UDC 004.738.5:378.018.43

MOODLE MANUAL UPGRADE METHODOLOGY

Atanasov S. S.

Trakia University, Students campus, Stara Zagora, 6015, Bulgaria
ORCID: [1]https://orcid.org/0000-0002-2658-1611
E-mail: svetoslav.atanasov@trakia-uni.bg

DOI: https://doi.org/10.15673/atbp.v15i4.2723

Abstract. *This abstract outlines a systematic four-step methodology for upgrading Moodle from version 3.6.5 to 3.9.9. The process begins with comprehensive backup and preparation, followed by the replacement of the Moodle codebase and execution of the upgrade script. Post-upgrade checks are performed to ensure the smooth functioning of core and third-party components. Additionally, the abstract explores the actions and functions of the cron script, emphasizing its role in executing scheduled tasks critical for Moodle's operation.*

Furthermore, the document sheds light on common reasons behind slow Moodle operation and provides practical tips for enhancing system speed. Recommendations include optimizing server resources, database maintenance, strategic caching implementation, and the judicious use of content delivery networks (CDNs). The importance of upgrading PHP and database versions is highlighted, alongside the benefits of disabling unused features and employing server monitoring tools. The abstract emphasizes the significance of staying abreast of Moodle updates for continued performance improvements and bug fixes, offering a holistic guide for upgrading and optimizing Moodle systems.

Анотація. *У цій анотації викладено систематичну чотириетапну методологію для оновлення Moodle з версії 3.6.5 до 3.9.9. Процес починається з комплексного резервного копіювання та підготовки, після чого відбувається заміна кодової бази Moodle і виконання сценарію оновлення. Перевірки після оновлення виконуються, щоб забезпечити безперебійне функціонування основних і сторонніх компонентів. Крім того, анотація досліджує дії та функції сценарію cron, підкреслюючи його роль у виконанні запланованих завдань, критичних для роботи Moodle.*

Крім того, документ проливає світло на поширені причини повільної роботи Moodle і містить практичні поради щодо підвищення швидкості системи. Рекомендації включають оптимізацію ресурсів сервера, підтримку бази даних, впровадження стратегічного кешування та розумне використання мереж доставки вмісту (CDN). Підкреслюється важливість оновлення версій PHP і бази даних, а також переваги вимкнення невикористаних функцій і використання інструментів моніторингу сервера. Анотація наголошує на важливості бути в курсі оновлень Moodle для постійного покращення продуктивності та виправлення помилок, пропонуючи цілісне керівництво для оновлення та оптимізації систем Moodle.

Keywords: distand education, telelearning, moodle, administration, upgrade.
Ключові слова: дистанційна освіта, дистанційне навчання, moodle, адміністрування, оновлення.

I. INTRODUCTION

As you can see in this document and maybe already know the process of Moodle updating is *not a trivial one* and it is not just an update, but also a *huge upgrade*. Even if you have to make minor changes [1] because many things have to be considered. The process is even harder and complicated if you have to do it manually and if your hosting does not offer helper programs like cPanel and Softaculous. In general, there is a lack of broad information in this regard - how to approach the upgrade of Moodle under Linux or sftp. In this case, Ubuntu 20 is the server.

Why upgrade at all? As noted in Site administration > Notifications in the Moodle itself: "It is strongly recommended that you update your site to the latest version to obtain all recent security and bug fixes". In addition, higher versions of Moodle run on higher versions of PHP. For example, PHP 7.3 is 10-15% faster in service requests than PHP 7.1 [2, 3].

In this case, we have to upgrade from version 3.6.5+ (Build: 20190801) to version 3.9.9. ***However, the same procedure can be applied in the next versions of Moodle***. First, we'll upgrade to version **3.6.10** (released: 11 May 2020; support: maintained from December 2018 to May 2020), after that to **3.7.9** (released: 9 November 2020; support: maintained from May 2019 to November 2020), then to **3.8.5** (released: 9 November 2020; support: maintained from November 2019 to 9 November 2020, sustained status to 10 May 2021, receives only security updates) and finally to **3.9.9** (maintained from June 2020 to 10 May 2021, sustained status to 8 May 2023) [4].

Why stepwise? To check if the plugins (*450* are total plugins, *34* additionally added to this Moodle and the template (*Adaptable*) are compatible with newer versions of Moodle and will not be dropped.

The other thing to keep in mind is that it is not good to update to the latest version unless it is stable because there are many reports of problems.

1.1 Statistical description of the Moodle system

To get an idea of the scale of the system that we will update, statistical information is listed:

- Number of courses (1709);
- Number of users (23785) – in use for 8 years;
- Number of recently active users (1239);
- Number of role assignments (167913);
- Number of posts (990);
- Number of questions (7393966);
- Number of resources (22846);
- The average number of participants (99,40);
- Maximum number of users at the same time in the highest loads (500);
- The average number of recently active participants (15,44);
- The average number of course modules (28,15);
- Number of users with registered mobile devices (1118);
- Number of enabled prediction models (5);
- Number of generated predictions (125324);

Number of actions taken on generated predictions (4).

1.2 Statistical description of the database tables with the largest number of files

In addition, to get an idea of the complexity of the system and for completeness of the information presented, the following statistical information is listed:

- Table question_attempts: > 5 mln rows, > 9 GB;
- Table logstore_standard_log: > 21 mln rows, > 9 GB;
- Table question_attempt_steps: > 15 mln rows, > 3 GB;
- Table grade_grades_history: > 2 mln rows, > 1 GB;
- Table log: > 2 million rows, > 0.5 GB.

II. DISCUSSION

The upgrade methodology can be divided into four parts: *recommendatory pre-upgrade steps, mandatory pre-upgrade steps, peri-upgrade steps* (the upgrade itself) and *post-upgrade steps*.

2.1. Recommendatory pre-upgrade steps

They may include and are not limited to:

- Turn on Site administration > Server > Maintenance mode, with or without a message for the ordinary users Or
- Modifying the .htaccess file, allowing only the IP of the upgrade performer to access the Moodle during the upgrade itself;
- Another option if you do not filter the traffic according to the above point, is to configure the so-called upgrade key [5] in config.php. In this way, random users will not be able to activate the update process without the password from the upgrade key;
- Download all mentioned Moodle stable versions from Site administration > Notifications;
- Site administration > Development > Purge caches;
- Start cron - admin/cron.php?password="";
- Rename the folder of the "old" Moodle for example "Moodle_365" and extract the archive of the next version to which we plan to update;
- Transfer config.php from the "old" Moodle to the new one;
- Transfer .htaccess file from the "old" Moodle to the new one.

2.2. Mandatory pre-upgrade steps

Things to be considered before upgrade:

2.2.1. On the "mirror" copy

It is a good practice to clone the site (to make a "mirror copy, mirror environment" of your site). There to apply the updates and test the update after the update if everything is working fine. In addition, if everything works correctly to apply the updates on the real, so-called production site.

This can be done on your local computer using programs like xampp or on the server: On the DNS servers create a subdomain with an A record to the shared IP address of the server. Configure vhost for the new subdomain with standalone DocumentRoot. Copy the files from the production environment to the mirror environment. Clone the database (this can also be done with a dump file), and for the new database, you need to create a separate MySQL user. The new data is set in the configuration of the mirror environment.

2.2.2. On the production site

If the updates of the mirror copy have been ok, we move on to the production site:

- Dump the database for backup;
- Backup of the "moodledata" folder (this contains about 200,000 files and 65 GB);

- Transfer of the “moodledata” folder from the old Moodle to the new one. It is also possible without the following subfolders:

/moodledata/cache
/moodledata/localcache
/moodledata/sessions
/moodledata/temp
/moodledata/trashdir

- Backup of moodle folder in htdocs (web root) of the installed Moodle (this contains about 85,000 files and 10 GB);
- Transfer the moodle folder from the old to the new site;
- If you are paying a hosting company for a backup, you can skip the above steps;
- Copy and prepare all things which are missing from a pure, just extracted and installed Moodle installation, for example – if you have books in Book folder, in lib, mod and so one;
- Copy config.php from the "old" Moodle Or Modifying of config.php:

$CFG->wwwroot
$CFG->dataroot
$CFG->dbname
$CFG->dbuser
$CFG->dbpass

- *It is important just in case to save or make print screens of all settings in Site administration, especially if the template is additionally customized manually and it is not with default template settings;*
- *It is very important firstly to update the template, then to run the upgrade.*

2.2.3. Peri-upgrade steps

Things to be considered during the upgrade:

Visit your Moodle site with a browser. You should be taken to the upgrade script, which will lead you through the upgrade process. During the process, you will be notified about plugins requiring attention. Pay attention also if there are checks in red on the bottom. After starting, be patient and wait for the process to finish. It can take half an hour. Do the same for every stable planned to be installed version.

2.4. Post-upgrade steps

Things to be considered after the upgrade:

- Site administration > Appearance > Themes > Adaptable > Navbar Settings> Edit settings and Button display – after the upgrade in the "Adaptable" template, the button “Turn editing on” in a given course is missing;
- Move to the new installation all things which are missing from a pure, just extracted and installed Moodle installation, for example – if you have books in Book folder, in lib, mod and so one;
- Set up a cron task to call the file admin/cron.php every minute in Administration > Site administration > Server > Scheduled tasks;
- Start file admin/purgecaches.php;
- Settings in Registration and change of admin email address;
- Fixing if the template is customized and there are dislocations and lack of images;
- Turn off Maintenance mode;
- Finally, it is good to do *a test scenario* - check according to a preliminary plan whether all functionalities are working properly.

2.5. What is cron-script doing?

The following is a list of most cron activities:

- Convert submission attempt(s) for assignments;
- Predict models - Courses at risk of not being able to start;
- Execute adhoc task: mod_forum\task\send_user_notifications;
- Look for H5P updates;
- Removal of the old H5P records for mobile authentication;
- User suspension task: automatic user unsuspend from uploaded file;
- Clean-up contexts; Automated backups; Clean-up old sessions;
- Remove expired cache entries;
- Background processing for caches;
- Award badges;
- Indexing when searching the site;
- Complete learning plans which are due;
- Sync plans from learning plan template cohorts;
- Cleaning the recycle can on a course;
- Cleaning the recycle can on a category;
- Background processing for tags;

- Create missing contexts;
- Update all installed language packs;
- Cleanup of unverified incoming email;
- Activate/deactivate invalid rule subscriptions;
- Cleanup of temporary records for file conversions;
- Delete expired data request export files;
- Check for updates;
- Remove unused random questions;
- Execute adhoc task: core_course\task\course_delete_modules;
- Background processing for cleaning up question previews;
- Background processing for scheduled allocation;
- Incoming email pickup;
- Calculate regular completion data;
- Clean-up event monitor events.

2.6. Ways to increase the performance of slow Moodle systems and how the server to handle more concurrent users [6]:

- If a swap file is generated on the server (to 8-9 GB), this is a clear indicator of RAM lack. Buying more RAM will reduce disk-based activity related to swapping [7];
- There are benchmark plugins [8, 9, 10] that can find bottlenecks in the systems;
- Courses with over 1000 enrolled students load slowly;
- Courses with many materials uploaded in them, with many sections, blocks, resources and activities too. In this case, the following can be done: in the course settings to make not all sections to be on one page, but one section per page. “Show one section per page” can make the course page load 5%-20% faster [11];
- Set the cron to run every minute as instructed in the Moodle recommendations [12];
- Frequent upgrades and updates to Moodle and plugins. Of course a mandatory backup before that;
- Site administration > Reports > Performance overview > Enable statistics – if they are turned on and there are performance issues they can be temporarily stopped and the system behavior to be watched;
- Site administration > Server > Cleanup > Grade history lifetime – “If you experience performance problems or have limited database space, try to set lower value” – if there are problems the history of changes in the evaluation tables can be made first 1000 days, then 365 days, etc. and again to monitor the behavior of the system;
- Periodical checks for website security and malware scanner can be performed in appropriate free websites [13], also showing if there are any problems with the firewall;
- After all, the internet traffic, the bandwidth of the server is very important too. For example, experience shows, that to handle requests from about 500 concurrent users solving a test protected with a lockdown browser at the same time, up to about 150 - 220 Mbps (18.75 - 27.5 MB/s) is needed and this can be an extra bottleneck if everything else with the hardware is fine.

III. CONCLUSIONS

Comprehensive manual Moodle upgrade methodology in four steps is proposed, based on a real Moodle system described in detail.

The system maintenance and optimization operations performed by the cron script are described in detail.

Common problems with the slow operation of Moodle are shared both in courses with multiple users and materials and in the simultaneous use of the system by multiple users. Ways to solve them have been proposed.

IV. REFERENCES

Internet Sources, all accessed in 2023:
1. https://moodle.org/mod/forum/discuss.php?d=419838
2. https://bit.ly/3S69Nna
3. https://pressable.com/knowledgebase/differences-between-php-versions/
4. https://en.wikipedia.org/wiki/Moodle
5. https://docs.moodle.org/39/en/Upgrade_key
6. https://docs.moodle.org/39/en/Performance_FAQ
7. https://docs.moodle.org/39/en/Performance_recommendations
8. https://moodle.org/plugins/report_benchmark
9. https://moodle.org/mod/forum/discuss.php?d=57028&parent=772267%7Cthe
10. https://moodle.org/plugins/block_edwiser_site_monitor
11. https://www.elearningworld.org/making-your-moodle-course-faster/
12. https://docs.moodle.org/39/en/Cron
13. https://sitecheck.sucuri.net

Отримана в редакції 01.11.2023. Прийнята до друку 04.12.2023. Received 01 November 2023. Approved 12 December 2023. Available in Internet 03 January 2024