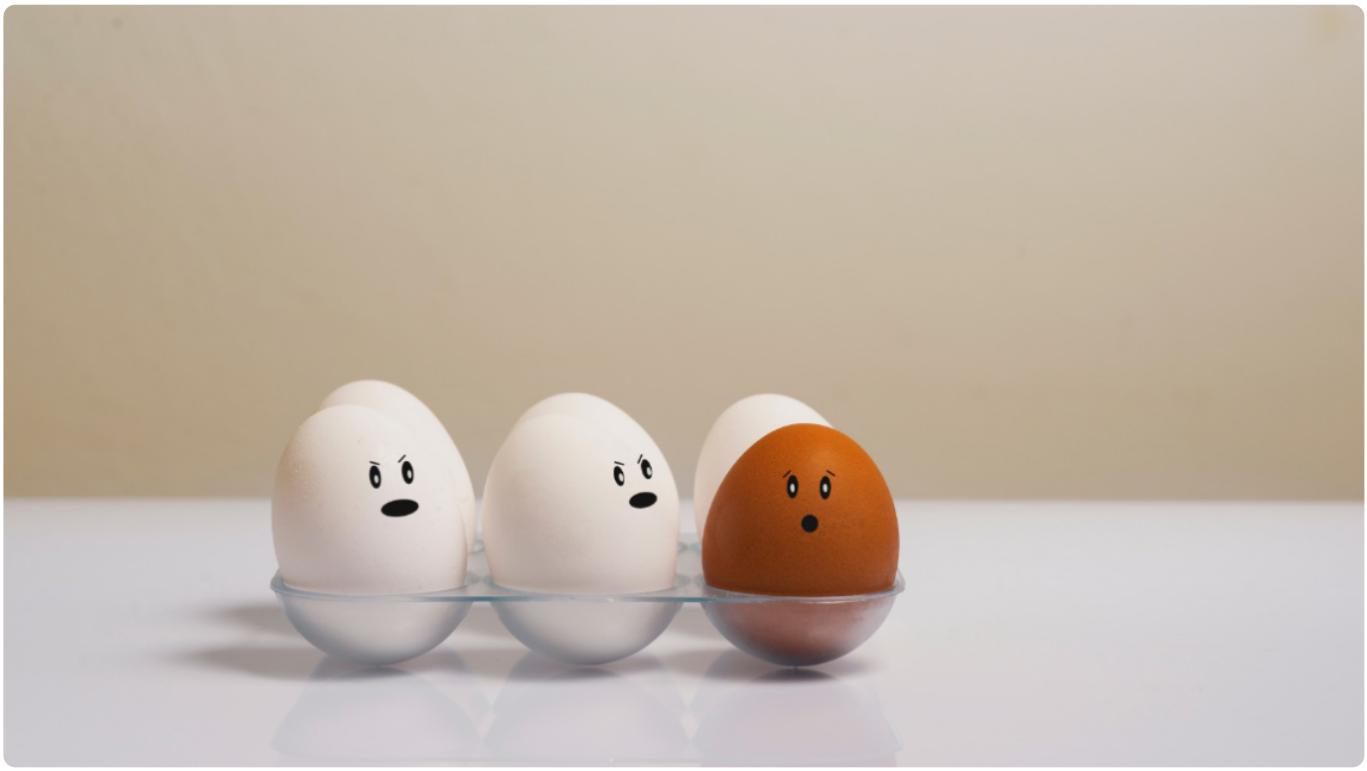
41 | 热点问题答疑（4）： Tomcat和Jetty有哪些不同？

2019-08-15 李号双



讲述：李号双
时长 04:35 大小 4.22M

作为专栏最后一个模块的答疑文章，我想是时候总结一下 Tomcat 和 Jetty 的区别了。专栏里也有同学给我留言，询问有关 Tomcat 和 Jetty 在系统选型时需要考虑的地方，今天我也会通过一个实战案例来比较一下 Tomcat 和 Jetty 在实际场景下的表现，帮你在做选型时有更深的理解。

我先来概括一下 Tomcat 和 Jetty 两者最大的区别。大体来说，Tomcat 的核心竞争力是**成熟稳定**，因为它经过了多年的市场考验，应用也相当广泛，对于比较复杂的企业级应用支持得更加全面。也因为如此，Tomcat 在整体结构上比 Jetty 更加复杂，功能扩展方面可能不如 Jetty 那么方便。

而 Jetty 比较年轻，设计上更加**简洁小巧**，配置也比较简单，功能也支持方便地扩展和裁剪，比如我们可以把 Jetty 的 SessionHandler 去掉，以节省内存资源，因此 Jetty 还可以

运行在小型的嵌入式设备中，比如手机和机顶盒。当然，我们也可以自己开发一个 Handler，加入 Handler 链中用来扩展 Jetty 的功能。值得一提的是，Hadoop 和 Solr 都嵌入了 Jetty 作为 Web 服务器。

从设计的角度来看，Tomcat 的架构基于一种多级容器的模式，这些容器组件具有父子关系，所有组件依附于这个骨架，而且这个骨架是不变的，我们在扩展 Tomcat 的功能时也需要基于这个骨架，因此 Tomcat 在设计上相对来说比较复杂。当然 Tomcat 也提供了较好的扩展机制，比如我们可以自定义一个 Valve，但相对来说学习成本还是比较大的。而 Jetty 采用 Handler 责任链模式。由于 Handler 之间的关系比较松散，Jetty 提供 HandlerCollection 可以帮助开发者方便地构建一个 Handler 链，同时也提供了 ScopeHandler 帮助开发者控制 Handler 链的访问顺序。关于这部分内容，你可以回忆一下专栏里讲的回溯方式的责任链模式。

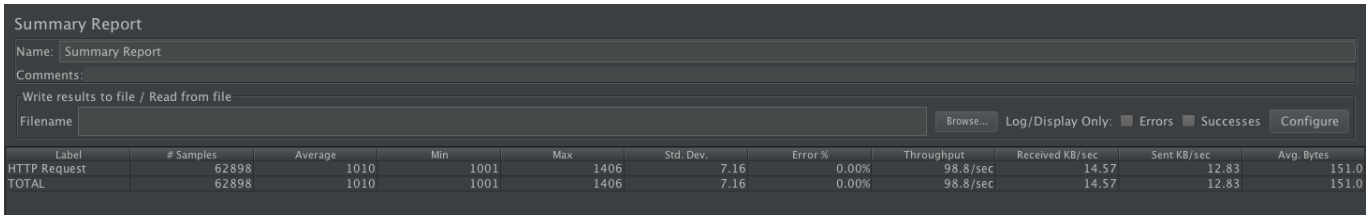
说了一堆理论，你可能觉得还是有点抽象，接下来我们通过一个实例，来压测一下 Tomcat 和 Jetty，看看在同等流量压力下，Tomcat 和 Jetty 分别表现如何。需要说明的是，通常我们从吞吐量、延迟和错误率这三个方面来比较结果。

测试的计划是这样的，我们还是用专栏第 36 期中的 Spring Boot 应用程序。首先用 Spring Boot 默认的 Tomcat 作为内嵌式 Web 容器，经过一轮压测后，将内嵌式的 Web 容器换成 Jetty，再做一轮测试，然后比较结果。为了方便观察各种指标，我在本地开发机器上做这个实验。

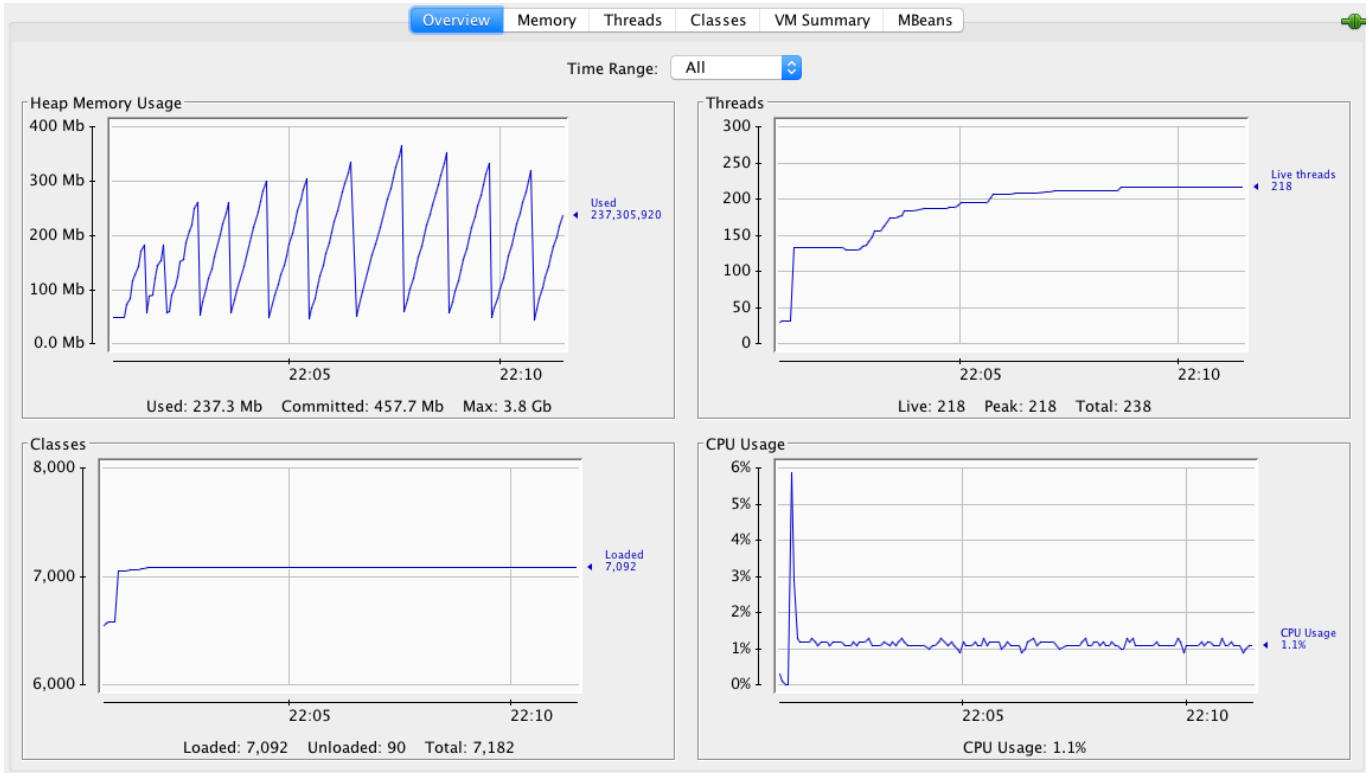
我们会在每个请求的处理过程中休眠 1 秒，适当地模拟 Web 应用的 I/O 等待时间。JMeter 客户端的线程数为 100，压测持续 10 分钟。在 JMeter 中创建一个 Summary Report，在这个页面上，可以看到各种统计指标。

第一步，压测 Tomcat。启动 Spring Boot 程序和 JMeter，持续 10 分钟，以下是测试结果，结果分为两部分：

吞吐量、延迟和错误率



资源使用情况



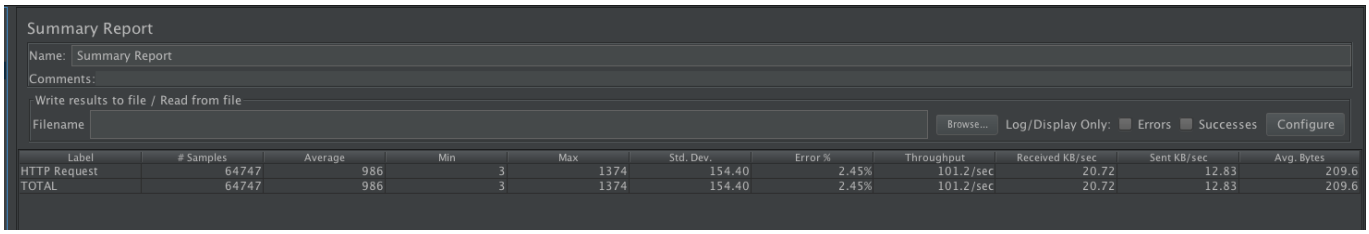
第二步，我们将 Spring Boot 的 Web 容器替换成 Jetty，具体步骤是在 pom.xml 文件中的 spring-boot-starter-web 依赖修改下面这样：

复制代码

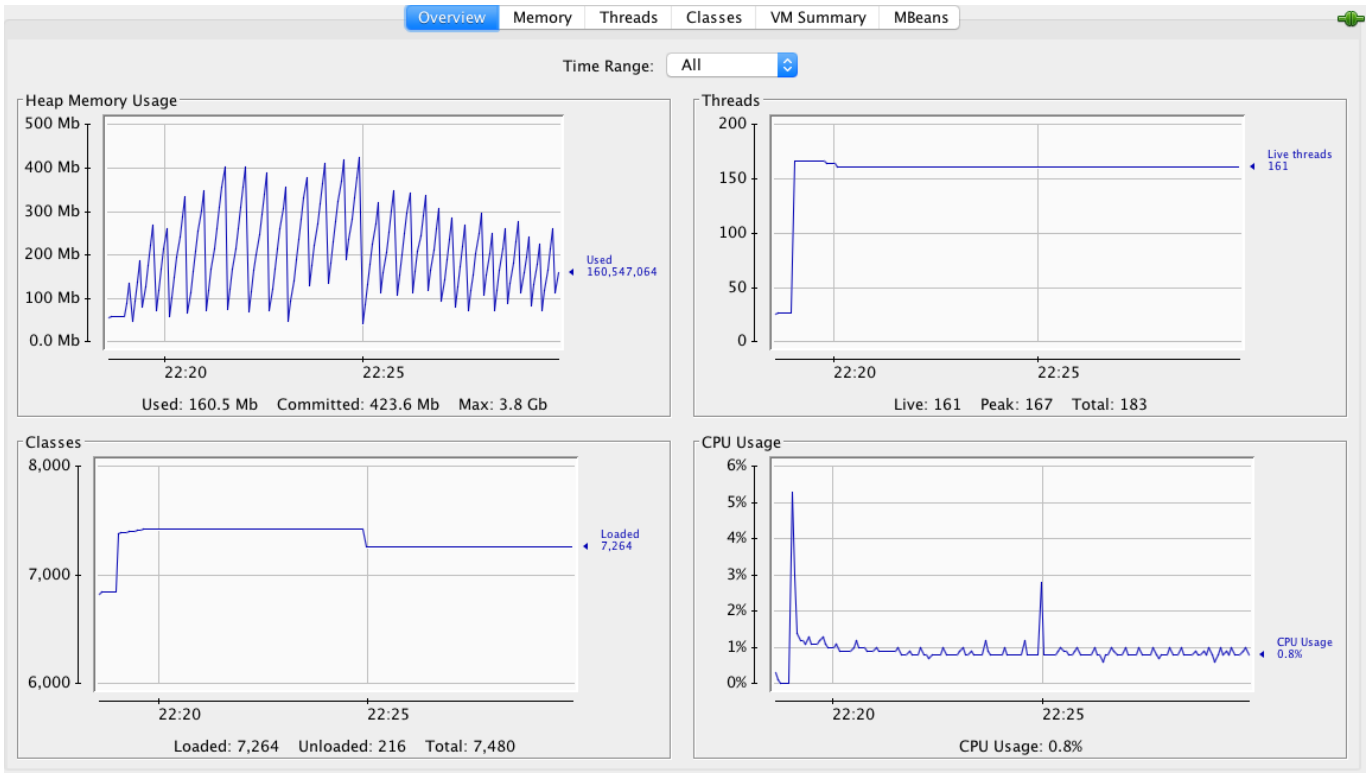
```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
    <exclusions>
        <exclusion>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-tomcat</artifactId>
        </exclusion>
    </exclusions>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-jetty</artifactId>
</dependency>
```

编译打包，启动 Spring Boot，再启动 JMeter 压测，以下是测试结果：

吞吐量、延迟和错误率



资源使用情况



下面我们通过一个表格来对比 Tomcat 和 Jetty：

	吞吐量 s	平均延迟 ms	错误率 %	线程数	CPU %	内存 MB	Class加载量
Tomcat	98.8	1010	0	218	0.8	约200	7092
Jetty	101.2	986	2.45	161	1.1	约160	7264

从表格中的数据我们可以看到：

Jetty 在吞吐量和响应速度方面稍有优势，并且 Jetty 消耗的线程和内存资源明显比 Tomcat 要少，这也恰好说明了 Jetty 在设计上更加小巧和轻量级的特点。

但是 Jetty 有 2.45% 的错误率，而 Tomcat 没有任何错误，并且我经过多次测试都是这个结果。因此我们可以认为 Tomcat 比 Jetty 更加成熟和稳定。

当然由于测试场景的限制，以上数据并不能完全反映 Tomcat 和 Jetty 的真实能力。但是它可以在我们做选型的时候提供一些参考：如果系统的目标是资源消耗尽量少，并且对稳定性要求没有那么高，可以选择轻量级的 Jetty；如果你的系统是比较关键的企业级应用，建议还是选择 Tomcat 比较稳妥。

最后用一句话总结 Tomcat 和 Jetty 的区别：**Tomcat 好比是一位工作多年比较成熟的工程师，轻易不会出错、不会掉链子，但是他有自己的想法，不会轻易做出改变。而 Jetty 更像是一位年轻的后起之秀，脑子转得很快，可塑性也很强，但有时候也会犯一点小错误。**

不知道今天的内容你消化得如何？如果还有疑问，请大胆的在留言区提问，也欢迎你把你的课后思考和心得记录下来，与我和其他同学一起讨论。如果你觉得今天有所收获，欢迎你把它分享给你的朋友。

上一篇　40 | 谈谈Jetty性能调优的思路

下一篇　结束语 | 静下心来，品味经典

精选留言 (3)

写留言

chun1123
2019-08-15

同样意犹未尽！

展开

1

许童童
2019-08-15

这篇文章有点短，没学够的感觉。

展开

1

QQ怪
2019-08-15

学的意犹未尽

展开

1