

ORACLE®

ORACLE®

MySQL 5.6 GA技術アップデート

MySQL Global Business Unit
MySQL Senior Sales Consultant, JAPAC
山崎 由章 / Yoshiaki Yamasaki



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。

Continuous Improvement

DRIVING MySQL INNOVATION

MySQL Enterprise Monitor 2.2
MySQL Cluster 7.1
MySQL Cluster Manager 1.0
MySQL Workbench 5.2
MySQL Database 5.5
MySQL Enterprise Backup 3.5
MySQL Enterprise Monitor 2.3
MySQL Cluster Manager 1.1

All GA!
2010

MySQL Enterprise Backup 3.7
Oracle VM Template for MySQL Enterprise Edition
MySQL Enterprise Oracle Certifications
MySQL Windows Installer
MySQL Enterprise Security
MySQL Enterprise Scalability

All GA!

MySQL Database 5.6 DMR*
MySQL Cluster 7.2 DMR

MySQL Labs!

("early and often")
2011

*Development Milestone Release

MySQL Cluster 7.2
MySQL Cluster Manager 1.4
MySQL Utilities 1.0.6
MySQL Migration Wizard
MySQL Enterprise Backup 3.8
MySQL Enterprise Audit
MySQL Database 5.6

All GA!

MySQL Cluster 7.3 DMR
Available Now!

**A BETTER
MySQL**
2012-13

ORACLE



40% DATA GROWTH
PER YEAR



400m TWEETS
PER DAY



5.9BN MOBILE SUBS IN 2011

OVER 1.2 BILLION iOS & ANDROID APPS IN 2012



2.2BN USERS
630M WEB SITES

Web
Mobile
Traffic
Revenue
Communications
eMail
Data
eCommerce
Social
Networking
Growth



955M USERS
552M DAILY ACTIVE

72 HOURS
UPLOADED EVERY
MINUTE



\$1TR BY 2013
\$700BN IN 2011



85% HANDSETS
SHIPPED WITH A BROWSER



MySQLのコミュニティ活動

- 全世界で1,500万インストール (弊社推計)
- Facebookのファン数: 163,000, 昨年比+35%増
- Twitterのフォロワー数: 25,000, 昨年比+67%増
- MySQL Newsletter: 140万サブスクライバ, 昨年比+20%増
- イベント: MySQL Dev Days & MySQL Tech Tours etc
- 雑誌などでの認知: CRN, Linux Journal, PHP Architect...
- MySQLはクラウドでの利用も拡大中

MySQL 5.6 GA



MySQL 5.6 GA

New!

- **オプティマイザ**: パフォーマンス&スケーラビリティ
- **パフォーマンス・スキーマ**: より詳細な統計情報
- **InnoDB**: トランザクション・スループットの向上
- **レプリケーション**: さらなる可用性とデータの整合性
- **「NotOnlySQL」オプション**: さらなる柔軟性

- ダウンロードはこちらから！

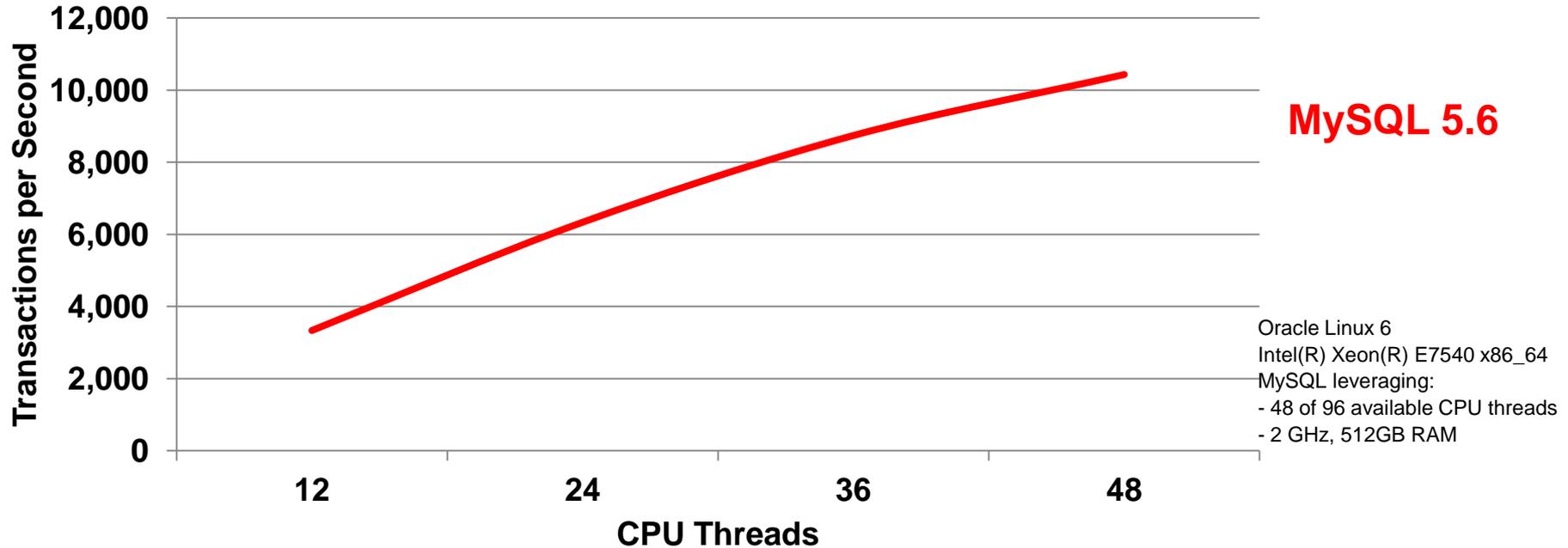
dev.mysql.com/downloads/mysql/



ORACLE

MySQL 5.6: Scalability

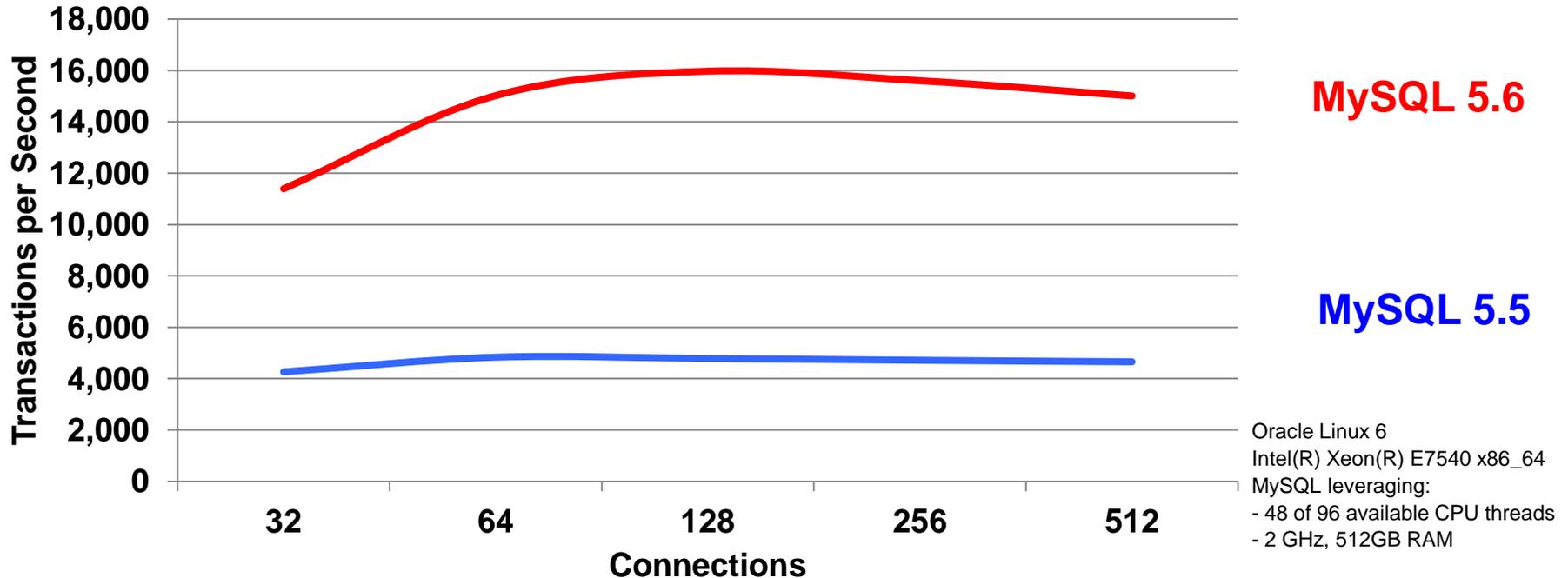
MySQL 5.6 Read Write (Linux)



■ 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)



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

MySQL : 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

Better Performance, Scalability

- 複数の内部実装の改良 (例:カーネルミューテックスの分割、バッファプールのフラッシュの効率改善など)
- 参照専用トランザクションの実装
- オプティマイザ統計の永続化
 - 安定して正確な実行計画
 - ユーザから制御、自動/手動
- SSDへの最適化
 - 4, 8kページサイズ
 - .ibdファイルをデータディレクトリ以外へ
 - UNDOログ表領域を分離



MySQL 5.6: InnoDB

参照処理の性能向上

- 参照処理の同時実行が多いWebアプリケーションなどで効果大
- 開発者が参照専用トランザクションを選択することでオーバーヘッド削減

```
SET autocommit = 1;  
SELECT c FROM sbtest WHERE id=N;
```

デフォルト

```
SET autocommit = 0;  
START TRANSACTION READ ONLY;  
SELECT c FROM sbtest WHERE id=N;  
COMMIT;
```

参照専用トランザクション開始

<http://dev.mysql.com/doc/refman/5.6/en/innodb-performance.html#innodb-performance-ro-txn>

MySQL 5.6: InnoDB

オンラインでのDDL実行

- CREATE INDEX
- DROP INDEX
- AUTO_INCREMENTの値を変更
- ADD/DROP FOREIGN KEY
- RENAME COLUMN
- テーブルのROW FORMAT, KEY_BLOCK_SIZEを変更
- 列のNULL, NOT_NULLを変更
- 列の追加、削除、並び替え

スキーマの変更をオンラインで可能。停止時間を削減

MySQL 5.6: InnoDB

バッファプールのダンプ&リストア

- 起動直後からバッファプールにデータがキャッシュされた状態に
- シャットダウン/起動時に自動で、または手動で
- ディスク上にはテーブルスペースのページIDのみを書き出す

シャットダウン時に自動的にバッファプールの内容をダンプ:

```
mysql> SET innodb_buffer_pool_dump_at_shutdown=ON;
```

起動時にダンプされた内容をバッファプールにロード:

```
mysql> SET innodb_buffer_pool_load_at_startup=ON;
```

- 再起動直後や新しいサーバの起動直後でも性能劣化しない
- クラウド、ホスティング、SaaSなどの環境でもメリット

MySQL 5.6: InnoDB

テーブルスペースの可搬性の向上

- MySQLサーバ間でテーブルの移動やコピーが簡単に可能

Export:

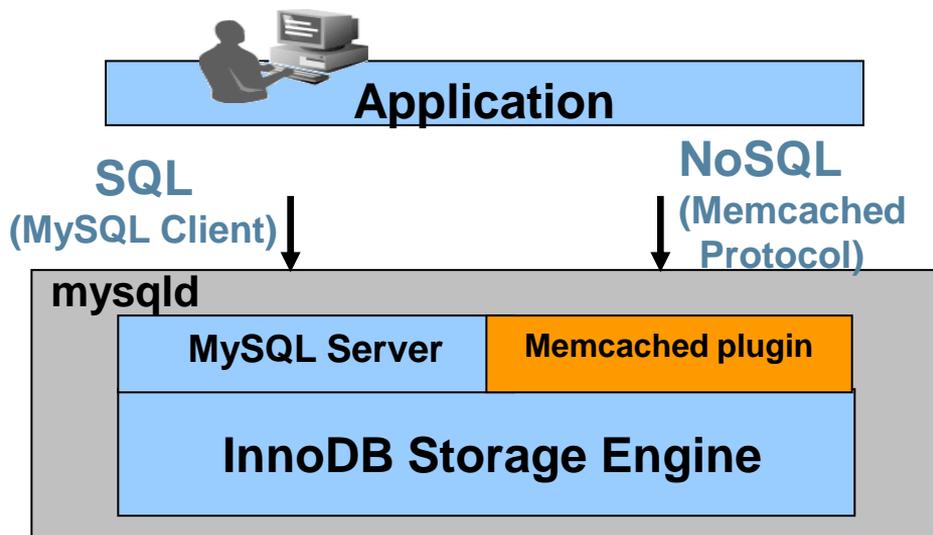
```
CREATE TABLE t(c1 INT) engine=InnoDB;  
FLUSH TABLE t FOR EXPORT; -- quiesce the table and create the meta data file  
$innodb_data_home_dir/test/t.cfg  
UNLOCK TABLES;
```

Import:

```
CREATE TABLE t(c1 INT) engine=InnoDB; -- if it doesn't already exist  
ALTER TABLE t DISCARD TABLESPACE;  
-- The user must stop all updates on the tables, prior to the IMPORT  
ALTER TABLE t IMPORT TABLESPACE;
```

MySQL 5.6: InnoDB

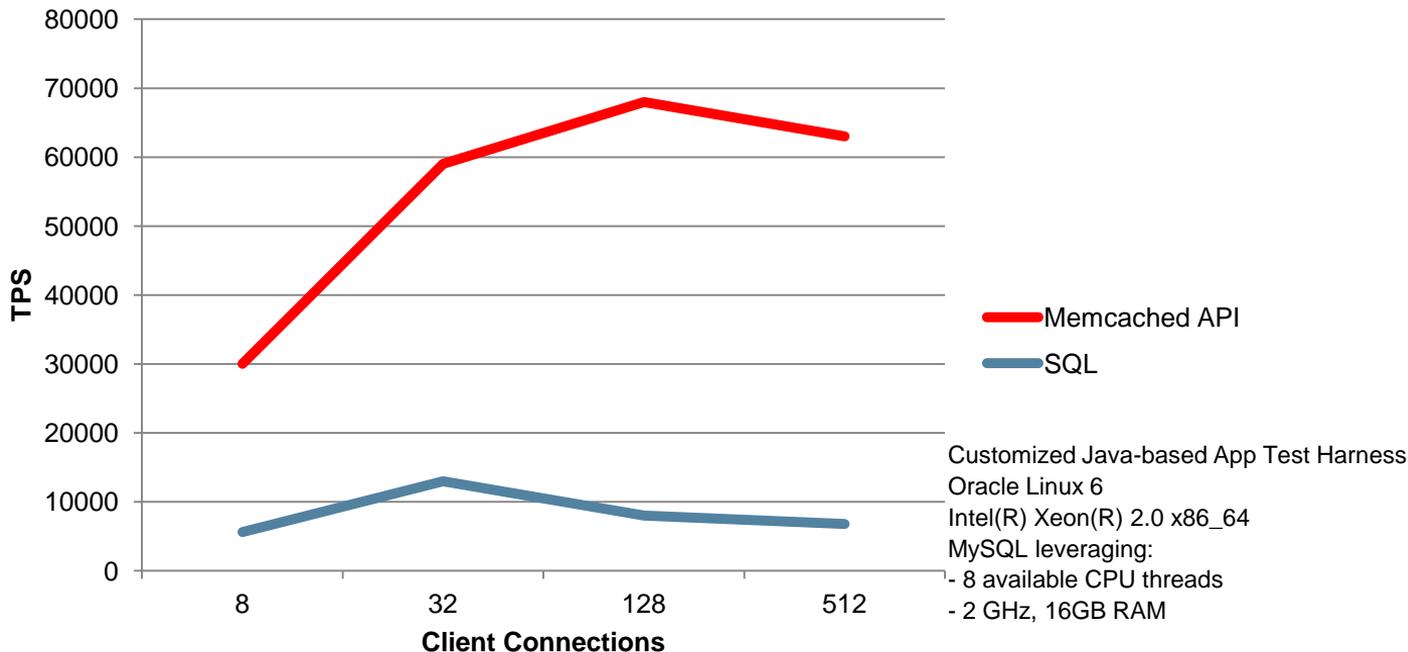
RDBMSとNoSQLの両立



- 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

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

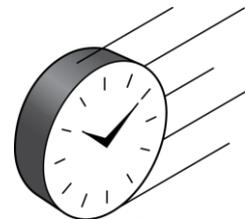
SQL実行性能の向上&処理時間の短縮

- 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: オプティマイザ



- サブクエリ的高速化
- LIMIT句で少数のレコードを取得する際のファイル・ソートを最適化
 - 4倍高速化 - 40秒から10秒に短縮
- インデックス条件のプッシュダウン
 - 160倍高速化 - 15秒から90ミリ秒に短縮
- FROMからのビュー／サブクエリの実データ取得を遅延
 - EXPLAINが240倍高速化 - 8分から2秒に短縮
- バッチ・キー・アクセスと複数範囲の読み取り
 - 280倍高速化 - 2800秒から10秒に短縮
- オプティマイザの統計情報の永続化

MySQL 5.6: オプティマイザ

サブクエリ的高速化

```
SELECT title FROM film WHERE film_id IN
  (SELECT film_id FROM film_actor
   GROUP BY film_id HAVING count(*) > 12);
```

- 開発者向けのメリット
 - 既存アプリケーションなどのサブクエリをJOINに書き換える必要なし
- パフォーマンス
 - DBT 3 のクエリ#13のベンチマーク
 - **数日**かかっていた処理が**数秒**で完了

MySQL 5.6: オプティマイザ

FROMからのビュー/サブクエリの実データ取得を遅延

```
EXPLAIN SELECT * FROM (SELECT * FROM a_big_table);  
SELECT * FROM t1 JOIN (SELECT t2.f1 FROM t2) AS derived_t2  
  ON t1.f2=derived_t2.f1  
  WHERE t1.f1 > 0;  
SELECT ... FROM derived_table AS dt  
          join table AS t WHERE dt.f1d = t.d1f
```

- マテリアライズを遅らせる
 - EXPLAINを高速化
 - 可能な場合には、マテリアライズを避ける
 - マテリアライズ化されたテーブルにインデックスを追加
- 240倍高速化 (8分から2秒に短縮)

MySQL 5.6: オプティマイザ

LIMIT句で少数のレコードを取得する際のファイル・ソートを最適化

```
CREATE TABLE products (  
  productid int auto_increment PRIMARY KEY,  
  productname varchar(200)  
);
```

Web use case – list top 100 products sorted by name

```
SELECT * FROM products ORDER BY productname LIMIT 100;
```

- `sort_buffer_size`内でソートできる場合、ソートの中間ファイル作成を回避する
- 1回のテーブルスキャンでソートされた結果セットを作成する
- 上記の例) 2000万行、デフォルトの`sort_buffer_size`
 - 4倍高速化 (40秒から10秒に短縮)

MySQL 5.6: オプティマイザ

Index Condition Pushdown (ICP)

```
CREATE TABLE person (  
  personid INTEGER PRIMARY KEY,  
  firstname CHAR(20),  
  lastname CHAR(20),  
  postalcode INTEGER,  
  age INTEGER,  
  address CHAR(50),  
  KEY k1 (postalcode, age)  
) ENGINE=InnoDB;
```

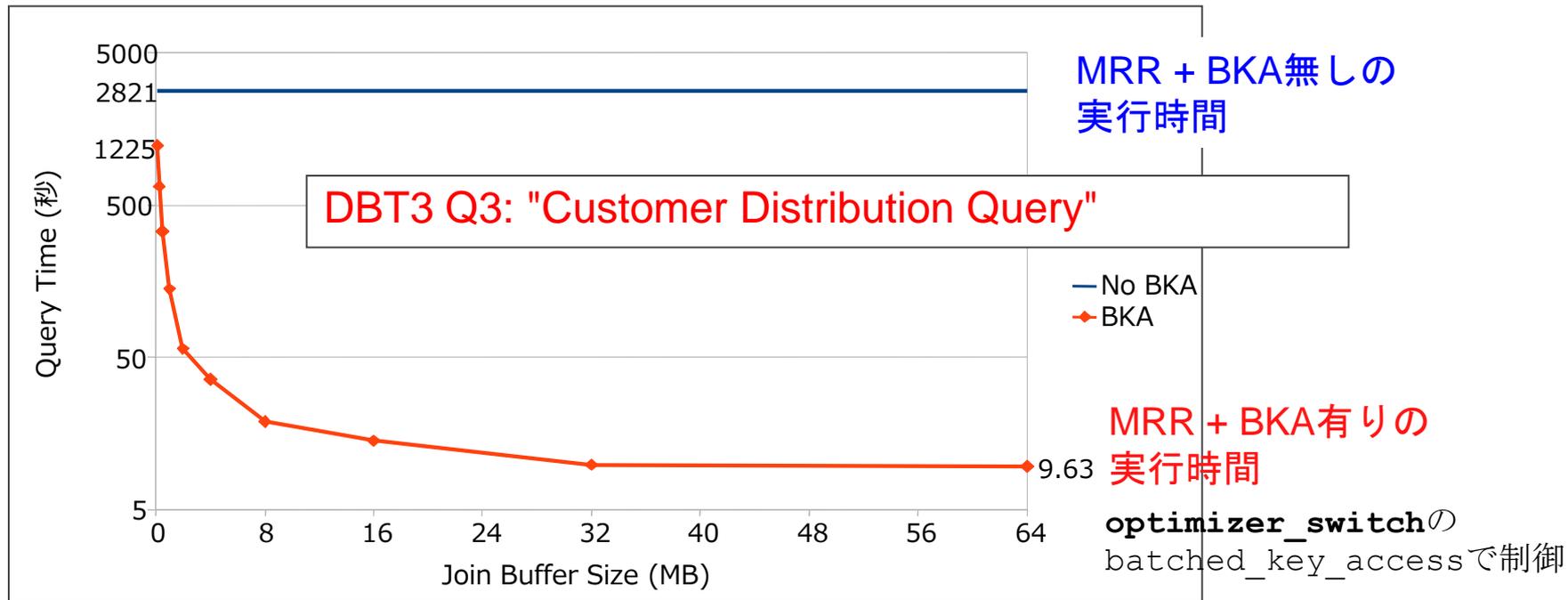
```
SELECT lastname, firstname FROM person  
WHERE postalcode BETWEEN 5000 AND 5500 AND age BETWEEN 21 AND 22;
```

- ICP 無効の場合
 - 15 s (buffer pool 128 Mb)
 - 1.4 s (buffer pool 1.5 Gb)
- ICP 有効の場合
 - 両方のケースで90 ms に短縮
- optimizer_switch の
index_condition_pushdownで制御

MySQL 5.6: オプティマイザ

バッチ・キー・アクセス (BKA) およびマルチ・レンジ・リード (MRR)

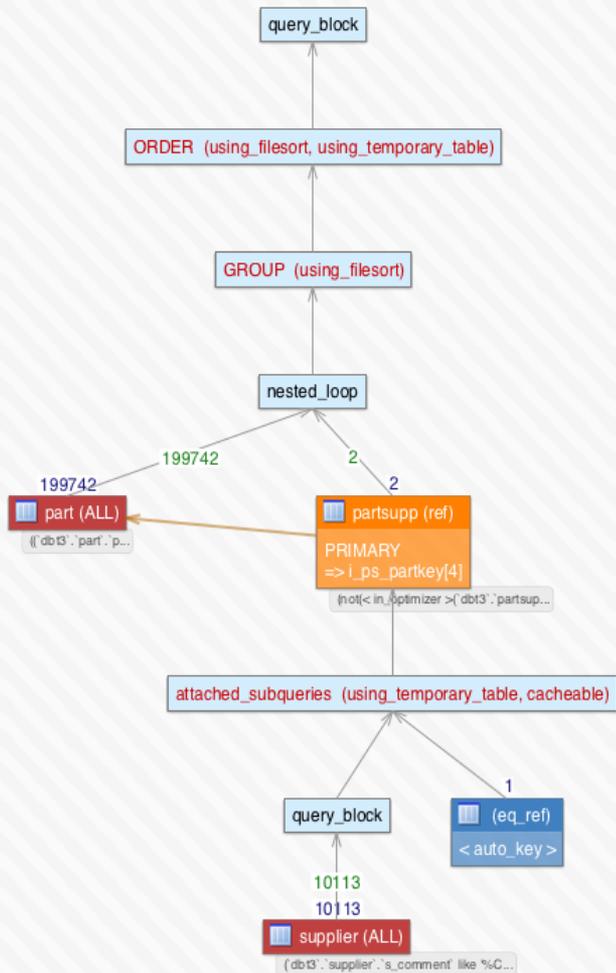
ディスクアクセスの多いJOINクエリのパフォーマンスを向上



MySQL 5.6: オプティマイザ

- EXPLAIN
 - JSONフォーマットでの出力
 - SELECT, INSERT, UPDATE, DELETE
- オプティマイザ・トレース

```
SET SESSION OPTIMIZER_TRACE='enabled=on';
SELECT (SELECT 1 FROM t6 WHERE d = c)
AS RESULT FROM t5;
SELECT * FROM information_schema.OPTIMIZER_TRACE;
```



MySQL 5.6: Replication改善点



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

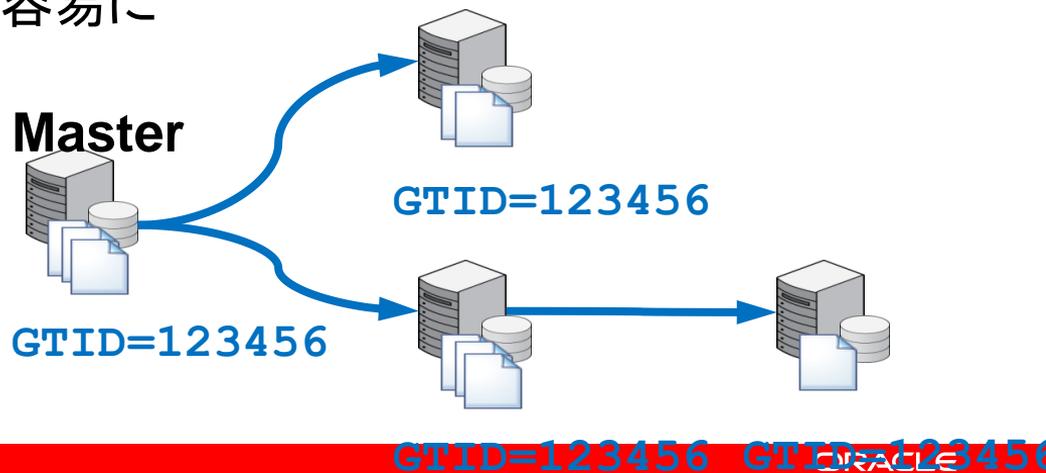
- レプリケーションチェックサム

開発 & 管理の簡素化

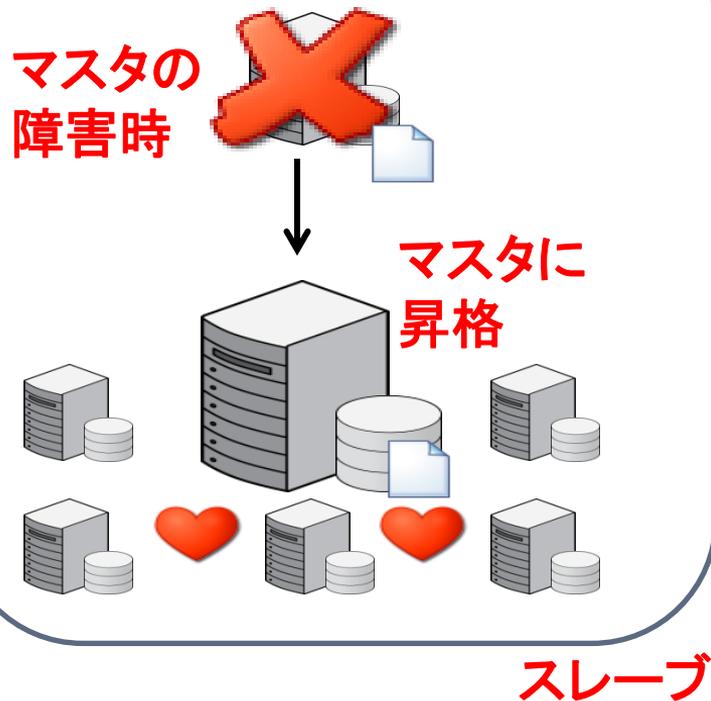
- 遅延レプリケーション
- リモートからのバイナリログのバックアップ
- ログへのメタデータの追加

MySQL 5.6: グローバルトランザクションID

- 複数台のレプリケーション環境でも容易にトランザクションの追跡/比較が可能
 - トランザクションを一意に識別できる識別子をバイナリログに記録
- フェイルオーバーのために、最も最新のスレーブを自動認識
- 多段構成のレプリケーションが容易に



フェールオーバー: レプリケーションの自動復旧

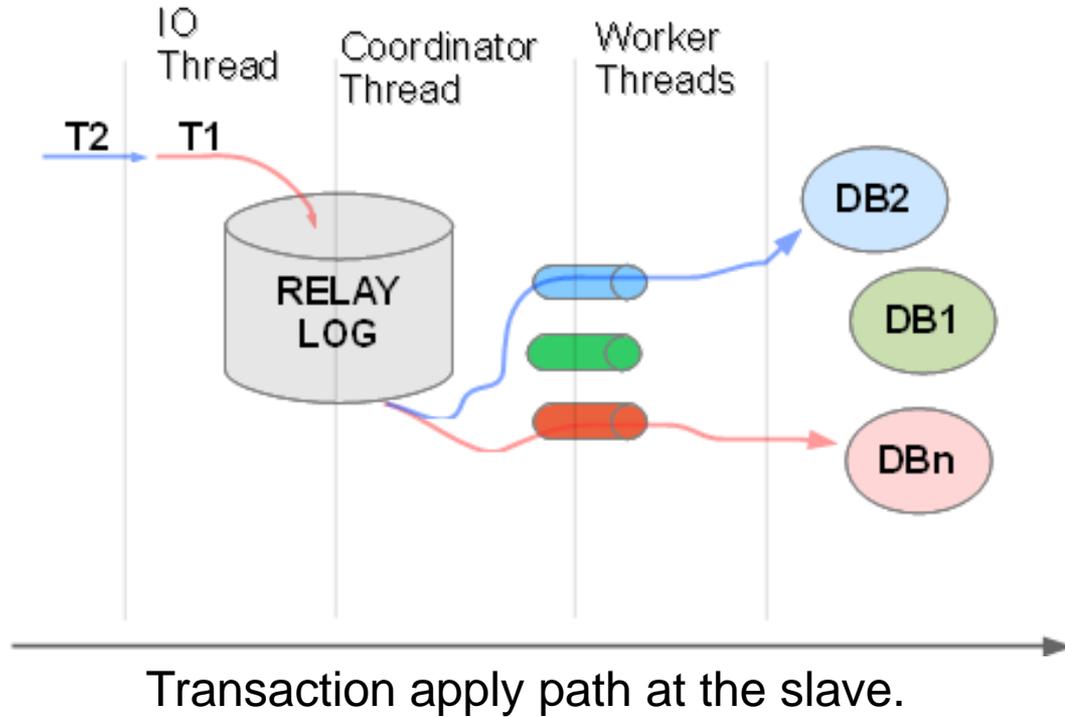


- Global Transaction Identifiers
 - レプリケーションの進捗を簡単に確認可能
- MySQL HA Utilities
 - 自動フェールオーバー、切り替え、リカバリ
 - スレーブの昇格ポリシーを設定可能
- スレーブ&バイナリログの耐障害性向上
 - ロールバックとレプリケーション再開処理の自動化

MySQL 5.6: マルチスレッドスレーブ

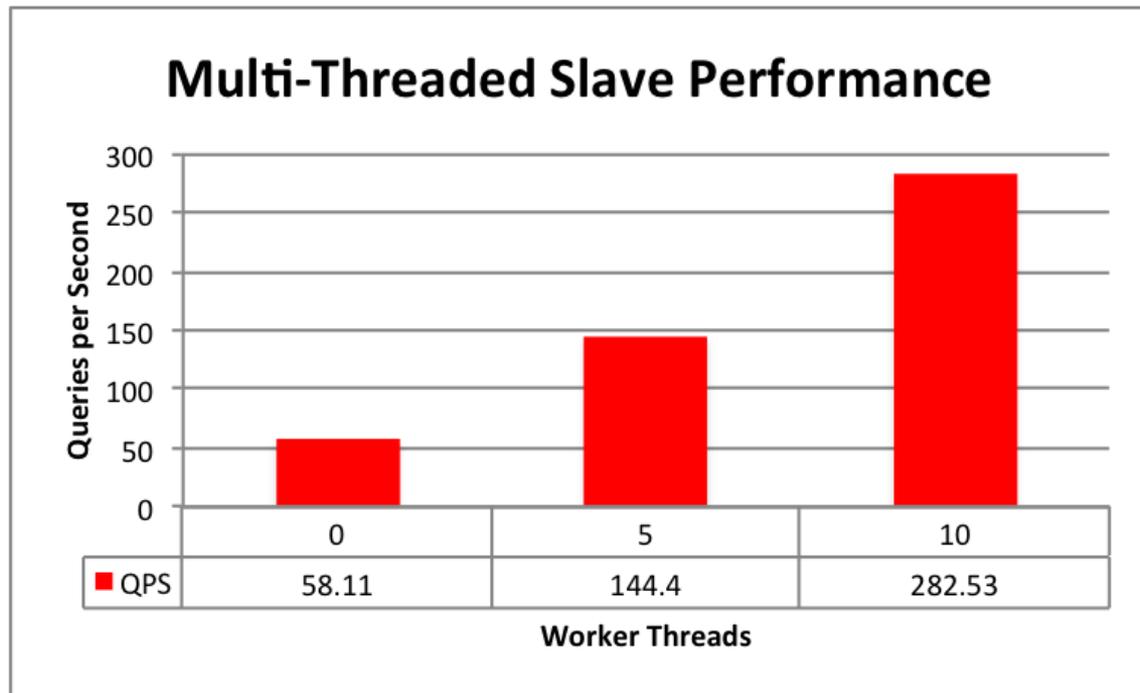
- スレーブのSQLスレッドを複数起動可能
 - slave_parallel_workers オプションで設定
 - 複数のworker thread が起動し、SQLスレッドはコーディネーターとして動作
- 並列化の単位はスキーマ(データベース)
 - マルチテナントシステムと相性がいい
- マスターとスレーブでは、トランザクションの実行順序が異なる可能性がある
 - データベース内での整合性は保証されている
 - 複数データベース間での整合性は、結果整合性(eventual consistency)

MySQL 5.6: マルチスレッドスレーブ



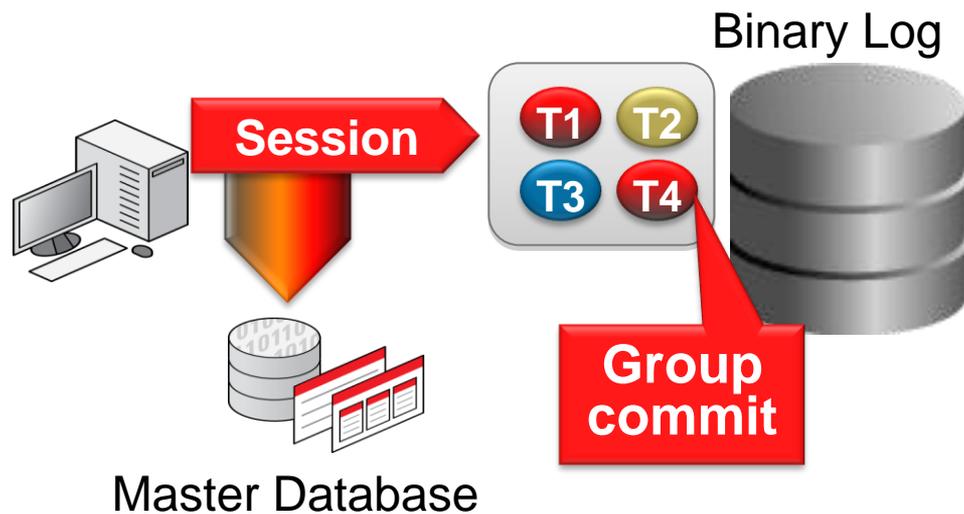
MySQL 5.6: マルチスレッドスレーブ

5倍の性能向上



- Sun Fire X4170 M2 Server
- 64bit OEL 6.1
- 48GB RAM
- 2 sockets, 6 cores with hyper-threading, 2930MHz
- Workload: Sysbench (10 schemas, 1 table/schema)

MySQL 5.6: バイナリログのグループコミット

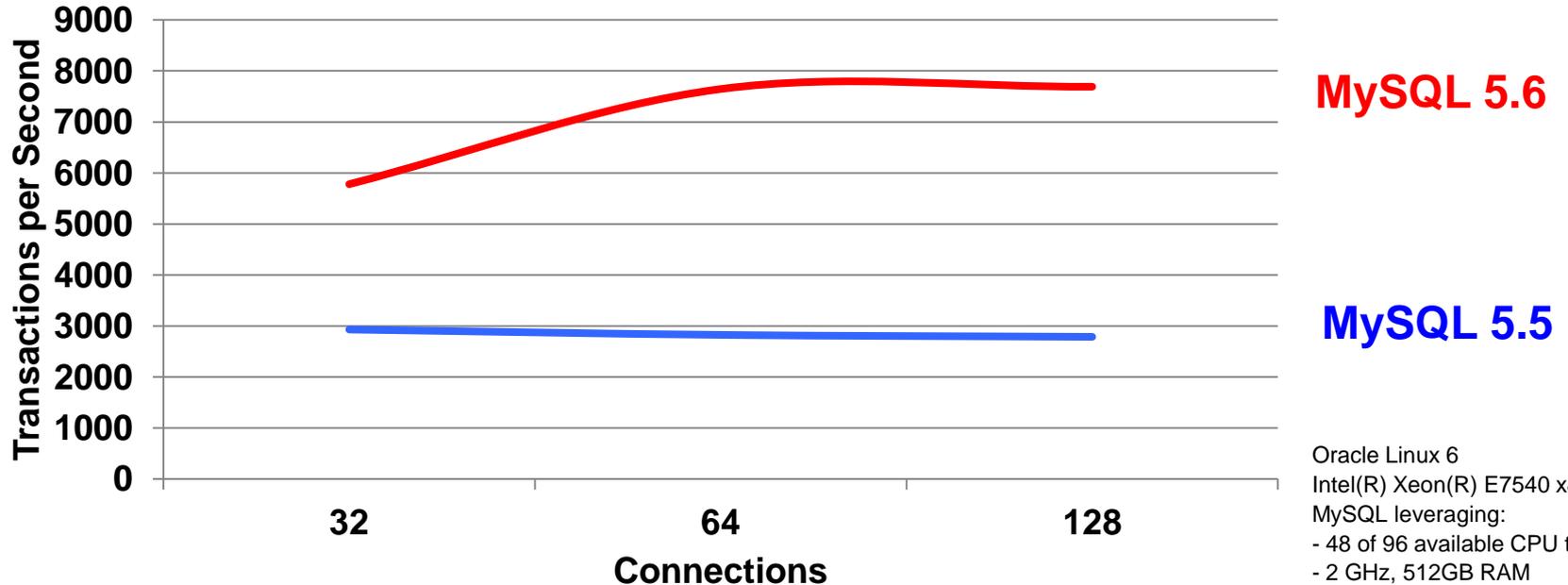


- 複数トランザクションの情報をまとめてバイナリログに記録
- Sync_binlog=1の場合のオーバーヘッドを削減
- バイナリログコミット部分のソースをリファクタし、よりメンテナンスしやすいコードに改善

Binary Log Group Commit Performance

Binlog=1

MySQL 5.6 vs. 5.5 - Read Write (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

180% Performance Gain

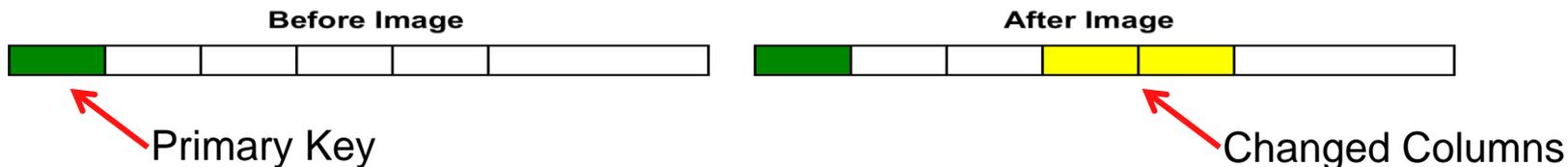
ORACLE

MySQL 5.6: 行ベースレプリケーションの最適化

Prior to MySQL 5.6 or `binlog-row-image = full`

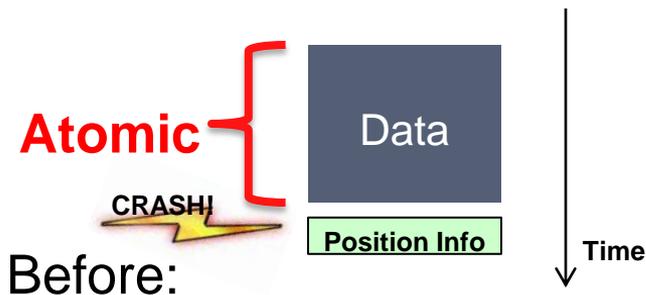


With MySQL 5.6, `binlog-row-image = minimal`

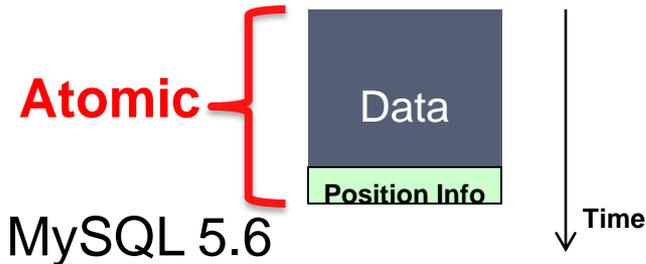


- 新しいオプション: `binlog-row-image=minimal`
- マスター/スレーブのスループットを向上
 - バイナリログのサイズ削減、メモリ使用量削減、ネットワーク転送量削減
- 変更された行イメージだけを保持

クラッシュセーフなスレーブ



- Transaction Data: **in tables**
- Replication Info: **in files**



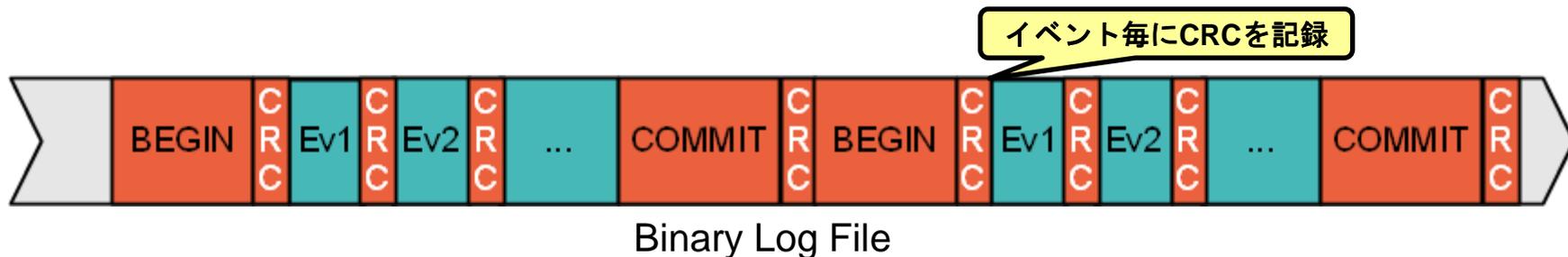
- Transaction Data: **in tables**
- Replication Info: **in tables**

- スレーブクラッシュ時の自動的なリカバリ
 - ポジション情報とデータは一貫性がある
- 管理者の介入なしにレプリケーションを再開可能
 - 最後にコミットしたイベントに自動的にロールバック

データの損失や破損のリスクを排除

チェックサムの追加

- バイナリログにチェックサムを記録可能(CRC32)
 - binlog-checksum オプションで有効/無効を制御
- マスターは、バイナリログにイベントを記録する時に、CRC32を生成
- 複数個所で誤り検出可能
 - マスターとスレーブでの誤り検出は、個別に有効/無効の設定が可能 (master-verify-checksum option、slave-sql-verify-checksum)



Binary Log File

チェックサムの追加

- 誤りが早期に検出されることにより、データの破損を防ぐ
- 誤り発生個所をより正確に特定可能(disk、network、memory)
 - サポート時間の削減
- 内部テストでは、チェックサム使用時に大きなオーバーヘッドは検出されず(STATEMENT、ROW、MIXED の全てにおいて)

MySQL 5.6: パフォーマンススキーマ

新たな統計情報

- SQL文/実行ステージ
- テーブルとインデックスのI/O
- テーブルロック
- ユーザ/ホスト/アカウント
- ネットワークI/O

新機能

- ホストキャッシュの表示
- サマリーテーブルの追加
- 設定の簡素化
 - my.cnfに設定を追加して起動
 - 自動的にチューニング
- オーバーヘッドの削減

MySQL 5.6 – パフォーマンス・スキーマの計測

データベース・プロファイリング／アプリケーション・チューニングの向上

- ステートメント／ステージ
 - 最もリソース消費の多いクエリは？ 時間がかかっているのはどこか？
- テーブル／インデックスI/O、テーブル・ロック
 - 最も負荷や競合が多いアプリケーション・テーブル／インデックスは？
- ユーザー／ホスト／アカウント
 - リソースをもっとも消費しているアプリケーション・ユーザ／ホスト／アプリケーションは？
- ネットワークI/O
 - ネットワークの負荷は？ セッションのアイドル・タイムは？
- 集計値
 - スレッド、ユーザー、ホスト、アカウント、オブジェクトごとに統計情報を集計

MySQL 5.6: セキュリティ

- パスワード管理を一新、暗号化を改良
 - パスワードの平文での表示を抑制
 - パスワード強度の検証およびポリシーの強制
 - 次回ログイン時にパスワードの変更を強制
 - より安全なSHA256ハッシュによるパスワード暗号化

MySQL 5.6: データ型

- TIME/TIMESTAMP/DATETIME - 秒以下の単位を格納

```
CREATE TABLE t1 (t TIME(3), dt DATETIME(6));
```

- TIMESTAMP/DATETIME – デフォルト/自動更新 される値に
CURRENT_TIMESTAMPを設定可能

```
CREATE TABLE t1 (  
ts TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,  
dt DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP );
```

- TIMESTAMP – NULLを格納可能に

開発者の使い勝手を向上

MySQL 5.6: その他の改良点

- デフォルト設定の最適化
- パーティショニングの改良
 - パーティション数が多い場合の性能改善
 - パーティションごとのインポート/エクスポート
 - パーティションの明示的な選択
- GIS: より正確な図形処理



Get it now!

dev.mysql.com/downloads/mysql/

ORACLE

MySQLサーバ開発の優先課題

- Web、クラウドインフラ、組み込みでの利用への最適化
- プラグインによるアーキテクチャの簡素化
 - メンテナンス効率および拡張性の向上
 - さらなるNoSQLオプションの追加 (HTTP, JSON, JavaScript など)
- 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/パーサー/プロトコル
- InnoDB
 - 主要な使い方の改良(参照専用、高速なりカバリ)、GIS
 - 全文検索の日本語対応 (中国語、韓国語も)
- 高可用性構成の簡素化、レプリケーション強化、シャーディング

5.6

MySQL Server – GA

InnoDBの改良やオプティマイザの刷新による性能&拡張性向上
レプリケーションの可用性向上 & NoSQLインタフェース追加

7.2

MySQL Cluster - GA

JOIN性能の大幅な向上
ディザスタリカバリの信頼性の向上

7.3

MySQL Cluster - DMR

外部キーサポート
Node.js APIの追加、運用性能向上

Hardware and Software

ORACLE®

Engineered to Work Together

ORACLE®