

BestTest 悦分享越快乐

——专业的软件测试自学与分享平台

linux监控分析调优 www.besttest.cn
<http://weibo.com/besttest>

安大叔

官方网站：www.besttest.cn

官方微博：<http://weibo.com/besttest>

微信订阅号：besttest

BestTest 悦分享越快乐

- 欢迎加入技术交流群共同探讨测试技术：
- 性能测试交流群：244735278
- 纯测试技术交流群：249854093
- 自动化测试交流群：37753989
- 安全测试交流群：82280498
- 软件测试招聘群：202830590
- 软件测试吐槽群（可闲聊的群）：64779175



自学与分享平台：www.besttest.cn 微博：<http://weibo.com/besttest> 微信订阅号：besttest

tomcat目录

目录:

- tomcat介绍
- tomcat安装
- tomcat使用
- tomcat监控
- tomcat分析调优

www.besttest.cn

<http://weibo.com/besttest>

Tomcat简单介绍

- Tomcat 服务器是一个免费的开放源代码的轻量级 Web 应用服务器，如apache处理静态HTML能力突出不同，tomcat处理动态HTML能力相当强大，因此一般项目都是部署apache+tomcat，由apache处理程序的静态文件，有tomcat处理JSP页面和Servlet

Tomcat目录

•tomcat目录结构

- /bin : 存放windows或Linux平台上启动和关闭Tomcat的[脚本文件](#)
- /conf : 存放Tomcat服务器的各种全局[配置文件](#) , 其中最重要的是server.xml和web.xml
- /doc : 存放Tomcat文档
- /server : 包含三个子目录 : classes、lib和webapps
- /server/lib : 存放Tomcat服务器所需的各种JAR文件
- /server/webapps : 存放Tomcat自带的两个WEB应用admin应用和 manager应用
- /common/lib : 存放Tomcat服务器以及所有web应用都可以访问的jar文件
- /shared/lib : 存放所有web应用都可以访问的jar文件 (但是不能被Tomcat服务器访问)
- /logs : 存放Tomcat执行时的日志文件
- /src : 存放Tomcat的[源代码](#)
- /webapps : Tomcat的主要Web发布目录 , 默认情况下把Web应用文件放于此目录
- /work : 存放JSP编译后产生的[class文件](#)

Tomcat配置方法

- 添加JVM
- catalina.sh中添加
- `JAVA_OPTS="-Xms256m -Xmx512m -XX:MaxPermSize=128m -Dcom.sun.management.jmxremote.port=1090 -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.authenticate=false"`

Tomcat配置方法

•部署一个web应用

- 有两个办法可以在系统中部署web服务。
- 拷贝WAR文件或者web应用文件夹（包括该web的所有内容）到\$CATALINA_BASE/webapps目录下。
- 如果有一个WAR文件，想部署它，则只需要把该文件简单的拷贝到CATALINA_BASE/webapps目录下即可，文件必须以“.war”作为扩展名。一旦Tomcat监听到这个文件，它将（缺省的）解开该文件包作为一个子目录，并以WAR文件的文件名作为子目录的名字。接下来，Tomcat将在内存中建立一个context，就好象在server.xml文件里建立一样。当然，其他必需的内容，将从server.xml中的DefaultContext获得。
- appBase自定义工程路径

Tomcat后台监控

- 在conf文件夹下找到一个tomcat-users.xml 里增加用户

```
<?xml version='1.0' encoding='utf-8'?>
```

```
<tomcat-users>
```

```
<role rolename="tomcat"/>
```

```
<role rolename="role1"/>
```

```
<role rolename="manager"/>
```

```
<role rolename="admin"/>
```

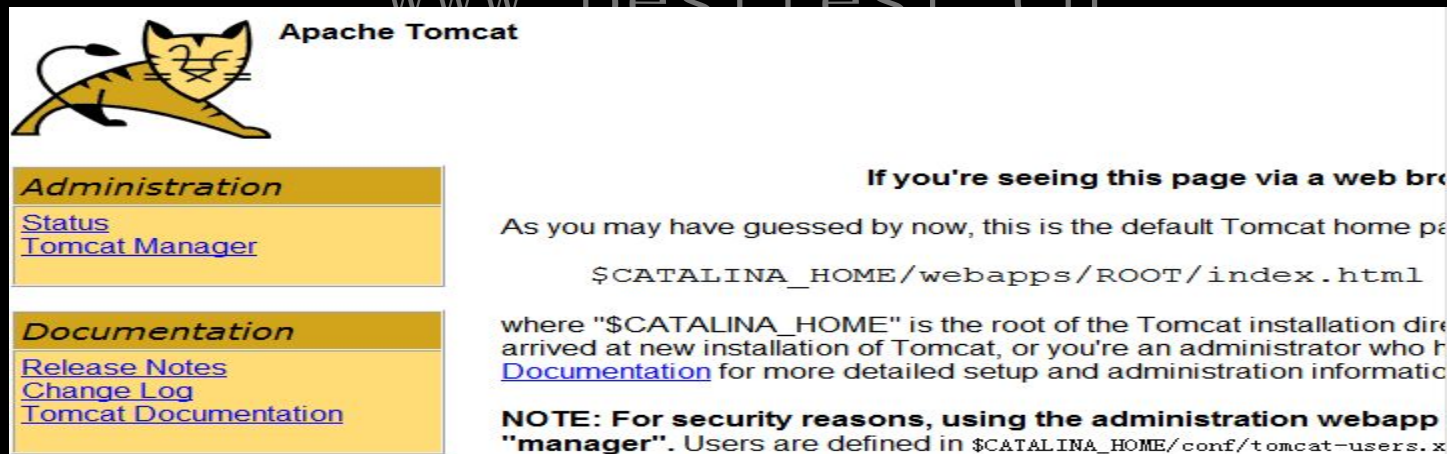
```
<user username="tomcat" password="tomcat"  
roles="tomcat"/>
```

```
<user username="both" password="tomcat"  
roles="tomcat,role1"/>
```

```
<user username="role1" password="tomcat" roles="role1"/>
```


Tomcat后台监控

- 启动Tomcat后（假设%Tomcat_Home%/conf/server.xml中的默认服务端口仍是8080），在浏览器中键入<http://localhost:8080>，出现启动成功画面



Tomcat后台监控

- Status控制台是用于监控Web应用的状态。
- Manager控制台可以部署、监控Web应用。

`www.besttest.cn`
`http://weibo.com/besttest`

Tomcat后台监控

监控台：

Server Status

Manager

List Applications	HTML Manager Help	Manager Help	Complete Server Status
-----------------------------------	-----------------------------------	------------------------------	--

Server Information

Tomcat Version	JVM Version	JVM Vendor	OS Name	OS Version	OS Architecture
Apache Tomcat/6.0.36	1.6.0_02-b06	Sun Microsystems Inc.	Windows Vista	6.1	x86

JVM

Free memory: 1.58 MB Total memory: 6.46 MB Max memory: 63.56 MB

http-8080

Max threads: 200 Current thread count: 6 Current thread busy: 3
Max processing time: 156 ms Processing time: 0.763 s Request count: 19 Error count: 5 Bytes received: 0.00 MB Bytes sent: 0.12 MB

Stage	Time	B Sent	B Recv	Client	VHost	Request
K	4290 ms	?	?	0:0:0:0:0:0:1	?	?
K	1903 ms	?	?	0:0:0:0:0:0:1	?	?
R	?	?	?	?	?	?
R	?	?	?	?	?	?
S	15 ms	0 KB	0 KB	0:0:0:0:0:0:1	localhost	GET /manager/status HTTP/1.1
R	?	?	?	?	?	?

P: Parse and prepare request S: Service F: Finishing R: Ready K: Keepalive

jk-8009

Max threads: 200 Current thread count: 4 Current thread busy: 1
Max processing time: 0 ms Processing time: 0.0 s Request count: 0 Error count: 0 Bytes received: 0.00 MB Bytes sent: 0.00 MB

Stage	Time	B Sent	B Recv	Client	VHost	Request
-------	------	--------	--------	--------	-------	---------

P: Parse and prepare request S: Service F: Finishing R: Ready K: Keepalive

Tomcat后台监控

监控台关注指标以及解释

关注指标↗	监控↗	指标解释↗	功能列↗
Free memory↗	server-status ↗	剩余内存↗	Status-JVM↗
Total memory↗		总内存↗	
Max memory↗		最大内存↗	
Max threads↗		最大线程数↗	Status-HTTP↗
Current thread count↗		当前线程数↗	
Current thread busy↗		当前繁忙线程数↗	
Max processing time↗		最大处理时间↗	
ms Processing time↗		最短处理时间↗	
Request count↗		请求数↗	
Error count↗		错数数↗	
Bytes received↗		接收字节↗	
Bytes sent↗		发送字节↗	

Tomcat调优之集成apache

- 虽然Tomcat也可以作web服务器，但其处理静态html的速度比不上Apache，且其作为web服务器的功能远不如Apache，因此把Apache和Tomcat集成起来，将html和Jsp的功能部分进行明确分工，让Tomcat只处理Jsp部分，其他的由Apache，IIS等web服务器去处理，由此大大提高Tomcat的运行效率。
- **【部署步骤】**
 1. 安装Apache服务器
 2. 部署Tomcat
 3. 将mod_jk.so拷贝到modules目录下面
 4. 修改httpd.conf和mod_jk.conf
- **【适用场景】** 大量使用静态页面的应用系统

Tomcat调优之apache+tomcat集群

- 对于并发要求很高的系统，我们需要采取负载均衡的方式来分担Tomcat服务器的压力。负载均衡实现大概有四种：第一是通过DNS，但只能简单的实现轮流分配，不能处理故障；第二是基于MS IIS, windows 2003 server本身就带了负载均衡服务；第三是硬件方式，通过交换机功能或专门的负载均衡设备来实现；第四种是软件的方式，通过一台负载均衡服务器进行，上面安装软件。使用Apache Httpd Server做负载均衡器，Tomcat集群节点使用Tomcat就可以做到上述第四种方式，这种方式比较灵活，成本相对较低，另外一个很大的优点就是可以根据应用情况和服务器的情况做一些灵活的配置。所以推荐使用Apache+Tomcat集群来实现负载均衡。
采用Tomcat集群可以最大程度的发挥服务器的性能，可以在配置较高的服务器上部署多个Tomcat，也可以在台服务器上分别部署Tomcat，Apache和Tomcat整合的方式还是JK方式。经过验证，系统对大用户量使用的响应方面，Apache+3Tomcat集群 > Apache+2Tomcat集群 > Apache集成Tomcat > 单个Tomcat。并且采用Apache+多Tomcat集群的部署方式时，如果一个Tomcat出现宕机，系统可以继续使用，所以在硬件系统性能足够优越的情况下，需要尽量发挥软件的性能，可以采用增加Tomcat集群的方式。Apache+Tomcat集群的方式使用到得配置文件有httpd.conf、mod_jk.conf、workers.properties。其中mod_jk.conf是对JK信息的配置，包括JK的路径等，workers.properties配置文件是对Tomcat服务器的连接定义文件。Apache需要调整运行参数，这样才能构建一个适合相应网络环境的web服务

Tomcat调优之apache+tomcat集群

- 其中可进行的优化配置如下：
 1. 设置MPM (Multi Processing Modules多道处理模块)。ThreadPerChild, 这个参数用于设置每个进程的线程数, 在Windows环境下默认值是64, 最大值是1920, 建议设置为100-500之间, 服务器性能高的话值大一些, 反之小一些。MaxRequestPerChild表示每个子进程能够处理的最大请求数。这个参数的值更大程度上取决于服务器的内存, 如果内存比较大的话可以设置为很大的参数, 否则设置一个较小的值, 建议值是3000.
 2. 关闭DNS和名字解析 HostnameLookups off
 3. 打开UseCanonicalName模块 UseCanonicalName on
 4. 关闭多余模块 一般来说, 不需要加载的模块有, mod_include.so、mod_autoindex.so、mod_access.so、mod_auth.so.
 5. 打开KeepAlive支持
KeepAlive on, KeepAliveTimeout 15 MaxKeepAliveRequests 1000

根据实际经验, 通过Apache和Tomcat集群的方式提高系统性能的效果十分明显, 这种方式可以最大化的利用硬件资源, 通过多个Tomcat的处理来分担单Tomcat时的压力。

Tomcat调优之apache+tomcat集群

- **【部署步骤】**

1. 安装Apache服务器
2. 部署Tomcat集群，即多个相同的Tomcat。
3. 将mod_jk.so拷贝到modules目录下面
4. 修改httpd.conf、mod_jk.conf和workers.properties

- **【适用场景】** 并发用户量及在线使用用户数量比较高的系统。

Tomcat调优之JVM

- Linux启动参数：

```
export JAVA_OPTS="-server -Xms1400M -Xmx1400M -Xss512k  
-XX:+AggressiveOpts -XX:+UseBiasedLocking -  
XX:PermSize=128M -XX:MaxPermSize=256M -  
XX:+DisableExplicitGC -XX:MaxTenuringThreshold=31 -  
XX:+UseConcMarkSweepGC -XX:+UseParNewGC -  
XX:+CMSParallelRemarkEnabled -  
XX:+UseCMSCompactAtFullCollection -  
XX:LargePageSizeInBytes=128m -  
XX:+UseFastAccessorMethods -  
XX:+UseCMSInitiatingOccupancyOnly -  
Djava.awt.headless=true "
```

- 具体参数解释以及优化见《JVM参数解释以及优化建议》

Tomcat调优之禁用DNS查询

- 当web应用程序向要记录客户端的信息时，它也会记录客户端的IP地址或者通过域名服务器查找机器名转换为IP地址。DNS查询需要占用网络，并且包括可能从很多很远的服务器或者不起作用的服务器上去获取对应的IP的过程，这样会消耗一定的时间。为了消除DNS查询对性能的影响我们可以关闭DNS查询，方式是修改server.xml文件中的enableLookups参数值改为false：enableLookups="false"

Tomcat调优之tomcat容器优化

- 通过应用程序的连接器（Connector）进行性能控制的参数是创建的处理请求的线程数。Tomcat使用线程池加速响应速度来处理请求。在Java中线程是程序运行时的路径，是在一个程序中与其它控制线程无关的、能够独立运行的代码段。它们共享相同的地址空间。多线程帮助程序员写出CPU最大利用率的高效程序，使空闲时间保持最低，从而接受更多的请求，以下以tomcat5为例：
 - maxThreads：Tomcat使用线程来处理接收的每个请求。这个值表示Tomcat可创建的最大线程数。
 - acceptCount：指定当所有可以使用的处理请求的线程数都被使用时，可以放到处理队列中的请求数，超过这个数的请求将不予处理。
 - connectionTimeout：网络连接超时，单位：毫秒。设置为0表示永不超时，这样设置有隐患的。通常可设置为30000毫秒。
 - minSpareThreads：Tomcat初始化时创建的线程数。
 - maxSpareThreads：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。
- 最好的方式是多设置几次并且进行测试，观察响应时间和内存使用情况。在不同的机器、操作系统或虚拟机组合的情况下可能会不同，而且并不是所有人的web站点的流量都是一样的，因此没有一刀切的方案来确定线程数的值。

Tomcat调优之tomcat容器优化

- Tomcat中使用APR库，其实就是在Tomcat中使用JNI的方式来读取文件以及进行网络传输。可以大大提升Tomcat对静态文件的处理性能，同时如果你使用了HTTPS方式传输的话，也可以提升SSL的处理性能。
一般在Windows下，可以直接下载编译好的二进制版本的dll库文件来使Tomcat启用APR，一般建议拷贝库文件tcnative-1.dll到Tomcat的bin目录下。而在Linux下，可以直接解压和安装bin目录下的tomcat_native.tar.gz文件，编译之前要确保apr库已经安装。
如何才能判断Tomcat是否已经启用了APR库呢？方法是通过看Tomcat的启动日志：<http://weibo.com/besttest>
如果没有启用APR，则启动日志一般有这么一条：
org.apache.coyote.http11.Http11Protocol start
如果启用了APR，则这条日志就会变成：
org.apache.coyote.http11.Http11AprProtocol start
tcnative-1.dll 下载地址：<http://tomcat.heanet.ie/native/>

Tomcat调优之tomcat容器优化

- HTTP 压缩可以大大提高浏览网站的速度，它的原理是，在客户端请求网页后，从服务器端将网页文件压缩，再下载到客户端，由客户端的浏览器负责解压缩并浏览。相对于普通的浏览过程HTML,CSS,Javascript,Text，它可以节省40%左右的流量。更为重要的是，它可以对动态生成的，包括CGI, PHP, JSP, ASP, Servlet,SHTML等输出的网页也能进行压缩，压缩效率惊人。
 - 1)compression="on" 打开压缩功能
 - 2)compressionMinSize="2048" 启用压缩的输出内容大小，这里面默认为2KB
 - 3)noCompressionUserAgents="gozilla, traviata" 对于以下的浏览器，不启用压缩
 - 4)compressableMimeType="text/html,text/xml" 压缩类型

Tomcat调优之调优综述

- 调优综述

根据以上分析，如果想要Tomcat达到最优的效果，首先要争取使得操作系统以及网络资源达到最优，并且最好使用高版本的JDK。对于有大量静态页面的系统，采用Apache集成Tomcat的方式，把静态页面交由Apache处理，动态部分交由Tomcat处理，能极大解放Tomcat的处理能力。使用ARP库也能极大的提高Tomcat对静态文件的处理能力。对于并发要求较高的系统，采用Apache加Tomcat集群的方式，将负载分别分担到多个Tomcat上，能很大的提高系统的性能，充分利用硬件资源。同时需要对Tomcat自身进行优化，包括增大内存、调节并发线程数等。

我们的共同目标

迈向成功之路

欢迎加入技术交流群共同探讨测试技术：

性能测试交流群：244735278

纯测试技术交流群：249854093

自动化测试交流群：37753989

安全测试交流群：82280498

软件测试招聘群：202830590

软件测试吐槽群（可闲聊的群）：64779175

自学与分享平台：www.besttest.cn 微博：<http://weibo.com/besttest> 微信订阅号：besttest

感谢您的支持

www.besttest.cn

<http://weibo.com/besttest>



安大叔

官方网站：www.besttest.cn

官方微博：<http://weibo.com/besttest>

微信订阅号：[besttest](#)