# Study and Optimization Based on MySQL Storage Engine

**Article** *in* Advances in Intelligent and Soft Computing · January 2011

DOI: 10.1007/978-3-642-25986-9_28



**3 authors**, including:

Y.H. Gu
Beijing University of Posts and Telecommunications
**16** PUBLICATIONS **71** CITATIONS

SEE PROFILE

# Study and Optimization Based on MySQL Storage Engine

*Xiaolong Pan*
School of Computer Science
Beijing University of Posts and Telecommunications
Beijing, China
wfpxl@163.com

Weiming Wu and Yonghao Gu
School of Computer Science
Beijing University of Posts and Telecommunications
Beijing, China
wuwming@bupt.edu.cn

***Abstract*—MySQL is an open database of the most common. One of the important characteristics of MySQL is that it provides rich storage engines. In the structure of Web applications, it is very important for the performance of database that the system must choose different storage engines based on different needs. But the default settings performance of MySQL is very not good, so it is necessary to optimize when the system uses MySQL. This paper discusses the most two important storage engines in MySQL, and raises the performance of MySQL by optimize MySQL based on different storage engines.**

**Keywords- MySQL; Storage Engines; Performance Optimization**

## 1 Introduction

With the increasing development of the information technology, information of the world has expanded rapidly and database has been widely used. Database has become the base and core of modern computer systems and computer applications.

As the most widely used open source database, MySQL provides different kinds of storage engine (table type). Different storage engine uses different storage mechanism, indexing techniques and lock level so they can provide different features and capabalities.

The most basic task of database is storing and managing data, and the only feature that end-user can feel is the performance of the database: what is the speed of the database to process a inquiry action and return the result to the user's application. Most application examples show that as the dynamic growth of the data in applications, the efficiency of data query will reduce. Especially for massive data management and query, the problem will become more prominent. So it will be much important to improve the efficiency of dynamic growing database query.

This paper uses the MySQL application experience to compare some current important storage engines, and then elaborates MySQL database optimization depending on different storage engines.

## 2 My SQL Storage Engines

The data in MySQL is stored in files (or memory) by different techniques. By choosing different techniques, the users can get extra speed or functions and then improve the overall function of the application.

These different techniques and their supporting related functions are called storage engine (or table type) in MySQL as shown in figure 1. MySQL has the default configuration of different kinds of storage engines. Users can pre-set or start them in MySQL server.

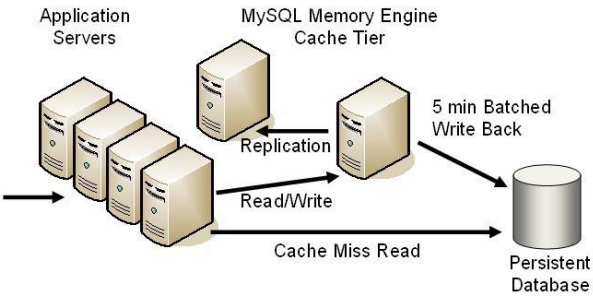


**Figure 1. The Working principle of Memory Engine**

In MySQL (this case for MySQL server 5.1.55), users can use “show engines” commands to get a list of available engines.

```
mysql> show engines;
+------------+---------+----------------------------------------------------------------
--------+
| Engine | Support | Comment |
+------------+---------+----------------------------------------------------------------
--------+
| MyISAM | DEFAULT | Default engine as of MySQL 3.23 with great performance |
| MEMORY | YES | Hash based, stored in memory, useful for temporary tables |
| InnoDB | YES | Supports transactions, row-level locking, and foreign keys |
| BLACKHOLE | NO | /dev/null storage engine (anything you write to it disappears) |
| ARCHIVE | NO | Archive storage engine |
| CSV | NO | CSV storage engine |
| FEDERATED | NO | Federated MySQL storage engine |
| MRG_MYISAM | YES | Collection of identical MyISAM tables |
+------------+---------+----------------------------------------------------------------
--------+
8 rows in set (0.01 sec)
```

This form shows all the available database engines and whether the current database server can support these engines.

## 2.1 MyISAM 引擎

## 2.1 My ISAM Engine

（1）MyISAM

MyISAM is the default storage engine in MySQL 5.1, and it is first provided by MySQL. MyISAM doesn’t support transaction and foreign key. Its superiority is that it has a high access speed. The applications that don’t have a requirement of transactional integrity or SELECT & INSERT-based applications basically use this engine to create tables. Most websites are based on inquiry so that they can use this engine.

Every table of MyISAM is stored as three files in the disk. The files and table names are the same, but the extensions are as follows:

①.frm(store the definition of tables)；

② .MYD(my data, store the data of the tables)；

③ .MYI(my index, store the indexes of the tables，MyISAM forms use B-tree indexes)。

（2）MyISAM Engine Classification

MyISAM is divided into static MyISAM, dynamic MyISAM and compressed MyISAM:

Static MyISAM: if the length of data columns in the table is pre-fixed, the server will use this table type automatically. Because the space of every record in the table is the same, the storage and updating efficiency of this table is very high.

Dynamic MyISAM: if varchar,xxxtext or xxxBLOB field are occurred in the table, the server will use this table type automatically. Relative to static MyISAM, this king of table has a small occupied storage space. But because the length of every record is different, after several revisions of data, the data of the table will be stored in the memory discretely, then the efficiency will drop.

Compressed MyISAM: the two table types mentioned above both can be compressed by myisamchk. This kind of table further reduces the storage space. But it can not be modified after compressed.

However, no matter which kind of MyISAM table, it can now support transaction, row-level locking and foreign key constraint.

## 2.2 InnoDB Engine

（1）InnoDB

InnoDB provides MySQL transaction-safe tables which have the transaction,rollback and crash recovery capabilities, multi-versioned concurrency control functions. These features increase multi-user and deployment functions. There is noneed to expand the lock function in InnoDB, because row-level locking in InnoDB fits very small space. In SQL query, users can mix up tables of InnoDB and of other types freely, even in the same query.

The design goal of InnoDB is to maximize performence as handling large volumes of data. It has the best CPU utilization of all other disk-based relational database.

Technically, InnoDB storage engines are fully integrated with the MySQL server. InnoDB storage engines maintain their own buffer pool to cache data and indexes in the memory.

InnoDB put data and indexes in table space. Which is different from others is that the table space may include multiple files. For example, in MyISAM, tables are stored in a single file. The table size of InnoDB is only limited by the

file size of the operating system. Each table can also has its own table space just by staring "innodb_file_per_table".

## 3 MyISAM 与 InnoDB 对比研究

InnoDB and MyISAM are the most common storage engines when people use MySQL. Each has advantages and disadvantages depend on different applications.

MyISAM is the extension of earl ISAM. ISAM is designes to suit for the situation that frequencies are much greater than the write rate. ISAM also occupies less space in the memory. Except inheriting this advantage, MyISAM also provide a number of useful new features and tools to keep up with the times. For example, it provides table-level lock considering the concurrency control. Though MyISAM doesn't support fault-tolerant itself, it can recovery the fault by myisamchk. And that because in MyISAM， each table has its own storage file (MYD data file and MYD index file), it will be very convinient to backup and restore. MyISAM has mosts tools with check and repair functions compared with other storage engines. MyISAM tables can be compressed, and they support fulltext retrieval. They are not transactional-safe, and they don't support foreign key.

InnoDB is designed for the situation of high concurrent read and write. It uses MVCC (Multi-Version Concurrency Control) and row-level locking to provide transaction support complied with ACID. InnoDB supports foreign key referential integrity and it has the capabality of fault recovery. In addition, the performence of InnoDB is not bad, especially in the case of handling large amounts of data. The words of official are: the CPU efficiency of InnoDB is the best of all the disk-based relational databases. It is not applicative for it to use shared table space and simple copy methods. It must recover the data after suspending MySQL. It has the same features with BDB. It also support foreign key. InnoDB table is very fast and has richer features than BDB. So if a transactional-safe storage engine is needed, InnoDB is a good choice.

The comparison of the two storage engines is shown in table 1:

| Attribute | MyISAM | InnoDB |
|---|---|---|
| Transactions | No | Yes |
| Lock granularity | Table | Row |
| Storage | Split files | Table space(s) |
| Isolation levels | None | All |
| Portable format | Yes | Yes |
| Referential integrity | No | Yes |
| Primary key with data | No | Yes |
| MySQL caches data records | No | Yes |
| Availability | All versions | All Versions |

Table-1 Comparison of MyISAM and InnoDB

To sum up, it is clear that：

●MyISAM is more suitable for the case that there is less insert and more query: It doesn’t support transaction, foreign key and WAL(write ahead logging) and it can only lock the full table.

●InnoDB is suitable for the case of large concurrent write and query: it supports transaction (ACID compatible), row-level locking, and foreign key. And it has its own memory buffer pool and independent table space (without large limitation).

## 4 Performance Comparisons and Optimization of the Two Storage Engines

### 4.1 Performance Comparisons

To further understand the specific differences in performance of the tow storage engines, this paper specifically give a simple test:

The tested database version： mysql Ver 14.14 Distrib 5.1.55, for Win32 (ia32)。

[Inserted data-1] (innodb_flush_log_at_trx_commit=1)
MyISAM 10W：22/s
InnoDB 10W：2010/s

[Inserted data-2] (innodb_flush_log_at_trx_commit=0)
MyISAM 10W：28/s
InnoDB 10W：27/s

[Inserted data-3] (innodb_buffer_pool_size=1024M)
InnoDB 1W：3/s
InnoDB 10W：33/s
InnoDB 100W：607/s

[Inserted data-4] (innodb_buffer_pool_size=256M, innodb_flush_log_at_trx_commit=1, set autocommit=0)

InnoDB 1W：3/s

InnoDB 10W：26/s

InnoDB 100W：379/s

It is clear that in MySQL 5.0, the performance differences between MyISAM and InnoDB is not big.

### 4.2 Performance Optimization

Based on the above test data, different optimization strategies are needed for different storage engines.

(1) For MyISAM， the main optimization aspects include:

● key_buffer_size: Number of buffers which are distributed to MyISAM index cache

● query_cache_size: The number of caches distributed to query cache.

● long_query_time: Set slow query time

● external-locking: Prohibit the use of external lock and prevent deadlock

● back_log: The number of requests can be stacked before temporary stops responding new requests. If the users need to allow a large number of collections in a short time, this valued can be raised.

● table_cache: The number of cache data table to avoid the spending to open the table repeatedly.

● thread_cache_size: The number of threads the cache can use to reduce the spending of creating new thread.

● sort/join/read buffer size: The memory distributed to every thread to process sorting or scanning table link and index.

(2) (2) For InnoDB, the main optimization aspects include:

●If the CPU usage of database is lower than 70 percent, the pressure of MySQL may be disk factor. There may be too many transactions and submissions or the buffer pool is too small.

● Put INSERT, UPDATE and DELETE into the same transaction. But the resulting efficiency should also be paid attention.

● innodb_flush_log_at_trx_commi: Set is as 0(refresh every second), 1(real-time refresh),2(only write log file and not refresh to the disk).

● Use bigger log file and log buffer.

● Close autocommit mode when importing data into InnoDB, or it will refresh new data to the disk.

● innodb_buffer_pool_size: Control the size of buffer pools which are distributed to cluster data and secondary index page. The default is 16MB.

● innodb_additional_mem_pool_size: Control the size of pools which are distributed to InnoDB internal data dictionary to sort. The default is 1MB.

● innodb_log_buffer_size: Control the size of buffers which are distributed to InnoDB memory to write log file in advance. The default is 1MB.

● innodb_log_files_in_group: The number of log files in the log group. InnoDB write files circularly. The default is 2.

● innodb_log_file_size: The default is 5MB. The recommended value is from 1MB to one Nth of the buffer pool. N is the number of log files in the group.

## 5 Conclude

Database storage engine is a important part of database. To assure the conditions of each storage engine is an important precondition of optimizing the database storage engines. This paper sums up the features of the two engines and introduces the optimization strategies by studying the two main engines. Through research and optimization, the advantages of MySQL can be played better.

## References


[1] PL/.SQL User's Guide and Reference Release 1 (10.1) http://www.stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10807/toc.htm

[2] Dan Tow. SQL Tuning. O'Reilly Media. November 2003.

[3] Cary Millsap, Jeff Holt. Optimizing Oracle Performance. Sept. 16, 2003.

[4] Liu Liping. How to Improve the Inquirement & Statistics Speed of MySQL ZCOM,2005