MySQL Cluster 7.4
開発最新動向

Ryusuke Kajiyama / 梶山隆輔
MySQL Sales Consulting Senior Manager, Asia Pacific & Japan

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Today's Database Requirements

Extreme Write Scalability

Real-Time User Experience

Rock Solid Reliability

Rapid Service Innovation

Today's Database Requirements

Extrem

ence

Transactional Integrity
OLTP + Real-Time Analytics
Standards & Skillsets

Rock Solid Reliability

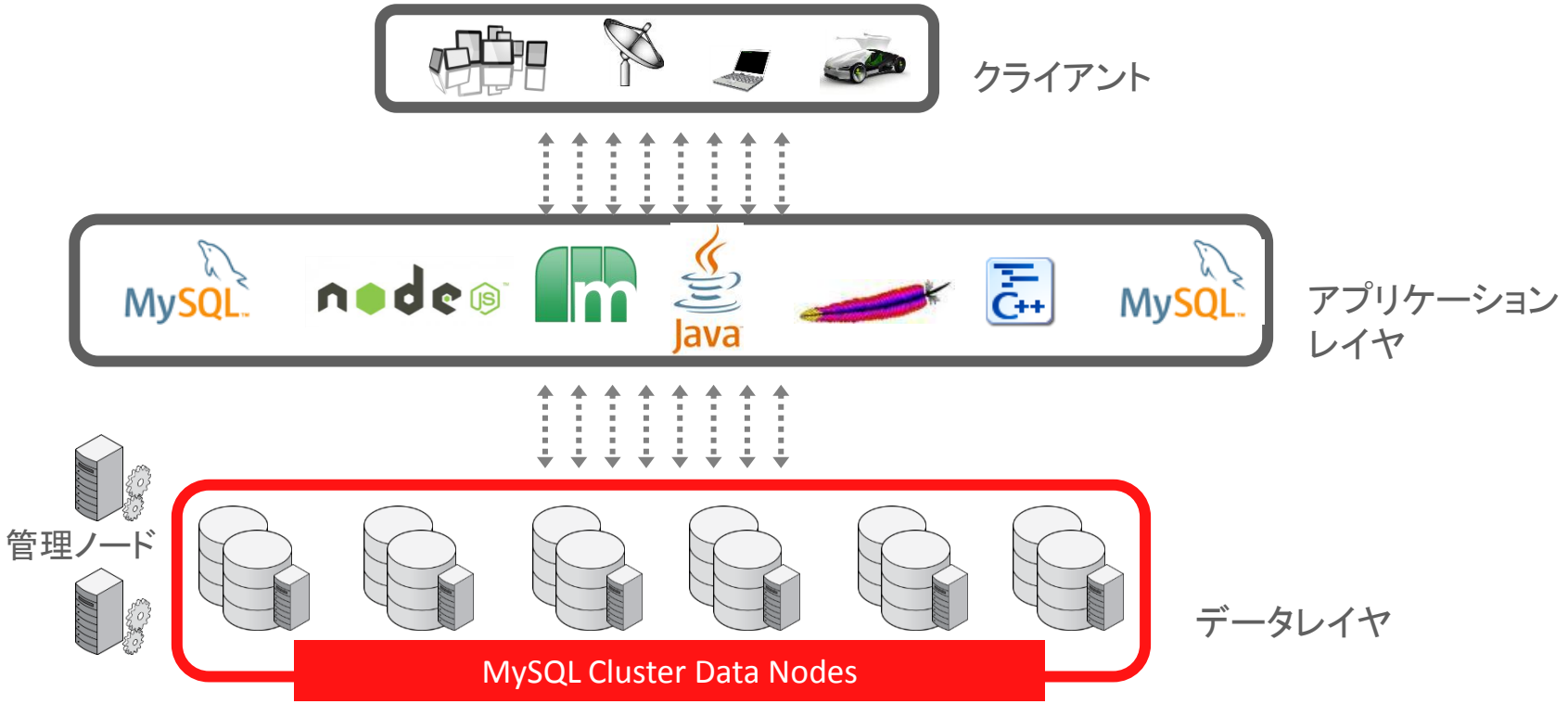
Rapid Service Innovation

Who's Using MySQL Cluster

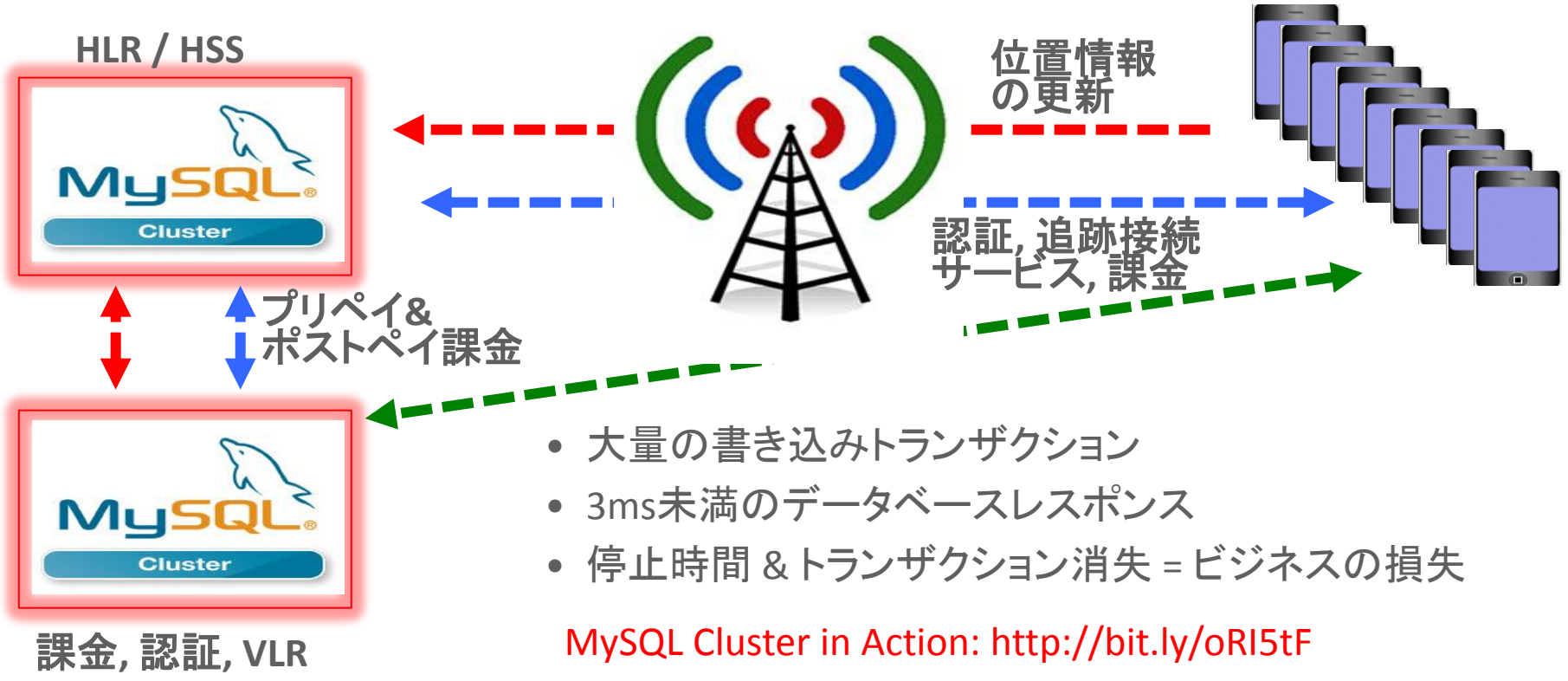
MySQL Cluster

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	• シェアードナッシング、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・バリュー型＋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

MySQL Cluster アーキテクチャ



導入事例: 携帯電話ネットワーク

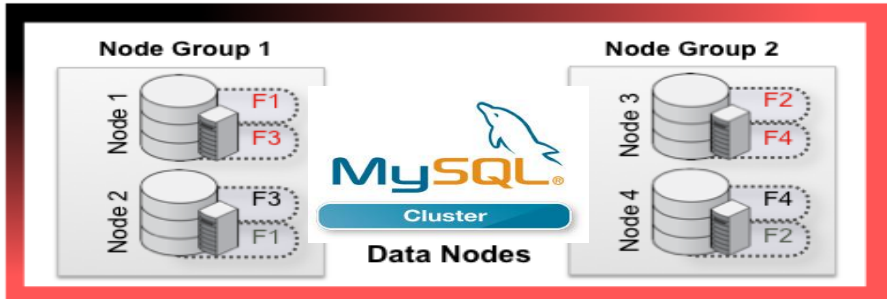


- 大量の書き込みトランザクション
- 3ms未満のデータベースレスポンス
- 停止時間 & トランザクション消失 = ビジネスの損失

MySQL Cluster in Action: http://bit.ly/oRI5tF

ORACLE

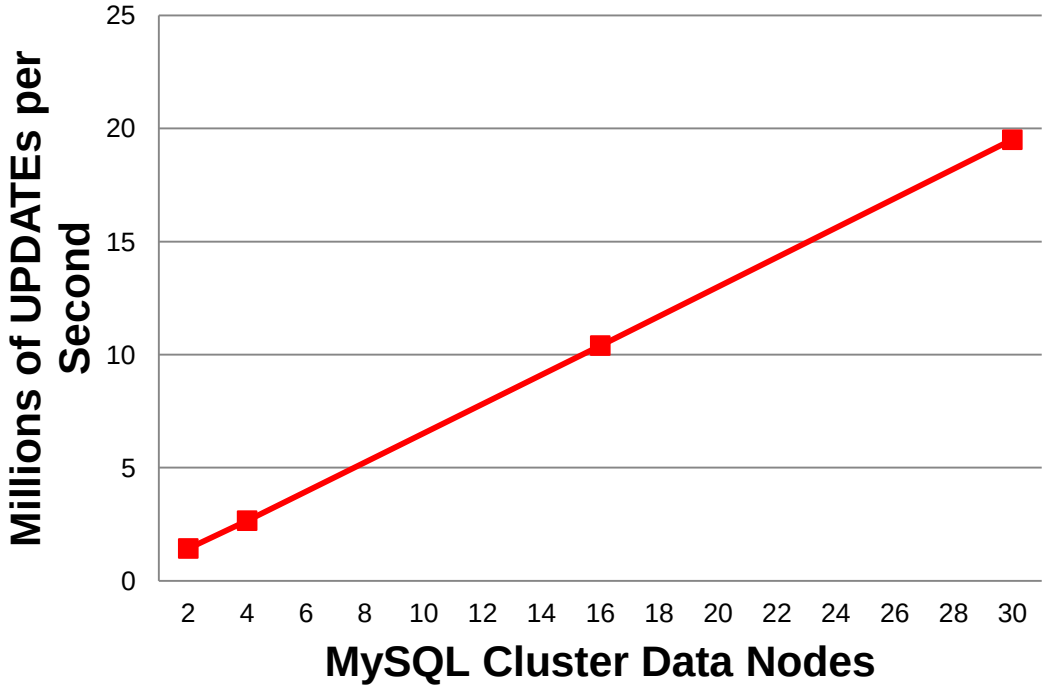
導入事例: 航空機管制システム



- 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

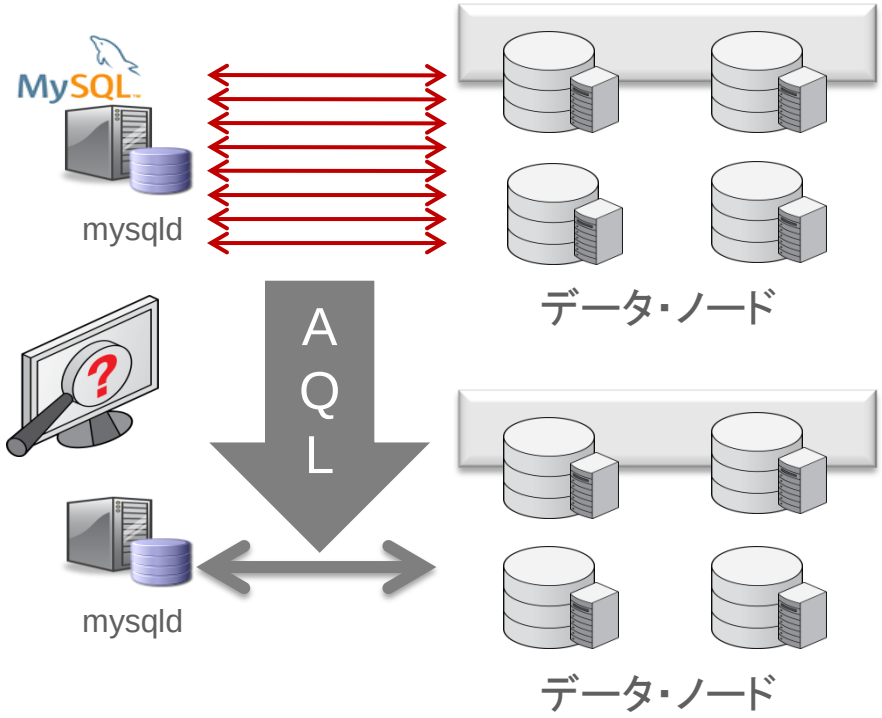
1.2 Billion UPDATEs per Minute



- NoSQL C++ API, flexaSynch benchmark
- 30 x Intel E5-2600 Intel Servers, 2 socket, 64GB
- ACID Transactions, with Synchronous Replication

アダプティブ・クエリー・ローカライゼーション

70倍以上の
パフォーマンス向上

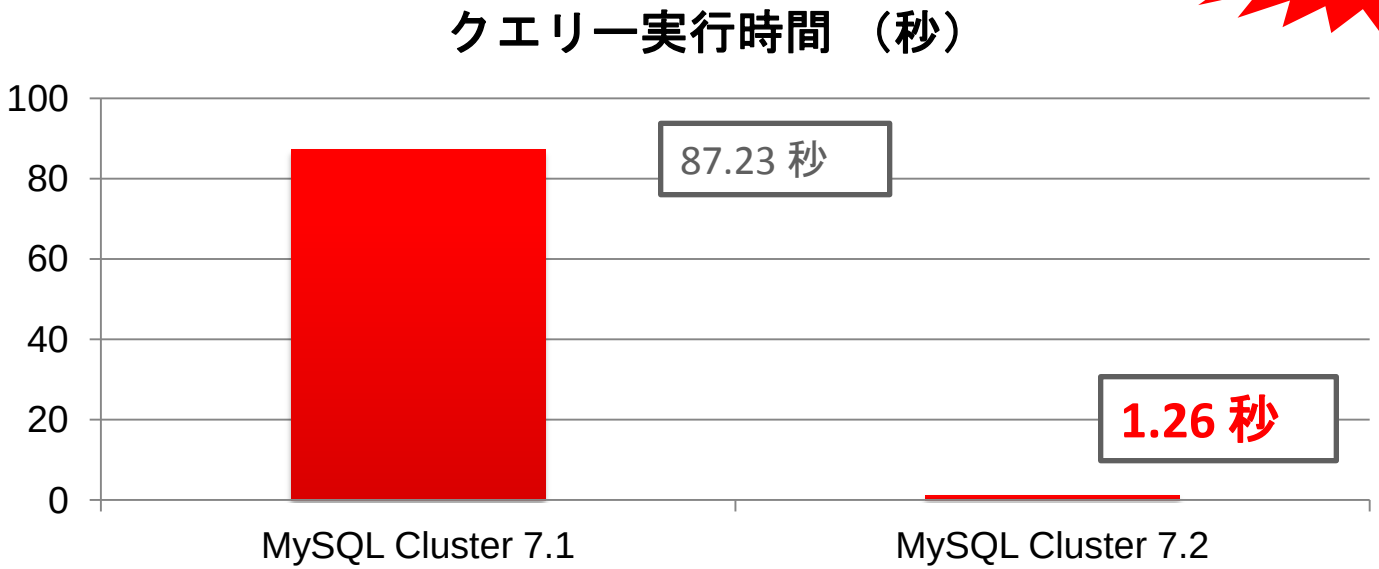


- シャード間で複雑なクエリを実行
  - JOIN処理をデータ・ノードに移行
  - 並列実行
  - 一つの結果セットをMySQLに戻す
- これまでは性能的に難しかった処理も実行可能に
  - リアルタイム分析
  - レコメンデーション・エンジン
  - クリックストリームを分析

機能性を犠牲にせず、スケールアウトを向上!!

実例テストケース

70倍高速



- Webベース・コンテンツ管理システム
  - 11テーブル、33,500行を結合(JOIN)
  - 結果セット：2,060行、1行あたり19列を返す

MySQL Cluster 7.3

Developer Power

Developer Simplicity

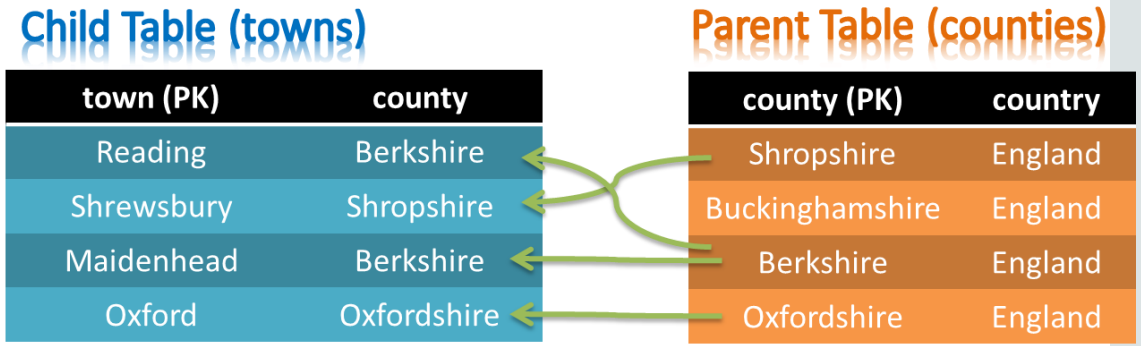
Learn More »

- Foreign Key Support
- MySQL 5.6
- Auto-Installer
- NoSQL JavaScript for node.js

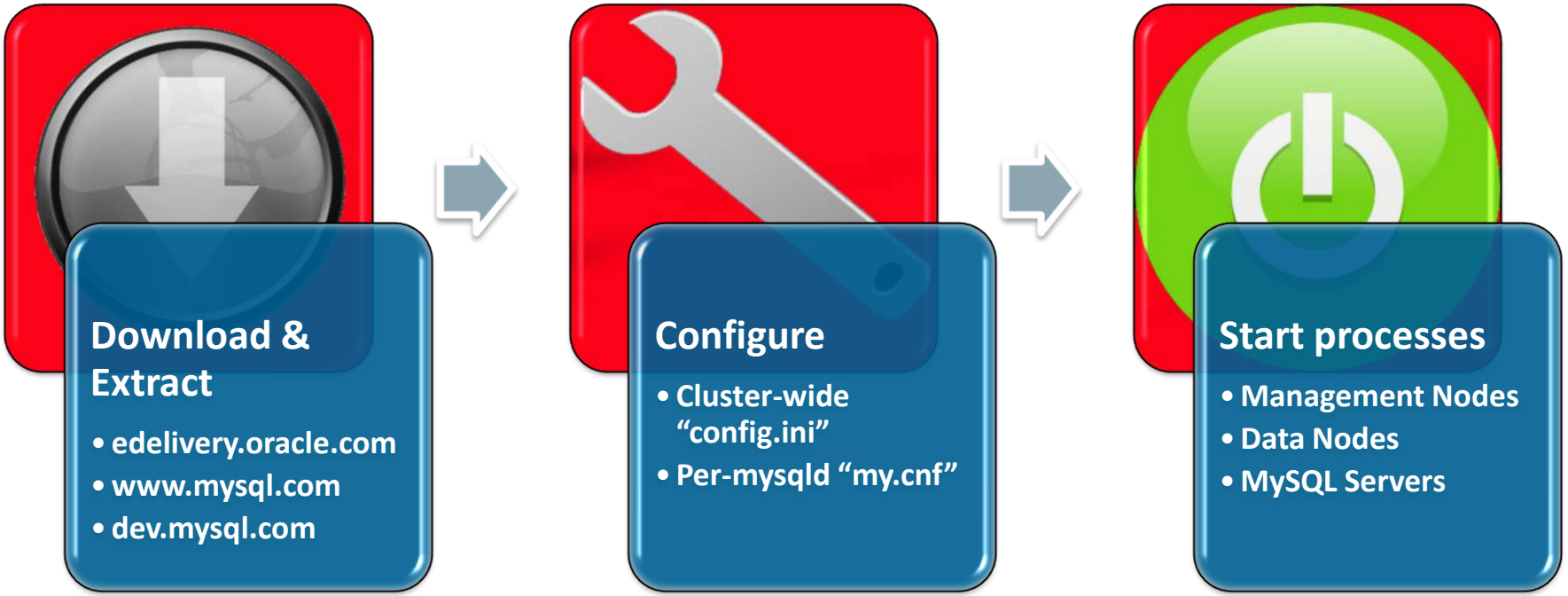
ORACLE

MySQL Cluster 7.3 GA: 外部キー

- MySQL Clusterの適用範囲がより広範囲に
  - パッケージアプリケーション、カスタムプロジェクト
- 複雑さを軽減しつつ、強力な機能を追加
  - アプリケーションロジック & データモデル
- デフォルトで使用可能
- SQL&NoSQLの両方で使用可能
- オンラインで追加/削除可能

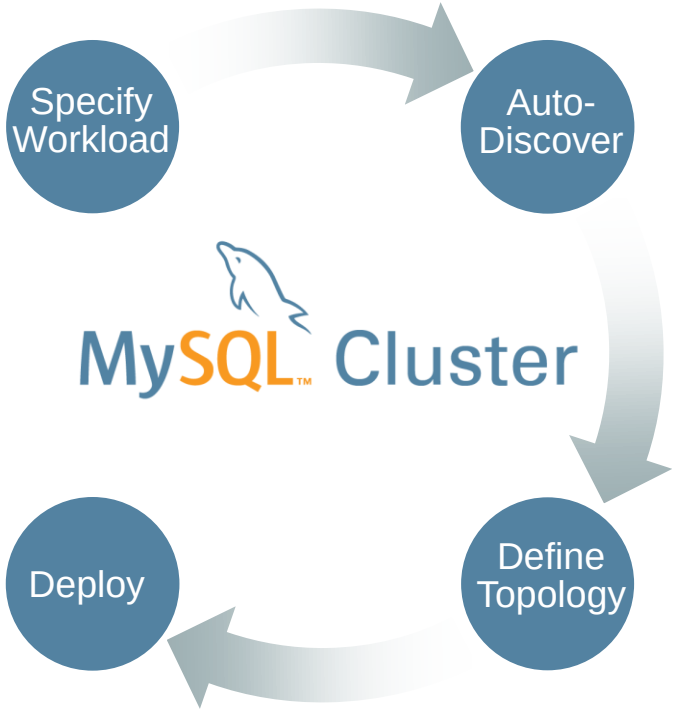


Creating & running your first Cluster by Hand



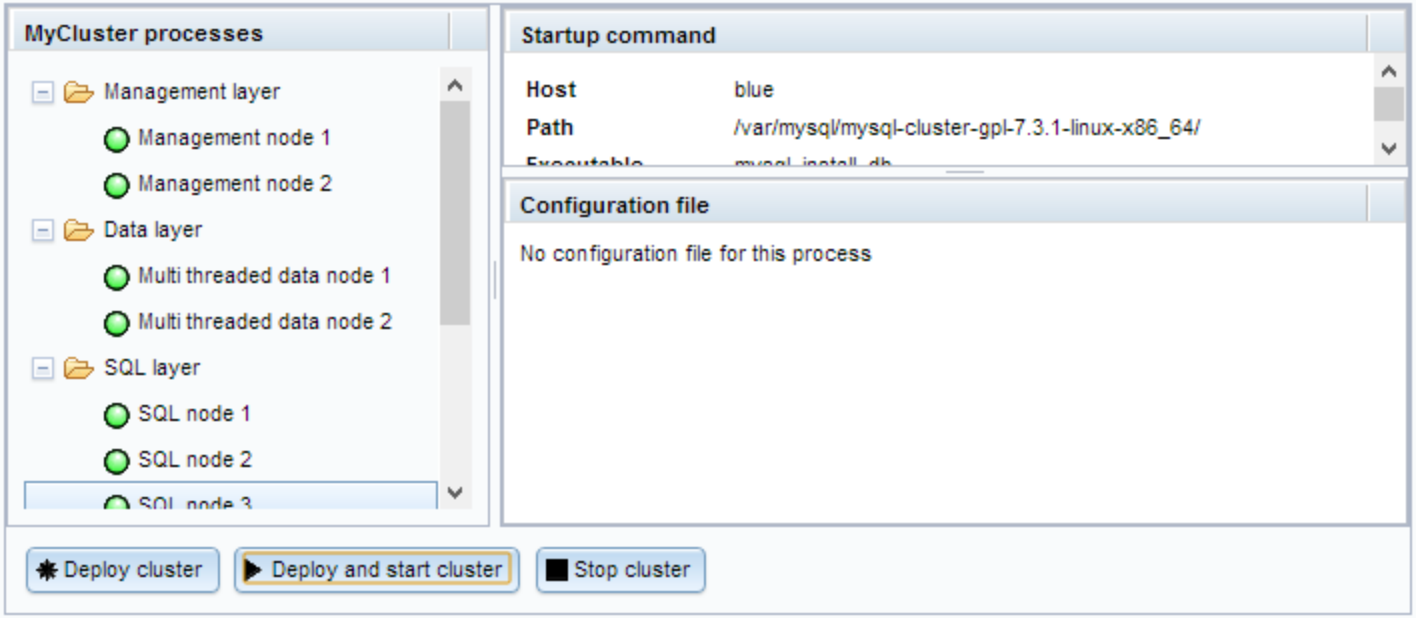
MySQL Cluster 7.3: Auto-Installer

- 素早く設定可能
- リソースを自動検出
- ワークロードに合わせた最適化
- 再現可能なベストプラクティス
- MySQL Cluster 7.2 + 7.3 で使用可能



Deploy Configuration and start MySQL Cluster

Your MySQL Cluster configuration can be reviewed below. To the left are the processes you have defined, ordered by their startup sequence. Please select a process to view its startup command(s) and configuration file. Note that some processes do not have configuration files. At the bottom of the center panel, there are buttons to *Deploy*, *Start* and *Stop* your cluster. Please note that starting the cluster may take up to several minutes depending on the configuration you have defined. In the process tree, the icons reflect the status of the process as reported by the management daemon: : *unknown* or if the management daemon does not reply, : *connected* or *started*, : *starting* or *shutting down*, and : *not connected* or *stopped*



MySQL Cluster Manager

MySQL Cluster Manager

Enhancing DevOps Agility, Reducing Downtime

Automated Management

- Start / Stop node or whole cluster
- On-Line Scaling
- On-Line Reconfiguration
- On-Line Upgrades
- On-Line Backup & Restore
- Import Running Cluster

Self-Healing

- Node monitoring
- Auto-recovery extended to SQL + mgmt nodes

HA Operations

- Cluster-wide configuration consistency
- Persistent configurations
- HA Agents

ORACLE

MySQL Cluster Managerの導入効果

MySQL Cluster 7.1から7.3へのアップグレード

MySQL Cluster Manager

MySQL Cluster Manager 導入前

- 1 x クラスタの事前状態チェック
- 8 x ssh コマンド/サーバー
- 8 x 停止コマンド/プロセス
- 4 x 構成ファイルの転送 - scp
  (2 x mgmd & 2 x mysqld)
- 8 x プロセスごとの開始コマンド
- 8 x 開始または再参加プロセスの確認
- 8 x 完了確認処理
- 1 x クラスタ全体の完了確認
- 各構成ファイルの手動による編集を除く

合計: 46 コマンド-
2.5 時間の作業

MySQL Cluster Manager 導入後

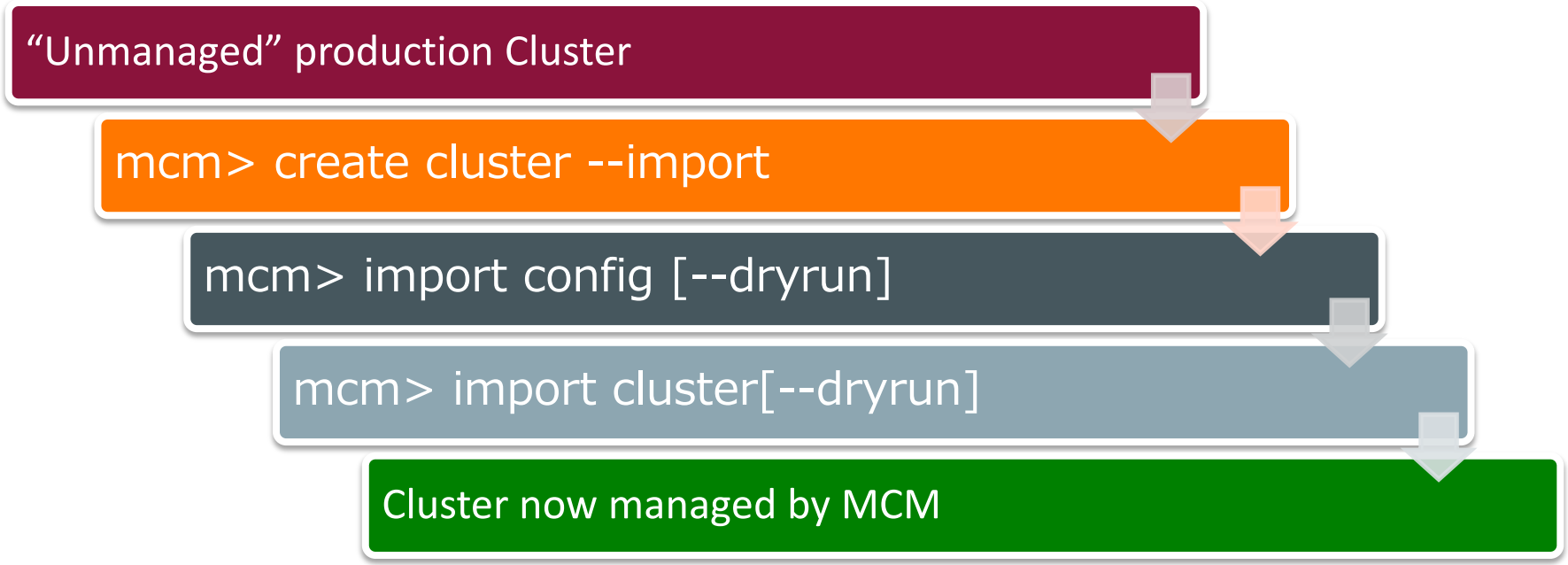
upgrade cluster --package=7.3 mycluster;

合計: 1 コマンド -
完全自動処理

ORACLE

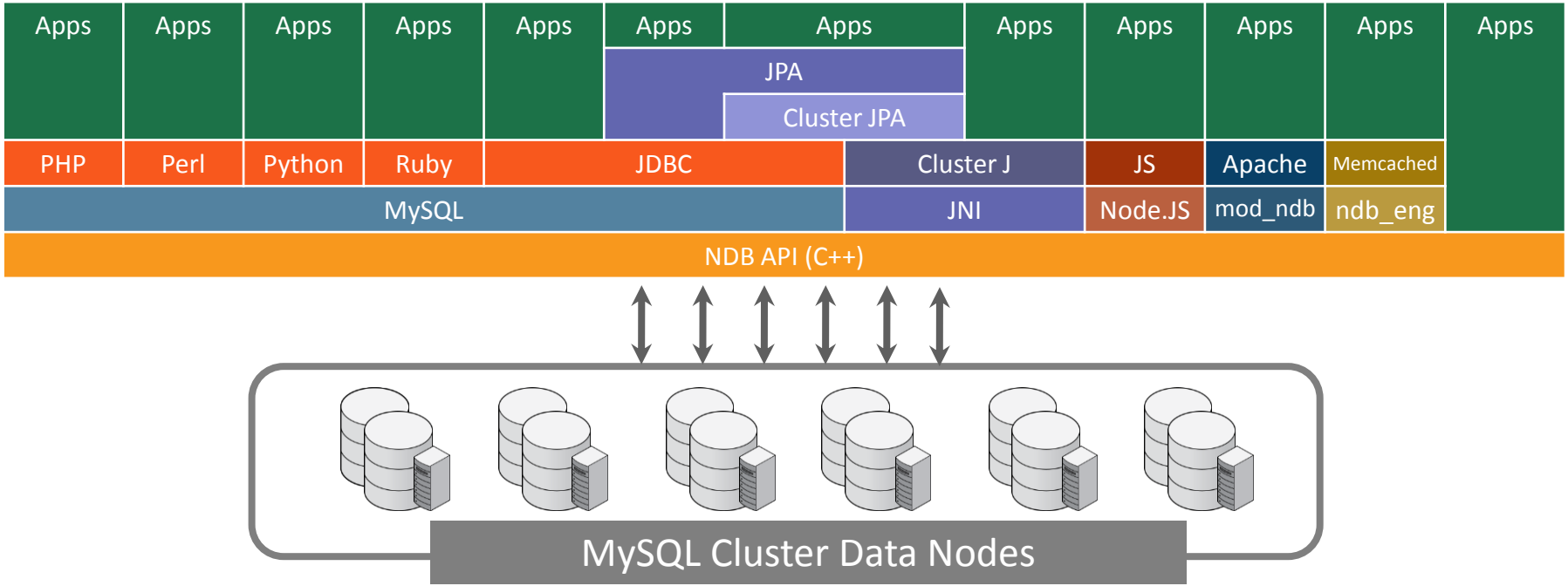
New! MySQL Cluster Manager 1.3.2 GA

Import a running Cluster into MCM




SQL
?
NoSQL

NoSQL Access to MySQL Cluster data



Cluster & Memcached – Schema-Free

Application view

key value

<town:maidenhead,SL6>

SQL view

key value

<town:maidenhead,SL6>

Key	Value
town:maidenhead	SL6

generic table

Cluster & Memcached - Configured Schema

key value

<town:maidenhead,SL6>

Application view

SQL view

prefix key value

<town:maidenhead,SL6>

Prefix	Table	Key-col	Val-col	policy
town:	map.zip	town	code	cluster

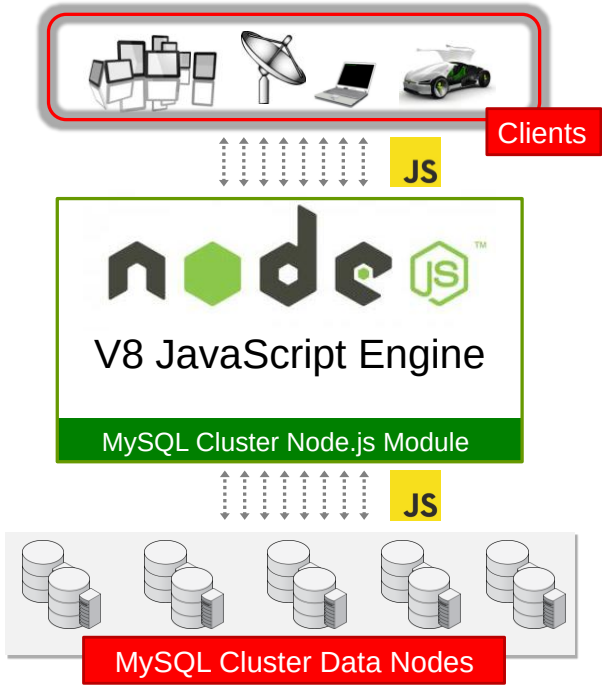
Config tables

town	...	code	...
maidenhead	...	SL6	...

map.zip

ORACLE

Node.js NoSQL API



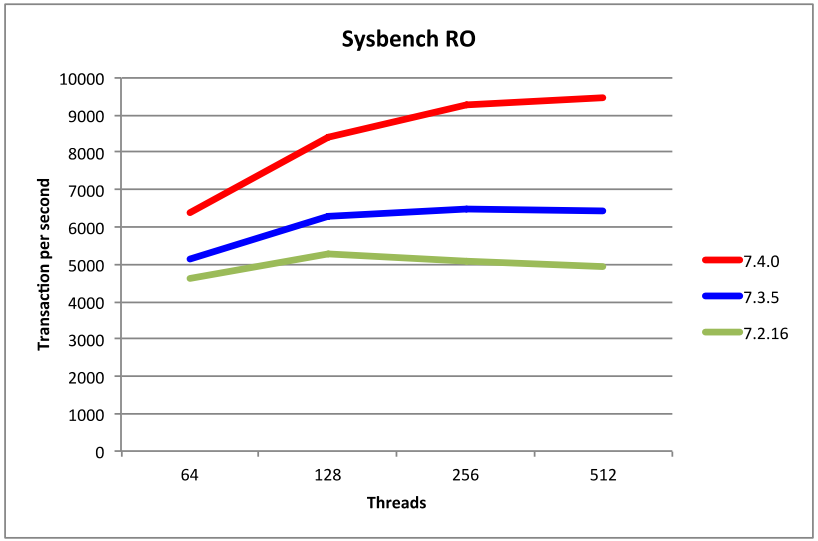
- Native JavaScript access to MySQL Cluster
  - End-to-End JavaScript: browser to the app & DB
  - Storing and retrieving JavaScript objects directly in MySQL Cluster
  - Eliminate SQL transformation
- Implemented as a module for node.js
  - Integrates Cluster API library within the web app
- Couple high performance, distributed apps, with high performance distributed database
- Optionally routes through MySQL Server

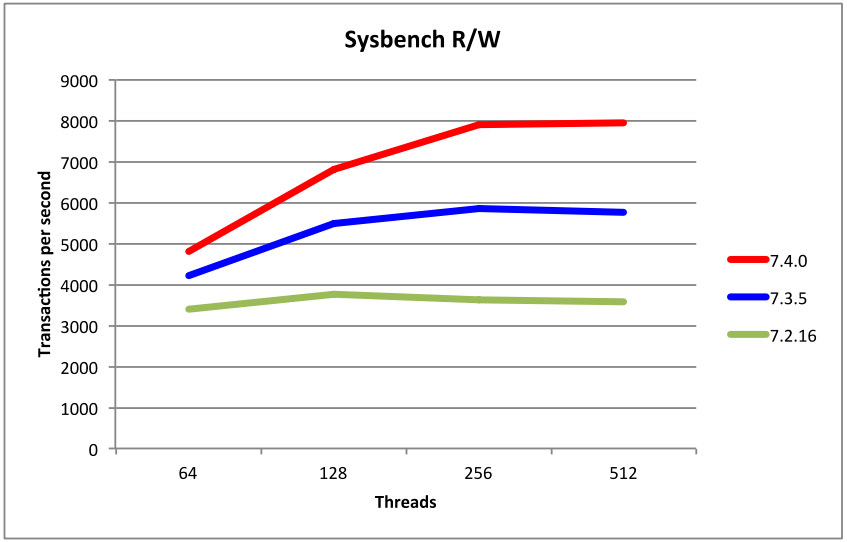
MySQL Cluster 7.4.1 DMR

Available Now!

MySQL Cluster 7.4.1 DMR

Better performance and operational simplicity





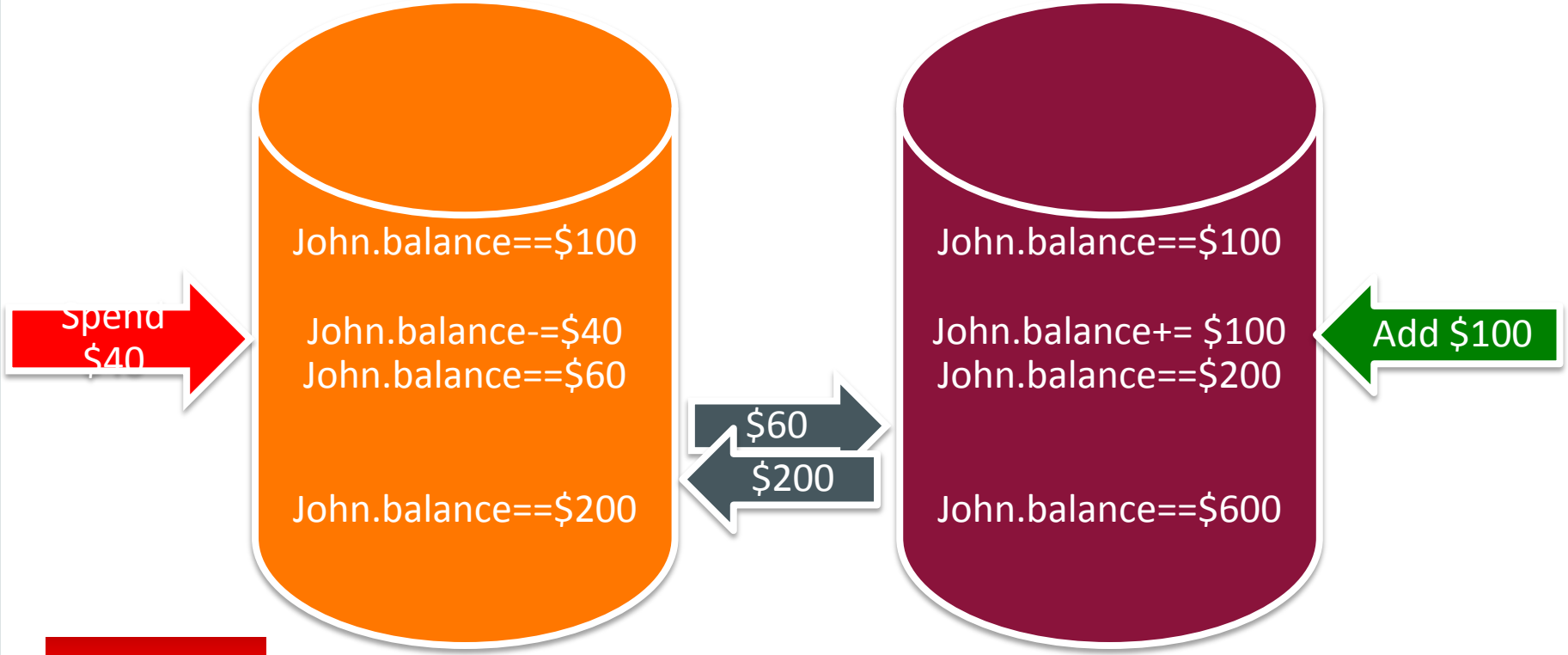
- Performance gain over 7.3
  - 47% (Read-Only)
  - 38% (Read-Write)

- Faster maintenance operations
  - Nodal & Rolling restarts
  - Upgrades

Active-Active Geo-Replication

- Asynchronous replication between MySQL Clusters
- Active-Active
  - Update anywhere
  - Conflict detection
    - Application notified through exception tables
    - Can opt to have conflicts resolved automatically
  - Auto-conflict-resolution
    - Conflicting transaction and dependent ones are rolled-back
- No changes to application schema

What is a conflict?



Handling of Conflicts

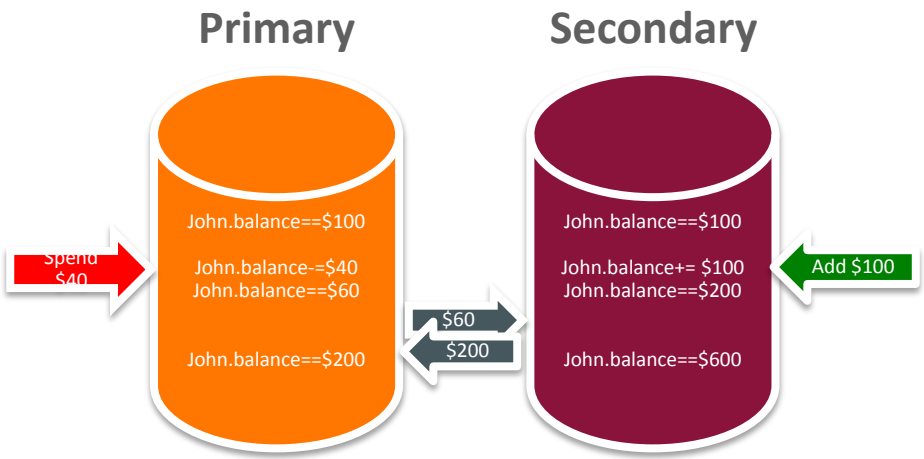
MySQL Cluster 7.4.1 DMR

- Detects conflicting inserts/updates
- Entire transactions (and dependent ones) rolled back
- All conflicts are handled before switching primary

Later in MySQL Cluster 7.4

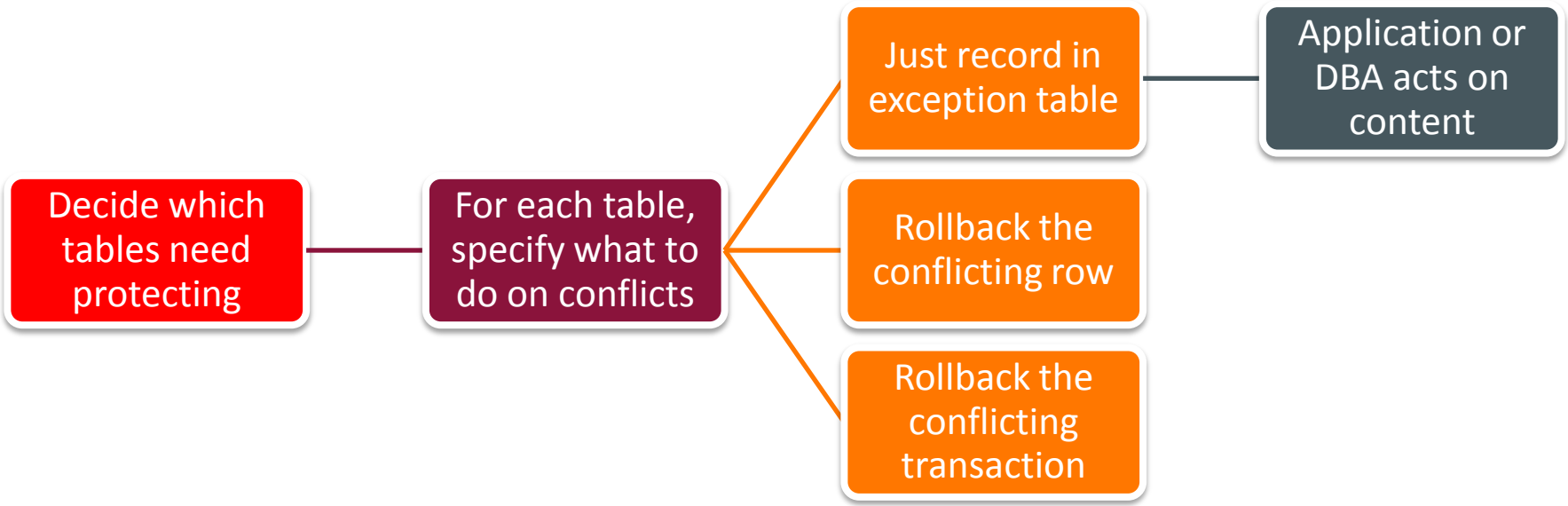
- Conflicting deletes
- Rolling back of transactions that read conflicted data

Detecting Conflicts - Reflected GCI



- Primary store logical timestamp (GCI) against updated row
  - Window for conflict opens
- GCI replicated with updated row to Secondary
- The same row and GCI is replicated back (reflected) from Secondary to Primary after it has been applied
  - Closing window for conflict
- Primary checks every event originating from the Secondary to ensure it isn't for a 'conflictable' row

How to Use Conflict Detection/Resolution



Restart Times

What operations benefit?

- Restarting data node with locally checkpointed data
  - Major improvement
- Restarting data node which must recover data from peer
  - Major improvement
  - Further speedups to come in 7.4.X (greater parallelization)
- Upgraded/rolling restarts
  - Major improvement
- Cluster shutdown and restart
  - Minor improvement

Enhanced Memory Reporting

See how much memory a table is using

```
mysql> CREATE DATABASE clusterdb;USE clusterdb;
mysql> CREATE TABLE simples (id INT NOT NULL AUTO_INCREMENT PRIMARY KEY) ENGINE=NDB;
mysql> select node_id AS node, fragment_num AS frag, fixed_elem_alloc_bytes alloc_bytes,
fixed_elem_free_bytes AS free_bytes, fixed_elem_free_rows AS spare_rows from memory_per_fragment
where fq_name like '%simples%';
+------+------+-------------+-----------+-----------+
| node | frag | alloc_bytes | free_bytes | spare_rows |
+------+------+-------------+-----------+-----------+
|    1 |    0 |      131072 |      5504 |       172 |
|    1 |    2 |      131072 |      1280 |        40 |
|    2 |    0 |      131072 |      5504 |       172 |
|    2 |    2 |      131072 |      1280 |        40 |
|    3 |    1 |      131072 |      3104 |        97 |
|    3 |    3 |      131072 |      4256 |       133 |
|    4 |    1 |      131072 |      3104 |        97 |
|    4 |    3 |      131072 |      4256 |       133 |
+------+------+-------------+-----------+-----------+
```

Oracle MySQL HA & Scaling Solutions

	MySQL Replication	MySQL Fabric	Oracle VM Template	Oracle Clusterware	Solaris Cluster	Windows Cluster	DRBD	MySQL Cluster
App Auto-Failover	✖	✔	✔	✔	✔	✔	✔	✔
Data Layer Auto-Failover	✖	✔	✔	✔	✔	✔	✔	✔
Zero Data Loss	MySQL 5.7	MySQL 5.7	✔	✔	✔	✔	✔	✔
Platform Support	All	All	Linux	Linux	Solaris	Windows	Linux	All
Clustering Mode	Master + Slaves	Master + Slaves	Active/Passive	Active/Passive	Active/ Passive	Active/Passive	Active/Passive	Multi-Master
Failover Time	N/A	Secs	Secs +	Secs +	Secs +	Secs +	Secs +	< 1 Sec
Scale-out	Reads	✔	✖	✖	✖	✖	✖	✔
Cross-shard operations	N/A	✖	N/A	N/A	N/A	N/A	N/A	✔
Transparent routing	✖	For HA	✔	✔	✔	✔	✔	✔
Shared Nothing	✔	✔	✖	✖	✖	✖	✔	✔
Storage Engine	InnoDB+	InnoDB+	InnoDB+	InnoDB+	InnoDB+	InnoDB+	InnoDB+	NDB
Single Vendor Support	✔	✔	✔	✔	✔	✖	✔	✔

When to Consider MySQL Cluster

MySQL Cluster

- Scalability demands
  - Sharding for write performance?
- Latency demands
  - Cost of each millisecond?
- Uptime requirements
  - Cost per minute of downtime?
  - Failure versus maintenance?
- Application agility
  - Developer languages and frameworks?
  - SQL or NoSQL?

ORACLE

Next Steps

Learn More

- www.mysql.com/cluster
- Authentic MySQL Curriculum: http://oracle.com/education/mysql

Try it Out

- dev.mysql.com/downloads/cluster/

Let us know what you think

- bugs.mysql.com
- forums.mysql.com/list.php?25

ORACLE


ORACLE®