Tuning-1: MySQL / MariaDB Daemon

Inhaltsverzeichnis

MySQL auf Debian mit Plesk

PageSpeed messen: http://developers.google.com/speed/pagespeed/insights/

Szenario

Für MySQL läuft EIN Daemon-Prozess, der alle Datenbanken des Servers mittels Threads bedient. Plesk aber muss davon ausgehen, dass viele Websites auf den Server kommen (jedenfalls bei Reseller-Hosting). Ferner ist es am Wichtigsten, möglichst viel Platz im RAM für VFS-Cache (Virtual File System) frei zu halten („warm“ VFS). Daher sind die Standard-Konfigwerten für MySQL bei Plesk eher mickrig – was eine sinnlose Einschränkung darstellt, wenn kein Reseller-Hosting vorliegt und Sie einen ganzen Server bzw. VPS für sich haben.

Spätestens dann wenn Sie versuchen mit ownCloud viele kleine Dateien auf den Server hochzuladen werden Sie merken, dass die Standard-Einstellungen vom MySQL-Daemon alles andere als optimal sind – der vom mysqld Daemon verbrauchte CPU-Zeit schnellt in die Höhe – während andere Prozesse CPU-Zeit in insgesamt Sekunden verbrauchen, hat mein mysqld eher Stunden verbraucht...

http://www.askapache.com/mysql/performance-tuning-mysql.html

und damit wird es klar, dass jede Instanz von MySQL nur dann sinnvoll „getunt“ werden kann, nachdem es einige Tage durchgelaufen ist.

So wird’s gemacht

Die Tools holen:

http://www.thomas-krenn.com/de/wiki/MySQL_Performance_Tuning

Script per **apt-get install mysqltuner** installieren; Aufruf einfach „mysqltuner“.

https://launchpad.net/mysql-tuning-primer (Script hier holen und selber installieren)

https://dev.mysql.com/doc/refman/5.5/en/server-system-variables.html

Ferner bietet phpMyAdmin – bei Verwendung auf der Server-Ebene – einen Hauptmenüpunkt „Status“ und darunter (ganz rechts) einen „Ratgeber“, der recht gut ist.

Im Plesk ist phpMyAdmin auf der Server-Ebene wei folgt erreichbar:
Tools & Settings | Applications & Databases | Database Servers | Werkzeug-Icon

MySQL-Config befindet sich in **/etc/mysql/my.cnf** ; aber nicht überschreiben, da die Datei evtl. von Plesk überbuttert wird; fügen Sie stattdessen eine Datei in **/etc/mysql/conf.d** hinzu (muss mit „.cnf“ enden um ausgeführt zu werden).

Zum Zeitpunkt des Schreibens habe ich 17 MySQL-DBs auf dem Server, die von Plesk, WordPress, ownCloud, eGroupware et. a.l benutzt werden; da jede Installation wohl einmalig ist – in Zusammensetzung und Nutzung (Last) – muss jede Installation individuell „getunt“ werden. Das Ergebnis meines eigenen Tunings gebe ich als Beispiel im Anhang.

Nach dem Einspielen der Tuning-Config: MySQL Service startet nicht mehr?

Im Error-Log* steht: InnoDB: Error: log file ./ib_logfile0 is of different size

Siehe Fix hier. * /var/log/mysql/error.log

Übrigens: DataDir ist bei Ubuntu: /var/lib/mysql (hier befindet sich die DB-Datei),

also: `rm -f /var/lib/mysql/ib_logfile*; service mysql start`

Joins ohne Indexe

Häufig als Problem genannt (und gefunden) sind Tabellen-Joins, die keine oder falschkonfigurierte Indexe verwenden. Die Joins sind dann ungleich aufwendiger für MySQL auszuführen als sonst – man wird angehalten, seine Tabellen mit Indexen zu ergänzen bzw. Abfragen umzuschreiben. Schön und gut, aber was ist wenn die Software, die diese Joins verursacht, nicht von einem selbst geschrieben wurde?

Ich habe beim Tuning das „Slow Query Log“ aktiviert und „log_queries_not_using_indexes“ aktiviert – und siehe da, gibt es Einträge von

- Cyclos (nun deaktiviert da nicht in Gebrauch)
- Plesk Backup Manager (habe nirgends finden können wie man ihn ausschaltet)
- Plesk Passwort-Abfragen (da werde ich partout nichts ändern)
- WordPress Optionen-Abfragen (aber von welchem Plugin?)
- PiWiK

ABER in fast allen Fällen handelt es sich um einen Fehlalarm: Der eigentliche Abfrage geht ein „SET timestamp“ voraus, vermutlich gemeint um die Abfrage aus dem Slow Query Log fern zu halten, aber mit umgekehrter Wirkung.

MySQL LOAD DATA INFILE

Als Thema entdeckt: Piwik | Administration | Diagnostic | System Check

„Using LOAD DATA INFILE will greatly speed Piwik's archiving process up. To make it available to Piwik, try updating your PHP & MySQL software and make sure your database user has the FILE privilege. If your Piwik server tracks high traffic websites (eg. > 100,000 pages per month), we recommend to try fix this problem.“

Error: LOAD DATA INFILE failed... Error was:

Try #1: LOAD DATA INFILE : SQLSTATE[28000]: Invalid authorization specification: 1045 Access denied for user 'piwikadmin'@'%' (using password: YES)[28000]

Troubleshooting: FAQ on piwik.org: http://piwik.org/faq/troubleshooting/#faq_194

https://dev.mysql.com/doc/refman/5.5/en/load-data.html

https://dev.mysql.com/doc/refman/5.5/en/insert-speed.html

https://dev.mysql.com/doc/refman/5.5/en/privileges-provided.html#priv_file

In der Tat, wie die Handbuchseiten und Piwik-Meldung ahnen lassen, dem Plesk DB-User fehlt das FILE-Privileg.

In Plesk:

Home | Subscriptions | {Webspace wählen} | Databases [Reiter] | User Managment (Link oben rechts) | {User wählen} | Global Privileges: File (Häkchen setzen)

Damit hat der User das Privileg - doch es klappt noch nicht:

Error: LOAD DATA INFILE failed... Error was:

Try #1: LOAD DATA INFILE : SQLSTATE[HY000]: General error: 13 Can't get stat of '/var/www/vhosts/{htdocsdir}/tmp/assets/piwik_option-....csv' (Errcode: 13)

Denn der MySQL-Daemon braucht Zugriff auf den Webspace von Piwik:

"give the process mysqld executable-access (+x) to files in path/to/piwik/tmp/assets/* and all parent folders":

usermod -aG psaserv mysql

service mysql restart

Spezifika zu MariaDB auf Debian in 2021

Wait Timeout

Wenn man keine Prozesse hat, die inaktive Verbindungen mit temporären DB-Tabellen lange offen halten, kann man zur Sicherheit diese nach einer Anzahl Sekunden vom DB aus schließen lassen:

wait_timeout = 600 # Default 28800 (1)

interactive_timeout = 600 # Default 28800 (2)

(1) The number of seconds the server waits for activity on a noninteractive connection before closing it.

(2) The number of seconds the server waits for activity on an interactive connection before closing it.

ACHTUNG: Wer Matomo (als WordPress-Plugin jedenfalls) benutzt, darf diese Werte nicht herabsetzen, sonst hagelt es PHP-Fehler wegen zu früh gelöschter Temporärtabellen.

Begrenzung der Anzahl aller offenen Dateien

Ob ein Limit überschritten wird, sieht man im Protokoll vom Starten des DB-Services:

[Warning] Could not increase number of max_open_files to more than 16384 (request: 64122)

Hier kann man mit einer eigenen, ergänzenden Config-Datei die Limits hochsetzen:

md /etc/systemd/system/mariadb.service.d

cat >/etc/systemd/system/mariadb.service.d/mariadbIncreaseFileLimit.conf

[Service]

LimitNOFILE=65536

LimitMEMLOCK=65536

^D

systemctl daemon-reload

systemctl restart mysql

Weblinks zum Thema MySQl Tuning

stackoverflow.com/questions/13259275/mysql-tmp-table-size-max-heap-table-size

dev.mysql.com/doc/refman/5.5/en/server-system-variables.html

stackoverflow.com/questions/5474662/mysql-optimize-all-tables

tuning.cnf für MySQL Daemon

```
#
# Tim Reeves MySQL database server performance tuning file
#
# Tuned to Plesk / Ubuntu 12.04 MySQL 5.5 with 2 vCores and 4GB RAM
#          17 DBs mit insg. 667 InnoDB-Tabellen und 439 MyISAM-Tabellen.
#
# Copy this to directory /etc/mysql/conf.d

[mysqld]

thread_cache_size      = 64   # Was 8
max_connections        = 40   # Was 100 commented out => 151
max_user_connections   = 40   # Was not set => 0
max_connect_errors     = 20   # Was not set => 10

open_files_limit       = 3000 # Was default 1024
table_open_cache       = 3000 # Was not set, default 400
table_definition_cache = 3000 # Was not set, default 400

#thread_concurrency     = 10  # I have "2vCores", but lscpu shows 48...

# Query Cache Configuration
query_cache_limit        = 8M   # Was 1M
query_cache_size         = 192M # Was 16M - "32MB for every 1GB of RAM"
query_cache_min_res_unit = 1400 # Was 4K by default

# InnoDB
innodb_buffer_pool_size         = 512M # Was set to measly 2M, Default is 128M -
but ownCloud alone has 50M tables...
innodb_log_buffer_size          = 16M  # Default is 8M - was set to 500K
innodb_log_file_size            = 128M  # Was default 5M - set to one quarter of
innodb_buffer_pool_size
innodb_additional_mem_pool_size = 16M  # Default is 8M - was set to 500K
innodb_flush_log_at_trx_commit  = 0
  # Was 1 = Full ACID: 0 = mysqld process crash can erase last second transactions
innodb_thread_concurrency       = 0    # Was a measly 2
 # A recommended value is 2 times the number of CPUs plus the number of disks.
 # A value of 0 (the default) is interpreted as infinite concurrency (no
concurrency checking).
innodb_use_native_aio           = 1    # Should be ON by default (says docu) but
was off.

# Various

sort_buffer_size = 4M      # 1MB for every 1GB of RAM
read_buffer_size = 4M      # 1MB for every 1GB of RAM
read_rnd_buffer_size = 8M  # 1MB for every 1GB of RAM
join_buffer_size = 32M     # Was only 128K

tmp_table_size = 384M      # 128 reicht nicht
max_heap_table_size = 384M
```

```
key_buffer_size = 8M           # Was default 16M (we use mainly InnoDB)
myisam_sort_buffer_size = 8M   # This is the default

long_query_time = 5
# slow-query-log = 1
# slow_query_log_file = /var/log/mysql/mysql-slow.log
# log_queries_not_using_indexes = 1
```

Tipp: /etc/mysql/conf.d, Plesk 12

Mischen Sie die Art der Anweisungen in Zusatz-Konfigdateien in /etc/mysql/conf.d besser nicht, so kann man dann an deren Namen sehen, was sie machen (und im Umkehrschluss, was nicht). Hier ein Beispiel – namens **only-from-localhost.cnf**:

```
[mysqld]
bind-address = 127.0.0.1  # Prevents remote access by limiting to localhost
```

Plesk 12 (12.0 und 12.5) realisiert die Zugangskontrolle über die Eigenschaften der verschiedenen Benutzkonten, die er mit MySQL anmeldet. Diese kann man gut per phpMyAdmin auf der Server-Ebene sehen. Dennoch erlaubt diese Methode den Verbindungsaufbau von Außen von beliebigen Ips. Wenn Sie sicher sind, dass immer nur Verbindungen von localhost auf einem Server zulässig sind, können Sie dies in der grundsätzlichen Config des MySQL-Daemons wie oben einrichten.

Caveat: Der eigene Eingriff auf der Config-Ebene des MySQL-Daemon verhindert eine Öffnung des Servers für einen berechtigten Zugriff von außen mittels Plesk – doch das sieht man nicht in Plesk und wundert sich, warum es nicht klappt…