[乐搜（lesoo.com)](http://www.lesoo.com/)使用HAProxy已经很久了，但主要用在前端web请求的负载均衡和状态监控上，对于后端的Memcached以及MySQL却一直没有应用，最近对系统架构进行了小规模的整理，把MySQL也收编到了HAProxy下，经过一段时间的使用，体验还是不错的。

详细的HAProxy的安装配置我就不说了，主要针对HAProxy如何监控检测后端MySQL服务器的状态说两句。

我们的环境由两台master和若干台slave数据库组成，在HA的配置中，可以设置灵活的策略和进行slave的分组，如果系统更大了，将还会根据业务系统来拆分若干个master和slave组。

***声明：以下内容应用和验证环境是基于 FreeBSD 8.2 环境，MySQL在5.1以上版本，理论上 Linux 的各个版本基本都能通过，如果有不能通过的情况，请别随意张口扔板砖，欢迎添加评论留言向我提问***

HA对MySQL状态监控实现的方式很多，不过大多逻辑都一样，使用自己编写的脚本程序（可以是shell、php、perl等你熟悉的语言）监控 mysql的运行情况，然后通过HTTP的方式输出运行结果给HA，我使用的是shell程序来实现的监控，结合 FreeBSD的inetd服务来实现监控端口的提供

**1. 监控脚本编写**

下面是我写的一段监控shell脚本，用于监控其中一台叫mdb1的master数据库replication运行：

1. #!/usr/local/bin/bash
2. MYSQL\_HOST="mdb1"
3. MYSQL\_PORT="3307"
4. MYSQL\_USERNAME="michael"
5. MYSQL\_PASSWORD="michael@lesoo.com"
7. /usr/local/bin/mysql -h$MYSQL\_HOST -P$MYSQL\_PORT -u$MYSQL\_USERNAME -p$MYSQL\_PASSWORD -e "show slave status\G;" > /tmp/check\_mdb1\_rep.txt
8. iostat=`grep "Slave\_IO\_Running" /tmp/check\_mdb1\_rep.txt  |awk '{print $2}'`
9. sqlstat=`grep "Slave\_SQL\_Running" /tmp/check\_mdb1\_rep.txt |awk '{print $2}'`
10. #echo iostat:$iostat and sqlstat:$sqlstat
11. if [ "$iostat" = "Yes" ] && [ "$sqlstat" = "Yes" ];
12. then
13. # mysql is fine, return http 200
14. /bin/echo "HTTP/1.1 200 OK\r\n"
15. /bin/echo "Content-Type: Content-Type: text/plain\r\n"
16. /bin/echo "\r\n"
17. /bin/echo "MySQL ($MYSQL\_HOST:$MYSQL\_PORT) replication is running.\r\n"
18. /bin/echo "\r\n"
19. else
20. # mysql is down, return http 503
21. /bin/echo "HTTP/1.1 503 Service Unavailable\r\n"
22. /bin/echo "Content-Type: Content-Type: text/plain\r\n"
23. /bin/echo "\r\n"
24. /bin/echo "MySQL ($MYSQL\_HOST:$MYSQL\_PORT) replication  is \*down\*.\r\n"
25. /bin/echo "\r\n"

上面的代码大概有点shell脚本编写经验的都能明白什么意思，就是用mysql 命令连接指定数据库并 -e 方式执行MySQL命令 “show slave status\G”，用以输出当前MySQL服务器的replication运行状况，里面会有两个主要的状态参数Slave\_IO\_Running和 Slave\_SQL\_Running， 这两个参数用于表示当前数据库的replication运行是否正常，如果有一个是No，则表示数据库同步歇菜了，把状态据结果通过Linux的管道命令 输出到 /tmp/check\_mdb1\_rep.txt文件（你可以自己随便改个文件名），再使用grep和awk命令把文件中想要的两个参数文本行读取出来并 拆分，最后判断是否都是“Yes”， 对于正常情况，我们输出一个HTTP协议标准的header头信息，里面包含关键的Code 200或者Code 503，这个其实就是HAProxy监控最终需要的东西。

上面的解释如果有不明白的，可以留言来问我。

把以上这段代码保存为一个文件，比如 /bin/mdb1\_replication.sh ，记得 chmod +x

**2. 把脚本变成一个系统TCP监听服务**

编写/etc/service文件，在末尾添加如下一行

1. mdb1\_replication 8822/tcp

这段代码的意思就是定义一个系统的服务，名称叫mdb1\_replication， 占用8822端口，使用TCP协议访问（这里名称和端口可以自己定义，只要不和系统别的服务重名和端口重复即可）

然后编写/etc/inetd.conf，在文件末尾添加如下一行

1. mdb1\_replication stream tcp nowait root /bin/mdb1\_replication.sh mdb1\_replication

这段代码的意思是添加一个接受系统inetd管理的服务，服务名称是前面提到的mdb1\_replication， 执行的脚本是 /bin/mdb1\_replication.sh，其他的几个参数意思就自己man inetd.conf看看吧

然后运行修改运行 /etc/rc.d/inetd restart 启动服务

运行 netstat -anl |grep 8822 看看是否服务和端口起来了，没有起来的话，检查检查哪里出错了，看看系统日志。

最后执行 telnet localhost 8822，正常应该看到如下运行结果

1. %telnet localhost 8822
2. Trying 127.0.0.1...
3. Connected to localhost.
4. Escape character is '^]'.
5. HTTP/1.1 200 OK\r\n
6. Content-Type: Content-Type: text/plain\r\n
7. \r\n
8. MySQL (mdb2:3307) replication is running.\r\n
9. \r\n
10. Connection closed by foreign host.
11. %

**3. 配置Haproxy的监控**

在HA配置文件中添加类似下面的部分内容

1. listen mdb\_3307 192.168.0.8:3307
2. mode tcp
3. maxconn 100
4. balance roundrobin
5. option httpchk OPTIONS \* HTTP/1.1\r\nHost:\ www
6. server www1:3307 www1:3307 weight 1 check port 9922 inter 2s rise 2 fall 2
7. server mdb2:3307 mdb2:3307 weight 3 check port 9922 inter 2s rise 2 fall 2 backup
8. option tcpka

上面这段配置我就不过多说明了，这是HAProxy的配置语法，如果对HA有不明白的地方，欢迎讨论。

补充： 目前我们的系统已经实现了 Nginx / Memcached / Sphinx / MySQL 等各个后端应用的负载均衡和状态监控，还有短信报警等环节，有兴趣讨论交流的，欢迎和我联络！

来源： <<http://www.toplee.com/blog/1284.html>>