# 金佳金融线上平台部署

1. 修改服务器名称，修改为主机名称-数字格式

Hostname

Vi /etc/hosts

Vi /etc/sysconfig/network

Reboot

1. linux系统安装与优化

1.1 安装centos6.6 最小化安装

1.2 优化内核

net.ipv4.ip\_forward = 0  
net.ipv4.conf.default.rp\_filter = 1  
net.ipv4.conf.default.accept\_source\_route = 0  
kernel.sysrq = 0  
kernel.core\_uses\_pid = 1  
net.ipv4.tcp\_syncookies = 1  
kernel.msgmnb = 65536  
kernel.msgmax = 65536  
kernel.shmmax = 68719476736

#设置比物理内存小（单位字节）  
kernel.shmall = 4294967296  
net.ipv4.tcp\_max\_tw\_buckets = 6000  
net.ipv4.tcp\_sack = 1  
net.ipv4.tcp\_window\_scaling = 1  
net.ipv4.tcp\_rmem = 4096 87380 4194304  
net.ipv4.tcp\_wmem = 4096 16384 4194304  
net.core.wmem\_default = 8388608  
net.core.rmem\_default = 8388608  
net.core.rmem\_max = 16777216  
net.core.wmem\_max = 16777216  
net.core.netdev\_max\_backlog = 262144  
net.core.somaxconn = 262144  
net.ipv4.tcp\_max\_orphans = 3276800  
net.ipv4.tcp\_max\_syn\_backlog = 262144  
net.ipv4.tcp\_timestamps = 0  
net.ipv4.tcp\_synack\_retries = 1  
net.ipv4.tcp\_syn\_retries = 1  
net.ipv4.tcp\_tw\_recycle = 1  
net.ipv4.tcp\_tw\_reuse = 1  
net.ipv4.tcp\_mem = 94500000 915000000 927000000  
net.ipv4.tcp\_fin\_timeout = 1

net.ipv4.tcp\_keepalive\_probes = 3

net.ipv4.tcp\_keepalive\_intvl = 20  
net.ipv4.tcp\_keepalive\_time = 1800  
net.ipv4.ip\_local\_port\_range = 1024 65000

* 1. 系统安全

防止成为跳板机：iptables -A OUTPUT -p tcp -m multiport --dports 22,21,873 -j REJECT

* 1. 杀毒软件

1. tomcat安装与优化

2.1 安装java，先卸载掉系统自带的java

rpm -qa | grep java

rpm -e --nodeps tzdata-java-2012c-1.el6.noarch

rpm -e --nodeps java-1.6.0-openjdk-1.6.0.0-1.45.1.11.1.el6.x86\_64

2.2 设置java环境变量

rpm -e --nodeps tzdata-java-2012c-1.el6.noarch

rpm -e --nodeps java-1.6.0-openjdk-1.6.0.0-1.45.1.11.1.el6.x86\_64

设置java环境变量

JAVA\_HOME=/usr/local/java

CLASSPATH=$JAVA\_HOME/lib:$JAVA\_HOME/jre/lib

PATH=$PATH:$JAVA\_HOME/bin:$JAVA\_HOME/jre/bin

export PATH CLASSPATH JAVA\_HOME

source /etc/profile

* 1. 安装tomcat
  2. 修改server.xml

<Connector port="8080"

protocol="org.apache.coyote.http11.Http11AprProtocol"

connectionTimeout="20000"

redirectPort="8443"

maxThreads="700"

minSpareThreads="20"

acceptCount="100"

disableUploadTimeout="true"

enableLookups="false"

URIEncoding="UTF-8" />

<Connector port="8081"

executor="tomcatThreadPool"

protocol="org.apache.coyote.http11.Http11AprProtocol"

connectionTimeout="20000"

useURIValidationHack="false"

acceptCount="100"

disableUploadTimeout="true"

enableLookups="false"

URIEncoding="UTF-8"

redirectPort="8443" />

* 1. 安装apr，apr-iconv，apr-util，tomcat-native

cd apr-1.5.2

./configure --prefix=/data/apr

make ;make install

安装apr报错rm: cannot remove `libtoolT': No such file or

直接打开/usr/local/src/apr-1.5.2/configure

注释掉：$RM “$cfgfile”

cd apr-iconv-1.2.1

./configure --prefix=/data/apr-iconv --with-apr=/data/apr/

make ;make install

cd apr-util-1.5

./configure --prefix=/data/apr-util --with-apr=/data/apr --with-apr-iconv=/data/apr-iconv/bin/apriconv

make;make install

cd tomcat-native-1.1.33-src/jni/native/

./configure --with-apr=/data/apr --with-java-home=/usr/local/java/

make;make install

添加环境变量(必须是/usr/local/apr/lib这个路径)：

export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:/usr/local/apr/lib

启动tomcat看到如下字样，成功。



* 1. JVM优化

JAVA\_OPTS="-server –Xms4096M –Xmx4096M -Xss512k -XX:+AggressiveOpts -XX:+UseBiasedLocking -XX:PermSize=128M -XX:MaxPermSize=256M -XX:+DisableExplicitGC -XX:MaxTenuringThreshold=31 -XX:+UseConcMarkSweepGC -XX:+UseParNewGC -XX:+CMSParallelRemarkEnabled -XX:+UseCMSCompactAtFullCollection -XX:LargePageSizeInBytes=128m -XX:+UseFastAccessorMethods -XX:+UseCMSInitiatingOccupancyOnly -Djava.awt.headless=true "

1. mysql安装与优化

3.1 安装jemalloc

cd jemalloc-3.6.0

./configure

make && make install

echo '/usr/local/lib' > /etc/ld.so.conf.d/local.conf

ldconfig

3.2 MySQL/MaridDB 5.5编译方法，cmake预编译时加上下面参数

-DCMAKE\_EXE\_LINKER\_FLAGS="-ljemalloc" -DWITH\_SAFEMALLOC=OFF

3.2安装mysql

yum -y install patch cmake make gcc gcc-c++ gcc-g77 flex bison file libtool libtool-libs kernel-devel ncurses-devel

mkdir -p /opt/mysql

mkdir -p /opt/mysql/data

cmake -DCMAKE\_INSTALL\_PREFIX=/data/mysql -DMYSQL\_DATADIR=/data/mysql/data -DSYSCONFDIR=/data/mysql -DWITH\_MYISAM\_STORAGE\_ENGINE=1 -DWITH\_INNOBASE\_STORAGE\_ENGINE=1 -DWITH\_MEMORY\_STORAGE\_ENGINE=1 -DWITH\_PARTITION\_STORAGE\_ENGINE=1 -DMYSQL\_UNIX\_ADDR=/data/mysql/sock/mysqld.sock -DDEFAULT\_CHARSET=utf8 -DDEFAULT\_COLLATION=utf8\_general\_ci -DEXTRA\_CHARSETS:STRING=utf8,gbk -DWITH\_DEBUG=0 -DCMAKE\_EXE\_LINKER\_FLAGS="-ljemalloc" -DWITH\_SAFEMALLOC=OFF

make ;make install

groupadd mysql

useradd -s /sbin/nologin -g mysql -M mysql

/data/mysql/scripts/mysql\_install\_db --basedir=/data/mysql --datadir=/data/mysql/data --defaults-file=/data/mysql/my.cnf --user=mysql

chown -R mysql.mysql /data/mysql/

cp /data/mysql/support-files/my-default.cnf /etc/my.cnf # 添加mysql配置文件

cp /data/mysql/support-files/mysql.server /etc/init.d/mysql 启动mysql：service mysql start

vi /root/.bash\_profile #修改环境变量

PATH=$PATH:$HOME/bin:/data /mysql/bin:/data /mysql/lib

启动日志

/usr/local/mysql/data/localhost.err

source /root/.bash\_profile

cp support-files/mysql.server /etc/init.d/mysql #添加启动服务

service mysql start

mysqladmin -u root password 'jjjr@2015'

ps -ef | grep mysql

use mysql

grant all privileges on \*.\* to root@"%" identified by "jjjr@2015";

update user set Password = password('xxxxxx') where User='root';

flush privileges;

Mysql 主从，双方都执行一下命令

create user 'slave'@'%' identified by 'jjjrQWER@2015'; 创建用户

grant replication slave , replication client on \* . \* to 'slave'@'%' identified by 'jjjrQWER@2015'; 授权

一定要在配置完上面两句在做change，不然，什么时候都不成功。

show master status\G;

计算机生成了可选文字:
Pos I t Ion: 
Executed_Gti d _ set : 
bi nlog. 000002 
500 
1 row in set (0.02 sec) 

change master to master\_host='192.168.2.201',master\_user='slave',master\_port=3306,master\_password='1111',master\_log\_file='binlog.000002',master\_log\_pos=500;

对方ip 对方的file值 对方的position值

start slave;

show slave status\G;

计算机生成了可选文字:
Relay_Master_Log_Fi1e: bi nlog. 000003 
sl ng : 
ng: Ye 

都是yes就说明成功了

Slave\_IO\_Running:connecting 授权有问题用mysql -uslave -h192.168.3.207 -p 测试

1. nginx安装与优化

yum -y install gcc gcc+ gcc-c++ openssl openssl-devel pcre-devel zlib-devel

* 1. 安装jemalloc

cd jemalloc-3.6.0

./configure

make; make install

echo '/usr/local/lib' > /etc/ld.so.conf.d/local.conf

ldconfig

* 1. 安装nginx

cd nginx-1.8.0

./configure --prefix=/data/nginx --user=www --group=www --with-http\_stub\_status\_module --with-http\_ssl\_module --with-http\_gzip\_static\_module --with-ld-opt="-ljemalloc" --with-cc-opt='-O2'

make;make install

* 1. 检查jemalloc是否生效

lsof -n | grep jemalloc



1. zookeeper安装与优化(分布式集群)
   1. zookeeper 集群

Zookeeper 中分为三种角色： leader,follower,observer  四种状态：leading,following,observing,looking

observer 是在zookeeper3.3版本中引入用来解决拜占庭将军问题

observer 与 follower 作用一致

Zookeeper服务自身组成一个集群(2n+1个服务允许n个失效)。

Zookeeper服务有两个角色，

一个是leader，负责写服务和数据同步，

剩下的是follower，提供读服务，leader失效后会在follower中重新选举新的leader。

client 可以连接到每一个server 每个server都会保留一份完整的数据

每个follower 都和leader 保持连接  同步leader数据变更

server记录快照和持久存储到文件系统

* 1. 安装环境

安装java，更改环境变量

JAVA\_HOME=/data/java

CLASSPATH=$JAVA\_HOME/lib:$JAVA\_HOME/jre/lib

PATH=$PATH:$JAVA\_HOME/bin:$JAVA\_HOME/jre/bin

export PATH CLASSPATH JAVA\_HOME

* 1. 安装目录：data

mv zookeeper-3.4.6 /data/zookeeper

cd /data/zookeeper/

mv conf/zoo\_sample.cfg conf/zoo.cfg

* 1. 修改配置文件

#vi conf/zoo.cfg

tickTime=2000

initLimit=10

syncLimit=5

dataDir=/data/zookeeper/zdata ##注意后面需要新建myid，需要放在此文件夹内

clientPort=2181

server.1=10.10.112.108:2888:3888

server.2=10.10.111.160:2888:3888

server.3=10.10.101.201:2888:3888

根据上面的server.1 对应的IP地址 新建myid文件，如

在172.16.51.228 上执行

#echo "1”>>/data/zookeeper/zdata/myid

其他机器上则

#echo "2" >> /data/zookeeper/zdata/myid

#echo "3" >> /data/zookeeper/zdata/myid

在每隔机器上修改环境变量以便直接调用Zookeeper的相关命令

#vi /etc/profile

##ZOOKEEPER

############ZOOKEEPER#########

ZOOKEEPER\_HOME=/data/zookeeper

export ZOOKEEPER\_HOME

export PATH=$PATH:$ZOOKEEPER\_HOME/bin:$ZOOKEEPER\_HOME/conf

#source /etc/profile

分别启动Zookeeper

#zkServer.sh start ##启动Zookeeper

启动成功后可查看服务状态就当前等级，等级分别有follower，leader，leader挂了之后会在follower里面推选一个leader

[root@pre5 zookeeper]# zkServer.sh status

JMX enabled by default

Using config: /usr/local/zookeeper/bin/../conf/zoo.cfg

Mode: leader

或者

[root@pre4 zookeeper]# zkServer.sh status

JMX enabled by default

Using config: /usr/local/zookeeper/bin/../conf/zoo.cfg

Mode: follower

1. mongodb安装与优化
   1. mongodb 集群安装

Pre1:172.16.51.222 master

Pre2:172.16.51.223 slaver

Pre3:172.16.51.224 arbiter

# mv mongodb-linux-x86\_64-rhel62-3.0.2 /usr/local/mongodb

# mkdir -p /mongodb/data/master

# cd /usr/local/mongodb/

# mkdir /etc/mongodb

# mkdir /mongodb/log

* 1. 分别在各个服务器上创建相关配置文件

# vi /etc/mongodb/master.conf

#master.conf

dbpath=/mongodb/data/master ##数据库目录

logpath=/mongodb/log/master.log ##日志文件

pidfilepath=/mongodb/master.pid ##pid文件

directoryperdb=true ##为每个数据库按照数据库名创建目录

logappend=true ##以追加的方式记录日志

replSet=testrs ##replica set的名字

bind\_ip=172.16.51.222 ##绑定的本地IP地址

port=27017 ##端口

oplogSize=10000 ##mongodb操作日志文件的最大大小。单位为Mb，默认为硬盘剩余空间的5%

fork=true ##运行在后台

noprealloc=true ##不预先分配存储

#vi /etc/mongodb/slaver.conf

#slaver.conf

dbpath=/mongodb/data/slaver

logpath=/mongodb/log/slaver.log

pidfilepath=/mongodb/slaver.pid

directoryperdb=true

logappend=true

replSet=testrs

bind\_ip=172.16.51.223

port=27017

oplogSize=10000

fork=true

noprealloc=true

#vi /etc/mongodb/arbiter.conf

#arbiter.conf

dbpath=/mongodb/data/arbiter

logpath=/mongodb/log/arbiter.log

pidfilepath=/mongodb/arbiter.pid

directoryperdb=true

logappend=true

replSet=testrs

bind\_ip=172.16.51.224

port=27017

oplogSize=10000

fork=true

noprealloc=true

* 1. 然后分别启动mongodb

/usr/local/mongodb/bin/mongod -f /etc/mongodb/master.conf

/usr/local/mongodb/bin/mongod -f /etc/mongodb/slaver.conf

/usr/local/mongodb/bin/mongod -f /etc/mongodb/arbiter.conf

注意：可能启动需要一段时间

当提示：child process started successfully, parent exiting 即启动成功

* 1. 测试mongodb

配置主，备，仲裁节点

可以连接三个节点中的任意一个

# /usr/local/mongodb/bin/mongo 172.16.51.223:27017

MongoDB shell version: 3.0.2

connecting to: 172.16.51.222:27017/test

Server has startup warnings:

2015-05-11T16:02:04.623+0800 I CONTROL [initandlisten] \*\* WARNING: You are running this proce user, which is not recommended.

2015-05-11T16:02:04.623+0800 I CONTROL [initandlisten]

2015-05-11T16:02:04.623+0800 I CONTROL [initandlisten]

2015-05-11T16:02:04.623+0800 I CONTROL [initandlisten] \*\* WARNING: /sys/kernel/mm/transparentled is 'always'.

2015-05-11T16:02:04.623+0800 I CONTROL [initandlisten] \*\* We suggest setting it to 'ne

2015-05-11T16:02:04.623+0800 I CONTROL [initandlisten]

2015-05-11T16:02:04.623+0800 I CONTROL [initandlisten] \*\* WARNING: /sys/kernel/mm/transparentag is 'always'.

2015-05-11T16:02:04.623+0800 I CONTROL [initandlisten] \*\* We suggest setting it to 'ne

2015-05-11T16:02:04.623+0800 I CONTROL [initandlisten]

>use admin

>config= {\_id:"online", members:

... [ {\_id:0,host:'10.10.97.73:27017',priority:1},

... {\_id:1,host:'10.10.109.46:27017',priorit:1},

... {\_id:2,host:'10.10.102.155:27017',arbiterOnly:true}] };

##config 可以是任意的名字，priority设置优先级，数字越大的优先级越高，arbiterOnly:true 为设置仲裁节点，并且一定为true，注意大小写

> rs.initiate(config) ##初始化设置

在提示“OK” 后即配置成功

>rs.status() ##查看集群状态

> rs.conf() ##查看配置文件

State：1表示可以读写，2表示不能读写

Health：1表示正常，2表示异常

对于Replica Set来说，Secondary 节点默认是不可读的。

常用命令

testrs:PRIMARY> use test; ##切换到test数据库，没有会自动创建

testrs:PRIMARY> show dbs; ##显示数据库

testrs:PRIMARY> db.test.find() ##相当于select \* from database；

db.test.insert({“name”:”foobar”,”age”:25}) ##插入数据

1. Redis
   1. 安装jemalloc

cd jemalloc-3.6.0

./configure

make; make install

echo '/usr/local/lib' > /etc/ld.so.conf.d/local.conf

ldconfig

* 1. redis 集群安装

Redis 三主三从

Pre1 172.16.51.224 7001/7002

Pre2 172.16.51.226 7001/7002

Pre3 172.16.51.227 7001/7002

* 1. 安装redis

# tar zxvf redis-3.0.1.tar.gz ;mv redis-3.0.1 /usr/local/redis

# cd /usr/local/redis/

# yum install gcc gcc-c++ kernel-devel

# make MALLOC=libc

# make install

# cp /usr/local/redis/redis.conf /etc/redis/redis\_master.conf

# cp /usr/local/redis/redis.conf /etc/redis/redis\_slaver.conf

#mkdir /etc/redis

# mkdir /redis/data

# mkdir -p /data/redis\_master

# mkdir -p /data/redis\_slaver

* 1. 修改redis配置文件

daemonize yes ##后台

port 7000 #端口

cluster-enabled yes ##启动集群

cluster-config-file nodes.conf ##节点配置文件，redis自己创建

cluster-node-timeout 5000 ##超时时间

appendonly yes ##每次操作会记录一条log

dir /data/redis\_master ##数据保存路径 分别填写相应路径

* 1. 安装集群相关依赖包

# yum install ruby

# yum install rubygems

因为国内ruby源可能不可用，所以添加国内ruby源

# gem source ##查看当前源

# gem sources -r <http://rubygems.org/> ##删除官方源

# gem sources -a <http://ruby.taobao.org/> ##增加国内源

# gem source –u 更新源

# gem install redis --version 3.0.0 ##安装依赖环境

# cd /usr/local/redis/src/

在各机器上启动redis服务

# redis-server /etc/redis/redis.conf

#./redis-trib.rb create --replicas 1 10.10.100.166:7001 10.10.100.166:7002 10.10.99.195:7001 10.10.99.195:7002 10.10.100.44:7001 10.10.100.44:7002

* 1. 集群基本命令

创建集群

# ./redis-trib.rb create –replicas 节点角色有顺序决定，先master，然后slave

检查集群状态

# ./redis-trib.rb check 172.16.51.224:7000 (可以是任意一个节点)

添加新的master节点，如在本机创建一个节点

#cp /etc/redis/redis.conf/etc/redis/redis2.conf 在修改端口，不与本机冲突

将空节点加入集群

#redis-trib.rb add-node 172.16.51.224:7001

注意：此时新节点是没有任何数据的，因为没有包含任何的slot（插槽），新加入的是主节点，当集群需要叫某个节点升级为新的主节点的时候，新节点不会呗选中。

#redis-trib.rb reshard 172.16.51.224:7001

然后会提示要选择的slot 1-16384

1. cas
   1. 安装java和tomcat环境

设置环境变量

JAVA\_HOME=/usr/local/jdk1.7.0\_76

CLASSPATH=$JAVA\_HOME/lib:$JAVA\_HOME/jre/lib

PATH=$PATH:$JAVA\_HOME/bin:$JAVA\_HOME/jre/bin

export PATH CLASSPATH JAVA\_HOME

* 1. 安装cas server 端

将cas项目cp到tomcat目录下

cp cas-server-4.0.1/modules/cas-server-webapp-4.0.1.war /opt/tomcat/webapps/

修改名字

mv cas-server-webapp-4.0.0.war cas.war

* 1. 数据库连接

cp cas-server-4.0.0/modules/cas-server-support-jdbc-4.0.0.jar/opt/tomcat/webapps/cas/WEB-INF/lib/

mv mysql-connector-java-5.1.22-bin.jar /opt/tomcat/lib/

* 1. 生成证书并导出

keytool -genkey -alias mycas -keyalg RSA -keysize 2048 -keystore /opt/mycas.keystore

What is your first and last name?

[Unknown]: cas\_deve.jinjiajinrong.com 和域名一样

What is the name of your organizational unit?

[Unknown]: jinjiajinrong.com

What is the name of your organization?

[Unknown]: cas\_deve

What is the name of your City or Locality?

[Unknown]: sh

What is the name of your State or Province?

[Unknown]: sh

What is the two-letter country code for this unit?

[Unknown]: sso

keytool -export -file /opt/cas.crt -alias mycas -keystore /opt/mycas.keystore

8.修改tomcat文件

Server.xml

<Connector port="8443" protocol="org.apache.coyote.http11.Http11Protocol"

maxThreads="150" SSLEnabled="true" scheme="https" secure="true"

clientAuth="false" sslProtocol="TLS" keystoreFile="/opt/mycas.keystor" keystorePass="生成证书输入的密码" />

* 1. 客户端导入证书

keytool -import -keystore /usr/local/jdk1.7.0\_76/jre/lib/security/cacerts -file /opt/mycas.crt -alias mycas

* 1. 修改cas配置文件

/opt/tomcat/webapps/cas/WEB-INF/deployerConfigContext.xml

注释掉

<!--

<bean id="primaryAuthenticationHandler"

class="org.jasig.cas.authentication.AcceptUsersAuthenticationHandler">

<property name="users">

<map>

<entry key="casuser" value="Mellon"/>

</map>

</property>

</bean>

-->

增加以下内容

<bean id="primaryAuthenticationHandler" class="org.jasig.cas.adaptors.jdbc.QueryDatabaseAuthenticationHandler">

<property name="dataSource" ref="casDataSource" />

<property name="sql" value="select password from cas\_table where login\_name = ?" />

<property name="passwordEncoder" ref="passwordEncoder"/>

</bean>

<bean id="casDataSource" class="org.apache.tomcat.jdbc.pool.DataSource" destroy-method="close">

<property name="driverClassName">

<value>com.mysql.jdbc.Driver</value>

</property>

<property name="url">

<value>jdbc:mysql://172.16.51.212:3306/cas</value>

</property>

<property name="username">

<value>root</value>

</property>

<property name="password">

<value>jjjr@2015</value>

</property>

<property name="timeBetweenEvictionRunsMillis" value="900000" />

<property name="minEvictableIdleTimeMillis" value="1800000" />

<property name="validationQuery" value="SELECT 1" />

<property name="validationInterval" value="600000" />

<property name="testWhileIdle" value="true" />

<property name="testOnBorrow" value="true" />

</bean>

<bean id="passwordEncoder"

class="org.jasig.cas.authentication.handler.DefaultPasswordEncoder" >

<constructor-arg value="MD5"/>

</bean>

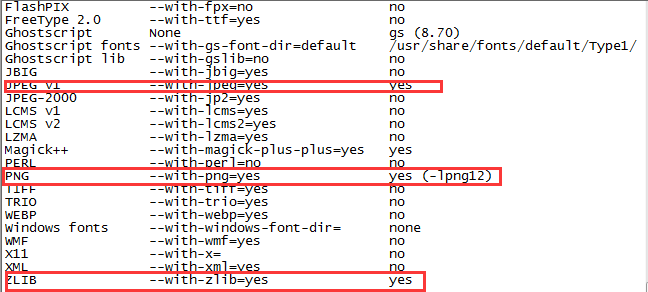
1. GraphicsMagick
   1. 安装必须包

yum install libjpeg libjpeg-devel libpng libpng-devel zlib-devel

* 1. 安装

cd GraphicsMagick-1.3.21

./configure



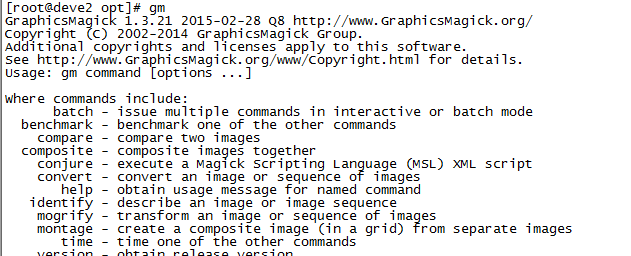
make;make install

which gm

PATH=$PATH:$JAVA\_HOME/bin:$JAVA\_HOME/jre/bin:/usr/local/bin/gm，添加/etc/profile

测试，能打开，证明安装成功

gm



gm convert -resize 100x80^ -gravity Center -crop 100x80+0+0 sim.png tt.png

1. Fastdfs

安装双组双Tracker

Pre1 172.16.51.222

Tracker1：22122

Storage：23000 group1

Pre2 172.16.51.223

Tracker2：22122

Storage：23000 group2

Pre3 172.16.51.224

Storage：23000 group1

Pre6 172.16.51.227

Storage：23000 group2

* 1. 首先卸载自带的zibevent

#rpm –e --nodeps libevent-1.4.13-4.el6.x86\_64

#cd /root/tmp

#tar zxvf libevent-2.0.21-stable.tar.gz

# cd libevent-2.0.21-stable

#./configure --prefix=/usr/local/libevent

# make && make install

#echo '/usr/local/libevent/lib/' >> /etc/ld.so.conf #加入动态链接库

#echo '/usr/local/libevent/linclude/' >> /etc/ld.so.conf #加入动态链接库

#ldconfig

安装Libfastcommon

#unzip master.zip

#cd libfastcommon-master/

#./make.sh && ./make.sh install

* 1. 安装Fastdfs

#tar zxvf fastdfs-5.05.tar.gz

# cd fastdfs-5.05

#./make.sh

#./make.sh install

#cd /etc/fdfs/

#cp storage.conf.sample storage.conf（storage服务器）

#cp tracker.conf.sample tracker.conf（tracker 服务器）

#cp client.conf.sample client.conf

* 1. 修改tracker服务器配置

#vi /etc/fdfs/tracker.conf

bind\_addr=172.16.51.222 ##绑定本机ip

base\_path=/home/data/tracker ##指定数据目录

reserved\_storage\_space = 10% ##保留存储空间，小于这个值就不能上传

* 1. 修改storage服务器配置

#vi /etc/fdfs/storage.conf

disabled=false

group\_name=group1 ##组名，根据所在服务器分配相应的组

bind\_addr=172.16.51.222 ##绑定的IP，127.0.0.1不可以

base\_path=/home/data/fastdfs ##数据库目录

store\_path0=/home/data/fastdfs ##存储目录

tracker\_server=172.16.51.222:22122 ##Tracker服务IP:PORT 多个Tracker服务器，每行一个

tracker\_server=172.16.51.223:22122 ##如上

Dubbo monitor（前提，必须先启动zookeeper）

1. 安装dubbo

新建目录：mkdir dubbo

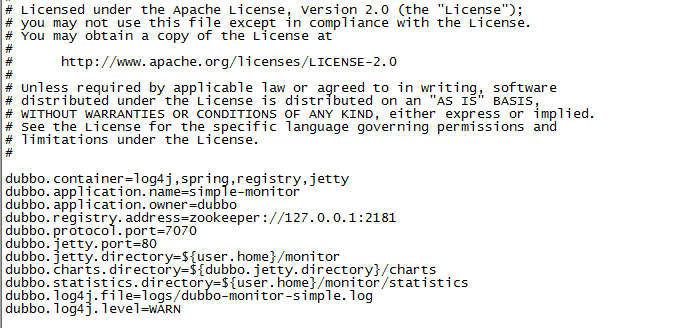
tar -xvf dubbo-monitor-simple-2.5.4.tar.gz -C /usr/local/dubbo

/usr/local/dubbo/bin/server.sh start

* 1. 修改配置文件

vi conf/dubbo.properties

修改后如下：



* 1. 添加启动

Vi /etc/init.d/duboomonitor

#!/bin/bash

#chkconfig:2345 20 90

#description:dubbomonitor

#processname:dubbomonitor

# Source function library.

start()

{

su root /usr/local/dubbomonitor/bin/start.sh

}

stop()

{

su root /usr/local/dubbomonitor/bin/stop.sh

}

restart()

{

su root /usr/local/dubbomonitor/bin/restart.sh

}

case $1 in

start) start;;

stop) stop;;

restart) restart;;

\*) echo "require start|stop|restart" ;;

esac

1. 安装redis主从

tar zxf redis-3.0.2.tar.gz

mv redis-3.0.2 /usr/local/redis

cd /usr/local/redis/

make

make install

vi /etc/init.d/redis

#!/bin/sh

# chkconfig: 345 86 14

# description: Startup and shutdown script for Redis

REDIS\_DIR=/usr/local

#export $REDIS\_DIR

REDIS\_CONF=/etc/redis/redis.conf

REDIS\_PID=/var/run/redis.pid

case $1 in

'start'|'s' )

if test -x $REDIS\_DIR/bin/redis-server

then

echo "Starting Redis..."

if $REDIS\_DIR/bin/redis-server $REDIS\_CONF

then

echo "OK"

else

echo "failed"

fi

else

echo "Couldn't find Redis Server ($REDIS\_DIR/bin/redis-server)"

fi

;;

'stop' )

echo "Stopping Redis..."

kill `cat $REDIS\_PID`

;;

'restart'|'reload'|'r' )

${0} stop

${0} start

;;

'list'|'l' )

ps aux | egrep '(PID|redis-server)'

;;

\*)

echo "usage: `basename $0` {start|restart|reload|stop|list}"

esac

chmod +x /etc/init.d/redis

mkdir /etc/redis

mkdir /data/redis

cp /usr/local/redis/redis.conf /etc/redis/

vi /etc/redis/redis.conf

daemonize yes ##启动后台

logfile "/var/log/redis.log" ##指定日志文件位置

dir /data/redis ##指定数据存储路径

Openvpn

yum install http://dl.fedoraproject.org/pub/epel/6/x86\_64/epel-release-6-8.noarch.rpm  
  
安装各种依赖关系  
yum -y install gcc gcc-c++ autoconf libjpeg libjpeg-devel libpng libpng-devel freetype freetype-devel libxml2 libxml2-devel zlib zlib-devel glibc glibc-devel glib2 glib2-devel bzip2 bzip2-devel ncurses ncurses-devel curl curl-devel e2fsprogs e2fsprogs-devel krb5 krb5-devel libidn libidn-devel openssl openssl-devel openldap openldap-devel nss\_ldap openldap-clients openldap-servers  
安装所需服务  
yum install pam\_krb5 pam\_mysql pam pam-devel  
yum install mysql mysql-server mysql-devel mysql-libs  
  
安装lzo  
wget http://www.oberhumer.com/opensource/lzo/download/lzo-2.03.tar.gz  
cd lzo-2.03 && ./configure && make && make install  
  
添加路径：  
cat >> /etc/ld.so.conf << EOF  
/lib  
/lib64  
/usr/lib  
/usr/lib64  
/usr/local/lib  
/usr/local/lib64  
EOF （退出）  
ldconfig  
  
注：这个文件记录了编译时使用的动态链接库的路径。  
默认情况下，编译器只会使用/lib和/usr/lib这两个目录下的库文件  
如果你安装了某些库，比如在安装gtk+-2.4.13时它会需要glib-2.0 >= 2.4.0,辛苦的安装好glib后  
没有指定 --prefix=/usr 这样glib库就装到了/usr/local下，而又没有在/etc/ld.so.conf中添加/usr/local/lib  
=================  
库文件的路径如 /usr/lib 或 /usr/local/lib 应该在 /etc/ld.so.conf 文件中，这样 ldd 才能找到这个库。在检查了这一点后，要以 root 的身份运行 /sbin/ldconfig 。  
=================  
将/usr/local/lib加入到/etc/ld.so.conf中，这样安装gtk时就会去搜索/usr/local/lib,同样可以找到需要的库  
将/usr/local/lib加入到/etc/ld.so.conf也是必须的，这样以后安装东东到local下，就不会出现这样的问题了。  
  
ldconfig：   
它是一个程序，通常它位于/sbin下，是root用户使用的东东。具体作用及用法可以man ldconfig查到  
简单的说，它的作用就是将/etc/ld.so.conf列出的路径下的库文件 缓存到/etc/ld.so.cache 以供使用  
因此当安装完一些库文件，(例如刚安装好glib)，或者修改ld.so.conf增加新的库路径后，需要运行一下/sbin/ldconfig  
使所有的库文件都被缓存到ld.so.cache中，如果没做，即使库文件明明就在/usr/lib下的，也是不会被使用的  
EOF本意是 End Of File,表明到了文件末尾  
  
**安装openvpn**  
wget http://down1.chinaunix.net/distfiles/openvpn-2.0.7.tar.gz  
tar -zxvf openvpn-2.0.7.tar.gz  
cd openvpn-2.0.7/  
./configure --prefix=/usr/local/openvpn && make && make install  
mkdir -p /etc/openvpn  
cd /root/openvpn-2.0.7  
cp -R easy-rsa /etc/openvpn  
cd /etc/openvpn/easy-rsa/2.0/  
cp vars vars\_bak  
  
如果下载其他版本的openvpn  如果没有easy-rsa  请到github下载  
<https://github.com/openvpn/easy-rsa>  
  
  
修改Vars文件的默认信息（可修改，可不修改，这些信息会在连接的日志中显示）  
vim vars  
###最下面修改内容：  
export KEY\_COUNTRY="CN"  
export KEY\_PROVINCE="BJ"  
export KEY\_CITY="beijing"  
export KEY\_ORG="beijingidc"  
export KEY\_EMAIL="你的邮箱地址"  
生成服务器及客户端需要的验证文件等。  
  
[root@monitor 2.0]# source ./vars

NOTE: If you run ./clean-all, I will be doing a rm -rf on /etc/openvpn/easy-rsa/2.0/keys

[root@monitor 2.0]# ./clean-all

[root@monitor 2.0]# ./build-ca ca

Generating a 1024 bit RSA private key

......++++++

........................++++++

writing new private key to 'ca.key'

-----

You are about to be asked to enter information that will be incorporated

into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

-----

Country Name (2 letter code) [CN]:

State or Province Name (full name) [SH]:

Locality Name (eg, city) [SH]:

Organization Name (eg, company) [17Money]:

Organizational Unit Name (eg, section) []:

Common Name (eg, your name or your server's hostname) [ca]:

Email Address [dengning@17money.com]:

[root@monitor 2.0]# ./build-key-server server

Generating a 1024 bit RSA private key

.........................++++++

.++++++

writing new private key to 'server.key'

-----

You are about to be asked to enter information that will be incorporated

into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

-----

Country Name (2 letter code) [CN]:

State or Province Name (full name) [SH]:

Locality Name (eg, city) [SH]:

Organization Name (eg, company) [17Money]:

Organizational Unit Name (eg, section) []:

Common Name (eg, your name or your server's hostname) [server]:

Email Address [dengning@17money.com]:

Please enter the following 'extra' attributes

to be sent with your certificate request

A challenge password []:jjjr@2015

An optional company name []:

Using configuration from /etc/openvpn/easy-rsa/2.0/openssl.cnf

Check that the request matches the signature

Signature ok

The Subject's Distinguished Name is as follows

countryName :PRINTABLE:'CN'

stateOrProvinceName :PRINTABLE:'SH'

localityName :PRINTABLE:'SH'

organizationName :PRINTABLE:'17Money'

commonName :PRINTABLE:'server'

emailAddress :IA5STRING:'dengning@17money.com'

Certificate is to be certified until Jul 11 11:55:29 2025 GMT (3650 days)

Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y

Write out database with 1 new entries

Data Base Updated

[root@monitor 2.0]# ./build-dh

Generating DH parameters, 1024 bit long safe prime, generator 2

This is going to take a long time

.........+...................................+..................+...+................................................................++\*++\*++\*