

Web 系统性能测试研究

兰景英¹, 王永恒²

(1. 西南科技大学 计算机科学与技术学院, 四川 绵阳 621010;
2. 西藏山南军分区司令部, 西藏 山南 856000)

摘要: Web 系统的性能测试是 Web 应用开发过程中的一个关键环节, 然而 Web 站点的复杂性及其用户行为的不可预见性使得对其性能测试极其困难, 而且测试环境和用户使用环境的差异性使得 Web 系统性能测试数据不精确或不可靠。分析了 Web 系统结构和 Web 系统的性能特点, 深入讨论了 Web 系统性能测试的内容和性能测试的原理, 并比较了目前常用的性能测试工具, 通过一个测试项目介绍了自动化测试的实施过程, 总结出一套切实可行的 Web 系统性能测试流程。

关键词: Web 系统; 性能测试; 负载; 压力; 测试工具

中图分类号: TP311.56

文献标识码: A

文章编号: 1673-629X(2008)11-0090-04

Research on Web Application Performance Testing

LAN Jing-ying¹, WANG Yong-heng²

(1. College of Computer Science and Techn., Southwest University of Science and Techn., Mianyang 621010, China;
2. Tibet Shannan Military Region Command, Shannan 856000, China)

Abstract: Performance testing of Web application is a crucial step in the Web application development. However, the complexity of Web and unpredictable operation of Web users make it very difficult to test the performance of Web. Analyzes the architecture of Web application and the performance characteristics of the Web application at first. Then discusses deeply on the content and principle, and compares some performance test tools. Through a test project, introduces the testing flow of automatic test.

Key words: Web application; performance testing; load; stress; test tools

0 引言

Web 正以其广泛性、交互性、快捷性和易用性等特点越来越受到企业和个人的青睐。Web 应用软件的规模不断扩大, 复杂性也逐渐增加, 随之带来的问题就是 Web 站点流量的迅速增长, 导致一些 Web 站点由于负载过重而变得反应迟缓^[1]。当站点的性能问题达到无法忍受的程度时, 就会导致用户过早地终止其在这个网站的事务。对于商业网站来说, 用户放弃他们的网站将导致收入的减少。

因此在 Web 应用发布之前, 就应该利用性能测试工具模拟成百上千个用户, 测试系统性能和行为, 对整个 Web 服务提供的各种网络服务的负载能力进行全面、有效地测试^[2]。获得在保证服务质量的前提下所能支持的最大用户数, 获得在各种负载条件下用户请求的响应时间、数据传输速率等关键信息。这样做不

仅优化了 Web 应用的性能, 加快了开发速度, 还增强了对系统的信心。在站点开始启用后, 还可以使用性能监视工具实时监视站点的执行情况, 检测和报告可能出现的性能问题。

1 Web 系统体系结构

对于基于 Web 的应用系统, 用户直接面对的是客户端浏览器^[3]。用户在使用系统时, 请求之后的事务逻辑处理和数据的逻辑运算由服务器与数据库系统共同完成, 对用户而言是完全透明的。运算后得到的结果再通过浏览器的方式返回给用户。这个过程可分成一些子步骤, 每一个子步骤的完成可理解为通过一个单独的应用服务器来处理, 这些应用服务器在最终得到用户所需的结论之前, 相互之间还会进行一定的数据交流和传递。图 1 是一种典型的 Web 应用体系结构。从 Web 系统的体系结构可以看出, Web 系统性能测试主要集中在三个方面: 应用在客户端的性能测试, 应用在网络上的性能测试和应用在服务器上的性能测试。

收稿日期: 2008-02-04

基金项目: 国家自然科学基金(10676029)

作者简介: 兰景英(1975-), 女, 四川隆昌人, 助教, 硕士, 研究方向为软件测试技术、网络安全。

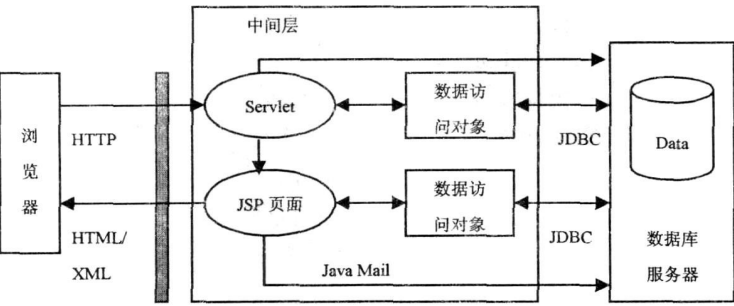


图 1 一种典型的 Web 应用体系结构

2 Web 系统性能测试

2.1 Web 系统的性能指标

在进行 Web 测试之前,需要考虑 Web 系统的性能指标。Web 系统常用的性能指标有以下四种^[1]:

(1) 响应时间。

响应时间也称为用户的等待时间。从用户的角度来看,指的是从用户发出请求到收到服务器的响应所经历的延迟。一般来说,使用站点的用户数越少,服务器处理的请求也就越少,响应时间就会比较短,当用户数目增多时,服务器比较繁忙,也就是说服务器要承受的负载会比较多,响应时间就会增加。

Web 响应时间与很多因素有关,比如 HTTP 协议,Web 服务器,网络带宽,用户的位置等。在测试时要考虑在真实环境中的响应时间,以便得到真实的负载。

(2) 吞吐量。

吞吐量是指在某一个特定的时间单位内 Web 应用所处理的用户请求数目。总的来说,用户的请求数目越多,吞吐量就会越大。但是当用户请求达到 Web 应用所能够并发处理的最大用户请求数目时,此时的用户请求数目就是最大的吞吐量。通过吞吐量可以找到系统的瓶颈问题。

(3) 用户放弃率。

用户作为 Web 站点成功与否的最终的决定者,他感知的性能是至关重要的,不过由于众多的用户需求太复杂了,所以很难满足。最直观的反应就是站点的响应时间。响应时间的长短是判断用户满意程度的最重要的因素之一。有一个流行的“8 秒”定律,如果系统对用户的请求不能及时响应,那大多数新用户等待的时间上限是 8 秒钟。即当用户下载一个页面的时间超过 8 秒的时候,他们可能会放弃当前的网站。

(4) 用户行为。

在开放的网络中,用户行为是不可预见的,因为用户在使用网络的时候,他们的行为是不同的,有些用户的点击速度要比其他用户的快一些,这样在特定的时

间内,这些速度较快的用户将会产生更多的负载。在负载测试过程中考虑用户对网站的熟悉程度以及对网站的兴趣的大小将有助于设计出精确的测试。

在 Web 性能测试过程中,需要考虑的性能测试指标还有很多,如页面请求次数、用户会话、页面请求分布等等,在性能测试过程中要考虑全面的性能测试指标。

2.2 Web 系统性能测试内容

性能测试主要确定在用户可接受的响应时间内,系统能够承担的并发用户的数量,能够同时进行的交易的数目,以及不同负载情况下页面的下载时间和检测瓶颈可能发生的位置,以确保将来系统运行的安全性、可靠性和执行效率。Web 性能测试主要包括连接速度测试、负载测试、压力测试 3 个部分^[4]。负载测试是为了测量 Web 系统在某一负载级别上的性能,以保证 Web 系统在需求范围内能正常工作。压力测试是测试系统的限制和故障恢复能力,也就是测试 Web 应用系统会不会崩溃以及在什么情况下会崩溃。

Web 性能测试通过搭建与实际系统相类似的测试环境,模拟实际用户的操作在不同的负载条件下,客户端和服务端端的资源使用情况。一个完整的性能测试解决方案必须包括:

- (1) 测试一个组合了各种软件应用程序和硬件平台的系统;
- (2) 确定服务器与任何给定应用程序的适应性;
- (3) 模拟多个客户端与单个服务器应用程序之间进行交互的环境;
- (4) 在几十、几百甚至是上千个用户的负载情况下测试应用程序。

2.3 Web 性能测试的原理

一般而言,对 Web 应用系统进行测试包括下面两个部分:一部分是查看系统在一定的负载条件下运行是否合乎需求,另一部分是通过系统在一定负载下运行所得的数据分析它的瓶颈。因此,能够对当前的系统产生一定的负载,并且能够分析系统在一定负载下的表现是实现性能测试的关键。一般来说,使系统产生负载有以下 4 种方法^[5]:

(1) 重复。

重复是性能测试对系统产生压力的基本手段。功能验证测试可以用来弄清楚一个操作能否正常执行。而性能测试将确定一个操作能否正常执行,并且能否继续在每次执行时都正常。

(2) 并发。

并发是同时执行多个操作的行为。也就是说在同

一时间执行多个测试,例如在同一个服务器上同时调用许多 Web 服务。

(3) 量级。

性能测试应该应用于产品的方面是每个操作中的负载量。压力测试可以重复执行一个操作,但是操作自身也要尽量给产品增加负担。例如,一个 Web 服务允许客户机输入一条消息,您可以通过模拟输入超长消息的客户机来使这个单独的操作进行高强度的使用。单独的高强度操作自身可能发现不了代码错误(或者仅能发现功能上的缺陷),但与其它条件结合在一起时,您将可以增加发现问题的机会。

(4) 随机变化。

任何性能测试都具有一些随机性。如果对系统中需要输入的一些参数做一些变化,就能够在每次测试运行时应用许多不同的代码路径。比如:使用重复时,在重新启动或重新连接服务之前,可以改变重复操作间的时间间隔、重复的次数,或者也可以改变被重复的 Web 服务的顺序。

同时,性能测试应该记录下系统在压力下运行时的一些参数,例如响应时间、最大/最小并发数、失败的次数、正常连续运行的最长/最短时间、并发数与失败的关系等,这些数据能够帮助测试者很快地找到系统的瓶颈,并且对这些数据进行分析。

2.4 Web 性能测试工具

采用人工的方式模拟成千上万的用户进行性能测试几乎是不可能实现的,因此需要采用专门的性能测试工具来完成。性能测试工具可分为两类:一类用于观测服务器端的性能指标,另一类用于客户端模拟并发用户^[6-7]。性能监视器用于观测服务器端的性能指标,可以利用服务器上的工具程序,通过在性能监视器中添加所要观察的计数器(即性能指标),就可以获得程序运行时的性能指标变化图。用于客户端的 Web 应用程序的性能测试工具通过可重复的、真实的测试,能够彻底地度量应用的可扩展性和性能,可以在整个开发生命周期跨越多种平台最大执行测试任务,可以模拟成百上千的用户并发执行关键业务而完成对应用程序的测试。

目前 Web 性能测试工具主要有 Mercury LoadRunner, Radview WebLoad, Rational Performance Tester, WAS (Web Application Stress Tool), OpenSTA (Open, Systems Testing Architecture)等。

LoadRunner 是一种预测系统行为和性能的工业标准级负载测试工具。通过以模拟上千万用户实施并发负载及实时性能检测的方式来确认和查找问题,能够对整个企业架构进行测试^[8]。LoadRunner 能支持

广泛的协议和技术。

Radview WebLoad 可以让 Web 应用程序开发者通过模拟真实用户的操作,生成大量压力负载来测试 Web 的性能。WebLoad 可以预定义 Web 应用程序应该满足的性能指标,然后测试系统是否能满足这些指标,可以设定 WebLoad 采用自增用户数的循环方式进行测试,这样可以自动测试得到系统的最大用户容量。

WAS 是专门用来进行实际网站压力测试的一套工具,用来模拟 Web 浏览器对使用 http1.0 或 1.1 标准的 Web 服务器的请求,而不用考虑 Web 服务器运行在何种平台上。

OpenSTA 是用 C++ 语言开发的软件,可以执行分布式测试,通过简单的图表形式和分布的测试,对于简单的和可靠的 HTTP 测试来说是很好的软件。

3 Web 系统性能测试实例

文中对某校教学管理系统进行性能测试。该系统采用的是基于 J2EE 的三层体系结构,由于涉及到大量的用户(约 2 万师生),系统的可靠性、稳定性至关重要。采用 LoadRunner 对其进行性能测试。

采用 LoadRunner 进行测试的基本流程是:先将用户的实际操作录制成脚本,然后产生一定数量的虚拟用户运行脚本,最后生成相关的报告以及分析图。本系统的性能测试过程如下:

(1) 脚本录制。

通过 LoadRunner 的 VuGen 记录用户操作流程来创建脚本,并通过对录制后的脚本添加事务、集合点、参数等来强化脚本。

(2) 设计测试方案。

为了模拟系统真实的负载,则需要设定测试方案。用 LoadRunner 的 Controller 能很快组织起多用户的测试方案。Controller 提供了一个互动的环境,在其中既可以建立起持续且循环的负载,又能管理和驱动负载测试方案,并可利用它的日程计划来定义用户在什么时候访问系统以产生负载。另外,还可以用 Controller 来限定不同的负载方案。

在本方案中通过所有的用户同时执行一个动作(如登录到系统)来模拟峰值负载的情况。在测试系统登陆功能的性能时,分别设计了 50, 100, 500, 1000, 2000 个虚拟用户,并分别设置虚拟用户以一定的间隔时间按怎样的比例递增,逐步启动来与服务器建立连接。

(3) 执行测试。

运行方案时,LoadRunner 会为用户组分配负载生成器并执行它们的 Vuser 脚本。

(4)分析测试结果。

在性能测试脚本的执行过程中,会收集到大量的测试数据,包括脚本中定义的事务的持续时间,脚本执行过程中的错误、警告和通知信息等本次测试的部分测试数据如表 1。从表中的数据可以看出,当用户达到 1000 的时候,系统响应速度明显变慢,而且出现部分错误,因此系统在 1000 左右出现了性能瓶颈,需要对系统的性能进行调优。另外,还需要分析测试环境对测试结果的影响,因为测试的时候不可能完全模拟用户真实的使用情况。

表 1 不同虚拟用户规模下的性能测试数据

	Maximum Running Users	Average Throughput (bytes/second)	Total Hits	Average Hits per Second	Average Transaction Response Time	Fal/Error
10	10	90 949	624	28 364	15.275	0
100	100	690 136	6 274	216 345	17.994	0
500	293	971 353	31 247	303 369	20.933	1
1000	337	1 032 263	62 685	323 119	27.787	7

4 结束语

Web 系统设计的复杂性及其分布、异构、并发和平台无关的特性,使得 Web 系统的性能测试要比一般的性能测试复杂得多。在进行 Web 系统性能测试时存在的难点主要有负载的不可预知性、测试场景设计的困难性、测试环境和真实环境的差异性,因此 Web

系统性能测试比传统的应用软件要复杂得多。文中分析了 Web 系统测试的原理和内容,并根据项目中性能测试的实施情况,对 Web 系统性能测试过程进行了概述。下一步的主要工作是 Web 系统性能测试中场景的自动生成技术的研究和 Web 系统测试框架的研究。

参考文献:

[1] 芮素娟, 丁晓明. Web 应用性能测试进展[J]. 计算机科学, 2006, 33(8): 278—290.

[2] 刘苗苗. Web 性能测试的方法研究与工具实现[D]. 西安: 西安理工大学, 2007.

[3] 胡 蓉, 缪淮扣, 刘焕洲. 一种基于 Web 软件集成测试的建模方法[J]. 计算机科学, 2007, 34(6): 253—257.

[4] 张友生. 基于 Web 的系统测试方法[J/OL]. 2003—04. 中国系统分析员. <http://testing.csai.cn/testweb/NO226.htm>.

[5] 谭 浩, 关 昕, 马 力. 性能测试的原理及其自动化工具的实现[J]. 计算机工程与设计, 2006, 27(19): 3660—3662.

[6] 姜昌华, 朱 敏, 陈优广. Web 应用程序压力测试[J]. 计算机应用, 2003, 13(10): 75—77.

[7] 杨 志, 关 昕, 马 力. Web 系统性能测试实施过程[J]. 计算机工程与设计, 2007, 28(10): 2284—2286.

[8] 测试时代. 负载性能测试工具列表大全[EB/OL]. 2006—04. <http://www.testage.net/TestTech/AutoT/200604/438.htm>.

(上接第 89 页)

对较高。在图 5 中可以看到, Reactive 下的路由负载最低,而其它两种在网关广播广告消息时隙很短的情况下路由负载很高,这是因为网关不停地广播广告消息增加了网络路由负载。

4 结束语

文中研究了无线 Ad hoc 网络中的无线节点访问 Internet 技术,通过修改 AODV 协议,扩展其路由信息包,来实现无线节点之间,以及无线节点与 Internet 固定节点之间的数据传输。以网关作为无线节点与 Internet 固定节点之间的桥梁,提出了三种网关发现方式,并通过 NS2 模拟试验,对这三种方式下的网络性能进行了比较。三种方式下的分组传输率差不多,Reactive 下端到端时延要比另外两种高,是因为它在发送数据之前需要先建立到网关的路由。

在网关广播广告消息时隙短的情况下,Hybrid 下的路由负载比较高,其次是 Proactive, Reactive 最低,但

随着网关广播时隙的变长,前两种方式的路由负载都趋近于 Reactive。综合来说,如果把网关的广告消息广播时隙定在 40~50 秒之间,则 Hybrid 方式下的网络性能要较其它两种方式优越。

参考文献:

[1] 余冬梅, 赵文来, 张秋余. IPv6 下 Mobile IP 的实现[J]. 计算机工程 2002(11): 6—10.

[2] Perkins C, Belding—Royer E M. Ad Hoc on—demand Distance Vector (AODV) Routing for IP Version 6[R]. draft—perkins—manet—aodv6—01.txt, 2000.

[3] Macker J, Corson S. Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations[S]. RFC2501, 1999.

[4] Liang Qin. Pro—active Route Maintenance in DSR[D]. Ottawa: School of Computer Science—Carleton University, 2001.

[5] Sun Y, Belding—Royer E M, Perkins C E. Internet connectivity for Ad hoc mobile network[EB/OL]. 2002—07—15. <http://www.cs.ucsb.edu/~ebelding/txt/mip-aodv.ps>.