2022 31st Annual Conference of the European Association for Education in Electrical and Information Engineering (EAEEIE) | 978-1-6654-8442-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/EAEEIE54893.2022.9820369

Proposal of a technological cluster to support eLearning platform

Filipe Cardoso
Viseu Polytechnic - ESTGV
and *INESC Coimbra*
Viseu, Portugal
ORCID: 0000-0002-3916-5182

António Godinho
Coimbra Polytechnic - ISEC
Coimbra, Portugal
ORCID: 0000-0002-1146-464X

J. Rosado
Coimbra Polytechnic - ISEC
and *INESC Coimbra*
Coimbra, Portugal
ORCID: 0000-0001-5610-7147

Filipe Caldeira
Viseu Polytechnic - ESTGV
and CISeD - Research
Centre in Digital Services
Viseu, Portugal
ORCID: 0000-0001-7558-2330

Filipe Sá
Coimbra Polytechnic - ISEC
and *CISeD - Research*
Centre in Digital Services
Coimbra, Portugal
ORCID: 0000-0002-7846-8397

***Abstract*—Due to the SARS-COV-2 pandemic, educational institutions are immediately faced with a new challenge to adapt, forcing the transition from face-to-face teaching to distance learning in a short period. Distance education supported by technology is a challenge for educational institutions based on binomial technology/teaching. This paper presents a proposal for an e-learning technology structure, supported by a cluster of servers capable of responding to the requirements of distance learning based on the premises of High Availability, High Performance, Load Balancing. The beginning of this study consisted of a literature review to find the various existing technologies, a way to combine them and create a system capable of providing the necessary functionalities, and whose performance could host all the users of an institution simultaneously. The implemented system results from this combination of technologies and allows its capacity to be scaled at any moment according to momentary needs. In technological terms, the solution was based on a free Linux distribution, the Ubuntu Server installed inside a cluster of servers with VMware ESXi, and a cluster of database nodes based on Gallera technology. The eLearning platform used in this study was Moodle because it is one of the resources most used by institutions. The aspects of teaching, provision of content and execution of evaluation tests, were explored. With the implementation of the presented scenario, it was possible to guarantee the High Availability and load balancing of the platform and guarantee a high performance of the whole solution.**

***Keywords*—E-Learning, pandemic situation, Moodle, Cluster, High availability, High performance, Load Balancing**

I. Introduction

The pandemic originated by Covid-19 disease in 2019, affected the entire world population in all its aspects. As far as educational institutions are concerned, they had to adapt from face-to-face teaching to distance learning in the e-learning area. This transition had to take place with minimal impact on students and teachers in a short period. In the context of education, authors [1] define e-learning as the use of information and communication technologies in teaching and learning. In the same vein, [2] states that e-learning offers numerous advantages to organizations, universities, and users. Its market is continuously growing, and there are considerable initiatives to promote it. [3] write that perceived satisfaction can be affected by interactive learning environments, perceived self-efficacy, and perceived anxiety. [4] claims that raise achievement levels when learning computer programming. The same authors write that “students perceived self-efficacy, perceived anxiety, perceived usefulness, and memory strategies should be taken into account while designing learning environment”. It is almost impossible to implement teaching processes without using information and communication technologies, especially in higher education [5]. This article describes a low-cost solution of a project to implement an e-learning platform in an educational institution carried out by an IT team to respond to the urgent adaptation of the face-to-face online regime, derived from the confinements caused by the COVID2019 pandemic. Existing systems were not prepared to support an entire institution online since IT teams had to find solutions that could meet their needs. To this end, it was necessary to look within the various existing technologies to combine them and create a system capable of providing the necessary features and whose performance could accommodate all users of an educational institution at the same time. The implemented system results from this combination of technologies and scales its capacity according to momentary needs. The article is organized as follows—chapter one introduction to the project. In chapter two, the scenarios are presented. Chapter three presents the implementation of the cluster: The results can be found in the second last chapter. Finally, the article ends with the conclusions and future work.

978-1-6654-8442-8/22/$31.00 ©2022 IEEE

II. Scenarios

In this chapter, the implementation scenarios are presented. As an e-learning tool, the work team selected Moodle since it was the platform that was already in use in the institution. Moodle is one of the most widely used platforms in the world [6]. In the context of Evaluation of Learning Management Systems, [7] reports that Moodle was chosen eleven times, in first place ratings and only one third-place rating. According [8] is a free learning management system that enables you to create powerful, flexible, and engaging online learning experiences. The same authors deliberately use the phrase ”online learning experiences” instead of ”online courses.”

A. Standalone Scenario

In a standalone scenario (Fig. 1), Moodle is usually installed on a single server, which will process all web requests and database transactions. In such a scenario, course data is stored locally and the entire system. In terms of processing power, this only depends on the characteristics of the server itself, i.e., it depends on the existing hardware. Regarding service availability, in a standalone scenario, if the server is inaccessible for any reason (e.g., power failure, network failure, hardware failure, or operating system problems), the entire service is affected, leaving users without access to it. If it is necessary to reinstall the platform, downtime is high because it is necessary to reinstall and reconfigure a server with the same parameters as the initial server. The term scalability in this scenario does not apply because the only way to grow is to add hardware with better features.

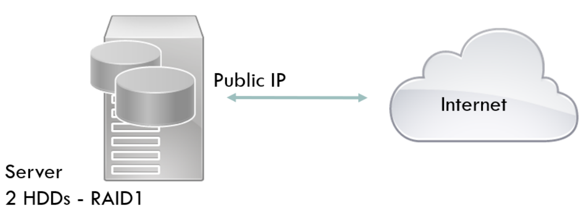

Fig. 1: Single Server Moodle Platform.

B. Cluster Scenario

Data clustering is a well-known technique in various areas of computer science and related domains. [9] [10], mention that data clustering is a process of partitioning or outliers detection to find a pattern, points, or objects. Objects within a valid cluster are more similar than the objects outside the cluster [9]. Clustering is the unsupervised grouping of objects into classes of similar objects. In e-learning, clustering can be used for finding clusters of students with similar behavior patterns. The challenge that arises is that it is necessary to provide a solution that can meet the following requirements:

- High Availability - HA;
- Load Balancing;
- High Performance;
- Scalable;
- Hardware and Software Independent;
- Easy to Migrate;
- Based on Open Source Software.

After some research, it is concluded that no ”turn-key” tool offers all the requirements. After this research, it was concluded that it is possible to create a solution capable of meeting the proposed requirements through the combination of several open-source tools. The proposed scenario is a cluster that uses the following tools:

- HAProxy – with loadbalancer; HAProxy is a free, very fast and reliable solution offering high availability, failover and load balancing mechanism, and proxying for TCP and HTTP based applications [11] and was used as techniques of web server load balancing;
- VirtualIP KeepAlived – used for virtual routing protocol;
- GalleraDB – as database cluster;
- NFS – for file repository.

Fig. 2 presents a cluster scenario that allows the requested requirements to be meet.

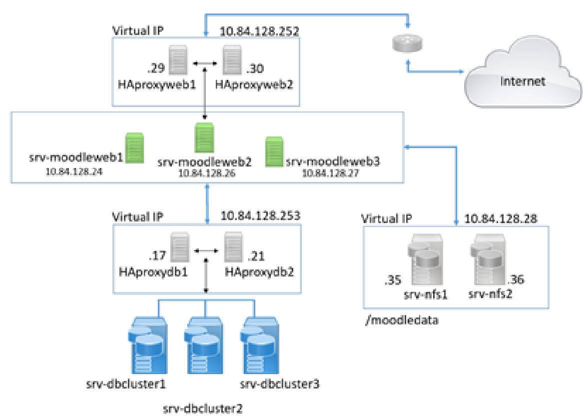

Fig. 2: Moodle Cluster.

III. Cluster Implementation

In order to guarantee one of the initial requirements (the use of opensource tools), a free Linux distribution, Ubuntu Server, was chosen. Each proposed server was installed in a virtual machine inside a server cluster with VMware ESXi 7. VMware ESXi is a virtualization solution that manages the distribution of hardware resources to each virtualized OS, which creates an isolated virtual environment for each one, is called the Virtual Machine Monitor [12]. VMware ESXi is a hypervisor that is built directly on top of x86 hardware. It abstracts the underlying hardware and allows multiple virtual machines to use the same hardware resources [13]. VMware ESXi leverages hardware support for MMU virtualization available in modern Intel/AMD CPUs [14]. The network provided by the virtualizers is based on 10GBASE-LX4 technology for both the Ethernet data network and the iSCSI network. The volumes for the installation of the various servers were made available through iSCSI and stored in a SAN array. Using these technologies alone guarantees redundancy because if one of the virtualizers fails, the VMs are automatically migrated to an available node. In turn, the data is stored in an array of storage, ensuring availability and integrity in case a controller or a unit (SAN) fails. Another advantage of the array is that the most accessible data is stored on high-speed disks.

A. *Data Base Cluster - Implementation*

The implementation of the database nodes (Fig. 3) uses the Gallera Cluster, HAPROXY, and KeepAlived. This scenario ensures that data is available on any of the nodes in the cluster (Gallera Cluster). If one of the nodes fails, the data is available on the remaining nodes. If we want to add another node to the cluster, the data is automatically synchronized between all cluster members. In order to ensure that the connection to the database servers is dynamic and transparent, KeepAlived was used. It provides an IP address that receives all requests and internally distributes them to the various nodes in the cluster. In order to introduce load balancing, a load balancer (the HAPROXY) was installed and configured. Note that this implementation is not restricted to the proposed scenario.

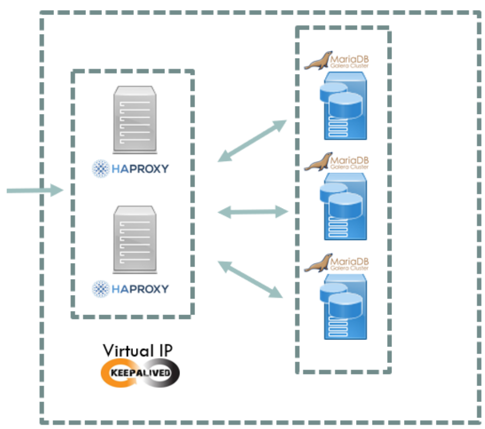

Fig. 3: Database Cluster.

In case of failure (Fig. 4) of one or two nodes of the DB cluster, the system would continue to function correctly. The same is true for the load balancer cluster because it will work correctly in a scenario where one of the nodes is down or inaccessible. After all, they both respond to a single IP address.

Adding nodes to the cluster can be straightforward if more resources are needed. Note that the nodes do not necessarily have to be in the same physical space. A hybrid scenario is possible, i.e., local nodes and nodes in the cloud (Fig. 5).

B. *Web Cluster - Implementation*

To implement the web cluster, three VMs were installed on Linux - Ubuntu Server, on which were installed:

- NGINX - Webserver;
- Mysql Client – Database Client;

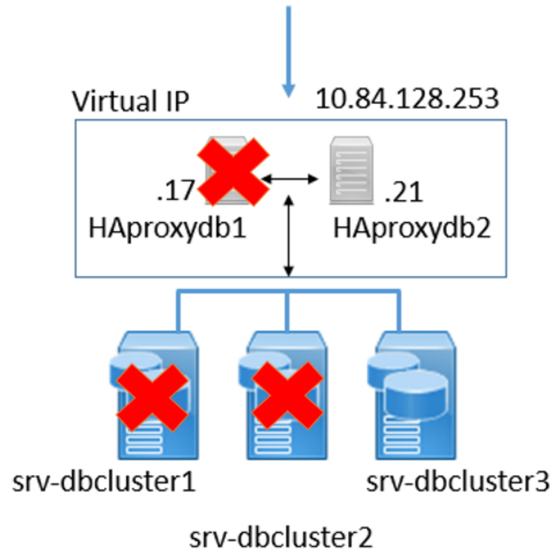

Fig. 4: Database Cluster Worst Failure Scenario.

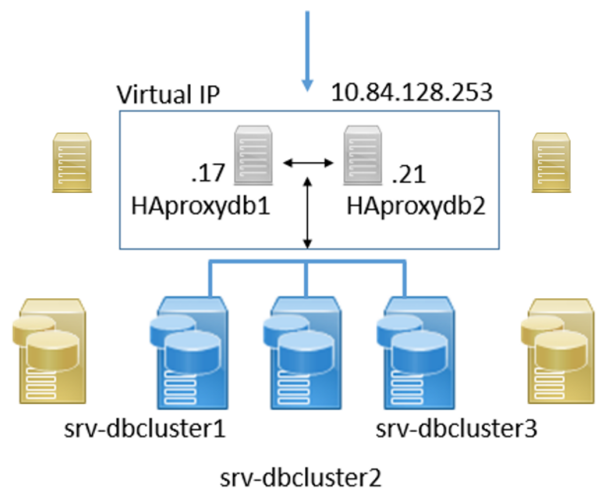

Fig. 5: Database Cluster Growth.

- NFS Client – File cluster Client.

After installing the requirements for the correct operation of Moodle (Fig. 6), the (remote) database was created on one of the nodes, and the latest stable version of the e-learning platform was downloaded.

The installation cannot be started at this stage because the data present in /moodledata (the directory in which Moodle stores the data) must also meet the requirements (high availability). This way, two nodes were installed (Fig. 7) that would host the data present in /moodledata.

For this cluster, we again used the virtual routing tool, KeepAlived, because it allows us to connect to any of the nodes through a single IP address. After creating and configuring the nodes of the NFS cluster, the installation was resumed.

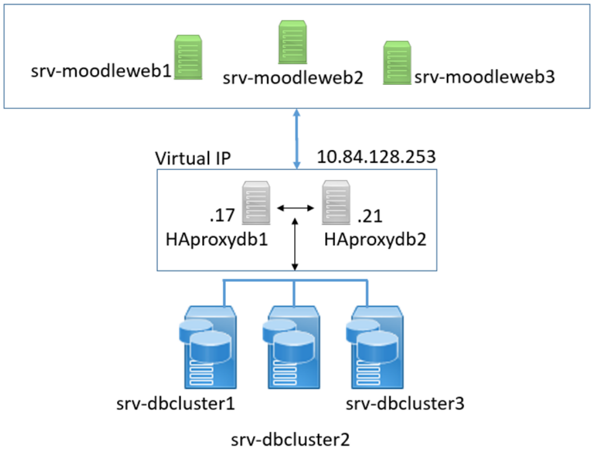

Fig. 6: Web Cluster Scenario.

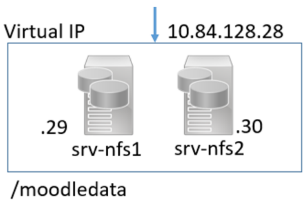

Fig. 7: NFS Cluster.

This installation is done in only one of the cluster nodes, which is later replicated to the other nodes through rsync. The replication process between all nodes should be repeated whenever a plugin is installed. It is possible to guarantee that the plugin is available on all nodes in this way. Fig. 8 shows the worst-case failure scenario with the presented solution.

C. Web cluster loab balancer - Implementation

The webserver cluster must respond only by a single IP address or name (DNS). So that users can access the service without having to memorize the IP addresses of each of the nodes (Fig. 9), it was necessary to configure the Virtual IP service (keepAlived). One of the initial requirements was that the system would allow several hundred simultaneous users. Thus, it was necessary to implement the load balancer service at this level.

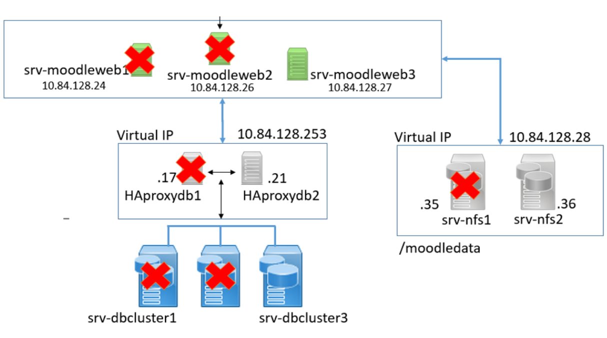

Fig. 8: Second Worst Failure Scenario.

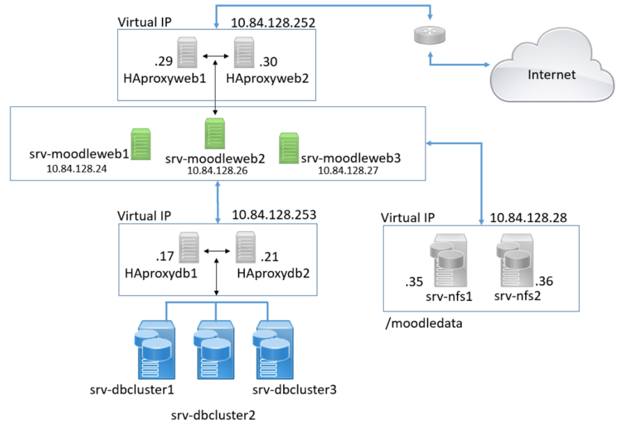

Fig. 9: Moodle Cluster Full Implementation.

Again, HAPROXY was used for load balancer and KeepAlived for routing and virtual IP address. This scenario allows a high number of failures (Fig. 10), because all services are redundant.

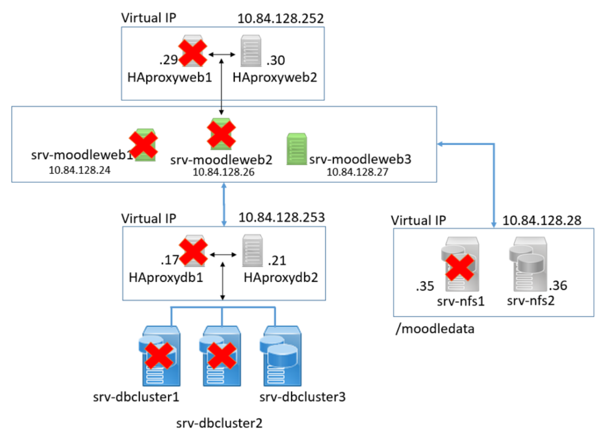

Fig. 10: Moodle Cluster Worst Case Failure.

As Moodle is an e-learning platform, one of the most used features is the online evaluation system. The need arose to ensure that a given user always connects to the same webserver while their session is active. To solve this situation (Fig. 11), it was enough for the HAproxyweb nodes to create a cookie identifying the web server the user is connected to. This way, whenever a request is made, it (always) automatically forwards to the same Moodle web node.

IV. Experimental Results

Since this is a practical and authentic project, conditioned by the Covid-19 pandemic, which forced the system to be put into production overnight to support online classes, the implementation methodology consisted of a development-reaction

```
backend moodleweb-cluster
    mode http
    balance roundrobin
    option httpchk
    option forwardfor
    http-request set-header X-Forwarded-Port %[dst_port]
    http-request add-header X-Forwarded-Proto https if { ssl_fc }
    option httpchk HEAD / HTTP/1.1rnHost:localhost
    cookie SERVERID insert indirect nocache
    server web1  10.84.128.24:80 cookie moodle1 weight 1
    server web2  10.84.128.26:80 cookie moodle2 weight 1
    server web3  10.84.128.27:80 cookie moodle3 weight 1
```

Fig. 11: HAPROXY cookie.

implementation. The system was made available online in March 2020. The GoAcess-webstat tool was used to monitor access, allowing viewing the platform usage in real-time.

Before the pandemic, the previous Moodle implementation had between 30k to 130k hits per day. After the cluster implementation, and Covid-19 restrictions, with online courses, most of the time, hits per day were above 220k, with a maximum of hit per day over 1 million (Fig. 12).

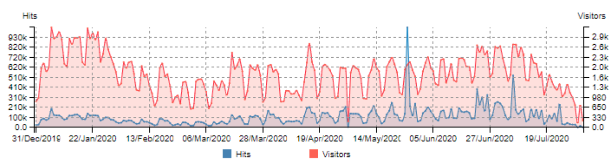

Fig. 12: Unique visitors per day.

V. Conclusions

Implementing a low-cost platform to support distance learning in a short time has been achieved. With the implementation of the scenario presented, it was possible to guarantee the platform's high availability and load balancing through the various load balancers that exist throughout the scenario. Due to this fact and because the proposed system is composed of several nodes with the same service, it is possible to guarantee a high performance of the entire solution. Since all VMs were installed on a virtualizer, they can easily be migrated to another system. Scalability is also guaranteed because, in case of need, we can add nodes to the cluster according to the level of requirements needed in each of the available services (Example: need to meet more web requests - add more moodle web nodes). If a high number of accesses is not required, the cluster allows part of the nodes to be shut down, maintaining its correct operation. This way, it was possible for the educational institution to provide a 100% functional solution to support online learning in a short period, using low-cost tools.

VI. Future Work

Since the implementation had unfeasible time requirements due to Covid19, it is now our intention to analyze the tools used based on scientific knowledge and review its entire implementation. In parallel, through the Delphi method, we intend to consult a set of specialists to assess what requirements and needs they identify for implementing this solution. We will also perform usability, quality, load, and performance tests to scale the system.

Acknowledgements

This work is partially funded by National Funds through the FCT - Foundation for Science and Technology, I.P., within the scope of the projects UIDB/00308/2020, UIDB/05583/2020 and MANaGER (POCI-01-0145-FEDER-028040). Furthermore, we would like to thank the Research Centre in Digital Services (CISeD) and the Polytechnics of Viseu and Coimbra for their support.

References

[1] L. Pham, Y. B. Limbu, T. K. Bui, H. T. Nguyen, and H. T. Pham, "Does e-learning service quality influence e-learning student satisfaction and loyalty? evidence from vietnam," *International Journal of Educational Technology in Higher Education*, vol. 16, no. 1, pp. 1–26, 2019.

[2] E. M. Caro, "E-learning: un análisis desde el punto de vista del alumno," *RIED. Revista iberoamericana de educación a distancia*, vol. 11, no. 2, pp. 151–168, 2008.

[3] S.-S. Liaw and H.-M. Huang, "Perceived satisfaction, perceived usefulness and interactive learning environments as predictors to self-regulation in e-learning environments," *Computers & Education*, vol. 60, no. 1, pp. 14–24, 2013.

[4] R. Phoophuangpairoj, "Identifying efficient determinant factors affecting students' achievement in learning computer programming," 2011.

[5] N. Kadoić and D. Oreški, "Analysis of student behavior and success based on logs in moodle," in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 2018, pp. 0654–0659.

[6] J. Hollowell, *Moodle as a curriculum and information management system*. Packt Publishing Ltd, 2011.

[7] B. Stewart, D. Briton, M. Gismondi, B. Heller, D. Kennepohl, R. McGreal, and C. Nelson, "Choosing moodle: An evaluation of learning management systems at athabasca university," *International Journal of Distance Education Technologies (IJDET)*, vol. 5, no. 3, pp. 1–7, 2007.

[8] W. H. Rice, "A complete guide to successful learning using moodle," 2006.

[9] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.

[10] A. Nagpal, A. Jatain, and D. Gaur, "Review based on data clustering algorithms," in *2013 IEEE conference on information & communication technologies*. IEEE, 2013, pp. 298–303.

[11] V. Kaushal and A. Bala, "Autonomic fault tolerance using haproxy in cloud environment," *Int. J. of Advanced Engineering Sciences and Technologies*, vol. 7, no. 2, pp. 54–59, 2011.

[12] D. Grosmann, N. F. S. de Sousa, F. A. Diniz, and T. R. da Silva, "Estudo comparativo sobre o uso do vmware e xen server na virtualização de servidores," *V Encontro Regional de Computação Ceará, Maranhão e Piauí - ERCEMAPI 2011. - Researchgate*.

[13] M. R. Kajamoideen and A. Sivaraman, *VMware ESXi Cookbook*. Packt Publishing, 2014.

[14] F. Guo, S. Kim, Y. Baskakov, and I. Banerjee, "Proactively breaking large pages to improve memory overcommitment performance in vmware esxi," in *Proceedings of the 11th ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, 2015, pp. 39–51.