MySQL Storage Engine Comparison Guide

June 2009

Table of Contents

Introduction

The MySQL pluggable storage engine architecture allows a database professional to select a specialized storage engine for a particular application need. It shields a developer or DBA from the need to manage any specific application coding requirements. The MySQL server architecture encapsulates the application programmer and DBA from the low-level implementation details at the storage level providing a consistent and easy application model and API. So while there are different capabilities across different storage engines, the application is shielded from these.

A MySQL storage engine is a low-level engine inside the database server that takes care of storing and retrieving data, and can be accessed through an internal MySQL API or, in some situations be accessed directly by an application. Note that one application can have more than one storage engine in use at any given time.

This document provides a description of all currently GA and nearly-GA internal (developed by MySQL/Sun) and external/third-party storage engines that may be used with the MySQL Server, and offers comparisons and contrasts that will help a database administrator, developer, or system architect make the correct choice regarding which storage engine(s) should be used for a particular application.

Key Storage Engine Differentiation Points

Each particular storage engine combines a unique set of properties and qualities, which help determine its fitness for a particular purpose. Some of these key differentiators include:

- **Concurrency** – some applications have more granular lock requirements (such as row-level locks) than others. Choosing the right locking strategy can reduce overhead and therefore help with overall performance. This area also includes support for capabilities like multi-version concurrency control or “snapshot” read.
- **Transaction Support** – not every application needs transactions, but for those that do, there are very well defined requirements like ACID compliance and varying levels of concurrency.
- **Referential Integrity** – the need to have the server enforce relational database referential integrity.
- **Physical Storage** – this involves everything from the overall page size for tables and indexes as well as the format used for storing data to physical disk.
- **Index Support** – different application scenarios tend to benefit from different index strategies, and so each storage engine generally has its own indexing methods, although some (like B-tree indexes) are common to nearly all engines.
- **Memory Caches** – different applications respond better to some memory caching strategies than others, so while some memory caches are common to all storage engines (like those used for user connections, MySQL’s high-speed Query Cache, etc.), others are uniquely defined only when a particular storage engine is put in play.
- **Performance Aids** – includes things like multiple I/O threads for parallel operations, thread concurrency, database checkpointing, bulk insert handling, and more.
- **Data Organization** – this generally refers to whether tables/indexes are row or column-oriented for applications like traditional OLTP vs. Data Warehousing.
- **Miscellaneous Target Features** – this may include things like support for geospatial operations, security restrictions for certain data manipulation operations, and other like items.

- **Future Features** – this could include features such as data encryption, data auditing, OLAP, and much more.

Each set of the pluggable storage engine infrastructure components are designed to offer a selective set of benefits for a particular application. Conversely, avoiding a set of component features helps steer clear of unnecessary overhead. So it stands to reason that understanding a particular application's set of requirements and selecting the proper MySQL storage engine can have a dramatic impact on overall system efficiency and performance.

MySQL/Sun Developed Engines

The following storage engines are developed and maintained internally by MySQL and Sun.

Archive

Overview

The Archive storage engine (available in MySQL 5.0 and above) is designed for non-transactional applications that manage historical, archived, and seldom-referenced data. The Archive engine uses compression algorithms to shrink most data down to 20% of its original size, and is therefore excellent for storing large amounts of data that may be queried, but won't need to be changed (the engine does not allow UPDATE/DELETE/REPLACE statements to be executed on its data). The Archive engine does not support the use of indexes, except for sparse indexes on autoincrement columns and has no crash-recovery support.

Target Applications

- Data warehouses / data marts
- Web / Web 2.0 logging applications
- Data auditing/security-based applications

Key Industries Utilizing

- eCommerce
- Manufacturing
- Media
- Retail
- Technology
- Telecom
- Travel

Pros and Cons

Pros
No practical storage limits
Best engine in terms of storage cost
Fastest insert speed of any disk-based storage engine
Special protection against UPDATE/DELETE changes
Fast query performance for large table scans (faster than MyISAM)

Unlimited row-level locking
MVCC-like/Snapshot read support

Cons
No transactional support
Index support only on auto-increment columns
No crash recovery
Blocking online backup
No support for foreign keys
No UPDATE/DELETE statements allowed
No memory cache support

Further Reading

MySQL Manual: http://dev.mysql.com/doc/refman/5.1/en/archive-storage-engine.html.
MySQL Forums: http://forums.mysql.com/list.php?112.
Dev Zone Article: http://dev.mysql.com/tech-resources/articles/storage-engine.html.

Blackhole

Overview

The Blackhole storage engine (available in MySQL 5.0 and above) accepts incoming data, but does not keep or retain it. The engine thereby becomes a "black hole" for data. The primary use of the Blackhole engine is to replicate data from a master server to many slaves, without retaining the data on the master server. The Blackhole storage engine has no support for transactional operations.

Target Applications

- Specialized replication and scale-out application designs

Key Industries Utilizing

- eCommerce
- Technology
- Web/Web 2.0

Pros and Cons

Pros
Good for certain replication situations
Can be used to measure overhead of replication functions

Cons
None that are meaningful given the purpose of the engine

Further Reading

MySQL Manual: http://dev.mysql.com/doc/refman/5.1/en/blackhole-storage-engine.html.

CSV

Overview

The CSV (comma separated value) storage engine (available in MySQL 5.1 and above) is designed for non-transactional applications that need the ability to reference data in flat file format, using standard file editors. The CSV engine stores data in text files using a comma-delimited values format and thus allows for flat file data-manipulation and access via standard file editors in addition to database operations, using SQL, via MySQL.

Target Applications

- Applications needing access to external data sources

Key Industries Utilizing

- eCommerce
- Manufacturing
- Media
- Technology

Pros and Cons

Pros
No practical storage limits
Instantaneous data loading into MySQL (simply rename any comma-separated value file to the table name created in MySQL) no matter how big the file size.
Provides easy access to external data sources

Cons
No transactional support
No index support
No crash recovery
Blocking online backup
No support for foreign keys
No memory cache support
Security vulnerability from external file manipulation.

Further Reading

MySQL Manual: http://dev.mysql.com/doc/refman/5.1/en/csv-storage-engine.html.
MySQL Forums: http://forums.mysql.com/list.php?127.
Dev Zone Article: http://dev.mysql.com/tech-resources/articles/csv-storage-engine.html.

Falcon

Overview

The Falcon storage engine is designed to handle high-speed transactional applications that require little or no user concurrency contention, and need rapid response times as well as crash-recovery support. The Falcon storage engine is currently in pre-production testing and is targeted for GA release in 2010.

Target Applications

- Web / Web 2.0 transactional applications / Web services
- Service Oriented Architecture applications
- eCommerce transactional systems
- Financial systems
- Health care applications
- Retail systems
- Embedded transactional applications
- General departmental/business systems

Key Industries Utilizing

- Legal
- eCommerce
- Defense
- Education
- Energy
- Finance/Banking
- Healthcare
- Manufacturing
- Media
- Retail
- Technology
- Telecom
- Travel
- Web / Web 2.0

Pros and Cons

Pros
ACID-transaction compliance
Support for crash recovery
High storage limit (110TB)
Efficient storage handling (compressed rows)
Optimized BLOB handling
Unlimited row-level concurrency
MVCC support (readers don't block writers and vice versa)
Advanced B-Tree index support
Very fast rollback of unwanted transactions
Advanced memory cache mechanisms

Cons
No current foreign key support
No full-text or GIS index support

Further Reading

MySQL Manual: http://dev.mysql.com/doc/falcon/en/index.html.
Dev zone article: http://dev.mysql.com/tech-resources/articles/falcon-transactional-engine-part1.html
Dev zone article: http://dev.mysql.com/tech-resources/articles/primer_on_falcon_tablespaces.html.

Federated

Overview

The Federated storage engine (available in MySQL 5.0 and above) is not a pure storage engine in the sense of the other MySQL storage engines, but is instead a distributed table definition that allows remote access to MySQL tables located on different physical servers. Its purpose is to allow the creation of logical databases from two or more physical database servers and it akin to database links or distributed tables used in other RDBMS systems.

Note that currently in MySQL 5.1 and above, the Federated storage engine is disabled by default due to some quality issues and other engineering priorities.

Target Applications

- Any applications requiring transparent distributed database support.

Key Industries Utilizing

- eCommerce
- Finance
- Government
- Manufacturing
- Media
- Retail
- Technology
- Telecom
- Travel
- Web/Web 2.0

Pros and Cons

Pros
Allows for seamless distributed table access
Works with all internally-developed MySQL storage engines and some third party (e.g. InnoDB)

Cons

No transactional support
Does not utilize the MySQL query cache
WHERE pushdown conditions are limited on remote servers
Currently disabled by default in MySQL 5.1 and higher

Further Reading

MySQL Manual: http://dev.mysql.com/doc/refman/5.1/en/federated-storage-engine.html.
MySQL Forums: http://forums.mysql.com/list.php?105.
Dev Zone Article: http://dev.mysql.com/tech-resources/articles/mysql-federated-storage.html.

Memory

Overview

The Memory storage engine is designed for non-transactional applications needing high-speed access to data that is not persistent. Data stored in the Memory engine disappears when the MySQL server is shut down (although the definition of tables using the memory engine is kept intact). As its name implies, the Memory engine keeps 100% of its data in memory.

Target Applications

- Applications requiring high-speed lookup tables
- Web / Web 2.0
- Web session management (shopping carts, etc.)
- Data warehouses / Data marts

Key Industries Utilizing

- Defense
- eCommerce
- Education
- Energy
- Healthcare
- Manufacturing
- Media
- Retail
- Technology
- Telecom
- Travel

Pros and Cons

Pros
Extremely fast read and write operations
Support for B-tree and Hash indexes
Main memory database management

Cons
Data is not persistent between server shutdown's

No transactional support
No support for foreign keys
No full-text or GIS index support
Table level locking
Cannot store BLOB or Text data

Further Reading

MySQL Manual: http://dev.mysql.com/doc/refman/5.1/en/memory-storage-engine.html.
MySQL Forums: http://forums.mysql.com/list.php?92.

Merge

Overview

The Merge storage engine is a container for grouping two or more MyISAM tables together to form one logical table. It is designed for applications that do not have transactional or crash-recovery needs, but instead require efficient management of large datasets.

Target Applications

- Data warehouses / data marts
- Web / Web 2.0 applications

Key Industries Utilizing

- Defense
- eCommerce
- Education
- Energy
- Healthcare
- Manufacturing
- Media
- Retail
- Technology
- Telecom
- Travel

Pros and Cons

Pros
No practical storage limits
Low storage cost (efficient storage handling)
Support for B-Tree and GIS indexes
Very fast insert performance
Fast query performance
Easy management of subset data
Custom placement of files/data on disk

Cons
No transactional support

Table-level locking
No crash recovery
Blocking online backup
No support for foreign keys
No Fulltext indexes
Slower reads on indexes than pure MyISAM

Further Reading

MySQL Manual: http://dev.mysql.com/doc/refman/5.1/en/merge-storage-engine.html.
MySQL Forums: http://forums.mysql.com/list.php?93.

MyISAM

Overview

The MyISAM storage engine is designed for applications that do not have transactional or crash-recovery needs, but instead require fast loading and retrieval of data. The MyISAM storage engine is the default storage engine for the MySQL server

Target Applications

- Web / Web 2.0 applications
- Data warehouses / data marts
- Embedded applications
- Hardware devices
- Blogs
- Logging
- Authorization/Authentication

Key Industries Utilizing

- Defense
- eCommerce
- Education
- Energy
- Healthcare
- Manufacturing
- Media
- Retail
- Technology
- Telecom
- Travel
- Web / Web 2.0

Pros and Cons

Pros
No practical storage limits; table sizes constrained only by file size limits
Low storage cost (efficient storage handling)
Support for B-Tree, FullText, and GIS indexes

Very fast insert performance
Very fast query performance
Maintains accurate count of number of rows stored in table (SELECT COUNT(*) very fast).
Support for prefix-length index keys

Cons
No transactional support
Table-level locking
No crash recovery
Blocking online backup (due to be resolved in MySQL's online backup coming in 2010)
No support for foreign keys

Further Reading

MySQL Manual: http://dev.mysql.com/doc/refman/5.1/en/myisam-storage-engine.html.
MySQL Forums: http://forums.mysql.com/list.php?21.

NDB/MySQL Cluster

Overview

The NDB/Cluster storage engine is designed for transactional applications requiring extreme degrees of high-availability and crash-recovery support. NDB utilizes built-in clustering and synchronous replication to provide up to 99.999% availability, in addition to main-memory database capabilities that offer rapid response times for both query and data manipulation operations.

Target Applications

- Telecommunication systems
- Web / Web 2.0
- Web session management (shopping carts, etc.)
- Embedded transactional applications
- Search engines
- Trading and investment management systems Exchange systems
- Instant gaming and betting applications

Key Industries Utilizing

- Telecommunications
- Network
- eCommerce
- Trading and Finance
- Retail

Pros and Cons

Pros

ACID-transaction compliance
Shared nothing architecture
Built-in clustered database support
Fast and automatic failover
Support for crash recovery
Main memory database management
Optional disk support for tables
Support for B-tree and hash indexes
Unlimited row-level concurrency
Extremely fast query speed (especially for primary key lookups)
Direct access API's for high performance

Cons
Indexes are main-memory only
No support for foreign keys
No full-text or GIS index support

Further Reading

MySQL Manual: http://dev.mysql.com/doc/refman/5.1/en/mysql-cluster.html.
MySQL Forum: http://forums.mysql.com/list.php?25.
Cluster API Guide: http://dev.mysql.com/doc/ndbapi/en/index.html.

Third-Party Storage Engines

This section describes the storage engines available in the MySQL server that are supplied by third party organizations or the MySQL community.

For further information, see http://solutions.mysql.com/engines.html.

Transactional Engines

The following engines support the use of typical OLTP applications.

IBMDB2I

Overview

The IBMDB2I Storage Engine allows users to create MySQL tables with a DB2 format type on the IBM i5 platform. As opposed to other storage engines, the IBMDB2I storage engine is more of an adaptor that allows access to DB2 tables on i5 platforms through a MySQL-based interface. the IBMDB2I Storage Engine stores the data in native DB2 format. Data in the DB2 format can be accessed either via MySQL or native i5 applications.

Data can be easily converted from a MySQL native storage engine into a DB2 data store. This makes it more convenient for other IBM i5 applications to access and exchange data with MySQL tables in a simple manner through DB2 for i5 native and SQL interfaces.

Further Reading

Overview of IBMDB2I engine: http://solutions.mysql.com/engines/ibm_db2_storage_engine.html
IBMDB2I FAQ: http://solutions.mysql.com/engines/ibm_db2faq.html

InnoDB

Overview

The InnoDB storage engine is produced by Innobase Oy, which is owned by Oracle Corporation. The InnoDB storage engine is designed to handle transactional applications that require crash recovery, automatically-enforced data integrity, high levels of user concurrency, and good response times. InnoDB is fully open source / free software and available under the GPL license for anyone to use, modify and redistribute.

A new plug-in version of the InnoDB storage engine was released in April 2008, which features data compression, new row formats, and faster index creation. The plug-in is available for download from the Innobase and MySQL forge web sites.

Target Applications

- Web / Web 2.0 transactional applications
- eCommerce transactional systems

- Financial systems
- Health care applications
- Retail systems
- Embedded transactional applications

Key Industries Utilizing

- Defense
- eCommerce
- Education
- Energy
- Finance/Banking
- Healthcare
- Manufacturing
- Media
- Retail
- Technology
- Telecom
- Travel
- Web / Web 2.0

Pros and Cons

Pros
ACID-transaction compliance
Support for crash recovery
High storage limit (64TB per tablespace)
Unlimited row-level locking
Support for foreign keys
MVCC support (readers don't block writers and vice versa)
Clustered, B-Tree index support (also on-demand Hash indexes)
MySQL-supplied online, non-blocking backup
Advanced memory cache mechanisms

Cons
No full-text or GIS index support
Faster online backup utility an add-on cost option

Further Reading

MySQL Manual: http://dev.mysql.com/doc/refman/5.1/en/innodb.html
MySQL Forums: http://forums.mysql.com/list.php?22
Web Site: http://www.innodb.com/wp/products/innodb/

MariaDB

Overview

MariaDB is produced by Monty Program AB and was created as a fork of the MyISAM storage engine, with the current alpha release providing MyISAM benefits plus crash recovery. Future releases of MariaDB are planned to include transaction support and data warehousing enhancements.

Target Applications

- Web / Web 2.0 applications
- Data warehouses / data marts
- Embedded applications
- Hardware devices
- Blogs
- Logging
- Authorization/Authentication

Key Industries Utilizing

- In alpha mode currently; community testing

Pros and Cons

Pros
All the benefits of MyISAM (see above section on MyISAM)
Crash recovery

Cons
Con's of MyISAM (see above section on MyISAM)

Further Reading

MySQL Forums: http://forums.mysql.com/list.php?157.

PBXT

Overview

The PrimeBase XT (PBXT) storage engine is produced by SNAP Innovation GmbH. The PBXT storage engine is designed for transactional applications that need crash recovery support and must handle tables with large and varying row-sizes (including BLOBs); disk space is efficiently managed using non-blocking, asynchronous defragmentation and a file-per-table storage scheme. The PBXT storage engine is currently in Release Candidate stage (June 2009).

Target Applications

- Web / Web 2.0 applications
- eCommerce transactional systems
- Financial systems
- Health care applications
- Retail systems
- Print / Pre-press / Publishing systems

Pros and Cons

Pros
ACID-transaction compliance
Support for crash recovery
High storage limit, 256TB per table
Unlimited row-level locking
Online backup using snapshot read
MVCC support (readers don't block writers and vice versa)
Support for foreign keys / server-enforced referential integrity
Write-once / write-through (data written directly to database)
Advanced caching and disk management (defragmentation / file-per-table).
Instant commit/rollback/recovery (cleanup done asynchronously)

Cons
No Full-text or GIS index support
Only supports row-based replication
Currently no support for Check and Repair table (planned for GA)
Currently no support for XA and Savepoints (planned for v1.5)
Currently no support for Analyze table (planned for v1.5)

Further Reading

Dev Zone Article: http://dev.mysql.com/tech-resources/articles/pbxt-storage-engine.html
Web Site: http://www.primebase.org/

ScaleDB

Overview

The ScaleDB storage engine is produced by ScaleDB Inc, and is designed for transactional applications that handle large data volume, requiring high-availability, and scalability. ScaleDB is based on a shared-disk clustering architecture that enables multiple nodes in a cluster to operate over a single physical database. ScaleDB is useful for large web applications since multiple MySQL servers can share the same physical data without the need to partition the data. ScaleDB is also appropriate for cloud computing, because servers can be added and removed dynamically according to changing needs. ScaleDB is not open source and the engine is in beta as of this writing.

Target Applications

- Web / Web 2.0 transactional applications
- Large Web applications
- eCommerce transactional systems
- Financial systems
- Health care applications
- Retail systems

Key Industries Utilizing

- eCommerce
- Web / Web 2.0

Pros and Cons

Pros
ACID-transaction compliance
Shared-disk architecture eliminates the need to partition data
Dynamic scalability enables adding/removing nodes on the fly, good for cloud computing
Support for crush recovery of any node in the cluster
Master-only cluster eliminates slave replication and promotion
Row-level locking
Support for foreign keys
Support for referential integrity
Multi-Table Index (provides the functionality of materialized views)
Highly compressed Indexes
Supports read committed

Cons
No full-text or GIS index support
No MVCC (planned for future release)

Further Reading

ScaleDB Website: http://www.scaledb.com
ScaleDB Overview: http://scaledb.com/pdfs/Overview_Datasheet.pdf
Comparing Shared-Disk and Shared Nothing: http://scaledb.com/pdfs/WP_SDvSN.pdf

ScaleDB amd Cloud Computing: http://scaledb.com/pdfs/CloudComputingDaaS.pdf
Multi-Table Index Overview: http://scaledb.com/pdfs/MTI_Datasheet.pdf
Multi-Table Index White Paper: http://scaledb.com/pdfs/MultiTableTechnologyOverview.pdf

XtraDB

Overview

XtraDB is produced by Percona and is a clone of the InnoDB storage engine, but also includes additional features such as support for multiple rollback segments, additional performance diagnostics and speed enhancements, and more.

Target Applications

- Web / Web 2.0 transactional applications
- eCommerce transactional systems
- Financial systems
- Health care applications
- Retail systems
- Embedded transactional applications

Key Industries Utilizing

- Technology
- Web / Web 2.0

Pros and Cons

Pros
ACID-transaction compliance
Support for crash recovery
High storage limit (64TB per tablespace)
Unlimited row-level locking
Support for foreign keys
MVCC support (readers don't block writers and vice versa)
Clustered, B-Tree index support (also on-demand Hash indexes)
MySQL-supplied online, non-blocking backup
Advanced memory cache mechanisms
Online backup utility offered free of charge (nearly GA as of this writing)

Cons
No full-text or GIS index support

Further Reading

XtraDB Home: http://www.percona.com/docs/wiki/Percona-XtraDB:start
XtraDB on Launchpad: https://launchpad.net/percona-xtradb

Data Warehouse / BI Storage Engines

Calpont

Overview

Calpont has developed a multi-threaded, scale out, distributed processing engine specifically built for analytics and data warehousing. Calpont can be deployed on single server or deployed against a server-grid architecture to enable its scale-out capabilities, with the engine being capable of true massive parallel processing (MPP) actions. Calpont employs a column-oriented architecture that significantly reduces I/O for analytic queries and has been developed to provide out of the box performance with very low administrative overhead.

Target Applications

- Data warehouses / data marts
- Web / Web 2.0 applications
- Historical data analysis and reporting

Pros and Cons

Pros
No practical storage limits
Scale-out architecture enables high performance computing
Scale-out architecture enables customers to easily grow with data and user requirements; true MPP capabilities for MySQL
Extent Map capabilities provides accelerated access to data without indexes
Very fast bulk load performance
Very fast query performance
Architecture enables the user to scale both physical and logical I/O
Column-oriented data store that significantly reduces I/O for analytic queries
MVCC-compliant transaction support
Shared nothing or shared-disk deployment capability

Cons
No storage compression (coming in a future release)
Transactions are currently serial in nature, future release to enable concurrent transactional behavior
No support for foreign keys or other referential integrity checks
No replication support (coming in a future release)

InfoBright – BrightHouse

Overview

The BrightHouse storage engine is produced by Infobright Inc. The BrightHouse storage engine is designed for large scale data warehouses (TB range) – both traditional and archive/historical – with a support range up to about 30TB on a single server. BrightHouse utilizes a column-oriented database structure and compresses data at an average compression ratio of 10:1 (including all additional structures).

Infobright offers a community edition of their server (ICE – Infobright Community Edition), which is open source, and an enterprise edition (IEE), which is proprietary.

Target Applications

- Web / Web 2.0 applications
- Data warehouses / data marts
- Historical data stores with Analytical query

Key Industries Utilizing

- eCommerce
- Media
- Retail

Pros and Cons

Pros
No practical storage limits
Extremely Low storage cost
Knowledge Grid replaces need for indices
Separate optimizer, which outpaces standard MySQL optimizer in most cases
Very fast load performance
Very fast query performance
Advanced memory management
Revised Optimizer using knowledge grid minimizing decompression of data
Column-oriented data store
No need for partitioning (all handled by Knowledge Grid)

Cons
No transactional support
Table-level locking
No crash recovery
No support for foreign keys or other referential integrity checks
No replication support
Currently limited to supporting about 30 concurrent connections (more concurrency support coming in future release)
A query can only make use of one CPU/core currently

Further Reading

Dev Zone Article: http://dev.mysql.com/tech-resources/articles/datawarehousing_mysql_infobright.html
Community Web Site: www.infobright.org
Company Web Site: www.infobright.com

Kickfire

Overview

The Kickfire storage engine is produced by Kickfire Inc. The Kickfire storage engine is designed for supporting data warehouses up to about 3TB (with current 1.0 product). Kickfire is an appliance-based solution and is not open source. The solution's key differentiator is a SQL chip that accelerates SQL performance much like a graphics card does with graphics on a display device. Kickfire also utilizes a column-oriented database structure and compresses data at an average compression ratio of 2-3:1.

Kickfire offers a number of different appliances depending on database size and application needs.

Target Applications

- Web / Web 2.0 applications
- Standard data warehouse / mart use cases
- Historical data analysis and reporting

Key Industries Utilizing

- E-commerce/retail
- High-tech/SaaS
- Marketing analytics
- Web / Web 2.0
- Media

Pros and Cons

Pros
Terabyte warehouse support
Low storage cost with compression
Very fast load performance
Fast query performance
Built-in hardware parallelism with SQL chip
Transactional in nature (full ACID compliance); crash recovery supported
Standard indexing supported
Separate optimizer, which runs BI queries faster than normal MySQL optimizer
Column-oriented data store
Support for foreign keys/referential integrity constraints
High user concurrency
On/offline backup supported
Concurrent query with no blocking during load
High availability options

Cons
Table-level locking
No replication support
Limited to about 3TB in version 1.0 product

Further Reading

Company Web Site: www.kickfire.com
TPC-H Benchmark Results: http://www.tpc.org/tpch/results/tpch_price_perf_results.asp

TokuDB

Overview

TokuDB's Fractal Tree™ indexing technology purports to provide high performance insert/update/delete operations on large tables with indexes, enabling fast queries. TokuDB is scalable, and reports to provide linear, predictable performance on very large tables. TokuDB runs on commodity servers, and includes compression, substantially reducing disk footprint. TokuDB v2.0 with crash safety is generally available as of this writing, and a fully ACID compliant version with fast recovery will be released later in 2009.

Target Applications

- Web / Web 2.0 applications
- Data warehouses / data marts
- Embedded applications
- Hardware devices
- Blogs
- Logging
- Authorization/Authentication

Key Industries Utilizing

- Defense
- eCommerce
- Media
- Retail
- Technology
- Telecom
- Travel
- Web / Web 2.0

Pros and Cons

Pros
Fractal Tree indexes enable fast indexed insert/update/delete and fast queries
Predictable, linear performance scaling on large tables. May eliminate the need for sharding or partitioning
Compression: 2x–12x smaller disk footprint
No aging or fragmentation – consistent performance without dump/re-load
Unlimited, pessimistic row-level locking
ACID-transaction compliance (available later in 2009)

Support for fast crash recovery (available later in 2009)
Fast single threaded insert/update/delete – reduces slave replication lag
No practical storage limits; table sizes constrained only by file size limits
Online backup

Cons
No Full-text or GIS index support
No MVCC capabilities
Currently no support for Check and Repair table (tables are not corrupted on crashes)
Currently no support for XA and Savepoints

Further Reading

Percona Review: http://www.mysqlperformanceblog.com/2009/04/28/detailed-review-of-tokutek-storage-engine/
Web Site: http://tokutek.com

Common Questions and Answers

This section covers some commonly asked questions about MySQL storage engines.

Question: What are the most popular MySQL storage engines?
Answer: MyISAM continues to be the most used storage engine, followed very closely by InnoDB, Memory, MySQL Cluster, and Archive.

Question: How do I switch between different storage engines?
Answer: Any existing MySQL table can be migrated to a new storage engine via the `ALTER TABLE … ENGINE=` command.

Question: Can I disable the use of various storage engines?
Answer: Yes. Most engines can be skipped during the compilation of the MySQL binary, or you can use –skip flags in the `my.cnf` configuration file to disable an engine's use. Beginning in MySQL 5.1, you can also use the new `INSTALL PLUGIN/UNINSTALL PLUGIN` commands do introduce or remove storage engines and/or new versions of existing storage engines.

Question: Can I use different storage engines in the same database/application?
Answer: Yes.

Question: What backup/recovery considerations are there for databases with multiple storage engines?
Answer: The `mysqldump` backup utility can perform online/non-blocking backups (DML only) for all transactional storage engines (except NDB, which has its own special backup program). Non-transactional storage engines can be backed up online also, except that DML activity is suspended until the backup operation completes.

Question: What are the main differences between MySQL storage engines?
Answer: See the section entitled "Key Storage Engine Differentiation Points" in this document.

Question: Are all external storage engines true engines in that they simply plug into a standard MySQL build and work?
Answer: No – engines such as Infobright and Calpont have made software additions to the non-storage engine layers of MySQL that require a separate download of their software that includes a special MySQL binary. Kickfire works in the same vein, although it is an appliance-based solution.

MySQL Transactional Storage Engine Feature Comparison Grid

This section compares the feature sets of the various MySQL transactional storage engines.

Feature	InnoDB	Falcon	Cluster	XtraDB	PBXT	ScaleDB
General						
Crash Recovery	☑	☑	☑	☑	☑	☑
Online parameter support			☑			
Foreign Key support	☑			☑	☑	☑
Materialized views						☑
Memory						
Data and index caches	☑	☑	☑	☑	☑	☑
Data dictionary caches	☑	☑		☑	☑	☑
Transaction management in memory		☑	☑			
Main memory tables/indexes			☑			
Storage						
Configurable Page Sizes		☑				
Tablespace containers	☑	☑	☑	☑		☑
Assign objects to specific files / tablespaces		☑				
Configurable file size limits	☑	☑	☑	☑		☑
Data file autoextend	☑	☑		☑	☑	☑
Redo file autoextend		☑				
Undo file autoextend	☑	☑		☑	☑	☑
Compressed data support	☑	☑				
Undo Log	☑	☑		☑	☑	☑
Multiple Undo Logs				☑		☑
Redo Log	☑	☑	☑	☑	☑	☑
External file access/association						
Shared disk clustering						☑
Concurrency Support						
Row-level locks	☑	☑	☑	☑	☑	☑
Row to table escalation locks	☑			☑		☑
Explicit table locks	☑			☑	☑	☑
Page level locks						
MVCC support	☑	☑		☑	☑	
Isolation levels					☑	
• Repeatable Read	☑	☑		☑	☑	
• Serializable	☑			☑	☑	☑
• Read Uncommitted	☑			☑		☑
• Read Committed	☑	☑	☑	☑	☑	☑
Deadlock detection	☑	☑		☑	☑	☑
Configurable Lock timeouts	☑	☑	☑	☑		☑
Transaction Support						
ACID level transactions	☑	☑	☑	☑	☑	☑
XA support	☑	☑		☑		☑
Small-medium sized transaction	☑	☑	☑	☑	☑	☑

support (< 1GB)						
Large (GB) sized transaction support	☑	☑		☑	☑	☑
Configurable checkpoints	☑		☑	☑		☑
Implicit savepoints	☑	☑		☑		☑
Group commit	☑	☑		☑		
Two-phase commit	☑		☑	☑		☑
I/O Support						
Asynchronous I/O						
Prefetch Mechanism	☑			☑		
Configurable read/write worker count						
Configurable Log flush setting	☑			☑		
Write-once (bypasses log)					☑	
Index Support						
B-Tree indexes	☑	☑	T-Trees	☑	☑	
Hash indexes			☑			
Clustered indexes	☑			☑		
Full-Text Indexes						
R-Tree (spatial)						
Balanced patricia tie indexes						☑
Utility Support						
Online Backup	☑	☑	☑	☑	☑	☑
Defragment/Optimize table	☑	☑		☑	☑	☑
Analyze table	☑	☑		☑		☑
Repair table					☑	
Check table					☑	
Replication	Statement / Row	Row	Statement / Row	Statement / Row	Row	NA

MySQL Third Party Data Warehouse Storage Engine Feature Comparison Grid

This section compares the feature sets of the various third-party MySQL data warehouse storage engines.

Feature	InfoBright	Kickfire	Tokutek	Calpont
General				
Crash Recovery		☑	☑	☑
Online parameter support	☑			☑
Foreign Key support		☑		
Transaction support (ACID)		☑	☑	☑
Column-oriented architecture	☑	☑		☑
Row-oriented architecture			☑	
Group commit				☑
Configurable lock timeouts				☑
High availability options		☑		☑
Additional optimizer	☑	☑		☑
Appliance-based solution		☑		
Software-based solution	☑		☑	☑
Hardware acceleration		☑		
MPP (massive parallel processing) capable				☑
Data Loading				
Character/binary bulk loading	☑	☑	☑	☑
Incremental loading		☑	☑	
DML support	☑*	☑	☑	☑
Non-blocking query while load		☑	☑	
Memory				
Data cache	☑	☑	☑	☑
Index cache	NA	☑	☑	NA
Data dictionary caches	☑	☑		☑
Main memory tables/indexes				☑
Storage				
Configurable Page Sizes			☑	
Assign objects to specific files		☑		
Tablespace containers		☑		☑
Configurable file size limits				☑
Data file autoextend		☑	☑	☑
Undo Log		☑	☑	☑
Redo Log		☑	☑	☑
Undo Log autoextend		☑	☑	
Compression	☑	☑	☑	
External file access/association		☑		
Concurrency Support				
Row-level locks			☑	
Row to range escalation locks			☑	

Row to table escalation locks			☑	
Table level locks	☑	☑		
Page level locks				☑
MVCC support				☑
Isolation levels				
• Repeatable Read				
• Serializable		☑	☑	
• Read Uncommitted			☑	
• Read Committed	☑			☑
Deadlock detection	☑	☑	☑	☑
Configurable Lock timeouts				☑
Concurrency support > 100		☑	☑	☑
I/O Support				
Asynchronous I/O			☑	☑
Prefetch Mechanism			☑	☑
Configurable read/write worker count				☑
Configurable Log flush setting		☑	☑	
Index Support				
No requirement for indexing (alternative technology)	☑			☑
B-Tree indexes		☑		
Hash indexes				
Clustered indexes			☑	
Secondary Clustered indexes			☑	
Fractal-Tree indexes			☑	
Full-Text Indexes				
Sparse Indexes				
R-Tree (spatial)				
N-Tree Indexes (aggregate info pre-calculated and stored)				
Utility Support				
Defragment/Optimize table		☑		
Online (blocking DML) backup; Online for query		☑	☑	☑
Analyze table	☑	☑		
Repair table				
Check table				
Replication			☑	
Explain plan	☑	☑	☑	☑
Data migration utility		☑		
Software upgrade utility		☑		
Additional monitoring diagnostics		☑		
Backup & restore via snapshots (built-in utility)		☑		
Active system monitoring / notifications		☑		

* in enterprise edition

MySQL Non-Transactional Storage Engine Feature Comparison Grid

This section compares the feature sets of the various MySQL non-transactional storage engines.

Feature	MyISAM	Memory	Merge	Archive	CSV	Blackhole	MariaDB
General							
Crash Recovery							☑
Online parameter support	☑		☑				☑
Foreign Key support							
Memory							
Data cache		☑					
Index cache	☑	☑	☑				☑
Data dictionary caches							
Main memory tables/indexes		☑					
Storage							
Configurable Page Sizes	☑		☑				☑
Assign objects to specific files	☑		☑	☑	☑		☑
Tablespace containers							
Configurable file size limits	☑		☑				☑
Data file autoextend	☑		☑	☑			☑
Undo Log							
Redo Log							
Undo Log autoextend							
Compressed row support	☑			☑			☑
External file access/association					☑		
Concurrency Support							
Row-level locks				☑			
Row to table escalation locks							
Table level locks	☑	☑	☑		☑		☑

Page level locks							
MVCC support				☑			
Isolation levels							
• Repeatable Read							
• Serializable							
• Read Uncommitted							
• Read Committed							
Deadlock detection							
Configurable Lock timeouts							
I/O Support							
Asynchronous I/O							
Prefetch Mechanism							
Configurable read/write worker count							
Configurable Log flush setting	NA	NA	NA	NA	NA	NA	
Write-once (bypasses log)	NA	NA	NA	NA	NA	NA	
Index Support							
B-Tree indexes	☑	☑	☑				☑
Hash indexes		☑					
Clustered indexes							
Full-Text Indexes	☑						☑
Sparse Indexes				☑			
R-Tree (spatial)	☑						☑
N-Tree Indexes (aggregate info pre-calculated and stored)							
Utility Support							
Defragment/Optimize table	☑		☑	☑			☑
Online (blocking) backup	☑		☑	☑	☑		☑
Analyze table	☑		☑				☑
Repair table	☑		☑	☑			☑
Check table	☑		☑	☑	☑		☑
Replication	Statement / Row	Statement / Row	Statement / Row	Statement / Row	Statement / Row	Statement / Row	Statement / Row