

MYSQL中间件介绍

ATLAS 配置文件示例

```
-----  
[mysql-proxy] #不需要改  
plugins = admin, proxy #Atlas加载的模块名称，不需要改  
  
admin-username = user #管理接口的用户名  
admin-password = pwd #管理接口的密码  
admin-lua-script = /usr/local/mysql-proxy/lib/mysql-proxy/lua/admin.lua #实现管理  
接口的Lua脚本所在路径  
  
proxy-backend-addresses = 127.0.0.1:3306 #Atlas后端连接的MySQL主库的IP和端  
口，可设置多项，用逗号分隔  
proxy-read-only-backend-addresses = 127.0.0.1:3305@2 #Atlas后端连接的MySQL  
从库的IP和端口，2代表权重，用来作负载均衡，若省略则默认为1，可设置多项，用  
逗号分隔  
  
daemon = false #设置Atlas的运行方式，设为true时为守护进程方式，设为false时为  
前台方式，一般开发调试时设为false，线上运行时设为true  
keepalive = false #设置Atlas的运行方式，设为true时Atlas会启动两个进程，一个为  
monitor，一个为worker，monitor在worker意外退出后会自动将其重启，设为false时  
只有worker，没有monitor，一般开发调试时设为false，线上运行时设为true  
  
event-threads = 4 #工作线程数，推荐设置与系统的CPU核数相等  
log-level = message #日志级别，分为message、warning、critical、error、debug  
五个级别  
log-path = /usr/local/mysql-proxy/log #日志存放的路径  
instance = test #实例名称，用于同一台机器上多个Atlas实例间的区分  
  
proxy-address = 0.0.0.0:1234 #Atlas监听的工作接口IP和端口  
admin-address = 0.0.0.0:2345 #Atlas监听的管理接口IP和端口  
  
min-idle-connections = 128 #连接池的最小空闲连接数，可根据业务请求量大小适当  
调大或调小  
tables = person.mt.id.3 #分表设置，此例中person为库名，mt为表名，id为分表字  
段，3为子表数量，可设置多项，以逗号分隔  
pws = user1:+jKsgB3YAG8=, user2:GS+tr4TPgqc= #用户名与其对应的加密过的  
密码，密码使用加密程序encrypt加密，此设置项用于多个用户名同时访问同一个Atlas  
实例的情况，若只有一个用户名则不需要设置该项  
charset = utf8 #默认字符集，若不设置该项，则默认字符集为latin1
```

三、编译安装

```
-----  
依赖： glib(2.32.0以上)、libevent(1.4以上)、 Lua(5.1以上)、 OpenSSL(0.9.8以上)
```

```
./bootstrap.sh #可能需要修改其中的路径  
make  
sudo make install
```

四、启动与停止

进入PREFIX/conf目录，编辑instance.conf，此处instance的实际名称应与其中instance设置项相同，其他设置项含义见第二节。

启动：

```
PREFIX/bin/mysql-proxyd instance start
```

停止：

```
PREFIX/bin/mysql-proxyd instance stop
```

重启：

```
PREFIX/bin/mysql-proxyd instance restart
```

查看运行状态：

```
PREFIX/bin/mysql-proxyd instance status
```

-----HAProxy -----

一,HAProxy 介绍

反向代理服务器,支持双机热备支持虚拟主机，但其配置简单，拥有非常不错的服务器健康检查功能，当其代理的后端服务器出现故障，HAProxy会自动将该服务器摘除，故障恢复后再自动将该服务器加入。新的1.3引入了frontend,backend,frontend根据任意HTTP请求头内容做规则匹配，然后把请求定向到相关的backend.

二，利用HAProxy实现负载均衡

1. 利用HAProxy实现负载均衡

```
192.168.169.137 (haproxy)-----负载均衡-----  
(192.168.169.117;192.168.169.118)
```

安装配置HAProxy

```
cd /usr/local/
```

```
wget http://haproxy.1wt.eu/download/1.3/src/haproxy-1.3.14.2.tar.gz
```

```
tar zxvf haproxy-1.3.14.2.tar.gz
```

```
mv haproxy-1.3.14.2 haproxy
```

```
cd haproxy
```

```
make TARGET=linux26
```

2. 创建配置文件

```
# vi haproxy.cfg
```

复制内容到剪贴板

代码：

```
global
```

```
    maxconn 5120
```

```
    chroot /usr/local/haproxy
```

```
    uid 99
```

```
    gid 99
```

```
    daemon
```

```
    quiet
```

```
    nbproc 2 #通过nbproc多设置几个haproxy并发进程，这样每个进程的  
task_queue相对就会短很多，性能自然就能提高不少
```

```
    #pidfile /var/run/haproxy-private.pid
```

```

defaults
    log global
    mode http
    option httplog
    option dontlognull
    log 127.0.0.1 local3
    retries 3
    option redispatch
    maxconn 2000
    timeout 5000
    clitimeout 50000
    srvtimeout 50000

listen webfarm 0.0.0.0:80
    mode http
    stats uri /haproxy-stats #监控haproxy状态
    stats realm Haproxy\ statistics
    stats auth netseek:52netseek #设置状态监控的用户名为netseek密码为
52netseek
    balance roundrobin #负载均衡算法
    cookie SERVERID insert indirect
    option httpclose #
    option forwardfor #apache日志转发功能
    option httpchk HEAD /check.txt HTTP/1.0 #健康检测
server app_bbs1 192.168.169.117:80 cookie app1inst1 check inter 2000 rise 2 fall 5
server app_bbs2 192.168.169.118:80 cookie app1inst2 check inter 2000 rise 2 fall 5

```

syslog.conf里加一行
local3.* /var/log/haproxy.log

```

# touch /var/log/haproxy.log
# chown haproxy:haproxy /var/log/haproxy.log
# chmod u+x /var/log/haproxy.log
# tail -f /var/log/haproxy.log 监控日志
# ./haproxy -f haproxy.cfg 启动服务。

```

监控状态图示<http://192.168.169.137/haproxy-stats> ,输入用户名密码查看状态。