NoSQL & SQL
Blending the best of both worlds

Andrew Morgan
@andrewmorgan
www.clusterdb.com
Safe Harbour 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.
What NoSQL must deliver

- Massive scalability
  - No application-level sharding
- Performance
- High Availability/Fault Tolerance
- Ease of use
  - Simple operations/administration
  - Simple APIs
  - Quickly evolve application & schema
Still a role for the RDBMS?

- No best single solution fits all
- Mix and match

<table>
<thead>
<tr>
<th>Scalability</th>
<th>Performance</th>
<th>HA</th>
<th>Ease of use</th>
<th>SQL/Joins</th>
<th>ACID Transactions</th>
</tr>
</thead>
</table>

### NoSQL
- Simple access patterns
- Compromise on consistency for performance
- Ad-hoc data format
- Simple operation

### RDBMS
- Complex queries with joins
- ACID transactions
- Well defined schemas
- Rich set of tools
MySQL Cluster Architecture

MySQL Cluster Data Nodes

Data Layer

Clients

Application Layer

Management

MySQL Cluster Architecture

Scalability
Performance
HA
Ease of use
SQL/Joins
ACID Transactions

MySQL
node.js
Java

Copyright © 2013, Oracle and/or its affiliates. All rights reserved. | 3rd Feb 2013 | Oracle reserves the right to change the timing and content of any future release
MySQL Cluster Architecture

- Scalability
- Performance
- HA
- Ease of use
- SQL/Joins
- ACID Transactions

MySQL Cluster Data Nodes

Data Layer

Application Layer

Clients

Management

MySQL Cluster

Java

Node

MySQL

Management
MySQL Cluster Architecture

Scalability
Performance
HA
Ease of use
SQL/Joins
ACID Transactions

MySQL Cluster Data Nodes

Data Layer

Clients

Application Layer
1.2 Billion UPDATEs per Minute

- 30 x Intel E5-2600 Intel Servers
- NoSQL C++ API, flexaSynch benchmark
- ACID Transactions, with Synchronous Replication
<table>
<thead>
<tr>
<th>Feature</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scalability</td>
<td>✔</td>
</tr>
<tr>
<td>Performance</td>
<td>✔</td>
</tr>
<tr>
<td>HA</td>
<td>✔</td>
</tr>
<tr>
<td>Ease of use</td>
<td></td>
</tr>
<tr>
<td>SQL/Joins</td>
<td>✔</td>
</tr>
<tr>
<td>ACID Transactions</td>
<td>✔</td>
</tr>
</tbody>
</table>
Creating & running your first Cluster

The traditional way (pre-MCM) – Up and running in 15 mins

  - Versions for Linux, Windows & Solaris
<table>
<thead>
<tr>
<th>Feature</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scalability</td>
<td>✔</td>
</tr>
<tr>
<td>Performance</td>
<td>✔</td>
</tr>
<tr>
<td>HA</td>
<td>✔</td>
</tr>
<tr>
<td>Ease of use</td>
<td>✔</td>
</tr>
<tr>
<td>SQL/Joins</td>
<td>✔</td>
</tr>
<tr>
<td>ACID Transactions</td>
<td>✔</td>
</tr>
</tbody>
</table>
MySQL Cluster Manager

Bootstrap single host Cluster

1. Download MCM/Cluster package from edelivery.oracle.com:
2. Unzip
3. Run agent, define, create & start Cluster!

$> bin\mcmd --bootstrap

MySQL Cluster Manager 1.1.2 started
Connect to MySQL Cluster Manager by running "D:\Andrew\Documents\MySQL\mcm\bin\mcmd" -a NOVA:1862
Configuring default cluster 'mycluster'...
Starting default cluster 'mycluster'...
Cluster 'mycluster' started successfully
  ndb_mgmd NOVA:1186
  ndbd NOVA
  ndbd NOVA
  mysql NOVA:3306
  mysql NOVA:3307
  ndbapi *
  Connect to the database by running "D:\Andrew\Documents\MySQL\mcm\cluster\bin\mysql" -h NOVA -P 3306 -u root
MySQL Cluster 7.3 EA: Auto-Installer

- Fast configuration
- Auto-discovery
- Workload optimized
- Repeatable best practices
- For MySQL Cluster 7.2 + 7.3
# Downloading MySQL Cluster EA

MySQL Server Snapshots :: MySQL-Cluster-Auto-Installer

⚠️ Warning! For testing purposes only!

These binaries were created by MySQL testing servers. They are **NOT FIT FOR PRODUCTION**.

They are provided solely for testing purposes, to try the latest bug fixes and generally to keep up with the development.

- Please, **DO NOT USE THESE BINARIES IN PRODUCTION**.
- Instead, install them on a spare server.
- If you are looking for production ready binaries, please visit MySQL Downloads.
- MySQL Software is provided under the GPL License

## MySQL-Cluster-Auto-Installer

<table>
<thead>
<tr>
<th>Filename</th>
<th>Version</th>
<th>Date</th>
<th>Platform</th>
<th>Size</th>
<th>MD5Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>mysql-cluster-7.2.8-linux-x86_64.tar.gz</td>
<td>7.2.8</td>
<td>2012-09-27</td>
<td>linux-x86_64</td>
<td>253.2 M</td>
<td>12929a15b72027c8a4e91d0be26ef048</td>
</tr>
<tr>
<td>mysql-cluster-gpl-7.2.8-solaris10-sparc-64bit.tar.gz</td>
<td>7.2.8</td>
<td>2012-09-27</td>
<td>solaris10-sparc-64bit</td>
<td>561.9 M</td>
<td>663c7a063397bfff7d751e0acc1aace3</td>
</tr>
<tr>
<td>mysql-cluster-client-gpl-7.2.8-1.el6.i686.rpm</td>
<td>7.2.8</td>
<td>2012-09-27</td>
<td>source</td>
<td>13.8 M</td>
<td>e26858cc20d2cb214b28323eba9e8285b</td>
</tr>
<tr>
<td>Feature</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td>----</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scalability</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HA</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ease of use</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQL/Joins</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACID Transactions</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
NoSQL APIs
MySQL Cluster 7.1: ClusterJ/JPA

- 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
MySQL Cluster 7.2: Memcached

- Memcached is a distributed memory based hash-key/value store with no persistence to disk
- NoSQL, simple API, popular with developers
- MySQL Cluster already provides scalable, in-memory performance with NoSQL (hashed) access as well as persistence
  - Provide the Memcached API but map to NDB API calls
- Writes-in-place, so no need to invalidate cache
- Simplifies architecture as caching & database integrated into 1 tier
- Access data from existing relational tables
Cluster & Memcached – Schema-Free

Application view

SQL view

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>town:maidenhead</td>
<td>SL6</td>
</tr>
</tbody>
</table>

key: <town:maidenhead,SL6>

value: SL6
Cluster & Memcached – Configured Schema

Application view

SQL view

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

Config tables

map.zip

town  code  SL6

Prefix: <town:maidenhead,SL6>
Memcached NoSQL Access with InnoDB

- Memcached as a plugin of MySQL Server; same process space, with very low latency access to data
- Memcapable: supports both memcached ascii protocol and binary protocol
- Support multiple columns: users can map multiple columns into “value”
- Optional local caching: “innodb-only”, “cache-only”, and “caching”
- Batch operations for performance
**MySQL Cluster 7.3 EA: Node.js NoSQL API**

- Native JavaScript access to MySQL Cluster
  - End-to-End JavaScript: browser to the app and database
  - Storing and retrieving JavaScript objects directly in MySQL Cluster
  - Eliminate SQL transformation

- Implemented as a module for node.js
  - Integrates full Cluster API library within the web app

- Couple high performance, distributed apps, with high performance distributed database
Try Node.js example for yourself

https://github.com/mysql/mysql-js/tree/master/samples

<table>
<thead>
<tr>
<th>name</th>
<th>age</th>
<th>message</th>
<th>history</th>
</tr>
</thead>
<tbody>
<tr>
<td>create.sql</td>
<td>11 days ago</td>
<td>first version [bo]</td>
<td></td>
</tr>
<tr>
<td>delete.js</td>
<td>11 days ago</td>
<td>first version [bo]</td>
<td></td>
</tr>
<tr>
<td>find.js</td>
<td>11 days ago</td>
<td>first version [bo]</td>
<td></td>
</tr>
<tr>
<td>insert.js</td>
<td>11 days ago</td>
<td>first version [bo]</td>
<td></td>
</tr>
<tr>
<td>lib.js</td>
<td>11 days ago</td>
<td>first version [bo]</td>
<td></td>
</tr>
</tbody>
</table>
Next Steps

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

Try it Out
- dev.mysql.com/cluster
- labs.mysql.com
- github.com/mysql/mysql-js

Let us know what you think
- clusterdb.com
- @clusterdb
- forums.mysql.com/list.php?25