

NSD OPERATION DAY04

1. [案例1：Nginx常见问题处理](#)
2. [案例2：安装部署Tomcat服务器](#)
3. [使用Tomcat部署虚拟主机](#)

1 案例1：Nginx常见问题处理

1.1 问题

本案例要求对Nginx服务器进行适当优化，以提升服务器的处理性能：

- 不显示Nginx软件版本号
- 如果客户端访问服务器提示 “Too many open files” 如何解决
- 如何解决客户端访问头部信息过长的问题
- 开启gzip压缩功能，提高数据传输效率
- 如何让客户端浏览器缓存数据
- 如何自定义返回给客户端的404错误页面

然后客户机访问此Web服务器验证效果：

- 使用ab压力测试软件测试并发量
- 编写测试脚本生成头部信息的访问请求
- 客户端访问不存在的页面，测试404错误页面是否重定向

1.2 方案

使用2台RHEL7虚拟机，其中一台作为Nginx服务器（192.168.4.5）、另外一台作为测试用的Linux客户机（192.168.4.100），如图-1所示。

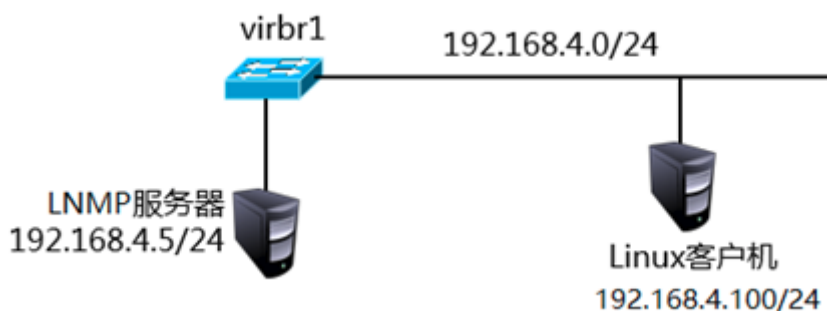


图-1

Nginx优化主要从配置文件着手，常见优化参数如下：

- worker_processes //与CPU核心数量一致
- worker_connections //每个worker最大并发连接数
- server_tokens //服务器版本号信息
- client_header_buffer_size //默认请求包头信息的缓存
- large_client_header_buffers //大请求包头信息的缓存个数与容量
- error_page 404 //自定义404错误页面

[Top](#)

1.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：构建Nginx服务器

1) 源码安装Nginx软件

```
01. [ root@svr5 ~] # yum -y install gcc pcre-devel openssl-devel //安装常见依赖包
02. [ root@svr5 ~] # useradd -s /sbin/nologin nginx
03. [ root@svr5 ~] # tar -zxvf nginx-1.8.0.tar.gz
04. [ root@svr5 ~] # cd nginx-1.8.0
05. [ root@svr5 nginx-1.8.0] # ./configure \
06. >-- prefix=/usr/local/nginx \ //指定安装路径
07. >-- user=nginx \ //指定用户
08. >-- group=nginx \ //指定组
09. >-- with-http_ssl_module //开启SSL加密功能
10. [ root@svr5 nginx-1.8.0] # make && make install //编译并安装
```

2) 启用Nginx服务并查看监听端口状态

```
01. [ root@svr5 ~] # /usr/local/nginx/sbin/nginx
02. [ root@svr5 ~] # netstat -anptu | grep nginx
03. tcp      0      0 0.0.0.0:80      0.0.0.0:*      LISTEN    10441/nginx
```

步骤二：优化前从客户机访问Nginx服务器测试

1) 使用ab高并发测试

```
01. [ root@svr100 ~] # ab -n 2000 -c 2000 http://192.168.4.5/
02. Benchmarking 192.168.4.5 ( be patient)
03. socket: Too many open files ( 24) //提示打开文件数量过多
```

2) 使用脚本测试长头部请求是否能获得响应

```
01. [ root@svr100 ~] # cat buffer.sh
02. #!/bin/bash
03. URL=http://192.168.4.5/index.html?
04. for i in { 1..5000}
05. do
06.     URL=${URL}v$i=$i
```

[Top](#)

```

07. done
08. curl $URL
09. [ root@svr100 ~] # chmod +x buffer.sh
10. [ root@svr100 ~] # ./buffer.sh
11. ...
12. <center><h1>414 Request-URI Too Large</h1></center> //提示头部信息过大

```

3) 使用Firefox浏览器测试客户端缓存

以Firefox浏览器为例，只要在地址栏内输入 `http://192.168.4.5/a.jpg`，回车后即连接目标主机192.168.4.5的Web服务，获得服务器上的a.jpg图片资源。若访问成功，再次，在Firefox地址栏内输入`about:cache`将显示Firefox浏览器的缓存信息，如图-2所示。



图-2

4) 客户端使用浏览器访问不存在的页面

```

01. [ root@svr5 ~] # firefox http://192.168.4.5/tt.html //访问不存在的页面

```

步骤三：优化Nginx服务器

1) 修改Nginx配置文件

```

01. [ root@svr5 ~] # vim /usr/local/nginx/conf/nginx.conf
02. ...
03. worker_processes 2; //与CPU核心数量一致
04. events {
05.     worker_connections 65535; //每个worker最大并发连接数
06.     use epoll;

```

[Top](#)

```

07.     }
08.     http {
09.         server_tokens off;           //不显示nginx版本号信息
10.         client_header_buffer_size 1k; //默认请求包头信息的缓存
11.         large_client_header_buffers 4 4k; //大请求包头部信息的缓存个数与容量
12.         gzip on;
13.         gzip_min_length 1000;
14.         gzip_comp_level 4;
15.         gzip_types text/plain text/css application/json application/x-javascript text/xml application,
16.
17.     server {
18.         listen 80;
19.         server_name www.tarena.com;
20.         location / {
21.             root html;
22.             index index.html index.htm;
23.         }
24.         location ~* \.(jpg|jpeg|gif|png|css|js|ico|xml) $ {
25.             expires 30d;           //定义客户端缓存时间为30天
26.         }
27.         error_page 404 /40x.html; //自定义错误页面
28.         location = /40x.html {
29.             root html;
30.         }
31.     }
32. }

```

2) 修改Linux操作系统最大打开文件数

通过修改/etc/security/limits.conf文件修改打开文件最大数量：

```

01. [ root@svr5 ~ ] # vim /etc/security/limits.conf
02. *                soft nofile 100000
03. *                hard nofile 100000
04. [ root@svr5 ~ ] # ulimit - Hn 100000
05. [ root@svr5 ~ ] # ulimit - Sn 100000

```

3) 提前生成404错误页面，供测试使用：

[Top](#)

```
01. [root@svr5 ~] # vim /usr/local/nginx/html/40x.html
02. <h1>~~~~~Error~~~~~</h1>
```

步骤四：优化后从客户机访问Nginx服务器测试

对Nginx服务器进行各种参数优化后，在客户端访问服务器页面，对比优化前与优化后的区别，验证优化是否生效。

2 案例2：安装部署Tomcat服务器

2.1 问题

本案例要求部署Tomcat服务器，具体要求如下：

- 安装部署JDK基础环境
- 安装部署Tomcat服务器
- 创建JSP测试页面，文件名为test.jsp，显示服务器当前时间

然后客户机访问此Web服务器验证效果：

- 使用火狐浏览器访问Tomcat服务器的8080端口，浏览默认首页
- 使用火狐浏览器访问Tomcat服务器的8080端口，浏览默认测试页面

2.2 方案

使用2台RHEL7虚拟机，其中一台作为Tomcat服务器（192.168.2.5）、另外一台作为测试用的Linux客户机（192.168.2.100），如图-3所示。

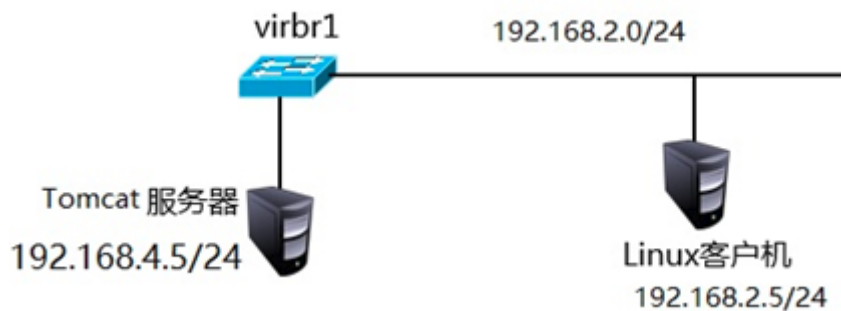


图-3

使用RPM安装JDK基础环境

使用源码安装部署Tomcat服务器

调整Tomcat配置文件，禁用默认的8009端口

2.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：部署服务器软件

1) 使用RPM安装JDK环境

[Top](#)

01. [root@svr5 ~] # yum - y install java- 1.8.0- openjdk //安装JDK
02. [root@svr5 ~] # yum - y install java- 1.8.0- openjdk- headless //安装JDK
03. [root@svr5 ~] # java - version //查看JAVA版本

2) 安装Tomcat

01. [root@svr5 ~] # tar - xzf apache- tomcat- 8.0.30.tar.gz
02. [root@svr5 ~] # mv apache- tomcat- 8.0.30 /usr/local/tomcat
03. [root@svr5 ~] # ls /usr/local/tomcat
04. bin/ //主程序目录
05. lib/ //库文件目录
06. logs/ //日志目录
07. temp/ //临时目录
08. work/ //自动编译目录jsp代码转换servlet
09. conf/ //配置文件目录
10. webapps/ //页面目录

步骤二：修改Tomcat配置文件

1) 创建测试JSP页面

01. [root@svr5 ~] # vim /usr/local/tomcat/webapps/ROOT/test.jsp
02. <html>
03. <body>
04. <center>
05. Now time is: <%=new java.util.Date() %> //显示服务器当前时间
06. </center>
07. </body>
08. </html>

2) 启动服务

01. [root@svr5 ~] # /usr/local/tomcat/bin/startup.sh

步骤三：验证测试

[Top](#)

1) 服务器验证端口信息

```

01. [root@svr5 ~]# netstat -nltlp | grep java //查看java监听的端口
02. tcp      0      0 0.0.0.0:8080 0.0.0.0:* LISTEN 2778/java
03. tcp      0      0 0.0.0.0:8005 0.0.0.0:* LISTEN 2778/java

```

2) 客户端浏览测试页面

```

01. [root@client ~]# firefox http://127.0.0.1:8080
02. [root@client ~]# firefox http://127.0.0.1:8080/test.jsp

```

3 使用Tomcat部署虚拟主机

3.1 问题

沿用练习二，使用Tomcat部署加密虚拟主机，实现以下要求：

- 实现两个基于域名的虚拟主机，域名分别为：www.aa.com和 www.bb.com
- 使用www.aa.com域名访问的页面根路径为/usr/local/tomcat/aa/ROOT
- 使用www.bb.com域名访问的页面根路径为/usr/local/tomcat/bb/ROOT
- 访问页面时支持SSL加密通讯
- 私钥、证书存储路径为/usr/local/tomcat/conf/cert
- 每个虚拟主机都拥有独立的访问日志文件

3.2 方案

修改server.xml配置文件，创建两个域名的虚拟主机，修改如下两个参数块：

```

01. # cat /usr/local/tomcat/conf/server.xml
02. ... ..
03. <Host name="www.aa.com" appBase="aa" unpackWARs="true" autoDeploy="true">
04. </Host>
05. <Host name="www.bb.com" appBase="bb" unpackWARs="true" autoDeploy="true">
06. </Host>
07. ... ..

```

提示：修改server.xml配置文件，通过在<Host>中添加Context，我们也可以指定页面的根路径位置，默认为ROOT

```

01. <Context path="/test" docBase="/var/www/html/" />

```

[Top](#)

```
01. #key tool - genkey pair - alias tomcat - key alg RSA - key store /usr/local/tomcat/key store
```

3.3 步骤

实现此案例需要按照如下步骤进行。

步骤一：配置服务器虚拟主机

1) 修改server.xml配置文件，创建虚拟主机

```
01. [ root@svr5 ~] # vim /usr/local/tomcat/conf/server.xml
02. ... ..
03. <Host name="www.aa.com" appBase="aa" unpackWARS="true" autoDeploy="true">
04. </Host>
05. <Host name="www.bb.com" appBase="bb" unpackWARS="true" autoDeploy="true">
06. </Host>
```

2) 创建虚拟主机对应的页面根路径

```
01. [ root@svr5 ~] # mkdir -p /usr/local/tomcat/{ aa,bb }/ROOT
02. [ root@svr5 ~] # echo "AAA" > /usr/local/tomcat/aa/ROOT/index.html
03. [ root@svr5 ~] # echo "BBB" > /usr/local/tomcat/bb/ROOT/index.html
```

3) 重启Tomcat服务器

```
01. [ root@svr5 ~] # /usr/local/tomcat/bin/shutdown.sh
02. [ root@svr5 ~] # /usr/local/tomcat/bin/startup.sh
```

4) 客户端设置host文件，并浏览测试页面进行测试

```
01. [ root@client ~] # vim /etc/hosts
02. ... ..
03. 192.168.4.5 www.aa.com www.bb.com
04. [ root@client ~] # firefox http://www.aa.com:8080/
05. [ root@client ~] # firefox http://www.bb.com:8080/
```

[Top](#)

步骤二：修改网站的首页目录（非必须）

1) 使用docBase参数可以修改默认网站首页路径

```
01. [ root@svr5 ~] # vim /usr/local/tomcat/conf/server.xml
02. ... ..
03.
04. <Host name="www.aa.com" appBase="aa" unpackWARS="true" autoDeploy="true">
05. <Context path="" docBase="base" />
06. </Host>
07.
08. <Host name="www.bb.com" appBase="bb" unpackWARS="true" autoDeploy="true">
09. </Host>
10. ... ..
11. [ root@svr5 ~] # mkdir /usr/local/tomcat/aa/base
12. [ root@svr5 ~] # echo "BASE" > /usr/local/tomcat/aa/base/index.html
13. [ root@svr5 ~] # /usr/local/tomcat/bin/shutdown.sh
14. [ root@svr5 ~] # /usr/local/tomcat/bin/startup.sh
```

2) 测试查看页面是否正确

```
01. [ root@client ~] # firefox http://www.aa.com:8080/ //结果为base页面的内容
```

步骤三：跳转（非必须）

1) 当用户访问http://www.aa.com/test打开/var/www/html目录下的页面

```
01. [ root@svr5 ~] # vim /usr/local/tomcat/conf/server.xml
02. ... ..
03.
04. <Host name="www.aa.com" appBase="aa" unpackWARS="true" autoDeploy="true">
05. <Context path="/test" docBase="/var/www/html/" />
06. </Host>
07.
08. <Host name="www.bb.com" appBase="bb" unpackWARS="true" autoDeploy="true">
09. </Host>
10. ... ..
11. [ root@svr5 ~] # echo "Test" > /var/www/html/index.html
12. [ root@svr5 ~] # /usr/local/tomcat/bin/shutdown.sh
13. [ root@svr5 ~] # /usr/local/tomcat/bin/startup.sh
```

[Top](#)

2) 测试查看页面是否正确

01. [root@client ~] # firefox http://www.aa.com: 8080/test
02. //返回/var/www/html/index.html的内容

步骤四：配置Tomcat支持SSL加密网站

1) 创建加密用的私钥和证书文件

01. [root@svr5 ~] # keytool - genkey pair - alias tomcat - key alg RSA - key store /usr/local/tom



2)再次修改server.xml配置文件，创建支持加密连接的Connector

01. [root@svr5 ~] # vim /usr/local/tomcat/conf/server.xml
02.
03. <Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
04. maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
05. keyStoreFile="/usr/local/tomcat/keystore" keyStorePass="123456" clientAuth="false" ssl



3) 重启Tomcat服务器

01. [root@svr5 ~] # /usr/local/tomcat/bin/catalina.sh stop
02. [root@svr5 ~] # /usr/local/tomcat/bin/catalina.sh start

4) 客户端设置host文件，并浏览测试页面进行测试

01. [root@client ~] # vim /etc/hosts
02.
03. 192.168.4.5 www.aa.com www.bb.com
04. [root@client ~] # firefox https://www.test.com: 8443/
05. [root@client ~] # firefox https://www.test.com: 8443/

[Top](#)

步骤四：配置Tomcat日志

1)为每个虚拟主机设置不同的日志文件

```
01. [ root@svr5 ~] # vim /usr/local/tomcat/conf/server.xml
02. <Host name="www.aa.com" appBase="aa" unpackWARs="true" autoDeploy="true">
03.     <Valve className="org.apache.catalina.valves.AccessLogValve"
04.         prefix="aa_access" suffix=".log"
05.         pattern="common"/>
06. </Host>
07. <Host name="www.bb.com" appBase="bb" unpackWARs="true" autoDeploy="true">
08.     <Valve className="org.apache.catalina.valves.AccessLogValve"
09.         prefix="bb_access" suffix=".log"
10.         pattern="common"/>
11. </Host>
```

2) 重启Tomcat服务器

```
01. [ root@svr5 ~] # /usr/local/tomcat/bin/catalina.sh stop
02. [ root@svr5 ~] # /usr/local/tomcat/bin/catalina.sh start
```

3) 查看服务器日志文件

```
01. [ root@svr5 ~] # ls /usr/local/tomcat/logs/
```