Liferay Digital Experience Platform Upgrade Benchmark

Benchmark Study of Upgrade Performance in Liferay DXP 7.1

Liferay

Table of Contents

Executive Summary

Liferay Digital Experience Platform (DXP) creates, manages and optimizes digital experiences across multiple customer touchpoints. Liferay provides a mechanism that automatically upgrades DXP data to the latest DXP versions.

The Liferay Support and Liferay Engineering tuned and tested realistic databases to optimize and time upgrading them to Liferay DXP 7.1.

The goals of this study were to:

- Gather statistics to help Liferay Global Services, Liferay Enterprise Subscription clients, and Liferay Service Partners plan capacity for upgrades.
- Provide optimal tuning instructions per database vendor.
- Detect and address upgrade bottlenecks and performance issues.

This document offers starting points and concepts for tuning DXP upgrades.

Important: Make sure to test your database configuration to determine tuning that's best for your system, and consult your DBA as appropriate.

Test Scenario

The original Liferay platform version tested was Liferay 6.2 EE Fix Pack 166. The test database was pruned following the pre-upgrade instructions.

Important: Pruning your database of unused objects can make upgrades faster. See the pre-upgrade instructions for details.

Here was the size of the database and document library and the number of languages the content was translated to:

- 3.2 GB Database
- 15 GB Document Library Filesystem
- Content translated to 3 different languages

In the test database, these entities had the most records:

- 1,694,000 rating entries
- 1,605,000 permissions (ResourcePermission objects)
- 871,000 assets (AssetEntry objects)
- 400,000 users
- 400,000 sites (Group objects)

- 402,000 images
- 259,000 message forum threads and posts
- 200,000 documents
- 193,000 portlet preferences
- 103,000 web contents (JournalArticle objects)
- 50,600 pages
- 3,276 journal article images
- 3,100 document folders

Benchmark Configuration and Methodology

The upgrade tests were performed in the following system conditions:

- Upgrade server with:
  - Ubuntu 16.04.3 LTS
  - 2 x Intel 8 Cores 2GHz CPU, 4Mb L2 Cache
- Liferay DXP 7.1 with Fix Pack 4 and the following fixes:
  - LPS-86779: To be completely added in a future fix pack.
  - LPS-90112: Added to fix pack 9.
  - LPS-92048: To be added in a future fix pack.
- Upgrade tool executed with Java version 1.8.0_171 and the following configuration:
  - Xmx 15 Gb RAM
  - File encoding UTF8
  - User timezone GMT

```
./db_upgrade.sh -j "-Xmx15000m -Dfile.
encoding=UTF8 -Duser.timezone=GMT"
```

- Database in the same network as the upgrade server. More information about the database server and configuration is provided for each database vendor next.

Benchmark Database Configuration

An upgrade's environment and database activities differ from those in production:

- Upgrade processes execute many more update statements (INSERT, UPDATE or DELETE) than SELECT statements.
- It's a completely safe environment because it's executed on a database backup on a DXP server separate from the current production server.

Taking these differences into account, we tuned the databases using these approaches:

- Deactivate data integrity measures that impact performance. Restore the backup if there is a failure during the process.
- Make commit-related I/O operations during transactions asynchronous.
- Increase the interval to flush commits to disk.
- Adjust the database log file capacity based on the total log capacity used in the test upgrade and the database vendor's recommendation.

Every vendor provides different ways to tune or configure the database server. We've assembled vendor-specific configurations that optimize upgrade performance for the test scenario described.

Note: The following database configuration is only valid for DXP upgrades. Never use this configuration to run DXP.

Note: On databases that have multiple schemas, global properties and configurations affect all the schemas. Before changing global properties or configurations, determine whether to isolate your test schema to its own database.

Microsoft SQL Server

Tests were performed using Microsoft SQL Server 2016 (RTM) - 13.0.1601.5 (X64) running on the following hardware platform:

- Windows Server 2012 R2
- 8 Gb RAM
- 2 x 4 Cores CPU 2.2 Ghz

The following setting was made in addition to the default configuration:

- Transaction Durability was set to FORCED (See the SQL Server documentation for details)

PostgreSQL

Tests were performed using Postgres 10 running on the same server as the upgrade tool.

The following changes were applied to the default configuration:

- Synchronous_commit set to OFF (PostgreSQL reference)
- Wal_writer_delay set to 1000ms (PostgreSQL reference)

Oracle Database

Tests were performed using Oracle 12.2.0.1.0 (12c Release 2) running on the following hardware platform:

- Windows Server 2012 R2
- 8 Gb RAM
- 2 x 4 Cores CPU 3.59 Ghz

Tests performed well from the beginning with no additional tuning. Oracle's default configuration already defines asynchronous I/O to disk.

MySQL

Tests were performed using MySQL 5.7 running on the same server as the upgrade tool.

The following modifications were applied to the default configuration:

- Innodb_flush_log_at_trx_commit set to 0 (MySQL reference)
- Innodb_doublewrite set to OFF (MySQL reference)

IBM DB2

Tests were performed using DB2 11.1 running on the following hardware platform:

- Windows Server 2012 R2
- 8 Gb RAM
- 2 x 4 Cores CPU 3.59 Ghz

During the tests, we collected statistics about the log usage in order to define the proper values. IBM has several articles about how to collect this information, and this one was particularly useful.

Here were the modifications applied to default database configuration:

- LOGFILSIZ set to 115250 (DB2 reference)
- LOGPRIMARY set to 6 (DB2 reference)
- LOGSECOND set to 3 (DB2 reference)
- Chngpgs_thresh set to 95 (DB2 reference)

This database type performed worse than other database types. Java batches executed during the upgrade took most of the time. Because the auto-commit parameter was set to true by default at JDBC driver level, every SQL statement ran alone in its own batch instead of being grouped with other statements. Batches using single updates don't behave properly with this configuration as it is explained in this IBM article. We'll investigate Java code changes to improve database upgrade performance for DB2. That analysis will be carried out in LPS-93363.

Benchmark Results

The following total results were obtained after the analysis of the database statistics and memory and CPU consumption during the upgrade. Several iterations were needed to properly tune and adjust each database and Java process.

Database	Core upgrade time	Modules upgrade time	Total time
Microsoft SQL Server	1 hour and 24 minutes	2 hours and 29 minutes	3 hours and 53 minutes
PostgreSQL	1 hour and 39 minutes	1 hour and 11 minutes	2 hours and 50 minutes
Oracle	40 minutes	1 hour and 44 minutes	2 hours and 34 minutes
MySQL	1 hour and 54 minutes	1 hour and 29 minutes	3 hours and 23 minutes
IBM DB2	4 hours and 44 minutes	12 hours and 15 minutes	16 hours and 59 minutes *

* See information above since it requires further analysis of auto-commit configuration

Conclusion

This study demonstrated the importance of database tuning for upgrade processes. Proper database tuning made database upgrades up to 78% faster.

Please take this benchmark as a starting point to plan your own database upgrade project. Optimal tuning depends on your data, infrastructure conditions and database vendor.

Analyze your data, tune for upgrades and time your test upgrades. Use this information to determine optimal database and Java process configurations for your DXP data upgrade.

Moving Forward

If you have any questions, please reach out to sales@liferay.com.

Liferay makes software that helps companies create digital experiences on web, mobile and connected devices. Our platform is open source, which makes it more reliable, innovative and secure. We try to leave a positive mark on the world through business and technology. Hundreds of organizations in financial services, healthcare, government, insurance, retail, manufacturing and multiple other industries use Liferay. Visit us at liferay.com.

191010