

# Web 服务器性能测量系统的研究进展<sup>\*</sup>

姚磊 戴冠中 张慧翔 潘文平

(西北工业大学自动化学院 西安 710072)

**摘要** 本文首先讨论了 Web 服务器测试的重要性及性能测试的一般原理;接着研究了现有的 Web 服务器测试系统,包括 WebStone、SPECWeb99、S-Client 和 SURGE;分析了它们的优点和不足之处;最后指出了 Web 服务器性能测试系统的发展趋势和需要进一步研究的问题。

**关键词** Web 服务器,测量系统,自相似

## Survey of Measurement Systems for Web Server Performance

YAO Lei DAI Guan-Zhong ZHANG Hui-Xiang PAN Wen-Ping

(College of Automation, Northwestern Polytechnical University, Xi'an 710072)

**Abstract** In this paper, we first discuss the importance and principles of Web Server testing, and then describe the existing Web Server test system, including WebStone, SPECWeb99, S-Client and SURGE. Their strengths and weaknesses are analyzed in detail. In conclusion, the development of Web Server performance test system and the further research aspects are pointed out.

**Keywords** Web server, Measurement systems, Self-similar

## 1 引言

随着基于 Web 的互联网应用越来越广泛,越来越多的工作都依赖于高质量的 Web 服务。如何提高 Web 服务器的性能、服务质量等问题成为现在研究的热点。为了解决以上问题,必须对 Web 服务器进行测试,取得测试结果,根据结果有目的地改进服务器的性能指标。所以,如何模拟互联网上用户的请求来测量服务器,是提高 Web 服务器性能的首要工作。

负载发生器(Workload Generator)可以模拟互联网上大量用户的请求,观测 Web 服务器对大量的、变化范围很大的负载处理情况。然后可以找到 Web 服务器存在的问题,为解决问题提供依据,并可以测试改进后的 Web 服务器性能。

## 2 性能测试系统的一般工作原理

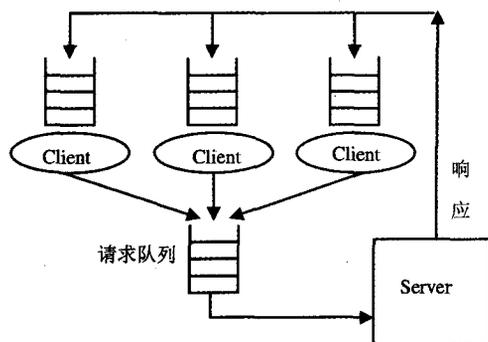


图 1 性能测试系统的一般工作原理

在互联网上,客户端与服务端是不对称的,数据的传输量也是不对称的。一个服务器服务于多个客户端。负载发生器

模拟的情况要与互联网的真实情况一致。一个负载发生器可以模拟多个用户,向一个服务器发出请求,如图 1 所示。

首先,服务器上要放置一些符合一定分布的文件,具体分布在后文中详述。负载发生器会模拟大量用户,请求服务器上的文件。服务器把响应结果返回到模拟的客户端。负载发生器要具有鲁棒性,例如对服务器上文件找不到等情况进行处理。

## 3 几种 Web 服务器测试系统

目前广泛采用的 Web 服务器性能测试工具主要有 WebStone 和 SPECweb99,常见的还有 S-Client 和 SURGE。

### 3.1 WebStone

WebStone<sup>[3,13]</sup>被看作第一代 Web 服务器性能测试工具。WebStone 是由 SGI 开发的网络测试软件,运行在 UNIX 环境下的开源(Open Source)软件。

#### 3.1.1 WebStone 的工作原理

WebStone 测试 Web Server 的两个性能指标:每秒请求数(Request per Second)和延迟时间(Latency)。WebStone 的工作原理如图 2 所示<sup>[3]</sup>。服务器上放置 32 个大小不同的文件,测试时对测试软件进行配置来模拟大量的客户端。模拟的客户端对服务器上的文件发出请求,从请求响应的结果中可以测量出服务器的性能指标。

WebStone 是通过模拟多客户行为在 Web Server 上产生负载的,其功能是从服务器上获取特定的文件。为测试 Web 服务器的性能,WebStone 采用一个 WebMaster 程序控制多个 WebClient 程序。WebMaster 是一个控制所有网络客户端的程序,可运行在客户计算机上或者任何独立的计算机上。WebStone 可以在一台计算机上运行超过 100 个的 WebCli-

<sup>\*</sup>西北工业大学研究生创业种子基金(Z200562)。姚磊 博士研究生,主要研究方向为网络化控制和流量控制;戴冠中 教授、博士生导师,主要研究领域为网络化控制、信息安全;张慧翔 博士研究生,研究方向为流媒体传输控制、拥塞控制;潘文平 博士研究生,研究方向为网络服务质量。

ent,而且为了在对服务器上产生足够的负载,可以将 WebClient 分布在同一网络中的不同客户计算机上。

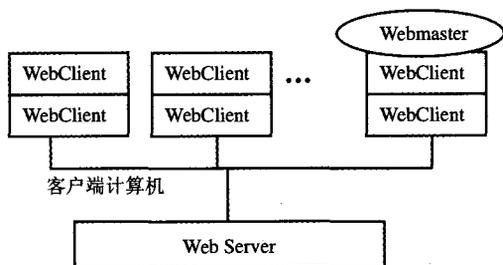


图2 WebStone 的工作原理

首先,WebMaster 产生出描述客户端的配置文件,并把这个文件传给所有的 WebClient。WebClient 根据配置文件创建多个子线程,每个线程产生 HTTP 请求,以一定的概率访问 Web 服务器上的某个文件。服务器返回结果后,子线程立刻开始下一次访问请求。所以,这种连续的访问是一种压力测试,可以得到 Web 服务器对重压的反应特性。

### 3.1.2 WebStone 的不足

WebStone 是一种压力测试工具,只是尽可能多地产生请求序列。文[4]指出,忽视 OFF Time,尽可能快地发送请求,不可能产生具有自相似特性的请求序列。

文[5]指出,WebStone 在服务器上放置 32 个文件。这些文件被一次次地传输,可能存在于 CPU 或文件系统缓存中,忽视掉磁盘的 I/O 时间,导致错误的结果。

对 Web 服务器测试的深度与广度不够。WebStone 的测试指标只有两个,反应不出平均完成一个请求需要的时间,而这个指标在评价电子商务之类应用的 Web 服务器是非常重要的。

并且 WebStone 只能处理静态页面,不支持对动态页面的访问。

### 3.2 SPECWeb99

SPECWeb99<sup>[6,14]</sup>是 SPEC(Standard Performance Evaluation Corporation,标准性能评测协会)开发的 Web 服务器性能基准测试,被看作现行的工业标准。

#### 3.2.1 SPECWeb99 的工作原理

SPECWeb99 测量的指标是:在预定的基准测试工作负荷下,仍然满足一定的吞吐量和错误率要求下最大的并发连接数。使用一个或多个客户系统为服务器产生工作负载。每个客户发送 HTTP 请求到服务器,然后验证接受到的应答。在 HTTP 请求序列发送完毕后,主客户搜集来自所有客户的数据,计算本次测试的结果。类似的过程进行 3 次,最后的测试报告就是 3 次测试结果的综合<sup>[6]</sup>。

在测试过程中,各 Client 上的进程在读来自 Server 的应答时,接收单元的大小不能超过 1460Bytes。加上长度为 40Bytes 的 HTTP 头,即最大传输单元 MTU (Maximum Transmit Unit) = 1500Bytes。最大段生存时间 (Maximum Segment Lifetime)被确定为 60s,因为缩短这个时间会导致人为地提高服务器的性能<sup>[15]</sup>。

对每个连接而言,只有在满足数据传输速率在 320~400kbps 之间的要求时,此连接才可计入最大同时连接数。

在一次测试中,只有当测出的最大同时连接数不小于指定最大同时连接数的 95% 时,才可以认为该次测试正确。

#### 3.2.2 SPECWeb99 的工作负荷

SPECWeb99 的工作负荷包括静态和动态的两种。

静态工作负荷模拟 Web 服务器中大量的不同页面。测试使用大小不同、且被访问概率也不同的文件代表真实 Web 服务器上的页面。文件的大小和访问概率是通过分析真实 Web 服务器上的数据得到的。工作负荷文件集包括了大量的目录,目录的数量由下式给出:

$$\text{目录数} = 25 + ((400000.0 / 122000.0) * \text{并发连接数}) / 5.0$$

每个目录中都包含 4 类文件,如表 1<sup>[6]</sup>所示。每类文件的中每个文件的访问概率符合 Zipf 分布<sup>[2,9,19]</sup>。即如果将文件按照被引用的次数  $p$  从高到低排名,则文件的排名  $r$  和被引用次数  $p$  有如下关系:  $p \approx kr^{-1}, k > 0$ 。

表 1 4 类文件的比例

文件大小	百分比 (%)
0~1kB	35.00
1~10kB	50.00
10~100kB	14.00
100KB~1MB	1.00

动态内容是网站页面中一个重要的部分<sup>[16~18]</sup>。动态工作负荷是模拟 Web 服务器的两个特点:广告和用户注册。动态负荷分为 4 类:标准的动态 GET(dynamic GET)、带用户广告的动态 GET(dynamic GET with custom ad rotation)、动态 POST(dynamic POST)、调用 CGI 代码的动态 GET(dynamic GET calling CGI code),它们占全部工作负荷的 30%。表 2<sup>[6]</sup>总结了不同类型请求在混合作业负荷中的比例。

表 2 不同类型请求占总负荷的比例

请求类型	百分比 (%)
Static GET	70.00
Dynamic GET	12.45
Dynamic GET with Custom ad rotation	12.60
Dynamic POST	4.80
Dynamic GET calling CGI code	0.15

#### 3.2.3 SPECWeb99 的局限性

虽然在连续请求之间引入了睡眠时间(OFF time),但是睡眠时间的长短并没有用类似 Pareto 分布或随机中点置换法 RMD 等可能产生自相似的模型来控制,故 SPECWeb99 中产生的请求序列与自相似相差甚远<sup>[7]</sup>。

SPECWeb99 主要测试 Web 服务器在保证一定数据传输率的前提下能维持的连接数,但是往返时间也是一个关心的重点。往返时间是完成每个请求所需的平均时间,并不是每秒请求数的倒数。

负载强度不够。由于产生的流量不具有自相似性,所以当客户机增多时(就是客户线程增多时)突发负荷将被吸收掉,从而导致服务器不能工作在明显的高负载条件下。

### 3.3 S-Client

S-Client<sup>[8,21,22]</sup>(Scalable Client)是 Rice 大学开发的一个 Web 服务器性能测试工具。它能产生突发的流量(bursty traffic),负载的峰值可以超出服务器的处理能力。可以观测在这种高负载的情况下 Web 服务器的性能指标,主要是服务器的吞吐量。

#### 3.3.1 S-Client 的工作原理

基本的测试结构如图 3<sup>[8]</sup>所示。P 个客户机连接到被测的服务器。每个客户机运行一定数量的 S-Client 进程。客户

机通过一个路由器(Router)与被测服务器连接。这个路由器必须有足够的容量通过可预计的最大流量。路由器的目的是在测试环境中引入人工的延迟来模拟 WAN 中的延迟。

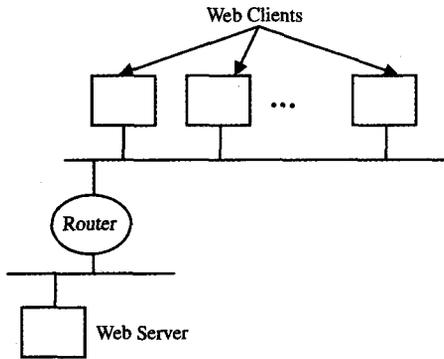


图 3 S-Client 测试结构

S-Client 与其它 Web 服务器测试工具最大的不同是：它是一个开环(open-loop)的系统。它在一个请求的响应还没有返回之前，就产生下一个请求。而其他工具都是在一个请求完成后再产生下一个请求。所以 S-Client 可以产生超出服务器能力的负载，给服务器更大的压力。

### 3.3.2 S-Client 的结构

一个 S-Client 中包含有一对进程，它们通过 Unix Domain Socketpair 连接。一个进程是连接建立进程(The Connection Establishment Process)，它的功能是以一定的速率和分布产生 HTTP 请求。在一个连接已经建立后，连接建立进程发送一个 HTTP 请求到服务器，然后把这个连接交给另一个进程：连接处理进程(The Connection Handling Process)。连接处理进程用来处理服务器端返回的 HTTP 响应。

连接建立进程的工作原理如下：此进程使用  $D$  个套接字以非阻塞模式(non-blocking mode)与服务器建立  $D$  个连接。这  $D$  个连接请求分布在  $T$  毫秒内。要求  $T$  大于在客户机和服务器之间的往返延迟(round-trip delay)。

连接处理进程的工作原理如下：此进程等待在活动连接中到达的数据和从连接建立进程传递过来的连接。如果到达的是数据，连接处理进程就读数据；如果完成了服务器的响应，就关闭套接字；如果到达的是连接，就把此连接放到活动连接集中。

为达到最大的连接请求率，可以采取减小  $T$  或增大  $D$  的措施。如前所述， $T$  必须大于客户机和服务器之间最大的往返延迟。给定  $T$  后， $D$  的最大值通常是由操作系统强制约束的，同时依赖于客户机的性能。一个具有很大  $D$  的 S-Client 很有可能使客户机瘫痪。

### 3.3.3 S-Client 的不足之处

文[8]中提到 Think Time 的分布，但是为了使连接的速率比较高，现在的 S-Client 并没有加入 Think Time 的分布。所以，S-Client 产生的 HTTP 请求不具有自相似的特性，只是一个高负载的 HTTP 请求序列。

在文[8]提出的试验环境中，S-Client 使用一个固定大小(1294 bytes)的文件作为服务器上的测试文件，所有的 HTTP 连接都是请求这个文件，服务器上文件的分布不符合实际情况。

## 3.4 SURGE

SURGE<sup>[4,20]</sup>(Scalable URL Reference Generator)模型是

由 Boston 大学的 Paul Barford 和 Mark Crovella 提出的。严格地说，SURGE 并不是一个 Web 服务器测试工具，它的目的是用来产生有代表性的 Web 负载(representative web workload)，而不是测试服务器的性能。SURGE 没有涉及到 Web 服务器性能测试指标，但是 SURGE 关于产生有代表性的 Web 负载的方法值得借鉴。SURGE 在产生 Web 负载时采用的是自相似模型。

在传统的 Web 服务器测试系统中，对网络流量采用的模型是无记忆的马尔可夫模型。这种模型认为，当前时间  $t$  与过去时间  $(t-s)$ ，若  $s$  足够大，则  $t$  与  $(t-s)$  时的业务量是不相关的，即仅考虑  $s$  较小时业务到达间的相关性，为短程相关(SRD)模型。而实际的网络业务流在很长的时间范围内都具有相关性，即业务到达是长程相关的(LRD)。因此，泊松过程已经不能恰当地描述网络流量特征<sup>[12]</sup>。当前网络业务更适宜采用具有长期相关性的模型，自相似模型就是其中的一个代表<sup>[1,2]</sup>。

在自然界中，自相似特性描述了事物的局部以某种方式与自身相似的现象。而在通信网络中，描述的是网络中离散的数据报文序列在时间尺度上具有统计特性相似的特点。我们给出长程相关过程和自相似过程的数学定义<sup>[1]</sup>。

考察一个广义平稳过程，

$$X=(X_t; t=0,1,2,3,\dots)$$

$X_t$  表示第  $t$  个单位时间里到达的网络业务实体的数目。记  $\mu = E[X_t]$ ,  $\sigma^2 = E[(X_t - \mu)^2]$ ，自相关函数

$$r(k) = E[(X_t - \mu)(X_{t+k} - \mu)] / \sigma^2$$

如果  $\sum r(k) = \infty$ ，则称平稳过程  $X$  具有长相关性(long-range dependence)；反之，则称平稳过程  $X$  具有短相关性(short-range dependence)。

令  $X_k^{(m)} = 1/m(X_{k-m+1} + \dots + X_k)$ ,  $k=1,2,3,\dots$ ，代表长度为  $m$  的聚集过程。对每个  $m$ ,  $X(m)$  都定义了一个广义平稳随机过程， $r(m)$  为其对应的自相关函数。

如果对所有  $m$  的均有：

$$r^{(m)}(k) = r(k) \sim k^{-\beta}, 0 < \beta < 1$$

当  $k \rightarrow \infty$  时，称  $X$  为精确二阶自相似过程，并称  $H = 1 - \beta/2$  为其自相似参数。自相似参数  $H$  又称为 Hurst 参数，是描述自相似特性的唯一参数。 $H$  的取值范围是  $(0.5, 1)$ 。 $H$  越大，过程自相似程度越高。

另一个与自相似性有关的概念是重尾分布(Heavy-tail Distribution)。分组到达时间间隔或突发数据长度的重尾分布是网络自相似特征的主要物理根源。重尾分布的定义如下：若一随机变量满足重尾分布，则

$$P[X > x] \sim x^{-\alpha}$$

当  $x \rightarrow \infty, 0 < \alpha < 2$ 。最简单的重尾分布是 Pareto 分布<sup>[2]</sup>，其概率密度函数为

$$p(x) = \alpha k^\alpha x^{-(\alpha+1)}, x \geq k > 0$$

当  $\alpha$  减小，大量的概率质量集中在分布的尾部，其自相似参数  $H = (3 - \alpha)/2$ 。有关自相似、重尾分布及 Pareto 分布的详细讨论，可参考文献[10]。

假设网络中的一个业务源，在发送数据和不发送数据两种状态间交替转换。在发送状态(称为 ON 期间)以固定的速率发送数据，而在不发送数据状态(称为 OFF 期间)保持空闲。这种业务源称为 ON/OFF 源<sup>[1]</sup>。文[11]给出证明，如果 ON 和 OFF 时间长度为相互独立且均服从 Pareto 重尾分布

(下转第 99 页)

- ings of the 2002 International Conference on Data Engineering, 2002
- 14 Ives Z, Levy A, Weld D. Efficient Evaluation of Regular Path Expressions on Streaming XML Data; [ Technical report]. University of Washington, 2000
  - 15 Lakshmanan L V S, Parthasarathy S. On Efficient Matching of Streaming XML Documents and Queries. In: Proceedings of ED-BT, 2002
  - 16 Sarvega's XML Content Router. Sarvega, INC. 2005, <http://www.sarvega.com/xml-routing.html>
  - 17 Chan Chee-Yong, Felber P, Garofalakis M, et al. Efficient Filtering of XML Documents with XPath Expressions. In: Proceedings of the 18<sup>th</sup> International Conference on Data Engineering (ICDE'02), 2002
  - 18 Green T J, Miklau Miklau, Onizuka M, et al. Processing XML Streams with Deterministic Automata. In: Processing of ICDDT, 2003. 173~189

- 19 Onizuka M. Light-weight XPath Processing of XML Stream with Deterministic Automata. In: CIKM'03, New Orleans, Louisiana, USA, November 2003
- 20 Gupta A K, D Gupta. Stream Processing of XPath Queries with Predicates. In: SIGMOD 2003, San Diego, CA, June 2003
- 21 Qlteanu D, Kiesling T, Bry F. An Evaluation of Regular Path Expressions with Qualifiers against XML Streams. In: Proceedings of the 19<sup>th</sup> International Conference on Data Engineering (ICED'03), IEEE Computer Society, 2003
- 22 高军, 杨冬青, 唐世渭, 等. 基于树自动机的 XPath 在 XML 数据流上的高效执行. 软件学报, 2005, 16(2): 223~232
- 23 Bruno N, Koudas N, Srivastava D. Holistic Twig Joins: Optimal XML Pattern Matching. In: Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, 2002
- 24 潘志庚, 姜晓红, 等. 分布式虚拟环境综述. 软件学报, 2000, 11(4): 461~467

(上接第 88 页)

的随机变量, 则叠加大量的 ON/OFF 源就可以导致网络流量呈现出自相似的特性。

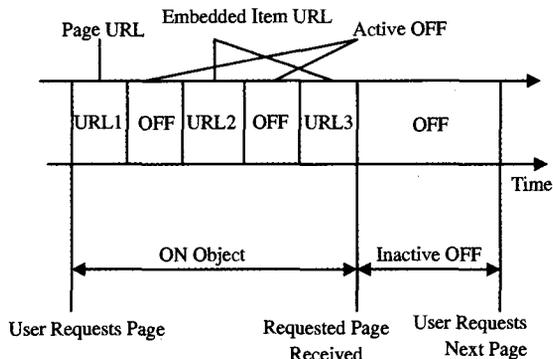


图 4 SURGE 中使用的 ON/OFF 模型

SURGE 就是使用大量的 ON/OFF 源叠加, 生产具有自相似特征的 HTTP 请求序列。图 4<sup>[4]</sup> 是 SURGE 中使用的 ON/OFF 模型。

#### 4 现有 Web 服务器测试系统存在的问题

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 (System Under Test, SUT 被测系统) 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 只能使 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, 访问可能省去磁盘 I/O 时间, 与 Web 服务器的典型行为不符。

**总结** 本文分析了 Web 服务器的性能测试原理, 研究并阐述了网络流量的自相似特征。这些特征是测试工具要考虑的。对 Web 服务器性能进行评价, 可以发现存在的问题, 以便进行分析和改进。所以, 掌握 Web 服务器的性能测试方法很有必要。

本文所论述的 Web 服务器测试工具都有各自的优点, 但也存在一定的局限性。今后的主要工作可以放在具有请求序列高负载、自相似, 且服务器上文件的分布符合实际情况的 Web 服务器测试工具研究上, 这样才能够测量出 Web 服务器在真实环境下的性能。

#### 参考文献

- 1 Leland W E, Taqqu M S, Willinger W, et al. On the Self-Similar Nature of Ethernet Traffic. IEEE/ACM Transactions on Networking, 1994, 2(1): 1~15
- 2 Crovella M E, Bestavros A. Self-similarity in word wide web traffic: Evidence and possible cause. In: Proceedings of the 1996 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, 1996
- 3 <http://www.mindcraft.com/webstone/>
- 4 Barford P, Crovella M. Generating Representative Web Workloads for Network and Server Performance Evaluation. In: Proceedings of Performance '98/ACM SIGMETRICS '98, 1998
- 5 Hu Yiming, Nanda A, Yang Qing. Measurement, Analysis and Performance Improvement of the Apache Web Server. In: 18<sup>th</sup> IEEE International Performance Computing and Communications Conference, Phoenix, 1999
- 6 Standard Performance Evaluation Corporation. <http://www.spec.org/osg/web99/>
- 7 欧灿辉, 李晓明. Web 服务器性能评测. 计算机研究与发展, 2002, 39(5): 540~547
- 8 Banga G, Druschel P. Measuring the Capacity of a Web Server. In: Proceedings of the USENIX Symposium on Internet Technologies and Systems, California, 1997
- 9 Padmanabhan, V N, Qui Lili. The content and access dynamics of a busy Web site: Findings and implications. In: ACM SIGCOMM Symposium on Communications Architectures and Protocols, 2000. 111~123
- 10 Stallings W. High-Speed Networks TCP/IP and ATM Design Principles. Englewood Cliffs, NJ: Prentice Hall, 1998
- 11 Willinger W, Paxon V, Taqqu M S. Self-Similarity and Heavy Tails: Structural Modeling of Network Traffic [M]. In: A Practical Guide to Heavy Tails: Statistical Techniques and Applications, Birkhauser, Boston; Chapman & Hall, 1998. 27~53
- 12 Paxon V, Floyd S. Wide area traffic: the failure of Poisson modeling. IEEE/ACM Transactions on Networking (TON), 1995; 226~244
- 13 Trent G, Sake M. WebSTONE: The First Generation in HTTP Server Benchmarking. <http://www.mindcraft.com/webstone/server.html>, 1995
- 14 Nahum E. Deconstructing SPECWeb99. In: Proceedings of the 7th International Workshop on Web Content Caching and Distribution, 2002
- 15 Aron M, Druschel P. TCP implementation enhancements for improving Web server performance: [Technical Report TR99-335]. Computer Science Dept, Rice University, 1999
- 16 Amza C, Cox A, Zwaenepoel W. Scaling and availability for dynamic content web sites: [Technical Report TR02-395]. Rice University, 2002
- 17 Challenger J, Iyengar A, Witting K et al. A publishing system for efficiently creating dynamic Web content. In: Proceedings of the Conference on Compute Communications (IEEE Infocom), Tel-Aviv, Israel, March 2000
- 18 Iyengar A, Challenger J. Improving Web server performance by caching dynamic data. In: Proceedings of the USENIX Symposium on Internet Technologies and Systems, Monterey, CA, 1997
- 19 Arlitt, M F, Williamson C L. Internet Web servers: Workload characterization and performance implications. IEEE/ACM Transactions on Networking, 1997, 5(5): 631~646
- 20 Barford P, Crovella M. A performance evaluation of hyper text transfer protocols. In: Proceedings of the ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems, Atlanta, GA, 1999
- 21 Banga G, Druschel P. Measuring the capacity of a web server under realistic loads. World Wide Web Journal (Special Issue on World Wide Web Characterization and Performance Evaluation), 1999
- 22 Banga G, Mogul J C. Scalable kernel performance for internet servers under realistic loads. In: USENIX Annu Tech Conf., New Orleans, LA, 1998