

MySQL 5.6 Performance Schema

梶山隆輔 / Ryusuke KAJIYAMA
MySQL Sales Consulting Senior Manager
MySQL Global Business Unit
March 12, 2015



ORACLE

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. |

Safe Harbor Statement

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Performance Schemaとは

- MySQL 5.5で導入された性能統計情報の分析のためのしくみ
- ストレージエンジン: 性能統計情報を記録
 - PERFORMANCE_SCHEMA
 - メモリ上にロックフリーハッシュを持つバッファを用意
- データベーススキーマ: 性能統計情報を表示
 - performance_schema
- インタフェース: MySQLサーバ内でシステムコールをラップ
 - 例) pthread_mutex_lock() -> mysql_mutex_lock()

ストレージエンジン or データベーススキーマ

```
mysql> SHOW DATABASES LIKE 'performance_schema'\G
***** 1. row *****
Database (performance_schema): performance_schema
1 row in set (0.01 sec)

mysql> SHOW ENGINES\G
<略>
***** 9. row *****
      Engine: PERFORMANCE_SCHEMA
      Support: YES
      Comment: Performance Schema
Transactions: NO
              XA: NO
      Savepoints: NO
9 rows in set (0.00 sec)
```

Performance Schemaとは

- MySQLサーバ内の「イベント」毎の処理時間を記録
 - MySQL サーバのソースコード中のinstrumentation pointにて計測
 - 処理時間はピコ秒単位での表示
 - ただし実際の精度はプラットフォームや設定による
- その他必要となる情報を記録
 - 処理データ量
 - ソースコードでの位置
 - 各種メタデータ

イベントの階層構造



- Instruments名やテーブル名は階層構造の各レベルに該当
- Session: 接続
- Transaction: トランザクション
*MySQL 5.7から
- Statement: SQL文
- Stage: SQL実行プロセスのステップ
- Wait: 処理実行待ち

セットアップ関連テーブル

- 各種の設定を動的に設定可能
- UPDATE文にて動的に変更可能

```
mysql> SHOW TABLES LIKE 'setup%';
+-----+
| Tables_in_performance_schema (setup%) |
+-----+
| setup_actors                          |
| setup_consumers                       |
| setup_instruments                     |
| setup_objects                         |
| setup_timers                          |
+-----+
5 rows in set (0.00 sec)
```

粗データ関連テーブル

- イベント、オブジェクト、インスタンスなど各instrumentsからの粗データを格納
- 粗データの現状および履歴を確認可能

```
mysql> SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES
-> WHERE TABLE_SCHEMA='performance_schema'
-> AND TABLE_NAME NOT LIKE 'setup%'
-> AND TABLE_NAME NOT LIKE '%summary%';
+-----+
| TABLE_NAME |
+-----+
| accounts |
| cond_instances |
| events_stages_current |
| events_stages_history |
| events_stages_history_long |
| events_statements_current |
| events_statements_history |
| events_statements_history_long |
| events_waits_current |
| events_waits_history |
| events_waits_history_long |
| file_instances |
| host_cache |
| hosts |
| mutex_instances |
| performance_timers |
| rwlock_instances |
| session_account_connect_attrs |
| session_connect_attrs |
| socket_instances |
| ... |
+-----+
22 rows in set (0.01 sec)
```


集計データ関連テーブル

- 各種の側面からデータ集計
- より簡単に稼働状況を確認可能

```
mysql> SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES
-> WHERE TABLE_SCHEMA='performance_schema'
-> AND TABLE_NAME LIKE '%summary%';
+-----+
| TABLE_NAME |
+-----+
| events_stages_summary_by_account_by_event_name |
| events_stages_summary_by_host_by_event_name |
| events_stages_summary_by_thread_by_event_name |
| events_stages_summary_by_user_by_event_name |
| events_stages_summary_global_by_event_name |
| events_statements_summary_by_account_by_event_name |
| events_statements_summary_by_digest |
| events_statements_summary_by_host_by_event_name |
| events_statements_summary_by_thread_by_event_name |
| events_statements_summary_by_user_by_event_name |
| events_statements_summary_global_by_event_name |
| events_waits_summary_by_account_by_event_name |
| events_waits_summary_by_host_by_event_name |
| events_waits_summary_by_instance |
| events_waits_summary_by_thread_by_event_name |
| events_waits_summary_by_user_by_event_name |
| events_waits_summary_global_by_event_name |
| file_summary_by_event_name |
| file_summary_by_instance |
| objects_summary_global_by_type |
| socket_summary_by_event_name |
...
+-----+
25 rows in set (0.01 sec)
```

Performance Schemaの設定

Performance Schemaの設定

3種類の設定方法

- 測定データに割り当てるメモリ量
 - オプションファイルにてパラメタを設定、動的変更は不可
- 起動時に測定項目(Instruments)をオン/オフ
 - オプションファイルにてパラメタを設定、MySQL 5.6から設定可能
- 動的に測定項目(Instruments)をオン/オフ
 - 各setupテーブルにて動的に変更

Performance Schemaの設定

メモリの設定

- Performance Schema関連のメモリ設定は2種類に分かれる
- 各イベントタイプのクラスとインスタンスをどれだけ追跡するか
 - Performance Schemaの全データに影響
- どれだけの集計と履歴を格納するか

クラス/インスタンスの設定

- クラス
Instruments実装をカウント
例) wait/io/file/sql/binlog
- インスタンス
サーバ稼働時の実体をカ
ウント
例) /u01/mysql/binlog.0001

```
mysql> SELECT VARIABLE_NAME, VARIABLE_VALUE
-> FROM INFORMATION_SCHEMA.GLOBAL_VARIABLES
-> WHERE VARIABLE_NAME LIKE 'PERF%CLASSES'
-> OR VARIABLE_NAME LIKE 'PERF%INSTANCES'
-> ORDER BY VARIABLE_NAME;
```

VARIABLE_NAME	VARIABLE_VALUE
PERFORMANCE_SCHEMA_MAX_COND_CLASSES	80
PERFORMANCE_SCHEMA_MAX_COND_INSTANCES	3504
PERFORMANCE_SCHEMA_MAX_FILE_CLASSES	50
PERFORMANCE_SCHEMA_MAX_FILE_INSTANCES	7693
PERFORMANCE_SCHEMA_MAX_MUTEX_CLASSES	200
PERFORMANCE_SCHEMA_MAX_MUTEX_INSTANCES	15906
PERFORMANCE_SCHEMA_MAX_RWLOCK_CLASSES	40
PERFORMANCE_SCHEMA_MAX_RWLOCK_INSTANCES	9102
PERFORMANCE_SCHEMA_MAX_SOCKET_CLASSES	10
PERFORMANCE_SCHEMA_MAX_SOCKET_INSTANCES	322
PERFORMANCE_SCHEMA_MAX_STAGE_CLASSES	150
PERFORMANCE_SCHEMA_MAX_STATEMENT_CLASSES	168
PERFORMANCE_SCHEMA_MAX_TABLE_INSTANCES	12500
PERFORMANCE_SCHEMA_MAX_THREAD_CLASSES	50
PERFORMANCE_SCHEMA_MAX_THREAD_INSTANCES	402

15 rows in set (0.01 sec)

クラス/インスタンスの設定

- ステータス変数の* lostが増加するようであれば、設定を変更する
- クラスの設定値の変更が必要になるケースは少ないが、インスタンスは変更を検討するケースも

```
mysql> SELECT VARIABLE_NAME, VARIABLE_VALUE
-> FROM INFORMATION_SCHEMA.GLOBAL_STATUS
-> WHERE VARIABLE_NAME LIKE 'PERF%CLASSES_LOST'
-> OR VARIABLE_NAME LIKE 'PERF%INSTANCES_LOST'
-> ORDER BY VARIABLE_VALUE;
```

VARIABLE_NAME	VARIABLE_VALUE
PERFORMANCE_SCHEMA_COND_CLASSES_LOST	0
PERFORMANCE_SCHEMA_SOCKET_INSTANCES_LOST	0
PERFORMANCE_SCHEMA_MUTEX_CLASSES_LOST	0
PERFORMANCE_SCHEMA_THREAD_CLASSES_LOST	0
PERFORMANCE_SCHEMA_SOCKET_CLASSES_LOST	0
PERFORMANCE_SCHEMA_FILE_INSTANCES_LOST	0
PERFORMANCE_SCHEMA_TABLE_INSTANCES_LOST	0
PERFORMANCE_SCHEMA_RWLOCK_INSTANCES_LOST	0
PERFORMANCE_SCHEMA_FILE_CLASSES_LOST	0
PERFORMANCE_SCHEMA_STATEMENT_CLASSES_LOST	0
PERFORMANCE_SCHEMA_RWLOCK_CLASSES_LOST	0
PERFORMANCE_SCHEMA_COND_INSTANCES_LOST	0
PERFORMANCE_SCHEMA_STAGE_CLASSES_LOST	0
PERFORMANCE_SCHEMA_MUTEX_INSTANCES_LOST	0
PERFORMANCE_SCHEMA_THREAD_INSTANCES_LOST	0

```
15 rows in set (0.01 sec)
```

履歴件数の設定

- タイプ毎の行数を設定
- *HISTORY SIZEはスレッド毎の値
- *HISTORY LONG SIZEはサーバ全体の値
- オプションファイルのみで設定可能

```
mysql> SELECT VARIABLE_NAME, VARIABLE_VALUE
-> FROM INFORMATION_SCHEMA.GLOBAL_VARIABLES
-> WHERE VARIABLE_NAME LIKE 'PERF%'
-> AND VARIABLE_NAME NOT LIKE '%CLASSES'
-> AND VARIABLE_NAME NOT LIKE '%INSTANCES'
-> ORDER BY VARIABLE_NAME;
```

VARIABLE_NAME	VARIABLE_VALUE
PERFORMANCE_SCHEMA	ON
PERFORMANCE_SCHEMA_ACCOUNTS_SIZE	100
PERFORMANCE_SCHEMA_DIGESTS_SIZE	10000
PERFORMANCE_SCHEMA_EVENTS_STAGES_HISTORY_LONG_SIZE	10000
PERFORMANCE_SCHEMA_EVENTS_STAGES_HISTORY_SIZE	10
PERFORMANCE_SCHEMA_EVENTS_STATEMENTS_HISTORY_LONG_SIZE	10000
PERFORMANCE_SCHEMA_EVENTS_STATEMENTS_HISTORY_SIZE	10
PERFORMANCE_SCHEMA_EVENTS_WAITS_HISTORY_LONG_SIZE	10000
PERFORMANCE_SCHEMA_EVENTS_WAITS_HISTORY_SIZE	10
PERFORMANCE_SCHEMA_HOSTS_SIZE	100
PERFORMANCE_SCHEMA_MAX_FILE_HANDLES	32768
PERFORMANCE_SCHEMA_MAX_TABLE_HANDLES	4000
PERFORMANCE_SCHEMA_SESSION_CONNECT_ATTRS_SIZE	512
PERFORMANCE_SCHEMA_SETUP_ACTORS_SIZE	100
PERFORMANCE_SCHEMA_SETUP_OBJECTS_SIZE	100
PERFORMANCE_SCHEMA_USERS_SIZE	100

16 rows in set (0.00 sec)

setup_timer

- データ取得タイミングの精度を設定
- Instrumentsのタイプ毎に異なるタイマーを設定可能

```
mysql> SELECT * FROM setup_timers;
+-----+-----+
| NAME      | TIMER_NAME |
+-----+-----+
| idle      | MICROSECOND |
| wait      | CYCLE       |
| stage     | NANOSECOND  |
| statement | NANOSECOND  |
+-----+-----+
4 rows in set (0.01 sec)
```


performance_timers

- 利用可能なタイマーの一覧
- 利用の可否とどれだけのCPUサイクルのオーバーヘッドとなるかを表示
- 実際にオーバーヘッドとなるCPUサイクル数はここで表示された値の2倍

```
mysql> SELECT * FROM performance_timers\G
***** 1. row *****
      TIMER_NAME: CYCLE
      TIMER_FREQUENCY: 2299238805
      TIMER_RESOLUTION: 1
      TIMER_OVERHEAD: 30
***** 2. row *****
      TIMER_NAME: NANOSECOND
      TIMER_FREQUENCY: 1000000000
      TIMER_RESOLUTION: 1
      TIMER_OVERHEAD: 102
***** 3. row *****
      TIMER_NAME: MICROSECOND
      TIMER_FREQUENCY: 1000000
      TIMER_RESOLUTION: 1
      TIMER_OVERHEAD: 102
***** 4. row *****
      TIMER_NAME: MILLISECOND
      TIMER_FREQUENCY: 1037
      TIMER_RESOLUTION: 1
      TIMER_OVERHEAD: 107
***** 5. row *****
      TIMER_NAME: TICK
      TIMER_FREQUENCY: 99
      TIMER_RESOLUTION: 1
      TIMER_OVERHEAD: 2479
5 rows in set (0.00 sec)
```

setup_instruments

- 処理時間や待機時間を収集するMySQLサーバのソースコード内に設けられたinstrumentの設定
- 個別に有効無効や時間の記録の有無を設定
- UPDATE文で動的に変更可

```
mysql> SELECT * FROM setup_instruments;
```

NAME	ENABLED	TIMED
wait/synch/mutex/sql/TC_LOG_MMAP::LOCK_tc	NO	NO
wait/synch/mutex/sql/LOCK_des key file	NO	NO
wait/synch/mutex/sql/MYSQL_BIN_LOG::LOCK_commit	NO	NO
wait/synch/mutex/sql/MYSQL_BIN_LOG::LOCK_commit_queue	NO	NO
wait/synch/mutex/sql/MYSQL_BIN_LOG::LOCK_done	NO	NO
wait/synch/mutex/sql/MYSQL_BIN_LOG::LOCK_flush_queue	NO	NO
wait/synch/mutex/sql/MYSQL_BIN_LOG::LOCK_index	NO	NO
wait/synch/mutex/sql/MYSQL_BIN_LOG::LOCK_log	NO	NO
wait/synch/mutex/sql/MYSQL_BIN_LOG::LOCK_binlog_end_pos	NO	NO
wait/synch/mutex/sql/MYSQL_BIN_LOG::LOCK_sync	NO	NO
wait/synch/mutex/sql/MYSQL_BIN_LOG::LOCK_sync_queue	NO	NO
wait/synch/mutex/sql/MYSQL_BIN_LOG::LOCK_xids	NO	NO
wait/synch/mutex/sql/MYSQL_RELAY_LOG::LOCK_commit	NO	NO
wait/synch/mutex/sql/MYSQL_RELAY_LOG::LOCK_commit_queue	NO	NO
wait/synch/mutex/sql/MYSQL_RELAY_LOG::LOCK_done	NO	NO
wait/synch/mutex/sql/MYSQL_RELAY_LOG::LOCK_flush_queue	NO	NO
wait/synch/mutex/sql/MYSQL_RELAY_LOG::LOCK_index	NO	NO
wait/synch/mutex/sql/MYSQL_RELAY_LOG::LOCK_log	NO	NO
wait/synch/mutex/sql/MYSQL_RELAY_LOG::LOCK_sync	NO	NO
wait/synch/mutex/sql/MYSQL_RELAY_LOG::LOCK_sync_queue	NO	NO

setup_consumers

- どれだけの情報を取得するかの設定(現在のみ、履歴を含むなど)
- UPDATE文で動的に変更可
- イベントテーブルで各SQL文を示すのDigest(MD5ハッシュ値)が有効となっている

```
mysql> SELECT * FROM setup_consumers;
+-----+-----+
| NAME                                | ENABLED |
+-----+-----+
| events_stages_current               | NO      |
| events_stages_history               | NO      |
| events_stages_history_long          | NO      |
| events_statements_current           | YES     |
| events_statements_history           | NO      |
| events_statements_history_long      | NO      |
| events_transactions_current         | YES     |
| events_transactions_history         | NO      |
| events_transactions_history_long    | NO      |
| events_waits_current                | NO      |
| events_waits_history                | NO      |
| events_waits_history_long           | NO      |
| global_instrumentation              | YES     |
| thread_instrumentation              | YES     |
| statements_digest                   | YES     |
+-----+-----+
15 rows in set (0.00 sec)
```

setup_objects

- 監視対象とするテーブルなどのオブジェクトを設定
- システム系のスキーマなどはデフォルトで監視対象外
- UPDATE文で動的に変更可
- setup_actorsでは監視対象とするユーザアカウントを設定可能

```
mysql> SELECT * FROM setup_objects;
```

OBJECT_TYPE	OBJECT_SCHEMA	OBJECT_NAME	ENABLED	TIMED
EVENT	mysql	%	NO	NO
EVENT	performance_schema	%	NO	NO
EVENT	information_schema	%	NO	NO
EVENT	%	%	YES	YES
FUNCTION	mysql	%	NO	NO
FUNCTION	performance_schema	%	NO	NO
FUNCTION	information_schema	%	NO	NO
FUNCTION	%	%	YES	YES
PROCEDURE	mysql	%	NO	NO
PROCEDURE	performance_schema	%	NO	NO
PROCEDURE	information_schema	%	NO	NO
PROCEDURE	%	%	YES	YES
TABLE	mysql	%	NO	NO
TABLE	performance_schema	%	NO	NO
TABLE	information_schema	%	NO	NO
TABLE	%	%	YES	YES
TRIGGER	mysql	%	NO	NO
TRIGGER	performance_schema	%	NO	NO
TRIGGER	information_schema	%	NO	NO
TRIGGER	%	%	YES	YES

```
20 rows in set (0.00 sec)
```

Performance Schemaの設定

- オプションファイルで設定する場合 (MySQL 5.6以上)
- Instrumentの有効無効を設定
 - `performance_schema_instrument='instrument名=値'`
 - 値 = [on | true | 1] / [off | false | 0] / [counted] (回数計測のみ)
 - Instrument名にはワイルドカード利用可能 例) `wait/synch/mutex/%`
- Consumerの有効無効を設定
 - `performance_schema_consumer_consumer名=value`
 - 例) `performance_schema_consumer_events_stages_current=on`

Performance Schema ロードマップ

MySQL 5.5のPerformance Schema

- 稼働統計情報を取得するフレームワークとして設計
- まずは低レベルな稼働統計情報の収集に重点
 - 負荷に耐えうるかの確認および新規に開発されたため
- 性能へのオーバーヘッドに課題があり、デフォルトではオフ

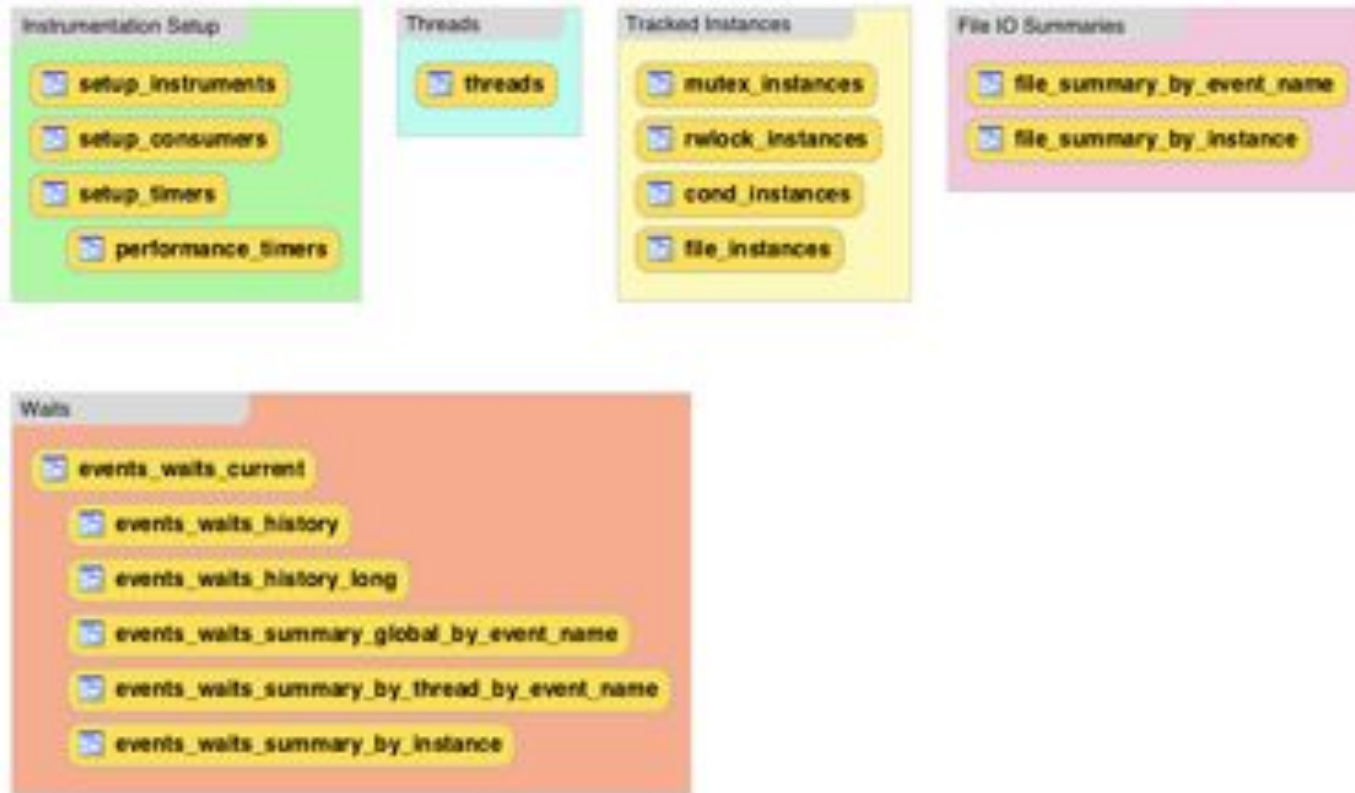
MySQL 5.5のPerformance Schema

- 17 Tables
- 222 Instruments



Instrument	Event Class
File IO	wait/io/file/%
Mutexes	wait/synch/mutex/%
Read/Write Locks	wait/synch/rwlock/%
Conditions	wait/synch/cond/%

MySQL 5.5 Performance Schema



MySQL 5.6のPerformance Schema

- 性能に関する課題を多数改善
- データベース管理者やアプリ開発者に役立つ情報の追加に重点
 - SQL文 / SQL文ダイジェスト
 - テーブルIO、インデックスIO、テーブルロック
 - ネットワークIO
- デフォルトでオン、ただしいくつかの詳細項目はオフ

MySQL 5.6のPerformance Schema

- StatementとStageイベント追加
- Statementには2種類のイベント
 - SQL文 `statement/sql/%`
 - COMコマンド `statement/com/%`
- Stageは各スレッドの内部処理ステップ
 - SQL文実行ステップ `stage/sql/%`

MySQL 5.6 Performance Schema

- 52 Tables (+35)
- 545 Instruments (+323)



Instrument Type	Event Class
Statements	statement/%
Stages	stage/%
Table IO	wait/io/table/%
Table Locks	wait/lock/table/%
Network IO	wait/io/socket/%
Idle Timing	idle

MySQL 5.6 Performance Schema



- メモリ利用状況
- メタデータロック
- レプリケーション設定 & 状況
- プリペアドステートメント
- トランザクション
- ストアドプロシージャ & ファンクション



MySQL 5.7.4 DMRのPerformance Schema

- 75 Tables (+23)
- 784 Instruments (+239)



Instrument Type	Event Class
Transactions	transaction
Memory	memory/%
Metadata Locks	wait/lock/metadata/%

MySQL 5.7.4 DMRのPerformance Schema



メモリ関連のInstruments

- 212のメモリ関連Instrumentsを追加 (5.7.4時点)
- 現在の利用量やそれまでの最高/最低利用量などを記録
- メモリ割り当てのレイテンシは記録しない
- 5つの新しいテーブル
 - memory_summary_by_account_by_event_name
 - memory_summary_by_host_by_event_name
 - memory_summary_by_thread_by_event_name
 - memory_summary_by_user_by_event_name
 - memory_summary_global_by_event_name

メモリ関連のInstruments – 全体サマリ

```
mysql> SELECT * FROM sys.memory_global_by_current_allocated\G
***** 1. row *****
      event_name: memory/performance_schema/internal_buffers
      current_count: 60
      current_alloc: 497.00 MiB
current_avg_alloc: 8.28 MiB
      high_count: 60
      high_alloc: 497.00 MiB
      high_avg_alloc: 8.28 MiB
***** 2. row *****
      event_name: memory/mysys/KEY_CACHE
      current_count: 3
      current_alloc: 8.00 MiB
current_avg_alloc: 2.67 MiB
      high_count: 3
      high_alloc: 8.00 MiB
      high_avg_alloc: 2.67 MiB
```

メモリ関連のInstruments – スレッド別サマリ

```
mysql> select * from memory_by_thread_by_current_allocated\G
***** 1. row *****
      user: sql/main
      current_count: 2407
      current_alloc: 10.89 MiB
current_avg_alloc: 4.63 KiB
current_max_alloc: 8.00 MiB
      total_allocated: 30.55 MiB
      thread_id: 1
***** 2. row *****
      user: mem@localhost
      current_count: 1914
      current_alloc: 1.50 MiB
current_avg_alloc: 824 bytes
current_max_alloc: 816.67 KiB
      total_allocated: 9.25 GiB
      thread_id: 4336
```

メモリ関連のInstruments – スレッド別詳細

```
mysql> SELECT event_name,  
->         sys.format_bytes(current_number_of_bytes_used) AS current_used  
-> FROM performance_schema.memory_summary_by_thread_by_event_name  
-> WHERE thread_id = 24  
-> ORDER BY current_number_of_bytes_used DESC;
```

event_name	current_used
memory/sql/Filesort_buffer::sort_keys	255.94 KiB
memory/sql/sp_head::main_mem_root	103.64 KiB
memory/mysys/IO_CACHE	64.05 KiB
memory/mysys/lf_dynarray	46.17 KiB
memory/mysys/array_buffer	24.20 KiB
memory/sql/thd::main_mem_root	23.95 KiB
memory/sql/String::value	16.13 KiB
memory/sql/TABLE	9.44 KiB
memory/sql/TABLE_SHARE::mem_root	8.70 KiB
memory/myisam/MI_INFO	7.07 KiB
memory/sql/THD::transactions::mem_root	4.02 KiB
memory/myisam/MYISAM_SHARE	3.29 KiB

SYSスキーマ

MySQL SYS Schema

Performance Schemaとインフォメーションスキーマをシンプルなビューに

- データベース管理者のタスクを支援
 - 稼働統計や成長率などの主要な統計値の監視
 - 性能問題の検出、診断および改善
- 状況の詳細の確認をシンプルに
 - IO量の高いファイルや処理
 - 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
 - レイテンシや待ち時間の分析
 - ロック
 - InnoDBの稼働統計



MySQL SYS Schema

- ps_helperから得られたフィードバックから改良
 - 80以上のビュー、自動更新、サーバーバージョン別
 - MySQL 5.5, 5.6, 5.7対応
- 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)
 - IBM DB2 SYSIBMカタログ
- Workbench 6.1から設定、またはGitHubからダウンロード可能
 - Workbenchには簡単に利用可能なレポート機能あり

MySQL SYS View

- 運用時に便利な各種のビュー
- performance_schemaとINFORMATION_SCHEMAのデータを表示
- 単位付きデータと単位無しデータの両方のビューを用意
 - x\$で始まるテーブルは単位無しのデータ
 - 単位無しのデータはツールでの利用に便利

x\$statement_analysis & statement_analysis

```
mysql> SELECT * FROM x$statement_analysis
-> WHERE query LIKE '%City%\G
***** 1. row
*****
      query: SELECT * FROM `City` WHERE
`CountryCode` = ?
      db: world
      full_scan:
      exec_count: 1
      err_count: 0
      rows_sent: 248
      rows_sent_avg: 248
      rows_examined: 248
      rows_examined_avg: 248
      tmp_tables: 0
      tmp_disk_tables: 0
      rows_sorted: 0
      sort_merge_passes: 0
      digest: 229d113f3761eff531ba57b9553d9964
      first_seen: 2015-03-06 21:43:05
      last_seen: 2015-03-06 21:43:05
1 row in set (0.00 sec)
```

```
total_latency: 13202612000
max_latency: 13202612000
avg_latency: 13202612000
lock_latency: 127000000
```

```
mysql> SELECT * FROM statement_analysis
-> WHERE query LIKE '%City%\G
***** 1. row
*****
      query: SELECT * FROM `City` WHERE
`CountryCode` = ?
      db: world
      full_scan:
      exec_count: 1
      err_count: 0
      rows_sent: 248
      rows_sent_avg: 248
      rows_examined: 248
      rows_examined_avg: 248
      tmp_tables: 0
      tmp_disk_tables: 0
      rows_sorted: 0
      sort_merge_passes: 0
      digest: 229d113f3761eff531ba57b9553d9964
      first_seen: 2015-03-06 21:43:05
      last_seen: 2015-03-06 21:43:05
1 row in set (0.01 sec)
```

```
total_latency: 13.20 ms
max_latency: 13.20 ms
avg_latency: 13.20 ms
lock_latency: 127.00 us
```

ユーザ/ホスト 集計ビュー

- ユーザ/ホスト関連の概要
- 集計カテゴリ
 - ファイルIO
 - Stage
 - SQL文

```
user_summary
user_summary_by_file_io
user_summary_by_file_io_type
user_summary_by_stages
user_summary_by_statement_latency
user_summary_by_statement_type
host_summary
host_summary_by_file_io
host_summary_by_file_io_type
host_summary_by_stages
host_summary_by_statement_latency
host_summary_by_statement_type
```

ファイルIO 集計ビュー

- スレッド毎のファイルIO
- ファイル別およびクラス別のバイト数およびレイテンシの集計

```
io_by_thread_by_latency  
io_global_by_file_by_bytes  
io_global_by_file_by_latency  
io_global_by_wait_by_bytes  
io_global_by_wait_by_latency
```

スキーマ分析ビュー

- オブジェクトの概要
- テーブルの利用統計
- インデックスの利用統計
- 未使用のインデックスやフルテーブルスキャンを行ったテーブルの表示

```
schema_index_statistics  
schema_object_overview  
schema_table_statistics  
schema_table_statistics_with_buffer  
schema_tables_with_full_table_scans  
schema_unused_indexes
```

「待ち」分析ビュー

- クラス別の「待ち」集計
- 「待ち」の詳細
 - ユーザ
 - ホスト
 - サーバ全体

```
wait_classes_global_by_avg_latency  
wait_classes_global_by_latency  
waits_by_host_by_latency  
waits_by_user_by_latency  
waits_global_by_latency
```

SQL文分析ビュー

- SQL文の概要
- 問題の起こりえるSQL文の抽出
 - エラーや警告
 - フルテーブルスキャン
 - ソートあり
 - 一時表作成
 - 遅延が95パーセンタイル超

`statement_analysis`

`statements_with_errors_or_warnings`

`statements_with_full_table_scans`

`statements_with_sorting`

`statements_with_temp_tables`

`statements_with_runtimes_in_95th_percentile`

プロセスリスト

- SHOW PROCESSLISTと比較して
 - mutex競合無し
 - リアルタイムで実行中のクエリを確認可能

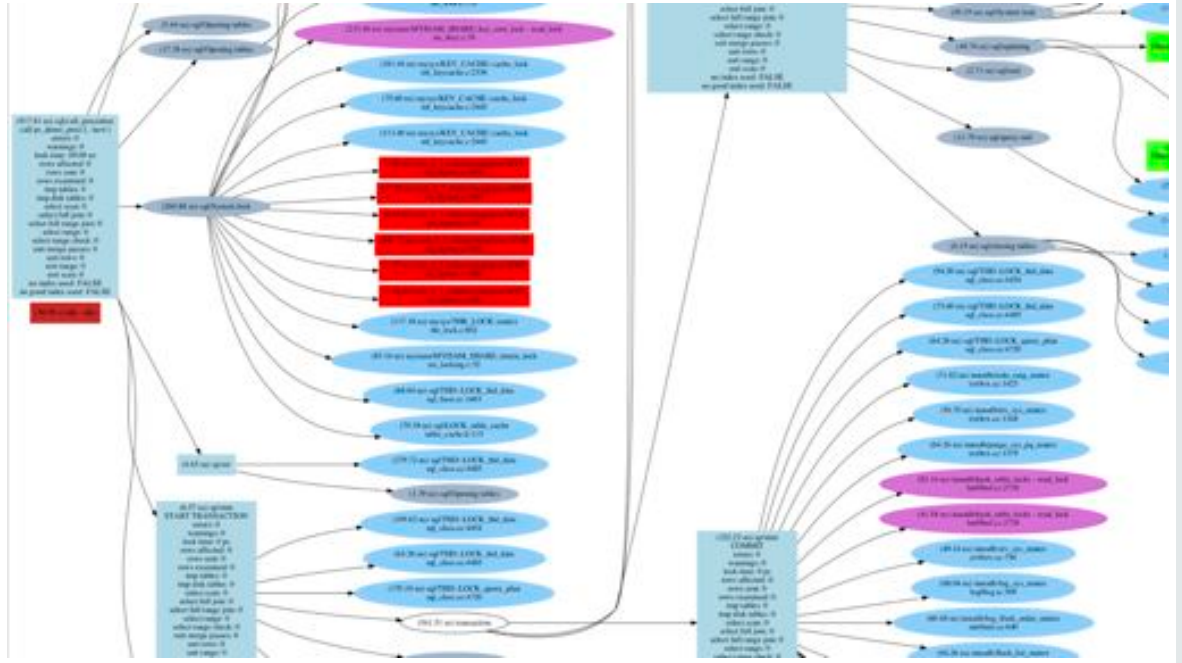
`processlist`

ps_trace_statement_digest プロシージャ

- 指定したSQL文を指定した時間だけリアルタイムで集計
 - SQL文稼働統計概要
 - 最も処理時間が長かった際の統計詳細
 - 実行計画
- SQL文の指定はdigestで行う

ps_trace_thread プロシージャ

- 指定したスレッドの詳細な稼働状況を収集
 - イベントの階層構造を収集
- “DOT”フォーマットで出力
 - 各種ツールにて描画可能
Graphvizなど



MySQL Workbench

localhost

Administration - Performance Reports

Performance Reports

Report

- Hot Spots for I/O
 - Top File I/O Activity Report
 - Top I/O by File by Time
 - Top I/O by Event Category
 - Top I/O in Time by Event Category
 - Top I/O Time by User/Thread
- High Cost SQL Statements
 - Statement Analysis
 - Statements in Highest 5 Percent b...
 - Using Temp Tables
 - With Sorting
 - Full Table Scans
 - Errors or Warnings
- Database Schema Statistics
 - Schema Object Overview (High Ov...
 - Schema Index Statistics
 - Schema Table Statistics
 - Schema Table Statistics (with Inno...
 - Tables with Full Table Scans
 - Unused Indexes
- Wait Event Times (Expert)
 - Global Waits by Time
 - Waits by User by Time
 - Wait Classes by Time
 - Waits Classes by Average Time
- InnoDB Statistics
 - InnoDB Buffer Stats by Schema

Statement Analysis

Lists statements with various aggregated statistics

Query	Full Table Scan Executed (N)	Errors (N)	Warnings (N)	Total Time (s)	Max Time
INSERT INTO t1 VALUES (...)	1574113	5	0	2991071965.70	310533
SELECT intcol1, charcol1 FROM t1	714360	28	0	2285362694.76	93782
CREATE TABLE 't1' (intcol1 INTEGER (7), charcol1 VAR...	7233	1	0	428166890.57	486906
DROP SCHEMA IF EXISTS 'mysqslap'	7231	0	0	15344778.32	52336
CREATE SCHEMA 'mysqslap'	7233	0	0	2089026.22	25052
SHOW GLOBAL STATUS	224	0	0	223946.12	17496
SHOW SCHEMAS	1	0	0	141164.53	141164
SELECT * FROM 'events_waits_summary_global_by_even...	7	0	0	54825.92	9746
SELECT @@ version_comment LIMIT 7	2	0	0	46168.69	45926
SHOW FULL TABLES FROM 'sys' WHERE 'Table_type' = ?	1	0	0	38448.93	38448
DROP SCHEMA IF EXISTS 'mysqslap'	2	0	0	27182.63	22306
SHOW 'TABLES'	1	1	0	23529.29	23526
SELECT * FROM 'events_waits_summary_global_by_even...	1	0	0	19300.62	19306
SHOW SCHEMAS	2	0	0	15573.32	15176
USE 'world'	2	0	0	15393.34	15346
SELECT 'EVENT_NAME', 'COUNT_STAR', 'SUM_TIMER...	1	0	0	10133.47	10133
SELECT * FROM 'world' . 'City'	1	0	0	7613.15	7613
SHOW FIELDS FROM 'world' . 'Country'	1	0	0	5825.12	5825
SELECT * FROM 'events_stages_summary_global_by_eve...	2	0	0	5433.83	5086
SHOW FIELDS FROM 'world' . 'City'	1	0	0	5244.54	5244
SELECT * FROM 'setup_instruments' WHERE NAME LIKE ?	4	0	0	4185.91	2283
SET SESSION TRANSACTION ISOLATION LEVEL REPEATAB...	4	0	0	3270.84	3115

Export... Copy Selected Copy Query Refresh

SQL Editor Opened

Hardware and Software Engineered to Work Together

ORACLE®