



# High-availability with Galera Cluster for MySQL

LinuxTag 2014  
10. Mai 2014, Berlin

by [oli.sennhauser@fromdual.com](mailto:oli.sennhauser@fromdual.com)

[www.fromdual.com](http://www.fromdual.com)

# About FromDual GmbH

- FromDual provides neutral and independent:
  - Consulting for MySQL, Galera Cluster, MariaDB and Percona Server
  - Support for all MySQL and Galera Cluster
  - Remote-DBA Services for all MySQL
  - MySQL Training
- Open Source Business Alliance (OSBA)
- Member of SOUG, DOAG, /ch/open



[www.fromdual.com](http://www.fromdual.com)

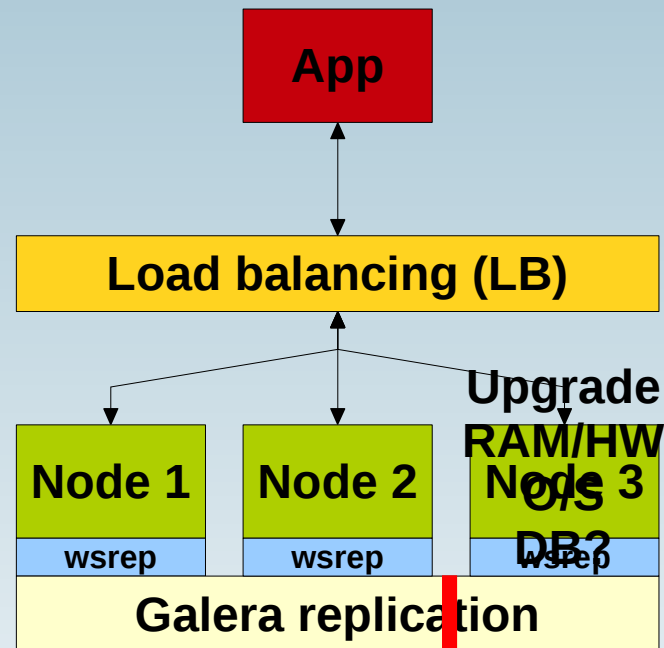
# High-Availability :-( ---

- Its also about maintenance...
- Who loves night-shifts?
- Who loves weekend-work?
- Who does regular upgrade (DB, kernel, etc.)?
- Who does regular reboots (after kernel upgrade)?
- Why are you not doing it in your office hours?



# The Galera Cluster for MySQL

# Maintenance time...



# Advantages / Disadvantages

- **Based on InnoDB SE**
- **Synchronous replication → No lost transaction**
- **Active-active multi-master Cluster**
  - **Read and write to any cluster node (no r/w split any more!)**
- **Read scalability and higher write throughput (Flash-Cache?)**
- **Automatic node membership control**
- **Rolling Restart (Upgrade of Hardware, O/S, DB release, etc.)**
- **True parallel replication, on row level**
  - **No slave lag**
- **A bit more complicated than normal MySQL, but similar complexity as M/S Replication!**
- **No original MySQL binaries → Codership MySQL binaries**
- **Be aware of Hot Spots on rows: Higher probability of deadlocks**

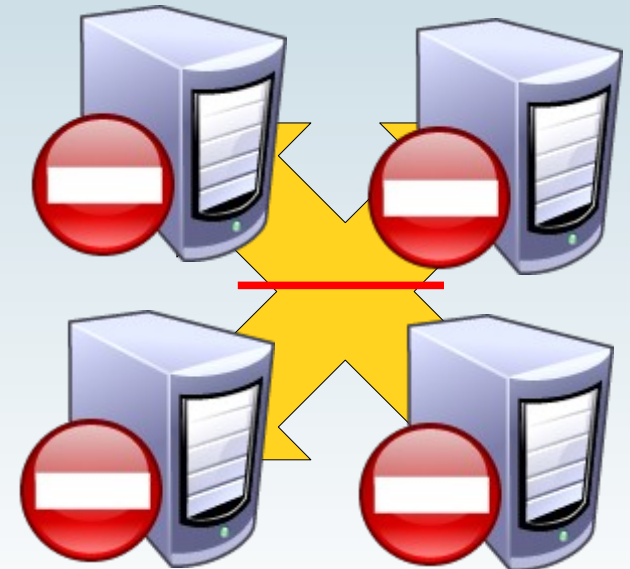
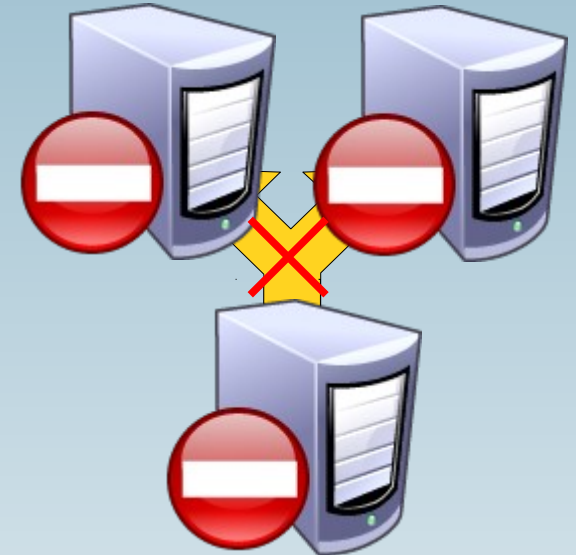
# Quorum and split-brain

- What is the problem?
- Split-brain → bad!



- Galera is a pessimistic Cluster → good!
- Quorum:  $\text{FLOOR}(n/2+1)$ 
  - more than half! → 3-node Cluster (or 2+1)

# Quorum







# Installation and Configuration

# Installation

- **Galera Cluster consists of:**
  - A patched Codership MySQL (`mysqld`)
    - Or MariaDB Galera Cluster
    - Or Percona XtraDB Cluster
  - The Galera Plugin (`libgalera_smm.so`)
- **Ways of installation**
  - Packets (RPM, DEB)
  - Binary tar-ball
  - Patch MySQL source and compile both
- **Download <http://galeracluster.com/downloads/>**

# MySQL Configuration

## `my.cnf`

```
[mysqld]

default_storage_engine      = InnoDB
binlog_format               = row

innodb_autoinc_lock_mode   = 2    # parallel applying

innodb_flush_log_at_trx_commit = 0  # performance only!

query_cache_size           = 0    # Galera 3 → experimental
query_cache_type           = 0    # Mutex! Consistency!
```

# Galera Configuration

## `my.cnf (conf.d/wsrep.cnf)`

```
[mysqld]

# wsrep_provider                = none
wsrep_provider                  = ../lib/plugin/libgalera_smm.so

# wsrep_cluster_address        = "gcomm://"
wsrep_cluster_address          = "gcomm://ip_node2,ip_node3"

wsrep_cluster_name              = 'Galera Cluster'
wsrep_node_name                 = 'Node A'

wsrep_sst_method                = mysqldump
wsrep_sst_auth                  = sst:secret
```



# Operations

# Initial Cluster start

- Start very 1<sup>st</sup> node with:

```
wsrep_cluster_address = "gcomm://"
```

or

```
mysqld_safe --wsrep-cluster-address="gcomm://"
```

- → this tells the node to be the first one!
- All other nodes normal:  

```
service mysqld start
```

# Rolling Restart

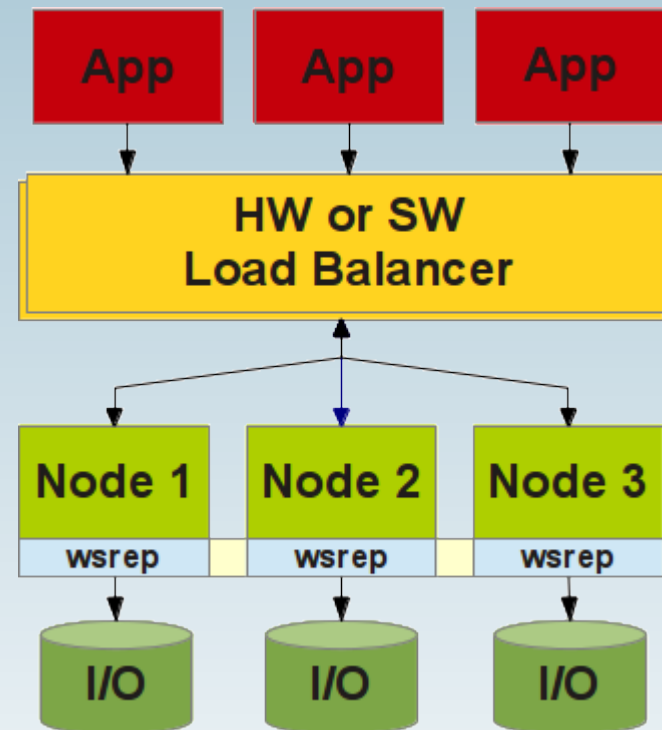
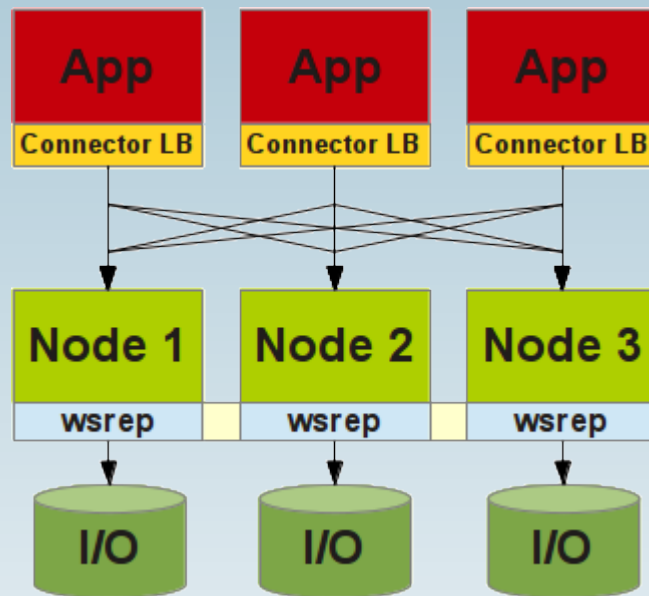
- **Scenario:**
  - **Hardware-, O/S-, DB- and Galera-Upgrade**
  - **MySQL configuration change**
  - **During full operation!!! (99.999% HA, 5x9 HA)**
- **→ Rolling Restart**
  - **Start one node after the other in a cycle**
  - **New features or settings are used after Rolling Restart is completed**

# Load Balancing

- **Connectors**
  - Connector/J
  - PHP: MySQLnd replication and load balancing plug-in
- **SW Load Balancer**
  - GLB, LVS/IPVS/Ldirector, HAProxy
- **HW Load Balancer**



# Location of Load Balancing





**Demo?**

# Online Schema Upgrade (OSU)

- **Schema Upgrade = DDL run against the DB**
  - Change DB structure
  - Non transactional
- **2 Methods:**
  - Total Order Isolation (TOI) (default)
  - Rolling Schema Upgrade (RSU)
- **`wsrep_osu_method = {TOI | RSU}`**



# Online Schema Upgrade

- **Total Order Isolation (TOI) (default)**
  - Part of the database is locked for the duration of the DDL.
  - + Simple, predictable and guaranteed data consistency.
  - - Locking operation
  - Good for fast DDL operations
- **Rolling Schema Upgrade (RSU)**
  - DDL will be only processed locally at the node.
  - Node is desynchronized for the duration of the DDL processing.
  - After DDL completion, delayed write sets are applied (similar to IST).
  - DDL should be manually executed at each node.
  - + only blocking one node at a time
  - - potentially unsafe and may fail if new and old schema are incompatible
  - Good for slow DDL operations

# Wir suchen noch:



- **Datenbank Enthusiast/in für Support / remote-DBA / Beratung**

# Q & A



**Questions ?**

**Discussion?**

**We have time for some face-to-face talks...**

- **FromDual provides neutral and independent:**
  - **Consulting**
  - **Remote-DBA**
  - **Support for MySQL, Galera, Percona Server and MariaDB**
  - **Training**

[www.fromdual.com](http://www.fromdual.com)