

MySQL/MariaDB Multi-Master Replication & Failover

A HA Solution using MMM and MySQL/MariaDB



Arjen Lentz & Walter Heck

arjen@openquery.com

walter@openquery.com

MySQL/MariaDB Multi-Master Replication & Failover © 2009-2010 Open Query



MySQL/MariaDB Multi-Master Replication & Failover © 2009-2010 Open Query



Overview

- Prepare virtual machines
 - this presentation explains MMM
- Configure MariaDB/MySQL
 - setup replication / my.cnf
- Install and configure MMM
- Test out scenarios

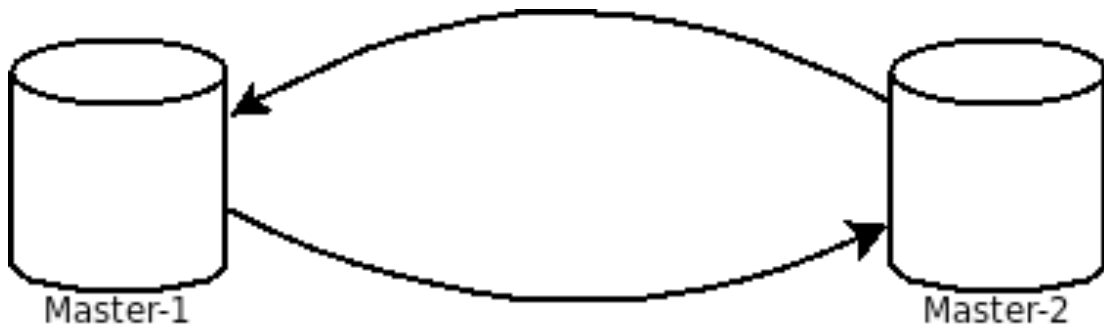


Introduction

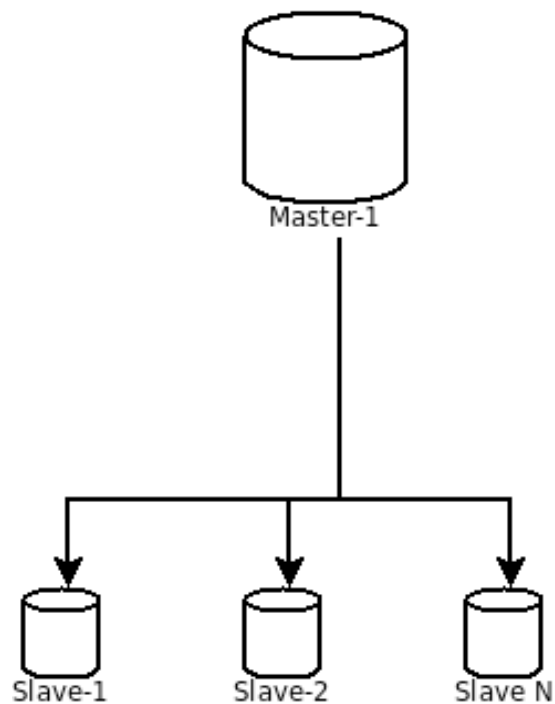
- What is MMM?
 - HA solution for *near* immediate failover
 - Automatic failover for slaves to another host
- What is MMM not?
 - Load balancer
 - 100% Data Reliability (Replication is not perfect!)



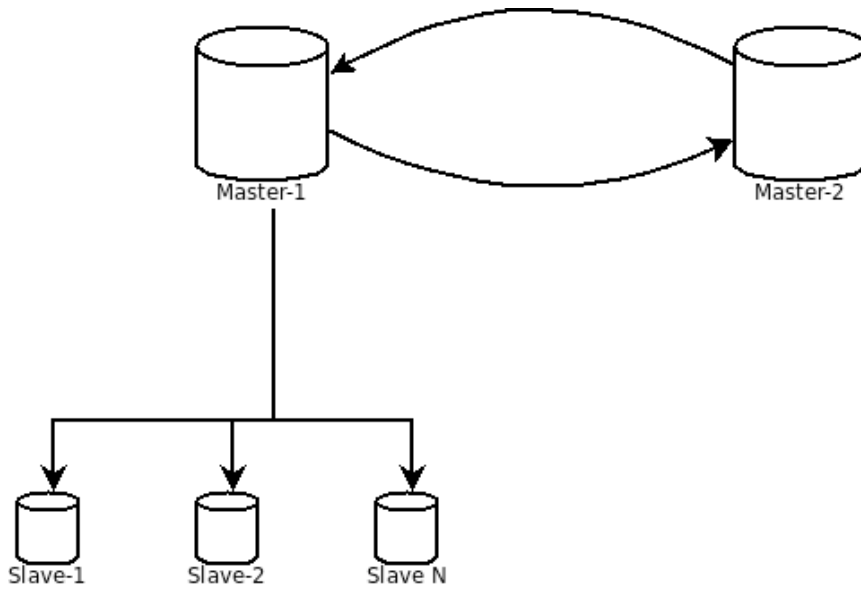
How MMM works



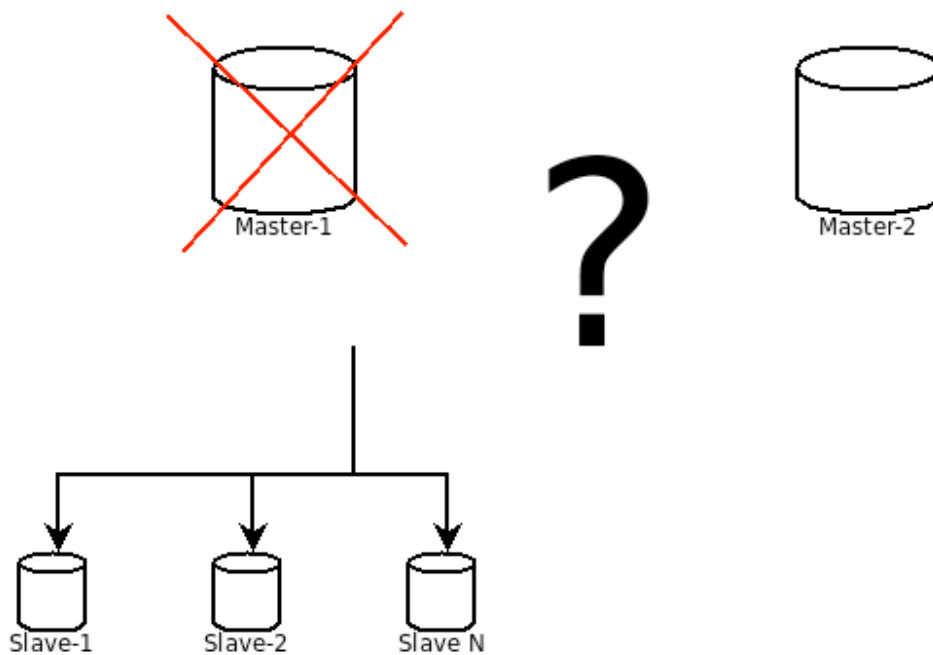
How MMM works



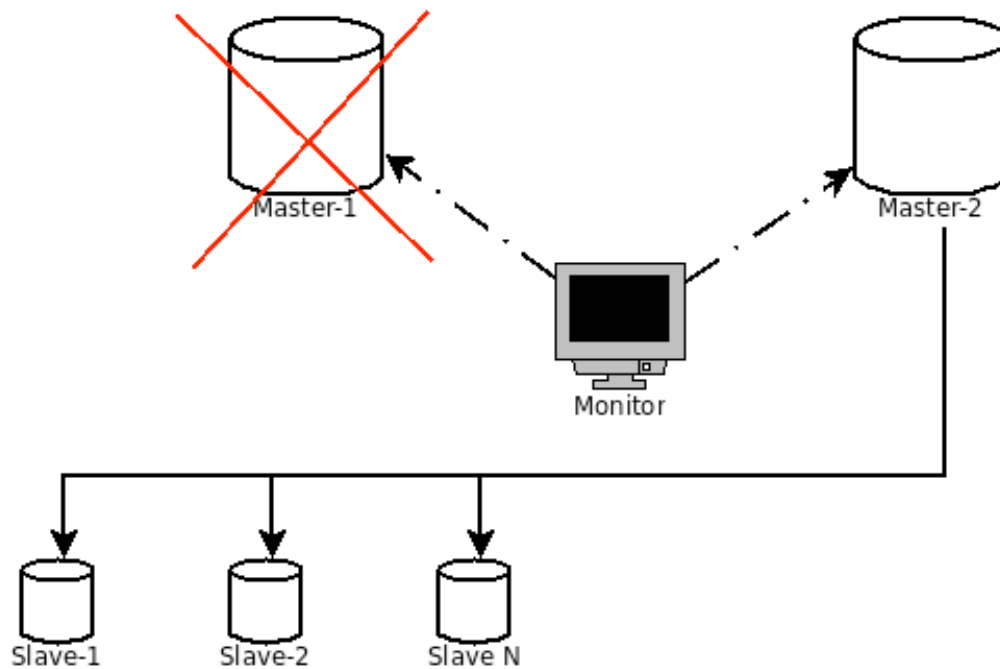
How MMM works



How MMM works



How MMM works



MySQL/MariaDB Multi-Master Replication & Failover © 2009-2010 Open Query



Virtual IP Voodoo

- Exclusive role
 - One virtual IP for multiple machines
 - machine goes down? → ip is moved to another machine
 - Usually used for write-mostly nodes (e.g. masters)
- Balanced role
 - One virtual IP for each machine
 - machine goes down? → ip is moved to another machine
 - Usually used for read-only nodes (e.g. slaves)
 - Caution! Nothing to do with load balancing!
 - One machine can have multiple virtual IPs

MySQL/MariaDB Multi-Master Replication & Failover © 2009-2010 Open Query



MMM 2.2.X

- Actively developed
 - current development done by Pascal Hoffman
- Complete rewrite from MMM 1.x
 - much more proper perl
- Uses highly customisable Log4Perl
- Built-in angel processes
- Passive mode
- Preferred roles



MMM 2.2.X – checks

- Checks allow mmm_mon to monitor health of the cluster
- 4 default checks included
 - Ping → server reachable?
 - rep_backlog → replication behind?
 - rep_threads → replication running
 - Mysql → mysql server reachable



MMM 2.2.X – Agent states (1)

- ONLINE
 - All is peachy, only state in which a node can have a role assigned
- REPLICATION_DELAY
 - replication backlog is too big (Check rep_backlog failed)
- REPLICATION_FAIL
 - replication threads are not running (Check rep_threads and rep_backlog failed)
- AWAITING_RECOVERY
 - Host is awaiting recovery. Entered after HARD_OFFLINE → all is ok



MMM 2.2.X – Agent states (2)

- HARD_OFFLINE
 - Host is offline (Check ping and/or mysql failed)
- ADMIN_OFFLINE
 - host was set to offline manually
- UNKNOWN
 - Host is in unknown state



Typical MMM 2.2.X 'hardware'/network requirements

- 5 (virtual) machines
 - Monitor → very lightweight
 - Data nodes → your choice
 - When using VM's:
 - distribute properly, clone smartly!
- Static IP for each machine
- 3 virtual IP's
 - 1 for master
 - 2 for slaves



Tutorial: Setting up MMM @ MySQLConf2K10



Tutorial: Setting up MMM

- using Ubuntu 9.10 VM specifically prepared for this session
 - already has mariadb 5.1.44 installed
 - has all prerequisites installed
 - follow instructions to change IP and hostname



MMM 2.2.X – MySQL/MariaDB

- Minimal settings for all data nodes:
 - enable log-bin
 - set unique server-id
- Settings for masters:
 - enable log-slave-updates



MMM 2.2.X – MySQL/MariaDB Replication

- Set up MySQL/MariaDB replication
 - Make master-2, slave-1 and slave-2 slaves of master-1
 - Make master-1 slave of master-2
- Start slaves and check if everything works properly
- If all == ok, start loading data into master1 and watch it replicate through the whole cluster
 - There's a file with 1 million records in it in ~/1000000.csv



MMM 2.2.X - sample config

```
active_master_role      writer                <host slave-1>
<host default>         ip 192.168.2.208
cluster_interface      eth0                 mode slave
pid_path                /var/run/mmm_agentd.pid </host>
bin_path                /usr/local/mysql-mmm/bin
replication_user        replication          <role writer>
replication_password    openquery            hosts master-1, master-2
agent_user              mmm_agent            ips 192.168.2.201
agent_password          openquery            mode exclusive
</host>                 </role>

<host master-1>        <role reader>
ip 192.168.2.206       hosts slave-1, slave-2
mode master            ips 192.168.2.202, 192.168.2.203
peer master-2          mode balanced
</host>                 </role>
===
this master-1
```



MMM 2.2.X – mmm_control

- mmm_control is used to control the cluster
 - Needs root privileges because of reading config files

MMM 2.2.X – common operations

- Move masters so you can do maintenance
 - mmm_control move_role writer my-master-1
- Set a slave offline for maintenance
 - mmm_control set_offline my-slave-1
 - mmm_control set_online my-slave-1
- Put MMM in passive state so it doesn't interfere
 - mmm_control set_passive/set_active
- Check the status of all checks
 - mmm_control checks

MMM 2.2.X – cluster state

- 'mmm_control show' shows the current state of all agents

```
[openquery@mmm2-monitor ~]$ sudo mmm_control show
master-1(192.168.2.206) master/ONLINE. Roles: writer(192.168.2.201)
master-2(192.168.2.207) master/ONLINE. Roles:
slave-1(192.168.2.208) slave/ONLINE. Roles: reader(192.168.2.202)
slave-2(192.168.2.209) slave/ONLINE. Roles: reader(192.168.2.203)
```

- 'mmm_control mode' shows whether the cluster is active or passive



Optional Extras

- Make configuration go through puppet
 - Makes adding new slaves extremely easy
- Install MySQL/MariaDB/MMM monitoring
 - Open Query uses Zabbix: lp:ourdelta-zabbix-scripts
- Preferred hosts
- When using LVM for MySQL/MariaDB, use MMM tools to clone/backup nodes
- Use SSL for MMM connections
- Mix and match solutions!
 - Use HW loadbalancer for reads
- cron-jobs with mk-table-checksum / mk-table-sync
 - Make sure schema is suitable



Ideas for improvement of MMM 2.2.X

- Add 'real' loadbalancer
 - haProxy
 - MySQL Proxy?
- Remove monitor as a SPoF
 - Keepalived, Heartbeat
- Proper packaging
 - Has just been added to Fedora 11&12 / EPEL



Credits / Links

<http://mysql-mmm.org>

<http://ourdelta.org>

<http://openquery.com>

<http://www.zabbix.com>



Thank you!

Arjen Lentz & Walter Heck

arjen@openquery.com / walter@openquery.com

