



OpenStackにおけるMySQLの活用

OpenStackリポジトリ、DBサービス基盤としてのMySQL

2015/03/26

Shinya Sugiyama / 杉山真也

MySQL Principal Sales Consult, MySQL Global Business Unit

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。

Agenda

1 MySQL概要

- Community Edition (GPL) / Commercial Edition (Commercial)

2 OpenStackのリポジトリとしてのMySQL

- OpenStackリポジトリとしてのMySQL
- MySQL可用性とスケール(DRBD, MySQL Cluster)

3 OpenStackにおけるDatabaseのAuto Scale

- OpenStackにおけるDBaaS (Trove)概要
- MySQL Fabric基本機能ご紹介
- MySQL FabricによるDatabaseの自動プロビジョニング

参考情報 : MySQL Commercial版概要



The world's most popular open source database

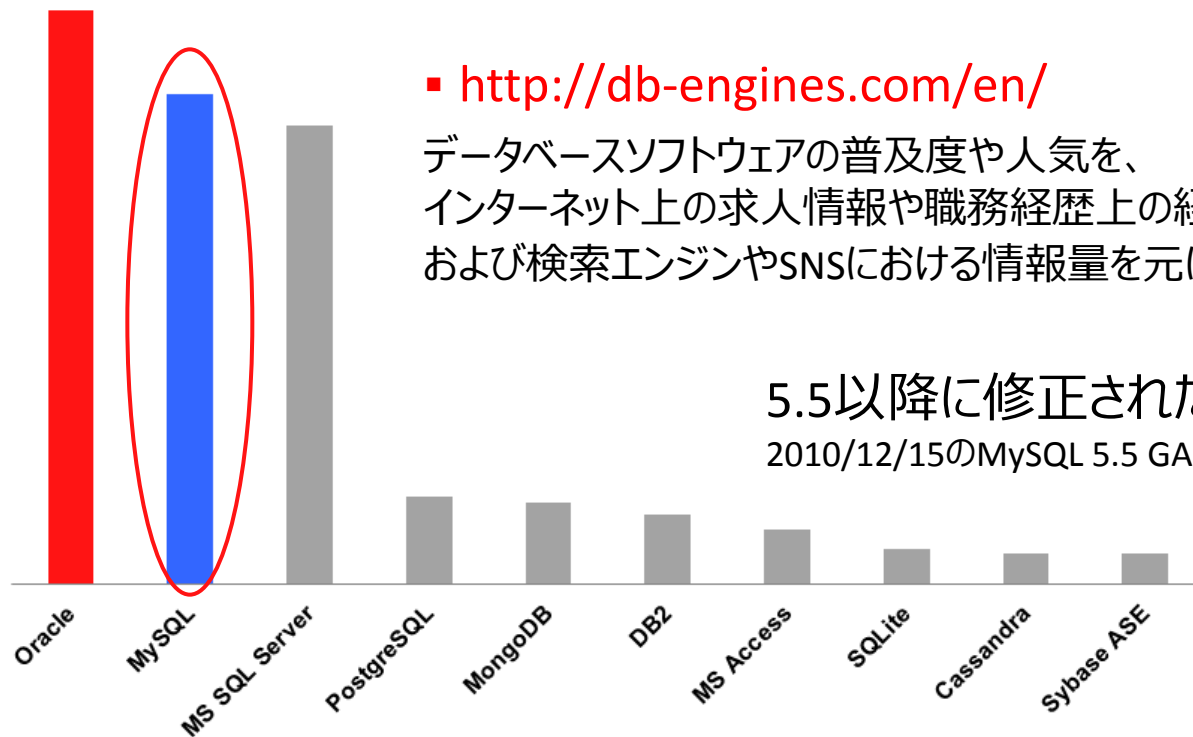
世界で最も普及しているオープンソースデータベース

<https://github.com/mysql>

<https://registry.hub.docker.com/u/mysql/mysql-server/>

The world's most popular open source database

参照1) DB-Enginesによる調査結果



▪ <http://db-engines.com/en/>

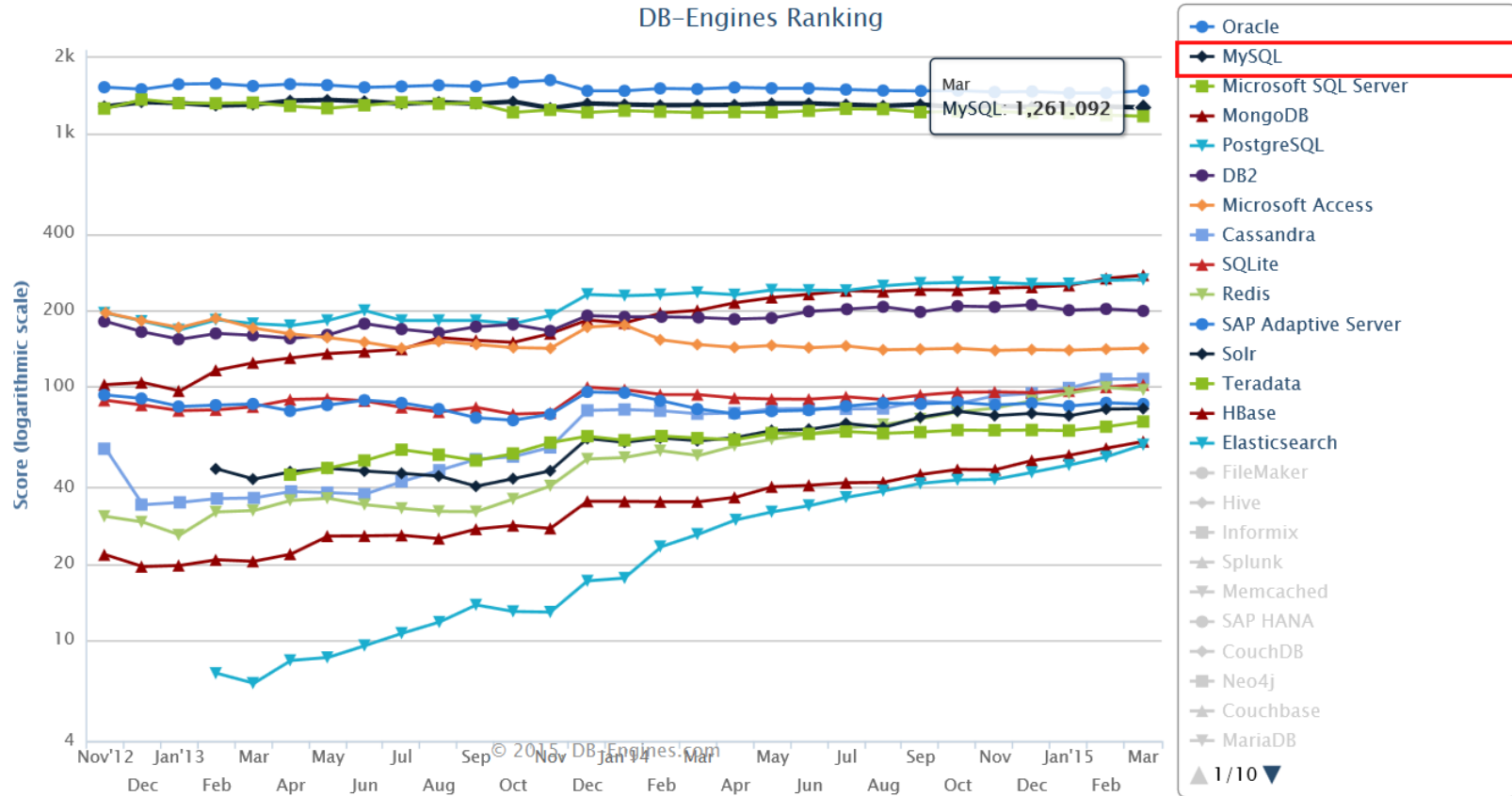
データベースソフトウェアの普及度や人気を、インターネット上の求人情報や職務経歴上の経験、および検索エンジンやSNSにおける情報量を元に毎月作成し公開。

5.5以降に修正されたバグの合計: **8259**

2010/12/15のMySQL 5.5 GA以降~ (2014/10)

参照2) DB-Enginesによる調査結果 期間：11/2012～03/2015

as of 2015/03/24



MySQL: From the Web to the Cloud

Web			 WIKIPEDIA		
Enterprise			 <i>Taking Care of Business</i>		
Embedded					
Cloud		 smart computing			

[WebScaleSQL](#)(MySQL branch) - Alibaba, Facebook, Google, LinkedIn, and Twitter

MySQLのコミュニティ版と商用版

OpenStack環境においては、Community Editionが多く使われていますが、Commercial Editionをご利用頂く事でツールやサポートも利用可能です。

Community Edition (GPL)

- MySQL Community Server
- MySQL Cluster
- MySQL GUI管理ツール
- MySQLコネクタ (JDBC, ODBC, etc.)
- ドキュメント
- フォーラム

Commercial Edition

- Standard Edition
- **Enterprise Edition**
- MySQL Cluster Carrier Grade Edition
- 商用ライセンス (組み込み用)
- プロフェッショナルサービス
 - トレーニング、コンサルティング、サポート

- コミュニティ版ソフトウェアはGPLでソースコードも公開し提供
- 商用版は、付加価値として**技術サポートや管理機能、拡張機能を有償で提供**

[参照: MySQL Downloads](#)

ORACLE

MySQL Enterprise Edition サービスカテゴリー



管理ツール

- 監視/モニタリング
- **バックアップ**
- 開発サポート
- 管理全般
- マイグレーション



拡張機能

- 拡張性
- 高可用性
- **セキュリティ**
- 監査
- 暗号化



サポート

- **技術サポート**
- コンサルティングサポート
- オラクル製品との動作保証



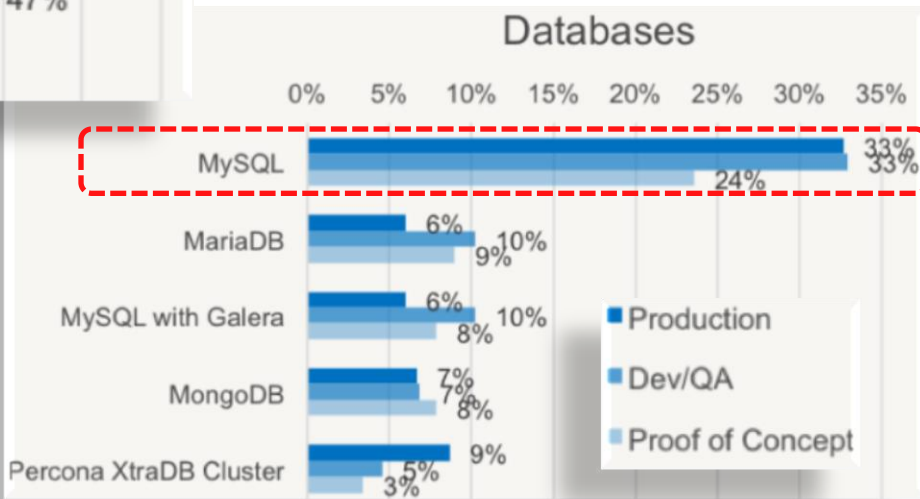
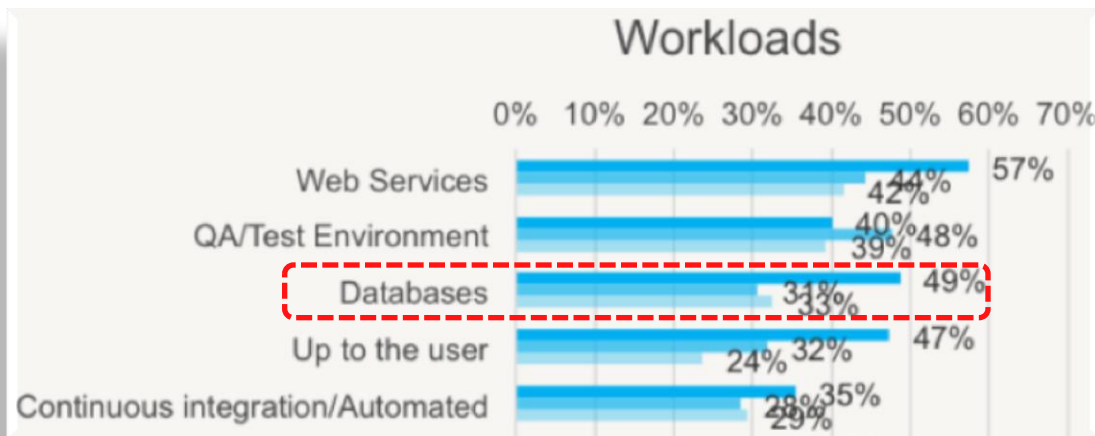
詳細に関しては, <http://www-jp.mysql.com/products/>を参照下さい。

ORACLE



OpenStackのリポジトリとしてのMySQL

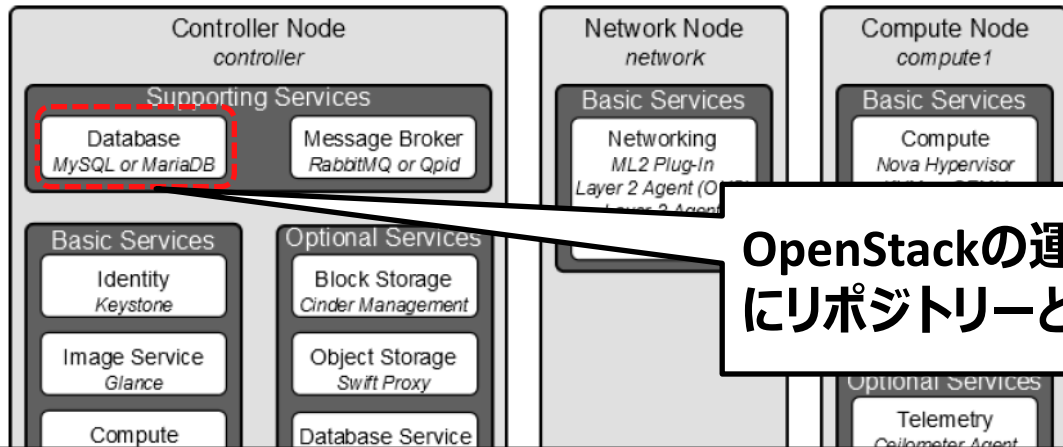
OpenStack Survey Nov/2014



MySQLはOpenStackにおいても、
多くのサービスで利用頂いております

参照: [OpenStack User Survey Insights: November 2014](#)

MySQL in OpenStack



OpenStackの運用上、重要なデータが永続的にリポジトリとしてのMySQLに保管されます。

MySQL database

[Controller setup](#)
[Node setup](#)

Most OpenStack services require a database to store information. These examples use a MySQL database that runs on the controller node. **You must install the MySQL database on the controller node.** You must install MySQL client software on any additional nodes that access MySQL.

抜粋: [OpenStack.org](https://openstack.org)

ORACLE

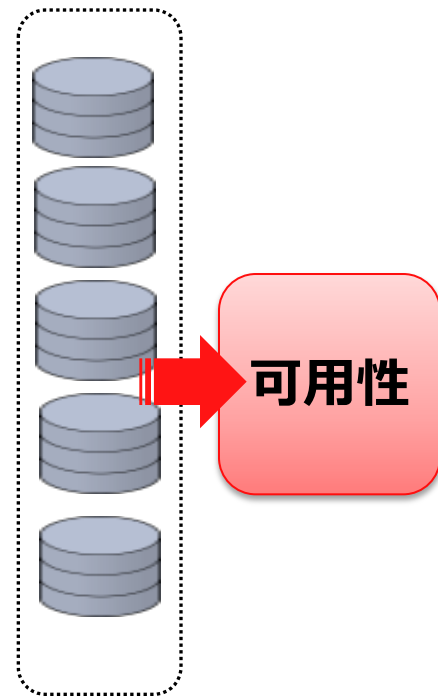
管理情報リポジトリDB(可用性)

OpenStackは可用性、性能、拡張性、管理性、コスト最適化を必要とします。
MySQLは、サービスを支える為の管理情報リポジトリDBとして広く利用されています。

Keystone	認証
Nova	IaaS (例 : EC2)
Cinder	ブロックストレージ
Glance	VMイメージ管理
Neutron	仮想ネットワーク管理

```
+-----+-----+
| TABLE_SCHEMA | Number of tables |
+-----+-----+
| cinder         |          19      |
| glance        |           8      |
| keystone      |          16      |
| nova          |         108     |
| ovs_neutron   |          29      |
+-----+-----+
5 rows in set (0.01 sec)
```

これらのデータはOpenStackの管理に必要なメタデータの為、
サービスをコントロールする上で、非常に重要なデータと
なっています。また、安定したサービスを継続的に提供する為に、
リポジトリの可用性構成も非常に重要な構成要素の一つです。



管理情報リポジトリDB (拡張性)

基本的に、構成管理用のリポジトリとしての役割がメインなので、大規模なシステムを管理しない限りは、それ程大きなデータベース負荷は発生しない事が想定されます。しかし、[keystone](#)のように、データベースに継続的にアクセストークンを蓄積し続けるケースもあるので、データ量が多くなっても、処理が重たくならないようにするか、定期的に削除するなどの運用を検討する必要があります。

また、[Ceilometer](#)等を利用して課金の為などに、メータリング情報を収集する場合は、規模によってはインサートや参照に性能課題が発生する可能性があります。

※Table Partitioningも選択肢として検討

```
mysql> desc token;
```

Field	Type	Null	Key	Default	Extra
id	varchar(64)	NO	PRI	NULL	
expires	datetime	YES	MUL	NULL	
extra	text	YES		NULL	
valid	tinyint(1)	NO		NULL	
trust_id	varchar(64)	YES		NULL	
user_id	varchar(64)	YES		NULL	

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

```
mysql> select id,expires from token limit 0,10;
```

id	expires
80c91125db88167f997c066eb8cc42e8	2014-11-27 01:26:07
030039767e1322c2044cf7a027c09e17	2014-11-27 01:26:07
05d5171229881d674b470dc2f384eeec	2014-11-27 01:26:07
068869123f444ce2d0b80367443fe4f9	2014-11-27 01:26:07
07cfa8a6398417c731fa66e8df49ce81	2014-11-27 01:26:07
084b056e3cf9c00b90cac959506ab311	2014-11-27 01:26:07
084be2411e04da737c0ef917c477678a	2014-11-27 01:26:07
0898f22665f48639a9456c61dd7547e2	2014-11-27 01:26:07
0a470b6703c8742750fec2720aad9375	2014-11-27 01:26:07
0bb52abeb199e440ccd9d46c69a94b6b	2014-11-27 01:26:07

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

```
mysql> |
```



MySQLリポジトリの可用性と拡張性

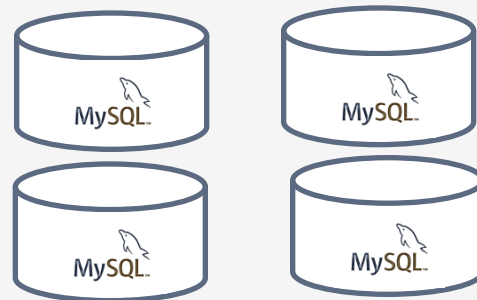
```
$ nova list
```

ID	Name	Status	Task State	PowerState	Networks
05682b91-81a1-464c-8f40-8	demo-ins1	ACTIVE	-	Running	demo-net=192.168.1.3

DRBD



MySQL Cluster



MySQL Cloud Controller: High Availability/Scalability



Clustering Mode	Auto-Failover	Failover Time	Scale-out
Active/Passive	Yes	Secs+	No



OpenStack HAMニュアルに
詳細が記載されています

- ☰ Preface
- ☰ 1. Introduction to OpenStack High Availability
- ☰ HA using active/passive
 - ☰ 2. The Pacemaker cluster stack
 - ☰ 3. Cloud controller cluster stack
 - ☑ Highly available MySQL
 - ☑ Highly available RabbitMQ
 - ☰ 4. API node cluster stack
 - ☰ 5. Network controller cluster stack
- ☰ HA using active/active
- ☰ A. Community support

Highly available MySQL

[Configure DRBD](#)
[Creating a file system](#)
[Prepare MySQL for Pacemaker high availability](#)
[Add MySQL resources to Pacemaker](#)
[Configure OpenStack services for highly available MySQL](#)

MySQL is the default database server used by many OpenStack services. Making the MySQL service highly available involves

- Configure a DRBD device for use by MySQL,
- Configure MySQL to use a data directory residing on that DRBD device,
- Select and assign a virtual IP address (VIP) that can freely float between cluster nodes,
- Configure MySQL to listen on that IP address,
- Manage all resources, including the MySQL daemon itself, with the Pacemaker cluster manager.



Note

[MySQL/Galera](#) is an alternative method of configuring MySQL for high availability. It is likely to become the preferred method of achieving MySQL high availability once it has sufficiently matured. At the time of writing, however, the Pacemaker/DRBD based approach remains the recommended one for OpenStack environments.

参照 : Highly available MySQL <http://docs.openstack.org/high-availability-guide/content/s-mysql.html>

Configuring MySQL for HA on OpenStack

Using DRBD, Pacemaker, Corosync

- MySQLを使用するようにDRBDを設定
- DRBDのデータディレクトリを使用するようにMySQLを構成
- Clusterノード間で共通のVIPを利用するように設定
- MySQLはVIPでListenするように設定
- OpenStack関連サービスはMySQLに設定されたVIPに接続

```
sql_connection = mysql://glancedbadmin:<password>@192.168.42.101/glance
```

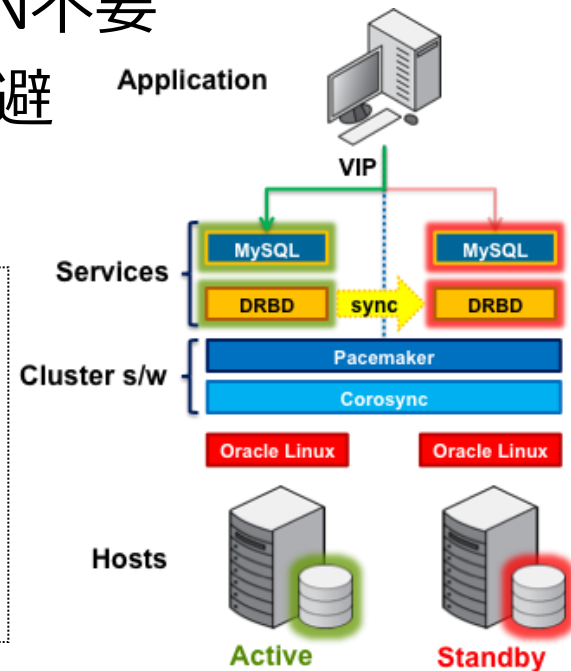
- ※ システム障害発生時は、VIPに紐づいたノード間でフェイルオーバー
- ※ 拡張は、基本的にサーバーのスケールアップで対応

High Availability with DRBD

- 分散ストレージを利用するため、共有ディスクやSAN不要
- 同期レプリケーションによってデータを失うリスクを回避
- オープンソースで実績の多いソリューション

Oracle Linux + DRBD Stack

- Oracle認定構成：フルスタックサポート
- Oracle Linux Unbreakable Enterprise Kernel R2にDRBDが統合
- Oracle Linux 6.2以上で使用可能
- クラスタリングとフェイルオーバーにて、PacemakerとCorosyncを使用
- 追加機能として、[MySQL DTrace probes](#)を利用可能



DRBD - Configuration and Deployment Guide

検索

5.4 Tracing mysqld Using Dtrace

検索

MySQL Cloud Controller: High Availability/Scalability



Clustering Mode	Auto-Failover	Failover Time	Scale-out
Multi-Master	Yes	< 1 Sec	Yes

MySQL Clusterとは？

- MySQLとは開発ツリーの異なる別製品
- 共有ディスクを使わずに、アクティブ-アクティブのクラスタ構成が組める
- 元々はSQLを使わないデータベース(NOSQL)だったが、MySQLと統合されSQLも追加で使えるようになった(NoSQLとSQLの両方が使えるデータベース)
- 各テーブルのストレージエンジンを選択する事が出来る(InnoDB or NDB)
- SQL, NoSQL共にACID処理可能なインメモリーデータベース

適したシステム

- 高可用性が求められるシステム
- 同時多発的に大量の短いトランザクションが発生するシステム
- 読み込み処理だけでなく、書き込み処理に対しても拡張性が求められるシステム



参照 : [MySQL Cluster Evaluation Guide](#)

MySQL Clusterの特徴

参照更新性能の
高い拡張性

- 自動シャーディング、マルチマスタ
- ACIDトランザクション、OLTPとリアルタイム分析

99.999% の可用性

- シェアードキャッシング、単一障害点無し
- 自動復旧、オンラインメンテナンス

リアルタイム

- インメモリ処理に最適化 + ディスク併用可能
- 低レイテンシ

SQL + NoSQL

- キー・バリュ型 + 複雑なリレーショナルな処理
- SQL+Memcached+JavaScript+Java+HTTP/REST&C++

低コスト

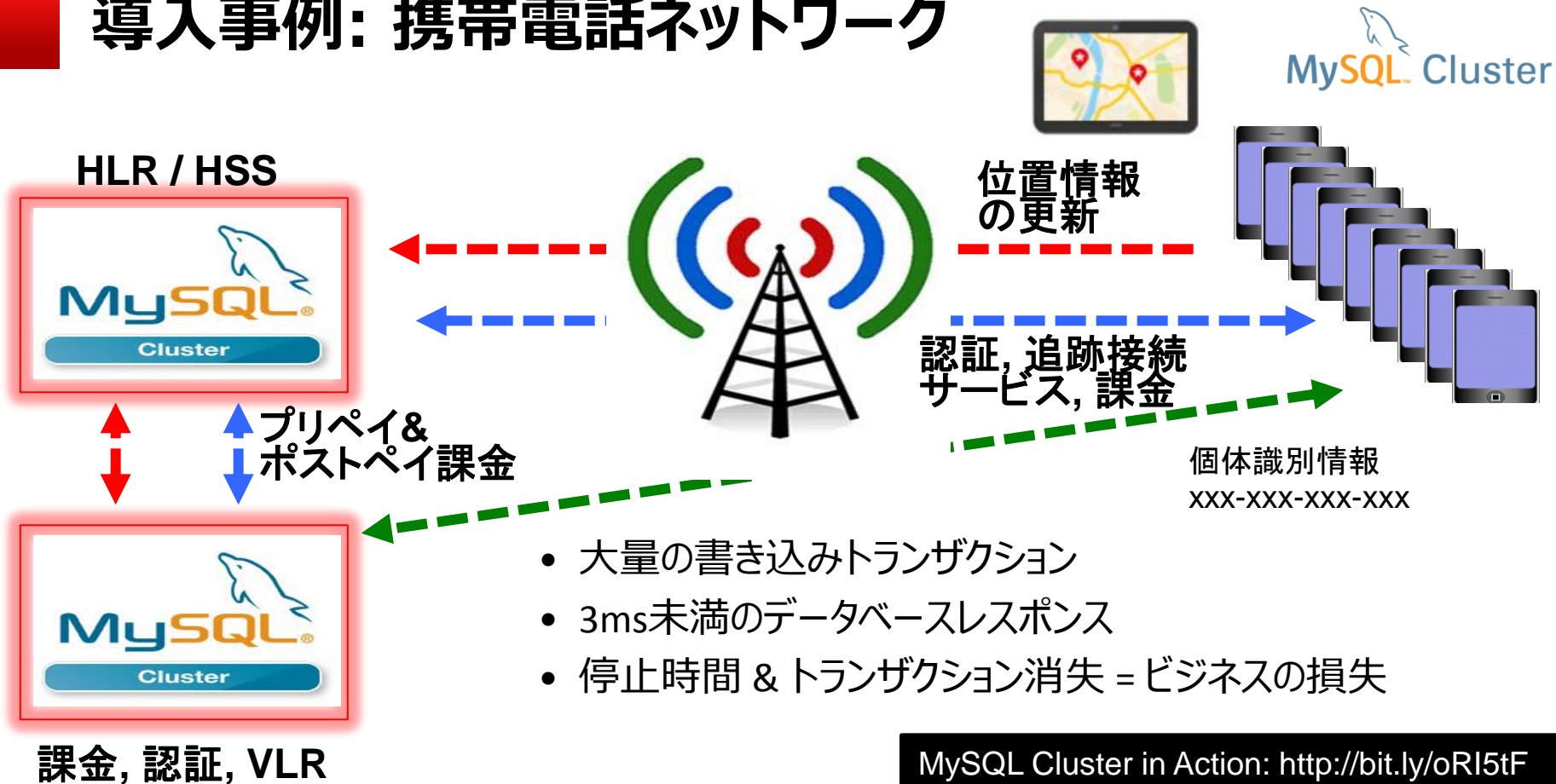
- オープンソース + 商用版運用支援ツール
- 特殊なハードウェア不要、管理監視ツール群、サポート

Who's Using MySQL Cluster?



ORACLE

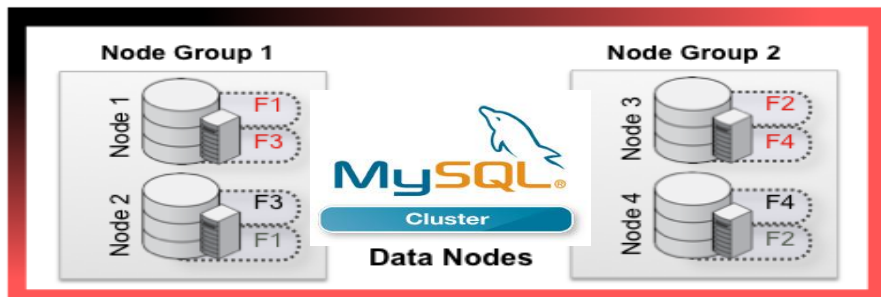
導入事例：携帯電話ネットワーク



導入事例：航空機管制システム



- 米国海軍航空母艦
- 包括的航空機運用管制システム
 - メンテナンス記録
 - 燃料搭載量管理
 - 気象状況
 - 飛行甲板管理
- システム要件
 - 単一障害点無し
 - 完全な冗長性
 - 小さなフットプリント & 過酷な利用環境での利用
- 4台のMySQL Clusterノード LinuxおよびWindows



MySQL User Conference Session: <http://bit.ly/ogeid3>

導入事例: PayPal®



PAYPAL OVERVIEW

- Processed \$145bn in transactions (CY2012)
- 前年比成長率22%、1億23百万アカウント、190マーケット

課題/機会

- グローバル単位での不正検知システム構築
- ユーザセッション、トランザクション追跡のリアルタイム処理

DATABASEに求められた要件

- 100TB のデータ & 1億ユーザに対応可能
- ACID 準拠のトランザクション
- 書込まれたデータを1秒以内に全世界のどこからでも確認可能
- ユーザのトランザクション履歴をリアルタイムで分析可能
- クラウド環境における、高い性能拡張性と99.999%の高可用性

<http://www.mysql.com/customers/view/?id=1223>

CUSTOMER PERSPECTIVE

"Technologies such as MySQL Cluster enables users to get the best of both world's...the agility of NoSQL systems with the trust, maturity and reliability of the SQL model "

Daniel Austin, Chief Architect, PayPal

課題解決策

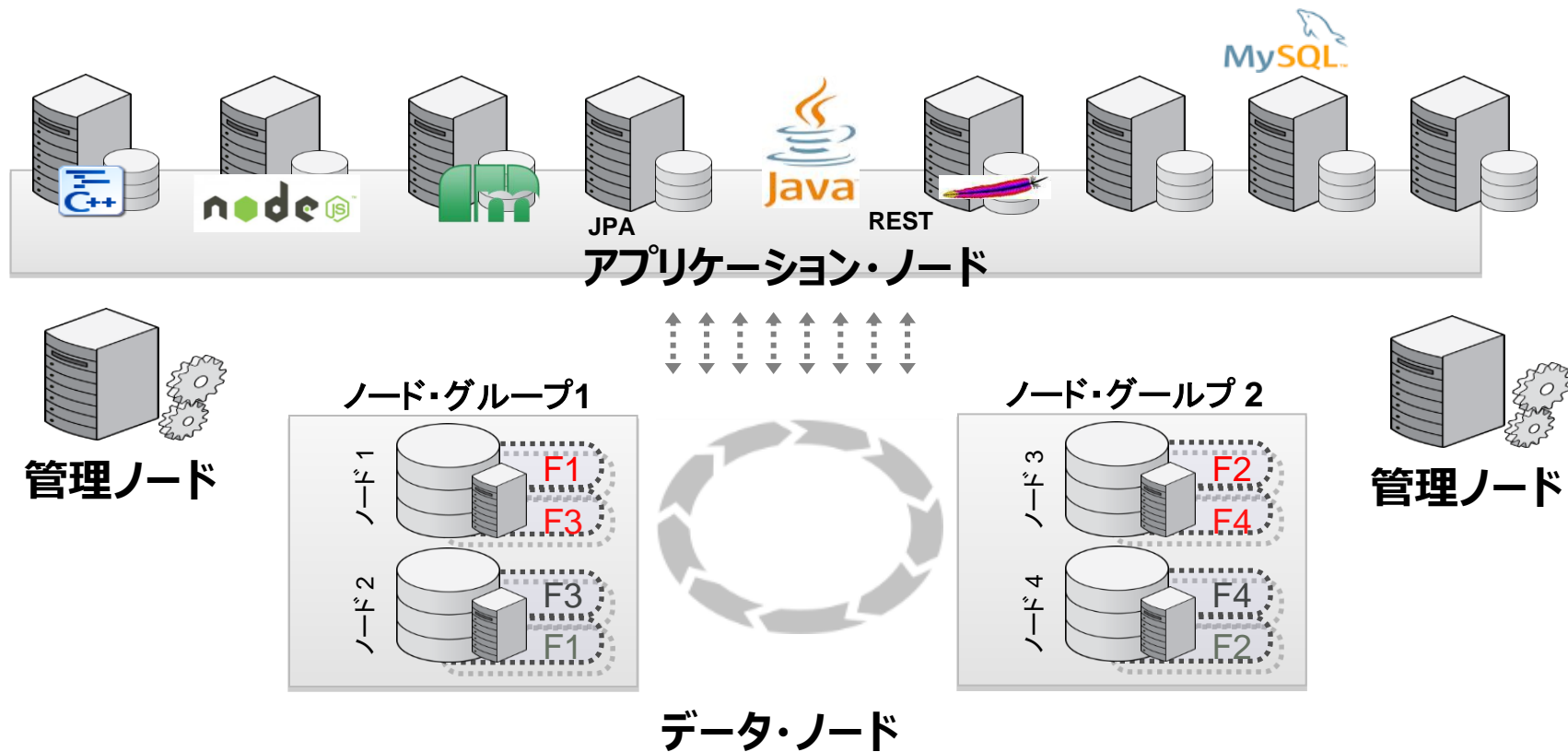
- MySQL Cluster 7.2 with Geo-Replication
- AWS

結果

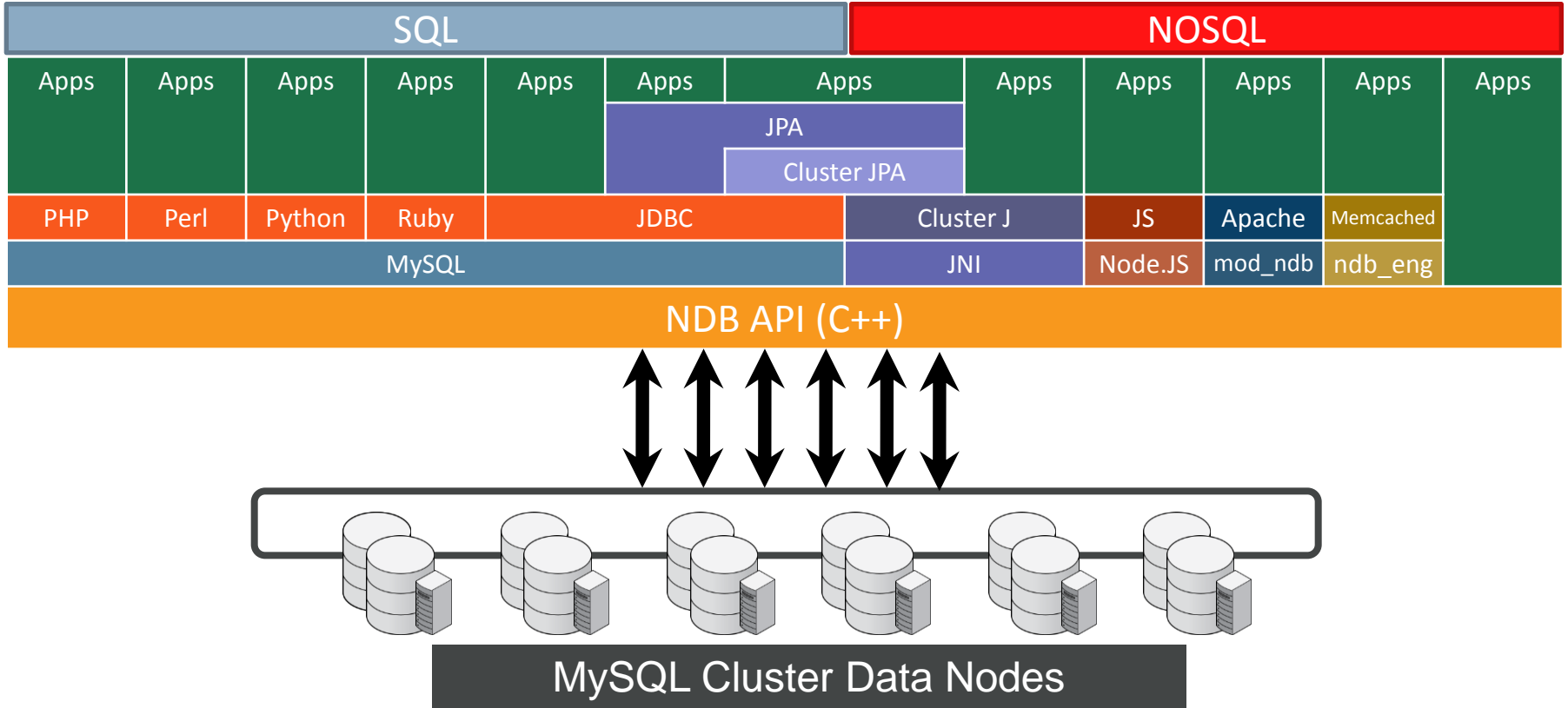
- 設計目標より3倍速いパフォーマンス
- 40TB のデータをCluster間で同期
- 5つのAWS リージョンにMySQL Clusterを構築
- Self-healing



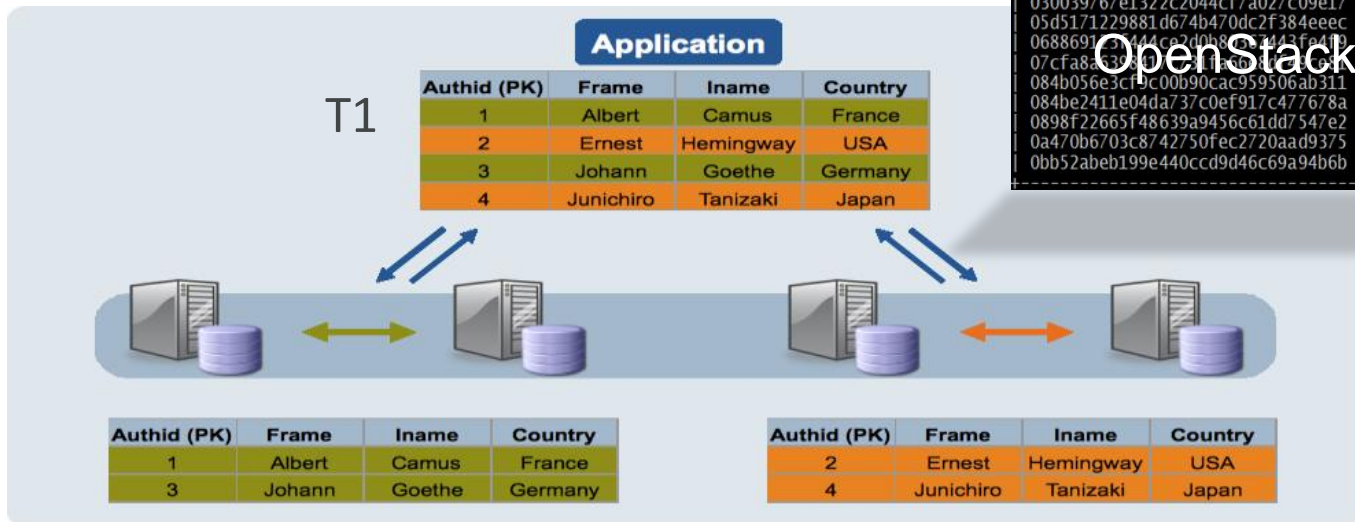
MySQL Cluster アーキテクチャ



NoSQL & SQL Access to MySQL Cluster data



自動シャーディングによる高可用性



- アプリケーションに影響なし
 - アプリケーション層でのシャーディング・ロジックは不要
 - 主キーの全てまたは一部のハッシュをベースとしてパーティショニング
 - 各ノードが1つのパーティションの主フラグメントを格納し、他のノードがそのバックアップを格納
- フェイルオーバー、アップグレード、スケールアウト時も処理を継続可能
- アプリケーションを1シャードだけ使用するトランザクションに制限する必要なし

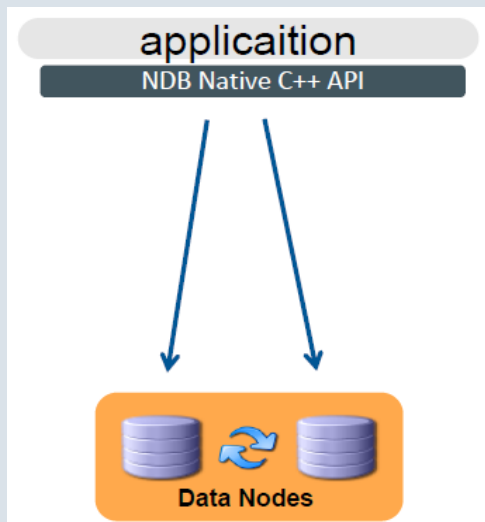
Data Nodeの拡張(例) NDB API Case

Data nodeは最大48台まで増設可能 (Node IDは、1~48を使用)

→ ノードを増やすことで、処理とデータの分散を行うので負荷を分散する事が可能

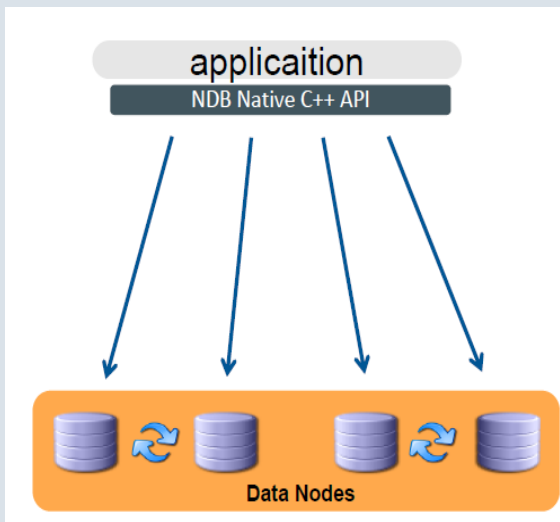
SMALL

例) Max 200M QPM and 20M UPM.



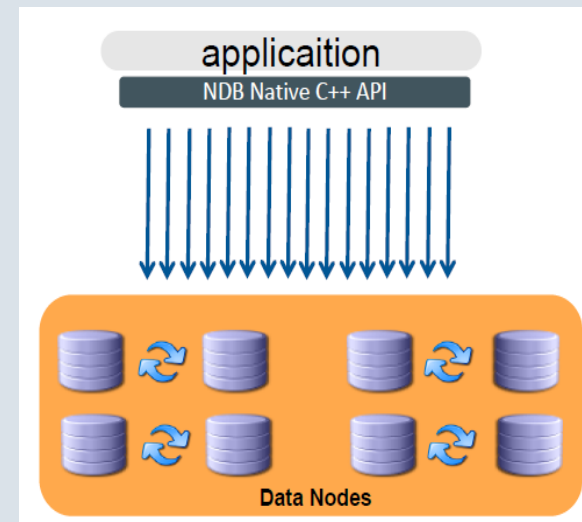
Medium

例) Max 600M QPM and 55M UPM.



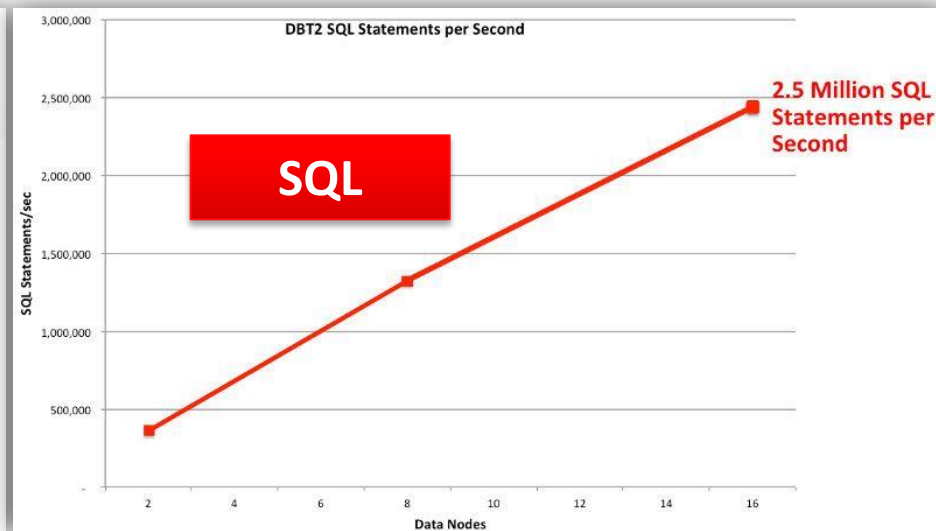
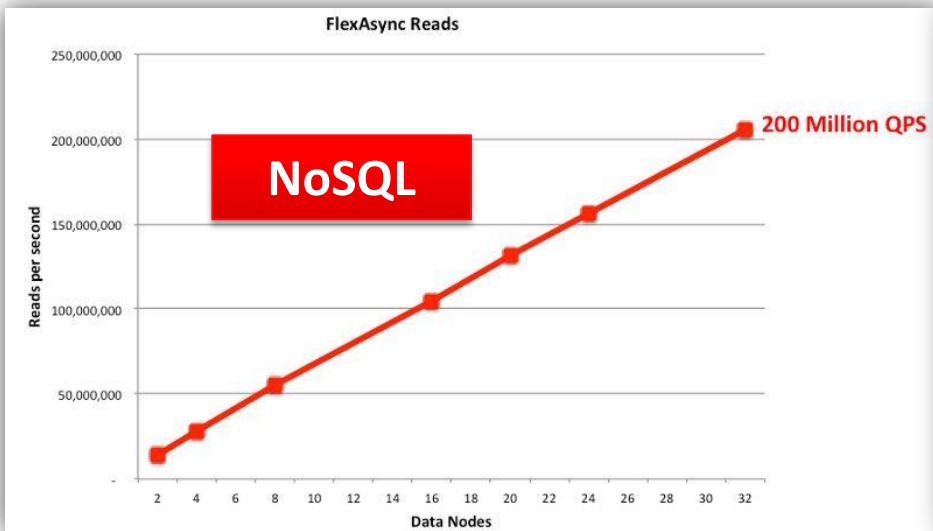
Large

例) Max 1100M QPM and 110M UPM



Data Nodeの拡張とリクエスト処理

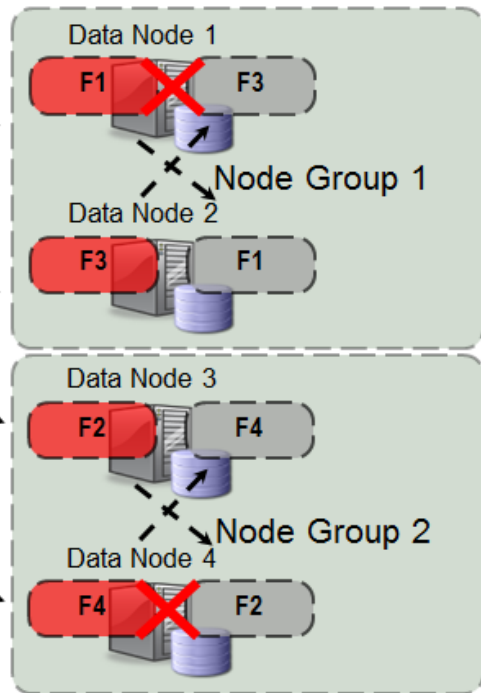
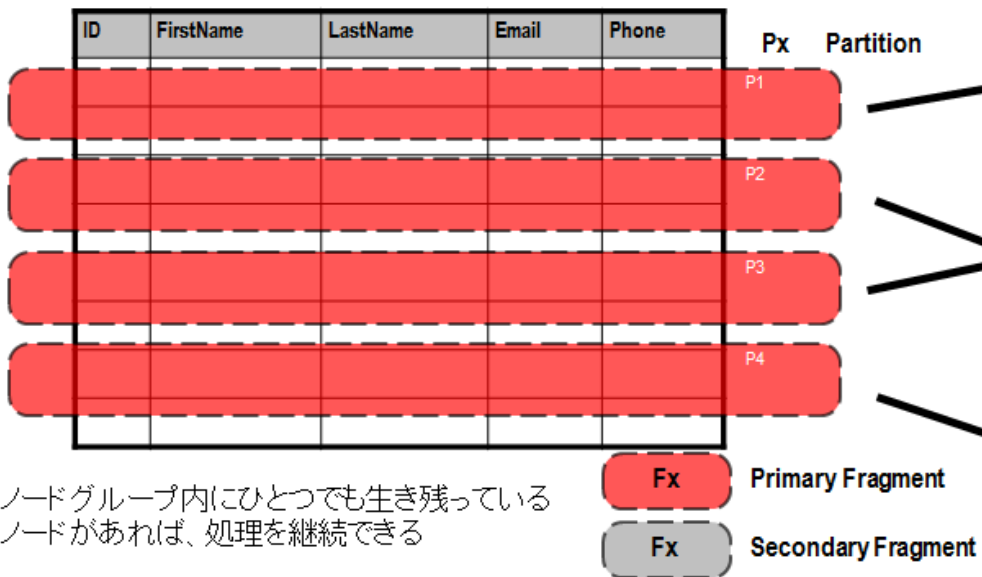
最新版のMySQL Cluster 7.4では、メモリ最適化テーブルやリニアなスケールアウト性能により、NoSQLで2億クエリ／秒の読み出し(32データノード), SQLで250万回／秒のSQL文実行(16データノード)などのパフォーマンスを実現している。



MySQL Cluster - Extreme Resilience (耐障害性)

Table T1

4パーティション × 2レプリカ = 8フラグメント

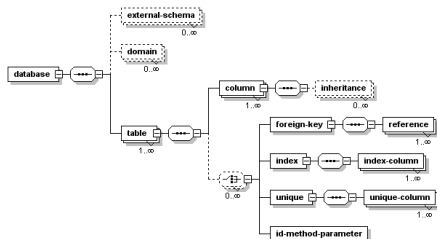


オンラインスケジュールメンテナンス



Scale

ノード追加 & 削除



Evolve

テーブル再分割 & スキーマ変更

オンラインオペレーション
高可用性・スケールアウト
構成でOpenStack
レポジトリ管理が可能

 MySQL Cluster



Backup

バックアップ



Upgrade

MySQL Cluster & OS



OpenStackにおける、 DatabaseのAuto Scale and Provisioning

Auto Scale ?

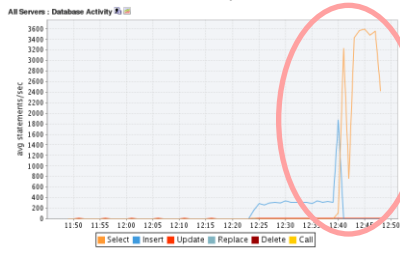


アクセス急増



Auto Scale

※ 要一定条件,トリガー定義



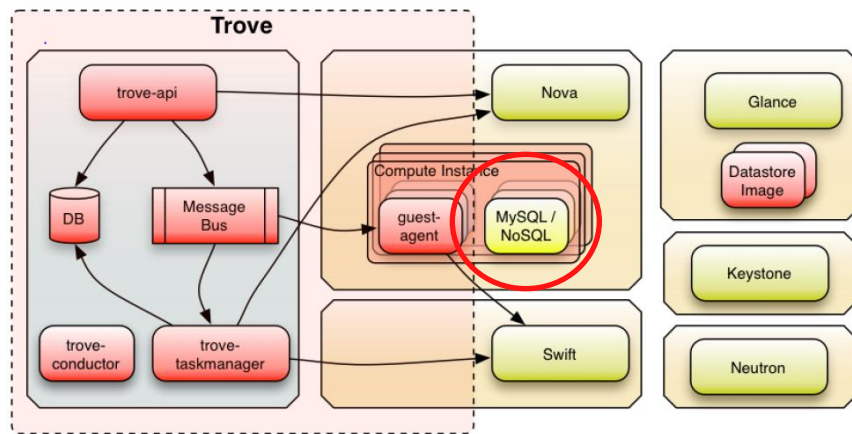
自動拡張により,アクセス急増時もリクエストを分散処理しサービスの安定稼働をサポート

MySQL & OpenStack Trove

Community Edition Only

Icehouse ~ (2014-4-17)

- Database as a ServiceをOpenStackで提供
- 複数データベースインスタンスのプロビジョニングと管理
- Single Tenant Database in Compute (Nova) Instance
- ユーザー/データベースの管理
- MySQLのようなRDBMSだけでなくNoSQLにも対応
- REST APIで全て管理する事が可能
 - インスタンスの準備
 - 複製作成
 - インスタンスのサイズ変更
 - ユーザーとDBの追加と権限の管理
 - データベースのバックアップを管理
 - データベース設定変更 (Group)



trove create

```
usage: trove create <name> <flavor_id>
       [--size <size>]
       [--databases <databases> [<databases> ...]]
       [--users <users> [<users> ...]] [--backup <backup>]
       [--availability_zone <availability_zone>]
       [--datastore <datastore>]
       [--datastore_version <datastore_version>]
       [--nic <net-id=net-uuid,v4-fixed-ip=ip-addr,port-id=port-uuid>]
       [--configuration <configuration>]
       [--replica_of <source_id>]
```

mysqlがサポートする機能を利用すると
mysqlfabric server create

```
$ trove create 名前2 --size=2 --databases DBNAME ¥
--users USER:PASSWORD --datastore_version mysql-5.6 --datastore mysql
```

参照: <https://wiki.openstack.org/wiki/Trove>
http://docs.openstack.org/cli-reference/content/troveclient_commands.html

High Availability/Scalability/Auto Scale & Provisioning



Clustering Mode	Auto-Failover	Failover Time	Scale-out
Master + Slaves	Yes	Secs	Yes

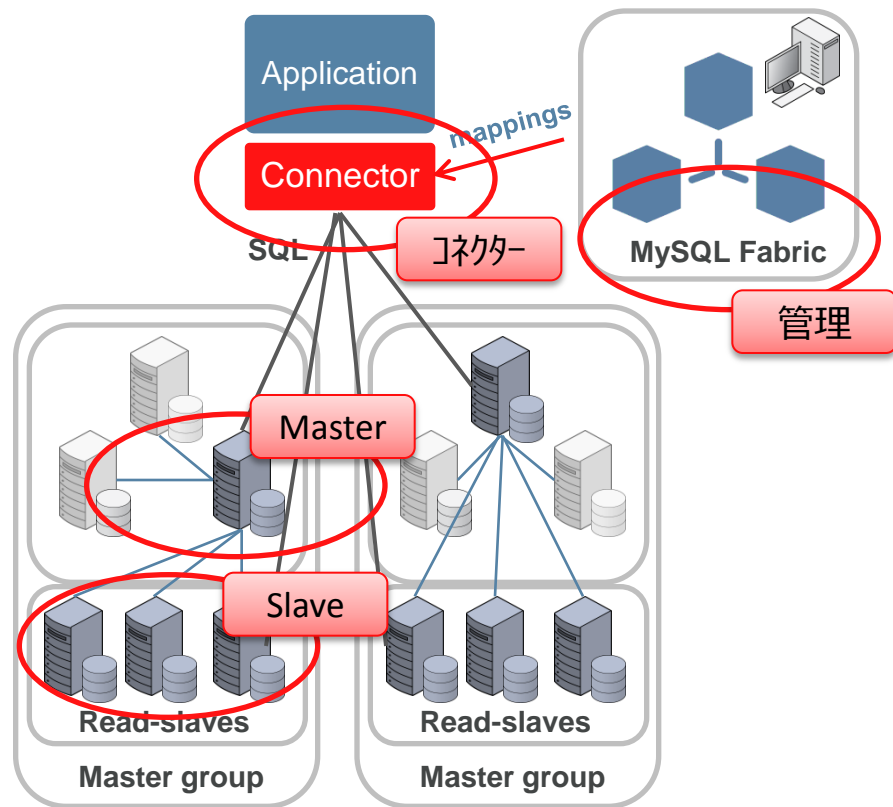
HA及びScale可能なシステム構成でOpenStack Controller用DBにも利用可能ですが、データベース接続をMySQL Fabricコネクタを使うように変更する必要がある為、Fabric概要とNovaとの連携を中心にご紹介させていただきます。

MySQL Fabricとは？

- MySQLサーバー群を管理する
統合型のフレームワーク
- 高可用性とデータ・シャーディング
による拡張性を実現する事が可能
- **OpenStack Novaとの連携**
- MySQL Utilitiesの一部として提供
(2014-05-27 ver.1.4.3～)
 - GTIDモードによるレプリケーション
機能を活用している
(MySQL 5.6.5以降で使用可能)

参照：[MySQL Fabric の特徴と利点](http://www-jp.mysql.com/products/enterprise/fabric/features.html)

<http://www-jp.mysql.com/products/enterprise/fabric/features.html>



MySQL Fabricと管理データベース

管理ノード

管理サーバーでは、XML-RPCとMySQL Protocolにてノードと通信しFabricの状態を管理



MySQL
Backing
Store

```
mysql> show tables from fabric;
+-----+
| Tables_in_fabric |
+-----+
| checkpoints      |
| error_log        |
| group_replication|
| group_view       |
| groups           |
| log              |
| machines         |
| permissions      |
| proc_view        |
| providers        |
| role_permissions |
| roles            |
| servers          |
| shard_maps       |
| shard_ranges     |
| shard_tables     |
| shards           |
| user_roles       |
| users            |
+-----+
19 rows in set (0.00 sec)
```

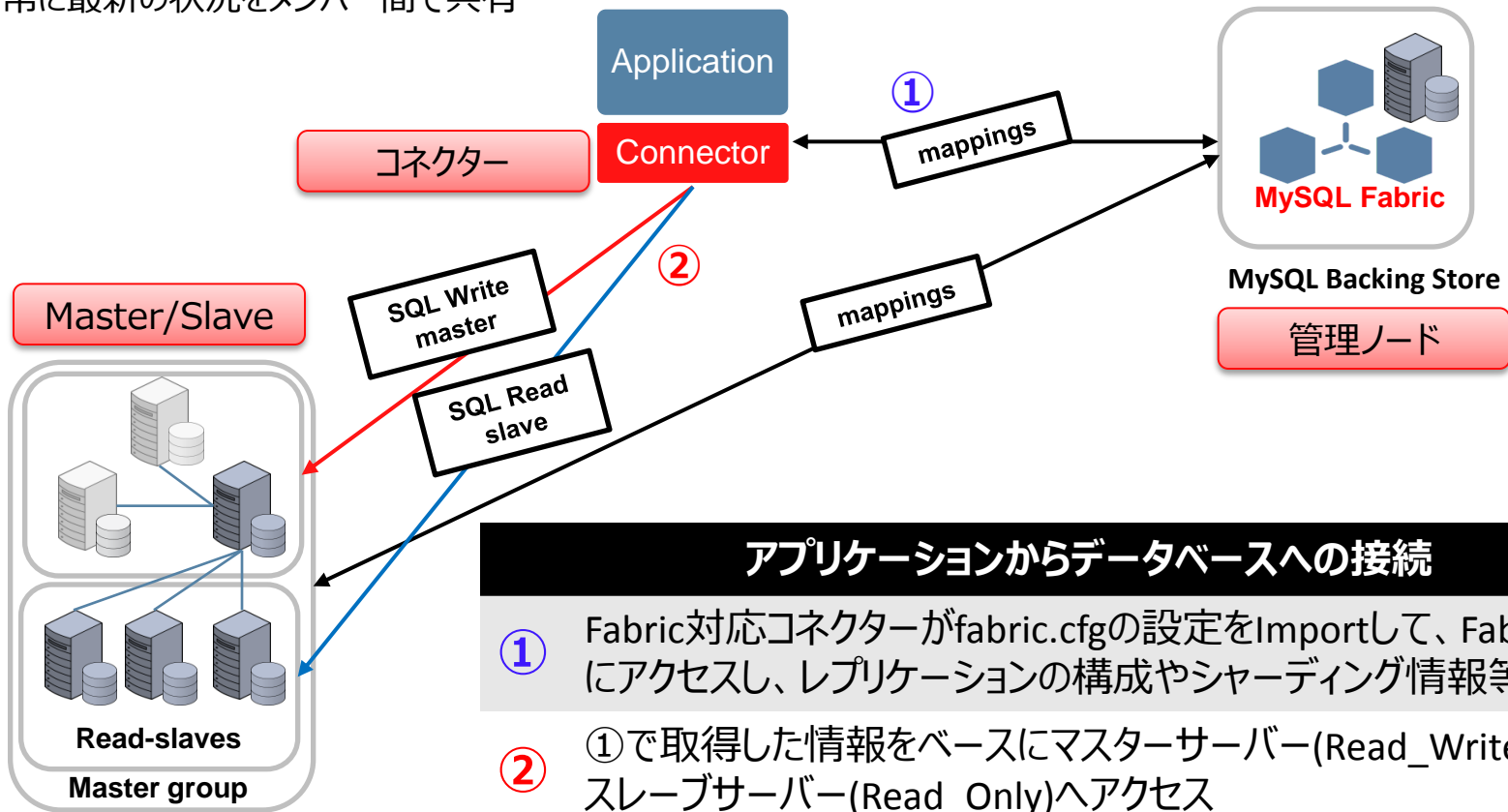
```
mysql> select * from group_view;
+-----+-----+-----+
| group_id | promote_count | demote_count |
+-----+-----+-----+
| global  | 1             | 0             |
| shard1  | 1             | 0             |
| shard2  | 1             | 0             |
+-----+-----+-----+
3 rows in set (0.08 sec)

mysql> select * from servers;
+-----+-----+-----+-----+-----+-----+
| server_uuid | server_address | mode | status | weight | group_id |
+-----+-----+-----+-----+-----+-----+
| 2dc929e4-b899-11e4-bbed-080027d65c57 | 127.0.0.1:63301 | 3    | 3      | 1      | global  |
| 2e7510f9-b899-11e4-bbed-080027d65c57 | 127.0.0.1:63302 | 1    | 2      | 1      | global  |
| 2eefc9f6-b899-11e4-bbed-080027d65c57 | 127.0.0.1:63303 | 1    | 2      | 1      | global  |
| 2f8c8c02-b899-11e4-bbed-080027d65c57 | 127.0.0.1:63304 | 3    | 3      | 1      | shard1  |
| 30359409-b899-11e4-bbed-080027d65c57 | 127.0.0.1:63305 | 1    | 2      | 1      | shard1  |
| 30ccd42e-b899-11e4-bbed-080027d65c57 | 127.0.0.1:63306 | 1    | 2      | 1      | shard1  |
| 316bea5a-b899-11e4-bbed-080027d65c57 | 127.0.0.1:63307 | 3    | 3      | 1      | shard2  |
| 31fe0dbc-b899-11e4-bbed-080027d65c57 | 127.0.0.1:63308 | 1    | 2      | 1      | shard2  |
| 32a746f9-b899-11e4-bbed-080027d65c57 | 127.0.0.1:63309 | 1    | 2      | 1      | shard2  |
+-----+-----+-----+-----+-----+-----+
```

管理用データベースには、Fabricで構成されたデータベースの最新状況を保管

Connectorからデータベースへのアクセス

常に最新の状況をメンバー間で共有



MySQL Fabricについての詳細

本日は、基本動作をデモにて確認して頂きます。
詳細は、下記Webページをご確認下さい。

MySQL Fabric

<https://www-jp.mysql.com/products/enterprise/fabric.html>

MySQL Fabric - adding High Availability and Scaling to MySQL

<http://www.clusterdb.com/mysql-fabric/mysql-fabric-adding-high-availability-and-scaling-to-mysql>

MySQL Fabric 基本機能Short Demo (可用性・読み込みスケール)

1) Fabricグループ構成の確認

```
mysqlfabric group lookup_servers global
```

2) FabricグループにSlaveの追加 (対応例:参照増加)

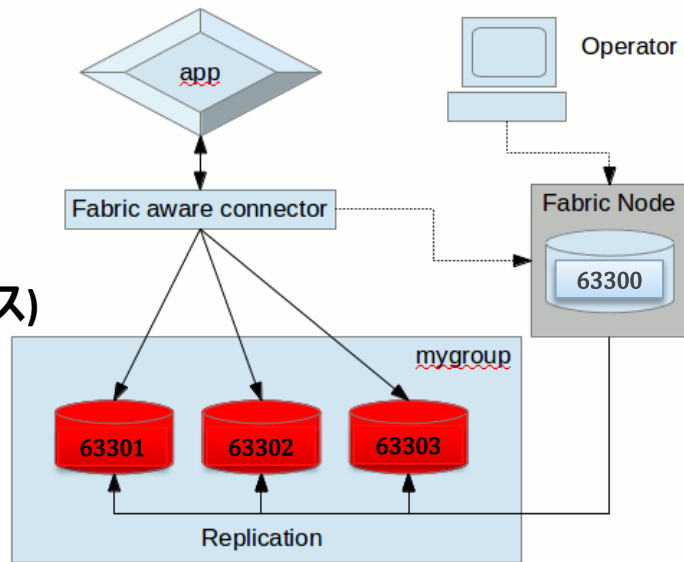
```
mysqlfabric group add global 127.0.0.1:63304  
mysqlfabric group add global 127.0.0.1:63305
```

3) 特定のSlaveをマスターに昇格 (対応例:メンテナンス)

```
mysqlfabric group promote global --slave_id=xxx
```

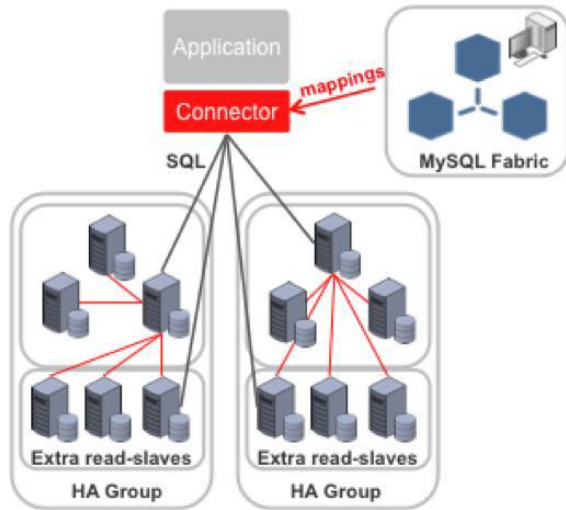
4) 疑似障害と可用性確認 (対応例:マスター障害)

```
kill <pid>
```

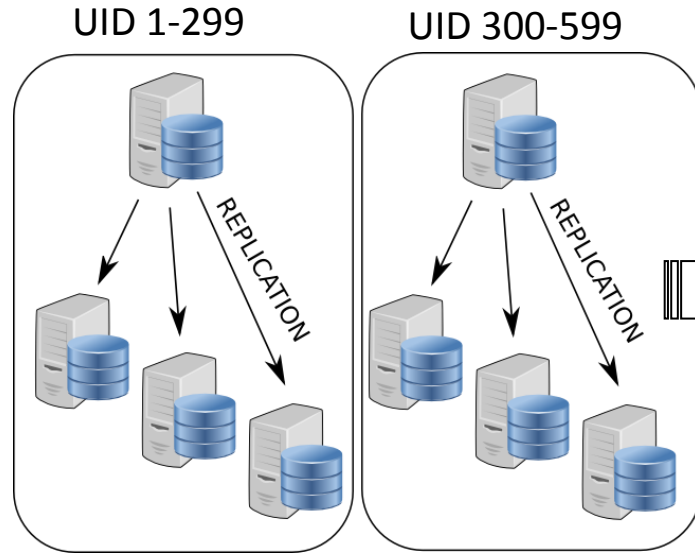


Sharding with Fabric

- 書き込みスケーラビリティ
 - 大規模なデータセット
 - 性能改善
- より多くの書き込みを処理することが可能
大き過ぎるデータベース/単一サーバーに収まらないデータ
小さなインデックスサイズ/ワーキングセットに分割



Single Master in each HA Group.



```
mysql> select * from shard_ranges;
+-----+-----+-----+
| shard_mapping_id | lower_bound | shard_id |
+-----+-----+-----+
| 1 | 1 | 1 |
| 1 | 300 | 2 |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from shard_tables;
+-----+-----+-----+
| shard_mapping_id | table_name | column_name |
+-----+-----+-----+
| 1 | test.employees | emp_no |
+-----+-----+-----+
1 row in set (0.00 sec)
```

キー範囲分割
UID 600-899
UID 900-1199
UID 1200-1499
.....

MySQL Fabric 書き込みSharding Short Demo

1) Fabricグループ構成の確認

2) FabricにSharding Groupを追加 (対応例:データ増加による遅延)

```
mysqlfabric group create shard#
```

```
mysqlfabric group add shard# サーバー
```

```
mysqlfabric sharding add_table 1 test.employees emp_no
```

```
mysqlfabric sharding add_shard 1 shard1/1,shard2/300
```

3) Fabricグループ構成の確認

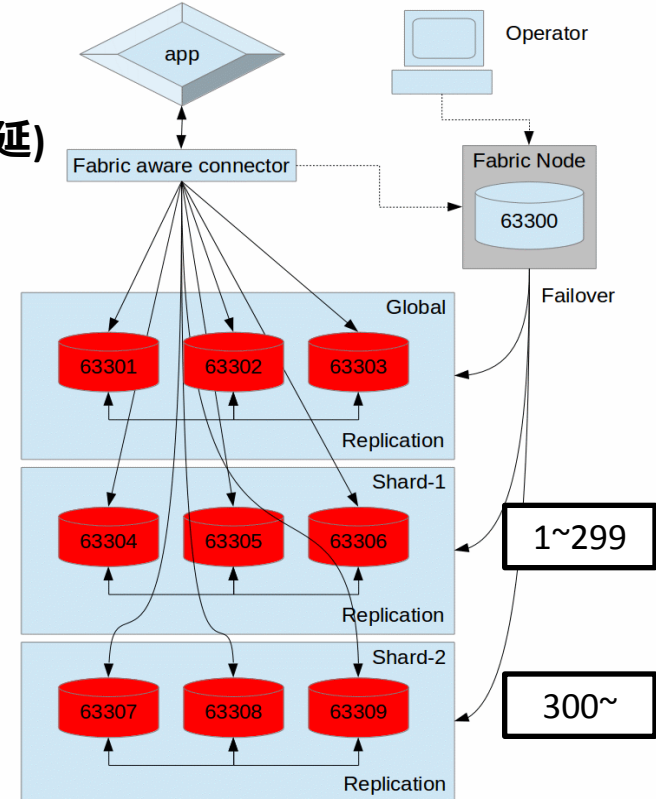
```
# shard1 (Server:63304,63305,63306)
```

```
# shard2 (Server:63307,63308,63309) 300以上
```

```
mysqlfabric dump sharding_information
```

```
mysqlfabric group lookup_servers グループ名
```

4) 書き込みデータの分散状況確認



Provisioning by MySQL Fabric with Nova

MySQL Fabricは、OpenStackなどのクラウドフレームワークと連携し、ベアメタル及び仮想環境においてMySQL導入の自動化をサポートします。また、柔軟にMySQLを準備しスケールアウトさせる事を可能にします。新規サーバー利用開始時には、透過的にMySQLとレプリケーションを設定します。

参照負荷増加
参照処理の増加

問題:
参照処理の高負荷状態によるレスポンス遅延

対応:
参照サーバーをコマンド1つで動的に追加

書き込み処理増加
書き込み増・高負荷

問題:
書き込みデータ増加による、全体的な処理遅延の発生


対応:
テーブルの書き込みデータを複数サーバーにシャーディング

サーバー障害発生時
グループ内サーバー障害発生時

問題:
サーバー障害による、サービス停止と機会損失

対応:
スレーブサーバーをマスターへ自動的に昇格

mysqlfabricコマンドとOpenStack(Nova)連携

インスタンス名	シ名	IP アドレス	サイズ	ペア	状態	ディザーン	タスク	稼働状態	稼働時間	アクション
<input type="checkbox"/> nova_mysql01	ol65	192.168.56.242	mysql1.micro 256MB メモリー 1仮想 CPU 8.0GB ディスク	-	Build	nova		No State	0分	Floating IPの割り当て

1項目を表示中

```
[admin@Fabric01 ~]$ mysqlfabric help provider register
provider register provider_id username password url [--tenant=NONE]
[--provider_type=OPENSTACK] [--default_image=NONE] [--default_flavor=NONE]
[--extra=NONE] [--synchronous]
Register a provider.
```

プロバイダー, APIアドレス, ユーザー, パスワード等の登録

```
[admin@Fabric01 ~]$ mysqlfabric help server create
server create provider_id [--image=NONE] [--flavor=NONE] [--number_machines=1]
[--availability_zone=NONE] [--key_name=NONE] [--security_groups=NONE]
[--private_network=NONE] [--public_network=NONE] [--userdata=NONE] [--swap=NONE]
[--scheduler_hints=NONE] [--meta=NONE] [--datastore=NONE]
[--datastore_version=NONE] [--size=NONE] [--databases=NONE] [--users=NONE]
[--configuration=NONE] [--security=NONE] [--skip_store] [--wait_spawning]
[--synchronous]
Create a virtual machine instance:
```

登録済みプロバイダーから, イメージ指定したプロビジョニング等


※ コマンド実行には、[OpenStack コマンドラインクライアント](#)のインストールが必要です。

Server Provisioning – NovaとFabricの連携

プロバイダー登録

```
> mysqlfabric provider register mysqlfabric01 my_user my_password ¥  
> http://8.21.28.222:5000/v2.0/ --tenant=my_user_role --provider_type=OPENSTACK ¥  
> --default_image=mysql_img --default_flavor=nova_mysql01
```

```
-bash-4.2$ mysqlfabric provider list  
Fabric UUID: 5ca1ab1e-a007-feed-f00d-cab3fe13249e  
Time-To-Live: 1
```



provider_id	type	username	url	tenant	default_image	default_flavor	extra
mysqlfabric01	OPENSTACK	my_user	http://8.21.28.222:5000/v2.0/	my_user_role	mysql_img	nova_mysql01	None

サーバーとMySQLのプロビジョニング

```
> mysqlfabric server create my_stack ¥  
> --image name="Oracle Linux 7 amd64" ¥  
> --flavor name=m1.small ¥  
> --userdata=mysql-oracle-linux-init
```



1. OpenStack Nova APIをCALL
- サーバー準備
2. Slaveクローンの作成
3. レプリケーションの自動設定
4. カスタム操作を実行

まとめ

#	概要
1	OpenStackリポジトリデータベースの可用性構成として、DRBD + MySQL, MySQL Cluster
2	MySQL Clusterは、OpenStackリポジトリデータベースの書き込みデータ増加に対応可能なシャーディングをサポート
3	MySQL Fabricは可用性・シャーディングをサポートし、OpenStack Novaと連携し、MySQLのAuto Scaleをサポート

有難うございました



参考情報:MySQL Commercial版概要

MySQL Enterprise Edition 管理ツールと拡張機能概要

MySQL Enterprise Edition

MySQL Enterprise Monitor	複数サーバの一括管理、クエリ性能分析
MySQL Enterprise Backup	高速なオンラインバックアップ、ポイントインタイムリカバリ
MySQL Enterprise Scalability	Thread Poolプラグインによる性能拡張性の向上
MySQL Enterprise Security	LDAPやActive Directoryとの外部認証と統合管理
MySQL Enterprise Audit	ユーザ処理の監査、Oracle DBと同じツールで管理可能
MySQL Enterprise Encryption	非対称暗号化(公開鍵暗号)の業界標準機能を提供
Oracle Enterprise Manager for MySQL	Oracle Enterprise ManagerからMySQLを統合管理
Oracle Premier Support	24x365, インシデント無制限、コンサルティングサポート

	MySQL Editions		
	Standard SE	Enterprise EE	Cluster CGE
機能概要			
MySQL Database	✓	✓	✓
MySQL Connectors	✓	✓	✓
MySQL Replication	✓	✓	✓
MySQL Fabric		✓	✓
MySQL Partitioning		✓	✓
MySQL Utilities		✓	✓
Storage Engine: MyISAM, InnoDB	✓	✓	✓
Storage Engine: NDB (ndbcluster)			✓
MySQL Workbench SE/EE*	✓	✓	✓
MySQL Enterprise Monitor*		✓	✓
MySQL Enterprise Backup*		✓	✓
MySQL Enterprise Security (外部認証サポート) *		✓	✓
MySQL Enterprise Audit (ポリシーベース監査機能) *		✓	✓
MySQL Enterprise Encryption (非対称暗号化) *		✓	✓
MySQL Enterprise Scalability (スレッドプール) *		✓	✓
MySQL Enterprise High Availability (HAサポート) *		✓	✓
Oracle Enterprise Manager for MySQL*		✓	✓
MySQL Cluster Manager (MySQL Cluster管理) *			✓
MySQL Cluster Geo-Replication			✓

Community EditionでFabricをご利用可能ですが、Enterprise版ではサポートも提供させて頂いております。

	MySQL Editions		
	Standard SE	Enterprise EE	Cluster CGE
Oracle Premium Support			
24時間365日サポート	✓	✓	✓
インシデント数無制限	✓	✓	✓
ナレッジベース	✓	✓	✓
バグ修正&パッチ提供	✓	✓	✓
コンサルティングサポート	✓	✓	✓
オラクル製品との動作保証			
Oracle Linux	✓	✓	✓
Oracle VM	✓	✓	✓
Oracle Solaris	✓	✓	✓
Oracle Enterprise Manager		✓	✓
Oracle Golden Gate		✓	✓
Oracle Data Integrator		✓	✓
Oracle Fusion Middleware		✓	✓
Oracle Secure Backup		✓	✓
Oracle Audit Vault and Database Firewall		✓	✓

MySQL Enterprise Editionを利用した場合のPDCAライフサイクル

開発段階における、データベースデザインツールとドキュメント保存及び納品。

導入段階における、パフォーマンスの確認と可視化による、アプリケーション構成の最適化と確認。

運用段階における、サービスレベルの安定化と機会損失削減をサポート。

開発

- MySQL Workbench
 - スキーマデザイン
- コンサルティングサポート
 - 開発段階におけるMySQL構成のアドバイス

導入

- MySQL Enterprise Monitor (POC)
 - 負荷試験
- MySQL Workbench
 - DB移行

運用

- MySQL Enterprise Monitor
 - パフォーマンス管理
 - Monitor/Alert
- Enterprise Backup
 - 可用性の向上
- サポート
 - 24x365対応、無制限

開発フェーズ

Design

- Workbench

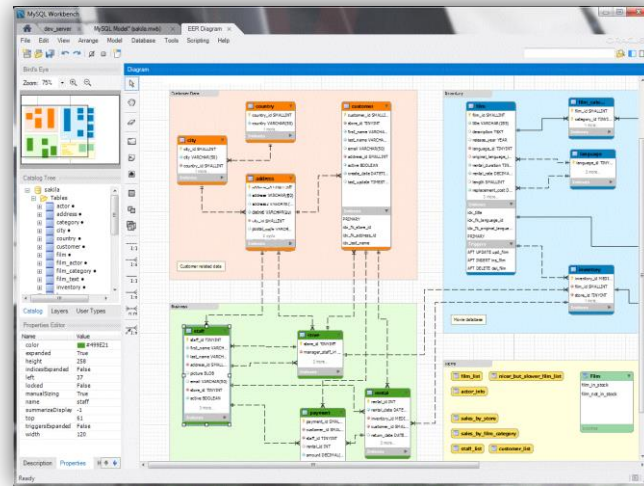
Review

- コンサルティングサポート



Out Put

- ドキュメント作成 (納品物)



DBドキュメント出カ,データモデルの検証

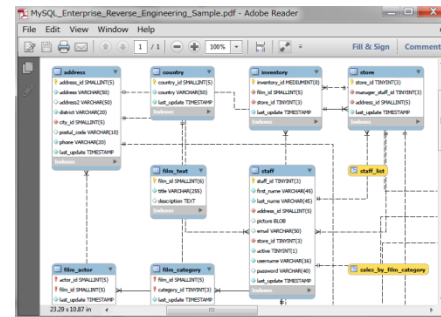
- リモート・トラブル・シューティング
- レプリケーション・レビュー
- パーティショニング・レビュー
- スキーマ・レビュー
- クエリー・レビュー
- パフォーマンス・チューニング
- ...and more

Table actor					
Key	Column Name	Data Type	Not Null	Default	Comment
PK	actor_id	SERIAL(4)	Yes		
	first_name	VARCHAR(45)	Yes		
	last_name	VARCHAR(45)	Yes		
	last_update	TIMESTAMP	Yes	CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	

Indexes						
Index Name	Columns	Primary	Unique	Type	Incl	Comment
PRIMARY	actor_id	Yes	No	PRIMARY		
	idx_actor_last_name	No	No	INDEX		

Relationships				
Relationship Name	Relationship Type	Parent Table	Child Table	Card.
fk_actor_actor	Non-Identifying	actor	fk_actor	1:n

Table address					
Key	Column Name	Data Type	Not Null	Default	Comment
PK	address_id	SERIAL(4)	Yes		
	address	VARCHAR(45)	Yes		
	address2	VARCHAR(45)	No	NULL	
	address3	VARCHAR(45)	Yes		



導入フェーズ

Design

- 最終レビュー

Review

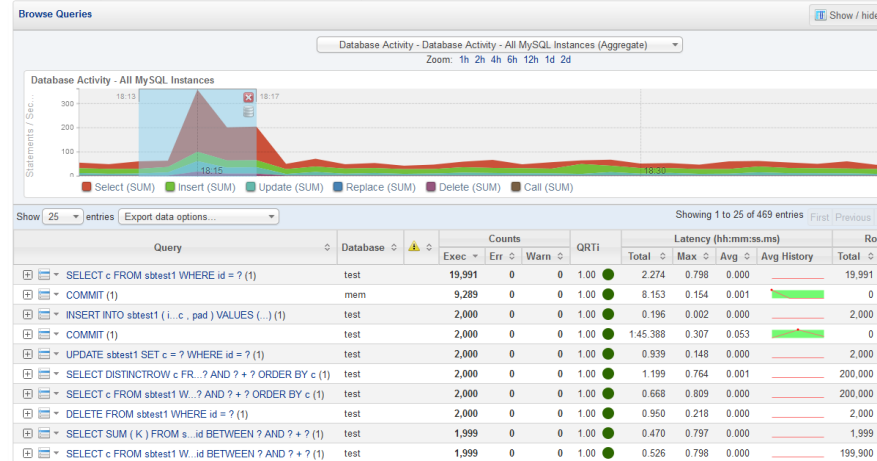
- POC (本番若しくは、本番同等環境)

Out Put

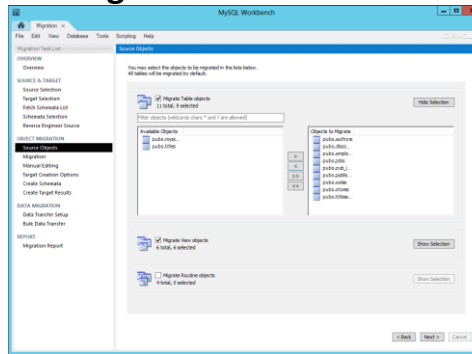
- 導入&リリース

- 負荷試験 (Benchmark)
- 冗長化試験 (Replication, Cluster,その他)
- Slow Log確認
- パフォーマンス確認(Enterprise Monitor)
- DB設定確認(Advisor)
- ...and more

Enterprise Query Analyzer



DB Migration Tool



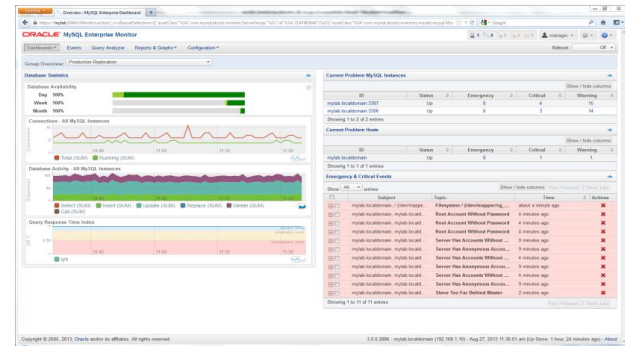
Best Practice Advisors

Advisor	Configured
Administration	Configured: 26 of 26
Agent	Configured: 3 of 3
Availability	Configured: 6 of 6
Backup	Configured: 5 of 5
Cluster	Configured: 10 of 10
Graphing	Configured: 87 of 87
Memory Usage	Configured: 6 of 6
Monitoring and Support Services	Configured: 5 of 5
Operating System	Configured: 5 of 5
Performance	Configured: 23 of 23
Query Analysis	Configured: 4 of 4
Replication	Configured: 13 of 13
Schema	Configured: 17 of 17
Security	Configured: 26 of 26

運用フェーズ

MySQL Enterprise Monitor

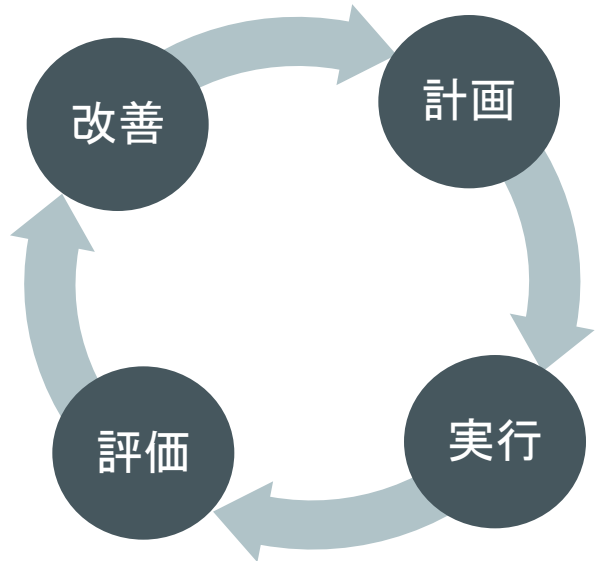
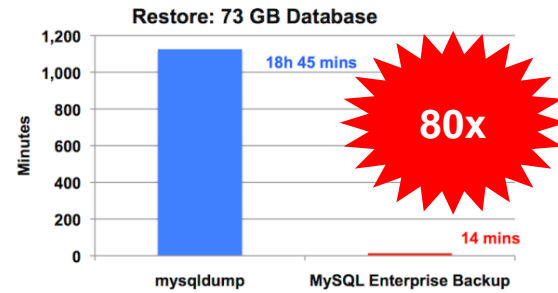
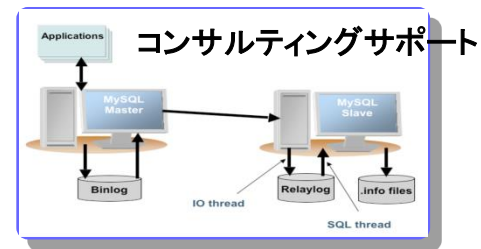
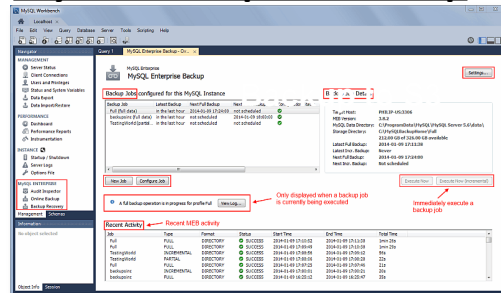
- Monitoring Dashboard
- Enterprise Query Analyzer
- Enterprise Replication
- Best Practice Advisors
- Backup Monitoring



MySQL Utilities

```
# Replication Topology Graph
192.168.56.101:3306 (MASTER)
+--- 192.168.56.102:3306 [10: Yes, SQL: Yes] - (SLAVE)
+--- 192.168.56.112:3306 [10: Yes, SQL: Yes] - (SLAVE)
mysql>
```

MySQL Enterprise Backup



その他、運用サポート

MySQL Enterprise Security

LDAPやWindows Active Directoryとの外部認証と統合管理

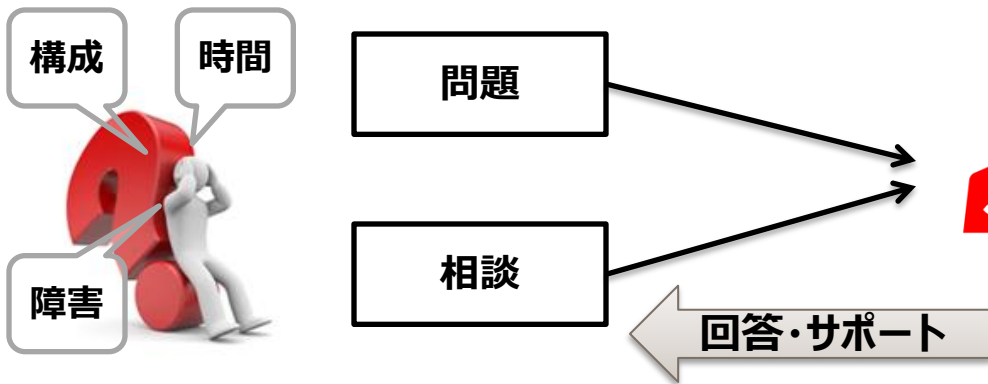
MySQL Enterprise Audit

ユーザ処理の監査、Oracle DBと同じツールで管理可能

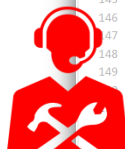
MySQL Enterprise Encryption

非対称暗号化([公開鍵暗号](#))の業界標準機能を提供

24x7x365 インシデント対応無制限, コンサルティングサポート, 様々なオプションとツールが利用可能に



```
138 /**
139  * Replication binlog transmitter (binlog dump) observer parameter.
140  */
141 typedef struct Binlog_transmit_param {
142     uint32 server_id;
143     uint32 flags;
144     /* Let us keep 1-16 as output flags and 17-32 as input flags */
145     static const uint32 F_OBSERVE= 1;
146     static const uint32 F_DONT_OBSERVE= 2;
147
148     void set_observe_flag() { flags|= F_OBSERVE; }
149     void set_dont_observe_flag() { flags|= F_DONT_OBSERVE; }
150 }
151 /**
152  * If F_OBSERVE is set by any plugin, then it should observe binlog
153  * transmission, even F_DONT_OBSERVE is set by some plugins.
154
155  * If both F_OBSERVE and F_DONT_OBSERVE are not set, then it is an old
156  * plugin. In this case, it should always observe binlog transmission.
157  */
158 bool should_observe()
159 {
160     return (flags & F_OBSERVE) || !(flags & F_DONT_OBSERVE);
161 }
162 } Binlog_transmit_param;
```



MySQL Enterprise Edition効果



- 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
- 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



Get Started Today!

MySQL Enterprise Edition Trial



30日間トライアル

Oracle Software Delivery Cloud

<http://edelivery.oracle.com/>

製品パックを選択：“MySQL Database”

事例紹介：<http://www.mysql.com/why-mysql/case-studies/#ja-5-0>

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

MySQL Editions詳細

<http://www-jp.mysql.com/products/>

Contact a MySQL Sales Rep



[MySQL お問い合わせ窓口]

電話：0120-065556

【受付時間】平日 9:00-12:00/13:00-18:00
(祝日及び年末年始休業日を除きます)

メール：MySQL-Sales_jp_grp@oracle.com

URL：<http://www.mysql.com/about/contact/>

ORACLE

【MySQL Cluster参考情報】

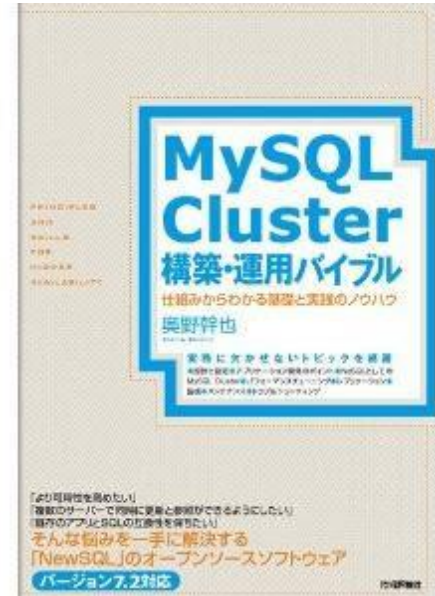
Community Edition (GPL) <http://dev.mysql.com/downloads/cluster/>



Commercial Edition 30日間トライアル <https://edelivery.oracle.com/>



MySQL Cluster 関連書籍



【MySQL最新情報】

■ NoSQLで2億クエリ／秒など高いパフォーマンスと可用性を実現 [2015-02-26]

パフォーマンス大幅向上、「MySQL Cluster 7.4」GAリリース

<http://ascii.jp/elem/000/000/985/985073/>

■ MySQL 5.7.6 DMRリリース [2015-03-26]

A character-based ngram full-text parser that supports Chinese, Japanese, and Korean (CJK), and a word-based MeCab parser plugin that supports Japanese were introduced in MySQL 5.7.6, for use with InnoDB tables.

<http://dev.mysql.com/doc/relnotes/mysql/5.7/en/news-5-7-6.html>

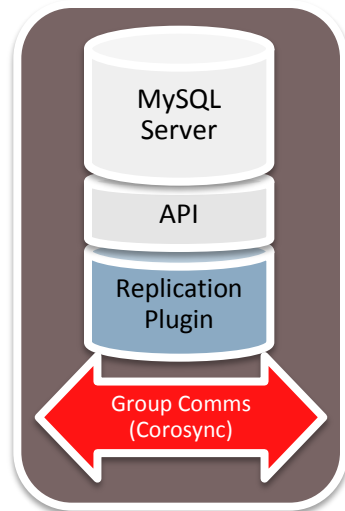
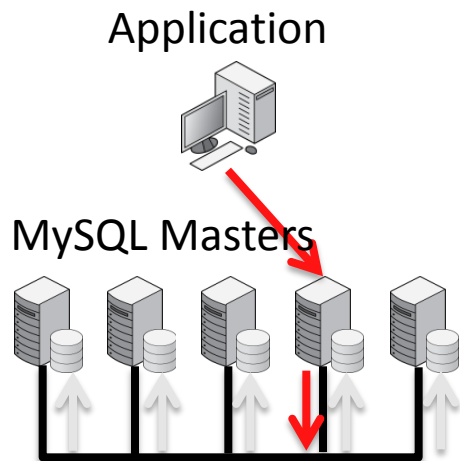
■ Oracle's MySQL image in Docker Hub Registry [2015-02]

<https://registry.hub.docker.com/u/mysql/mysql-server/>

Upcoming ... MySQL 5.7: グループレプリケーション



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



- シェアード・ナッシング型”疑似”同期レプリケーション
- 更新はマルチ・マスタ型でどこでも可能
 - 矛盾の検知と解決(トランザクションのロールバック)
 - “Optimistic State Machine” レプリケーション
- グループメンバーの管理と障害検知を自動化
 - サーバのフェールオーバー不要
 - 構成の拡張/縮小の柔軟性
 - 単一障害点無し
 - 自動再構成
- 既存構成との統合
 - InnoDB
 - GTIDベースのレプリケーション
 - PERFORMANCE_SCHEMA