

Objective Analysis of MySQL and MySQLVariant

Vikas Khanna¹, Dr. Mohammad Amjad²

¹Department of Computer Science & Engineering, Al-Falah School of Engineering & Technology
Dhauj, Faridabad, Haryana, India

²Department of Computer Engineering, Jamia Millia Islamia University, New Delhi, India

ABSTRACT

RDBMS technology is favorable in software based organizations for more than three decades. In year 2008, MySQL AB Company was acquired by Sun Microsystem due to most popular open source RDBMS program MySQL and first choice of large data management corporations spanning over a wider range of internet. The biggest event happened in year 2009 by acquisition of Sun Microsystem with Oracle. The decision was criticized by Michael Widenius and David Axmark, the co-founders of MySQL AB. Michael Widenius resigned from Sun Microsystems after acquisition and eventually developed a fork of MySQL called MariaDB before merger of Sun Microsystem with Oracle. Forks can be considered as a mini standard version of MySQL. Different forks were created over the period with respect to this most widely used open source relational database management software program. For an example Percona server with XtraBackup was created from MySQL 5.0 GA release. The different forks can be considered as variants of MySQL with different capabilities. The paper presents an idea is to do the analysis of MySQL and MySQL variants like MariaDB with respect to different sources owned over the period.

Keywords: MySQL database; Relational Database; MySQL variants, MariaDB

I. INTRODUCTION

In early 1990's the use of database in web based application was the choice of MySQL due to its fast and flexible capabilities. "Monty" Widenius developed this database and named after his daughter name My and SQL that stands for structured query language. The alternate database won't be a choice for people if they want to have an ease of use, reliable and open source database. Over the years this database is more stable and reliable and it is used by small scale to large based organizations. The number of year's effort placed by MySQL Ab Company to produce such a capabilities based product into open source market.

The company was acquired by Sun Microsystems in year 2008. In turn, Sun Microsystems was acquired by Oracle Corporation in 2009. There are two release of MySQL editions after merger of Sun

Microsystems with Oracle. One is released under open source community edition that uses the GPL license [1] and another released as an enterprise edition treated as a closed source. The founding father of MySQL called Monty and the core developers with the MySQL development team developed MariaDB. The reason was to develop this database due to quality of releases released by Oracle for MySQL in early days after acquisition period. The most and but obvious fear was the open source code base relational database management system will turned down towards closed source database in future and due to this reason Monty created a fork of MySQL version into open source community and used this fork to develop MySQL variant named as a MariaDB under GPL v2 license[2].

II. METHODS AND MATERIAL

A. Evolution of MySQL and MySQL Variants

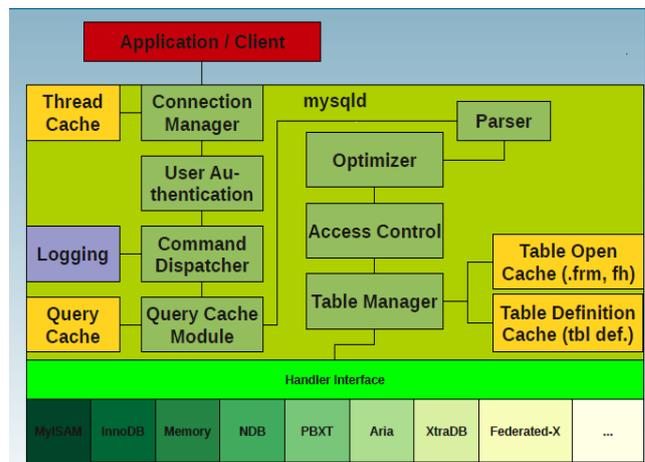
MySQL is open source code based software programs and downloaded more than 100 million times. Unireg database is the root of MySQL server development that was developed by Monty for a Swedish company named as TcX in the period of 1980. The SQL interface was added by Monty to Unireg and the first release of MySQL server was done in 1995. In this period MySQL AB Company was formed by Monty, David and Allan Larsson together [3]. The first transaction support provided by MySQL in year 2001 after integration of Berkley database and InnoDB engines. MySQL server had added number of capabilities to make it enterprise database server over the period with several releases. The following table presents different releases of MySQL and most often used MySQL variants after the acquisition of Sun Microsystem by Oracle.[4][5][6]

MySQL Release History		
Year	Version	Release By
1995	MySQL First Release	TcX
2001	MySQL Version 3.2.3	MySQL AB
2003	MySQL Version 4.0	MySQL AB
2004	MySQL Version 4.1	MySQL AB
2005	MySQL Version 5.0	MySQL AB
2005	Oracle acquired InnoDB Database Engine was the part of MySQL database	
2008	MySQL Version 5.1	MySQL.com under the hood of Sun Microsystem
2010	MySQL Version 5.5	Oracle after the acquisition of Sun in year 2009
2013	MySQL Version 5.6	Oracle
2014	MySQL Version 5.7	Oracle
2015	Active Development on MySQL Version 5.5,5.6 and 5.7	Oracle
MySQL Variant Percona Server Release History		
Year	Version	Release By
2009	Percona Server Version 1.0.3	Percona Server
2010	Percona Server Version 5.1	Percona Server
2011	Percona Server Version 5.5	Percona Server

2013	Percona Server Version 5.6	Percona Server
MySQL Variants MariaDB Release History		
2010	MariaDB Version 5.1/5.2	MariaDB Foundation
2012	MariaDB Version 5.3	MariaDB Foundation
2013	MariaDB Version 5.3	MariaDB Foundation
2014	MariaDB Version 10	MariaDB Foundation

B. MySQL And MySQL Variants Architecture

MySQL server was written with number of modules and those modules are developed to perform specific functions. While start of MySQL server the module responsible for initialization read and parse the configuration file typically named as my.cnf or my.ini and loads the necessary global memory and required structures to perform another module related tasks. The following architecture diagram was referenced from internet. [7]



- Connection Manager Module
It is responsible to take connections from Application/Client once MySQL database server was started properly with complete initializations
- User Authentication Module
This module responsible for user authentication who connects to MySQL database server.
- Command Dispatcher Module
Once successfully authentication takes place, the thread responsible for making a connection will pass requested data to command dispatcher module. The requested data can be classified into two categories

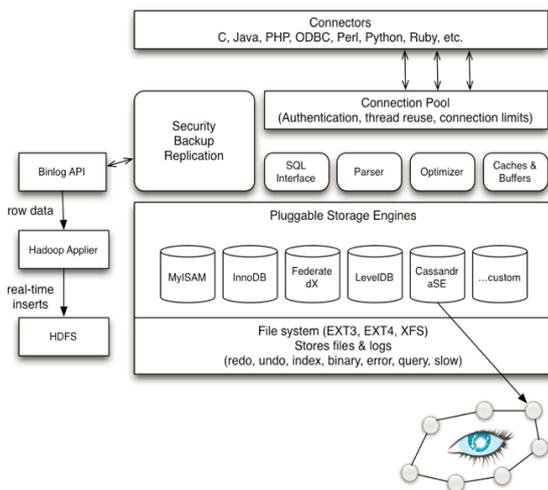
mainly command and query. The execution of command type of request does not require a parser but for query execution parser is an essential requirement.

○ Query Cache Module

This module is responsible to pass queries to parser. This module keeps track of the query can be cached. If the same query exists into cached result and still valid then this module will serve the requested data back to connected client from the cache.

○ Parser Module

This module is responsible to pass queries to parser once requested data is not available into query cache. For different type of requested data parser module will transfer to different modules. For an example the select type of operation it will pass the information to optimizer for query optimization. For update type of operation it will pass information to table modification module. Once parser module pass control to different type of modules based on requested data which in turn passed that data to access control and then to table manager modules to open the tables and acquire necessary locks. In this stage table operation will continue by doing the interaction with different storage engines through handler interface. The MariaDB architecture is similar to the architecture of MySQL as it is the



fork edition of original MySQL Server with extra added capabilities with the addition of different storage engines like Apache Cassandra Storage engine. The MariaDB server over the years provided more capabilities with the clear objective of making source code will never be a closed entity of any organization in

future. As the early hype created during acquisition done by Oracle to Sun Microsystem started remaining truth because of late release of MySQL server capabilities as a general availability release and closing the bugs reported database to open source community.

C. MySQL build using source code

Due to nature of open source code anyone who is interested to understand the code structure component wise then first that individual requires to download the code from source code repositories and do the required configuration with all pre-requisites software under different operating systems. The process to configure and build the MySQL server from source code can be considered broadly for two operating systems Microsoft Operating System and another one is UNIX based operating system i.e. Linux.

The following is the pre-requisite list of software programs to build and configure the source code for MySQL before running into system.[8]

- ✓ Bazaar Version Control System
- ✓ GNU Make
- ✓ AutoConf version 2.5.8 or greater
- ✓ Libtool 1.5.24 or greater
- ✓ M4
- ✓ Bison
- ✓ Visual Studio Express of Microsoft Operating System
- ✓ Windows Platform SDK
- ✓ CMake
- ✓ Cygwin
- ✓ WiX

The steps wise instructions placed below to build and configure the MySQL from source code for Linux platform.

Step 1: MySQL Repository Initialization Using Bazaar VCS

```
> mkdir mysql-server
> bzz init-repo-trees mysql-server
Step 2: MySQL Branch at local system
> cd mysql-server
> bzz branch lp:mysql-server/5.1 mysql-5.1
```

Step 3: Copy Active branch

```
> cd mysql-server
```

```
>bzr branch mysql-5.1 mysql-5.1-build
```

Step 4: Prepare Source Tree

```
> cd mysql-5.1
```

```
> autoreconf --force --install
```

Step 5: Configure and Compile

```
> ./configure [options]
```

```
> make
```

```
> make install
```

Step 6: Make and Install

```
> ./configure [options]
```

```
> make
```

The steps wise instructions placed below to build and configure the MySQL from source code for window based platform.[9]

Step 1: Obtain MySQL Source code and configure using any version control system like Bazaar VCS.

Step 2: Source build using CMake utility Win/configure.js used to create the configuration file used by CMake. The configuration options can be defined based out the capabilities to be the part of MySQL Server installation. For an example

```
cscript win\configure.js
```

```
WITH_INNOBASE_STORAGE_ENGINE
```

```
WITH_PARTITION_STORAGE_ENGINE
```

```
WITH_BLACKHOLE_STORAGE_ENGINE
```

Step 3: Create Visual Studio solution file CMake utility will create the solution under each directory based out configuration opted for building MySQL server.

Step 4: Build the solution Build files will build the solution under work directory. The only important point is to choose correct one build according to the visual studio environment installed into system.

Step 5: Open solution in Visual Studio Environment and choose the configuration and press the build button. The tool will compile the source code.

D. MySQL/MariaDB and NoSQL Integration

The handler socket plugin was included into MySQL and MariaDB [10]. It operated as a daemon process inside mysql process i.e. mysqld. The handler socket is much faster than mysql daemon process due to primary reason of no involvement in parsing SQL statements. It

does not require table open process along with locking mechanism. It also not involved in SQL execution plans as well as unlocking and closing of tables.

E. MySQL and MariaDB Differences

This is very common question about both have common functionalities and then where is the difference lies between these two different programs. MariaDB have added new features and security improvements as well as better way for testing along with bug fixes. The primary reason behind core developer MySQL team was remaining all time with Monty and now they all are working under MariaDB foundation. The major classification in terms of differences is

- ✓ Easy to Use
- ✓ User statistics features introduced to identify the source of load into databases by just querying few more commands
- ✓ All date time related data types have a highest precision up to microsecond support.
- ✓ Test suite involved more test cases to conduct the test in more better manner
- ✓ Limited number of bugs and warnings
- ✓ Performance improvement
- ✓ Addition of more storage engine like Apache Cassandra engine to handle big data volumes
- ✓ Connect storage engine provides variety of data formats for read/write data
- ✓ TokuDB native plugin into MariaDB provides 20 times fast insertion of data in compare to InnoDB engine

III. RESULTS AND DISCUSSION

MySQL Plugin Development

The plugin application programming interface provided by MySQL and MySQL variants allowed creating component using any third party languages and load those components during runtime environment. The server extension can be done using MySQL plugin API. The component was developed using curl library integration to get the data from hypertext protocol through structured query language interface. In similar manner the data can be post, put and delete through SQL interface.

```
mysql>select
http_get('http://maps.google.cn/maps/api/geocode/xml?address=Race+Course+New+Delhi,+Mountain+View,+In')
+
|<?xml version="1.0" encoding="UTF-8"?>
<GeocodeResponse>
<status>OK</status>
<result>
<type>locality</type>
<type>political</type>
</formatted_address>
<address_component>
<type>locality</type>
<type>political</type>
</address_component>
<address_component>
<short_name>DL</short_name>
<type>administrative_area_level_1</type>
<type>political</type>
</address_component>
<address_component>
<short_name>IN</short_name>
<type>country</type>
<type>political</type>
</address_component>
<geometry>
<location>
<lat>28.6139391</lat>
<lng>77.2090212</lng>
</location>
<location_type>APPROXIMATE</location_type>
<viewport>
<southwest>
<lat>28.4010669</lat>
<lng>76.8396999</lng>
</southwest>
<northeast>
<lat>28.8898160</lat>
<lng>77.3418147</lng>
</northeast>
</viewport>
<bounds>
<southwest>
<lat>28.4010669</lat>
<lng>76.8396999</lng>
</southwest>
<northeast>
<lat>28.8898160</lat>
<lng>77.3418147</lng>
</northeast>
</bounds>
</geometry>
<partial_match>true</partial_match>
<place_id>ChIJLbZ-NFv9DDkRzk0gTkm3wII</place_id>
</result>
</GeocodeResponse>
|
1 row in set (3.46 sec)
```

```
mysql>select
http_get('http://maps.google.cn/maps/api/geocode/xml?address=Race+Course+New+Delhi,+Mountain+View,+In')
+
|<?xml version="1.0" encoding="UTF-8"?>
<GeocodeResponse>
<status>OK</status>
<result>
<type>locality</type>
<type>political</type>
</formatted_address>
<address_component>
<type>locality</type>
<type>political</type>
</address_component>
<address_component>
<short_name>DL</short_name>
<type>administrative_area_level_1</type>
<type>political</type>
</address_component>
<address_component>
<short_name>IN</short_name>
<type>country</type>
```

IV. CONCLUSION

This paper introduced the overview of Mysql and its variants database management system and surveyed the current state-of-the-art. Due to many challenging research problems and a wide variety of practical applications, the research in the field has been very active in past seven years with the advent of MySQL derivatives like Percona server, MariaDB etc. At the end of this section wherein some of the Computer Programs were written to demonstrate the plugin component and the results for the same are also placed as an output.

V. REFERENCES

- [1] The Definitive Guide to MySQL 5
- [2] GPLv2 license is available at:
- [3] <http://kb.askmonty.org/en/mariadb-license>
- [4] MySQL Administrative Bible's
- [5] <http://www.mysql.com>
- [6] <http://www.percona.com>
- [7] <https://mariadb.org>
- [8] <http://www.fromdual.com/>
- [9] <https://dev.mysql.com/doc/refman/5.1/en/installing-development-tree.html>
- [10] <https://dev.mysql.com/doc/refman/5.1/en/windows-source-build.html>
- [11] <https://mariadb.com/kb/en/mariadb/handlersocket/>