SUPPORTING EFFECTIVE AND USEFUL WEB-BASED DISTANCE LEARNING

Santi Caballé, Thanasis Daradoumis
Department of Computer Science, Multimedia, and Telecommunication, Open University of Catalonia
Rbla. Poblenou, 156, 08018, Barcelona, Spain

Fatos Xhafa, Joan Esteve
Department of Languages and Informatics Systems and Computing Laboratory, Faculty of Informatics of Barcelona
Polytechnic University of Catalonia, Jordi Girona 1-3, 08034 Barcelona, Spain

Keywords: Web-based Education, User Modelling, Grid Computing, Distributed and Parallel Applications.

Abstract: Learners interacting in a Web-based distance learning environment produce a variety of information elements during their participation; these information elements usually have a complex structure and semantics, which makes it rather difficult to find out the behavioural attitudes and profiles of the users involved. User modelling in on-line distance learning is an important research field focusing on two important aspects: describing and predicting students' actions and intentions as well as adapting the learning process to students' features, habits, interests, preferences, and so on. This work provides an approach that can be used to greatly stimulate and improve the learning experience by tracking the students' intentions and helping them reconduct their actions that could evolve accordingly as the learning process moves forward. In this context, user modelling implies a constant processing and analysis of user interaction data during long-term learning activities, which produces large and considerably complex information. In this paper we show how a Grid approach can considerably decrease the time of processing log data. Our prototype is based on the master-worker paradigm and is implemented using a peer-to-peer platform called Juxtacat running on the Planetlab nodes. The results of our study show the feasibility of using Grid middleware to speed and scale up the processing of log data and thus achieve an efficient and dynamic user modelling in on-line distance learning.

1 INTRODUCTION

Online learning constitutes a complex task both as regards its utility and its capability to reach an effective and useful learning. The more and more frequent employment of constructive learning requires the support and use of modern educational tools that allow learners to be aware of their own knowledge-building activities and other learners' collaboration in order to achieve effective learning and performance (Horton, 2000).

Information and communication technologies have shown a significant advance and fast progress as regards performance and usability. At the same time, new educational needs demand more innovative ways to work, collaborate, and learn. These aspects justify the grown interest and development of technology for knowledge management, mobility and conceptual awareness in the area of distributed collaborative work and learning (Begole et al., 2002).

As a consequence of the complex processes involved in learning, we need to capture all and each type of possible data gathered in log files. However, the information generated in web-based learning applications can be of a great variety of type and formats (Xhafa et al., 2004). Moreover, these applications are characterized by a high degree of user-user and user-system interaction which stresses the amount of interaction data generated. Therefore, there is a strong need for powerful solutions that record the large volume of interaction data and can be used to perform an efficient interaction analysis and knowledge extraction.

Caballé S., Daradoumis T., Xhafa F. and Esteve J. (2007).
SUPPORTING EFFECTIVE AND USEFUL WEB-BASED DISTANCE LEARNING.
In *Proceedings of the Third International Conference on Web Information Systems and Technologies - Society, e-Business and e-Government / e-Learning*, pages 536-539
DOI: 10.5220/0001284105360539

As a matter of fact, the computational cost is the main obstacle to processing data in real time. Hence, in real learning situations, this processing tends to be done offline so as to avoid harming the performance of the logging application, but as it takes place after the completion of the learning activity has less impact on it (Caballé et al., 2005).

Based on the Grid vision (Foster and Kesselman, 1998), a preliminary study was conducted (Xhafa et al., 2004). This study showed that a parallel approach based on the Master-Worker (MW) paradigm might increase the efficiency of processing a large amount of information from user activity log files (for more information about MW, please follow the link: http://www.cs.wisc.edu/condor/mw).

In this paper we show, first, the main challenges to be faced in modelling students' behaviour in Web-based learning environments, and, then, how a Grid-based approach can deal with them. In order to show the feasibility of our approach, we use the log data from the internal campus of the Open University of Catalonia (the UOC is found at: http://www.uoc.edu), though our approach is generic and can be applied for reducing the processing time of log data from web-based applications in general.

Our ultimate objective is to make it possible to continuously monitor and adapt the learning process and objects to the actual students' learning needs as well as to validate the campus' usability by analyzing and evaluating its actual usage.

2 MODELING STUDENTS' BEHAVIOR IN WEB-BASED DISTANCE LEARNING

Our real web-based learning context is the Open University of Catalonia (UOC), which offers distance education through the Internet in different languages. As of this writing, about 40,000 students, lecturers, and tutors from everywhere participate in some of the 23 official degrees and other PhD and post-graduate programs, resulting in more than 600 official courses. The campus is completely virtualized. It is made up of individual and community areas (e.g. personal electronic mailbox, virtual classrooms, digital library, on-line bars, virtual administration offices, etc.), through which users are continuously browsing in order to fully satisfy their learning, teaching, administrative and social needs.

From our experience at the UOC, the description and prediction of our students' behaviour and navigation patterns when interacting with the campus is a first issue. Indeed, a well-designed system's usability is a key point to stimulate and satisfy the students' learning experience. In addition, the monitoring and evaluation of real, long-term, complex, problem-solving situations is a must in our context. The aim is both to adapt the learning process and objects to the actual students' learning needs as well as to validate the campus' usability by monitoring and evaluating its actual usage.

In order to achieve these goals, the analysis of the campus activity and specifically the users' traces captured while browsing the campus is essential in this context. The collection of this information in log files and the later analysis and interpretations of this information provide the means to model the actual user's behaviour and activity patterns.

2.1 The Collection of Information from On-line Learning Activity

The on-line web-based campus of the UOC is made up of individual and community virtual areas such as mailbox, agenda, classrooms, library, secretary's office, and so on. Students and other users (lecturers, tutors, administrative staff, etc.) continuously browse these areas where they request for services to satisfy their particular needs and interests. For instance, students make strong use of email service so as to communicate with other students and lecturers as part of their learning process.

All users' requests are chiefly processed by a collection of Apache web servers as well as database servers (Apache is found at: http://httpd.apache.org) and other secondary applications, all of which provide service to the whole community and thus satisfy a great deal of users' requests. For load balance purposes, all HTTP traffic is smartly distributed among the different Apache web servers available. Each web server stores in a log file all users' requests received in this specific server as well as the information generated from processing the requests. Once a day (namely, at 01:00 a.m.), all web servers in a daily rotation merge their logs producing a single very large log file containing the whole user interaction with the campus performed in the last 24 hours.

A typical daily log file size may be up to 10 GB. This great amount of information is first pre-processed using filtering techniques in order to remove a lot of futile, non relevant information (e.g. information coming from automatic control processes, the uploading of graphical and format elements, etc.). However, after this pre-processing,

about 1.8 GB of potentially useful information corresponding to 3,500,000 of log entries in average still remains (Carbó at al., 2005).

For the purpose of registering the campus activity, log files entries were set up with the purpose of capturing the following information: who performed a request (i.e. user's IP address along with a session key that uniquely identifies a user session); when the request was processed (i.e. timestamp); what type of service was requested (a URL string format description of the server application providing the service requested along with the input values) and where (i.e. an absolute URL containing the full path to the server application providing the service requested).

At this point, we point out some problems arisen by dealing with these log files. Each explicit user request generates al least an entry in the log file and after being processed by a web server, other log entries are generated from the response of this user request; certain non-trivial requests (e.g. user login) involve in turn requesting others and hence they may implicitly trigger new log entries; the what and where fields contain very similar information regarding the URL strings that describe the service requested and the parameters with the input values; certain information is found in a very primitive form and is represented as long text strings (e.g. user session key is a long 128-character string).

Therefore, there is a high degree of redundancy, tedious and ill-formatted information as well as incomplete as at some cases certain user actions do not generate any log entry (e.g. user may leave the campus by either closing or readdressing the browser) and have to be inferred. As a consequence, treating this information is very costly in terms of time and space needing a great processing effort.

3 AN EFFICIENT PROCESSING OF LOG DATA

In order to deal with the above mentioned problems and inconvenients, we have developed a simple application in Java, called *UOCLogsProcessing* that processes log files of the UOC. In particular, this application runs offline on the same machine as the logging application server. It uses, as an input, the daily log files obtained as a result of merging all web servers' log files. The following process is run: (i) identify the log entries boundaries and extract the fields that make up each entry, (ii) capture the specific information contained in the fields about users, time, sessions, areas, etc., (iii) infer the missing information, (iv) map the information obtained to typed data structures, and (v) store these data structures in a persistent support.

However, as the processing is done sequentially, it takes too long to complete the work and it has to be done after the completion of the learning activity, which makes the construction of effective real-time user models not possible.

3.1 Juxta-CAT: a JXTA-based Grid Platform

In this section, we briefly introduce the main aspects of the grid platform, called Juxta-CAT (Esteve and Xhafa, 2006), which we have used for parallelizing the processing of log files.

The Juxta-CAT platform has been developed using the JXTA protocols and offers a shared Grid where client peers can submit their tasks in the form of java programs stored on signed jar files and are remotely solved on the nodes of the platform (JXTA web page is found at: http://www.jxta.org). The architecture of Juxta-CAT platform is made up of two types of peers: *common client peers* and *broker peers*. The former can create and submit their requests using a GUI-based application while the later are the administrators of the Grid, which are in charge of efficiently assigning client requests to the Grid nodes and notify the results to the owner's requests. To assure an efficient use of resources, brokers use an allocation algorithm, which can be viewed as a price-based economic model, to determine the best candidate node to process each new received petition.

The Juxta-CAT platform has been deployed in a large-scale, distributed and heterogeneous P2P network using nodes from PlanetLab platform (PlanetLab web page is found at http://www.planet-lab.org). At the time of this writing, PlanetLab's node distribution is: 715 nodes over 338 sites (Polytechnic University of Catalonia has joined PlanetLab with several proper nodes). Juxta-CAT Project and its official web site have been hosted in Java.NET community (please see the following link: https://juxtacat.dev.java.net).

3.2 Experimental Results

We present, in this section, the main experimental results obtained for a test battery in order to measure the efficiency obtained by the grid processing. This battery test uses both large amounts of log information (i.e. daily log files) and well-stratified short samples consisting of representative daily periods with different activity degrees (e.g. from 7

p.m. to 1 a.m. as the most active lecturing period and from 1 a.m. to 7 a.m. as the period with least activity in the campus). On the other hand, other tests involved a few log files with selected file size forming a sample of each representative stratum. This allowed us to obtain reliable statistical results using an input data size easy to use.

The battery test was processed by the *UOCLogsProcessing* application executed on single-processor machines involving usual configurations. The battery test was executed several times with different workload in order to have more reliable results in statistical terms involving file size, number of log entries processed and execution time along with other basic statistics. On the other hand, the same battery test was processed by Juxta-CAT using different number of nodes, specifically, 2, 4, 8, and 16 nodes, using PlanetLab nodes. A sample of the results is shown in Figure 1, while Table 1 shows the gain in parallel speed-up and efficiency we achieved

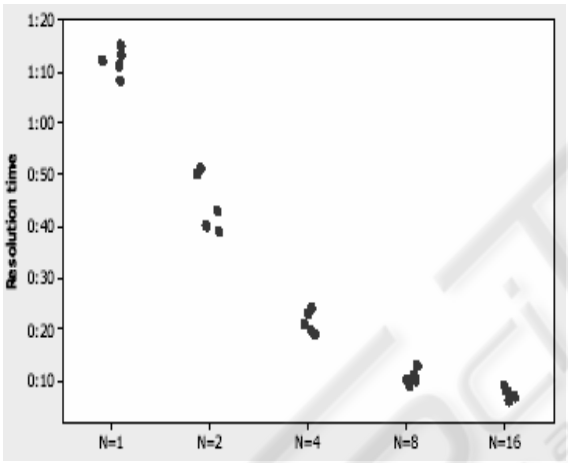

Figure 1: An execution time result for log files with sizes of 36MB; x-axis indicates the number of processors and y-axis the processing time (mm:ss).

Table 1: Parallel speed-up and efficiency.

Log file size	Speed-up	Efficiency
12 MB	6.1	38.2 %
24 MB	7.4	46,2 %
36 MB	9.1	56.8 %

4 CONCLUSIONS AND FURTHER WORK

In this paper, we have shown how to model the learner's behaviour and activity pattern by using user modelling tracking-based techniques. However, the information generated from tracking the learners when interacting with the virtual learning environment is usually very large, tedious, redundant and ill-formatted and as a result processing this information is time-consuming. In order to overcome this problem, in this paper we have proposed a Grid-aware implementation that considerably reduces the processing time of log data and makes it possible to build and constantly maintain user models, even in real time. For the purposes of both showing the problem of dealing with log data and testing our grid prototype we have described and used the log data coming from the virtual campus of the Open University of Catalonia.

Further work will include the implementation of a more thorough mining process of the log files, which due to the nature of the log files of our virtual campus will require more processing time in comparison to the log processor used in this work.

ACKNOWLEDGEMENTS

This work has been partially supported by the Spanish MCYT project TSI2005-08225-C07-05.

REFERENCES

Begole, J.; Tang, J.; Smith, R.; Yankelovich, N., 2002. Work rhythms: analyzing visualizations of awareness histories of distributed groups. Computer Supported Cooperative Work; In proceedings of the *2002 ACM conference on Computer Supported Cooperative Work*, p 334 - 343; New Orleans, Louisiana, USA.

Caballé, S., Paniagua, C., Xhafa, F., and Daradoumis, Th., 2005. A Grid-aware Implementation for Providing Effective Feedback to On-line Learning Groups. In: *proc. of the GADA'05*, Cyprus.

Carbó, JM., Mor, E., Minguillón, J., 2005. User Navigational Behavior in e-Learning Virtual Environments. *The 2005 IEEE/WIC/ACM International Conference on Web Intelligence*'05, pp. 243-249.

Esteve J., Xhafa F., 2006. Juxta-CAT: A JXTA-based Platform for Distributed Computing. *The ACM International Conference on Principles and Practice of Programming in Java'06*.

Foster, I. and Kesselman, C., 1998. *The Grid: Blueprint for a Future Computing Infrastructure*. pp. 15-52. Morgan Kaufmann, San Francisco, CA.

Horton, W., 2000. *Designing Web-Based Training*. Ed. Wiley, New York.

Xhafa, F., Caballé, S., Daradoumis, Th. and Zhou, N., 2004. *A Grid-Based Approach for Processing Group Activity Log Files*. In: proc. of the GADA'04, Cyprus.