1. 目的

通过优化 tomcat 提高网站的并发能力。

2. 服务器资源

服务器所能提供 CPU、内存、硬盘的性能对处理能力有决定性影响。

3. 优化配置

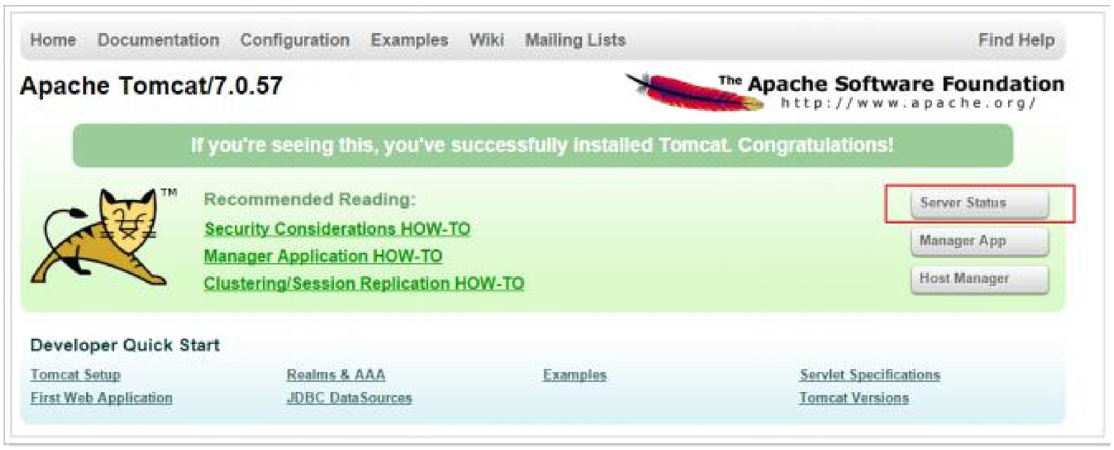
3.1. 配置 tomcat 管理员账户

在 conf/ tomcat-users.xml 下添加用户：

```
<role rolename="manager"/>
<role rolename="manager-gui"/>
<role rolename="admin"/>
<role rolename="admin-gui"/>
<user username="tomcat" password="tomcat"
roles="admin-gui,admin,manager-gui,manager"/>
```

启动 tomcat，登录查看信息：

http://127.0.0.1:8080/



Home Documentation Configuration Examples Wiki Mailing Lists Find Help

Apache Tomcat/7.0.57

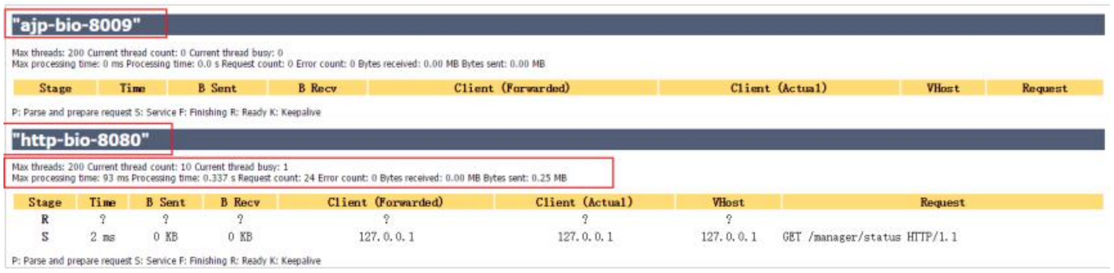
The Apache Software Foundation http://www.apache.org/

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:
Security Considerations HOW-TO
Manager Application HOW-TO
Clustering/Session Replication HOW-TO

Server Status
Manager App
Host Manager

Developer Quick Start
Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Examples
Servlet Specifications
Tomcat Versions



"ajp-bio-8009"

```
Max threads: 200 Current thread count: 0 Current thread busy: 0
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 (Forwarded)	Client (Actual)	VHost	Request

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

"http-bio-8080"

```
Max threads: 200 Current thread count: 10 Current thread busy: 1
Max processing time: 93 ms Processing time: 0.337 s Request count: 24 Error count: 0 Bytes received: 0.00 MB Bytes sent: 0.25 MB
```

Stage	Time	B Sent	B Recv	Client (Forwarded)	Client (Actual)	VHost	Request
R	?	?	?	?	?	?	
S	2 ms	0 KB	0 KB	127.0.0.1	127.0.0.1	127.0.0.1	GET /manager/status HTTP/1.1

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

3.2. tomcat 的 3 种运行模式

tomcat 的运行模式有 3 种：

1、 bio

默认的模式,性能非常低下,没有经过任何优化处理和支持.

2、 nio(new IO)

利用 java 的异步 io 护理技术,no blocking IO 技术.

3、 apr
安装起来最困难,但是从操作系统级别来解决异步的 IO 问题,大幅度的提高性能.

3.2.1. 启动 NIO 模式

修改 server.xml 里的 Connector 节点,修改 protocol 为
org.apache.coyote.http11.Http11NioProtocol

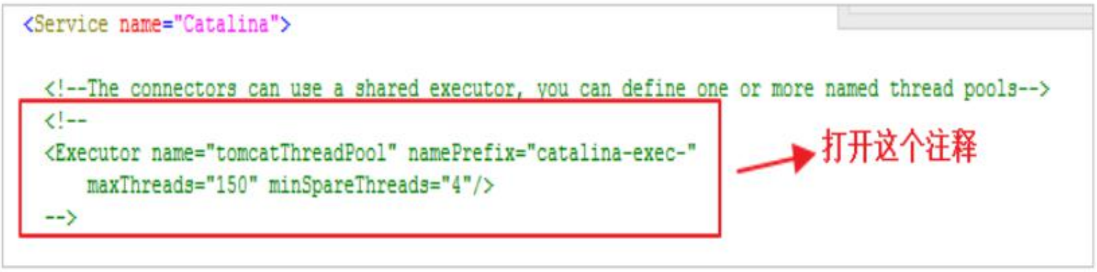
```
<Connector port="8080" protocol="org.apache.coyote.http11.Http11NioProtocol"
           connectionTimeout="20000"
           redirectPort="8443" />
```

3.3. 执行器（线程池）

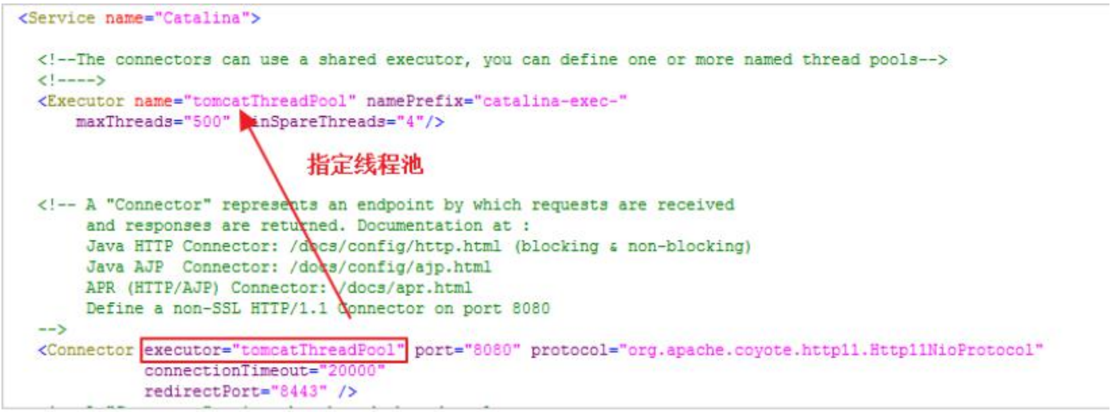
在 tomcat 中每一个用户请求都是一个线程，所以可以使用线程池提高性能。

3.3.1. 开启并且使用

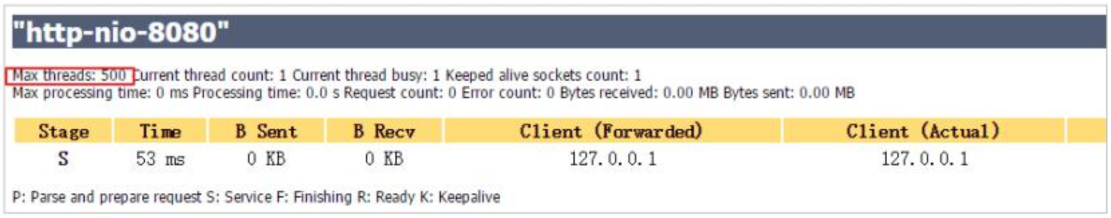
配置



```
<Service name="Catalina">

  <!--The connectors can use a shared executor, you can define one or more named thread pools-->
  <!--
  <Executor name="tomcatThreadPool" namePrefix="catalina-exec-"
      maxThreads="150" minSpareThreads="4"/>
  -->
```



```
<Service name="Catalina">

  <!--The connectors can use a shared executor, you can define one or more named thread pools-->
  <!---->
  <Executor name="tomcatThreadPool" namePrefix="catalina-exec-"
      maxThreads="500" minSpareThreads="4"/>

  <!-- A "Connector" represents an endpoint by which requests are received
       and responses are returned. Documentation at :
       Java HTTP Connector: /docs/config/http.html (blocking & non-blocking)
       Java AJP  Connector: /docs/config/ajp.html
       APR (HTTP/AJP) Connector: /docs/apr.html
       Define a non-SSL HTTP/1.1 Connector on port 8080
  -->
  <Connector executor="tomcatThreadPool" port="8080" protocol="org.apache.coyote.http11.Http11NioProtocol"
             connectionTimeout="20000"
             redirectPort="8443" />
```



"http-nio-8080"

Max threads: 500 Current thread count: 1 Current thread busy: 1 Keeped alive sockets count: 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 (Forwarded)	Client (Actual)
S	53 ms	0 KB	0 KB	127.0.0.1	127.0.0.1

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

3.3.2. 参数说明

Attribute	Description
threadPriority（优先级）	(int) The thread priority for threads in the executor, the default is 5 (the value of theThread.NORM_PRIORITY constant)
daemon（守护进程）	(boolean) Whether the threads should be daemon threads or not, the default is true
namePrefix（名称前缀）	(String) The name prefix for each thread created by the executor. The thread name for an individual thread will be namePrefix+threadNumber
maxThreads（最大线程数）	(int) The max number of active threads in this pool, default is 200
minSpareThreads（最小活跃线程数）	(int) The minimum number of threads always kept alive, default is 25
maxIdleTime（空闲线程等待时间）	(int) The number of milliseconds before an idle thread shutsdown, unless the number of active threads are less or equal to minSpareThreads. Default value is 60000(1 minute)
maxQueueSize(最大的等待队里数，超过则请求拒绝)	(int) The maximum number of runnable tasks that can queue up awaiting execution before we reject them. Default value is Integer.MAX_VALUE
prestartminSpareThreads（是否在启动时就生成minSpareThreads个线程）	(boolean) Whether minSpareThreads should be started when starting the Executor or not, the default is false
threadRenewalDelay（重建线程的时间间隔）	(long) If a ThreadLocalLeakPreventionListener is configured, it will notify this executor about stopped contexts. After a context is stopped, threads in the pool are renewed. To avoid renewing all threads at the same time, this option sets a delay between renewal of any 2 threads. The value is in ms, default value is 1000 ms. If value is negative, threads are not renewed. 。重建线程池内的线程时，为了避免线程同时重建，每隔 threadRenewalDelay（单位： ms ）重建一个线程。默认值为 1000 ，设置为负则不重建

3.3.3. 最佳实践

```
<!--The connectors can use a shared executor, you can define one or more named thread pools-->
<!---->
<Executor name="tomcatThreadPool" namePrefix="catalina-exec-"
    maxThreads="800" minSpareThreads="100" maxQueueSize="100" prestartminSpareThreads="true"/>
```

3.4. 连接器（Connector）

Connector 是 Tomcat 接收请求的入口，每个 Connector 有自己专属的监听端口
Connector 有两种：HTTP Connector 和 AJP Connector

3.4.1. 通用属性（高亮的是重点）

Attribute	Description
allowTrace	A boolean value which can be used to enable or disable the TRACE HTTP method. If not specified, this attribute is set to false. 如果需要服务器能够处理用户的 HAED/TRACE 请求，这个值应该设置为 true， 默认值是 false
asyncTimeout	The default timeout for asynchronous requests in milliseconds. If not specified, this attribute is set to 10000 (10 seconds). 默认超不时候以毫秒为单位的异步恳求。若是没有指定， 该属性被设置为 10000（10 秒）。
enableLookups	Set to `true` if you want calls to `request.getRemoteHost()` to perform DNS lookups in order to return the actual host name of the remote client. Set to `false` to skip the DNS lookup and return the IP address in String form instead (thereby improving performance). By default, DNS lookups are disabled. 若是你想 request.getRemoteHost（） 的调用 履行， 以便返回的长途客户端的实际主机名的 DNS 查询， 则设置为 true。设置为 false 时跳过 DNS 查找，并返回字符串情势的 IP 地址（从而提高性能）。默认景象下，禁用 DNS 查找。
maxHeaderCount	The maximum number of headers in a request that are allowed by the container. A request that contains more headers than the specified limit will be rejected. A value of less than 0 means no limit. If not specified, a default of 100 is used. 容器允许的请求头字段的最大数目。请求中包含比指定的限制更多的头字段将被拒绝。值小于 0 表示没有限制。如果没有指定， 默认设置为 100。
maxParameterCount	The maximum number of parameter and value pairs (GET plus POST) which will be automatically parsed by the container. Parameter and value pairs beyond this limit will be ignored. A value of less than 0 means no limit. If not specified, a default of 10000 is used. Note that `FailedRequestFilter` filtercan be used to reject requests that hit the limit. 将被容器自动解析的最大数量的参数和值对（GET 加上 POST）。参数值对超出此限制将被忽略。值小于 0 表示没有限制。如果没有指定， 默认为 10000。请注意， FailedRequestFilter 过滤器可以用来拒

	绝达到了极限值的请求。
maxPostSize	The maximum size in bytes of the POST which will be handled by the container FORM URL parameter parsing. The limit can be disabled by setting this attribute to a value less than or equal to 0. If not specified, this attribute is set to 2097152 (2 megabytes). 将被容器以 FORM URL 参数形式处理的最大长度（以字节为单位）的 POST。通过设置此属性的值小于或等于 0 可以禁用该限制。如果没有指定，该属性被设置为 2097152（2 兆字节）。
parseBodyMethods	A comma-separated list of HTTP methods for which request bodies will be parsed for request parameters identically to POST. This is useful in RESTful applications that want to support POST-style semantics for PUT requests. Note that any setting other than POST causes Tomcat to behave in a way that goes against the intent of the servlet specification. The HTTP method TRACE is specifically forbidden here in accordance with the HTTP specification. The default is POST 以逗号分隔的 HTTP 方法列表，通过方法列表，等同于 POST 方法，request 正文将被解析成请求参数。这在 RESTful 应用程序要支持以 POST 式的语义解析 PUT 请求中是非常有用的。需要注意的是设置其他值（不是 POST）会导致 Tomcat 的行为违反 servlet 规范的目的。在这里为了符合 HTTP 规范明确禁止 HTTP 方法 TRACE。默认值是 POST
port	The TCP port number on which this **Connector** will create a server socket and await incoming connections. Your operating system will allow only one server application to listen to a particular port number on a particular IP address. If the special value of 0 (zero) is used, then Tomcat will select a free port at random to use for this connector. This is typically only useful in embedded and testing applications. TCP 端口号，连接器利用该端口号将创建一个服务器套接字，并等待传入的连接。你的操作系统将只允许一个服务器应用程序在一个特定的 IP 地址侦听特定的端口号。如果使用特殊值 0(零)，则 Tomcat 将为连接器随机选择一个空闲的端口。这是通常只用在嵌入式和测试应用程序。

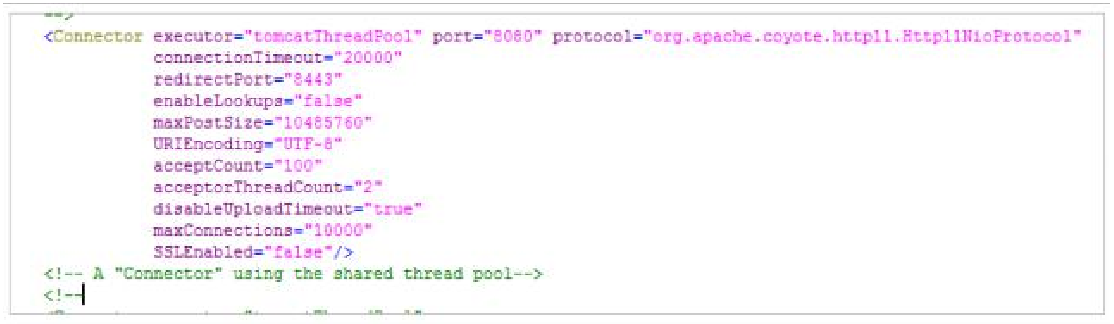
proxyName	If this **Connector** is being used in a proxy configuration, configure this attribute to specify the server name to be returned for calls to `request.getServerName()`. See Proxy Support for more information. 如果这个连接正在使用的代理服务器配置，配置该属性指定的服务器的名称，可以调用 request.getServerName（）返回。有关更多信息，请参见代理支持。
proxyPort	If this **Connector** is being used in a proxy configuration, configure this attribute to specify the server port to be returned for calls to `request.getServerPort()`. See Proxy Support for more information. 如果这个连接正在使用的代理服务器配置，配置该属性指定服务器端口，可以调用 request.getServerPort（）返回。有关更多信息，请参见代理支持。
redirectPort	If this **Connector** is supporting non-SSL requests, and a request is received for which a matching`<security-constraint>` requires SSL transport, Catalina will automatically redirect the request to the port number specified here. 如果该连接器支持非 SSL 请求，并且接收到的请求为满足安全约束需要 SSL 传输， Catalina 将自动将请求重定向到指定的端口号。
scheme	Set this attribute to the name of the protocol you wish to have returned by calls to`request.getScheme()`. For example, you would set this attribute to "`https`" for an SSL Connector. The default value is "`http`". 将该属性设置为你想调用 request.getScheme（）返回的协议的名称。例如，对于 SSL 连接器，你会将此属性设置为“HTTPS ”。默认值是“ HTTP ”。
secure	Set this attribute to `true` if you wish to have calls to `request.isSecure()` to return `true` for requests received by this

NIO 的具体配置

Attribute	Description
pollerThreadCount	(int)The number of threads to be used to run for the polling events. Default value is 1 per processor up to and including version 7.0.27. Default value as of version 7.0.28 is 1 per processor but not more than 2. When accepting a socket, the operating system holds a global lock. So the benefit of going above 2 threads diminishes rapidly. Having more than one thread is for system that need to accept connections very rapidly. However usually just increasing acceptCount will solve that problem. Increasing this value may also be beneficial when a large amount of send file operations are going on. （int）用来处理轮询事件的线程的数量。在版本 7.0.27 及以前版本，默认值是每个处理器 1 个。版本 7.0.28 的默认值是每个处理器 1 个，但不超过 2 个。 当接受一个套接字，操作系统拥有全局的锁。所以超过 2 个线程的好处而迅速减小。有一个以上的线程是因为系统需要非常迅速地接受连接。但通常只要增加 acceptCount 值就可以解决这个问题。增加该值也可能是有用的，当大量发送文件操作发生的时候。
pollerThreadPriority	(int)The priority of the poller threads. The default value is 5 (the value of the java.lang.Thread.NORM_PRIORITY constant). See the JavaDoc for the java.lang.Thread class for more details on what this priority means. （int）轮询线程的优先级。默认值是 5（java.lang.Thread.NORM_PRIORITY 常量值）。优先级的更多详细信息，可以查考 java.lang.Thread 类的 JavaDoc 。
selectorTimeout	(int)The time in milliseconds to timeout on a select() for the poller. This value is important, since connection clean up is done on the same thread, so do not set this value to an extremely high one. The default value is 1000 milliseconds. （int）选择轮询器 select（）的超时时间（以毫秒为单位）。这个值非常重要，因为连接清理工作也是在同一个线程里的，所以不要将此值设置为一个非常高的。默认值是 1000 毫秒。
useComet	(bool)Whether to allow comet servlets or not. Default value is true. （bool）是否允许 Comet servlet。默认值是 true。
useSendfile	(bool)Use this attribute to enable or disable sendfile capability. The default value istrue. （bool）使用此属性来启用或禁用 sendfile 的能力。默认值是 true。

socket.directBuffer	(bool)Boolean value, whether to use direct ByteBuffers or java mapped ByteBuffers. Default is false. When you are using direct buffers, make sure you allocate the appropriate amount of memory for the direct memory space. On Sun's JDK that would be something like -XX:MaxDirectMemorySize=256m. （bool)选择使用直接 ByteBuffers 或 Java 映射的 ByteBuffers。默认是 false。 当您使用直接 ByteBuffers，请确保你分配适当的内存量给直接内存空间。在 Sun 的 JDK 中，配置如-XX：MaxDirectMemorySize＝256M。
socket.appReadBufSize	(int)Each connection that is opened up in Tomcat get associated with a read ByteBuffer. This attribute controls the size of this buffer. By default this read buffer is sized at8192 bytes. For lower concurrency, you can increase this to buffer more data. For an extreme amount of keep alive connections, decrease this number or increase your heap size. （int）在 Tomcat 中每个连接的开辟连接一个读 ByteBuffer。此属性控制这个缓冲区的大小。默认情况下，这个读缓冲区大小为 8192 字节。对于较低的并发，你可以增加这个值以缓冲更多的数据。对于长连接数很多的情况，你需要降低这个数值或者增加堆大小。
socket.eventCache	(int)Tomcat will cache PollerEvent objects to reduce garbage collection. The integer value specifies how many objects to keep in the cache at most. The default is 500. Other values are -1 for unlimited cache and 0 for no cache. （int）以减少垃圾收集，Tomcat 缓存 PollerEvent 对象。该值指定保持在缓存中最多有多少个对象。默认值是 500。-1 表示不限制缓存大小，0 表示不缓存。
selectorPool.maxSelectors	(int)The max selectors to be used in the pool, to reduce selector contention. Use this option when the command line org.apache.tomcat.util.net.NioSelectorShared value is set to false. Default value is 200. （int）以减少选择器的争用，在池中使用的选择器最大个数。命令行 org.apache.tomcat.util.net.NioSelectorShared 值设置为 false 时，使用此选项。默认值是 200。
selectorPool.maxSpareSelectors	(int)The max spare selectors to be used in the pool, to reduce selector contention. When a selector is returned to the pool, the system can decide to keep it or let it be GC'd. Use this option

3.4.4. 最佳实践

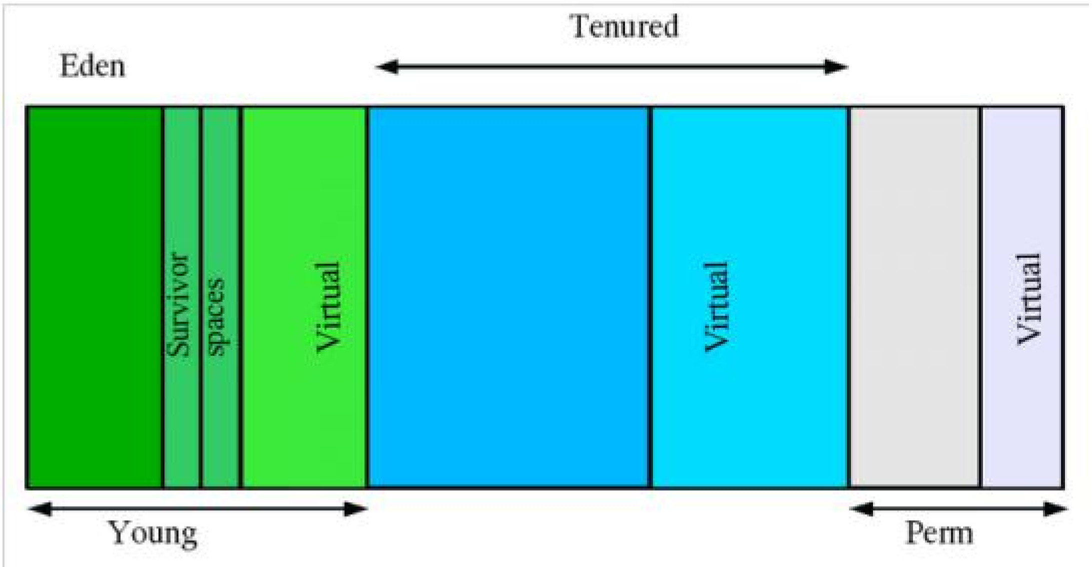


```
<Connector executor="tomcatThreadPool" port="8080" protocol="org.apache.coyote.http11.Http11NioProtocol"
           connectionTimeout="20000"
           redirectPort="8443"
           enableLookups="false"
           maxPostSize="10485760"
           URIEncoding="UTF-8"
           acceptCount="100"
           acceptorThreadCount="2"
           disableUploadTimeout="true"
           maxConnections="10000"
           SSLEnabled="false"/>
<!-- A "Connector" using the shared thread pool-->
<!--
```

5. JVM 参数的优化

适当调整 Tomcat 的运行 JVM 参数可以提升整体性能。

5.1 JVM 内存模型



Java 的内存模型分为

Young（年轻代）

Tenured（终身代）

Perm（永久代）

有如下原因可能导致

Full GC：

1.年老代（Tenured）被写满

2. 持久代（Perm）被写满

3. System.gc()被显示调用

4. 上一次 GC 之后 Heap 的各域分配策略动态变化

5.2. JVM 参数

修改文件：bin/catalina.sh

JAVA_OPTS="-Dfile.encoding=UTF-8 -server -Xms1024m -Xmx1024m -XX:NewSize=512m -XX:MaxNewSize=512m

-XX:PermSize=256m -XX:MaxPermSize=256m -XX:NewRatio=2 -XX:MaxTenuringThreshold=50

-XX:+DisableExplicitGC"
参数说明：
1、 file.encoding 默认文件编码
2、 -Xmx1024m 设置 JVM 最大可用内存为 1024MB
3、 -Xms1024m 设置 JVM 启动时初始分配的内存大小为 1024m。此值可以设置与-Xmx 相同，以避免每次
垃圾回收完成后 JVM 重新分配内存。
4、 -XX:NewSize 设置年轻代大小
5、 XX:MaxNewSize 设置最大的年轻代大小
6、 -XX:PermSize 设置永久代大小------热部署时会出现永久代溢出
7、 -XX:MaxPermSize 设置最大永久代大小
8、 -XX:NewRatio=4:设置年轻代（包括 Eden 和两个 Survivor 区）与终身代的比值（除去永久代）。设置为 4，
则年轻代与老年代所占比值为 1：4，年轻代占整个堆栈的 1/5
9、 -XX:MaxTenuringThreshold=0：设置垃圾最大年龄，默认为：15。如果设置为 0 的话，则年轻代对象不经
过 Survivor 区，直接进入年老代。对于年老代比较多的应用，可以提高效率。如果将此值设置为一个较
大值，则年轻代对象会在 Survivor 区进行多次复制，这样可以增加对象再年轻代的存活时间，增加在年
轻代即被回收的概论。
10、 -XX:+DisableExplicitGC 这个将会忽略手动调用 GC 的代码使得 System.gc()的调用就会变成一个空调
用，完全不会触发任何 GC
11、 补充：---- gc 的日志的设置

5.3. 在 tomcat 中设置 JVM 参数

5.3.1. windows

修改 bin/catalina.bat 文件设置参数（第一行）

```
set JAVA_OPTS=-Dfile.encoding=UTF-8 -server -Xms1024m -Xmx2048m -XX:NewSize=512m
-XX:MaxNewSize=1024m -XX:PermSize=256m -XX:MaxPermSize=256m
-XX:MaxTenuringThreshold=10
-XX:NewRatio=2 -XX:+DisableExplicitGC
```

5.3.2. linux

修改 bin/catalina.sh 文件参数（第一行）

```
JAVA_OPTS="-Dfile.encoding=UTF-8 -server -Xms1024m -Xmx2048m -XX:NewSize=512m
-XX:MaxNewSize=1024m
-XX:PermSize=256m -XX:MaxPermSize=256m -XX:MaxTenuringThreshold=10 -XX:NewRatio=2
-XX:+DisableExplicitGC"
```