InnoDB and MyISAM Tuning Fundamentals

Shlomi Noach
openark.org

MySQL Users Group Meeting
Israel, March 3rd. 2009
Parameter Tuning

- One of the last things to tune.
- Data types, Schema structure, Queries: usually have more impact.
- However...
Defaults can be unrealistic

- innodb_buffer_pool_size = 8M
- innodb_log_file_size = 5M
- table_cache = 64
- etc.
- Some parameters should be set up front
Parameter Tuning

- When all else is properly tuned, there's room for fine tuning.
- Understanding the basics is crucial for proper tuning.
MyISAM

Table data: not cached by MySQL

Keys: cached by MySQL
Key buffer

- Holds keys for all tables.
- key_buffer_size=1G
Key buffer issues

- Has a single access lock.
- Flushes relevant pages on every write.
Overcoming lock bottleneck

MyISAM allows for multiple key buffers.

```%
post_buffer.key_buffer_size = 256M
user_buffer.key_buffer_size = 128M

General key buffer    user_buffer    post_buffer
1G                   128M           256M

CACHE INDEX post IN post_buffer;
%```
Overcoming key flush

- Use `delay_key_write`
- Flush only occurs when table is closed / page must be removed from cache.
- Corruption is imminent. Any crash *will* corrupt indexes.
Overcoming key flush

- Aggregate operations with LOCK TABLES.
- Flushes on a table's index are suspended until table is unlocked.
- Cause long waits for locks.
Table locks

SELECT   DELETE
SELECT    UPDATE
SELECT    INSERT
Concurrent INSERTs & SELECTs

- Use `concurrent_insert`
- **0**: No concurrency. Table fully locked.
- **1**: INSERTs appended to table's end, provided no holes exist.
- **2**: INSERTs appended to table's end when concurrent SELECTs take place, otherwise fill in holes.
InnoDB

- Tables are clustered by PRIMARY KEY
- Secondary indexes point to PRIMARY KEY values
Choose an appropriate PRIMARY KEY!
The buffer pool

- One single buffer pool.
- Caches keys + table data.
- Does not rely on OS cache (and in fact may prefer to override it)
- `buffer_pool_size=8G`
Transaction writes

- Transactions are written both to buffer pool and to transaction logs.
- Buffer pool is not flushed on transaction commit.
Transaction logs

- Transaction logs are 'undo logs'.
- Larger transaction logs make for less buffer pool flush.
- `innodb_log_file_size=128M`
Transaction flush

- Log flushed per transaction => many disk flush operations per sec
- `innodb_flush_logs_at_trx_commit`
- 0: Weakest: writes & flushes once per second.
- 1: Strongest: writes & flushes on each commit.
- 2: Writes on each commit, flushes once per second.
Many more parameters

- `innodb_flush_method`
- `innodb_file_per_table`
- `MyISAM ROW_FORMAT`
- etc.
- Tune carefully, test & benchmark!
Thank you!

Hope to see you in the next MySQL Users Group meeting!