提到MySQL高可用性，很多人会想到[MySQL Cluster](http://www.mysql.com/products/cluster/)，亦或者[Heartbeat](http://linux-ha.org/wiki/Heartbeat)+[DRBD](http://www.drbd.org/)，不过这些方案的复杂性常常让人望而却步，与之相对，利用MySQL复制实现高可用性则显得容易很多，目前大致有[MMM](http://mysql-mmm.org/)，[PRM](https://launchpad.net/percona-prm)，[MHA](http://code.google.com/p/mysql-master-ha/)等方案可供选择：MMM是最常见的方案，可惜它带来的问题往往比解决的问题还多（[What’s wrong with MMM?](http://www.xaprb.com/blog/2011/05/04/whats-wrong-with-mmm/)）；至于PRM，它还是个新项目，暂时不推荐用于产品环境，不过作为[Percona](http://www.percona.com/)的作品，它值得期待；如此看来目前只能选MHA了，好在经过[DeNA](http://dena.jp/)大规模的实践应用证明它是个靠谱的工具。

**安装：**

作为前提条件，应先配置[MySQL复制](http://huoding.com/2011/04/05/59)，并设置[SSH公钥免密码登录](https://help.ubuntu.com/community/SSH/OpenSSH/Keys)。下面以CentOS为例来说明，最好先安装[EPEL](http://fedoraproject.org/wiki/EPEL)，不然YUM可能找不到某些软件包。

MHA由Node和Manager组成，Node运行在每一台MySQL服务器上，也就是说，不管是MySQL主服务器，还是MySQL从服务器，都要安装Node，而Manager通常运行在独立的服务器上，但如果硬件资源吃紧，也可以用一台MySQL从服务器来兼职Manager的角色。

安装Node：

shell> yum install perl-DBD-MySQL

shell> rpm -Uvh http://mysql-master-ha.googlecode.com/files/mha4mysql-node-0.52-0.noarch.rpm

安装Manager：

shell> yum install perl-DBD-MySQL

shell> yum install perl-Config-Tiny

shell> yum install perl-Log-Dispatch

shell> yum install perl-Parallel-ForkManager

shell> rpm -Uvh http://mysql-master-ha.googlecode.com/files/mha4mysql-node-0.52-0.noarch.rpm

shell> rpm -Uvh http://mysql-master-ha.googlecode.com/files/mha4mysql-manager-0.52-0.noarch.rpm

**配置：**

配置全局设置：

shell> cat /etc/masterha\_default.cnf

[server default]

user=...

password=...

ssh\_user=...

配置应用设置：

shell> cat /etc/masterha\_application.cnf

[server\_1]

hostname=...

[server\_2]

hostname=...

注：MHA配置文件中参数的详细介绍请参考[官方文档](http://code.google.com/p/mysql-master-ha/wiki/Parameters)。

**检查**

检查MySQL复制：

shell> masterha\_check\_repl --conf=/etc/masterha\_application.cnf

检查SSH公钥免密码登录：

shell> masterha\_check\_ssh --conf=/etc/masterha\_application.cnf

**实战**

首先启动MHA进程：

shell> masterha\_manager --conf=/etc/masterha\_application.cnf

注：视配置情况而定，可能会提示read\_only，relay\_log\_purge等警告信息。

然后检查MHA状态：

shell> masterha\_check\_status --conf=/etc/masterha\_application.cnf

注：如果正常，会显示『PING\_OK』，否则会显示『NOT\_RUNNING』。

到此为止，一个基本的MHA例子就能正常运转了，不过一旦当前的MySQL主服务器发生故障，MHA把某台MySQL从服务器提升为新的MySQL主服务器后，如何通知应用呢？这就需要在配置文件里加上如下两个参数：

* [master\_ip\_failover\_script](http://code.google.com/p/mysql-master-ha/wiki/Parameters#master_ip_failover_script)
* [master\_ip\_online\_change\_script](http://code.google.com/p/mysql-master-ha/wiki/Parameters#master_ip_online_change_script)

说到Failover，通常有两种方式：一种是虚拟IP地址，一种是全局配置文件。MHA并没有限定使用哪一种方式，而是让用户自己选择，虚拟IP地址的方式会牵扯到其它的软件，这里就不赘述了，以下简单说说全局配置文件，以PHP为实现语言，代码如下：

#!/usr/bin/env php

<?php

$longopts = array(

'command:',

'ssh\_user:',

'orig\_master\_host:',

'orig\_master\_ip:',

'orig\_master\_port:',

'new\_master\_host::',

'new\_master\_ip::',

'new\_master\_port::',

);

$options = getopt(null, $longopts);

if ($options['command'] == 'start') {

$params = array(

'ip' => $options['new\_master\_ip'],

'port' => $options['new\_master\_port'],

);

$string = '<?php return ' . var\_export($params, true) . '; ?>';

file\_put\_contents('config.php', $string, LOCK\_EX);

}

exit(0);

?>

注：用其它语言实现这个脚本也是OK的，最后别忘了给脚本加上可执行属性。

如果要测试效果的话，可以kill掉当前的MySQL主服务器，稍等片刻，MHA就会把某台MySQL从服务器提升为新的MySQL主服务器，并调用master\_ip\_failover\_script脚本，如上所示，我们在master\_ip\_failover\_script脚本里可以把新的MySQL主服务器的ip和port信息持久化到配置文件里，这样应用就可以使用新的配置了。

有时候需要手动切换MySQL主服务器，可以使用[masterha\_master\_switch](http://code.google.com/p/mysql-master-ha/wiki/masterha_master_switch)命令，不过它调用的不是master\_ip\_failover\_script脚本，而是master\_ip\_online\_change\_script脚本，但调用参数类似，脚本可以互用。

shell> masterha\_master\_switch --conf=/etc/masterha\_application.cnf --master\_state=dead --dead\_master\_host=...

shell> masterha\_master\_switch --conf=/etc/masterha\_application.cnf --master\_state=alive --new\_master\_host=...

注：针对原来的MySQL主服务器是否已经宕机，执行命令所需的参数有所不同。

需要说明的是，如果MHA检测到MySQL主服务器连续发生宕机，且两次宕机时间间隔不足八小时的话，则不会进行Failover，之所以这样限制是为了避免ping-pong效应。不过为了自动化，我们往往希望能取消这种限制，此时可以用如下方式启动Manager：

shell> nohup masterha\_manager --conf=/etc/masterha\_application.cnf --ignore\_last\_failover --remove\_dead\_master\_conf &

注：请确保Manager的运行用户对masterha\_application.cnf有写权限。