NSD Project1 DAY02

1. 案例1：网站架构演变
2. 案例2：LNP+Mariadb数据库分离
3. 案例3：Web服务器集群

1 案例1：网站架构演变

1.1 问题

学习从单机架构到集群架构的演变之路：

- 单机版LNMP
- 独立数据库服务器
- Web服务器集群与Session保持
- 动静分离、数据库集群
- 各种缓存服务器
- 业务模型

1.2 步骤

此案例主要是学习网站架构演变的过程，以拓扑图和理论为主，具体实现还需要结合具体的软件。

步骤一：单机版LNMP

单机版网站，拓扑如图-1所示。

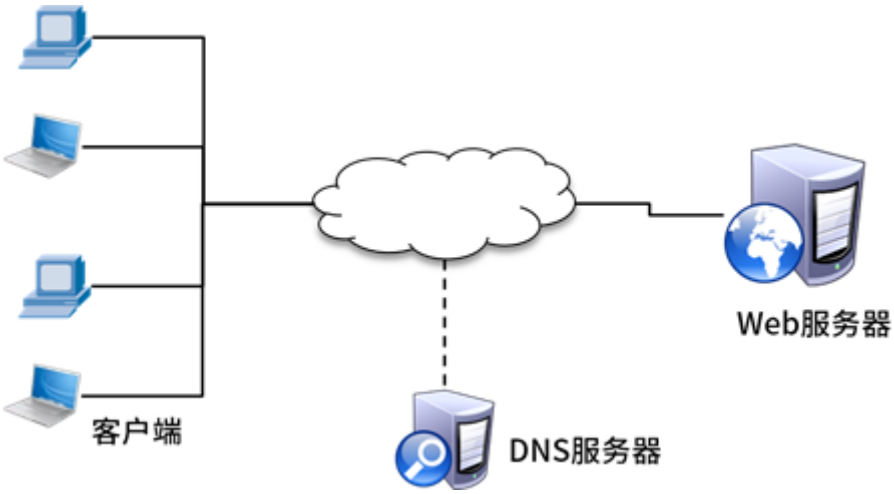

图-1 单机版网站服务器

用户量少时使用，简单、成本低、存在单点故障。

步骤二：独立数据库服务器

独立数据库服务器是将网站静态文件、代码文件等资料与数据库分离的架构，当用户量增加时单机的处理能力有限，PHP或JAVA代码的执行需要消耗大量CPU资源，数据库的增删改查需要调用大量的内存资源，将两者分离可以减轻服务器的压力，其拓扑结构如图-2所示。

Top

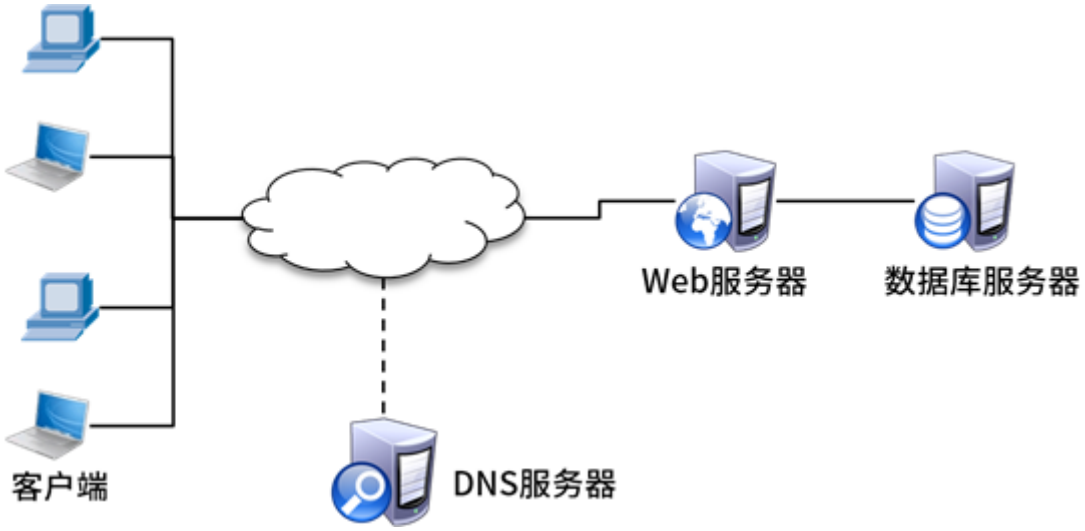

图-2 web服务器与数据库分离

Web服务器和数据库服务器的压力都可以得到有效改善，访问量有所增加。但是服务器依然存在单点故障问题。

步骤三：Web服务器集群与Session保持

我们可以通过Nginx、Haproxy代理服务器实现Web负载均衡集群，也可以使用LVS调度器实现Web负载均衡集群。部署完Web集群后还需要考虑如何进行Session会话保持，方法很多，如：根据源IP保持，代理服务器重写Cookie信息，共享文件系统保存session，使用数据库共享session等等。

该架构拓扑如图-3所示。

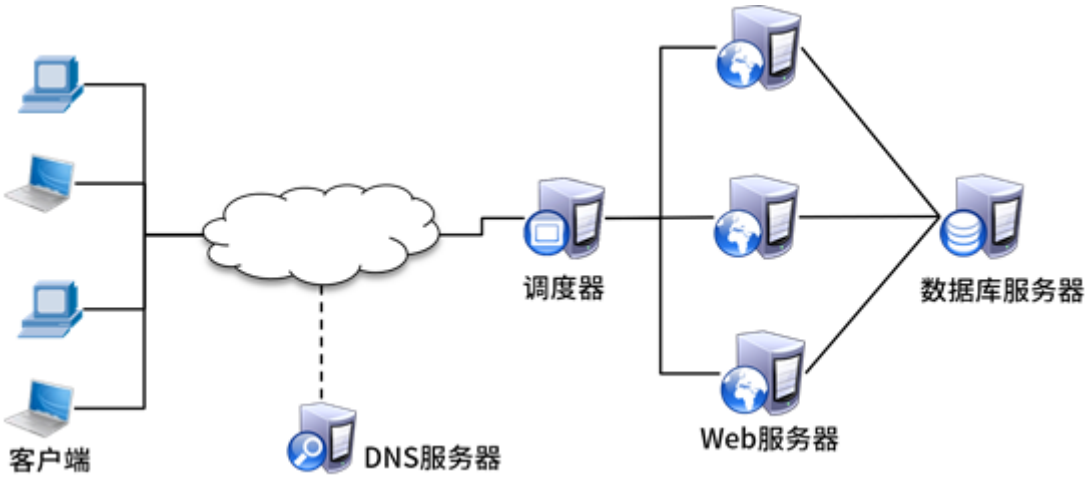

图-3

但是如果只有一台调度器依然会导致单点故障的问题，因此还需要使用Keepalived或Heartbeat之类的软件进行高可用配置，如图-4所示。

Top

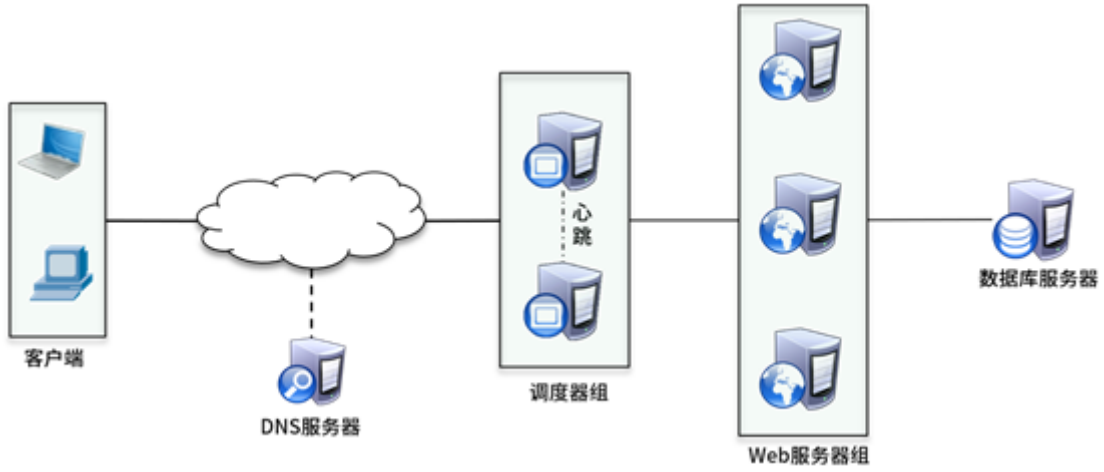

图-4

对于网站内容而言可以分离为动态页面和静态页面，静态页面就需要数据文件，动态页面则需要CPU解析代码，需要消耗大量的CPU资源，因此可以将静态和动态分离为两组服务器，动态页面有脚本代码组成，是一种基于网页的应用程序，因此这一组服务器也称为应用服务器，其架构如图-5所示。

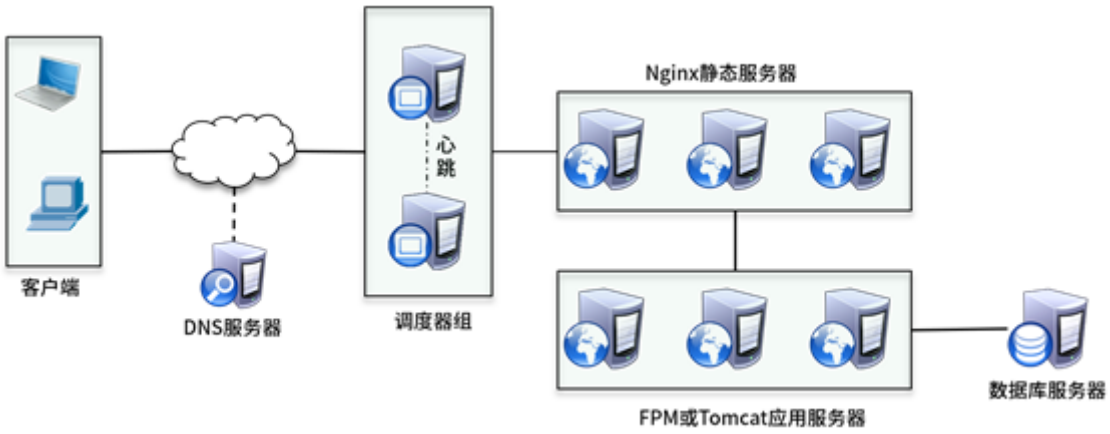

图-5

步骤四：动静分离、数据库集群

随着服务器的增加，虽然性能与并发量得到了明显的提升，但是数据的一致性、管理的便利性成为了新的问题，因此就需要增加统一的存储服务器，实现数据的同步一致，可以使用NFS，GlusterFS、Ceph等软件实现该功能，其架构如图-6所示。

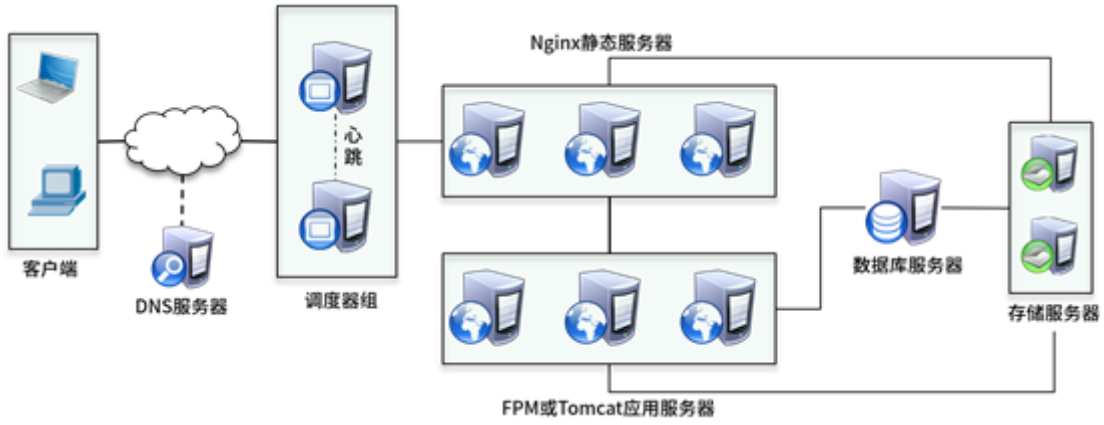

图-6

Top

此时所有应用服务器都连接一台数据库服务器进行读写操作，而且后期随着数据库中的数据不断增加，会导致数据库成为整个网站的瓶颈！这就需要我们对数据进行分库分表，创建数据库主从

或者数据库集群，实现读写分离，其拓扑如图-7所示。

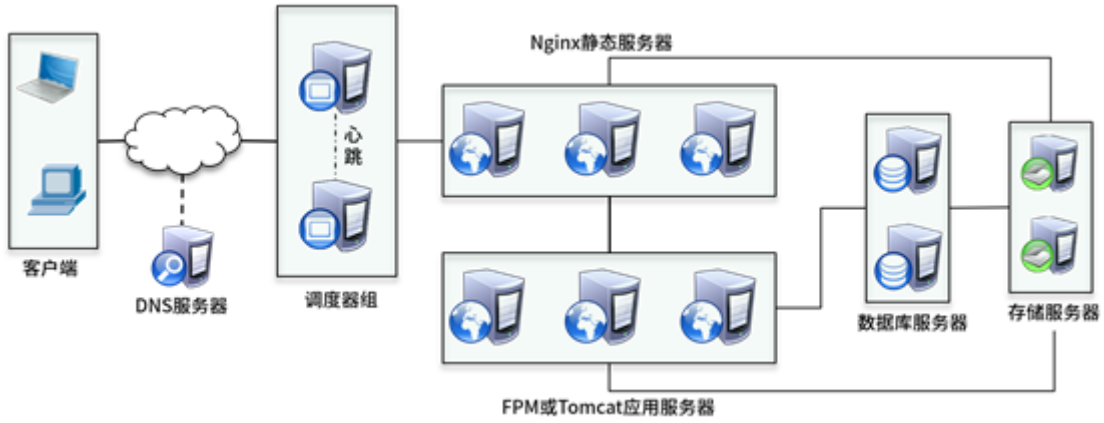

图-7

步骤四：缓存服务器与业务模型

对于静态数据我们可以通过varnish、squid或者nginx进行缓存，将数据缓存到距离用户更近的位置，构建CDN（内容分发网络）架构。

对于传统的SQL数据库而言，我们也可以通过增加NoSQL数据库，实现数据缓存的功能，提升数据库的访问速度。

备注：数据库相关知识在第三阶段课程有详细介绍，第二阶段项目暂时不做数据库优化。

最后，基于前面的架构，我们还可以将网站按照公司的业务进行分离，每个业务都可以是一个独立的集群，如图-8所示。

图-8

2 案例2：LNP+Mariadb数据库分离

2.1 问题

部署LNP+Mariadb实现数据库与Web服务器分离，实现以下目标：

- 将旧的数据库备份，迁移到新的服务器
- 修改配置调用新的数据库服务器

2.2 方案

实验拓扑如图-9所示，做具体实验前请先配置好环境。

Top

图-9

主机配置如表-1所示。

表-1

主机角色	IP 地址
client	private2 (192.168.2.254/24)
Web 服务器	eth1(192.168.2.11/24)
数据库服务器	eth1(192.168.2.21/24)

2.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：部署数据库服务器

1）准备一台独立的服务器，安装数据库软件包

```
[root@database ~]# yum -y install mariadb mariadb-server mariadb-deve
[root@database ~]# systemctl start mariadb
[root@database ~]# systemctl enable mariadb
```

2)将之前单机版LNMP网站中的数据库迁移到新的数据库服务器。

登陆192.168.2.11主机，备份数据库并拷贝给新的服务器，关闭旧的数据库服务。

```
[root@centos7 ~]# mysqldump wordpress > wordpress.bak
[root@centos7 ~]# scp wordpress.bak 192.168.2.21:/root/
[root@centos7 ~]# systemctl stop mariadb
[root@centos7 ~]# systemctl disable mariadb
```

登陆192.168.2.21主机，使用备份文件还原数据库。

创建空数据库：

```
[root@database ~]# mysql
MariaDB [(none)]> create database wordpress character set utf8mb4;
MariaDB [(none)]> exit
```

Top

使用备份文件还原数据：

```
[root@database ~]# mysql wordpress < wordpress.bak
```

重新创建账户并授权访问：

```
[root@database ~]# mysql
MariaDB [(none)]> grant all on wordpress.* to wordpress@'%' identifie
MariaDB [(none)]> flush privileges;
MariaDB [(none)]> exit
```

3）修改wordpress网站配置文件，调用新的数据库服务器。

Wordpress在第一次初始化操作时会自动生产配置文件：wp-config.php，登陆192.168.2.11修改该文件即可调用新的数据库服务。

```
[root@centos7 ~]# vim /usr/local/nginx/html/wp-config.php
修改前内容如下：
define('DB_HOST', '192.168.2.11');
修改后内容如下：
define('DB_HOST', '192.168.2.21');
```

步骤二：客户端测试

1）客户端使用浏览器访问wordpress网站。

```
[root@client ~]# firefox http://192.168.2.11
```

3 案例3：Web服务器集群

3.1 问题

使用HAProxy部署Web服务器集群，实现以下目标：

- 部署三台Web服务器
- 迁移网站数据，使用NFS实现数据共享
- 部署HAProxy代理服务器实现负载均衡
- 部署DNS域名解析服务器

Top

3.2 方案

实验拓扑如图-10所示，做具体实验前请先配置好环境。

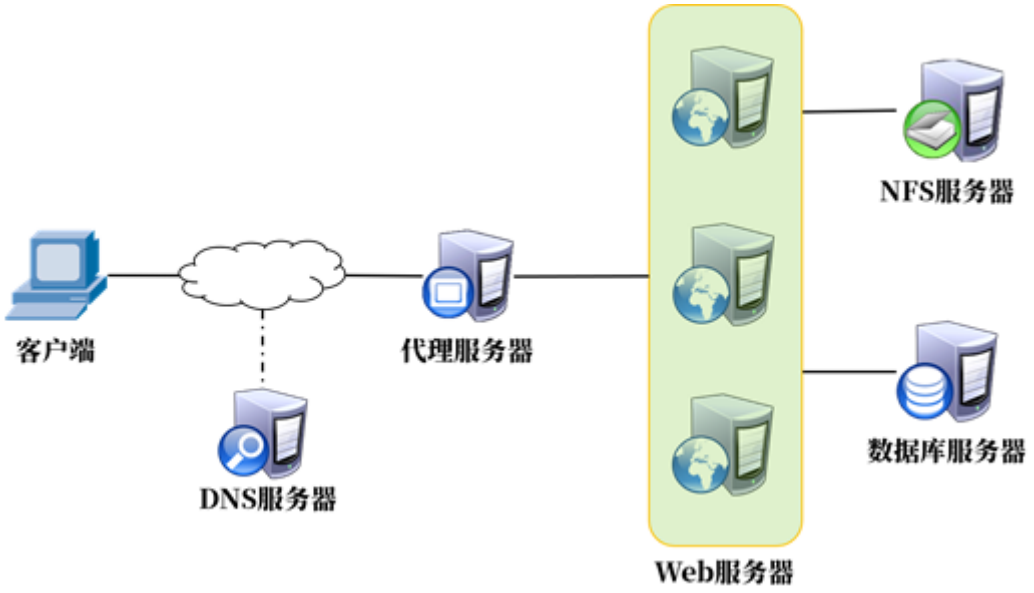

图-10

备注：实际操作中DNS服务代理服务器部署在同一台主机上（节约虚拟机资源）。

主机配置如表-2所示。

表-2

主机角色	主机名称	IP 地址
client	room9pc01	private2 (192.168.2.254/24)
代理服务器 DNS 服务器	proxy	eth0(192.168.4.5/24) eth1(192.168.2.5/24)
Web1 服务器	web1	eth1(192.168.2.11/24)
Web2 服务器	web2	eth1(192.168.2.12/24)
Web3 服务器	web3	eth1(192.168.2.13/24)
数据库服务器	database	eth1(192.168.2.21/24)
NFS 服务器	nfs	eth1(192.168.2.31/24)

3.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：部署web2和web3服务器

1）安装LNP软件包

```
01.    [root@web2 ~]# yum -y install gcc pcre-devel openssl-devel
02.    [root@web2 lnmp_soft]# tar -xf nginx-1.12.2.tar.gz
03.    [root@web2 lnmp_soft]# cd nginx-1.12.2/
04.    [root@web2 nginx-1.12.2]# ./configure \
05.    --with-http_ssl_module \
06.    --with-http_stub_status_module
07.    [root@web2 nginx-1.12.2]# make && make instal
```

Top

```
[root@web2 ~]# yum -y install php php-fpm php-mysql mariadb-devel


[root@web3 ~]# yum -y install gcc pcre-devel openssl-devel
[root@web3 lnmp_soft]# tar -xf nginx-1.12.2.tar.gz
[root@web3 lnmp_soft]# cd nginx-1.12.2/
[root@web3 nginx-1.12.2]# ./configure \
--with-http_ssl_module \
--with-http_stub_status_module
[root@web3 nginx-1.12.2]# make && make instal
[root@web3 ~]# yum -y install php php-fpm php-mysql mariadb-devel
```

2）修改nginx配置实现动静分离（web2和web3操作）

web2修改默认首页index.php，配置两个location实现动静分离。

```
[root@web2 ~]# vim /usr/local/nginx/conf/nginx.conf
location / {
            root   html;
            index  index.php index.html index.htm;
        }

location ~ \.php$ {
            root           html;
            fastcgi_pass   127.0.0.1:9000;
            fastcgi_index  index.php;
            include        fastcgi.conf;
        }
```

web3修改默认首页index.php，配置两个location实现动静分离。

```
[root@web3 ~]# vim /usr/local/nginx/conf/nginx.conf
location / {
            root   html;
            index  index.php index.html index.htm;
        }

location ~ \.php$ {
            root           html;
            fastcgi_pass   127.0.0.1:9000;
```

Top

```
10.                 fastcgi_index  index.php;
11.                 include        fastcgi.conf;
12.             }
```

3）启动相关服务

```
01.    [root@web2 ~]# echo "/usr/local/nginx/sbin/nginx" >> /etc/rc.local
02.    [root@web2 ~]# chmod +x /etc/rc.local
03.    [root@web2 ~]# /usr/local/nginx/sbin/nginx
04.    [root@web2 ~]# systemctl start  php-fpm                    #启动php-fp
05.    [root@web2 ~]# systemctl enable php-fpm
06.
07.    [root@web3 ~]# echo "/usr/local/nginx/sbin/nginx" >> /etc/rc.local
08.    [root@web3 ~]# chmod +x /etc/rc.local
09.    [root@web3 ~]# /usr/local/nginx/sbin/nginx
10.    [root@web3 ~]# systemctl start  php-fpm                    #启动php-fp
11.    [root@web3 ~]# systemctl enable php-fpm
```

步骤二：部署NFS，将网站数据迁移至NFS共享服务器

1）部署NFS共享服务器

```
01.    [root@nfs ~]# yum install nfs-utils
02.    [root@nfs ~]# mkdir /web_share
03.    [root@nfs ~]# vim /etc/exports
04.    /web_share  192.168.2.0/24(rw,no_root_squash)
05.
06.    [root@nfs ~]# systemctl restart rpcbind
07.    [root@nfs ~]# systemctl eanble rpcbind
```

NFS使用的是随机端口，每次启动NFS都需要将自己的随机端口注册到rpcbind服务，这样客户端访问NFS时先到rpcbind查询端口信息，得到端口信息后再访问NFS服务。

```
01.    [root@nfs ~]# systemctl restart nfs
02.    [root@nfs ~]# systemctl enable nfs
```

Top

2）迁移旧的网站数据到NFS共享服务器

将web1（192.168.2.11）上的wordpress代码拷贝到NFS共享。

```
[root@web1 ~]# cd /usr/local/nginx/
[root@web1 nginx]# tar -czpf html.tar.gz html/
[root@web1 nginx]# scp html.tar.gz 192.168.2.31:/web_share/
```

登陆nfs服务器，将压缩包解压

```
[root@nfs ~]# cd /web_share/
[root@nfs web_share]# tar -xf html.tar.gz
```

3)所有web服务器访问挂载NFS共享数据。

```
[root@web1 ~]# rm -rf /usr/local/nginx/html/*
[root@web1 ~]# yum -y install nfs-utils
[root@web1 ~]# echo "192.168.2.31:/web_share/html /usr/local/nginx/ht
[root@web1 ~]# mount -a

[root@web2 ~]# yum -y install nfs-utils
[root@web2 ~]# echo "192.168.2.31:/web_share/html /usr/local/nginx/ht
[root@web2 ~]# mount -a

[root@web3 ~]# yum -y install nfs-utils
[root@web3 ~]# echo "192.168.2.31:/web_share/html /usr/local/nginx/ht
[root@web3 ~]# mount -a
```

步骤三：部署HAProxy代理服务器

1）部署HAProxy

安装软件，手动修改配置文件，添加如下内容。

```
[root@proxy ~]# yum -y install haproxy
[root@proxy ~]# vim /etc/haproxy/haproxy.cfg
listen wordpress *:80
  balance roundrobin
  server web1 192.168.2.11:80 check inter 2000 rise 2 fall 3
  server web2 192.168.2.12:80 check inter 2000 rise 2 fall 3
```

Top

```
07.      server web3 192.168.2.13:80 check inter 2000 rise 2 fall 3
08.
09.    [root@proxy ~]# systemctl start haproxy
10.    [root@proxy ~]# systemctl enable haproxy
```

步骤三：部署DNS域名服务器

1）安装DNS相关软件（192.168.4.5操作）。

```
01.     [root@proxy ~]# yum -y  install bind bind-chroot
```

2）修改主配置文件，添加zone。

```
01.    [root@proxy ~]# vim /etc/named.conf
02.    options {
03.            listen-on port 53 { any; };            #服务监听的地址与端口
04.            directory       "/var/named";          #数据文件路径
05.            allow-query     { any; };              #允许任何主机访问DNS服务
06.    ... ...
07.    };
08.
09.    zone "lab.com" IN {                           #定义正向区域
10.            type master;
11.            file "lab.com.zone";
12.    };
13.
14.    #include "/etc/named.rfc1912.zones";         #注释掉改行
15.    #include "/etc/named.root.key";              #注释掉改行
16.
17.    [root@proxy ~]# named-checkconf /etc/named.conf             #检查语法
```

3）修改正向解析记录文件。

注意：保留文件权限。

```
01.    [root@proxy named]# cp -p /var/named/named.localhost /var/named/lab.c
02.    [root@proxy named]# vim /var/named/lab.zone
03.    $TTL 1D
04.    @       IN SOA  @ rname.invalid. (
```

Top

```
05.                                             0       ; serial
06.                                             1D      ; refresh
07.                                             1H      ; retry
08.                                             1W      ; expire
09.                                             3H )    ; minimum
10.     @       NS      dns.lab.com.
11.     dns     A       192.168.4.5
12.     www     A       192.168.4.5
```

4）启动服务

```
01.     [root@proxy named]# systemctl start named
02.     [root@proxy named]# systemctl enable named
```

5）客户端修改DNS解析文件

提示：做完实验修改回原始内容。

```
01.     [root@room9pc01 data]# cat /etc/resolv.conf
02.     # Generated by NetworkManager
03.     search tedu.cn
04.     nameserver 192.168.4.5
05.     nameserver 172.40.1.10
06.     nameserver 192.168.0.220
```

步骤四：修改wordpress配置文件

1）修改wp-config.php

在define('DB_NAME', 'wordpress')这行前面添加如下两行内容：

```
01.     [root@web3 html]# vim /usr/local/nginx/html/wp-config.php
02.     define('WP_SITEURL', 'http://www.lab.com');
03.     define('WP_HOME', 'http://www.lab.com');
```

如果不添加这两行配置，浏览器访问网站某个子页面后，URL会固定到某一台后端服务器不轮询。

Top