ULE项目报告书

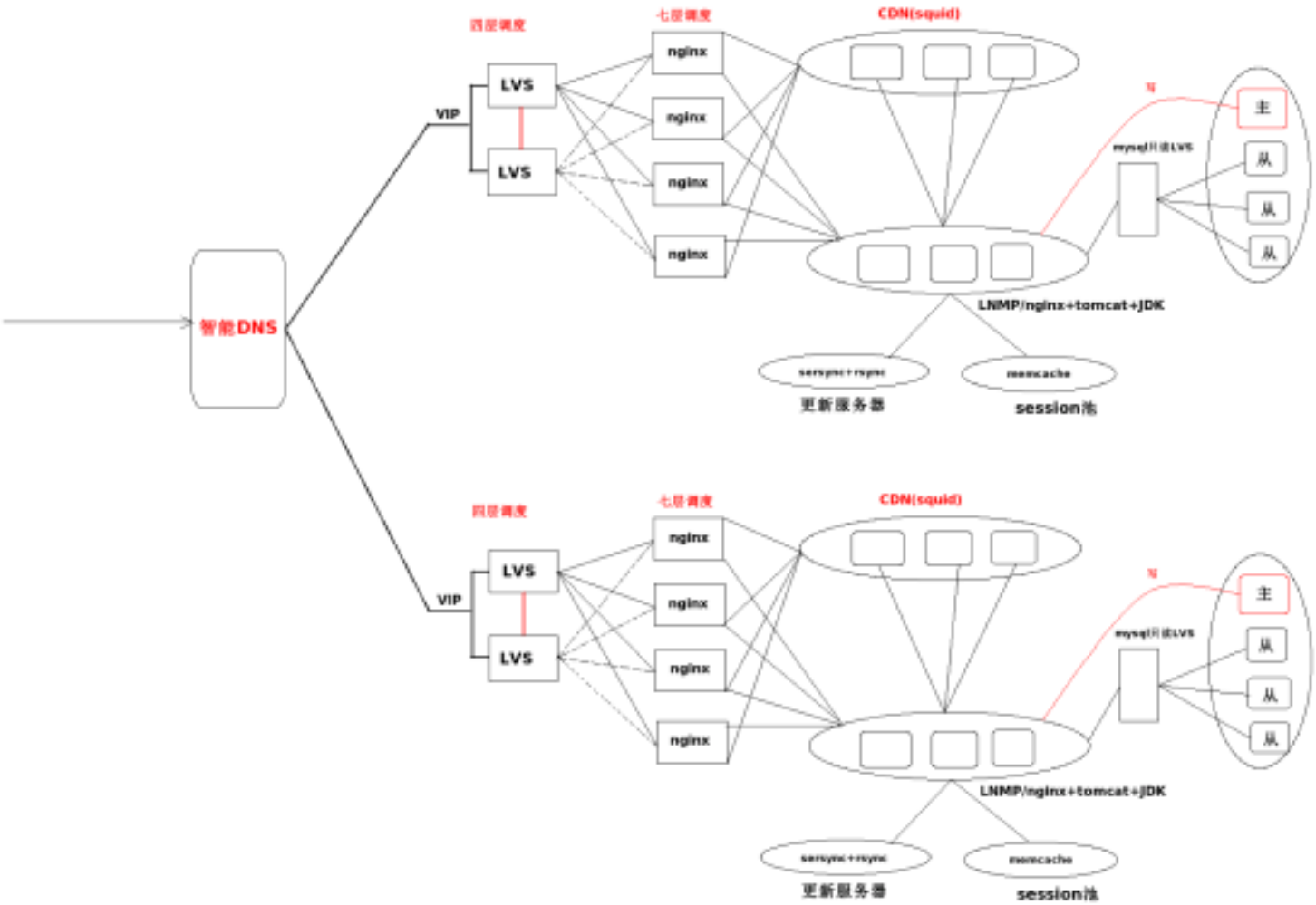
曾志成

摘要： 本文详细的阐述了高伸缩高可用大并发的网站架构（以JSP网站为例）的搭建。搭建过程中重温了四个月学习的MYSQL集群架构、Nginx+tomcat集群架构、内容分发网络(CDN)架构、Nginx的七层调度、负载均衡(LB)、LVS+heartbeat的四层调度、分布式存储(Glusterfs)集群的架构、Memcached缓存技术以及一系列服务的基本操作。

根据项目要求：高伸缩高可用大并发的网站架构（以JSP网站为例）的搭建要遵循以下要求：

1、MYSQL集群的搭建
2、nginx的七层的负载均衡集群
3、tomcat的集群（后端节点会话（session）的一致性）
4、利用分布性存储（glusterfs）实现页面一致性
5、引入CDN内容分发网络，实现网站静态元素加速
6、利用nginx七层分发器实现基于内容的分发
7、利LVS实现网站流量高效快速的分发
8、利用智能DNS实现大并发流量切割

总的拓扑图如下



项目准备工作：

本项目环境使用的是KVM虚拟机。由于条件有限，只能使用十五台虚拟机，根据本机的CPU和内存调整虚拟机的内存和CPU(eg：物理机内存3G,每台虚拟机之多分到256MB)，IP规划如下

172.25.254.254 classroom.example.com
172.25.254.250 i.example.com
172.25.254.12 f12
172.25.12.101 vm01.uplooking.com vm01 nginx1
172.25.12.102 vm02.uplooking.com vm02 tomcat1
172.25.12.103 vm03.uplooking.com vm03 tomcat2
172.25.12.104 vm04.uplooking.com vm04 mandg1
172.25.12.105 vm05.uplooking.com vm05 mandg2
172.25.12.106 vm06.uplooking.com vm06 squid1

172.25.12.107 vm07.uplooking.com vm07 squid2
172.25.12.108 vm08.uplooking.com vm08 lvs1
172.25.12.109 vm09.uplooking.com vm09 lvs2
172.25.12.110 vm10.uplooking.com vm10 nginx2
172.25.12.111 vm11.uplooking.com vm11 mysql-proxy
172.25.12.112 vm12.uplooking.com vm12 mysqlmaster1
172.25.12.113 vm13.uplooking.com vm13 mysqlmaster2
172.25.12.114 vm14.uplooking.com vm14 mysqlslave1
172.25.12.115 vm15.uplooking.com vm15 mysqlslave2

使用本地rhel6.3镜像搭建一个模版机

默认rhel6（rhel6.3）
一. 安装系统
镜像格式: qcow2
磁盘size: 20G
磁盘bus: virtio
网卡model: virtio

二. yum update（rhel7 yum install bash-completion&& yum update ,命令补全）

三. ntp，时间同步

四. network
chkconfig NetworkManager off
>/etc/udev/rules.d/70-persistent-net.rules （清空网卡的启动设置，避免复制的时候出现网卡累加）
vim /etc/sysconfig/network-scripts/ifcfg-eth0 (删除mac,uuid) --- br0
vim /etc/sysconfig/network-scripts/ifcfg-eth1 (删除mac,uuid) ---virbr0

五. yum configure

六. iptables/selinux
iptables -F
service iptables save
sed -ri '/^SELINUX=/cSELINUX=disabled' /etc/selinux/config

七. 配置本地console连接
--extra-args="console=tty0 console=ttyS0,115200"
(rhel6)#vim /etc/grub.conf 找到kernel启动项，在最后添加console
(rhel7)#vim /etc/sysconfig/grub
添加GRUB_CMDLINE_LINUX="--extra-args="console=tty0 console=ttyS0,115200" "
grub2-mkconfig -o /boot/grub2/grub.cfg

八. ssh连接慢的问题
vim /etc/ssh/sshd_config
UseDNS no
GSSAPIAuthentication no

九. 安全初始化配置
对制定的服务进行所需的安全初始化

十、创建初始化脚本
 vim /root/kvm_init.sh
 #!/bin/bash
read -p "Please input your hostname:" NAME
hostname $NAME.example.com
sed -i "s/^HOSTNAME=.*/HOSTNAME=$NAME.example.com/" /etc/sysconfig/network

cat > /etc/hosts << EOT
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1   localhost localhost.localdomain localhost6 localhost6.localdomain6
172.25.254.254 classroom.example.com
172.25.254.250 i.example.com
172.25.254.12 f12
172.25.12.101 vm01.uplooking.com vm01 nginx1
172.25.12.102 vm02.uplooking.com vm02 tomcat1
172.25.12.103 vm03.uplooking.com vm03 tomcat2
172.25.12.104 vm04.uplooking.com vm04 mandg1
172.25.12.105 vm05.uplooking.com vm05 mandg2
172.25.12.106 vm06.uplooking.com vm06 squid1

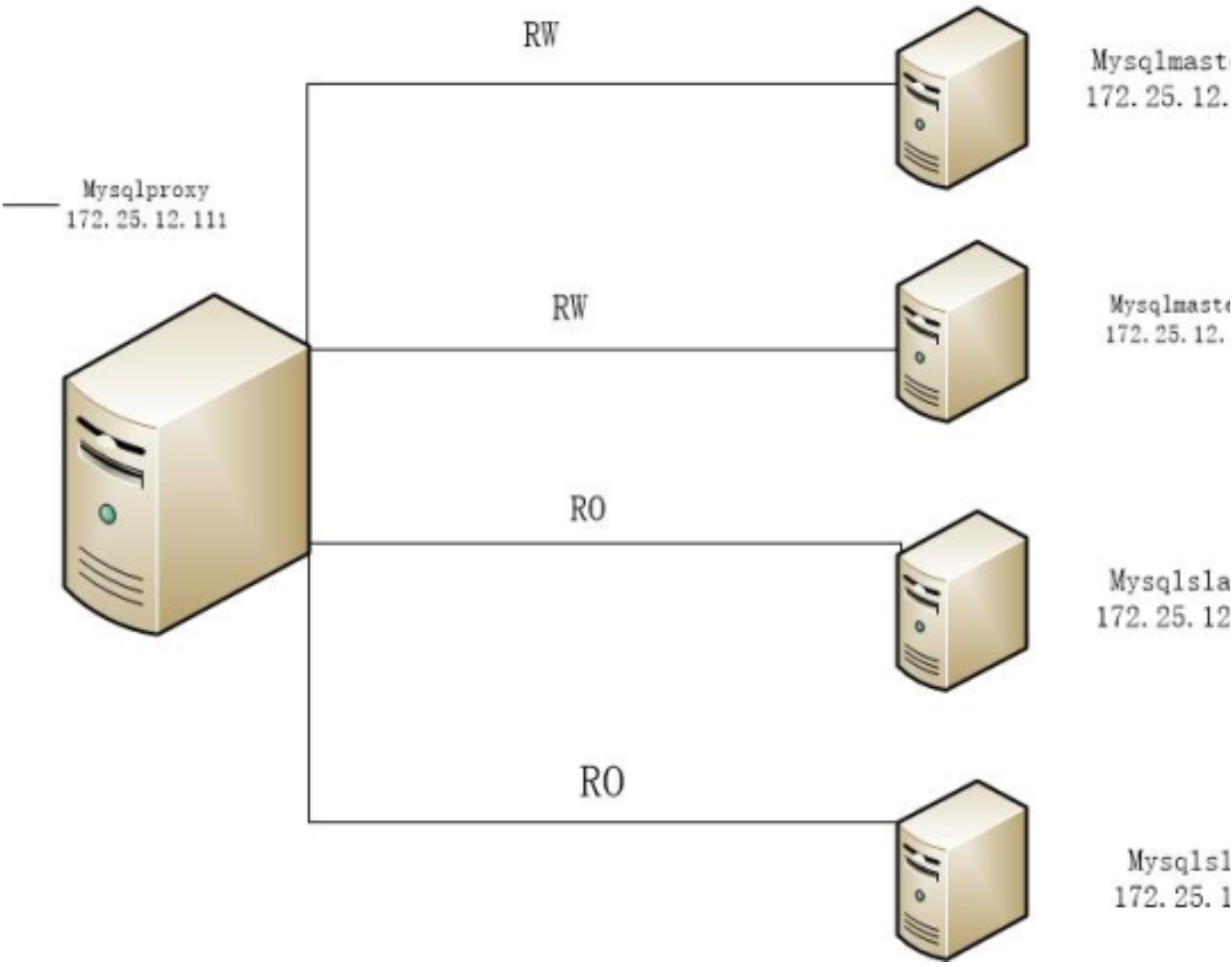
```
172.25.12.107 vm07.uplooking.com vm07 squid2
172.25.12.108 vm08.uplooking.com vm08 lvs1
172.25.12.109 vm09.uplooking.com vm09 lvs2
172.25.12.110 vm10.uplooking.com vm10 nginx2
172.25.12.111 vm11.uplooking.com vm11 mysql-proxy
172.25.12.112 vm12.uplooking.com vm12 mysqlmaster1
172.25.12.113 vm13.uplooking.com vm13 mysqlmaster2
172.25.12.114 vm14.uplooking.com vm14 mysqlslave1
172.25.12.115 vm15.uplooking.com vm15 mysqlslave2
EOT


iptables -F
service iptables save
chkconfig iptables off


setenforce 0
sed -i 's/^SELINUX=.*/SELINUX=disabled/' /etc/selinux/config

ntpdate -u 172.25.254.254 &> /dev/null



echo "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCbx4nrqEaq0iF2ns54AmAG6QPti0xt8VgdAelUMEmyjDjO2/
z0SjC3nS8hsBD1t5nDxGyYdOH5ewumTADPuvUPdGoSCJxdcceGFcYE21rm1QEvTPm44IcShyeER/
pxZOtNajys6RxU7yaU3avboDBIgifB4ogvXuUfVwUywIzMGhiBEPL/
qBF03I8Imf68yvFAa870Txf1ZU8NYjMMViBdG2ArOp4Cc6YoYIe7Hg9NvAZvRBfIbgopqlXFqEDu4jezOsqCNl87CaR1ezWU6gPmLm41W
+r7/Rz7ngKIML37e3JRrSm6QH0qcy4lS9/GKoM1iKp1dutDuQPky+Ek8kwB root@foundation12.ilt.example.com" >> /root/.ssh/
authorized_keys




cat > /etc/yum.repos.d/server.repo << EOT
[server]
name = rhel6.5 repos
baseurl = http://classroom.example.com/content/rhel6.5/x86_64/dvd/
enable=1
gpgcheck=0

[LB]
baseurl=http://classroom.example.com/content/rhel6.5/x86_64/dvd/LoadBalancer
gpgcheck=0

[HA]
baseurl=http://classroom.example.com/content/rhel6.5/x86_64/dvd/HighAvailability
gpgcheck=0

[RS]
baseurl=http://classroom.example.com/content/rhel6.5/x86_64/dvd/ResilientStorage
gpgcheck=0

[FS]
baseurl=http://classroom.example.com/content/rhel6.5/x86_64/dvd/ScalableFileSystem
gpgcheck=0
EOT

[SS]
baseurl=http://classroom.example.com/content/Storage/glusterfs/x86_64/dvd/rhel6/
gpgcheck=0

IP1=$(grep $NAME /etc/hosts |grep $NAME |awk '{print $1}')



cat > /etc/sysconfig/network-scripts/ifcfg-eth0  << EOT
DEVICE=eth0
TYPE=Ethernet
ONBOOT=yes
```

```
NM_CONTROLLED=no
BOOTPROTO=none
IPADDR=$IP1
NETMASK=255.255.255.0
GATEWAY=172.25.12.254
EOT

service network restart
reboot

#chmod +x  kvm_init.sh

chattr +i XXX.img (+a只能添加)
注：不能使用该镜像启动系统

创建模版完成后在物理机执行初始化命令建立十五台虚拟机

for i in {01..15}
do
    virt-clone -o rhel6.3 -n vm$i -f /var/lib/libvirt/images/vm$i.qcow2
    sed -i "s/domain-rhel6u3/domain-vm$i/" /etc/libvirt/qemu/vm$i.xml
    virsh define /etc/libvirt/qemu/vm$i.xml
    virsh start vm$i
done

使用console进入虚拟机执行初始化脚本
virsh console vm$i
./kvm_init.sh

检查脚本的运行是否正常
for i in {01..15}
do
    ssh root@172.25.12.1$i date
done

到此，初步的初始化操作完成。
```

一、MYSQL集群的搭建(Mysql-Cluster+Mysql-proxy)

功能实现：优点-----非索引数据的磁盘存储、增加数据节点能够在线扩展、使用ndinfo表来管理集群、配置和管理集群的脚本、多线程操作、下推（push-down）的关联（现在称为自适应查询本地化）、能够处理BLOB列和很多列的表、集中式的用户管理，以及通过像memcache协议一样的NDB API来实现NoSQL访问，通过代理服务器进行通讯，代理服务器实现负载均衡，实现读写分离。

缺点-----代理服务器由于条件有限没有做HA，容易产生单点故障，从而影响性能。



架构拓扑图：

运用脚本实现之自动化搭建

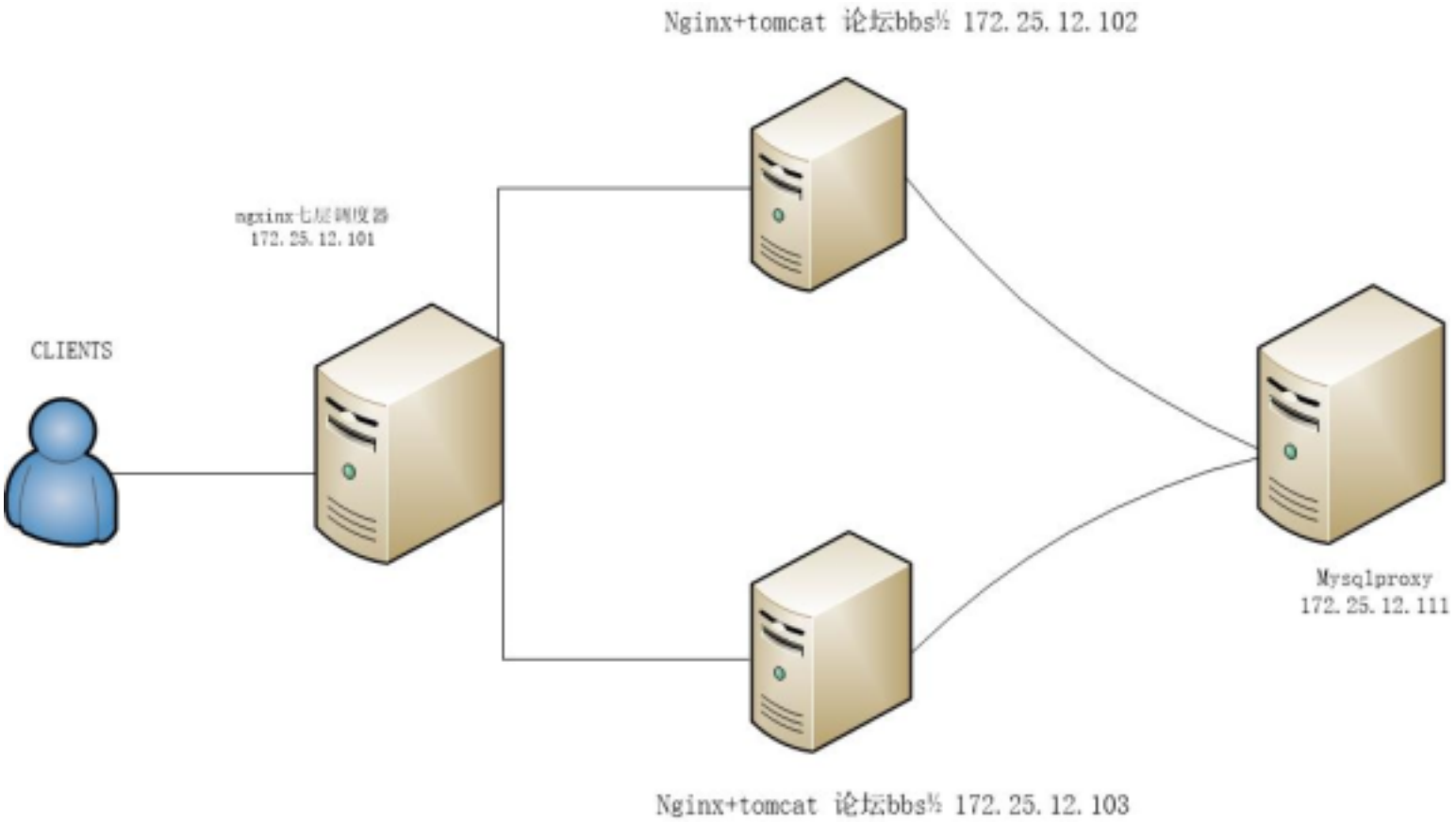
```
#vim auto_mysql.sh
#!/bin/bash
#同步读写分离

cat > /tmp/hosts << EOT
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
172.25.254.254 classroom.example.com
172.25.254.250 i.example.com
172.25.254.12 f12
172.25.12.111 vm11.uplooking.com vm11 mysql-proxy
172.25.12.112 vm12.uplooking.com vm12 mysqlmaster1
172.25.12.113 vm13.uplooking.com vm13 mysqlmaster2
172.25.12.114 vm14.uplooking.com vm14 mysqlslave1
172.25.12.115 vm15.uplooking.com vm15 mysqlslave2
EOT

for i in {11..15}
do
        scp /tmp/hosts root@172.25.12.1$i:/etc/hosts
done
```

```
for i in {11..15}
do
	ssh root@172.25.12.1$i "setenforce 0 ; sed -i s/^SELINUX=.*/SELINUX=disabled/  /etc/selinux/config ;  iptables -F ;
service iptables save;service iptables stop;chkconfig iptables off"
done

for i in {11..15}
do
	ssh root@172.25.12.1$i "yum -y install  rsync wget rpc-bind nfs-utils  vim rsync net-tools"
done
#对mysql服务器进行操作
for i in {12..15}
do
	ssh root@172.25.12.1$i "cd /etc/yum.repos.d/ &&  wget http://classroom.example.com/materials/mysql-5.7.repo  ;  wget
http://classroom.example.com/materials/thirdpart.repo ; yum clean all ; yum repolist"
done

for i in {12..15}
do
	ssh root@172.25.12.1$i "yum -y install mysql-server mysql"
done

for i in {12..15}
do
	ssh root@172.25.12.1$i "service mysqld start ;chkconfig mysqld on;
mysql -uroot <<END
delete from mysql.user where user='';
update mysql.user set password=password('uplooking') where user='root';
delete from mysql.db where user='';
flush privileges;
END"
done
#mysql_secure_installation set password uplooking and restart mysql

cat > /tmp/galera.cnf <<EOT
[galera]
wsrep_on=ON
wsrep_provider=/usr/lib64/galera/libgalera_smm.so
wsrep_cluster_address='gcomm://'
#所有node必须一样
wsrep_cluster_name='galera'
#节点地址
wsrep_node_address='172.25.12.112'
#节点名称
wsrep_node_name='galera1'
#Snapshot State Transter快照状态转移方法:mysqldump/rsync,默认mysqldump
wsrep_sst_method=rsync
#binlog的格式也有三种：STATEMENT，ROW，MIXED
binlog_format=row
default_storage_engine=InnoDB
#调整锁策略的
innodb_autoinc_lock_mode=2
bind-address=0.0.0.0
EOT

for i in {12..15}
do
	scp /tmp/galera.cnf root@172.25.12.1$i:/etc/my.cnf.d/
	ssh root@172.25.12.11 "service mysqld start"
done

	ssh root@172.25.12.113 "sed -i 's/wsrep_cluster_address=.*/wsrep_cluster_address='gcomm:\/\/172.25.12.112'/'  /etc/
my.cnf.d/galera.cnf"
	ssh root@172.25.12.113 "sed -i 's/wsrep_node_address=.*/wsrep_node_address='172.25.12.113/'  /etc/my.cnf.d/
galera.cnf"
	ssh root@172.25.12.113 "sed -i 's/wsrep_node_name=.*/wsrep_node_name=mysqlmaster2/'  /etc/my.cnf.d/galera.cnf"

	ssh root@172.25.12.114 "sed -i 's/wsrep_cluster_address=.*/wsrep_cluster_address='gcomm:\/\/172.25.12.112'/'  /etc/
my.cnf.d/galera.cnf"
```

```
	ssh root@172.25.12.114 "sed -i 's/wsrep_node_address=.*/wsrep_node_address='172.25.12.114'/'  /etc/my.cnf.d/
galera.cnf"
	ssh root@172.25.12.114 "sed -i 's/wsrep_node_name=.*/wsrep_node_name=mysqlslave1/'  /etc/my.cnf.d/galera.cnf"

	ssh root@172.25.12.115 "sed -i 's/wsrep_cluster_address=.*/wsrep_cluster_address='gcomm:\/\/172.25.12.112'/'  /etc/
my.cnf.d/galera.cnf"
	ssh root@172.25.12.15 "sed -i 's/wsrep_node_address=.*/wsrep_node_address='172.25.12.115'/'  /etc/my.cnf.d/
galera.cnf"
	ssh root@172.25.12.15 "sed -i 's/wsrep_node_name=.*/wsrep_node_name=mysqlslave2/'  /etc/my.cnf.d/galera.cnf"


for i in {12..15}
do
	ssh root@172.25.12.1$i   "service mysqld restart"
done

	ssh root@172.25.12.112 "mysql -uroot -puplooking -e 'show global status like 'wsrep_cluster%'' | grep wsrep_cluster_size
| awk -F'' print{$2} >>/tmp/num"
	ssh root@172.25.12.112 "[ `echo $(cat /tmp/num)` -eq 4 && echo "OK" ]"

#install dbproxy

	ssh root@172.25.12.111 "yum -y install install  mysql  gcc* lua lua-devel libevent libevent-devel glib2 glib2-devel
pkgconfig  flex openssl-devel"

	ssh root@172.25.12.111 "wget http://172.25.254.254/content/courses/db100/rhel7.2/materials/mysql-proxy-0.8.5.tar.gz"
	ssh root@172.25.12.111 "tar -zxf mysql-proxy-0.8.5.tar.gz ;cd mysql-proxy-0.8.5"
	ssh root@172.25.12.111 "cd mysql-proxy-0.8.5 ; ./configure --prefix=/usr/local/mysql-proxy ; make ; make install"
	ssh root@172.25.12.111 "cp ./mysql-proxy-0.8.5/lib/rw-splitting.lua /usr/local/mysql-proxy/"
	ssh root@172.25.12.111 "echo "cd /usr/local/mysql-proxy/ && ./mysql-proxy -P 0.0.0.0:3306 -b 172.25.12.111:3306 -b
172.25.12.112 -r 172.25.12.113:3306 -r 172.25.12.114:3306  -s /usr/local/mysql-proxy/rw-splitting.lua" >> /etc/rc.d/rc.local"
	ssh root@172.25.12.111 "chmod +x /etc/rc.d/rc.local ;source /etc/rc.d/rc.local"

ssh root@172.25.12.111 "mysql -uroot -puplooking <<END
create database db1;
use db1;
create table t1 (num int not null);
insert into t1 values (1);
grant all on db1.* to 'dbadmin'@'%' identified by 'uplooking';
grant all on *.* to 'dbadmin'@'%' identified by 'uplooking' with grant option;
flush privileges;
END"

#安装完成后测试数据库，测试完成后把测试库删除，使用数据库是通过代理172.25.12.111进行连接
```

二，nginx的七层的负载均衡集群

架构拓扑图:



1，在后端节点搭建nginx+tomcat
1)下载、安装、启动软件
lftp 172.25.254.250 → cd /notes/ula/item -→ mirror softwares
cd soft# tar xf jdk-7u15-linux-x64.tar.gz -C /opt
tar xf apache-tomcat-7.0.37.tar.gz -C /opt
cp tomcat-init.sh /etc/rc.d/init.d/tomcat
chmod +x /etc/rc.d/init.d/tomcat
cd /opt
mv apache-tomcat-7.0.37/ tomcat
mv jdk1.7.0_15/ jdk
cd /opt/tomcat/bin/
tar xf commons-daemon-native.tar.gz
cd commons-daemon-1.0.13-native-src/unix/
./configure --with-java=/opt/jdk && make
cp jsvc /opt/tomcat/bin/
service tomcat start
chkconfig tomcat on

2)配置bbs1.uplooking.com/bbs2.uplooking.com两个网站(172.25.12.102)
#vim /opt/tomcat/conf/server.xml
```
<?xml version='1.0' encoding='utf-8'?>
<Server port="8005" shutdown="SHUTDOWN">
  <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />
  <Listener className="org.apache.catalina.core.JasperListener" />
  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />
  <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />
  <GlobalNamingResources>
    <Resource name="UserDatabase" auth="Container"
              type="org.apache.catalina.UserDatabase"
              description="User database that can be updated and saved"
              factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
              pathname="conf/tomcat-users.xml" />
  </GlobalNamingResources>
  <Service name="Catalina">
    <Connector port="8080" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />
    <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
```

```
    <Engine name="Catalina" defaultHost="bbs1.uplooking.com">
      <Realm className="org.apache.catalina.realm.LockOutRealm">
        <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
               resourceName="UserDatabase"/>
      </Realm>
      <Host name="bbs1.uplooking.com"  appBase="webapps"
            unpackWARs="true" autoDeploy="true">
        <Valve className="org.apache.catalina.valves.AccessLogValve" directory="/var/log/tomcat"
               prefix="bbs1.uplooking.com_access" suffix=".log"
               pattern="%h %l %u %t &quot;%r&quot; %s %b" />
        <Context path="" docBase="/webroot/bbs1.uplooking.com" />
      </Host>

      <Host name="bbs2.uplooking.com"  appBase="webapps"
            unpackWARs="true" autoDeploy="true">
        <Valve className="org.apache.catalina.valves.AccessLogValve" directory="/var/log/tomcat"
               prefix="bbs2.uplooking.com_access" suffix=".log"
               pattern="%h %l %u %t &quot;%r&quot; %s %b" />
        <Context path="" docBase="/webroot/bbs2.uplooking.com" />
      </Host>
    </Engine>
  </Service>
</Server>

# mkdir /var/log/tomcat
# mkdir /webroot/bbs{1,2}.uplooking.com -p
# cd /root/softwares
# unzip ejforum-2.3.zip
# unzip JavaCenter_Home_2.0_Source_UTF8.zip
# mv JavaCenter_Home_2.0_Source_UTF8/source/WebRoot/* /webroot/bbs1.uplooking.com/
# mv ejforum-2.3/ejforum/* /webroot/bbs2.uplooking.com/
# service tomcat restart

3)连接数据库，为两个网站创建用户和数据库
#mysql -udbadmin -puplooking -h172.25.12.111<<EOF
create database bbs1 default charset utf8;
create database bbs2 default charset utf8;
grant all on bbs1.* to runbbs1@'%' identified by 'uplooking';
grant all on bbs2.* to runbbs2@'%' identified by 'uplooking';

4)配置网络连接到数据库上(测试机要写/etc/hosts文件)
bbs1通过页面安装
# firefox http://bbs1.uplooking.com:8080/install

bbs2手动配置连接到数据库
# yum -y install mysql
# mysql -urunbbs2 -puplooking -h172.25.12.111  bbs2 < /root/ejforum-2.3/install/script/easyjforum_mysql.sql
# vim /webroot/bbs2.uplooking.com/WEB-INF/conf/config.xml
<?xml version="1.0" encoding="UTF-8"?>
<config>
      <database maxActive="10" maxIdle="10" minIdle="2" maxWait="10000"
                username="runbbs2" password="uplooking"
                driverClassName="com.mysql.jdbc.Driver"
                url="jdbc:mysql://172.25.12.111:3306/bbs2?
characterEncoding=gbk&amp;autoReconnect=true&amp;autoReconnectForPools=true&amp;zeroDateTimeBehavior=convertToNull
                sqlAdapter="sql.MysqlAdapter"/>
      <system adminUser="admin"/>
      <misc>
            <maxMemberPages>20</maxMemberPages>
            <maxSessionPosts>10</maxSessionPosts>
            <maxFavorites>50</maxFavorites>
            <maxShortMsgs>50</maxShortMsgs>
            <maxAvatarPixels>150</maxAvatarPixels>
      </misc>
</config>
# service tomcat restart
# firefox http://bbs2.uplooking.com:8080/ &

 5)nginx实现页面的动静分离
 #cd  /root/softwares
 # rpm -ivh /root/nginx-1.4.7-1.el6.x86_64.rpm
```

```
# vim /etc/nginx/nginx.conf
user  nginx nginx;
worker_processes  1;
error_log  /var/log/nginx/error.log  info;
pid        /var/run/nginx.pid;


events {
    worker_connections  65535;
    use epoll;
}


http {
    include       mime.types;
    default_type  application/octet-stream;
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';
    access_log  /var/log/nginx/access.log  main;
    sendfile        on;
    tcp_nopush     on;
    keepalive_timeout  30;
    gzip  on;
        upstream tomcat_upstream {
                server 127.0.0.1:8080;
        }
    server {
        listen       80;
        server_name  bbs1.uplooking.com;
        charset utf8;
        access_log  /var/log/nginx/bbs1.uplooking.com.access.log  main;

        location / {
                root   /webroot/bbs1.uplooking.com;
                index  index.html index.htm index.jsp;
        }

        location ~ \.jsp$ {
                proxy_pass http://tomcat_upstream;
                proxy_set_header Host $host;
                proxy_set_header X-Forward-For $remote_addr;
        }
    }

    server {
        listen       80;
        server_name  bbs2.uplooking.com;
        charset utf8;
        access_log  /var/log/nginx/bbs2.uplooking.com.access.log  main;

        location / {
                root   /webroot/bbs2.uplooking.com;
                index  index.html index.htm index.jsp;
        }

        location ~ \.jsp$ {
                proxy_pass http://tomcat_upstream;
                proxy_set_header Host $host;
                proxy_set_header X-Forward-For $remote_addr;
        }

        location ~ ^/forum-[0-9]-[0-9]-[0-9]\.html$ {
                proxy_pass http://tomcat_upstream;
                proxy_set_header Host $host;
                proxy_set_header X-Forward-For $remote_addr;
        }
    }
}

# mkdir /var/log/nginx
```

```
#mkdir /var/run/nginx -p
# ulimit -HSn 65535
# vim /etc/security/limits.conf
root           -       nproc          65535
root           -       nofile         65535
# cp /root/softwares/nginx.init.d   /etc/rc.d/init.d/nginx
# service nginx start
# chkconfig nginx on
```

2,将部署的网站同步到另一节点(tomcat2)

1)初始化nginx

```
# lftp 172.25.254.250 → cd /notes/ula/item   -→ mirror  softwares
#cd /root/softwares
# rpm -ivh nginx-1.4.7-1.el6.x86_64.rpm
# mkdir /var/log/nginx
#mkdir /var/run/nginx -p
# ulimit -HSn 65535
# vim /etc/security/limits.conf
root           -       nproc          65535
root           -       nofile         65535
# cp /root/softwares/nginx.init.d   /etc/rc.d/init.d/nginx
# rsync -avzR /opt/ /webroot/ /etc/rc.d/init.d/tomcat  /etc/rc.d/init.d/nginx  /etc/nginx/nginx.conf /etc/security/limits.conf
root@172.25.12.103:/
# service tomcat start
# service nginx start
# chkconfig tomcat on
# chkconfig nginx on
```

2)后端节点建立测试页面

172.25.12.102

```
# vim /webroot/bbs1.uplooking.com/test.jsp
<html>
      <body bgcolor="red">
            <center>
            <%out.print(request.getSession().getId()) ;%>
            <h1>Tomcat1</h1>
      </body>
</html>
```

172.25.12.103

```
# vim /webroot/bbs1.uplooking.com/test.jsp
<html>
      <body bgcolor="blue">
            <center>
            <%out.print(request.getSession().getId()) ;%>
            <h1>Tomcat2</h1>
      </body>
</html>
```

3)在172.25.12.111安装和使用七层负载均衡器(nginx)

```
# lftp 172.25.254.250 → cd /notes/ula/item   -→ mirror  softwares
#cd /root/softwares
# rpm -ivh nginx-1.4.7-10.x86_64.rpm
# mkdir /var/log/nginx
#mkdir /var/run/nginx -p
# ulimit -HSn 65535
# vim /etc/security/limits.conf
root           -       nproc          65535
root           -       nofile         65535
# cp /root/softwares/nginx.init.d   /etc/rc.d/init.d/nginx
# vim /etc/nginx/conf/nginx.conf
user  nginx nginx;
worker_processes  1;
error_log  /var/log/nginx/error.log  info;
pid        /var/run/nginx.pid;


events {
   worker_connections  15000;
   use epoll;
}
```

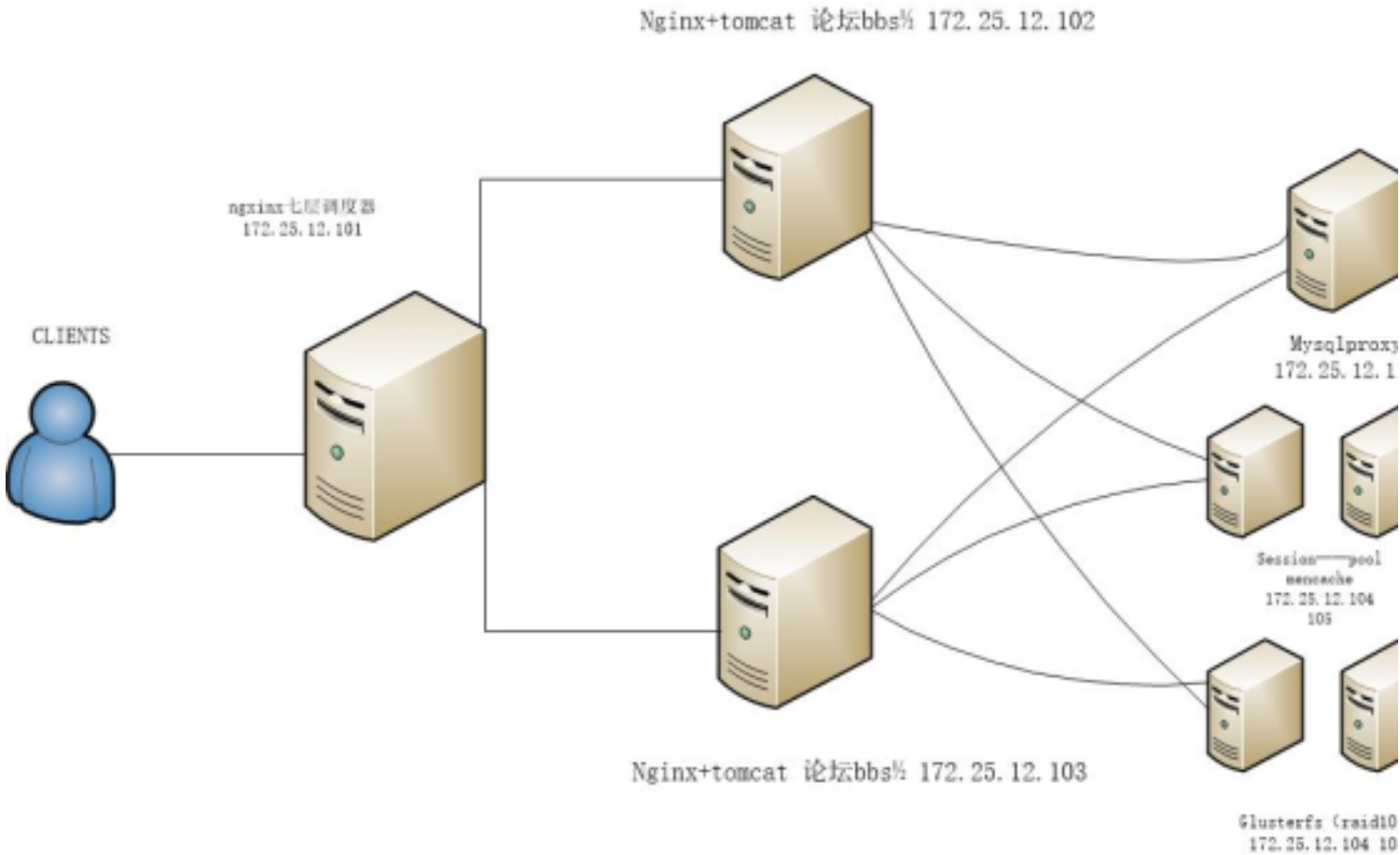
```
http {
    include       mime.types;
    default_type  application/octet-stream;
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';
    access_log  /var/log/nginx/access.log  main;
    sendfile        on;
    tcp_nopush     on;
    keepalive_timeout  30;
    gzip  on;
        upstream nginx_upstream {
                server 172.25.12.102:80;
                server 172.25.12103:80;
        }

        upstream tomcat_upstream {
                server 172.25.12.102:8080 weight=10;
                server 172.25.12.103:8080 weight=10;
        }


    server {
        listen       80;
        server_name  localhost;
        charset utf8;

        location / {
                index  index.html index.htm index.jsp;
                proxy_pass http://nginx_upstream;
                proxy_set_header Host $host;
                proxy_set_header X-Forward-For $remote_addr;
        }

        location ~ \.jsp$ {
                proxy_pass http://tomcat_upstream;
                proxy_set_header Host $host;
                proxy_set_header X-Forward-For $remote_addr;
        }
        location ~ ^/forum-[0-9]-[0-9]-[0-9]\.html$ {
                proxy_pass http://tomcat_upstream;
                proxy_set_header Host $host;
                proxy_set_header X-Forward-For $remote_addr;
        }


    }
}
```

4测试七层负载均衡调度器
1)关掉后端nginx
firefox http://bbs1.uplooking.com/index.jsp & --结果看不到静态元素

2）关掉是后端的tomcat
firefox http://bbs1.uplooking.com/template/default/image/guest_bg.jpg & – – 可以看到图片

三、利用Memcached实现tomcat的集群（后端节点会话（session）的一致性）和利用分布性存储（glusterfs）实现页面一致性

条件有限，只能将Memcached和glusterfs搭建在同两台服务器上(172.25.12.104 vm04.uplooking.com vm04 mandg1
172.25.12.105 vm05.uplooking.com vm05 mandg2)
实现后端节点会话一致性，页面一致性。
拓扑图：



一)Mencached实现会话一致性，即用户访问该页面，则会一直锁定在同一页面

1、在mandg1和mandg2部署Mencached

```
# yum -y install memcached
# service memcached start
# chkconfig memcached on
# netstat -tunlp |grep memcache
tcp        0      0 0.0.0.0:11211               0.0.0.0:*                   LISTEN      9052/memcached
tcp        0      0 :::11211                    :::*                        LISTEN      9052/memcached
udp        0      0 0.0.0.0:11211               0.0.0.0:*                               9052/memcached
udp        0      0 :::11211                    :::*                                    9052/memcached

#  ps aux |grep mem
498       9052  0.0  1.2 333920  3040 ?        Ssl  Apr07   0:01 memcached -d -p 11211 -u memcached -m 75 -c 1024 -P /var/run/
memcached/memcached.pid
```

如果内存使用不合理，则进入/etc/sysconfig/memcached 修改缓存大小。

2、在tomcat1和tomcat2添加会话用户(memcache client)

```
#lftp 172.25.254.250 --> cd /notes/ula/item -->mirror softwares
#cd softwares/session/
#mv *.jar /opt/tomcat/lib/
# vim /opt/tomcat/conf/context.xml
<?xml version='1.0' encoding='utf-8'?>
<Context>
        <Manager className="de.javakaffee.web.msm.MemcachedBackupSessionManager"
                  memcachedNodes="n1:172.25.12.104:11211,n2:172.25.12.105:11211"
                  failoverNodes="n2"
                  requestUriIgnorePattern=".*\.(ico|png|gif|jpg|css|js)$"
                 transcoderFactoryClass="de.javakaffee.web.msm.serializer.kryo.KryoTranscoderFactory"
        />
   <WatchedResource>WEB-INF/web.xml</WatchedResource>
</Context>


# vim /opt/tomcat/conf/context.xml
<?xml version='1.0' encoding='utf-8'?>
<Context>
        <Manager className="de.javakaffee.web.msm.MemcachedBackupSessionManager"
```

```
                    memcachedNodes="n1:192.168.0.2:11211,n2:192.168.0.3:11211"
                    failoverNodes="n2"
                    requestUriIgnorePattern=".*\.(ico|png|gif|jpg|css|js)$"
                   transcoderFactoryClass="de.javakaffee.web.msm.serializer.kryo.KryoTranscoderFactory"
        />
  <WatchedResource>WEB-INF/web.xml</WatchedResource>
</Context>


 # service tomcat restart

 观察日志输出
# tail /opt/tomcat/logs/catalina-daemon.out  -f
INFO: MemcachedSessionService starts initialization... (configured nodes definition
n1:172.25.12.104:11211,n2:172.25.12.105:11211, failover nodes n2)
2017-04-07 12:43:31.477 INFO net.spy.memcached.MemcachedConnection:  Added {QA sa=/172.25.12.104:11211,
#Rops=0, #Wops=0, #iq=0, topRop=null, topWop=null, toWrite=0, interested=0} to connect queue
2017-04-07 12:43:31.478 INFO net.spy.memcached.MemcachedConnection:  Added {QA sa=/172.25.12.105:11211,
#Rops=0, #Wops=0, #iq=0, topRop=null, topWop=null, toWrite=0, interested=0} to connect queue

将修改好的tomcat同步到另一节点
# rsync -avzR /opt/tomcat root@172.25.12.103:/
# service tomcat restart

 3、测试
 # firefox http://bbs1.uplooking.com/test.jsp          --可以看到会话不会发现变化(会话一致性) --但是页面内容不一致

       二)glusterfs实现页面一致性，即用户访问的页面内容保持一致
       使用raid10保证数据一致，拓扑图(略)
       1、mandg1和mandg2各自添加两个同样大小的磁盘(vdb vdc),格式化和永久挂载
# mkfs.ext4 -L brick1 /dev/vdb(-L 加卷标)
# mkfs.ext4 -L brick2 /dev/vdc
# vim /etc/fstab
LABEL="brick1"        /strip-rep01          ext4    defaults       0 0
LABEL="brick2"        /strip-rep02          ext4    defaults       0 0

#mount -a
#df -h
# yum -y glusterfs.x86_64 glusterfs-fuse.x86_64 glusterfs-server.x86_64
# /etc/init.d/glusterd start
# chkconfig glusterd on
----------以上两个节点都要配置----------------------------------------------------------------
       2、配置gluster服务i端
在vm04节点操作
# gluster peer probe vm05.uplooking.com
peer probe: success.

# gluster peer status
Number of Peers: 1

Hostname: vm05.uplooking.com
Uuid: 59bdb8eb-d693-466c-a6ec-70d8d002e53f
State: Peer in Cluster (Connected)

# gluster volume create glusterfs stripe 2 replica 2  vm04.uplooking.com:/strip-rep01 vm04.uplooking.com:/strip-rep02
vm05.uplooking.com:/strip-rep01 vm05.uplooking.com:/strip-rep02 force
ps：force的添加---gluster默认raid10需要四台服务器，由于节省资源则在两台服务器上用四个硬盘模拟，不加force会报错。

# gluster volume info

Volume Name: glusterfs
Type: Striped-Replicate
Volume ID: 588864fc-ecd5-4321-b3ca-debfda3401eb
Status: Started
Number of Bricks: 1 x 2 x 2 = 4
Transport-type: tcp
Bricks:
Brick1: vm04.uplooking.com:/strip-rep01
Brick2: vm04.uplooking.com:/strip-rep02
Brick3: vm05.uplooking.com:/strip-rep01
```

Brick4: vm05.uplooking.com:/strip-rep02
Options Reconfigured:
performance.readdir-ahead: on

gluster volume start glusterfs

3、在tomcat1和tomcat2安装glusterfs客户端并挂载存储
yum -y install glusterfs glusterfs-fuse
service tomcat stop && service nginx stop
mount.glusterfs 172.25.12.104:/glusterfs /mnt
cp -av /webroot/* /mnt
umount /mnt
mount.glusterfs 192.168.0.4:/glusterfs /webroot/
service tomcat start
service nginx start

另一tomcat上
service tomcat stop && service nginx stop
mount.glusterfs 172.25.12.105:/glusterfs /webroot/
service tomcat start
service nginx start

4、客户端实现永久挂载分布式存储
vim /etc/rc.d/rc.local
#!/bin/sh
#
This script will be executed *after* all the other init scripts.
You can put your own initialization stuff in here if you don't
want to do the full Sys V style init stuff.

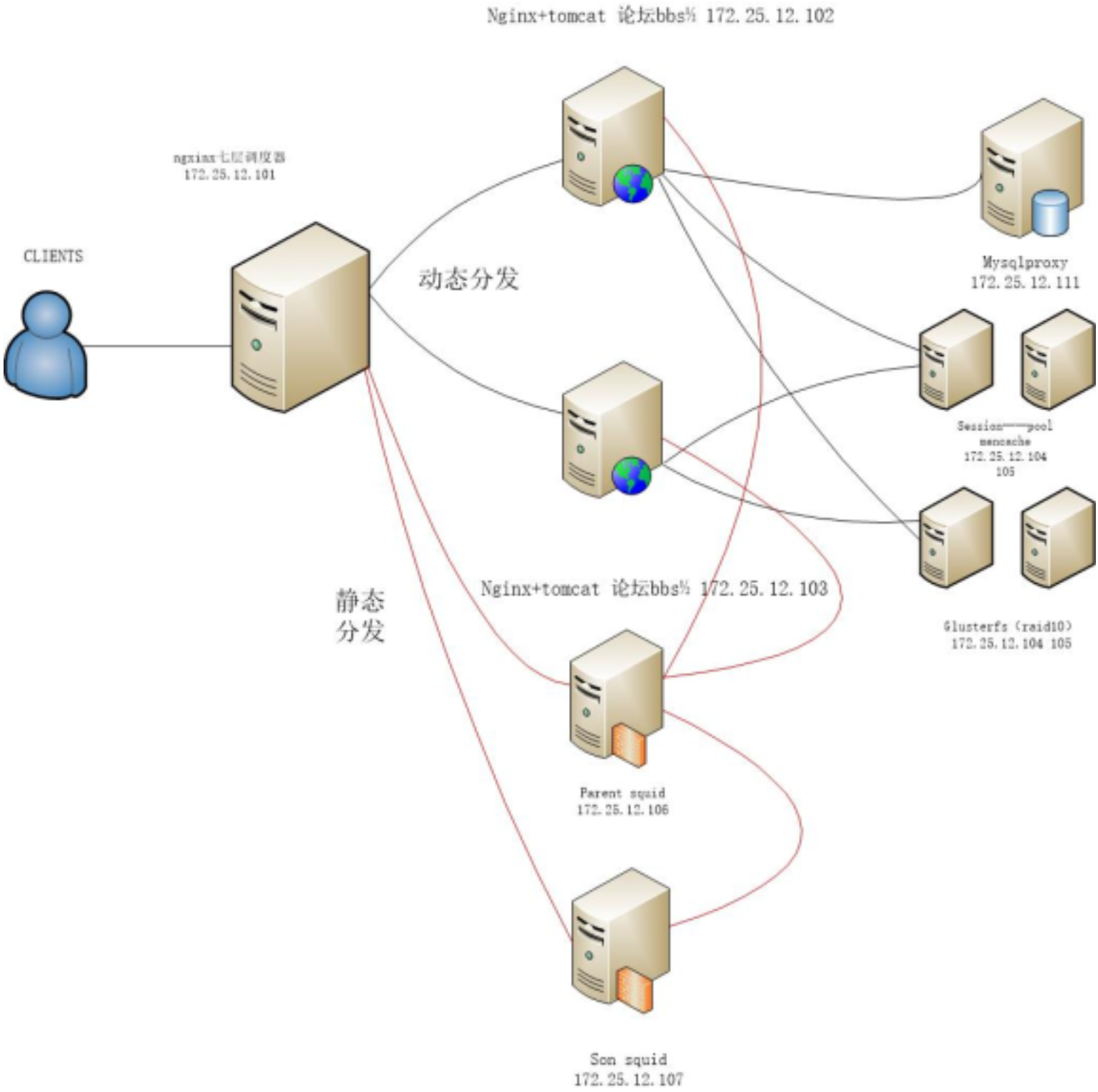
touch /var/lock/subsys/local
/usr/sbin/ntpdate 192.168.122.1
/bin/mount mount -t glusterfs vm04.uplooking.com:/glusterfs /webroot/

另一节点
vim /etc/rc.d/rc.local
#!/bin/sh
#
This script will be executed *after* all the other init scripts.
You can put your own initialization stuff in here if you don't
want to do the full Sys V style init stuff.

touch /var/lock/subsys/local
/usr/sbin/ntpdate 192.168.122.1
/bin/mount mount -t glusterfs vm05.uplooking.com:/glusterfs /webroot/

5、测试
firefox http://bbs2.uplooking.com/test.jsp 会话和页面内容均不会变化

四、引入CDN内容分发网络，实现网站静态元素加速
拓扑图



1、在squid1和squid2安装squid
yum -y instal squid

2 父squid配置如下
vim /etc/squid/squid.conf
Recommended minimum configuration:
#
acl manager proto cache_object
acl localhost src 127.0.0.1/32 ::1
acl to_localhost dst 127.0.0.0/8 0.0.0.0/32 ::1

Example rule allowing access from your local networks.
Adapt to list your (internal) IP networks from where browsing
should be allowed

```
acl localnet src 10.0.0.0/8     # RFC1918 possible internal network
acl localnet src 172.16.0.0/12  # RFC1918 possible internal network
acl localnet src 192.168.0.0/16 # RFC1918 possible internal network
acl localnet src fc00::/7       # RFC 4193 local private network range
acl localnet src fe80::/10      # RFC 4291 link-local (directly plugged) machines

acl SSL_ports port 443
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443         # https
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl Safe_ports port 3130        #icp
acl CONNECT method CONNECT

#
# Recommended minimum Access Permission configuration:
#
# Only allow cachemgr access from localhost
http_access allow manager localhost
http_access deny manager

# Deny requests to certain unsafe ports
http_access deny !Safe_ports

# Deny CONNECT to other than secure SSL ports
http_access deny CONNECT !SSL_ports

# We strongly recommend the following be uncommented to protect innocent
# web applications running on the proxy server who think the only
# one who can access services on "localhost" is a local user
#http_access deny to_localhost

#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#

# Example rule allowing access from your local networks.
# Adapt localnet in the ACL section to list your (internal) IP networks
# from where browsing should be allowed
http_access allow localnet
http_access allow localhost

# And finally deny all other access to this proxy
http_access deny all
http_port 3128 vhost vport
icp_port 3130
# Squid normally listens to port 3128
visible_hostname vm06.uplooking.com
# We recommend you to use at least the following line.
hierarchy_stoplist cgi-bin ?

# Uncomment and adjust the following to add a disk cache directory.
cache_mem 100 MB
cache_dir ufs /var/spool/squid 100 16 256

# Leave coredumps in the first cache dir
coredump_dir /var/spool/squid

# Add any of your own refresh_pattern entries above these.
refresh_pattern ^ftp:           1440    20%     10080
refresh_pattern ^gopher:        1440    0%      1440
refresh_pattern -i (/cgi-bin/|\?) 0     0%      0
refresh_pattern .               0       20%     4320
cache_peer 172.25.12.102 parent 80 0 no-query originserver no-digest name=w1
cache_peer 172.25.12.102 parent 80 0 no-query originserver no-digest name=w2
```

```
cache_peer 172.25.12.103 parent 80 0 no-query originserver no-digest name=w3
cache_peer 172.25.12.103 parent 80 0 no-query originserver no-digest name=w4
cache_peer 172.25.12.106 sibling 3128 3130 name=cache0
cache_peer 172.25.12.107 sibling 3128 3130 name=cache1
cache_peer_domain w1    bbs1.uplooking.com
cache_peer_domain w2    bbs2.uplooking.com
cache_peer_domain w3    bbs1.uplooking.com
cache_peer_domain w4    bbs2.uplooking.com
~
```

3、在子squid配置

```
 vim /etc/squid/squid.conf
# Recommended minimum configuration:
#
acl manager proto cache_object
acl localhost src 127.0.0.1/32 ::1
acl to_localhost dst 127.0.0.0/8 0.0.0.0/32 ::1

# Example rule allowing access from your local networks.
# Adapt to list your (internal) IP networks from where browsing
# should be allowed
acl localnet src 10.0.0.0/8     # RFC1918 possible internal network
acl localnet src 172.16.0.0/12  # RFC1918 possible internal network
acl localnet src 192.168.0.0/16 # RFC1918 possible internal network
acl localnet src fc00::/7       # RFC 4193 local private network range
acl localnet src fe80::/10      # RFC 4291 link-local (directly plugged) machines

acl SSL_ports port 443
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443         # https
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl Safe_ports port 3130        #icp
acl CONNECT method CONNECT

#
# Recommended minimum Access Permission configuration:
#
# Only allow cachemgr access from localhost
http_access allow manager localhost
http_access deny manager

# Deny requests to certain unsafe ports
http_access deny !Safe_ports

# Deny CONNECT to other than secure SSL ports
http_access deny CONNECT !SSL_ports

# We strongly recommend the following be uncommented to protect innocent
# web applications running on the proxy server who think the only
# one who can access services on "localhost" is a local user
#http_access deny to_localhost

#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#

# Example rule allowing access from your local networks.
# Adapt localnet in the ACL section to list your (internal) IP networks
# from where browsing should be allowed
http_access allow localnet
http_access allow localhost
```

```
# And finally deny all other access to this proxy
http_access deny all
http_port 3128 vhost vport
icp_port 3130
# Squid normally listens to port 3128
# We recommend you to use at least the following line.
hierarchy_stoplist cgi-bin ?
cache_mem 100 MB
cache_dir ufs /var/spool/squid 100 16 256
# Uncomment and adjust the following to add a disk cache directory.
visible_hostname vm07.uplooking.com
# Leave coredumps in the first cache dir
coredump_dir /var/spool/squid

# Add any of your own refresh_pattern entries above these.
refresh_pattern ^ftp:           1440    20%     10080
refresh_pattern ^gopher:        1440    0%      1440
refresh_pattern -i (/cgi-bin/|\?) 0     0%      0
refresh_pattern .               0       20%     4320
cache_peer 172.25.12.106 parent 3128 0 no-query originserver no-digest name=w1
cache_peer 172.25.12.106 parent 3128 0 no-query originserver no-digest name=w2
cache_peer 172.25.12.107 sibling 3128 3130 name=cache1
cache_peer 172.25.12.106 sibling 3128 3130 name=cache0
cache_peer_domain w1    bbs1.uplooking.com
cache_peer_domain w2    bbs2.uplooking.com

#service squid start && chkconfig squid on
```

4、测试squid代理
测试机上先修改对应的/etc/hosts文件

```
# vim /etc/hosts
172.25.12.106     bbs1.uplooking.com      bbs2.uplooking.com
```

父squid的测试结果

```
# curl -I http://bbs1.uplooking.com:3128/template/default/image/guest_bg.jpg
HTTP/1.0 200 OK
Server: nginx/1.4.7
Date: Fri, 06 Feb 2015 08:04:35 GMT
Content-Type: image/jpeg
Content-Length: 37373
Last-Modified: Wed, 23 Mar 2011 06:33:46 GMT
ETag: "4d89944a-91fd"
Accept-Ranges: bytes
X-Cache: MISS from vm06.uplooking.com                           – – 第一次是miss
X-Cache-Lookup: MISS from vm06.uplooking.com:3128
Via: 1.0 vm06.uplooking.com (squid/3.1.10)
Connection: keep-alive


# curl -I http://bbs1.uplooking.com:3128/template/default/image/guest_bg.jpg
HTTP/1.0 200 OK
Server: nginx/1.4.7
Date: Fri, 06 Feb 2015 08:04:35 GMT
Content-Type: image/jpeg
Content-Length: 37373
Last-Modified: Wed, 23 Mar 2011 06:33:46 GMT
ETag: "4d89944a-91fd"
Accept-Ranges: bytes
Age: 50
X-Cache: HIT from vm06.uplooking.com                – – 第二次以后都是hit
X-Cache-Lookup: HIT from vm06.uplooking.com:3128
Via: 1.0 vm06.uplooking.com (squid/3.1.10)
Connection: keep-alive
```

```
# vim /etc/hosts
172.25.12.107     bbs1.uplooking.com      bbs2.uplooking.com
```

子squid的测试结果

```
# curl -I http://bbs1.uplooking.com:3128/template/default/image/guest_bg.jpg
HTTP/1.0 200 OK
Server: nginx/1.4.7
Date: Fri, 06 Feb 2015 08:04:35 GMT
Content-Type: image/jpeg
Content-Length: 37373
Last-Modified: Wed, 23 Mar 2011 06:33:46 GMT
ETag: "4d89944a-91fd"
Accept-Ranges: bytes
Age: 230
X-Cache: HIT from vm06.uplooking.com
X-Cache-Lookup: HIT from vm06.uplooking.com:3128
X-Cache: MISS from vm07.uplooking.com
X-Cache-Lookup: MISS from vm07.uplooking.com:3128
Via: 1.0 vm06.uplooking.com (squid/3.1.10), 1.0 vm07.uplooking.com (squid/3.1.10)
Connection: keep-alive


# curl -I http://bbs1.uplooking.com:3128/template/default/image/guest_bg.jpg
HTTP/1.0 200 OK
Server: nginx/1.4.7
Date: Fri, 06 Feb 2015 08:04:35 GMT
Content-Type: image/jpeg
Content-Length: 37373
Last-Modified: Wed, 23 Mar 2011 06:33:46 GMT
ETag: "4d89944a-91fd"
Accept-Ranges: bytes
X-Cache: HIT from vm06.uplooking.com
X-Cache-Lookup: HIT from vm06.uplooking.com:3128
Age: 265
X-Cache: HIT from vm07.uplooking.com
X-Cache-Lookup: HIT from vm07.uplooking.com:3128
Via: 1.0 vm06.uplooking.com (squid/3.1.10), 1.0 vm07.uplooking.com (squid/3.1.10)
Connection: keep-alive
```

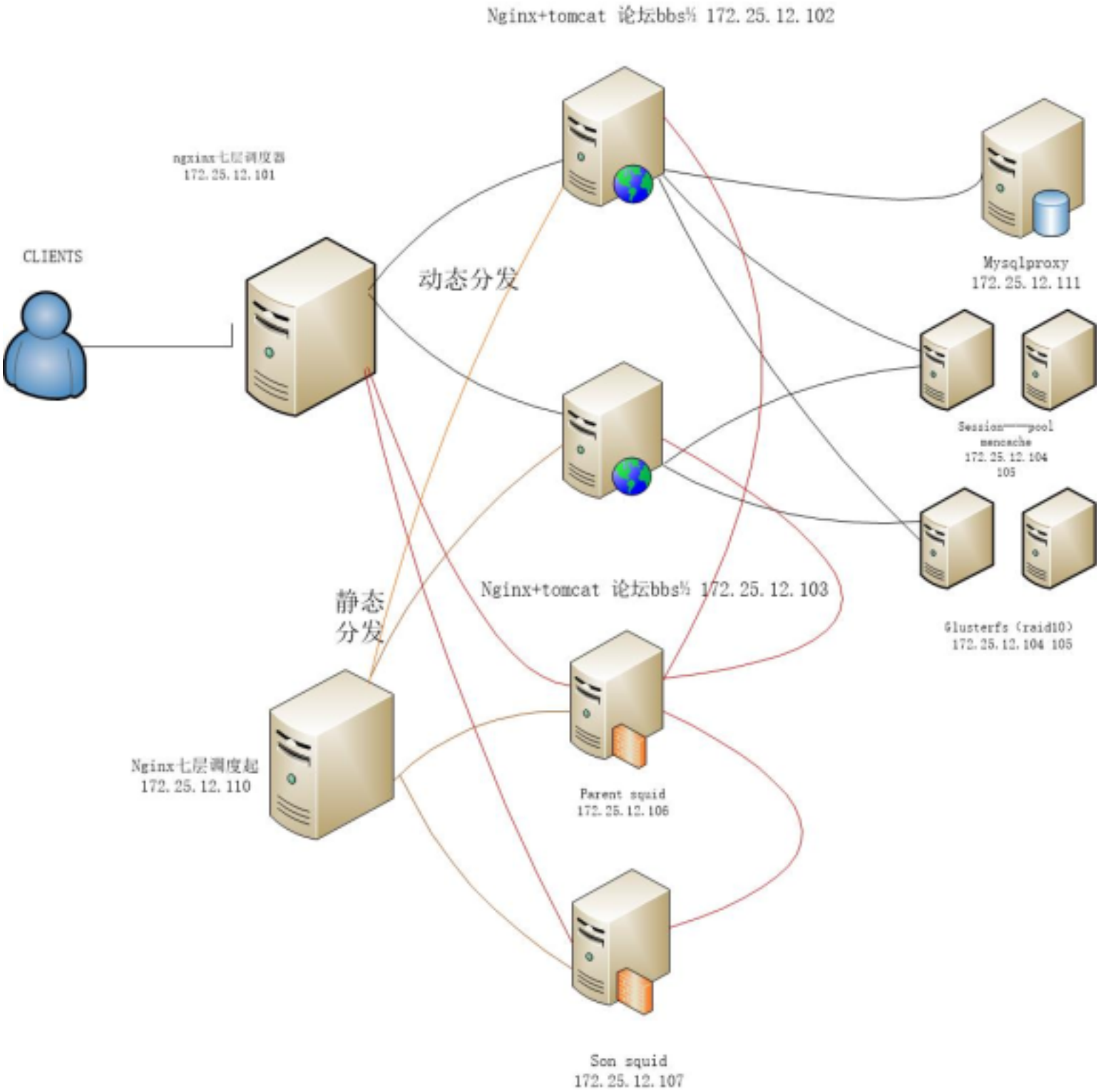
5、在nginx1调度器上增加squid的分发

```
# vim /etc/nginx/nginx.conf
user  nginx nginx;
worker_processes  1;
error_log  /var/log/nginx/error.log  info;
pid        /var/run/nginx.pid;


events {
    worker_connections  65535;
    use epoll;
}


http {
    include       mime.types;
    default_type  application/octet-stream;
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';
    access_log  /var/log/nginx/access.log  main;
    sendfile        on;
    tcp_nopush     on;
    keepalive_timeout  30;
    gzip  on;
        upstream squid_upstream {
                server 172.25.12.106:3128 weight=1;
                server 172.25.12.107:3128 weight=10;
                # hash $request_uri; (安装nginx时需要编译url_hash才能开启选项--add-module=url_hash)
                # hash_again 10;
        }
        upstream tomcat_upstream {
                server 172.25.12.102:8080 weight=10;
                server 172.25.12.103:8080 weight=10;
```

```
    }


  server {
    listen       80;
    server_name  localhost;
    charset utf8;

    location / {
        index  index.html index.htm index.jsp;
        proxy_pass http://squid_upstream;
        proxy_set_header Host $host;
        proxy_set_header X-Forward-For $remote_addr;
    }

    location ~ \.jsp$ {
        proxy_pass http://tomcat_upstream;
        proxy_set_header Host $host;
        proxy_set_header X-Forward-For $remote_addr;
    }
    location ~ ^/forum-[0-9]-[0-9]-[0-9]\.html$ {
        proxy_pass http://tomcat_upstream;
        proxy_set_header Host $host;
        proxy_set_header X-Forward-For $remote_addr;
    }
    location ~ ^/ok\.html$ {
        index index.html;
        root /var/nginx/html;
    }

  }
}

# service nginx reload


++++++++++++++++++++++++
测试：将测试机的/etc/hosts文件修改好
# vim /etc/hosts
172.25.12.101  bbs1.uplooking.com  bbs2.uplooking.com

直接访问 看到完整页面
关闭squid
则访问不了静态页面
```

五、利用nginx七层分发器实现基于内容的分发

拓扑图



lftp 172.25.254.250 → cd /notes/ula/item -→ mirror softwares
#cd /root/softwares
rpm -ivh nginx-1.4.7-10.x86_64.rpm
mkdir /var/log/nginx
#mkdir /var/run/nginx -p
ulimit -HSn 65535
vim /etc/security/limits.conf
root - nproc 65535
root - nofile 65535
cp /root/softwares/nginx.init.d /etc/rc.d/init.d/nginx
rsync -avzR /etc/nginx/nginx.conf 172.25.12.110:/
#service nginx restart && chkconfig nginx on

测试分发结果

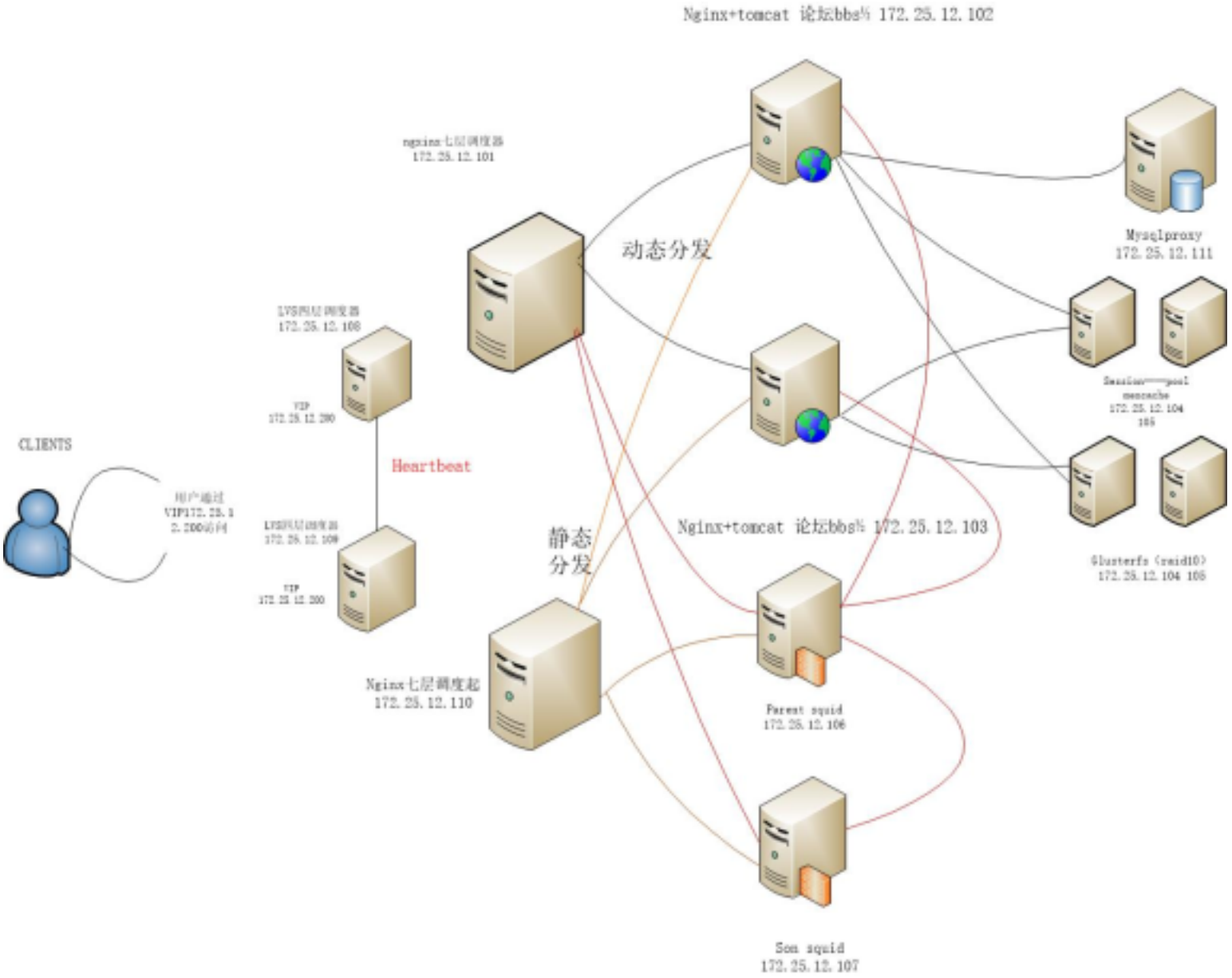
六、利LVS实现网站流量高效快速的分发(LVS DR)
方案1：
直接分发(lvs1配置)
yum -y install ipvsadm
ipvsadm -A -t 172.25.12.100:80 -s rr
ipvsadm -a -t 172.25.12.100:80 -r 172.25.12.101 -g
ipvsadm -a -t 172.25.12.100:80 -r 172.25.12.110 -g

ifconfig eth0:100 172.25.12.100/32 up

2、在后端nginx节点172.25.12.101和172.25.12.110做如下操作
vim /opt/lvs-dr.sh
#!/bin/sh
VIP=172.25.12.100
/sbin/ifconfig lo:0 $VIP broadcast $VIP netmask 255.255.255.255 up
/sbin/route add -host $VIP dev lo:0
echo "1" >/proc/sys/net/ipv4/conf/lo/arp_ignore
echo "2" >/proc/sys/net/ipv4/conf/lo/arp_announce
echo "1" >/proc/sys/net/ipv4/conf/all/arp_ignore
echo "2" >/proc/sys/net/ipv4/conf/all/arp_announce

/bin/bash /opt/lvs-dr.sh

++
方案2
heartbeat+LVS
拓扑图



1)下载软件
lftp 172.25.254.250 --> cd /notes/ula/cluster --> mirror heartbeat/

2)lvs1和lvs2安装软件
yum -y remove cluster-glue-libs
yum -y install openhpi-libs OpenIPMI-libs
rpm -ivh heartbeat-stonith-2.1.4-11.el6.x86_64.rpm heartbeat-pils-2.1.4-11.el6.x86_64.rpm
yum -y localinstall heartbeat-ldirectord-2.1.4-11.el6.x86_64.rpm
rpm -ivh heartbeat-2.1.4-11.el6.x86_64.rpm libnet-1.1.5-1.el6.x86_64.rpm

3)在lvs1配置heartbeat

cd /usr/share/doc/heartbeat-2.1.4/
cp ha.cf haresources authkeys /etc/ha.d/
vim /etc/ha.d/ha.cf
debugfile /var/log/ha-debug
keepalive 2
deadtime 30
warntime 10
initdead 60
udpport 694
bcast eth0
auto_failback off
node vm08.uplooking.com
node vm09.uplooking.com
ping 172.25.12.254

```
watchdog /dev/watchdog
respawn hacluster /usr/lib64/heartbeat/ipfail
apiauth ipfail gid=haclient uid=hacluster


cp /usr/share/doc/heartbeat-ldirectord-2.1.4/ldirectord.cf  /etc/ha.d/
# vim /etc/ha.d/ldirectord.cf
checktimeout=3
checkinterval=1
fallback=127.0.0.1:80
autoreload=yes
logfile="/var/log/ldirectord.log"
logfile="local0"
emailalert="root@localhost"
quiescent=yes

virtual=172.25.12.100:80
        real=172.25.12.101:80 gate
        real=172.25.12.110:80 gate
        fallback=127.0.0.1:80 gate
        service=http
        request="ok.html"
        receive="uplooking"
        scheduler=rr
        #persistent=600
        #netmask=255.255.255.255
        protocol=tcp
        checktype=negotiate
        checkport=80

# vim /etc/ha.d/haresources
vm08.example.com ldirectord::ldirectord.cf IPaddr2::172.25.12.100/32/eth0:100

# vim /etc/ha.d/authkeys
auth 2
#1 crc
2 sha1 c4627ddc77c3e6b4d3274742d0808dc8971b1d84
#3 md5 Hello!

#  chmod 600 /etc/ha.d/authkeys
#  rsync -avzR /etc/ha.d/ lvs2:/

+++++++以下两个lvs都要做+++++++++=
# /etc/init.d/heartbeat start
# chkconfig heartbeat on

后端调度器设置测试页面
# echo "uplooking" > /var/nginx/html/index.html
# vim /etc/nginx/nginx.conf
        location ~ ^/ok\.html$ {
                index index.html;
                root /var/nginx/html;
        }
# service nginx reload
# rsync -avzR /etc/nginx/nginx.conf /var/nginx/html/index.html vm10:/
# /etc/init.d/nginx reload

使用心跳ip访问后端网页顺利访问
```

七、利用智能DNS实现大并发流量切割

拓扑图(略)
智能DNS条件有限，只是理解了，并做了一个小测试，但是并没有加入到集群里。

```
 制作智能DNS（172.25.12.100／172.25.13.100）
# yum -y install bind bind-utils
# vim /etc/named.conf
options {
        listen-on  { any; };
        directory  "/var/named";
        dump-file "/var/named/data/cache_dump.db";
```

```
        statistics-file    "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";
        dnssec-enable no;
        dnssec-validation no;
        dnssec-lookaside no;
        bindkeys-file "/etc/named.iscdlv.key";

};

logging {
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};
view dxview {
        match-clients { 172.25.12.0/24; };
        include "/etc/named.rfc1912.zones";
        zone "." {
                type hint;
                file "named.ca";
        };
        zone "uplooking.com" IN {
                type master;
                file "/var/named/uplooking.com.zone.dx";
                allow-update { none; };
        };
};
view wtview {
        match-clients { 172.25.13.0/24; };
        include "/etc/named.rfc1912.zones";
        zone "." {
                type hint;
                file "named.ca";
        };
        zone "uplooking.com" IN {
                type master;
                file "/var/named/uplooking.com.zone.wt";
                allow-update { none; };
        };
};


# vim /var/named/uplooking.com.zone.dx
$TTL 1D
@       IN SOA  dns.uplooking.com. admin.uplooking.com. (
                                        0       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum

@               IN          NS          dns.uplooking.com.
dns             IN          A           172.25.12.254

bbs1            IN          A           172.25.12.100
bbs2            IN          A           172.25.12.100




# vim /var/named/uplooking.com.zone.wt
$TTL 1D
@       IN SOA  dns.uplooking.com. admin.uplooking.com. (
                                        0       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum
```

```
@       IN      NS      dns.uplooking.com.
dns     IN      A       172.25.13.254

bbs1            IN      A       172.25.13.100
bbs2            IN      A       172.25.13.100
```

```
# service named start
```

测试DNS是否准确可用

```
# host bbs1.uplooking.com 172.25.12.254
Using domain server:
Name: 172.25.12.254
Address: 172.25.12.254#53
Aliases:

bbs1.uplooking.com has address 172.25.12.100


# host bbs1.uplooking.com 172.25.13.254
Using domain server:
Name: 172.25.13.254
Address: 172.25.13.254#53
Aliases:

bbs1.uplooking.com has address 172.25.13.100
```

5、找两个客户端测同一网段试一下

```
# vim /etc/resolv.conf
nameserver 172.25.12.254
# firefox http://bbs1.uplooking.com/
# firefox http://bbs2.uplooking.com/


# vim /etc/resolv.conf
nameserver 172.25.13.254
# firefox http://bbs1.uplooking.com/
# firefox http://bbs2.uplooking.com/
```

增强DNS的安全性

```
 yum -y install bind-chroot
```

---硬连接

```
 [root@master ~]# ll -di /etc/named.conf
395642 -rw-r----- 1 root named 1426 9月  26 00:45 /etc/named.conf
[root@master ~]# ll -di /var/named/chroot/etc/named.conf
395642 -rw-r----- 1 root named 1426 9月  26 00:45 /var/named/chroot/etc/named.conf

[root@master ~]# ps aux |grep named
named    15398  0.0  0.1 321664 23188 ?        Ssl  00:53   0:00 /usr/sbin/named -u named -t /var/named/chroot
```