Jetty与tomcat的区别

Jetty和Tomcat为目前全球范围内最著名的两款开源的webserver/servlet容器。

由于它们的实现都遵循Java Servlet规范，一个java Web应用部署于两款容器的任意一个皆可。

但选择哪个更优？也许这得看场景。

近期很多人关注Jetty，也许主要是因为GAE放弃了Tomcat而选择了Jetty。于是，以很直接的想法，Jetty更符合GAE的需求、即云环境的需求，亦分布式环境的需求。

那Jetty与Tomcat比较，有哪差异呢？

## Jetty更轻量级。这是相对Tomcat而言的。

由于Tomcat除了遵循Java Servlet规范之外，自身还扩展了大量JEE特性以满足企业级应用的需求，所以Tomcat是较重量级的，而且配置较Jetty亦复杂许多。但对于大量普通互联网应用而言，并不需要用到Tomcat其他高级特性，所以在这种情况下，使用Tomcat是很浪费资源的。这种劣势放在分布式环境下，更是明显。换成Jetty，每个应用服务器省下那几兆内存，对于大的分布式环境则是节省大量资源。而且，Jetty的轻量级也使其在处理高并发细粒度请求的场景下显得更快速高效。

## jetty更灵活，体现在其可插拔性和可扩展性，更易于开发者对Jetty本身进行二次开发，定制一个适合自身需求的Web Server。相比之下，重量级的Tomcat原本便支持过多特性，要对其瘦身的成本远大于丰富Jetty的成本。用自己的理解，即增肥容易减肥难。

## 然而，当支持大规模企业级应用时，Jetty也许便需要扩展，在这场景下Tomcat便是更优的。

总结：

**Jetty更满足公有云的分布式环境的需求，而Tomcat更符合企业级环境。**

--------------------------------------------------------------------------------

GAE放弃了Tomcat，选择了Jetty，正是因为Jetty的体积和灵活性，Google可以更好地定制一个足够小的Java Web Server为其GAE服务。

而Tomcat为满足更多的企业级需求，增加了JEE特性，在服务企业级应用时，它的支持优于Jetty。然而，即使Tomcat性能略优于Jetty，但对于大多非企业级应用而言，配置复杂体积庞大的Tomcat显得过于重量级。

正因为这个，实验室的云平台实现便是把

云平台本身的门户网站放在Tomcat内，

而云台托管的Java Web应该是部署在Jetty内的。

Jetty和tomcat的比较

相同点：

1.Tomcat和Jetty都是一种Servlet引擎，他们都支持标准的servlet规范和JavaEE的规范。

不同点：

## 架构比较

Jetty的架构比Tomcat的更为简单

Jetty的架构是基于Handler来实现的，主要的扩展功能都可以用Handler来实现，扩展简单。

Tomcat的架构是基于容器设计的，进行扩展是需要了解Tomcat的整体设计结构，不易扩展。

## 性能比较

Jetty和Tomcat性能方面差异不大

Jetty可以同时处理大量连接而且可以长时间保持连接，适合于web聊天应用等等。

Jetty的架构简单，因此作为服务器，Jetty可以按需加载组件，减少不需要的组件，减少了服务器内存开销，从而提高服务器性能。

Jetty默认采用NIO结束在处理I/O请求上更占优势，在处理静态资源时，性能较高

## 少数非常繁忙;Tomcat适合处理少数非常繁忙的链接，也就是说链接生命周期短的话，Tomcat的总体性能更高。

Tomcat默认采用BIO处理I/O请求，在处理静态资源时，性能较差。

## 其它比较

Jetty的应用更加快速，修改简单，对新的Servlet规范的支持较好。

Tomcat目前应用比较广泛，对JavaEE和Servlet的支持更加全面，很多特性会直接集成进来。