



基幹システムでも使える MySQL EE(商用版) 活用方法 第3回 MySQL Enterprise Monitor

MySQL Global Business Unit
Updated : 2015/09/17

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 Enterprise Monitor: Agenda

- 1: データベース安定運用における課題
- 2: MySQL Enterprise Monitorの活用
- 3: Query改善プロセス
- 4: MySQL設定課題・検知・確認
- 5: MySQL設定改善プロセス
- 6: まとめ

A dolphin is captured in mid-leap, emerging from the surface of the ocean. The dolphin's body is sleek and dark, with a lighter patch on its belly. It is surrounded by a spray of white water droplets and bubbles. The background is a clear, deep blue sky. The dolphin's reflection is visible in the calm water below the surface.

データベース安定運用における課題

MySQL DBA チェックリスト

1. 本番データベースが使用可能であることを確認。 ✓
2. 24x365 MySQLのパフォーマンスを監視 ✓
3. MySQLのレプリケーションが正常に動作していることを確認 ✓
4. バックアップが正常に完了したことを確認 ✓
5. MySQLのディスク容量が不足しないように監視 ✓
6. 定期的に監視し、ブロッキングの問題を特定 ✓
7. データベーススキーマ変更の有無を確認 ✓
8. OSにおける、異常なイベントを確認 ✓
9. セキュリティの脆弱性をチェックする ✓
10. メモリ使用状況を監視して分析 ✓

MySQL DBA チャレンジ

- 「データベースが遅いです。どのようなチューニングが必要ですか? 」
- 「最もコストが高いクエリはどれでしょうか? 」
- 「インデックスは、最適化されていますか? 」
- 「レプリケーションの遅延が問題になっていませんか? 」
- 「最後のバックアップは成功しましたか? 」
- 「いつ頃、ディスクがいっぱいになりそうでしょうか? 」
- 「いつ頃、スケールアウトに追加のハードウェアが必要になりますか? 」
- 「データベーススキーマは変わりましたか? 」
- 「何か、対応しなければならない、セキュリティの脆弱性がありますか? 」



対応例：クエリパフォーマンス確認

MySQLのパフォーマンスモニタリングオプションやツールを利用して確認

```
+-----+-----+
| Variable_name | Value |
+-----+-----+
| slow_query_log | ON |
| slow_query_log_file | slow.log |
+-----+-----+
```

```
+-----+-----+
| Variable_name | Value |
+-----+-----+
| long_query_time | 0.500000 |
+-----+-----+
```

```
+-----+-----+
| Variable_name | Value |
+-----+-----+
| log_queries_not_using_indexes | ON |
+-----+-----+
```

```
[root@GA01 performance]#
/usr/local/mysql/bin/mysqldumpslow -s t -t 3
/usr/local/mysql/data/slow.log
```

```
Reading mysql slow query log from
/usr/local/mysql/data/slow.log
Count: 4124 Time=0.00s (18s) Lock=0.00s (2s)
Rows=0.0 (46), admin[admin]@3hosts
SELECT * FROM mysql.backup_progress WHERE
backup_progress.backup_id > 'S' AND
backup_progress.current_time > 'S'
```

```
Count: 4124 Time=0.00s (17s) Lock=0.00s (1s)
Rows=0.0 (11), admin[admin]@3hosts
SELECT * FROM mysql.backup_history WHERE
backup_history.backup_Id > 'S' AND
backup_history.end_time > 'S'
```

その他: SHOW [FULL] PROCESSLIST , FLUSH STATUS; <run query>; SHOW STATUS;

対応例：データベースイベント確認

エラーログなどを確認し、データベースにおけるエラーや警告を確認

```
[root@GA01 performance]# cat /usr/local/mysql/data/GA01.err
150413 21:56:37 mysqld_safe Starting mysqld daemon with databases from /usr/local/mysql/data
2015-04-13 21:56:37 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Please
  use --explicit_defaults_for_timestamp server option (see documentation for more details).
2015-04-13 21:56:37 0 [Note] /usr/local/mysql/bin/mysqld (mysqld 5.6.24-enterprise-
  commercial-advanced) starting as process 2569 ...
2015-04-13 21:56:37 2569 [Note] Plugin 'FEDERATED' is disabled.
2015-04-13 21:56:37 2569 [Note] InnoDB: Using atomics to ref count buffer pool pages
2015-04-13 21:56:37 2569 [Note] InnoDB: The InnoDB memory heap is disabled
2015-04-13 21:56:37 2569 [Note] InnoDB: Mutexes and rw_locks use GCC atomic builtins
```

```
[root@GA01 performance]# cat /var/log/messages | grep Warning
Sep  7 10:02:24 GA01 kernel: Warning: Intel CPU model - this hardware has not undergone
  upstream testing. Please consult http://wiki.centos.org/FAQ for more information
Sep  7 12:58:43 GA01 kernel: Warning: Intel CPU model - this hardware has not undergone
  upstream testing. Please consult http://wiki.centos.org/FAQ for more information
Sep  7 21:59:50 GA01 kernel: Warning: Intel CPU model - this hardware has not undergone
  upstream testing. Please consult http://wiki.centos.org/FAQ for more information
```

MySQL Enterprise Monitorは、MySQLに特化したモニタリングツールで、サーバステータス、クエリパフォーマンス、データベースにおけるイベントを統合監視する事が可能なツールです。

また、MySQLに最適なパラメータやセキュリティ設定などもアドバイスしてくれる為、MySQLを最適な設定に保つ事が出来、データベース管理工数を削減する事が見込まれます。



サーバーパフォーマンス

MySQLデータベースパフォーマンス

SQLステートメントパフォーマンス

MySQLデータベースイベント監視

MySQLデータベースパラメータ監視

MySQLデータベースセキュリティ監視

その他、クラスター、レプリケーション等

A dolphin is captured in mid-leap, emerging from the surface of a clear blue ocean. The dolphin's body is sleek and dark, with water droplets glistening on its skin. The background is a deep, vibrant blue, suggesting a clear sky and water. The dolphin's reflection is visible in the water below.

MySQL Enterprise Monitorの活用 運用の最適化とシステムの安定稼働

MySQL Enterprise Edition (MySQL Enterprise Monitor)

- ビジネス・クリティカルな環境において、最高レベルのMySQLスケーラビリティ、セキュリティ、信頼性、アップタイムを実現し、ビジネス・クリティカルな環境においてリスクとコストを削減

Performance

- パフォーマンスと拡張性
- モニタリング・チューニング



TCO

- ダウンタイム(機会損失)回避
- 生産性の向上(サポート)
- ROIの最適化をサポート



DevOps Agility

- スケールと管理の自動化
- 柔軟で高速なバックアップ



Peace of Mind

- 保険としての保守
- 必要な時に迅速なサポート



Risks

- セキュリティ,コンプライアンス
- MySQLチームからのサポート



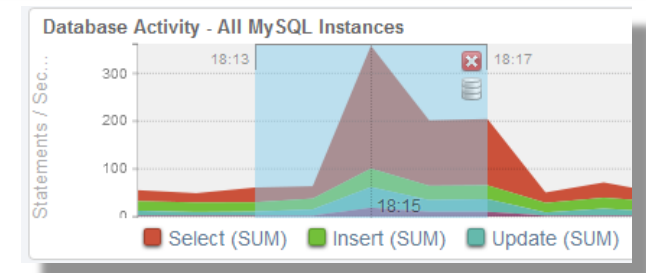
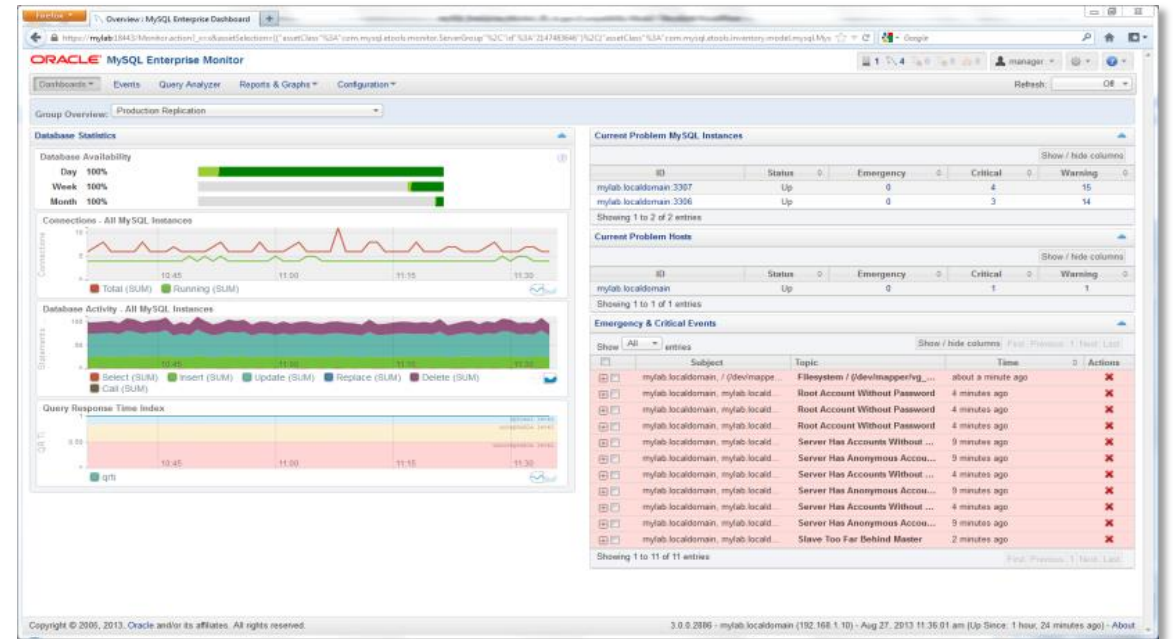
Customer Satisfaction

- MySQL導入の最適化
- サービスパフォーマンス
- サービス可用性



MySQL Enterprise Monitor

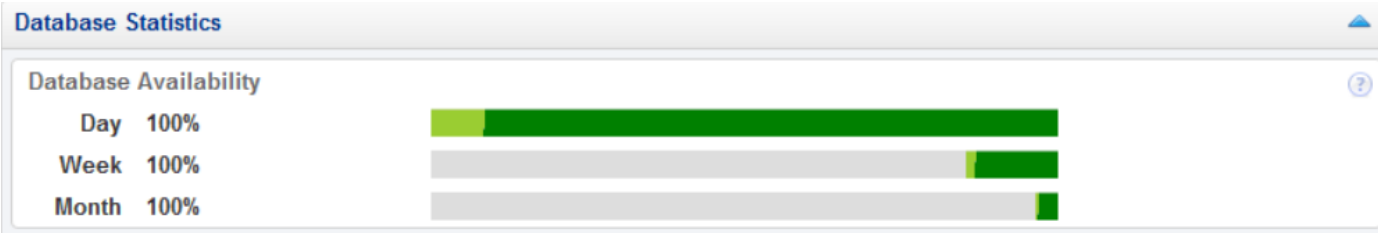
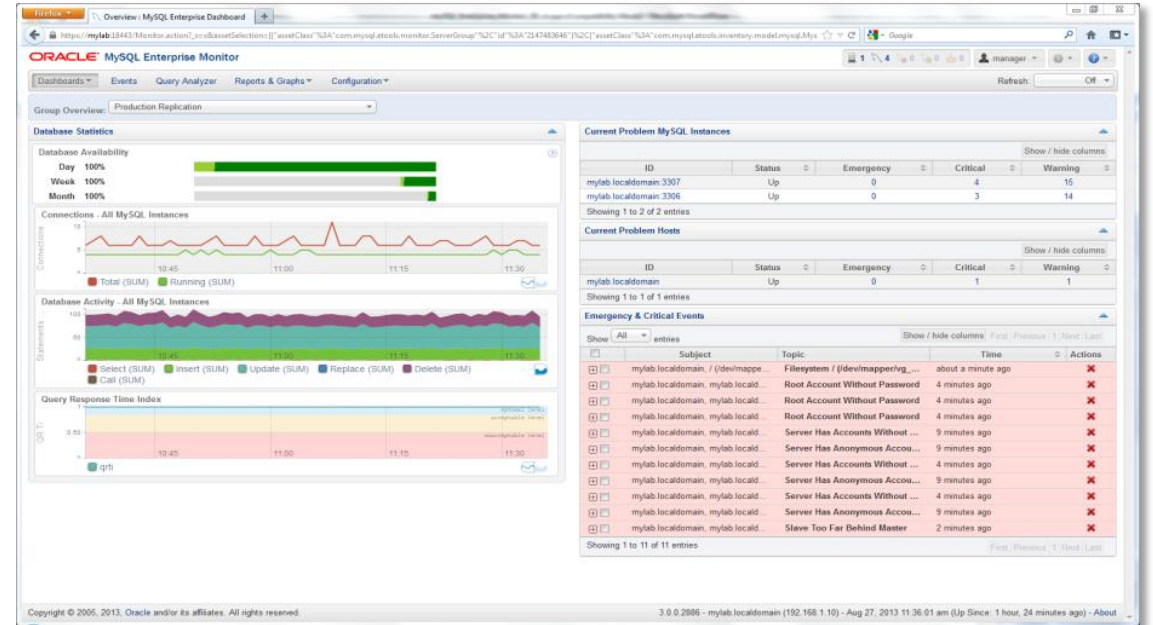
- パフォーマンスと可用性の監視
- リアルタイム監視およびアラート生成
- 問題のあるSQL文の検知
- ディスク監視と容量プランニング
- MySQL設定の確認とアドバイス
- クラウド対応アーキテクチャ
 - ポリシーベースの設定
 - エージェント導入不要
- MySQL監視を10分以内で開始可能
- レプリケーションの監視
- バックアップの監視等



	Current	Worst	Subject	Topic
<input type="checkbox"/>	🔴	🔴	mylab.localdomain, mylab.localdomain:3306	Root Account Without Password
<input type="checkbox"/>	🔴	🔴	mylab.localdomain, mylab.localdomain:3306	Server Has Accounts Without A Password
<input type="checkbox"/>	🟢	🔴	mylab.localdomain, mylab.localdomain:3306	Average Statement Execution Time Excess...
<input type="checkbox"/>	🟢	🔴	mylab.localdomain, mylab.localdomain:3306	SQL Statement Generates Errors or Warnings
<input type="checkbox"/>	🔴	🔴	mylab.localdomain, mylab.localdomain:3306	Server Has Anonymous Accounts
<input type="checkbox"/>	🟢	🔴	mylab.localdomain, mylab.localdomain:3306	MySQL Instance Is Experiencing A Query P...
<input type="checkbox"/>	🟡	🟡	mylab.localdomain, mylab.localdomain:3306	InnoDB Log Buffer Flushed To Disk After Ea...
<input type="checkbox"/>	🟡	🟡	mylab.localdomain, mylab.localdomain:3306	User Has Rights To Database That Does Not...

Enterprise Monitor Dashboard

- サービスレベルのモニタリング
- リアルタイムパフォーマンス監視
- 警告と通知による迅速な対応
- ベストプラクティスアドバイザー
- 全てのMySQLサーバを視覚的に管理



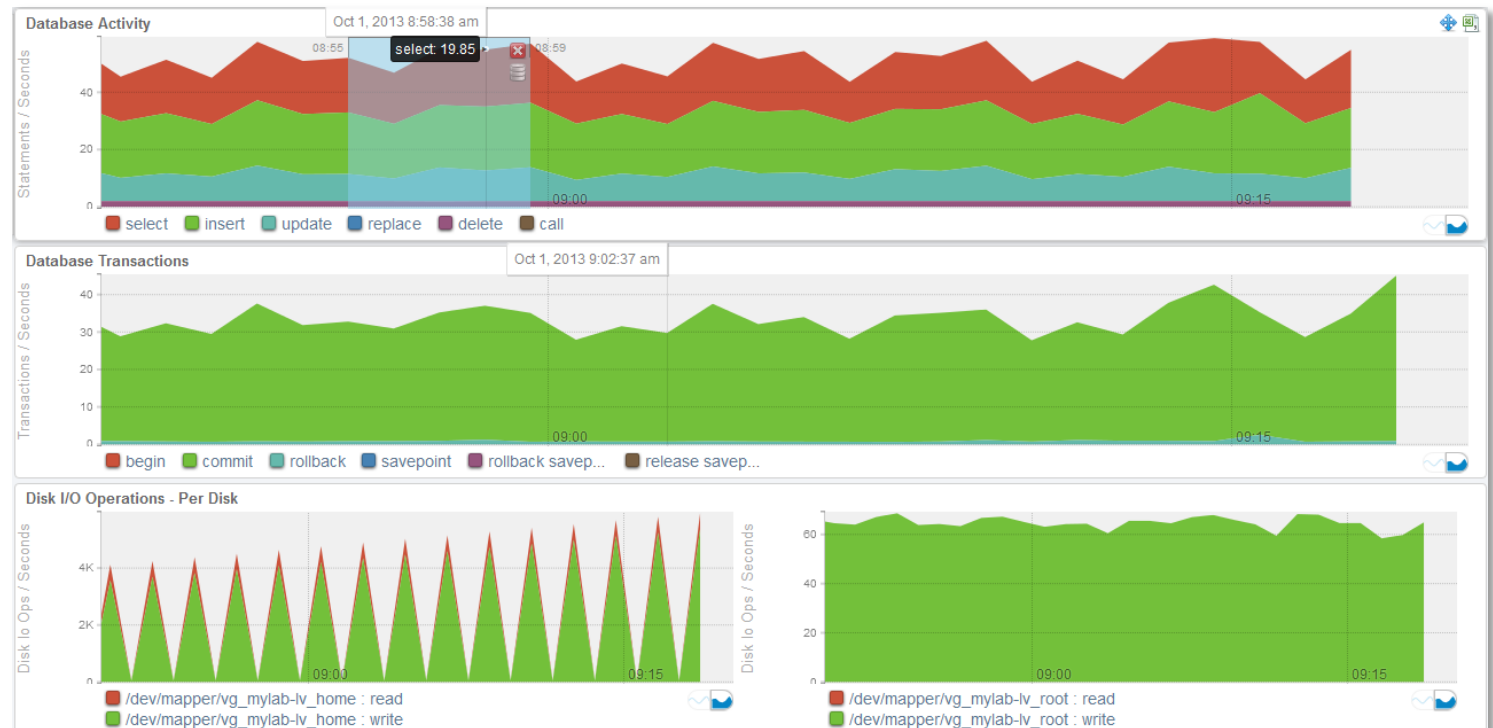
"The MySQL Enterprise Monitor is an absolute must for any DBA who takes his work seriously."

- Adrian Baumann, System Specialist
Federal Office of Information Technology &
Telecommunications

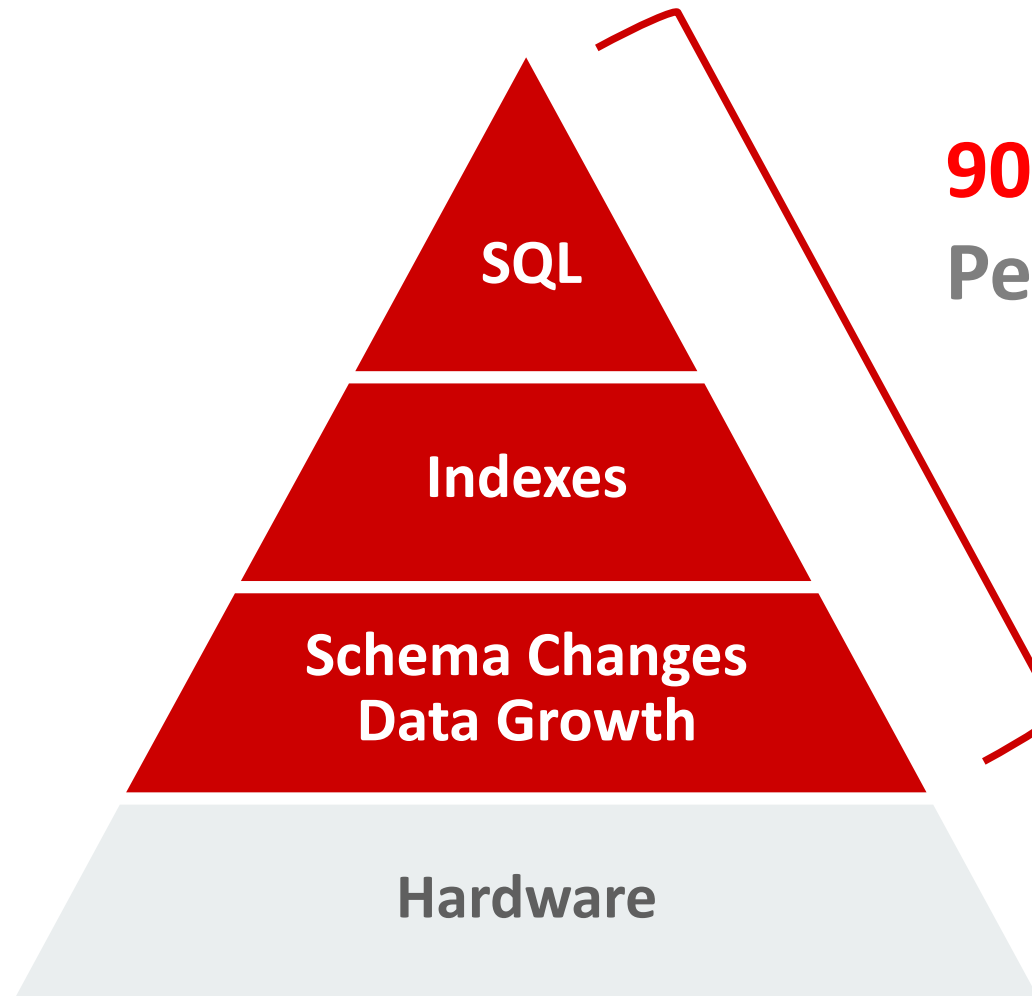


Performance Monitoring

- MySQLパフォーマンスのモニタリング
- OSパフォーマンスのモニタリング
- クエリーパフォーマンスのモニタリング
- インデックス利用をアドバイス
- 問題発生時の警報

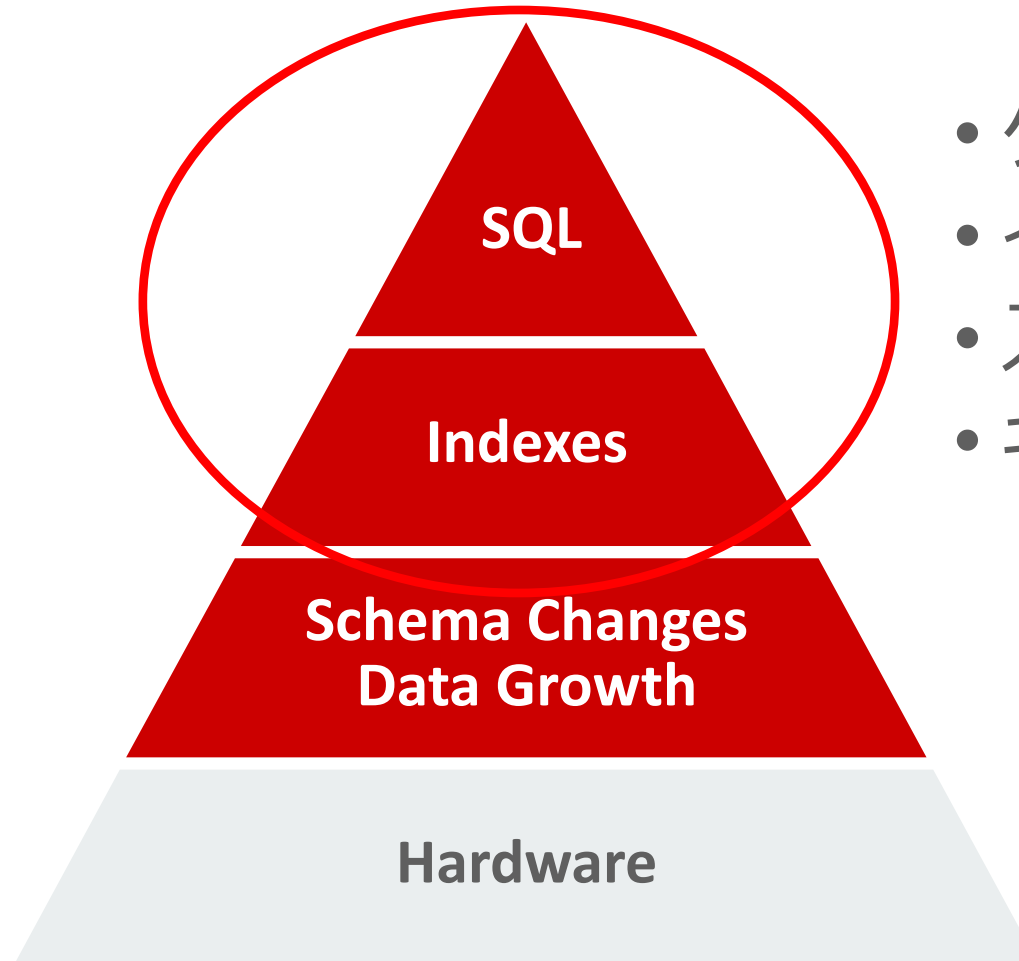


データベース・パフォーマンスの問題の原因



90% of
Performance Problems

データベース・パフォーマンスの問題の原因



- クエリーチューニング
- インデックスの追加
- スキーマのチューニング
- キャッシュヒット率の改善

クエリパフォーマンスの問題の解決ステップ

MySQL Enterprise Monitor

MySQL Workbench & CLI

- 視覚的にスロークエリを特定
- 相関グラフ
- クエリ応答時間指数 (QRTi)
- 実行統計

Step1

Step2

Step3

Step4

- クエリチューニング
- インデックスの追加
- スキーマのチューニング
- キャッシュヒット率の改善

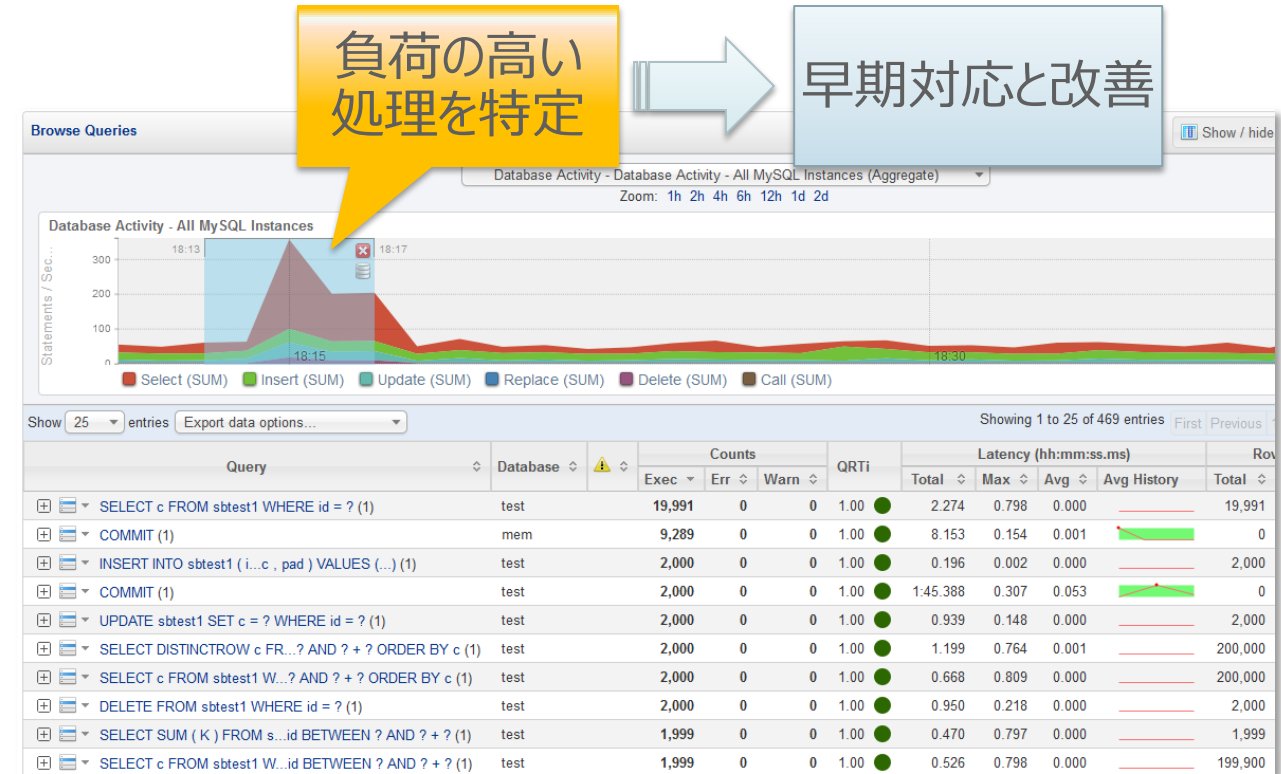
- MySQL 実行プラン(Explain)
- サンプルクエリ
- クエリグラフ

パフォーマンスの向上

Enterprise Query Analyzer

Step1

- 全クエリーのリアルタイム統合監視
- パフォーマンスの可視化
- コストの大きいクエリーの特定
- クエリー統計詳細の確認
- Query Response Time index (QRTi)
 - クエリーサービスレベル指針
 - サーバー、インスタンスのサービスレベル
 - クエリーパフォーマンス指標



Advantages of the Query Analyzer **over** Slow Query Log

Step1

- 実行統計を参照
- 原因となるアプリケーションの特定
- 一定期間の間、全体的なクエリーのパフォーマンスを確認
- 該当のクエリーが発生した時間軸で確認可能
- 実行計画を参照可能
- 特定クエリー、特定ホスト、期間に焦点を置いて確認可能
- 他のパフォーマンスグラフと特定クエリーの相関性を確認可能

Query Response Time Index (QRTi)

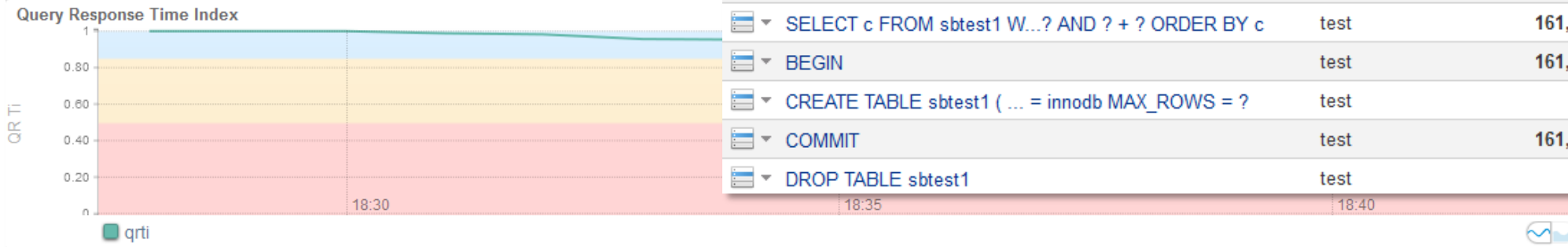
Step1

- 各クエリの「サービス品質」(QoS) を測定
- サーバ、グループ、またはすべてのインスタンスのQoS測定
- クエリパフォーマンス確認の為の単一測定基準

Query Response Time

Green (Optimum)	< 100ms
Yellow (Acceptable)	100ms < 400ms
Red (Unacceptable)	400ms <

Query	Database	Counts			QRTi
		Exec	Err	Warn	
INSERT INTO sbtest1 (i...c , pad) VALUES (...)	test	161,133	0	0	1.00
INSERT INTO sbtest1 (k...LUES (...)/* , ... */	test	20	0	0	0.50
SELECT SUM (K) FROM s...id BETWEEN ? AND ? + ?	test	161,156	0	0	1.00
SELECT DISTINCTROW c FR...? AND ? + ? ORDER BY c	test	161,160	0	0	1.00
CREATE INDEX k_1 ON sbtest1 (k)	test	1	0	0	0.00
SELECT c FROM sbtest1 W...? AND ? + ? ORDER BY c	test	161,157	0	0	1.00
BEGIN	test	161,139	0	0	1.00
CREATE TABLE sbtest1 (... = innodb MAX_ROWS = ?	test	1	0	0	0.50
COMMIT	test	161,091	0	0	0.69
DROP TABLE sbtest1	test	1	0	0	0.50



参照 : Query Response Time index (QRTi)

<http://dev.mysql.com/doc/mysql-monitor/3.0/en/mem-features-qrti.html>

Missing Indexes

Step1

Indexが無い為、テーブル全体のデータを処理している。
メモリーでソート処理出来ない場合は、ディスクのTemp Tableで
処理する為パフォーマンスが落ちる原因になります。

Temp Tables	
Total	Disk %
30	100
36	100

クエリを参照

Show 10 entries データのエキスポートオプション

Showing 1 to 10 of 647 entries First Previous 1 2 3 4 5 Next Last

クエリ	データベース	警告	カウント			QRTI	待ち時間 (hh:mm:ss.ms)				行数	
			実行	エラー	警告		合計	最高	平均	平均値の履歴	合計	平均
SELECT COUNT (*) AS ...RE `state` = ? LIMIT ? (1)	mysql		22	0	0	0.86	2.384	7.688	0.108		22	
SELECT `plugin_name` FR...ORDER BY `plugin_name` (1)	mysql		26	0	0	0.79	3.498	6.254	0.135		754	
SELECT `plugin_status` ...ugin_name` = ? LIMIT ? (1)	mysql		2	0	0	0.00	6.862	5.589	3.431		1	
COMMIT (1)	mem		15,423	0	0	1.00	1:33.520	4.259	0.006		0	
UPDATE `mem__inventory` ...e` = ? WHERE `hid` = ? (1)	mem		897	0	0	0.99	14.739	4.136	0.016		897	
SELECT GROUP_CONCAT (`...in_status` = ? LIMIT ? (1)	mysql		1	0	0	0.00	4.022	4.022	4.022		1	
UPDATE `mem__inventory` ...p` = ? WHERE `hid` = ? (1)	mem		25	0	0	0.92	4.029	3.949	0.161		25	
DELETE FROM `mem__quan` ...timestamp` < ? LIMIT ? (1)	mem		22	0	0	0.95	3.954	3.949	0.180		0	
UPDATE `mem__inventory` ...p` = ? WHERE `hid` = ? (1)	mem		1	0	0	0.00	3.942	3.942	3.942		1	
UPDATE `mem__inventory` ...p` = ? WHERE `hid` = ? (1)	mem		93	0	0	0.97	0.977	3.942	0.011		93	

データのエキスポートオプション

First Previous 1 2 3 4 5 Next Last

New, Un-optimized Statements

新規追加若しくは実行されたクエリーの為、まだ最適化されていない状態

Step1

クエリを参照 Show / hide columns

Show 10 entries データのエキスポートオプション... Showing 1 to 10 of 450 entries First Previous 1 2 3 4 5 Next Last

クエリ	データベース	警告	カウント			QRTI	待ち時間 (hh:mm:ss.ms)				行数		Temp Tables		初回実行
			実行	エラー	警告		合計	最高	平均	平均値の履歴	合計	平均	Total	Disk %	
BY TIMESTAMP (1)	mem	1	0	0	1.00	0.001	0.001	0.001		28	28	3	0	10:23:59	
BY TIMESTAMP (1)	mem	1	0	0	1.00	0.001	0.001	0.001		28	28	3	0	10:23:59	
BY TIMESTAMP (1)	mem	1	0	0	1.00	0.001	0.001	0.001		28	28	3	0	10:23:59	
BY TIMESTAMP (1)	mem	1	0	0	1.00	0.001	0.001	0.001		28	28	3	0	10:23:59	
er_id' = ? (1)	mem	3	0	0	1.00	0.006	0.005	0.002		3	1	0	0	10:15:04	
63_', ... (1)	mem	4	0	0	1.00	0.010	0.013	0.002		8	2	0	0	2015/03/01 22:03:10	
= ?) (1)	mem	1	0	0	1.00	0.001	0.017	0.001		2	2	0	0	2015/03/01 22:03:09	
es` AS ... (1)	mem	2	0	0	1.00	0.002	0.180	0.001		44	22	0	0	2015/03/01 21:48:21	
... (1)	mem	6	0	0	0.92	0.220	0.461	0.037		6	1	0	0	2015/03/01 21:42:33	
ctions` ... (1)	mem	6	0	0	0.92	0.227	0.672	0.038		6	1	0	0	2015/03/01 21:42:33	

データのエキスポートオプション... First Previous 1 2 3 4 5 Next Last

High Rate of Low Latency Statements

高い確率で遅延が発生しているクエリーの特徴

Step1

クエリを参照

Show / hide columns

Show 10 entries データのエクスポ... Showing 1 to 10 of 448 entries First Previous 1 2 3 4 5 Next Last

クエリ	データベース	警告	カウント			QRTi	待ち時間 (hh:mm:ss.ms)				行数		Temp
			実行	エラー	警告		合計	最高	平均	平均値の履歴	合計	平均	
UES (...)(1)	mem		8,476	0	0	0.95	4:28.917	1.660	0.032		8,545	1	0
	mem		23,397	0	0	1.00	1:39.170	4.259	0.004		0	0	0
een') (1)	mem		8,491	5	0	1.00	59.801	1.150	0.007		16,989	2	0
, ...)(1)	mem		493	0	0	0.90	19.218	1.071	0.039		493	1	0
dTotal') AS ...)(1)	mem		48	0	0	0.50	12.150	2.220	0.253		480	10	96
DM (SELECT ...)(1)	mem		48	0	0	0.50	10.050	0.542	0.209		480	10	96
E `hid` = ? (1)	mem		1,334	0	0	1.00	9.737	4.136	0.007		1,334	1	0
E `id` = ? (1)	mem		662	0	0	0.98	8.684	3.277	0.013		663	1	0
ble') (1)	mem		29	0	0	0.71	5.071	0.778	0.175		29	1	0
_ ...)(1)	mem		66	0	0	0.89	3.538	1.248	0.054		66	1	0

データのエクスポ...

First Previous 1 2 3 4 5 Next Last

Query改善プロセス



CLI: 問題のあるクエリを解決 – EXPLAIN;

EXPLAIN文でオプティマイザの解析結果を確認

Step2

```
explain select * from
world.CountryLanguage
where CountryCode = 'JPN'¥G

***** 1. row *****
      id: 1
  select_type: SIMPLE
        table: CountryLanguage
         type: ref
possible_keys: PRIMARY, CountryCode
         key: PRIMARY
    key_len: 3
         ref: const
        rows: 6
   Extra: Using where
1 row in set (0.00 sec)
```

解析

- インデックスがどのように使用されているか?
- ファイルソートが必要だったか?
- どのテーブル、カラムがクエリで使用されているか?

修正/チューニング – 以下の作業を繰り返し行う:

- インデックスを追加/変更
- テーブル定義、データ型など変更
- クエリ構造を変更

CLI: EXPLAINの各項目

- テーブルごとに1行出力される。
 - id... クエリのID (テーブルのIDではないので注意)
 - select_type... クエリの種類を表す
 - table... 対象のテーブル
 - type... レコードアクセスタイプ。どのようにテーブルにアクセスされるかを示す。
 - possible_keys... 利用可能なキー。
 - key/key_len... 選択されたキーとその長さ。
 - rows... 行数の概算見積もり。
 - Extra... オプティマイザヒント。

CLI: 問題のあるクエリを解決 – EXPLAIN EXTENDED;

EXTENDED キーワードを付けて使用すると、出力にfiltered カラムが含まれます。

Step2

```
[world]> explain extended select * from world.CountryLanguage where CountryCode = 'JPN'¥G show warnings¥G
***** 1. row *****
      id: 1
  select_type: SIMPLE
        table: CountryLanguage
         type: ref
possible_keys: PRIMARY, CountryCode
         key: PRIMARY
    key_len: 3
         ref: const
         rows: 6
  filtered: 100.00
    Extra: Using where
1 row in set, 1 warning (0.00 sec)
***** 1. row *****
  Level: Note
   Code: 1003
Message: /* select#1 */ select `world`.`CountryLanguage`.`CountryCode` AS `CountryCode`,
`world`.`CountryLanguage`.`Language` AS `Language`, `world`.`CountryLanguage`.`IsOfficial` AS `IsOfficial`,
`world`.`CountryLanguage`.`Percentage` AS `Percentage` from `world`.`CountryLanguage`
where (`world`.`CountryLanguage`.`CountryCode` = 'JPN')
1 row in set (0.00 sec)
```

テーブル条件によってフィルタ処理されるテーブル行の推定の割合

Warningには、オプティマイザが SELECT ステートメント内のテーブルおよびカラム名をどのように修飾するか、書き換えおよび最適化ルールの適用後に SELECT がどのように見えるか、および場合によって最適化プロセスに関するその他のメモが表示されます。

GUI: MySQL 5.6 + MySQL Workbench (Visual Explain)

Step2

JOINの順番も一目で確認可能
オブジェクトへのアクセスパターンを色で識別

- **赤色**(confirm)
 - ALL(full table scan)
- **橙色**(confirm)
 - Full index scan
 - Full Text Index Search
- **緑色**(good)
 - Range (>,<, between)
 - Reference
- **青色**(Best)
 - EQ_REF

MySQL Workbench

```
1 SELECT CONCAT(customer.last_name, ', ', customer.first_name) AS customer,  
2 address.phone, film.title  
3 FROM rental INNER JOIN customer ON rental.customer_id = customer.customer_id  
4 INNER JOIN address ON customer.address_id = address.address_id  
5 INNER JOIN inventory ON rental.inventory_id = inventory.inventory_id  
6 INNER JOIN film ON inventory.film_id = film.film_id
```

Query Plan | Raw Explain Data | Explain

Display Cost Info: []

Switch EXPLAIN views

Switch between 'Result' and 'Explain' views

query_block #1

nested loop 1 row

nested loop 1 row

nested loop 1 row

nested loop 2 rows

1000 rows

Full Table Scan film

Non-Unique Key Lookup film

Non-Unique Key Lookup film

Unique Key Lookup customer PRIMARY

Unique Key Lookup address PRIMARY

Result 1 Explain

Query Completed

Hover for additional information

Access Type: ALL
Full Table Scan
Cost Hint: Very High - very costly for large tables (not so much for small ones).
No usable indexes were found for the table and the optimizer must search every row.
Consider adding an index.

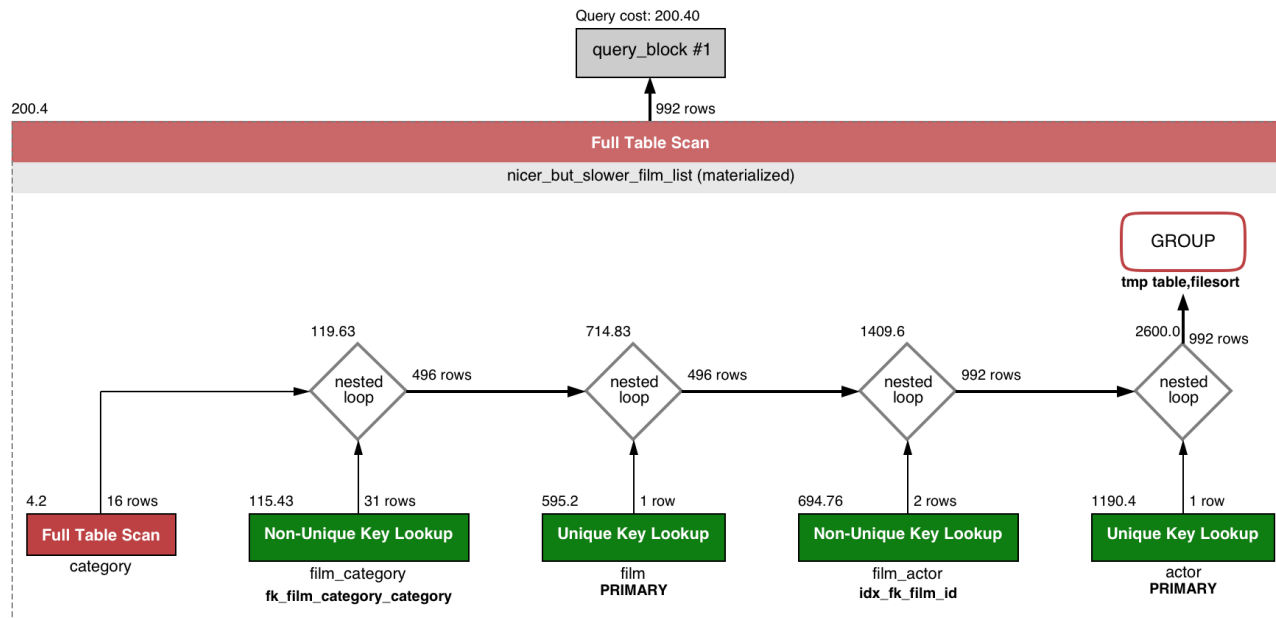
Key/Index: -
Possible Keys: PRIMARY

Rows Examined per Scan: 1000
Rows Produced per Join: 1000
Filtered (ratio of rows produced per rows examined): 100%
Hint: 100% is best, <= 1% is worst
A low value means the query examines a lot of rows that are not returned.

参考) MySQL5.7 - Visual and 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"
    ],
    ...
  }
}
```


MEM: 問題のあるクエリを解決 – EXPLAIN;

MySQL Enterprise Monitor上のEXPLAINでオプティマイザの解析結果を確認

Step2

クエリ情報 クエリ例 **EXPLAIN クエリ** グラフ

期間内のクエリのEXPLAIN出力(通常最も遅い)。

説明

id	select_type	table	type	possible_keys	key	key_len	ref	rows	extra
1	SIMPLE	Country	ALL	PRIMARY	null	null	null	239	Using where; Using temporary; Using filesort
1	SIMPLE	City	eq_ref	PRIMARY	PRIMARY	4	world.Country.Capital	1	Using index
1	SIMPLE	CountryLanguage	ref	PRIMARY,CountryCode	CountryCode	3	world.Country.Code	2	Using index

Close

上級フィルタを表示
フィルタの適用 デフォルトとし

Show 10 entries データのエク

クエ

+ EXPLAIN SELECT @@SE

+ EXPLAIN SELECT LANGU

+ SET character_set_results = ?(2)

SELECT Language, COUNT(... BY LANGUAGE LIMIT ?,10(2))

+ SET `character_set_results` = ?(2)

+ /* mysql-connector-java...olation, @@wait_timeout

+ SELECT @@SESSION . `aut...ation , @@wait_timeout (2)

+ SELECT VALUE FROM `mysq...entory` WHERE NAME = ? (2)

+ SELECT SYSTEM_USER () (2)

+ SELECT `connection_id` () (2)

world 6 0 0 1.00 ● 0.018 0.006 0.003

world 6 0 0 1.00 ● 0.030 0.008 0.005

0.008 0.001 0.000

0.014 0.007 0.002

world 45 0 0 1.00 ● 0.015 0.002 0.000

world 45 0 0 1.00 ● 0.013 0.001 0.000

world 45 0 0 1.00 ● 0.009 0.001 0.000

world 45 0 0 1.00 ● 0.009 0.001 0.000

SELECT Language, COUNT(?) FROM CountryLanguage JOIN Country ON CountryLanguage.CountryCode...

※ [MySQL Enterprise Monitor Proxy and Aggregator](#)若しくは、[MySQL Enterprise Plugin for Connector/J](#) が必要です。

select_type:クエリの種類を把握し対応

Step2

- select_typeでクエリの構造を確認

- JOIN

- SIMPLE

- UNION

- UNION, UNION RESULT

- サブクエリ

- FROM句のサブクエリ... PRIMARY, DERIVED
- その他... PRIMARY, SUBQUERY, DEPENDENT SUBQUERY, UNCACHEABLE SUBQUERYなど。
- サブクエリがUNIONになっている場合... DEPENDENT UNION, UNCACHEABLE UNION

クエリの種類に応じて
チューニングを実施

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	CountryLanguage	ref	PRIMARY,CountryCode	PRIMARY	3	const	6	Using where

type: レコードアクセス

Step2

- const・・・PRIMARY/UNIQUEによるルックアップ
- system・・・レコードか1行のテーブル
- ALL・・・テーブルスキャン
- index・・・インデックススキャン
- eq_ref・・・PRIMARY/UNIQUEによるJOIN
- ref・・・ユニークでないインデックスによる等価比較
- ref_or_null・・・ユニークでないインデックスによる等価比較とIS NULLのOR
- range・・・範囲検索
- fulltext・・・全文検索
- index_merge・・・2つの異なるインデックスをAND/OR
- unique_subquery・・・サブクエリ内でPRIMARY/UNIQUEでルックアップ
- index_subquery・・・サブクエリ内でref

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	CountryLanguage	ref	PRIMARY, CountryCode	PRIMARY	3	const	6	Using where

Extra: オプティマイザヒント

Step2

- Using where... テーブルから行がフェッチされた後、WHERE句の条件によってさらに絞り込まれる
- Using index... インデックスだけを使ってクエリを解決出来ることを示す
- Using filesort... ファイルソート（クイックソート）によってソートをする
- Using temporary... クエリを解決するのにテンポラリテーブルが必要
- Using where with pushed condition... engine condition pushdownを利用
- Using index for group-by... MIN()またはMAX()によって集計を行う際、クエリがインデックスだけを用いて解決できる
- Distinct... JOINにおいてDISTINCTによって一意な結果を生成しなければならない場合、内部表のキーでDISTINCTを解決出来る
- Range checked for each record (index map: N)... 遅いJOIN。ほぼフルJOINに近い
- Not exists... LEFT JOINにおいて、内部表にマッチする行が存在しないレコードだけを探したいことを示す
- Using join buffer... JOINバッファが利用されていることを示す

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	CountryLanguage	ref	PRIMARY,CountryCode	PRIMARY	3	const	6	Using where

対応: Queryの見直し

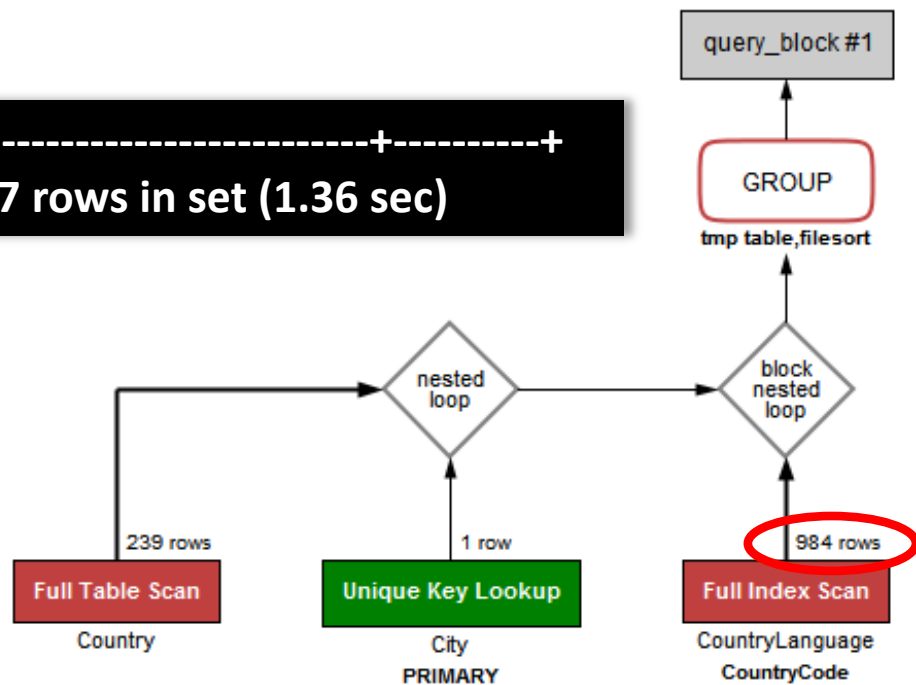
Queryの条件が不足していないか? 書き方や順序を変えてみると?

Step3

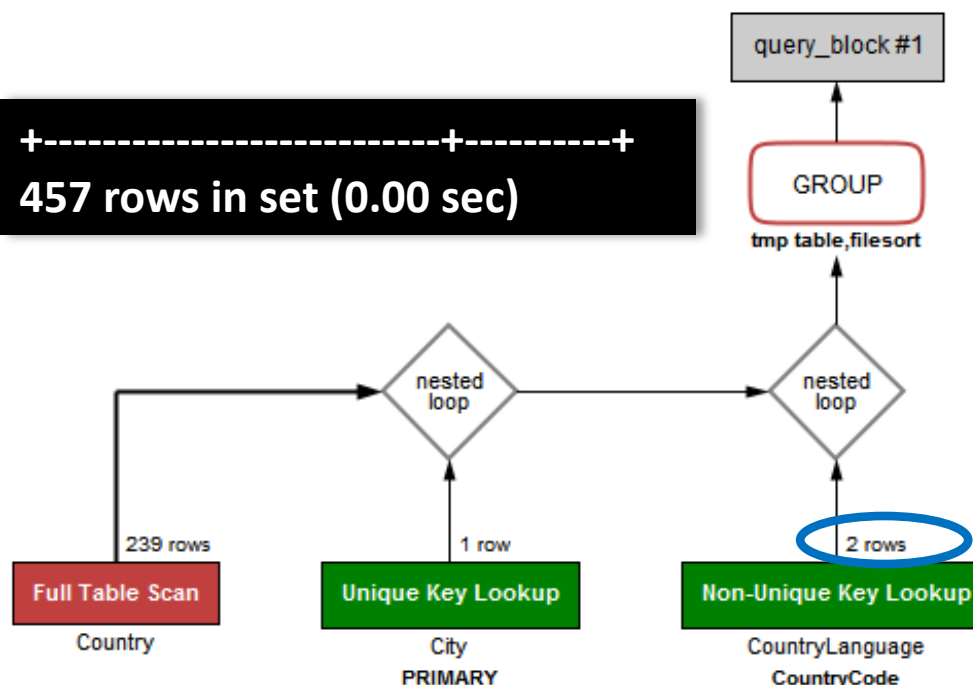
```
SELECT Language, COUNT(1) FROM  
CountryLanguage JOIN Country WHERE Capital IN  
(SELECT Id FROM City)GROUP BY LANGUAGE;
```

```
SELECT Language, COUNT(1) FROM  
CountryLanguage JOIN Country ON  
CountryLanguage.CountryCode =  
Country.CodeWHERE Capital IN  
(SELECT Id FROM City) GROUP BY LANGUAGE;
```

+-----+-----+
457 rows in set (1.36 sec)



+-----+-----+
457 rows in set (0.00 sec)



対応: インデックス

Step3

- インデックスは参照時の性能は向上するが、更新時はオーバーヘッド
 - 小さなインデックス、プレフィックス index(name(8))
- MySQLはインデックス内で順序が先の列のみ利用可能
 - key (a,b) where b=5 はインデックスを使わない
- インデックスは必要最小限に留めること
 - 例) 性別に対するインデックス (ビットマップ・インデックス)
- ユニークなインデックスにはUNIQUE キーワードをつける
- 重複するようなインデックスの利用は避けるべき
 - key(a, b) があるなら key(a) は削除
- BTREE インデックスはソートされた結果を返す
 - select * from t where b=5 order by c ... key(b,c) optimal
- “covering indexes”は高速、行のデータのフェッチが不要
 - select c from t where b=5 ...の場合、key(b,c) が良い
- OPTIMIZE TABLEやALTER TABLE ... でインデックスのソートと最適化

対応: SQLオプティマイザの制御

Step3

- SELECT STRAIGHT_JOIN * from tbl1,tbl2 ...
 - SQL文に書かれたテーブルの順に処理を行う
- USE INDEX / FORCE INDEX / IGNORE INDEX
 - SELECT * FROM Country USE INDEX(PRIMARY)
 - ヒント句、MySQLでの利用ケースはあまり多くない
 - インデックスを強制的に使わせるケースはある
- ANALYZE TABLE ...
 - 通常はあまり必要とされないが、大量にデータの更新があった後などに実行することも

```
root@localhost [world]> explain select * from City force index(CountryCode) where CountryCode = 'JPN' and ID = 1769;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | type | possible_keys | key          | key_len | ref          | rows | Extra          |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1  | SIMPLE      | City  | const | CountryCode   | CountryCode | 7        | const,const | 1    | Using where   |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

対応: データ型選択時における注意点

- 出来るだけ小さいサイズの型を使う。
 - 本当にその文字列長は必要か？
 - 文字コードを工夫（日本語が不要ならlatin1やascii、binaryを利用する）
 - ENUMの活用。文字列だが数値として格納される。例) ENUM('YES','NO')
- PROCEDURE ANALYSE()の活用
 - 格納されているデータを分析
 - 最適なデータ型を提案

```
CREATE TABLE `Country` (  
  `Code` char(3) NOT NULL DEFAULT "",  
  `Name` char(52) NOT NULL DEFAULT "",  
  `Continent` enum('Asia','Europe','North America','Africa','Oceania','Antarctica','South America')  
  NOT NULL DEFAULT 'Asia',  
  `Region` char(26) NOT NULL DEFAULT "",
```

マルチカラムインデックスの活用

- MySQLのオプティマイザが同時に利用出来るインデックスはひとつだけ
 - インデックスマージが効く場面は限定的
 - インデックスマージより遙かに高速
- 複数の検索条件、ソート条件がある場合はマルチカラムインデックスが有効

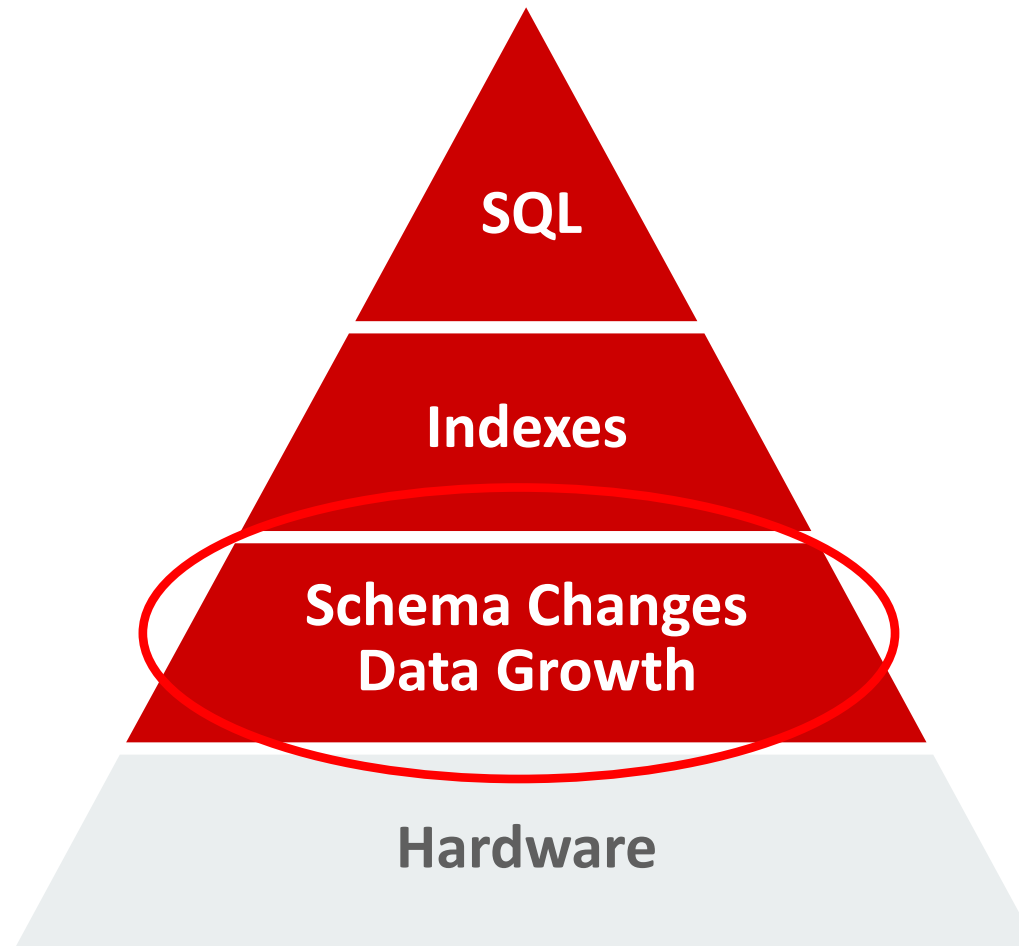
```
CREATE TABLE `CountryLanguage` (  
  `CountryCode` char(3) NOT NULL DEFAULT "",  
  `Language` char(30) NOT NULL DEFAULT "",  
  `IsOfficial` enum('T','F') NOT NULL DEFAULT 'F',  
  `Percentage` float(4,1) NOT NULL DEFAULT '0.0',  
  PRIMARY KEY (`CountryCode`,`Language`),  
  KEY `CountryCode` (`CountryCode`),
```

Covering Index

- インデックスだけでクエリを解決出来るようにすること
 - データへアクセスしないのでとても高速
 - EXPLAINコマンドのExtraフィールドは「Using Index」
 - これを目的として多くのインデックスを作成しすぎると更新性能は劣化するため注意が必要
- `SELECT col1, col2 WHERE col1 = x ORDER BY col2;`

(col1, col2)というインデックスがあればデータへのアクセスは不要

データベース・パフォーマンスの問題の原因



- パーティショニング
- スキーマデザイン
- I/O処理分散

MySQLにおけるパーティショニング

Step3

- パーティショニングとは？
 - データを特定のカラムの値によって分類
 - それぞれにデータ、インデックスを持つ
- 5.1から導入
 - 現行は同一のストレージエンジンを使用したもののみをサポート
 - 同じテーブル内に違うストレージエンジンのパーティションを持つことは出来ない。
 - 5.7ではhandlerの処理を改善し、多くのパーティション環境でメモリー消費を抑制
- パーティションのタイプは4つ
 - Range ... カラム値の範囲を指定。
 - List ... カラム値をリストアップ。
 - Hash ... 数値カラムのハッシュ値で分ける。
 - Key ... 文字列カラムのハッシュ値で分ける。

MySQLにおけるパーティショニング (例)

Step3

```
CREATE TABLE `tbl_partition` (  
  `member_id` varchar(40) NOT NULL,  
  `platform` varchar(10) NOT NULL,  
  `year` smallint(5) unsigned NOT NULL,  
  `month` tinyint(2) unsigned NOT NULL,  
  `updated_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY (`member_id`, `year`, `month`),  
  KEY `idx_tbl_partition` (`year`),  
  KEY `idx_tbl_partition_id` (`member_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
/*!50100 PARTITION BY RANGE (`year`)  
(PARTITION p2013 VALUES LESS THAN (2013) ENGINE = InnoDB,  
PARTITION p2014 VALUES LESS THAN (2014) ENGINE = InnoDB,  
PARTITION p2015 VALUES LESS THAN (2015) ENGINE = InnoDB,  
PARTITION p2016 VALUES LESS THAN (2016) ENGINE = InnoDB,  
PARTITION p2017 VALUES LESS THAN (2017) ENGINE = InnoDB,  
PARTITION p2018 VALUES LESS THAN (2018) ENGINE = InnoDB,  
PARTITION p2019 VALUES LESS THAN (2019) ENGINE = InnoDB,  
PARTITION p2020 VALUES LESS THAN (2020) ENGINE = InnoDB,  
PARTITION pmax VALUES LESS THAN MAXVALUE ENGINE = InnoDB) */
```

MySQLにおけるパーティショニング (例)

データサイズが多いテーブルで、インデックスが効率よく利用出来ていない場合等は、パーティション化する事で読みこむデータ量を減らす事が出来、パフォーマンスが大きく改善する可能性があります。

```
root@localhost [test]> root@localhost [test]> explain partitions select * from tbl_partition where year = 2015;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	tbl_partition	p2016	ref	idx_tbl_partition	idx_tbl_partition	2	const	1	NULL

```
-rw-rw----. 1 mysql mysql 131072 8月 5 10:06 tbl_partition#P#p2013.ibd
-rw-rw----. 1 mysql mysql 131072 8月 5 10:06 tbl_partition#P#p2014.ibd
-rw-rw----. 1 mysql mysql 131072 8月 5 10:06 tbl_partition#P#p2015.ibd
-rw-rw----. 1 mysql mysql 131072 8月 5 10:06 tbl_partition#P#p2016.ibd
-rw-rw----. 1 mysql mysql 131072 8月 5 10:06 tbl_partition#P#p2017.ibd
-rw-rw----. 1 mysql mysql 131072 8月 5 10:06 tbl_partition#P#p2018.ibd
-rw-rw----. 1 mysql mysql 131072 8月 5 10:06 tbl_partition#P#p2019.ibd
-rw-rw----. 1 mysql mysql 131072 8月 5 10:06 tbl_partition#P#p2020.ibd
-rw-rw----. 1 mysql mysql 131072 8月 5 10:06 tbl_partition#P#pmax.ibd
-rw-rw----. 1 mysql mysql 8712 8月 5 10:06 tbl_partition.frm
-rw-rw----. 1 mysql mysql 84 8月 5 10:06 tbl_partition.par
```

スキーマのデザイン

Step3

- **正規化**

- OLTPや書き込み多い環境
- データの冗長性を削減
- JOINのオーバーヘッド
- トータルのデータサイズは小さくなる
- E/R図とスムーズに連携

- **非正規化**

- OLAPやレポーティング
- データの冗長性が高い
- JOINを削減できる

- **データ型**

- tinyint, smallint, mediumint, などでデータ量を削減
- JOINに使う列は同じデータ型に
- char(64)ではなくvarchar(64)
- 可能なところはNOT NULLを宣言
- varchar(255)ではなくvarchar(64)

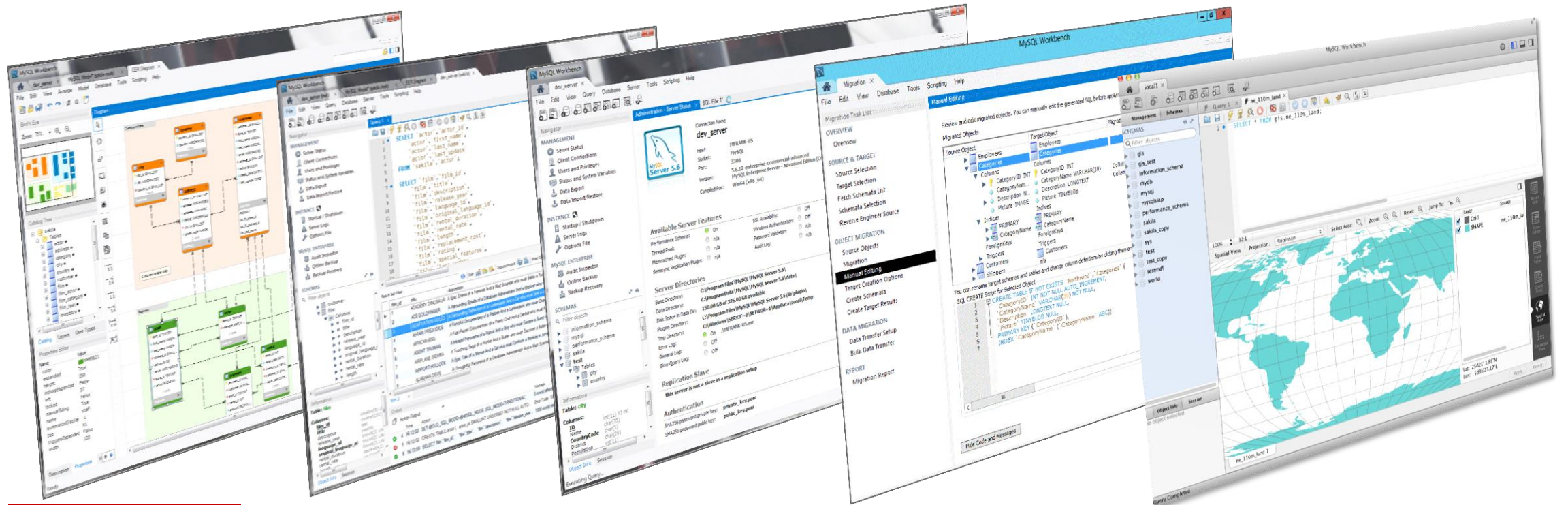
- **インデックス**

- 複数列
- プレフィックス
- “covering index”

参考) MySQL Workbenchによるスキーマデザインのサポート

MySQL Workbench は、データベースアーキテクト、開発者、DBA のための統合GUIツールです。

- MySQLの公式GUIツール
- MySQL Databaseの統合開発環境
- Windows, OS X, Linux 対応



参考) MySQL Workbenchの機能 (コミュニティ&商用共通)

• 管理

- 起動/停止、システム変数/ステータス変数確認、ログ確認、ユーザ管理、セッション管理、オブジェクト管理、データ編集、パフォーマンスダッシュボード&レポート、GIS Viewer, Dump/Import

• 設計

- E-R図作成、フォワードエンジニアリング、リバースエンジニアリング

• 開発

- SQLエディタ、SQL整形、SQLコード補完、SQLシンタックスハイライト、SQL Snippets(ステートメント再利用)、Visual Explain

• マイグレーション

- 各種RDBMSからのマイグレーション
- PostgreSQL, MS SQL Server, MS Access, SQLite, SQL Anywhere, Sybase ASE

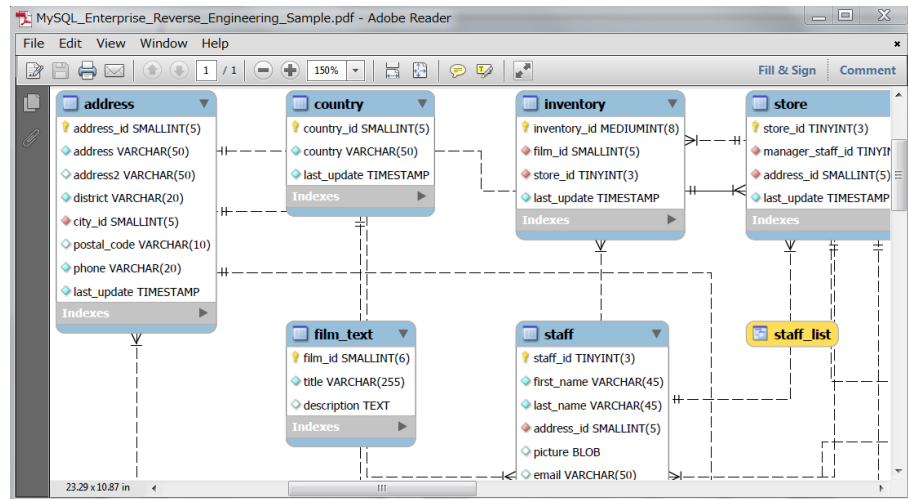
参考) MySQL Workbenchの機能 (商用版限定機能)

• DBドキュメント出力

- データベーススキーマの情報をドキュメント化(テーブル定義書を自動作成)

• MySQL Enterprise Backup用GUI

- バックアップジョブの管理



• データモデルの検証

- DB設計上の間違いや懸念事項を提示
- RDBMS全般/MySQL独自の観点でチェック

• MySQL Enterprise Audit用GUI

- 監査ログの分析、フィルタリング

The screenshot shows the MySQL Workbench interface with the 'Table customer' details. The 'Columns' section displays a table with the following data:

Key	Column Name	Datatype	Not Null	Default	Comment
PK	customer_id	SMALLINT(5)	Yes		
	store_id	TINYINT(3)	Yes		
	first_name	VARCHAR(45)	Yes		
	last_name	VARCHAR(45)	Yes		
	email	VARCHAR(50)	No	NULL	
	address_id	SMALLINT(5)	Yes		
	active	TINYINT(1)	Yes	'1'	
	create_date	DATETIME	Yes		
	last_update	TIMESTAMP	Yes	CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	

The 'Indexes' section displays a table with the following data:

Index Name	Columns	Primary	Unique	Type	Kind	Comment
PRIMARY		Yes	No	PRIMARY		
idx_fk_store_id		No	No	INDEX		
idx_fk_address_id		No	No	INDEX		
idx_last_name		No	No	INDEX		

参考) MySQL5.7 InnoDB - General Tablespace

I/O分散

General TablespaceはSystem Tablespaceに類似した共有型のテーブルスペースです。その為、複数のテーブルを保持する事が可能で、全てのテーブルフォーマットにも対応しています。相対、若しくは独立したディレクトリーに作成する事が可能です。

```
[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.INNO_DB_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](#)

参考) MySQL5.7 InnoDB - Temporary Tables

I/O分散

- 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)
```

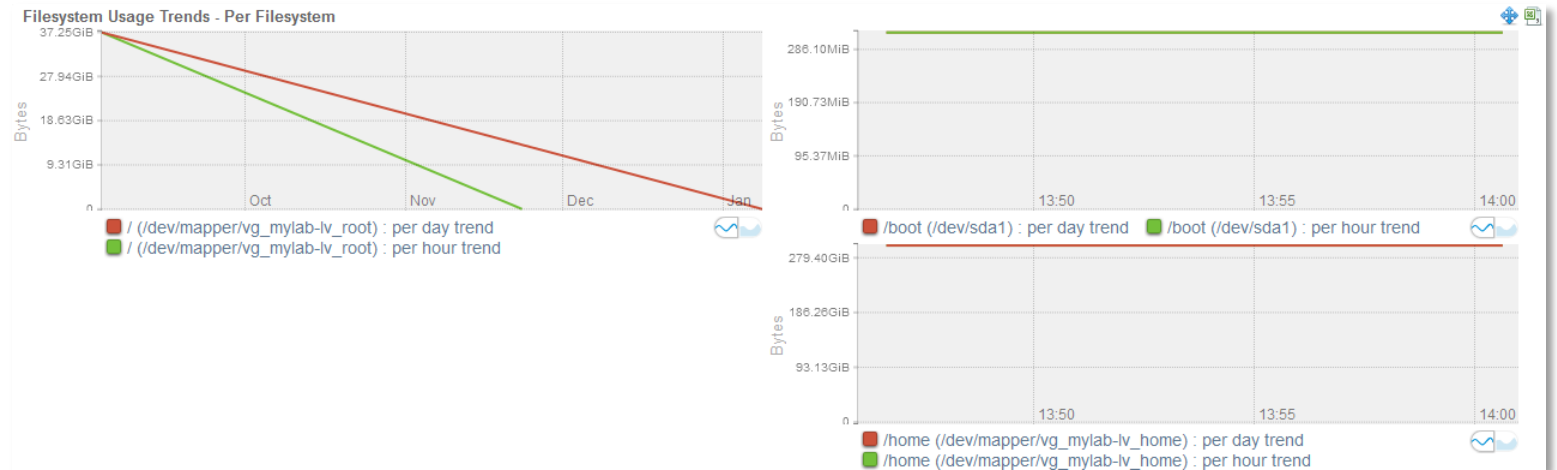
参照: [14.11 InnoDB Startup Options and System Variables](#)



MySQL設定改善プロセス

Trends & Predictive Problem Detection

- ビジュアルトレンド分析
- 異常値の特定
- 問題発生を回避
- 差し迫った容量の問題に対処



Topic: Filesystem / (dev/mapper/vg_mylab-lv_root) Running Out Of Space In about 13 days

Categories: Operating System

Advisor: Filesystem Free Space

Current State: Open

Current Status: Critical

Worst Status: Critical

Auto-Closes by Default: Yes

Last Checked: Sep 3, 2013 2:10:39 PM

Worst Alarm Time: Sep 3, 2013 2:08:06 PM

Notes:

No notes provided.

Details:

Problem Description

Databases use disks and filesystems to store data, indexes, logs, and other artifacts. When space gets low, it can adversely affect the performance of your system, and in extreme cases may cause your application to halt or crash.

Advice

Investigate why filesystem / (dev/mapper/vg_mylab-lv_root) has a 24-hour average growth rate of 2.8 GiB. At that rate, the 28 GiB remaining (out of 49 GiB) will run out in about 13 days, around September 16, 2013.

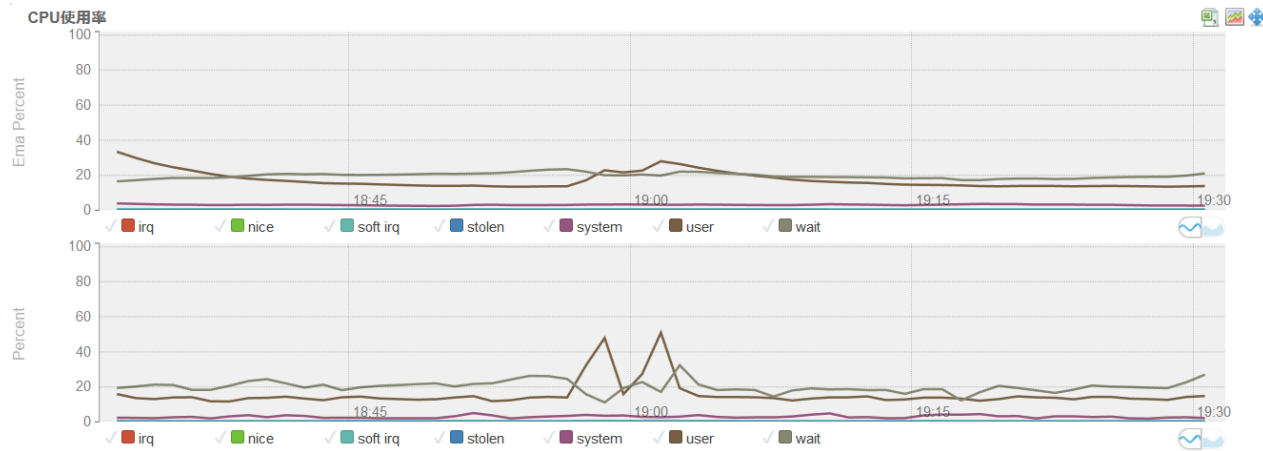
Consider archiving and deleting large files that are no longer needed, as well as temporary files (e.g. files in /tmp on Linux and %TEMP% on Windows). Look for files that are growing rapidly and consider alternatives (e.g. rotating logs, moving those files to another filesystem, etc).

Links and Further Reading

- [MySQL Manual: Managing Disk I/O and File Space for InnoDB Tables](#)
- [Article: Managing Disk Space in Linux](#)
- [Oracle Solaris Administration: Displaying Information About Files and Disk Space](#)

OS Monitoring

- キャパシティプランニング
- 将来的に必要なキャパシティ確認
- 見通し
- 傾向分析
- 時系列データ



データ消去操作

指定よりも古い履歴データを削除します。

4週間

4週間

5週間

6週間

7週間

8週間

3月

6月

12月

18月

24日

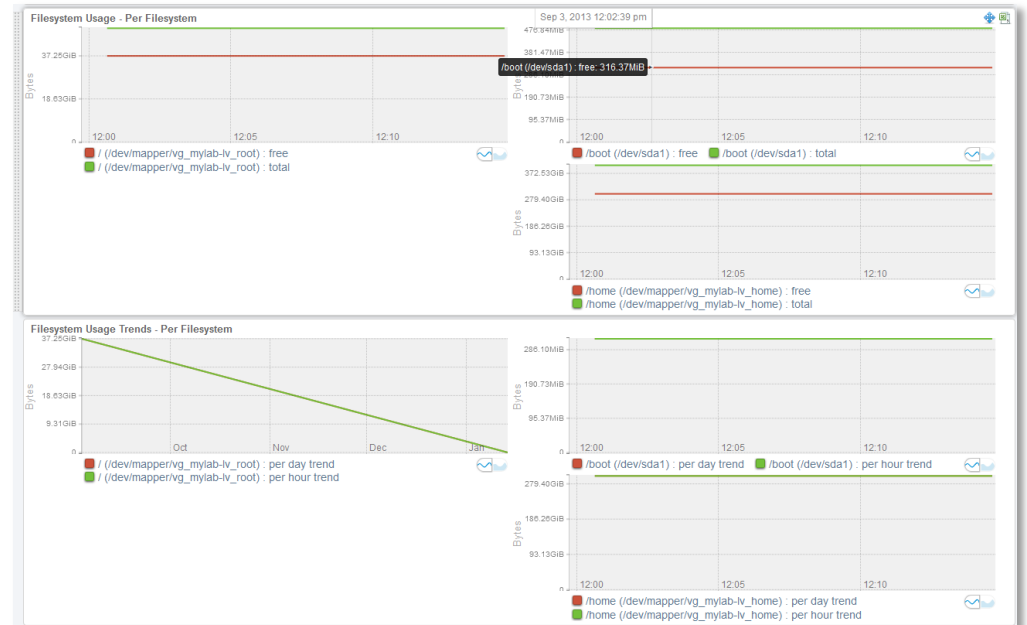
指定のデータを削除

説明

プランのサービス・リクエスト情報の取得と表示に使用されます (アクティブなアカウントのみ)

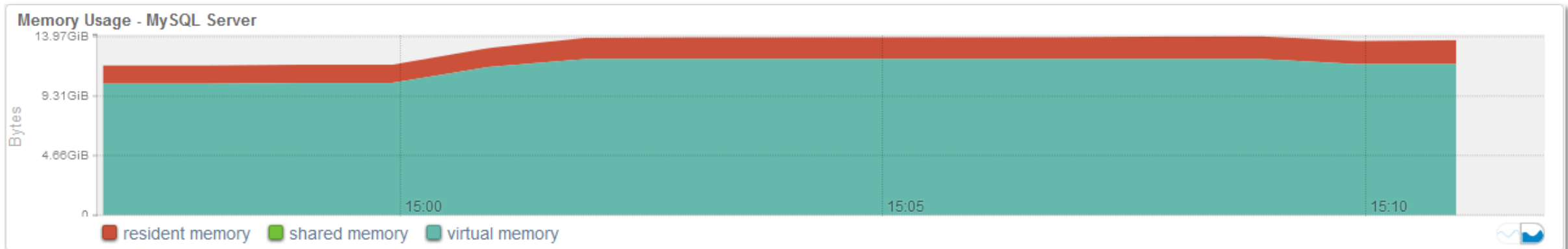
名前 (電子メールアドレス)

保存



Memory Usage Monitoring

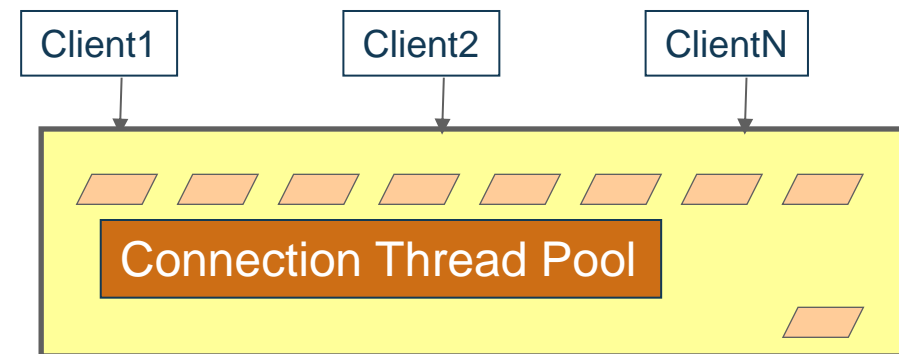
- OSのメモリ使用量を監視
- MySQLのメモリ使用量を監視
- 設定のアドバイス
- メモリ使用量に関する警告



サーバのコネクション&スレッド

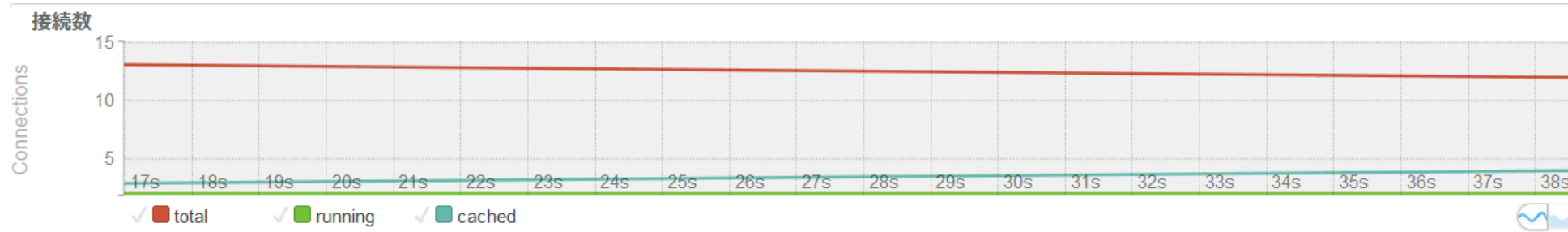
設定ファイル my.cnf :

- **max_connections (151)**
 - サーバが許容可能なコネクション数。
多すぎるとメモリを消費しきる可能性あり
- **thread_cache_size (9) ※**
 - スレッドをコネクションの切断後にもキャッシュしておく数一般的には $\text{max_connections}/3$
※計算式により自動計算: $8 + (\text{max_connections} / 100)$



mysql> show status;

- **Max_used_connections**
 - max_connections とあわせてチェック
- **Threads_created**
 - thread_cache ミス
 - 低い数値であるべき



参照 : http://dev.mysql.com/doc/refman/5.6/ja/server-system-variables.html#sysvar_thread_cache_size

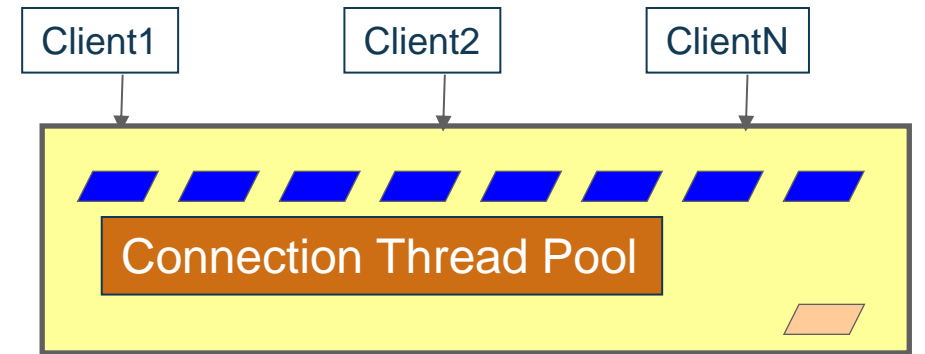
コネクションスレッド毎のバッファ

設定ファイル my.cnf:

- **sort_buffer_size (256KB)**
 - ソート用メモリサイズ。このサイズを超えるソートはディスク利用。
MySQL 5.6でデフォルト値が縮小された。(2MB⇒256KB)
- バッチ処理などの場合、処理実行前に 動的にこれらのパラメタを変更する。

mysql> show status;

- **Sort_merge_passes**
 - ファイルを利用したマージソートのパス数
 - ソートがメモリ上だけで収まらない場合には要確認
 - インデックスの利用を検討

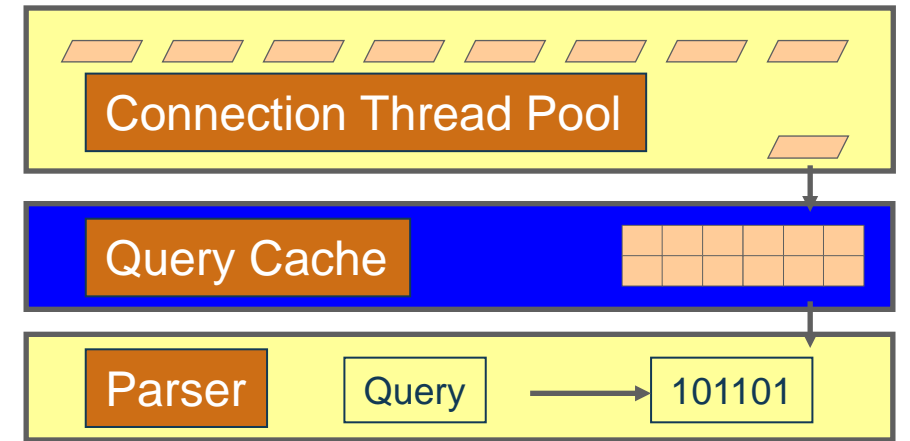


待ち時間 (hh:mm:ss.ms)				行数		Temp Tables		初回実行
合計	最高	平均	平均値の履歴	合計	平均	Total	Disk %	
2.513	0.188	0.028		3,371	37	630	43	2015/08/27 21:46:34

クエリキャッシュ

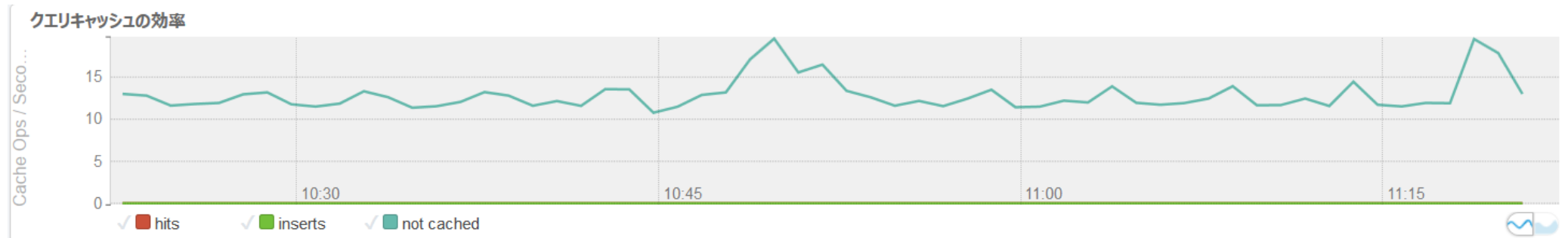
設定ファイル my.cnf:

- **query_cache_size (1MB)**
 - クエリキャッシュに割り当てるメモリサイズ
 - 一般的には32MでOK
- **query_cache_type (OFF)**
 - 最悪のケースではパフォーマンスのオーバーヘッドが約15%
 - SELECTの比率が高いサーバで有効
 - DEMANDに設定すると、クエリ実行時にSQL_CACHE句を付けたクエリだけキャッシュ可能



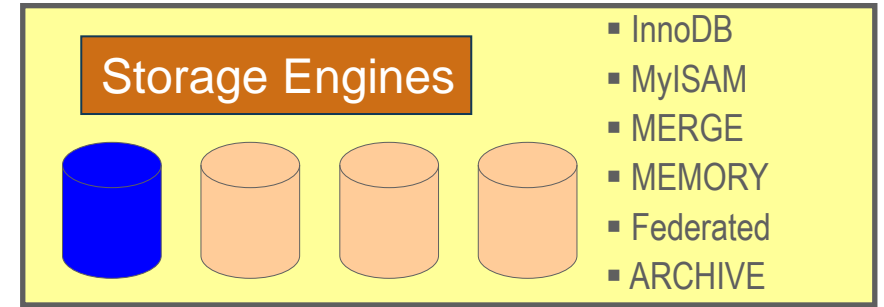
mysql> show status;

- **Qcache_hits, Qcache_inserts** :
キャッシュヒット/登録件数、ヒット率が小さければクエリキャッシュを無効にすることも検討
- **Qcache_lowmem_prunes** :
メモリ不足のためにキャッシュが削除された回数



InnoDB パフォーマンス Tips

- **innodb_buffer_pool_size**
 - MySQL&InnoDBのみを利用していれば、メモリの80%程度設定
 - データとインデックスの両方をキャッシュ
- **innodb_log_file_size**
 - innodb_buffer_pool_sizeの25%～100%
 - ログファイルがどの程度頻繁に切り替わっているかをチェック
 - 値を大きくするとクラッシュ後のリカバリ時間が長くなる
- **innodb_flush_log_at_trx_commit**
 - 1 (遅い) コミット時にログをフラッシュ。真のACID
 - 2 (速い) コミット時にはOSのキャッシュにログをフラッシュ、ディスクとのSYNCは毎秒1回
 - 0 (最速) ログを毎秒1回(またはそれ以下)フラッシュ
- **innodb_file_per_table**
 - MySQL 5.6からデフォルト”ON”
 - ディスクI/Oの分散やibdataファイルの肥大化を防ぐために”ON”を推奨



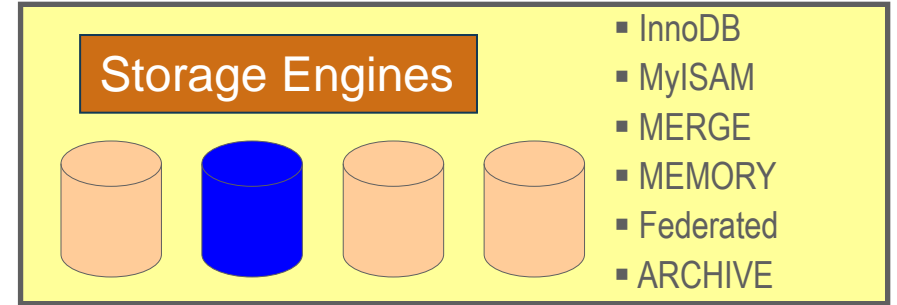
`mysql> SHOW ENGINE INNODB STATUS;`

InnoDBの内部での稼働情報

- ファイル I/O
- バッファプール
- ログ情報
- 行/ロック情報

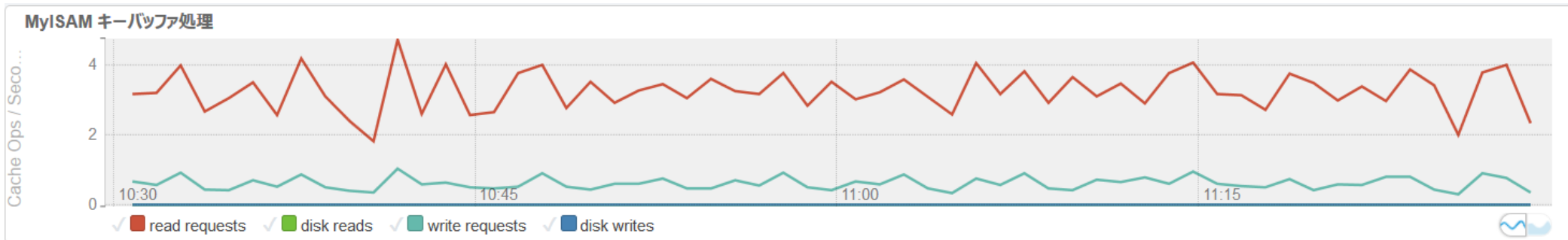
MyISAM パフォーマンス Tips

- **key_buffer_size (8M)**
 - MySQL&MyISAMのみを利用していれば、メインメモリの25-33%を割り当てる
- **myisam_sort_buffer_size (8M)**
 - インデックス作成時には大きくしておく
- **myisam_recover=FORCE,BACKUP**
 - テーブルアクセス時に破損をチェック。
- **delay_key_write=ALL**
 - インデックスをディスクに反映させるのをテーブルが閉じられたタイミングのみ
 - クラッシュ時にデータが破損する可能性が向上するため注意



mysql> show status like 'key%'

- **Key_read_requests, Key_reads**
 - Key_reads/Key_read_requests
キャッシュミス率
- **Key_write_requests, Key_writes**
 - キャッシュミス率は高いことが多い
- **Key_blocks_unused**
 - 値が大きい場合はキャッシュサイズが大きすぎる可能性大



Best Practice Advisors

MySQLに特化した、アドバイザーが複雑な設定をアドバイス

- MySQLベストプラクティスを適用可能
- 14アドバイザーのカテゴリー
- 250以上のサブアドバイザカテゴリー
- 閾値ベースの警告管理
 - 指数移動平均
 - 変更率の検知
- 問題解決のアドバイスを提供
 - カスタムスクリプトの作成、展開、バージョン管理にかかる時間を短縮。
 - データベース管理者だけでは発見不可能な問題やチューニング方法をアドバイス。



Category	Configured
Administration	Configured: 22 of 22
Agent TODO	Configured: 3 of 3
Memory Usage	Configured: 6 of 6
Query Analysis	Configured: 4 of 4
アベイラビリティ	Configured: 5 of 5
オペレーティングシステム	Configured: 5 of 5
クラスター	Configured: 10 of 10
グラフ	Configured: 93 of 93
スキーマ	Configured: 17 of 17
セキュリティ	Configured: 29 of 29
バックアップ	Configured: 1 of 1
パフォーマンス	Configured: 21 of 21
レプリケーション	Configured: 19 of 19
監視とサポートサービス	Configured: 6 of 6

"I definitely recommend the MySQL Enterprise Monitor to DBAs who don't have a ton of MySQL experience. It makes monitoring MySQL security, performance and availability very easy to understand and to act on."

Sandi Barr
Sr. Software Engineer
Schneider Electric



Best Practice Advisors (有効・無効・閾値・間隔・イベント処理)

Memory Usage		Configured: 6 of 6					
Item	Info	Coverage	Schedule	Event Handling			Parameters
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> innodb_buffer_cache_hit_rate_not_optimal.name	?	100% (5/5)	5m	0	0	0	95 85 75
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> キーバッファサイズがキーキャッシュに対して最適化されていない可能性があります	?	100% (5/5)	5m	0	0	0	95 85 75
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> クエリキャッシュのサイズが小さすぎる可能性があります	?	100% (5/5)	5m	0	0	0	0
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> クエリキャッシュのヒット率が最適化されていません	?	100% (5/5)	5m	0	0	0	60 50 40
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> スレッドキャッシュサイズが最適化されていないようです	?	100% (5/5)	5m	0	0	0	95 85 75
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> テーブルキャッシュが最適化されていません	?	100% (5/5)	5m	0	0	0	20 40 60
Query Analysis		Configured: 4 of 4					
アベイラビリティ		Configured: 5 of 5					
オペレーティングシステム		Configured: 5 of 5					
Item	Info	Coverage	Schedule	Event Handling			
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> CPU使用率アドバイザ	?	100% (5/5)	1m	1	0	0	Moving Average Window: 5m Enable CPU Outlier Detectio
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> MySQL プロセス	?	100% (3/3)	1m	0	0	0	Moving average window size: 5m
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> ram_usage_excessive.name	?	100% (5/5)	1m	0	0	0	300 200 100
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> ネットワークトラフィックグラフ	?	100% (17/17)	1m	0	0	0	
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> ファイルシステムの空き領域	?	100% (10/10)	1m	1	0	0	Extrapolate Free Space To Zero In Graphs: true Analyze F

<input type="checkbox"/>	現在 <input type="checkbox"/>	最重要 <input type="checkbox"/>	サーバー	説明	時刻	アクション
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	misc:3306	MyISAM の同時挿入が適切に設定されていない可能性があります	about an hour ago	<input checked="" type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	misc:3306	名前の大文字小文字を区別するためデータベースの互換性が低いと考えられます	about an hour ago	<input checked="" type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	misc:3306	max_prepared_stmt_countにデフォルト値が使用されています	about an hour ago	<input checked="" type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	misc:3306	シンボリックリンクが有効です	about an hour ago	<input checked="" type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	misc:3306	Slave SQL Processing Not Multi-Threaded	about an hour ago	<input checked="" type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	misc:3306	Binary Log Debug Information Disabled	about an hour ago	<input checked="" type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	misc:3306	一般クエリログが有効になっています	about an hour ago	<input checked="" type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	misc:3306	InnoDBログバッファがトランザクション後毎回ディスクにフラッシュされています	about an hour ago	<input checked="" type="checkbox"/>

Topic: InnoDBログバッファがトランザクション後毎回ディスクにフラッシュされています

Categories: パフォーマンス

Advisor: InnoDBログバッファがトランザクション後毎回ディスクにフラッシュされています

Current State: オープン

Closed By: Closed:

ワーストステータス Warning

Auto-Closes by Default: Yes

Worst Alarm Time: 2015/01/24 14:25:17

Notes:

メモの入力はありません。

Details:

問題の説明

デフォルトでは、InnoDBのログバッファはログファイルにトランザクションのコミットごとに書き出され、ディスクへのフラッシュ操作がログファイルに対して実行されます。この仕組みによりACIDに準拠しています。クラッシュ時、1秒程度のトランザクションをロストすることが許容できるなら、**innodb flush log at trx commit**を0または2に設定することでパフォーマンスを向上できます。この値を2に設定した場合、オペレーティングシステムがクラッシュするか電源異常が発生したときのみ、最後の1秒分のトランザクションが消去される場合があります。これはスレーブサーバーでは有効です。1秒程度のデータロストは、必要に応じてマスタサーバーから復元することができるからです。

アドバイス

my.cnf/my.iniファイルにおいて **innodb_flush_log_at_trx_commit=2** と設定し、サーバーを再起動してください。

警告: ACIDに準拠する場合は値は1にする必要があります。値を2に設定した場合、オペレーティングシステムがクラッシュするか電源異常が発生したとき、最後の1秒分のトランザクションが消失する場合があります。これは使用中のアプリケーションや環境によっては重大ではない場合もあります。特にスレーブサーバーでは、1秒間分のデータをロストしたとしても、マスタから復元することができるからです。

Enterprise Replication Monitor

- レプリケーショントポロジーの自動検知
- マスター/スレーブのパフォーマンス監視
- レプリケーションアドバイザーによるサポート
- レプリケーションのベストプラクティスを提示

Replicationの遅延を検知して通知

アドバイザー ?

スレーブが大幅にマスタから遅れています ×

Event Statuses ?

Emergency TODO ×

Event Handling

SMTP Notification Groups ?

Replication ×

SMTP Notification Policy ?

Notify on event escalation TODO

Replication Monitoring

Servers	Type	Threads		Time Behind	Binary Logs		Position	Log Space		
		IO	SQL		Current File	Position		Binary Logs	Relay Logs	
Replication 1 (4)	MIXED	✓	✓							
mylab.localdomain:3306	master/slave	✓	✓	00:00:00	mylab-bin.000001	791	mylab-bin.000001	791	791 B	1.1 KB
mylab.localdomain:3307	master/slave	✓	✓	00:00:00	mylab-bin.000001	791	mylab-bin.000001	791	791 B	1.1 KB
mylab.localdomain:3308	master/slave	✓	✓	00:00:00	mylab-bin.000001	986	mylab-bin.000001	791	0.96 KB	1.1 KB
MLORD-PC:3306	slave	✓	✓	00:00:00			mylab-bin.000001	986		1.29 KB

Security Administration

- アカウント管理
- 権限設定変更時期を確認可能
- ベストプラクティス・アドバイス
- アクセスの問題を監視

mylab.localdomain, mylab.localdomain:3306 Server Has Accounts Without A Password 6 minutes ago

Topic: Server Has Accounts Without A Password
Categories: Security **Advisor:** Server Has Accounts Without A Password
Current State: Open **Current Status:** Critical **Worst Status:** Critical
Auto-Closes by Default: Yes **Last Checked:** Sep 3, 2013 1:17:40 PM **Worst Alarm Time:** Sep 3, 2013 11:57:39 AM

Notes:
No notes provided.

Details:

Problem Description
Accounts without passwords are particularly dangerous because an attacker needs to guess only a username. Assigning passwords to all accounts helps prevent unauthorized users from accessing the system.

Advice
Assign a strong password to the following accounts on server mylab.localdomain:3306:

- '@localhost',
- '@mylab.localdomain',
- 'root@127.0.0.1',
- 'root@::1',
- 'root@mylab.localdomain'

A strong password should be at least 8 characters long, contain lowercase and uppercase characters, numbers and symbols, and not contain words found in a dictionary.

Recommended Action
SET PASSWORD FOR 'user_name'@'host_name' = PASSWORD('new_pass');

Links and Further Reading
[Securing Your MySQL Installation](#)
[Securing a MySQL Server on Windows](#)

Expression
%user% != THRESHOLD

Evaluated Expression

- '@localhost',
- '@mylab.localdomain',
- 'root@127.0.0.1',
- 'root@::1',
- 'root@mylab.localdomain' != "

<input type="checkbox"/>	Current	Worst	Subject	Topic
<input type="checkbox"/>			mylab.localdomain, mylab.localdomain:3306	Root Account Without Password
<input type="checkbox"/>			mylab.localdomain, mylab.localdomain:3306	Server Has Accounts Without A Password
<input type="checkbox"/>			mylab.localdomain, mylab.localdomain:3306	Average Statement Execution Time Excess...
<input type="checkbox"/>			mylab.localdomain, mylab.localdomain:3306	SQL Statement Generates Errors or Warnings
<input type="checkbox"/>			mylab.localdomain, mylab.localdomain:3306	Server Has Anonymous Accounts

Monitoring the Firewall

Enterprise MonitorとEnterprise Firewallの連携

グラフによりデータベースへの攻撃を可視化し、管理者へAlertを送信し早期確認を可能に

Event Handler Name [?](#)
Firewall Attack Detection

Filters

アセット名 [?](#)
MySQLServer: GA01:3306 ✕

グループ [?](#)
Linux ✕

アドバイザ [?](#)
Firewall Statistics ✕

Event Statuses [?](#)
Warning TODO ✕

Event Handling

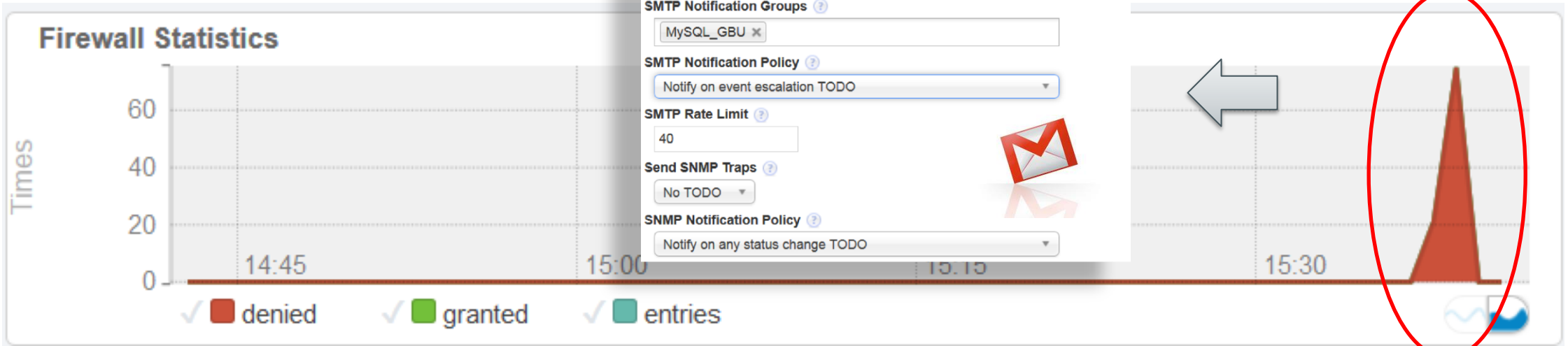
SMTP Notification Groups [?](#)
MySQL_GBU ✕

SMTP Notification Policy [?](#)
Notify on event escalation TODO

SMTP Rate Limit [?](#)
40

Send SNMP Traps [?](#)
No TODO

SNMP Notification Policy [?](#)
Notify on any status change TODO



MySQL Enterprise Monitor + Backup

- バックアップ結果のモニタリング
- バックアップパフォーマンスのモニタリング

最新バックアップ取得を確実にする事が可能

ORACLE MySQL Enterprise Monitor

Dashboards ▾ Events Query Analyzer Reports & Graphs ▾ Configuration ▾

Advisors

Edit Selected Advisors Disable Selected Advisors Create Advisor Import/Export

Administration	Configured: 26 of 26				
Agent	Configured: 3 of 3				
Availability	Configured: 6 of 6				
Backup	Configured: 5 of 5				
<input type="checkbox"/> Item	Info	Coverage	Schedule	Event Handling	
MySQL Enterprise Backup Failed		100% (1/1)	5m	0 0 0	"FAILURE"
MySQL Enterprise Backup Succeeded		100% (1/1)	5m	1 0 0	"SUCCESS"
Last Full MySQL Enterprise Backup Too Old		100% (1/1)	6h	1 0 0	7
Incremental MySQL Enterprise Backups Not Enabled		100% (1/1)	6h	1 0 0	0
MySQL Enterprise Backup Lock Time Excessive		100% (1/1)	5m	0 0 0	10 60

カスタマイズ & 拡張性

- カスタマイズ
 - グループ
 - アドバイザー
 - グラフ
 - フィルター
 - イベントハンドラ
- and more ...

The screenshot displays the Oracle MySQL Enterprise Monitor configuration interface. The main window is titled 'Manage Rule: MySQL Enterprise Dashbo...' and shows the 'Create Rule' configuration page. The interface is divided into several sections: 'Rule Definition' (Rule Name, Advisor Category, Version, Expression), 'Thresholds' (Critical Alert, Warning Alert, Info Alert), 'Variable Assignment' (Variable, Data Item, Instance), and 'Rule Info' (Problem Description, Advice, Recommended Action). On the right side, there is a 'Edit Event Handler' panel with fields for 'Event Handler Name' (Slave is too far behind!), 'Assets' (Os: mylab.localdomain), 'Groups' (Production Replication), 'Advisors' (Slave Too Far Behind Master), 'Event Statuses' (Emergency, Critical), 'SMTP Notification Groups' (DBA Team), 'SMTP Notification Policy' (Notify on event escalation), 'SMTP Rate Limit' (10), 'Send SNMP Traps' (No), 'SNMP Notification Policy' (Notify on any status change), and 'Auto-Close Events' (No). At the bottom right of the event handler panel are 'Save Event Handler' and 'Cancel' buttons. The footer of the browser window shows 'Copyright © 2005, 2013, Oracle and/or its affiliates. All rights reserved.' and '3.0.0.2887 - mylab.localdomain (192.168.1.10) - Sep 5, 2013 8:58:26 am (Up Since: 5 minutes ago) - About'.

Custom Data Collection

<http://dev.mysql.com/doc/mysql-monitor/3.0/en/memcustomdata-collection.html>

Overview of Advisor Creation

<http://dev.mysql.com/doc/mysql-monitor/3.0/en/memcreating-advisors-overview.html>

カスタマイズ & 拡張性 (Event Handling)

関係者へのエスカレーション

- SMTP
- SNMP

メール通知グループ

Create Notification Group

グループ名	宛て先	件名の行	SMS
MySQL_DBA	shinya.sugiyama@oracle.com	MySQL Enterprise Monitor {0} 警告: {1} ({2})	false

メールの設定

メール通知の有効化

差出人アドレス: "MySQL_DBA"<mysql_dba@oracle.com>

SMTPサーバー: mail.domain.com:25

SMTPサーバーのログイン: admin

保存時にパスワードを更新する

JavaMail TLS/SSLの無効化

保存時にテストメールメッセージを送信する宛先: (オプション)

Emailの設定の保存

SNMPの構成

SNMP通知の有効化

SNMPv1 を使用

SNMPv2 を使用

ターゲット 1: [] ポート番号: []

ターゲット 2: [] ポート番号: [] (オプション)

コミュニティ文字列: []

リモート MySQL エージェントホストの IP アドレスを、アドバイザトラップのための SNMP トラップ エージェントアドレスとして使用する (オプション)

内部的に生成されたトラップ用の SNMP トラップエージェントアドレス: [] (オプション)

保存時にテストトラップを送信する (オプション)

SNMPの構成の保存

SNMP通知の状態

一般的なパフォーマンスの問題の解決

パフォーマンスの問題の原因	MySQL Enterprise Monitor Helps
テーブルスキャンを行うクエリ	すぐにこれらのクエリを確認してください。
過度の一時テーブル利用(ディスク)	原因となるクエリを確認してください。
CPUスパイク発生	引き起こした原因を調べて下さい。
ディスクI/Oの飽和状態	引き起こした原因を調べて下さい。
内部ロック	ロック問題が発生時にアラートを取得
ハードウェアの問題	ディスクとネットワーク問題発生時にアラート取得
データベースおよびスキーマの変更	変更発生時にアラート取得
新しいクエリが導入	いつ、どのようなQueryが追加されたか確認
MySQLの設定不備	設定の推奨値やチューニングに関するアドバイスを取得

解決された上位10のパフォーマンス問題

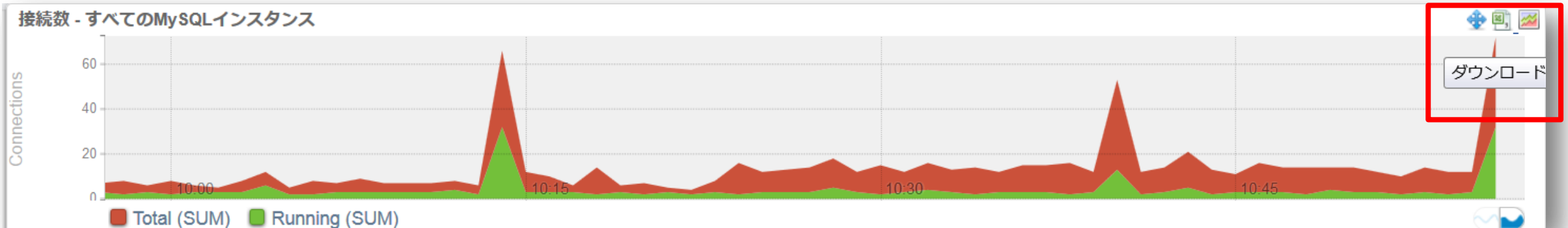
- ✓ 最もコストが高いクエリの特典
- ✓ フルスキャン処理(インデックス無し)
- ✓ 過度な一時テーブル処理
- ✓ 大量または頻繁なfilesorts処理
- ✓ CPU使用率過負荷
- ✓ ディスクI/Oの飽和
- ✓ ハードウェアの問題
- ✓ データベースおよびスキーマの変更
- ✓ 新規クエリーの検知
- ✓ MySQL設定の問題
- ✓ 内部ロックの問題
- ✓ セキュリティ脆弱性への対応

補足情報

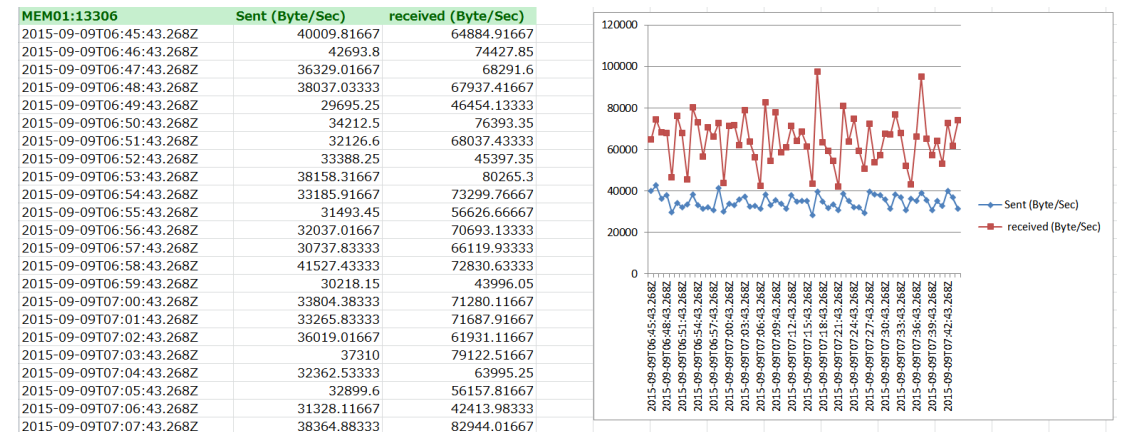
コマンドラインクライアントによる定期的なデータの収集

MEMは、標準的なWebアプリケーションの為、wgetやCurlを用いてデータを/取り出す事も可能。

MEMは、HTTP基本認証やSSLに対応しています。



例) `curl --user user:pass --insecure https://etoolsstable.no.oracle.com:30000/v2/rest/[asset-selection]?format=CSV -o throughput.csv`



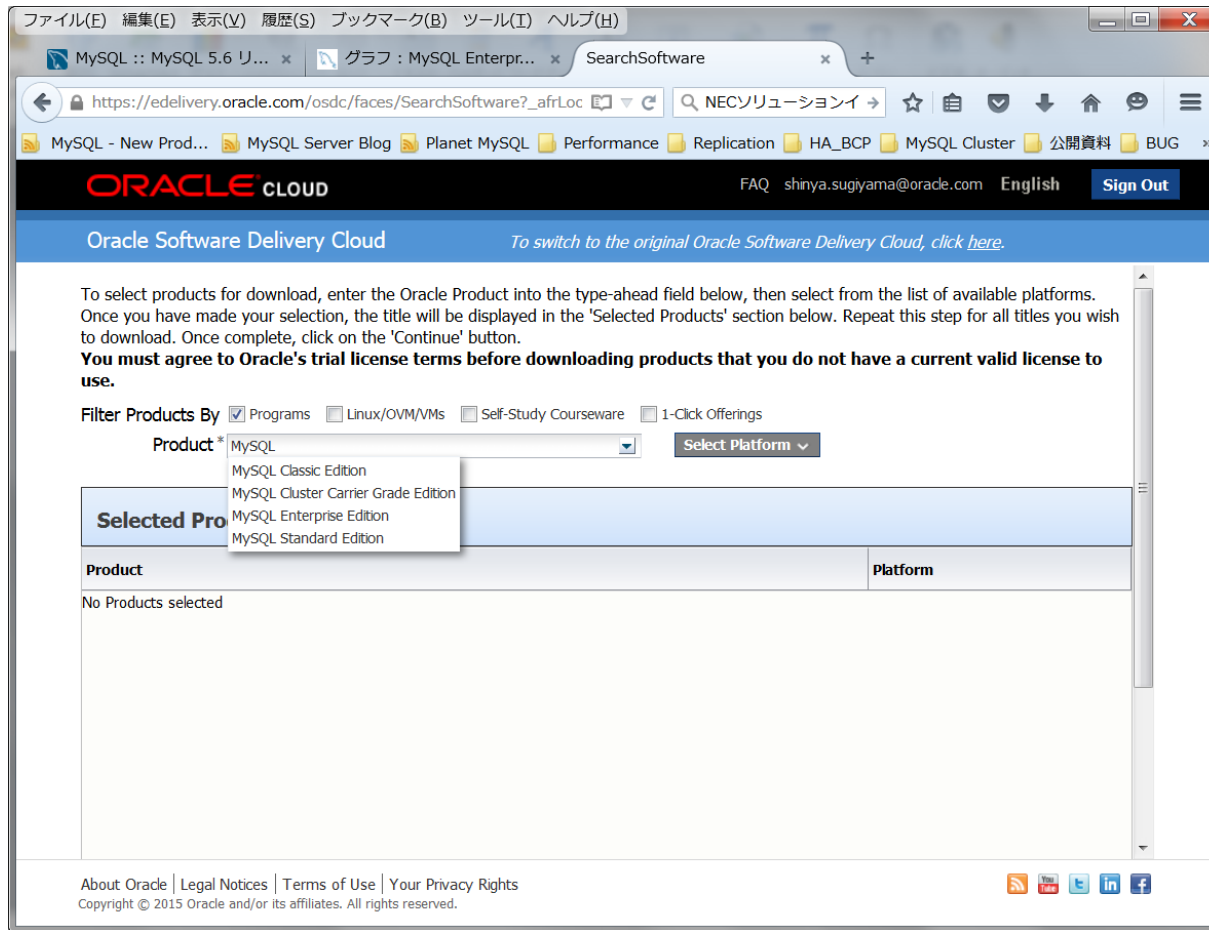
まとめ

1	MySQL Enterprise Monitorは、MySQLに特化したモニタリングツールで、サーバーステータス、クエリーパフォーマンス、データベースにおけるイベントを統合監視する事が可能なツールです。
2	セキュリティを含む設定が複雑な、MySQLのパラメータ設定なども、アドバイザーが推奨値をアドバイスしてくれる為、データベース管理工数を削減すると同時にMySQLを最適な設定に保つ事が可能です。
3	MySQL Enterprise Monitorは、MySQLで広く使われている、レプリケーション、クラスタリング、バックアップ等の状態も監視し、問題が発生した場合は、SNMPやSMTPで管理者に知らせる事が可能。

※ サポートを併用して頂く事で、問題発生時も迅速に対応する事が可能な為、機会損失を最小限にする事が可能。

MySQL Enterprise Edition & Cluster CGEの試使用

30日間トライアル



- Oracle Software Delivery Cloud
<http://edelivery.oracle.com/>

- 製品パックを選択：
“Product”にMySQLと入力し、
OSを選択し”Continue”

- 製品マニュアル
<http://dev.mysql.com/doc/index-enterprise.html>

MySQL Enterprise Monitor on Youtube Tutorial

Installing MySQL Enterprise Monitor

<https://www.youtube.com/watch?v=GbqsBqfT-mQ&index=2&list=PL9756ECC6D1DC01D2>

Real-time MySQL Performance & Availability Monitoring

<https://www.youtube.com/watch?v=e5HhC0Xii0Y&index=5&list=PL9756ECC6D1DC01D2>

Remotely Monitor MySQL in the Cloud

https://www.youtube.com/watch?v=a_91iSUfeWA&list=PL9756ECC6D1DC01D2&index=3

Performance Tuning with MySQL Query Analyzer

<https://www.youtube.com/watch?v=pjRTLPeUOug&list=PL9756ECC6D1DC01D2&index=4>

MySQL Enterprise Monitor: Integration With Your Private Cloud and DevOps Environments

<https://www.youtube.com/watch?v=cK5HxhJrXns&index=6&list=PL9756ECC6D1DC01D2>

MySQL Enterprise Monitor 3.0.18 マニュアル

<http://dev.mysql.com/doc/mysql-monitor/3.0/ja/index.html>

有難うございました

A dolphin is captured in mid-leap, emerging from the surface of a clear blue ocean. The dolphin's body is sleek and dark, with water droplets and splashes around it, creating a sense of motion and energy. The background is a deep, uniform blue, suggesting a vast, open sea. The dolphin's head is pointed upwards, and its tail is visible as it breaks the surface.

参考情報

Performance Schema

メモリ統計情報

- 統計情報の収集
 - メモリ利用タイプ別 (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 Performance Schema

- パフォーマンスのボトルネックを特定
- 問題のあるクエリーを識別
- LOCKに関する詳細をリアルタイムに取得
- MySQL内部情報を正確に参照可能
- MySQLの内部情報をリアルタイムに取得
- クエリーの実行中,リアルタイムに洞察を取得

```
mysql> select * from host_summary_by_stages;
```

host	event_name	total	wait_sum	wait_avg
hal	stage/sql/Opening tables	889	1.97 ms	2.22 us
hal	stage/sql/Creating sort index	4	1.79 ms	446.30 us
hal	stage/sql/init	10	312.27 us	31.23 us
hal	stage/sql/checking permissions	10	300.62 us	30.06 us
hal	stage/sql/freeing items	5	85.89 us	17.18 us
hal	stage/sql/statistics	5	79.15 us	15.83 us
hal	stage/sql/preparing	5	69.12 us	13.82 us
hal	stage/sql/optimizing	5	53.11 us	10.62 us
hal	stage/sql/Sending data	5	44.66 us	8.93 us
hal	stage/sql/closing tables	5	37.54 us	7.51 us
hal	stage/sql/System lock	5	34.28 us	6.86 us
hal	stage/sql/query end	5	24.37 us	4.87 us
hal	stage/sql/end	5	8.60 us	1.72 us
hal	stage/sql/Sorting result	5	8.33 us	1.67 us
hal	stage/sql/executing	5	5.37 us	1.07 us
hal	stage/sql/cleaning up	5	4.60 us	919.00 ns

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]>
```

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 * 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)
```

```
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>

 sys_56.sql	Adding the view innodb_lock_waits that displays a snapshot of the tra...	2 months ago
 sys_57.sql	Merge pull request #31 from JesperWisborgKrogh/dev/20141021_innodb_lo...	2 months ago

■ 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/>