MySQL & InnoDB fundamentals

Justin Swanhart
PLMCE 2015

MYSQL WORLDWIDE CONFERENCE & EXPO

PERCONA LIVE

MySQL is a multi-threaded daemon

- mysqld
  - Accepts TCP connections
  - Authenticates users
  - Answers queries
  - One thread per connection model
    - pool-of-threads plugin available for Percona Server, MariaDB, MySQL Enterprise edition (not free)

my.cnf file

- MySQL settings normally in /etc/my.cnf
  - On Windows my.ini
- [mysqld_safe]
  - Settings for startup, like open-files-limit
- [mysqld]
  - Settings for MySQL, like sort_buffer_size

MySQL capabilities

- SQL92 compatible parser and feature set
- Storage engines are used for tables
  - Storage engine selection is extremely important
- Indexes are btree indexes
- Joins are indexed-nested-loop
  - If no indexes for join, block nested loop
  - No hash joins

Transactions

- Transactions are implemented by storage engines
- InnoDB and TokuDB support ACID transactions and XA (distributed) transactions
- InnoDB also supports foreign keys

Transactions

- MySQL defaults to auto_commit
- Start transaction with
  - BEGIN
  - START TRANSACTION
  - START TRANSACTION WITH CONSISTENT SNAPSHOT

Transactions

- Use COMMIT to finalize changes into database
- Use ROLLBACK to undo changes
- Use SAVEPOINTS to stage changes into multiple sets
    - ROLLBACK TO SAVEPOINT …

Transactions

- ACID
  - Atomicity
  - Consistency
  - Isolation
  - Durability

Storage engines - MyISAM

- MyISAM
  - Replaced original ISAM storage engine
  - Stores data in a heap structure
  - Can use fixed or dynamic row formats
  - Indexes point to physical file offsets
  - *Table level locking* (not good for high concurrency)

Storage engines – MyISAM (cont)

- MyISAM caches indexes in the key buffer
  - *key_buffer_size* – controls how large it is
  - MyISAM stores indexes in .MYI files
  - Key buffer larger than sum of .MYI file sizes
  - More than one key buffer can be created and assigned to tables

Storage engines - MyISAM

- MyISAM does not cache data
  - Relies on OS file cache
  - Must ensure enough free memory to cache data in OS
  - Don’t set key_buffer_size too large

Storage engines - MyISAM

- MyISAM is *not transactional!*
- *Can not roll back*
  - Pressing ctrl-C (or network error) in the middle of an update will leave some rows updated, others not
  - This can cause replication to stop (diff number of rows affected on slave and master) and slave becomes inconsistent
- Server crash can leave tables in inconsistent state

Storage engines - MyISAM

- Repair after crash
  - REPAIR TABLE command
  - Conceptually similar to fsck on non-logging filesystem
  - Can take a VERY long time on tables of significant size

Storage engines - MyISAM

- MyISAM best for non-critical data that can be recreated if necessary
- MyISAM does not have checksums (no data protection)
- Lack of concurrency for writes means MyISAM is only good for read-mostly workload

Storage engines - MyISAM

- MyISAM best for non-critical data that can be recreated if necessary
- MyISAM does not have checksums (no data protection)
- Lack of concurrency for writes means MyISAM is only good for read-mostly workload

Storage engines - InnoDB

- InnoDB is 100% ACID compliant
  - Uses a log file to ensure changes are not lost
  - Default is to fsync log at every commit
    - Unless few writes, battery-backed RAID or flash needed
    - Can be relaxed (innodb_flush_log_at_trx_commit)
    - If relaxed, up to one second of transactions can be lost
  - Roll forward / roll back recovery after crash

Storage engines - InnoDB

- InnoDB caches indexes and data
  - Cache area is called InnoDB buffer pool
  - Configured with innodb_buffer_pool_size
  - General rule of thumb is 80% of ram but if you have >100GB of ram, might be higher

Storage engines - InnoDB

- Buffer pool is an LRU
  - Least recently used list
  - Caches *pages*
- Adaptive hash indexes
  - In order to reduce cost of secondary lookups, builds in-memory hash of hot pages

Storage engines - InnoDB

- InnoDB tables are *clustered* tables
  - Also known as *index-organized tables*
  - Data in the table is stored with the PRIMARY KEY
  - This is significantly different compared to MyISAM

Storage engines - InnoDB

- PRIMARY KEY is *very* important
  - Should be short (auto_increment often best)
  - INSERT data in sorted order by PK as much as possible (prevents expensive page splits)
  - Secondary indexes contain a copy of the PK (not physical offset) so long PK will bloat table

Storage engines - InnoDB

- InnoDB uses more space than MyISAM
  - Each row has transaction_id and rollback pointer
  - Pages have a 15/16 fill factor, meaning some space is reserved for updates to prevent page splits

Storage engines - InnoDB

- SELECT statements do not take locks
  - Exception is SELECT .. [FOR UPDATE | LOCK IN SHARE MODE]
- ROW level locking for writes
  - Fine grain locking locks individual rows
  - In a range scan, all scanned rows are locked*
- Locks are held for duration of transation

Storage engines - InnoDB

- ROLLBACK / UNDO
  - InnoDB has rollback segments
  - Each transaction gets a rollback segment 'slot'
  - Rollback segment can not be cleaned up past oldest open transaction!
  - Indexes can contain multiple copies of a row with different visiblity

Storage engines – InnoDB important settings

- Innodb_buffer_pool_size
- Innodb_log_file_size
  - By default two logs are created of this size
  - Innodb_log_files_in_group configures this
- Innodb_log_buffer_size
  - Defaults to 8MB
- Innodb_file_per_table

Other important settings

- Sort_buffer_size
- Query_cache_size (should be 0!)
- Table_open_cache / table definition cache
- Log_slow_queries /slow_query_log_file