**Tomcat(雄猫) : 开源的Web应用服务器**

**背景： 从前，Apache服务器对静态html有很强的处理能力，但是它不支持运行JSP与Servlet， 故扩展开发了Tomcat服务器，并将java的解析任务交给Tomcat服务器，但因为Tomcat 本身也具有处理静态html的能力，并且可以独立运行， 所以现在很多开发者图方便 就让Tomcat 单独承担了整个web项目，事实上较好的处理方式是： 由tomcat解析 并运行java， 由apache处理静态资源**

**开发者：Apache , Sun , 其他公司及个人**

**使用条件：for free(免费)**

**context.xml**

**主要功能：Tomcat公用的环境配置**

**特点：动态加载**

**web.xml**

**主要功能：与webProject中的web.xml类似，只不是级别不同，这里它属于tomcat级别**

**server.xml**

**主要功能：部署工程，其次可以设置端口号，添加虚拟机等，是对服务器的设置**

**maxThreads**：tomcat起动的最大线程数，即同时处理的任务个数，默认值为200

**acceptCount**：当tomcat起动的线程数达到最大时，接受排队的请求个数，默认值为100

**catalina.bat**

**主要功能：设置并检查 目录变量， jvm参数等等**

例：**set JAVA\_OPTS="-server -Xms2048m -Xmx2048m -Xss512k"**

放在catalina.out内容中的一堆rem注释之后

\* -server:一定要作为第一个参数，在多个CPU时性能佳   
\* -Xms：初始Heap大小，使用的最小内存,cpu性能高时此值应设的大一些 物理内存的1/64  
\* -Xmx：java heap最大值，使用的最大内存 物理内存的1/4  
\* -XX:PermSize:设定内存的永久保存区域 0m?  
\* -XX:MaxPermSize:设定最大内存的永久保存区域 4m  
 -XX:MaxNewSize:   
 +XX:AggressiveHeap 会使得 Xms没有意义。这个参数让jvm忽略Xmx参数,疯狂地吃完1 个G物理内存,再吃尽一个G的swap。  
 -Xss：每个线程的Stack大小 512k  
 -verbose:gc 现实垃圾收集信息  
 -Xloggc:gc.log 指定垃圾收集日志文件  
 -Xmn：young generation的heap大小，一般设置为Xmx的1/4  
 -XX:+UseParNewGC ：缩短minor收集的时间  
 -XX:+UseConcMarkSweepGC ：缩短major收集的时间

**压力测试**

**Tomcat6**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **配置** | **threads** | **count** | **connects** | **time** | **connected/time** | **time/connect** |
| **默认配置** | **10** | **100** | **1000** | **0.397** | **2517.84** | **0.0004** |
| **100** | **10** | **1000** | **0.467** | **2141.02** | **0.00047** |
| **100** | **100** | **10000** | **3.732** | **2679.62** | **0.00037** |
| **800** | **100** | **80000** | **3.997** | **2556.54** | **0.00039** |
| **100** | **800** | **80000** | **0.366** |  |  |
| **Ps:如果一个线程持续发起超过401次（count >= 402）请求，无法发起请求** | | | | | |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |