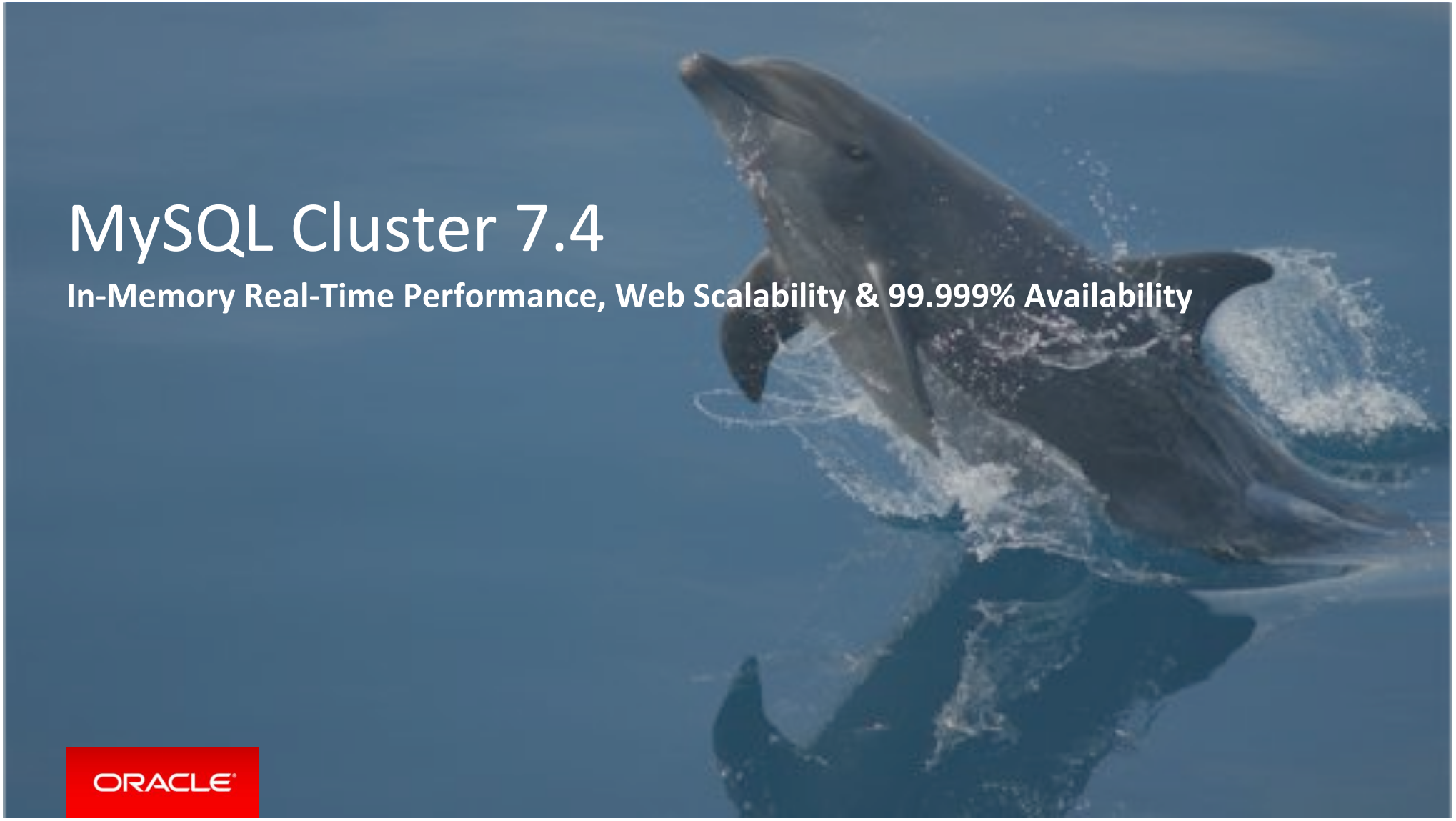

MySQL Cluster 7.4
In-Memory Real-Time Performance, Web Scalability & 99.999% Availability
ORACLE

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.

MySQL Cluster 7.4 GA

- 200 Million NoSQL Reads/Sec
- 2.5M SQL Ops/Sec
- 50% Faster Reads
- 40% Faster Mixed

Performance

- Active-Active Geographic Redundancy
- Conflict Detection/ Resolution

Active-Active

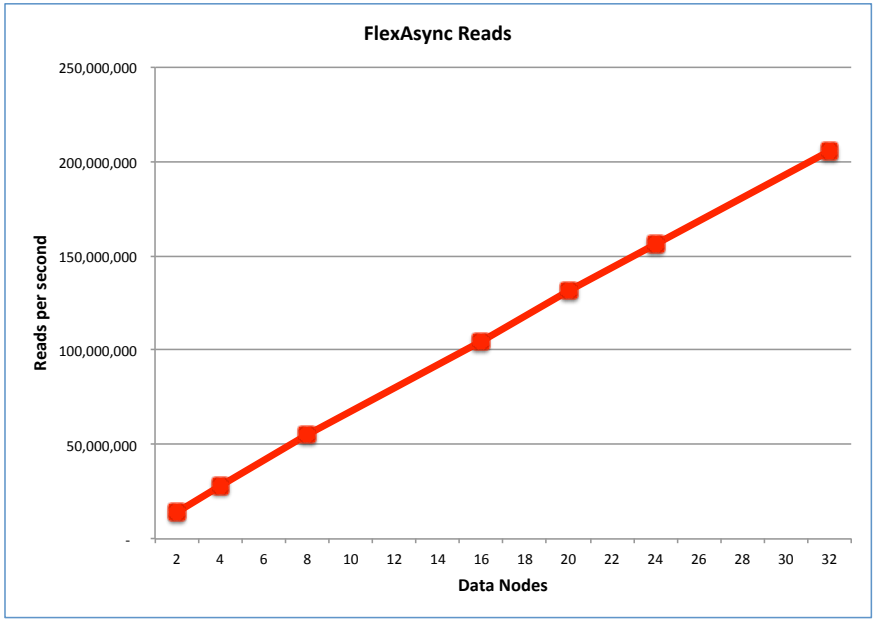
- 5X Faster Maintenance Ops
- Detailed Reporting

Management

ORACLE

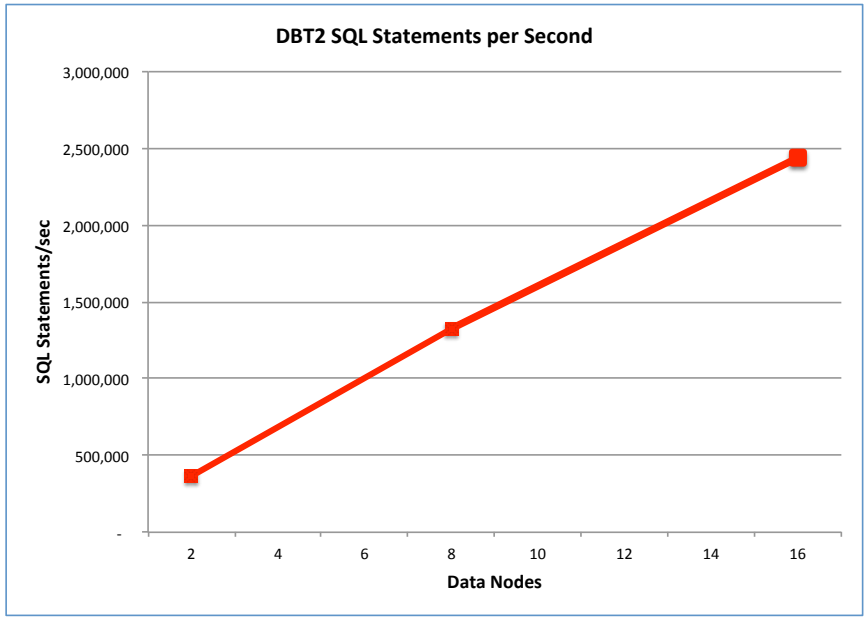
MySQL Cluster 7.4 NoSQL Performance
200 Million NoSQL Reads/Second

- 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 FlexAsych
  - 200M NoSQL Reads/Second



MySQL Cluster 7.4 SQL Performance
2.5M SQL Statements/Second

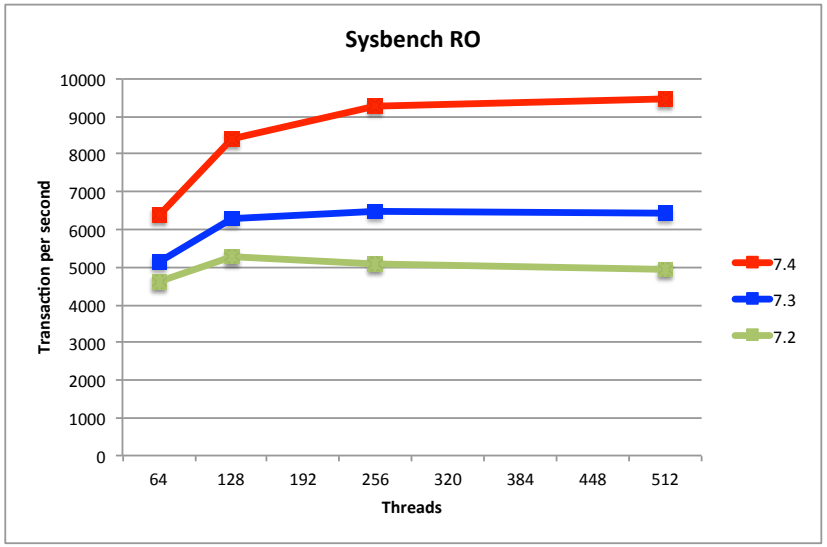
- Memory optimized tables
  - Durable
  - Mix with disk-based tables
- Massively concurrent OLTP
- Distributed Joins for analytics
- Parallel table scans for non-indexed searches
- MySQL Cluster 7.4 DBT2 BM
  - 2.5M SQL Statements/Second



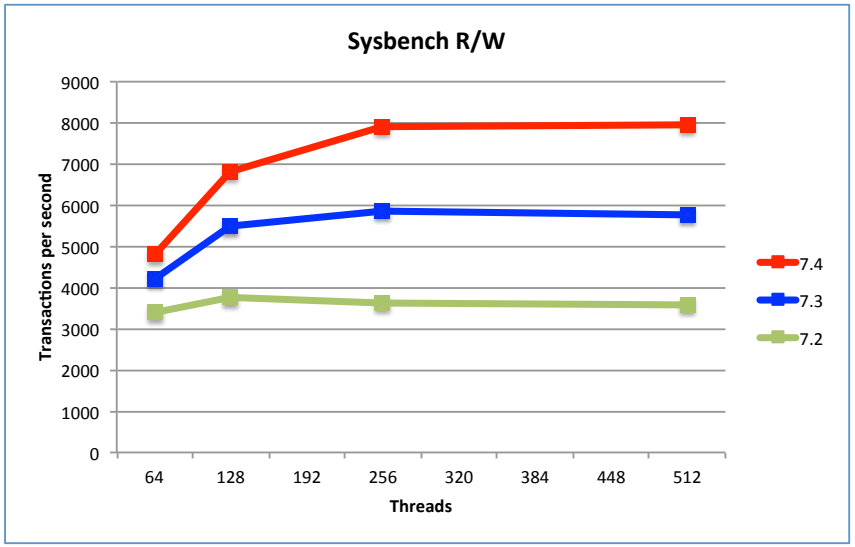
ORACLE

MySQL Cluster 7.4

Performance Enhancements



50% Read-Only Increase



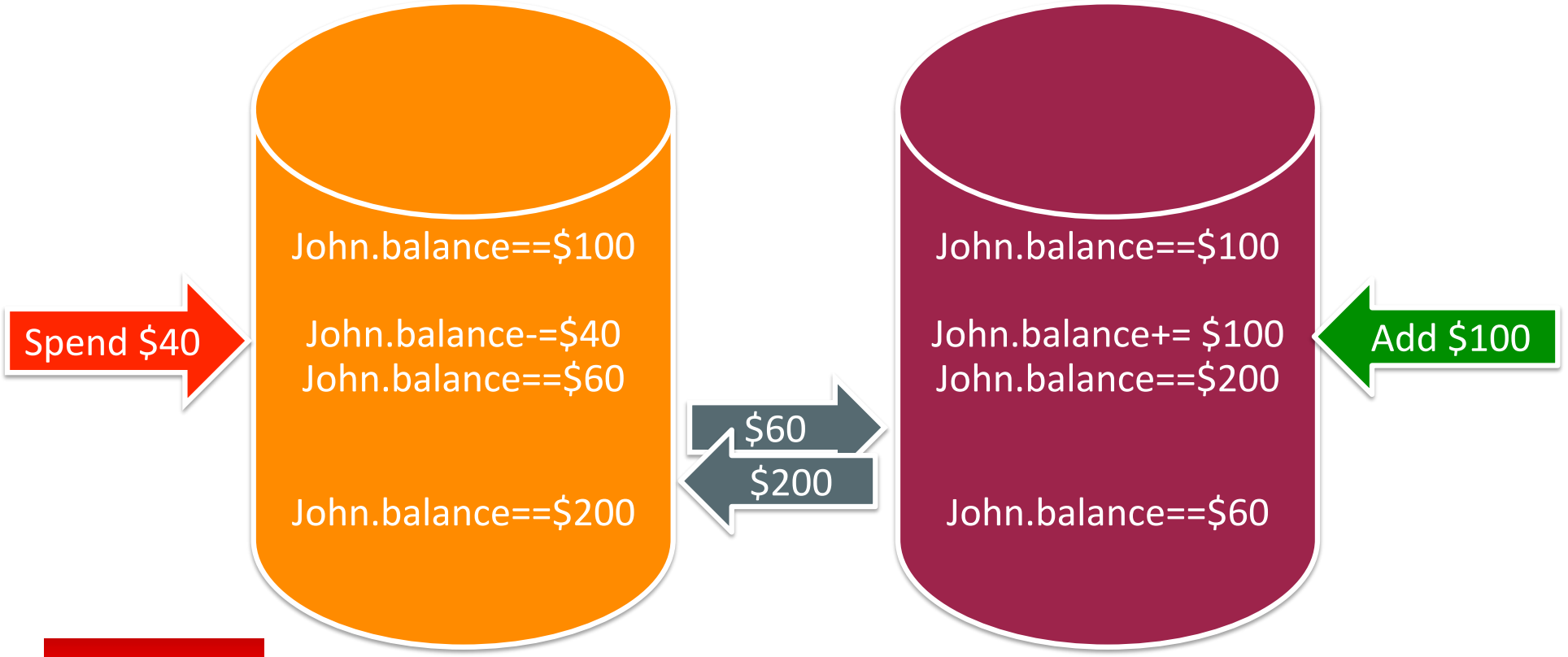
40% Read/Write Increase

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

ORACLE

What is a conflict?

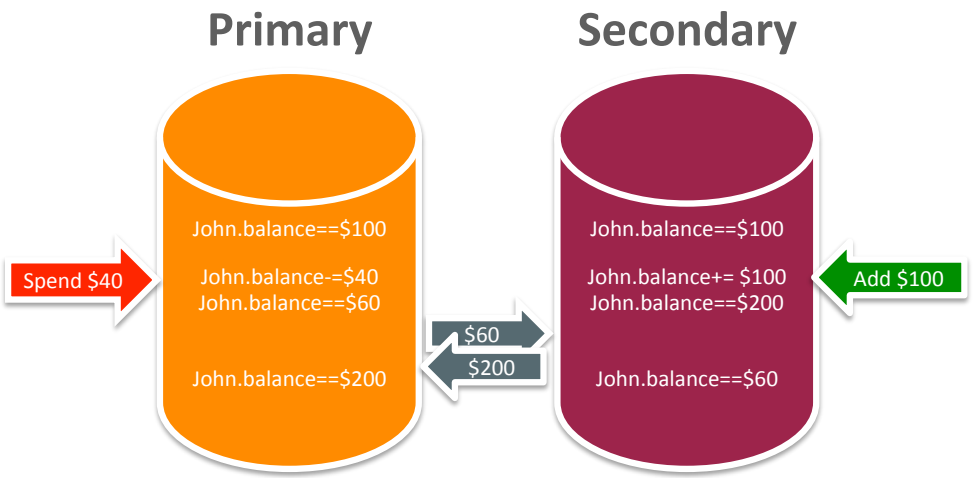


Handling of Conflicts – Extensions in MySQL Cluster 7.4

- NDB$EPOCH2 and NDB $EPOCH2_TRANS introduced
- Detects conflicting inserts/updates/ deletes/reads
- Entire transactions (and dependent ones) rolled back

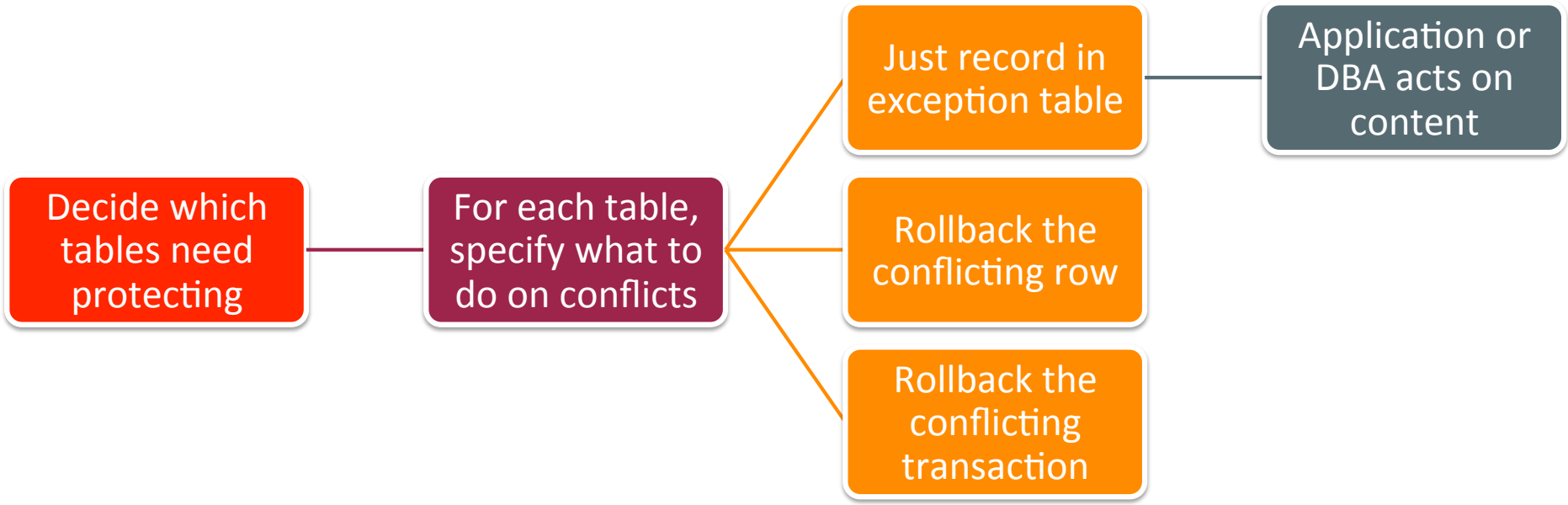
- Rolling back of transactions that read conflicted data
- Improved NDB Exceptions table format
  - Non-PK columns, operation type, transaction id, before and after values
- Online conflict role change

Detecting Conflicts - Reflected GCI



- Primary stores 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



MySQL Cluster 7.4 Restart Improvements

Make Data Node Restarts Fast!

- Duration of long-running maintenance activities dominated by Data Node restart times
- MySQL Cluster 7.4 = **5.5X faster restarts**
- Benefits both nodal and rolling restarts
  - Upgrades, add-node,…
- Benefits both SQL and NoSQL APIs
- Benefits both ”manual” a MySQL Cluster Manager operations
- Achieve 5x as much during a single maintenance window

MySQL Cluster 7.4 Restart Improvements

Observability improvements

- Verbose logging
  - Task start/completion
  - Data volumes
  - Parallelism & Wait times
- NDBINFO for recent node restarts
- More documentation of stages
- Goal: Make analysis of a slow restart possible
  - Determine cause; Detect patterns; Understand the impact of indexes, local checkpoints etc.


```
2014-11-12 16:30:47 [ndbd] INFO     -- Start phase 0 completed
2014-11-12 16:30:47 [ndbd] INFO     -- Phase 0 has made some file system initialisations
2014-11-12 16:30:47 [ndbd] INFO     -- Starting QMGR phase 1
2014-11-12 16:30:47 [ndbd] INFO     -- DIH reported initial start, now starting the Node Inclusion Protocol
2014-11-12 16:30:50 [ndbd] INFO     -- findNeighbours from: 2220 old (left: 65535 right: 65535) new (3 1)
2014-11-12 16:30:50 [ndbd] INFO     -- Include node protocol completed, phase 1 in QMGR completed
2014-11-12 16:30:50 [ndbd] INFO     -- Start phase 1 completed
2014-11-12 16:30:50 [ndbd] INFO     -- Phase 1 initialised some variables and included node in cluster, locked memory if configured to do so
2014-11-12 16:30:50 [ndbd] INFO     -- Asking master node to accept our start (nodeId = 1 is master), GCI = 0
2014-11-12 16:30:50 [ndbd] INFO     -- findNeighbours from: 2132 old (left: 3 right: 1) new (3 2)
2014-11-12 16:30:50 [ndbd] INFO     -- NDBCNTR master accepted us into cluster, start NDB start phase 1
2014-11-12 16:30:50 [ndbd] INFO     -- We are performing initial start of cluster
2014-11-12 16:30:50 [ndbd] INFO     -- LDM instance 1: Starting REDO log initialisation
2014-11-12 16:30:50 [ndbd] INFO     -- LDM instance 4: Starting REDO log initialisation
2014-11-12 16:30:50 [ndbd] INFO     -- LDM instance 2: Starting REDO log initialisation
2014-11-12 16:30:50 [ndbd] INFO     -- LDM instance 3: Starting REDO log initialisation
2014-11-12 16:30:50 [ndbd] INFO     -- LDM instance 3: Completed REDO log initialisation of logPart = 0
2014-11-12 16:30:50 [ndbd] INFO     -- LDM instance 3: Completed REDO initialisation
2014-11-12 16:30:50 [ndbd] INFO     -- LDM instance 1: Completed REDO log initialisation of logPart = 0
2014-11-12 16:30:50 [ndbd] INFO     -- LDM instance 1: Completed REDO initialisation
2014-11-12 16:30:50 [ndbd] INFO     -- LDM instance 4: Completed REDO log initialisation of logPart = 0
2014-11-12 16:30:50 [ndbd] INFO     -- LDM instance 4: Completed REDO initialisation
2014-11-12 16:30:50 [ndbd] INFO     -- LDM instance 2: Completed REDO log initialisation of logPart = 0
2014-11-12 16:30:50 [ndbd] INFO     -- LDM instance 2: Completed REDO initialisation
2014-11-12 16:30:50 [ndbd] INFO     -- Schema file initialisation Starting
2014-11-12 16:30:50 [ndbd] INFO     -- Schema file initialisation Completed
2014-11-12 16:30:51 [MgmtSrvr] INFO     -- Node 1: Make On-line Database recoverable by waiting for LCP Starting, LCP id = 1
2014-11-12 16:30:52 [MgmtSrvr] INFO     -- Node 1: Local checkpoint 1 started. Keep GCI = 1 oldest restorable GCI = 1
2014-11-12 16:30:52 [MgmtSrvr] INFO     -- Node 2: LDM instance 1: Completed LCP, num fragments = 4 num records = 528, num bytes = 16976
2014-11-12 16:30:52 [MgmtSrvr] INFO     -- Node 2: LDM instance 2: Completed LCP, num fragments = 4 num records = 471, num bytes = 15152
2014-11-12 16:30:52 [MgmtSrvr] INFO     -- Node 2: LDM instance 3: Completed LCP, num fragments = 4 num records = 536, num bytes = 17232
2014-11-12 16:30:52 [MgmtSrvr] INFO     -- Node 2: LDM instance 4: Completed LCP, num fragments = 4 num records = 513, num bytes = 16496
2014-11-12 16:30:52 [MgmtSrvr] INFO     -- Node 1: LDM instance 2: Completed LCP, num fragments = 4 num records = 471, num bytes = 15152
2014-11-12 16:30:52 [MgmtSrvr] INFO     -- Node 1: LDM instance 3: Completed LCP, num fragments = 4 num records = 536, num bytes = 17232
2014-11-12 16:30:52 [MgmtSrvr] INFO     -- Node 1: LDM instance 1: Completed LCP, num fragments = 4 num records = 528, num bytes = 16976
2014-11-12 16:30:52 [MgmtSrvr] INFO     -- Node 1: LDM instance 4: Completed LCP, num fragments = 4 num records = 513, num bytes = 16496
2014-11-12 16:30:52 [MgmtSrvr] INFO     -- Node 1: Local checkpoint 1 completed
2014-11-12 16:30:52 [MgmtSrvr] INFO     -- Node 1: Make On-line Database recoverable by waiting for LCP Completed, LCP id = 1
2014-11-12 16:30:52 [MgmtSrvr] INFO     -- Node 4: Start phase 5 completed (initial start)
…
2014-11-12 16:30:52 [MgmtSrvr] INFO     -- Node 4: Start phase 6 completed (initial start)
…
2014-11-12 16:30:52 [MgmtSrvr] INFO     -- Node 4: Start phase 7 completed (initial start)
```


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)
- Upgrade/rolling restarts
  - Major improvement
- Cluster shutdown and restart
  - Minor improvement

MySQL Cluster 7.4 Restart Improvements

Speedups

- Faster Local Checkpoints
  - Parallel
  - Tuning & Adaptive throttling
- Parallel operations
  - Adaptive to avoid starving remaining system of resources
- Feedback loop
  - Determine Cluster 7.3 baseline for your environment
  - Experiment with Cluster 7.4 settings
  - Consult Oracle with results to establish improvements to be made in configuration

MySQL Cluster 7.4 – Enhanced memory reporting

- `ndbinfo.memory_per_fragment` memory usage information for each fragment replica, for each table and index
- Allocated memory and how much of that is actually in use.
- Exposes
  - Fragmentation of fixed and var-sized fragment pages
  - Accurate Data and Index Memory use
  - Comparison of Primary and Backup fragment usage
  - Partitioning effectiveness

```
mysql> desc ndbinfo.memory_per_fragment;
+-------------------------+---------------------+------+-----+---------+-------+
| Field                   | Type                | Null | Key | Default | Extra |
+-------------------------+---------------------+------+-----+---------+-------+
| fq_name                 | varchar(512)        | YES  |     | NULL    |       |
| parent_fq_name          | varchar(512)        | YES  |     | NULL    |       |
| type                    | varchar(512)        | YES  |     | NULL    |       |
| table_id                | int(10) unsigned    | YES  |     | NULL    |       |
| node_id                 | int(10) unsigned    | YES  |     | NULL    |       |
| block_instance          | int(10) unsigned    | YES  |     | NULL    |       |
| fragment_num            | int(10) unsigned    | YES  |     | NULL    |       |
| fixed_elem_alloc_bytes  | bigint(20) unsigned | YES  |     | NULL    |       |
| fixed_elem_free_bytes   | bigint(20) unsigned | YES  |     | NULL    |       |
| fixed_elem_size_bytes   | int(10) unsigned    | YES  |     | NULL    |       |
| fixed_elem_count        | bigint(20) unsigned | YES  |     | NULL    |       |
| fixed_elem_free_count   | decimal(16,0)       | YES  |     | NULL    |       |
| var_elem_alloc_bytes    | bigint(20) unsigned | YES  |     | NULL    |       |
| var_elem_free_bytes     | bigint(20) unsigned | YES  |     | NULL    |       |
| var_elem_count          | bigint(20) unsigned | YES  |     | NULL    |       |
| hash_index_alloc_bytes  | bigint(20) unsigned | YES  |     | NULL    |       |
+-------------------------+---------------------+------+-----+---------+-------+
16 rows in set (0.00 sec)
```


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 |
+------+------+-------------+------------+------------+
```

Enhanced Memory Reporting

See how memory is made available after deleting rows

```
+------+------+-------------+------------+------------+
| 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 |
+------+------+-------------+------------+------------+
mysql> DELETE FROM clusterdb.simples LIMIT 1;
+------+------+-------------+------------+------------+
| node | frag | alloc_bytes | free_bytes | spare_rows |
+------+------+-------------+------------+------------+
|    1 |    0 |      131072 |       5504 |        172 |
|    1 |    2 |      131072 |       1312 |         41 |
|    2 |    0 |      131072 |       5504 |        172 |
|    2 |    2 |      131072 |       1312 |         41 |
|    3 |    1 |      131072 |       3104 |         97 |
|    3 |    3 |      131072 |       4288 |        134 |
|    4 |    1 |      131072 |       3104 |         97 |
|    4 |    3 |      131072 |       4288 |        134 |
+------+------+-------------+------------+------------+
```

Enhanced Memory Reporting

Check how well partitioned/sharded a table is

```
mysql> CREATE TABLE simples (id INT NOT NULL AUTO_INCREMENT, species VARCHAR(20) DEFAULT "Human",
PRIMARY KEY(id, species)) engine=ndb PARTITION BY KEY(species);

// Add some data

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 ndbinfo.memory_per_fragment
where fq_name like '%simples%';
+------+------+-------------+------------+------------+
| node | frag | alloc_bytes | free_bytes | spare_rows |
+------+------+-------------+------------+------------+
|    1 |    0 |           0 |          0 |          0 |
|    1 |    2 |      196608 |      11732 |        419 |
|    2 |    0 |           0 |          0 |          0 |
|    2 |    2 |      196608 |      11732 |        419 |
|    3 |    1 |           0 |          0 |          0 |
|    3 |    3 |           0 |          0 |          0 |
|    4 |    1 |           0 |          0 |          0 |
|    4 |    3 |           0 |          0 |          0 |
+------+------+-------------+------------+------------+
```

MySQL Cluster 7.4 – Enhanced activity reporting

- `ndbinfo.operations_per_fragment` activity counters for each fragment replica, for each table and index
- PK & scan access – requests, bytes, rows…
- Exposes
  - How traffic maps to tables and indices
  - Query execution, use of indexes etc.
  - LDM and node imbalances
  - Hotspots and scan overloads

```
mysql> desc operations_per_fragment;
+-------------------------+---------------------+------+-----+---------+-------+
| Field                   | Type                | Null | Key | Default | Extra |
+-------------------------+---------------------+------+-----+---------+-------+
| fq_name                 | varchar(512)        | YES  |     | NULL    |       |
| parent_fq_name          | varchar(512)        | YES  |     | NULL    |       |
| type                    | varchar(512)        | YES  |     | NULL    |       |
| table_id                | int(10) unsigned    | YES  |     | NULL    |       |
| node_id                 | int(10) unsigned    | YES  |     | NULL    |       |
| block_instance          | int(10) unsigned    | YES  |     | NULL    |       |
| fragment_num            | int(10) unsigned    | YES  |     | NULL    |       |
| tot_key_reads           | bigint(20) unsigned | YES  |     | NULL    |       |
| tot_key_inserts         | bigint(20) unsigned | YES  |     | NULL    |       |
| tot_key_updates         | bigint(20) unsigned | YES  |     | NULL    |       |
| tot_key_writes          | bigint(20) unsigned | YES  |     | NULL    |       |
| tot_key_deletes         | bigint(20) unsigned | YES  |     | NULL    |       |
| tot_key_refs            | bigint(20) unsigned | YES  |     | NULL    |       |
| tot_key_attrinfo_bytes  | bigint(20) unsigned | YES  |     | NULL    |       |
| tot_key_keyinfo_bytes   | bigint(20) unsigned | YES  |     | NULL    |       |
| tot_key_prog_bytes      | bigint(20) unsigned | YES  |     | NULL    |       |
| tot_key_inst_exec       | bigint(20) unsigned | YES  |     | NULL    |       |
| tot_key_bytes_returned  | bigint(20) unsigned | YES  |     | NULL    |       |
| tot_frag_scans          | bigint(20) unsigned | YES  |     | NULL    |       |
| tot_scan_rows_examined  | bigint(20) unsigned | YES  |     | NULL    |       |
| tot_scan_rows_returned  | bigint(20) unsigned | YES  |     | NULL    |       |
| tot_scan_bytes_returned | bigint(20) unsigned | YES  |     | NULL    |       |
| tot_scan_prog_bytes     | bigint(20) unsigned | YES  |     | NULL    |       |
| tot_scan_bound_bytes    | bigint(20) unsigned | YES  |     | NULL    |       |
| tot_scan_inst_exec      | bigint(20) unsigned | YES  |     | NULL    |       |
| tot_qd_frag_scans       | bigint(20) unsigned | YES  |     | NULL    |       |
| conc_frag_scans         | int(10) unsigned    | YES  |     | NULL    |       |
| conc_qd_frag_scans      | bigint(11) unsigned | YES  |     | NULL    |       |
| tot_commits             | bigint(20) unsigned | YES  |     | NULL    |       |
```


General Usage Considerations

- MySQL Cluster is designed for
  - Short transactions
  - Many parallel transactions
- Utilize simple access patterns for high running transactions
  - Use efficient scans and batching interfaces
  - AQL gives huge performance in JOIN operations
- Storage engine configurable for each table…InnoDB or NDB

MySQL Cluster Evaluation Guide

http://mysql.com/why-mysql/white-papers/mysql_cluster_eval_guide.php

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

- clusterdb.com
- @clusterdb
- forums.mysql.com/list.php?25

ORACLE

