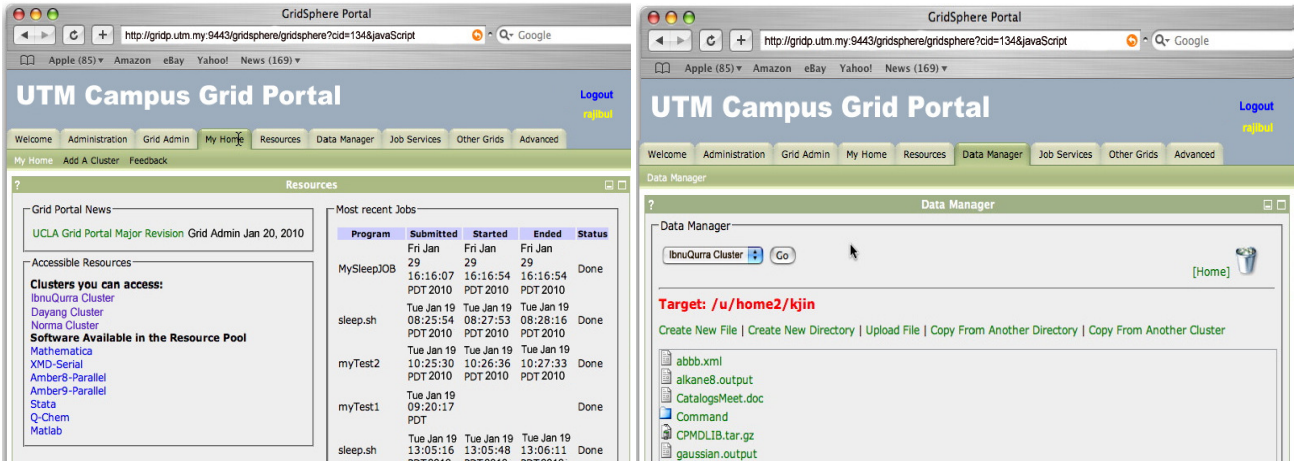
Cluster users can use the Grid Portal to access files on and submit jobs to the clusters they have access to.
Cluster users can also submit jobs to resource pools as a Pool User.
Pool-Only Users – Students, staff, and faculty members who do not have login ids on any of the clusters can easily sign up on the Grid Portal to be Pool-Only Users.
Each Pool-Only User is assigned a storage area on the Storage Server connected to the Grid Portal.
The Pool-Only User can submit jobs to resource pools.

5. Web Service Performance

The process of visualization has becomes extremely fast, reliable and precise with high performance computing. The parallel performance makes the product really attractive because of its high speed, efficient, effectiveness and high temporal performance algorithm [11] [12]. In terms of numerical performance, the result are also precise, highly convergence, stable and accurate to the exact solution.

Low cost High Performance Computer (HPC)

The operating system uses is open source. There is one of the most significant examples of free software in Linux Fedora and open source development, its underlying source code can be freely modified, used, and redistributed by anyone, so long as they fully comply with the GPL License. Linux is one of the most prominent examples of free software and open source development; its underlying source code can be freely modified, used, and redistributed by anyone, so long as they fully comply with the GPL License. The platform is capable in performing a complex computational to solve the real end-to-end solution run on high performance and high-productivity computing.



(a) (b)

Figure 3: (a) Web portal interface for users showing all cluster lists along with list of applications, (b) Web portal interface for users showing contents for sharing

Robustness: The robustness of the software that well suits on any future upgrade distributed memory architecture. Productivity is understood to be a composite of system performance, portability, user friendly, administrative concerns and reduced the "expertise gap".

Open source web based software: The open source product built on Linux platform in web-based format is really famous nowadays. This feature makes the software easy to be reach and access by user at any level instead of providing latest information.

Feature of the software development: The visualization is presented in webPerl-CGI and PHP are emphasized to develop the software instead of MySQL database to store significant information. The productivity is understood to be a composite of system performance, system robustness, programmability, portability, and administrative concerns.

Real time solution: The mathematical modeling grants user the accurate prediction of engineering problems on a real time solution. This includes efficient visualization between mathematical simulation and exact solution of engineering problems.

Table 1: Comparison between open source and price based software

Open source (free)	Price based on version
Capture the smooth graph	Coarse graph
Highly convergent to exact solutions	Slowly convergent
Simulation based on web portal	Simulation is not provided
Server is provided	Server is not provided
High speed	Low speed
High performance of parallel computing	Low performance of sequential computing

Based on the Grid Portal with distributed parallel computer systems in University Technology Malaysia (see Figure 3), we have been developing a numbers of software to assist users in numerical field and software engineer in manufacturing industries. The comparison with price based software is as follows:

This research will provide the following benefits to students and participating universities:

- Facilitate and support work group students in their design and developing a courseware.
- The successful application of authoring activities environment through grid technology provides enhancements in work group performance, helps to lower cost, and encourages innovation.
- Learners and faculties can promote the exchange of ideas, information, knowledge, and joint research and development of Web-based teaching materials.
- Help member universities build a network of facilitators to support e-learners (forum with advanced Information and Communication Technology (ICT), i.e., with the use of massive parallel processors of globally distributed and yet interconnected mini-supercomputers through global neural computer network).
- Researchers can partner with colleagues in more advanced faculties, and perform joint collaborative research and development with the use of the emerging global GRID computer networking technology.

6. Conclusion

Grid technology defines as a new powerful computing paradigm by analog to the eletric Power Grid. Users of

the GRID will then be able (a) to use his/her private workplace to invoke any application from a remote system, (b) to use the best suited system for executing their desired particular application, (c) to access data securely and consistently from remote sites, (d) to exploit multiple systems to complete complex tasks in design or authoring a courseware, or (e) to use multiple systems to solve large problems that exceed the capacity of a single one. In this vision, the sharing doesn't mean simply exchange of data or files but rather a concrete access to resources (especially design and authoring a courseware in a group in parallel fashion).

This paper described how the proposed grid portal authoring activities infrastructure and workflow solutions were integrated by SOA, NetBean in order to achieve the goal of this study. The grid portal technology with parallel computing system represents an effective upgraded approach to deploy C and JAVA programme code applications as Grid services. The more pervasive take-up of Grid technology requires high-level Grid application environments where users can easily create complex Grid workflows including different Grid enabled applications. The user only has to provide several mandatory parameters to an HTML pages in PERL-CGI Scripts based Code Interface Description File and PERL-CGI enables the code application to be run from a Grid service client. All these C and JAVA programme codes were executed from a single workflow and the execution output was visualized by the portal. We believe that by this study, an efficient authoring activities based collaborative learning environment will be created as well as developing strong foundation in parallel computing which emphasizes on theories and hands-on activities to the market potential. Based on the strong foundations, hopefully the users are ready to apply their knowledge, creativity and leadership to fulfil the need of their future career development.

Acknowledgment

The authors gratefully acknowledge the financial assistance from the Research Management Centre (RMC), UTM as well as the University Technology Malaysia to provide the facilities for the research.

Reference

[1] Begoña Ferrero, Maite Martín, Ainhoa Alvarez, Maite Urretavizcaya and Isabel Fernández-Castro, Authoring and Diagnosis of Learning Activities with the KADDET Environment, Journal of Universal Computer Science, vol. 11, no. 9, 2005, pp. 1530-1542.

[2] D. Dicheva, L. Aroyo & A. Cristea, Collaborative Courseware Authoring Support, CATE'02, ACTA Press, ISBN:0-88986-332-6, 2002, pp.52-57.

[3] D. Garrison and W. Archer, A transactional perspective on teaching and learning: A framework for adult and higher education. Pergamon, UK, 2000.

[4] K. Illeris, The three dimensions of learning. Roskilde University Press, Denmark, 2002.

[5] G. Wells, Dialogic inquiry: Towards a sociocultural practice and theory of education. Cambridge University Press: US, 1999.

[6] Bing Wu, Matthew Dovey, Muan Hong Ng, Kaihsu Tai1, Stuart Murdock, Hans Fangohr, Steven Johnston, Paul Jeffreys, Simon Cox, Jonathan W. Essex and Mark S.P. Sansom, A Web / Grid Portal Implementation of BioSimGrid: A Biomolecular Simulation Database, Journal of Digital Information Management, Vol. 2, No. 2, 2004, pp. 74-78.

[7] Zamoya, A.~Y., Parallel and Distribution Computing Handbook, McGraw Hill, 1996.

[8] Kacsuk, P., Goyeneche, A., Delaitre, T., Kiss, T., Farkas Z. and Boczko, T., High-Level Grid Application Environment to Use Legacy Codes as OGSA Grid Services, In Proc. of Fifth IEEE/ACM International Workshop on Grid Computing (GRID'04), pp. 428-435.

[9] Jakob, the POVBench Parallel Processor Benchmark. Retrieved (March 3, 2004), from http://www.haveland.com/ index.htm?povbench/index.htm

[10] Thomas, M.P., Burruss, J., Cinquini, L., Fox, G., Gannon, D., Gilbert, L., von Laszewski, G., Jackson, K., Middleton, D., Moore, R., Pierce, M., Plale, B., Rajasekar, A., Regno, R., Roberts, E., Schissel, D., Seth, A., Schroeder, W., Grid Portal Architectures for Scientific Applications. Journal of Physics: Conference Series, vol. 16, 2005, pp. 596–600.

[11] Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R., Sunderam, V., PVM: Parallel Virtual Machine & User's Guide and Tutorial for Networked Parallel Computing, MIT Press, Cambridge, Mass, 1994.

[12] Lewis,T.G. and EL-Rewini,H., Distributed and Parallel Computing, Manning Publication, USA, 1998.