

# PerformanceTuning

#summary A collection of tips and tricks for improving Marmotta performance in different contexts

## Introduction

This page is intended as a collection of tips and tricks on how to improve Marmotta performance for different situations. The default configuration is meant to run on moderate machines and laptops, and therefore a bit conservative. If you have a high-end server with a lot of resources (processors, memory), you should consider the following improvements:

## Disabling Unnecessary Components

The Marmotta distribution comes with a big collection of components. If you don't need these functionalities, consider turning off or removing them. Removing a component is as easy as removing the dependency in the `pom.xml` or removing the jar file from the `WEB-INF/lib` directory. The following components are good candidates:

- marmotta-versioning: carries out versioning after each transaction and thus roughly doubles the time needed for committing data to the database; can be turned off by setting the configuration option "versioning.enabled" to false
- marmotta-reasoner: if reasoning rules are installed, the reasoner will be triggered by each transaction to run the rules over updated triples; can be disabled by setting `reasoner.enabled` to false or by not installing any rules

## Using PostgreSQL

In production environments with big amounts of data, you should use the PostgreSQL database instead of the embedded H2 database. PostgreSQL itself offers considerably better performance, and Marmotta implements a number of optimizations that make use of PostgreSQL features, e.g. native SPARQL querying. Therefore, **PostgreSQL is the recommended database** for real usages.

The database can be changed in the configuration section at the core module. . A restart is not required, the connection is changed in live. Currently, however, the data from the embedded database is NOT copied to the new database, so you would need to manually re-import your data.

In the default configuration, PostgreSQL is not configured for good performance. In order to get a decent database performance, improve PostgreSQL performance settings (see [http://wiki.postgresql.org/wiki/Tuning\\_Your\\_PostgreSQL\\_Server](http://wiki.postgresql.org/wiki/Tuning_Your_PostgreSQL_Server)). Try setting at least `shared_buffers` to 256M, `work_mem` to a higher value, and optionally also play with the `synchronous_commit` option.

## Multithreaded Task Execution

Marmotta is highly parallelized and thread safe and on appropriate machines making use of parallel task execution (e.g. imports, indexing, querying) may result in a considerable performance improvement.

## Memory and Caching

The minimum recommended, and what usually comes by default in the installed, is 1GB of memory for Marmotta. If you have enough memory, increasing this value will give Marmotta the opportunity to work with bigger transactions and do more caching and thus reduce the amount of I/O requests needed. You can increase the value by changing the `-Xmx` setting. On good machines, you can set it to 8GB or even higher.

To tune the caching settings, take a look at `ehcache*.xml` files and update the settings for the different caches there. You can monitor cache usage in the Admin Interface: Core, System menu.

For avoiding issues in some big data scenarios, due we guess some third-party libraries could be doing a [wrong usage of explicit garbage collection](#), we recommend to add `-XX:+DisableExplicitGC` to your JVM options.

## High-Performance Hardware

Given the tuning options described above, it is obvious that Marmotta can benefit significantly from faster hardware if configured properly. If you intend to run on high-end hardware, take into account the following parameters:

- number of processors/cores: can affect the number of threads you can run in parallel to perform tasks
- I/O performance: this is the most important parameter; run the PostgreSQL database and Marmotta Home directory on a Solid State Disk and you will get dramatic performance improvements
- memory: up to a certain level, increasing your memory can help, but the effect will not be as big as the previous two options. More than 12GB or RAM for Marmotta are rarely reasonable.