参考：<http://blog.csdn.net/centre10/article/details/50639693>

# 如何优化tomcat配置(从内存、并发、缓存4个方面)优化

优化Tomcat有很多方面，我从内存、并发、缓存四个方面介绍优化方法。

## 一.Tomcat内存优化

　　Tomcat内存优化主要是对 tomcat 启动参数优化，我们可以在 tomcat 的启动脚本 catalina.sh 中设置 java\_OPTS 参数。   
　　JAVA\_OPTS参数说明   
　　-server 启用jdk 的 server 版；   
　　-Xms java虚拟机初始化时的最小内存；   
　　-Xmx java虚拟机可使用的最大内存；   
　　-XX: PermSize 内存永久保留区域   
　　-XX:MaxPermSize 内存最大永久保留区域   
　　服务器参数配置

　　　　JAVA\_OPTS=’-Xms1024m -Xmx2048m -XX: PermSize=256M -XX:MaxNewSize=256m -XX:MaxPermSize=256m’

　　配置完成后可重启Tomcat ，通过以下命令进行查看配置是否生效：　　首先查看Tomcat 进程号：

　　sudo lsof -i:9027

　　我们可以看到Tomcat 进程号是 12222 。   
　　查看是否配置生效：

　　sudo jmap – heap 12222

　　我们可以看到MaxHeapSize 等参数已经生效。

## 二.Tomcat并发优化

\*\*

　　1.Tomcat连接相关参数connector

　　在Tomcat 配置文件 server.xml 中的

　　<Connector port="9027"

　　protocol="HTTP/1.1"

　　maxHttpHeaderSize="8192"

　　minProcessors="100"

　　maxProcessors="1000"

　　acceptCount="1000"

　　redirectPort="8443"

　　disableUploadTimeout="true"/>

　　2.调整连接器connector的并发处理能力

　　1>参数说明

　　maxThreads 客户请求最大线程数   
　　minSpareThreads Tomcat初始化时创建的 socket 线程数   
　　maxSpareThreads Tomcat连接器的最大空闲 socket 线程数   
　　enableLookups 若设为true, 则支持域名解析，可把 ip 地址解析为主机名   
　　redirectPort 在需要基于安全通道的场合，把客户请求转发到基于SSL 的 redirectPort 端口   
　　acceptAccount 监听端口队列最大数，满了之后客户请求会被拒绝（不能小于maxSpareThreads ）   
　　connectionTimeout 连接超时   
　　minProcessors 服务器创建时的最小处理线程数   
　　maxProcessors 服务器同时最大处理线程数   
　　URIEncoding URL统一编码

2>Tomcat中的配置示例

　　<Connector port="9027"

　　protocol="HTTP/1.1"

　　maxHttpHeaderSize="8192"

　　maxThreads="1000"

　　minSpareThreads="100"

　　maxSpareThreads="1000"

　　minProcessors="100"

　　maxProcessors="1000"

　　enableLookups="false"

　　URIEncoding="utf-8"

　　acceptCount="1000"

　　redirectPort="8443"

　　disableUploadTimeout="true"/>

3.Tomcat缓存优化-也在connector上配置

　　1>参数说明

　　compression 打开压缩功能   
　　compressionMinSize 启用压缩的输出内容大小，这里面默认为2KB   
　　compressableMimeType 压缩类型   
　　connectionTimeout 定义建立客户连接超时的时间. 如果为 -1, 表示不限制建立客户连接的时间

　　2>Tomcat中的配置示例

　　<Connector port="9027"

　　protocol="HTTP/1.1"

　　maxHttpHeaderSize="8192"

　　maxThreads="1000"

　　minSpareThreads="100"

　　maxSpareThreads="1000"

　　minProcessors="100"

　　maxProcessors="1000"

　　enableLookups="false"

　　compression="on"

　　compressionMinSize="2048"

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

　　connectionTimeout="20000"

　　URIEncoding="utf-8"

　　acceptCount="1000"

　　redirectPort="8443"

　　disableUploadTimeout="true"/>

4.参考配置

　　1>旧有的配置

　　参考网络对服务器做过如下配置，拿出来分享下：

　　<Connector port="9027"

　　protocol="HTTP/1.1"

　　maxHttpHeaderSize="8192"

　　maxThreads="1000"

　　minSpareThreads="25"

　　maxSpareThreads="75"

　　enableLookups="false"

　　compression="on"

　　compressionMinSize="2048"

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

　　connectionTimeout="20000"

　　URIEncoding="utf-8"

　　acceptCount="200"

　　redirectPort="8443"

　　disableUploadTimeout="true" />

后来发现在访问量达到3 百万多的时候出现性能瓶颈。   
2>更改后的配置

　　<Connector port="9027"

　　protocol="HTTP/1.1"

　　maxHttpHeaderSize="8192"

　　maxThreads="1000"

　　minSpareThreads="100"

　　maxSpareThreads="1000"

　　minProcessors="100"

　　maxProcessors="1000"

　　enableLookups="false"

　　compression="on"

　　compressionMinSize="2048"

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

　　connectionTimeout="20000"

　　URIEncoding="utf-8"

　　acceptCount="1000"

　　redirectPort="8443"

　　disableUploadTimeout="true"/>