



MySQL™ Connect




What's New in MySQL 5.7

梶山 隆輔

MySQL Sales Consulting Manager, JAPAC

ORACLE®





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は、米国オラクル・コーポレーション及びその子会社、関連会社の米国及びその他の国における登録商標または商標です。他社名又は製品名は、それぞれ各社の商標である場合があります。



Over 3.5 Years of MySQL Innovation

MySQL Cluster 7.3

MySQL Workbench 6.0

MySQL Migration
Wizard

MySQL 5.6

MySQL 5.5

Windows installer & Tools

MySQL Cluster
Manager

MySQL
Applier for
Hadoop

MySQL Enterprise Monitor 2.3

MySQL Enterprise

Backup

MySQL Utilities

MySQL Workbench 5.2

Security

Scalability

MySQL Cluster 7.2
MySQL Cluster 7.1

MySQL Enterprise
Oracle Certifications

HA

Audit

DMRs – Development Milestone Releases

- リリース候補版の品質になった機能のみを含む
- 年に2-4回リリース
- 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



MySQL Database 5.7 DMR 新機能





MySQL 5.7.2 DMR

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
- レプリケーション より高い拡張性と可用性
- Performance Schema より詳細な性能統計情報
- オプティマイザ クエリパフォーマンスのEXPLAIN可視化の改良

Available Now! Get it here:

dev.mysql.com/downloads/mysql/



ベンチマーク

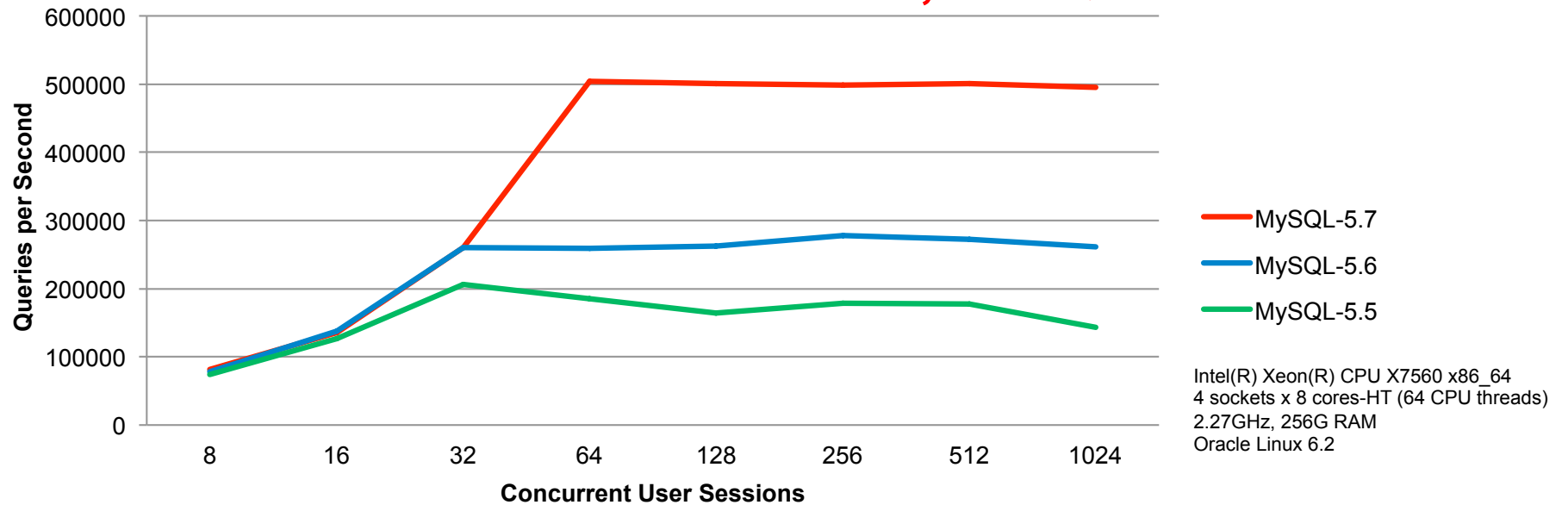




MySQL 5.7.2 Sysbench Benchmarks

Sysbench Point Select

500,000 QPS

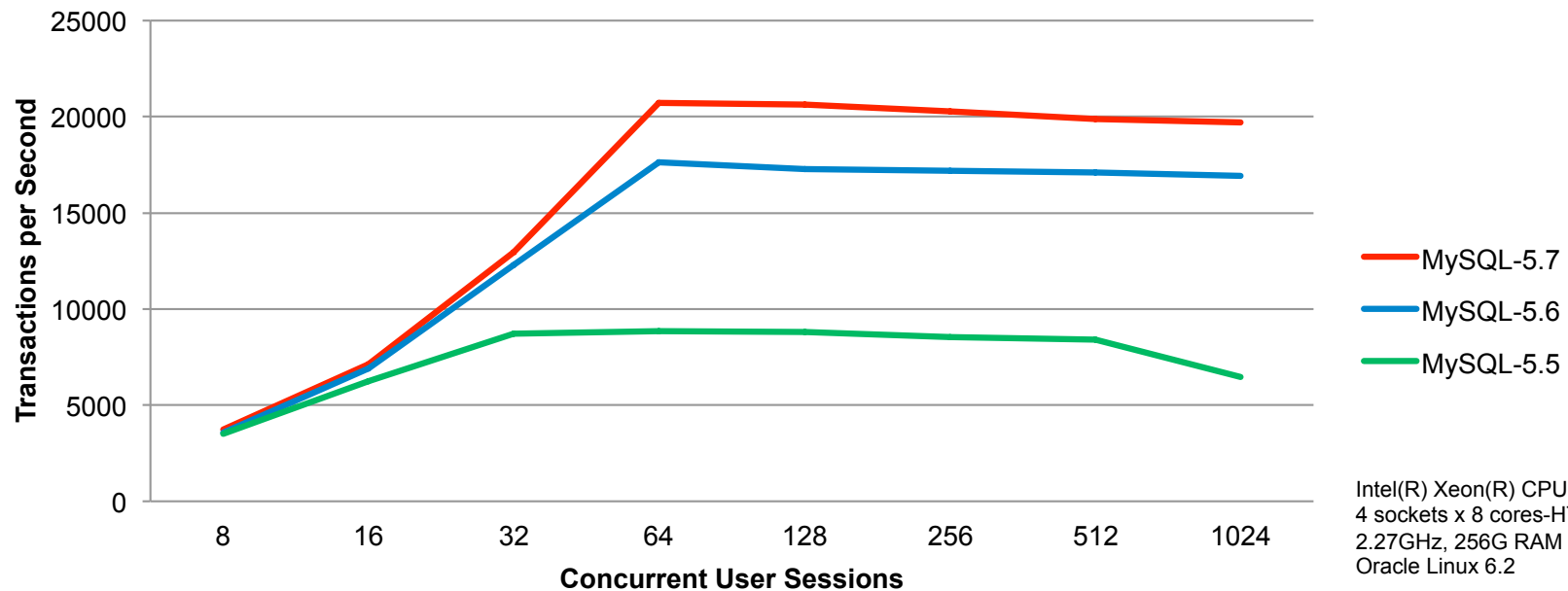


95% Faster than MySQL 5.6
172% Faster than MySQL 5.5



MySQL 5.7.2 Sysbench Benchmarks

OLTP Read Only

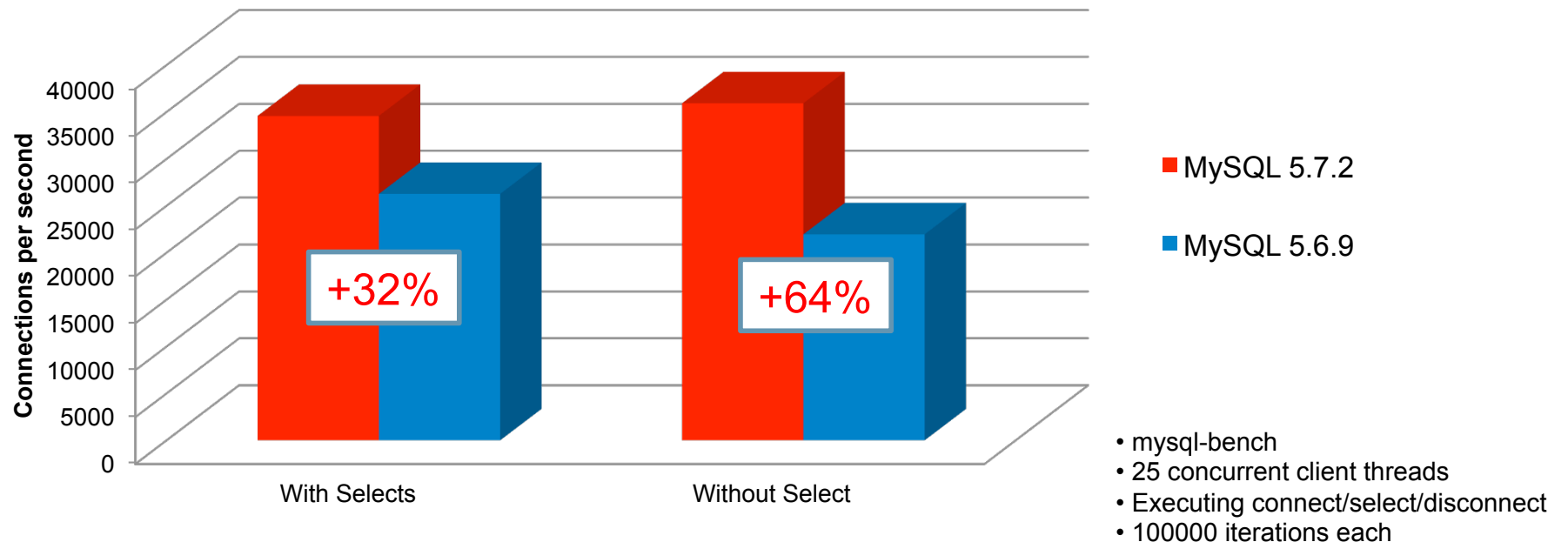


Intel(R) Xeon(R) CPU X7560 x86_64
4 sockets x 8 cores-HT (64 CPU threads)
2.27GHz, 256G RAM
Oracle Linux 6.2
sb_OLTP_RO_1M_8tab-ps Max-TPS

17% Faster than MySQL 5.6
134% Faster than MySQL 5.5

MySQL 5.7.2: Connections / second

新規接続の処理の高速化



Facebookからの要望をベースに開発

THDの初期化処理とネットワークの初期化処理をワーカースレッドに分離

MySQL 5.7.2: InnoDB Temporary Tables

性能の向上

11倍 高速化

CREATE/DROP

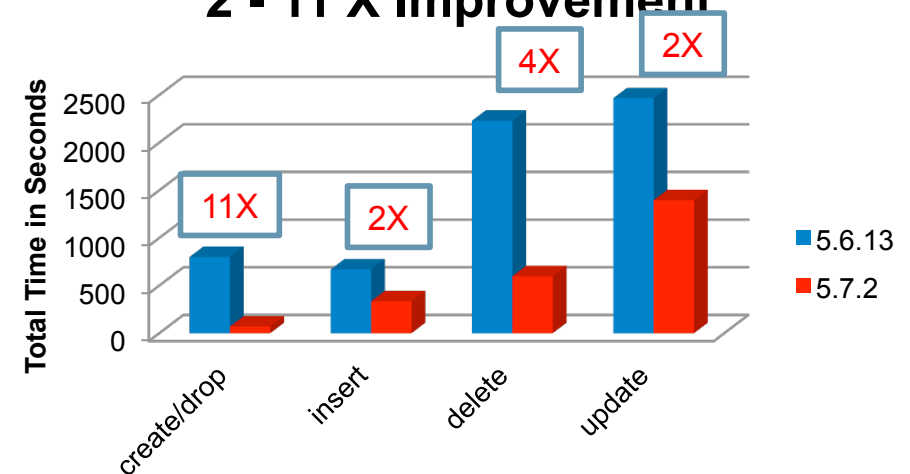
- テーブルの作成や削除処理中のIOを削減

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

オプティマイザ





MySQL 5.7.2: オプティマイザ

稼働中のクエリに対するEXPLAIN

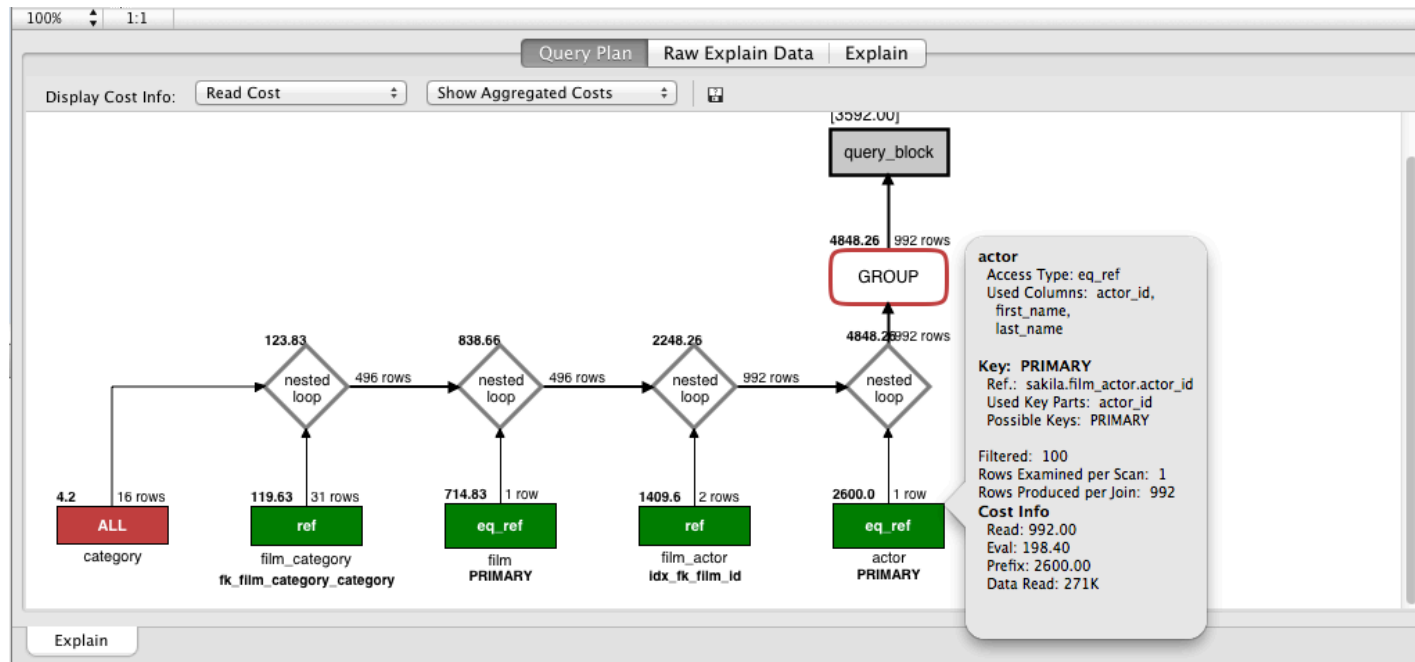
- 課題 - あるセッションのクエリの処理に長い時間がかかる
- 新しいオプション
 - 他のセッションに対して **EXPLAIN FOR CONNECTION**

```
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: オプティマイザ

JSON EXPLAINの拡張



Total query cost and Per table

Volume (in gb/mb/kb) = (# of rows) * (record width)



Performance Schema



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

統計情報の例

```
SELECT * FROM setup_instruments;
```

- クエリの最適化やクエリ実行のメインのバッファに必要な領域
memory/sql/thd::main_mem_root
- プリペアードステートメントに必要な領域
memory/sql/Prepared_statement::main_mem_root
- ソートのバッファ領域
memory/sql/Filesort_buffer::sort_keys
- JOINバッファ領域
memory/sql/JOIN_CACHE

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を拡張し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

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
 - ストアドプログラムに対するプライマリキー
- COUNT_STAR, SUM/MIN/AVG/MAX TIMER_WAIT
 - 処理全体の統計: 実行回数と実行時間
- COUNT_STATEMENTS, SUM/MIN/AVG/MAX STATEMENTS
 - ストアドプログラム内で実行されたSQL文の統計
- SUM_LOCK_TIME, ..., SUM_NO_GOOD_INDEX_USED
 - 全てのストアドプログラムの統計の合算



InnoDB




MySQL 5.7.2: InnoDB

Better Performance, Scalability



- Sysbench READ ONLY の性能が 50-100% 向上(8テーブル)
 - Sysbench POINT SELECTにて秒間 約500,000トランザクション
 - Sysbench OLTP_ROにて秒間 20,000トランザクション超
- 5.7では明示的に **START TRANSACTION READ ONLY** を実行する必要無し



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しIOを抑制
- DMLの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利用しない、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ログには一切書き込まない



MySQL 5.7.2: InnoDB


オンラインALTER TABLE の拡張

- オンラインでのインデックス名の変更

```
ALTER TABLE t RENAME INDEX i1 to i2;
```

- オンラインでのVARCHAR列の変更

```
ALTER TABLE t1 ALGORITHM=INPLACE,  
CHANGE COLUMN c1 c1 VARCHAR(255);
```

MySQL 5.7.2: InnoDB

Innochecksum の改良

- 5.6の innochecksum
 - InnoDBの表領域ページのチェックサムを算出、記録し出力
- 5.7での改良点:
 - アルゴリズムを選択可能に (innodb/crc32/none)
 - 現在のチェックサムを新しいアルゴリズムで上書き可能
 - 新しいチェックサムで強制的に上書き可能 (無効の場合)
 - 処理中止前に、何件の不一致まで処理するか指定可能
 - 複数の表領域、および複数の表領域ファイルに対して実行可能
 - 2GB 以上の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



レプリケーション



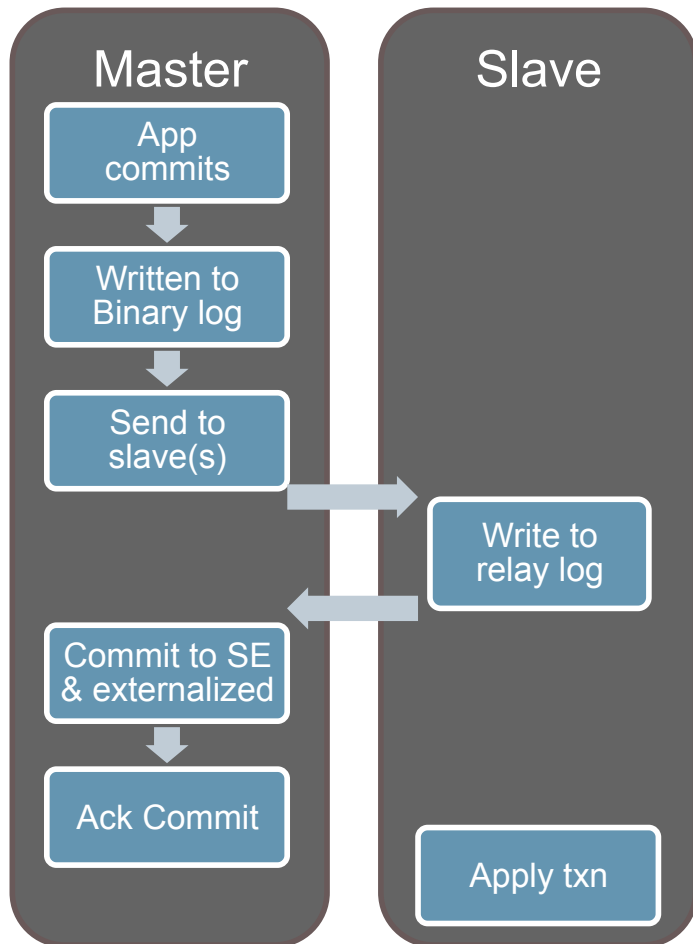


同スキーマ内でのマルチスレッド スレーブ

スレーブのスループットを向上

- スレーブ内の複数のワーカースレッドがトランザクションを並列で処理
 - 並列処理されるトランザクションの対象データが同一で無い場合に限る
 - **同スキーマ内**でもマルチスレッドで処理可能
- トランザクションの一貫性を保って処理
- スレーブのスループットを向上
- 現在開発中
 - リファクタリング完了
 - チューニングや最適化は現在作業中

“Lossless”準同期レプリケーション



- マスターは指定のスレーブがトランザクションを受信するまで以下の処理を待つ
 - ストレージエンジンへのコミット
 - 他のクライアントから変更へのアクセス
 - アプリケーションへのコミットの応答
- スレーブが変更点を反映させるまでは待たない
 - 遅延を最小化
- スレーブに安全にコピーされるまで他のトランザクションが新しいデータを変更できないように



セキュリティ



MySQL 5.7.2: セキュリティ – 監査ログ

MySQL Enterprise Audit

- MySQL監査プラグインを有効にして監査ログを出力
 - 「誰が、いつ、何を、どこで、どのような」
 - ユーザ、日時、ホスト、ステータス ...
 - 接続/切断、クエリ、DML、DDL、サーバ&監査の開始/停止...
- MySQL 5.7 での改良点
 - Oracle Audit Vault との統合を改良
 - XMLのサブ要素を使用
 - タイムゾーン



MySQL 5.7.2: セキュリティ – 監査ログ

5.6 と 5.7 の比較

5.6

```
<AUDIT_RECORD  
TIMESTAMP="2013-04-15T15:27:27"  
NAME="Query"  
CONNECTION_ID="3"  
STATUS="0"  
SQLTEXT="SELECT 1"  
>
```

5.7

```
- <AUDIT_RECORD>  
  <TIMESTAMP>2013-04-15T15:27:27 UTC</TIMESTAMP>  
  <RECORD_ID>3998_2013-04-15T15:27:27</RECORD_ID>  
  <NAME>Query</NAME>  
  <CONNECTION_ID>3</CONNECTION_ID>  
  <STATUS>0</STATUS>  
  <STATUS_CODE>0</STATUS_CODE>  
  <USER>root[root] @ localhost [127.0.0.1]</USER>  
  <OS_LOGIN />  
  <HOST>localhost</HOST>  
  <IP>127.0.0.1</IP>  
  <COMMAND_CLASS>select</COMMAND_CLASS>  
  <SQLTEXT>SELECT 1</SQLTEXT>  
</AUDIT_RECORD>
```




その他の改良点





MySQL 5.7.2: トリガの改良

一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: トリガーの改良

一つのテーブルに複数のトリガー

- テーブル t1 に3個の AFTER INSERT トリガー
 - 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:トリガの改良

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) または 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: Error Reporting

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: Error Reporting

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 Utilities & MySQL Fabric



MySQL Utilities

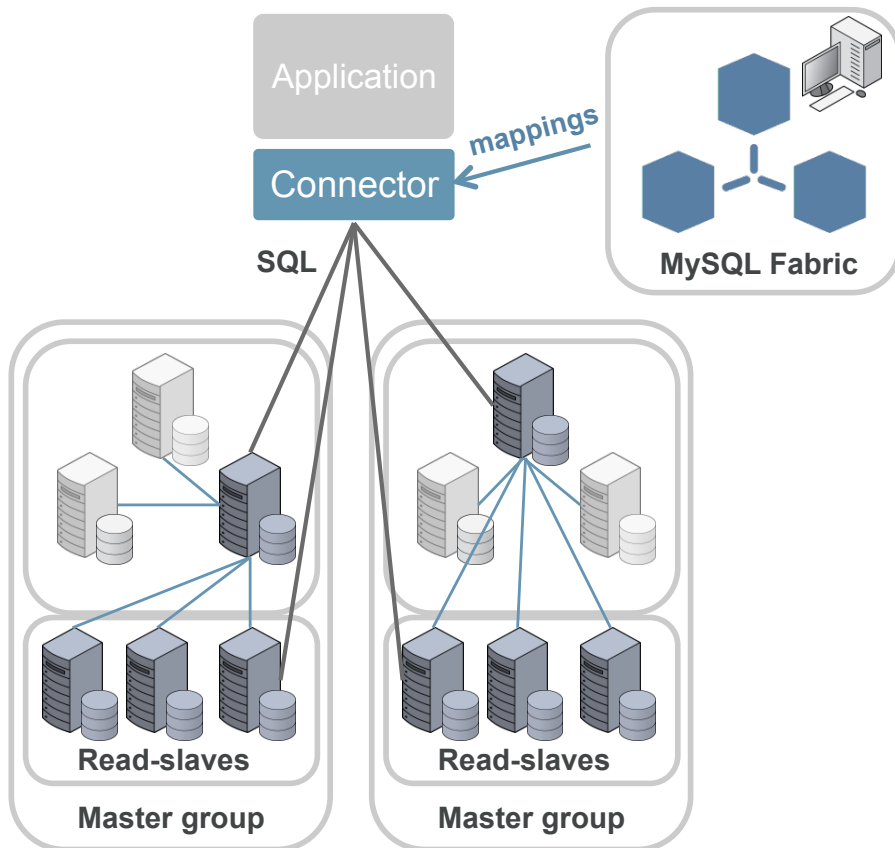
Powerful DevOps Management tools for MySQL

- New, standalone package: MySQL Utilities 1.3
- Automate common Dev/Ops tasks
 - Replication: provisioning, testing, monitoring and failover (now as daemon)
 - Database comparisons: consistency checking
 - Database administration: users, connections, tables, etc
 - Auditing
- Python scripts
 - Standalone or launched from MySQL Workbench
 - Available from dev.mysql.com
 - Extensible to include custom scripting; Python library for extensibility

MySQL Utilities - Fabric

Scale-out using Sharding

labs.mysql.com



- Scale-out through sharding
- Connectors
 - Python
 - Java
 - PHP
- Application provides shard key
 - Range or Hash
 - Tools for resharding
 - Global updates & tables
- Available in MySQL Utilities 1.4.0



開発の優先課題





MySQL開発の優先課題

- Web、クラウドインフラ、組み込みへの最適化
- よりシンプルに、プラグブル アーキテクチャ
 - メンテナンス性向上、さらなる拡張性
 - さらなるNoSQLオプション(HTTP, JSON, JavaScript, etc.)
- リファクタリング
 - InnoDBのデータディクショナリ
 - オプティマイザ/パーサ/プロトコル
- InnoDB
 - 性能、オンラインDDL、バッファプール
- 高可用性構成、レプリケーション、シャーディング

ORACLE®



MySQL Connect

