

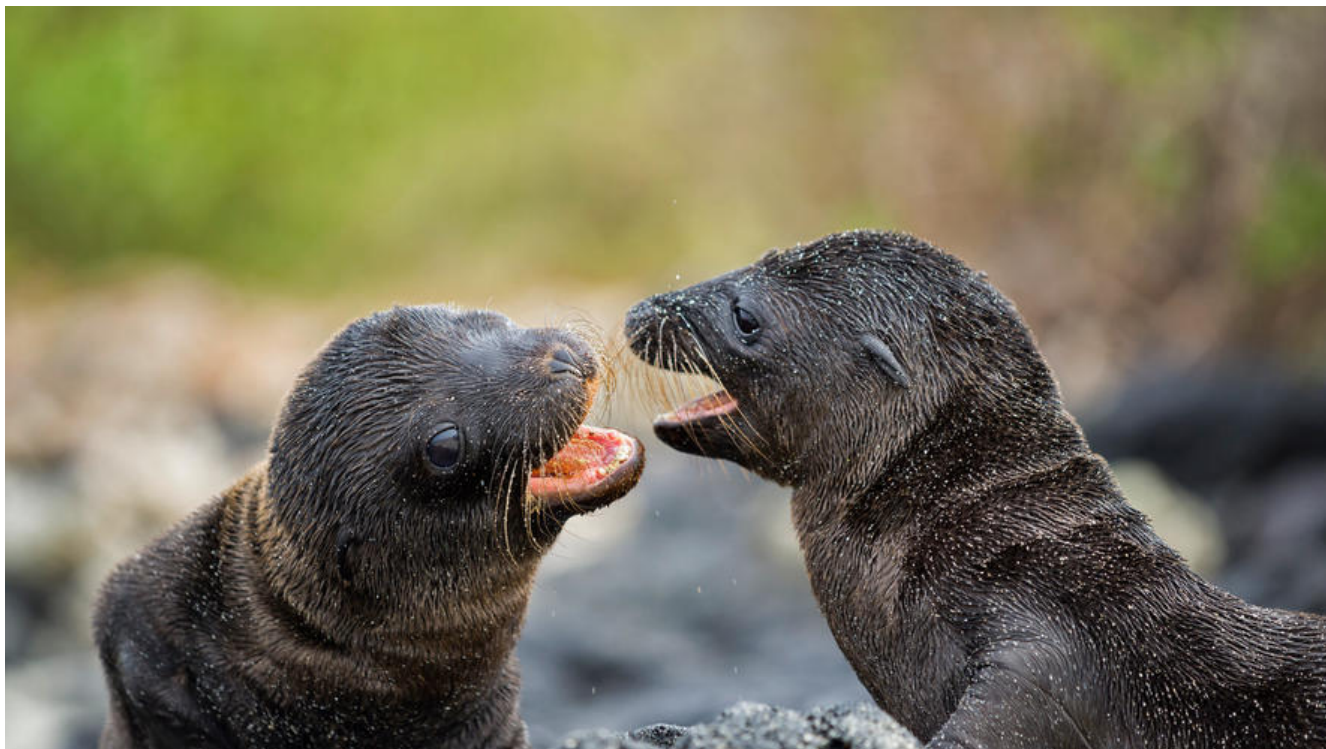
Tomcat、Jetty、Undertow、Netty 等容器的区别

作者: [sumoonyoko](#)

原文链接: <https://ld246.com/article/1575252759539>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



- undertow, jetty 和 tomcat 可以说是 javaweb 项目当下最火的三款服务器，tomcat 是 apache 下的一重量级的服务器，不用多说历史悠久，经得起实践的考验。
- 然而：当下微服务兴起，spring boot，spring cloud 越来越热的情况下，选择一款轻量级而性能越的服务器是必要的选择。spring boot 完美集成了 tomcat，jetty 和 undertow，本文将通过对 jetty undertow 服务器的分析以及测试，来比较两款服务器的性能如何。
- 值得一提的是 jetty 和 undertow 都是基于 NIO 实现的高并发轻量级的服务器，支持 servlet3.1 和 websocket。

压测记录：

容器	负载 (min、max、avg)	QPS	处理时间 (max、avg)	io	gc
tomcat	4.4、6.7、5.5	3000 ~ 3100	1100、65	2000kb	正常
undertow	3.5、5.9、4.5	3300 ~ 3400	404、48	2000kb	正常
jetty	4.5、6.8、5.8	3000 ~ 3100	1200、62	2000kb	正常

因此：**Springboot 使用轻量级的 undertow 容器替代重量级 tomcat，支持 HTTP2.0，性能更好稳定性更好**

- pom.xml 配置如下：

```
<!-- 添加spring-boot-starter-web，默认使用tomcat作为web容器 -->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
  <exclusions>
    <exclusion>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-logging</artifactId>
    </exclusion>
  </exclusions>
</dependency>
```

```
<exclusion>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-tomcat</artifactId>
</exclusion>
</exclusions>
</dependency>
<!-- 去除tomcat, 将undertow作为容器 -->
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-undertow</artifactId>
</dependency>
```

- application.properties配置如下

server:

undertow:

设置IO线程数, 它主要执行非阻塞的任务, 它们会负责多个连接, 默认设置每个CPU核心一个线程

不要设置过大, 如果过大, 启动项目会报错: 打开文件数过多

io-threads: 8

阻塞任务线程池, 当执行类似servlet请求阻塞IO操作, undertow会从这个线程池中取得线程

它的值设置取决于系统线程执行任务的阻塞系数, 默认值是IO线程数*8

worker-threads: 64

以下的配置会影响buffer, 这些buffer会用于服务器连接的IO操作, 有点类似netty的池化内存管理

每块buffer的空间大小, 越小的空间被利用越充分, 不要设置太大, 以免影响其他应用, 合适即可

buffer-size: 1024

是否分配的直接内存(NIO直接分配的堆外内存)

direct-buffers: true