通信和网络，2010年，2，200-206。

doi：10.4236 / cn.2010.23029 2010年5月在线发布（http://www.SciRP.org/journal/cn）。

Java Web服务的性能评估：开发人员的角度。

Sanjay Ahuja，Je-Loon Yang。

美国佛罗里达州杰克逊维尔市北佛罗里达大学计算机学院。

电子邮件：{sahuja，n00168553}@unf.edu。

2010年3月7日收到; 2010年7月2日修订;接受2010年7月3日。

抽象。

随着互联网人口的迅速增长，网络技术的发展变得非常重要。对于Web 2.0的发展，Web服务至关重要。Web服务是一种程序，它允许Web上不同平台的计算机以交互方式进行通信，而无需为人类阅读界面和诸如网页结构等格式提供额外数据。由于Web服务是互联网发展的未来趋势，因此用于开发的工具也很重要。虽然Web服务框架有多种选择可供选择，但开发人员应根据框架的性能，时间和精力选择最适合其应用的框架。在这个项目中，我们选择了四个通用框架来比较定性和定量指标。运行测试后，结果由SAS进行统计分析。

关键词：Web服务，框架，性能，Java，开发人员。

1.介绍。

对于前往其他国家或国家的旅行，该人通常需要购买机票，租车，并预订酒店留下。在处理飞机票时，大多数时候人们甚至不得不为一些站点购买几张票，而不是将票直接带到最终目的地的票。查看飞机的到达和离开时间以连接每个航班可能需要数小时的搜索和计划。如果有一个虚拟代理可以在几秒钟内完成这件事情会怎么样？所以通常人们会寻找代理人为他们做这件事。但是如果这个代理实际上是在线虚拟代理呢？如果该人员刚刚输入他想从哪里开始的位置，目的地，期望的出发或到达时间以及计算机所需的所有信息，则计算机将显示所有结果供人选择，并且购买门票。更好的是，这些虚拟代理商还可以显示目的地的汽车租赁和酒店信息，并为您预留它们。通过使用这种类型的虚拟代理，可以节省许多精力和时间，并且也可以比人工代理更准确。这种技术依赖于Web服务的发展和普及。

没有从头开始开发Web服务应用程序，有许多开源框架可以使开发更容易。哪个框架将成为Web服务应用程序开发的更好选择？这项研究通过做几次测试和分析比较了四种流行的开源框架的定性和定量。这四个框架是Apache Axis，JBossWS，XFire和Hessian。第2部分介绍了更多Web服务的介绍。第3部分介绍了本研究中使用的四个框架。在第4节中，将更详细地解释用于度量框架的度量标准。第5节介绍用于分析测量结果的统计分析方法。在第6节中，显示和分析了测试结果。结论在第7节。

2. Web服务框架。

由于Web服务旨在以常用方式传输数据，因此为了方便Web服务开发人员，一些公司和团队开发了Web服务框架，以便他们无需从头开始编写完整的Web服务。一些流行的框架是Apache Axis，JBossWS，Codehaus XFire和Caucho Hessian。在本节中，将介绍这些框架。

2.1。 Apache Axis。

Apache Axis（Apache可扩展交互系统）是Apache Software Foundation（ASF）创建的基于Java和XML的开源Web服务框架。该基金会是一家非营利性公司，主要生产网络使用软件，例如服务器和服务器框架。他们的项目众所周知是协作性的，基于共识的开发流程和免费或开源软件。 Apache Axis包具有用于生成和部署Web服务应用程序的SOAP服务器和API的实现。 SOAP引擎构造SOAP处理器，如客户端，服务器和网关。这允许服务器和客户端通过SOAP消息进行通信。该API支持多种语言。除了Java版本之外，C ++实现也是可用的。它允许开发人员以各种方式构建他们的应用程序。最简单的方法只需要将文件扩展名从“.java”更改为“.jws”。这种方法的缺点是缺乏进一步配置的灵活性。

2.2。将JBossWS。

JBossWS是JBoss的J2EE兼容Web服务的实现。该框架旨在更好地适应整体JBoss体系结构，并且通常更适合Web服务的特定J2EE要求。 JBoss没有为这个框架使用传统的Apache服务器，而是拥有自己的服务器，并建议在该服务器上使用该框架以获得最佳性能。与ASF类似，JBoss社区是一群专注于开源项目的人。他们的项目强调Java Enterprise Middleware的开发，这些软件在应用程序，操作系统或两者之间起着桥梁作用。

2.3。 Codehaus XFire。

Codehaus XFire是下一代Java SOAP框架。它是一个免费且开源的SOAP框架，可以让您轻松简单地实现Web服务。它还提供了许多Web服务规范中标识的功能，这些功能在大多数商业或开源工具中尚不可用。据称它具有更高的性能，因为它基于低内存StAX（基于XML的Streaming API）模型，但没有数据可以证明这一事实。

2.4。黑森州。

Hessian二进制Web服务协议使得开发Web服务变得简单而且可用，无需大型框架，因此开发人员不需要花费更多的时间和精力来学习一系列的协议。由于它是一种二进制协议，因此它在发送二进制数据时运行良好，无需使用附件扩展协议。像手机PDA这样的J2ME设备可以使用Hessian以更好的性能连接到Web服务，因为它是一个小型协议。黑森州是以黑森州布料命名的，这是英国的布尔拉普术语。它是这样命名的，因为粗麻布很简单，实用，而且非常有用，但却是非常普通的材料，就像Hessian协议的特点。

3.评估指标。

在比较这个项目中的四个框架时考虑了不同的因素。一些指标用于确定性能和效率;有些是为了展示透明度和抽象。本节介绍这些指标。

3.1。潜伏。

就网络而言，等待时间表示将数据发送回请求需要多少时间。这包括请求发送到服务器的时间，服务器处理任务的时间以及结果发回的时间。网络延迟由许多因素造成，如传播，传输，调制解调器和路由器处理以及存储延迟。传播是一个物体（如数据）以光速从一个位置转移到另一个位置所需的时间。传输是光纤或无线网络等媒介的延迟。调制解调器和路由器需要一些时间来检查数据包的标题。存储延迟是实际硬件存储（例如硬盘驱动器）存储接收到的数据所花费的时间。在这个项目中，测试延迟是在不同的场景下进行的，例如请求1,2,3,4和5 MB数据以及1,5,10,15,20个客户端同时请求数据。从这些测试的结果中，可以找到并比较每个框架的趋势。

3.2。吞吐量。

吞吐量是在一定时间内处理的客户端或数据的数量，如秒数。它与延迟高度相关，因为具有高延迟的情况会导致低吞吐量，而低延迟情况会导致高吞吐量。但是，通过查看延迟图，我们只能说出响应时间的趋势，而我们可以通过查看吞吐量图来确定框架的最有效方案。

3.3。内存使用情况。

在计算中，内存是数据存储器，用于临时存储用于计算计算机的数据。有各种各样的内存，如高速缓存，闪存，随机存取内存（RAM），虚拟内存等。无论是内存，由于成本和空间的原因，它们在服务器上都是有限的。使用更少内存的框架将具有允许服务器具有更高容量的优势。

3.4。 CPU使用率。

中央处理单元（CPU），也称为处理器，是计算机中用于解释程序指令和处理数据的组件。虽然它一次只能处理一个任务，但当有多个任务需要完成时，而不是完成一个任务，然后转到另一个任务，CPU设计为在必要时完成一个任务之前切换到其他任务，以便它会像是在同一时间执行多个任务一样行事。但是，大型任务可能占用大量CPU时间，这会减少计划用于其他任务的时间。使用较少CPU的框架将允许服务器有更多时间来执行其他任务。

3.5。代码的源代码行。

框架中使用的源代码行（SLOC）可以指示框架的透明度和抽象度。框架的主要目标是通过不必从头编写整个代码来节省开发人员的时间和精力。因此，框架所需的代码量越少，这个框架就可以节省更多的时间和精力。但是，由于某些行可能很长，而某些行很短，因此代码行不能完全精确。所以文件的数量和文件的大小也是一个考虑因素。

4.统计分析方法。

检索测试数据以比较性能后，我们需要一种方法来分析结果。通过简单地计算平均响应时间并将它们绘制成图形是不足以进行分析的。查看平均响应时间1.5秒和1.6秒，我们不能确定这是否有很大的差异。因此，需要统计分析方法来判断差异是否显着。在这个项目中，一般线性模型（GLM）[9]和双向方差分析（双向ANOVA）用于统计分析。此外，统计分析系统（SAS）[10]被用作帮助计算所需统计分析的工具。

4.1。 SAS系统。

SAS系统是具有各种统计模块和程序的统计分析软件。他们使用第四代编程语言（4GL）代码，程序由三个主要组件组成 - 数据步骤，过程步骤和宏语言。数据步骤用于输入数据，如将数据插入代码或从数据文件读取数据。程序步骤是使用统计方法和模型来分析在数据步骤中读取的数据。宏语言是为了减少整个程序中反复使用的函数的冗余。

4.2。 GLM模型。

GLM模型是一般情况下使用的统计线性模型。它是许多统计分析的基础，如t检验，方差分析，协方差分析（ANCOVA）等。理解GLM模型如何工作的最简单情况是双变量情况。此分析的目标是使用一种方法来准确描述此图中的信息。使用GLM模型，我们试图找出最接近图中所有点的直线。这条线可以写成：y = b0 + b1x + e，其中y是y轴变量，x是x轴变量，b0是截距（当x等于0时y的值），b1是直线的斜率，e是误差。通过求解b0和b1，我们可以得到关于描述图中点的这条直线的信息。在其他情况下，如果有两个以上变量，公式可以扩展为：y = b0 + b1x1 + b2x2 + b3x3 + ... + bnxn + e，其中n是情境变量的数量。但解决这些问题的机制是一样的。

5.结果和分析。

为了从SAS分析中获得最好的结果，每个案例都进行了20次测试。由于四个框架和五个不同数量的客户端进行了测试，有二十种不同的情况。为20个不同情况添加20个测试时间将导致SAS计算400个数据集。除了客户数量之外，还考虑数据的大小。在发送五种不同大小的数据的情况下，将有二十种情况下总共400个数据集。通过记录调用Web服务之前的时间并在接收到请求的数据后立即记录时间，然后减去时间差来测量响应时间。

5.1。结果