Efficient Enabling of Real Time User Modeling in On-line Campus

Santi Caballé, Fatos Xhafa, Thanasis Daradoumis and Raul Fernandez

Open University of Catalonia, Department of Computer Science and Multimedia
Av. Tibidabo, 39-43, 08035 Barcelona, Spain
{scaballe,fxhafa,adaradoumis,rfernandezco}@uoc.edu

Abstract. 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. The aim is to greatly stimulate and improve the learning experience. In this context, user modeling implies a constant processing and analysis of user interaction data during long-term learning activities, which produces large and considerably complex information. As a consequence, processing this information is costly and requires computational capacity beyond that of a single computer. In order to overcome this obstacle, in this paper we show how a parallel processing approach can considerably decrease the time of processing log data that come from on-line distance educational web-based systems. The results of our study show the feasibility of using Grid middleware to speed and scale up the processing of log data and thus achieving an efficient and dynamic user modeling in on-line distance learning.

1 Introduction

User modeling [1] is a mature research field mostly involved in the information technology context. It is mainly utilized in software systems for inferring the users' goals, skills, knowledge, needs and preferences and thus achieving more adequate adaptation and personalization on the basis of the user activity pattern built. This inference process relies in turn on being able to track the users' actions when interacting with the application such as the users' choice of buttons and menu items [2].

In this paper, we focus on and are interested in web-based applications that support on-line distance learning. These applications, due to the high degree of user interaction, take great advantage of the tracking-based techniques of user modeling such as providing broader and better support for the users of Web-based educational systems [2]. Indeed, the data analysis of the information captured from the actions performed by learners is a core function for the modeling of the learner's behavior during the learning process and of the learning process itself as well. In addition, the building of learner models may help identify navigation patterns and adapt the system's usability to the actual learners' needs resulting in a great stimulation of the learning experience. However, the information generated in web-based learning applications can be of a

great variety of type and formats [3]. 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.

Based on this vision, a preliminary study was conducted [3] to show that a Grid [4] approach might increase the efficiency of processing a large amount of information from user activity log files. In order to show the feasibility of our approach, we used the log data from the internal campus of the Open University of Catalonia though it can be applied for reducing the processing time of log data from web-based application 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 Settings: the Case of the Open University of Catalonia

Our real web-based learning context is the Open University of Catalonia (UOC) [5] which offers distance education through the Internet in different languages. As of this writing, about 40,000 students, lectures 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.

From our experience at the UOC, the description and prediction of our students' behavior and navigation patterns when interacting with the virtual 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 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 behavior and activity patterns. However, in the context of the UOC, the whole user interaction generates a great amount of information a day (about 10 GB) which is filtered and collected in large daily log files. Furthermore, this large information is found in an ill-structured highly redundant form needing a great amount of computational power to constantly process log data [5]. As a matter of fact, the computational cost is the main obstacle to processing this data in real time [3] and hence in our real situation this processing tends to be done offline in order to avoid harming the performance of the logging application, but as it takes place after the completion of the learning activity it has less impact on it.

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. 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.

The distributed platform has been developed using the JXTA [7] 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. The architecture of the JXTA platform is made up of two types of peers: *common client peers* and *broker peers*. The former can create and submit their requests 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 different allocation algorithms, which can be viewed as economic models, to determine the best candidate node to process each new received request. The implementation and design of peers, groups, job and presence discovery, pipe-based messaging, etc. are developed using the latest updated JXTA libraries [7]. This distributed platform has been deployed in a large-scale, distributed and heterogeneous P2P network using nodes from PlanetLab[1] platform.

3.1 Parallelizing the Processing of Log Files

The parallel implementation follows the Master-Worker (MW) [8] paradigm. In a nutshell, the log file is split off into a certain number of parts, which can be exactly equal to the number of grid nodes (slaves) that will participate in the processing or can be larger. In this later case some peer nodes could receive more than one part for processing. By splitting the original file into more parts than peer slave candidates for processing, we can achieve different degrees of granularity of the parallel processing. Achieving different degrees of granularity is very desirable in Grid environments given the high heterogeneity of computing resources. Note that we have a perfect split of the problem in many independent parts. In the end, the master node just needs to append to a unique file the arriving of partial solutions (partial result files after processing). The main steps of the MW parallel algorithm to process a log file in the JXTA platform are as follows:

1. [**Pre-processing phase**]: *UOCLogsProcessing* counts the total number of lines of the log file, totalNbLines, and knowing the total number of parts to split the file off, nbParts, each peer node will receive and process a totalNbLines/nbParts of lines from the file.

[1] http://www.planet-lab.org. As of Feb. 24, 2007, PlanetLab consists of 755 nodes at 363 sites.

2. [**Master Loop**]: Repeat
    a. Read totalNbLines/nbParts lines from the original file and create a file with them.
    b. Create a request and submit it to JXTA platform
    c. [**Juxta-cat processing**]:
        i. The request is received by a broker of JXTA platform and it is assigned to a peer node of the platform.
        ii. The peer node, upon receiving and accepting the request, notifies it to the Broker node.
        iii. The peer node receives the corresponding part of the file to process by direct JXTA transfer from the Master node.
        iv. The peer runs *UOCLogProcessing* functionality for processing the lines of the file, one at a time, and stores the results of the processing in a buffer.
        v. The peer node, once the processing of the request is done, sends back to the master node the content of the buffer.

    Until the original log file has been completely scanned.
3. [**Master's final phase**]: Receive messages (partial files) from peers and append in the correct order the newly received resulting file to the final file containing the information extracted from the original log file.

3.2 Experimental Results

In this section we present the experimental results from measuring the speedup obtained by the grid processing. Battery test involved 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. In addition, other tests included 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.

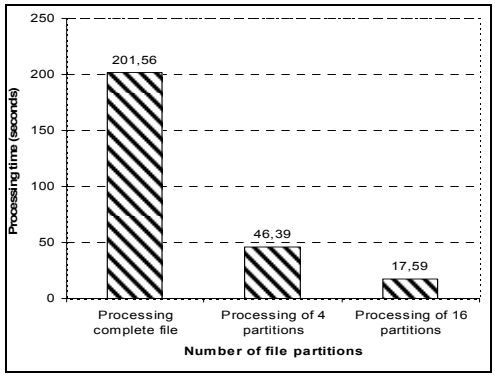

Fig. 1. Processing times of a log file of 100Mb for the case of processing without partitions and by partitioning into 4 and 16 parts, respectively, resulting in a speed-up of 0,543 and 0,71 respectively.

The battery test was processed by the *UOCLogsProcessing* application executed on single-processor machines involving usual configurations. Moreover, it 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. The same battery test was processed by JXTA platform using 8 peer nodes and by considering 4, 8 and 16 parts of the original file.

Parallel efficiency and speed up are then computed involving the number of grid nodes and the time needed by the grid to process each log file. Fig. 1 shows the considerable decrease in execution time we achieved using the JXTA platform.

4 Conclusions and Further Work

In this paper, we have shown how to model the learner's behavior and activity pattern by using user modeling tracking-based techniques. However, the information generated from tracking the learners is usually very large, tedious, and ill-formatted and as a result processing this information is time-consuming. In order to overcome this problem, we have proposed a Grid-aware implementation that considerably reduces the processing time of log data and allow us to build and constantly maintain user models.

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

1. Bushey, R., Mauney, JM., and Deelman, T. (1999). The Development of Behaviour-Based User Models for a Computer System. In Judy Kay (ed.), User Modeling: Proceedings of the UM99. Springer Wien New York, pp. 109-118
2. Gaudioso, E., Boticario, J.G. (2003). Towards web-based adaptive learning communities. Proceedings of Artificial Intelligence in Education 2003, Sydney, Australia. IOS Press.
3. 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.
4. Foster, I. and Kesselman, C. The Grid: Blueprint for a Future Computing Infrastructure. Morgan Kaufmann, San Francisco, CA, 1998. pp. 15-52
5. Open University of Catalonia: http://www.uoc.edu (web page as of February 2007)
6. 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 (WI'05), pp. 243-249
7. JXTA: http://www.jxta.org/ (web page as of February 2007)
8. Master-Worker: http://www.cs.wisc.edu/condor/mw/ (web page as of February 2007)