

Apache and MySQL

TYPO3 websites are usually operated on a LAMP environment (Linux/Apache/MySQL/PHP). In case when one has root on the environment of the webserver and the provider does not have TYPO3 expertise or cannot offer tuning services, one has to adjust the interplay of MySQL, PHP and Apache with the existing hardware. The server should only be used for operating the webserver. Email, anti virus, anti spam and other server services do not belong on a TYPO3 server. These services only increase the server load, complicate tuning especially for TYPO3 and offer additional surfaces for potential attacks – not only DoS attacks.

Available RAM of the server plays an important role for tuning. MySQL and Apache have to share the RAM. One should always keep in mind that the RAM restrictions should not be ignored: If this happens, the server needs to swap – outsource files from RAM to the harddrive and later get it back from there which often completely overloads the server.

Usually Apache is configured with „prefork“, which is the easiest alternative with PHP as a module. Each Apache process allocates its own memory. Keep in mind that each process can only serve one client at a time. This needs to be considered in the preferences „MaxClients“ for the peak time as well as for the average delivery time and available RAM.

Apache prefork Tunings

```
<IfModule prefork.c>
StartServers 10
MinSpareServers 12
MaxSpareServers 20
MaxClients 20
MaxRequestsPerChild 20
</IfModule>
# In order to keep less clients in "keep alive" state:
KeepAliveTimeout 5
```

Listing 2

When configuring „MaxClients“ for Apache one needs to find the right balance between „saving resources and being able to serve all requests immediately. Each pro Apache process including PHP module and memory for a regular TYPO3 delivery calls for 15 MB of RAM at the least. This means that „MaxClients 20“ in a „peak“ in which all clients access a TYPO3 site already allocates 300 MB. If one has a GB of RAM at one's disposal, this is not a problem, for a small virtual machine with 256 MB of RAM this can already be too much.

Now MySQL enters the stage: The MySQL tuning should be worked out roughly once before going live. It should be revised though about a week after going live because only then MySQL has been able to collect the necessary statistics, which should be used for tuning. In the ideal case this tuning should be repeated from time to time in order to initiate potential hardware upgrades in time. The RAM restrictions need to be considered for MySQL tuning as well and the parts which are allocated for Apache and the OS need to be subtracted. The preference „max_connections“ has direct impact on the overall size of the RAM which MySQL allocates. This should be in accord with „MaxClients“ of Apache.

MySQL-Tuning, my.cnf

```
[mysqld]
long_query_time = 2
# MaxClients of Apache
max_connections = 30
key_buffer_size = 16M
table_cache = 1024
```

```
query_cache_size = 32M
sort_buffer_size = 4M
```

Listing 3

Depending on the TYPO3 environment, the MySQL preferences should be adjusted. Especially the query cache should be set higher. Using an evaluation script [1] makes the tuning of the numerous MySQL parameters easier. In order to not exceed „max_connections“, PHP should not open a persistent MySQL connection, this can be done by including „mysql.allow_persistent = Off“ in php.ini „[MySQL]“.

Scalability

If the server needs to deliver many static files, includes several virtual servers or flexibility is an issue, it pays off to replace Apache „prefork“ with „workers“. This way it is possible to abstain from „mod_php“ and instead operate PHP as FastCGI: Leaner Apache processes and per-VirtualHost-PHP environments are possible then [2]. If one reaches the limits of scalability of a server, more powerful hardware is the easiest solution. If scalability needs to be considered from the start (for example to compensate short access peaks after sending out mass newsletters) one should think about using a proxy server in front of the webserver, using a webserver cluster in front of a database or applying methods such as file based caching.

Additional Details

The steps mentioned above are only the beginning of tuning. Operators of small sites need to choose the right provider (fast internet connection, state of the art hardware, appropriate operating system, good TYPO3 service) and can then tune their individual environments with simple means. Operators of professional sites with higher numbers in requests, guaranteed availability etc. should get professional help and favor a provider who is specialized on TYPO3 hosting. TYPO3 hosting packages or managed TYPO3 servers are usually preconfigured well for the standard use. In these cases the provider often is the TYPO3 service provider at the same time and can offer according services for tuning or a complete performance concept. ☒

Links and Literature [↗ Softlink 2326](#)

- [1] Tuning-Primer-Skript: <http://www.day32.com/MySQL/>
- [2] Using PHP with mod_fcgid: <http://typo3.org/development/articles/using-php-with-mod-fcgid/>

The Author



Ernesto Baschny has a master degree in computer sciences and is one of the TYPO3 core developers. He is manager of cron IT and has years of experience in operating webserver – especially in connection with TYPO3 – which he made when operating the data center of cron IT in Frankfurt.