
ORACLE®

ORACLE®

MySQL Cluster for Real Time, HA Services

Bill Papp (bill.papp@oracle.com)
Principal MySQL Sales Consultant
Oracle

Agenda

- Overview of MySQL Cluster
  - Design Goals, Evolution, Workloads, Users
  - Architecture and Core Technology
- Deep Dive, New Features & Capabilities
  - MySQL Cluster 7.1
  - MySQL Cluster Manager
- Resources to Get Started

MySQL Cluster

MySQL Cluster Goals

- High Performance: Write Scalability & Low Latency
- 99.999% Availability
- Low TCO

MySQL Cluster - Key Advantages

High Throughput Reads & Writes	Distributed, Parallel architecture Transactional, ACID-compliant relational database
Carrier-Grade Availability	Shared-nothing design, synchronous data replication Sub-second failover & self-healing recovery
Real-Time Responsiveness	Data structures optimized for RAM. Real-time extensions Predictable low latency, bounded access times
On-Line, Linear Scalability	Incrementally scale out, scale up and scale on-line Linearly scale with distribution awareness
Low TCO, Open platform	GPL & Commercial editions, scale on COTS Flexible APIs: SQL, C++, Java, OpenJPA, LDAP & HTTP

MySQL Cluster Highlights

- *Distributed Hash Table backed by an ACID Relational Model*
- *Shared-Nothing Architecture, scale-out on commodity hardware*
- *Implemented as a pluggable storage engine for the MySQL Server with additional direct access via embedded APIs.*
- *Automatic or user configurable data partitioning across nodes*
- *Synchronous data redundancy*

MySQL Cluster Highlights (cont.)

- *Sub-second fail-over & self-healing recovery*
- *Geographic replication*
- *Data stored in main-memory or on disk (configurable per-column)*
- *Logging and check pointing of in-memory data to disk*
- *Online operations (i.e. add-nodes, schema updates, maintenance, etc)*

MySQL Cluster – Users & Applications

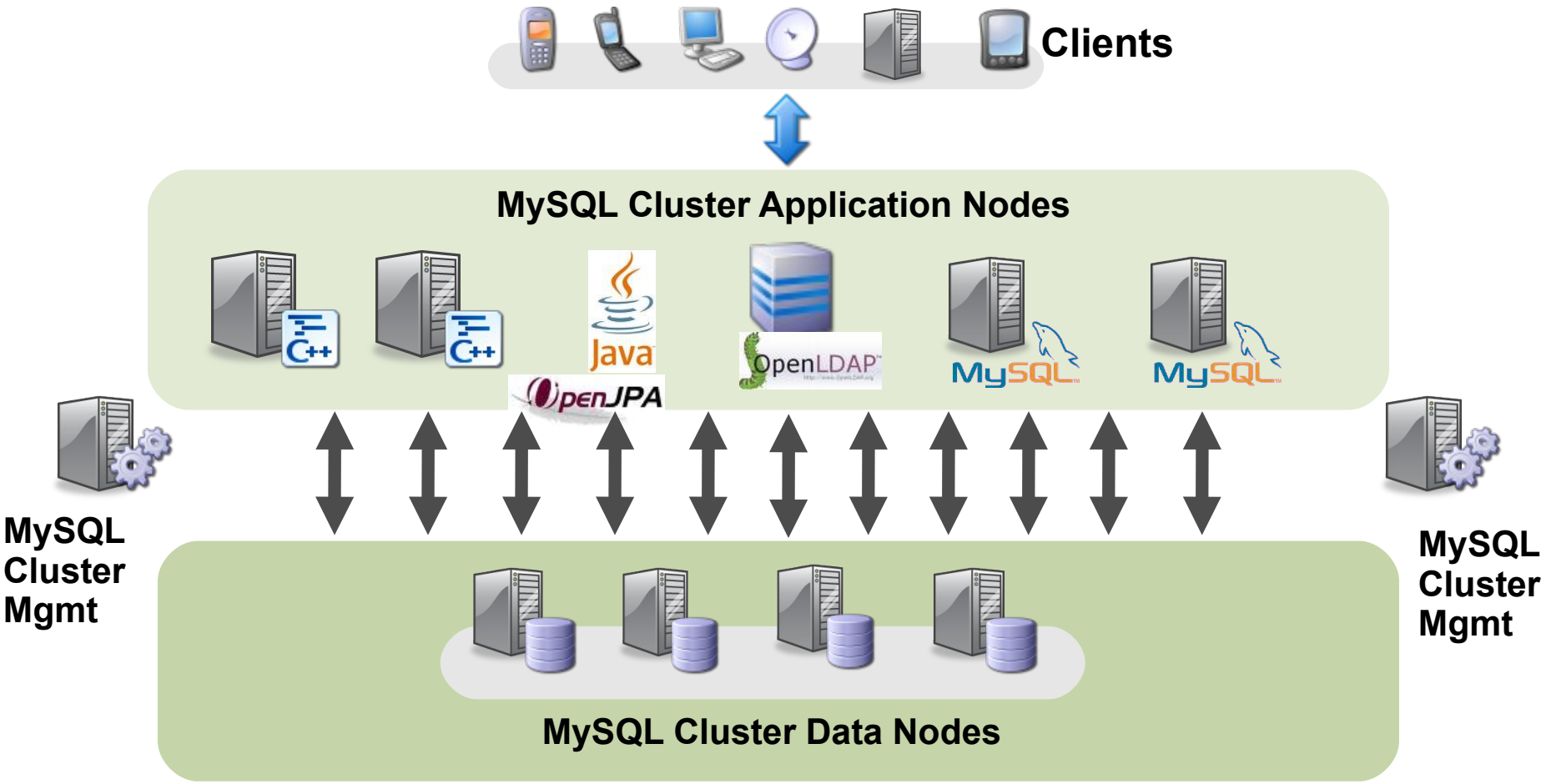
HA, Transactional Services: Web & Telecoms

- Telecoms
  - Subscriber Databases (HLR/HSS)
  - Service Delivery Platforms
  - VoIP, IPTV & VoD
  - Mobile Content Delivery
  - On-Line app stores and portals
  - IP Management
  - Payment Gateways
- Web
  - User profile management
  - Session stores
  - eCommerce
  - On-Line Gaming
  - Application Servers

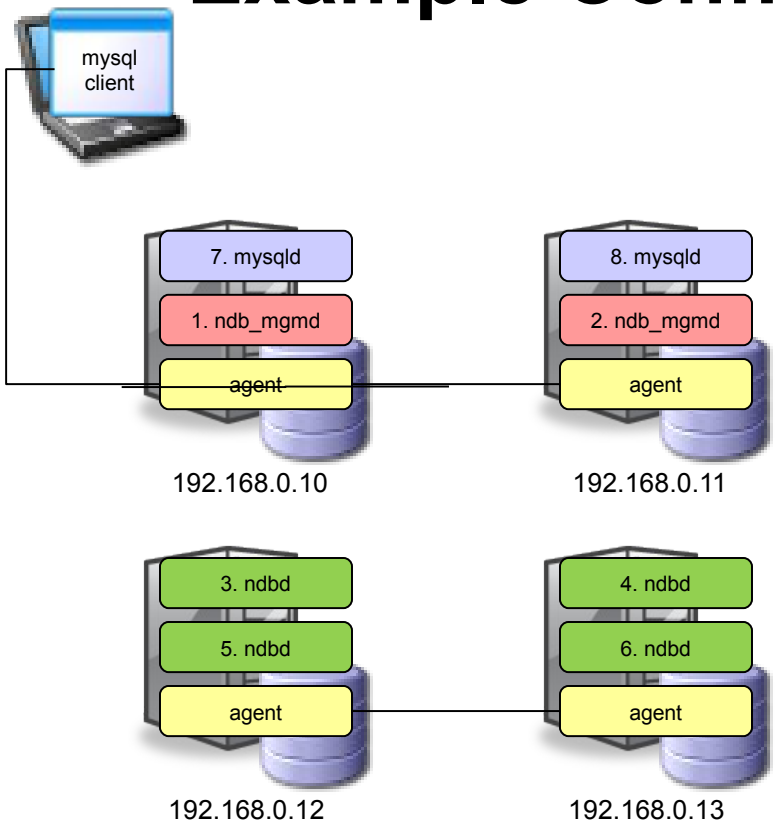
http://www.mysql.com/customers/cluster/

MySQL Cluster Architecture

Parallel Database with no SPOF: High Read & Write Performance & 99.999% uptime



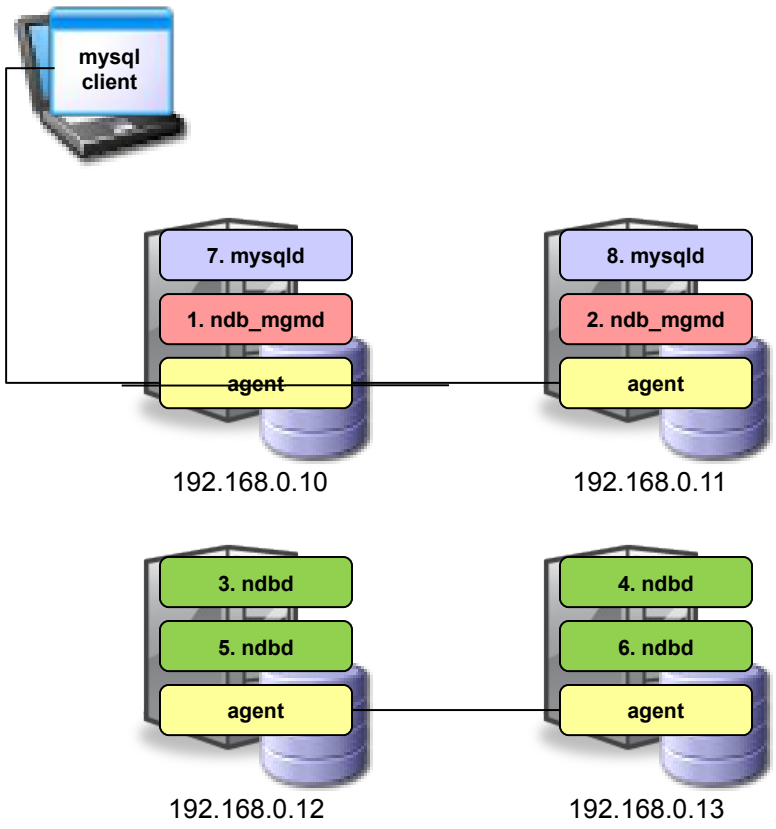
Example Configuration



n. mysqld	MySQL Server (ID=n)
n. ndb_mgmd	Management Node (ID=n)
n. ndbd	Data Node (ID=n)
agent	MySQL Cluster Manager agent

- *MySQL Cluster Manager agent runs on each physical host*
- *No central process for Cluster Manager – agents co-operate, each one responsible for its local nodes*
- *Agents are responsible for managing all nodes in the cluster*
- *Management responsibilities*
  - *Starting, stopping & restarting nodes*
  - *Configuration changes*
  - *Upgrades*
  - *Host & Node status reporting*
  - *Recovering failed nodes*

Creating & Starting a Cluster



1. Define the site:

```
Mysql> create site --hosts=192.168.0.10,192.168.0.11,
-> 192.168.0.12,192.168.0.13 mysite;
```

2. Expand the MySQL Cluster tar-ball(s) from mysql.com to known directory
3. Define the package(s):

```
Mysql> add package --basedir=/usr/local/mysql_6_3_26 6.3;
Mysql> add package --basedir=/usr/local/mysql_7_0_7 7.0;
```

Note that the basedir should match the directory used in Step 2.

4. Create the Cluster

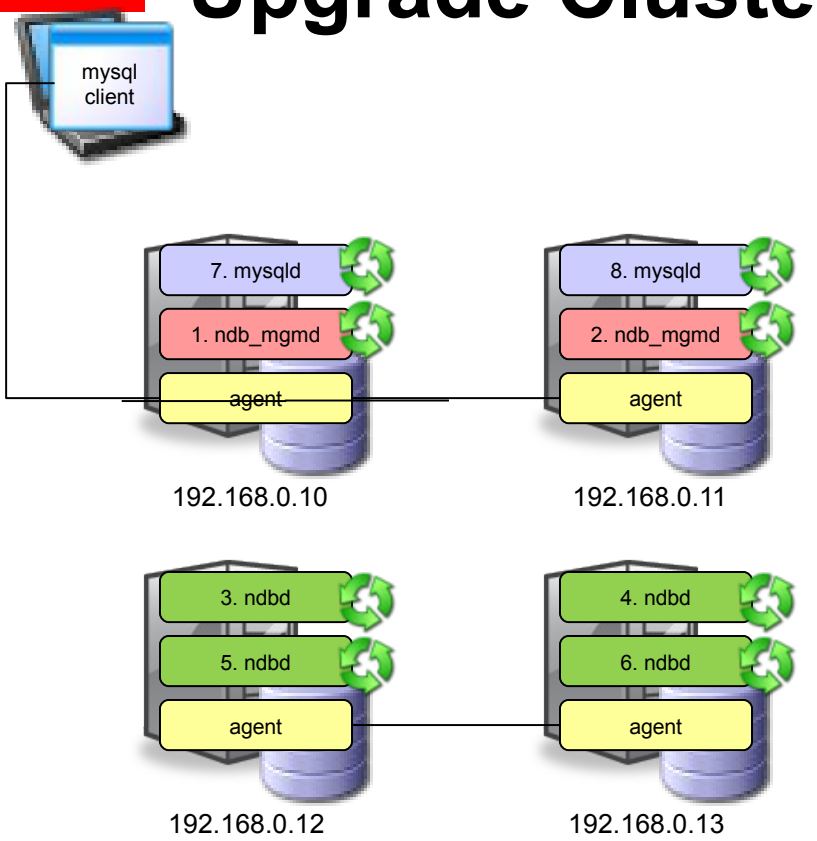
```
Mysql> create cluster --package=6.3
-> --processhosts=ndb_mgmd@192.168.0.10,ndb_mgmd@192.168.0.11,
-> ndbd@192.168.0.12,ndbd@192.168.0.13, ndbd@192.168.0.12,
-> ndbd@192.168.0.13,mysqld@192.168.9.10,mysqld@192.168.9.11
-> mycluster;
```

This is where you define what nodes/processes make up the Cluster and where they should run

5. Start the Cluster:

```
Mysql> start cluster mycluster;
```

Upgrade Cluster

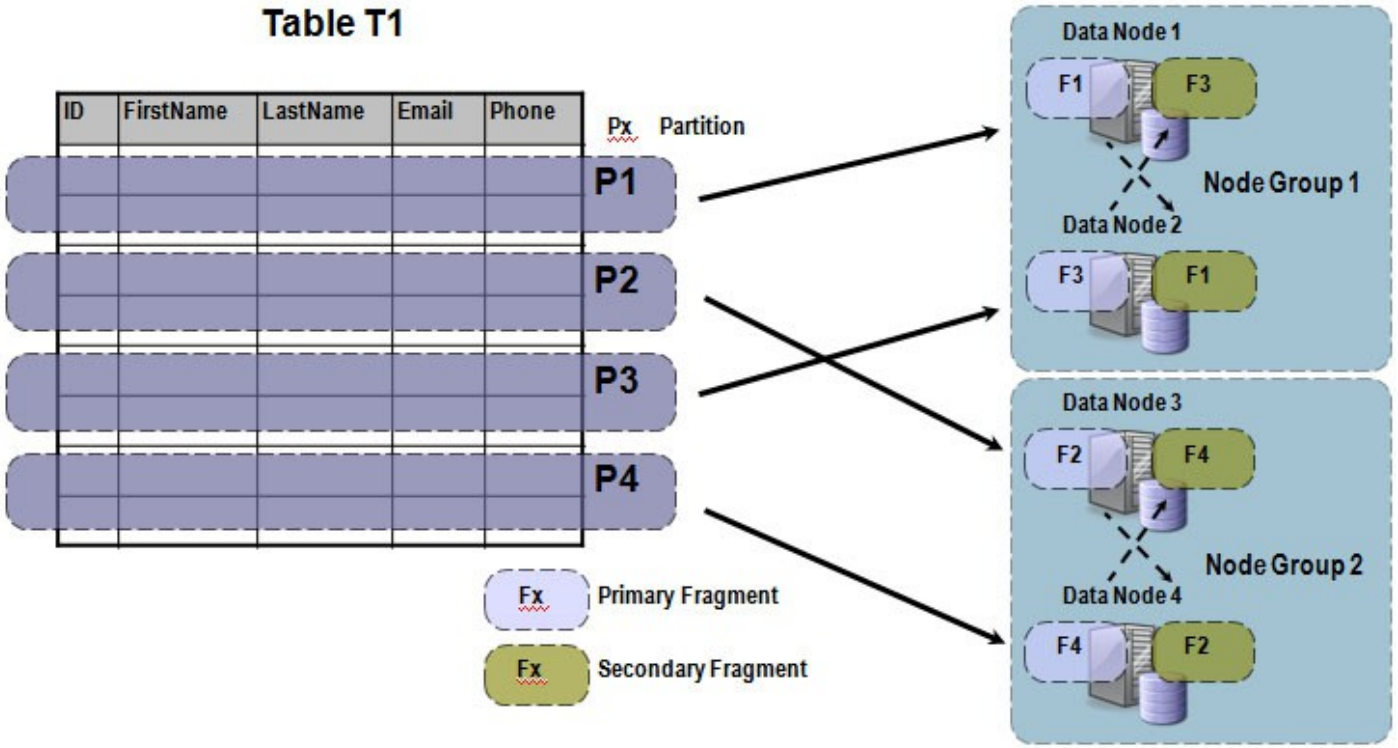


- Upgrade from MySQL Cluster 6.3.26 to 7.0.7:

```
mysql> upgrade cluster --package=7.0 mycluster;
```

- Automatically upgrades each node and restarts the process – in the correct order to avoid any loss of service
- Without MySQL Cluster Manager, the administrator must stop each process in turn, start the process with the new version and wait for the node to restart before moving onto the next one

Out of the Box Scalability: Data Partitioning

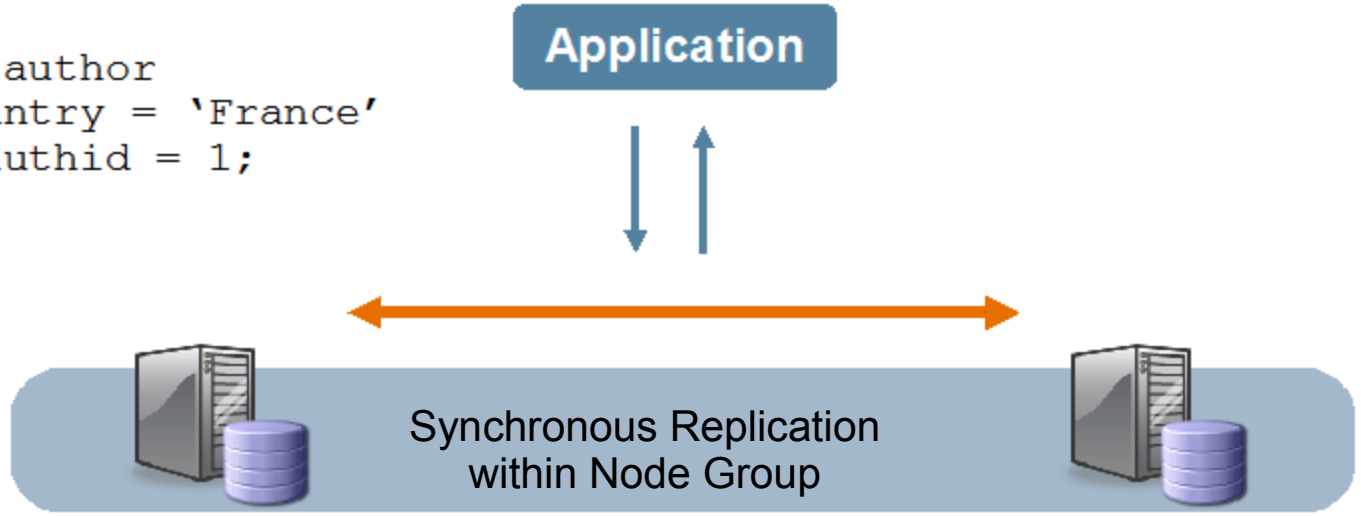


- Data partitioned across Data Nodes
- Rows are divided into partitions, based on a hash of all or part of the primary key
- Each Data Node holds primary fragment for 1 partition
  - Also stores secondary fragment of another partition
- Records larger than 8KB stored as BLOBs

Shared-Nothing Architecture for High Availability

Update:

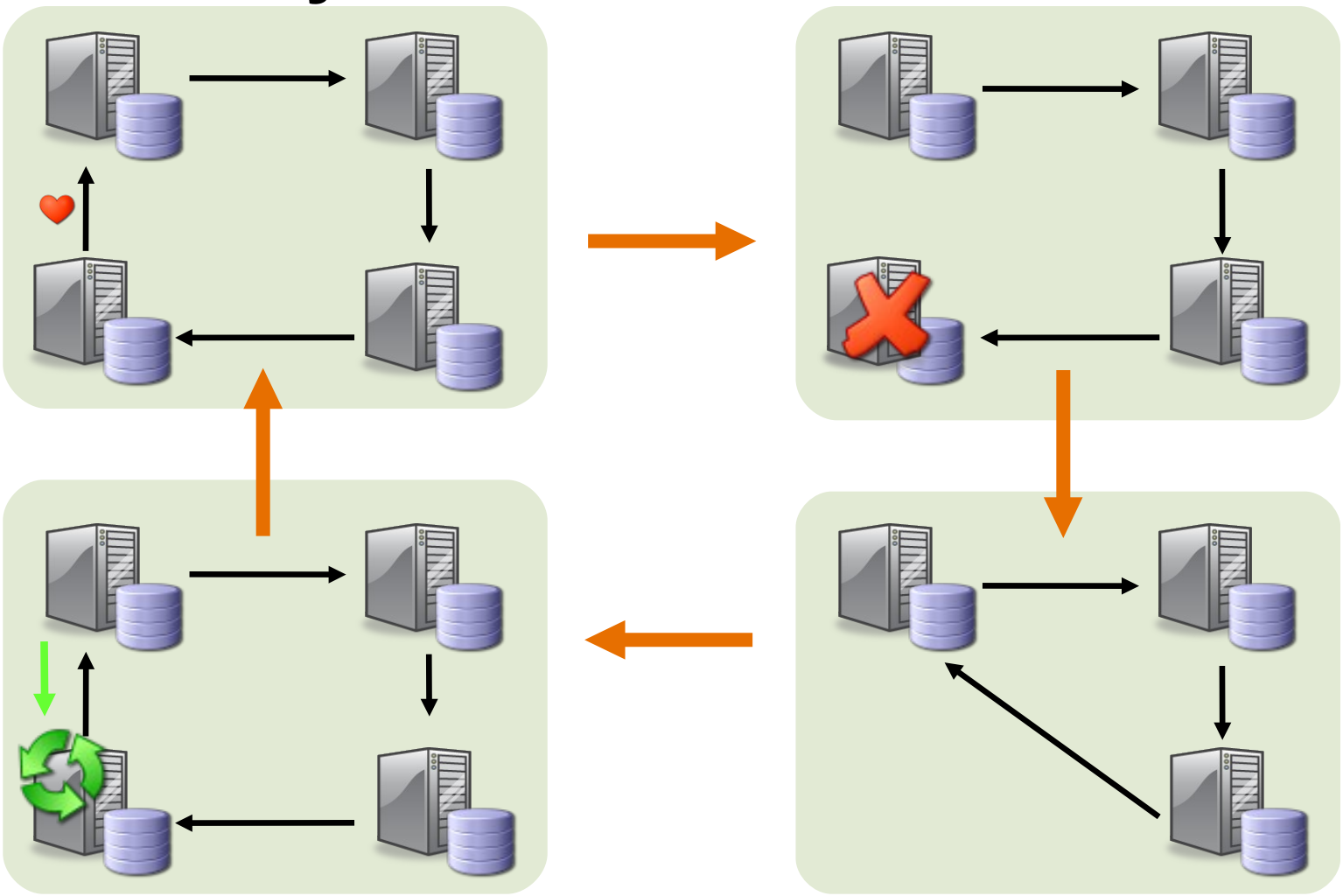
```
UPDATE author
SET country = 'France'
WHERE authid = 1;
```



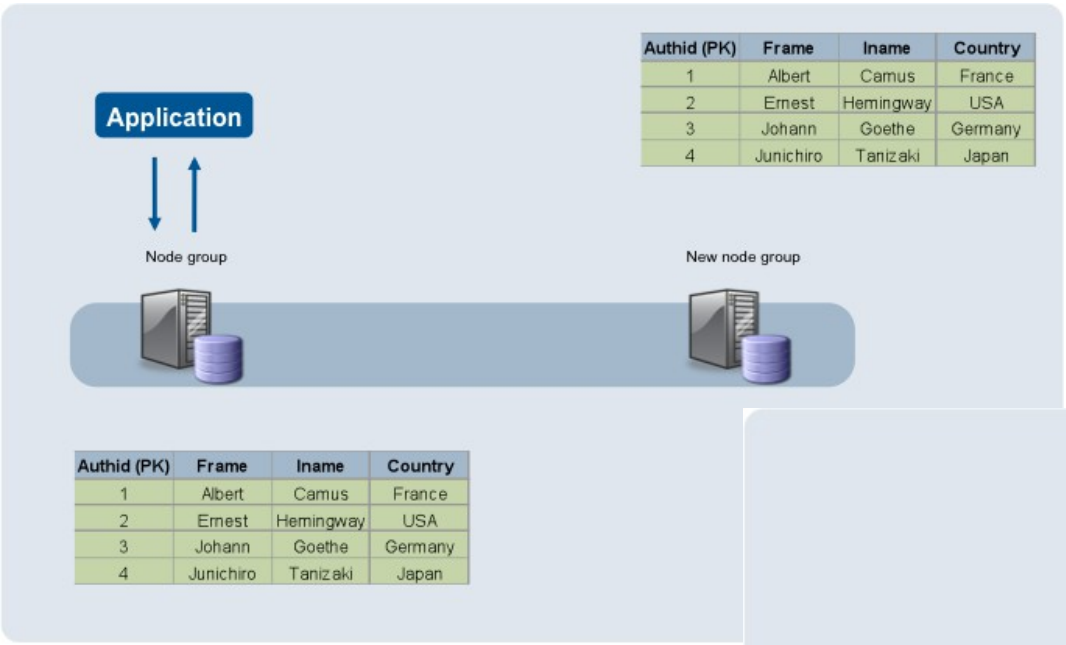
Authid (PK)	Frame	lname	Country
1	Albert	Camus	France
2	Ernest	Hemingway	Cuba
3	Johan	Goethe	Germany
4	Junichiro	Tanizaki	Japan

Authid (PK)	Frame	lname	Country
1	Albert	Camus	France
2	Ernest	Hemingway	Cuba
3	Johan	Goethe	Germany
4	Junichiro	Tanizaki	Japan

Node Failure Detection & Self-Healing Recovery



On-Line Scaling & Maintenance

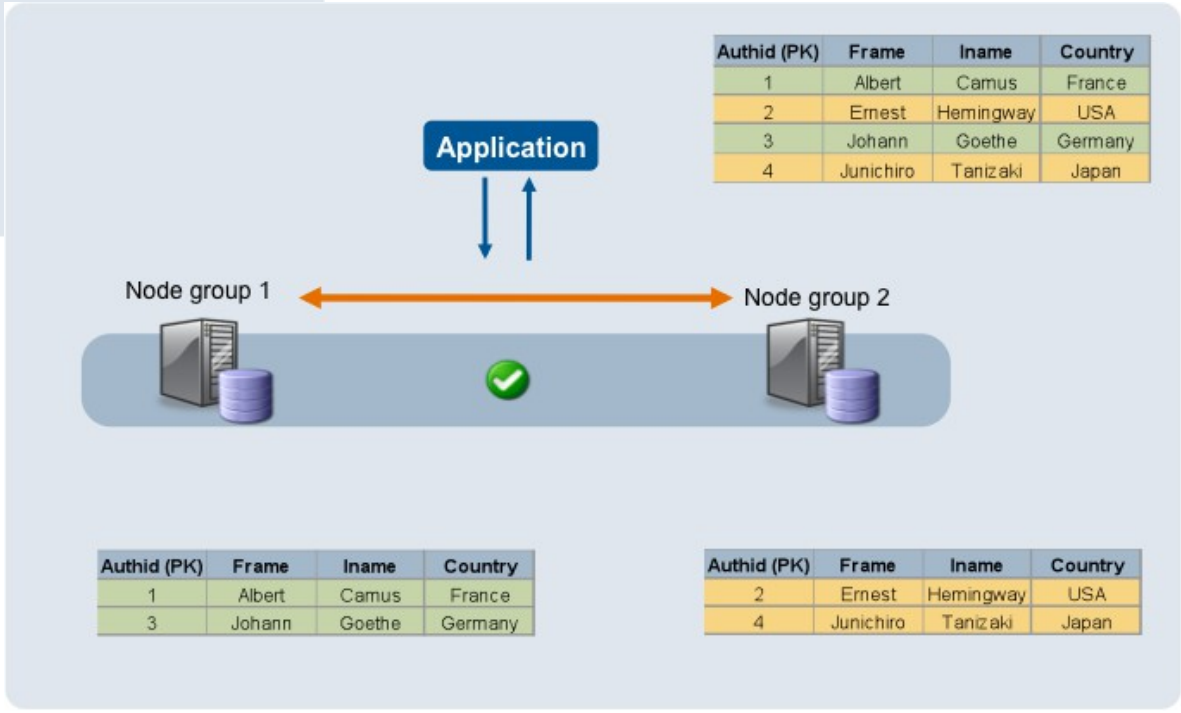


Authid (PK)	Frame	Iname	Country
1	Albert	Camus	France
2	Ernest	Hemingway	USA
3	Johann	Goethe	Germany
4	Junichiro	Tanizaki	Japan

Authid (PK)	Frame	Iname	Country
1	Albert	Camus	France
2	Ernest	Hemingway	USA
3	Johann	Goethe	Germany
4	Junichiro	Tanizaki	Japan

1. New node group added
2. Data is re-partitioned
3. Redundant data is deleted
4. Distribution is switched to share load with new node group

- Can also update schema on-line
- Upgrade hardware & software with no downtime
- Perform back-ups on-line

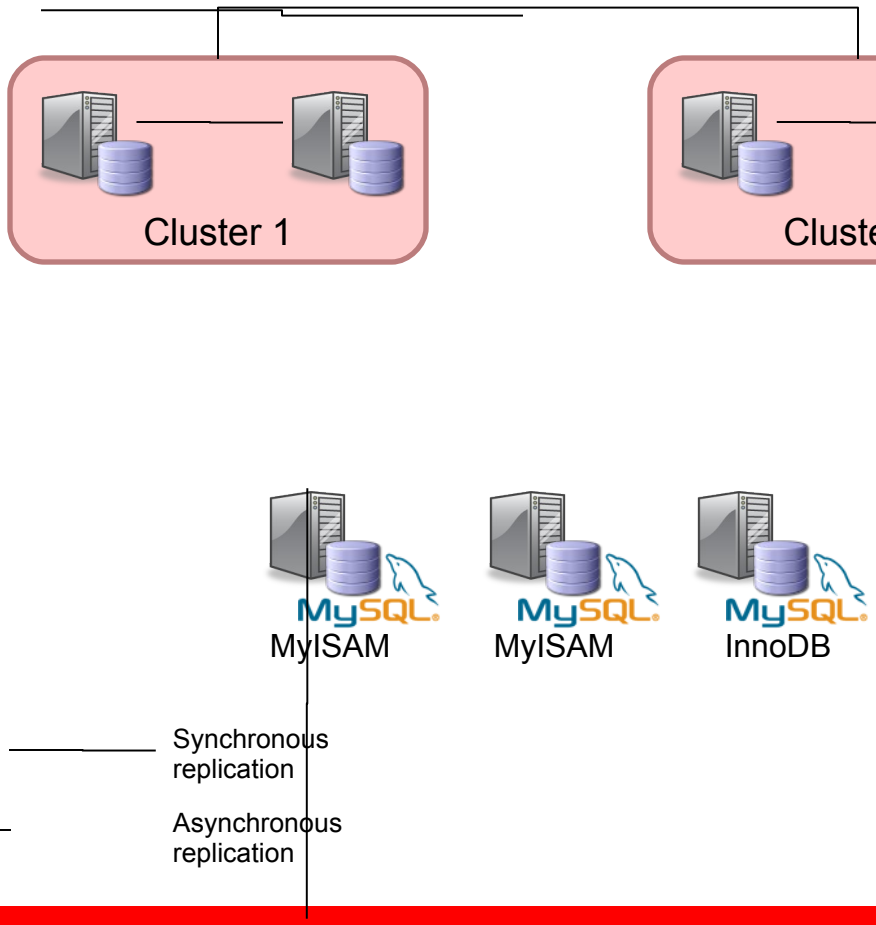


Authid (PK)	Frame	Iname	Country
1	Albert	Camus	France
2	Ernest	Hemingway	USA
3	Johann	Goethe	Germany
4	Junichiro	Tanizaki	Japan

Authid (PK)	Frame	Iname	Country
1	Albert	Camus	France
3	Johann	Goethe	Germany

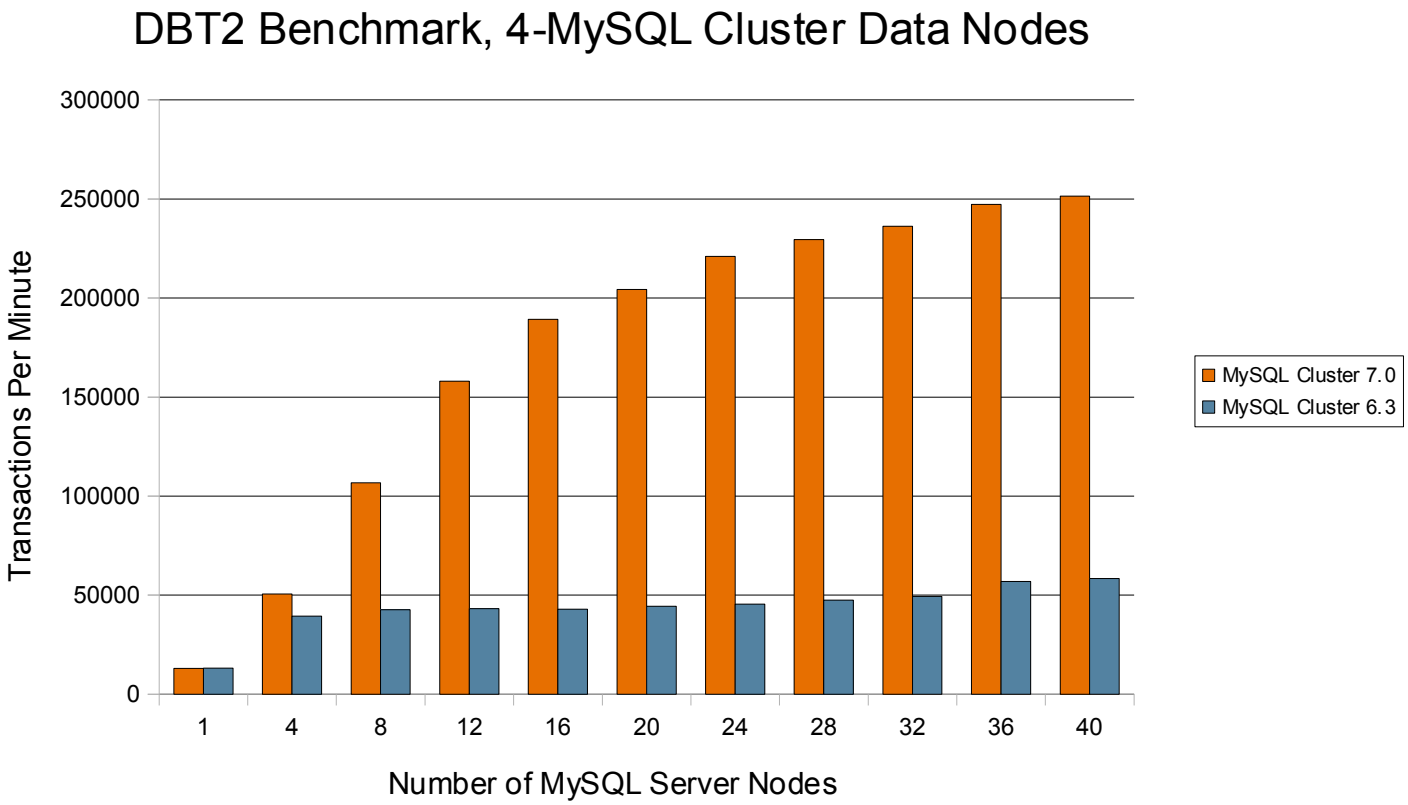
Authid (PK)	Frame	Iname	Country
2	Ernest	Hemingway	USA
4	Junichiro	Tanizaki	Japan

Geographic Replication



- ***Synchronous replication within a Cluster node group for HA***
- ***Bi-Direction asynchronous replication to remote Cluster for geographic redundancy***
- ***Asynchronous replication to non-Cluster databases for specialised activities such as report generation***
- ***Mix and match replication types***

High Throughput, Low Latency Transactional Performance

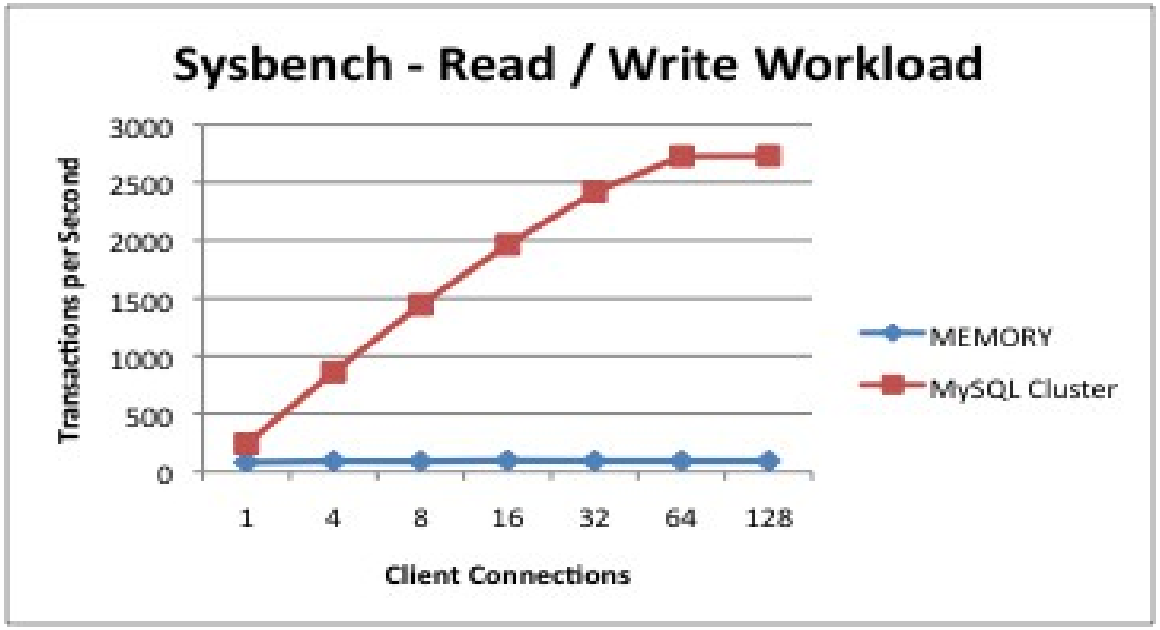


http://www.mysql.com/why-mysql/benchmarks/mysql-cluster/

- MySQL Cluster delivered:
  - 250k TPM, 125k operations per second
  - Average 3ms response time
  - 4.3x higher throughput than previous MySQL Cluster 6.3 release

MySQL Cluster vs MySQL MEMORY:

30x Higher Throughput / 1/3rd the Latency on a single node



- *Table level locking inhibits MEMORY scalability beyond a single client connection*
- *Check-pointing & logging enabled, MySQL Cluster still delivers durability*
- *4 socket server, 64GB RAM, running Linux*

MySQL Cluster CGE 7.1 – Key Enhancements

Reducing Cost of Operations

Simplified Management & Monitoring:

NDBINFO

MySQL Cluster Manager (part of CGE only)

Faster Restarts

Delivering up to 10x higher Java Throughput

MySQL Cluster Connector for Java:

Native Java API

OpenJPA Plug-In

Real-Time Metrics w/ ndbinfo

```
mysql> use ndbinfo
mysql> show tables;
+--------------------+
| Tables_in_ndbinfo  |
+--------------------+
| blocks             |
| config_params      |
| counters           |
| logbuffers         |
| logspaces          |
| memoryusage        |
| nodes              |
| resources          |
| transporters       |
+--------------------+
```

- ***New database (ndbinfo) which presents real-time metric data in the form of tables***
- ***Exposes new information together with providing a simpler, more consistent way to access existing data***
- ***Examples include:***
  - ***Resource usage (memory, buffers)***
  - ***Event counters (such as number of READ operations since last restart)***
  - ***Data node status and connection status***

Real-Time Metrics w/ ndbinfo (cont.)

- Example 1: Check memory usage/availability

```
mysql> select * from memoryusage;
+---------+--------------+------+------+
| node_id | DATA_MEMORY  | used | max  |
+---------+--------------+------+------+
|       3 | DATA_MEMORY  |  594 | 2560 |
|       4 | DATA_MEMORY  |  594 | 2560 |
|       3 | INDEX_MEMORY |  124 | 2336 |
|       4 | INDEX_MEMORY |  124 | 2336 |
+---------+--------------+------+------+
```

- ***Note that there is a DATA_MEMORY and INDEX_MEMORY row for each data node in the cluster***
- ***If the Cluster is nearing the configured limit then increase the DataMemory and/or IndexMemory parameters in config.ini and then perform a rolling restart***

Real-Time Metrics w/ ndbinfo (cont.)

- Example 2: ***Check how many table scans performed on each data node since the last restart***

```
mysql> select node_id as 'data node', val as 'Table Scans'
  from counters where counter_name='TABLE_SCANS';
+-----------+-------------+
  | data node | Table Scans |
  +-----------+-------------+
  |         3 |           3 |
  |         4 |           4 |
  +-----------+-------------+
```

- You might check this if your database performance is lower than anticipated
- If this figure is rising faster than you expected then examine your application to understand why there are so many table scans

MySQL Cluster 7.1: ndbinfo

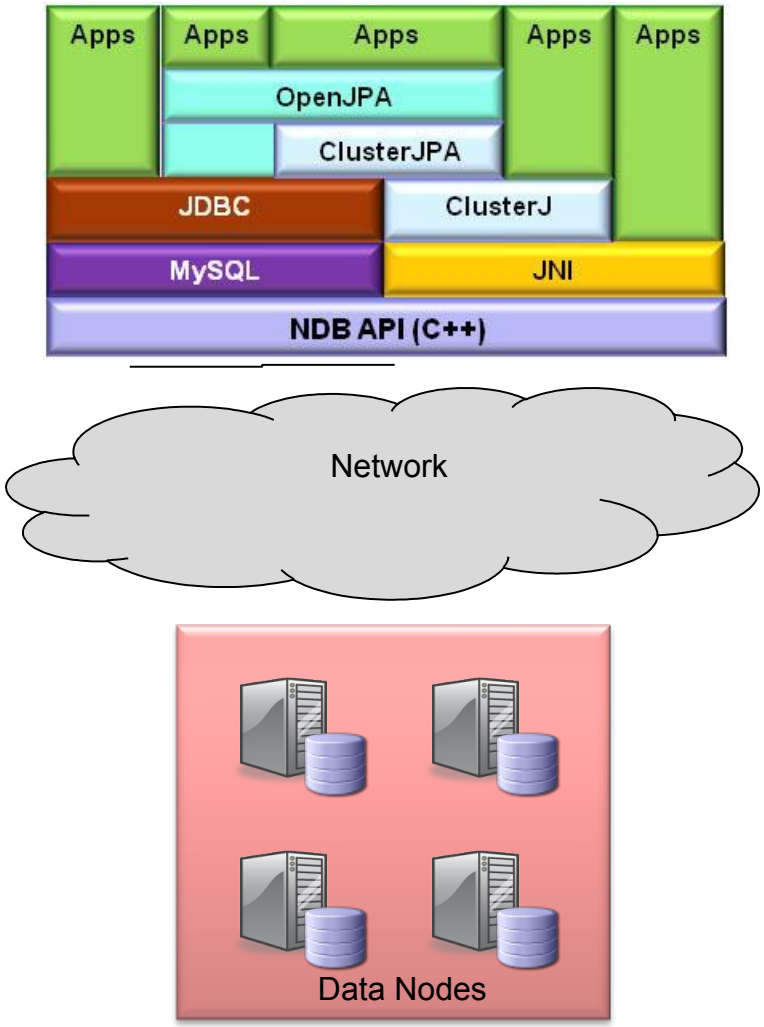
- Example 3: ***Check if approaching the point at which the undo log completely fills up between local checkpoints (which could result in delayed transactions or even a database halt if not addressed):***

```
mysql> select node_id as 'data node', total as 'configured undo log
  buffer size', used as 'used buffer space' from logbuffers where
  log_type='DD-UNDO‘;

  +-----------+-----------------------------------+-------------------+
  | data node | configured undo log buffer size | used buffer space |
  +-----------+-----------------------------------+-------------------+
  |         3 |                           2096128 |                 0 |
  |         4 |                           2096128 |                 0 |
  +-----------+-----------------------------------+-------------------+
```

- If log buffer is almost full then increase size of log buffer

MySQL Cluster Connector for Java

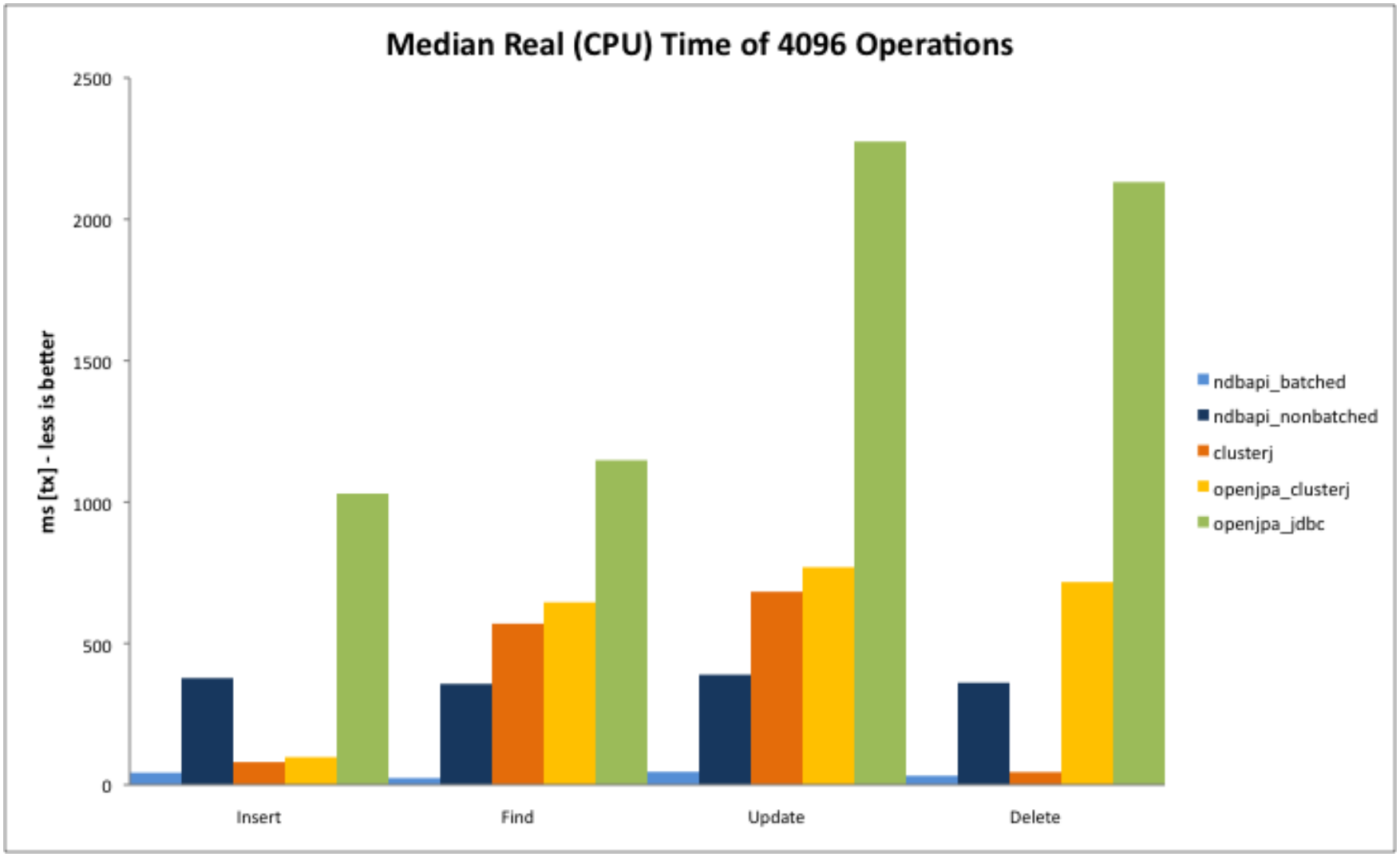


- *New Domain Object Model Persistence API (ClusterJ) :*
  - *Java API*
  - *High performance, low latency*
  - *Feature rich*
- *JPA interface built upon this new Java layer:*
  - *Java Persistence API compliant*
    - *Implemented as an OpenJPA plugin*
  - *Uses ClusterJ where possible, reverts to JDBC for some operations*
  - *Higher performance than JDBC*
  - *More natural for most Java designers*
  - *Easier Cluster adoption for web applications*

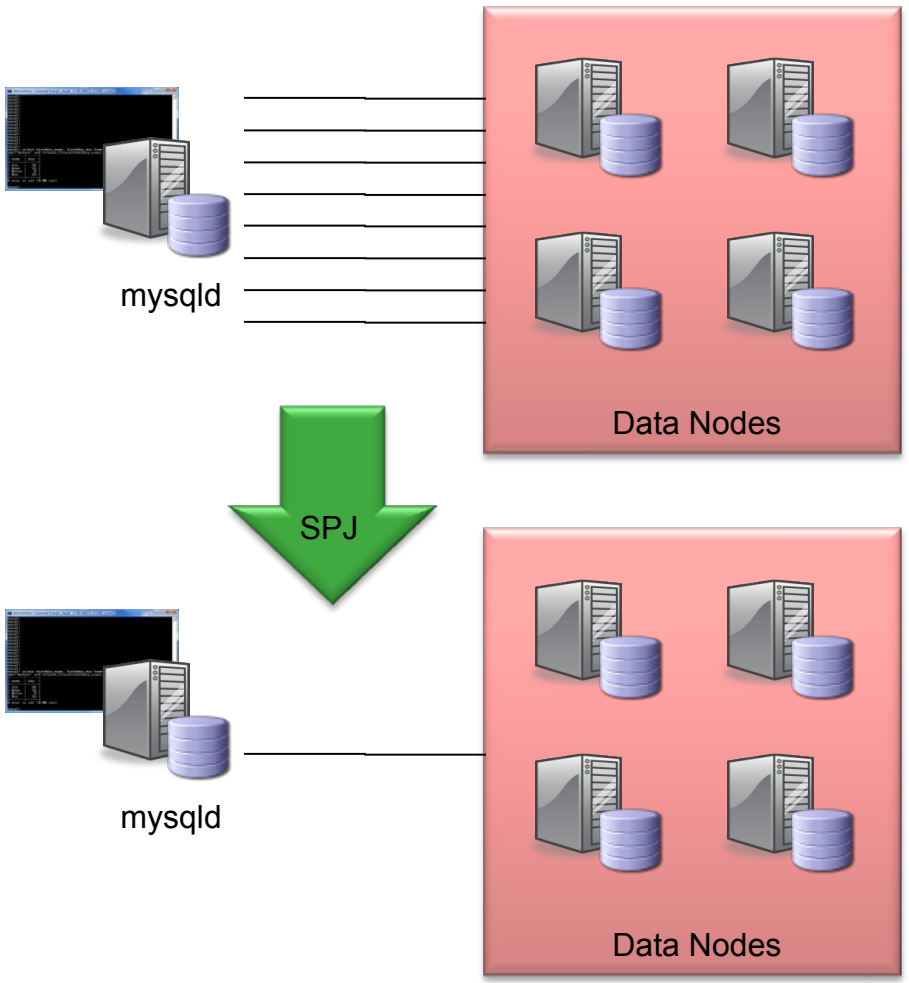
ClusterJPA

- Removes ClusterJ limitations:
  - Persistent classes
  - Relationships
  - Joins in queries
  - Lazy loading
  - Table and index creation from object model
- Implemented as an OpenJPA plugin
- Better JPA performance for insert, update, delete

Performance

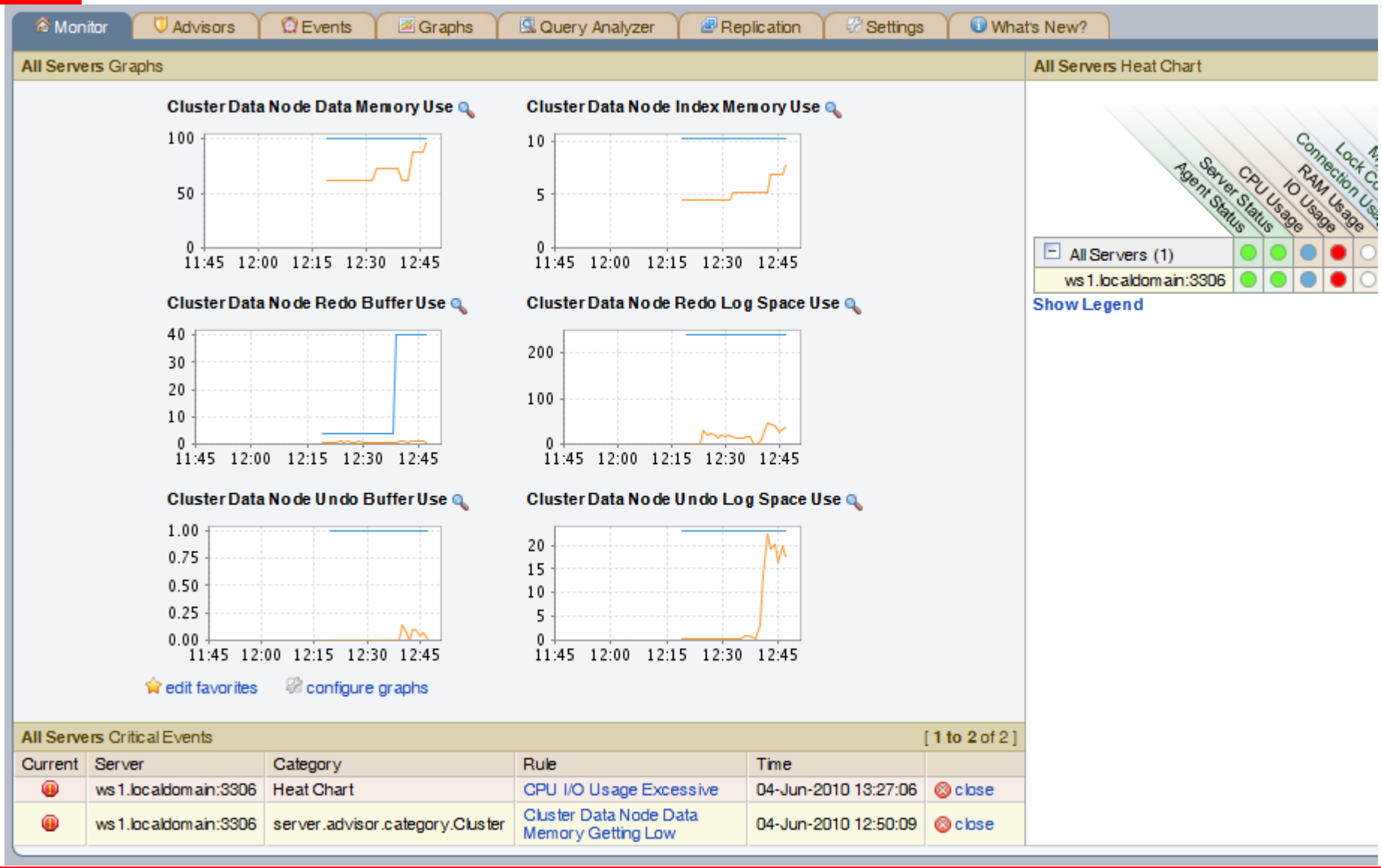


Beyond 7.1: SPJ – Push Down Joins



- *A linked operation is formed by the MySQL Server from the SQL query and sent to the data nodes*
- *For a linked operation, first part of query can be a scan but should result in primary key lookups for the next part*
- *More complex queries could be sent as multiple linked operations*
- *Reduces latency and increases throughput for complex joins*
  - *Qualifies MySQL Cluster for new classes of applications*
- *Also possible directly through NDB API*
- *Up to 42x performance gain in PoC!*

MySQL Enterprise Monitor 2.3 (pre GA)

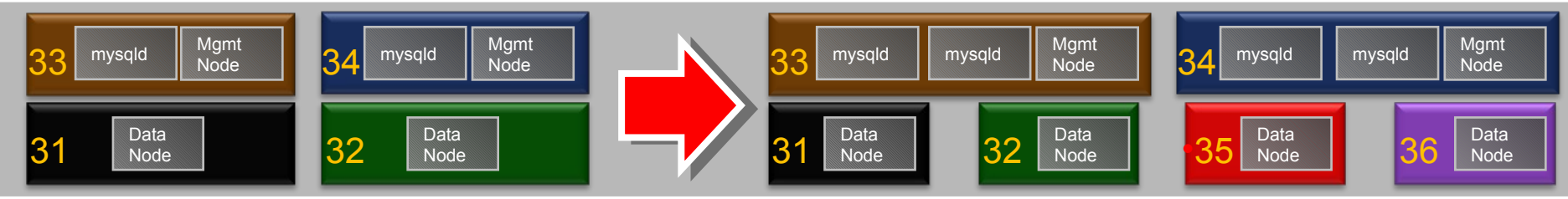


MySQL Cluster Manager 1.0 Features

Automated Management	Monitoring	HA Operations
Cluster-Wide Management Process Management On-Line Operations (Upgrades / Reconfiguration)	Status Monitoring & Recovery	Disk Persistence Configuration Consistency HA Agent Operation

MySQL Cluster Manager

Current Development Projects



- On-line add-node

```
mysql> add hosts --hosts=192.168.0.35,192.168.0.36 mysite;
mysql> add package --basedir=/usr/local/mysql_7_0_7 -
  hosts=192.168.0.35,192.168.0.36 7.0;
mysql> add process
  --processhosts=mysqld@192.168.0.33,mysqld@192.168.0.34,ndbd
  @192.168.0.35,ndbd@192.168.0.36 mycluster;
mysql> start process --added mycluster;
```

- Restart optimizations
  - Fewer nodes restarted on some parameter changes

- Application: Service Delivery Platform
  - Roaming platform to support 7m roaming subscribers per day FIFA World Cup 2010
  - Database supports AAA, routing, billing, messaging, signalling, payment processing
  - MySQL Cluster 7.1 delivered 1k TPS on 1TB data with carrier-grade availability
- Key business benefits
  - Local carriers to monetize new subscribers
  - Users enjoy local pricing with full functionality of their home network
  - Reduced deployment time by 75%

”MySQL Cluster 7.1 gave us the perfect combination of extreme levels of transaction throughput, low latency & carrier-grade availability. We also reduced TCO by being able to scale out on commodity server blades and eliminate costly shared storage”

- Phani Naik, Head of Technology at Pyro Group

Shopatron: eCommerce Platform

Shopatron

- Applications
  - Ecommerce back-end, user authentication, order data & fulfilment, payment data & inventory tracking. Supports several thousand queries per second
- Key business benefits
  - Scale quickly and at low cost to meet demand
  - Self-healing architecture, reducing TCO
- Why MySQL?
  - Low cost scalability
  - High read and write throughput
  - Extreme availability

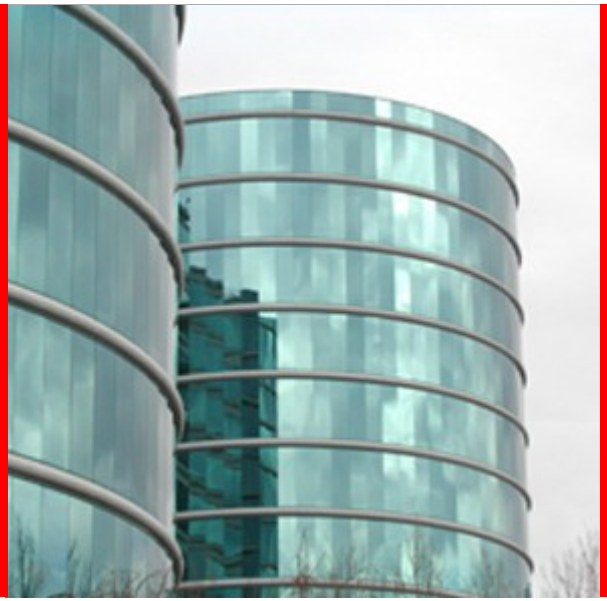
"Since deploying MySQL Cluster as our eCommerce database, we have had continuous uptime with linear scalability enabling us to exceed our most stringent SLAs"

— Sean Collier, CIO & COO, Shopatron Inc

Resources to Get Started

- MySQL Cluster Quick Start Guides
  - http://www.mysql.com/products/database/cluster/get-started.html#quickstart
- MySQL Cluster 7.1, Architecture and New Features
  - http://www.mysql.com/why-mysql/white-papers/mysql_wp_cluster7_architecture.php
- MySQL Cluster on the Web
  - http://www.mysql.com/products/database/cluster/

The preceding 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.



MySQL™

ORACLE®