MariaDB

Meet InnoDB 5.7.14 in 10.2.2

Jan Lindström,
Principal Engineer,
MariaDB Corporation

Contents

- What to expect from MariaDB 10.2.2 with InnoDB 5.7.14 ?
- What is missing ?
- Why mtr says it is xtradb ?
- XtraDB 5.7.14 ?
- Future of InnoDB in MariaDB

What to expect

- All InnoDB 5.7.14 code base is on MariaDB 10.2.2
- BUT: Some of the features are not compiled
  - #undef MYSQL_57_SELECT_COUNT_OPTIMIZATION
    #undef MYSQL_COMPRESSION
    #undef MYSQL_ENCRYPTION
    #undef MYSQL_FT_INIT_EXT
    #undef MYSQL_INNODB_API_CB
    #undef MYSQL_INNODB_PARTITIONING
    #undef MYSQL_PFS
    #undef MYSQL_RENAME_INDEX
    #undef MYSQL_REPLACE_TRX_IN_THD
    #undef MYSQL_SPATIAL_INDEX
    #undef MYSQL_STORE_FTS_DOC_ID
    #undef MYSQL_TABLESPACES
    #undef MYSQL_VIRTUAL_COLUMNS

What to expect

- GIS (rtree) indexes
- Optimized temporary tables
  - Does not use redo
  - Used dedicated tablespace
  - Intrinsic temporary tables do not have undo log
    Improved buffer pool dump and load
- TRUNCATE Undo tablespaces
- Dynamic buffer pool resize

What to expect

- Better full text pluggable parser (Ngram/MeCab)
- Recovery improvements (Redo log changes)
- Multiple page cleaners support
- Numa Support MySQL 5.7
- More online DDL support:
  - Online varchar extension
- Improved online DDL - bottom up page build (Sorted Index Builds)
  – Page fill factor configurable
- Scalability enhancements

What to expect

Improved defaults

- File format is now Barracuda – dynamic row format
- Large index prefix on by default
- Multiple page cleaners and purge treads
- Buffer pool dump and restore (25%) on by default
- InnoDB strict mode on by default
- Checksum algorithm is CRC32

WL#6943 InnoDB FULLTEXT INDEX: support external parser

- Support external parser in InnoDB following the fts plugin parser framework.
- This makes InnoDB compatible with myisam in terms of functionality.
- CREATE TABLE t1(id INT AUTO_INCREMENT PRIMARY KEY, doc CHAR(255), FULLTEXT INDEX (doc) WITH PARSER my_parser) ENGINE=InnoDB;
- ALTER TABLE articles ADD FULLTEXT INDEX (body) WITH PARSER my_parser;
- CREATE FULLTEXT INDEX ft_index ON articles(body) WITH PARSER my_parser;
- We can also create a myisam table with parser, and alter it to innodb.
- http://dev.mysql.com/doc/refman/5.5/en/writing-full-text-plugins.html
- http://dev.mysql.com/doc/refman/5.6/en/fulltext-boolean.html

5.7 Performance Schema (MDEV-6114)

- WL#7777 Integrate PFS memory instrumentation with InnoDB
- WL#6629 Performance Schema, Status Variables
- WL#7445 PERFORMANCE SCHEMA: instrument SX-lock for rw_lock

WL #6968 InnoDB GIS: R-tree index support.

- Support Spatial Index in InnoDB.
- It is also umbrella worklog for following 2 worklogs:
  - #WL 6609: InnoDB GIS: Support Predicate Locking for GIS index
  - #WL 6745: InnoDB GIS: Support DML for InnoDB GIS index

What is missing?

- Spatial data type
- JSON data type
- Virtual columns and indexes
- Native partitioning
- Transportable tablespaces for partitioned tables
- General tablespaces
- Index rename
- MySQL compression/encryption/key management plugin

WL#6035 Native InnoDB Partitioning

- Implementing partitioning natively in InnoDB, so it does not need ha_partition generic partition engine for supporting partitioning.
- ha_partition uses one ha_innobase handler for each partition which does not share common data with each other resulting in high resource usage which this WL fix.
- It also makes it easier to support other InnoDB features that partitioning currently does not support.
- A new handler ha_innopart is added, inheriting both ha_innobase (for InnoDB access) and Partition_helper (for partitioning support, see wl#4807).

WL#6035 Native InnoDB Partitioning

- And to avoid a proxy object for Partition_handler (see wl#4807) it is also inherited.
- Also ha_partition is changed to inherit Partition_handler directly.
- As a result of this there is no longer any need for .par files for partitioned InnoDB tables, since InnoDB can use its internal data dictionary for finding partitions during rename and delete.
- The optimizer estimate is also changed, especially for records_in_range, where all used partitions are checked instead of only the biggest ones.
- http://mysqlserverteam.com/innodb-native-partitioning-early-access/

WL#6205: InnoDB: Implement CREATE TABLESPACE for general use

- Also WL#7957: Add MDL for tablespaces
- CREATE TABLESPACE `tblspace_name` ADD DATAFILE 'tablespace.ibd' [FILE_BLOCK_SIZE=n];
- CREATE TABLE tbl_name TABLESPACE=`tblspace_name`;
- ALTER TABLE tbl_name TABLESPACE=`tblspace_name`;
- DROP TABLESPACE `tblspace_name`;

Index on non-materialized virtual columns

- WL#8149 B-tree Index Support on non-materialized virtual columns
- WL#8114 Don't store virtual generated columns in database
- WL#8227 Support SEs to create index on virtual generated columns
- WL#8481 Callback for computation of virtual column index values from InnoDB purge threads

WL#8003: Server support for attachable transactions.

- The patch introduces the server-side support (as opposed to the InnoDB-side support) of attachable transactions.
- Attachable transaction is a tiny subclass of a nested transaction. Attachable transactions are AC-RO-RC-NL transactions (auto-commit, read-only, read-committed, non-locking).
- Attachable transactions will be used later by to read from transactional system tables.

JSON

- WL#7909: Server side JSON functions
- WL#8132: JSON datatype and binary storage format
- WL#8170: Expression analyzer for GC
- WL#8249: JSON comparator
- WL#8539: Ordering of scalar JSON values

Other

- WL#7123 Additional query optimization for Fulltext Search
- WL#411: Generated columns (some new thd_* functions ?)
- WL#6835: InnoDB: GCS Replication: Deterministic Deadlock Handling (High Prio Transactions in InnoDB)
- WL#6711: Support InnoDB as additional storage engine for tmp table
- WL#6555 "Online rename index".
- WL#8190: Refactor low-level thread handling (my_thread.h)

Other

- WL#5769: Keyring service for MySQL
- WL#8821: Innodb tablespace encryption key rotation SQL commands
  - ALTER INSTANCE ROTATE INNODB MASTER KEY
- WL#8548: InnoDB: Transparent data encryption
  - ENCRYPTION="Y"/"N"

Why mtr thinks it is xtradb?

- Currently innodb_plugin that is normally used as shared library using dynamic loading is compiled to static library.
- Thus it is a bug on mtr that it does not fully recognize different innodb's

Xtradb 5.7.14 ?

- Started
  - 116 conflicting files remaining
  - From experience there is 6-8 hard one’s
  - From experience most of the errors last time I made in ha_innodb.cc

Future of InnoDB in MariaDB ?

- We have InnoDB from Oracle
- We have InnoDB from Percona (xtradb)
  - There are some small differences but nothing major
- We have merged changes from WebScaleSQL to both Oracle and Percona InnoDB
- Our own development done to both
- Do we really need two InnoDB storage engines ???

“Cracy Ideas”

- Columnar storage for InnoDB
- New redo log
  - More like a ARIES like crash recovery
  - No more 2 passes over redo log
  - Easier point-in-time recovery
- New storage format for tables
  - Bigger and better compressed storage