MySQL 5.7 Technical Update

2015/04/24

MySQL Global Business Unit
Shinya Sugiyama / 杉山真也
MySQL Principal Sales Consult, MySQL Global Business Unit

ORACLE

MySQL

SAFE HARBOR STATEMENT

以下の事項は、弊社の一般的な製品の方向性に関する概要を説明するものです。
また、情報提供を唯一の目的とするものであり、いかなる契約にも組み込むことはできません。
以下の事項は、マテリアルやコード、機能を提供することをコミットメントするものではない為、
購買決定を行う際の判断材料になさらないで下さい。

オラクル製品に関して記載されている機能の開発、リリースおよび時期については、
弊社の裁量により決定されます。

MySQL 5.7 Release Candidate Available!

RC

パフォーマンス & 拡張性

- MySQL 5.6比2倍の速度
- InnoDBの機能拡張:
  Online&Bulk load オペレーション高速化
- レプリケーションの改善
  (multi-source, multi-threaded slaves等)
- 新しいオプティマイザコストモデル:
  greater user control & better query performance

管理性

- Performance Schema改善
- MySQL SYS Schema改善
- セキュリティの向上:
  より安全な初期化, セットアップ&管理
- NEW! JSONのSupport (now in labs)

And many more new features and enhancements... http://mysqlserverteam.com/the-mysql-5-7-7-release-candidate-is-available/

ORACLE

1 MySQL 5.7におけるパフォーマンスと拡張性

- Performanceベンチマーク結果
- Optimizer
- General Tablespace Support
- Native partitioning Support
- Temporary Tables
- INDEX全般
- Lock処理
- 全文検索

2 MySQL 5.7における管理面の強化

- Performance Schema (SYS)
- オンライン処理の拡張
- SYSLOGサポート
- GIS
- セキュリティ

MySQL 5.7における パフォーマンスと拡張性の改良

MySQL 5.7 RCにおける主な改良点

- **InnoDB:** トランザクション処理性能、可用性、IO性能の向上
- **Optimizer:** より詳細なEXPLAIN、パーサ、SQL処理性能
- **Lock:** 単一コネクションで複数のユーザレベルロック、リファクタリング
- **機能拡張:** Trigger, 32,64K Page, クエリ・リライト・プラグイン,生成列
- **全文検索:** InnoDB 全文検索、中国語、日本語、韓国語対応

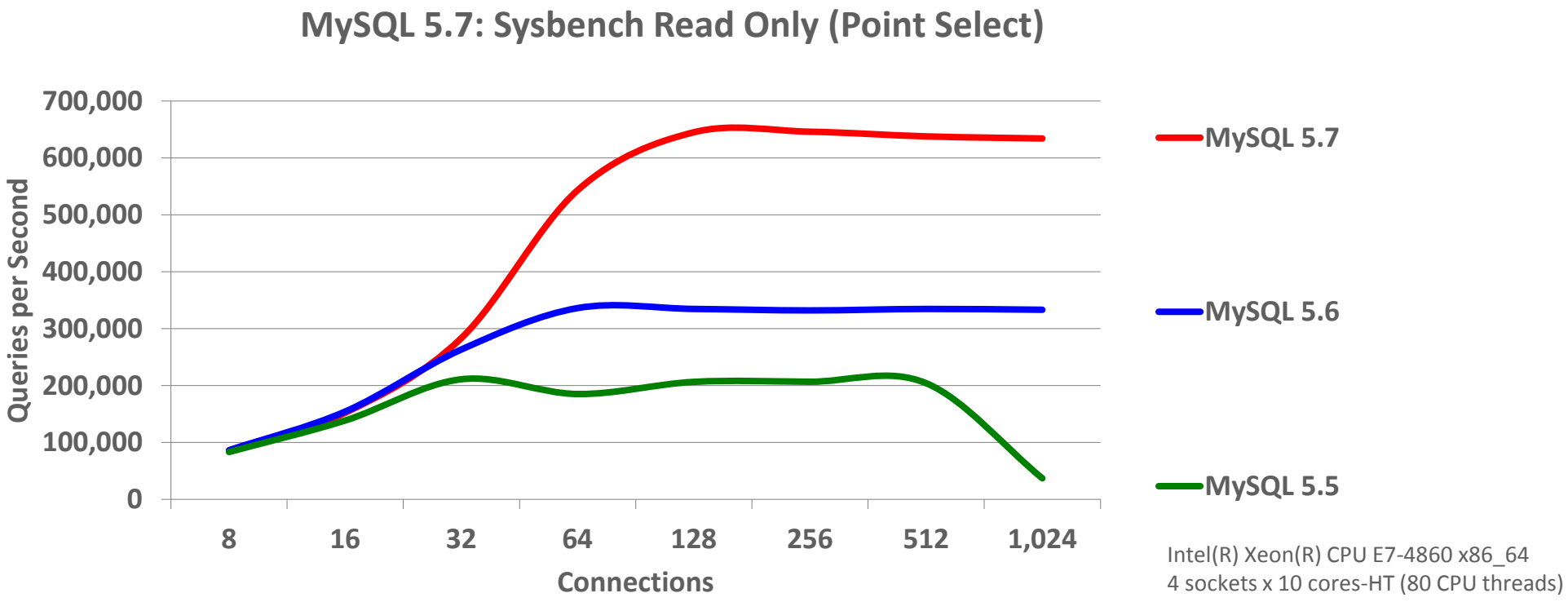
Available Now! dev.mysql.com/downloads/mysql/

MySQL 5.7: Sysbench Benchmark

MySQL 5.6より2倍高速

MySQL 5.5より3倍高速

645,000 QPS



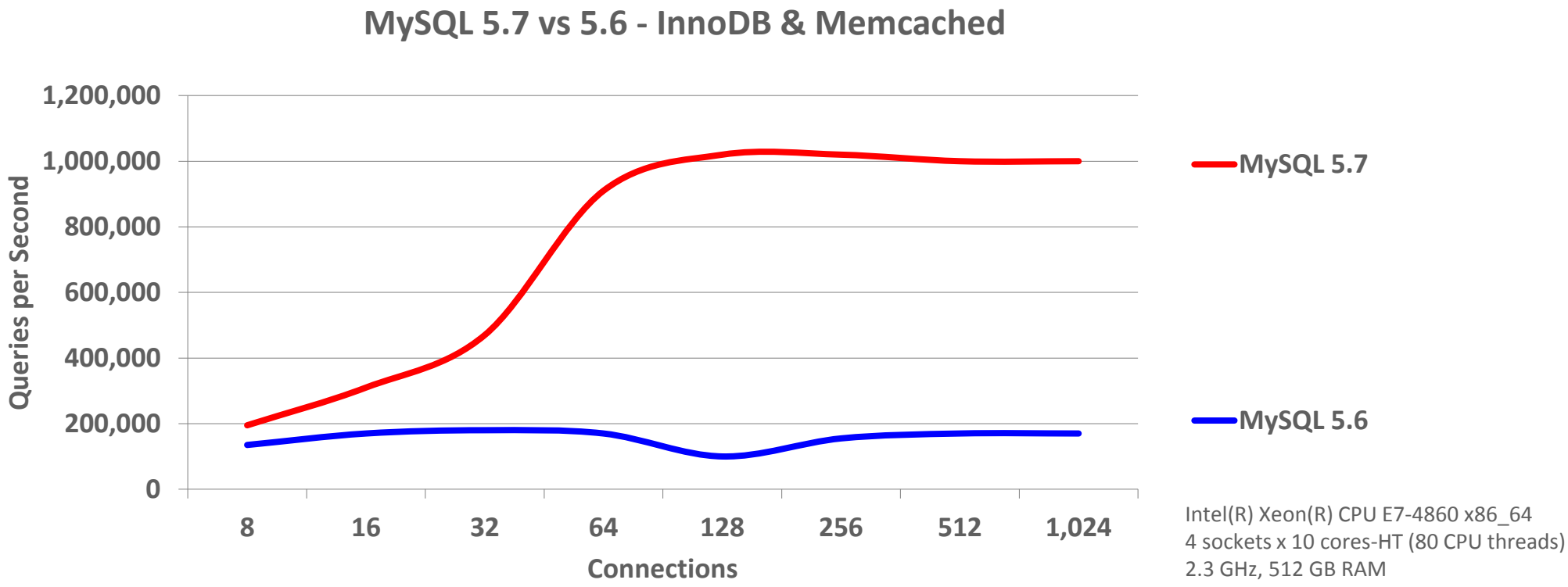
Intel(R) Xeon(R) CPU E7-4860 x86_64
4 sockets x 10 cores-HT (80 CPU threads)
2.3 GHz, 512 GB RAM
Oracle Linux 6.5

MySQL 5.7: InnoDB, NoSQL With Memcached

MySQL 5.6より6倍以上高速

Thank you, Facebook

1,000,000 QPS

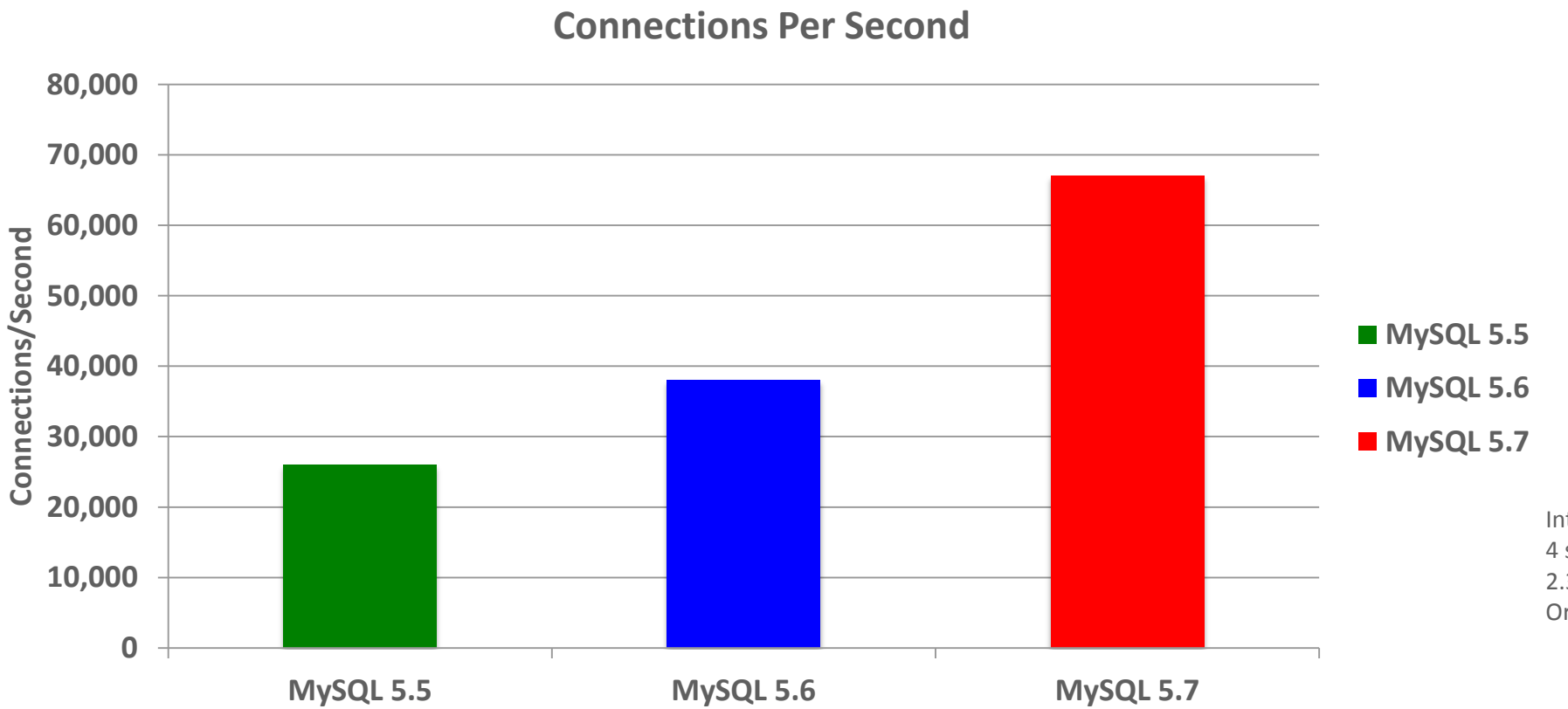


Intel(R) Xeon(R) CPU E7-4860 x86_64
4 sockets x 10 cores-HT (80 CPU threads)
2.3 GHz, 512 GB RAM
Oracle Linux 6.5

MySQL 5.7: 秒間接続数

MySQL 5.6より1.7倍高速
MySQL 5.5より2.5倍高速

67,000 接続/秒



Intel(R) Xeon(R) CPU E7-4860 x86_64
4 sockets x 10 cores-HT (80 CPU threads)
2.3 GHz, 512 GB RAM
Oracle Linux 6.5

参照：Improving connect/disconnect performance

オプティマイザ – 新コストモデル

SQL文の実行性能を向上

- 新しいコストモデルによりストレージエンジンでの処理を改善
  - より正確で動的なコスト見積もり
  - キーの参照、テーブルスキャン、レンジスキャン、インデックススキャンなど
  - コストベースの意思決定で、経験則のみに基づく判断を回避
- インデックスからレコードへの参照の見積もり改善
- コスト値はEXPLAINのJSON出力に含まれる
- 様々な追加要素にてコストを設定可能
  - ディスクI/O処理性能
  - メモリ処理性能

```
root@localhost [nyosm]>SELECT * FROM mysql.server_cost;
+------------------------------+------------+---------------------+---------+
| cost_name                    | cost_value | last_update         | comment |
+------------------------------+------------+---------------------+---------+
| disk_temptable_create_cost   |       NULL | 2014-12-18 17:43:11 | NULL    |
| disk_temptable_row_cost      |       NULL | 2014-12-18 17:43:11 | NULL    |
| key_compare_cost             |       NULL | 2014-12-18 17:43:11 | NULL    |
| memory_temptable_create_cost |       NULL | 2014-12-18 17:43:11 | NULL    |
| memory_temptable_row_cost    |       NULL | 2014-12-18 17:43:11 | NULL    |
| row_evaluate_cost            |       NULL | 2014-12-18 17:43:11 | NULL    |
+------------------------------+------------+---------------------+---------+
6 rows in set (0.00 sec)

root@localhost [nyosm]>SELECT * FROM mysql.engine_cost;
+-------------+-------------+------------------+------------+---------------------+---------+
| engine_name | device_type | cost_name        | cost_value | last_update         | comment |
+-------------+-------------+------------------+------------+---------------------+---------+
| default     |           0 | io_block_read_cost |       NULL | 2014-10-09 18:51:46 | NULL    |
+-------------+-------------+------------------+------------+---------------------+---------+
1 row in set (0.00 sec)
```

オプティマイザ – その他

- INを用いた問いクエリ処理の最適化
- フルテキストクエリ処理の最適化
- より効率の良いソート処理
- SQL標準に準拠した”ONLY_FULL_GROUP_BY” SQLモード

Setting the SQL Mode

The default SQL mode in MySQL 5.7 is

`ONLY_FULL_GROUP_BY,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES`. The

`ONLY_FULL_GROUP_BY` and `STRICT_TRANS_TABLES` modes were added in MySQL 5.7.5, and

`NO_AUTO_CREATE_USER` in MySQL 5.7.7.

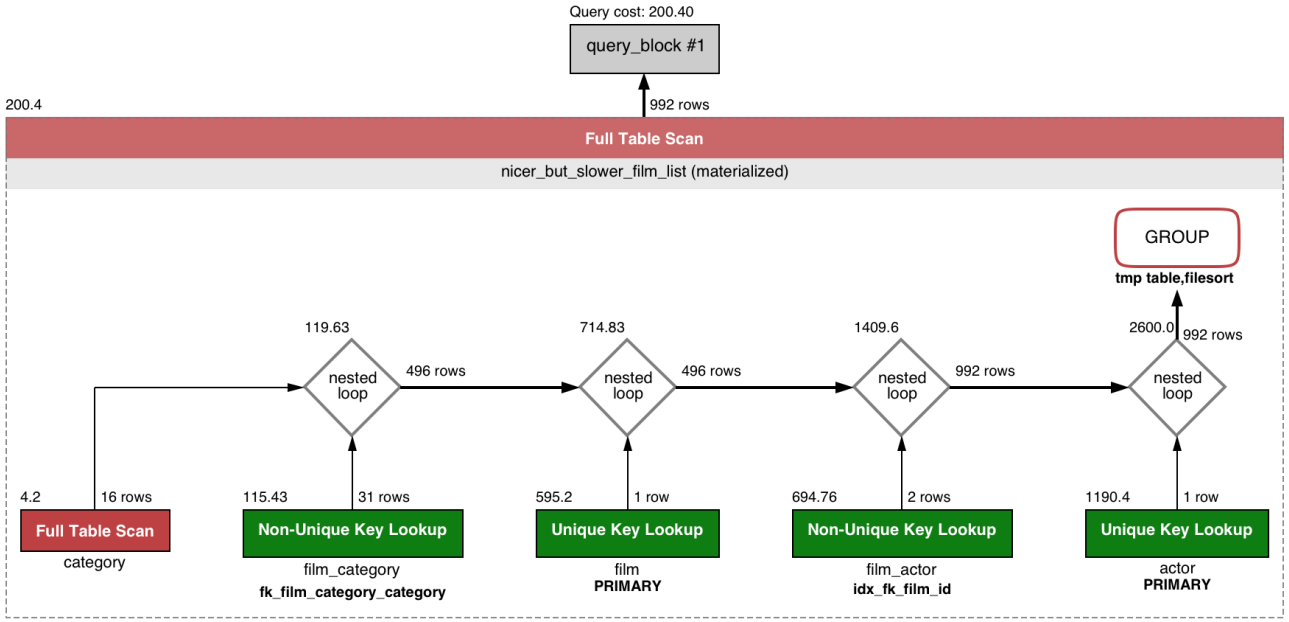
```
+---------------+-----------------------------------------------------------------------------+
| Variable_name | Value                                                                       |
+---------------+-----------------------------------------------------------------------------+
| sql_mode      | ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION |
+---------------+-----------------------------------------------------------------------------+
```

```
root@localhost [test]> select text,count(*) from T_ONLINE_DDL where text = 'Group by Test A';
ERROR 1140 (42000): Mixing of GROUP columns (MIN(),MAX(),COUNT(),...) with no GROUP columns is illegal if there is no GROUP BY clause
root@localhost [test]> select text,count(*) from T_ONLINE_DDL where text = 'Group by Test A' group by text;
+-----------------+----------+
| text            | count(*) |
+-----------------+----------+
| Group by Test A |        3 |
+-----------------+----------+
1 row in set (0.00 sec)
```

オプティマイザ – JSON EXPLAINへのコスト情報追加

より具体的な値をベースに最適化を実施

- JSON EXPLAINを拡張
  - 出力可能なコスト情報を全て表示
  - MySQL WorkbenchのVisual Explainにも表示




```
{
 "query_block": {
  "select_id": 1,
  "cost_info": {
   "query_cost": "200.40"
  },
  "table": {
   "table_name": "nicer_but_slower_film_list",
   "access_type": "ALL",
   "rows_examined_per_scan": 992,
   "rows_produced_per_join": 992,
   "filtered": 100,
   "cost_info": {
    "read_cost": "2.00",
    "eval_cost": "198.40",
    "prefix_cost": "200.40",
    "data_read_per_join": "852K"
   },
   "used_columns": [
    "FID",
    "title",
    "description",
    "category",
    "price",
    "length",
    "rating",
    "actors"
   ],
...
```

InnoDB - General Tablespace Support

A general tablespace is a shared tablespace, similar to the system tablespace. It can hold multiple tables, and supports all table row formats. General tablespaces can also be created in a location relative to or independent of the data directory.

```
[USER01]> CREATE TABLESPACE U_TABLESPACE01 ADD DATAFILE '/home/mysql/user_tablespace01.ibd' Engine=InnoDB;
Query OK, 0 rows affected (0.01 sec)
[USER01]> CREATE TABLESPACE U_TABLESPACE02_8K ADD DATAFILE '/home/mysql/user_tablespace02_8k.ibd' FILE_BLOCK_SIZE =  8192 Engine=InnoDB;
Query OK, 0 rows affected (0.01 sec)

[USER01]> CREATE TABLE `T_USER01` (
>  `id` int(11) NOT NULL AUTO_INCREMENT, `text` varchar(100) DEFAULT NULL, PRIMARY KEY (`id`)
>  ) TABLESPACE = U_TABLESPACE01 ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8mb4;
Query OK, 0 rows affected (0.01 sec)

 [USER01]> CREATE TABLE `T_USER02_8K` (
> `id` int(11) NOT NULL AUTO_INCREMENT, `text` varchar(100) DEFAULT NULL, PRIMARY KEY (`id`)
>) TABLESPACE = U_TABLESPACE02_8K ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8mb4
> ROW_FORMAT=COMPRESSED KEY_BLOCK_SIZE =8;
Query OK, 0 rows affected (0.00 sec)
```

```
root@localhost [USER01]> show variables like 'datadir';
+---------------+-----------------------+
| Variable_name | Value                 |
+---------------+-----------------------+
| datadir       | /usr/local/mysql/data/ |
+---------------+-----------------------+
1 row in set (0.00 sec)

root@localhost [USER01]> system ls /home/mysql/
user_tablespace01.ibd  user_tablespace02_8k.ibd
root@localhost [USER01]>
```

```
root@localhost [USER01]> SELECT * FROM information_schema.INNODB_SYS_TABLESPACES where NAME like 'U_%';
+-------+--------------------+------+-------------+------------+-----------+----------------+------------+
| SPACE | NAME               | FLAG | FILE_FORMAT | ROW_FORMAT | PAGE_SIZE | ZIP_PAGE_SIZE | SPACE_TYPE |
+-------+--------------------+------+-------------+------------+-----------+----------------+------------+
|   165 | U_TABLESPACE01     | 2048 | Any         | Any        |     16384 |              0 | General    |
|   166 | U_TABLESPACE02_8K  | 2089 | Barracuda   | Compressed |     16384 |           8192 | General    |
+-------+--------------------+------+-------------+------------+-----------+----------------+------------+
2 rows in set (0.00 sec)
```

参照：13.1.15 CREATE TABLESPACE Syntax

InnoDB - Native partitioning Support

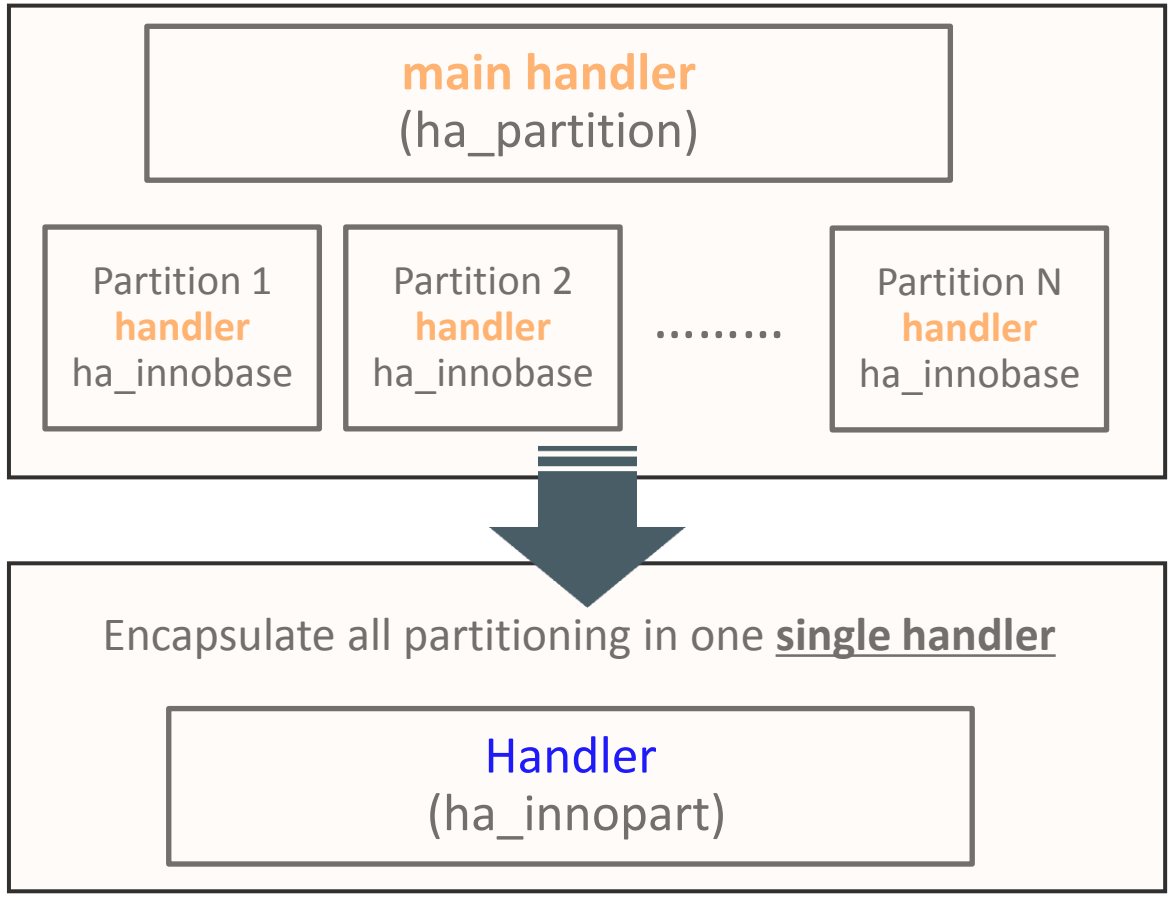
InnoDBのネイティブパーティショニング

- これまでのパーティショニング固有の制限事項を排除可能に
- 多数のパーティションが存在する場合のメモリ消費を抑制

```
CREATE TABLE `t_partition_h` (
  `UID` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `COMMENT` varchar(1024) DEFAULT NULL,
  PRIMARY KEY (`UID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
/*!50100 PARTITION BY HASH (UID) PARTITIONS 8192 */
```

```
CREATE TABLE `t_partition_r` (
  `UID` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `COMMENT` varchar(1024) DEFAULT NULL,
  PRIMARY KEY (`UID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
/*!50100 PARTITION BY RANGE (UID)
(PARTITION p0 VALUES LESS THAN (100) ENGINE = InnoDB,
 PARTITION p1 VALUES LESS THAN (200) ENGINE = InnoDB,
 PARTITION p2 VALUES LESS THAN (300) ENGINE = InnoDB,
 PARTITION p3 VALUES LESS THAN (400) ENGINE = InnoDB,
 PARTITION pmax VALUES LESS THAN MAXVALUE ENGINE = InnoDB) */
```

'Main' handler (pointed out by TABLE::file) which in turn have one handler per partition where all read/writes are forwarded.



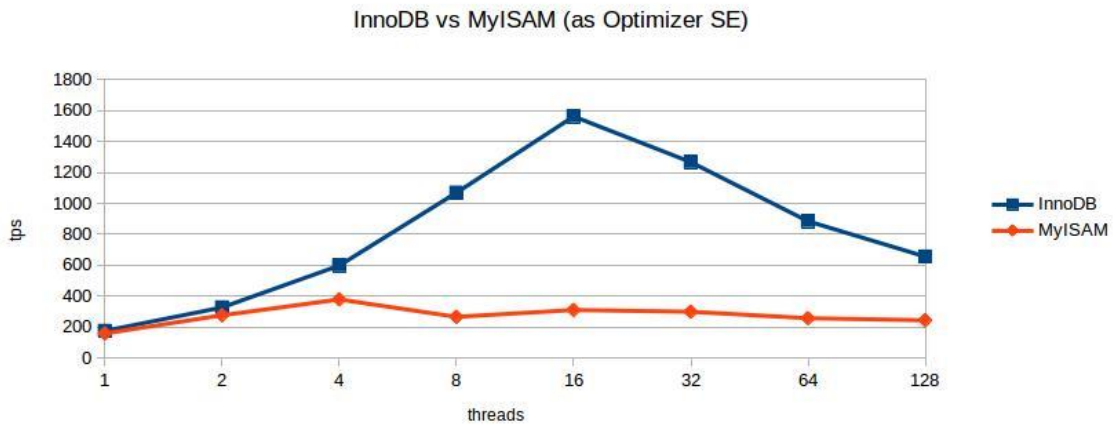
参照: InnoDB Native Partitioning – Early Access

InnoDB - Temporary Tables

一時テーブルを利用した処理の高速化

This variable was added in MySQL 5.7.5.
The default value was changed to INNODB in MySQL 5.7.6.

この変更により、optimizerはディスク上の内部一時テーブルとして**InnoDB**ストレージエンジンを使用するようになりました。



internal_tmp_disk_storage_engine

```
root@localhost [USER01]> SHOW GLOBAL VARIABLES like 'default%';
+-------------------------------+----------------------+
| Variable_name                 | Value                |
+-------------------------------+----------------------+
| default_authentication_plugin | mysql_native_password |
| default_password_lifetime     | 360                  |
| default_storage_engine        | InnoDB               |
| default_tmp_storage_engine    | InnoDB               |
| default_week_format           | 0                    |
+-------------------------------+----------------------+
5 rows in set (0.00 sec)

root@localhost [USER01]> show variables like 'internal_tmp_disk_storage_engine';
+---------------------------------+--------+
| Variable_name                   | Value  |
+---------------------------------+--------+
| internal_tmp_disk_storage_engine | INNODB |
+---------------------------------+--------+
1 row in set (0.00 sec)
```

Contains metadata about active InnoDB temporary tables

```
root@localhost [USER01]> CREATE TEMPORARY TABLE t1 (c1 INT PRIMARY KEY) ENGINE=INNODB;
Query OK, 0 rows affected (0.00 sec)

root@localhost [USER01]> SELECT * FROM INFORMATION_SCHEMA.INNODB_TEMP_TABLE_INFO;
+----------+------------+--------+-------+----------------------+---------------+
| TABLE_ID | NAME       | N_COLS | SPACE | PER_TABLE_TABLESPACE | IS_COMPRESSED |
+----------+------------+--------+-------+----------------------+---------------+
|    16571 | #sqlba3_7_0 |      4 |   160 | FALSE                | FALSE         |
+----------+------------+--------+-------+----------------------+---------------+
1 row in set (0.00 sec)
```

参照： 8.4.4 How MySQL Uses Internal Temporary Tables

InnoDB - Temporary Tables

- 一時テーブル専用の表領域を新規追加
  - CREATE/DROPのパフォーマンスを改善
  - DDLによる変更が短縮され,一部ディスクI/Oも削減
- DMLオペレーションの最適化
  - No REDO logging, no change buffering, less locking
- 内部的な新たなテンポラリーテーブル
  - ACID/MVCCに対応した専用の一時テーブル
  - 軽量且つ超高速で、中間のクエリの実行操作に最適

```
root@localhost [(none)]> show variables like 'innodb_temp_data_file_path';
+---------------------------+-----------------------+
| Variable_name             | Value                 |
+---------------------------+-----------------------+
| innodb_temp_data_file_path | ibtmp1:12M:autoextend |
+---------------------------+-----------------------+
1 row in set (0.00 sec)

root@localhost [(none)]> system ls -lh /usr/local/mysql/data/ibtmp*
-rw-r-----. 1 mysql mysql 12M  4月 22 08:10 /usr/local/mysql/data/ibtmp1
root@localhost [(none)]>
```

```
root@localhost [USER01]> show variables like 'internal_tmp_disk_storage_engine';
+---------------------------------+--------+
| Variable_name                   | Value  |
+---------------------------------+--------+
| internal_tmp_disk_storage_engine | INNODB |
+---------------------------------+--------+
1 row in set (0.00 sec)
```

参照：14.11 InnoDB Startup Options and System Variables

Avoid Creating Temporary Table for UNION ALL

UNION ALLクエリ実行時、一時テーブルを不使用

```
SELECT * FROM T_UNION01 UNION ALL SELECT * FROM T_UNION02;
```

- 5.6: 常にすべてのUNIONの結果を一時テーブルで実現する。
- 5.7: ソート処理に使用されない限り、結果は一時テーブルを利用せずクライアントに直接送信されます
- 5.7: 最後の問合せブロックが終了するまで待つ必要が無い為、クライアントは最初の行を直ぐに受け取る事が可能
- 5.7: 少ないメモリとディスク消費量

```
root@localhost [test]> select @@version;
+-----------------------------------------+
| @@version                               |
+-----------------------------------------+
| 5.6.24-enterprise-commercial-advanced-log |
+-----------------------------------------+
1 row in set (0.00 sec)

root@localhost [test]> explain SELECT * FROM T_UNION01 UNION ALL SELECT * FROM T_UNION02;
+----+--------------+------------+------+---------------+------+---------+------+------+-----------------+
| id | select_type  | table      | type | possible_keys | key  | key_len | ref  | rows | Extra           |
+----+--------------+------------+------+---------------+------+---------+------+------+-----------------+
|  1 | PRIMARY      | T_UNION01  | ALL  | NULL          | NULL | NULL    | NULL |    3 | NULL            |
|  2 | UNION        | T_UNION02  | ALL  | NULL          | NULL | NULL    | NULL |    3 | NULL            |
| NULL | UNION RESULT | <union1,2> | ALL  | NULL          | NULL | NULL    | NULL | NULL | Using temporary |
+----+--------------+------------+------+---------------+------+---------+------+------+-----------------+
3 rows in set (0.00 sec)
```

```
root@localhost [USER01]> select @@version;
+-------------+
| @@version   |
+-------------+
| 5.7.7-rc-log |
+-------------+
1 row in set (0.00 sec)

root@localhost [USER01]> explain  SELECT * FROM T_UNION01 UNION ALL SELECT * FROM T_UNION02;
+----+-------------+-----------+------------+------+---------------+------+---------+------+------+----------+-------+
| id | select_type | table     | partitions | type | possible_keys | key  | key_len | ref  | rows | filtered | Extra |
+----+-------------+-----------+------------+------+---------------+------+---------+------+------+----------+-------+
|  1 | PRIMARY     | T_UNION01 | NULL       | ALL  | NULL          | NULL | NULL    | NULL |    3 |   100.00 | NULL  |
|  2 | UNION       | T_UNION02 | NULL       | ALL  | NULL          | NULL | NULL    | NULL |    3 |   100.00 | NULL  |
+----+-------------+-----------+------------+------+---------------+------+---------+------+------+----------+-------+
2 rows in set, 1 warning (0.00 sec)
```

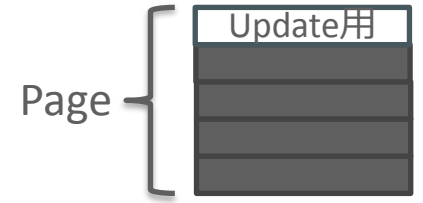
参照：MySQL 5.7: only_full_group_by Improved

INDEXにおける改善

- BULK LOADSによる高速なINDEX作成

  As of MySQL 5.7.5, InnoDB performs a bulk load when creating or rebuilding indexes. Prior to the introduction of bulk load index creation, index entries were inserted into the B-tree one record at a time using insert APIs.

- ボトムアップソートを行いINDEXを構築
  - Improves speed by increasing locality and decreasing node splitting
- PAGEがフルになった場合のみ圧縮が行われる
- 各PAGEの空き容量を”innodb_fill_factor”オプションでコントロール



  innodb_fill_factor defines the percentage of space on each B-tree page that is filled during a sorted index build, with the remaining space reserved for future index growth.

- パフォーマンス向上
  - 2-3x　　ADD/CREATE INDEXのパフォーマンスの改善
  - 2-5%　　標準INSERTオペレーションのパフォーマンス改善

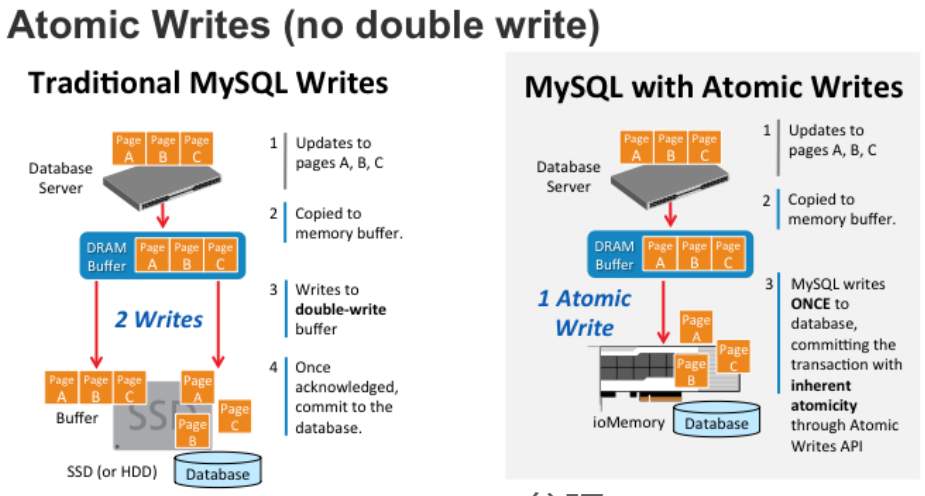
```
+--------------------+-------+
| Variable_name      | Value |
+--------------------+-------+
| innodb_fill_factor | 100   |
+--------------------+-------+
```

参照：14.12.17 Bulk Load for CREATE INDEX

InnoDB Compression

Thank you, SanDisk Fusion-io

- ページレベルでの透過的圧縮
  - バックグラウンドスレッドにより自動的に圧縮
  - IOレイヤにて管理
  - スパースファイルを使用
    サポート済みOSカーネルおよびファイルシステム(NVMFS)が必要
- IO削減
  - MySQLの性能向上
  - ストレージ利用効率向上
  - 書き込みサイクル削減、SSDのライフサイクルを維持
- 全てのInnoDBのデータ、システム表領域、UNDOログが対象



参照：fusionio.com

MySQL 5.7におけるロック

Thank you, Konstantin Osipov!

単一コネクションで複数のユーザレベルロック

- ユーザレベルロックによって相互に排他制御を利用
  - 複数のリソースにアクセスする場合
  - テーブルレベルまたは行レベルのロックが不適切な場合
- 一連のGET_LOCK()関数にて複数のロックを要求
- 独自のユーザレベルロック実装を置き換え
  - MDL(メタデータロック)ロックマネージャを利用
  - ユーザレベルロック、メタデータロック、テーブルのフラッシュの待ちなどによるデッドロックを検知し通知

Improved metadata locking(MDL)

- Fast-path for DML locks　(“fast-path”)
- Lock-free DML lock acquisition
- Lock-free hash
  - Now uses MurmurHash library

Excellent performance - measured on an Intel Core 2 Duo @ 2.4 ghz

```
OneAtATime - 354.163715 mb/sec
FNV - 443.668038 mb/sec
SuperFastHash - 985.335173 mb/sec
lookup3 - 988.080652 mb/sec
MurmurHash 1.0 - 1363.293480 mb/sec
MurmurHash 2.0 - 2056.885653 mb/sec
```

参照：　https://sites.google.com/site/murmurhash/

- 単一テーブルへのDMLアクセスに関してのボトルネックを排除
  - 10% スループット向上 : OLTP_RO/POINT_SELECT sysbench
  - 負荷の高い典型的なDML処理を最適化

参照: Removing Scalability Bottlenecks in the Metadata Locking and THR_LOCK Subsystems in MySQL 5.7
WL#7304: Improve MDL performance and scalability by implementing "fast-path" for DML locks
WL#7305: Improve MDL scalability by using lock-free hash
MySQL-5.7 improves DML oriented workloads

Improved SQL Semantics

- **Triggers**
  - テーブル毎に複数のトリガを作成可能

> As of MySQL 5.7.2, it is possible to define multiple triggers for a given table that have the same trigger event and action time. For example, you can have two BEFORE UPDATE triggers for a table.

```
root@localhost [USER01]> SELECT EVENT_OBJECT_TABLE,TRIGGER_NAME,ACTION_TIMING,ACTION_ORDER,EVENT_MANIPULATION
    -> FROM information_schema.triggers
    -> WHERE information_schema.triggers.event_object_table="T_Trigger";
+--------------------+-----------------+--------------+--------------+--------------------+
| EVENT_OBJECT_TABLE | TRIGGER_NAME    | ACTION_TIMING | ACTION_ORDER | EVENT_MANIPULATION |
+--------------------+-----------------+--------------+--------------+--------------------+
| T_Trigger          | ins_transaction | BEFORE       |            1 | INSERT             |
| T_Trigger          | ins_sum         | BEFORE       |            2 | INSERT             |
| T_Trigger          | upd_check       | BEFORE       |            1 | UPDATE             |
+--------------------+-----------------+--------------+--------------+--------------------+
3 rows in set (0.00 sec)
```

- **Error reporting**
  - 診断エリアをクリア

Error Code	Description
1000 ～	Codes are currently numbered consecutively (～1974使用済)
2000 ～	Second range of error codes used by the client
3000 ～	The new 5.7 server error codes will now instead start at 3000 ～

```
root@localhost [USER01]> SHOW ERRORS;
+-------+------+-----------------------------------------------------------------------------------------------------------------------------------------------------------+
| Level | Code | Message                                                                                                                                                   |
+-------+------+-----------------------------------------------------------------------------------------------------------------------------------------------------------+
| Error | 1064 | You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '' at line 1 |
+-------+------+-----------------------------------------------------------------------------------------------------------------------------------------------------------+
```

参照：Support for multiple triggers per table for the same value of action/timing.

Additional Support for 32K and 64K pages size

MySQLインスタンス内の全てのInnoDBのテーブルスペースのページ·サイズを指定します。
この値はインスタンス構築時に設定し、その後設定した値が継続して適用されます。

innodb_page_size

Command-Line Format	--innodb_page_size=#k	
System Variable	**Name**	innodb_page_size
	Variable Scope	Global
	Dynamic Variable	No
Permitted Values (<= 5.7.5)	**Type**	enumeration
	Default	16384
	Valid Values	4k
		8k
		16k
		4096
		8192
		16384
Permitted Values (>= 5.7.6)	**Type**	enumeration
	Default	16384
	Valid Values	4k
		8k
		16k
		32k
		64k
		4096
		8192
		16384
		32768
		65536

innodb_log_buffer_size

Command-Line Format	--innodb_log_buffer_size=#	
System Variable	**Name**	innodb_log_buffer_size
	Variable Scope	Global
	Dynamic Variable	No
Permitted Values (<= 5.7.5)	**Type**	integer
	Default	8388608
	Min Value	262144
	Max Value	4294967295
Permitted Values (>= 5.7.6)	**Type**	integer
	Default	16777216
	Min Value	1048576
	Max Value	4294967295

The size in bytes of the buffer that InnoDB uses to write to the log files on disk.
The default value changed from 8MB to 16MB in 5.7.6 with the introduction of 32k and 64k innodb_page_size values.

```
root@localhost [USER01]> show variables like 'innodb_page_size';
+------------------+-------+
| Variable_name    | Value |
+------------------+-------+
| innodb_page_size | 16384 |
+------------------+-------+
1 row in set (0.00 sec)

root@localhost [USER01]> show variables like 'innodb_log_buffer_size';
+------------------------+----------+
| Variable_name          | Value    |
+------------------------+----------+
| innodb_log_buffer_size | 16777216 |
+------------------------+----------+
1 row in set (0.00 sec)
```

クエリ・リライト・プラグイン

- クエリの書き換え (パースの前と後)
- パースした後での書き換えプラグイン
  - アプリケーションを変更することなく問題のあるクエリを書き換え
  - ヒントの追加
  - JOIN順の変更
- ORマッパーやサードパーティ製のアプリなどが発行する問題となり得るクエリなどに対応

クエリ・リライト・プラグイン – Sample View

admin@Labs01 :~/sakila-db | root@Labs01 :/home/admin

```
root@localhost [sakila]>select @@version;
+-------------------------+
| @@version               |
+-------------------------+
| 5.7.5-labs-preview-log |
+-------------------------+
1 row in set (0.00 sec)

root@localhost [sakila]>select * from query_rewrite.rewrite_rules;
+--------------------------------------+------------------+-----------------------------------------------------------------------------+---------+---------+
| pattern                              | pattern_database | replacement                                                                 | enabled | message |
+--------------------------------------+------------------+-----------------------------------------------------------------------------+---------+---------+
| select * from actor where first_name = ? | sakila           | select * from actor force index(idx_actor_first_name) where first_name = ? | Y       | NULL    |
+--------------------------------------+------------------+-----------------------------------------------------------------------------+---------+---------+
1 row in set (0.00 sec)

root@localhost [sakila]>select * from actor where first_name = 'JOHNNY'
    -> ;
+----------+------------+-------------+---------------------+
| actor_id | first_name | last_name   | last_update         |
+----------+------------+-------------+---------------------+
|        5 | JOHNNY     | LOLLOBRIGIDA | 2006-02-15 04:34:33 |
|       40 | JOHNNY     | CAGE        | 2006-02-15 04:34:33 |
+----------+------------+-------------+---------------------+
2 rows in set, 1 warning (0.00 sec)
```

Rewrite Rule

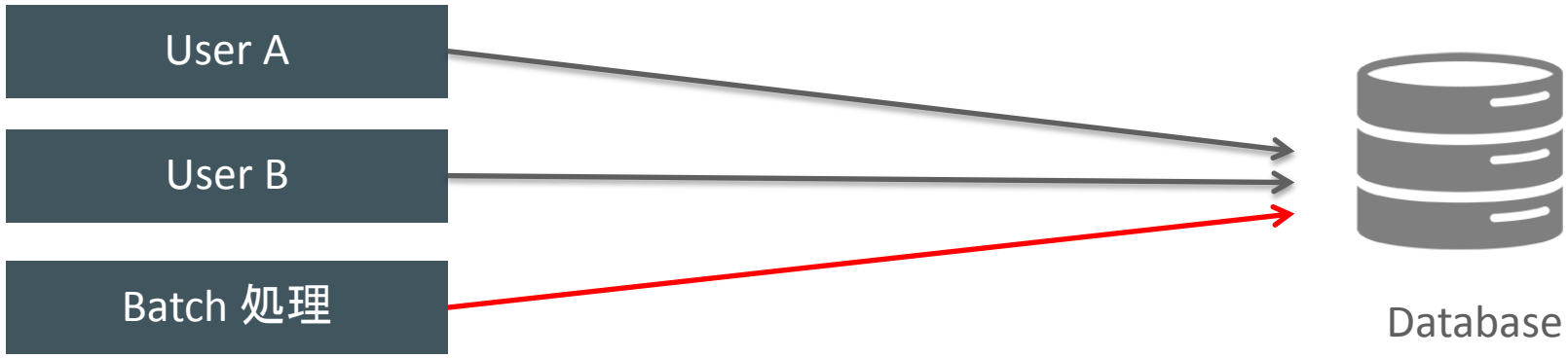
```
[root@Labs01 admin]# tail -f -n 3 /usr/local/mysql2/data/general.log



2014-12-31T13:21:22.423241Z    9 Query  select @@version
2014-12-31T13:21:26.996802Z    9 Query  select * from query_rewrite.rewrite_rules
2014-12-31T13:21:32.765325Z    9 Query  select * from actor force index(idx_actor_first_name) where first_name = 'JOHNNY'
```

Rewrite Result

MySQL 5.7: サーバサイドでのSQL文タイムアウト

Thank you Davi Arnaut!

- サーバサイドにてSQL文をタイムアウト
  - サーバ全体、セッション単位、SELECT文単位で設定可能
- WindowsおよびSolarisにも対応




```
SELECT MAX_STATEMENT_TIME = 109 * FROM my_table;
```

Generated Column Support

```
<type> [ GENERATED ALWAYS ] AS ( <expression> ) [ VIRTUAL|STORED ] [ UNIQUE [KEY] ] [ [PRIMARY] KEY ] [ NOT NULL ] [ COMMENT <text> ]
```

式から生成される列(2種類)

- VIRTUAL(default) : 読み込み時に計算され, データ保存されず,インデックスの作成不可
- STORED: inserted/updated時に計算され,データは保存され,インデックス作成可能

Useful for:

- Functional index: create a stored column, add a secondary index
- Materialized cache for complex conditions
- Simplify query expression

```
CREATE TABLE T_Generated_Column (
> pid int(10) unsigned NOT NULL AUTO_INCREMENT,
> pname varchar(1024) DEFAULT NULL,
> price decimal(10,2),qty   int(10),
> total decimal(10,2) GENERATED ALWAYS AS (price * qty) STORED,
> PRIMARY KEY (pid)
> ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
Query OK, 0 rows affected (0.02 sec)
```

```
root@localhost [USER01]> desc T_Generated_Column;
+-------+------------------+------+-----+---------+------------------+
| Field | Type             | Null | Key | Default | Extra            |
+-------+------------------+------+-----+---------+------------------+
| pid   | int(10) unsigned | NO   | PRI | NULL    | auto_increment   |
| pname | varchar(1024)    | YES  |     | NULL    |                  |
| price | decimal(10,2)    | YES  |     | NULL    |                  |
| qty   | int(10)          | YES  |     | NULL    |                  |
| total | decimal(10,2)    | YES  |     | NULL    | STORED GENERATED |
+-------+------------------+------+-----+---------+------------------+
5 rows in set (0.00 sec)

root@localhost [USER01]> insert into T_Generated_Column(pname,price,qty) values('自転車',10000,3);
Query OK, 1 row affected (0.00 sec)

root@localhost [USER01]> select * from T_Generated_Column;
+-----+----------+----------+------+----------+
| pid | pname    | price    | qty  | total    |
+-----+----------+----------+------+----------+
|   1 | 自転車   | 10000.00 |    3 | 30000.00 |
+-----+----------+----------+------+----------+
1 row in set (0.00 sec)
```

Generated Column Support

Generated Columnに対してIndex作成が可能なので、Where句はindexを利用し最適なコストでデータを抽出可能

```
root@localhost [USER01]> select * from T_Generated_Column;
+-----+-----------+----------+-----+-----------+
| pid | pname     | price    | qty  | total     |
+-----+-----------+----------+-----+-----------+
|   1 | 自転車    | 10000.00 |    3 |  30000.00 |
|   2 | TV        | 30000.00 |    5 | 150000.00 |
|   3 | 冷蔵庫    | 50000.00 |    1 |  50000.00 |
+-----+-----------+----------+-----+-----------+
3 rows in set (0.01 sec)

root@localhost [USER01]> explain select * from T_Generated_Column where total > 100000;
+----+-------------+--------------------+------------+------+---------------+------+---------+------+------+----------+-------------+
| id | select_type | table              | partitions | type | possible_keys | key  | key_len | ref  | rows | filtered | Extra       |
+----+-------------+--------------------+------------+------+---------------+------+---------+------+------+----------+-------------+
|  1 | SIMPLE      | T_Generated_Column | NULL       | ALL  | NULL          | NULL | NULL    | NULL |    3 |    33.33 | Using where |
+----+-------------+--------------------+------------+------+---------------+------+---------+------+------+----------+-------------+
1 row in set, 1 warning (0.01 sec)

root@localhost [USER01]> CREATE INDEX IDX_TOTAL ON T_Generated_Column (total);
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

root@localhost [USER01]> explain select * from T_Generated_Column where total > 100000;
+----+-------------+--------------------+------------+-------+---------------+-----------+---------+------+------+----------+-----------------------+
| id | select_type | table              | partitions | type  | possible_keys | key       | key_len | ref  | rows | filtered | Extra                 |
+----+-------------+--------------------+------------+-------+---------------+-----------+---------+------+------+----------+-----------------------+
|  1 | SIMPLE      | T_Generated_Column | NULL       | range | IDX_TOTAL     | IDX_TOTAL | 6       | NULL |    1 |   100.00 | Using index condition |
+----+-------------+--------------------+------------+-------+---------------+-----------+---------+------+------+----------+-----------------------+
1 row in set, 1 warning (0.00 sec)
```

参照：[Generated Columns in MySQL 5.7.5]

InnoDB - Full Text Search (FTS) - ngram

InnoDB Full Text Search (FTS) にて 中国語, 韓国語，日本語をサポート

N-gram support for Chinese and Korean, additional MeCab support for Japanese

```
CREATE TABLE `N_DEMO` ( `FTS_N_ID` bigint(20) unsigned NOT NULL AUTO_INCREMENT, `title` varchar(100) DEFAULT NULL,
PRIMARY KEY (`FTS_N_ID`), FULLTEXT KEY `ngram_idx` (`title`) /*!50100 WITH PARSER `ngram` */
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8mb4;
```

```
root@localhost [ngram]> show variables like 'ngram_token_size';
+------------------+-------+
| Variable_name    | Value |
+------------------+-------+
| ngram_token_size | 2     |
+------------------+-------+
1 row in set (0.00 sec)
```

```
root@localhost [ngram]> SELECT * FROM N_DEMO WHERE MATCH (title) AGAINST ('sql' IN NATURAL LANGUAGE MODE);
+----------+-------+
| FTS_N_ID | title |
+----------+-------+
|        1 | mysql |
|        2 | MYSQL |
|        3 | MySQL |
|        9 | sq    |
|       11 | ql    |
+----------+-------+
5 rows in set (0.00 sec)

root@localhost [ngram]> SELECT * FROM N_DEMO WHERE MATCH(title) AGAINST('sql' IN BOOLEAN MODE);
+----------+-------+
| FTS_N_ID | title |
+----------+-------+
|        1 | mysql |
|        2 | MYSQL |
|        3 | MySQL |
+----------+-------+
3 rows in set (0.00 sec)

root@localhost [ngram]>
```

```
root@localhost [ngram]> root@localhost [ngram]> SELECT * FROM N_DEMO WHERE MATCH(title) AGAINST('きゅー' IN BOOLEAN MODE);
+----------+------------------------------+
| FTS_N_ID | title                        |
+----------+------------------------------+
|        6 | まいえすきゅーえる           |
|       12 | まいーえすきゅーえる         |
|       13 | まいえーすきゅーえる         |
+----------+------------------------------+
3 rows in set (0.00 sec)

root@localhost [ngram]> SELECT * FROM N_DEMO WHERE MATCH (title) AGAINST ('+きゅー -まいー' IN BOOLEAN MODE);
+----------+------------------------------+
| FTS_N_ID | title                        |
+----------+------------------------------+
|        6 | まいえすきゅーえる           |
|       13 | まいえーすきゅーえる         |
+----------+------------------------------+
2 rows in set (0.00 sec)

root@localhost [ngram]> SELECT * FROM N_DEMO WHERE MATCH (title) AGAINST ('+きゅー -えー' IN BOOLEAN MODE);
+----------+------------------------------+
| FTS_N_ID | title                        |
+----------+------------------------------+
|        6 | まいえすきゅーえる           |
|       12 | まいーえすきゅーえる         |
+----------+------------------------------+
2 rows in set (0.00 sec)
```

InnoDB - Full Text Search (FTS) - mecab

InnoDB Full Text Search (FTS) にて 中国語, 韓国語，日本語をサポート

N-gram support for Chinese and Korean, additional MeCab support for Japanese

```
CREATE TABLE `M_DEMO` ( `FTS_M_ID` bigint(20) unsigned NOT NULL AUTO_INCREMENT, `title` varchar(100) DEFAULT NULL,
PRIMARY KEY (`FTS_M_ID`),  FULLTEXT KEY `mecab_idx` (`title`) /*!50100 WITH PARSER `mecab` */
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8mb4;
```

```
root@localhost [ngram]> show variables like '%ft_min%';
+--------------------------+-------+
| Variable_name            | Value |
+--------------------------+-------+
| ft_min_word_len          | 4     |
| innodb_ft_min_token_size | 2     |
+--------------------------+-------+
2 rows in set (0.00 sec)
```

```
root@localhost [mecab]>  SELECT FTS_M_ID,title,MATCH (title) AGAINST('日本の首都' IN NATURAL LANGUAGE MODE) AS score FROM M_DEMO;
+----------+--------------------------------------+--------------------+
| FTS_M_ID | title                                | score              |
+----------+--------------------------------------+--------------------+
|        1 | 東京都は日本の首都です               |  1.178047776222229 |
|        2 | 京都と大阪は日本の府です             | 0.3624762296676636 |
|        3 | mysql                                |                  0 |
|        4 | MYSQL                                |                  0 |
|        5 | MySQL                                |                  0 |
|        6 | マイエスキューエル                   |                  0 |
|        7 | ﾏｲｴｽｷｭｰｴﾙ                          |                  0 |
|        8 | まいえすきゅーえる                   |                  0 |
+----------+--------------------------------------+--------------------+
8 rows in set (0.00 sec)

root@localhost [mecab]> SELECT FTS_M_ID,title,MATCH (title) AGAINST('日本の首都' IN BOOLEAN MODE) AS score FROM M_DEMO;
+----------+--------------------------------------+--------------------+
| FTS_M_ID | title                                | score              |
+----------+--------------------------------------+--------------------+
|        1 | 東京都は日本の首都です               | 1.178047776222229  |
|        2 | 京都と大阪は日本の府です             |                  0 |
|        3 | mysql                                |                  0 |
|        4 | MYSQL                                |                  0 |
|        5 | MySQL                                |                  0 |
|        6 | マイエスキューエル                   |                  0 |
|        7 | ﾏｲｴｽｷｭｰｴﾙ                          |                  0 |
|        8 | まいえすきゅーえる                   |                  0 |
+----------+--------------------------------------+--------------------+
8 rows in set (0.00 sec)
```

MySQL 5.7における管理面の強化

ORACLE

MySQL

MySQL 5.7 RCにおける運用サポート機能

- **Performance Schema:** 性能統計情報のさらなる追加
- **オンライン処理**においての機能拡張
- **SYSLOG**をLinux/Windows共にネイティブサーポート
- **GIS機能**をInnoDBの空間インデックスとBoost.Geometryの統合でサポート
- **Security**強化として,簡単で安全な初期設定と管理をサポート

- **Replication** for better scalability and availability
- **Fabric** for high availability and sharding

MySQL 5.7: Performance Schema

メモリ統計情報

- 統計情報の収集
  - メモリの利用タイプ別 (キャッシュ、内部バッファ…)

  memory_summary_＊テーブル
  - スレッド/アカウント/ユーザ/ホスト毎のメモリ処理
- 含まれる属性情報
  - メモリ利用量 (バイト)
  - 処理数
  - 最大/最小

SQL文統計情報

- ストアドプロシージャ
- ストアドファンクション
- プリペアードステートメント
- トランザクション

追加情報

- レプリケーションスレーブ情報
- MDLロック統計情報
- スレッドごとのユーザ変数
- Server stage tracking
- 長時間実行されているSQL文
- メモリフットプリントとオーバーヘッドの削減

MySQL SYS Schema

DB管理者、開発者や運用担当者を支援

- **DB管理者や運用担当者の作業効率を改善**
  - サーバの稼働状況、ユーザやホストの状況、主要な稼働指標
  - 性能問題の発見、分析および改善

- **状況をより簡単に把握し理解するための複数のビュー**
  - IO量の高いファイルや処理、ロック、コストの高いSQL文
  - テーブル、インデックス、スキーマの統計
- 他のデータベースにおけるSYS類似機能:
  - Oracle V$表　(動的パフォーマンスビュー)
  - Microsoft SQL Server DMV　(Dynamic Management Views)

```
root@localhost [sys]>select * from x$user_summary limit 0,1\G
*************************** 1. row ***************************
                  user: root
            statements: 465
     statement_latency: 1591048261000
 statement_avg_latency: 3421609163.4409
           table_scans: 28
             file_ios: 2471
      file_io_latency: 1004145900811793
   current_connections: 1
     total_connections: 1
          unique_hosts: 1
        current_memory: 0
total_memory_allocated: 0
1 row in set (0.02 sec)

root@localhost [sys]>select * from user_summary limit 0,1\G
*************************** 1. row ***************************
                  user: root
            statements: 466
     statement_latency: 1.62 s
 statement_avg_latency: 3.47 ms
           table_scans: 29
             file_ios: 2476
      file_io_latency: 00:16:49.39
   current_connections: 1
     total_connections: 1
          unique_hosts: 1
        current_memory: 0 bytes
total_memory_allocated: 0 bytes
1 row in set (0.02 sec)

root@localhost [sys]>
```

MySQL SYS Schema – sample view (1)

Global Memory usage broken down by allocation type

```
root@localhost [sys]>select * from memory_global_by_current_allocated \G
*************************** 1. row ***************************
       event_name: memory/performance_schema/internal_buffers
    current_count: 60
    current_alloc: 89.50 MiB
current_avg_alloc: 1.49 MiB
       high_count: 60
       high_alloc: 89.50 MiB
   high_avg_alloc: 1.49 MiB
1 row in set (0.00 sec)

root@localhost [sys]>
```

```
root@localhost [sys]>select * from
    -> sys.schema_object_overview where db = 'sys';
+-----+-------------+-------+
| db  | object_type | count |
+-----+-------------+-------+
| sys | FUNCTION    |    11 |
| sys | VIEW        |    84 |
| sys | PROCEDURE   |    22 |
+-----+-------------+-------+
3 rows in set (0.03 sec)
```

InnoDB Buffer Memory Usage

```
root@localhost [sys]>select * from sys.innodb_buffer_stats_by_table order by data desc limit 0,10;
+---------------+--------------------+-----------+-----------+-------+-------------+-----------+-------------+
| object_schema | object_name        | allocated | data      | pages | pages_hashed | pages_old | rows_cached |
+---------------+--------------------+-----------+-----------+-------+-------------+-----------+-------------+
| InnoDB System | SYS_TABLES         | 32.00 KiB | 6.98 KiB  |     2 |            2 |         2 |          58 |
| InnoDB System | SYS_INDEXES        | 16.00 KiB | 6.23 KiB  |     1 |            1 |         1 |          90 |
| mysql         | engine_cost        | 16.00 KiB | 54 bytes  |     1 |            1 |         1 |           1 |
| InnoDB System | SYS_FIELDS         | 16.00 KiB | 5.04 KiB  |     1 |            1 |         1 |         116 |
| mysql         | server_cost        | 16.00 KiB | 279 bytes |     1 |            1 |         1 |           6 |
| InnoDB System | SYS_DATAFILES      | 16.00 KiB | 2.64 KiB  |     1 |            1 |         1 |          49 |
| InnoDB System | SYS_TABLESPACES    | 16.00 KiB | 2.59 KiB  |     1 |            1 |         1 |          49 |
| mysql         | innodb_index_stats | 48.00 KiB | 18.11 KiB |     3 |            3 |         3 |         190 |
| InnoDB System | SYS_COLUMNS        | 48.00 KiB | 17.53 KiB |     3 |            3 |         3 |         275 |
| InnoDB System | SYS_FOREIGN        | 32.00 KiB | 1.99 KiB  |     2 |            2 |         2 |          22 |
+---------------+--------------------+-----------+-----------+-------+-------------+-----------+-------------+
10 rows in set (0.04 sec)

root@localhost [sys]>select count(*) from myhttp.hello;
+----------+
| count(*) |
+----------+
|        6 |
+----------+
1 row in set (0.00 sec)

root@localhost [sys]>select * from sys.innodb_buffer_stats_by_table order by data desc limit 0,10;
+---------------+--------------------+-----------+-----------+-------+-------------+-----------+-------------+
| object_schema | object_name        | allocated | data      | pages | pages_hashed | pages_old | rows_cached |
+---------------+--------------------+-----------+-----------+-------+-------------+-----------+-------------+
| InnoDB System | SYS_TABLES         | 32.00 KiB | 6.98 KiB  |     2 |            2 |         2 |          58 |
| InnoDB System | SYS_INDEXES        | 16.00 KiB | 6.23 KiB  |     1 |            1 |         1 |          90 |
| mysql         | engine_cost        | 16.00 KiB | 54 bytes  |     1 |            1 |         1 |           1 |
| InnoDB System | SYS_FIELDS         | 16.00 KiB | 5.04 KiB  |     1 |            1 |         1 |         116 |
| myhttp        | hello              | 16.00 KiB | 330 bytes |     1 |            1 |         1 |           6 |
| mysql         | server_cost        | 16.00 KiB | 279 bytes |     1 |            1 |         1 |           6 |
| InnoDB System | SYS_DATAFILES      | 16.00 KiB | 2.64 KiB  |     1 |            1 |         1 |          49 |
| InnoDB System | SYS_TABLESPACES    | 16.00 KiB | 2.59 KiB  |     1 |            1 |         1 |          49 |
| mysql         | innodb_index_stats | 48.00 KiB | 18.11 KiB |     3 |            3 |         3 |         190 |
| InnoDB System | SYS_COLUMNS        | 48.00 KiB | 17.53 KiB |     3 |            3 |         3 |         275 |
+---------------+--------------------+-----------+-----------+-------+-------------+-----------+-------------+
10 rows in set (0.03 sec)
```

MySQL SYS Schema – sample view (2)

user_summary_by_file_io_type

```
root@localhost [sys]>select * from user_summary_by_file_io_type;
+------------+---------------------------------------+--------+-----------+-------------+
| user       | event_name                            | total  | latency   | max_latency |
+------------+---------------------------------------+--------+-----------+-------------+
| background | wait/io/file/innodb/innodb_data_file  |    599 | 808.41 ms | 75.15 ms    |
| background | wait/io/file/sql/FRM                  |    910 | 250.59 ms | 37.22 ms    |
| background | wait/io/file/myisam/kfile             |     67 | 241.19 ms | 207.83 ms   |
| background | wait/io/file/sql/slow_log             |      4 | 84.44 ms  | 84.38 ms    |
| background | wait/io/file/innodb/innodb_log_file   |     18 | 63.67 ms  | 28.24 ms    |
| background | wait/io/file/myisam/dfile             |     42 | 51.38 ms  | 46.13 ms    |
| background | wait/io/file/sql/binlog_index         |     15 | 23.19 ms  | 20.17 ms    |
| background | wait/io/file/sql/binlog               |     24 | 19.77 ms  | 13.42 ms    |
| background | wait/io/file/sql/ERRMSG               |      5 | 12.67 ms  | 9.93 ms     |
| background | wait/io/file/mysys/charset            |      3 | 12.60 ms  | 12.55 ms    |
| background | wait/io/file/mysys/cnf                |      5 | 11.35 ms  | 11.25 ms    |
| background | wait/io/file/sql/query_log            |      4 | 1.45 ms   | 1.39 ms     |
| background | wait/io/file/sql/casetest             |     10 | 316.11 us | 151.75 us   |
| background | wait/io/file/sql/pid                  |      3 | 236.21 us | 197.57 us   |
| background | wait/io/file/sql/global_ddl_log       |      2 | 22.64 us  | 18.71 us    |
| root       | wait/io/file/csv/data                 | 190276 | 3.18 s    | 436.44 ms   |
| root       | wait/io/file/sql/FRM                  |    442 | 159.43 ms | 19.82 ms    |
| root       | wait/io/file/myisam/kfile             |    424 | 74.92 ms  | 37.00 ms    |
| root       | wait/io/file/csv/metadata             |     21 | 72.04 ms  | 31.49 ms    |
| root       | wait/io/file/myisam/dfile             |    113 | 23.30 ms  | 13.54 ms    |
| root       | wait/io/file/sql/file_parser          |    156 | 1.73 ms   | 64.16 us    |
| root       | wait/io/file/sql/dbopt                |     14 | 542.22 us | 369.09 us   |
+------------+---------------------------------------+--------+-----------+-------------+
22 rows in set (0.01 sec)
```

statement_analysis

```
root@localhost [sys]>select * from statement_analysis limit 2\G!
*************************** 1. row ***************************
           query: SELECT * FROM `user_summary_by ...` ( `performance_schema` . ...
              db: sys
       full_scan: *
      exec_count: 1
       err_count: 0
      warn_count: 0
   total_latency: 542.33 ms
     max_latency: 542.33 ms
     avg_latency: 542.33 ms
    lock_latency: 539.52 ms
       rows_sent: 22
   rows_sent_avg: 22
   rows_examined: 592
rows_examined_avg: 592
      tmp_tables: 0
 tmp_disk_tables: 0
     rows_sorted: 22
sort_merge_passes: 0
          digest: feced1fb353a2384ab8e8e8139fea619
      first_seen: 2015-02-19 13:04:45
       last_seen: 2015-02-19 13:04:45
```

SYS Schema補足情報

■ The MySQL SYS Schema設定

GUI: http://www-jp.mysql.com/products/workbench/

SCRIPT:https://github.com/MarkLeith/mysql-sys

■ The MySQL SYS Schemaドキュメント

https://oracleus.activeevents.com/2014/connect/fileDownload/session/72527FD42DFF7B2148314B9E72BE7B6A/CON3751_Leith-mysql_sys_schema_oow_2014.pdf

■ The MySQL SYS Schema (Video Streaming)

トピック: 管理及びモニタリング, パフォーマンスとスケーラビリティ

プレゼンター: Mark Leith, Senior Software Development Manager, Oracle

http://www-jp.mysql.com/news-and-events/web-seminars/the-mysql-sys-schema/

オンライン処理の拡張

- **Resize the InnoDB Buffer Pool online**
  - オンラインでのバッファーサイズのチューニング
  - データベースの使用パターンの変化にリアルタイムで適応
- **Separate UNDO tablespace**
  - 自動オンラインUNDOログ切り捨て(MySQL 5.7.5～)
  - UNDOログファイルサイズの増加を回避する事が可能
- **Dynamic configuration**
  - Making existing settings dynamically configurable
  - As a design principle for new features & settings
  - その他、幾つかのレプリケーションの設定変更等も
    オンラインで変更可能になりました。

innodb_buffer_pool_size

Command-Line Format	--innodb_buffer_pool_size=#	
System Variable (<= 5.7.4)	Name	innodb_buffer_pool_size
	Variable Scope	Global
	Dynamic Variable	No
System Variable (>= 5.7.5)	Name	innodb_buffer_pool_size
	Variable Scope	Global
	Dynamic Variable	Yes
Permitted Values (32-bit platforms)	Type	integer
	Default	134217728
	Min Value	5242880
	Max Value	2**32-1
Permitted Values (64-bit platforms)	Type	integer
	Default	134217728
	Min Value	5242880
	Max Value	2**64-1

```
+--------------------------+-------+
| Variable_name            | Value |
+--------------------------+-------+
| innodb_undo_directory    | .     |
| innodb_undo_log_truncate | OFF   |
| innodb_undo_logs         | 128   |
| innodb_undo_tablespaces  | 0     |
+--------------------------+-------+
```

```
+--------------------------+------------+
| Variable_name            | Value      |
+--------------------------+------------+
| innodb_max_undo_log_size | 1073741824 |
+--------------------------+------------+
```

参照：14.4.8 Truncating Undo Logs That Reside in Undo Tablespaces

オンライン処理の拡張 (DDL)

Additional Online ALTER TABLE support

- VARCHARサイズ拡張　(例) varchar(100) → varchar(255)

```
ALTER TABLE T_ONLINE_DDL ALGORITHM=INPLACE, CHANGE COLUMN text text VARCHAR(255);
```

```
root@localhost [test]> select @@version;
+-----------------------------------------+
| @@version                               |
+-----------------------------------------+
| 5.6.24-enterprise-commercial-advanced-log |
+-----------------------------------------+
1 row in set (0.00 sec)

root@localhost [test]> desc T_ONLINE_DDL;
+-------+--------------+------+-----+---------+----------------+
| Field | Type         | Null | Key | Default | Extra          |
+-------+--------------+------+-----+---------+----------------+
| id    | int(11)      | NO   | PRI | NULL    | auto_increment |
| text  | varchar(100) | YES  |     | NULL    |                |
+-------+--------------+------+-----+---------+----------------+
2 rows in set (0.00 sec)

root@localhost [test]> ALTER TABLE T_ONLINE_DDL ALGORITHM=INPLACE, CHANGE COLUMN text text VARCHAR(255);
ERROR 1846 (0A000): ALGORITHM=INPLACE is not supported. Reason: Cannot change column type INPLACE. Try ALGORITHM=COPY.
root@localhost [test]>
```

```
root@localhost [USER01]> select @@version;
+--------------+
| @@version    |
+--------------+
| 5.7.7-rc-log |
+--------------+
1 row in set (0.00 sec)

root@localhost [USER01]> desc T_ONLINE_DDL;
+-------+--------------+------+-----+---------+----------------+
| Field | Type         | Null | Key | Default | Extra          |
+-------+--------------+------+-----+---------+----------------+
| id    | int(11)      | NO   | PRI | NULL    | auto_increment |
| text  | varchar(100) | YES  |     | NULL    |                |
+-------+--------------+------+-----+---------+----------------+
2 rows in set (0.00 sec)

root@localhost [USER01]> ALTER TABLE T_ONLINE_DDL ALGORITHM=INPLACE, CHANGE COLUMN text text VARCHAR(255);
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0

root@localhost [USER01]>
```

- Rename Index

```
ALTER TABLE T_ONLINE_DDL RENAME INDEX idx_text TO index_text;
```

```
root@localhost [USER01]> ALTER TABLE T_ONLINE_DDL RENAME INDEX idx_text TO index_text;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

MySQL 5.7 : Linux/Unix環境でのSyslogサポート

Thank you, Simon Mudd at booking.com

- syslogのネイティブサポート
- シンプルなオプションでsyslogにログを出力
- サーバの起動オプションとして設定
- サーバ稼働中に動的に変更可能
  - システム変数 log_syslog (ON/OFF, デフォルトはOFF).

```
root@localhost [(none)]>show variables like '%syslog%';
+-------------------------+-------------+
| Variable_name           | Value       |
+-------------------------+-------------+
| log_syslog              | ON          |
| log_syslog_facility     | daemon      |
| log_syslog_include_pid  | ON          |
| log_syslog_tag          | MySQL57_TEST|
+-------------------------+-------------+
4 rows in set (0.00 sec)
```

```
[root@Labs01 log]# cat /var/log/messages | grep MySQL57_TEST
Dec 18 11:31:54 Labs01 mysqld-MySQL57_TEST[5368]: /usr/local/mysql/bin/mysqld (mysqld 5.7.5-labs-http-log) starting as process 5368 ...
Dec 18 11:31:54 Labs01 mysqld-MySQL57_TEST[5368]: InnoDB: Mutexes and rw_locks use GCC atomic builtins
Dec 18 11:31:54 Labs01 mysqld-MySQL57_TEST[5368]: InnoDB: Uses event mutexes
Dec 18 11:31:54 Labs01 mysqld-MySQL57_TEST[5368]: InnoDB: GCC builtin __sync_synchronize() is used for memory barrier
Dec 18 11:31:54 Labs01 mysqld-MySQL57_TEST[5368]: InnoDB: Compressed tables use zlib 1.2.3
Dec 18 11:31:54 Labs01 mysqld-MySQL57_TEST[5368]: InnoDB: Using Linux native AIO
Dec 18 11:31:54 Labs01 mysqld-MySQL57_TEST[5368]: InnoDB: Number of pools: 1
Dec 18 11:31:54 Labs01 mysqld-MySQL57_TEST[5368]: InnoDB: Not using CPU crc32 instructions
Dec 18 11:31:54 Labs01 mysqld-MySQL57_TEST[5368]: InnoDB: Initializing buffer pool, total size = 128.0M, instances = 1, chunk size = 1
Dec 18 11:31:54 Labs01 mysqld-MySQL57_TEST[5368]: InnoDB: Completed initialization of buffer pool
Dec 18 11:31:54 Labs01 mysqld-MySQL57_TEST[5368]: InnoDB: Highest supported file format is Barracuda.
Dec 18 11:31:54 Labs01 mysqld-MySQL57_TEST[5368]: InnoDB: Creating shared tablespace for temporary tables
Dec 18 11:31:54 Labs01 mysqld-MySQL57_TEST[5368]: InnoDB: Setting file './ibtmp1' size to 12 MB. Physically writing the file full; Ple
Dec 18 11:31:54 Labs01 mysqld-MySQL57_TEST[5368]: InnoDB: File './ibtmp1' size is now 12 MB.
Dec 18 11:31:54 Labs01 mysqld-MySQL57_TEST[5368]: InnoDB: 96 redo rollback segment(s) found. 96 redo rollback segment(s) are active.
Dec 18 11:31:54 Labs01 mysqld-MySQL57_TEST[5368]: InnoDB: 32 non-redo rollback segment(s) are active.
```

Improved Tools

- Improved innochecksum
  - Specify checksum algorithm (innodb,crc32,none)
  - Rewrite current checksum (--write)
  - supports files greater than 2GB in size

```
[root@misc01 mysql]# /usr/local/mysql/bin/innochecksum -C crc32 --verbose user_tablespace01.ibd

Variables (--variable-name=value)
and boolean options {FALSE|TRUE}  Value (after reading options)
--------------------------------- ----------------------------------------
verbose                           TRUE
count                             FALSE
start-page                        0
end-page                          0
page                              0
strict-check                      crc32
no-check                          FALSE
allow-mismatches                  0
write                             crc32
page-type-summary                 FALSE
page-type-dump                    (No default value)
log                               (No default value)
[root@misc01 mysql]#
```

- SSL support to mysqlbinlog tool

  リモートインスタンスとの通信を暗号化

```
[root@misc01 mysql]# ./bin/mysqlbinlog --help | egrep ssl
  --ssl               If set to ON, this option enforces that SSL is
                      server. To disable client SSL capabilities use --ssl=OFF.
                      (Defaults to on; use --skip-ssl to disable.)
```

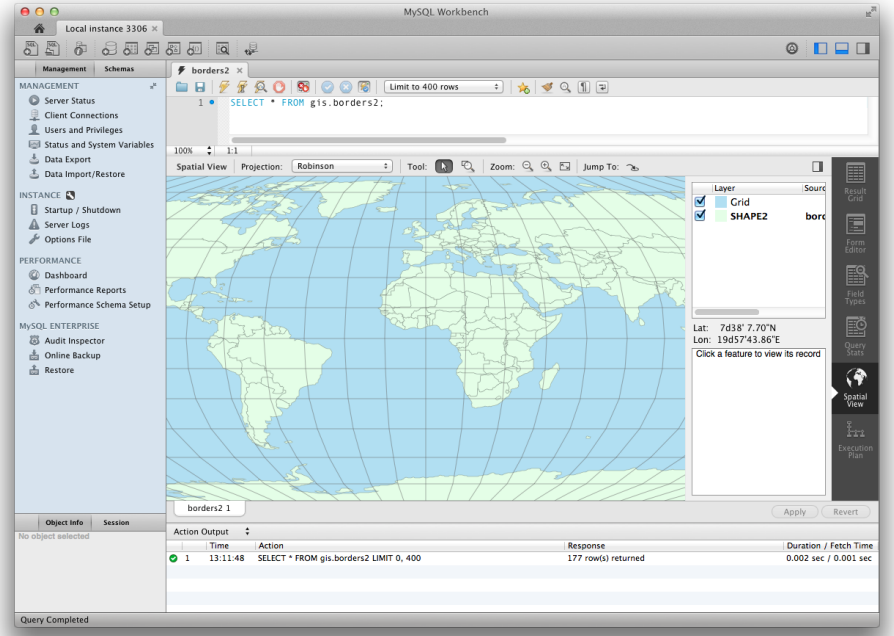
- Rewrite of mysql_upgrade into C from perl
  - Allows direct calls to things like mysqlcheck

- Rewrite of mysql_secure_installation and mysql_install_db scripts to C++
  - Improve security and fix issues

MySQL 5.7: GIS - Boost.Geometryとの統合

- 独自コードの置き換え
  - 空間図形情報の計算
  - 空間図形情報の分析
- OGC(Open Geospatial Consortium)準拠
  - パフォーマンスの向上
- Boost.Geometryによる効果
  - エキスパートとの交流
  - 非常に活発なコミュニティ
- Boost.Geometryへのコントリビュートも

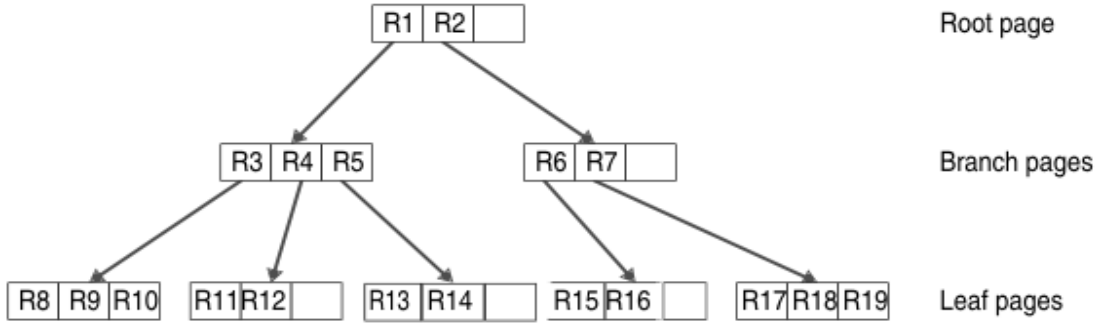
例）ALTER TABLE テーブル名 add SPATIAL index(列名);

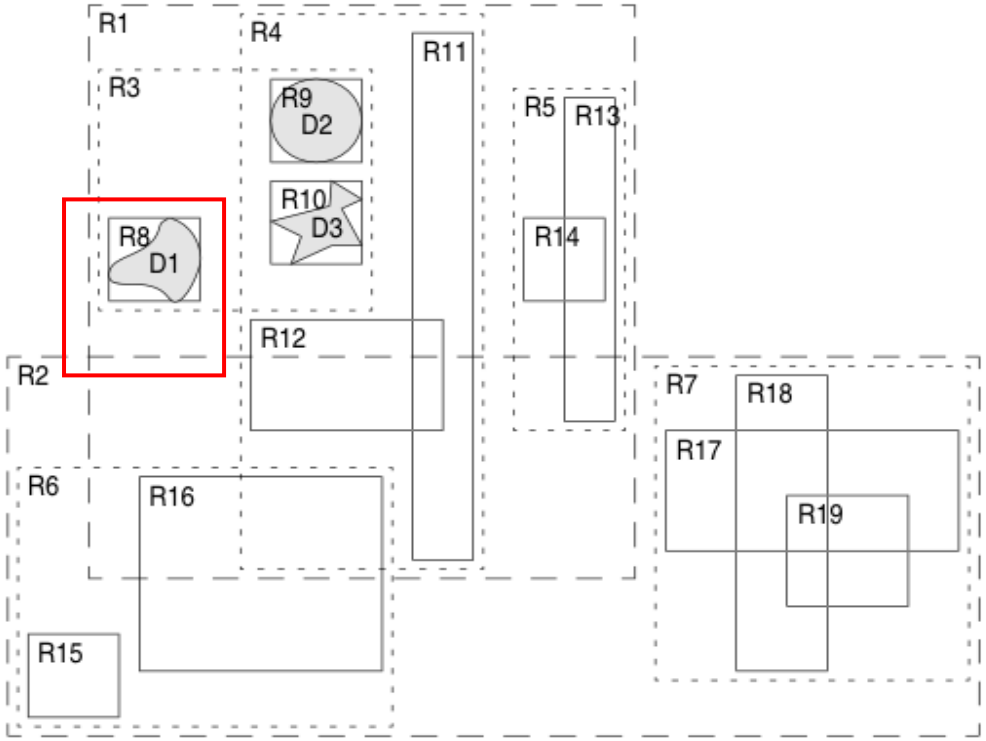



```
root@localhost [nyosm]>show create table nodes\G
*************************** 1. row ***************************
       Table: nodes
Create Table: CREATE TABLE `nodes` (
  `id` bigint(20) DEFAULT NULL,
  `geom` geometry NOT NULL,
  `user` varchar(50) DEFAULT NULL,
  `version` int(11) DEFAULT NULL,
  `timestamp` varchar(20) DEFAULT NULL,
  `uid` int(11) DEFAULT NULL,
  `changeset` int(11) DEFAULT NULL,
  `tags` text,
  UNIQUE KEY `i_nodeids` (`id`),
  SPATIAL KEY `i_geomidx` (`geom`),
  FULLTEXT KEY `tags` (`tags`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
1 row in set (0.00 sec)
```

GIS - InnoDB Spatial Indexes

- **R-tree based**
  - Full transactional support
  - Predicate locking to prevent phantoms
  - Records contain minimum bounding box
    - Small and compact
  - Currently only supports 2D data
    - We would like to add 3D support in the future
  - Supports historical spatial index DDL syntax

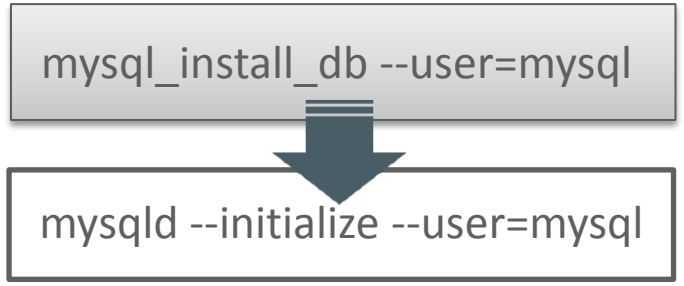




MySQL 5.7.7 セキュリティの強化

ユーザ管理とセキュリティ

- mysql_install_dbコマンド非推奨
  - mysqldの--initializeまたは--initialize-insecureオプションで初期化
- CREATE USER文とALTER USER文にオプション追加
  - SSL, PASSWORD EXPIRE, ACCOUNT [LOCK | UNLOCK]
- mysql.userテーブルのPassword列がauthentication_stringに変更に
- SET PASSWORD文およびPASSWORD()関数が非推奨
  - ALTER USER文での設定を推奨
- ENCRYPT, DES_ENCRYPT, DES_DECRYPT関数非推奨 AES推奨



Security - Encryption, Passwords, Installation

- AES 256 Encryption (Default in MySQL 5.7)
- パスワードローテーションポリシー
  - インスタンス全体、ユーザー単位で設定可能
- Deployment: デフォルトで安全に無人インストールを行う事が可能
  - インストール時にランダムなパスワードを設定/匿名のアカウントを削除
  - テストアカウント, スキーマ(test), デモファイルは作成されなくなりました

```
| ssl_type                | enum('','ANY','X509','SPECIFIED') | NO   |     |                     |       |
| ssl_cipher              | blob                              | NO   |     | NULL                |       |
| x509_issuer             | blob                              | NO   |     | NULL                |       |
| x509_subject            | blob                              | NO   |     | NULL                |       |
| max_questions           | int(11) unsigned                  | NO   |     | 0                   |       |
| max_updates             | int(11) unsigned                  | NO   |     | 0                   |       |
| max_connections         | int(11) unsigned                  | NO   |     | 0                   |       |
| max_user_connections    | int(11) unsigned                  | NO   |     | 0                   |       |
| plugin                  | char(64)                          | NO   |     | mysql_n             |       |
| authentication_string   | text                              | YES  |     | NULL                |       |
| password_expired        | enum('N','Y')                     | NO   |     | N                   |       |
| password_last_changed   | timestamp                         | YES  |     | NULL                |       |
| password_lifetime       | smallint(5) unsigned              | YES  |     | NULL                |       |
| account_locked          | enum('N','Y')                     | NO   |     | N                   |       |
+-------------------------+-----------------------------------+------+-----+---------------------+-------+
45 rows in set (0.01 sec)
```

[Global Configuration]

SET GLOBAL default_password_lifetime = 180;

[Individual user accounts]

ALTER USER joro@localhost PASSWORD EXPIRE INTERVAL 90 DAY;

ALTER USER joro@localhost PASSWORD EXPIRE DEFAULT;

ALTER USER joro@localhost PASSWORD EXPIRE NEVER;

Security – SSL, Proxy User

- SSL
  - Enabled by default
  - Auto-detection of existing keys and certs
  - Auto generation of keys and certs when needed
  - New helper utility: mysql_ssl_rsa_setup
- Extended Proxy User Support
  - Added Built-in Authentication Plugins support for Proxy Users
  - Allows multiple users to share a single set of managed privileges

Proxyユーザーは、MySQL5.5からサポートされている機能ですが、
追加の認証プラグインが必要でした。MySQL5.7.7RCからは、MySQL標準の
mysql_native_password とsha256_password認証をサポートする事で、
より柔軟にアカウント管理を行う事が出来るようになりました。

```
SET @@global.check_proxy_users = ON;
SET @@global.mysql_native_password_proxy_users = ON;
```

```
root@localhost [mysql]> CREATE USER proxy_base@localhost;
Query OK, 0 rows affected (0.00 sec)

root@localhost [mysql]> CREATE USER admin_1@localhost;
Query OK, 0 rows affected (0.00 sec)

root@localhost [mysql]> CREATE USER admin_2@localhost;
Query OK, 0 rows affected (0.00 sec)

root@localhost [mysql]> GRANT PROXY ON proxy_base@localhost TO admin_1@localhost;
Query OK, 0 rows affected (0.07 sec)

root@localhost [mysql]> GRANT PROXY ON proxy_base@localhost TO admin_2@localhost;
Query OK, 0 rows affected (0.00 sec)

root@localhost [mysql]> GRANT SELECT ON USER01.* TO proxy_base@localhost;
Query OK, 0 rows affected (0.00 sec)
```

```
[admin@misc01 ~]$ /usr/local/mysql/bin/mysql -u admin_1 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.7.7-rc-log MySQL Community Server (GPL)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

admin_1@localhost [(none)]> SELECT USER(), CURRENT_USER(), @@session.proxy_user;
+-------------------+----------------------+-----------------------+
| USER()            | CURRENT_USER()       | @@session.proxy_user  |
+-------------------+----------------------+-----------------------+
| admin_1@localhost | proxy_base@localhost | 'admin_1'@'localhost' |
+-------------------+----------------------+-----------------------+
1 row in set (0.00 sec)

admin_1@localhost [(none)]>  SHOW GRANTS;
+-----------------------------------------------------------+
| Grants for proxy_base@localhost                           |
+-----------------------------------------------------------+
| GRANT USAGE ON *.* TO 'proxy_base'@'localhost'            |
| GRANT SELECT ON `USER01`.* TO 'proxy_base'@'localhost'    |
+-----------------------------------------------------------+
2 rows in set (0.00 sec)

admin_1@localhost [(none)]>
```

まとめ

1	MySQL5.7はMySQL 5.6と比較しても大幅にパフォーマンスが向上しています。 Read Only (Point Select)で約2倍の処理性能。 将来的なデータやユーザー増加に対応可能な設定オプションも追加された事で、 既存のシステムリソースをより、有効活用する事が可能になりました。
2	General Tablespace, Temporary Tables,全文検索等のInnoDB関連の改良が 多く組み込まれていると同時に、オプティマイザーの性能も向上しているので、 それぞれの環境に応じて、柔軟にパフォーマンスの最適化を行う事が可能です。
3	運用においては、SYSスキーマを有効活用し、MySQLの状況を工数を殆どかけずに 柔軟に把握する事が可能な為、少ないコストでシステムをより安定稼働させる事が 可能になりました。また、MySQL5.6で追加されたオンラインDDL処理に関しても、 MySQL5.7でも新たに追加され、更にサービスダウンタイムを回避する事が可能です。

有難うございました