MySQL 8.0
What's new in InnoDB

Bin Su
Oracle, MySQL
Mar 2018

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Agenda

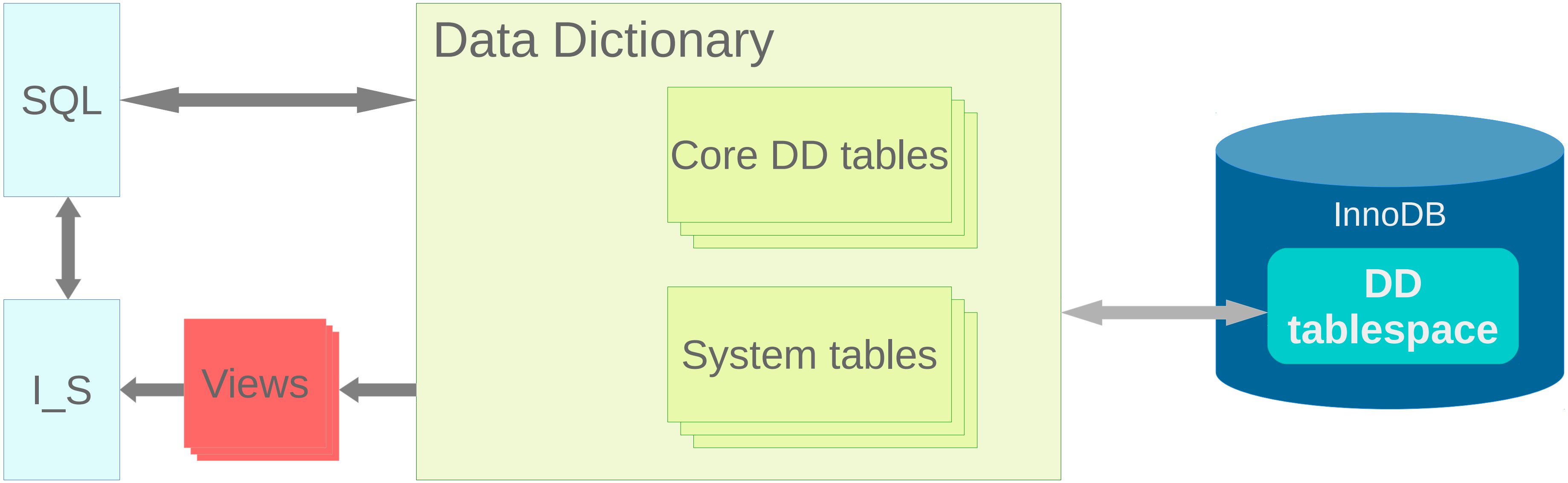
1. Feature improvements
2. Performance improvements
3. Summary

Feature improvements

Data Dictionary(DD)

- **Legacy – Multiple Data Dictionaries (.frm & InnoDB DD)**
  - changes not atomic
  - mismatch possible
  - concurrent access had to be managed
  - not crash proof

New Data Dictionary



New Data Dictionary

- Stored in new InnoDB system tables
- Single set of persisted metadata for all storage engines
- Control meta-data access using single locking mechanism
- No .frm files for temporary tables – meta-data in memory only
- Improves table spaces by removing .frm files
- Makes atomic DDL possible
- Required for transactional DDL (future)

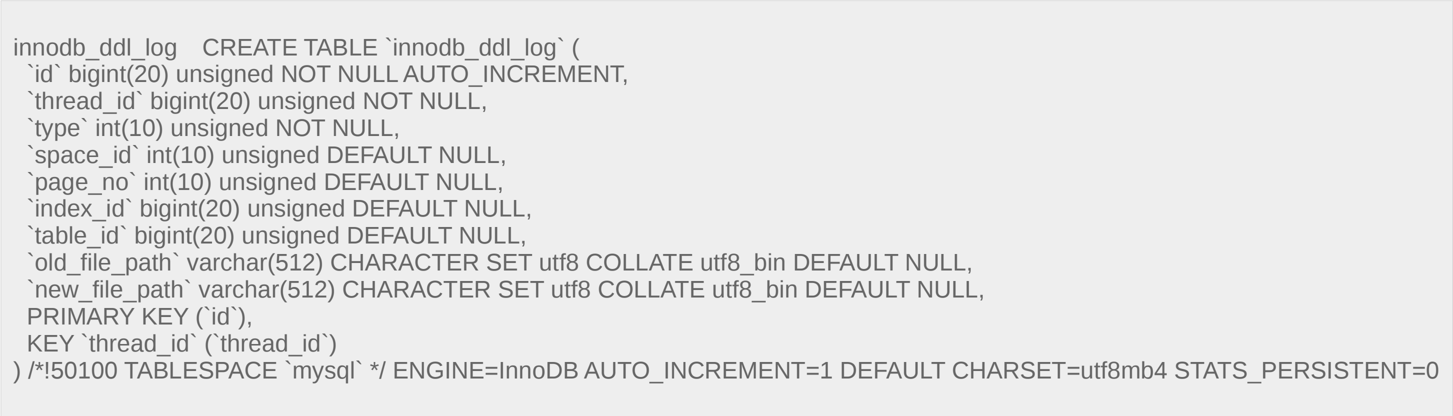
InnoDB and new DD

- **InnoDB serves 2 roles**
  - Data Dictionary store for all storage engines
  - implements atomic DDL (DDL_Log)
- **InnoDB gets metadata from the server not from system tables**
- **8.0 is transitional so InnoDB will have mutex for this reason but will go away in future releases**
- **–innodb-read-only semantics change**

Atomic DDL

- **Prerequisites**
  - Both atomic DD updates and data file updates
  - Storing DD metadata in transactional SE
  - Single DD transaction to update for DDL
  - Writing necessary SE DDL logs

DDL log table



```
innodb_ddl_log    CREATE TABLE `innodb_ddl_log` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `thread_id` bigint(20) unsigned NOT NULL,
  `type` int(10) unsigned NOT NULL,
  `space_id` int(10) unsigned DEFAULT NULL,
  `page_no` int(10) unsigned DEFAULT NULL,
  `index_id` bigint(20) unsigned DEFAULT NULL,
  `table_id` bigint(20) unsigned DEFAULT NULL,
  `old_file_path` varchar(512) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT NULL,
  `new_file_path` varchar(512) CHARACTER SET utf8 COLLATE utf8_bin DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `thread_id` (`thread_id`)
) /*!50100 TABLESPACE `mysql` */ ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8mb4 STATS_PERSISTENT=0
```

- One of the DD tables resides in DD tablespace
- A non-locking table
- One DDL will generate several logs
- Changes are persisted immediately, exempted from innodb_flush_log_at_trx_commit
- Once one DDL finished, DDL logs would be removed

SDI

- **SDI(Serialized Dictionary Information)**
  - Metadata stored in addition to the DD itself
  - To make the tablespace self descriptive
- **The SDI is stored in tablespace**
  - Stored in the form of B-tree
  - Compressed JSON format
  - Updated on DDL
- **ibd2sdi to extract SDI from tablespaces**
- **IMPORT/EXPORT**

Persistent AUTOINC

- Doesn’t reset to SELECT MAX(AUTOINC_COL) FROM T; on restart
- Probably the most requested feature since v3.x
- Bug 199 – created 27 March 2003

Native partitioning

- **Partitioning storage engine has been removed**
  - InnoDB supports native partitioning
  - Code reengineering according to new Data Dictionary
- **InnoDB supports 'ALTER ... PARTITION' natively**
  - ADD / DROP / COALESCE / REORGANIZE / REBUILD / EXCHANGE PARTITION
  - 'ALGORITHM = ..., LOCK = ...' is also supported now
  - Less logs would be written, so better performance
  - It paves the way for future improvement

Descending index

- **Support descending index on B-tree**
  - Backward index scan is noticeably slower
  - Provide the possibility to prevent file sort for ORDER BY

Examples:

```
CREATE TABLE t1(
  a INT, b INT,
  KEY a_desc_b_asc (a DESC, b));

-- Should use (forward) index scan
SELECT * FROM t1 ORDER BY a DESC;
SELECT * FROM t1 ORDER BY a DESC, b ASC;

-- Should use backward index scan
SELECT * FROM t1 ORDER BY a ASC;
SELECT * FROM t1 ORDER BY a ASC, b DESC;

-- Should use filesort, not index
SELECT * FROM t1 ORDER BY a ASC, b ASC;
SELECT * FROM t1 ORDER BY a DESC, b DESC;
```

Encryption

- **Encrypt redo and undo logs**
  - --innodb-redo-log-encrypt
  - --innodb-undo-log-encrypt
- **Encrypt shared tablespace (TBD)**

Undo tablespace

- **Ability to manage Undo tablespace**
  - Increase and decrease undo tablespaces
  - Default of 2 undo tablespaces required
  - [Info] InnoDB: Setting 'innodb_undo_tablespaces' to 0 is deprecated and will not be supported in a future release.
  - Undo truncate on by default
  - --innodb_rollback_segments is now per undo tablespace
  - --innodb_undo_logs and status Innodb_available_undo_logs are removed
- **Will provide more functions on undo tablespaces**

Information_schema

- **INNODB_CACHED_INDEXES**
  - Pages cached in the InnoDB buffer pool cache
- **INNODB_TABLESPACE_BRIEF**
  - The short and brief statistics for tablespaces

Memcache

- **Multiple get**
  - ‘get key1, key2, key3…’
  - Keys should be in the same table
  - Result set size has a limitation of 128M
- **Range search**
  - @>value, etc.
  - Compare symbols: <, <=, >, >=
  - Support only one range

Temp table

- **New In-Memory storage engine**
  - For internal use only
  - Not shared across connections
  - Lifetime limited to query life time
  - Limited size, bound by ram used, –temptable-max-ram

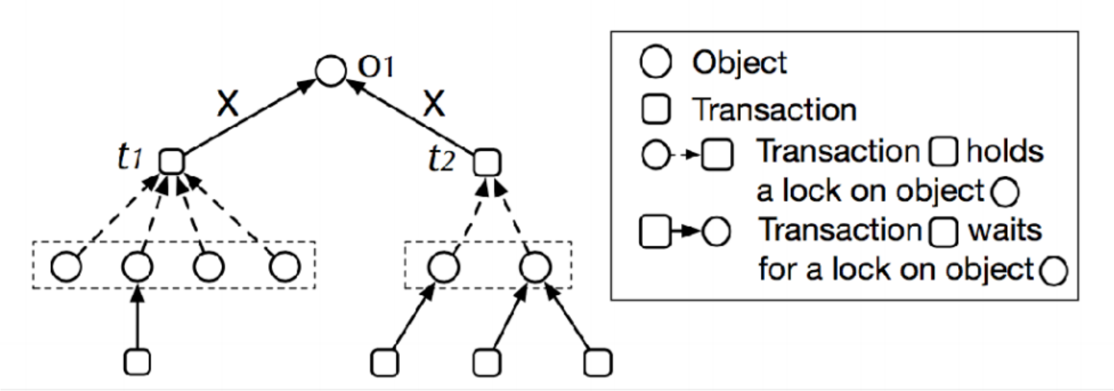
Dedicated server

- **--innodb-dedicated-server (default OFF)**
- **Sets default value basedon physical memory available**
- **Dynamically sets the following variables (UNIX only)**
  - –innodb-log-file-size
  - –innodb-buffer-pool-size
  - –innodb-flush-method

Performance improvements

CATS(Contention Aware Trans Scheduling)

- **Contributed by University of Michigan DB researchers**
- **Key idea is transaction has its own weight**
  - The weight is related to how it blocks other transactions


o1
X
X
t1
t2
○ Object
□ Transaction
○→□ Transaction □ holds a lock on object ○
□→○ Transaction □ waits for a lock on object ○

CATS(Contention Aware Trans Scheduling)

- **No configuration required**
- **Switches between FIFO and CATS automatically**
  - Threshold is >= 32 waiting threads
- **Help a lot for workload hitting row lock contentions**

Redo log re-design

- **Redo log was one big(if not biggest) bottleneck**
  - At first, even writing to disk blocks other worker threads
  - This gets fixed in both 5.7 and pre-8.0.5
- **However, this is still not enough!**
  - Contention on log buffer still exists
  - Worker threads are still busy writing redo logs
  - Users can’t tune the redo logs too much
  - Etc, etc.
- **With getting rid of these problems, performance will improve significantly!**

BLOB

- **Support partial fetch and update**
  - The internal LOB index
- **Plan to make streaming easier**

Buffer pool

- **Remove the buffer pool mutex (Percona contribution)**
  - Took a long time to fix problems in the contributed patch
  - QA team found lots of problems in edge cases
  - Foundation for more improvements in the future

Others

- **Cost Based Optimizer statistics**
  - Number of pages in RAM per index
- **--innodb_stats_include_delete_marked**
  - Include/exclude rows marked as deleted
- **Group records by table id when purging**
  - Reduces contention of dict_index_t::lock when multiple purge threads enabled
- **--innodb_detect_deadlock**
  - On high concurrent workloads deadlock detector becomes expensive so this turns it off and rely on rollback

Summary

Upgrade steps

- **Upgrade from 5.7 only**
  - Upgrade automatically
  - Make sure no crash and previous innodb_fast_shutdown is not 2
  - Create new DD tables in DD tablespace
  - Update all tables to new DD tables
  - Handle Undo tablespaces
  - Create SDI
  - Finally, InnoDB system tables get dropped
- **Downgrade is not allowed for now**
- **Incompatibility and crash can be handled**

Summary

- **Aim to easier use**
- **Aim to better performance**
- **Fix lots of bugs**
- **Easy upgrade**
- **Download 8.0-labs / 8.0-rc, and enjoy!**

Thank you!