

Title

MariaDB Administration Part 1

Author

Bernard Szlachta (NobleProg Ltd)

Subfooter

MariaDB Administration Part 1

Bernard Szlachta (NobleProg Ltd)



Slide Show

Contents

- 1 MODULE 1
 - 1.1 MySQL and MariaDB history
 - 1.2 Release Background
 - 1.3 MariaDB Server Installation
 - 1.4 Installing in Ubuntu
 - 1.5 Starting and stopping server
 - 1.6 Installing on Windows
 - 1.7 Hardware and MariaDB
 - 1.8 MySQL and Platform
 - 1.9 MariaDB and OS
 - 1.10 MariaDB Storage Engines
 - 1.11 MySQL Storage Engines
 - 1.12 MyISAM vs. InnoDB
 - 1.13 MyISAM vs. InnoDB
 - 1.14 MariaDB Server Files and Scripts
 - 1.15 MariaDB Programs
 - 1.16 MariaDB Programs
 - 1.17 Ways of starting a server
 - 1.18 Stopping MariaDB Server
 - 1.19 Changing Forgotten Password
 - 1.20 Changing Forgotten Password
 - 1.21 Running Scripts
 - 1.22 Using Client Tools
- 2 MODULE 2
 - 2.1 MariaDB Server Configuration
 - 2.2 mysqld server options
 - 2.3 Passing options to MariaDB server
 - 2.4 Checking variable value
 - 2.5 Parameters Formats
 - 2.6 Restricting maximum value of a variable
 - 2.7 Server Options for Loading Plugins
 - 2.8 System Variables
 - 2.9 Session System Variables
 - 2.10 Server Status Variables
 - 2.11 Local (non system) Variables
 - 2.12 Using Local Variables to replace subqueries
 - 2.13 Select into variable
 - 2.14 Server SQL Modes
 - 2.15 The Shutdown Process
 - 2.16 MODULE 3
 - 2.17 MySQL Security Issues
 - 2.18 General Security Guidelines
 - 2.19 Security Guidelines
 - 2.20 Security Issues with LOAD DATA LOCAL
 - 2.21 Memory Configuration

MODULE 1

MySQL and MariaDB history

- MySQL founded 1995
- acquired by Sun Microsystems 2008
- now owned by Oracle Corporation since 2009
- soon after MariaDB started by original authors of MySQL

Release Background

- 1994 - MySQL by Michael (Monty) Widenius and David Axmark
- First internal release on 23 May 1995
- Windows version was released on 8 January 1998 for Windows 95 and NT
- Version 3.23 January 2001
- Version 4.0 March 2003 (unions)
- Version 4.1 October 2004 (R-trees and B-trees, subqueries, prepared statements)
- Version 5.0 October 2005 (cursors, stored procedures, triggers, views)
- Sun Microsystems acquired MySQL AB on 26 February 2008.
- Version 5.1 November 2008 (event scheduler, partitioning, plugin API, row-based replication, server log tables)
- MySQL 5.1 and 6.0 showed poor performance when used for data warehousing — partly due to its inability to utilize multiple CPU cores for processing a single query.[26]
- MariaDB initial release - 22 January 2009
- Oracle acquired Sun Microsystems on 27 January 2010. Oracle and Sun
- December 2012 - MariaDB foundation has been established
- On October 1, 2014, SkySQL Corporation Ab changed its name to MariaDB Corporation Ab

MariaDB Server Installation

- Install the server discussing different options
- GUI Tools Installation
- Quick Overview on GUI Tools

Installing in Ubuntu

- <https://downloads.mariadb.org/mariadb/repositories/#mirror=coreix>

Starting and stopping server

Linux

```
sudo service mysql start
sudo service mysql stop
```

Windows (as a service)

```
net start mysql
```

MariaDB 10.0 starting from command line

```
mysqld --console
```

Installing on Windows

- <https://downloads.mariadb.org/>
- Follow installer steps

Hardware and MariaDB

- A lot of MariaDB features has been created when memory and CPU power was expensive and not flexible
- I try to avoid things specific to MariaDB
- For example, the physical hard drive failure is not an issue with cloud computing
- As for partitioning, it can be handle by the storage itself (S3, RAID, etc...) Usually you can buy it very cheaply
- SSD drives are the most disruptive technology
- Optimization vs hardware, it is usually cheaper to buy extra resources, more complicated optimization is the last resort

MySQL and Platform

- Using Debian, Ubuntu or other Linux distrubtion, you don't have to do upgrade and huge chunk of maintenance work yourself
- Windows 32bit memroy limitation is 3.2 GB, around 60% of optimization problems was related to this limitation
- Changing default behaviour means that the changes must be documented or automated during the application installation process. This can depend on the platform as well.
- The most suitable platforms are Linux and Windows. Other systems (despite vendors promises) do not have so big community support
- Choosing platform is important and does effect the effort needed to maintain the system

MariaDB and OS

- Some configurations options of the server depend of the platform.
- Using frameworks and database separation libraries (persistence), like Spring, Drupal, etc... can avoid common mistakes
- If your database is too big and you "think" you need new features, look at the data from "business" perspective.
 - Is it redundant?
 - Does it has to be stored in the main database?
 - Can we delete or move old data?
- Usually good data design and cleansing process solves 80% of cases I encountered. Using partitioning, replication or cluster is much more expensive compare to business like solution.
- Smart SOA allows you to avoid big monolithic applications with huge databases

MariaDB Storage Engines

- MariaDB supports several storage engines that act as handlers for different table types.
- Some table are transaction-safe and nontransaction-safe

```
SHOW ENGINES; shows supported storage engine
```

MySQL Storage Engines

InnoDB/XtraDB

- A transaction-safe (ACID compliant)
- Has commit, rollback, and crash-recovery capabilities
- InnoDB row-level locking and Oracle-style consistent nonlocking reads
- Good for increase multi-user concurrency and performance
- InnoDB stores user data in clustered indexes to reduce I/O for common queries based on primary keys
- Supports FOREIGN KEY referential-integrity constraints

MyISAM

- Mainly for web and data warehouses
- Doesn't support transactions and foreign keys

Memory (HEAP)

- Stores all data in RAM for extremely fast access in environments that require quick lookups of reference and other like data

Merge

- Logically groups a series of identical MyISAM tables and reference them as one object
- Good for VLDB environments such as data warehousing

Archive

- Provides the perfect solution for storing and retrieving large amounts of seldom-referenced historical, archived, or security audit information.

CSV

- Stores data in text files using comma-separated values format.
- Easily exchange data between other software and applications that can import and export in CSV format
- Use to store log files

Blackhole

- accepts but does not store data and retrievals always return an empty set
- Can be used in distributed database design where data is automatically replicated, but not stored locally.

More:

- <https://mariadb.com/kb/en/mariadb/documentation/storage-engines/>

MyISAM vs. InnoDB

MyISAM	InnoDB
Few DML operations	OLTP with a lot of DML
Very fast for selects	Slower for selects
Some data can get corrupted	Good crash-recovery facility
Cannot be a MASTER server in replication, but can be a SLAVE	Perfect for MASTER server in replication
	Supports FOREIGN KEY
	Binary logs
	Point-in-time recovery

MyISAM vs. InnoDB

Feature	MyISAM	InnoDB
Storage limits	256TB	64TB
Transactions	No	Yes
Locking granularity	Table	Row
MVCC	No	Yes
Geospatial indexing support	Yes	No
Full-text search indexes	Yes	No
Clustered indexes	No	Yes
Data caches	No	Yes
Cluster database support	No	No
Foreign key support	No	Yes
Point-in-time recovery	No	Yes

MariaDB Server Files and Scripts

- MariaDB Programs
- Server Programms
- Client Programms
- GUI and other tools

MariaDB Programs

- SERVER
 - **mysqld** - the MySQL server
 - **mysqld_safe** - a script which starts mysqld, recommended of starting the server
 - restarts the server when an error occur
 - logs run-time and failure information
 - **mysqld_multi** - script that can start or stop multiple servers installed on the

system (different ports or sockets)

- CLIENT
 - **mysql** - a command line client
 - **mysqladmin** - administrative operations
 - reloading the grant tables
 - creating/dropping database
 - flushing tables to disk
 - managing log files
 - shutdown the database

MariaDB Programs

- MariaDB Database
 - Visual database design application
- MariaDB Galera Cluster
 - Database Cluster
- Client Libraries (Java/C, etc..) and ODBC Drivers
 - Connectivity from applications
- Graphical Clients
 - <https://mariadb.com/kb/en/mariadb/documentation/clients-and-utilities/graphical-clients/>

Ways of starting a server

- Windows or Unix service
 - `sudo /etc/init.d/mysql start`
- Mysqld_safe
 - `sudo mysqld_safe`
- Command line mysqld (Unix)

```
sudo mysqld
```

- Command line Windows (MariaDB)

```
mysqld --console
```

Stopping MariaDB Server

- Windows/Unix Service
 - `sudo /etc/init.d/mysql stop`
 - `sudo service mysql stop`
 - `net stop mysql` (Windows)
- mysqladmin
 - `mysqladmin -u root -p shutdown`

Changing Forgotten Password

- Run

```
mysqld --skip-grant-tables --skip-networking
```

- This option causes the server to start without using the privilege system at all, which gives anyone with access to the server unrestricted access to all databases
- Executing
 - mysqladmin flush-privileges or
 - mysqladmin reload or
 - FLUSH PRIVILEGES
- Change your password
 - set password for root@localhost = password('asdfasdf')

Changing Forgotten Password

- In order to prevent external connections accessing the databases you can change the port and the bind interface

```
mysqld --bind-address=127.0.0.1 --port 333 --skip-granttables
```

- You can connect to the database with the command below

```
mysql -P 333 -h 127.0.0.1
```

Running Scripts

- download File:Tables.sql

From MySQL prompt

```
use test
source filename
```

From Command Line

```
cat filename | mysql (Unix only)
```

Windows

```
type c:\Tables.sql | mysql -u root -pasdf test
mysql -u root -pasdf test < c:\Tables.sql
```

- GUI Tools

Running a SQL statement from shell

- echo select * from EMP | mysql test;

Saving errors to a file

- mysql --force < tables.sql 2> error.log

Using Client Tools

- System Variables
 - Windows

```
SET USER=your_name
```

- Unix the currently logged in user in the system

```
MYSQL_TCP_PORT=3306; export MYSQL_TCP_PORT
mysql -u user -ppassword -h host database
```

- Config file (my.ini or /etc/mysql/my.cnf)

```
[client]
host=host_name
user=user_name
password=your_pass
```

MODULE 2

MariaDB Server Configuration

- mysqld Options
- Server System Variables
- Dynamic System Variables
- Server Status Variables
- The Server SQL Mode
- Shutdown Process

mysqld server options

- MariaDB Server manages access to the MySQL data directory that contains databases and tables.
- For a complete list of options, run this command:

```
mysqld --verbose --help
```

Passing options to MariaDB server

- System variable
- Cmd-Line
- Option file
- System Var
- Status Var [session or global or both]

Checking variable value

```
mysql> show variables like '%query_cache%'
```

Variable_name	Value
have_query_cache	YES
query_cache_limit	1048576
query_cache_min_res_unit	4096
query_cache_size	0
query_cache_type	ON
query_cache_wlock_invalidate	OFF

6 rows in set (0.01 sec)

Parameters Formats

- Command Line Format

```
mysqld --sync-binlog=#
```

- Config File Format

```
sync-binlog
```

- Option Sets Variable Yes,

```
sync_binlog
```

- Variable Name

```
sync_binlog
```

- Variable Scope Global
- Dynamic Variable
- Permitted Values

```
Type numeric  
Default 0  
Range 0-4294967295
```

Restricting maximum value of a variable

```
mysqld --maximum-var_name=value
```

Server Options for Loading Plugins

- Enable the plugin

```
--plugin_name[=ON]  
--plugin_name=1  
--enable-plugin_name
```

If plugin initialization fails, start the server anyway, but with the plugin disabled

```
--plugin_name=FORCE
```

If plugin initialization fails, do not start the server

- Disable plugin

```
--plugin_name=OFF  
--skip-plugin_name  
--plugin_name=0  
--disable-plugin_name
```

System Variables

- System variables
 - indicate how the server is configured
 - accessible via "Select @@variablename"
- Session Variables
 - cannot be set at server startup
- Status Variable

```
show status;
```

- Scope of the variable
 - GLOBAL
 - SESSION

Session System Variables

- exist only as session variables
- these cannot be set at server startup
- can be assigned values at runtime using the SET statement (except for those that are read only)
- Most of them are not displayed by SHOW VARIABLES, but you can obtain their values using SELECT.

```
E.g. autocommit
select @@autocommit
MariaDB [(none)]> set SESSION autocommit=OFF;
MariaDB [(none)]> SELECT  @@GLOBAL.autocommit;
+-----+
| @@GLOBAL.autocommit |
+-----+
|                      1 |
+-----+
1 row in set (0.00 sec)
```

```
MariaDB [(none)]> SELECT  @@SESSION.autocommit;
+-----+
| @@SESSION.autocommit |
+-----+
|                      0 |
+-----+
1 row in set (0.00 sec)
```

Server Status Variables

- Status variables provide information about its operation
- SHOW [GLOBAL | SESSION] STATUS
- GLOBAL keyword aggregates the values over all connections
- SESSION shows the values for the current connection.
- FLUSH STATUS statement restarts the status variables
- Cannot be access like system variables via select statement "select @@tmpdir"
- SHOW GLOBAL STATUS;
- SHOW SESSION STATUS LIKE 'Com_show_status';

Local (non system) Variables

- Single "at character"
- Visible only in the session
- Useful with SQL scripts

```
mysql> set @max_salary=20000;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select @max_salary;
+-----+
| @max_salary |
+-----+
|          20000 |
+-----+
1 row in set (0.00 sec)
```

```
MariaDB [test]> select * from emp where sal > @max_salary;
```

Using Local Variables to replace subqueries

```
MariaDB [test]> select * from emp where sal > (select avg(sal) from emp);
```

ID	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	BUSH	PRESIDENT	NULL	1981-11-17	5000.00	NULL	10
7698	BLAIR	MANAGER	7839	1981-05-01	2850.00	NULL	30
7782	MERKEL	MANAGER	7839	1981-06-09	2450.00	NULL	10
7566	PUTIN	MANAGER	7839	1981-04-02	2975.00	NULL	20
7902	TOOSK	ANALYST	7566	1981-12-03	3000.00	NULL	20
7788	CARNEGIE	ANALYST	7566	1982-12-09	3000.00	NULL	20

```
6 rows in set (0.00 sec)
```

```
MariaDB [test]> select avg(sal) from emp;
```

avg(sal)
2073.214286

```
1 row in set (0.00 sec)
```

```
MariaDB [test]> select avg(sal) from emp into @avg_sal;
Query OK, 1 row affected (0.00 sec)
```

```
MariaDB [test]> select * from emp where sal > @avg_sal;
```

ID	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	BUSH	PRESIDENT	NULL	1981-11-17	5000.00	NULL	10
7698	BLAIR	MANAGER	7839	1981-05-01	2850.00	NULL	30
7782	MERKEL	MANAGER	7839	1981-06-09	2450.00	NULL	10
7566	PUTIN	MANAGER	7839	1981-04-02	2975.00	NULL	20
7902	TOOSK	ANALYST	7566	1981-12-03	3000.00	NULL	20
7788	CARNEGIE	ANALYST	7566	1982-12-09	3000.00	NULL	20

```
6 rows in set (0.01 sec)
```

Select into variable

```
select password from mysql.user where host='localhost' and user='root' into @pass;
```

Server SQL Modes

- An SQL mode changes the syntax of the SQL and some client behaviours
- Modes can apply differently for different clients

- Application written to a different SQL dialects can be used or migrated with less effort
- The default value is empty (no modes set)
- Can cause corruption with partitioning

```
SET [GLOBAL|SESSION] sql_mode='modes '
SELECT @@GLOBAL.sql_mode;
SELECT @@SESSION.sql_mode;
```

- ANSI, DB2, MSSQL, ORACLE, POSTGRESQL, TRADITIONAL
- More info

<http://dev.mysql.com/doc/refman/5.7/en/sql-mode.html>

The Shutdown Process

- The shutdown process can be initiated several ways.
 - execute a mysqladmin shutdown command.
 - system-specific methods: Unix - SIGTERM signal, Windows - services manager
- The server might create a thread to handle the shutdown process if necessary.
- The server stops accepting new connections to prevent new activity from being initiated during shutdown
- The server terminates current activity
 - open transactions are rolled back
 - nontransactional table, multiple-row UPDATE or INSERT may leave the table partially updated
- Storage engines are shut down or closed.
 - table cache is flushed
 - all open tables are closed.
- The server exits.

MODULE 3

MySQL Security Issues

- Securing MySQL Against Attacks
- Security-Related mysqld Options
- Security Issues with LOAD DATA LOCAL

General Security Guidelines

- Don't display MySQL credentials in your applications
- Do not use root account
- Never grant privileges to all hosts
- Try mysql -u root
- SHOW GRANTS statement to check which accounts have access to what
- Do not choose passwords from dictionaries
- Put MySQL behind the firewall or in a demilitarized zone (DMZ).
- SQL InJection
- Check the size of data before passing it to MySQL
- Use socket instead of tcp/ip

Security Guidelines

- The password is not transmitted in clear text over the connection
- All other information is transferred as text
 - SSL, VPN, SSH
- Don't run MySQL server as root (Unix/Linux) or administrator (Windows)
- Do not grant the PROCESS or SUPER privilege to nonadministrative users. SHOW PROCESSLIST can contain passwords
- Do not grant the FILE privilege to nonadministrative users
- If you do not trust your DNS, you should use IP numbers

- Control number of connections with max_user_connections variable or GRANT statement

Security Issues with LOAD DATA LOCAL

- The LOAD DATA loads file on the server host or on the client (LOCAL keyword)
- In a Web environment LOAD DATA LOCAL can read any files that the Web server process has read access
- You can disable all LOAD DATA LOCAL commands from the server side by starting mysqld with the --local-infile=0 option.

Memory Configuration

```
tmp_table_size=16M
sort_buffer_size=208K
key_buffer_size=13M
read_buffer_size
innodb_buffer_pool_size=22M
```