



**ORACLE®**



## **MySQL Cluster**

**Delivering Scalable & Highly Available Session Management**

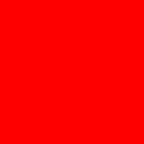
Mat Keep (mat.keep@oracle.com)  
MySQL Cluster Product Management

Bernd Ocklin (bernd.ocklin@oracle.com)  
Director, MySQL Cluster Engineering

# Session Agenda

- Overview of MySQL Cluster
- Challenges of Session Management
- Implementing Session Management with PHP and MySQL Cluster
- Resources to Get Started





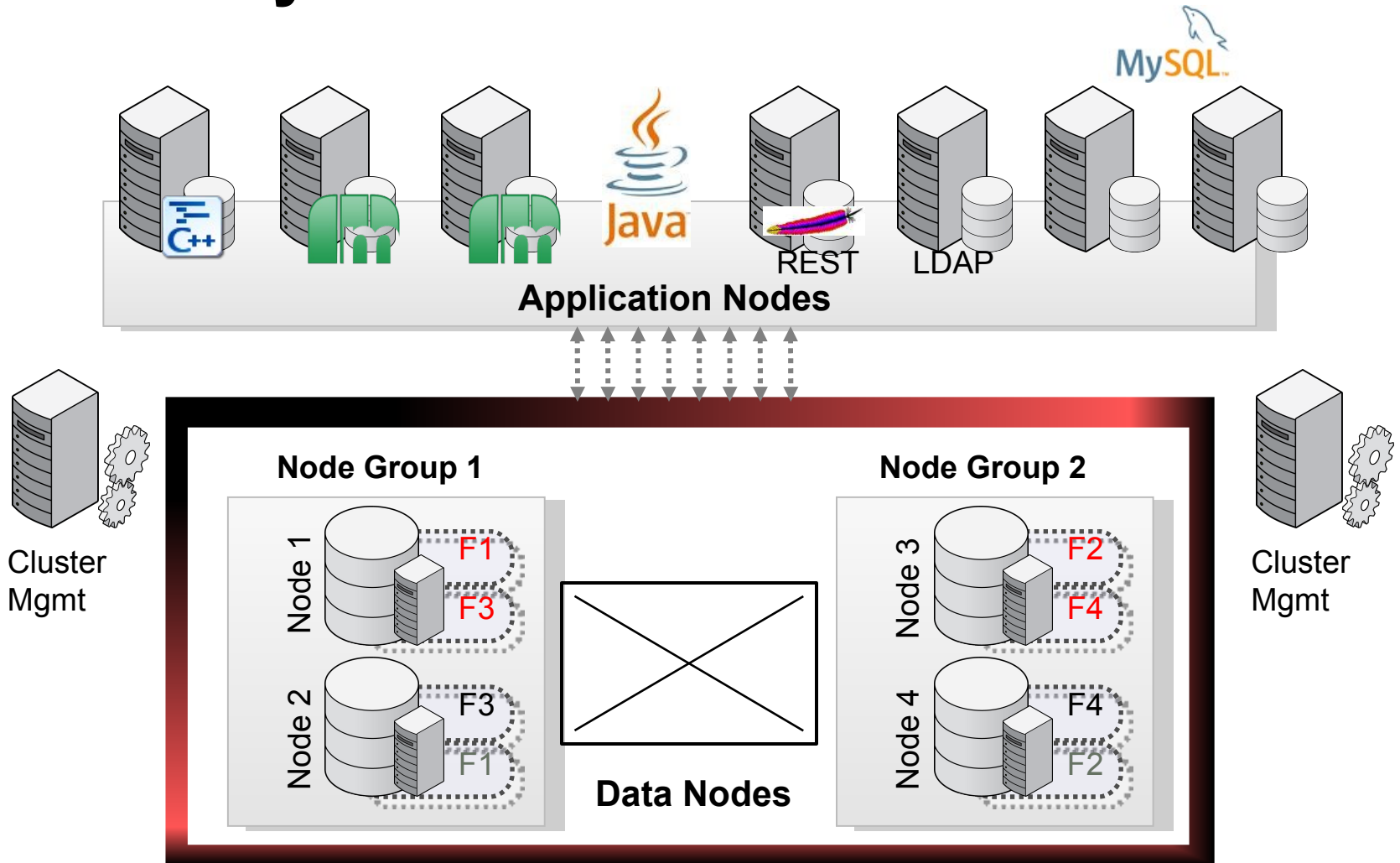
The presentation is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.

The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

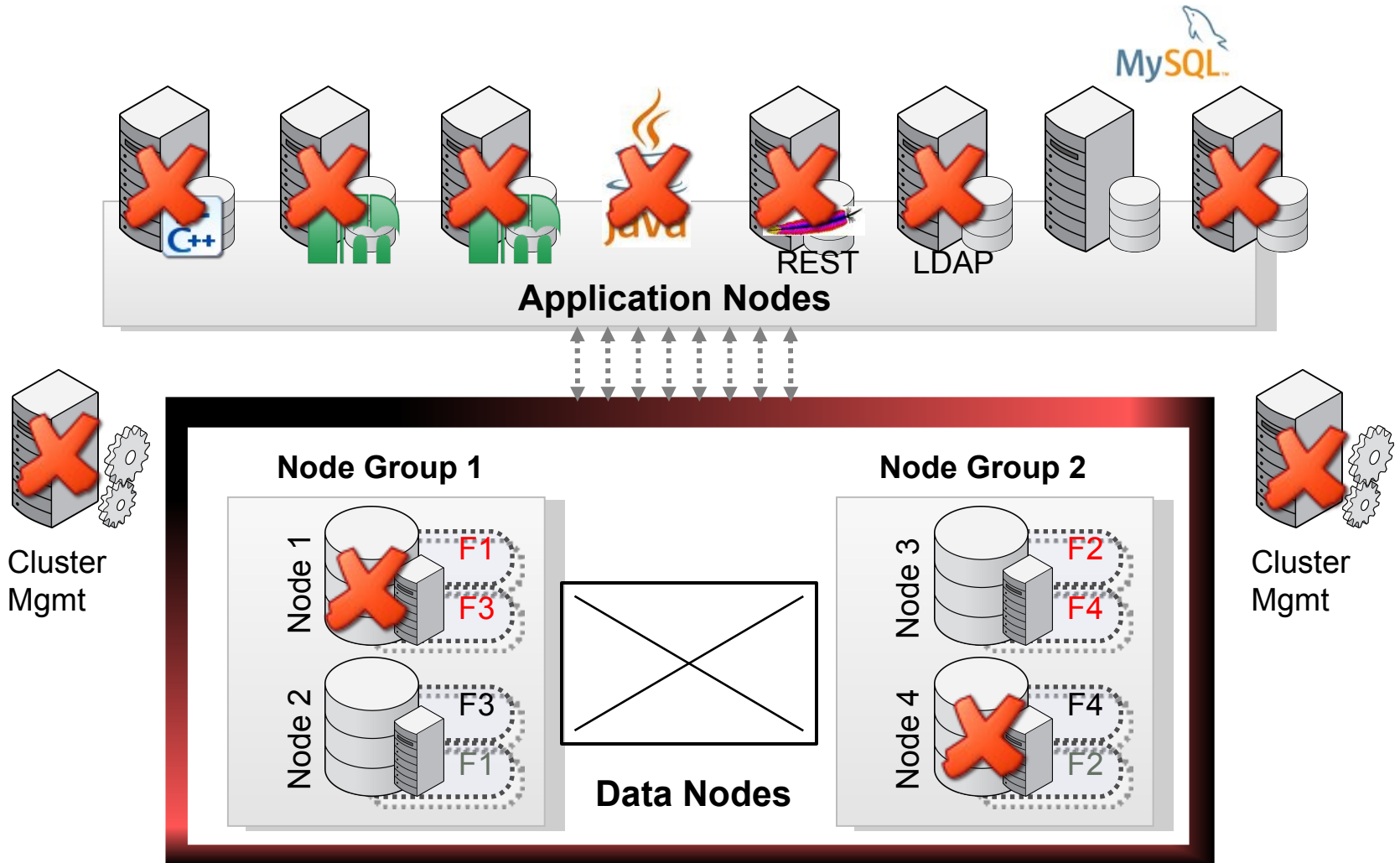
# MySQL Cluster Overview

- ACID Compliant Relational Database
  - SQL & NoSQL interfaces
- Write-Scalable & Real-Time
  - Distributed, auto-partitioning (sharding), multi-master
- 99.999% Availability
  - Shared-nothing, integrated clustering & sub-second recovery, local & geographic replication, on-line operations
- Low TCO
  - Open-source, management & monitoring tools, scale-out on commodity hardware

# MySQL Cluster Architecture



# MySQL Cluster - Extreme Resilience



# 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 7.1 Momentum



MySQL & Pyro Score at  
**FIFA 2010 World Cup**

[Learn More »](#)

**1,000 Downloads per Day**

Windows GA

Fully Automated  
Management

Pro-active Cluster  
Monitoring

10x Higher Java  
Performance

**Pyro**

*“MySQL Cluster 7.1 gave us the perfect combination of extreme levels of transaction throughput, low latency & carrier-grade availability, while reducing TCO”*

Phani Naik, Pyro Group

# Current Enhancement Plans– (For information only)

## Performance

- Adaptive Query Localization (Download NOW: 7.2 DM)

## Capacity

- Increased number of columns (Download NOW: 7.2 DM)

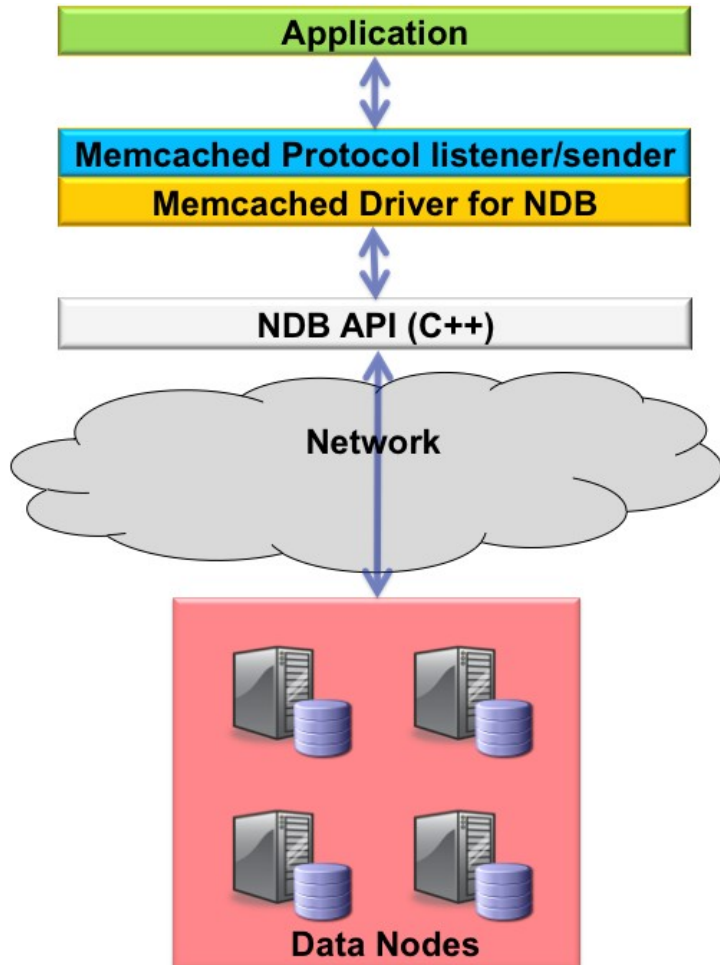
## Ease of use

- Memcached-API (Download NOW: labs.mysql.com)
- Geo-Replication: simplified conflict detection/resolution
- System Table Consolidation (Download NOW: 7.2 DM)

Note that not all of these items will necessarily be in the next MySQL Cluster release.

The existence, content and timing of future releases described here is included for information only and may be changed at Oracles discretion. April 13, 2011

# 7.2 DM: Memcached Key-Value API



- Build update-intensive, highly available services with MySQL Cluster back-end
  - Accessed via memcached API
- Consolidate caching and database tiers
  - Use existing memcached clients & avoid application changes
  - Support for update-intensive workloads, eliminate cache invalidation
  - Scalable, persistent, HA data store
  - Simpler re-use of data across services
- Implementation
  - Memcached driver for NDB plug-in to memcached server
  - Direct access to NDB API



# MySQL Cluster & Session Management

# Session Management Primer

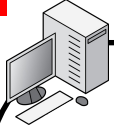
- HTTP protocol is “stateless”
  - HTTP is used in the interaction between browsers and web servers and is stateless by nature
  - “Stateless” means the HTTP protocol does not store/save information about pages visited or interactions/transactions
- State of the application needs to be stored between HTTP requests
  - No permanent connection between web client and server
  - Isolated single requests and their reply
  - No persistence of a connection or client identification from one request to next

# Session Management Examples

Uses	Examples
Shopping Carts	eCommerce
Customized Web Pages	App Stores
Inactivity Timeouts or Disconnects	Online Banking
Settings & Preferences	Mobile Broadband
Product Recommendations	eCommerce
Wish Lists	eCommerce
Targeted Advertising	Social Networking & LBS
Previous Searches	eCommerce
Profiles & Accounts	Social Networking
Address Books	Communications Networks

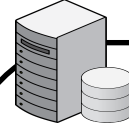
*Any kind of “Know / Recognize” your visitor*

# Session Management Solutions



## Browser-based

- Good if data is not complex or large
- Hard limits on # or size of cookies
- Privacy and security concerns
  
- Examples:
  - Cookies / Client Side HTML5
  - URL rewriting
  - Hyperlinks
  - Forms



## Server-based

- Scalable
- Manageable
- Performance
- Security
  
- Examples:
  - Web Server
  - Application Server
  - Database Server

## Considerations

**Platform Interoperability**

**Performance & Throughput**

**Lock and Transactional Support**

**Scalability**

**Load Balancing**

**Security**

**High Availability**

**Data Management**

**Timeouts and Inactivity**

**End-User Experience**

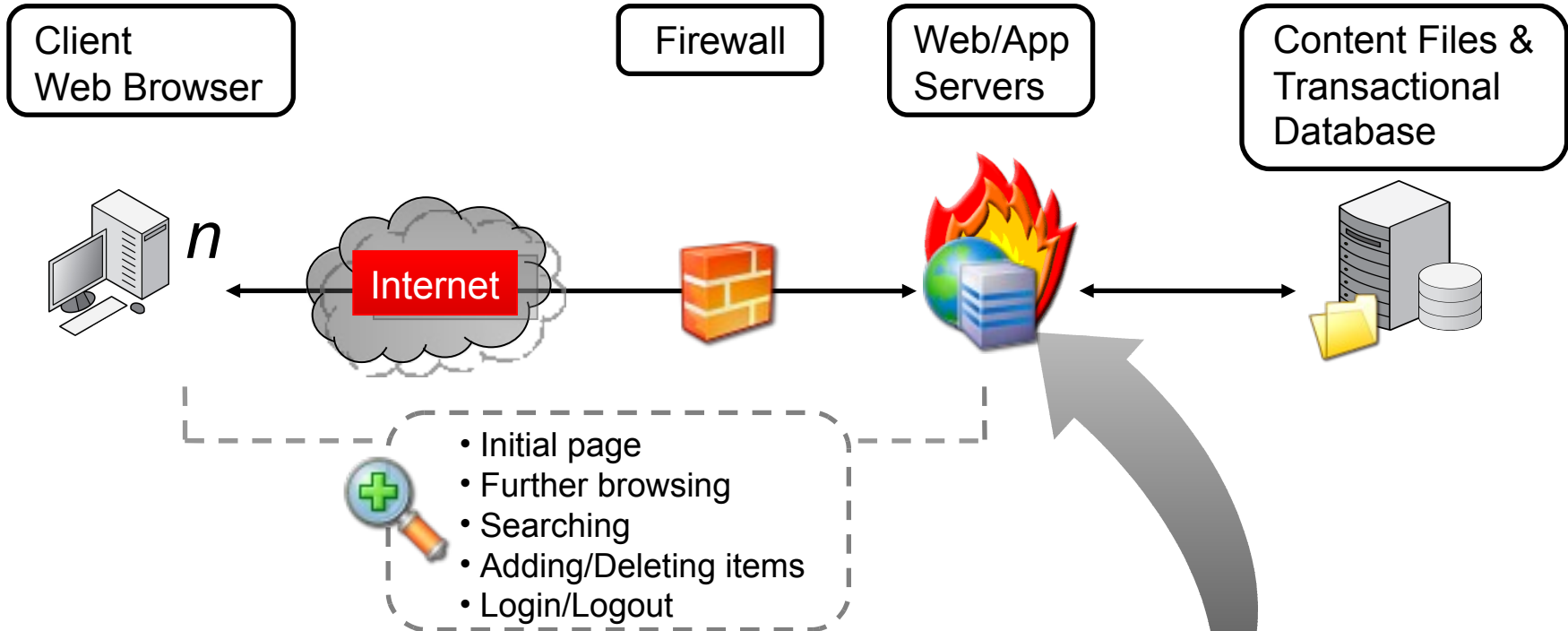
# Implementing Sessions in Web/App Layer

## Scalability, Locking....

- Each HTTP request needs to be processed in the context of the session variables for which the request belongs
- Web Server example:
  - After browsing the initial page, subsequent requests like further browsing, searching, add/deleting items in a cart, or logging in or out, must be processed on the web server holding the session variables
- Difficult to distribute requests across Web Servers
  - Makes Scale-Out difficult
- File system may cause potential bottleneck
  - Does not provide an efficient locking mechanism for concurrent access
- Other considerations
  - Security, Availability, Managing Timeouts

# Implementing Sessions in Web/App Layer

## Scalability and Load Balancing



*When managing sessions on the Web Server, expect an increased workload as the number of clients and requests multiply, creating a potential bottleneck.*

# Using memcached for Sessions

- Many users do
  - Read / Write session data to memcached
  - Set expiration time at 30 minutes
- Kick some or all browsers off your site.....
  - Failures
  - Adding or removing memcached servers
  - Upgrades to memcached or the underlying infrastructure
- New requirements
  - Persist session data for mining and analytics
  - Maintain sessions for disconnected mobile clients
  - Eliminate duplication of data across the infrastructure

# Addressing Session Management Challenges



## Concerns

*Can we handle an increase in traffic for the holidays?*

*How can I balance load across my Web Servers?*

*How can we create a customized end-user experience?*

*Why do we have so many inactive connections tying up resources?*

*How can we make our website more secure?*

*How can we deliver targeted advertising?*



## Benefits of storing session data in the database

*Helps control user interaction across multiple devices*

*Reduces processing and resource consumption on the Server-side*

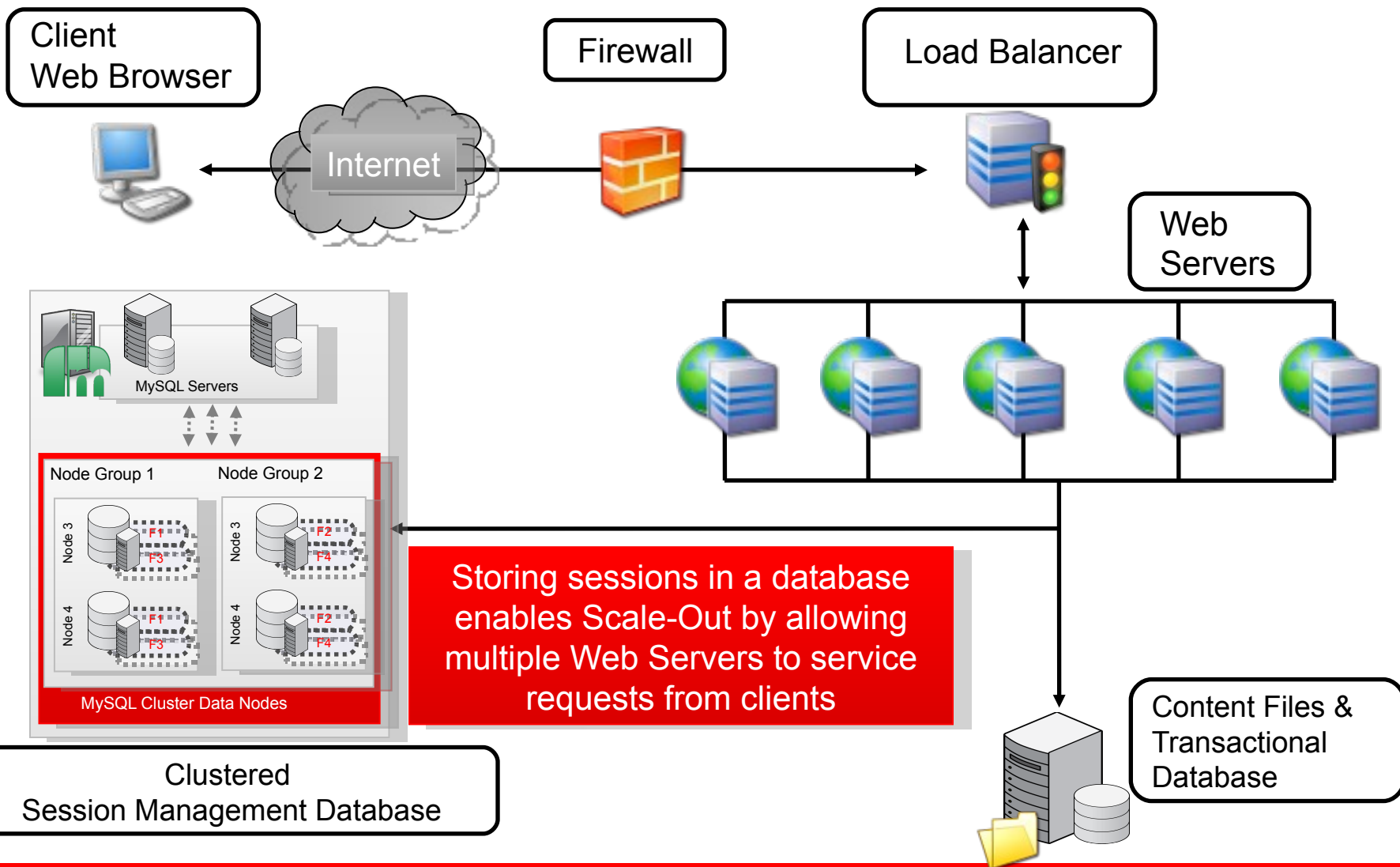
*Handles increased volume, size and complexity of data*

*Increased security*

*Helps manage user inactivity and timeouts*

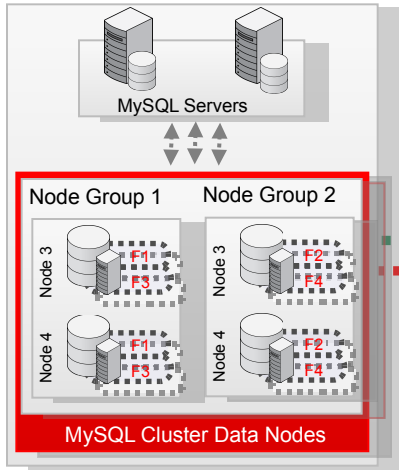
# Implementing Sessions in MySQL Cluster

## Supporting High Update Rates and 99.999% Uptime

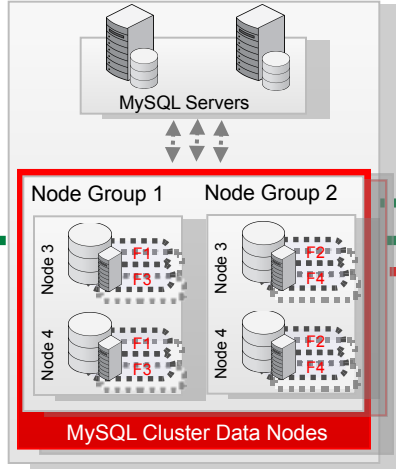


# MySQL Web Reference Architectures

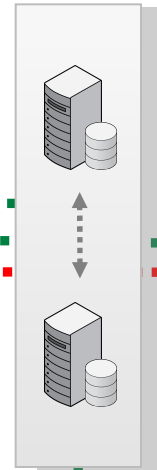
## Session Management



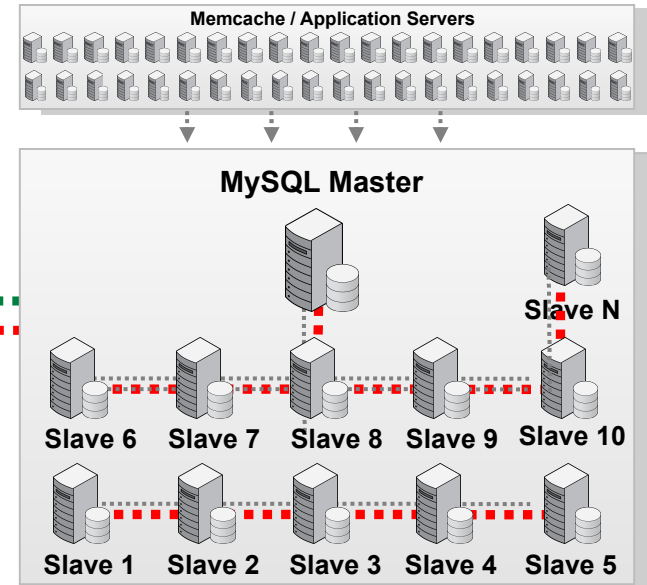
## eCommerce



## Data Refinery

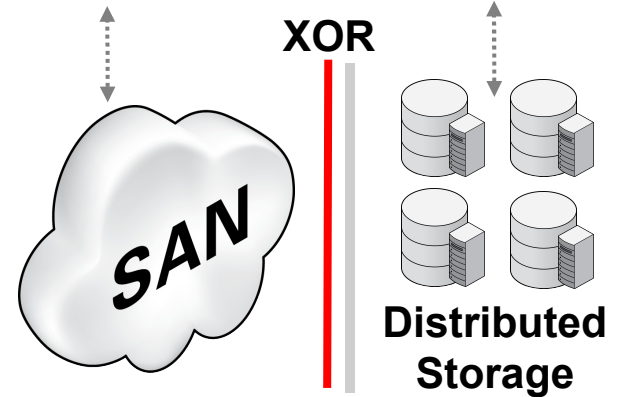
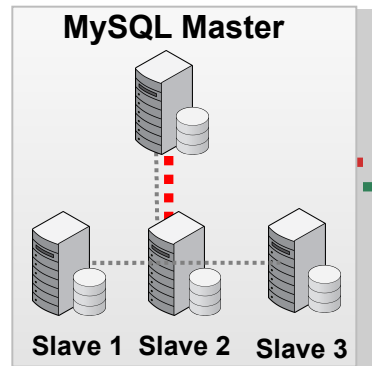


## Content Management



- 4 x Data Nodes support ~6k page hits per second
- Each page hit generating 8 – 12 database operations

## Analytics



# Session management in PHP

- `$_SESSION[]` as array containing session variables
- `start_session()` create or resume a session

```
<?php
```

```
    session_start();
```

```
    $_SESSION['cake'] = 'bakery ocklin';
```

```
    $_SESSION['pastery'] = 'bread baker';
```

```
    // cookie
```

```
    echo '<a href="smart_baker_jones.php">next baker</a>';
```

```
    // session_id in URL
```

```
    echo '<a href="cookie_less.php?' . SID . '">next baker</a>';
```

# PHP session persistence

- using files/memory by configuring `php.ini`  
`session.save_handler = „files“`  
`session.save_path = „/tmp/my“`
- via a customized `session_set_save_handler()`  
providing custom callback functions

# PHP session persistence via SQL

```
class SessionHandler {
    SessionHandler() {
        session_set_save_handler(
            array($this, 'open'),
            array($this, 'close'),
            ...,
            array($this, 'gc'));
        session_start();
    }
    function open($save_path, $session_name) ...
    function close() ...
    function read($id) ...
    function write($id, $sess_data) ...
    function destroy($id) ...
    function gc($maxlifetime) ...
}
```

# PHP session persistence via SQL

```
class SessionHandler {
    ...
    function read($id) {
        SELECT data FROM session_tab WHERE id = $id;
    }
    function write($id, $data) {
        UPDATE session_tab SET data = $data WHERE id = $id;
    }
    function gc($maxlifetime) {
        DELETE FROM session_tab
            WHERE expires < time() - $maxlifetime;
    }
    ...
}
```

# Using the session class

```
<?php
    require_once("SessionHandler.php");
    $sess = new SessionHandler();

    if (!isset($_SESSION["count"])) $_SESSION["count"]=0;
    if (!isset($_SESSION["start"])) $_SESSION["start"]=time();

    $_SESSION["count"]++;
    $duration = time() - $_SESSION["start"];
?>
...
<body>
    <br>count = <?php echo $count; ?>.
    <p>Session lasted <?php echo $duration; ?> seconds.
</body>
```

# Using memcached on top of cluster

- your sessions are gone when memcached goes
- but on top of cluster: safe & keep on scaling

- install PECL memcached
- run cluster & the memcache ndb plugin

```
bin/memcached -E lib/ndb_engine.so
```

```
-e "connectstring=localhost:1186;role=db-only"
```

- PHP.INI

```
session.save_handler=memcache
```

```
session.save_path=tcp://127.0.0.1:11211
```



## COMPANY OVERVIEW

- Division of DocuDesk
- Deliver Document Management SaaS

## CHALLENGES / OPPORTUNITIES

- Provide a single repository for customers to manage, archive, and distribute documents
- Implement scalable, fault tolerant, real time data management back-end
- PHP session state cached for in-service personalization
- Store document meta-data, text (as BLOBs), ACL, job queues and billing data
- Data volumes growing at 2% per day

## SOLUTION

- MySQL Cluster deployed on EC2

## USER PERSPECTIVE

*“MySQL Cluster exceeds our requirements for low latency, high throughput performance with continuous availability, in a single solution that minimizes complexity and overall cost.”*

-- Casey Brown, Manager of Dev & DBA Services,  
DocuDesk



## RESULTS

- Successfully deployed document management solution, eliminating paper trails from legal processes
- Integrate caching and database into one layer, reducing complexity & cost
- Support workload with 50:50 read/write ratio
- Low latency for real-time user experience and document time-stamping
- Continuous database availability

# Summary

- Session Management critical to deliver high quality, personalized web experiences
  - Growing requirement to persist session data for analytics
  - Session data volume and complexity growing
- MySQL Cluster proven for session management
  - High read and write performance, with scalability
  - Automated load balancing and data partitioning
  - In-memory management deliver real-time responsiveness
  - 99.999% uptime ensures service availability
- Memcached key-value API delivers more development and deployment flexibility

# Getting Started

## Learn More – GA Release

Architecture &  
New Features  
Guide  
[www.mysql.com/cluster/](http://www.mysql.com/cluster/)



## Evaluate MySQL Cluster 7.2

Download Today

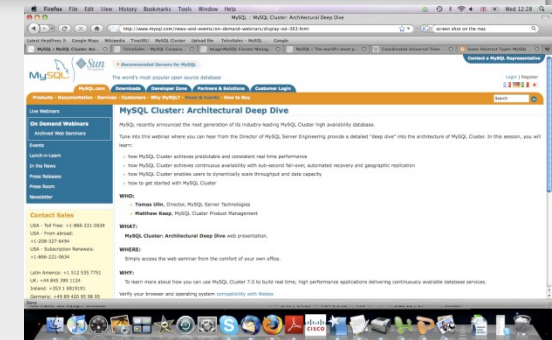
<http://dev.mysql.com/download>

<http://labs.mysql.com>  
(memcached)



## Quick Start Guides

Linux, Solaris,  
Windows  
<http://tinyurl.com/5wkl4dy>



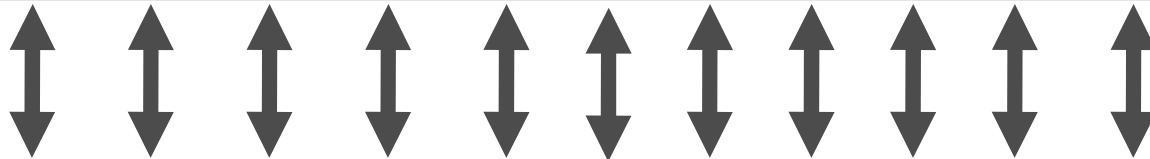
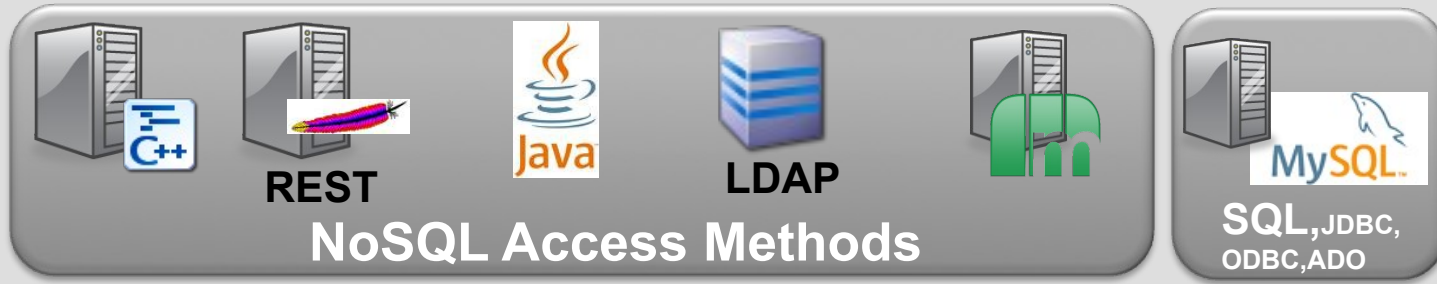
**ORACLE®**

# MySQL Cluster Architecture

Multi-Master, no SPOF: High Write Performance, Real-Time & 99.999% Uptime



## MySQL Cluster Application Nodes



MySQL Cluster Mgmt



# What is Session Management?

- "Session management allows a 'virtual connection' to be established between the web browser and server to personalize web services."
- "The ability to track the state and interaction of a users activity on a system."
- "A session refers to a series of related interactions between a user and your Web application. Session management refers to how your application handles and protects these interactions."

“DocuDesk **relies on MySQL Cluster** to support our DocQ SaaS offering which **demands high update rates, low latency and continuous availability** from the database. Testing of **Adaptive Query Localization** has yielded over 20x higher performance on complex queries within our application, enabling DocuDesk to expand our use of MySQL Cluster into a broader range of highly dynamic web services.”

## **Casey Brown**

Manager, Development & DBA Services, DocuDesk

# SQL and NoSQL Access Methods

	SQL	C++ (NDB API)	Java (ClusterJ/JPA)	LDAP	Memcached (planned enhancement)	mod-ndb
Description	Provided by MySQL Server, access to all MySQL connectors	Native API directly into the data nodes	Java ORM that accesses the NDB API directly	Plugin to allow direct access from LDAP server to NDB API	Presents popular Memcached API with direct access to NDB API	Apache module that provides a REST web services API for MySQL Cluster
Use cases	Applications wanting the simplicity and richness of SQL.	Where the lowest, most predictable latency & highest throughput is needed.	Java applications wanting > simplicity and > performance than JDBC	LDAP based applications looking for high write performance & scalability	Developers familiar with Memcached but wanting simplicity of a single, distributed data store. Scalable, persistent key-value store with simple API	Web applications wanting fast, access to Highly Available data using REST API
Support	<b>GA. Supported by Oracle</b>	<b>GA. Supported by Oracle</b>	<b>GA. Supported by Oracle</b>	<b>Supported by Symas</b>	<b>Not Applicable</b>	<b>Community</b>
Schema changes	<b>Yes (online)</b>	<b>Through SQL (online)</b>	<b>Yes (online), if using JPA option</b>	<b>Relational schema built from LDAP schema</b>	<b>Key-value store</b>	<b>Use MySQL Server (online)</b>
Read/Write Performance	<b>High and scalable</b>	<b>Extreme and scalable</b>	<b>Very high and scalable using ClusterJ</b>	<b>Very High and scalable</b>	<b>Very high and scalable</b>	<b>Very high and scalable</b>
Ease of use	<b>Simple</b>	<b>Complex</b>	<b>Very simple to Java developers</b>	<b>Very simple to LDAP developers</b>	<b>Very simple</b>	<b>Simple</b>
Language	<b>Regular SQL, common to MySQL &amp; other RDMS</b>	<b>Unique to Cluster</b>	<b>ClusterJ – unique to Cluster JPA – open &amp; common</b>	<b>LDAP is a common directory access protocol</b>	<b>Memcached API is commonly used – esp in web</b>	<b>Very common: HTTP, XML, JSON, HTML</b>
JOINS	<b>Yes (getting faster)</b>	<b>Programmatically</b>	<b>Using JPA</b>	<b>No</b>	<b>No</b>	<b>Programmatically</b>

# 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

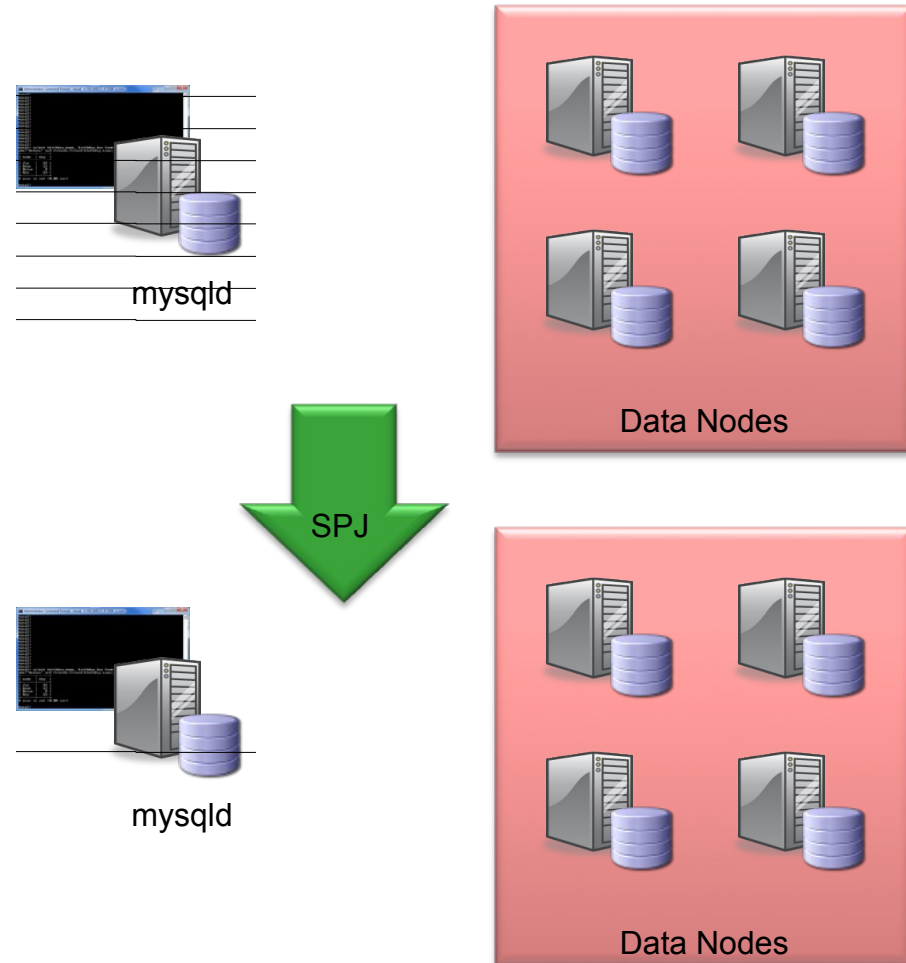
Low TCO,  
Open platform

GPL & Commercial editions, scale on COTS  
Flexible APIs: SQL, C++, Java, OpenJPA, LDAP & HTTP

# Introducing MySQL Cluster

- 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
- SQL/JDBC/ODBC/standards
- NoSQL Access Methods: C++, Java, LDAP, REST...Memcached (future)
- Enables agile development – on-line schema changes
- Automatic or user configurable data partitioning across nodes
- Synchronous data redundancy
- 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. upgrades, add-nodes, schema updates, etc)

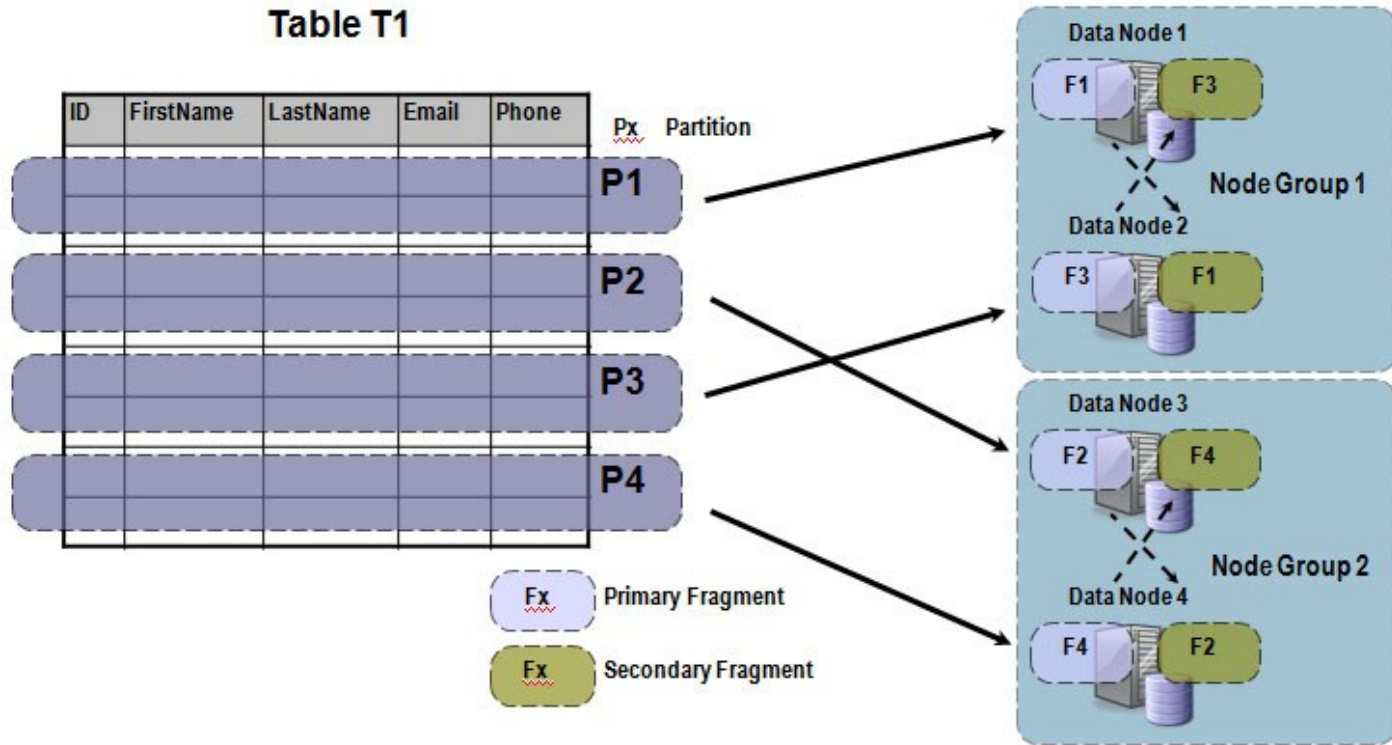
# 7.2 DM: Adaptive Query Localization



- 'Complex' joins often much slower on MySQL Cluster
  - Complex = lots of levels and interim results in JOIN
- JOIN was implemented in the MySQL Server:
  - Nested Loop join
  - When data is needed, it must be fetched over the network from the Data Nodes; row by row
  - This causes latency and consumes resources
- Can now push the execution down into the data nodes, greatly reducing the network trips
- **25x performance gain in customer PoC!**

The existence, content and timing of future releases described here is included for information only and may be changed at Oracles discretion. February 10, 2011

# Out of the Box Scalability: Data Partitioning

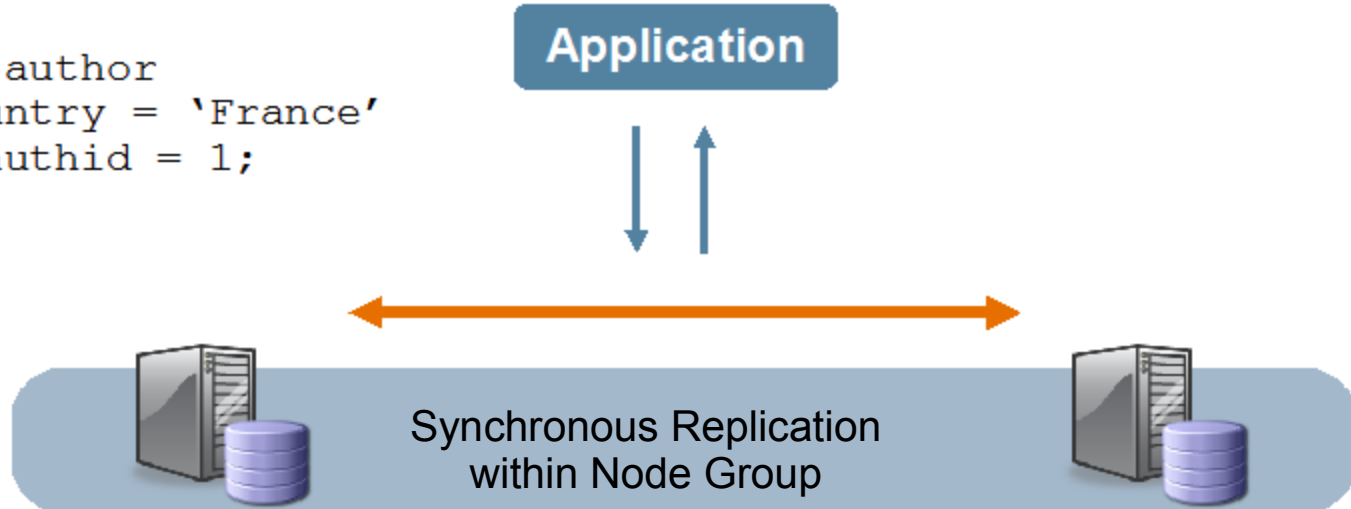


- Data partitioned across Data Nodes
- Rows are divided into partitions, based on a hash of all or part of the primary key
- Each Data Node holds primary fragment for 1 partition
  - Also stores secondary fragment of another partition
- Records larger than 8KB stored as BLOBs

# Shared-Nothing Architecture for High Availability

## Update:

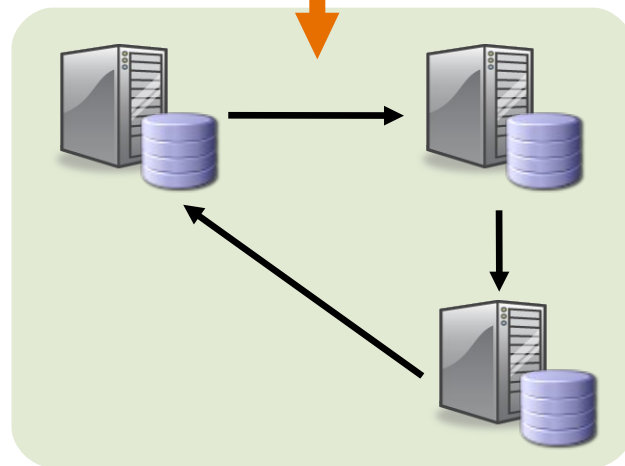
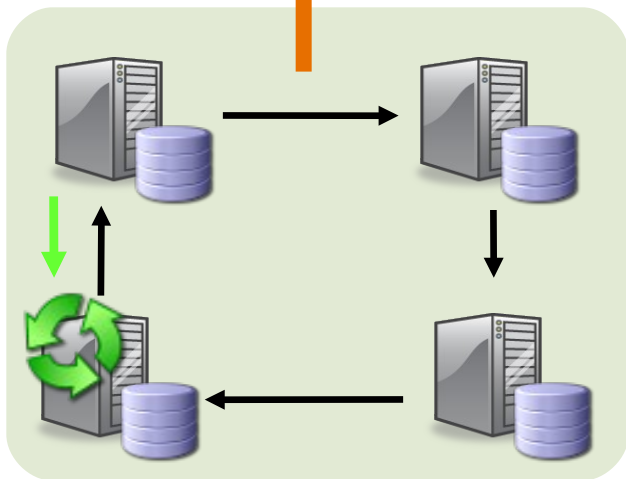
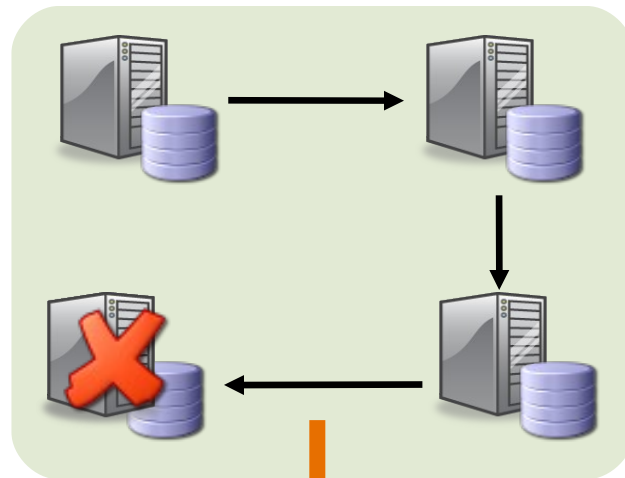
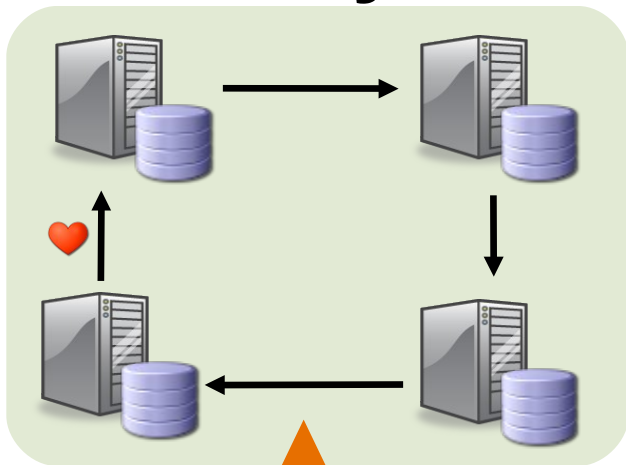
```
UPDATE author  
SET country = 'France'  
WHERE authid = 1;
```



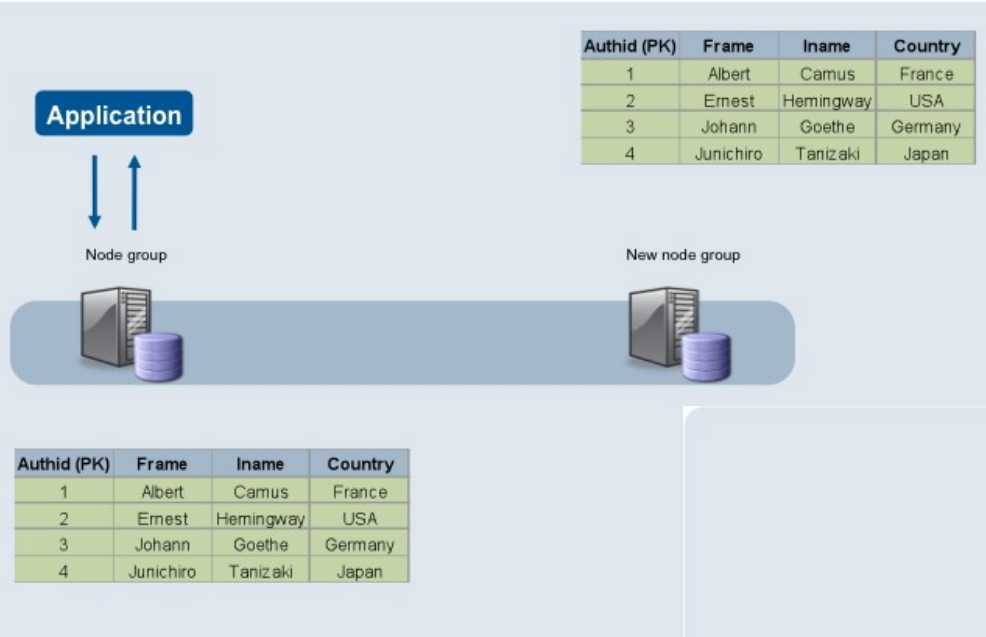
Authid (PK)	Frame	Iname	Country
1	Albert	Camus	France
2	Ernest	Hemingway	Cuba
3	Johan	Goethe	Germany
4	Junichiro	Tanizaki	Japan

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

# Node Failure Detection & Self-Healing Recovery

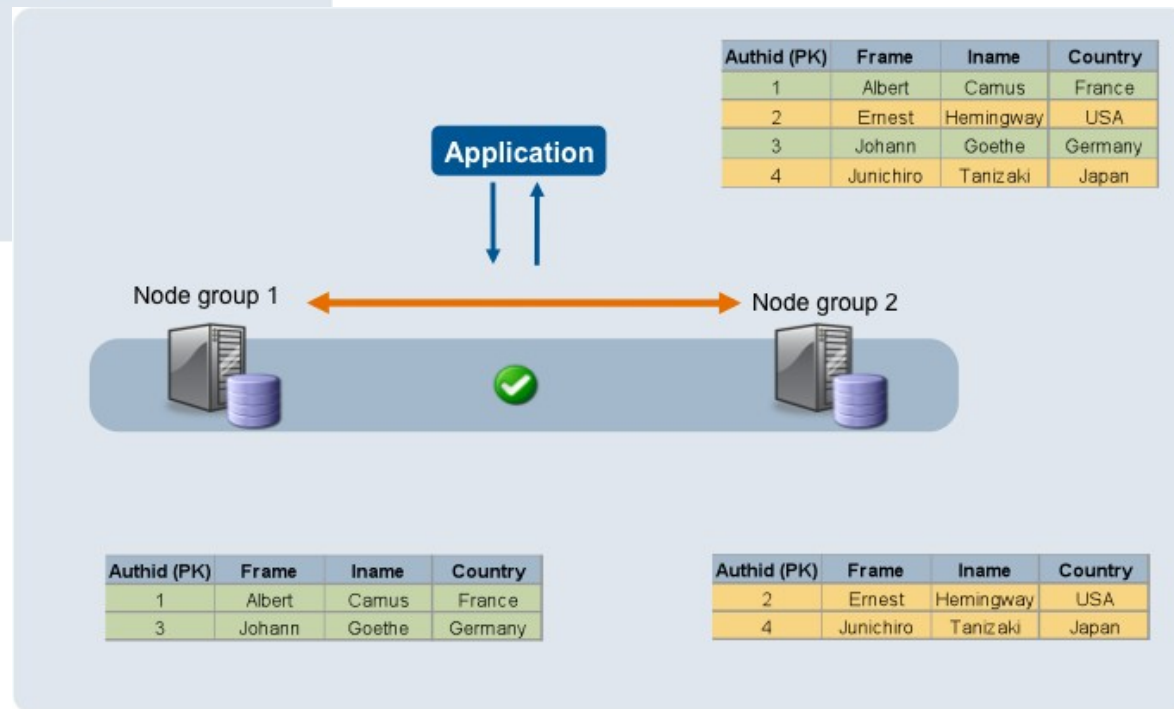


# On-Line Scaling & Maintenance

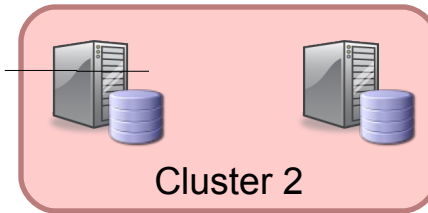


1. New node group added
2. Data is re-partitioned
3. Redundant data is deleted
4. Distribution is switched to share load with new node group

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



# 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



Synchronous  
replication

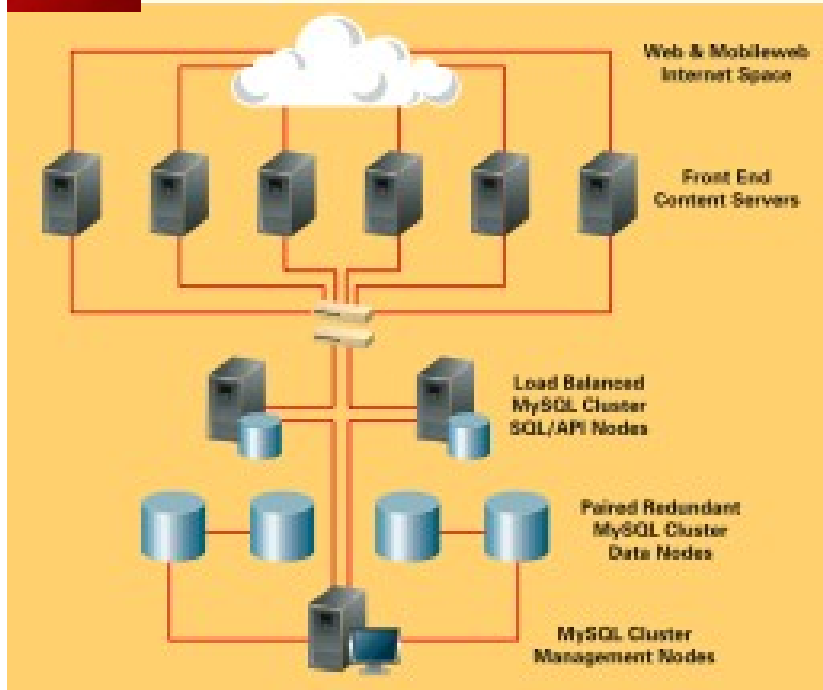
Asynchronous  
replication

- [http://www.ocklin.de/session\\_management\\_cluster.html](http://www.ocklin.de/session_management_cluster.html)
- <http://dormando.livejournal.com/495593.html>

# go2 Media: Mobile Media Publishing Platform

[http://www.mysql.com/why-mysql/case-studies/mysql\\_cs\\_Go2Media.php](http://www.mysql.com/why-mysql/case-studies/mysql_cs_Go2Media.php)

go2



- Application
  - Web-based city entertainment guide, accessed via mobile devices, with social networking integration
  - MySQL Cluster used to store user profiles, preferences and historic session state
- Key business benefits
  - On-Demand scalability, no up-front investments
  - Personalized, low latency user experience
- Why MySQL?
  - Freedom to download, develop and deploy without up-front costs
  - 99.999% availability, self healing
  - High throughput reads and writes, 1,100 QPS

“By building our infrastructure on MySQL Cluster, go2 has achieved a more stable environment, improved our user experience and now have the ability to efficiently scale our platform with the growth of the mobile web” — Dan Smith, Co-Founder & CEO, go2 Media