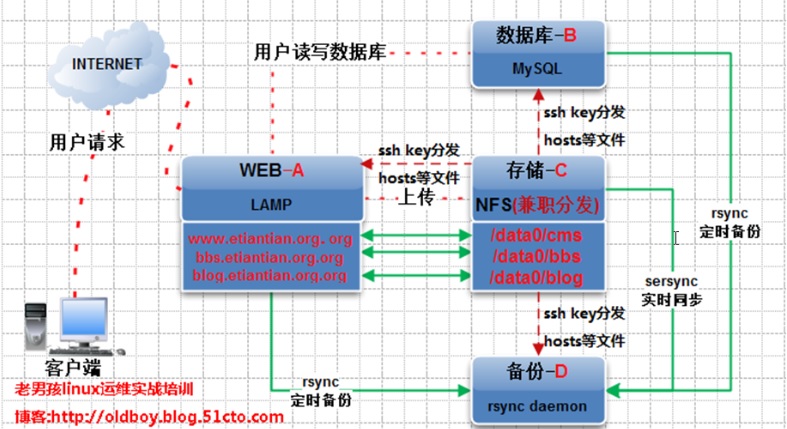
LAMP架构运维实战(二)

# 3.1 LAMP架构生产环境部署和配置

## 3.1.1 架构逻辑图



## 3.1.2 数据流说明

1)用户通过INTERNET访问到WEB服务器www.etiantian.org,blog.etiantian.org,bbs.etiantian.org域名

2)用户更新帖子,发布博客,将数据写入数据库(注意数据库和web服务器是分离的)

3)用户上传图片，将通过web服务器把数据上传到NFS存储上，而不保留在本地

4)所有的数据最终都备份到备份服务器上留存

5)NFS存储兼职分发数据库，会把需要分发的文件批量分发的其他网内节点服务器

# 3.2 生产环境搭建需求

+--------+----------------------+

|linux | centos6.5 64bit |

+--------+----------------------+

|Apache | 2.2.37 |

+--------+----------------------+

|MySQL | 5.5.37 |

+--------+----------------------+

|php | 5.3.27 |

+--------+----------------------+

# 3.3 ip地址的需求

+-------+-----------------+-----------------+

| |内网 |外网

+-------+-----------------+-----------------+

|LAMP | 10.0.0.100 | 192.168.100.100|

+-------+-----------------+-----------------+

|MYSQL | | 192.168.100.101|

+-------+-----------------+-----------------+

|NFS | | 192.168.100.102|

+-------+-----------------+-----------------+

|rsync | | 192.168.100.103|

+-------+-----------------+-----------------+

|说明：IP地址自行定义，最好分内外网卡，这样 |

|更接近实际环境，即办公室的10.0.0.0/24网段作|

|为外网IP，在添加一个10.0.1.0/24网段作为内网|

|网段，WEB服务器又双IP，其他服务器仅有内网IP|

|网段，所有，内部网段通过WEB服务器上网 |

+-------+-----------------+-----------------+

# 3.4 服务器需求

+-------+-----------------+-----------------+

|部署的服务|服务器编号 |服务器硬件 |

+------------+------------+-----------------+

|Mysql,apache| | |

|php,crontab |LAMP-server | |

+-------+-----------------+-----------------+

|MYSQL | MySQL-server | |

+-------+-----------------+-----------------+

|NFS,sersync | rsync-client| |

|sshkey | | |

+-------+-----------------+-----------------+

|rsync |rsync-server | |

+-------+-----------------+-----------------+

3.5 架构需求

1:配置WEB服务器A，搭建三个虚拟主机，虚拟主机分别是www.etiantian.org,bbs.etiantian.org,blog.etiantian.org

2:虚拟主机站点目录结构要求

tree /data0/

/data/

`-- www

|-- bbs

|-- blog

|-- www

提示：WEB服务器上不需要跑数据库服务，数据在专门的数据库服务器上。

3:搭建专用数据库服务器B，存储WEB服务器上的CMS,bbs,blog动态数据,如博文和帖子

4:搭建NFS共享存储服务器C,存储WEB服务器上的CMS,BBS,blog资源数据，如图片，附件，头像等

注意：NFS共享存储服务器只存放对于用户来说上传的资源数据。

5:需要搭建备份服务器D,用来备份WEB，数据库及NFS存储上的数据

6:NFS存储服务器上有写入数据时，要求即将NFS上的图片，附件，头像等资源同步到rsync服务的热备服务器(当NFS存储宕机，可以手动切换，继续替代NFS提供服务)

7:WEB服务器每天00点备份程序目录及访问日志并推送到备份服务器上(WEB服务器本地保留7天数据，备份服务器上保留3个月的数据副本)

8：MYSQL服务器每天00:30点备份数据库并推送到热备服务器(数据库本地保留7天)

，备份服务器上保留3个月数据副本)

9:NFS存储C兼职分发服务器(也可以单独搞，会把需要分发的文件批量分发到其他网

内节点服务器，如A,B,D.

10:所有服务器配置定时时间同步，这里会遭遇到问题，所有内网服务器必须通过

WEB SERVER上网，当然你可以配置NTP SERVER

## 3.6 实战配置过程

## 3.6.1 linux系统优化配置

1)启动服务器

2)配置CRT连接服务器

3)查看硬件信息

4)查看系统信息

5)初始化系统

1：调整防火墙和SELINUX

chkconfig iptables off

/etc/init.d/iptables stop

sed -i 's@SELINUX=enforcing@SELINUX=disable@g'/etc/selinux/conf

setenforce 0

2: 添加一个系统普通账户

useradd -r oldboy

useradd oldboy

id oldboy

echo "oldboy123" | passwd --stdin oldboy

history -c

3:配置YUM源

cd /etc/yum.repos.d/

bin/mv CentOS-Base.repo CentOS-Base.repo.ori

wget http://mirrors.sohu.com/help/CentOS-Base-Sohu.repo

rpm -import /etc/pki/rpm-gpg/RPM-GPG-KEY\*

yum upgrade

4:安装必要的软件

yum install lrzsz sysstat-y

yum groupinstall "x software development"

yum grouplist

## 3.6.2 安装后的基本调优及安全设置

### 3.6.2.1 清理开机自启动的服务

1:首先通过下面命令关闭所有的开机自启动的服务器

for oldboy in `chkconfig --list|grep 3:on|awk '{print $1}'`;do

chkconfig --level 3 $oldboy off;done

或者使用:for service in `chkconfig --list|grep 3:on|sed 's/0:off.\*$//g;'`;do chkconfig --level 3 $service off;done

执行如下命令开启需要开机自动的服务

for oldboy in crond network syslog sshd;do chkconfig --level 3 $oldboy on;done

chkconfig --list |grep 3:on

2:更改SSH的配置

|  |
| --- |
| cp /etc/ssh/sshd\_config /etc/ssh/ssh\_config.ori  sed -i 's%#Port 22%Port 52113%' /etc/ssh/sshd\_config  sed -i 's%#PermitRootLogin yes%PermitRootLogin no%' /etc/ssh/sshd\_config  sed -i 's%#PermitEmptyPassword no%PermitEmptyPassword no%' /etc/ssh/sshd\_config  sed -i 's%#UseDNS yes%UseDNS no%' /etc/ssh/sshd\_config  egrep "UseDNS|52113|RootLogin|EmptyPass" /etc/ssh/sshd\_config  /etc/init.d/sshd reload |

3:将普通用户加入到sudo管理

执行如下命令，即可打开visudo的配置文件

visudo #相当于直接编辑/etc/sudoer,但用命令方式更安全，推荐

在文件的中间如下内容的下面需要root权限的用户名，格式如下

#User privilege sepcification

root ALL=(ALL)ALL

oldboy ALL=(ALL)ALL #这个配置表示oldboy可拥有完全的系统管理员权限，

有关更细致的权限规划，请参看笔者的有关sudo的详细文章。

将以上信息更改后，保存退出

此时等用oldboy登陆系统时，可以通过执行类似sudo ls -l /root 来以root用户的权限操作系统了

4：修改中文显示

通过快捷的命令方式在/etc/sysconfig/i18n中添加如下内容

cp /etc/sysconfig/i18n /etc/sysconfig/i18n.ori

echo 'LANG="zh\_CN.GB18030"' >/etc/sysconfig/i18n

相当于用vim /etc/sysconfig/i18n添加LANG="zh\_CN.GB18030"内容

source /etc/sysconfig/i18n #使上文修改生效

提示：这个中文显示配置要跟你自己的SSH客户端配置一致，默认情况下就是我演示的配置

5:服务器时间同步

echo '\*/5 \* \* \* \* /usr/sbin/ntpdate time.windows.com>/dev/null 2>&1'>>/var/spool/cron/root

这个命令其实就是写一个定时任务，相当于执行crontab -e 然后加入内容，\*/5 \* \* \* \* /usr/sbin/ntpdate time.windows.com>/dev/null 2>&1

保存退出，有关crontab定时任务后文有详细解释，大家不必在这里纠缠

补充一个定时清理优化任务，find /var/spool/clientmqueue/ -type -mtime +30|xargs rm -rf

提示：在机器数量少时，以上定时任务同步时间就可以了，如果机器数量大时，可以在网内载部署一个时间同步服务器ntp server,此处不再深入讲解，大家可自行查阅相关文章

6：加大文件描述符

echo '\* - nofile 65535'>>/etc/security/limits.conf

tail -l /etc/security/limits.conf

注明：配置完成后，重新登陆即可查看

ulimit -n 65535

提示：也有人把ulimit -SHn 65535 命令加入到/etc/rc.local 然后每次重启生效

7：调整内核参数文件/etc/sysctl.conf

以下是我的生产环境的某非服务器的配置

|  |
| --- |
| cp /etc/sysctl.conf /etc/sysctl.conf.$(date +%F)  cat >> /etc/sysctl.conf << EOF  net.ipv4.tcp\_timestamps = 0  net.ipv4.tcp\_synack\_retries = 2  net.ipv4.tcp\_syn\_retries = 2  net.ipv4.tcp\_mem= 94500000 915000000 927000000  net.ipv4.tcp\_max\_orphans = 3276800  net.ipv4.core.wmem\_default = 8388608  net.ipv4.core\_rmem\_default = 8388608  net.ipv4.core\_rmem\_max = 16777216  net.ipv4.core\_wmem\_max = 16777216  net.ipv4.tcp\_rmem = 4096 87380 16777216  net.ipv4.tcp\_wmem = 4096 65536 16777216  net.core.netdev\_max\_backlog = 32768  net.core.somaxconn = 32768  net.ipv4.tcp\_syncookies = 1  net.ipv4.tcp\_tw\_reuse = 1  net.ipv4.tcp\_tw\_recycle = 1  net.ipv4.tcp\_fin\_timeout = 1  net.ipv4.tcp\_keepalive\_time = 1200  net.ipv4.tcp\_max\_syn\_backlog = 65536  net.ipv4.ip\_local\_port\_range = 1024 65535  EOF  sysctl -p #让配置生效 |

## 3.7 系统安装后的高级调优

包括内核深入优化细节及各种raid制作，此处省略

# 4.1 配置业务服务器lamp,rsync

+--------+---------+--------------------------+

|服务器名| IP | 服务器描述 |

+--------+---------+--------------------------+

|LAMP-server1|192.168.100.192|apache+php+mysql|

| | | rsync client |

+------------+----------+---------------------+

|mysql-server|192.168.100.193| mysql数据库, |

| | | rsync client |

+------------+----------+---------------------+

## 4.1.1 单实例MYSQL数据库的安装

### 4.1.1.1 建立MYSQL账号

手下以root身份登陆到linux系统，然后执行如下命令创建mysql组及用户

sudo su -

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

#useradd命令参数简要说明

#-s /sbin/nologin 表示禁止该用户登录，加强安全

#-g mysql指定属于mysql组

#-M 表示不创建用户家目录

#以上命令也可用useradd -s /sbin/nologin -M mysql替代上面的配置

检查创建的用户

tail -1 /etc/passwd

建立mysql软件目录

mkdir -p /home/oldboy/tools

cd /home/oldboy/tools

## 4.1.1.2 编译安装mysql

### 4.1.1.2.1 获得mysql软件

通过rz命令从本地上传mysql软件包如:mysql-5.1.62.tar.gz，当然你也可以到网上去下载。

本例通过rz命令来演示，如提示无rz命令，可先执行yum install lrzsz -y 来安装你也可以通过wget http://down1.chinaunix.net/distfiles/mysql-5.1.62.tar.gz方式下载

4.1.1.2.2 安装MYSQL数据库

1)解压

tar xf mysql-5.1.62.tar.gz

cd mysql-5.1.62

./configure \

--prefix=/usr/local/mysql \

--with-unix-socket-path=/usr/local/mysql/tmp/mysql.sock \

--localstatedir=/usr/local/mysql/data \

--enable-assembler \

--with-thread-safe-client \

--enable-mysqld-user=mysql \

--with-big-tables \

--without-debug \

--with-pthread \

--enable-assembler \

--with-extra-charset=complex \

--with-readline \

--with-ssl \

--with-embedded-server \

--enable-local-infile \

--with-plugins=partition,innobase \

--with-plugin-PLUGIN \

--with-mysqld-ldflags=-all-static \

--with-client-ldflags=-all-static

#说明

#可以通过执行./configure -help命令来详细查看以上个参数的用途

#以上配置中的 "\"斜线表示换行

#configure编译常用参数的说明，请柬附录一

3)静态编译生成mysqld的执行文件

make

#提示：

#也可以执行下命令，make -j 4 加快mysql的编译 #-j表示希望使用cpu核数

#如果使用不当可能反而安装速度更慢

#注意编译的结果是否报错(带error字样)，有些朋友也不管有没有错，就直接make install

结果导致无法完成安装

4)安装mysql

#提示

#注意编译的结果是否报错(带error字样)，有些朋友也不管有没有错，就直接make install

结果导致无法完成安装

#获取MYSQL配置文件

ls -l support-files/\*.cnf

/bin/cp support-files/my-small.cnf /etc/my.cnf

4.1.1.2.3 创建MYSQL数据库文件

mkdir -p /usr/local/mysql/data

chown -R mysql.mysql /usr/local/mysql/data

/usr/local/mysql/bin/mysql\_install\_db --user=mysql

#设置MYSQL的系统环境变量

echo 'export PATH=$PATH:/usr/local/mysql/bin' >>/etc/profile

source /etc/profile

#配置/etc/init.d/mysqld start方式启动数据库

cp support-files/mysql.server /etc/init.d/mysqld

chmod 700 /etc/init.d/mysqld

/etc/init.d/mysqld start

mysql

#数据库优化

mysql> drop user ""@localhost;

mysql> drop user ""@mysql;

mysql> drop user "root"@mysql;

# 4.2 安装apache服务

1)查询当前系统中已安装apache软件包

通过rpm -qa httpd\*命令，我们可以查询当前系统中已安装的apache软件包，具体查询

操作过程如下

rpm -aq httpd\* #如果你的机器没有查到，就表示没有安装，则不用卸载了

#rpm命令-qa参数说明

#-q等同于--query，查询的意思

#-a等同于-all，表示范围是系统所有软件包

#http\*中,http为apache软件包名的关键字，\*为模糊匹配，表示包含http字样的软件包

2)卸载查询出来的apache软件包

如果查询出来已经安装了apache的rpm软件包后，就需要通过rpm -e --nodeps后面分别接

查询到的apache软件包完整名称，即可卸载当前系统中通过rpm安装方式的apache软件包

具体卸载操作过程如下

rpm -e --nodeps httpd-2.2.3-43.el5.centos.3

rpm -e --nodeps httpd-devel-2.2.3-43.el5.centos.3

rpm -e --nodeps httpd-manual-2.2.3-43.el5.centos.3

#rpm命令-e --nodeps参数说明

#-e等同于--erase表示移除的意思

#--nodeps的意思是不做软件间的依赖关系

3)安装MYSQL软件包

cd /home/oldboy/tools

wget http://ldbs.renren.com/apache-mirror//httpd/httpd-2.2.22.tar.gz

#通过wget命令下载apache软件包

tar xf httpd-2.2.22.tar.gz

cd httpd-2.2.22

./configure \

--prefix=/application/apache2.2.22 \

--enable-deflate \

--enable-expire \

--enable-headers \

--enable-modules=most \

--enable-so \

--with-mpm=worker \

--enable-rewrite

#./configure后的参数说明

1)--prefix=/application/apache2.2.22 表示指定apache安装路径为/application/apache2-2.22，如果不指定安装路径，则默认路径为/usr/local/apache2

2)--enable-deflate 提供对内容的压缩传输编码支持，一般html,js,css等内容的站点，使用此参数功能会大大提高传输速度，提升访问者访问体验，在生产环境中，这是apache调优的一个重要选项之一

3)--enable-expire激活运行通过配置文件控制HTTP的"Expires:"和"Cache-control:"头内容，即对网站图片，js,css等内容，提供在客户端浏览器缓存的设置，这是apache调优的一个重要选项

4)--enable-headers提供运行对http请求头的控制

5)--with-mpm=worker 选择apache mpm的模式为worker模式，因worker模式原理是更多的使用线程来处理请求，所以可以处理更多的并发请求，而系统资源的开销小于进程的MPM prefork，如果不指定此参数，默认的模式是prefork进程模式，这是apache调优的一个重要选项之一

6)--enable-rewrite 提供基于URL规则的重写功能，即根据已知URL地址，转换其他想要的URL地址，如前文讲解的伪静态就是这个模块实现的，这是apache在生产环境中必用的一个重要选项之一

make && make install

ln -s /application/apache2.2.22 /applicaion/apache

cd /application/apache/conf/

cp httpd.conf httpd.conf.ori

vim httpd.conf

修改以下选项参数

修改#ServerName www.example.com:80为ServerName 127.0.0.1:80

修改#Include conf/extra/httpd-vhosts.conf为Include conf/extra/httpd-vhosts.conf

修改完成后，保存退出

#安装cronolog日志轮询工具

cd /home/oldboy/tools

wget http://cronolog.org/download/cronolog-1.6.2.tar.gz

tar xf cronolog-1.6.2.tar.gz

cd cronolog-1.6.2

./configure

make

make install

mkdir /app/logs -p

ll /app -ld

#配置apache虚拟主机

vim extra/httpd-vhosts.conf

修改配置文件的内容为：

NameVirtualHost \*:80

<VirtualHost \*:80>

ServerAdmin 49000448@qq.com

DocumentRoot "/data/www"

ServerName www.etiantian.org

ServerAlias etiantian.org

ErrorLog "logs/www-error\_log"

CustomLog "|/usr/local/sbin/cronolog /app/logs/access\_www\_%Y%m%d.log" combined

</VirtualHost>

<VirtualHost \*:80>

ServerAdmin 49000448@qq.com

DocumentRoot "/data/bbs"

ServerName bbs.etiantian.org

ErrorLog "logs/bbs-error\_log"

CustomLog "|/usr/local/sbin/cronolog /app/logs/access\_bbs\_%Y%m%d.log" combined

</VirtualHost>

<VirtualHost \*:80>

ServerAdmin 49000448@qq.com

DocumentRoot "/data/blog"

ServerName blog.etiantian.org

ErrorLog "logs/blog-error\_log"

CustomLog "|/usr/local/sbin/cronolog /app/logs/access\_blog\_%Y%m%d.log" combined

</VirtualHost>

vim /application/apache/conf/httpd.conf

修改如下内容

<Directory "/data/">

Options FollowSymLinks

AllowOverride None

Order allow,deny

Allow from all

</Directory>

../sbin/apachectl graceful

#常见网站站点目录

mkdir /data/{www,bbs,blog} -p

for n in `ls /data`;do echo "$n" >/data/$n/index.html;done

../../sbin/apachectl start

4.3 安装PHP

1)检查PHP安装的依赖的软件包

rpm -aq zlib libxml libjpeg freetype libpng gd curl libiconv zlib-devel libxml2-devel libjpeg-devel freetype-devel libpng-devel gd-devel curl-devel

cd /home/oldboy/tools

tar xf libiconv-1.13.1.tar.gz

cd libiconv-1.13.1:

./configure --prefix=/usr/local/libiconv

make && make install

#技巧:\*可以复制多行命令一次输入执行

相关地址:http://www.gnu.org/software/libiconv/

2)下载并安装PHP

tar xf php-5.3.15.tar.bz2

cd php-5.3.15

./configure \

--prefix=/application/php5.3.27 \

--with-apxs2=/application/apache/bin/apxs \ #给apache添加模块的

--with-mysql=/application/mysql \

--with-xmlrpc \

--with-openssl \

--with-zlib \

--with-freetype-dir \

--with-gd \

--with-jpeg-dir \

--with-png-dir \

--with-iconv=/usr/local/libiconv \

--enable-short-tags \

--enable-sockets \

--enable-zend-multibyte \

--enable-soap \

--enable-mbstring \

--enable-static \

--enable-gd-native-ttf \

--with-curl \

--with-xsl \

--enable-ftp \

--with-libxml-dir

make && make install

ln -s /application/php5.3.15 /application/php

# 5.1 配置rsync备份服务

+--------+---------+--------------------------+

|服务器名| IP | 服务器描述 |

+--------+---------+--------------------------+

|rsync-server|10.0.0.190|rsync server |

+------------+----------+---------------------+

|rsync-client|10.0.0.191|rsync client |

+------------+----------+---------------------+

1)创建rsync的配置文件/etc/rsyncd.conf,并添加如下参数

uid = root

gid = root

use chroot = no

max connections = 2000

timeout = 600

pid file = /var/run/rsyncd.pid

lock file = /var/run/rsync.lock

log file = /var/log/rsyncd.log

ignore errors

read only = false

list = false

hosts allow = 10.0.0.0/24

hosts deny = 0.0.0.0/32

auth users = rsync\_backup

secrets file = /etc/rsync.password

###################################

[www]

comment = www by wpaccp 17:59 2017-03-06

path = /data/www/

###################################

[bbs]

comment = bbs by wpaccp 17:59 2017-03-06

path = /data/bbs/

#####################################

[blog]

comment = blog by wpaccp 17:59 2017-03-06

path = /data/blog/

#创建用户权限文件

echo "rsync\_backup:wpaccp" >/etc/rsync.password

chmod 600 /etc/rsync.password

cat /etc/rsync.password

ll /etc/rsync.password

#创建rsync server端的文件目录

mkdir -p data/{www,bbs,blog} -p

#启动rsync服务

#以守护进程方式启动rsync服务器

#设置开机自启动

#停止命令

rsync --daemon

echo "/usr/bin/rsync --daemon" >>/etc/rc.local

ps -ef | grep rsync

#rsync客户端的配置

echo "wpaccp" >/etc/rsync.password

chmod 600 /etc/rsync.password

cat /etc/rsync.password

ll /etc/rsync.password

#在客户端测试rsync的配置效果

rsync -avzP /etc/passwd rsync\_backup@10.0.0.190::www/ --password-file=/etc/rsync.password

# 5.2 rsync+sersync实时同步的配置

+--------+---------+--------------------------+

|服务器名| IP | 服务器描述 |

+--------+---------+--------------------------+

|sersync-client|10.0.0.190|sersync客户端 |

+------------+----------+---------------------+

|sersync-server|10.0.0.191|sersync服务器 |

+------------+----------+---------------------+

1)在sersync-server上解压sersync2.5.4\_64bit\_binary\_stable\_final.tar.gz

cd /home/oldboy/tools

mkdir /application

tar xf sersync2.5.4\_64bit\_binary\_stable\_final.tar.gz -C /application/

cd /application/

mv GNU-Linux-x86 sersync

cp sersync/confxml.xml sersync/confxml.xml.$(date +%F)

cp sersync/confxml.xml sersync/www\_confxml.xml

cp sersync/www\_confxml.xml sersync/bbs\_confxml.xml

cp sersync/www\_confxml.xml sersync/blog\_confxml.xml

2)在sersync-server上对www\_confxml.xml配置文件进行更改

a) 修改24--28行，原来的配置为：

<localpathwatch="/opt/tongbu"> # 定义本地要同步的目录

<remote ip="127.0.0.1"name="tongbu1"/>

<!--<remoteip="192.168.8.39" name="tongbu"/>--> # 同步到哪台机器上 tongbu模块rsync端模块名字

<!--<remoteip="192.168.8.40" name="tongbu"/>--> # 同步到哪台机器上 tongbu模块

</localpath>

修改为以下内容配置：

<localpath watch="/data/bbs">

<remote ip="10.0.0.190" name="bbs"/>

</localpath>

<!--################################################## -->

b)修改配置，原来的配置如下：

<rsync>

<commonParamsparams="-artuz"/>

<auth start="false"users="root" passwordfile="/etc/rsync.pas"/>

<userDefinedPortstart="false" port="874"/><!-- port=874 -->

<timeoutstart="false" time="100"/><!-- timeout=100 -->

<sshstart="false"/>

</rsync>

修改为以下内容配置：

<rsync>

<commonParams params="-artuz"/>

<auth start="true" users="rsync\_backup" passwordfile="/etc/rsync.password"/>

<userDefinedPort start="false" port="874"/><!-- port=874 -->

<timeout start="true" time="100"/><!-- timeout=100 -->

<ssh start="false"/>

</rsync>

c)修改37行,原来的配置

<failLogpath="/tmp/rsync\_fail\_log.sh"timeToExecute="60"/><!--default every 60mins execute once-->

修改为以下内容配置：

<failLog path="/application/sersync/rsync\_fail\_log.sh" timeToExecute="60"/><!--default every 60mins execute once-->

#在sersync-server上对bbs\_confxml.xml配置文件进行更改

a) 修改24--28行，原来的配置为：

<localpathwatch="/opt/tongbu"> # 定义本地要同步的目录

<remote ip="127.0.0.1"name="tongbu1"/>

<!--<remoteip="192.168.8.39" name="tongbu"/>--> # 同步到哪台机器上 tongbu模块rsync端模块名字

<!--<remoteip="192.168.8.40" name="tongbu"/>--> # 同步到哪台机器上 tongbu模块

</localpath>

修改为以下内容配置：

<localpath watch="/data/www">

<remote ip="10.0.0.190" name="www"/>

</localpath>

<!--################################################## -->

b)修改配置，原来的配置如下：

<rsync>

<commonParamsparams="-artuz"/>

<auth start="false"users="root" passwordfile="/etc/rsync.pas"/>

<userDefinedPortstart="false" port="874"/><!-- port=874 -->

<timeoutstart="false" time="100"/><!-- timeout=100 -->

<sshstart="false"/>

</rsync>

修改为以下内容配置：

<rsync>

<commonParams params="-artuz"/>

<auth start="true" users="rsync\_backup" passwordfile="/etc/rsync.password"/>

<userDefinedPort start="false" port="874"/><!-- port=874 -->

<timeout start="true" time="100"/><!-- timeout=100 -->

<ssh start="false"/>

</rsync>

c)修改37行,原来的配置

<failLogpath="/tmp/rsync\_fail\_log.sh"timeToExecute="60"/><!--default every 60mins execute once-->

修改为以下内容配置：

<failLog path="/application/sersync/rsync\_fail\_log.sh" timeToExecute="60"/><!--default every 60mins execute once-->

#在sersync-server上对blog\_confxml.xml配置文件进行更改

a) 修改24--28行，原来的配置为：

<localpathwatch="/opt/tongbu"> # 定义本地要同步的目录

<remote ip="127.0.0.1"name="tongbu1"/>

<!--<remoteip="192.168.8.39" name="tongbu"/>--> # 同步到哪台机器上 tongbu模块rsync端模块名字

<!--<remoteip="192.168.8.40" name="tongbu"/>--> # 同步到哪台机器上 tongbu模块

</localpath>

修改为以下内容配置：

<localpath watch="/data/www">

<remote ip="10.0.0.190" name="www"/>

</localpath>

<!--################################################## -->

b)修改配置，原来的配置如下：

<rsync>

<commonParamsparams="-artuz"/>

<auth start="false"users="root" passwordfile="/etc/rsync.pas"/>

<userDefinedPortstart="false" port="874"/><!-- port=874 -->

<timeoutstart="false" time="100"/><!-- timeout=100 -->

<sshstart="false"/>

</rsync>

修改为以下内容配置：

<rsync>

<commonParams params="-artuz"/>

<auth start="true" users="rsync\_backup" passwordfile="/etc/rsync.password"/>

<userDefinedPort start="false" port="874"/><!-- port=874 -->

<timeout start="true" time="100"/><!-- timeout=100 -->

<ssh start="false"/>

</rsync>

c)修改37行,原来的配置

<failLogpath="/tmp/rsync\_fail\_log.sh"timeToExecute="60"/><!--default every 60mins execute once-->

修改为以下内容配置：

<failLog path="/application/sersync/rsync\_fail\_log.sh" timeToExecute="60"/><!--default every 60mins execute once-->

d)开启sersync守护进程同步数据

/application/sersync/sersync2 -d -r -o /application/sersync/www\_confxml.xml

/application/sersync/sersync2 -d -r -o /application/sersync/bbs\_confxml.xml

/application/sersync/sersync2 -d -r -o /application/sersync/blog\_confxml.xml

echo "/application/sersync/sersync2 -d -r -o /application/sersync/www\_confxml.xml >/dev/null 2>&1" >>/etc/rc.local

echo "/application/sersync/sersync2 -d -r -o /application/sersync/bbs\_confxml.xml >/dev/null 2>&1" >>/etc/rc.local

echo "/application/sersync/sersync2 -d -r -o /application/sersync/blog\_confxml.xml >/dev/null 2>&1" >>/etc/rc.local

#配置sersync环境变量

echo "PATH=$PATH:/application/sersync/" >>/etc/profile

source /etc/profile

sersync2

# 5.3 NFS服务器的安装和配置

+--------+---------+--------------------------+

|服务器名| IP | 服务器描述 |

+--------+---------+--------------------------+

|NFS-server|10.0.0.191|NFS服务器,sersync服务器|

| | |分发服务器

+------------+----------+---------------------+

|NFS-client|10.0.0.192|NFS客户端，LAMP服务器 |

+------------+----------+---------------------+

1)检测NFS是否安装

rpm -aq nfs-utils rpcbind

2)如果检测该软件包，没有安装，就使用YUM源安装NFS

yum install -y nfs rpcbind

3)启动NFS服务,并配置NFS开机自启动

/etc/init.d/nfs start

echo "/etc/init.d/rpcbind start >/dev/null 2>&1" >>/etc/rc.local

echo "/etc/init.d/nfs start >/dev/null 2>&1" >>/etc/rc.local

4)配置NFS服务端

mkdir /backup/{www,bbs,blog} -p

useradd nfs

id nfs

echo "/backup 10.0.0.0/24(rw,sync,all\_squash,anonuid=500,anongid=500)" > /etc/exports

chown -R nfs /backup

/etc/init.d/rpcbind reload

/etc/init.d/nfs reload

5)NFS客户端挂载共享目录

rpm -aq nfs-utils rpcbind

yum install nfs-utils rpcbind -y

/etc/init.d/rpcbind start

echo "/etc/init.d/rpcbind start" >>/etc/rc.local

showmount -e 10.0.0.191

echo "mount -t nfs 10.0.0.191:/data/www /data/www" >>/etc/rc.local

echo "mount -t nfs 10.0.0.191:/data/bbs /data/bbs" >>/etc/rc.local

echo "mount -t nfs 10.0.0.191:/data/blog /data/blog" >>/etc/rc.local

6) CMS,BBS,BLOG产品的部署

1：创建数据库和授权权限的用户

create database www;

create database bbs;

create database blog;

grant all on www.\* to www@'10.0.0.%' identified by 'oldboy123';

grant all on bbs.\* to bbs@'10.0.0.%' identified by 'oldboy123';

grant all on blog.\* to blog@'10.0.0.%' identified by 'oldboy123';

2：CMS,BBS,BLOG产品的部署

groupadd -g 500 www

useradd -g 500 -u 500 -s /sbin/nologin -M www

mkdir /home/oldboy/tools

cd /home/oldboy/tools

wget http://www.dedecms.com/dl/dl.php?action=dl&type=1&lang=gbk

tar xf wordpress-3.4.1-zh\_CH.tar.gz

ll wordporess

mv wordpress/\* /data/blog

tar xf DedeCMS-V5.7-GBK-SP1-Full.tar.gz

cd DedeCMS-V5.7-GBK-SP1-Full

mv DedeCMS-V5.7-GBK-SP1-Full/uploads/\* /data/www/

mkdir bbs

mv Discuz\_X2.5\_SC\_GBK.zip bbs/

unzip Discuz\_X2.5\_SC\_GBK.zip

mv upload/\* /data/bbs/

================================================

cd ../

cd /data/www/

chown -R www dede/ a/ install/ special/ uploads/

chown ett -R ../www

================================================

cd ../bbs/

chown -R www data/ uc\_\* config/

================================================

cd ../

chown -R www blog

================================================

# 5.4 分发服务器的配置

一:手工方式

su - oldboy

ssh-keygen -t dsa

ssh-copy-id -i .ssh/id\_rsa.pub "-p 52113 oldboy@10.0.0.191"

ssh-copy-id -i .ssh/id\_rsa.pub "-p 52113 oldboy@10.0.0.192"

ssh-copy-id -i .ssh/id\_rsa.pub "-p 52113 oldboy@10.0.0.193"

二:用expect实现自动分发

1)安装Expect软件

首先，配置好YUM安装源，并且确保机器可以上网，然后执行yum install expect y即可按

装Expect软件

rpm -aq | grep expect

yum install -y expect

2)使用脚本分发sshkey文件

su - oldboy

mkdir scripts

mv oldboy-6.exp.good fenfa\_sshkey.exp

mv fenfa\_sshkey.exp scripts/

mv fenfa\_sshkey.sh scripts/

=================================================================

vim fenfa\_sshkey.exp

#!/usr/bin/expect

#

if { $argc !=2 } {

send\_user "usage: expect scp-expect.exp file host dir\n"

exit

}

#define var

set file [lindex $argv 0]

set host [lindex $argv 1]

set password "wpaccp"

spawn ssh-copy-id -i $file root@$host

#spawn ssh-copy-id -i $file "-p 52113 oldboy@$host"

expect {

"yes/no" {send "yes\r";exp\_continue}

"\*password" {send "$password\r"}

}

expect eof

exit -onexit {

send\_user "oldboy say good by to you!\n"

}

测试：

expect fenfa\_sshkey.exp /home/oldboy/.ssh/id\_dsa.pub 10.0.0.191

===============================================================

vim fenfa\_sshkey.sh

#!/bin/sh

. /etc/init.d/functions

[ ! -d ~/.ssh ] && ssh-keygen -t dsa

for ip in `cat iplist`; do

expect fenfa\_sshkey.exp ~/.ssh/id\_dsa.pub $ip >/dev/null 2>&1

if [ $? -eq 0 ];then

action "$ip" /bin/true

else

action "$ip" /bin/false

fi

done

================================================================

vim iplist

10.0.0.190

10.0.0.192

10.0.0.193

或者

================================================================

#!/bin/sh

. /etc/init.d/functions

for ip in 190 192 193; do

expect fenfa\_sshkey.exp .ssh/id\_rsa.pub 192.168.100.$ip >/dev/null 2>&1

if [ $? -eq 0 ]; then

action 192.168.100.$ip /bin/true

else

action 192.168.100.$ip /bin/false

fi

done

================================================================

chmod 755 \*

sh -x fenfa\_sshkey.sh

# 5.5 自动分发/etc/hosts配置文件的脚本

vim fenfa\_hosts.sh

#!/bin/sh

. /etc/init.d/functions

for ip in 101 102; do

rsync -avzP $1 root@$ip:$2 >/dev/null 2>&1

if [ $? -eq 0 ]; then

action 192.168.100.$ip /bin/true

else

action 192.168.100.$ip /bin/false

fi

done

测试脚本命令：

sh fenfa\_hosts.sh /etc/hosts /etc/

案例二

=================================================================

vim fenfa\_file.exp

#!/usr/bin/expect

#

if { $argc !=3 } {

send\_user "usage: expect scp-expect.exp file host dir\n"

exit

}

#define var

set file [lindex $argv 0]

set host [lindex $argv 1]

set dir [lindex $argv 2]

set password "wpaccp"

#spawn ssh-copy-id -i $file root@$host

#spawn ssh-copy-id -i $file "-p 52113 oldboy@$host"

spawn rsync -avzP $file "-p 22 root@$host:$dir

expect {

"yes/no" {send "yes\r";exp\_continue}

"\*password" {send "$password\r"}

}

expect eof

exit -onexit {

send\_user "oldboy say good by to you!\n"

}

============================================================

vim fenfa\_file.sh

#!/bin/sh

. /etc/init.d/functions

for ip in 101 102; do

expect fenfa\_file.exp /etc/host $ip /etc/ >/dev/null 2>&1

if [ $? -eq 0 ]; then

action 192.168.100.$ip /bin/true

else

action 192.168.100.$ip /bin/false

fi

done

sh fenfa\_file.sh

分库分表脚本

vim datebase\_backup.sh

=============================================================

#!/bin/bash

#

databases=$(mysql -s -r -uroot -poldboy123 -e "SHOW DATABASES;"|egrep -v "base|mysql|schema")

mysqldump="mysqldump -uroot -poldboy123"

mysqlcmd="mysql -uroot -poldboy123"

path=/backup

backup\_database(){

for dbname in ${databases[\*]}; do

if [ ! -d $path/${dbname}\_$(date +%F) ]; then

mkdir $path/${dbname}\_$(date +%F) -p

fi

$mysqldump -B --events $dbname >$path/${dbname}\_$(date +%F)/${dbname}\_$(date +%F).sql

done

find $path -name "\*\_\*" -mtime +7|xargs rm -rf

}

backup\_tables(){

for dbname in ${databases[\*]}; do

if [ ! -d $path/${dbname}\_$(date +%F) ]; then

mkdir $path/${dbname}\_$(date +%F) -p

fi

for tables in `$mysqlcmd -s -r -e "show tables from $dbname;"|grep -v "Tables\_in\_${dbname}"`; do

if [ ! -d $path/${dbname}\_$(date +%F)/tables\_$(date +%F) ]; then

mkdir $path/${dbname}\_$(date +%F)/tables\_$(date +%F)

fi

$mysqldump $dbname $tables >$path/${dbname}\_$(date +%F)/tables\_$(date +%F)/${tables}\_$(date +%F).sql

done

done

find $path -name "\*\_\*" -mtime +7|xargs rm -rf

}

case $1 in

database)

backup\_database

;;

table)

backup\_tables

;;

esac

#定时任务配置

echo '00 00 \* \* \* /bin/sh /server/script/datebase\_backup.sh table >/dev/null 2>&1' >>/var/spool/cron/root

echo "30 00 \* \* 6 /bin/sh /server/script/datebase\_backup.sh database >/dev/null 2>&1" >> /var/spool/cron/root

=====================================================================================

数据库远程推送脚本

vim file\_backup.sh

#!/bin/bash

#

ip=$(ifconfig eth0 | sed -n "2p" | awk -F "[ :]+" '{print $4}')

[ ! -d /mysql\_backup/$ip ] && mkdir /mysql\_backup/$ip -p

cd /mysql\_backup/$ip

tar czf mysql\_$(date +%F).tar.gz /backup/\*\_$(date +%F) /backup/\*\_$(date +%F)/tables\_$(date +%F) /var/spool/cron /application/mysql/ /etc/rc.local

rsync -avzP mysql\_$(date +%F).tar.gz rsync\_backup@192.168.100.190::backup/mysql/ --password-file=/etc/rsync.password

find /backup -type f -name "\*.tar.gz" -mtime +3|xargs rm -f

#定时任务配置

echo "00 00 \* \* \* /bin/sh /server/script/file\_backup.sh >/dev/null 2>&1" >> /var/spool/cron/root

=====================================================================================

apache服务器备份远程推送脚本

vim apache\_file.sh

#!/bin/bash

#

ip=$(ifconfig eth0 | sed -n "2p" | awk -F "[ :]+" '{print $4}')

[ ! -d /web\_backup/$ip ] && mkdir /web\_backup/$ip -p

cd /web\_backup/$ip

tar czf apache\_$(date +%F).tar.gz /etc/rc.local /etc/rsync.password /var/spool/cron/\* /server/script/\* /application/apache/\*

rsync -avzP apache\_$(date +%F).tar.gz rsync\_backup@192.168.100.190::backup/web/ --password-file=/etc/rsync.password

find /web\_backup -type f -name "\*.tar.gz" -mtime +3|xargs rm -f

#定时任务配置

echo "00 00 \* \* \* /bin/sh /server/script/apache\_file.sh >/dev/null 2>&1" >> /var/spool/cron/root

=====================================================================================

网站备份远程推送脚本

vim web\_backup.sh

#!/bin/bash

#

ip=$(ifconfig eth0 | sed -n "2p" | awk -F "[ :]+" '{print $4}')

[ ! -d /web\_backup/$ip ] && mkdir /web\_backup/$ip -p

cd /web\_backup/$ip

tar czf web\_$(date +%F).tar.gz /data/\* /etc/rc.local /var/spool/cron/\* /etc/rsync.password /etc/exports /application/sersync/\*

rsync -avzP web\_$(date +%F).tar.gz rsync\_backup@192.168.100.190::backup/web/ --password-file=/etc/rsync.password

find /web\_backup -type f -name "\*.tar.gz" -mtime +3|xargs rm -f

#定时任务配置

echo "30 00 \* \* \* /bin/sh /server/script/web\_backup.sh >/dev/null 2>&1" >> /var/spool/cron/root

=====================================================================================

#分发文件脚本

vim fenfa\_file.sh

#!/bin/sh

. /etc/init.d/functions

for ip in 190 192 193; do

rysnc -avzP $1 "-p 22 root@192.168.100.$ip:$2"

if [ $? -eq 0 ]; then

action 192.168.100.$ip /bin/true

else

action 192.168.100.$ip /bin/false

fi

done

sh fenfa\_file.sh /server/script/\* /backup/web/