

ORACLE®

MySQL 5.7開発最新動向

日本オラクル株式会社

山崎 由章 / MySQL Senior Sales Consultant,
Asia Pacific and Japan



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(確約)するものではないため、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。

OracleとJavaは、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。文中の社名、商品名等は各社の商標または登録商標である場合があります。

MySQL 5.6: GA 14 Months Ago – Best GA Ever

IMPROVED PERFORMANCE AND SCALABILITY

- Scales to 48 CPU Threads
- Up to 230% performance gain over MySQL 5.5

IMPROVED INNODB

- Better transactional throughput and availability

IMPROVED OPTIMIZER

- Faster query execution and diagnostics for query tuning and debugging

IMPROVED REPLICATION

- Higher performance, availability and data integrity

IMPROVED PERFORMANCE SCHEMA

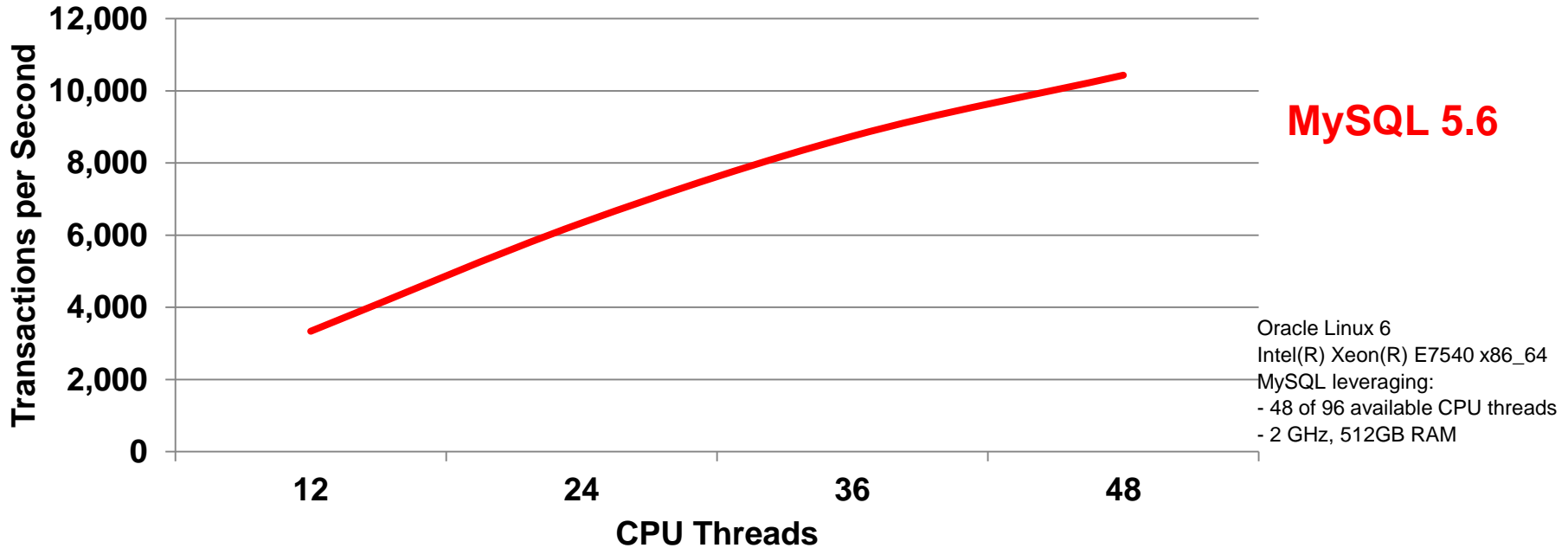
- Better Instrumentation, User/Application level statistics and monitoring

New! NoSQL ACCESS TO INNODB

- Fast, Key Value access with full ACID compliance, better developer agility

MySQL 5.6: Scalability

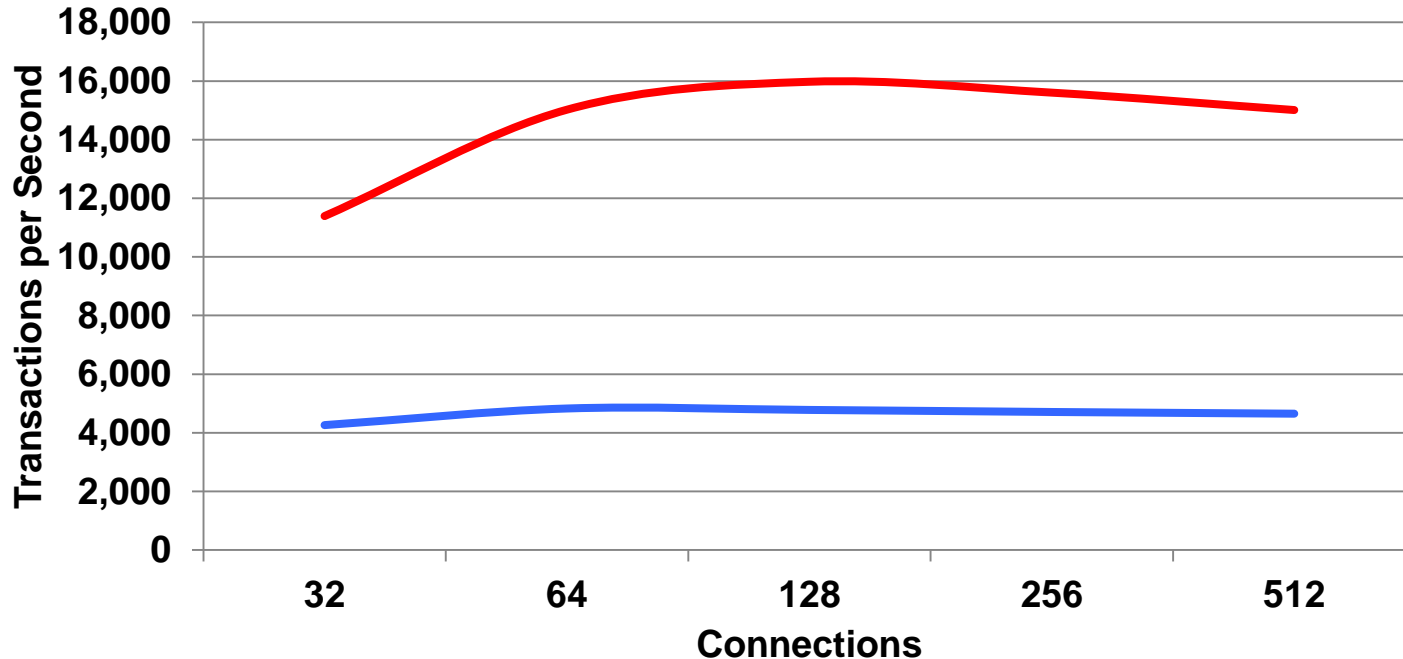
MySQL 5.6 Read Write (Linux)



■ 最新のハードウェアやOSに対応した性能拡張性

MySQL 5.6: SysBench Benchmarks

MySQL 5.6 vs. 5.5 - Read Only (Linux)



MySQL 5.6

MySQL 5.5

Oracle Linux 6
Intel(R) Xeon(R) E7540 x86_64
MySQL leveraging:
- 48 of 96 available CPU threads
- 2 GHz, 512GB RAM

Up to 234% Performance Gain

ORACLE

MySQL 5.6: InnoDB

リソースの有効活用による性能と拡張性能向上

- レガシーなボトルネックの削減
- スレッドの同時実行性能の向上
- 参照専用トランザクションの実装
- SSDへの最適化

可用性の向上 & 拡張性の改良

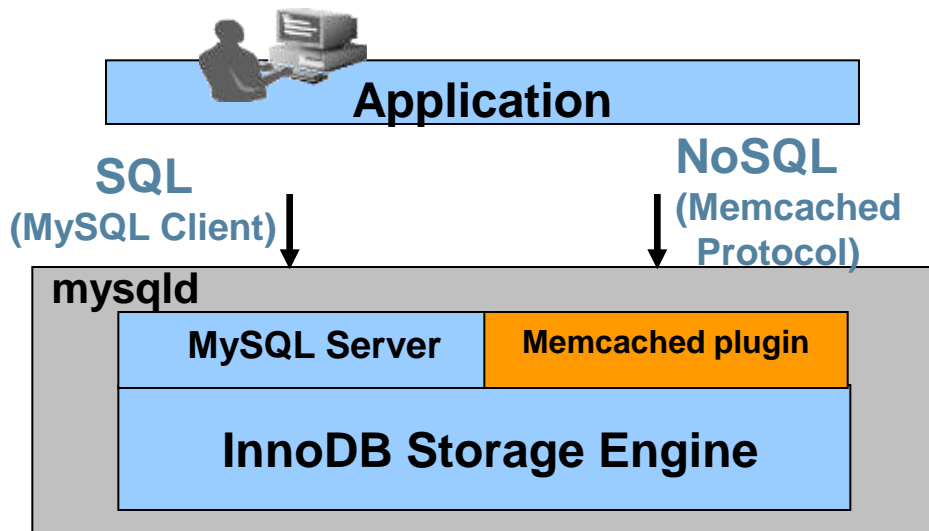
- オンラインでのDDL実行
- テーブルスペースの可搬性の向上
- バッファプールのダンプ&リストア

開発の柔軟性の向上

- NoSQLでのInnoDBへのキーバリュ型アクセス

MySQL 5.6: InnoDB

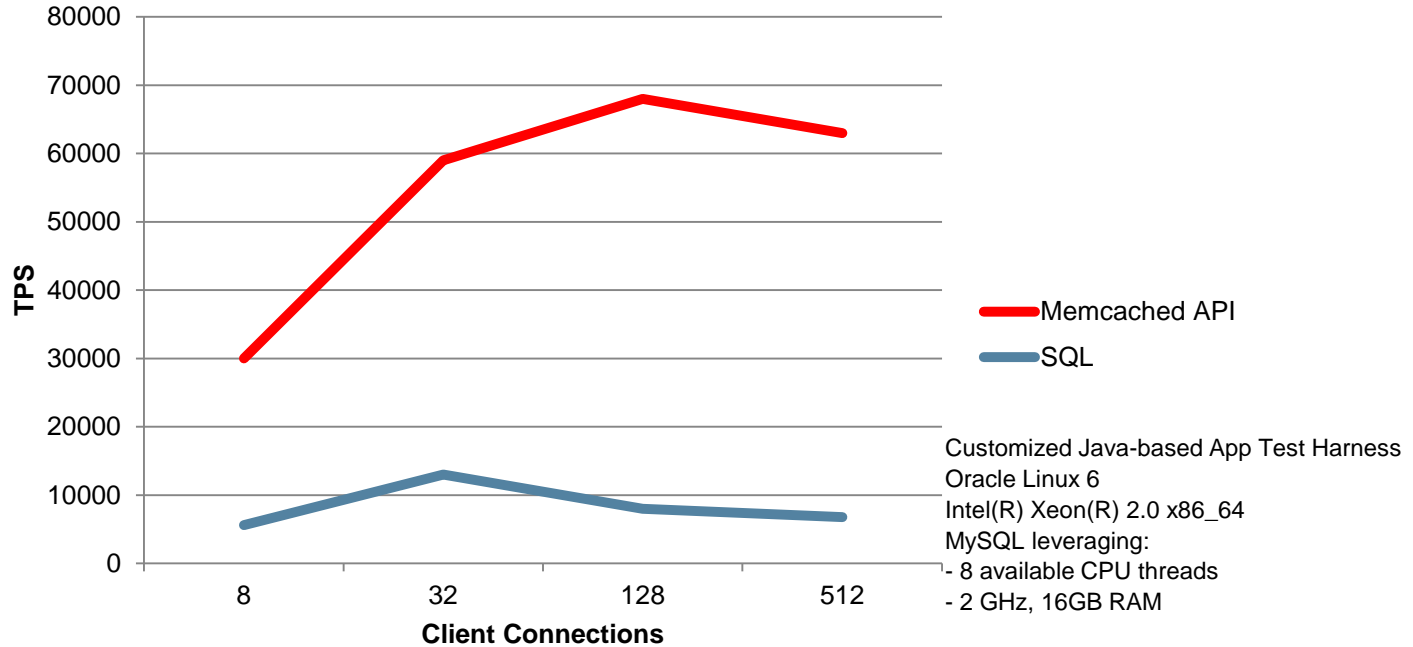
RDBMSとNoSQLの両立



- InnoDBに素早く、簡単にアクセス
 - Memcached API経由のアクセス
 - 既存のMemcachedクライアントを使用
 - SQL変換をバイパス
- NotOnlySQLアクセス
 - キー・バリュー操作
 - 複雑なクエリやJOIN、FKにはSQLを使用
- 実装
 - mysqlにMemcachedをデーモン・プラグインとして統合
 - ネイティブInnoDB APIをmemcachedプロトコルにマッピング
 - 超低レイテンシ用の共有プロセス・スペース

MySQL 5.6: NoSQL APIによる性能

MySQL 5.6: NoSQL Benchmarking



Up to 9x Higher “SET / INSERT” Throughput

blogs.oracle.com/mysqlinnodb/entry/new_enhancements_for_innodb_memcached

ORACLE

MySQL 5.6: オプティマイザ

SQL実行性能の向上&処理時間の短縮

- サブクエリの最適化
- ファイルソートの性能向上
- Index Condition PushdownによるJOIN性能の向上
- Batched Key AccessおよびMulti-Range ReadによるJOIN性能の向上

より詳細な状況監視

- INSERT, UPDATE, DELETEでのEXPLAIN文の利用
- JSONフォーマットでのEXPLAIN文の出力
- Optimizer Traces

MySQL 5.6: Replication



パフォーマンス

- マルチスレッド スレーブ
- バイナリログのグループコミット
- 行ベース レプリケーションの転送データ量の削減

フェールオーバー & リカバリ

- Global Transaction Identifiers
- レプリケーション フェールオーバー & 管理ユーティリティ
- スレーブ&バイナリログの耐障害性向上

データの正確性

- レプリケーション チェックサム

開発 & 管理の簡素化

- 遅延レプリケーション
- リモートからのバイナリログのバックアップ
- ログへのメタデータの追加

Raising the Bar: **MySQL 5.7**

5.6 GA
Feb 2013

5.7 DMR 1
April 2013

DMR 2
Sep 2013

DMR 3
Dec 2013

...
DMR 4
New features in
labs.mysql.com
Available Now!

DMRs: Development Milestone Releases

- 年に2-4回リリース
- リリース候補版の品質になった機能のみを含む
- コミュニティでのテストや使用、フィードバックのため
- 機能拡張の加速
- 品質の改善





Warning! For testing purposes only!

- テスト目的で、より先進的/実験的な機能をリリース
 - Laboratory: 実験室
- コミュニティでのテストや使用、フィードバックのため
 - 本番環境では使用しないでください



最近の Labs Releases

- Hadoop Applier for MySQL
- MySQL Fabric
- Multi-Source Replication
- Multi-threaded Slave
- MySQL JSON UDF
- **NEW** – InnoDB GIS+Optimizer GIS+Parser Refactoring!
- **NEW** – InnoDB Page/IO Compression!
- **NEW** – MySQL Cluster 7.4.0

InnoDB GIS+Optimizer GIS+Parser Refactoring

- パーサーのリファクタリング
- InnoDBが空間インデックスをサポート
 - R-treeインデックス
- Boost.Geometryとの統合
 - Improved accuracy
 - Support for additional types, combinations, and functions
 - Improved performance
 - Full Open Geospatial Consortium compliance
 - Enhancements to Boost.Geometry pushed upstream

MySQL 5.7: InnoDB Compression

labs.mysql.com

Thank you, Fusion-io

- 透過的なページレベルの圧縮
 - バックグラウンドスレッドによって透過的に圧縮/解凍される
 - IOレイヤーによって完全に管理される
 - 既存の圧縮テーブルと同時に使用可能
 - Fusion-ioから提供されたPOCパッチ、現時点ではLinuxのみ
- IOを削減
 - パフォーマンスの改善
 - 書込み量の削減 ⇒ SSDの寿命が延びる
- テーブル、システム表領域、UNDOログの全てに適用される

MySQL 5.7: InnoDB Compression

labs.mysql.com

Thank you, Fusion-io

- 要件

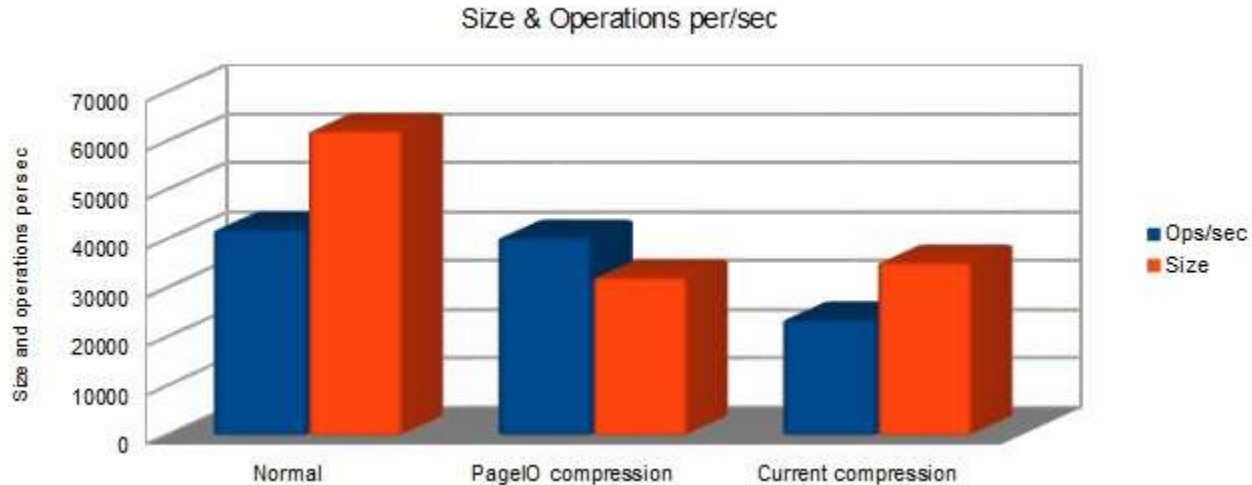
- sparse file support : NVMFS, XFS, EXT4, ZFS & NTFS
- Linux 2.6.39+ added PUNCH HOLE support

MySQL 5.7: InnoDB Compression

Thank you, Fusion-io

labs.mysql.com

FusionIO – 25G BP – maxid 50Million 64 Requesters



ORACLE

MySQL 5.7 DMR の軌跡

DMR0

Internal Release Only

- Error issued when creating duplicate index.
 - Warning when strict SQL Mode removed.
- Ctl-C fixed in MySQL CLI!
- Non-blocking SHOW SLAVE STATUS
- Stacked Diagnostics Area
- Idempotent mysqlbinlog mode.

DMR1

Released April 2013

- Online DDL Now supports RENAME INDEX and VARCHAR extensions.
- Metadata Locking (MDL) Performance Improvements.
- InnoDB Temporary Tables.
- InnoDB support for Spatial Data Types.
- Rewrite-db option for mysqlbinlog with RBR.

DMR2

Released September 2013

- Refactored internal logging + introduced `log_error_verbosity` configuration setting.
 - Timestamps are now written with log entries in YYYY-MM-DDThh:mm:ss.uuuuuu format.
- Performance Schema now instruments memory usage, stored program execution and replication.
- `mysql_secure_installation` changed to C++ program.
- Semi sync replication improvements.

DMR2(続き)

Released September 2013

- Multiple Triggers per event per table.
 - Lifts a restriction since MySQL 5.0!
- InnoDB Improvements for Large Tables, Read Only Transactions, Temporary Tables, Scalability.
- Parallel Replication (Intra-schema)
- InnoDB Locking Code refactored.
- Improved support for high connection creation rate.
- EXPLAIN FOR CONNECTION
- EXPLAIN showing cost information.

MySQL 5.7.2: InnoDB

一時表の性能向上

- 一時表のための表領域を分離
 - 表領域ibtmp1, 起動時に再作成
- 一時表の CREATE/DROP 性能を向上
 - DDLによる変更を最小限にLIOを抑制
- DML処理を一時表に最適化
 - REDOログに書き込まない、チェンジバッファを使用しない、ロックしない
- InnoDBの一時表のUNDOログは一時表の表領域に
 - 一時表の表領域のUNDOログのみ利用、REDOログには一切書込まない

MySQL 5.7.2: InnoDB Temporary Tables

性能の向上

11倍高速化

CREATE/DROP

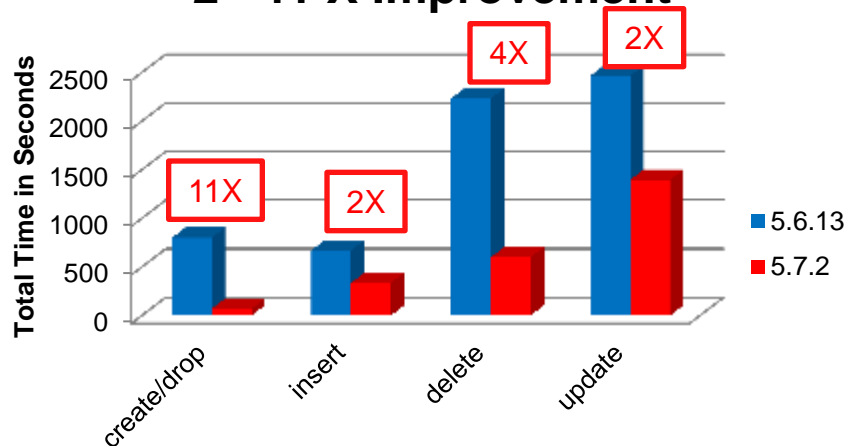
- テーブルの作成や削除処理中のIOを削減

2-4倍高速化

INSERT/DELETE/ UPDATE

- REDOログバッファの書き込み、ロックおよびチェンジバッファの利用を削減

MySQL 5.7 versus 5.6 2 - 11 X Improvement



Total time to complete in seconds

* sql-bench, modified to use temp-tables

* Seed size of 10K for Create/Drop

* Seed size of 5M for Insert/Delete/Update

MySQL 5.7.2: InnoDB

Better Performance, Availability, Checksums

- オンラインALTER TABLEの拡張
 - オンラインでのインデックス名変更、VARCHAR列の変更
- 5.6でのinnochecksum
 - InnoDB表領域のチェックサムを計算し、比較、レポート
- 5.7での拡張
 - アルゴリズムを選択可能 (innodb/crc32/none), 新しいアルゴリズムで上書き可能
 - チェックサムを上書き可能(無効な場合)
 - 処理中止前に、何件の不一致まで処理するか指定可能
 - 複数のテーブルスペース、複数のテーブルに対して実行可能
 - 2GBを超える表領域ファイルに対して実行可能
 - ページタイプごとのサマリ表を出力可能

MySQL 5.7.2: Replication

より良いパフォーマンス、使い勝手の向上、HAの強化

- スレーブのスループット向上: 同スキーマであってもマルチスレッドで処理
 - トランザクションの一貫性を保つ
 - 現在開発中(チューニング、ボトルネックの排除)
- マスターのスループット向上
 - バイナリログへアクセスする際のロックを削減
- 準同期の強化
 - “Lossless”準同期レプリケーション
- スレーブの稼働統計を確認するためのパフォーマンス・スキーマ
 - SQLを使ってスレーブのステータスを確認

MySQL 5.7.2: Performance Schema

メモリ利用状況



- メモリに関する利用統計情報
 - 200件以上のメモリ関連Instruments(統計項目)
- 利用統計情報をグルーピング
 - メモリの利用タイプ (キャッシュ、内部バッファなど)
 - スレッド/アカウント/ユーザ/ホストの間接的なメモリ利用
- Attributes(属性情報)の種類
 - 利用量 (バイト単位)、処理回数、最大値/最小値

MySQL 5.7.2: Performance Schema

ストアドプログラムの稼働統計



- SQL文の稼働統計を拡張
 - 5.6を拡張子ストアドプログラムをカバー、必要なコマンドとSQLを追加
- 新しいInstruments
 - statement/sp/cclose, cfetch, copen
 - statement/scheduler/event
- 新しいオブジェクトタイプ
 - EVENT, FUNCTION, PROCEDURE, TRIGGER
- 新しいサマリーテーブル
 - events_statements_summary_by_program

MySQL 5.7.2: Performance Schema

ストアドプログラムの稼働統計

新しいサマリーテーブル: `events_statements_summary_by_program`

- `OBJECT_TYPE`, `OBJECT_SCHEMA`, `OBJECT_NAME`
 - the primary key that identifies the SP.
- `COUNT_STAR`, `SUM/MIN/AVG/MAX_TIMER_WAIT`
 - statistics about overall execution: number of times invoked and execution time.
- `COUNT_STATEMENTS`, `SUM/MIN/AVG/MAX_STATEMENTS`
 - statistics about nested statements invoked during the SP execution
- `SUM_LOCK_TIME`, ..., `SUM_NO_GOOD_INDEX_USED`
 - all the SP statements statistics, aggregated

MySQL 5.7.2: Trigger Improvements

1つのテーブルに複数のトリガ

- これまで
 - 各トリガの組み合わせはテーブル当たり1つずつ
 - (INSERT, UPDATE, DELETE) X (BEFORE, AFTER)
- 5.7
 - 複数のトリガを指定可能
 - 実行順を指定可能

MySQL 5.7.2: Trigger Improvements

1つのテーブルに複数のトリガ

- テーブル t1 に3個の AFTER INSERT トリガ
 - CREATE TRIGGER **TRIG1** AFTER INSERT ON t1 FOR EACH ROW ... ;
 - CREATE TRIGGER **TRIG2** AFTER INSERT ON t1 FOR EACH ROW ... ;
 - CREATE TRIGGER **TRIG1A** AFTER INSERT ON t1 FOR EACH ROW **FOLLOWS TRIG1** ... ;
- 実行順は: 1) **TRIG1** 2) **TRIG1A** 3) **TRIG2**
- 関連する Information_Schema
 - SELECT CREATED FROM INFORMATION_SCHEMA.TRIGGERS
 - SELECT ACTION_ORDER FROM INFORMATION_SCHEMA.TRIGGERS
 - SHOW CREATE TRIGGER

MySQL 5.7.2: Trigger Improvements

SQL文実行の最後で制約のチェック

- テーブル 't1' には
 - 'c1' 列がNOT NULLとなっている;
 - BEFORE INSERTトリガが `SET NEW.c1 = 1;` を実行 (NULLではない)

- SQL文

```
INSERT INTO t1 VALUES (NULL) or UPDATE t1 SET c1 = NULL
```

- 5.6 - 失敗
 - "ERROR 1048 (23000): Column 'C1' cannot be null"
- 5.7 - 成功

MySQL 5.7.2: エラーログ

ログレベルの制御

- 3つのログレベル:
 1. エラー(ERROR)のみ
 2. エラー(ERROR) + 警告(WARNING)
 3. エラー(ERROR) + 警告(WARNING) + 情報(NOTE) ※デフォルト値
- 設定ファイルまたは起動オプションで指定

```
--log-error-verbosity=1
```

- 稼働中

```
SET GLOBAL log_error_verbosity=1;
```

MySQL 5.7.2: エラーログ

出力形式の改良

- UTC タイムスタンプ

```
--log_timestamps = SYSTEM | UTC (default)
```

- フォーマットの変更
 - タイムスタンプは ISO 8601 / RFC 3339 形式, マイクロ秒単位
 - ログテーブル, TIMESTAMP(6)
 - プロセスIDは起動時のみ出力
 - 可能な場合は接続IDを出力、それ以外は “0”

MySQL 5.7.2: エラーハンドリング

Stacked Diagnostics

- スタックされた例外情報を取得するコマンドを実装 (SQL標準で定義)

GET STACKED DIAGNOSTICS

- スタックされた領域から情報を取得
 - GET [CURRENT] DIAGNOSTICS は現在の診断領域から情報を取得
-
- ストアドプログラムの開発者にメリット
 - より多くの情報を取得可能
 - エラー情報を調査して修正が可能

MySQL 5.7.2: エレレポータイング

例: Stacked Diagnostics

```
CREATE TABLE ids(id INT NOT NULL, UNIQUE(id));
CREATE TABLE errors(id INT NOT NULL, text VARCHAR(2000));
```

```
CREATE PROCEDURE insert_id(id INT)
BEGIN
  DECLARE CONTINUE HANDLER FOR SQLEXCEPTION
  BEGIN
    GET STACKED DIAGNOSTICS CONDITION 1 @msg = MESSAGE_TEXT;
    INSERT INTO errors VALUES(id, @msg);
    GET STACKED DIAGNOSTICS CONDITION 1 @errno = MYSQL_ERRNO;
    IF @errno = 1213 THEN
      RESIGNAL;
    END IF;
  END;
  INSERT INTO ids VALUES(id);
END;
```

Running

```
> CALL insert_id(0);CALL insert_id(1);CALL insert_id(0);
> select text from errors;
Duplicate entry '0' for key 'id'
```

MySQL 5.7.2: エラーレポートイング

例: Clear Diagnostics Area

```
mysql> DROP TABLE no_such_table;  
ERROR 1051 (42S02): Unknown table 'test.no_such_table'  
  
mysql> CREATE EVENT e ON SCHEDULE EVERY 1 SECOND DO DELETE FROM t;  
Query OK, 0 rows affected (0.35 sec)
```

// 5.6:

```
mysql> SHOW ERRORS;  
+-----+-----+-----+  
| Level | Code | Message |  
+-----+-----+-----+  
| Error | 1051 | Unknown table 'test.no_such_table' |  
+-----+-----+-----+  
1 row in set (0.00 sec)
```

// 5.7:

```
mysql> SHOW ERRORS;  
Empty set (0.00 sec)
```

MySQL 5.7.2: Optimizer

稼働中のクエリに対するEXPLAIN

- 課題 – あるセッションのクエリの処理に長い時間がかかる
- 新しいオプション
 - 他のセッションに対してEXPLAIN FOR CONNECTION

```
EXPLAIN [FORMAT=(JSON|TRADITIONAL)] [EXTENDED] FOR CONNECTION <id>;
```

- 遅延の根本原因を発見できる
 - クエリの最適化につなげられる
- コネクションの<id> は SHOW PROCESSLIST で確認

MySQL 5.7.2: Optimizer

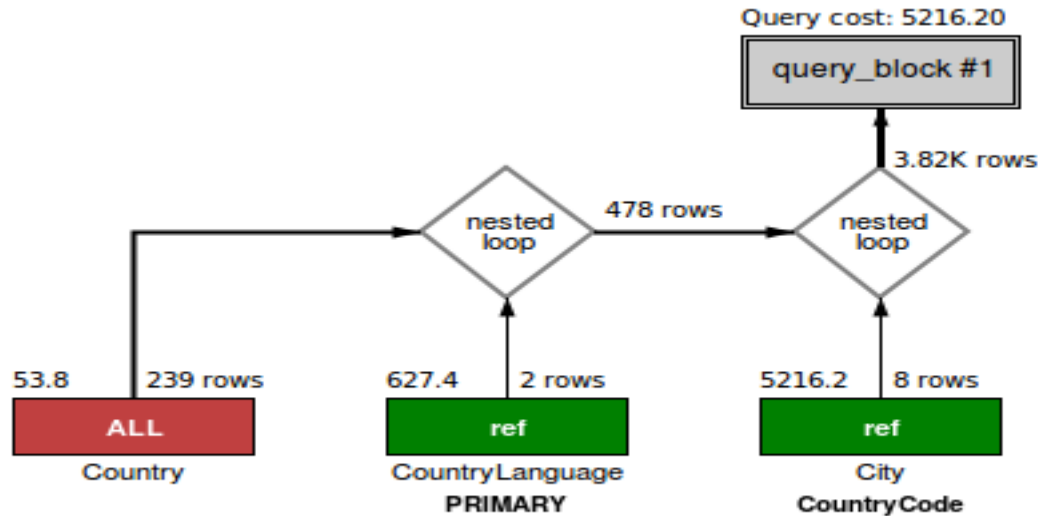
EXPLAINにコストの情報を追加 例) EXPLAIN FORMAT=JSON FOR CONNECTION 2;

***** 1. row *****

```
EXPLAIN: {
  "query_block": {
    "select_id": 1,
    "cost_info": {
      "query_cost": "4786.00"
    },
    "ordering_operation": {
      "using_temporary_table": true,
      "using_filesort": true,
      "cost_info": {
        "sort_cost": "2151.00"
      },
      "nested_loop": [
        {
          "table": {
            ..
          },
          "key_length": "3",
          "rows_examined_per_scan": 239,
          "rows_produced_per_join": 239,
          "filtered": 100,
          "using_index": true,
          "cost_info": {
            "read_cost": "6.00",
            "eval_cost": "47.80",
            "prefix_cost": "53.80",
            "data_read_per_join": "61K"
          },
          ..
        }
      ]
    }
  }
}
```


MySQL 5.7.2: Optimizer

Visual EXPLAINにコスト情報を追加、MySQL Workbenchから取得



DMR3

Released December 2013

- UNION ALL no longer requires temporary table.
- Filesort now packs VARCHAR and CHAR keys in temporary files.
- Optimizer Support for row value constructors.
 - i.e. WHERE (a,b) in (('a', 'b'), ('c', 'd'));
- Performance Schema now instruments Metadata Locking (MDL) and Transactions.
- Improvements to SSL Support.

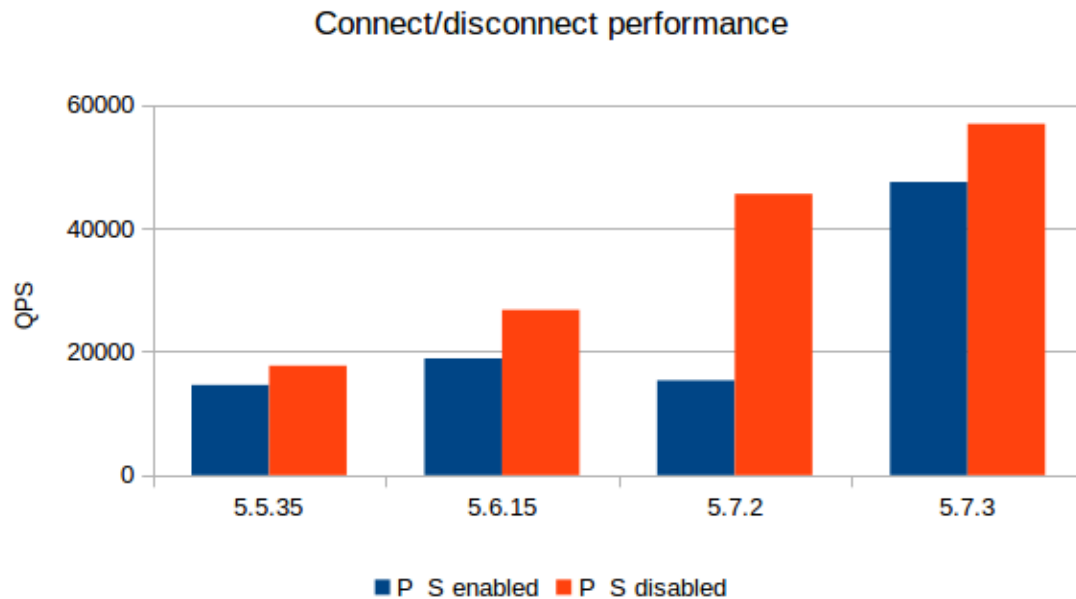
DMR3(続き)

Released December 2013

- InnoDB Scalability Improvements.
- Server Scalability.
- InnoDB Fulltext now supports external parsers.
- Dynamic replication filters.
- Support ICP optimization for partitioned tables.

MySQL 5.7.3: Connections / Second

新規コネクションの生成を高速化



5.7.3 is faster by
2.1X over 5.6
3.3X over 5.5

Built with input from Facebook

- mysql-bench
- P_S is Point Select Test Case
- 25 concurrent client threads
- Executing connect/select/disconnect
- 100000 iterations each

MySQL 5.7.3: Optimizer

IN クエリーのパフォーマンスを改善

- これまでは、WHERE句を書換えしないとレンジスキャンが使用できなかった
- 例
 - 書換え前: `WHERE (column_1, column_2) IN (('a', 'b'), ('c', 'd'))`
 - 書換え後: `WHERE (column_1 = 'a' AND column_2 = 'b') OR (column_1 = 'c' AND column_2 = 'd')`
- 5.7.3 では、書換えなくてもレンジスキャンが可能
- See [Bug#31188](#) reported by Mark Callaghan (Facebook).

例) EXPLAIN SELECT * FROM City WHERE (CountryCode,Name) IN (('USA', 'San Francisco'), ('USA', 'Chicago'), ('CAN', 'Toronto'))

MySQL 5.6:

```
***** 1. row *****
      id: 1
    select_type: SIMPLE
      table: City
      type: ALL
possible_keys: NULL
      key: NULL
     key_len: NULL
       ref: NULL
      rows: 4188
  Extra: Using where
1 row in set (0.00 sec)
```

MySQL 5.7:

```
***** 1. row *****
      id: 1
    select_type: SIMPLE
      table: City
    partitions: NULL
      type: range
possible_keys: CountryCode,CountryCode$Name
      key: CountryCode$Name
     key_len: 38
       ref: NULL
      rows: 3
  filtered: 100.00
  Extra: Using where
```

MySQL 5.7.3: Optimizer

UNION ALL 時、一時表の作成が不要に

- UNION ALLの結果に対して、一時表の作成を回避
- パフォーマンスを改善し、レスポンスタイムを短縮
 - 一時表の作成、書込み、読込みのコストを削減
 - 結果が大きい場合は、ディスクに書込まれる場合もある
 - 必要なディスク領域の削減

例) EXPLAIN SELECT * FROM a UNION ALL SELECT * FROM b

MySQL 5.6:

```
***** 1. row *****
  id: 1
  select_type: PRIMARY
  table: table_a
  type: ALL
  possible_keys: NULL
  key: NULL
  key_len: NULL
  ref: NULL
  rows: 3
  Extra: NULL
***** 2. row *****
  id: 2
  select_type: UNION
  table: table_b
  type: ALL
  possible_keys: NULL
  key: NULL
  key_len: NULL
  ref: NULL
  rows: 3
  Extra: NULL
***** 3. row *****
  id: NULL
  select_type: UNION RESULT
  table: <union1,2>
  type: ALL
  possible_keys: NULL
  key: NULL
  key_len: NULL
  ref: NULL
  rows: NULL
  Extra: Using temporary
```

MySQL 5.7:

```
***** 1. row *****
  id: 1
  select_type: PRIMARY
  table: table_a
  partitions: NULL
  type: ALL
  possible_keys: NULL
  key: NULL
  key_len: NULL
  ref: NULL
  rows: 3
  filtered: 100.00
  Extra: NULL
***** 2. row *****
  id: 2
  select_type: UNION
  table: table_b
  partitions: NULL
  type: ALL
  possible_keys: NULL
  key: NULL
  key_len: NULL
  ref: NULL
  rows: 3
  filtered: 100.00
  Extra: NULL
```


MySQL 5.7.3: Optimizer

ソートバッファ内で、ソートに使っていない列のサイズを縮小

- パフォーマンスの改善
 - ソートバッファの利用量削減
 - ソート中のディスク書き込みを避ける/減らす
- 例)
 - “SELECT * FROM t ORDER BY c1”
 - テーブルがCHAR or VARCHARのc2列, c3列を持っている場合
 - c2列, c3列が縮小される

MySQL 5.7.3 Performance Schema

メタデータロック (MDL) 稼働統計

- ロックの依存関係確認に役立つ
 - どのセッションがどのメタデータロックを持っているか
 - どのセッションがどのメタデータロックを待っているか

MySQL 5.7.3 Performance Schema

トランザクション稼働統計

- トランザクションの状況を明らかにする
 - サーバとストレージエンジンの中で
- 情報は定量的/定性的なデータも含む
 - トランザクション存続期間, トランザクション数,
 - トランザクション属性による頻度
 - 例) トランザクション分離レベル、アクセスモード
- ユーザやセッションの属性で集計可能

MySQL 5.7.3: パーティショニング

性能向上

- パーティションテーブルに対して、
Index Condition Pushdown (ICP) をサポート
- 今までの問題
 - テーブルをパーティション化すると
 - これまで
 - ICP最適化は失われる
 - これから
 - ICPはパーティションテーブルの形式でメンテナンスされる

MySQL 5.7.3: InnoDB

フルテキストインデックスの拡張

- 外部のパarserをサポート (MyISAMと同様に)

```
ALTER TABLE articles ADD FULLTEXT INDEX (body) WITH PARSER my_parser;
```

- 2つのモードで実行可能
 1. 組み込みのパarserを完全に置き換える
 - 入力を読み込み、単語に分割し、解析する
 2. 組み込みのパarserと一緒に動作する
 - 外部parserで入力からテキストを抽出し、組み込みparserに渡す
 - 組み込みparserは、ノーマルのルールを使って分割、解析する

<http://dev.mysql.com/doc/refman/5.7/en/writing-full-text-plugins.html>

MySQL 5.7.3: InnoDB

Faster redo logging

- Redoログ書込み部分のコードをリファクタリング
- 以下のどちらでもパフォーマンスを改善
 - `log_sys::mutex` が激しく使われている
 - もしくは
 - `innodb_flush_log_at_trx_commit = 2`

MySQL 5.7.3: Replication

セキュリティの向上、動的にフィルターを変更

- トランスポートレイヤーのセキュリティ
 - mysqlbinlog がSSLをサポート
 - 完全なSSLサポート
 - mysqlbinlog は最後のクライアントプログラム
- レプリケーションフィルターを動的に変更
 - スレーブの `-replicate-*` オプションを設定するための新コマンド
 - CHANGE REPLICATION FILTER
 - 設定変更のためにサーバの再起動不要

MySQL 5.7.3: Replication

準同期

- 新しい準同期のオプション
 - マスターがN個のスレーブから応答が返ってくるのを待つ
 - 従来は1つのスレーブだけ待っていた
- N個のスレーブを待つメリット ($N > 1$)
 - 2重障害に対する冗長性
 - トランザクションの永続性向上
 - トランザクションは2つ以上のスレーブに存続する

MySQL 5.7.3: Security

暗号化をクライアントに強制

- 今までの “--ssl” オプション
 - ssl無しでコネクションを確立することもある
 - 先ずはsslで接続を試みるが、サーバでsslが有効でなければ、ssl無効になる
- 新しい “--ssl” オプション
 - 指定すると、ssl経由でしか接続できない

DMR4

Released April 2014

- Test suite migrated to InnoDB.
- Atomics refactored / PS overhead reduction.
- Password expiration support.
 - Timestamp of password_last_changed in mysql.user table.
- Multiple page cleaner threads in InnoDB.
- Performance Schema now instruments Prepared Statements.
- Performance Schema improved to do lazy initialization on new connections.

DMR4(続き)

Released April 2014

- Metadata Locking (MDL) scalability improvements.
- Statement Timeout.
- Optimizer refactoring (multiple improvements)
- Change master without stopping SQL thread.
- Support to change AES key length and block encryption mode.
- Changed compiler from sun studio to gcc on Solaris.
- InnoDB code refactoring of tablespace class.

MySQL 5.7.4: Server-Side Statement Timeout

Incorporates Feedback from Davi Arnaut, the MySQL Community

- Server Side statement timeouts
 - グローバル単位、セッション単位、SELECT文単位で使用可能

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

- Based on contribution from Davi Arnaut
 - Expanded to Windows and Solaris

MySQL 5.7.4: InnoDB

Many Improvements

- Fusion-io との統合強化
 - DirectFS を検出すると、Doublewrite buffer を自動的に無効に
- “Dirty Page” フラッシングの並列化
 - スループット、パフォーマンス、スケーラビリティの向上
- パーティション – トランスポートابل・テーブルスペース(TTS)のサポート
 - パーティションテーブル全体だけでなく、個々のパーティション単位でも別のサーバーへ移動可能

MySQL 5.7: Improved MDL locking

- Removes bottlenecks around DML access to a single table
 - 10% increased throughput in OLTP_RO/POINT_SELECT sysbench tests on higher core counts
 - Optimized for typical DML heavy workloads
- Implemented fast-path for DML locks
- Implemented lock-free DML lock acquisition
- Implemented a lock-free hash
 - Now uses MurmurHash library

MySQL 5.7.4: セキュリティ

- パスワードのローテーションポリシー
 - `default_password_lifetime` システム変数でパスワードの有効期限を指定
 - 5.7.4でのデフォルト値は360(日)
- セキュアなインストールがデフォルトに
 - root ユーザーにランダムなパスワードが設定され、パスワードを変更するまで何も作業ができない
 - anonymous ユーザーを作成しない
 - test データベースを作成しない

MySQL 5.7: DMR 4

MySQL 5.6をベースに各種機能を改良

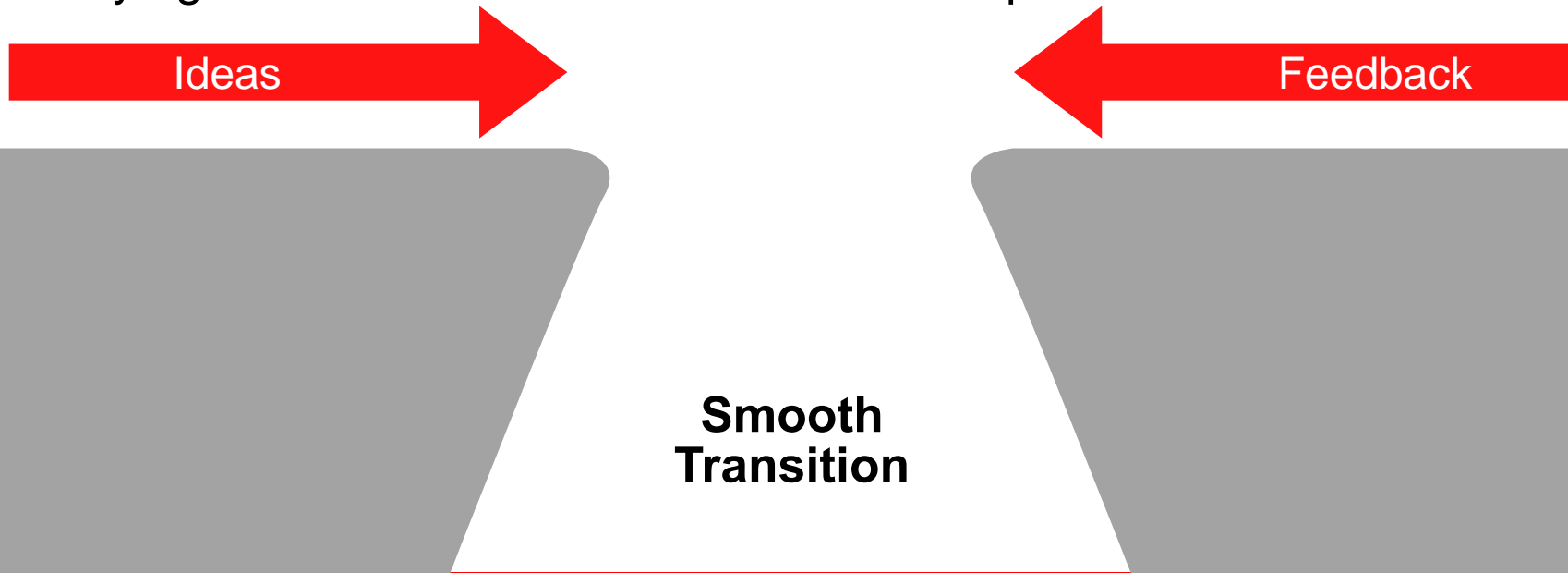
- **InnoDB** より高いトランザクション処理性能と可用性
- **Replication** より高い拡張性と可用性
- **Utilities** DevOpsの作業を自動化
- **Performance Schema** より詳細な性能統計情報
- **Optimizer** クエリパフォーマンスの改良、EXPLAINの可視化、強化されたバッファリングとパーティションの最適化
- **Connecting** セッション生成をより効率的に

Available Now! Get it here: dev.mysql.com/downloads/mysql/

We Want Your Feedback!

Help make the transition as smooth as possible.

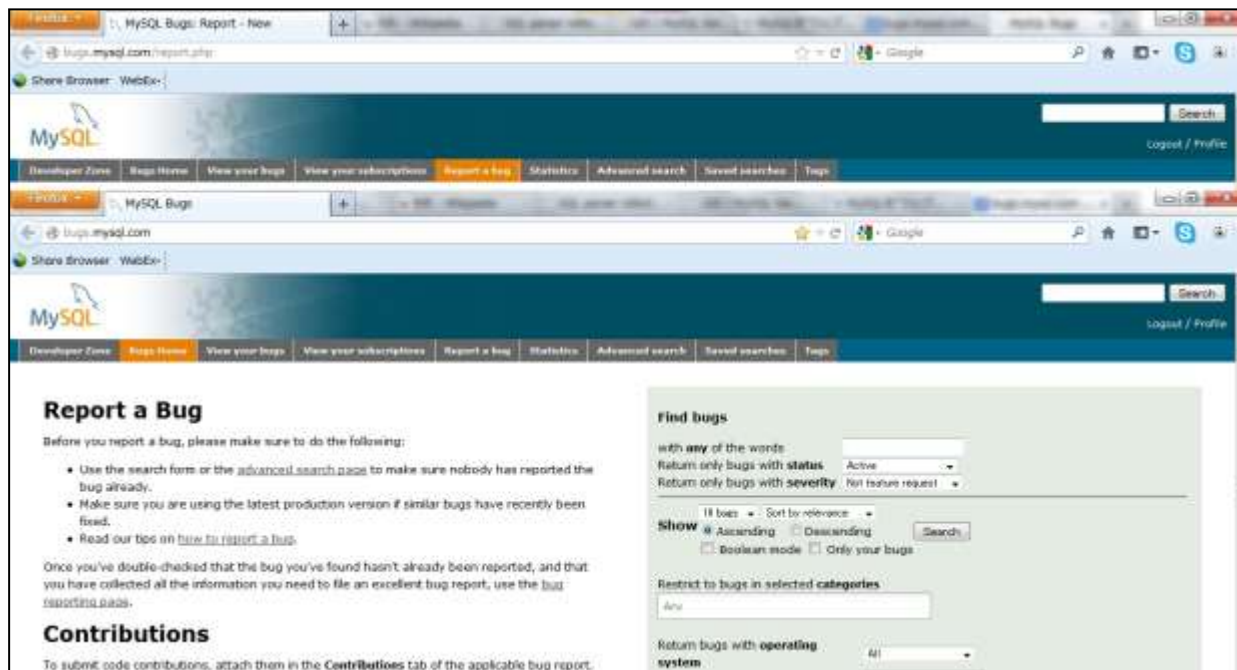
- Sharing our ideas.
Providing you an opportunity for feedback.
Trying to make the transition as smooth as possible.



バグ報告/機能追加要望はこちらまで

■ MySQL Bugs

<http://bugs.mysql.com/>



The screenshot displays two browser windows. The top window is titled 'MySQL Bugs: Report - New' and shows the 'Report a Bug' page. The bottom window is titled 'MySQL Bugs' and shows the 'Find bugs' page.

Report a Bug

Before you report a bug, please make sure to do the following:

- Use the search form or the [advanced search page](#) to make sure nobody has reported the bug already.
- Make sure you are using the latest production version if similar bugs have recently been fixed.
- Read our [tips on how to report a bug](#).

Once you've double-checked that the bug you've found hasn't already been reported, and that you have collected all the information you need to file an excellent bug report, use the [bug reporting page](#).

Contributions

To submit code contributions, attach them in the **Contributions** tab of the applicable bug report.

Find bugs

with **any** of the words:

Return only bugs with **status**:

Return only bugs with **severity**:

Show Sort by relevance

Ascending Descending

Boolean mode Only your bugs

Restrict to bugs in selected **categories**:

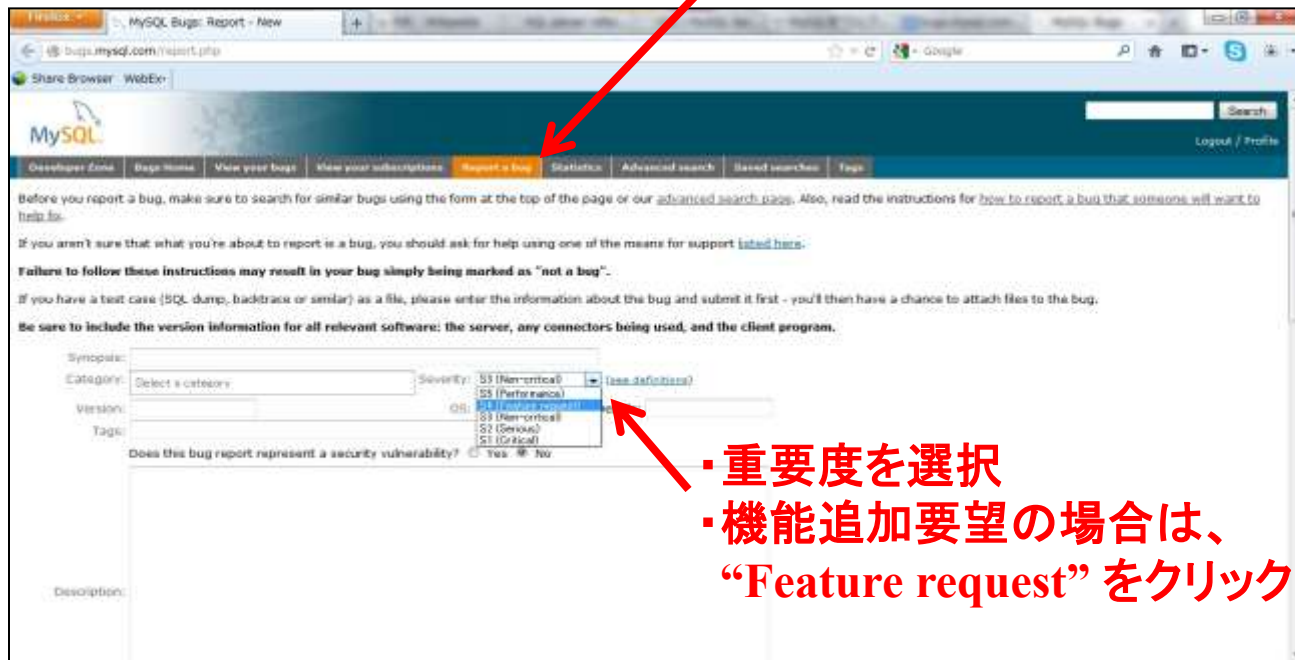
Return bugs with **operating system**:

バグ報告/機能追加要望はこちらまで

- MySQL Bugs

<http://bugs.mysql.com/>

“Report a bug” をクリック



The screenshot shows the MySQL Bugs report form. A red arrow points to the "Report a bug" button in the navigation bar. Another red arrow points to the "Severity" dropdown menu, which is open and showing options: "S3 (Minor)", "S2 (Performance)", "S1 (Feature request)", "S3 (Non-critical)", "S2 (General)", and "S1 (Critical)".

Before you report a bug, make sure to search for similar bugs using the form at the top of the page or our [advanced search page](#). Also, read the instructions for [how to report a bug that someone will want to help fix](#).

If you aren't sure that what you're about to report is a bug, you should ask for help using one of the means for support [listed here](#).

Failure to follow these instructions may result in your bug simply being marked as "not a bug".

If you have a test case (SQL dump, backtrace or similar) as a file, please enter the information about the bug and submit it first - you'll then have a chance to attach files to the bug.

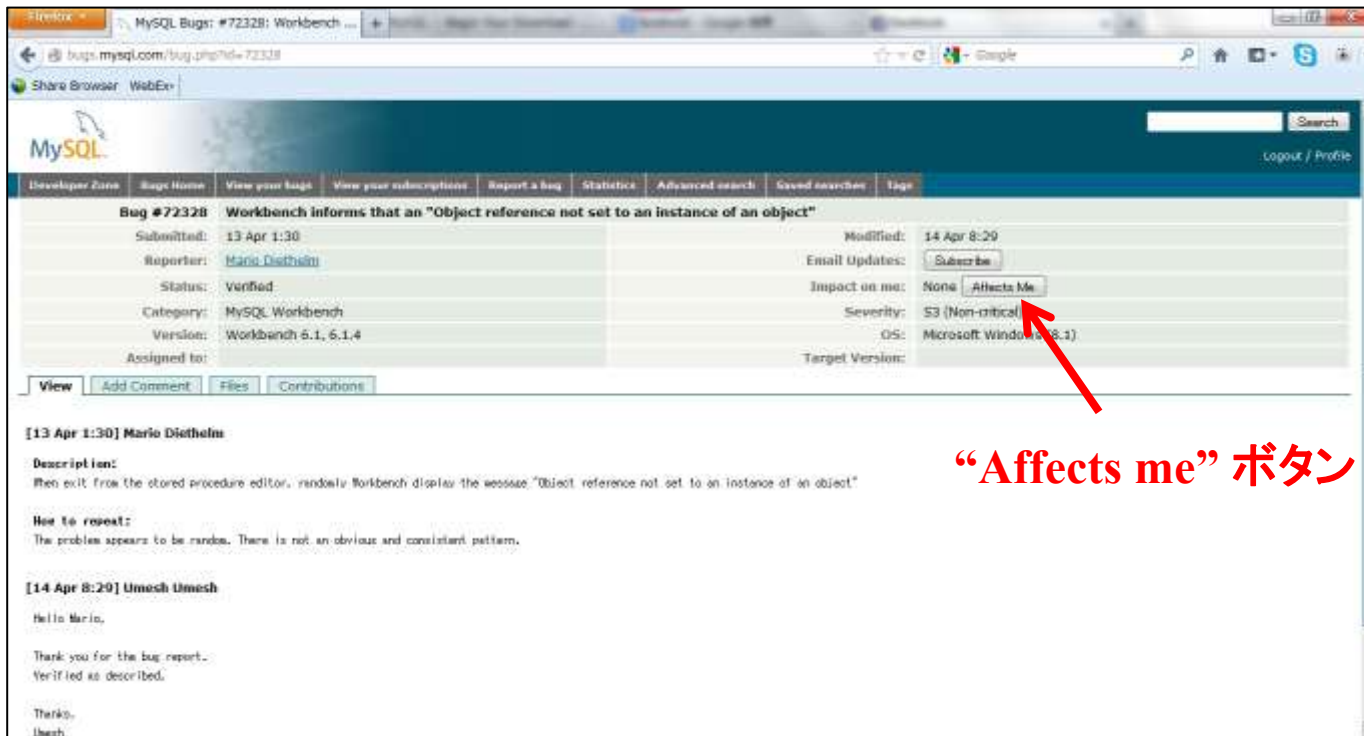
Be sure to include the version information for all relevant software: the server, any connectors being used, and the client program.

Synopsis:
Category: Severity: [\(see definitions\)](#)
Version:
Tag:
Does this bug report represent a security vulnerability? Yes No

Description:

重要度を選択
機能追加要望の場合は、
“Feature request” をクリック

既に報告されているバグ、機能追加要望の場合は、 ”Affects me”ボタンをクリック



The screenshot shows the MySQL Bugzilla interface for bug #72328. The bug title is "Workbench informs that an 'Object reference not set to an instance of an object'". The bug was submitted on 13 Apr 1:30 by Mario Diethelm. The status is "Verified". The category is "MySQL Workbench" and the version is "Workbench 5.1, 6.1.4". The target OS is "Microsoft Windows (8.1)". The impact on me is set to "None", and there is a red arrow pointing to the "Affects Me" button. The description of the bug is: "When exit from the stored procedure editor, randomly Workbench display the message 'Object reference not set to an instance of an object'". The problem appears to be random. There is not an obvious and consistent pattern. A response from Umesh Umesh on 14 Apr 8:29 says: "Hello Mario, Thank you for the bug report. Verified as described. Thanks, Umesh".

Submitted:	13 Apr 1:30	Modified:	14 Apr 8:29
Reporter:	Mario Diethelm	Email Updates:	<input type="button" value="Subscribe"/>
Status:	Verified	Impact on me:	None <input type="button" value="Affects Me"/>
Category:	MySQL Workbench	Severity:	S3 (Non-critical)
Version:	Workbench 5.1, 6.1.4	OS:	Microsoft Windows (8.1)
Assigned to:		Target Version:	

“Affects me” ボタン

ORACLE®