

# Aomi Online 性能测试報告

author: xiaoyulong

Aomi

2019 年 4 月 29 日

# 目录

## 1. 第一章节 Tomcat 配置

### 1.1 第一小节 TomCat 参数

### 1.2 第二小节 connector 参数

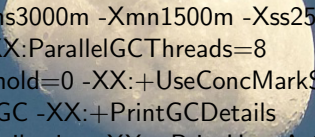
## 2. 第二章节基准测试

## 3. 参考文献

# 第一章 Tomcat

## 第一小节 TomCat 参数

### ▶ 参数配置



```
-server -Xmx3000m -Xms3000m -Xmn1500m -Xss256k  
-XX:SurvivorRatio=6 -XX:ParallelGCThreads=8  
-XX:MaxTenuringThreshold=0 -XX:+UseConcMarkSweepGC  
-verbose:gc -XX:+PrintGC -XX:+PrintGCDetails  
-XX:+PrintTenuringDistribution -XX:+PrintHeapAtGC  
-XX:+PrintGCTimeStamps -XX:+PrintGCDateStamps  
-XX:+HeapDumpOnOutOfMemoryError -Xloggc:../logs/gc.log
```

# 第一章 Tomcat

## 第一节 TomCat 参数解释

`-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

`-Xmx`: java heap最大值,使用的最大内存

上面两个值是分配JVM的最小和最大内存,取决于硬件物理内存的大小,建议均设为物理内存的一半。`-XX:PermSize`:设定内存的永久保存区域

`-XX:MaxPermSize`:设定最大内存的永久保存区域

`-XX:MaxNewSize`:

`-Xss 15120` 这使得JBoss每增加一个线程(thread)就会立即消耗15M内存,而最佳值应该是128K,默认值好像是512k。

`+XX:AggressiveHeap` 会使得 Xms没有意义。这个参数让jvm忽略Xmx参数,疯狂地吃完一个G物理内存,再吃尽一个G的swap。

`-Xss`: 每个线程的Stack大小

`-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3、4分之一

`-XX:+UseParNewGC` : 缩短minor收集的时间

`-XX:+UseConcMarkSweepGC` : 缩短major收集的时间

配置远程jmx监听

`-Dcom.sun.management.jmxremote.port=20086 -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.authenti`

# 第一章 Tomcat

## 第二小节 connector 参数

```
< Connector
port="8080"
protocol="org.apache.coyote.http11.Http11NioProtocol"
enableLookups="false"
maxThreads="1000"
minSpareThreads="100"
acceptCount="900"
disableUploadTimeout="true"
connectionTimeout="20000"
URIEncoding="UTF-8"
redirectPort="8443"
compression="on"
compressionMinSize="1024"
useSendfile="false"
noCompressionUserAgents="gozilla, traviata"
compressibleMimeType="text/html,text/xml,text/plain,text/css,text/javascript,application/javascript" />
```

图: connector 参数demo

# 第一章 Tomcat

## 第二小节 connector 参数说明

### ▶ 参数说明

org.apache.coyote.http11.Http11NioProtocol: 调整工作模式为Nio  
maxThreads: 最大线程数, 默认150。增大值避免队列请求过多, 导致响应缓慢。

minSpareThreads: 最小空闲线程数。

acceptCount: 当处理请求超过此值时, 将后来请求放到队列中等待。

disableUploadTimeout: 禁用上传超时时间

connectionTimeout: 连接超时, 单位毫秒, 0代表不限制

URIEncoding: URI地址编码使用UTF-8

enableLookups: 关闭dns解析, 提高响应时间

compression: 启用压缩功能

compressionMinSize: 最小压缩大小, 单位Byte

compressibleMimeType: 压缩的文件类型

# Tomcat 总体架构图

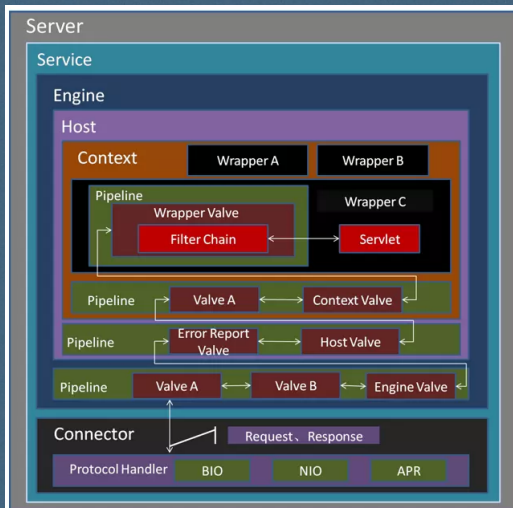


图: Architect

# Tomcat BIO

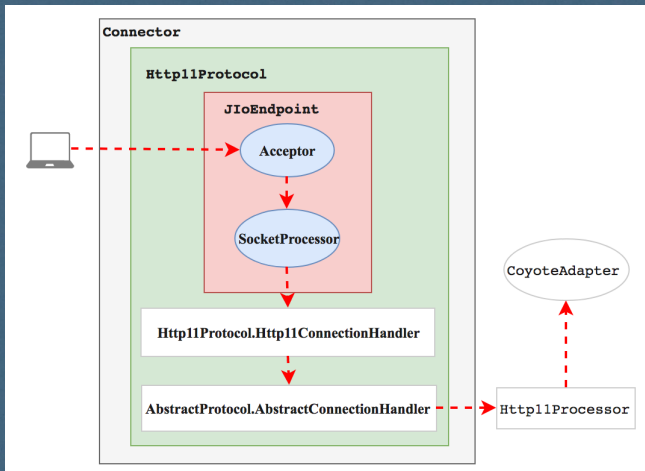


图: http-BIO



# Tomcat NIO

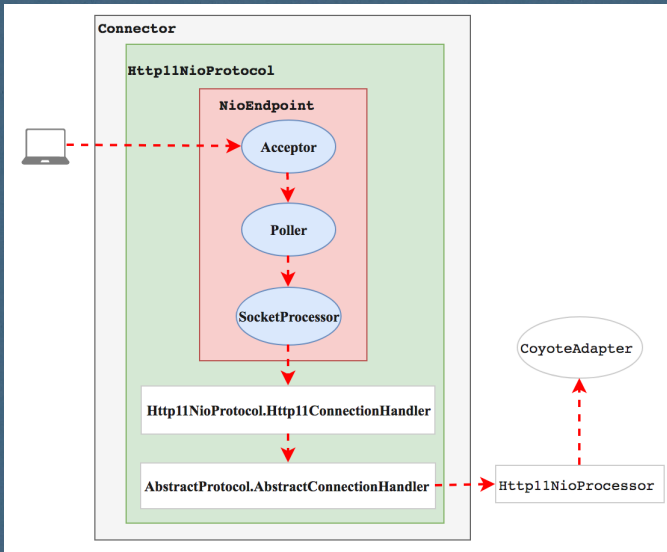


图: Http-NIO

# Tomcat NIO2

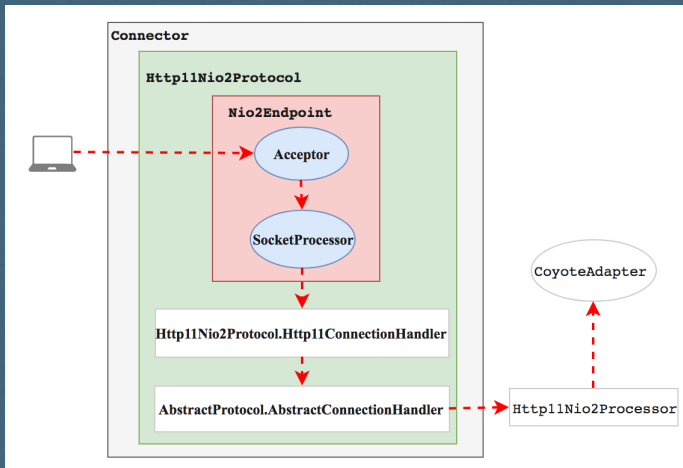


图: http-NIO2

# Tomcat APR

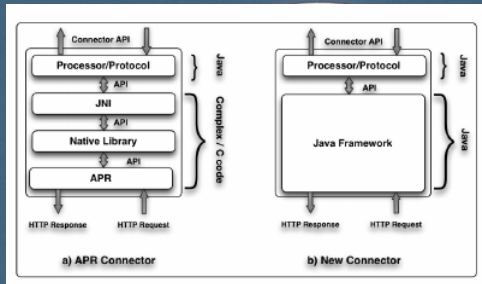


图: http-APR

# 目录

## 1. 第一章节 Tomcat 配置

### 1.1 第一小节 TomCat 参数

### 1.2 第二小节 connector 参数

## 2. 第二章节基准测试

## 3. 参考文献



## Tomcat 基准性能

tomcat bio、nio、apr 软件测试版本（10分钟压测结果）

tomcat默认首页

tomcat 模式	并发数	样本数	平均响应时间(ms)	吞吐量(s)	偏移	错误率(%)	KB/sec
bio	200	620140	114	1033	194	0	11479
bio	300	596220	174	990	505	0.01	10997
bio	500	460704	645	764	1994	0.65	8443
bio	700	440252	947	730	2914	1.65	8002
bio	1000	255481	2350	422	5020	5.87	4471
bio	1300	185352	4227	305	6767	12.56	3049
bio	1500	253813	3583	413	6565	11.23	4173
nio	200	840157	121	1396	68	0	15512
nio	300	805874	60	1340	145	0	14889
nio	500	817107	235	1354	205	0.01	15041
nio	700	817243	265	1356	283	0	15062
nio	1000	736392	262	1218	505	0	13526
nio	1300	738334	250	1217	567	0	13519
nio	1500	726248	252	1200	663	0	13330
apr	200	882166	94	1463	73	0	16247
apr	300	711858	139	1175	268	0	13056
apr	500	701720	155	1158	410	0.03	12862
apr	700	896486	127	1482	297	0	16465
apr	1000	709752	207	1169	689	0	12981
apr	1300	629729	385	1037	1145	0	11518
apr	1500	627767	390	1028	1253	0.01	11419

**图：基准性能图表**

# 目录

## 1. 第一章节 Tomcat 配置

### 1.1 第一小节 TomCat 参数

### 1.2 第二小节 connector 参数

## 2. 第二章节基准测试

## 3. 参考文献



## 参考文献 I

[1] 作者. 标题. 期刊.

[2] Author. Title. *Journal*.

<https://blog.csdn.net/mrleeapple/article/details/80420395>

<https://zh.wikipedia.org/wiki/Apache>




## 参考文献 I

JDK版本	默认GC	性能
JDK8	G1非并发	GC
JDK11	G1并发	GC
openjdk8	G1非并发	GC
openjdk11	G1并发	GC





# JDK版本与GC I



Connector	版本要求	IO
Http11Protocol	7、8、9	BIO
Http11NioProtocol	7、8、9	NIO
Http11Nio2Protocol	8、9	NIO2
Http11AprProtocol	7、8、9	APR

## Q & A

Thank you !

