



MySQL High Availability

InnoDB Cluster and NDB Cluster

Mario Beck
mario.beck@oracle.com



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.

100%

Almost all organizations require their most critical systems to be highly available

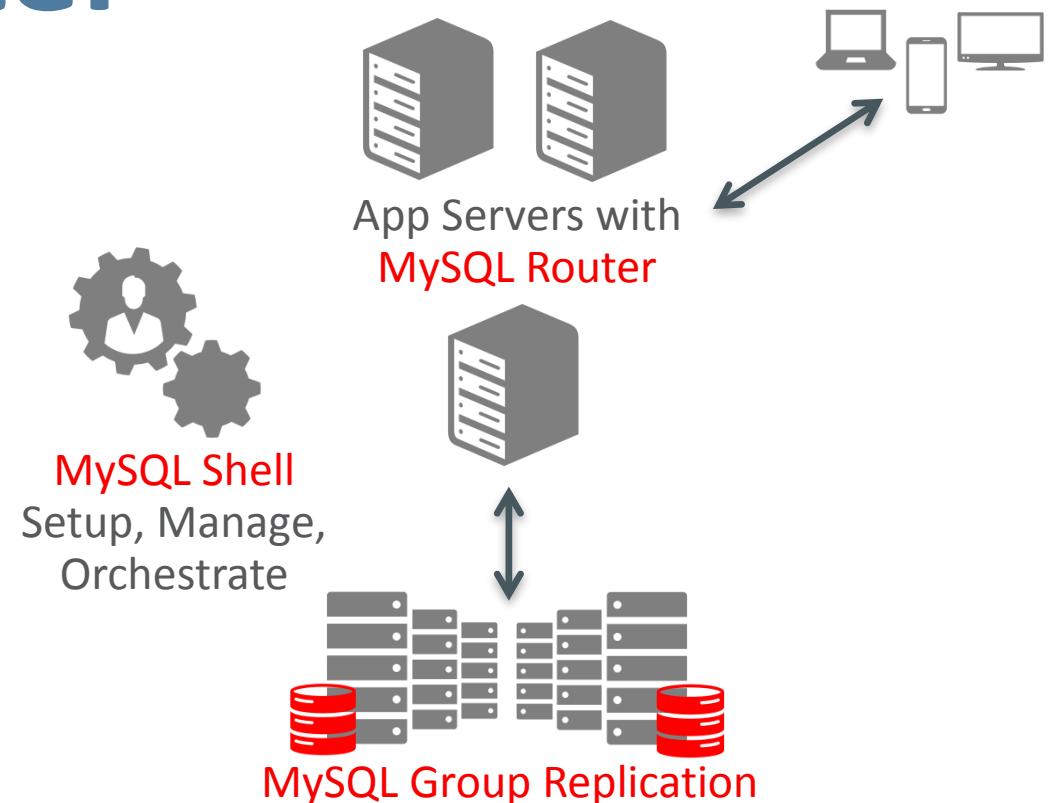


MySQL InnoDB Cluster



MySQL™ InnoDB Cluster

“High Availability becomes a core first class feature of MySQL!”



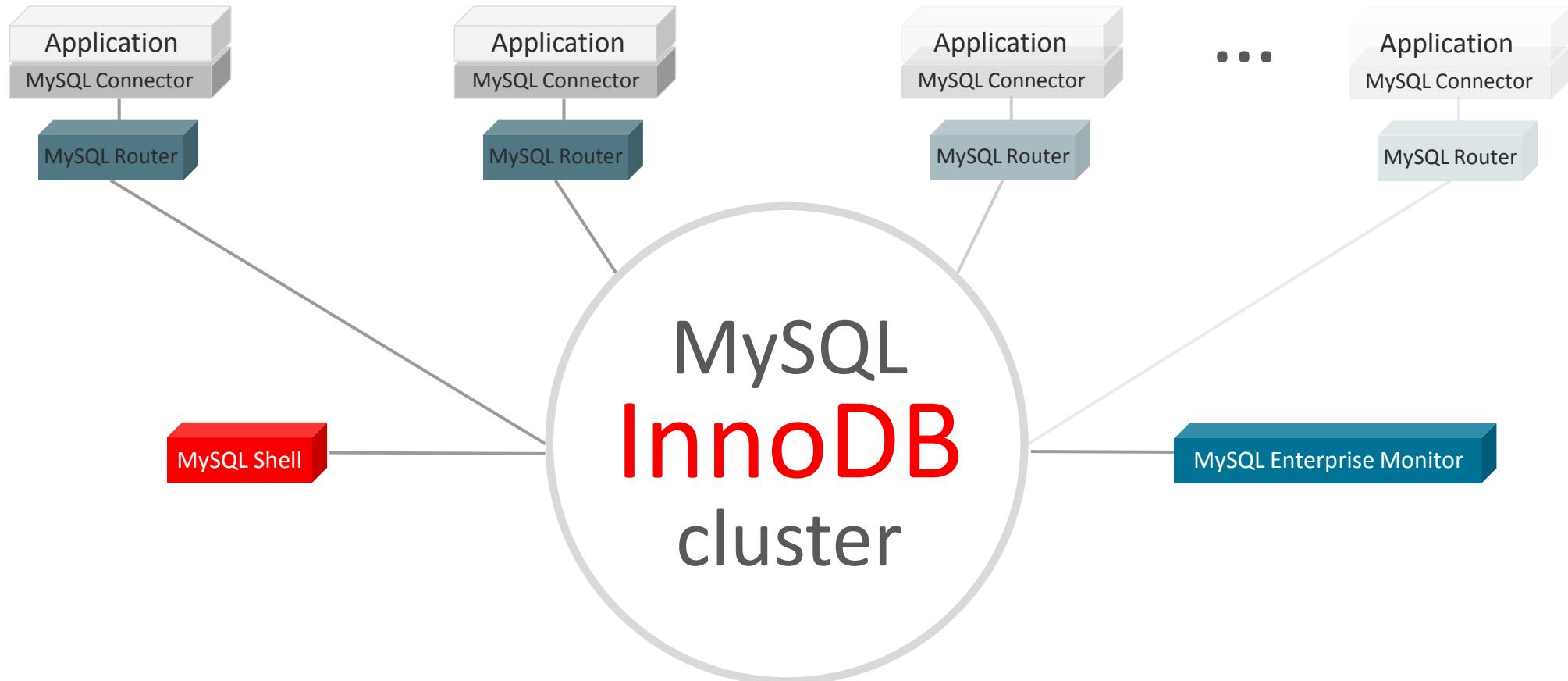
MySQL InnoDB Cluster: Vision

“A single product — MySQL — with high availability and scaling features baked in; providing an integrated end-to-end solution that is easy to use.”

MySQL InnoDB Cluster: Goals

- One Product: MySQL
 - All components created together
 - Tested together
 - Packaged together
- Easy to Use
 - One client: MySQL Shell
 - Easy packaging
 - Integrated orchestration
 - Homogenous servers
- Flexible and Modern
 - SQL and NoSQL together
 - Protocol Buffers
 - Developer friendly
- Support Read/Write Scale Out
 - Sharded clusters
 - Federated system of N replica sets
 - Each replica set manages a shard

MySQL InnoDB Cluster: High Level Architecture

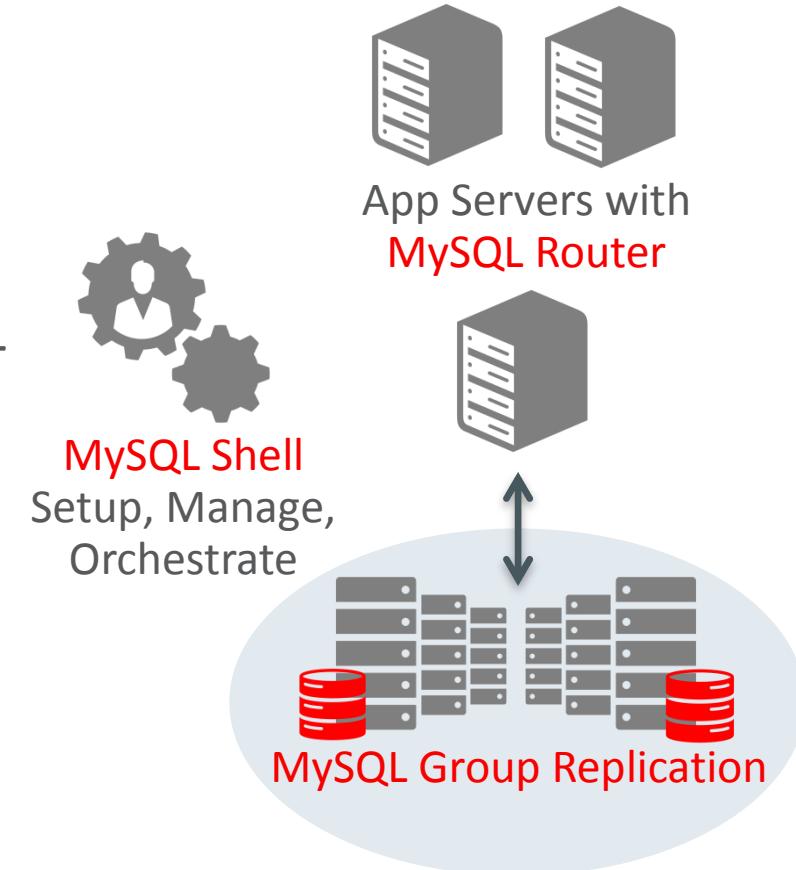


MySQL Group Replication

Natively distributed and highly available replica sets

MySQL Group Replication: What Is It?

- Group Replication library
 - Implementation of Replicated Database State Machine theory
 - MySQL GCS is based on Paxos (variant of Mencius)
 - Provides *virtually synchronous* replication for MySQL 5.7+
 - Supported on *all MySQL platforms*
 - Linux, Windows, Solaris, OSX, FreeBSD

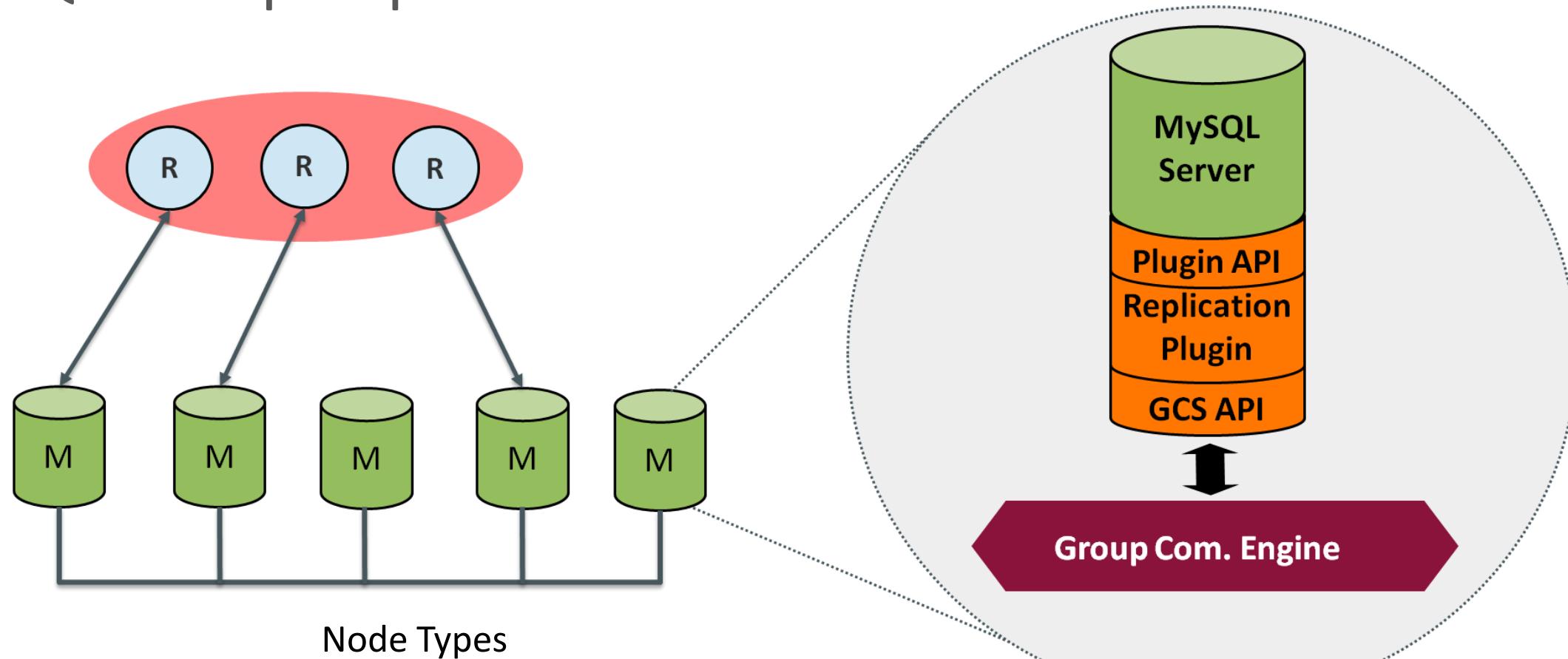


"Multi-master update anywhere replication plugin for MySQL with built-in conflict detection and resolution, automatic distributed recovery, and group membership."

MySQL Group Replication: What Does It Provide?

- A highly available distributed MySQL database service
 - Removes the need for manually handling server fail-over
 - Provides distributed fault tolerance
 - Enables Active/Active update anywhere setups
 - Automates reconfiguration (adding/removing nodes, crashes, failures)
 - Automatically detects and handles conflicts

MySQL Group Replication: Architecture



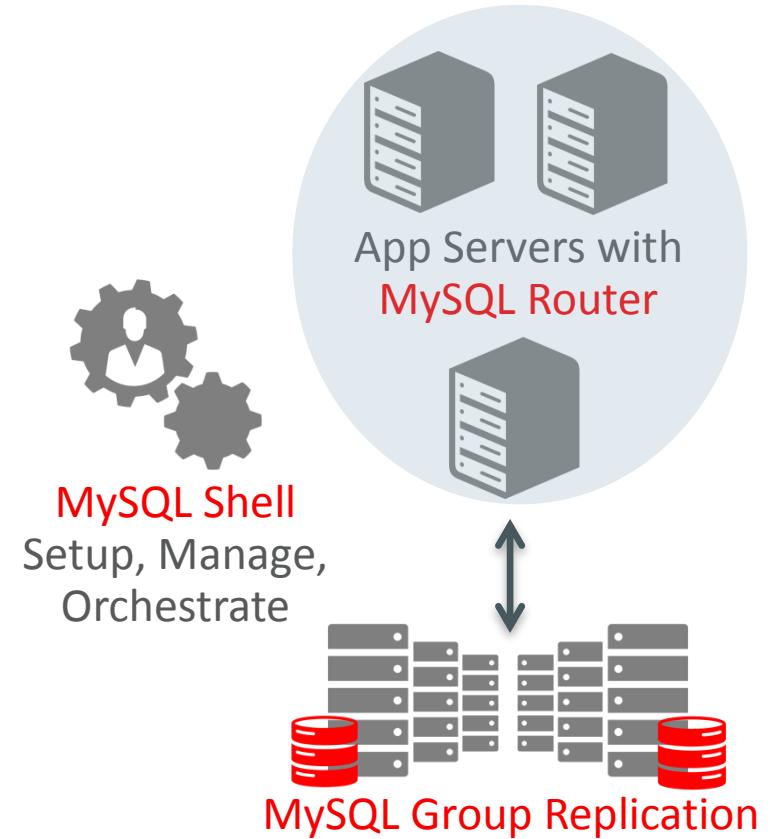
MySQL Router

Transparent application connection routing



MySQL Router: 2.1

- Native support for InnoDB clusters
 - Understands Group Replication topology
 - Utilizes metadata schema stored on each member
 - Bootstraps itself and sets up client routing for the InnoDB cluster
 - Allows for intelligent client routing into the InnoDB cluster
 - Supports multi-master and single primary modes
- Core improvements
 - Logging
 - Monitoring
 - Performance
 - Security



MySQL Shell

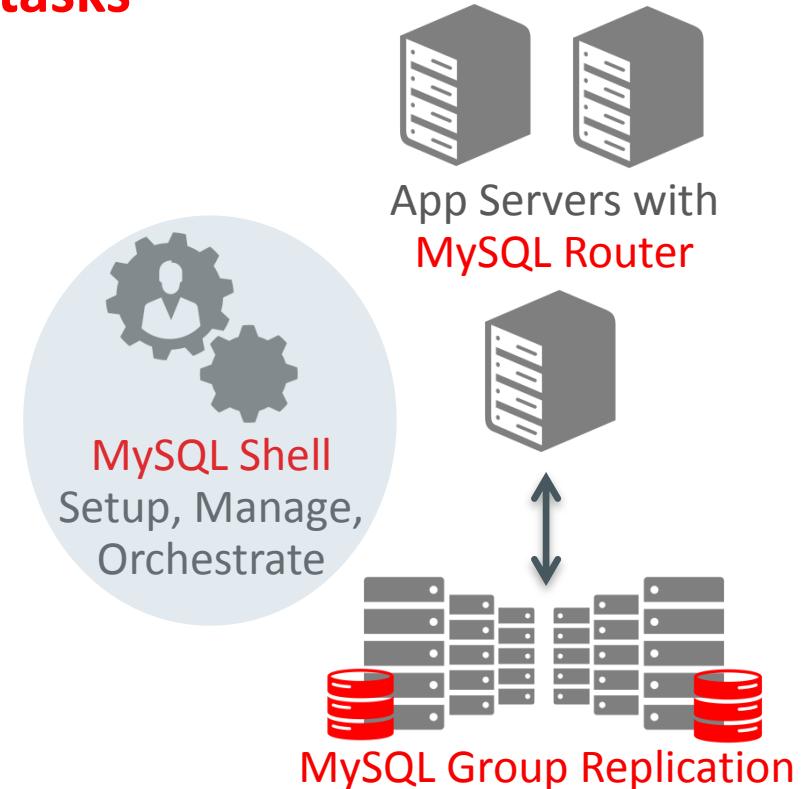
**Single tool for development, setup,
management, orchestration, and monitoring**

MySQL Shell

A single unified client for all administrative and operations tasks

- Multi-Language: JavaScript, Python, and SQL
 - Naturally scriptable
- Supports both Document and Relational models
- Exposes full Development and Admin API

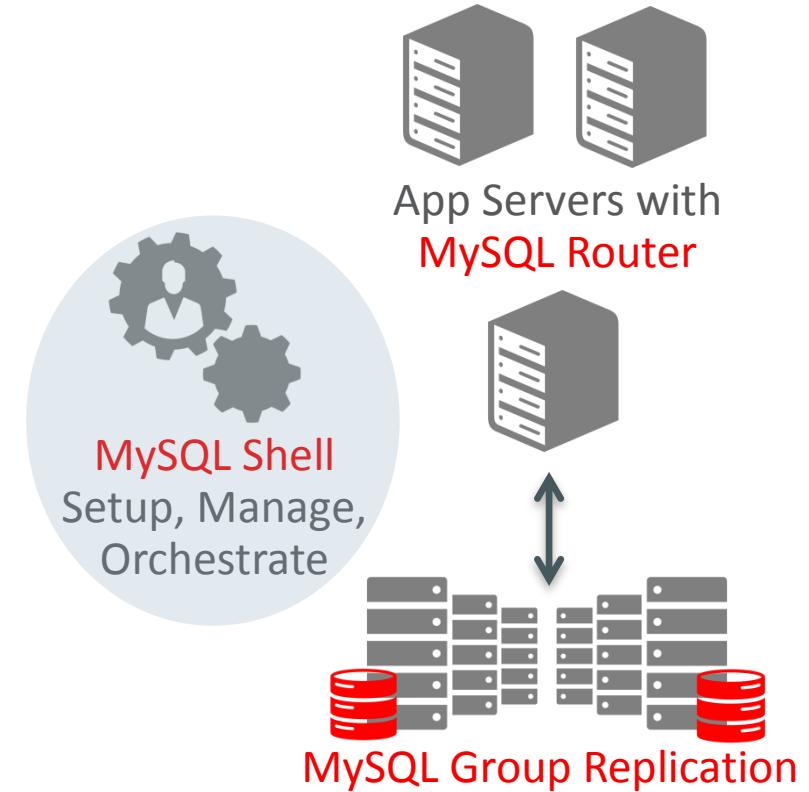
"MySQL Shell provides the developer and DBA with a single intuitive, flexible, and powerfull interface for all MySQL related tasks!"



MySQL Shell: Admin API

Database Administration Interface

- mysql> dba.help()
- The global variable 'dba' is used to access the MySQL AdminAPI
- Perform DBA operations
 - Manage MySQL InnoDB clusters
 - Create clusters
 - Deploy MySQL instances
 - Get cluster info
 - Start/Stop MySQL Instances
 - Validate MySQL instances ...



MySQL Shell – Deploy MySQL Instances

```
shell> mysqlsh
```

```
mysql-js> dba.deployLocalInstance(3306)
```

```
mysql-js> dba.deployInstance('hanode2:3306')
```

```
mysql-js> dba.deployInstance('hanode3:3306')
```

MySQL Shell – Create an InnoDB Cluster

```
mysql-js> \connect root@hanode1:3306
```

```
mysql-js> cluster = dba.createCluster('NewAppCluster')
```

```
mysql-js> cluster.addInstance('root@hanode2:3306')
```

```
mysql-js> cluster.addInstance('root@hanode3:3306')
```

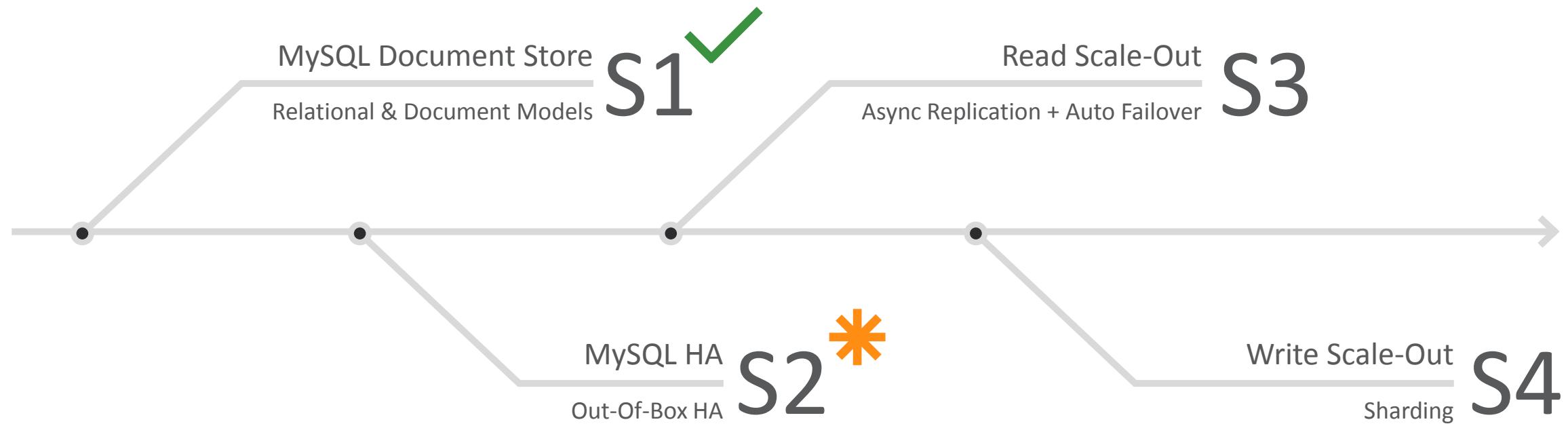
MySQL Shell – Add a MySQL Router

```
shell> mysqlrouter --bootstrap hanode1:3306
```

```
shell> mysqlrouter &
```

```
shell> mysqlsh --uri root@localhost:6446
```

What's Next



* Download InnoDB Cluster Preview Release from labs.mysql.com

MySQL Enterprise Monitor

- Native holistic support for InnoDB clusters
 - Topology views
 - Detailed metrics and graphs
 - Best Practice advice
- Monitoring of MySQL Routers
- Monitoring of Group Replication



The logo features the MySQL logo (blue "My" and orange "SQL™") followed by "NDB Cluster". Above the "My" in the MySQL logo is a stylized blue dolphin leaping out of the water.

Driving Database Requirements

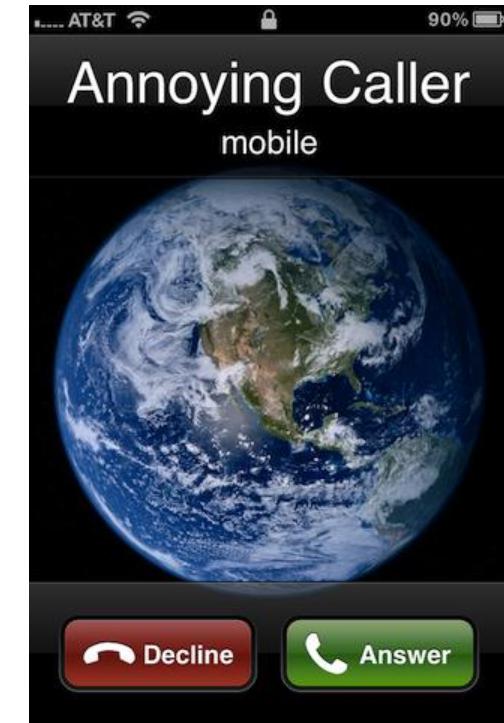
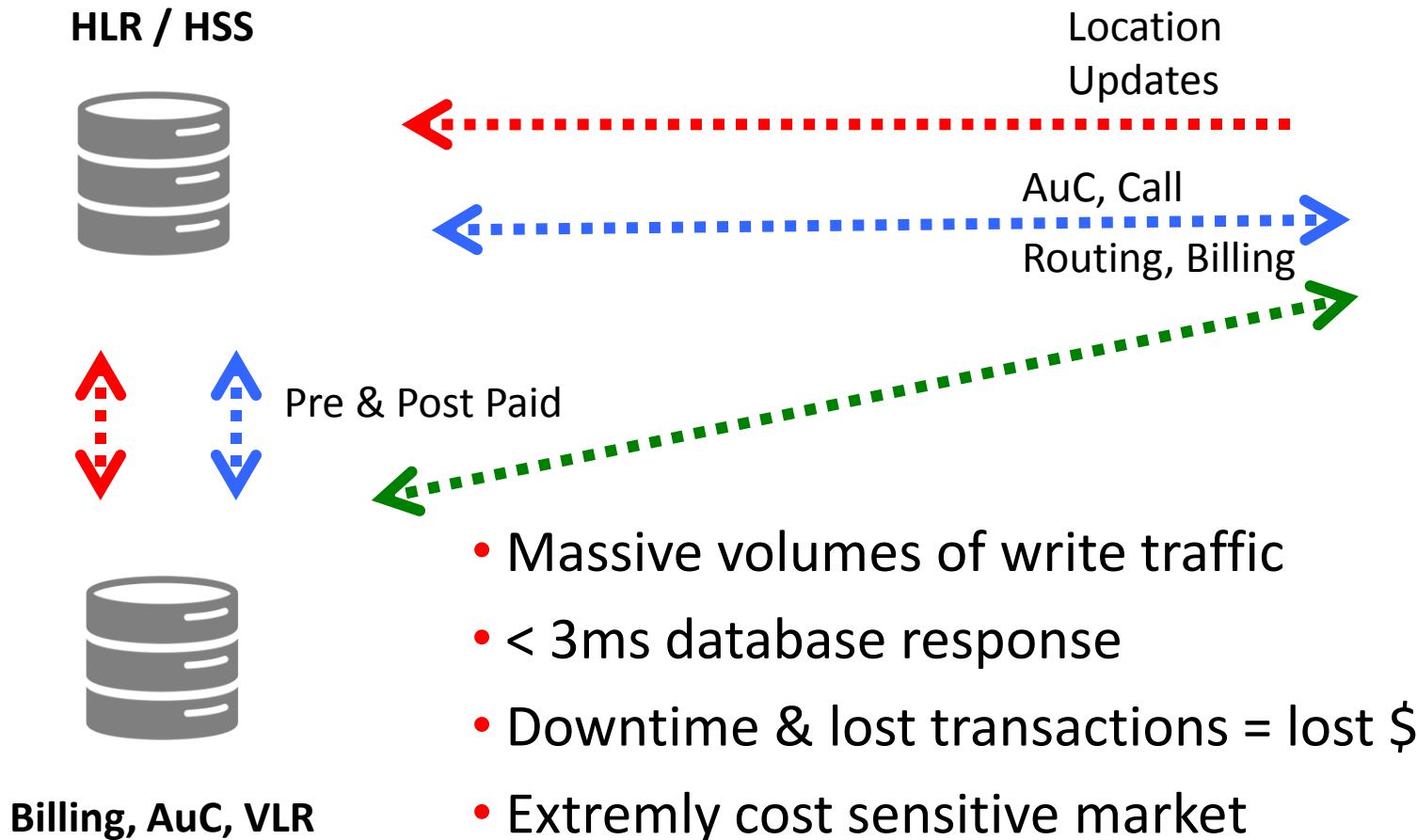
In-Memory
Real-Time
Performance

Extreme
Read & Write
Scalability

Rock Solid
Availability

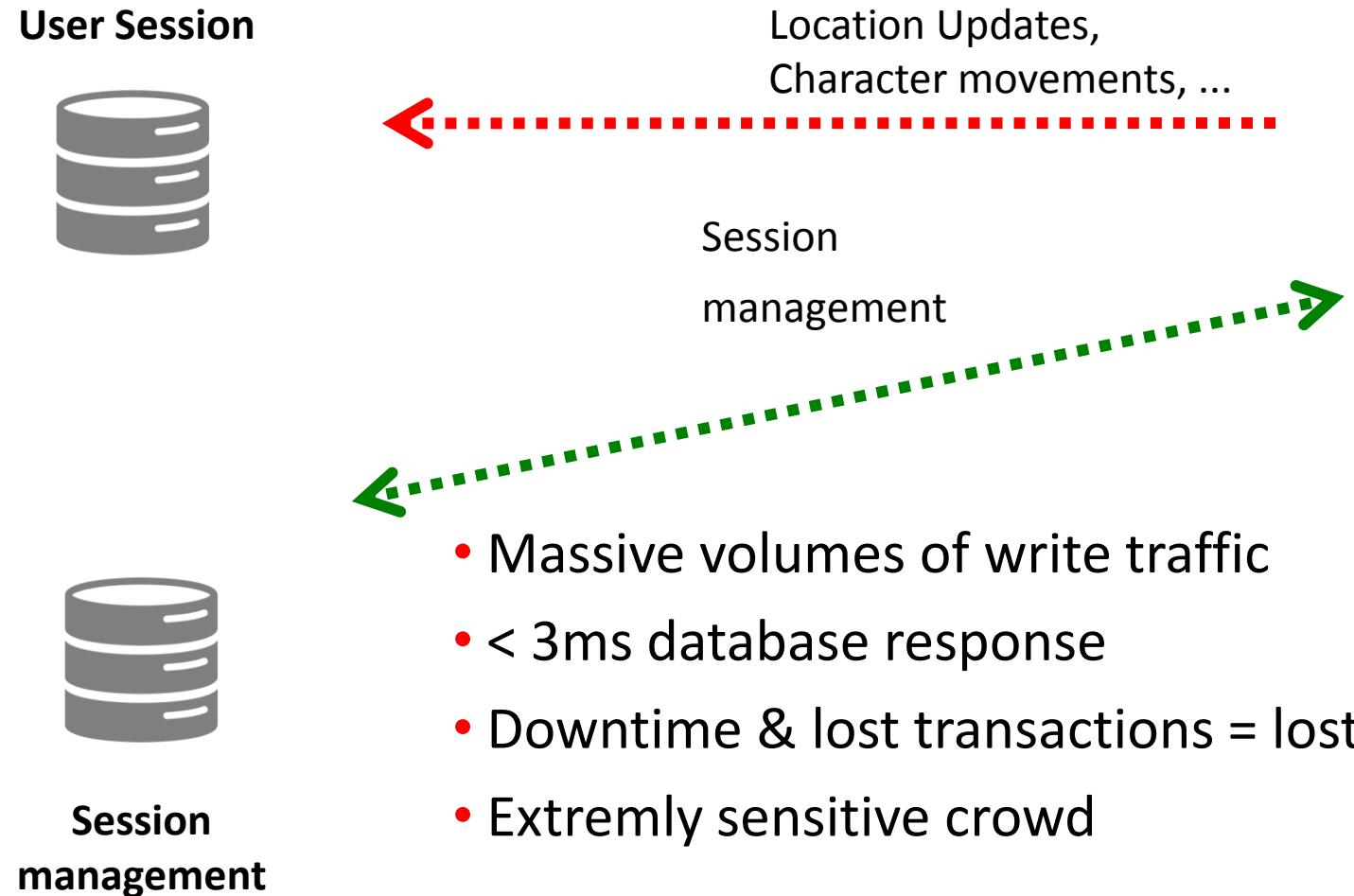
Elasticity

No Trade-Offs: Cellular Network



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

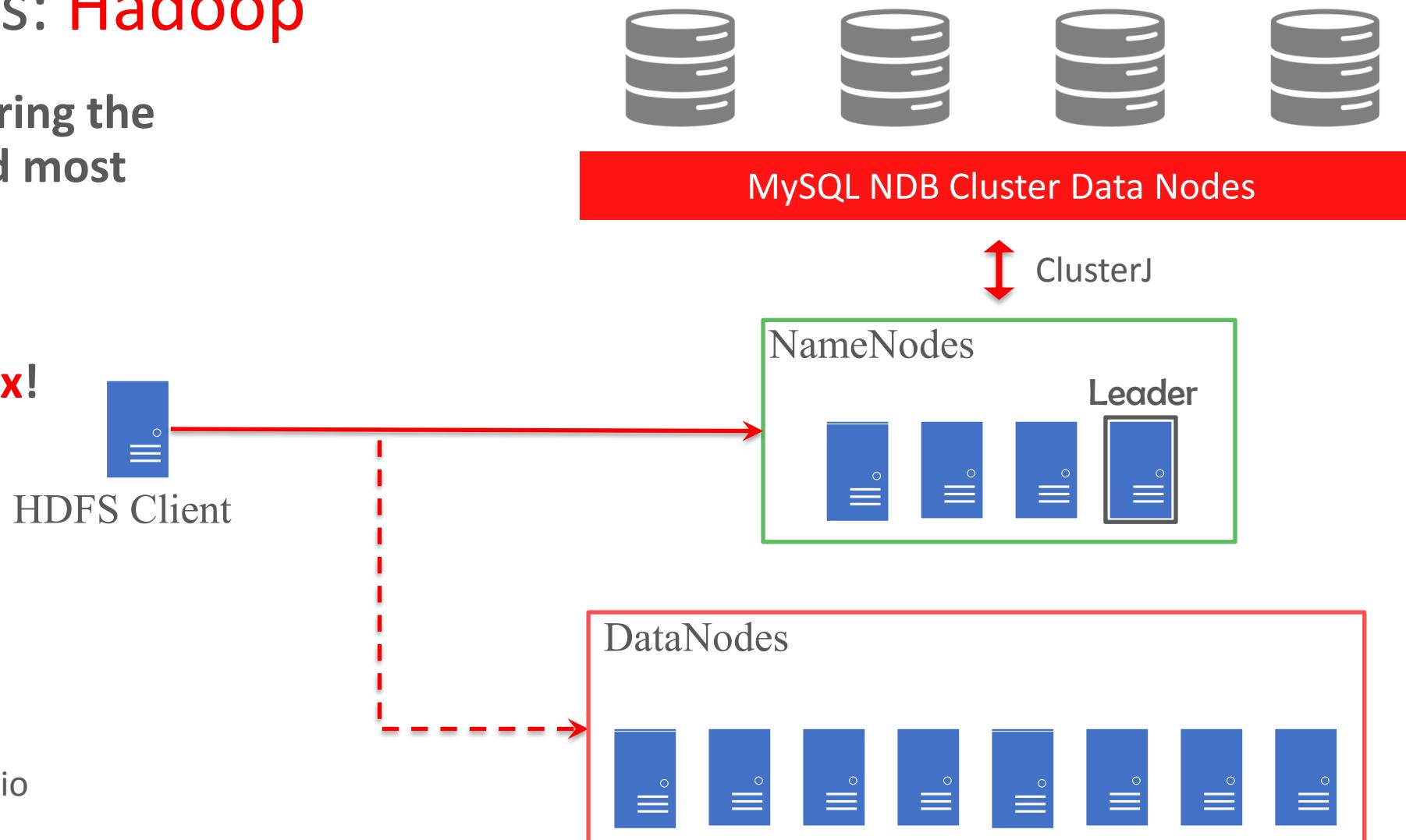
No Trade-Offs: Massive Parallel Online Games



No Trade-offs: Hadoop

NDB Cluster powering the world's fastest and most scalable Hadoop filesystem.

Scaling Hadoop **20x!**



MySQL Cluster Overview

REAL-TIME

- Memory optimized tables with durability
- Predictable Low-Latency, Bounded Access Time

HIGH SCALE, READS + WRITES

- Auto-Sharding, Active-Active
- ACID Compliant, OLTP + Real-Time Analytics

FULLY ELASTIC

- Add and remove storage and performance capacity in seconds
- Fully cloud enabled

99.999% AVAILABILITY

- Active-Active, Shared nothing, no Single Point of Failure
- Self Healing + On-Line Operations

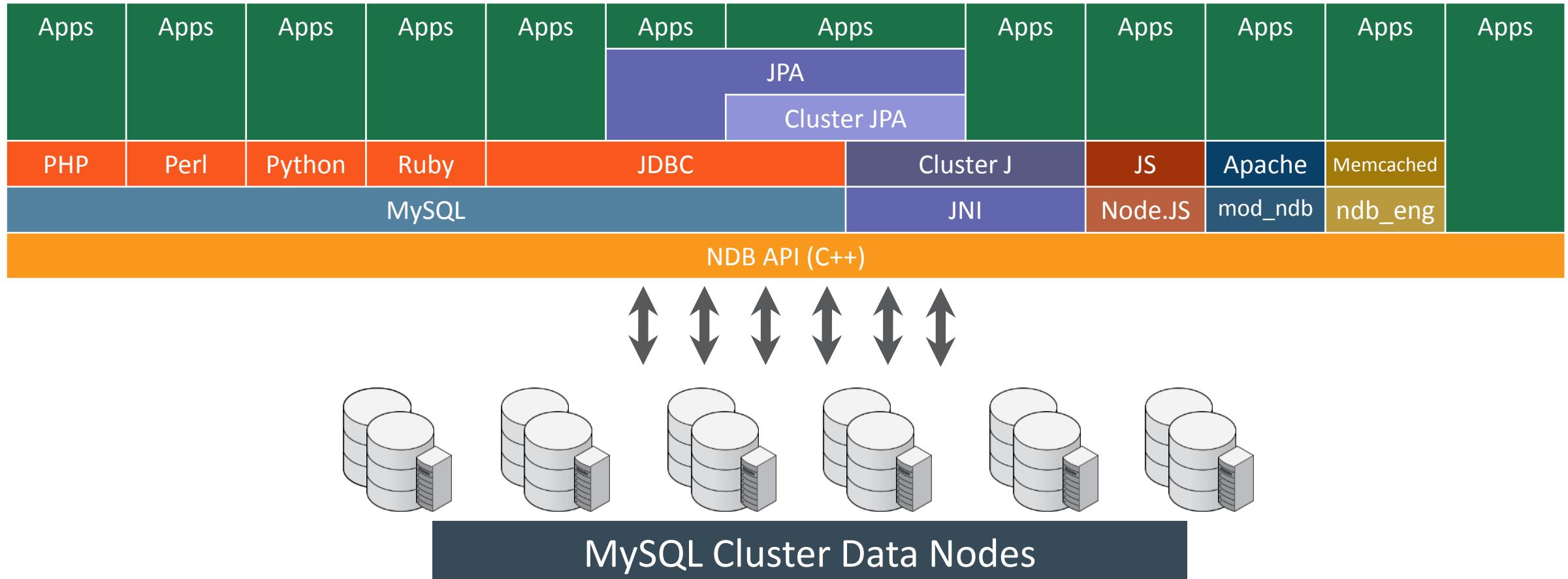
SQL + NoSQL

- Key/Value + Complex, Relational Queries
- SQL + Memcached + JavaScript + Java + HTTP/REST & C++

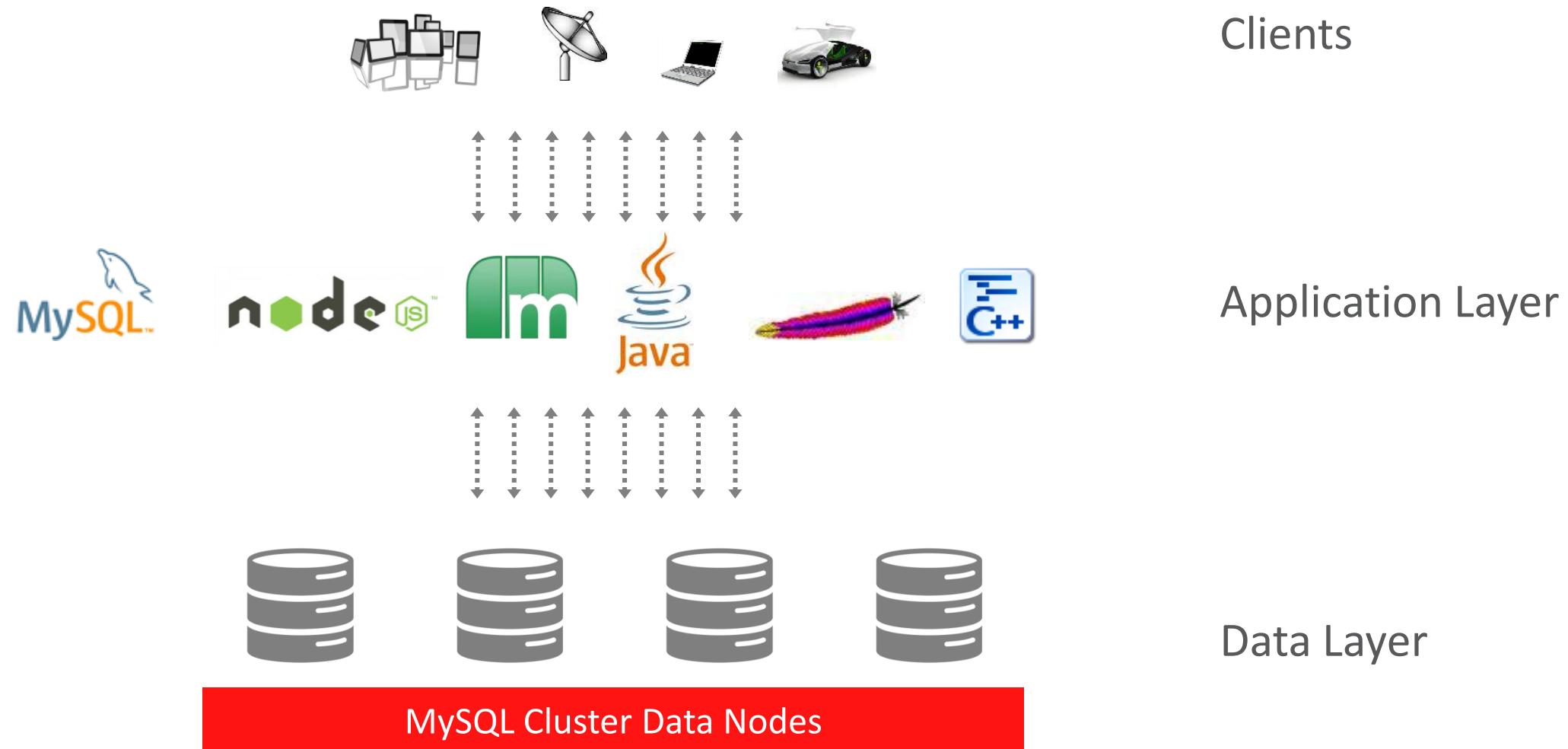
LOW TCO

- Open Source + Commercial Editions, Easy to use and deploy
- Commodity hardware + Management, Monitoring Tools

NoSQL Access to MySQL Cluster data

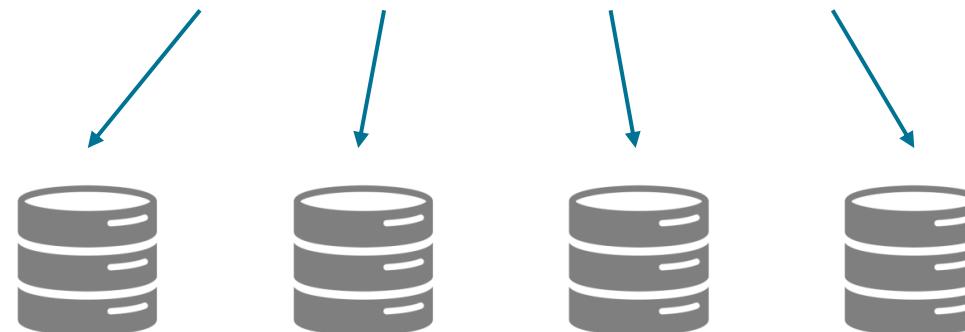


MySQL Cluster Architecture



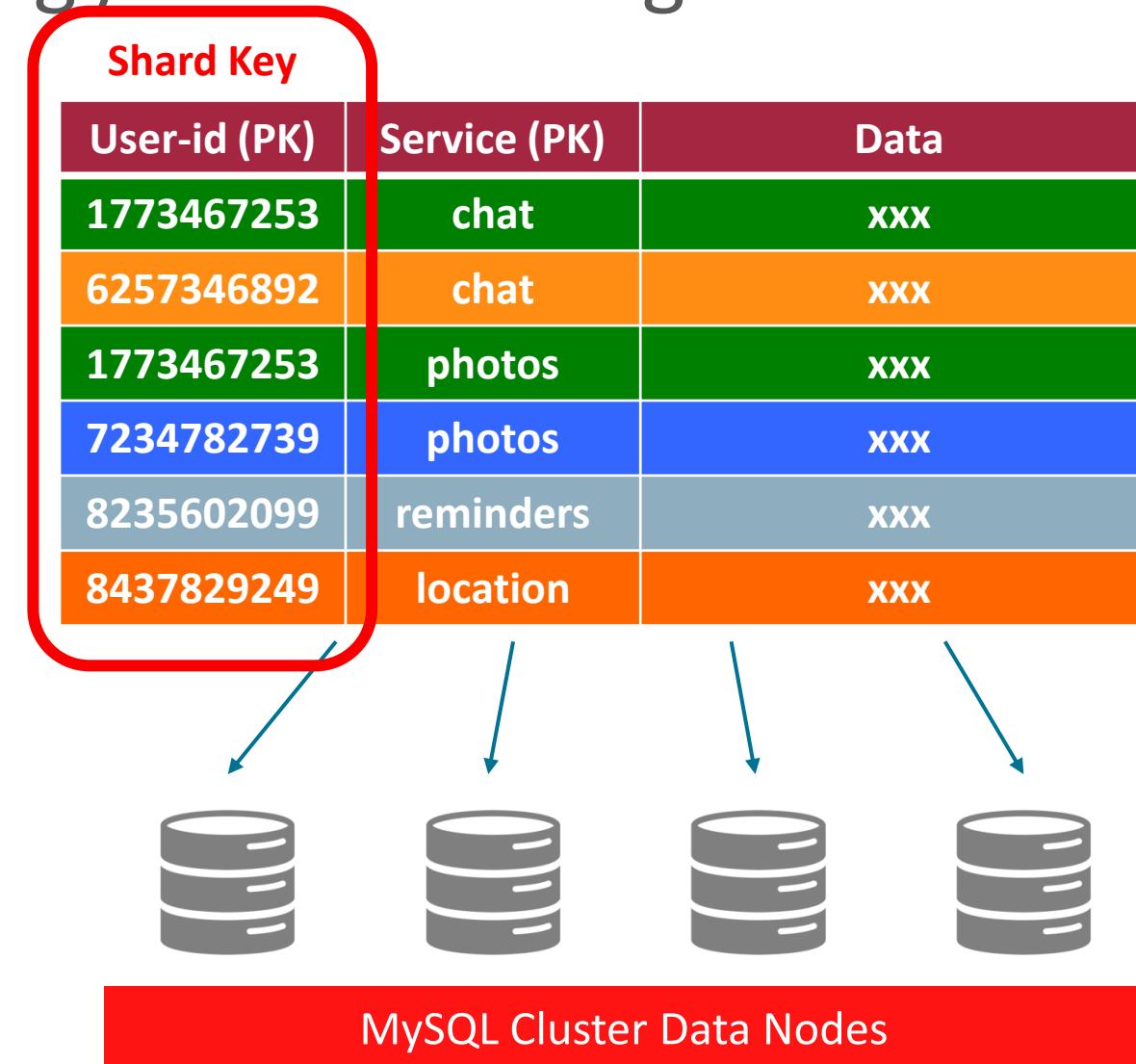
Data Partitioning / Auto-sharding

User-id (PK)	Service (PK)	Data
1773467253	chat	xxx
6257346892	chat	xxx
1773467253	photos	xxx
7234782739	photos	xxx
8235602099	reminders	xxx
8437829249	location	xxx

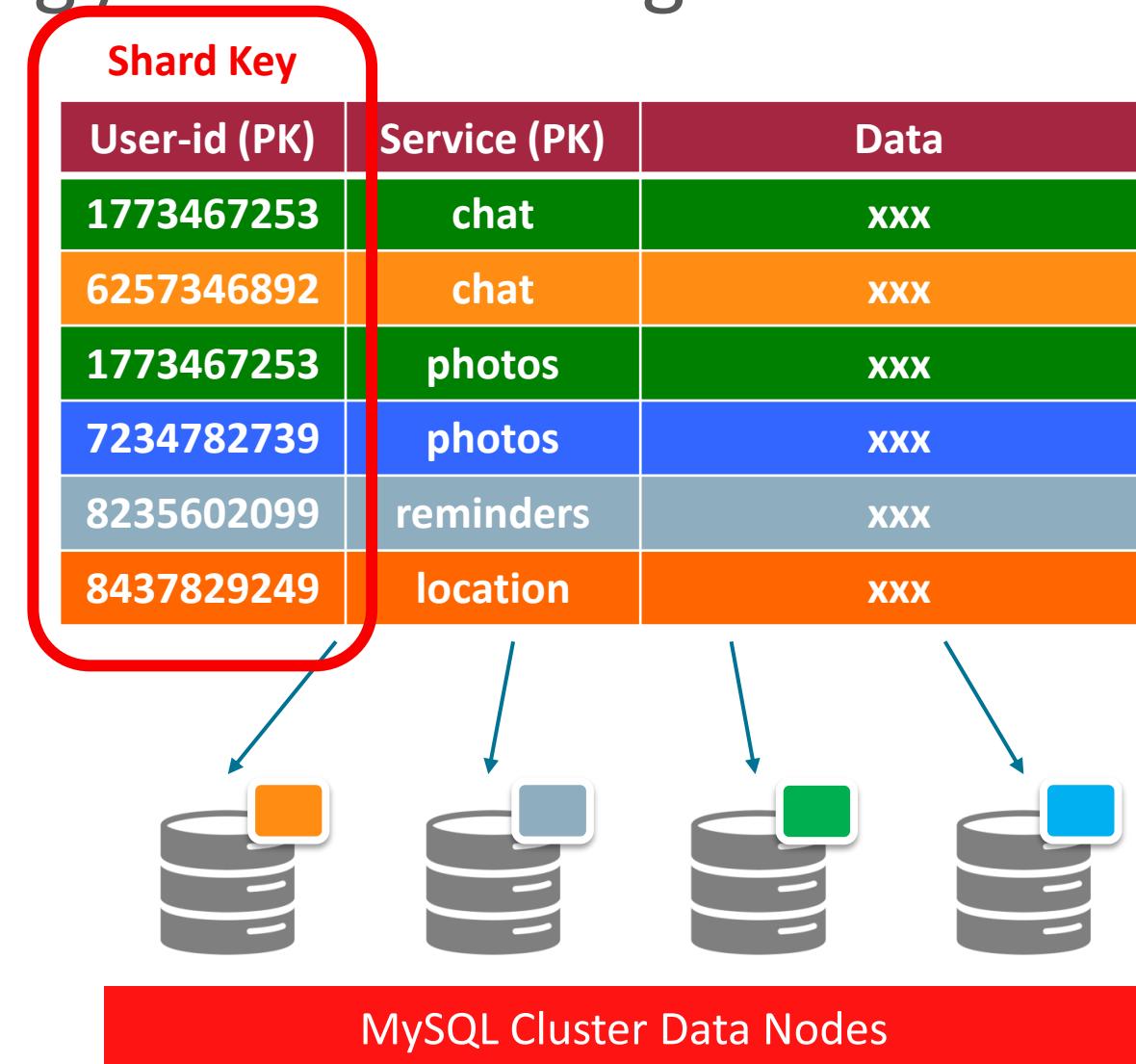


MySQL Cluster Data Nodes

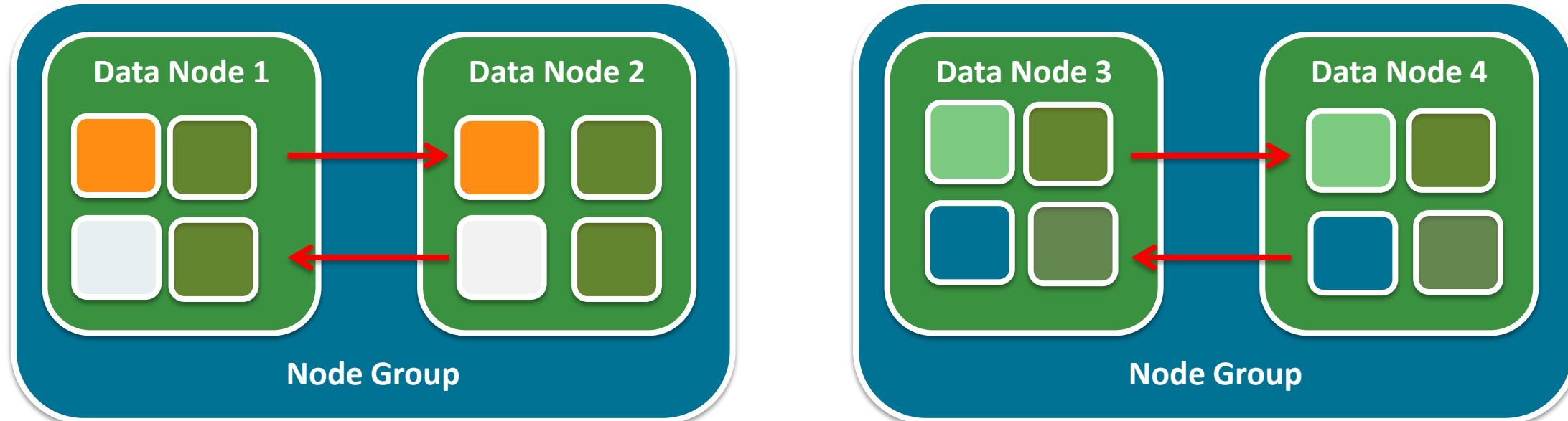
Data Partitioning / Auto-sharding



Data Partitioning / Auto-sharding



Data Partitioning / Auto-sharding

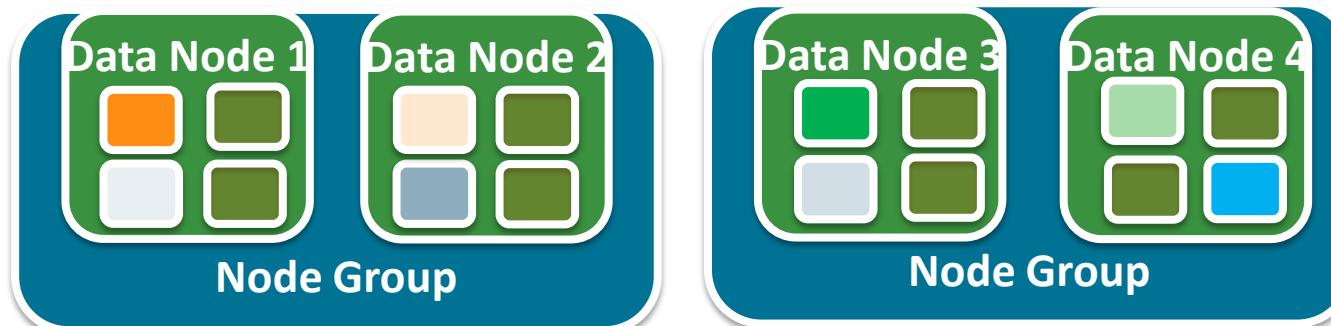


- Data transparently sharded between Node Groups
- Each fragment active in one Data Node with synchronous replication to 2nd Data Node in same Node Group

Data Partitioning / Auto-sharding

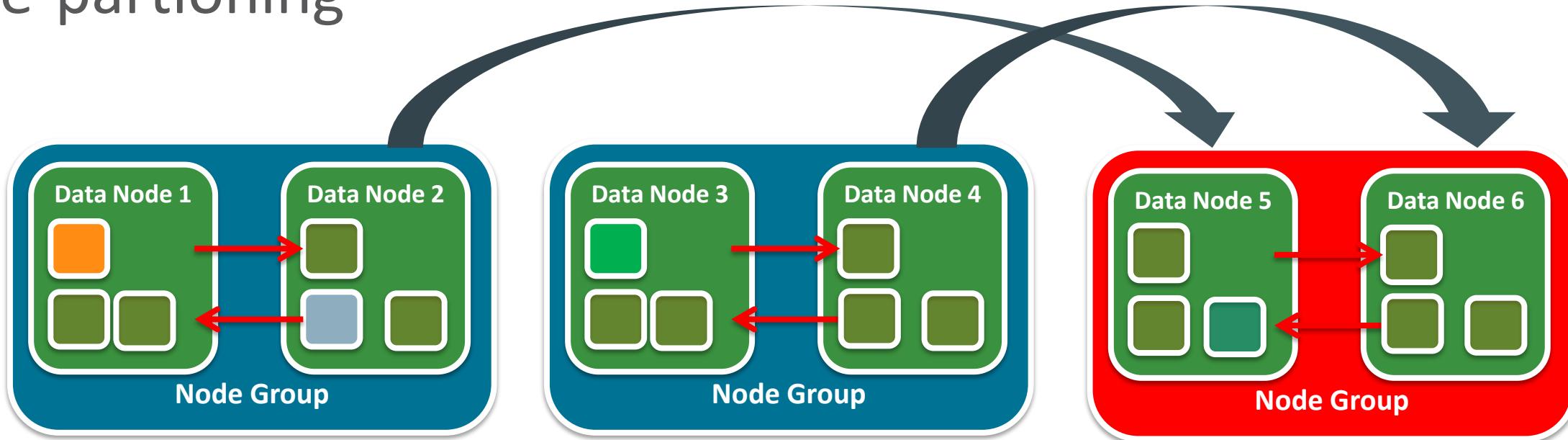
Shard Key		
User-id (PK)	Service (PK)	Data
1773467253	chat	xxx
6257346892	chat	xxx
1773467253	photos	xxx
7234782739	photos	xxx
8235602099	reminders	xxx
8437829249	location	xxx

- DBA chooses which part of Primary Key to use as shard key
- Fragment for each row decided by hashing the sharding key



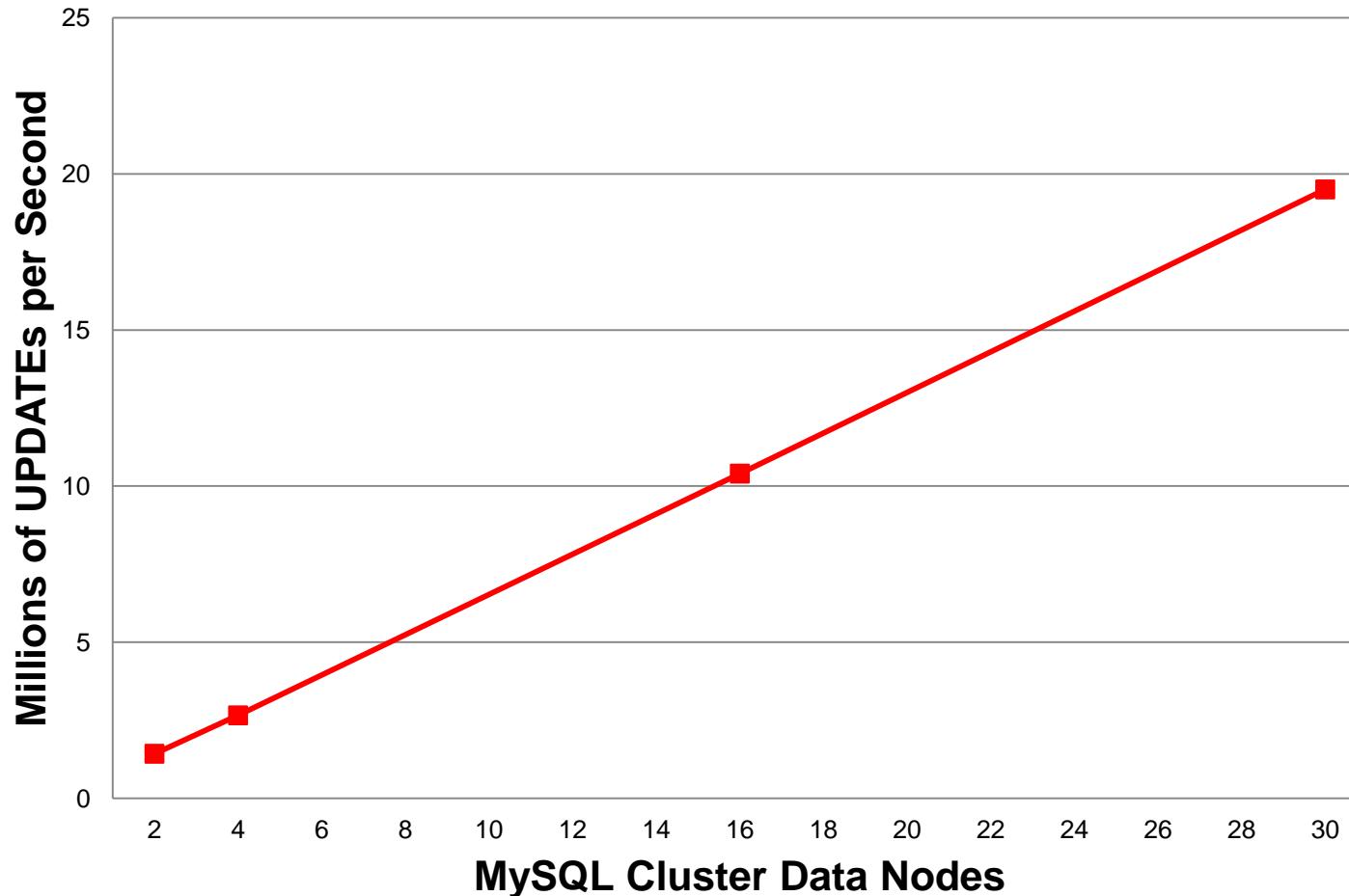
On-line Scaling and Elasticity

Re-partitioning



- Data automatically rearranged to use new capacity
- Designed to be a slow background process not impacting real-time performance.

MySQL Cluster 7.2: 1.2 Billion UPDATEs per Minute



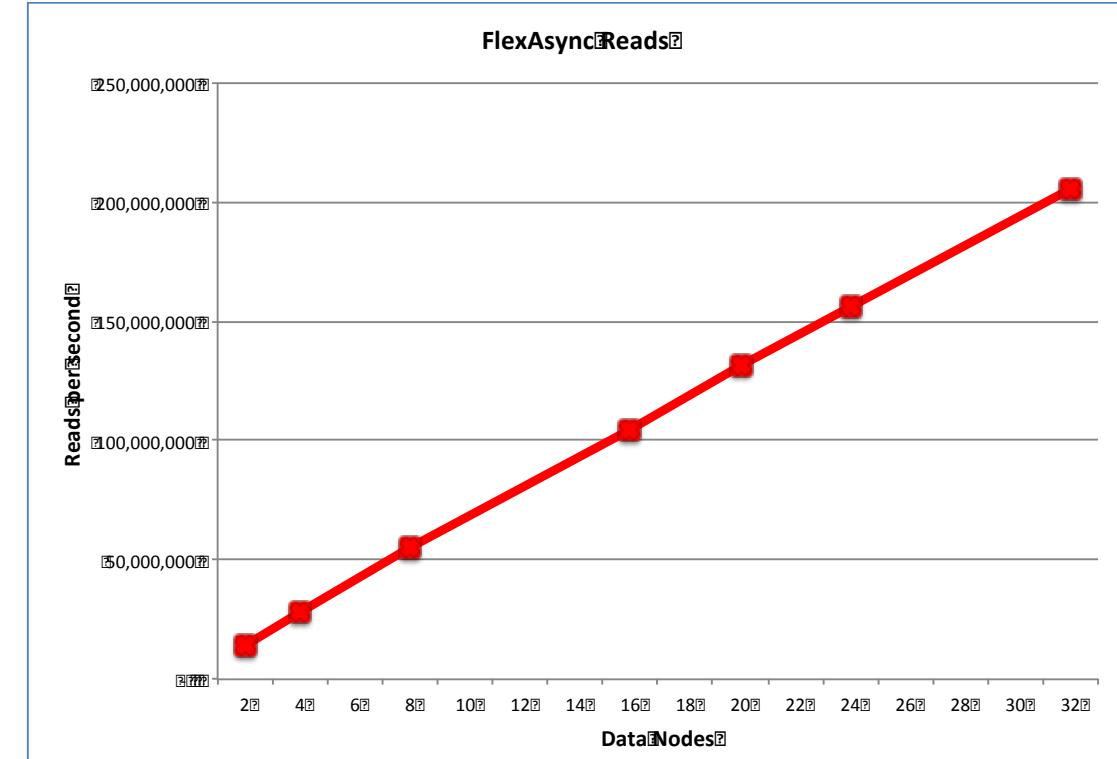
- NoSQL C++ API,
flexaSynch benchmark
- 30 x Intel E5-2600 Intel
Servers, 2 socket, 64GB
- ACID Transactions, with
Synchronous Replication

<http://www.mysql.com/why-mysql/white-papers/mysql-cluster-benchmarks-1-billion-writes-per-minute/>

MySQL Cluster 7.4 NoSQL Performance

200 Million NoSQL Reads/Second

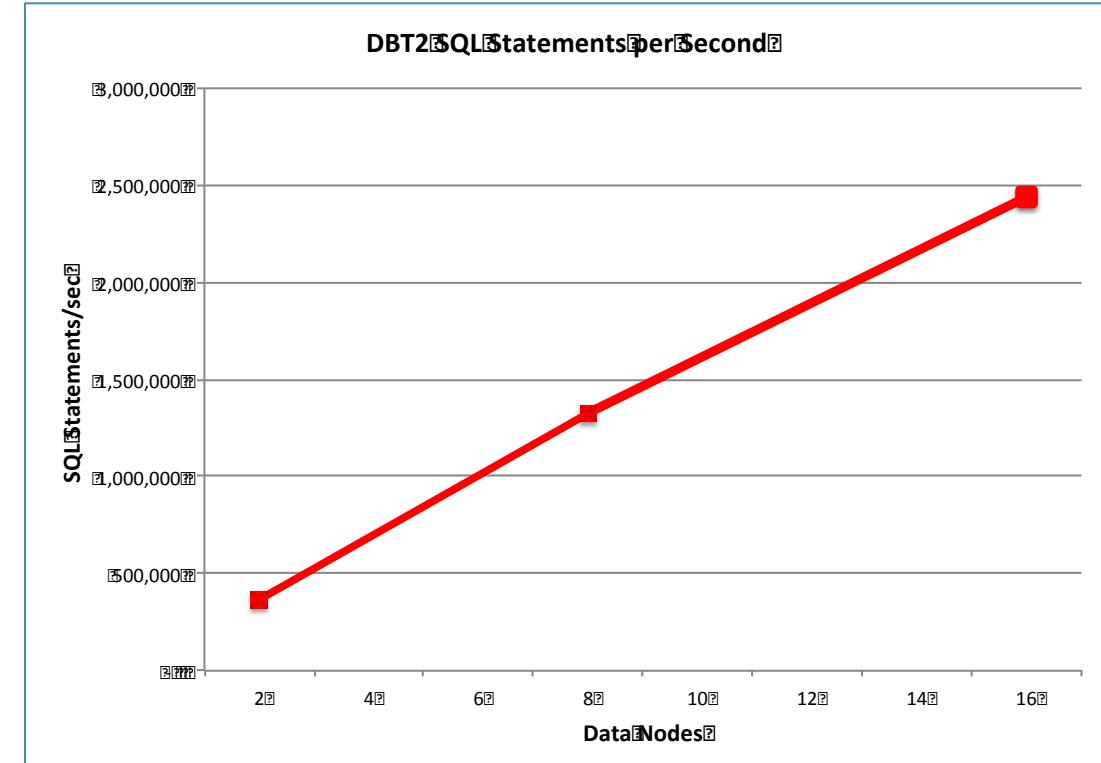
- Memory optimized tables
 - Durable
 - Mix with disk-based tables
- Massively concurrent OLTP
- Distributed Joins for analytics
- Parallel table scans for non-indexed searches
- MySQL Cluster 7.4 FlexAsynch
 - 200M NoSQL Reads/Second



MySQL Cluster 7.4 SQL Performance

2.5M SQL Statements/Second

- Memory optimized tables
 - Durable
 - Mix with disk-based tables
- Massively concurrent OLTP
- Distributed Joins for analytics
- Parallel table scans for non-indexed searches
- MySQL Cluster 7.4 DBT2 BM
 - 2.5M SQL Statements/Second



MySQL NDB Cluster 7.5

- Read Optimized Tables
- Fully Replicated

Capacity and Scale Out



- MySQL 5.7
- JSON Data Type
- Generated Columns
- Records-Per-Key Optimization

Improved SQL

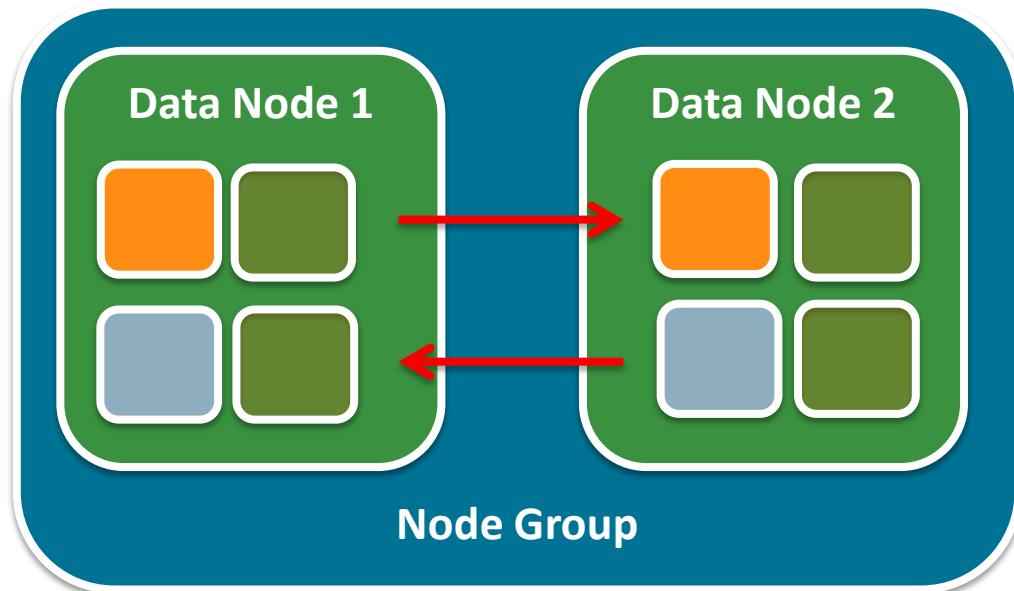


- Improved Reporting
- Improved Logging
- Improved Visibility
- Improved Restore

Management

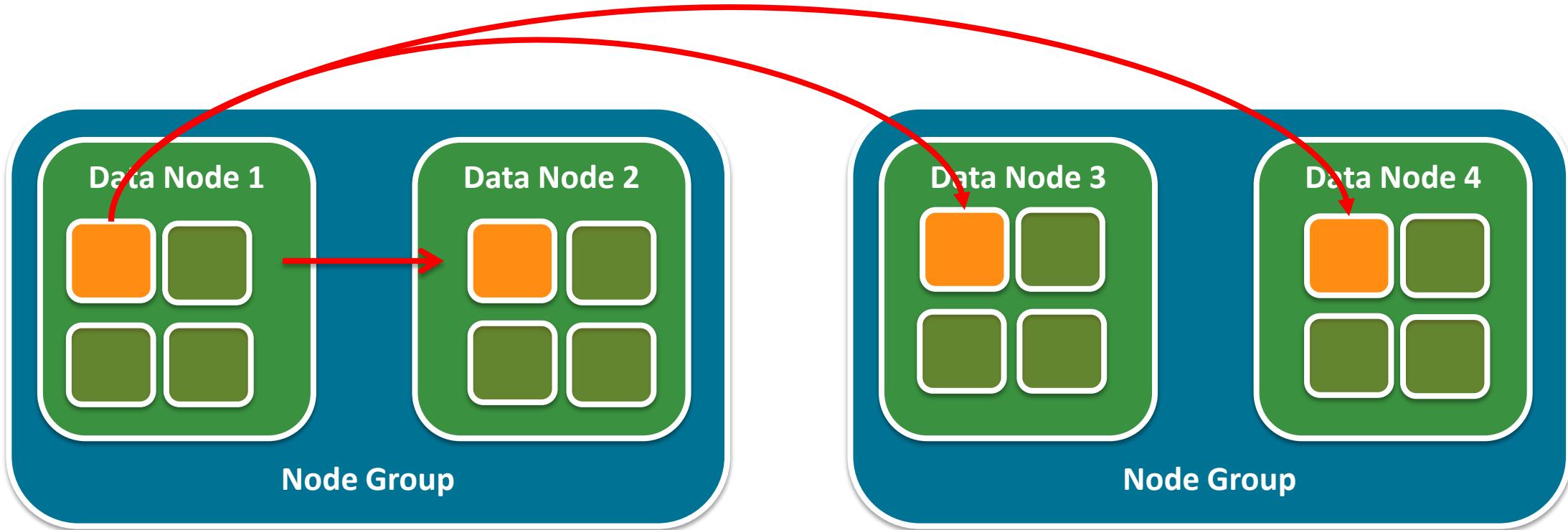


MySQL NDB 7.5: Reading from backup



- Reading from backup allows to read from any copy
- Previously all reads were directed towards the primary fragment only

MySQL NDB Cluster 7.5: Fully replicated



- Fully replicated allows a table to be read/written locally on any node
- Ideal for static data, faster join performance

Next Steps



Learn More

- www.mysql.com/cluster
- <https://dev.mysql.com/doc/mysql-innodb-cluster/en/>
- Authentic MySQL Curriculum: <http://oracle.com/education/mysql>



Try it Out

- dev.mysql.com/downloads/cluster/
7.4 GA and 7.5 DMR 1
- <https://labs.mysql.com/>



Let us know what you think

- <http://mysqlhighavailability.com/>
- forums.mysql.com/list.php?25



MySQL™ InnoDB Cluster

or

MySQL™ NDB Cluster

How Do The Two Compare I

- MySQL **InnoDB** Cluster
 - Easy HA built into MySQL 5.7+
 - Write consistency
 - Read Scalability
 - App failover using MySQL Router
 - Management via mysqlsh
 - InnoDB storage engine
- MySQL **NDB** Cluster
 - In memory database
 - Automatic sharding
 - Native access via several API
 - Read/write consistency
 - Read/write scalability
 - NDB storage engine

How Do The Two Compare II

	MySQL InnoDB Cluster	MySQL NDB Cluster
Storage Engine	InnoDB	NDBCLUSTER
Distributed Architecture	Shared nothing	Shared nothing
Consistency Model	Weak Consistency	Strong Consistency
Sharding	No	Yes
Arbitration	No	Yes
Load Balancing	No	Yes
NoSQL APIs	MySQL Document Store	Native NDB API
Operational Complexity	Medium	High
Administration	Standard (MySQL)	Custom (MySQL + NDB)

Thank you!

