第三十一章：Tomcat应用企业级优化

**一、定义默认页面、页面超时、禁止列出目录；**

**二、禁用AJP协议连接器；**

**三、配置网页传输压缩；**

**四、tomcat的三种模式切换；**

**五、解决重启tomcat服务后，8005端口延迟启动的问题；**

**六、生产环境tomcat配置；**

**七、JVM的优化管理；**

**案例环境：按照“第三十一章tomcat应用部署”文档进行部署**

**一、定义默认页面、页面超时、禁止列出目录；**

[root@www ~]# vi /usr/local/tomcat/conf/web.xml

115 <init-param>

116 <param-name>listings</param-name>

117 <param-value>false</param-value>

118 </init-param>

##禁止列出目录

606 <session-config>

607 <session-timeout>30</session-timeout>

608 </session-config>

##页面超时

4704 <welcome-file-list>

4705 <welcome-file>index.html</welcome-file>

4706 <welcome-file>index.htm</welcome-file>

4707 <welcome-file>index.jsp</welcome-file>

4708 </welcome-file-list>

##默认页面

:wq

[root@www ~]# /usr/local/tomcat/bin/shutdown.sh

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

[root@www ~]# netstat -utpln |grep java

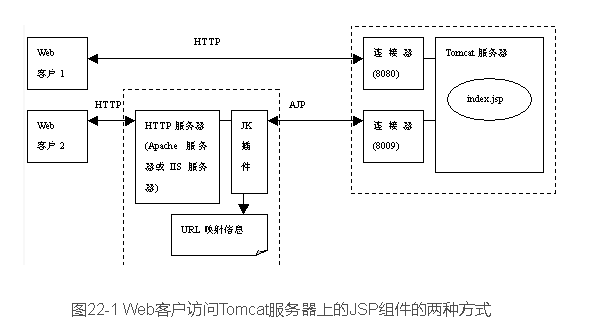
tcp 0 0 0.0.0.0:8080 0.0.0.0:\* LISTEN 14913/java

tcp 0 0 127.0.0.1:8005 0.0.0.0:\* LISTEN 14913/java

tcp 0 0 0.0.0.0:8009 0.0.0.0:\* LISTEN 14913/java

**二、禁用AJP协议连接器；**

**概述：**AJP13是定向包协议，通常用于代理或者防火墙重定向；



[root@www ~]# vi /usr/local/tomcat/conf/server.xml

115 <!-- Define an AJP 1.3 Connector on port 8009

116 <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />

117 -->

:wq

[root@www ~]# /usr/local/tomcat/bin/shutdown.sh

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

[root@www ~]# netstat -utpln |grep java

tcp 0 0 0.0.0.0:8080 0.0.0.0:\* LISTEN 14992/java

tcp 0 0 127.0.0.1:8005 0.0.0.0:\* LISTEN 14992/java

**三、配置网页传输压缩；**

**注：如若有apache、nginx等做代理，tomcat则不必配置传输压缩；**

[root@www ~]# vi /usr/local/tomcat/conf/server.xml

69 <Connector port="8080" protocol="HTTP/1.1"

70 connectionTimeout="20000"

71 redirectPort="8443"

72 compression="on" ##开启压缩

73 compressionMinSize="50" ##指定最小的压缩文件，单位是字节

74 noCompressionUserAgents="gozilla, traviata" ##此浏览器类型不进行压缩

75 compressableMimeType="text/html,text/xml,text/javascript,text/css,text/plain" /> ##压缩文件的格式

:wq

[root@www ~]# /usr/local/tomcat/bin/shutdown.sh

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

**四、tomcat的三种模式切换；**

**1.先了解四个概念：**

同步 ： 自己亲自出马持银行卡到银行取钱（使用同步IO时，Java自己处理IO读写）。

异步 ： 委托一小弟拿银行卡到银行取钱，然后给你（使用异步IO时，Java将IO读写委托给OS处理，需要将数据缓冲区地址和大小传给OS(银行卡和密码)，OS需要支持异步IO操作API）。

阻塞 ： ATM排队取款，你只能等待（使用阻塞IO时，Java调用会一直阻塞到读写完成才返回）。

非阻塞 ： 柜台取款，取个号，然后坐在椅子上做其它事，等号广播会通知你办理，没到号你就不能去，你可以不断问大堂经理排到了没有，大堂经理如果说还没到你就不能去（使用非阻塞IO时，如果不能读写Java调用会马上返回，当IO事件分发器会通知可读写时再继续进行读写，不断循环直到读写完成）。

**2.Java对BIO、NIO、AIO的支持：**

Java BIO ： 同步并阻塞，服务器实现模式为一个连接一个线程，即客户端有连接请求时服务器端就需要启动一个线程进行处理，如果这个连接不做任何事情会造成不必要的线程开销，当然可以通过线程池机制改善。

Java NIO ： 同步非阻塞，服务器实现模式为一个请求一个线程，即客户端发送的连接请求都会注册到多路复用器上，多路复用器轮询到连接有I/O请求时才启动一个线程进行处理。

Java AIO（NIO.2） ： 异步非阻塞，服务器实现模式为一个有效请求一个线程，客户端的I/O请求都是由OS先完成了再通知服务器应用去启动线程进行处理

**3.BIO、NIO、AIO适用场景分析:**

BIO方式适用于连接数目比较小且固定的架构，这种方式对服务器资源要求比较高，并发局限于应用中，JDK1.4以前的唯一选择，但程序直观简单易理解。

NIO方式适用于连接数目多且连接比较短（轻操作）的架构，比如聊天服务器，并发局限于应用中，编程比较复杂，JDK1.4开始支持。



**4.配置：**

**模式一 Bio：**

**模式二Nio：**

[root@www ~]# vi /usr/local/tomcat/conf/server.xml

<Connector port="8080" protocol="org.apache.coyote.http11.Http11NioProtocol"

connectionTimeout="20000"

redirectPort="8443"

compression="on"

compressionMinSize="50"

noCompressionUserAgents="gozilla, traviata"

compressableMimeType="text/html,text/xml,text/javascript,text/css,text/plain" />

:wq

[root@www ~]# /usr/local/tomcat/bin/shutdown.sh

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

[root@www ~]# tail /usr/local/tomcat/logs/catalina.out

20-Jul-2018 02:55:16.655 信息 [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["http-nio-8080"]

**模式三Apr：**

[root@www ~]# rpm -e apr --nodeps

[root@www ~]# yum -y install apr apr-devel

[root@www ~]# cp /usr/local/tomcat/bin/tomcat-native.tar.gz /root/

[root@www ~]# tar zxvf tomcat-native.tar.gz

[root@www ~]# cd tomcat-native-1.2.17-src/native/

[root@www tomcat-native-1.2.17-src]# ./configure --with-apr=/usr/bin/apr-1-config

[root@www tomcat-native-1.2.17-src # make && make install

Libraries have been installed in: ##安装成功的提示

/usr/local/apr/lib

[root@www tomcat-native-1.2.17-src]# cd

[root@www ~]# vi /usr/local/tomcat/bin/catalina.sh ##末尾添加，设置环境变量

CATALINA\_OPTS="-Djava.library.path=/usr/local/apr/lib"

[root@www ~]# vi /usr/local/tomcat/conf/server.xml

69 <Connector port="8080" protocol="org.apache.coyote.http11.Http11AprProtocol"

70 connectionTimeout="20000"

71 compression="on"

72 compressionMinSize="50"

73 noCompressionUserAgents="gozilla, traviata"

74 compressableMimeType="text/html,text/xml,text/javascript,text/css,text/plain" />

:wq

[root@www ~]# vi /etc/profile ##在最后一行添加

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

:wq

[root@www ~]# /usr/local/tomcat/bin/shutdown.sh

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

[root@www ~]# netstat -utpln |grep java

tcp 0 0 0.0.0.0:8080 0.0.0.0:\* LISTEN 2748/java

tcp 0 0 127.0.0.1:8005 0.0.0.0:\* LISTEN 2748/java

[root@www ~]# tail /usr/local/tomcat/logs/catalina.out

20-Jul-2018 02:58:42.867 信息 [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["http-apr-8080"]

**五、解决重启tomcat服务后，8005端口延迟启动的问题；**

[root@www ~]# vi /usr/local/java/jre/lib/security/java.security

117 securerandom.source=file:/dev/urandom

:wq

**六、生产环境tomcat配置；**

[root@www ~]# vi /usr/local/tomcat/conf/server.xml

<Connector port="8080" protocol="org.apache.coyote.http11.Http11NioProtocol"

URIEncoding="UTF-8"

minSpareThreads="25"

maxSpareThreads="75"

enableLookups="false"

disableUploadTimeout="true"

connectionTimeout="20000"

acceptCount="300"

maxThreads="300"

maxProcessors="1000"

minProcessors="5"

<!--

compression="on"

compressionMinSize="2048"

compressableMimeType="text/html,text/xml,text/javascript,text/css,text/plain"

-->

redirectPort="8443"

/>

<Connector port="8080" protocol="org.apache.coyote.http11.Http11NioProtocol"

#nio模式利用java的异步io护理技术,noblocking IO技术.

URIEncoding="UTF-8"

#设置编码

minSpareThreads="25"

#Tomcat初始化时创建的 socket线程数

maxSpareThreads="75"

#Tomcat连接器的最大空闲socket 线程数，一旦创建的线程超过这个值，Tomcat就会关闭不再需要的socket线程。默认值50

enableLookups="false"

#屏蔽DNS查询

disableUploadTimeout="true" #该标志位表明当执行servlet时，是否允许servlet容器使用一个不同的、更长的连接超时。启用该标志位将导致在上传数据时，要么使用更长的时间完成上传，要么出现更长的超时。如果不指定，该属性为“false”。

connectionTimeout="20000"

#网络超时时间

acceptCount="300"

#容许的最大连接数，一般设置为maxProcessors的1.5倍即可，满了之后客户请求会被拒绝（不能小于maxSpareThreads ）

maxThreads="300"

#客户请求最大线程数，默认值为“200”

maxProcessors="1000"

#最大连接线程数，即：并发处理的最大请求数，默认值为75 ，一旦创建的线程超过这个值，Tomcat就会关闭不再需要的socket线程

minProcessors="5"

#最小空闲连接线程数，用于提高系统处理性能，默认值为10

<!-- 前端使用nginx作为反向代理，不需要启用tomcat压缩功能。

compression="on"

#打开压缩功能

compressionMinSize="2048"

#启用压缩的输出内容大小，单位是字节，这里面默认为2KB

compressableMimeType="text/html,text/xml,text/javascript,text/css,text/plain"

#压缩类型

-->

redirectPort="8443"

#开启安全通道ssl

/>

**七、JVM的优化管理；**