一种新的面向广域 Web 服务器集群的动态负载平衡算法

李 捷, 刘景森, 刘先省
(河南大学计算机与信息工程学院, 河南开封 475004)

摘 要: 由于地理意义上的分散性和系统的动态变化性, 群集间的负载分配成为了一个棘手的问题. 首先给出了一种多量纲参数的融合策略, 然后提出了一种动态负载平衡算法. 通过对广域 Web 服务器集群环境建立数学模型, 设计并实现了试验平台. 仿真结果表明, 该算法可以满足系统的负载平衡需求, 因此较大程度地减少重定向引起的延时以及提高系统的准入概率.

关键词: Web; 服务质量; 集群; 负载平衡
中图分类号: TP393 文献标识码: A 文章编号: 0372-2112 (2007) 12-2425-05

Research on the Load Balance Strategy of Web Server Clustering

LI Jie, LIU Jing-sen, LIU Xiarr-xing
(*College of Computer & Information Engineer, Henan University, Kaifeng, Henan 475004, China*)

Abstract: The exponential growth of the Internet coupled with the increasing popularity of dynamically generated content on the World Wide Web, has created the need for more and faster Web servers capable of serving the over 100 million Internet users. Server clustering has emerged as a promising technique to solve the problem. In this paper, . the defects of the existing Web Server Clustering technologies are summarized firstly. To attain the goal of keeping the balance of the server cluster, an syncretic strategy is advanced for hybrid heterogeneous parameters, through which a load-assignment algorithm is presented. A testbed is implemented by modeling the Web Server clustering. The simulation result shows that the presented scheme can guarantee the load balance and thus can provide better QoS guarantees.

Key words: Web; QoS; clustering; load balance

1 引言

Internet 上 Web 应用和 HTTP 请求的爆炸性增长, 使得目前的 Web 站点经常面临服务器超载的问题, 集群技术[1]被认为是解决这一问题的有效手段. 已有的 Web 服务器集群系统可以分为由局部范围内的多台服务器组成的局域集群和多个局域集群在地理位置上广域分布而形成的广域集群. 对于 Web 服务器集群而言, 有效的负载均衡策略对其实现高性能和为用户提供 QoS 性能保证具有决定性作用, 而合理的负载分配机制则是实现负载均衡的前提[2]. 对于局域集群, 集中式的基于请求分配器(Dispathcher)的负载分配策略[1,3]已得到了较好的商业应用. 对于广域集群, 通常采用群集间的基于 DNS 和群集内的基于 Dispatcher 的两级负载分配机制[4]. 由于来自不同客户域负载的高度不一致性和真实 Web 工作负载的高可变性, DNS 需要利用额外的状态信息(如负载权、服务器的状态信息等)实现细粒度的负载均, 目前此类问题还是一个开放性课题[5]. 文[6]试图通过引入 HTTP 重定向机制实现第三级负载分配, 但是重定向的引入不仅增加了请求的响应延时, 而且会对整个问题的求解带来负面反馈, 使得整个问题的求解变得更加困难. 本文对一级分配机制进行了深入研究, 提出了一种新的资源配置优化算法, 通过改进资源管理的合理性为集群系统提供更为可靠的负载平衡保证.

下文组织如下. 第二节对相关工作进行了归纳; 第三节给出了一个改进的基于 DNS 的负载分配算法; 第四节包括实验系统的设计以及仿真结果的分析; 最后在第五节总结全文.

2 相关工作

2.1 广域 Web 集群体系结构

图 1 为一个广域集群的拓扑结构图, 它由四个局域集群组成, 局域集群中包括域名服务器 DNS、调度器和若干台镜像服务器. 每一个集群被分配一个唯一的虚拟 IP 地址, 集群与集群之间通过高速骨干网互联来实现彼此之间的信息交互和协同操作. 广域 Web 集群系统是一种典型的层次化结构. 因此采用两级负载分配机制. 首先利用局域集群中的 DNS 执行一级分配, 将客户端请求分配至各个局域集群中, 然后在局域集群中实现二级分配. 文[4]对二级分配策略进行了综述, 指出局域

收稿日期: 2005-10-12; 修回日期: 2007-10-08
基金项目: 国家自然科学基金(No. 60272024), 国家 863 计划项目(No. 2007AA01Z478)

集群的基于调度器能够完全控制所有到来的请求,并且实现精细粒度的负载平衡.本文的研究重点是广域分布的自治域之间的动态负载分配和负载平衡策略.

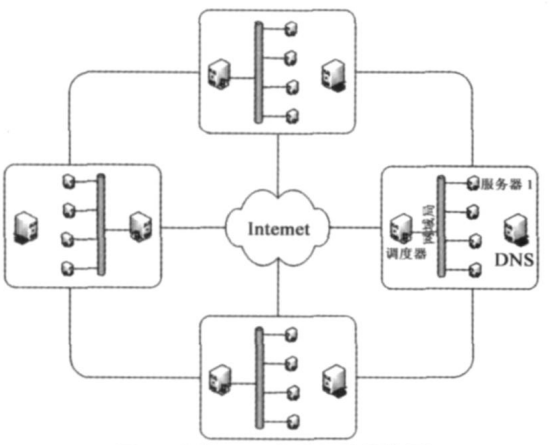

图 1 广域 Web 集群拓扑结构图

2.2 基于 DNS 的一级分配机制及改进思路

目前利用 DNS 进行一级负载分配时,所采用的策略是为每一个请求尽量分配距离其最近的局域集群.实现方法是利用 DNS 的地址映射缓存机制实现.但是由于地址映射的有效期,这种方法只能处理很少一部分地址请求,从而只能实现一种"尽力而为"的粗粒度负载分配.改进的方法有两种,一种是实现地址映射的有效期自适应调整,另一种是通过额外的状态信息直接进行地址映射获取最合适的局域集群地址.显然后者较前者具有普遍性.本文选取增强 DNS 地址解析能力作为求解目标.

3 改进的基于 DNS 的负载分配策略

3.1 多量纲性能指标融合策略

在负载分配时,需要根据处理机多个指标综合进行考虑.这些指标具有多量纲性.比如对于 Web 服务器节点,需考虑如 I/O 资源、存储器和 CPU 等计算资源等指标,而对于局域集群需考虑其入口处的网络带宽、接受的连接数等指标.因此必须对这些指标进行融合以消除其多量纲性.

符号 $F_k(x_1, x_2, x_3, \cdots, x_n)$,其中 $k, n \in \mathbf{N}$, F_k 为第 k 个处理机的性能影响因素集合,x_n 为对应的处理机的第 n 个影响因素,其中 x_i 和 x_j 的量纲不一定等同,为了为资源的调度提供科学决策依据,需要从这些数据中提取出有效的可比较数据,求解策略如下.

取某个因素集合 $F_k(x_1, x_2, x_3, \cdots, x_n)$,从 $x_1, x_2, x_3, \cdots, x_n$ 中任取 x_i 与 x_j,比较它们对处理机性能影响的大小,令 $L_{ij} = \Omega(x_i, x_j)$.函数 Ω 可根据处理机的实际需要进行定义,表 1 为一种 Ω 函数定义:

建立 F_k 的元素对,使得其中所有的元素均和除自身以外的元素进行比较.设已取得所有的 L_{ij}, $(I, j = 1, 2, \cdots, n)$,建立 n 阶方阵 $\boldsymbol{A} = (L_{ij})_{n\times n}$.由于矩阵 $\boldsymbol{A}$ 第 i 行第 j 列元素为 L_{ij},而第 j 行第 i 列元素为 L_{ji},它们互为倒数,并且其对角线元素为 1,所以它为逆对称矩阵.

表 1 一种 $(x_i, x_j,)$ 运算规则定义

Ω	Ω 输出
x_i 与 x_j 对处理机性能影响的程度相同	$L_{ij} = 1$
x_i 比 x_j 影响的程度略大	$L_{ij} = 3$
x_i 比 x_j 影响的程度大[ ] $L_j = 5$	
x_i 比 x_j 影响的程度大很多	$L_{ij} = 7$
x_i 的影响如此之大,x_j 根本不能和它相提并论	$L_{ij} = 9$
认为 L_{ij} 介于 $2n-1$ 和 $2n+1$ 之间	$L_{ij} = 2n,\ n = 1, 2, 3, 4,$
当且仅当 $L_{ij} = n$	$L_{ij} = 1/n,\ n = 1, 2, 3, 4, \cdots, 9$

对成对比较距阵 $\boldsymbol{A}$ 进行处理,建立向量的迭代序列如下:

$$\begin{cases} \boldsymbol{E}_0 = [1/n, 1/n, \cdots, 1/n]_{1\times n} \\ \boldsymbol{E}_K' = A\boldsymbol{E}_{k-1} \\ \boldsymbol{E}_k = \dfrac{\boldsymbol{E}_k'}{\| \boldsymbol{E}_k' \|} \end{cases} \tag{1}$$

其中 $\| \boldsymbol{E}_k' \|$ 为 $A\boldsymbol{E}_{k-1}$ 的 n 个分量之和,$k = 1, 2, \cdots, n$.

由 Perron 定理可知,迭代的 n 维向量序列 $\{\boldsymbol{E}_k\}$ 收敛,记其极限为 $\boldsymbol{e}$,且记

$$e = [c_1, c_2, \cdots, c_n]_{1\times n} \tag{2}$$

设 $w_i = c_i$, $i = 1, 2, 3, \cdots, n$,则 w_i 便是所求的权系数,它满足如下条件:

$$w_i \geqslant 0, \quad \sum_{i=1}^{n} w_i = 1 \tag{3}$$

取变量 y_k,并把它表示成 $x_1, x_2, \cdots, x_n$ 的线性组合: $y_k = w_1x_1 + w_2x_2 + \cdots + w_nx_n$

这样便求出了对应 F_k 的量化可比数据 y_k.依次类推解出所有 $F_i(i = 1, 2, \cdots)$ 对应的量化数据 y_i 便获得一个量化数组 $y(y_1, y_1, y_3, \cdots)$.

3.2 集中式的广域 DNS 负载分配算法

图 2 为图 1 所示的广域集群的负载分配层次结构图.考虑到系统的资源构成并不是非常复杂,只有较少的层次结构(本例中为三层),且局域集群中只是由有限个 Web 服务器结点构成.层次分析的主要目的是要确定最底层,既方案层中各个计算单元对于目标的总排序权重,以便找到最优的求解方案.

算法如图 3 所示,算法中的步骤 1- 8 对层次树中的每一个结点进行权系数构造,步骤 9- 11 为基于系数构造实现了资源的分配.首先每个局域集群内部的 Web 服务器按照节 3.1 的策略对自身性能进行评估并将结果传递给局域 DNS,各 DNS 将其域内各 Web 服务器评估参数与其自身性能参数再次进行融合得出其节点系

数. 然后广域 DNS 根据性能参数的比例关系将资源层层分配置各 Web 服务器中.

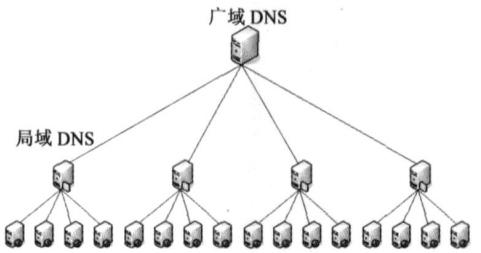

图 2　一种广域集群的负载分配层次图

1　K ← 局域集群数目;
2　for each 局域集群 LWC_k, $k=1, \cdots, K$
3　{ n ← LWC_k 的 Web 服务器(WS)数目;
4　for each Web 服务器 $WS_j, j=1, 2, \cdots, n$
5　{对其性能指标进行融合生成 x_i}
6　生成 $x_1, x_2, \cdots, x_n$;
7　LWC_k 对 $x_1, x_2, \cdots, x_n$ 进行融合生成 y_k; }
8　Q ← 总任务量;
9　广域 DNS 为局域集群 LWC_k, $k=1, \cdots, K$, LWC_k 分配负载 $Q_k = Q \times y_k \Big\backslash \sum_{j=1}^{K} y_j$
10　IWC_k 将 Q_k 向下分配给集群内 WS_j, $j=1, 2, \cdots, n$, 则 WS_j 分配负载 $Q_j = Q_k \times x_j \Big\backslash \sum_{j=1}^{n} y_i$

图 3　一种集中式的广域 DNS 负载分配算法

3.3　复杂度分析

首先分析时间复杂度. 该算法包括了层次树节点权系数构造和基于权系数的负载分配两个部分, 前者是从各个叶子节点出发的宽度优先遍历到根节点, 而后者是从根节点出发的叶子节点的宽度优先遍历到叶子节点, 两者的复杂度相等. 由于它们均有两层循环(一个循环 k 次, 一个循环 n 次), 所以其对应的总计算时间单元为 $2 \times k \times n$, 时间复杂度为 $O(k \times n)$.

然后分析空间复杂度. 该算法在执行之前, 必须在系统事先指定的位置为每一个计算节点分配空间以存储各自的处理机性能的指标以供设计使用. 这里设对于一个有 $m=3$ 层计算资源的广域集群结构, 其对应于计算节点 P_{ij}(对应第 i 层, 第 j 个节点)的处理机指标有 K_{ij} 个, 并设第 i 层对应的节点共有 N_i 个, 则系统应实现分配的静态空间单位 S 应为:

$$S = \sum_{i=1}^{m} \sum_{j=1}^{N_i} k_{ij} \tag{4}$$

该算法在执行的过程会根据系统提供的处理机性能指标生成对应的权系数, 因此应动态的提供适当的空间以保存计算的结果. 设记录每一个计算节点的空间单位为 R, 则程序动态创建的空间单元 D 为:

$$S = \sum_{i=1}^{m} \sum_{j=1}^{N_i} R \tag{5}$$

同时, 为了完成用户提交的计算任务 Q, 也要根据其任务量动态地创建空间以保存计算任务. 因此, 算法创建的空间单元开销为

$$S = \sum_{i=1}^{m} \sum_{j=1}^{N_i} (k_{ij} + R) + Q \tag{6}$$

从以上的算法的性能分析中可以看出, 该算法具有较小的空间复杂度和时间复杂度. 因此它不失为一个较为合理的调度算法.

4　实验系统设计

文[12]提出了利用随机高级 Petri 网对 Web 性能进行定量分析. 我们设计并实现了 Web 集群试验系统. 分别对 Web 服务器节点、其产生的数据流量以及局域集群间的网络通信建立数学模型. 试验采取了如图 1 所示的网络拓扑. 广域集群由四个局域集群组成, 每个局域集群包括四台 Web 服务器. 各局域集群通过高速骨干网实现互连.

4.1　Web 服务器节点模型

Web 服务器节点被设计为单服务窗等待制排队模型 $M/M/1$. 即系统内只有单个服务窗. 顾客按参数 λ 的泊松分布到达, 若窗口忙, 则排队等待服务; 且顾客到达的时间间隔与服务窗为每个顾客服务的时间均为负指数分布; 平均服务率为 $\mu(\mu > \lambda)$. 令 $\rho = \lambda / \mu$, 可推导系统中顾客的均值 L_a(包括正被服务和排队等候的顾客均值)以及正在接受服务的顾客均值 L_s 分别为:

$$L_a = \frac{\lambda}{\mu - \lambda}, \quad L_s = \rho \tag{7}$$

则 Web 服务器节点的指标为 CPU 占用量 U_{av}、I/O 占用量 I_{av}和网络带宽占用量 J_{av}分别计算如下:

$$U_{av} = \lambda \cdot L_s = \lambda^2 / \mu, \quad I_{av} = \lambda \cdot L_a = \frac{\lambda^2}{\mu - \lambda},$$
$$J_{av} = \lambda \tag{8}$$

仿真中, 服务器节点 i 随机生成常量 μ_i, 根据连接分配强度 λ 动态调整其相关参数.

4.2　Web 服务器流量模型

文献[7, 8]指出 Web 服务器的负载具有高可变性和自相似性, 这是由于重尾分布 ON-OFF 区间的重叠造成的. 对 Web 服务器响应进行了分析, 建立了 Web 服务器的流量模型.

连续两次请求间的间隔时间分布为平均开关时间为 1 秒, Pareto 整形参数为 1.4 的 Pareto on/off 分布.

用户请求涉及到的页面数分布为以 3.86 为期望, 9.46 为方差的逆高斯分布.

响应中的 HTML 文件的主体部分为期望为 7.630, 期望为 1.001 的对数正态分布, 其文件尾部为重尾 Pareto 分布, 平均长度为 10240byte, 整形参数为 1. 响应中的附属文件部分是期望为 8.125, 期望为 1.46 的对数

正态分布.

4.3 通信流量模型

文献[9]骨干网的流量特征和模式进行了分析, 文献[10]对基于不同传输协议的 HTTP 协议进行了性能比较. 针对本文的需要, 对其进行了借鉴, 并设计了局域集群间网络通信模型. 模型中采用了 HTTP/1.1[11]协议. 根据模型, 局域集群 i 和广域 DNS 之间进行一次通信的传输时间为:

$$T_i = 2\mathrm{RTT}_i + \sum_{k=1}^{n} [(\mathrm{req}_k + \mathrm{res}_k)/\mathrm{BW}_i] \qquad (9)$$

其中, RTT_i 和 BW_i 分别是局域集群 i 和广域 DNS 的传输延迟和可用带宽. req_k 和 res_k 分别是第 k 个请求的报文长度和响应报文长度. req_k 为以 358 字节为期望的指数分布的随机变量. res_k 的分布情况在节 4.1.1 进行了定义. 为了描述可用带宽的动态性, 将其设置为 4 种离散区间, BW_i 被设计为该 4 种区间上的平均分布, 相应的传输延迟 RTT_i 为则为每类区间 RTT 定义闭区间上的平均分布. 具体定义如表 2.

表 2 可用带宽区间定义

类型	可用带宽下限(Mbps)	传输延时 RTT(ms)
1	其他	[40, 70]
2	≤0.9	[120, 150]
3	≤0.7	[180, 210]
4	≤0.4	[270, 300]

5 仿真试验

实验系统开发工具为 Matlab 和 C++. 每次试验时间为 24 小时其采用不同的随机数种子, 共进行 10 次试验, 试验过程中采用随机方式采样试验数据, 最终取 10 次数据的平均值.

5.1 负载平衡验证

本算法的直接目的是保证整个广域 Web 集群系统的负载平衡. 对每个局域 Web 集群分得的连接数进行采样统计, 其结果如图 4 所示. 图 4(a)为请求连接数为 600 时, 采用本算法前后的负载分配情况. 没有采取本算法时, 广域 DNS 采用距离最近的局域集群的原则进行请求分配. 此时, 分配表现出较差的均衡性, 典型的, LWC_2 分得的连接数是集群LWC4的2.5倍左右. 而采用本算法后, 由于是根据各局域集群的资源情况的均匀分配, 因此系统保持了较好的均匀性, LWC_1 和 LWC_2 的连接数相差最大, 前者也只是后者的 1.05 倍. 算法运行过程中各 Web 服务器节点和局域集群 DNS 的权重系数如表 3 所示. 将请求总数设为 800, 再次对各局域集群分配的连接数进行统计, 结果如图 4(b)所示. 改进后的算法仍然表现出了很好的负载均衡性.

表 3 各节点权系数构造表(连接数= 600)

	局域集群 I	局域集群 II	局域集群 III	局域集群 IV
Web 服务器 1 权系数	0.4326	0.3273	0.1465	0.5883
Web 服务器 2 权系数	0.6656	0.1909	0.1746	0.1832
Web 服务器 3 权系数	0.1253	0.1867	0.1364	0.7258
Web 服务器 4 权系数	0.2877	0.0376	0.1892	0.1139
局域集群权系数	0.16585	0.17385	0.16915	0.1665
局域集群分配连接数	199	208	203	200

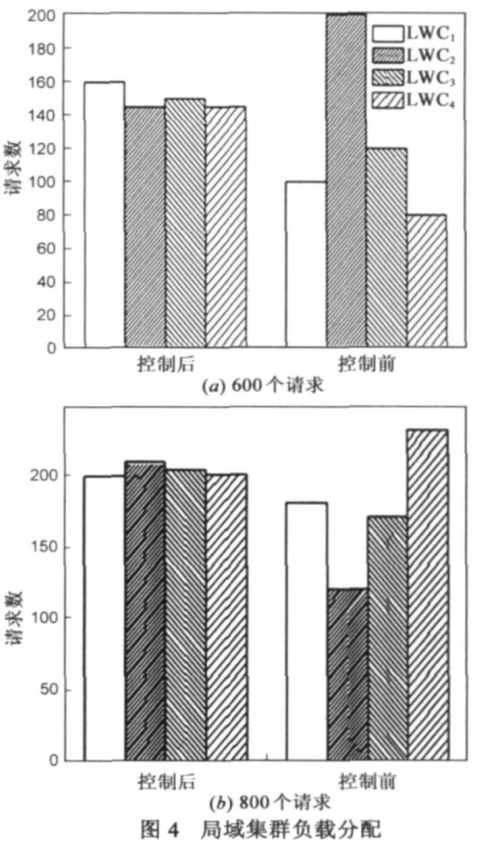

图 4 局域集群负载分配

5.2 响应延时

从用户对 Web QoS 感知的角度来说, 用户最关心的是整个 Web 系统的延时, 即用户从发出请求到打开网页的之间的时间长度. 这段时间不仅包括了用户请求和 Web 服务器的响应信息在网络上传输的时间, 还包括了 Web 服务器的处理请求的时间.

图 5(a)表示了系统的响应延时随着请求强度的增加的变化情况. 从中可以看出, 随着请求强度的加大, 系统的响应延时也随之增加. 连接强度小于 200 个/秒时, 采用算法前后的系统延时相差不大, 这是因为此时负载平衡的重要性尚未表现出来, 当连接强度大于 500 以后, 算法的优越性逐渐表现出来. 连接强度为 800 个/秒的情况下, 采用本算法前后的系统延时分别为 8.3 秒和 6.4 秒, 后者是前者的 75%. 从中可以看出利用本算法通过实现系统的负载平衡从而降低了因为负载增大引起的请求响应延时的增长幅度.

5.3 准入概率

准入概率是一个重要的指标, 因为客户的请求被不可用的服务器拒绝将严重影响用户感知的 Web QoS. 由于实验系统中的 Web 服务器结点采用了无限制的等待队列长度, 因此实验中订制了一个较为简单的准入控制策略. 广域 DNS 给某一个局域集群分配请求前, 对该段局域集群入口带宽进行判断, 若剩余带宽不足 20%, 则拒绝分配请求. 此时广域 DNS 对请求的准入概率为:

$$RA = 1 - \prod_{i=1}^{K} P(Rq_i \geqslant 0.8 b_i) \qquad (10)$$

其中 Rq_i 和 b_i 分别是局域集群 i 占用的带宽和其基本带宽.

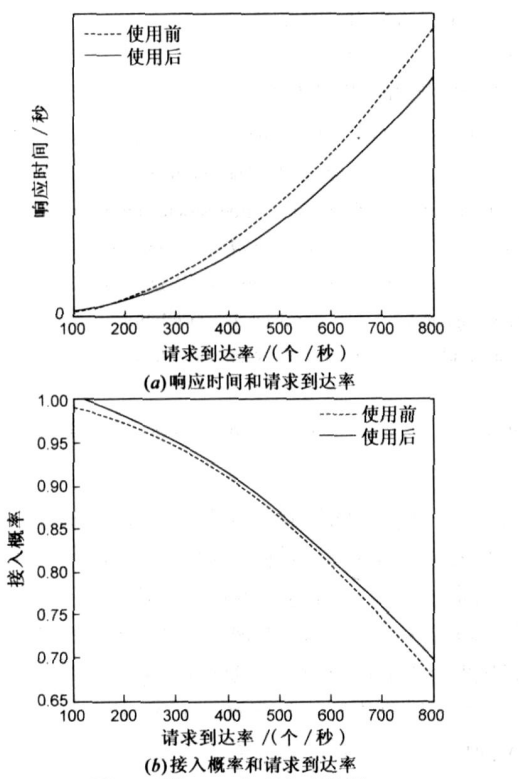

图 5　系统 QoS 相关仿真结果

图 5(b) 是采用本算法前后的接入概率的性能比较. 采用本算法后可提高广域集群的接入概率, 但是增长幅度不大, 约有 1.4% 左右. 这是两个原因造成, 一是实验系统中采用的广域集群规模(K)较小, 二是准入控制策略定制过于简单. 但是这已经说明通过将连接请求均匀地分配到各局域集群后, 可在一定程度上降低局域集群发生拥塞的概率, 从而提高广域集群的接入概率.

6　结论和工作展望

对于广域 Web 集群系统而言, 能否实现连接请求的公平分配对于 Web QoS 保证而言是一个至关重要的问题. 本文在给出的多量纲性能参数融合策略的基础上, 提出的请求分配算法可满足系统的负载平衡需求. 下一步我们将计划通过引入区分服务和用户请求的优先级分类, 从而实现 QoS 感知的广域 Web 集群系统. 另外我们还将对广域 Web 集群的实验系统进行深入加工, 使其更为合理, 比如 Web 服务器结点的排队论模型. 这些工作将在今后的工作中进行完善.

参考文献:

[1] T Schroeder, S Goddard, et al. Scalable Web server clustering technologies[J]. IEEE Network, 2000, 14(3): 38–45.

[2] 林闯, 单志广, 任丰原. 计算机网络的服务质量[M]. 北京: 清华大学出版社, 2004.

[3] G D H Hunt, G S Goldszmidt, et al. Network Dispatcher: A connection router for scalable Internet service[J]. Computer Networks and ISDN Systems, 1998, 30(1–7): 347–357.

[4] V Cardellini, M Colajanni, et al. Dynamical load balancing on Web-server systems[J]. IEEE Internet Computing, 1999, 3(3): 28–39.

[5] R L Carter, M E Crovella. Server selection using dynamic path characterization in wide-area networks[A]. Proc of IEEE INFOCOM'97[C]. Kobe, Japan: IEEE, 1997. 1014–1021.

[6] V Cardellini, M Colajanni, and P S Yu. Geographic load balancing for scalable distributed Web systems[A]. Proc of the 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems[C]. San Francisco, CA, USA, 2000. 20–27.

[7] M E Arlitt, R Friedrich, and T Jin. Workload characterization of a Web proxy in a cable modem environment[J]. ACM Performance Evaluation Review, 1999, 27(2): 25–36.

[8] J E Pitkow. Summary of WWW characterizations[J]. World Wide Web, 1999, 2(1–2): 3–13.

[9] K Thompson, G J Miller, and R Wilder. Wide-area Internet traffic patterns and characteristics[J]. IEEE Network, 1997, 6(11): 10–23.

[10] J Heidemann, K Obraczka, and J Touch. Modeling the performance of HTTP over several transport protocols[J]. IEEE Transaction on Nerworking, 1997, 5(5): 616–630.

[11] R Fielding, J Gettys, J C Mogul, et al. Hypertext transfer protocol-HTTP/1.1[S]. RFC 2068, 1997.

[12] 单志广, 戴琼海, 林闯, 等. Web 请求分配和选择的综合方案与性能分析[J]. 软件学报, 2001, 12(3): 355–366.

作者简介:

李　捷　男, 1975 年生, 博士, 讲师. 研究方向为网络管理、Web 质量保证. E-mail: jsjt9@henu.edu.cn

刘景森　男, 1968 年生, 博士生, 副教授. 研究方向为计算机网络与信息安全.

刘先省　男, 1964 年生, 博士, 教授, 中国电子学会高级会员. 主要研究方向为信息处理、网络建模.