Upgrading to MySQL 5.6

http://bit.ly/upgrade56

Sheeri Cabral
@sheeri
Mozilla Database Engineering

Briefly

- Stop MySQL
- Remove 5.5
- Install 5.6
- Start MySQL
- mysql_upgrade

Possible, but not the official way!

Major Version Upgrades

- **Export with mysqldump**
- Stop MySQL
- Remove 5.5
- **Remove data**
- Install 5.6
- Start MySQL
- **Import**
- mysql_upgrade

Major Version Upgrades

- **Export with mysqldump**
- Stop MySQL
- Remove 5.5
- **Remove data**

If migrating, this helps get rid of unused stuff

e.g. MariaDB 5.5 to MySQL 5.6

aria files

What if MySQL does not start?

Check the error log

Removed variables

Removed Variables

- have_csv, have_innodb
  - SHOW ENGINES
  - INFORMATION_SCHEMA.ENGINES
- log_slow_queries
  - slow_query_log
- log
  - general_log

Removed Variables

- engine_condition_pushdown
  - deprecated in 5.5
  - optimizer_switch variable
  - engine_condition_pushdown flag
    - on, off, default
    - Default is on
- other flags to optimizer_switch – join, subquery

Deprecated Variables

- Warnings, not errors
- Will be removed in the future
- Change these now
- Show how MySQL is changing

Ignored Variables

- “new”
  - Used in 4.0
  - Turned on 4.1 behavior
  - Always OFF
  - Kept for backwards compatibility
- ignore_builtin_innodb
  - All builtin, no plugin in 5.6

Deprecated Variables

- innodb_use_sys_malloc
  - Default is on
  - Why deprecated?

Deprecated Variables

- thread_concurrency
  - only used on Solaris
- Counterpart is NOT innodb_thread_concurrency
  - innodb_read_io_threads
  - innodb_write_io_threads
  - default 4 each

INSERT DELAYED

- All system variables deprecated
  - max_delayed_threads/max_insert_delayed_threads
  - delayed_insert_limit
  - delayed_insert_timeout
  - delayed_queue_size

INSERT DELAYED

- All status variables deprecated
  - Monitoring – alerting/graphing
  - Delayed_errors
  - Delayed_insert_threads
  - Delayed_writes
  - Not_flushed_delayed_rows

MySQL OK, can't connect

- secure_auth on by default
- No pre 4.1 passwords
- 4.1 was released as production Oct 2004

Tangent: SHA 256 Plugin

- Already loaded in MySQL 5.6
- Can globally require it
    - default_authentication_plugin = sha256_password
- old_passwords = 2
    - Changes the PASSWORD function to sha256

Tangent: SHA 256 Plugin

```
mysql> set old_passwords=0;
    -> SELECT PASSWORD('password');
Query OK, 0 rows affected (0.00 sec)

+-------------------------------------------+
| PASSWORD('password')                      |
+-------------------------------------------+
|*2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 |
+-------------------------------------------+
1 row in set (0.00 sec)
```

Tangent: SHA 256 Plugin

```
mysql> set old_passwords=1;
    -> SELECT PASSWORD('password');
Query OK, 0 rows affected (0.00 sec)

+----------------------+
| PASSWORD('password') |
+----------------------+
| 5d2e19393cc5ef67     |
+----------------------+
1 row in set (0.00 sec)
```

Tangent: SHA 256 Plugin

```
mysql> set old_passwords=2;
    -> SELECT PASSWORD('password');
Query OK, 0 rows affected (0.00 sec)

+-------------------------------------------
  ---------------------+
| PASSWORD('password')
                       |
+-------------------------------------------
  ---------------------+
| $5$`77WT3oSiBQ)*Ae]
  %f$1myhy9VDbxq76wxo1wqi3N5z/wSYgqr3.8nM6rUHzp1  |
+-------------------------------------------
  ---------------------+
1 row in set (0.01 sec)
```

Variables that are unused
and deprecated

Nothing to worry about

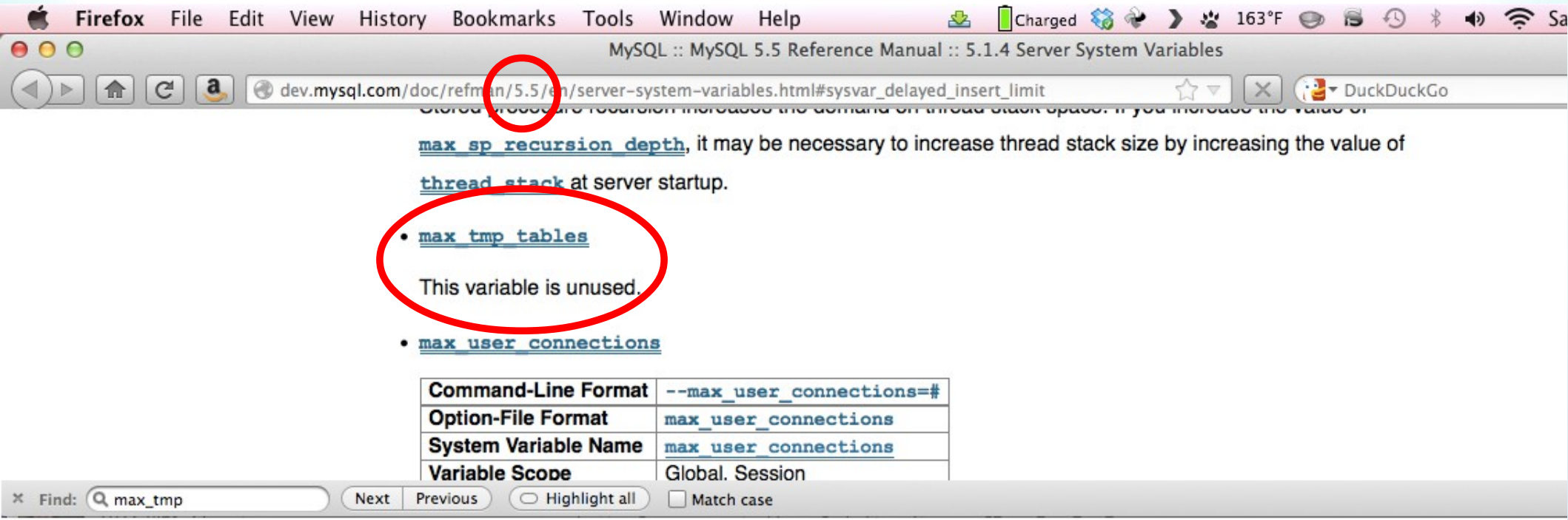
Probably not set anyway

Unused and Deprecated

- date_format
- datetime_format
- time_format
- innodb_mirrored_log_groups
- innodb_version
    - Not used past MySQL 5.5.30
    - Same version as MySQL version

Unused and Deprecated

- max_tmp_tables
  - Not size of temp tables



Deprecated, Replaced

- master_retry_count
  - CHANGE MASTER TO...MASTER_RETRY_COUNT
  - master-* is now gone
  - Except master-info-file location
  - master-host, master-user, master-password, master-port, master-ssl-*
    - Deprecated in 5.1
    - Removed in 5.5

Deprecated, Replaced

- storage_engine
  - default_storage_engine
- innodb_additional_mem_pool_size
  - Now multiple innodb buffer pools

innodb_locks_unsafe_for_binlog

- Default 0, meaning gap locking is enabled
- Searches/index scans use next-key locking
- If 1, searches/index scans use index-record locking
- Does not disable gap locking for:
  - Foreign key checks
  - Duplicate key checks
- Use READ COMMITTED isolation level instead

READ COMMITTED vs. innodb_locks_unsafe_for_binlog

- Isolation level is finer-grained, more flexible control
- Isolation level global and session
  - innodb_locks_unsafe_for_binlog global only
- Isolation level dynamic
  - innodb_locks_unsafe_for_binlog static

READ COMMITTED vs. innodb_locks_unsafe_for_binlog

- Disabling gap locks allows phantom row insertion

```
SELECT id FROM tbl
WHERE id>100 FOR UPDATE
```

  - `INSERT INTO tbl (id) VALUES (101);`

- SELECT in same transaction, new value
  - Why you need gap locking
  - Not SERIALIZABLE, READ COMMITTED at best

SHOW PROFILES

- Use PERFORMANCE_SCHEMA instead
  - PERFORMANCE_SCHEMA is better
- The end of an era
- http://bugs.mysql.com/bug.php?id=24795
- Demo'd Nov 2006
- Submitted bug Dec 4 2006
- Was not fully pushed until May 2009
- No longer takes 2.5 years to get patches in!

Pay attention

This probably affects you

Automatic Timestamping

- Auto-timestamp before 5.6
  - TIMESTAMP field
  - One per table
  - DEFAULT CURRENT_TIMESTAMP
  - ON UPDATE CURRENT_TIMESTAMP
  - Both

Automatic Timestamping

- Auto-timestamp in 5.6
  - TIMESTAMP or DATETIME field
  - Any/all in the table
  - DEFAULT CURRENT_TIMESTAMP
  - ON UPDATE CURRENT_TIMESTAMP
  - Both
  - Works the same

Automatic Timestamping in 5.6

- DEFAULT CURRENT_TIMESTAMP
  - Not auto-updated
- No DEFAULT specified
  - DATETIME defaults to NULL
  - If field is NOT NULL, defaults to 0
  - TIMESTAMP defaults to 0
  - If field allows NULL, defaults to NULL
- Watch out for SQL_MODE
  - NO_ZERO_DATE, TRADITIONAL

Automatic Timestamping in 5.6

- First TIMESTAMP field in table
  - DEFAULT CURRENT_TIMESTAMP, ON UPDATE CURRENT_TIMESTAMP
- Or set a DEFAULT
- Or set to allow NULL
  - NULL inserts NULL, not timestamp
  - TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP
- Or set explicit_defaults_for_timestamp

Changed behavior

ON/OFF, 0/1, TRUE/FALSE

e.g. slow_query_log=1 works

sort_buffer_size

- Entire sort_buffer allocated before 5.6
- Old default just under 2M
- MySQL estimates how much to allocate
  - Up to sort_buffer_size
- New default 256K

Variable with new defaults

If not set, default is changed

“auto tune”

If set, consider un-setting

innodb_autoextend_increment

- Size in MB of autoextend for ibdata
- Old default 8
- New default 64

innodb_buffer_pool_instances

- Number of buffer pools
- Each buffer pool manages:
  - Free lists
  - Flush lists
  - LRUs
  - Buffer pool mutex

innodb_buffer_pool_instances

- Old default 1
- New default 8
  - innodb_buffer_pool size is evenly split into these
- Except for Windows 32-bit, autosized (-1)

innodb_concurrency_tickets

- InnoDB Kernel

- Ticketing system
  - Old default 500
  - New default 5000

innodb_data_file_path

- File name : initial size : [autoextend]
- Old default: ibdata1:10M:autoextend
- New default ibdata1:12M:autoextend
- Not a big change

innodb_log_file_size

- Size of each logfile in group
- Old default: 5M
- New default 48M
- Crash recovery performance better since 5.5
  - larger size is deemed acceptable

innodb_old_blocks_time

- Buffer pool age
- LRU lists
- How long a block in the old sublist waits until moving to the new sublist, in ms
- Old default 0, moves immediately
- New default 1000, waits 1 sec
- Full table scans do not automatically kill buffer pool

innodb_purge_threads

- Number of background purge threads
- Old default 0
  - Part of the master thread
- New default 1
- New range up to 32

innodb_stats_on_metadata

- During metdata statements
- ANALYZE TABLE like stats updating
  - SHOW TABLE STATUS, SHOW INDEX
  - INFORMATION_SCHEMA.TABLES
  - INFORMATION_SCHEMA.STATISTICS
- Old default ON
- New default OFF
- Less overhead, but isn't it less accurate?

innodb_stats_on_metadata

Not with some new variables!

- Less overhead, but isn't it less accurate?

innodb_stats_persistent

- Written to disk
  - Persistent across mysql restarts
- Less overhead
- More consistent statistics
- Can set per-table
  - STATS_PERSISTENT

Sample Pages Defaults

- innodb_stats_persistent_sample_pages
  - 20 by default
- innodb_stats_transient_sample_pages
  - 8 by default
- Can set per table
  - STATS_SAMPLE_PAGES

innodb_open_files

- For multiple tablespaces
  - innodb_file_per_table
- How many open .ibd files at once
- Old default 300
- New default -1
- max(table_open_cache, 300)

table_open_cache

- Max number of open tables
- Old default 400
- New default 2000
- innodb_open_files is max(table_open_cache, 300)

table_definition_cache

- How many table definitions are cached
- Old default 400
- New default -1 (autosized)
- 400 + (table_open_cache / 2)
- Default 1400

back_log

- TCP queue
- Old default 50
- New default -1
- 50 + (max_connections / 5)
- max_connections default is 151
- 50 + (151 / 5) = 80
- SHOW VARIABLES LIKE 'back_log' shows value

binlog_row_event_max_size

- ROW format max size
- Events are grouped into chunks this size, if possible
- Old default 1024 (1k)
- New default 8192 (8k)

flush_time

- How frequently tables are flushed, synced to disk
- Frees up resources on systems w/ minimal resources
- Old default 1800 Windows, 0 other
- New default 0

join_buffer_size

- One buffer allocated for each join between tables
- At least minimum size is allocated
- Range is still up to 4Gb
- Old default minimum 128K
- New default minimum 256K

max_allowed_packet

- Max size of a packet
- Range is still up to 1 Gb
- Old default 1Mb
- New default 4 Mb

max_connect_errors

- How many connect errors in a row
  - For a fixed time
  - Aborted_connections
- FLUSH HOSTS if that happens
- Not flexible, errors per user nor FLUSH per user
- Old default 10
- New default 100

open_files_limit

- How many files mysqld can open
  - Depends on operating system
- Old default 0
  - MySQL cannot change
- New default -1 (autosized)

open_files_limit

- Maximum of:

- 10 + max_connections + (table_open_cache * 2)
- max_connections * 5
- open_files_limit value specified at startup XOR 5000

open_files_limit

max_connections default is 151

table_open_cache default is 2000

open_files_limit

- Maximum of:

  max_connections default is 151

  table_open_cache default is 2000
- 10 + max_connections + (table_open_cache * 2)
- max_connections * 5
- open_files_limit value specified at startup XOR 5000

open_files_limit default

- Maximum of:

  max_connections default is 151

  table_open_cache default is 2000
- 10 + 151 + (2000 * 2) = 4161
- 151 * 5 = 755
- open_files_limit value specified at startup XOR 5000

- If no variables set, default is 5000

Performance Schema

- performance_schema variable defaults to ON
  - Was OFF in 5.5
  - Much less overhead in 5.6
- Many performance schema variables are autosized
- Another presentation

  video: http://bit.ly/mysql_ps_video

  slides: http://bit.ly/mysql_ps

Query Cache

- query_cache_type
  - Old default ON
  - New default OFF
- query_cache_size
  - Old default 0
  - New default 1M

Query Cache

- Should be off and 0
- If you need it, use DEMAND type
  - SELECT SQL_CACHE

sql_mode

- Old default ''
- New default NO_ENGINE_SUBSTITUTION
  - Error if storage engine is not available

Replication file syncing

- sync_master_info
- sync_relay_log
- sync_relay_log_info
    - Old default 0
        - relies on OS to sync
    - New default 10000
        - syncs using fdatasync after 10,000 events

For master_info_repository=FILE

Replication file syncing

- New functionality
- Default 10000
    - Updates table after 10,000 events

For master_info_repository=TABLE

Questions/Comments/Feedback?

Slides:
http://bit.ly/upgrade56