

Optimizing the Access Records of Students in the Moodle Virtual Learning Environment Database

Eliakim Zacarias¹, Luciano R. de Almeida¹, Juliano B. Prettz¹, João Paulo C. L. da Costa^{1,2,3}, Edison P. de Freitas⁴, Edna D. Canedo¹, and Rafael T. de Sousa Júnior¹

¹Department of Electrical Engineering, University of Brasília, Brazil

²Institute for Information Technology, Ilmenau University of Technology, Ilmenau, Germany

³Fraunhofer Institute for Integrated Circuits IIS, Erlangen, Germany

⁴Institute of Informatics, Federal University of Rio Grande do Sul, Brazil

edison.pignaton@inf.ufrgs.br, ednacanedo@unb.br, eliakim.zacarias@gmail.com, joaopaulo.dacosta@ene.unb.br, prettzb@gmail.com, luciano.ralm@gmail.com, desousa@unb.br

Abstract: Moodle is a freeware solution for Learning Management System (LMS) applied to Virtual Learning Environments (VLS). Despite the several features to support online courses, some desirable information is not easily provided, for instance, the amount of time the user spent in the system or in a specific course. A custom implementation can measure the time spent on the user's searches into the internal Moodle log and the distance between the records through the session time. However, for a large number of recorded logs, searching a single record passing through the other records requires a large time processing. As a consequence, the generation of custom reports is also costly. This work proposes a real-time plugin for Moodle to count the time spent by a student logged into the system retrieving the information from a pre-computed central table. The objective of the solution is to improve the counting precision and allow a faster retrieval of data. Our proposed plugin is validated using real data from the National School of Public Administration (Enap) in Brazil.

© 2016, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Generation of Reports, Moodle Plugin, Time Counting Precision, Measuring Online Time.

1. INTRODUCTION

Although Moodle [1] is a popular platform for distance learning environments, there are still essential limitations such as the lack of tracking tools for the students. However, these limitations can be overcome by adding third-party solutions such as plug-ins or even a whole separate system [2].

The student tracking data is crucial to generate reports monitoring the student's behavior in the offered courses. Moodle has available the user activity log, which can be used to control and to capture relevant tracking information.

The National School of Public Administration, in Portuguese Escola Nacional de Administração Pública (Enap), provides distance learning courses for the federal employees in Brazil using Moodle. In order to support the students to improve their performance, Enap used an ad hoc solution that seeks all access records of selected students in a certain course. Search loops were repeated for all students passing through the activity log data by filtering course from the course start date until the end date. Despite the feasibility in small databases, such approach is unfeasible if the activity log increases. Therefore, the total query time was proportional to the amount of selected students.

This paper aims to provide a solution to the problem by standardizing data and separating them from the activity logs so the query can be made by consulting a single record.

This paper is divided into four sections besides this introduction. Section 2 presents and discusses the state of the art related to the subject. Section 3 presents the proposed approach to address the problem, while Section 4 presents the acquired results. Section 5 concludes the paper and also provides directions for future work.

2. RELATED WORKS

Moodle [1] is one of the leading platforms for free distance learning. It is an open source system that includes the participation of collaborative communities for their development. Its structure is formed by modules or extensions, smaller systems that provide specific or specialized features. These modules are also called plug-ins [2].

NAGI and SUESAWALUK [3] present an analysis based on reports of Moodle to measure the level of interactivity in the virtual environment of teaching courses at the University of the Assumption. The aim of [3] is the use of data captured behavior and iteration of students in the Moodle environment. These pieces of information are collected in real-time, including information that is not registered in the database, like views and clicks. One of the conclusions in [3] is that a communication platform to improve the interaction between students and instructors can be created and a ranking of the courses with more iterations and collaborations of students can be also made.

In BOVO, 2013 [4] an analysis of the log of the virtual system Moodle learning is made. With data mining techniques, machine learning and artificial intelligence indicates which groups are the most relevant to form support tools for managers and teachers. Such data mining techniques can also predict the behavior of the students and their performance. It can also allow the teachers and mentors trace paths and counseling so that students get better results.

In [5], managers and teachers needed the complete information on the behavior of students, in order to analyze and propose ways to improve student achievement. To meet this demand the Merlin was developed, composed of three layers. The first layer contains the data of the education system. The second layer consists of tools that analyze and form a database of system logs, cookies, logins on the system and polls of answers, all with the function to track and record the actions of students during activities in the virtual system. The third layer is the presentation of reports based on pooled data from data mining tools.

3. THE PROPOSED METHOD

The proposal of this paper seeks to avoid dependence on activity logs and centralize the storage of already processed data. Thus, the search of the required information is held in a separate table, which links the students and their total active time.

Since the solution does not require a history of accesses, but only the final total time the user remained active until completion of the course, the system keeps only the total sum of minutes. Thus, the user time registration is required, along with another record connecting the user and the course, storing the sum of this.

Considering the impossibility to increase the server's hardware capabilities for this case, and also the impossibility to wait for the full-time report generation that could exceed 10 minutes, the solution aims to replace the intensive processing load during the report generation through periodic requests. At the end of each minute, an asynchronous request is sent informing that the user remains accessing the system. Such a request is allowed through technology for Asynchronous Javascript and XML (AJAX), implemented in browsers as the XMLHttpRequest class, in the first level of development [6].

The AJAX technology is possible and accessible to browsers that support the Moodle system since version 2.3. Since Moodle itself requires JavaScript for its better functioning, the solution does not interfere with the usability and requirements, taking advantage of resources already offered.

To implement the solution, two parts were needed. A Javascript is added to a standard content to all pages, usually to the theme customization files. The code creates an event that happens every minute, which makes a call to a specific particular page on the server.

As shown in Figure 1, the Javascript code includes the header or footer of pages of Moodle, to print and run on all system pages. Inclusion is usually performed in the theme customization files. Then, an event is initiated and repeated at

a predefined interval as the minimum time that should be considered for user access, also in view of each iteration, a request to the server is made. The code makes the request to a specific page.

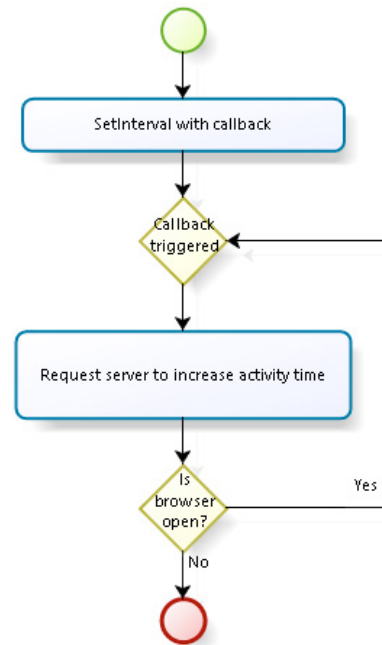


Fig. 1. Browser process: Diagram of the browser part of the plugin.

Figure 2 shows the process triggered in the server for each request so that the server is responsible for storage of the student's access time. Every time the server is requested, the records belonging to the user and the full system access are rescued. Next the minimum required time since the last update is checked, and if it is higher, the past period is added.

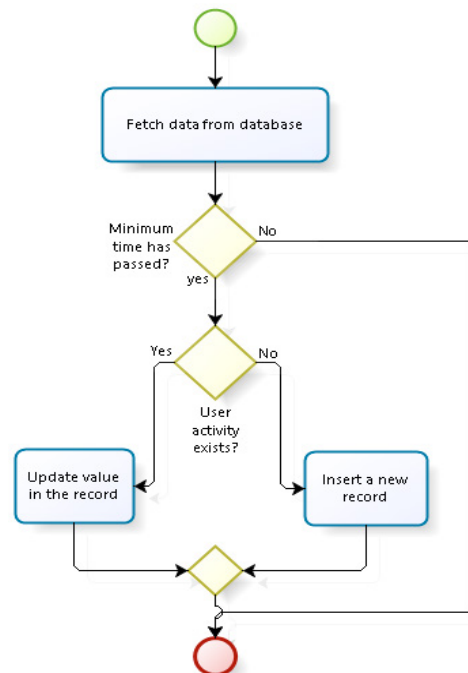


Fig. 2. Server process: Diagram of the server part of the plugin.

Download English Version:

<https://daneshyari.com/en/article/5001946>

Download Persian Version:

<https://daneshyari.com/article/5001946>

[Daneshyari.com](https://daneshyari.com)