**Tomcat 线程模型源码分析**

**课程概要：**

* Tomcat 支持四种线程模型介绍
* Tomcat BIO、NIO实现过程源码解析
* Tomcat connector 并发参数解读

## **Tomcat 支持四种线程模型介绍**

**各IO 简单说明:(20分钟)**

|  |  |
| --- | --- |
|  | 描述 |
| BIO | 阻塞式IO，即Tomcat使用传统的java.io进行操作。该模式下每个请求都会创建一个线程，对性能开销大，不适合高并发场景。优点是稳定，适合连接数目小且固定架构。 |
| NIO | 非阻塞式IO，jdk1.4 之后实现的新IO。该模式基于多路复用选择器监测连接状态在通知线程处理，从而达到非阻塞的目的。比传统BIO能更好的支持并发性能。Tomcat 8.0之后默认采用该模式 |
| APR | 全称是 Apache Portable Runtime/Apache可移植运行库)，是Apache HTTP服务器的支持库。可以简单地理解为，Tomcat将以JNI的形式调用Apache HTTP服务器的核心动态链接库来处理文件读取或网络传输操作。使用需要编译安装APR 库 |
| AIO | 异步非阻塞式IO，jdk1.7后之支持 。与nio不同在于不需要多路复用选择器，而是请求处理线程执行完程进行回调调知，已继续执行后续操作。Tomcat 8之后支持。 |

**使用指定IO模型的配置方式:**

配置 server.xml 文件当中的 <Connector protocol="HTTP/1.1"> 修改即可。

默认配置 8.0 protocol=“HTTP/1.1” 8.0 之前是 BIO 8.0 之后是NIO

**BIO**

protocol=“org.apache.coyote.http11.Http11Protocol“

**NIO**

protocol=”org.apache.coyote.http11.Http11NioProtocol“

**AIO**

protocol=”org.apache.coyote.http11.Http11Nio2Protocol“

**APR**

protocol=”org.apache.coyote.http11.Http11AprProtocol“

## **Tomcat BIO、NIO实现过程源码解析**

**提问：**

BIO 与NIO有什么区别？

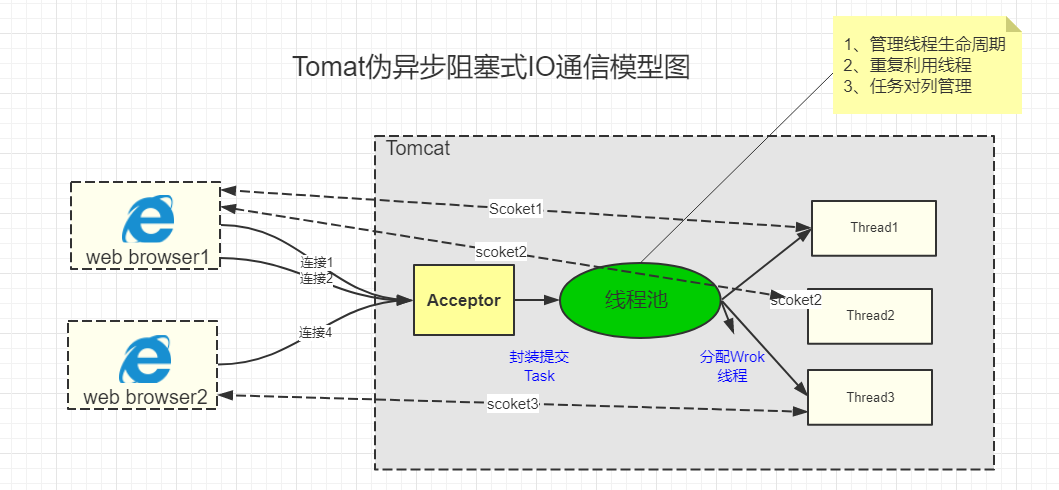
**分别演示在高并发场景下BIO与NIO的线程数的变化？(15分钟)**

**演示数据：**

|  |  |  |
| --- | --- | --- |
|  | 每秒提交数 | 执行线程 |
| BIO | 200 | 300 |
| NIO | 200 | 50 |

**结论：**

**BIO 线程模型讲解(5分钟)**



**NIO 线程模型讲解(5分钟)**

**BIO 源码解读**

**线程组：**

Accept 线程组 acceptorThreadCount=

exec 线程组 maxThread

JIoEndpoint

Acceptor extends Runnable

SocketProcessor extends Runnable

## **Tomcat connector 并发参数解读**

|  |  |
| --- | --- |
|  |  |
| acceptCount | 等待最大队列 |
| address | 绑定客户端特定地址，127.0.0.1 |
| bufferSize | 每个请求的缓冲区大小。 bufferSize \* maxThreads |
| compression | 是否启用文档 |
| compressableMimeTypes | text/html,text/xml,text/plain |
| connectionTimeout | 客户发起链接 到 服务端接收为止，中间最大的等待时间 |
| connectionUploadTimeout | upload 情况下连接超时时间 |
| disableUploadTimeout | true 则使用connectionTimeout |
| enableLookups | 禁用DNS查询 true |
| keepAliveTimeout | 当长链接闲置 指定时间主动关闭 链接 ，前提是客户端请求头 带上这个 head"connection" " keep-alive" |
| maxKeepAliveRequests | 最大的 长连接数 |
| maxHttpHeaderSize |  |
| maxSpareThreads | BIO 模式下 最多线闲置线程数 |
| maxThreads | 最大执行线程数 |
| minSpareThreads | BIO 模式下 最小线闲置线程数 |