MySQL Storage Engines Which Do You Use?

April, 25, 2017

Sveta Smirnova

OPEN SOURCE DATABASE CONFERENCE

PERCONA LIVE

Sveta Smirnova

- MySQL Support engineer
- Author of
  - MySQL Troubleshooting
  - JSON UDF functions
  - FILTER clause for MySQL
- Speaker
  - Percona Live, OOW, Fosdem, DevConf, HighLoad...

From Type to Engine

- MySQL $<$ 3.23 had only engine: ISAM

From Type to Engine

- MySQL < 3.23 had only engine: ISAM
- Version 3.23 introduced table types

```
mysql> CREATE TABLE plmce(
    -> id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    -> name VARCHAR(100)
    -> ) TYPE = MyISAM;
Query OK, 0 rows affected, 1 warning (0.10 sec)
```

From Type to Engine

- MySQL $<$ 3.23 had only engine: ISAM
- Version 3.23 introduced table types
- In year 2003 term ”Type” was deprecated

From Type to Engine

- MySQL $<$ 3.23 had only engine: ISAM
- Version 3.23 introduced table types
- In year 2003 term ”Type” was deprecated
- Engines were built-in into server

From Type to Engine

- MySQL $<$ 3.23 had only engine: ISAM
- Version 3.23 introduced table types
- In year 2003 term ”Type” was deprecated
- Engines were built-in into server
- Nobody could deliver engine independently

From Type to Engine

- MySQL $<$ 3.23 had only engine: ISAM
- Version 3.23 introduced table types
- In year 2003 term ”Type” was deprecated
- Engines were built-in into server
- Nobody could deliver engine independently
- Version 5.1 changed everything
  - Pluggable storage engine API was introduced

InnoDB

- Part of MySQL since version 3.23.24
  - Released at March, 10, 2001

InnoDB

- Part of MySQL since version 3.23.24
- Created by Innobase OY
  - Acquired by Oracle in 2005

InnoDB

- Part of MySQL since version 3.23.24
- Created by Innobase OY
- Major changes in 5.1
  - New tablespace format
  - Dynamic loading
  - Online index creation
  - ...
  - Released as a plugin

InnoDB

- Part of MySQL since version 3.23.24
- Created by Innobase OY
- Major changes in 5.1
  - Two versions in 5.1.38 - 5.1.73
    - Built-in
    - Pluggable

Pioneers

- Many others started own storage engines

Pioneers

- Many others started own storage engines
- Most notable
  - Tokutek
  - Primebase

Pioneers

- Many others started own storage engines
- Most notable
  - Tokutek
    - TokuDB
    - Write-scale
    - Acquired by Percona in 2015
  - Primebase

Pioneers

- Many others started own storage engines
- Most notable
  - Tokutek
    - TokuDB
    - Write-scale
    - Acquired by Percona in 2015
  - Primebase
    - PBXT
    - Better BLOB handling technology
    - Engine not supported now

In the Official Distribution

- Built-in engines were converted into plugins
- Some old engines were removed
  - BerkeleyDB
  - ISAM

Community

- Number of engine grows
- They can
  - Shard: Spider
  - Use any source of data: CONNECT
  - Connect to foreign sources: FederatedX
  - Perform full text search: SphinxSE
  - More
- MariaDB includes most of the engines

Simple and Complex Engines

- All engines
  - Store data
  - Retrieve data

Simple and Complex Engines

- All engines
- Simple engines
  - Use built-ins for all other job
    - Locking
    - Transactions support
    - Diagnostic

Simple and Complex Engines

- All engines
- Simple engines
- Complex engines
  - Implement
    - Own locking model
    - Transactions
    - Diagnostic
    - Log files
    - More

Three Majors: InnoDB, TokuDB, MyRocks

- All three
  - Transactional
  - Row-level locking
  - MVCC
  - ACID
  - XA
  - Automatic crash recovery

Three Majors: InnoDB, TokuDB, MyRocks

- All three
- InnoDB
  - Universal
  - Default since 5.5.5

Three Majors: InnoDB, TokuDB, MyRocks

- All three
- InnoDB
- TokuDB
  - Write optimized
  - Fine compression support
  - Best for big data

Three Majors: InnoDB, TokuDB, MyRocks

- All three
- InnoDB
- TokuDB
- MyRocks
  - Write and space optimized
  - Great compression support
  - Best for SSD

InnoDB

- B-Tree
  - Extremely fast read access
  - Needs to be re-balanced on write

InnoDB

- B-Tree
- Reach features set
  - Foreign keys
  - Locks at the engine level
    - Row
    - Gap
    - Auto-increment
    - Table
  - Compression
  - Extended crash recovery

B-Tree

- Initial Data

```
      11
     /  \
   10    12
   /      \
  2       24
 / \      / \
1   9    23 25
```

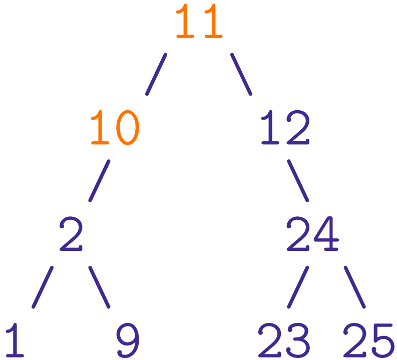
B-Tree

- SELECT 11

```
      11
     /  \
   10     12
   /       \
  2         24
 / \        / \
1   9     23  25
```

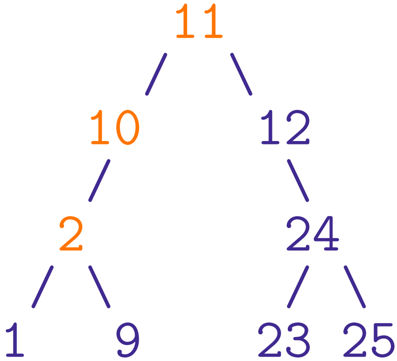
B-Tree

- SELECT 10



B-Tree

- SELECT 2



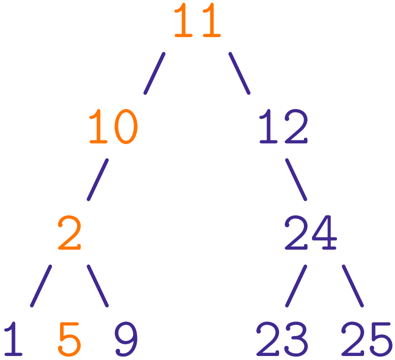
B-Tree

- SELECT 9

```
        11
       /  \
     10    12
     /       \
    2         24
   / \       /  \
  1   9     23  25
```

B-Tree

- INSERT 5



11

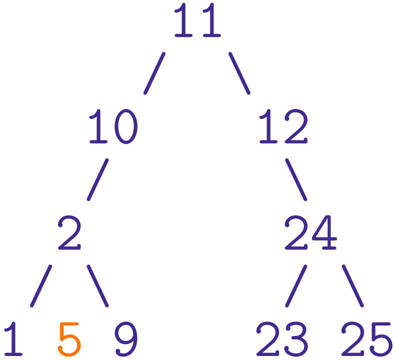
10 12

2 24

1 5 9 23 25

B-Tree

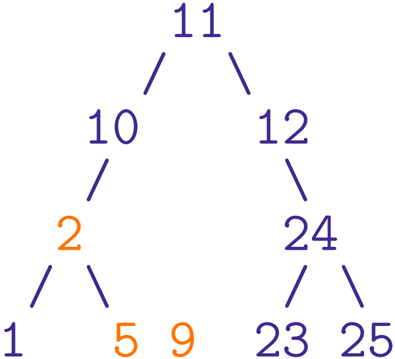
- INSERT 5



```
      11
     /  \
   10    12
   /      \
  2       24
 / \      / \
1 5 9    23 25
```

B-Tree

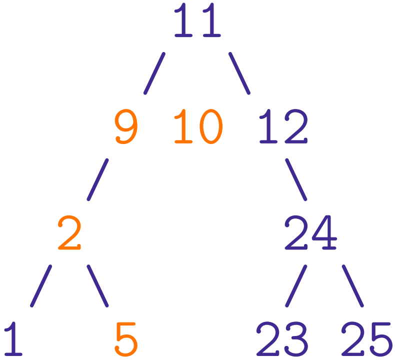
- INSERT 5



```
      11
     /  \
   10    12
   /      \
  2       24
 / \      / \
1   5 9  23 25
```

B-Tree

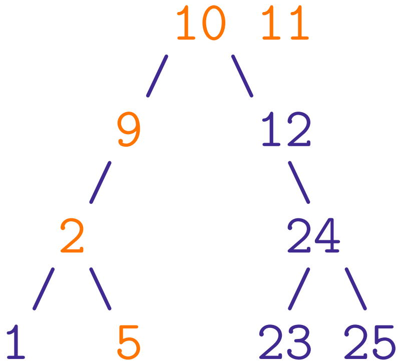
- INSERT 5



```
      11
     /  \
    9 10 12
   /      \
  2       24
 / \      / \
1   5    23 25
```

B-Tree

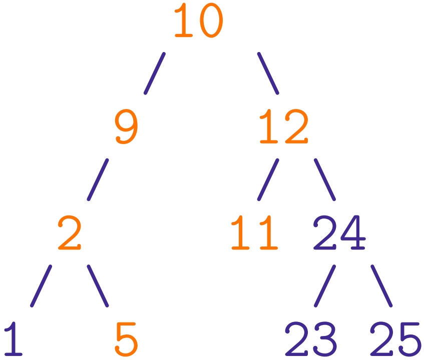
- INSERT 5



```
      10 11
     /  \
    9    12
   /      \
  2       24
 / \      / \
1   5    23 25
```

B-Tree

- INSERT 5



10

9 12

2 11 24

1 5 23 25

TokuDB

- Fractal Tree
  - Write optimized
  - All writes stored in buffers
  - Background thread flushes them
  - By default reads are slow

TokuDB

- Fractal Tree
- Optimizations for reads
  - Secondary Clustered Indexes
  - Read-free replication
  - No index fragmentation

TokuDB

- Fractal Tree
- Optimizations for reads
- Optimizations for writes
  - Fast inserts
  - Bulk loader
  - Compression

TokuDB

- Fractal Tree
- Optimizations for reads
- Optimizations for writes
- Other features and limitations
  - Reach set of locking diagnostic
  - No foreign key support
  - Crash recovery is limited if compare to InnoDB

MyRocks

- LSM Tree
  - Write and space optimized
  - All writes go to MemTable and WAL first
  - Data files are immutable
  - Compaction
  - Designed for small transactions

MyRocks

- LSM Tree
- Optimizations for reads
  - Bloom filter
  - ICP
  - No ”index dives”
  - Reverse column families
  - Read-free replication

MyRocks

- LSM Tree
- Optimizations for reads
- Optimizations for writes
  - Options for bulk operations
  - Compression

MyRocks

- LSM Tree
- Optimizations for reads
- Optimizations for writes
- Limitations
  - Two transaction isolation levels
    - READ COMMITTED
    - REPEATABLE READ

MyRocks

- LSM Tree
- Optimizations for reads
- Optimizations for writes
- Limitations
  - Two transaction isolation levels
  - No gap locking
  - No support for
    - Foreigh Keys
    - Full Text Keys
    - Spatial Keys

MyRocks

- LSM Tree
- Optimizations for reads
- Optimizations for writes
- Limitations
  - Two transaction isolation levels
  - No gap locking
  - No support for
  - Index only access for limited types
    - BINARY
    - Collation latin1_bin
    - Collation utf8_bin

MyRocks

- LSM Tree
- Optimizations for reads
- Optimizations for writes
- Limitations
  - Two transaction isolation levels
  - No gap locking
  - No support for
  - Index only access for limited types
  - Crash recovery is limited

Three Majors: comparison

	InnoDB	TokuDB	MyRocks
Reads	Fast	Slow	Slow
Writes	Comparatively Slow	Fast	Fast
Transaction Isolation Levels	4	4	2 (RR, RC)
Foreign Keys	Yes	Not	Not
Space Used	Plenty	Workload-depend	Small
Compression	Yes	Yes	Yes
Crash Recovery	Automatic, Tunable	Automatic	Automatic

Summary

- MySQL has many storage engines
- They provide a lot of flexibility
- Many extend server functionality
- Simple and complex engines exist
- InnoDB is feasible for most workloads
- TokuDB and MyRocks are best for write intensive applications

MySQL Storage Engine Sessions at Percona Live

- April, 25, 5:15pm, Balroom B
  - MyRocks Engineering: deploying a new MySQL storage engine to production
  - Herman Lee

MySQL Storage Engine Sessions at Percona Live

- April, 25, 5:15pm, Balroom B
- April, 26, 11:10am, Room 203
  - EVCache: Lowering Costs for a Low-Latency Cache with RocksDB
  - Scott Mansfield

MySQL Storage Engine Sessions at Percona Live

- April, 25, 5:15pm, Balroom B
- April, 26, 11:10am, Room 203
- April, 26, 2:00pm, Balroom C
  - MyRocks: best practice at Alibaba
  - dengcheng he, jiayi wang

MySQL Storage Engine Sessions at Percona Live

- April, 25, 5:15pm, Balroom B
- April, 26, 11:10am, Room 203
- April, 26, 2:00pm, Balroom C
- April, 26, 2:00pm, Room 203
  - Six New Important RocksDB Features And Planned Works
  - Siying Dong

MySQL Storage Engine Sessions at Percona Live

- April, 25, 5:15pm, Balroom B
- April, 26, 11:10am, Room 203
- April, 26, 2:00pm, Balroom C
- April, 26, 2:00pm, Room 203
- April, 26, 4:30pm, Ballroom E
  - Using SPIDER for sharding in production
  - Kayoko GOTO, Kentoku SHIBA

PERCONA

MySQL Storage Engine Sessions at Percona Live

- April, 25, 5:15pm, Balroom B
- April, 26, 11:10am, Room 203
- April, 26, 2:00pm, Balroom C
- April, 26, 2:00pm, Room 203
- April, 26, 4:30pm, Ballroom E
- April, 27, 11:00am, Ballroom E
  - MariaRocks: MyRocks in MariaDB
  - Sergei Petrunia

MySQL Storage Engine Sessions at Percona Live

- April, 25, 5:15pm, Balroom B
- April, 26, 11:10am, Room 203
- April, 26, 2:00pm, Balroom C
- April, 26, 2:00pm, Room 203
- April, 26, 4:30pm, Ballroom E
- April, 27, 11:00am, Ballroom E
- April, 27, 1:50pm, Ballroom A
  - TokuDB vs RocksDB
  - George Lorch, Vladislav Lesin

More informaiton

- InnoDB Documentation
- TokuDB Documentation
- MyRocks Wiki
- MySQL User manual on storage engines
- Experts MySQL
- MySQL 5.1 Plugin Development

Time For Questions

???

Thank you!

http://www.slideshare.net/SvetaSmirnova

https://twitter.com/svetsmirnova