Web 集群请求分配与选择算法性能评价模型的研究

刘安丰　陈志刚　阎朝坤　李　登
（中南大学信息科学与工程学院　长沙 410083）

摘　要　请求分配和选择算法是 Web 集群技术的重要研究内容。本文基于当前世界上在 Web 集群请求分配与选择算法研究上的主要结果，建立起一个系统的 Web 集群请求分配与选择算法的性能评价体系，提出了评价指标参数，并详细介绍了 Web 集群调度模拟器（Web Cluster Schedule Simulator，WCSS）的设计模型和实现方法，最后以作者提出的基于资源优化的调度策略进行了模拟，实验结果较为理想。

关键词　Web 集群，请求分配与选择算法，评价模型，集群调度模拟器

Research on the Evaluation Model for the Request Dispatching and Selecting Algorithm in Web Cluster

LIU An-Feng　CHEN Zhi-Gang　YAN Chao-Kun　LI Deng
（College of Information Science and Engineering，Central South University，Changsha 410083）

Abstract　Request dispatching and selecting algorithm is important to the research of Web cluster. Based on current research on request dispatching and selecting algorithm of Web cluster，a systemic architecture of performance evaluating of request dispatching and selecting algorithm of Web cluster is argued. Some criterion evaluating is introduced. The module designing and method realizing of Web cluster scheduling simulator is introduced in detail. We have simulated the scheduling policy based on resource optimizing and got ideal results.

Keywords　Web cluster，Request dispatching，Selecting algorithm，Evaluating model，Web cluster schedule simulator

1　引言

近十年来，Internet 在传播信息的范围和数量上都呈指数级增长，其中占绝大部分的是 Web 服务。这给 Internet 上的各种 Web 服务器提出了前所未有的挑战。比如 AOL 的 Web 缓存每天服务超过一百亿次点击[1]。同时，HTTP 请求经常以爆发的方式到达 Web 服务器[2]。文[3]指出，高峰时的 HTTP 请求率超过平均值的 8～10 倍，这时的负载一般超过 Web 服务器的负载能力，迫切需要 Web 服务器具有强大的系统处理能力。基于单一系统映像的 Web 服务器集群系统是满足强大处理能力的一种解决方法。这种方法把若干性能较低的服务器用局域网连接成一个性能较高的整体：即 Web 集群服务器，使其从客户端看来就如同一台服务器。在现有的服务器集群技术中，存在 3 种体系结构：基于 DNS 的、基于前端 Dispatcher（分发器）的、基于后端服务器的，其中基于 Dispatcher 的服务器集群的性能被证明是最好的[4]。它的基本结构如图 1 所示。

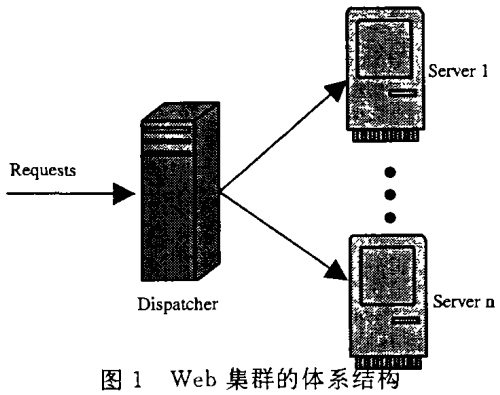

图 1　Web 集群的体系结构

请求分配和选择算法[5~9]是 Web 集群 QoS 控制技术的重要研究内容。其主要原理是将接收到的所有 HTTP 请求依据一定的原则把它们分配到集群中的各后台服务器上去进行处理，使各后台服务器的负载分布比较均衡，从而获得较高的整体吞吐能力和较快的反应速度。选择的目的是将已经分配到某服务器的请求按一定的策略调度以达到使系统性能优化的作用。一个 Web 集群性能评价模型包括：集群的体系结构研究，请求的分布模型分析，集群的调度策略，集群的服务质量（QoS）要求评价，如对集群服务延迟、吞吐量，反应速度等的要求。请求分配和选择方案算法也就是调度算法的优劣直接影响集群服务质量，因此，研究如何评价集群系统调度策略算法性能是至关重要的。

建立集群系统实验室来测试集群的性能十分重要，但这种方法有一些缺点[10]：建立大规模集群实验环境需要巨额资金，同时难以更改配置和共享，甚至有些新的思想和方法在现实集群环境下无法测试。采用建立 Web 集群测试模拟系统的方法虽然会丢失一些细节信息，但其具备更好的灵活性和共享性。而在实际中很多研究也是采用模拟 Web 集群来求证的[11]。

2　改进的请求分配和选择算法性能评价指标

目前，评价 Web 集群调度算法性能主要采用每秒连接数和每秒传输字节数。毫无疑问，每秒连接数是用户最关心的性能指标。但只有这一个指标是不够的，有时候用户更关心往返时间，即平均每请求完成所需的时间，请注意往返时间并不是每秒请求数的倒数。另一方面，对 Web 服务器的性能评测深度也不够，没有考虑动态负载加数据库存取这种情况，而现在 Web 服务器后端连接一个或多个数据库服务器的模式越来

*）基金项目：国家留学回国人员基金（教外司留[2000]479 号）；湖南省自然科学基金（02JJY2097）。**刘安丰**　博士研究生，主要研究方向为高性能 Web 集群系统，系统性能优化，Web QoS。**陈志刚**　博士，博士生导师，主要研究方向为网络计算与分布式处理。

越流行，动态负载加数据库存取(通过 CGI)所占的比重将越来越大。显然这类动态请求对系统性能负载很大。文[4]指出动态请求的负载是静态请求的 100 倍。因此，单靠以上指标难以反映系统性能。

为了更加准确地评价和比较 Web 集群调度算法的性能，我们提出了六种测量集群配置和调度算法的有效性的参数。这些参数都是根据集群模拟器所测得的 6 个统计数字计算而来的。依据不同的集群拓扑、不同的配置参数、不同的调度算法，通过集群模拟测试器可以得到在整个模拟期间 6 个统计数据：(1)＃Requsets，表示所到达的 Web 请求数目；(2)＃Successes，表示请求顺利完成的 Web 请求数目；(3)＃BeginRtime，表示请求到达时间；(4)＃EndRtime，表示请求完成时间；(5)＃ BPS(bytes per second)，每秒字节数，BPS 定义为单位时间内(1s)完成的请求传输的数据量(以字节记)，包含发送的请求、HTTP 头及接收的所有数据；(6)＃Rclass，每类请求的个数。

从模拟测试系统中可以获取以上 6 个统计数据，通过这 6 个参数就可以计算出几乎所有 Web 集群调度算法的性能：

(1) 成功完成请求率：

SRR＝＃Successes/＃Requests；

(2) 平均应答延迟：

MRD＝Σ(＃EndRtime－＃BeginRtime)/＃Requsets；

(3) 区分服务比率：DS＝＃Rclassi/Σ＃Rclass。

很明显可知，这些参数反映了 Web 集群系统调度算法的有效性、集群负载、集群体系结构等有效性等集群性能指标。例如：当集群负载很重时无论采用什么调度算法，MRD 必然较高，每秒完成请求数必然高，但是在以前的研究中，没有采用区分服务比率：DS，如果在一段时间动态请求比例很高，那么每秒处理连接率和吞吐率会很低，但系统负载大，相比，少的静态请求也会产生大的每秒请求率和吞吐率，负载并不大。因此我们提出的指标在一定程度上全面反映了系统调度算法的好坏。

3 Web 集群调度模拟器 WCSS 的设计

集群模拟器在研究和测试新的集群与协同计算技术方面有巨大的作用，并且避免了建立集群测试平台的高成本和复杂性。模拟器能够评价在不同集群体系结构下的集群性能，为集群与协同计算新技术的研究提供可靠的实验依据。一个好的模拟系统[10]应具有以下特性：

(1)能够支持大规模数据的测试。由于模拟系统主要通过统计特性反映集群性能，因此需要测试大量数据来分析其统计特性，例如：在分析集群连接率时，网络请求数应在 30 万左右[10]。

(2) 测试数据易修改。集群的性能与许多测试数据参数存在直接的关系，在模拟期间，经常需要调整 Web 集群参数，因此模拟器中应提供测试参数易于修改的途径。

(3)可扩充。Web 集群模拟器需不断补充新的功能模块，添加新的测试算法，因此必须提供良好的扩充编程接口；

(4)可视化界面。网络模拟器应能为研究者提供可视化界面，方便研究者分析实验结果，并能动态跟踪模拟的整个过程。

集群模拟器 WCSS 的设计主要是为了测试不同调度算法在不同集群参数下的性能指标。主要集群参数包括集群的拓扑、主机状态更新策略、请求流的特性及分布、性能评价指标、选择算法、调度算法、集群接入控制策略等。WCSS 的模型如图 2 所示。

集群模拟器 WCSS 主要包括集群结构生成器、集群负载生成器、集群模拟处理器和模拟结果收集统计分析器。

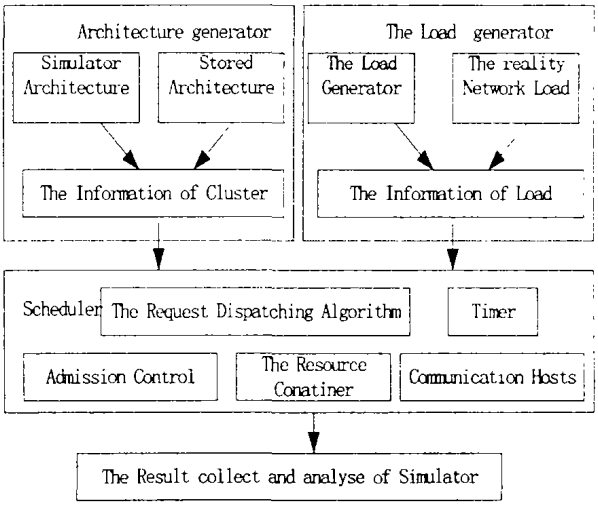

图 2 集群模拟器 WCSS 的模型

集群拓扑生成器主要功能是产生集群结构并构造初始集群状态信息。主要包括两种产生方式：人工产生拓扑和实际集群的拓扑结构。实际集群拓扑结构主要包括前端分发器，后台服务器(进行选择调度)及它们之间的组成关系等，集群拓扑信息库保存这些集群拓扑结构，在模拟过程中使用。人工产生拓扑主要是根据某种需要构造集群拓扑结构或随机产生，其目的主要是评价集群拓扑对集群性能的影响。

集群负载生成器负责产生整个模拟期间的集群请求。Web 集群负载生成器对调度算法评估有直接的重要作用。在这一方面，研究人员对大量 Web 请求数据进行比较精确的统计分析。Crovella 等人对 Web 业务的自相似性做了一系列的研究，并且通过对客户端 Web 请求踪迹(通过修改浏览器代码记录用户请求)的分析，提出了基于 Web 服务器端文档大小分布，用户对不同文档的喜好，用户思考时间("think"time)和缓存(cache)策略影响的一个解释[12,13]。Arlitt 和 Williamson[14]通过对 6 个不同层次服务器端日志的分析，总结了 10 条 Web 负载的特点。通过对代理服务器日志的分析统计，Abdulla[15]归纳了 9 条 Web 负载的特点。当然要模拟这些负载分布是很复杂的，我们采取 Web 请求踪迹的办法来产生负载(可从网上下载)。

集群模拟处理器是整个模拟器的核心部分，主要包括调度分发器、计时器、资源释放处理器、资源预留处理器、后台服务器状态更新处理器。调度分发器作为所有调度策略和算法的控制器，依赖于调度选择参数，为每个 Web 请求寻找合适的处理服务器。同时它实现了不同的调度策略，包括请求分配算法，如，转轮(round-robin)[5,6]、最少连接优先(least connections first)[5,7]、快速反应优先(faster response precedence)[5,7]和选择加权百分率(selected weighted percentage)[5,8]等。请求选择方案有随机均衡选择(random selecting)、列优先级(head of line)和队列长度阈值(queue length threshold)等[9]。计时器提供了调度计算所需的时钟信息。资源释放处理器和资源预留处理器利用请求消耗来实现资源的控制。后台服务器状态更新处理器的后台服务器信息中保存了后台服务器的基本信息，也是调度算法的依据。由于后台服务器状态信息是不断变化的，需要进行更新，而更新策略直接影响了状态信息的准确性和协议的开销。一般要说，要求后台服务器状态准确度高那么协议开销越大；相反，协议开

销小后台服务器状态信息不准确。目前主要的更新策略[16]有：

(1)定时更新。基于门限的更新方法，设定某一固定的常数 T 周期内更新；

(2) 事件触发更新。事件触发后向控制器报告方式。

模拟结果收集统计分析器的主要功能是收集在不同网络参数条件下(1)#Requsets (2)#Successes (3)#BegunRtime (4)#EndRtime (5)# BPS(bytes per second)的值，分析不同条件下的 Web 集群性能，为优化集群性能提供实验依据。

4 集群模拟器 WCSS 的实现

集群模拟器 WCSS 采用 VC++6.0 在 Windows98 环境下开发实现。网络模拟器 WCSS 的工作流程如图 3 所示。

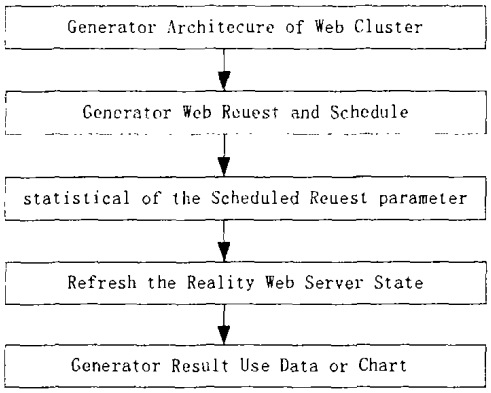

图 3 模拟器的工作流程

5 实验结果

为了证实我们的模拟器对不同调度算法模拟的有效性，我们做了如下实验：在我们的实验中采用排队论进行分析，首先产生负指数分布数据。但在测试前我们先用真实 Web 集群对算法进行了测试。然后，我们用模拟器 WCSS 做了测试后进行对比。

5.1 模拟数据产生

由于在排队分析中，服务时间和到达间隔均服从负指数分布，因此我们必须能产生服从指数分布的随机数。但计算机只能产生均匀分布的随机数，我们必须通过一些变换得到服从指数分布的随机数。具体过程如下：

假定要产生服从参数为 λ 的负指数分布的随机数，设 ζ 是[0,1]上均匀分布的随机变量，$F(x)$为任意分布函数，且存在逆函数 F^{-1}。

记 $\eta \equiv F^{-1}(\zeta)$，则 $P\{\eta \leqslant x\} = P\{F^{-1}(\zeta) \leqslant x\} = P\{\zeta \leqslant F(x)\} = F(x)$。

所以 η 为具有分布函数 $F(x)$的随机变量。由于 $1-\zeta$ 仍然是[0,1]均匀分布的随机变量，所以 $F^{-1}(1-\zeta)$的分布函数也是 $F(x)$ 。

特别地，令

$$F(x)=\begin{cases}1-e^{-\lambda x} & x>=0\\ 0 & x<0\end{cases}$$

它的逆函数为

$$F^{-1}(y)=-\frac{1}{\lambda}\ln(1-y)$$

令 $\eta=-\frac{1}{\lambda}\ln(1-\zeta)$

故 η 是具有负指数 $F(x)$分布的随机变量。

5.2 资源优化算法介绍

从 Web 服务的发展过程中可以发现，采用多处理机(multiprocessor)技术和多线程(multithread)技术是服务器发展的必然趋势。一个进程可以包含多个线程，各线程处理客户请求时采用分时共享 CPU 技术，当一个线程挂起时，同进程中的其它线程可继续进行，使得系统资源可以真正得到并行利用。

近来的研究一般将 Web 集群系统抽象为一个资源容器[17]，而近几十年来的各种调度处理研究表明，任务的调度运行是以资源的满足为前提的，如果能恰当调度请求分配和选择之间的关系，使系统资源能同时(并行)得到利用，无疑这时系统性能最高。

本文基于以上认识，提出了基于资源优化请求调度策略。依据这个调度策略，一个 Web 请求同时需要多维资源，如占用 CPU，I/O 和内存等资源，而不相同的请求对各种资源的需求是不相同的，例如：动态请求对 CPU 资源需求多，而对 I/O 需求相对较少，对于静态请求，主要是得到一个文件，对 I/O 资源要求较多，对 CPU 资源要求较低少，如果采用分离式调度策略[18]：将静态请求放在一起，而将动态请求放在一起调度，那么，在调度静态请求时，CPU 主要在一个周期内只要进行一次 I/O 中断即可，大部分时间空闲，相反动态请求时，CPU 资源紧张，而 I/O 资源利用不足。基于资源优化的调度算法就是以资源的优化为目的，即先计算各请求对各种资源的需求，再进行合理的调度使系统资源得到最大程度的并行利用，使系统性能最高。

我们用真实服务器做了上面提出的资源优化调度算法(ROSA)与分离式调度算法(SS)的对比测试，得到如下数据：

表 1 资源优化调度算法与分离式调度算法对比结果

方案		实施策略描述	完成指标
方案一 20 个动态页面，40 个静态页面	资源优化调度	采用先发二个静态页面，再发一个动态页面，循环 10 次	15 分 46 秒
	分离式调度	先全部发送静态页面，后发送动态页面，循环 10 次	16 分 2 秒
方案二 20 个动态页面，80 个静态页面	资源优化调度	先发 4 个静态页面，再发 1 个动态页面，循环 10 次	25.632 次/秒 用时 18 分 52 秒
	分离式调度	先发 80 个静态页面，再发 20 个动态页面，循环 10 次	25.169 次/秒 用时 19 分 24 秒

执行结果如图 4 所示。

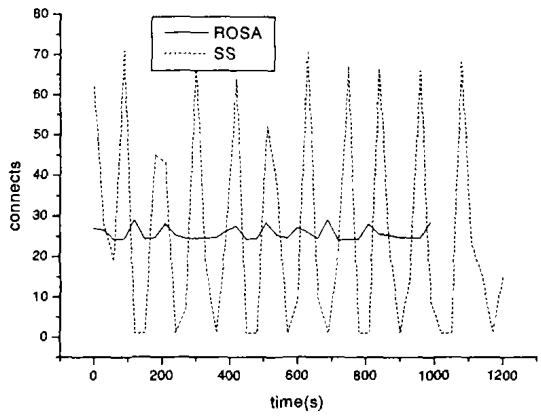

图 4 ROSA 与 SS 对比的真实结果

(下转第 69 页)

下面由隶属度矩阵计算排序权向量。对 B_3-C 隶属度矩阵，确定标准 V_1、V_2、V_3 的权重依次为 1/9、3/9、5/9，按照公式 $\boldsymbol{B}=\boldsymbol{A}\cdot\boldsymbol{R}'$ 求得各风险因素在准则 B_3-"不可控制性"下的相对权重为(0.4,0.422,0.311,0.333,0.378,0.311)，归一化后得到排序权向量为(0.186,0.196,0.144,0.155,0.175,0.144)。

计算各层权重乘积之和即得到各风险因素综合重要度，见表 4。

表 4 综合重要度

C_i	B_1	B_2	B_3	$W_i=\sum_{j=1}^{3} w_j \cdot w_{bj}$
	0.333	0.592	0.075	
C_1	0.212	0.199	0.186	0.203
C_2	0.127	0.164	0.196	0.154
C_3	0.154	0.152	0.144	0.152
C_4	0.188	0.117	0.155	0.143
C_5	0.209	0.212	0.175	0.210
C_6	0.112	0.158	0.144	0.142

根据以上分析，最后得到风险因素 C_1、C_2、…、C_6 的综合重要度分别为 0.203、0.154、0.152、0.143、0.210、0.142，风险较大的为 $B5$ 和 $B1$。要采取的风险保护措施首先必须控制未经授权而修改数据和软件并控制未授权的网络访问。

结束语 本文对网络安全风险评估的方法，考虑到了网络安全风险因素的不确定性和多变性，采用 AHP 法和模糊逻辑法相结合的方法，将风险因素分别从风险概率、风险影响和不可控制性方面进行专家评定，并用模糊逻辑法确定其权重，是一种将主客观相结合的方法，将主观评估做定量描述，可以对风险因素做比较客观的评估。

参 考 文 献

1 Baltimore. CMS Information Security Risk Assessment Methodology. Sep. 2002

2 Mustafa M A, FAI-Bahar J. Project risk assessment using the analytic hierarchy process[J]. IEEE Transactions on Engineering Management, 1991, 38(1): 46～52

3 That JHM, Carr V. A proposal for construction project risk assessment using fuzzy logic[J]. Construction Management and Economics, 2000, 18: 491～500

4 钟登华，张建设，曹广晶．基于 AHP 的工程项目风险分析方法．天津大学学报，2002，35(2)：162～165

5 李洪兴，汪培庄．模糊数学[M]．北京：国防工业出版社，1993

6 许树柏．层次分析法原理．天津：天津大学出版社，1988

(上接第 59 页)

5.3 与其它算法对比结果

采用 WCSS 得到如下对比结果。从对比结果来看，在系统未超过过载点时，二者的性能基本相近，但过超载点后，资源优化调度算法的错误连接率要好。与我们上面的真实结果较为符合。证实了 WCSS 对真实 Web 集群模拟的逼近性。

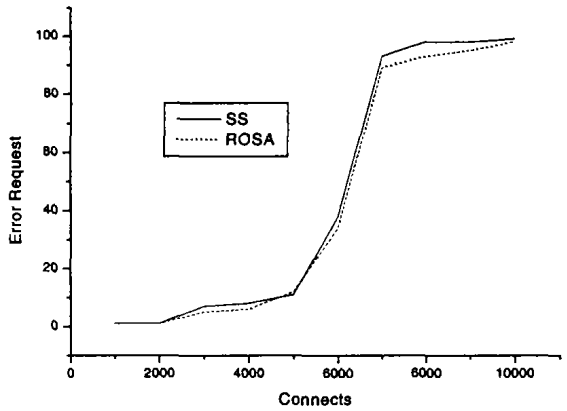

图 5 ROSA 与 SS 对比的模拟结果

结束语 本文主要探讨了 Web 集群调度算法的性能评价体系和集群模拟器 WCSS 的设计与实现，为 Web 集群调度算法的研究提供了很好的研究平台。进一步的研究工作主要包括支持各种不同 QoS 调度算法性能评价。

参 考 文 献

1 AOL. America Online Press Data Points. 2002. http://corp.aol.com/press/press_datapoints.html.

2 Crovella ME, Bestavros A. Self-Similarity in World Wide Web traffic: Evidence and possible causes. IEEE/ACM Transactions on Networking, 1997, 5(6): 835～846

3 Mogul JC. Network behavior of a busy web server and its clients. Technical Report, WRL 95/5, Palo Alto: DEC Western Research Laboratory, 1995

4 Cardellini V, Casalicchio E, Colajanni M, Yu P S. The state of the art in locally distributed Web-server systems. ACM Computing Surveys, 2002, 34(2): 1～49

5 Colajanni M, Yu P S, Cardellini V. Dynamic load balancing in geographically distributed heterogeneous web servers. In: Proc. of the 18th IEEE Intl. Conf. on Distributed Computing Systems (ICDCS'98). Amsterdam IEEE Computer Society, 1998. 295～302

6 Katz E D, Butler M, McGrath R. A scalable HTTP server: the NCSA prototype. Computer Networks and ISDN Systems, 1994, (28): 155～164

7 Cisco Systems Inc. Load balancing : amultifaceted solution for improving server avalability. Whitepaper, 2000. http: //www.cisco.com /warp/public/cc/pd/cxsr/400/tech/lobal-wp.htm.

8 Crovella M E, Carter R L. Dynamic server selection in the Internet: [TechnicalReport, TR-95-014]. Department of Com-puter Science, Bostor University, 1995. http://cs-www.bu.edu/faculty/crovdla/paper-archive/hpcs95/paper.html.

9 Lin, Chuang. Performance analysis of request dispatching and selecting in web server clusters. Chinese Journal of Computers, 2000, 23(5): 500～508

10 Estrin. Improving simulation for network research [R]: [TechRep. 99-702b]. University of Southern California Sep. 1999

11 李双庆，古平，程代杰．Web 集群系统负载均衡策略分析与研究，计算机工程与应用，2002，38(19)：40～42，64

12 Stallings W. High-Speed Networks TCIP/IP and ATM Design Principles. Englewood Cliffs, NJ: PrenticeHall, 1998

13 Cunha C, Bestavros A, Mcrovella. Characteristics of WWW client-based traces. Department of Computer Science of Boston University, [TechRep: TR-95-010]. 1995

14 Arlitt M, Williamson C. Web server workload characterization: The search for invariants (extended version). In: ACM SIGMETRICS'96. Marriott, PA, 1996

15 Abdulla G, Fox E, Abrams M. Shared user behavior on the world wide web. In: ProcWebNet97, Association for the Advancement of Computing in Education (AACE). 1997. URL. http://www.aace.org/

16 Shaikh A. Efficient dynamic routing in wide-area networks[D]: [PhD thesis]. University of Michigan, 1999

17 Banga G, Mogul J C. Resource containers: A new facility for resource management in server systems [J]. In: Proc. of the Third Symposium on Operating Systems Design and Implementation (OSDI'99), New Orleans, LA, Feb. 1999

18 雷迎春，张松，等．Web 集群服务器的分离式调度策略．计算机研究与发展，2002，39(9)：1093～1098