

MMM

MySQL Master-Master Replication Manager

Pascal Hofmann

Table of Contents

1	Overview	1
2	Typical use-cases	2
2.1	Two node setup	2
2.2	Two masters + one/many slaves	3
3	Requirements	4
3.1	Requirements - monitoring host	4
3.2	Requirements - nodes	5
3.3	Additional requirements for MMM tools	5
4	mmmd_agent - the agent	6
5	mmmd_mon - the monitor	7
5.1	States	7
5.2	Roles	7
5.3	Status file	8
5.4	Checks	8
5.5	Network check	8
5.6	Flapping	8
5.7	Passive mode	8
5.8	Startup	9
5.9	Role transition	9
5.9.1	Standard role	9
5.9.2	Active master role	9
5.10	Kill host functionality	9
6	mmm_control - controlling the monitor	10
6.1	help	10
6.2	ping	10
6.3	show	10
6.4	set_online <i>host</i>	10
6.5	set_offline <i>host</i>	11
6.6	mode	11
6.7	set_active	11
6.8	set_passive	11
6.9	move_role <i>role host</i>	11
6.10	move_role --force <i>role host</i>	12
6.11	set_ip <i>ip host</i>	12

7	Configuration	14
7.1	Config files	14
7.2	Configuration syntax	14
7.2.1	Comments	14
7.2.2	Variables	14
7.2.3	Specifying multiple values	14
7.2.4	Sections	14
7.2.4.1	Unique sections	14
7.2.4.2	Named sections	15
7.2.4.3	Default sections	15
7.2.5	Includes	15
7.3	Configuration variables	15
7.3.1	Global variables	15
7.3.2	<code>monitor</code> section	16
7.3.3	<code>host</code> sections	17
7.3.4	<code>role</code> sections	20
7.3.5	<code>check</code> sections	20
7.3.6	<code>socket</code> section	21
7.3.7	<code>copy_method</code> sections	21
8	Logging	23
8.1	Logging config files	23
8.2	Log4perl sample configuration	23
	Index	25

1 Overview

MMM (Master-Master Replication Manager for MySQL) is a set of flexible scripts to perform monitoring/failover and management of MySQL Master-Master replication configurations (with only one node writable at any time). The toolset also has the ability to read balance standard master/slave configurations with any number of slaves, so you can use it to move virtual IP addresses around a group of servers depending on whether they are behind in replication. In addition to that, it also has scripts for data backups, resynchronization between nodes etc.

The main functionality is provided through the following three scripts:

`mmmd_mon` monitoring daemon which does all monitoring work and makes all decisions about roles moving and so on.

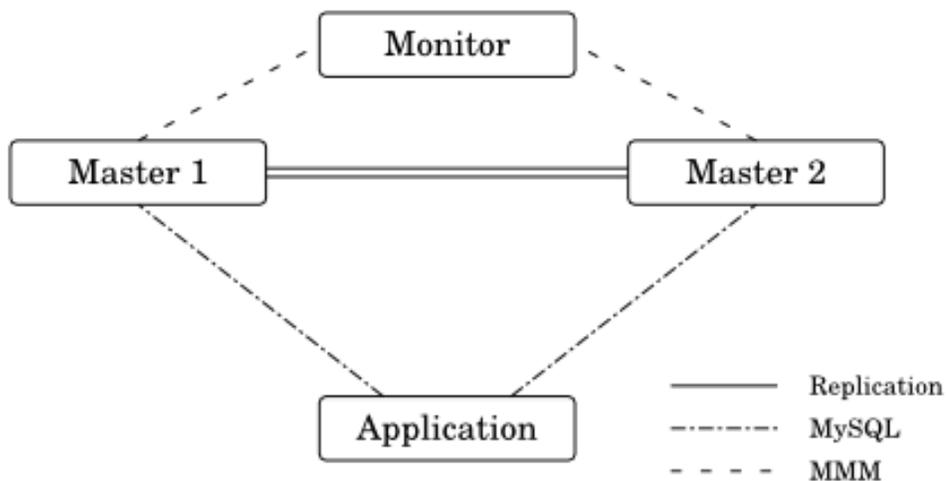
`mmmd_agent`
agent daemon which runs on each MySQL server and provides monitoring node with simple set of remote services.

`mmm_control`
simple script dedicated to management of the `mmmd_mon` processes by commands.

2 Typical use-cases

Typical use cases of MMM will be described in the following two sections.

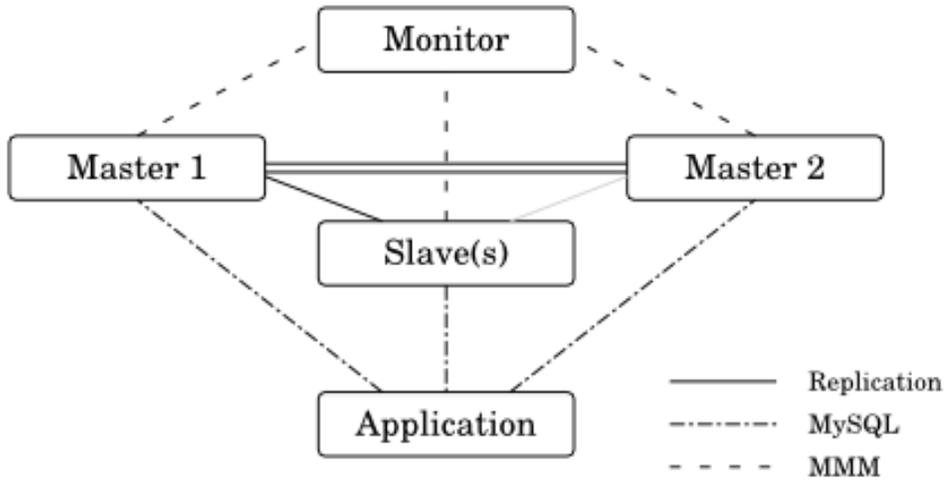
2.1 Two node setup



In two node master-master setup, MMM uses five IPs: single permanent IP for each node that is never changed, 2 reader IPs (read-only) and 1 writer IP (updates). Last three IPs are migrating between the nodes depending on node availability.

Normally (no replication failures, no replication delay etc) active master has 2 IPs (reader and writer), standby master - 1 IP (reader). In case of a failure, both - writer and reader roles migrate to the working node.

2.2 Two masters + one/many slaves



3 Requirements

For an MMM setup with n MySQL servers, you'll need

$n + 1$ hosts

One host for each MySQL server; one host for the MMM monitor.

$2 * (n + 1)$ IPs

One IP for each host (see above); one IP for the writer role; n IPs for one reader role per host.

monitor user

A MySQL user with privileges `REPLICATION CLIENT` for MMM monitor.

agent user

A MySQL user with privileges `SUPER`, `REPLICATION CLIENT`, `PROCESS` for MMM agent.

replication user

A MySQL user with privileges `REPLICATION SLAVE` used for replication.

tools user

A MySQL user with privileges `SUPER`, `REPLICATION CLIENT`, `RELOAD` for MMM tools.

3.1 Requirements - monitoring host

- perl with `ithreads` support
- `fping` (if you want to run `mmmd_mon` as non-root user)
- Perl modules:
 - `Algorithm::Diff`
 - `Class::Singleton`
 - `DBI` and `DBD::mysql`
 - `File::Basename`
 - `File::stat`
 - `File::Temp`
 - `Log::Dispatch`
 - `Log::Log4perl`
 - `Mail::Send`
 - `Net::Ping`
 - `Proc::Daemon`
 - `Thread::Queue`
 - `Time::HiRes`

3.2 Requirements - nodes

One should set `read-only=1` in the configuration of all MySQL servers, MMM will change that to `read-only=0` on the host with the `active_master_role`.

- perl
- iproute
- send_arp (solaris)
- Perl modules:
 - Algorithm::Diff
 - DBI and DBD::mysql
 - File::stat
 - Log::Dispatch
 - Log::Log4perl
 - Mail::Send
 - Net::ARP (linux)
 - Proc::Daemon
 - Time::HiRes

3.3 Additional requirements for MMM tools

If you want to use the MMM tools (`mmm_backup`, `mmm_restore`, `mmm_clone`) you have to use LVM for the partition on which your MySQL databases and logs reside. **Note:** You'll need free physical extends for the snapshots undo space (see [Estimating Undo Space needed for LVM Snapshot](#)).

You'll also need the following perl modules:

- Path::Class
- Data::Dumper

4 `mmmd_agent` - the agent

TODO short description

5 `mmd_mon` - the monitor

5.1 States

<code>ONLINE</code>	Host is running without any problems.
<code>ADMIN_OFFLINE</code>	host was set to offline manually.
<code>HARD_OFFLINE</code>	Host is offline (Check <code>ping</code> and/or <code>mysql</code> failed)
<code>AWAITING_RECOVERY</code>	Host is awaiting recovery
<code>REPLICATION_DELAY</code>	replication backlog is too big (Check <code>rep_backlog</code> failed)
<code>REPLICATION_FAIL</code>	replication threads are not running (Check <code>rep_threads</code> failed)

- Only hosts with state `ONLINE` may have roles. When a host switches from `ONLINE` to any other state, all roles will be removed from it.
- A host that was in state `REPLICATION_DELAY` or `REPLICATION_FAIL` will be switched back to `ONLINE` if everything is OK again, unless it is flapping (see [Section 5.6 \[Flapping\]](#), page 8).
- A host that was in state `HARD_OFFLINE` will be switched to `AWAITING_RECOVERY` if everything is OK again. If its downtime was shorter than 60 seconds and it wasn't rebooted or `auto_set_online` is `> 0` it will be switched back to `ONLINE` automatically, unless it is flapping (see [Section 5.6 \[Flapping\]](#), page 8 again).
- Replication backlog or failure on the active master isn't considered to be a problem, so the active master will never be in state `REPLICATION_DELAY` or `REPLICATION_FAIL`.
- Replication backlog or failure will be ignored on hosts whose peers got `ONLINE` less than 60 seconds ago (That's the default value of `master-connect-retry`).
- If both checks `rep_backlog` and `rep_threads` fail, the state will change to `REPLICATION_FAIL`.

5.2 Roles

There are two types of roles:

exclusive roles

Exclusive roles may only have one ip and are assigned to one host at a time. You may specify a *preferred* host, to which the role will always be moved, if it's online. **Note:** You can't move roles which are assigned to the preferred host, because they would be moved back to it immediately.

balanced roles

Balanced roles may have multiple ips. The ips will be balanced between the hosts, so that no host has two roles more than another host.

TODO describe `active_master_role` allow writes change master of all hosts with `mode=slave` to this host

5.3 Status file

information about host state and roles '`status_path`'

5.4 Checks

`mmd_mon` performs four different checks on each host to determine if it is OK. These checks are:

<code>ping</code>	host is pingable
<code>mysql</code>	MySQL server on the host is alive
<code>rep_threads</code>	replication threads are running
<code>rep_backlog</code>	replication backlog is not too big

5.5 Network check

`mmd_mon` has the ability to detect a non-functioning network connection. It regularly pings all `ping_ips` defined in the config. If at least one ip is reachable, the network is considered to be working.

Without working network connection `mmd_mon` will...

- ... ignore failed checks.
- ... not change the state of hosts.
- ... not send anything to agents.

If the network connection doesn't work during startup, `mmd_mon` will switch into passive mode (see [Section 5.7 \[Passive mode\]](#), page 8).

5.6 Flapping

`mmd_mon` supports the detection of hosts that are "flapping". Flapping occurs if a host which is `ONLINE` changes its state to `HARD_OFFLINE` / `REPLICATION_FAIL` / `REPLICATION_DELAY` too often and each time gets switched back to `ONLINE` because it has been down for less than 60 seconds. This may lead to roles getting switched between hosts very often.

To prevent this `mmd_mon` has a built in flap-detection which can be tuned in the configuration file. If a host goes down for more than `flap_count` times within `flap_duration` seconds it is considered as flapping and will not be set `ONLINE` automatically. It will stay in state `AWAITING_RECOVERY` until it gets set online (with `mmm_control set_online host`).

5.7 Passive mode

entered if no network connection during startup entered if discrepancies are detected during startup entered with `set_passive`

roles can be changed (unclean) with `set_ip` changed to active with `set_active`

roles get never changed automatically nothing is send to agents status file won't be updated

5.8 Startup

- Initial network check
- Initial host checks
- reads status information from ...
 - status file
 - agents (agent info)
 - hosts (system info)
- If status information doesn't match or network is down `PASSIVE` mode will be entered.

5.9 Role transition

5.9.1 Standard role

- IP is removed from old host
- IP is configured on new host
- New host sends arp packets to inform other hosts that it now has the IP

5.9.2 Active master role

- Writer role is removed from old master:
 1. MySQL is made `read_only`
 2. Active connections are removed
 3. IP is removed
- Slaves are informed. They:
 1. Try to catch up with the old master as far as possible
 2. Change master to the new master
- Writer role is added to the new master:
 1. MySQL is made writable
 2. IP is configured

5.10 Kill host functionality

If a host goes down and the agent on it can't be reached, the monitor executes a "kill host"-binary if present. You have to provide a custom binary for this which takes the hostname as first argument and the state of check ping (1 - ok; 0 - not ok) as second argument. This binary could for example:

- SSH to the failed host and use iptables to close mysql from the world by dropping all connections to port 3306.
- Tell a UPS to poweroff the failed host.
- ...

The default location for this binary is `'bin_path/monitor/kill_host'`. This may be changed via the variable `kill_host_bin` in the monitor section of the configuration.

6 `mmm_control` - controlling the monitor

The monitor daemon may be controlled with the help of `mmm_control`. If you have multiple clusters, you should always specify the name of the cluster you want to work with (i.e. to check C1 status, use "`mmm_control @C1 show`"). Otherwise - if you have only one MMM cluster, it can be used without cluster name.

6.1 `help`

Show help message.

6.2 `ping`

Ping the monitor daemon to check if it's still alive.

```
# mmm_control ping
OK: Pinged successfully!
```

When the monitor is down:

```
# mmm_control ping
ERROR: Can't connect to monitor daemon!
```

6.3 `show`

Show the current cluster status. See [Section 5.1 \[States\], page 7](#) for an explanation of the different node states.

```
# mmm_control show
db1(192.168.0.31) master/ONLINE. Roles: writer(192.168.0.50), reader(192.168.0.51)
db2(192.168.0.32) master/AWAITING_RECOVERY. Roles:
db3(192.168.0.33) slave/ONLINE. Roles: reader(192.168.0.52), reader(192.168.0.53)
```

6.4 `set_online host`

`set_online` is used to recover a node from a failure when it's state is `AWAITING_RECOVERY` or `ADMIN_OFFLINE`. In the following example, the host `db2` was rebooted. Here's the cluster status:

```
# mmm_control show
db1(192.168.0.31) master/ONLINE. Roles: writer(192.168.0.50), reader(192.168.0.51)
db2(192.168.0.32) master/AWAITING_RECOVERY. Roles:
db3(192.168.0.33) slave/ONLINE. Roles: reader(192.168.0.52), reader(192.168.0.53)
```

All roles have been moved to hosts `db1` and `db3` as `db2` has failed. Now that it's recovered, we should set it online:

```
# mmm_control set_online db2
OK: State of 'db2' changed to ONLINE. Now you can wait some time and check its new roles!■

# mmm_control show
db1(192.168.0.31) master/ONLINE. Roles: writer(192.168.0.50), reader(192.168.0.51)
db2(192.168.0.32) master/ONLINE. Roles: reader(192.168.0.52)
db3(192.168.0.33) slave/ONLINE. Roles: reader(192.168.0.53)
```

6.5 set_offline host

`set_offline` is used to bring a node down manually for maintenance. This will remove all roles and stop replication.

```
# mmm_control show
db1(192.168.0.31) master/ONLINE. Roles: writer(192.168.0.50), reader(192.168.0.51)
db2(192.168.0.32) master/ONLINE. Roles: reader(192.168.0.52)
db3(192.168.0.33) slave/ONLINE. Roles: reader(192.168.0.53)

# mmm_control set_offline db1
OK: State of 'db1' changed to ADMIN_OFFLINE. Now you can wait some time and check all roles!■

mon:~# mmm_control show
db1(192.168.0.31) master/ADMIN_OFFLINE. Roles:
db2(192.168.0.32) master/ONLINE. Roles: writer(192.168.0.50), reader(192.168.0.52)
db3(192.168.0.33) slave/ONLINE. Roles: reader(192.168.0.51), reader(192.168.0.53)
```

`set_online` is used to bring the node back online again. See [Section 6.4 \[set_online\]](#), [page 10](#).

6.6 mode

Print current mode. In the following example, the monitor is running in `ACTIVE` mode:

```
# mmm_control mode
ACTIVE
```

Here the monitor is in `PASSIVE` mode:

```
# mmm_control mode
PASSIVE
```

See [Section 5.7 \[Passive mode\]](#), [page 8](#).

6.7 set_active

Switch the monitor into `ACTIVE` mode:

```
# mmm_control set_active
OK: Switched into active mode.
```

See [Section 5.7 \[Passive mode\]](#), [page 8](#).

6.8 set_passive

Switch the monitor into `PASSIVE` mode:

```
# mmm_control set_passive
OK: Switched into passive mode.
```

See [Section 5.7 \[Passive mode\]](#), [page 8](#).

6.9 move_role role host

Used to move an exclusive role between the cluster nodes. This command is available in `ACTIVE` mode only. Lets assume the following situation:

```
# mmm_control show
db1(192.168.0.31) master/ONLINE. Roles: reader(192.168.0.51)
db2(192.168.0.32) master/ONLINE. Roles: writer(192.168.0.50), reader(192.168.0.52)
```

```
db3(192.168.0.33) slave/ONLINE. Roles: reader(192.168.0.53)
```

We use `move_role` to move the role `writer` to host `db1`:

```
# mmm_control move_role writer db1
OK: Role 'writer' has been moved from 'db2' to 'db1'. Now you can wait some time and check new roles info

# mmm_control show
db1(192.168.0.31) master/ONLINE. Roles: writer(192.168.0.50), reader(192.168.0.51)
db2(192.168.0.32) master/ONLINE. Roles: reader(192.168.0.52)
db3(192.168.0.33) slave/ONLINE. Roles: reader(192.168.0.53)
```

6.10 `move_role --force role host`

Can be used to move the `active_master_role` to a host with state `REPLICATION_FAIL` or `REPLICATION_BACKLOG`. Use this with caution! This command is available in `ACTIVE` mode only.

6.11 `set_ip ip host`

`set_ip` can be used to manipulate the roles in `PASSIVE` mode. The changes won't be applied until the monitor is switched into `ACTIVE` mode via `set_active`.

Let's assume we have our cluster up and running with the following status:

```
# mmm_control show
db1(192.168.0.31) master/ONLINE. Roles: writer(192.168.0.50)
db2(192.168.0.32) master/ONLINE. Roles: reader(192.168.0.51)
db3(192.168.0.33) slave/ONLINE. Roles: reader(192.168.0.52), reader(192.168.0.53)
```

Now, several bad things happen:

1. network connection to `db1` fails
2. `mmmd_mon` detects that `db1` has failed
3. `mmmd_mon` moves the `writer` role to `db2`, but can't remove it from `db1` (because it can't connect to it)
4. `mmmd_mon` crashes and the status file gets corrupted.
5. network connection to `db1` recovers
6. The admin restarts `mmmd_mon`

`mmmd_mon` has no status information now, and two nodes report, that they have the `writer` role, so `mmmd_mon` doesn't know what it should do and will switch into `PASSIVE` mode.

```
# mmm_control mode
PASSIVE

# mmm_control show
# --- Monitor is in PASSIVE MODE ---
# Cause: Discrepancies between stored status, agent status and system status during startup.■
#
# Stored status:
# db1(192.168.0.31) master/UNKNOWN. Roles:
# db2(192.168.0.32) master/UNKNOWN. Roles:
# db3(192.168.0.33) slave/UNKNOWN. Roles:
#
```

```

# Agent status:
# db1 ONLINE. Roles: writer(192.168.0.50). Master: ?
# db2 ONLINE. Roles: writer(192.168.0.50), reader(192.168.0.51). Master: ?
# db3 ONLINE. Roles: reader(192.168.0.52), reader(192.168.0.53). Master: db2
#
# System status:
# db1 writable. Roles: writer(192.168.0.50)
# db2 writable. Roles: writer(192.168.0.50), reader(192.168.0.51)
# db3 readonly. Roles: reader(192.168.0.52), reader(192.168.0.53)
#
db1(192.168.0.31) master/ONLINE. Roles: writer(192.168.0.50)
db2(192.168.0.32) master/ONLINE. Roles: reader(192.168.0.51)
db3(192.168.0.33) slave/ONLINE. Roles: reader(192.168.0.52), reader(192.168.0.53)

```

As you see, mmmd_mon tries to recover the status as well as possible. But in this situation it's wrong so one must move the writer role to db2 manually:

```

# mmm_control set_ip 192.168.0.50 db2
OK: Set role 'writer(192.168.0.50)' to host 'db2'.

```

Now take a look at the status, everything looks ok:

```

# mmm_control show
# --- Monitor is in PASSIVE MODE ---
# [...]
db1(192.168.0.31) master/ONLINE. Roles:
db2(192.168.0.32) master/ONLINE. Roles: writer(192.168.0.50), reader(192.168.0.51)
db3(192.168.0.33) slave/ONLINE. Roles: reader(192.168.0.52), reader(192.168.0.53)

```

Finally switch the monitor into active mode, so that it will apply the roles:

```

# mmm_control set_active
OK: Switched into active mode.

# mmm_control show
db1(192.168.0.31) master/ONLINE. Roles: reader(192.168.0.51)
db2(192.168.0.32) master/ONLINE. Roles: writer(192.168.0.50)
db3(192.168.0.33) slave/ONLINE. Roles: reader(192.168.0.52), reader(192.168.0.53)

```

Note: The role reader(192.168.0.51) has been moved to db1, because reader is a balanced role.

7 Configuration

7.1 Config files

Config files may resist in `/etc`, `/etc/mmm` or `/etc/mysql-mmm`. `/etc/mysql-mmm` should be preferred.

Program	Filename
<code>mmmd_agent</code>	<code>'mmm_agent.conf'</code>
<code>mmmd_mon</code>	<code>'mmm_mon.conf'</code> or <code>'mmm_mon_CLUSTER.conf'</code>
<code>mmm_control</code>	<code>'mmm_mon.conf'</code> or <code>'mmm_mon_CLUSTER.conf'</code>
<code>mmm_backup</code>	<code>'mmm_tools.conf'</code>
<code>mmm_clone</code>	<code>'mmm_tools.conf'</code>
<code>mmm_restore</code>	<code>'mmm_tools.conf'</code>

7.2 Configuration syntax

7.2.1 Comments

Lines which begin with the hash character `#` are considered comments, and are ignored. Comments may not be included on a line after a configuration directive. White space occurring before comments are ignored.

```
# This is a comment
debug 1 # this is no comment
```

7.2.2 Variables

```
asdsadasd
    debug    0
```

7.2.3 Specifying multiple values

For some variables you may specify multiple values seperated by a comma:

```
ips    192.168.0.51, 192.168.0.52, 192.168.0.53
```

7.2.4 Sections

The configuration is divided into two kinds of sections: unique and named.

7.2.4.1 Unique sections

Unique sections ... TODO

```
<monitor>
    ip    127.0.0.1
</monitor>
```

It is also possible to define empty unique sections:

```
<socket/>
```

7.2.4.2 Named sections

Named sections ... TODO

```
<host db1>
  ip    192.168.0.31
</host>
```

You may also define empty named sections:

```
<check mysql/>
```

7.2.4.3 Default sections

Values assigned in sections named *default* will be used as default values for all other sections of this type.

```
# Default mode of all hosts is 'slave'
<host default>
  mode slave
</host>
```

7.2.5 Includes

It is possible to split up the configuration into several files and include them via `include`.

```
include common.conf
```

7.3 Configuration variables

7.3.1 Global variables

- **this**

Description:	name of <i>this</i> host
Default value:	-
Used by:	agent, tools
- **debug**

Description:	Enable debug mode
Allowed values:	true/yes/1/on false/no/0/off
Default value:	0
Used by:	agent, monitor
- **active_master_role**

Description:	name of the role for which identifies the active master
Default value:	-
Used by:	agent, monitor
- **max_kill_retries**

Description:	Maximum number of retries when killing threads to prevent further writes during the removal of the <code>active_master_role</code> .
Default value:	10
Used by:	agent

- **default_copy_method**

Description: name of the default copy method
 Default value: -
 Used by: tools

- **clone_dirs**

Description: path(s) containing mysql data/logs (relative to mount point of logical volume)
 Default value: -
 Used by: tools

7.3.2 monitor section

The `monitor` section is required by `mmmd_mon` and `mmmd_control`.

- **ip**

Description: IP on which `mmmd_mon` listens
 Default value: -
 Used by: control, monitor

- **port**

Description: Port on which `mmmd_mon` listens
 Default value: 9988
 Used by: control, monitor

- **pid_path**

Description: Location of pid-file
 Default value: -
 Used by: monitor

- **bin_path**

Description: Path to directory containing MMM binaries
 Default value: -
 Used by: monitor

- **status_path**

Description: Location of status file
 Default value: -
 Used by: monitor

- **ping_interval**

Description: Break between network checks
 Default value: 1
 Used by: monitor

- **ping_ips**

Description: IPs used for network checks
 Default value: -
 Used by: monitor

- **flap_duration**

Description: Duration in seconds for flap detection. See `flap_count`

- | | |
|--------------------------|--|
| Default value: | 3600 |
| Used by: | monitor |
| ● flap_count | |
| Description: | Maximum number of downtimes within <code>flap_duration</code> seconds after which a host is considered to be flapping. |
| Default value: | 3 |
| Used by: | monitor |
| ● auto_set_online | |
| Description: | How many seconds to wait before switching node status from <code>AWAITING_RECOVERY</code> to <code>ONLINE</code> . 0 = disabled. |
| Default value: | 0 |
| Used by: | monitor |
| ● kill_host_bin | |
| Description: | Binary used to kill hosts if roles couldn't be removed because the agent was not reachable. You have to provide a custom binary for this which takes the hostname as first argument and the state of check ping (1 - ok; 0 - not ok) as second argument. |
| Default value: | <i>bin_path/monitor/kill_host</i> |
| Used by: | monitor |

7.3.3 host sections

- **ip**

Description:	IP of host
Default value:	-
Used by:	agent, monitor, tools
- **mode**

Description:	Mode of host. Either <code>master</code> or <code>slave</code> .
Default value:	-
Used by:	agent, monitor
- **peer**

Description:	Name of peer host (if mode is master)
Default value:	-
Used by:	agent, monitor
- **pid_path**

Description:	Location of pid-file
Default value:	-
Used by:	agent
- **bin_path**

Description:	Path to directory containing MMM binaries
Default value:	-
Used by:	agent
- **agent_port**

Description:	Port on which <code>mmmd_agent</code> listens
--------------	---

- | | |
|----------------|----------------|
| Default value: | 9989 |
| Used by: | agent, monitor |
- **cluster_interface**

Description:	network interface on which the IPs of the roles should be configured
Default value:	-
Used by:	agent
 - **mysql_port**

Description:	Port on which mysqld is listening
Default value:	3306
Used by:	agent, monitor, tools
 - **mysql_pidfile**

Description:	location of mysql pid-file
Default value:	/var/run/mysqld/mysqld.pid
Used by:	tools
 - **mysql_rcscript**

Description:	location of mysql rc-script
Default value:	/etc/init.d/mysql
Used by:	tools
 - **mysql_cnf**

Description:	location of my.cnf
Default value:	/etc/my.cnf
Used by:	tools
 - **agent_user**

Description:	mysql user for MMM Agent
Default value:	-
Used by:	agent
 - **agent_password**

Description:	mysql password for MMM Agent
Default value:	-
Used by:	agent
 - **monitor_user**

Description:	mysql user for MMM Monitor
Default value:	-
Used by:	monitor
 - **monitor_password**

Description:	mysql password for MMM Monitor
Default value:	-
Used by:	monitor
 - **replication_user**

Description:	mysql user used for replication
Default value:	-
Used by:	agent, tools

- **replication_password**
 - Description: mysql password used for replication
 - Default value: -
 - Used by: agent, tools
- **ssh_user**
 - Description: SSH user for MMM Tools
 - Default value: -
 - Used by: tools
- **ssh_port**
 - Description: SSH port for MMM Tools
 - Default value: 22
 - Used by: tools
- **ssh_parameters**
 - Description: additional SSH parameters for MMM Tools
 - Default value: -
 - Used by: tools
- **tools_user**
 - Description: mysql user for MMM Tools
 - Default value: -
 - Used by: tools
- **tools_password**
 - Description: mysql password for MMM Tools
 - Default value: -
 - Used by: tools
- **backup_dir**
 - Description: Target directory for backups
 - Default value: -
 - Used by: tools
- **restore_dir**
 - Description: Directory where backups should be restored to
 - Default value: -
 - Used by: tools
- **lvm_bin_lvcreate**
 - Description: Path to lvcreate binary
 - Default value: lvcreate
 - Used by: tools
- **lvm_bin_lvremove**
 - Description: Path to lvremove binary
 - Default value: lvremove
 - Used by: tools
- **lvm_snapshot_size**
 - Description: Size of LVM snapshot

- Default value: -
- Used by: tools
- **lvm_logical_volume**
 - Description: Logical volume where mysql data and logs reside
 - Default value: -
 - Used by: tools
- **lvm_volume_group**
 - Description: Volume group of logical volume with mysql data and logs
 - Default value: -
 - Used by: tools
- **lvm_mount_dir**
 - Description: Mount point for LVM snapshot
 - Default value: -
 - Used by: tools
- **lvm_mount_opts**
 - Description: Mount options used when mounting LVM snapshot
 - Default value: -
 - Used by: tools

7.3.4 role sections

This section defines what roles are in the cluster and which IPs will be used for each role. Since roles are moved among servers each server needs an IP (for the monitor to talk to it) and each role needs an IP.

- **mode**
 - Description: Mode of role. Either `balanced` or `exclusive` (see see [Section 5.2 \[Roles\], page 7](#)).
 - Default value: -
 - Used by: monitor
- **hosts**
 - Description: Hosts which may take over the role
 - Default value: -
 - Used by: monitor
- **ips**
 - Description: One or multiple IPs associated with the role
 - Default value: -
 - Used by: agent, monitor
- **prefer**
 - Description: The preferred host for this role. Only allowed for `exclusive` roles.
 - Default value: -
 - Used by: monitor

7.3.5 check sections

- **check_period**
 - Description: Perform check every 5 seconds

- Default value: 5
- Used by: monitor
- **trap_period**
 - Description: Check is considered as failed if it doesn't succeed for at least *trap_period* seconds.
 - Default value: 10
 - Used by: monitor
- **timeout**
 - Description: Check times out after *timeout* seconds
 - Default value: 2
 - Used by: monitor
- **restart_after**
 - Description: Restart checker process after *restart_after* checks
 - Default value: 10000
 - Used by: monitor
- **max_backlog**
 - Description: Maximum backlog for check *rep_backlog*.
 - Default value: 60
 - Used by: monitor

7.3.6 socket section

- **type**
 - Description: Socket type to use. Either *plain* or *ssl*
 - Default value: -
 - Used by: agent, control, monitor
- **cert_file**
 - Description: location of SSL certificate (if type is *ssl*)
 - Default value: -
 - Used by: agent, control, monitor
- **key_file**
 - Description: location of RSA private key (if type is *ssl*)
 - Default value: -
 - Used by: agent, control, monitor
- **ca_file**
 - Description: location of file containing certificate(s) of the reputable certificate authorities (if type is *ssl*)
 - Default value: -
 - Used by: agent, monitor

7.3.7 copy_method sections

- **backup_command**
 - Description: Command used for creation of backup
 - Default value: -
 - Used by: tools

- **restore_command**

Description: Command used for restoring backup
Default value: -
Used by: tools

- **incremental_command**

Description: Command used for showing available increments
Default value: -
Used by: tools

- **incremental**

Description: Indicates whether the copy method supports incremental backups
Default value: 0
Used by: tools

- **single_run**

Description: Indicates whether the copy method is able to copy all dirs in one single run
Default value: 0
Used by: tools

- **true_copy**

Description: Indicates whether the copy method creates a 1:1 copy of the data
Default value: 0
Used by: tools

8 Logging

8.1 Logging config files

MMM uses Log4perl for logging, so the logging abilities are very flexible. If the default logging configuration doesn't fit your needs you can create a config file which may resist in '/etc', '/etc/mmm' or '/etc/mysql-mmm'. '/etc/mysql-mmm' should be preferred.

The name of the file depends on the program you want to create it for:

Program	Filename
mmmd_agent	'mmm_agent_log.conf'
mmmd_mon	'mmm_mon_log.conf' or 'mmm_mon_log_CLUSTER.conf'
mmm_control	'mmm_mon_log.conf' or 'mmm_mon_log_CLUSTER.conf'
mmm_backup	-
mmm_clone	-
mmm_restore	-

8.2 Log4perl sample configuration

Here is the default logging configuration:

```
log4perl.logger = INFO, FileInfo, FileWarn, FileError, FileFatal, MailFatal

log4perl.appender.FileInfo = Log::Log4perl::Appender::File
log4perl.appender.FileInfo.Threshold = INFO
log4perl.appender.FileInfo.filename = /var/log/mysql-mmm/progam.info
log4perl.appender.FileInfo.recreate = 1
log4perl.appender.FileInfo.layout = PatternLayout
log4perl.appender.FileInfo.layout.ConversionPattern = %d %5p %m%n

log4perl.appender.FileWarn = Log::Log4perl::Appender::File
log4perl.appender.FileWarn.Threshold = WARN
log4perl.appender.FileWarn.filename = /var/log/mysql-mmm/progam.warn
log4perl.appender.FileWarn.recreate = 1
log4perl.appender.FileWarn.layout = PatternLayout
log4perl.appender.FileWarn.layout.ConversionPattern = %d %5p %m%n

log4perl.appender.FileError = Log::Log4perl::Appender::File
log4perl.appender.FileError.Threshold = ERROR
log4perl.appender.FileError.filename = /var/log/mysql-mmm/progam.error
log4perl.appender.FileError.recreate = 1
log4perl.appender.FileError.layout = PatternLayout
log4perl.appender.FileError.layout.ConversionPattern = %d %5p %m%n

log4perl.appender.FileFatal = Log::Log4perl::Appender::File
log4perl.appender.FileFatal.Threshold = FATAL
log4perl.appender.FileFatal.filename = /var/log/mysql-mmm/progam.fatal
log4perl.appender.FileFatal.recreate = 1
log4perl.appender.FileFatal.layout = PatternLayout
log4perl.appender.FileFatal.layout.ConversionPattern = %d %5p %m%n

log4perl.appender.MailFatal = Log::Dispatch::Email::MailSend
log4perl.appender.MailFatal.Threshold = FATAL
log4perl.appender.MailFatal.to = root
```

```
log4perl.appender.MailFatal.subject = FATAL error in progam
log4perl.appender.MailFatal.layout = PatternLayout
log4perl.appender.MailFatal.layout.ConversionPattern = %d %m%n
```

In addition to that, if debug mode is enabled:

```
log4perl.logger = DEBUG, ScreenLog, FileInfo, FileWarn, FileError, FileFatal, MailFatal
log4perl.appender.ScreenLog = Log::Log4perl::Appender::Screen
log4perl.appender.ScreenLog.stderr = 0
log4perl.appender.ScreenLog.layout = PatternLayout
log4perl.appender.ScreenLog.layout.ConversionPattern = %d %5p %m%n
```

Index

A

Agent 6

C

Checks 8

Comments 14

Config files 14

Configuration 14

Configuration syntax 14

F

Flapping, flap detection 8

H

help 10

K

Kill host functionality 9

L

Logging 23

Logging config files 23

M

mmm_control 10

mmmd_agent 6

mmmd_mon 7

Monitor 7

N

Named sections 15

Network check 8

O

Overview 1

P

passive mode 8

R

Requirements 4

Role transition 9

Roles 7

S

Sections 14

set_offline 11

set_online 10

show 10

Startup 9

States 7

Status file 8

T

typical use-cases 2

U

Unique Sections 14

V

Variables 14