



HAL
open science

MOHD : Moodle Optimisé Hautement Disponible

Christophe Gaboret

► **To cite this version:**

Christophe Gaboret. MOHD : Moodle Optimisé Hautement Disponible. JRES (Journées réseaux de l'enseignement et de la recherche) 2021, May 2022, Marseille, France. 2022. hal-04807453

HAL Id: hal-04807453

<https://hal.science/hal-04807453v1>

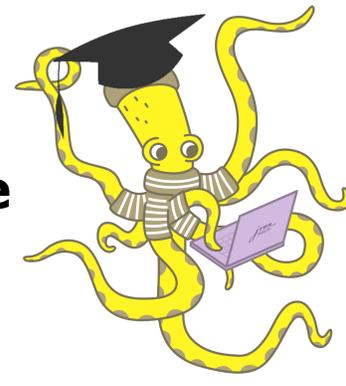
Submitted on 27 Nov 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License



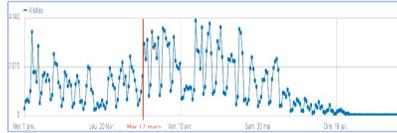
MOHD : Moodle Optimisé Hautement Disponible

jres MARSEILLE 2022

C. Gaboret

Depuis 2008, les écoles du campus d'Evry de l'IMT utilisent **Moodle** comme plateforme pédagogique. Suite au premier **confinement**, le 17 mars 2020, d'un simple support de cours, notre *Learning Management System* (LMS) devient alors le pivot de la continuité pédagogique. On note alors un accroissement des connexions à notre plateforme qui devait s'accroître à la rentrée 2020.

Nous présentons ici notre travail afin d'avoir des échanges sur les bonnes pratiques, d'améliorer ce que nous avons fait mais aussi d'en apprendre un peu plus sur ce que chacun a mis en place durant cette période compliquée.



Rentrée 2019

- 842 cours actifs,
- 13,5 Go de données,
- 1600 visites par jour.

Rentrée 2020

- 1609 cours actifs,
- 463 Go de données,
- 7000 connexions quotidiennes.

La DSI, qui hébergeait jusqu'alors Moodle sur un serveur unique, se devait de répondre aux besoins des enseignants et des étudiants pour la rentrée suivante. Nous avons donc décidé de :

1 Améliorer la résilience de la plateforme

2 Augmenter les performances de notre Moodle

3 Superviser toute l'infrastructure

4 Gérer le cycle de vie de notre LMS

1 Amélioration de la résilience de la plateforme

Contrairement à ce qu'on pourrait penser, différents retours d'expérience montrent qu'il faut surtout renforcer le serveur web [1]. Pour cela, nous avons :

- mis en place d'un serveur frontal HA-Proxy,
- implémenté plusieurs *backends* Apache.

Le frontal web est basé sur HA-Proxy qui est stable et performant. Il renvoie les requêtes aux différents backends, dont une partie peut être arrêtée sans générer d'interruption de la plateforme.

La base de donnée (DB) est également un élément déterminant d'une plateforme Moodle. Une solution pour assurer le *failover* et le *load-balancing* est de mettre en place un *cluster* Galera. Ce type d'architecture est éprouvé et maîtrisé et peut fonctionner sur un système de fichiers distribués GlusterFS. Cette solution est de loin la plus robuste mais elle est assez complexe à mettre en œuvre et potentiellement à exploiter [4][5]. Nous avons opté pour l'utilisation de **deux serveurs de bases de données répliqués en temps réel**, en suivant la méthode KISS et en respectant les recommandations suivantes :

- DB séparée dédoublée pour failover,
- stockage /var/lib/mysql sur des disques SSD,
- tuning DB pour utilisation de la mémoire qui doit être de 32G RAM au minimum,
- purge régulière de la table mdl_logstore_standard_log afin d'alléger la taille des binaires.

Finalement, le répertoire web de Moodle et le répertoire moodledata sont exportés en **NFSv3**, en suivant les préconisations de l'éditeur [3]. Idéalement, pour assurer une redondance, il conviendrait de mettre en place NFS sur DRDB (Distributed Replicated Block Device). Là encore, nous avons préféré nous conformer au principe KISS, nous laissant la possibilité d'améliorer notre plateforme si le besoin s'en fait ressentir dans l'avenir.

2 Augmenter les performances de notre Moodle

Le fait de distribuer l'architecture sur plusieurs serveurs dégrade les performances globales, notamment du fait des écritures sur disque. Il est tout de même possible d'en minimiser l'impact.

La première action entreprise a été de ne plus faire fonctionner les *backends* Apache en mode *Prefork*. Le module multi-traitements (MPM) *Prefork* serait seul compatible avec *mod_php* [2], ce qui en fait souvent le MPM par défaut de Moodle. Cependant *Prefork* n'est pas connu pour ses performances, nous avons donc utilisé le **MPM Event**.

Nous avons également implémenté le FastCGI Process Manager (**PHP-FPM**) sur ces *backends* Apache, comme préconisé dans la documentation officielle de Moodle. Le principe de PHP-FPM est de faire fonctionner PHP de manière séparée et d'installer le module FastCGI d'Apache afin de faire communiquer Apache et PHP. PHP-FPM fournit des fonctionnalités supplémentaires utiles pour les sites très fortement chargés.

Enfin, nous avons mis en **cache** les sessions sur un serveur **Redis**. Les sessions sont écrites en mémoire au lieu d'être sur le disque, ce qui améliore très nettement les performances.

3 Superviser toute l'infrastructure

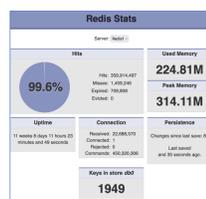
La surveillance d'une plateforme consiste en 3 points :

- la remontée d'alertes,
- les indicateurs,
- la supervision (monitoring).

La **remontée d'alertes** est encore souvent faite par nos utilisateurs mais nous travaillons à la mise en place de notifications par Chat avec un outil type **Mattermost**.

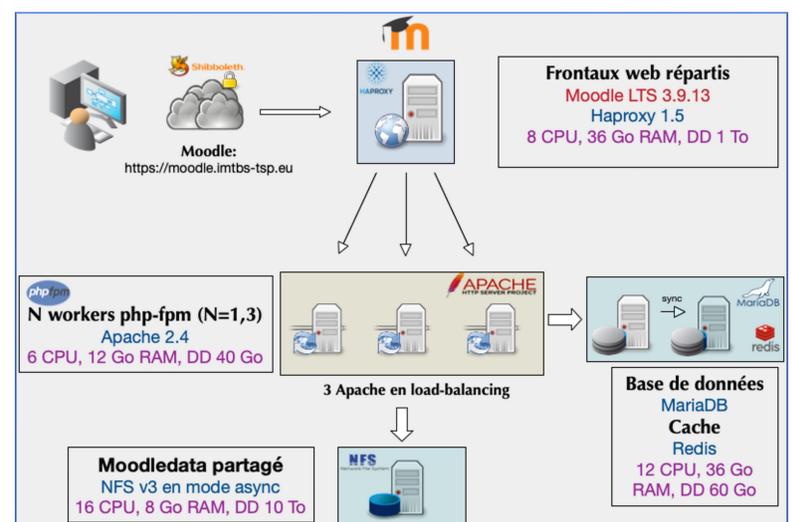
Les **indicateurs** sont de 2 sortes, ceux concernant les usages que nous collectons avec **Matomo** et ceux concernant la disponibilité de la plateforme sur une période longue, typiquement une année scolaire.

Avec l'infrastructure mise en place, la **supervision** ne se limite plus à des paramètres systèmes (charge, utilisation de la CPU, etc.) mais aussi à la réplication de la DB, l'état du cache Redis, etc. Nous collectons toutes ces données avec **Prometheus** et nous les mettons sous forme de graphes avec **Grafana**.



4 Gérer le cycle de vie de notre LMS

Plutôt qu'effectuer les mises à jour de Moodle et des plugins additionnels au fil de l'eau au fur et à mesure des alertes de CERT et des notifications de la plateforme, avec de fastidieux retours en arrière, nous avons mis en place un système d'intégration continue basée sur **Git**. Les mises à jour Moodle (mises à jour mineures et majeures) et de ses modules sont maintenant gérées de façon granulaire, depuis un dépôt git, hébergé sur le serveur **GitLab** de la DSI. Nous avons intégré ceci à un **runner** qui, au travers de « *pipelines* » nous permet de tester toutes les étapes avant la mise à jour effective de notre LMS .



Architecture du Moodle Optimisé Hautement Disponible mise en place dans notre établissement. Les versions logicielles, le nombre de CPU, de mémoire et la taille des disques durs sont également indiqués.

Références

1. Seclier B. **Moodle à l'université de Lorraine**. Dans Journée Technique Moodle, CSIESR/CNAM, février 2020 ; <https://mediaserver.cnam.fr/videos/08-univ-lorraine-jt-csiesr-2020-moodle>.
2. Germain A., Lesage J.B., Lefebvre O. et Bonamy V. **Optimisation Moodle à l'Université de Rouen Normandie en 2020 Version2**, mai 2020 ; <https://www.esup-portail.org/wiki/download/attachments/974651408/univ-rouen-moodle-optimisation.pdf>.
3. Red Hat Training. **Mounting NFS shares** ; https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/managing_file_systems/mounting-nfs-shares_managing-file-systems.
4. Ksiazek K. **Building a Highly Available Database for Moodle Using MariaDB (Replication & MariaDB Cluster)**, septembre 2020 ; <https://severalnines.com/database-blog/building-highly-available-database-moodle-using-mariadb-replication-mariadb-cluster>.
5. Stefas Y. **Mise en place de la haute disponibilité autour de MariaDB : retour d'expérience**, juin 2020 ; <https://www.syloe.com/haute-disponibilite-autour-de-mariadb/>.
6. Chrtophe. **Système de fichiers distribués**, novembre 2020 ; <https://chrtophe.developpez.com/tutoriels/filesystems-distribues>.



La méthode KISS