# Integrating Moodle with a Postgres-XC cluster database providing high availability and high performance at a low cost

Afonso Bione a

Rodrigo de Souza b, [[1]](#footnote-1)

Unidade Acadêmica de Educação à Distância e Tecnologia

Universidade Federal Rural de Pernambuco

Brazil

a [aagbione@gmail.com](mailto:aagbione@gmail.com), b [rsouza@deinfo.ufrpe.br](mailto:rsouza@deinfo.ufrpe.br)

**Abstract:** We study an integration of Moodle, a Virtual Learning Environment (VLE), with the distributed database system Postgres-XC, aiming at the construction of a pool of computers connected in parallel to ensure performance, as required by Moodle, at a low cost. The methodology used was the installation of 10 small computers in parallel, all running the free operating system Linux, forming a parallel processing server. As a result, we obtain an open, accessible and cheap platform for running VLE’s. This prototype has undergone real-world testing and yielded good results concerning usability, availability and performance of the VLE.

# Introduction

The main issue addressed by this research lies in the challenges faced by some educational institutions for the deployment of a Distance Education (DE) infrastructure, mainly in what concerns cost. It is well known that Virtual Learning Environments (VLE) require heavy hardware in order to assure acceptable performance in the context of hundreds of students (or much more, such as in Massive Open Online Courses – MOOC’s) operating the system simultaneously. This is true in particular for Moodle (<https://moodle.org/>), one of the most popular VLE’s, due inter alia to a large community of developers and the fact that it is open source. In Brazil, where this research has been developed, plenty of educational institutions use Moodle for on-line courses; this is the case for example of the EADTec Department at Universidade Federal Rural de Pernambuco (UFRPE) that hosts a number of DE based undergraduate courses. However, few of these institutions can afford the recommended minimal hardware to support the aimed public.

“Moodle has been designed to be compatible, flexible and easy to modify. It has been written using the popular and powerful PHP language, which runs on any computer platform with a minimum of effort, allowing teachers to set up their own servers using their desktop machines. Moodle is built in a highly modular fashion and uses common technologies such as shared libraries, abstraction, and Cascading Style Sheets to define the interfaces (while still working on old browser technology)” (Dougiamas, 2003).

“In a distributed environment data is replicated in order to achieve shorter response times, higher throughput and increased availability and reliability in case of failures.In a distributed environment data is replicated in order to achieve shorter response times, higher throughput and increased availability and reliability in case of failures” (Nicola, 2003). We propose a solution based on the integration of Moodle with the Postgres-XC distributed database management system (<http://sourceforge.net/projects/postgres-xc/>) running over a Linux cluster of cheap computers. As a result, we obtain an affordable, powerful and scalable platform, which most universities can build from existent equipment. To the best of our knowledge, no similar solution (involving Moodle and Postgres-XC) has been proposed in the literature of the subject. The choice of Postgres-XC is natural for, besides being open source, it is a clone of the popular Postgres database management system (for the deployment of databases in parallel), which is used natively by Moodle.

Another feature of our solution it that it can be implemented in the cloud – this opens new perspectives to institutions of any size interested in Distance Education but not willing to deal with physical hardware issues. In what follows, we describe two experiments, both based on the same integration scheme of Moodle and Postgres-XC, and both conduct at UFRPE: first, the construction of a local cluster of cheap computers; next, the deployment of our solution in a private cloud formed by leased hardware and software. Tests of real-word scenarios have been conduct with the help of the JMeter tool.

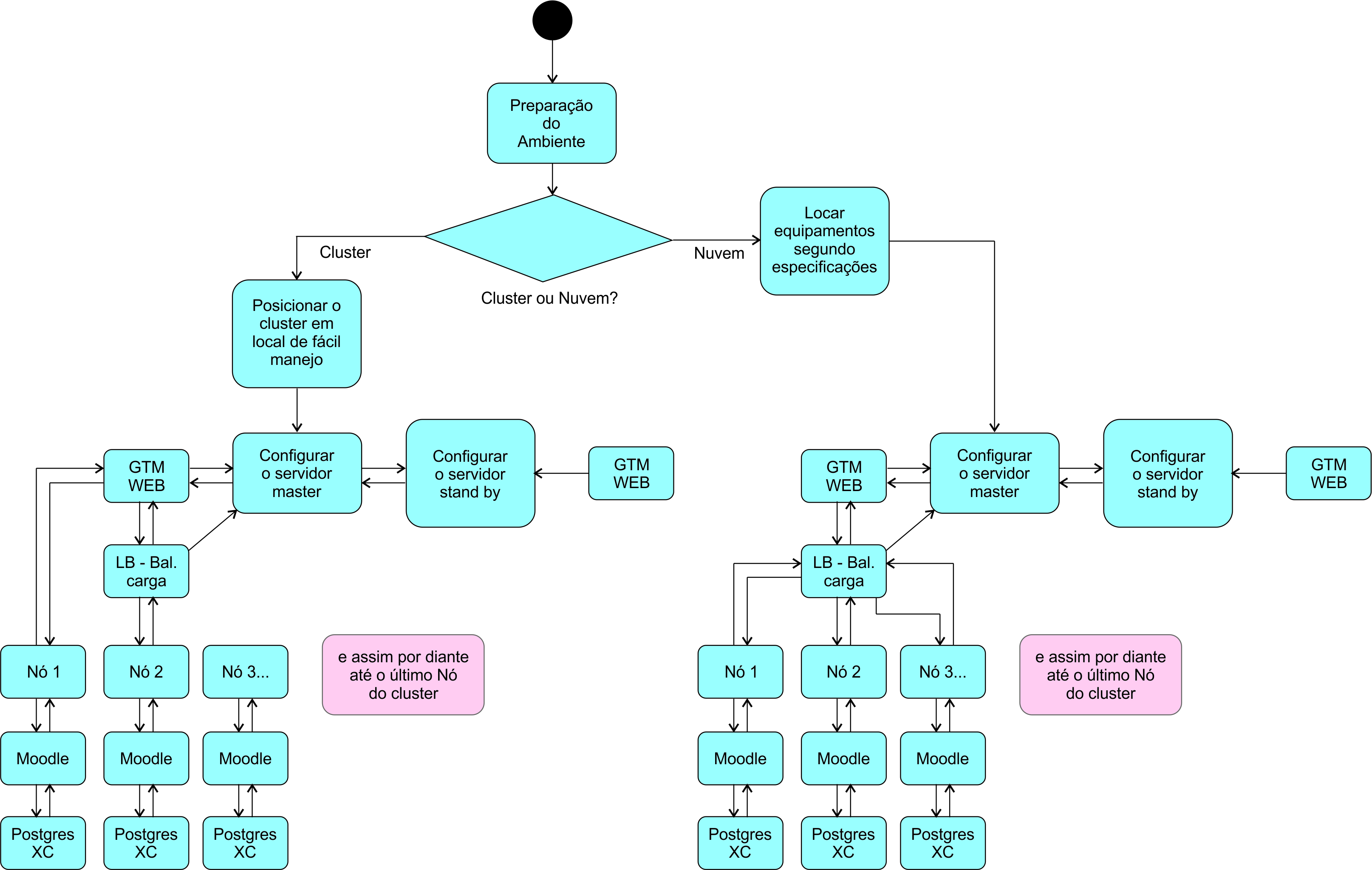
**Methodology**  
  
**Preparation of the physical environment network and cloud**

Figure 3 Activity Diagram

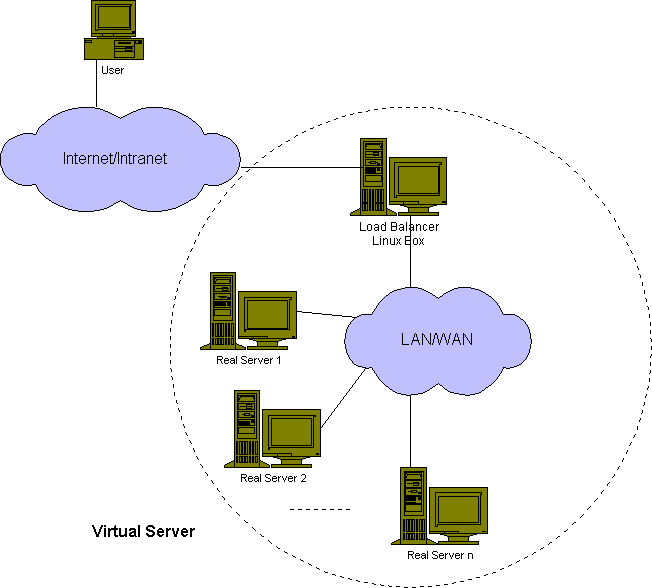
According to Figure 3 above in the preparation of the environment it is necessary to take a decision, which may be the use of an existing structure or to opt for a facility in the clouds, in our case we chose to test two options in order to demonstrate that this integration between Moodle and Postgres - XC would run on both platforms. For this we use in both environments the following hardware configurations on each node.

* Computer with 2GB Memory
* 500 GB Disk
* Processor Dual Core 2.6 Ghz.

We will use these computers servers with different types of services for the composition of our cluster and in our cloud also, this division occurs through software installed on the servers themselves provided by the Linux platform.

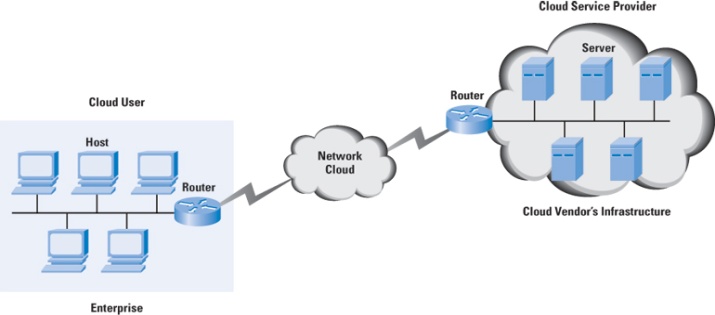
**Cluster instalation**

Linux centos 6.3 operating system was chosen and installed on all servers that make up the solution. On each computer, the network interface called Ethernet, which in Linux is mapped to the eth0 device should be connected to the same network that comprise the cluster, in this case, the ethernet switch cluster to do an internal network between computers. This network is considered private, ie all traffic on this network is physically separated from the external public network ( eg the Internet ). Using internal address, which we call the range of IP intranet, all IP 's addresses starting with 192, 172 and 10 are called private addresses.  
In the main computer cluster, put two ethernet interfaces that are mandatory. The interface eth0 is connected to the Ethernet network to the compute nodes. The eth1 interface was connected to the external Internet network.  
After physically set up our cluster, each node was configured to boot without a keyboard when connected. It is important that all services are configured for the cluster when the system restarts automatically they are working. We can get an idea of the physical topology as in Figure 4. This figure we can identify the central node where we put two network cards eth0 and eth1, and eth1 connected to the public network connected to eth0 and the internal network of the cluster. Just below we see the cluster nodes configured in the internal network.  
Figure 4 Topology Cluster Computers  
Source : postgresxc.org

  
**Deployment in the cloud**

We rented 4 machines in a provider of hosting services in http://server4you.com server4you called for the installation of the prototype, we follow the same line of the cluster setup, with same features, same operating system, amount of cpu and memory. The main difference was that the clouds have valid ips of internet1 not need to configure the internal network. So it is a simple and quick installation. We can get an idea of the physical topology in the clouds as in Figure 5. Identify this figure we can observe in the central node where the connection on eth0 connected directly to the internet. Just below we see the configured cluster nodes all on the Internet.  
Figure 5 Cluster Topology in the clouds

Source : postgres-XC



# Installation of database Postgres – XC

Postgres -XC is a set of data components that can be tightly coupled when installed in more than one hardware or virtual machines.

These features, described previously married strongly with our architecture as the postgres - XC is scalable and can be configured with as many database that can be used winning performance and can serve more concurrent users because servers decentralization bank data compared to a single database server, which has limited its memory and power procesador therefore given a limited amount of concurrent users.

Postgres - XC to run on multiple servers, your data is stored in a distributed manner, ie, distributed or replicated. In sql queries, Postgres - XC determines where the data is stored. The challenge was to install Moodle using this new tool because this Moodle approved for PostgreSQL and not for Postgres - XC is our case, especially This implementation required an extensive and detailed survey to staff software development postgres - XC, proven through the email listed in the appendix of this thesis where we can see the degree of difficulty and commitment so that we could reach the installation and adjustment of Moodle.

Moodle works recording all movements of the student on the platform,for example, if the user received a message in the forum and he just read, Moodle logs that drive the student in the database as well as he posted a new message on the forum, Moodle also adds this information to the database, whose logs will be analyzed by teachers, tutors and managers who are part of the virtual learning environment, VLE.

Postgres-XC consists of several components having specific functions to clear the cluster operate either in parallel, and the components can be installed in various topologies. The main components are : GTM - Global Transaction Manager, which is the overall management of transactions, GTM\_PROXY, is a performance enhancer for the GTM, the COORDINATOR is the module that communicates with the application which in our case with Moodle, and DataNode that is indeed who stores the data.

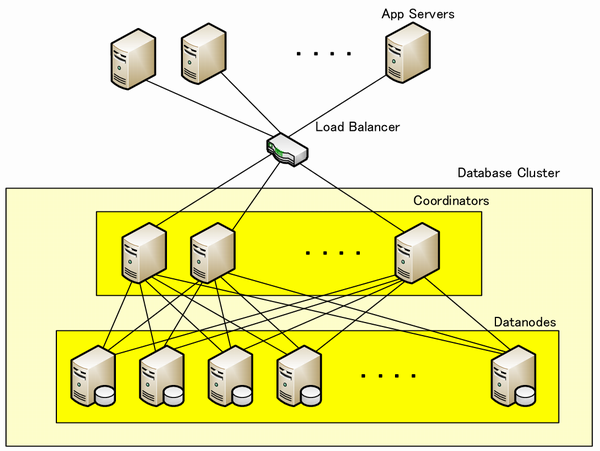
Our topology of Postgres - XC was installed with the GTM on the computer that we called node0, as the master computer at each node installed. The Coordinatos, on node1, node2, node3, node4, Node5, Node6, node7, node8, node9, and the GTM\_PROXY was also installed on each node as well as the datanodes.

We can observe the model as Figure 6 below, we have it on top of Moodle applications exchanging messages with a coordinator who plays the role of the database interface to the application. Now exchange information with the coordinator GTM - Global Transaction Manager that is the transaction manager, for every sql command are passed directed to the coordinator requesting a transaction to GTM after receiving the transaction ID, the coordinator executes the command in the datanode that is available in the cluster and returns the result to the application of the command execution.

In Figure 6 we can also observe that there is only one coordinator, although in our prototype we chose to use for each node in the cluster database a coordinator. This topology Postgres - XC is quite flexible. It is worth noted that the Global Catalog database is directly connected to the coordinators, for how the database is Postgres -XC transactions READ / WRITE, when a transaction takes effect in the catalog this communication is transmitted to the coordinators.

Figure 6 Diagram of the proposed Cluster

Source: postgresxc.org



# Tests

After approximately 20,000 test samples obtained an average of 4.500 milliseconds response with loss of 1.47% and a flow rate of 3.1 / s. This test consists of simulating a session by passing the user login and password. At this point Moodle stores the information of user access, and then the simulator executes the output of the system. As the graphs below, consisting of the response time graph and summary of connections. In Figure 7 the response time graph called the average attention in seconds, since the summary, in Figure below, we draw attention to the field of errors and to the field of flow.

The first 2 graphs below is about a network formed by 10 small computers the second test was made rented 4 servers in a cloud refer to the last two graphs.

Figure 7 *Jmeter response graph of the prototype MOODLE and POSTGRES-XC*

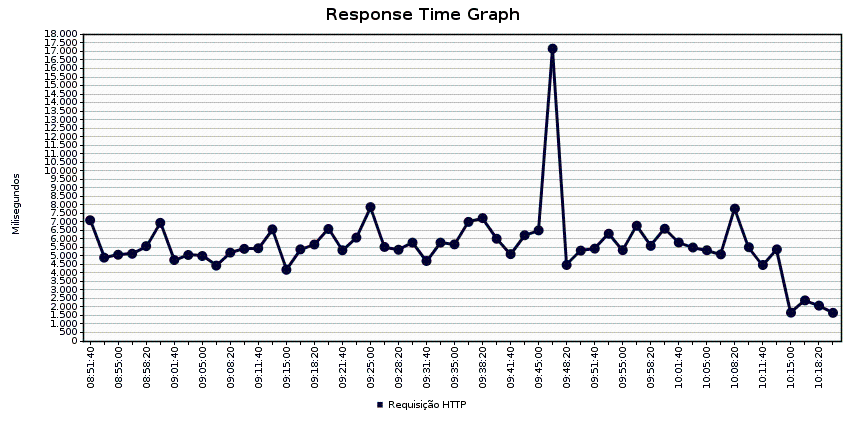


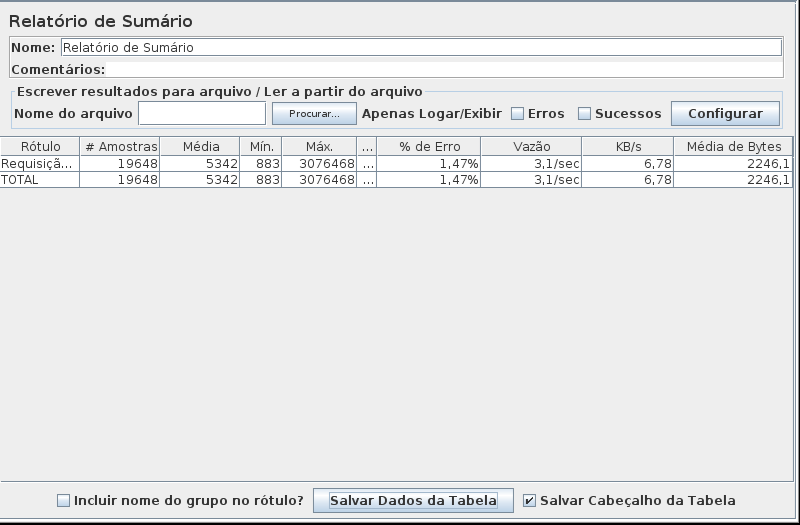
Figure 8 Summary of Jmeter prototype WITH MOODLE E POSTGRES-XC  


Figure 9 *Jmeter response graph of the prototype in the cloud - MOODLE and POSTGRES-XC*

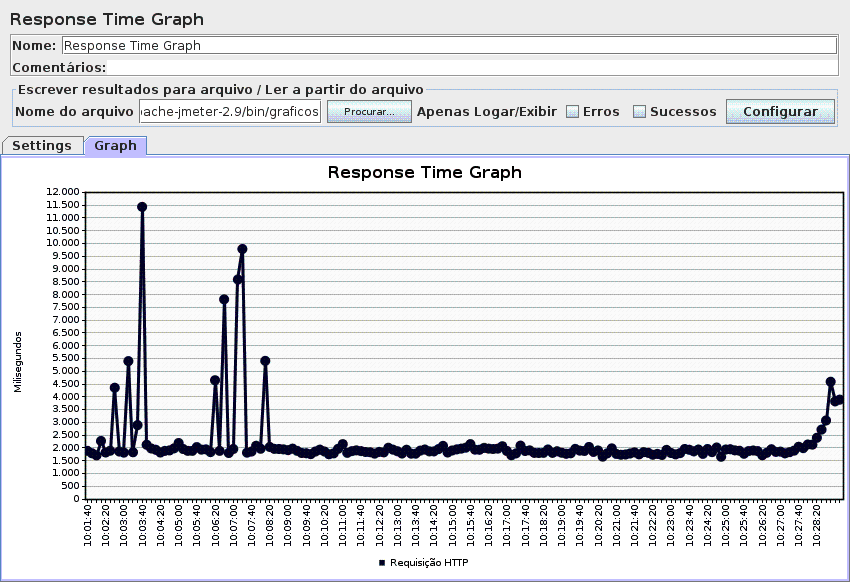
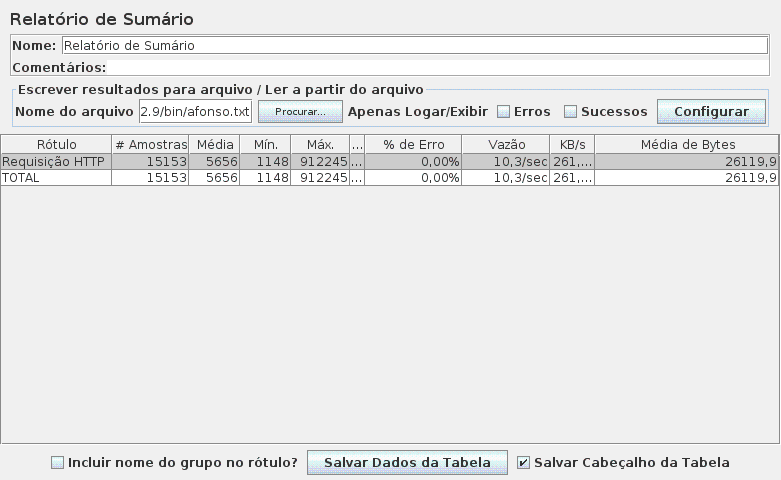


Figure 10 Summary of Jmeter prototypein the Cloud - MOODLE E POSTGRES-XC



**Conclusion**

With this work we conclude that indeed it is possible to build an infrastructure to cater for a range of services that require a complex network of computers, as the exhibition contained in Chapter 2 of the motivation for developing this infrastructure through existing computers in the context of the institution, that supported an economy of scale, which was reliable and always were in operation and were mainly low cost. Apparently a complex task, but with the advent of free programs, it means, without initial cost of purchasing the license, this mission was made possible because, as demonstrated in Chapter 3, where we detail the operation and implementation of the cluster successfully used in the tests this prototype.

In stress tests of the prototype simulations typing with Jmeter tool were made ​​to bring this information to be accessed by the program almost simultaneously, as if they were real students, requiring an infrastructure of high performance, reliable, scalable and have the own satisfaction rate about the response time of the Virtual Learning Environment. All these parameters above have been achieved from the implementation of the tests covering by The Waiariki Institute of Technology, in a cluster of high performance and low cost (Benner, 2010) which we replicated in our prototype and has been successful with the result of this implementation.  
 It is concluded that the approach described in this paper provides a feasible solution to provide a system with improved high availability, highly reliable system, with the potential to provide a high performance solution compared to similar architectures using the same means of communication as shown in the chapter of this dissertation review.

We also conclude that this infrastructure can be implemented on low-cost equipment, or existing equipment with one or two years of use, with new or old models from previous run, which cost cheaper acquisition processors, reusing these items for the construction of this infrastructure. Therefore, our hypothesis proved the potential of Linux Cluster to provide a platform for VLE using Moodle, able to be scalable, available and inexpensive. “Hardware solutions usually have the highest performance and security by copying or replicating whole disks. Using a hardware solution reduces the granularity and ability to determine when and what must be replicated. These solutions are also usually much more expensive than software solutions” (Whein-Jen, 2007).

It is noteworthy that as seen, are not necessary computing resources for complex and do not require large investments, to mount a virtual learning environment, and can even reuse computers deemed obsolete in the formation of the server pool. If you choose yourself by a cluster topology or you can rent a set of virtual servers in the cloud and provide the same service.

Our next step is to get together a free community that produces Moodle put this solution into the official distribution of Moodle, thereby making this platform available to the entire academic community. Introducing this platform officially showing that it is possible to build a further economic environment, which would run on a platform that Beowulf Cluster is to focus on the performance of central processing unit, in particular the server / database, as explained in the dissertation, running in parallel and using memory and cpu, and also the pool of servers and may make recommendations and suggestions for future work. One of our intentions in the short term, is doing the same integration with Amadeus Software AVA with Postgres - XC in cloud computing.

**References**

Benner, Gary. et al. (2008). Developing a high performance Linux Cluster at low cost. Rotorua, NZ: School of Business and Computing, Waiariki Instituteof Technology.

Dougiamas, M. & Taylor, P. (2003). Moodle: Using Learning Communities to Create an Open Source Course Management System. In D. Lassner & C. McNaught (Eds.), Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2003 (pp. 171-178). Chesapeake, VA: AACE.

Nicola, Matthias; JARKE, Matthias. (2000). Performance Modeling of Distributed and Replicated Databases.

Whei-Jen, Chen., Chandrasekaran, Chandramouli., Gneiting, Don., Castro, Gustavo., Descovich, Paul., Iwahashi, Tomohiro. (2007). High Availability and Scalability Guide for DB2 on Linux, UNIX, and Windows. RedBooks. First Edition.

1. This research is part of the project APQ-0833-1.03/12 *Modelos Formais de Computação: Alguma Teoria e seu Ensino* supported by FACEPE (Brazilian government). [↑](#footnote-ref-1)