

# InnoDB and MyISAM Tuning Fundamentals

Shlomi Noach  
[openark.org](http://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

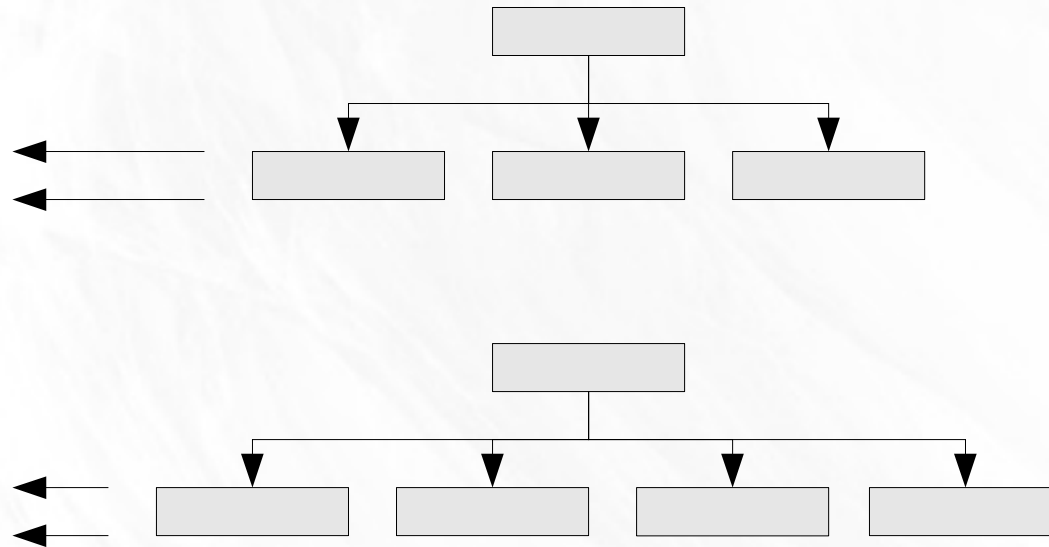
# Parameter Tuning

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

# MyISAM



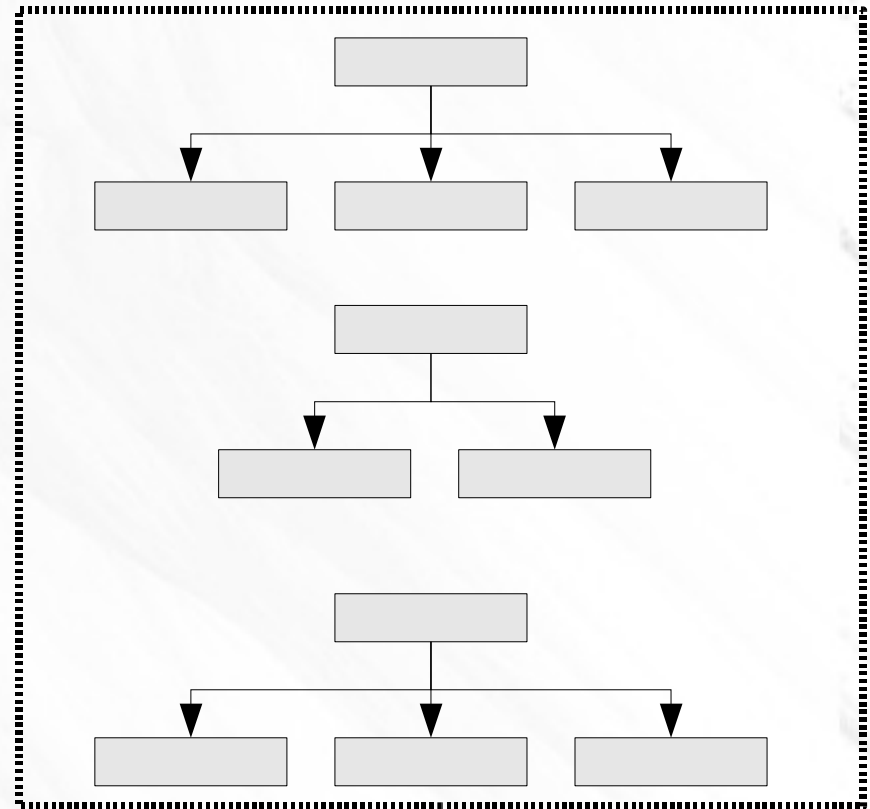
**Table data:  
not cached by MySQL**



**Keys: cached by MySQL**

# Key buffer

- Holds keys for all tables.
- **key\_buffer\_size=1G**



# Key buffer issues

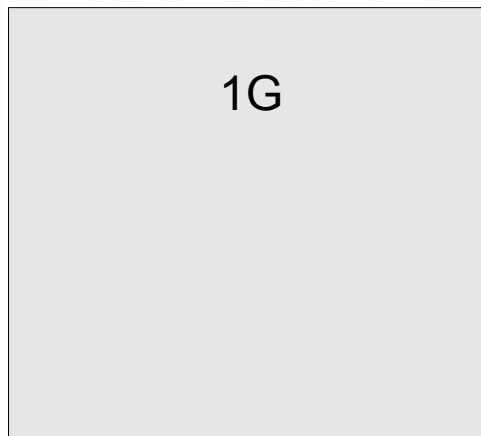
- Has a single access lock.
- Flushes relevant pages on every write.

# Overcoming lock bottleneck

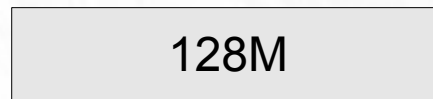
MyISAM allows for multiple key buffers.

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

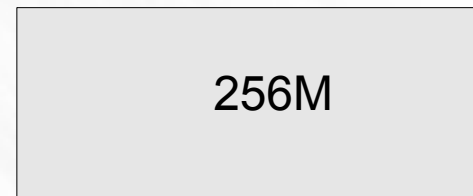
General key buffer



user\_buffer



post\_buffer



```
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

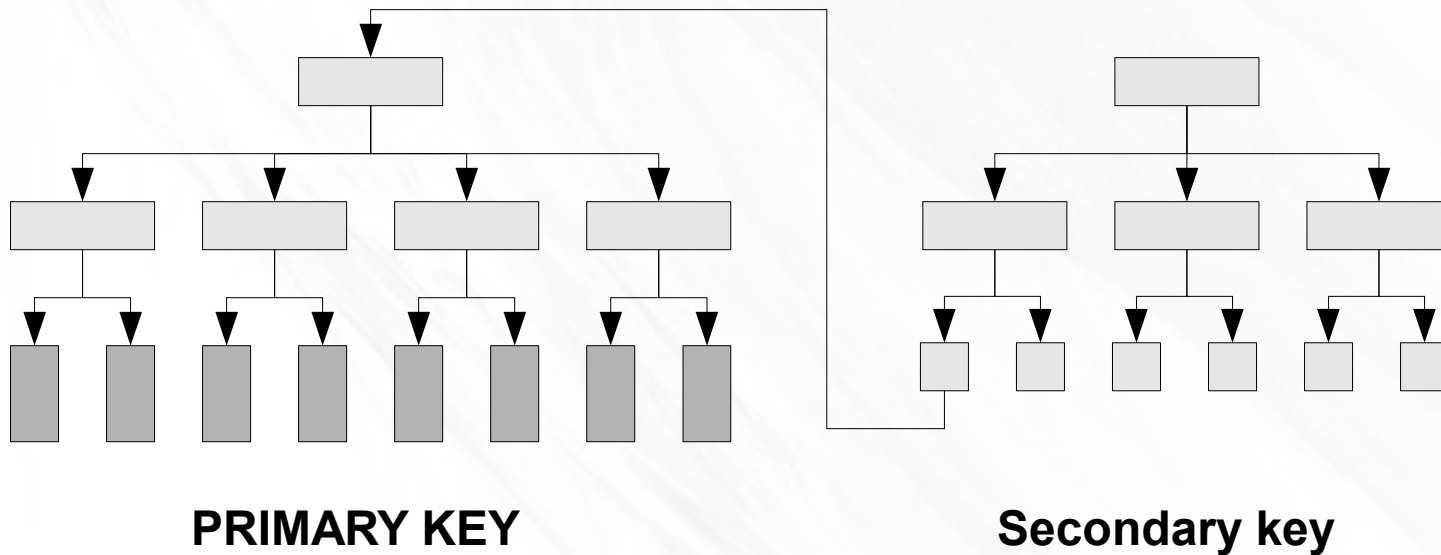


# Concurrent INSERTs & SELECTs

- 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



# InnoDB

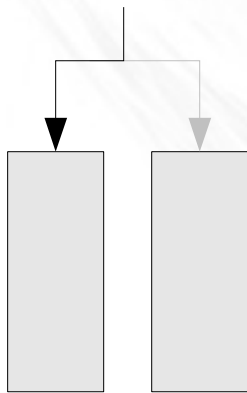
Choose an appropriate PRIMARY KEY!

# The buffer pool

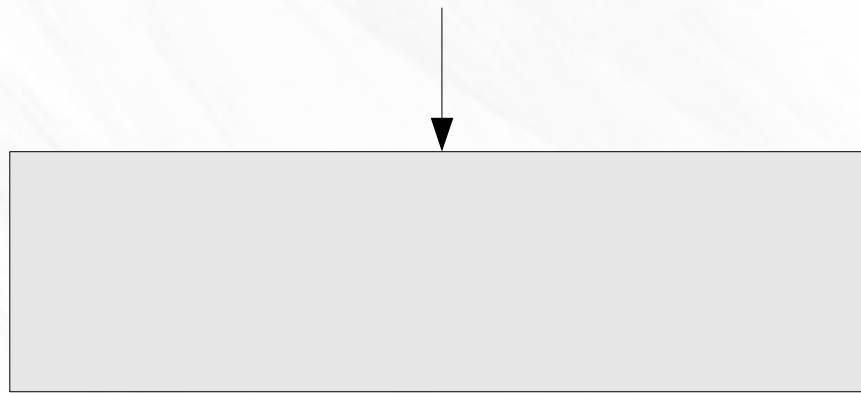
- One single buffer pool.
- Caches keys + table data.
- Does not rely on OS cache (and in fact may prefer to override it)
- **buffer\_pool\_size=8G**

# Transaction writes

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



**Transaction logs**



**Buffer pool**



# 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!**