Contents

Improve speed performance

BlueSpice is optimized for fast read accesses. If your BlueSpice is running slowly, it can have different reasons. Often the search engine is not set up or the corresponding Tomcat server is not available. Also, an incorrect configuration (e.g., caches) may be the cause. Or the performance of your server is too low.

In this Howto you will find the most important solutions for increasing the speed of your BlueSpice instance.

Contents

Apache Tomcat Server not available

BlueSpice uses the search engine Apache Lucene / Solr, which supports important services of the wiki. Should saving a page take a long time, this is an indication that you have a problem with the search engine. Or you can simply enter a search query: If the input and the search page does not look as described here, the search engine is not installed correctly.

- Please check the Installation manual to see if you have completed the installation.
- Restart the Apache Tomcat Server.

Measures to increase performance

Specify cache directory

When you specify a cache directory, the Wiki uses the filesystem for some internal caches instead of the database

```
## Set $wgCacheDirectory to a writable directory on the web server
## to make your wiki go slightly faster. The directory should not
## be publically accessible from the web.
$wgCacheDirectory = "$IP/cache";
```

Please make sure that the folder named here can be described by your web server. See also the notes in the Installation manual.

Learn more: https://www.mediawiki.org/wiki/Manual:$wgCacheDirectory

Resize the name resolution of the database server

Especially under Windows, the name resolution of "localhost" can lead to delays. Use the IP address instead of the DNS name in the database. To do this, change the server configuration accordingly in LocalSettings.php:

```
$wgDBserver = "127.0.0.1"; #change IP value to your DB server IP
```

Enable memcached

Use Memcached as the central cache type. Note that Memchached must be installed and active on the server!

To do this in LocalSettings.php insert this configuration at the end:

```
$wgMainCacheType = CACHE_MEMCACHED;
$wgMemCachedServers = array( "127.0.0.1:11233" );
```

Enable Bytecode Cache in PHP

Since PHP 5.5, the bytecode cache opcache is delivered with. This must however be activated in the php.ini.

```
opcache.enable=1
```

Additional settings:

```
opcache.memory_consumption=512
opcache.max_accelerated_files=10000
opcache.validate_timestamps=1
opcache.revalidate_freq=2
```

After changing the php.ini, the web server must be restarted.

Learn more: http://php.net/manual/de/opcache.configuration.php und http://www.hostingadvice.com/how-to/enable-php-5-5-opcache-ubuntu-14-04/

Increase PHP Memory Limit

A higher memory limit of PHP results in a faster execution. This must be specified in php.ini:

```
memory_limit = 256M
```

After changing the php.ini, the web server must be restarted.

Set up JobQueue

Some tasks from MediaWiki can be delayed. Typically, they are processed by calling pages on the wiki. This can lead to delays. To prevent this, the JobQueue must be set up. Create a cronjob or task that runs this script every 10 minutes:

```
php maintenance/runJobs.php --maxjobs 100
```

Renew the L10N cache

Only in the case of absolute emergency, you can manually update the local cache, otherwise it will be rewritten every page call. To do this in LocalSettings.php insert this configuration at the end:

```
$wgLocalisationCacheConf['manualRecache'] = true;
```

Then execute the following command:

```
php /maintenance/rebuildLocalisationCache.php --force --lang="de,de-formal,en"
```

This command should also be set up as an hourly cronjob.

Learn more: https://www.mediawiki.org/wiki/Manual:$wgLocalisationCacheConf

Related info

- Performance Optimization: MySQL (MariaDB)