

WebMark: 一个 Web 服务器性能测试工具*

张广艳⁺, 郑名扬, 鞠九滨

(吉林大学 计算机科学与技术学院, 吉林 长春 130012)

WebMark: A Tool for Testing Web Server Performance

ZHANG Guang-Yan⁺, ZHENG Ming-Yang, JU Jiu-Bin

(College of Computer Science and Technology, Jilin University, Changchun 130012, China)

+ Corresponding author: Phn: 86-431-5166476, E-mail: guangyan.zhang@163.com

<http://www.jlu.edu.cn>

Received 2002-05-16; Accepted 2002-08-14

Zhang GY, Zheng MY, Ju JB. WebMark: A tool for testing Web server performance. *Journal of Software*, 2003,14(7):1318~1323.

<http://www.jos.org.cn/1000-9825/14/1318.htm>

Abstract: In this paper, the metrics and principles of testing Web server performance are summarized, and a more rational metric, weighted response time, is proposed. A tool called WebMark is presented for testing Web server performance, which manages asynchronous I/O by event-driving mode and emulates Internet environment by modifying the TCP/IP stack of clients. The testing of the overall performance of Apache proves that developing such a powerful testing tool as WebMark is of great necessity.

Key words: Web server; performance testing; WebMark; asynchronous I/O management; Internet environment emulation; weighted response time

摘要: Web 及其应用程序的普及使得 Web 服务器的性能测试变得越来越重要,而现有的测试工具都具有一定的适应性限制。总结了 Web 服务器性能测试的指标和原则,提出了一个更加合理的测试指标——加权响应时间。介绍了一个 Web 服务器性能测试工具 WebMark。它用事件驱动的方式管理异步 I/O,通过修改 Client 的 TCP/IP 协议栈来模拟 Internet 环境。使用它对 Apache 进行了全面测试,证明了研制 WebMark 这样一个功能强大的测试工具是十分必要的。

关键词: Web 服务器;性能测试;WebMark;异步 I/O 管理;模拟 Internet 环境;加权响应时间

中图法分类号: TP393 文献标识码: A

Web 服务器性能测试是指,在一定的软硬件环境下,按照统一的度量标准,测试 Web 服务器对各种请求的响应速度、最大顺畅连接数等性能指标,给出公正、可靠的性能评价。

Web 及其应用程序的普及对 Web 服务器提出了越来越高的性能要求。目前可以见到各种 Web 服务器平台,然而根据 Mercury 的研究报告,98%的 Web 服务器都未能达到人们所期望的性能,平均只能发挥人们所期望性

* Supported by the National Natural Science Foundation of China under Grant No.60073040 (国家自然科学基金)

第一作者简介: 张广艳(1976—),男,吉林榆树人,硕士生,主要研究领域为计算机网络,分布式系统。

能的 1/6 左右^[1].Web 服务器性能测试能够确定影响 Web 服务器性能的关键因素,从而可以有针对性地进行分析和改进,避免 Web 服务器研究和优化过程中的盲目行为.另外,它也是选取不同 Web 服务器的重要参考.

本文介绍我们自行开发的 Web 服务器性能测试工具 WebMark.该工具采用可靠的方法,模拟真实负载,以各种合理的指标来评价 Web 服务器性能,具有实验容易控制、结果易于解释、工具本身性能较高及实验可重复的特性.

1 基本定义

定义 1(Web 系统). 以 Web 服务器为中心,由 Web 服务器及其所有相关事物组成的系统,包括 Web 服务器、许多 Web 客户机和连接二者的网络设施以及 Web 内容和用户访问模式等.

定义 2(Web 特征). 这是指 Web 系统或者它的某些组成部分在完成 Web 事务过程中表现出来的特点.

定义 3(测试指标). 评价 Web 服务器性能高低的尺度和依据.

常见的测试指标主要有:(1) 最大顺畅连接数,是指在满足 Web 用户可接受的响应时间或数据流量的前提下,Web 服务器所能支持的最大并发连接数.SpecWeb99^[2]认为,如果客户和服务器之间的连接对各种请求的平均带宽大于 320Kb/s,则称其为“顺畅连接”(conforming connection).(2) 连接建立时间.(3) 响应时间,是指发送 HTTP 请求到接收完响应的的时间.(4) 数据流量.(5) 每秒钟内 HTTP 的事务数.

本文综合连接建立时间和响应时间,提出一个新的测试指标.假设一个连接的建立时间为 S ;一个页面 P 包括 n 个文件,第 i 个文件的响应时间为 R_i ; T_0 和 T_1 分别为使用 HTTP/1.0 和 HTTP/1.1 获得页面 P 的总时间,则

$$T_0 = (S + R_1) + (S + R_2) + \dots + (S + R_n) = nS + \sum_{i=1}^n R_i; \quad T_1 = S + R_1 + R_2 + \dots + R_n = S + \sum_{i=1}^n R_i.$$

令 \bar{R} 表示各个文件响应时间的平均值,则使用 HTTP/1.0 和 HTTP/1.1 获得一个文件平均消耗的时间 \bar{T}_0 和 \bar{T}_1 分别为

$$\bar{T}_0 = \frac{T_0}{n} = S + \frac{1}{n} \sum_{i=1}^n R_i = S + \bar{R}; \quad \bar{T}_1 = \frac{T_1}{n} = \frac{S}{n} + \frac{1}{n} \sum_{i=1}^n R_i = \frac{S}{n} + \bar{R}.$$

由此提出了新的测试指标——加权响应时间,即 $T = \omega S + \bar{R}$.这里, ω 为通过一个连接所传送文件数目的倒数,即通过一个连接传送的各个文件分担建立该连接的开销.加权响应时间与连接建立时间和响应时间相比,更能体现用户的感受.

下面给出 Web 服务器性能测试的两点原则:

测试原则 1. 假设要对 Web 系统 S 中的 Web 服务器 T 进行性能测试,对于 S 中的任意一个 Web 特征 δ ,如果 δ 影响 T 的性能展示,那么 δ 应该在测试过程中被模拟.

测试原则 2. 要在测试环境 T 中对被测系统 S 进行性能测试,必须保证 T 中除 S 以外的任何其他组成部分不会成为 T 的性能瓶颈.

2 WebMark 的功能及特点

(1) 支持 HTTP/1.0 和 HTTP/1.1

由于历史原因,HTTP/1.0 还将在一段时间内使用,而 HTTP/1.1 以其更加良好的性能和可靠的安全性,将无可争议地取代 HTTP/1.0 而成为标准.Web 服务器使用 HTTP/1.1 比使用 HTTP/1.0 性能有明显的提高^[3],它的两个主要贡献因素是持续连接和 HTTP 请求的流水线操作^[4].

(2) 支持 SSL

SSL 已经成为用来确保安全性和验证 Web 数据传输的最常用的方法^[5].与明文传输相比,SSL 的密钥协商和数据加密传输使用了更多的计算资源和网络带宽,使性能降低到原来的 1/9~5/17^[6].

(3) 支持 Web 服务器过载情况下的性能测试

Web 请求的突发性经常使 Web 服务器达到过载.在过载情况下,由于处于内核空间的 TCP/IP 协议栈的调度优先级大于处于用户空间的 Web 服务器软件,Web 服务器把几乎所有的资源都用于处理到来的网络包,但由于没有 CPU 时间来运行接收这些包的 Web 服务器软件,后来又不得不把这些包丢弃.这将极大地降低响应速度,

甚至导致“活锁”状态.另外,性能测试主要是找出 Web 服务器的性能瓶颈,从而方便性能优化,而过载测试是找出性能瓶颈的有效途径.

(4) 有代表性的请求负载

测试环境(TestBed)是对 Web 系统的一个模拟.为了使 Web 服务器性能测试的结果更具说服力,与请求负载相关的 Web 特征应当服从实验测得的分布模型.

(5) 在局域网中模拟 Internet 环境

在局域网中模拟 Internet 环境包括模拟丢包和网络延迟.Internet 上的丢包增加了连接的建立和持续时间,进而增加了活动连接数以及服务器和网络上的负载.丢包后,TCP的拥塞窗口变小,也限制了下一步的网络流量.大量研究证实,丢包是影响 Web 服务器和网络性能的一个重要因素^[7,8].当 Web 服务器经受 4%~5%的丢包率时,高峰负载经常比没有丢包时超出 7%^[9].网络延迟在服务器上增加很多并发的活动连接,因为要保存连接状态,大量的连接增加了服务器的内存压力,减少了可用于文件 cache 的内存,进而降低了吞吐量.

WAWM^[10]在真正的 Internet 上进行性能测试.首先,Internet 的状态(包括网络延迟和丢包率)难以控制;其次,WAWM 的 Web 特征不具有可重复性.在局域网中模拟 Internet 环境成功地克服了 WAWM 的缺点.

WebMark 还具有以下功能和特点:支持动态内容;测试工具本身高性能;支持 Cookie 和 POST 操作;能够指导被测系统(system under test,简称 SUT)的改善;具有分布式、易扩充的体系结构;能够比较容易地根据需要而升级.

3 WebMark 的实现

3.1 软件体系结构

WebMark 包括一个 Manager 和若干个 Client(如图 1 左半部分所示).每个 Client 按照 Manager 的指示向 SUT 发送 HTTP 请求,生成 Web 负载,然后验证接收到的响应,记录自身的测试结果.Manager 除了同步各个 Client 的测试状态,还在测试的最后,收集来自所有 Client 的结果数据,形成统一的测试报告.SUT 上增加的 S-Setup 程序用来生成被测系统上存放的 Web 文件集.

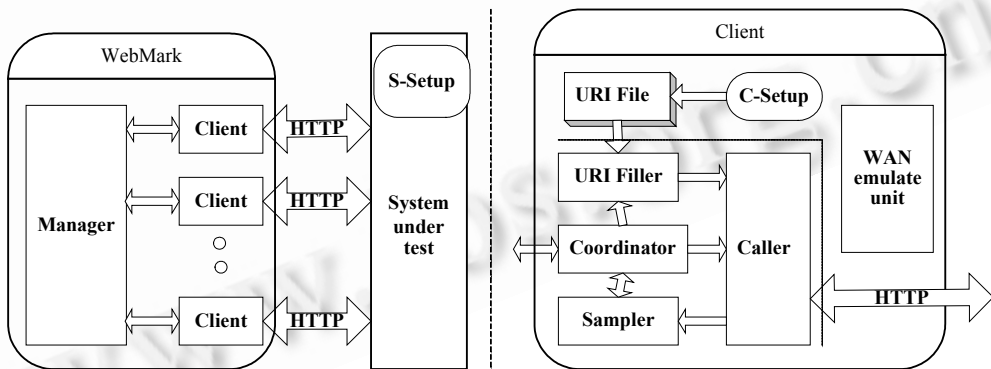


Fig.1 WebMark software architecture

图 1 WebMark 软件体系结构

每个 Client 主体在逻辑上分为 4 个模块(如图 1 右半部分所示):协调器、URI 填充器、呼叫器和采样器.协调器用于配合 Manager 完成同步及结果收集的功能;后 3 个模块用于完成自身的测试任务.URI 填充器负责在适当时间填充满足不同分布类型的请求 URI,以使 Web 服务器承受某一分布类型的负载.呼叫器处理与 SUT 的所有通信,如连接管理、发送 HTTP 请求及处理响应信息等.采样器在整个 HTTP 呼叫过程中随时采集有用信息,测试结束后,整理各种测量数据,并产生本地的测试结果.另外,Client 有两个辅助成分:C-Setup 和 WAN 模拟单元.前者在测试开始前生成 URI 文件,以供 URI 填充器在适当的时候读取;后者是通过修改 TCP/IP 协议栈获得的一个逻辑功能模块,实现在局域网中模拟 Internet 环境.

Client 采用单进程实现,4 个模块之间的交互通过事件驱动的方式.如果一个模块对某个事件 E_i 感兴趣,就注册 E_i 的一个处理器(即处理该事件的函数) h_{ij} .一旦 E_i 发生,就调用该事件的所有处理器 $h_{ij}(1 \leq j \leq N)$.其中, N 为注册到 E_i 上的处理器个数.一个事件可以注册多个处理器,而当事件发生时,这些处理器是按照注册的先后顺序来执行的,所以,如果一个事件的两个处理器对执行顺序敏感,注册时应保证所需顺序.

采用单进程来管理多个并发连接,必须采取异步 I/O 的形式,异步 I/O 与同步 I/O 相比,优点是速度快,缺点是难于控制.WebMark 采用事件驱动的方式管理异步 I/O,大大降低了管理复杂度.当异步 I/O 过程中的某一环节发生时,只需触发相应事件,而不必关心它是如何被处理的.

3.2 整体工作过程

Manager 先在一个指定端口监听来自 Client 的连接请求.然后,每个 Client 启动协调器,与 Manager 建立连接,Manager 发送配置信息.每个 Client 准备就绪后,报告 Manager 并等待 Manager 发布同步的启动指令.Client 一旦获得启动指令,其上的呼叫器就开始生成请求负载,URI 填充器准备请求内容,采样器采集测量数据,从而形成本地的测试结果.Client 完成自身的测试任务之后,协调器把本地的测试结果送给 Manager,生成总测试报告.

3.3 实现细节

为了在局域网中模拟 Internet 环境,修改了 Client 的 TCP/IP 协议栈中发送数据包的部分,而接收动作未作任何修改.为了模拟到来的包的丢失,忽略一定数量的到来的包不予确认;模拟发出的包的丢失是这样的,不发送相应的数据包,但内部行为跟发送了一样.模拟网络延迟是把数据包放到一个队列里,一个专门的内核进程连续扫描这个队列,达到延迟时限的包被发送.这实际上是用发送包的延迟来模拟一条路径上包往返两次的延迟.延迟的计算公式为

$$\text{delay}_{\text{packet}} = 2 \left(\frac{\text{size}_{\text{packet}}}{\text{bandwidth}_{\text{path}}} + \text{latency}_{\text{path}} \right).$$

文献[11]是通过修改路由器的 TCP/IP 协议栈来模拟丢包和网络延迟的.但是,路由器作为 TestBed 的集中环节,很容易成为系统的性能瓶颈.本文的方法则避免了这个问题.

SURGE^[10]捕获了 Web 站点的各种流量特性,所生成的负载是有代表性的.WebMark 模拟不定数量的用户群,无法实现 OFF 时间分布,这无疑会削弱请求到达过程的突发性,但是,实现 OFF 时间分布降低了负载强度.WebMark 生成 Web 文件集的 S-Setup 程序是和 SURGE 完全一样的;Client 中生成 URI 文件的 C-Setup 程序是通过改造 SURGE 而成的,相当于录制 SURGE 的 URI 请求序列.这就保证 WebMark 保留了 SURGE 除了 OFF 时间以外的所有分布,负载是有代表性的,而且又能保证负载强度.

WebMark 采用的是单进程结构.为使该进程获得更多的计算资源,采用 nice 命令提高它的调度优先级.另外,每个进程同时打开的 Socket 数是有限制的,采用 ulimit 可以把这个限制扩大.但如果采用 select 系统调用,由于 libc 中 FD_SETSIZE 的 1024 限制,ulimit 命令还是不能真正起作用.我们采用 poll 系统调用解决了这个问题.S-Client^[11]和 httpperf^[12]采用的都是单进程结构,然而,S-Client 根本没有注意到这两个问题;而 httpperf 虽然提到了第 2 个问题,却没有给出切实可行的解决办法.

4 使用 WebMark

使用 WebMark 对 Apache 服务器进行了性能测试.Web 服务器配置为 677MHz 赛扬 CPU,128M PC133 内存,100Mbps RTL8139 网卡,Linux-2.4.7-10 系统上运行 Apache-1.3.20 软件;4 台客户机配置均为 933MHz Pentium III CPU,128M PC133 内存,100Mbps RTL8139 网卡,Linux-2.4.7-10 操作系统;通过 100Mbps 交换机连接成独立网络.采用 SURGE 的标准配置参数,zipf 受欢迎程度参数为 2000,Web 文件集包括 2 000 个文件,约占 55Mb 空间.URI 文件包括 172 020 个 URI,每次都是完成这 172 020 个 URI 请求后结束实验.

首先,验证 Web 服务器在过载情况下的性能展示,如图 2 所示为通过 HTTP/1.0 协议测试的结果.随着每秒内 HTTP 请求数从 100 增加到 300,Web 服务器的吞吐量(每秒处理的 HTTP 事务数)由 100 线性增长到 300;之后,吞吐量的增长幅度变缓,当请求率到达 400 时,吞吐量达到最大值 351.824Trans/S.接着,Web 服务器达到过载,

吞吐量随着请求率的加大而逐渐减少.这与文献[11]中的结果基本吻合.

其次,通过 HTTP/1.1 进行测试,WebMark 按一定速率和 SUT 建立连接,通过每个连接完成 10 个 HTTP 事务.A 组是使用持续连接,但不使用请求的流水线操作;B 组则二者均使用,一条连接上的 10 个请求分为两组,一组 5 个同时发出,等这 5 个请求得到响应之后,再同时发出另外 5 个请求.得到两组实验结果如图 3 所示.两组结果与图 2 的性能走势大体相同,A 组最大吞吐量是 518.56 Trans/s,B 组则是 524.656 Trans/s,显然,Web 服务器使用 HTTP/1.1 协议比使用 HTTP/1.0 所展示的性能有明显的提高.使用请求的流水线操作的 B 组的最大吞吐量比 A 组没有明显的提高,但过载时,B 组吞吐量明显比 A 组下降得缓慢.

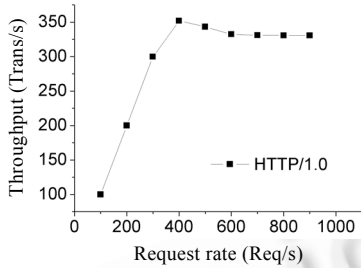


Fig.2
图 2

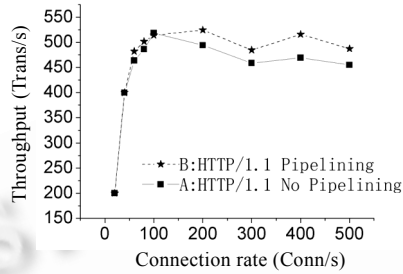


Fig.3
图 3

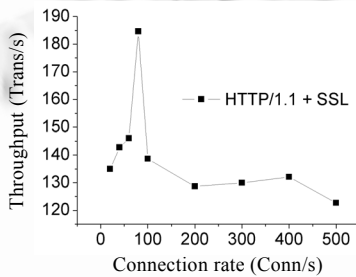


Fig.4
图 4

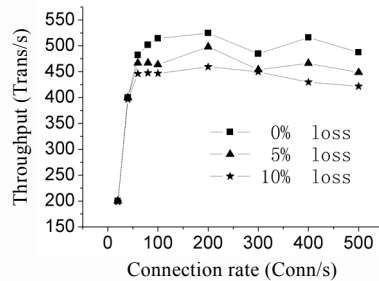


Fig.5
图 5

再次,进行和上面 B 组基本相同的实验,不同的是增加了 SSL.如图 4 所示的结果显示,SSL 对 Web 服务器的性能有很大的影响.当连接率为 80 的时候得到最大吞吐量 184.592Trans/s,是 B 组结果的 35.2%,且过载后 Web 服务器性能陡降.

最后,在丢包情况下进行测试,其他参数和 B 组实验相同.如图 5 所示,在丢包率为 5%时,最大吞吐量为 498.028Trans/s;在丢包率为 10%时,最大吞吐量为 459.336Trans/s;与没有丢包的 B 组实验的最大吞吐量 524.656 Trans/s 相比,分别下降了 5.08%和 12.45%.

5 同类工作比较

目前广泛采用的 Web 服务器性能测试工具有 WebStone^[13]和 SPECweb99,常见的还有 S-Client,SURGE 和 WebBench^[14].其中,WebStone 被看作第一代 Web 服务器性能测试工具,SPECweb99 被看作现行的工业标准.WebMark 和它们的功能对比见表 1.

以 WebStone,SPECweb 和 SURGE 为代表的一类方法,基本上是在前一个请求完成之后才发起新的请求,这就使 Client 和 SUT 步伐一致,所有 Client 的连接速率之和等于 SUT 的吞吐量.当 Client 的增加已使 SUT 满负荷时,再增加 Client 只能因 SUT 的接收队列变长而增加排队延迟,进而降低了 Client 的请求速率.所以,生成的请求最多能达到 SUT 的承受能力,但不会使 SUT 过载.

S-Client 可以产生过量的负载,但其功能过于简单,且每次都请求相同的静态文档,该文档会被缓存到内存中,每次访问都省去了磁盘 I/O 时间,明显地与 Web 服务器的典型行为不符.

Table 1 Function comparison among various tools
表 1 各种工具的功能对比

Function	WebMark	WebStone	SPECweb99	S-Client	SURGE	WebBench
Support HTTP/1.1	Y	N	Y	N	Y	Y
Support POST request	Y	N	Y	N	N	N
Support Cookie	Y	N	Y	N	N	N
Generate server overload	Y	N	N	Y	N	N
Support SSL	Y	N	N	N	N	Y
Support dynamic content	Y	Y	Y	N	N	Y
Representative workload	Y	N	Y	N	Y	N
Emulate Internet	Y	N	N	Y	N	N

WebMark 吸收了 S-Client 能生成过量负载的优点,克服了其功能过于简单及请求不具有代表性的缺点。

6 结 语

本文提出一个新的测试指标——加权响应时间,介绍了我们研制的一个 Web 服务器性能测试工具 WebMark。它用事件驱动的方式简化了异步 I/O 的管理;通过修改 Client 的 TCP/IP 协议栈来模拟 Internet 环境,从而避免了修改路由器带来的集中环节;能使 Web 服务器过载;负载有代表性。使用 WebMark 对 Apache 测试的结果充分证明了一个 Web 服务器性能测试工具具有这些功能和特点是非常必要的。

References:

- [1] Forum of Incident Response and Security Teams. New-Type software multiplies the performance of Web server 5 times. 2001. http://www.chinafirst.org.cn/new/more/d1.php?arc_id=5380 (in Chinese).
- [2] SPEC. SPECweb99 release 1.02. 2000. <http://www.spec.org/osg/web99/docs/whitepaper.html>.
- [3] Liu Z, Niclausse N, Jalpa-Villanueva C. Web traffic modeling and performance comparison between HTTP1.0 and HTTP1.1. In: Gelenbe E, ed. System Performance Evaluation: Methodologies and Applications. CRC Press, 1999.
- [4] Nielsen H, Gettys J, Baird-Smith A, Prud'hommeaux E, Lie HW, Lilley C. Network performance effects of HTTP/1.1, CSS1, and PNG. In: Proceedings of the SIGCOMM'97 Symposium. Cannes: Association of Computing Machinery, 1997. 155~166.
- [5] Netscape Communications Corporation. Introduction to SSL. 1998. <http://docs.iplanet.com/docs/manuals/security/ssl/index.htm>.
- [6] Coarfa C, Druschel P, Wallach DS. Performance analysis of TLS Web servers. In: Proceedings of the Network and Distributed System Security Symposium (NDSS 2002). San Diego, 2002.
- [7] Balakrishnan H, Padmanabhan V, Seshan S, Stemm M, Katz R. TCP behavior of a busy Internet server: Analysis and improvements. In: Proceedings of the IEEE Infocom'98. San Francisco: University of California, Berkeley and IBM T.J. Watson Research Center, 1998.
- [8] Stanford Linear Accelerator Center. Les Cotrell throughput versus loss. 2000. <http://www.slac.stanford.edu/comp/net/wan-mon/thru-vs-loss.html>.
- [9] Joi L. Chevalier, editorial director Internet average. Matrix Information and Directory Services, Inc., 2001. <http://average.miq.net/>.
- [10] Barford P. Modeling, measurement and performance of World Wide Web transactions [Ph.D. Thesis]. Boston University, 2001.
- [11] Banga G, Druschel P. Measuring the capacity of a Web server under realistic loads. World Wide Web Journal, 1999,2(1). <http://citeseer.nj.nec.com/banga99measuring.html>.
- [12] Mosberger D, Jin T. Httpperf—A tool for measuring Web server performance. In: SIGMETRICS Workshop on Internet Server Performance'98. Madison, 1998.
- [13] Trent G, Sake M. WebSTONE: The first generation in HTTP server benchmarking. 1995. <http://www.mindcraft.com/webstone/paper.html>.
- [14] ZD Labs. WebBench 4.1. 2001. <http://etestinglabs.com/benchmarks/webbench/webbench.asp>.

附中文参考文献:

- [1] 中国信息安全论坛. 新型软件使 Web 服务器性能提高 5 倍. 2001-02-20. http://www.chinafirst.org.cn/new/more/d1.php?arc_id=5380.