文章编号:1000-582X(2010)02-0073-06

动态反馈的异构集群负载均衡算法的实现

陈　伟,张玉芳,熊忠阳
(重庆大学 计算机学院,重庆 400044)

摘　要:虚拟服务技术(LVS)的集群负载调度系统中的加权调度算法权值是静态的,没有动态调整机制,不能依据真实服务器处理能力进行动态的任务分配;给出了一种通过量化调度器和真实服务器的实时反馈负载信息指标,采用服务器负载率和分配权值为计算指标,引入临界回归因子,利用动态反馈临界加速回归的算法思想,实现基于动态反馈机制的动态反馈临界加速回归分配算法。该算法能根据负载均衡调度器和业务处理服务器的实时反馈信息,及时进行负载调度,实现负载动态平衡,进一步提高服务器的利用效率和集群系统的吞吐率,并在 LVS 的负载平衡调度集群系统中进行了应用性的实验验证。测试结果表明,该算法可实时反馈负载信息,动态进行负载调度,整体上较好的实现了负载动态平衡,提高了服务器的利用效率和集群系统的吞吐率。

关键词:LVS 虚拟服务技术;异构;动态均衡;回归;反馈

Research and realization of the load balancing algorithm for heterogeneous cluster with dynamic feedback

CHEN Wei , ZHANG Yu-fang , XIONG Zhong-yang
(College of Computer Science, Chongqing University, Chongqing 400044, P. R. China)

Abstract: Linux virtual server (LVS) can establish a server cluster based on Linux with high availability, high capability, high reliabilities and scalability, while the weights of the weighted scheduling algorithm for the cluster load balancing system in LVS is static. Without the dynamic adjusting mechanism, the task can not been assigned dynamically according to the servers' real capabilities. A dynamic load balancing algorithm (Multiplicative Regression in Critical Area, MRC) is introduced to adjust weights of the servers dynamically. The MRC algorithm calculates the load ratios and distributed weights according to the real-time feedback information from the real servers, and introduces the threshold regression factors to schedule tasks for web server cluster to achieve load balance. The method improves server's efficiency and the capability of the whole system. The method is applied to web server cluster systems based on Linux virtual server. The experimental results show that the algorithm can achieve real-time feedback of the load information, dynamic load scheduling, and achieve dynamic load balance in a whole, which improves the server utilization efficiency and the capability of the whole system.

Key words: heterogeneous ; dynamic balancing; regression ; feedback

---

收稿日期:2009-10-12
基金项目:教育部留学回国人员启动基金资助项目[20071108-10]
作者简介:陈伟(1974-),男,主要从事数据挖掘、远程教育方向研究,(Tel)02386269009;
(E-mail)chenwei@mail.cqzkb.gov.cn。

Internet业务量爆炸性增长已经使网络服务器不堪重负,带来了服务器负载的不断增长,特别对于访问数量多的站点,服务器经常在短时间内过载。而服务器集群(Cluster)技术正成为实现高可伸缩的、高可用网络服务的有效架构之一。集群技术是将若干个松散连接的独立的服务器架构成具有高可靠性的和可扩展的集群服务器[1]。集群的内部结构对客户是透明的,客户端只看到一个高性能服务器。集群系统的优越性及作用主要有1)避免服务器软、硬件升级造成服务器暂时的中断;2)避免单点失效的出现;3)能够实现透明的负载均衡;4)能够真正的保证高可用性,高可靠性,高性能等特性。

集群系统中的一个重要问题就是负载均衡(load balancing),集群系统负载均衡的目标是根据各服务器性能来分配与其匹配负载,最大限度利用集群的优势并提供更好的网络服务质量,以最小化应用的执行时间,属最优化理论范畴。实际应用的集群系统多是异构的,即构成集群的各个服务器配置(硬件、操作系统等)是不完全相同的,在异构集群中衡量确定各服务器分配权值显得尤为重要。

1 负载均衡算法

实现负载均衡的算法有两大类:一种是静态的负载均衡算法,这种算法没有考虑节点运行时负载情况可能发生的变化,只是根据事先的决策进行任务分配;另一种是动态的负载均衡算法,这种算法以系统当前的负载情况为依据,可以动态的调整服务器的权值,充分利用节点的处理能力来减少服务的响应时间。传统负载均衡算法主要是建立在访问请求泊松到达和响应时间呈指数分布的假设上。如轮转法(RR)、加权轮转法(WRR)、最小连接数法(LC)、加权最小连接数法(WLC)[2]等,这类算法都是静态算法,没有考虑系统运行时状态,带有极大局限性,仅适用于小规模的单一配置的静态网页服务系统。而最少等待队列调度和最少利用率调度是常用的动态的负载均衡算法。

由于Web负载在到达时间间隔和响应时间上都明显呈现重尾特点[3],同一个请求分配到不同服务器所产生的影响也不同。因此,一些可以动态计算每个服务器负载情况并动态调整负载分配的算法被相继提出。WRR-num和WRR-time[4]是基于轮转法的动态加权算法,Round-trip和Xmitbyte[5]是基于最小连接数动态加权算法。上述4个算法只考虑了不同请求任务负载之间的差异,但没考虑每台服务器处理能力之间的差异,主要适用于相同配置服务器组成的集群系统。

基于实时动态调整的需要,设计了一种根据调度器和服务器的实时反馈信息,综合服务器能力和动态调整服务器分配权值的收敛算法,希望可以避免普通加权算法因权值固定可能出现的任务分配与真实服务器处理能力相背离的缺陷,可以剔除网络偶然阻塞、服务器瞬间峰值等小概率事件对负载率的影响,并且能够避免因调度器的频繁调整而造成额外的负担,实现基于动态反馈机制的负载均衡,从而进一步提高服务器的利用率及集群的吞吐率。

2 动态反馈负载平衡工作原理

单用户/多用户共享异构集群环境中,主要由负载均衡调度器、业务处理服务器和客户机组成(见图1)。在运行时依据对业务处理服务器的服务负载实时监视并反馈给负载均衡调度器,以此对系统不可预测的负载变化做出及时反应,从而保证系统良好的性能。集群动态负载的工作原理加入了动态反馈负载算法的基本思想,根据负载均衡调度器和业务处理服务器的实时反馈信息,采用服务器负载率和分配权值为计算指标,利用动态反馈临界加速回归的算法思想,实现基于动态反馈机制的负载均衡。

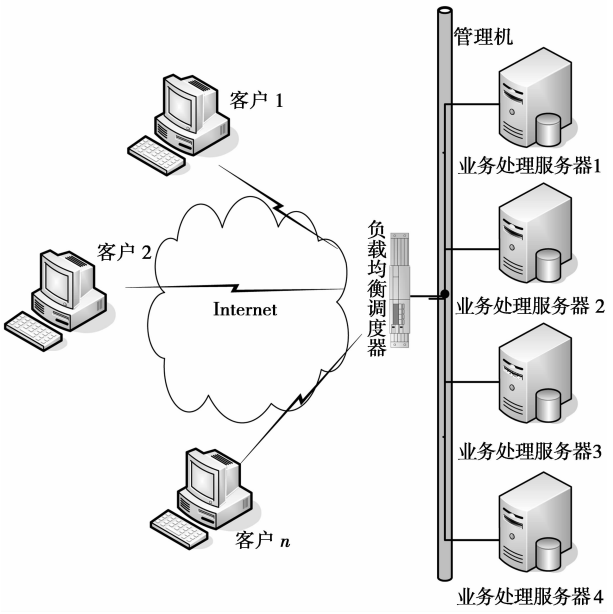

图1 动态负载均衡框架结构图

基于动态反馈的负载均衡算法的设计思想主要是为了保证负载均衡调度器具有较高的重定向效率,负载均衡调度器不对每个任务本身的负荷进行分析,而是动态监测负载被分配后对各个服务器所造成的影响,通过动态反馈业务处理服务器实际负载状态,再结合设定的业务处理服务器固有处理能力,可分别计算出服务器负载率 V_L^i 、平均负载率 β 和服务器分配权值 W_L ,根据服务器应分配与之能力相匹配的负载的原则,在后续分配中,对高于平均

负载率的服务器减少分配给它的请求任务数量，对低于平均负载率的服务器增加分配给它的请求任务数量，使集群系统的负载均衡度尽量小以达到负载均衡的目标。

2.1　集群服务器负载指标参数

根据能力与负载相匹配原则，设集群服务器系统由 m 台业务处理服务器组成，服务器 $S_i(i=1,2,...m)$ 的固有能力为 ω_i，其当前的负载为 L_i，将某台服务器当前的负载与其固有处理能力（通常为常量）之比定义为该服务器当前的负载率，记为

$$V_L^i = L_i/\omega_i, \mathrm{Load}_i = V_L^i。\quad (1)$$

集群服务器当前的平均负载率

$$\mathrm{Avg}(V_L) = 1/m \cdot \sum W_L^i,$$

$$\beta = \mathrm{Avg}(V_L)(1 \leqslant i \leqslant m)。\quad (2)$$

在集群服务器完全负载平衡时，有

$$v_l^1 = v_L^2 = v_l^3 = \cdots v_L^m = \mathrm{Avg}(V_L)。\quad (3)$$

考虑服务器之间的性能差异，因此每一台业务处理服务器赋一权重系数 $W_i \in [0,1]$（集群处理能力为：$1/m \cdot \sum W_i \cdot S_i, \sum W_i = 1$），系数越大服务器的性能越高，该系数可看成是服务器分配到负载的概率（即服务器分配权值 W_L）。当服务器投入集群系统中使用时，系统管理员对服务器 S 都设定一个初始权值 W_{Ti}（零负载权值），随着服务器负载的变化，对该权值进行调整。规定权值的范围 $[\mathrm{LOW}_i, \mathrm{HIGH}_i]$，服务器 S 当前权值记为 CT_i。服务器负载率 V_l 和分配权值 W_L 关系是：$W_L \propto V_L$，从而确定适当的 W_L 是算法的关键。

根据模糊集理论[5]，对于实际系统，任何一项资源（CPU、内存、网络、磁盘）或者集群在线连接率超过一定的使用率（比如 95%），系统就不再有能力承受额外负载。此特性决定了应该选用乘积平均值法。如前所述，系统使用 S 服务器当前 CPU 使用率、网络带宽使用率、内存使用率、磁盘 I/O 使用率和集群在线连接率 5 项指标对 S 服务器的负载状况进行评估。这 5 项指标与负载水平均成正比，各性能指标对不同应用的影响是不同的，为了体现各项指标对负载状态的影响程度，也为每一项指标赋予一个权重系数，来表示各个负载信息在综合负载中的权重和灵敏度。定义服务器 S_i 的 CPU 使用率为 $P[i]$，网络带宽使用率为 $N[i]$，内存使用率为 $M[i]$，磁盘 I/O 使用率为 $D[i]$，集群在线连接率为 $C[i]$（服务器 S 在 t 时刻的连接数为 N_i，服务器 S_i 的连接指标 CON_i 为其在线连接数与 m 台服务器收到连接数的比值，其计算公式为：$C[i] = N_i/\sum N_k (1 \leqslant k \leqslant m)$；对应表示各个指标的重要程度的系数分别为 $\lambda_p、\lambda_n、\lambda_m、\lambda_d、\lambda_c \in [0,1]$ 且 $\sum \lambda_k = 1$ 且。根据不同应用的需求，调整各个指标的系数。服务器 S_i 综合负载 Load_i 可以通过以下公式计算出：$\mathrm{Load}_i = \lambda_p \cdot P[i] = \lambda_n \cdot N[i] = \lambda_m \cdot M[i] = \lambda_d \cdot D[i] = \lambda_c \cdot C[i]$。(4)

由此可知

$0 \leqslant \min(P[i], N[i], M[i], D[i], C[i] \leqslant \mathrm{Load},$

$\leqslant \max(P[i], N[i], M[i], D[i], C[i] \leqslant 1$。

2.2　临界回归因子的变化规律

为使集群服务器系统负载均衡，负载均衡调度器以当前平均负载率为基准调整各业务服务器 S 的负载。首先，利用服务器负载测试软件对单台服务器的实际测试结果（如图 2）。

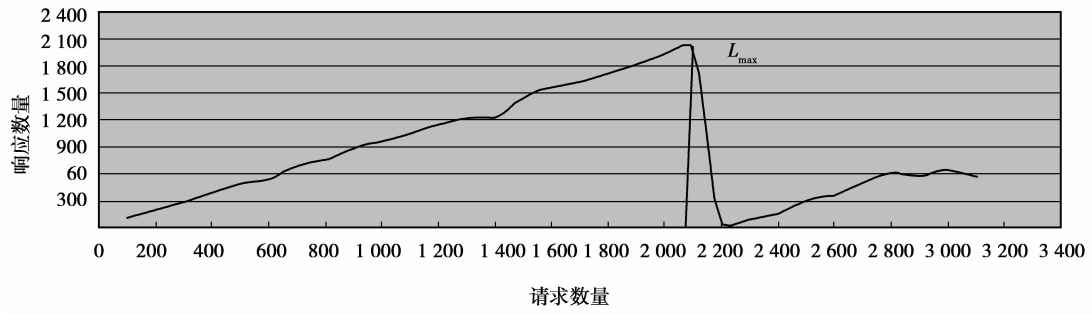

图 2　发送的请求个数与响应的请求个数关系图

从图 2 可以发现，当发送的请求数量达到一定值 L_{max} 时，服务器响应请求的数量急剧下降为零，并持续一段时间。这是服务器操作系统软件为防止死机而采取的措施，也称“拒绝服务”（DoS）。服务器出现 DoS 是因其服务能力已达到饱和状态，必须等待处理完一些已接收负载后，才有可用资源对外提供服务，当服务器出现 DoS 时，其响应时间急剧增加，超过 T_{max}（此时负载为 L_{max}，即服务器最大处理负载）。设定一旦请求响应时间超过 T_{max}，则服务器的可用资源为零。从图 3 中还可以看出，只有一个最基本的请求任务加载到服务器上时，请求的响应时间为 T_{bas}（定义此时负载 L_{bas} 为基本负载），如果忽略这

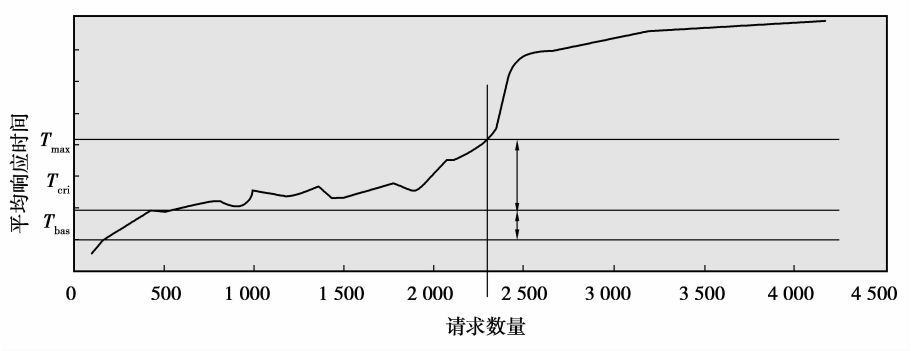

图3 发送的请求个数与平均响应时间关系图

个小负荷和小概率事件的影响，L_{bas}就是系统维持运转的基本负载；将（$L_{max}-L_{bas}$）定义为服务器的固有能力，即$\omega_i=(L_{imax}-L_{ibas})$。

令对某台服务器实测的当前响应时间为T_{mow}（此时的负载综合实测为T_{mow}）。将（$L_{mow}-L_{bas}$）定义为服务器的当前负载，即$L_i=(L_{inow}-L_{ibas})$。代入公式(1)有

$$V_L^i=L_i/\omega_i=(L_{imow}-L_{ibas})/(L_{imax}-L_{ibas}),$$
$$T_{ibas}\leqslant T_{inow}\leqslant T_{imax},V_L^i\in[0,1]。\quad(5)$$

公式(5)称为服务器S_I当前负载权值的等效变换。

从图3中还可以明显看出，当请求负载增加到一定量时，响应时间的变化出现抖动，表明系统即将达到饱和状态。将这一负载区域定义为DoS出现前的临界区，从响应时间上看为$T_{cri}\sim T_{max}$区间。理论上通过增加进入临界状态服务器的负载值而实际没有增加（服务器负载越大则减少下阶段分配的负载），从而可达到抑制服务器进入饱和状态的目的。定义负载权值变化的部分为临界回归因子Δ。加入Δ后服务器负载率计算公式和递减因子的计算公式如下

$$V_L=L_i/\omega_i=\Delta。\quad(6)$$

$$\Delta=\begin{pmatrix}-\varepsilon\cdot L_{cri}/L_{max},L_{inow}\in[L_{ibas},L_{icri}]\\=(\varepsilon/2)\cdot(1-L_{cri}/L_{max}),L_{inow}\in[L_{cri},L_{imax}]\end{pmatrix}。\quad(7)$$

上述公式中，Δ只在服务器负载进入临界区后才产生作用，且(L_{cri}/L_{max})为服务器的临界深度。由于每台服务器的临界深度不一样，设置参数ε来调整其递减的速度（通常ε取值0.618黄金分割比例数）。有了Δ，可以主动控制服务器不出现DoS状态。当服务器的负载接近L_{max}时已分配的负载会继续得到该服务器的服务，只是禁止该服务器参加下一阶段的请求分配过程，这样就避免原有的请求任务受系统能力饱和影响而出现DoS，并对其告警，如果多台服务器告警则启动冗余备份的服务器来分担请求负载；当服务器负载低于L_{cri}时可以继续给其分配任务。

由于对进入临界状态的服务器采用增加其负载加倍减少后续阶段分配负载（没有进入临界状态的服务器则减少其负载值加倍增加后续阶段分配负载）的策略，研究的算法也称为临界加速回归（multiplicative regression in critical area，MRC）负载分配算法。值得注意的是，临界因子Δ只是理论上变化用于计算分配权值反馈影响分配权值而实际上当前负载并没有发生变化）。计算出了每台服务器负载率后，就可以根据公式(2)得到平均负载率。在此基础上，通过调整下一阶段分配到每台服务器的请求负载达到负载均衡目标。由于无法确知后续请求数量和负载状况，依据当前的请求负载状况（负载值、分配的请求个数等来推算应做出的调整。令达到负载均衡状态时服务器S_i分配的请求个数为$\mathrm{Avg}(N_i)$，对应的负载状况为$\mathrm{Avg}(L_i)$；而服务器当前分配的请求个数N_i和负载状况L_i都可以测得。（负载临界值L_{cri}和负载平均值L_{wvg}的关系：）$L_{avg}\leqslant L_{cri}$。假设负载条件相同，那么分配的请求数量的变化与负载的变化应是一致的（$N_i\propto W_i$）。将每台服务器应分配的请求个数在整个集群系统应达到的请求个数中所占的比例定义为该服务器的分配权值记为W_{iL}，引入服务器分配权值计算公式

$$W_{iL}=\mathrm{Avg}(N_i)/\sum\mathrm{Avg}(N_i),i\in[1,m]。\quad(8)$$

负载均衡调度器按照集群中每台服务器的分配权值为其分配一定的请求任务负载。对于当前负载率大于平均负载率的服务器来说则会减少所分配的请求任务数量；反之，则会增加服务器所分配的请求任务数量。公式(7)是假设负载条件相同条件下根据等价变化而得出的理论值，应用中还需要根据服务器的当前综合负载值对服务器分配权值进行二次修正，使得系统负载率位于β的Δ邻域内。

当$\mathrm{Load}_i>\beta=\Delta$，权值$W_{iL}$变小，$W_{iL}\leftarrow\max$

$\{LOW_i,(1-\delta)\cdot CT_i\}$；

当 $Load_i\in[\beta-\Delta,\beta+\Delta]$，权值 W_{iL} 不变，$W_{iL}\leftarrow CT_i$；

当 $Load_i<\beta-\Delta$，权值 W_{iL} 变大，$W_{iL}\leftarrow\min\{(1=\delta)\cdot CT_i,HITGH_i\}$。　　(9)

在公式中，β 是想要达到的系统利用率（集群的理想负载率 $Avg(V_L)$）。当综合负载值位于 β 的 Δ 邻域内时，服务器分配权值不变；当综合负载值大于 $\beta+\Delta$ 时，权值变小；当综合负载值小于 $\beta-\Delta$ 时，权值变大。若新权值 W_{iL} 大于 $HIGH_i$，将新权值设为 $HIGH_i$，公式(9)是一个负反馈公式，最终会使权值达到一个稳定点，此时分配权值是不变的。

在实际使用中，若发现所有服务器的权值都接近他们的 LOW_i 或零，则说明整个服务器集群处于超载状态，这时需要加入新的服务器结点到集群中来分担部分负载；反之，若所有服务器的权值都接近于 $HTGH_i$，则说明当前系统的负载都比较轻。

3　算法性能比较

LVS 群集是一个被特别配置的、可提供高性能网络服务的服务器的集合。在 LinuxIPVS 的基础上结合单位现有的条件，实现了一个简单的动态负载平衡算法。对所实现的算法进行了性能测试。测试环境：后端采用四台操作系统和硬件配置均不相同的服务器；前端采用一台部署了 LinuxIPVS 的特殊的计算机负载均衡调度器，理论最大转发速率为每秒 18000 个 TCP 连接。客户端采用二十台普通 PC 机。集群内部网络连接设备采用 100M/s 的 Intel 交换机。前端分配器和后端服务器上运行我们设计的系统软件，分配算法包括了静态加权轮转法（WRR），基于请求数量的加权最小连接数优先法（WLC），基于响应时间的动态加权轮转法（DWRR），临界加速回归法（MRC）等。客户端运行 Web 请求仿真软件，请求事件流为符合负指数分布的泊松事件流，仿真软件要求能够生成持续时间有长有短的负载连接。

测试结果：分别使用上述 4 种请求分配算法测试对不同负载文件集的访问情况，观察请求任务平均响应（完成）时间的变化。测试结果如图 4、5 所示。

从图 4 和图 5 可以看出，MRC 算法的平均响应时间变化曲线最为平滑，这说明 MRC 算法在对集群服务器当前负载的评估更准确，而随机概率分配策略使访问负载的分布更均匀。

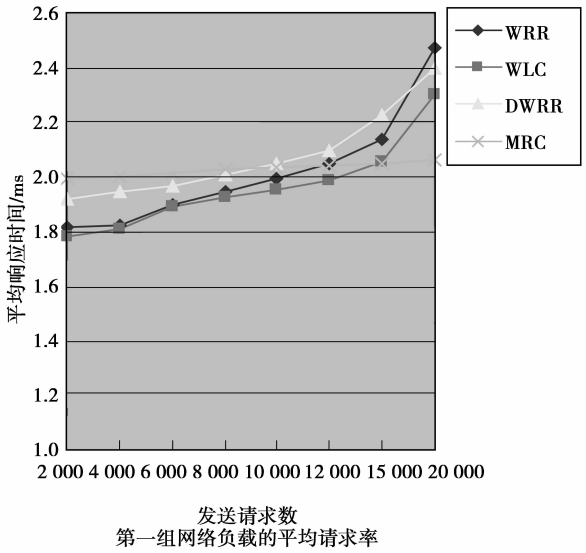

图 4　第 1 组访问负载条件下的平均请求响应时间

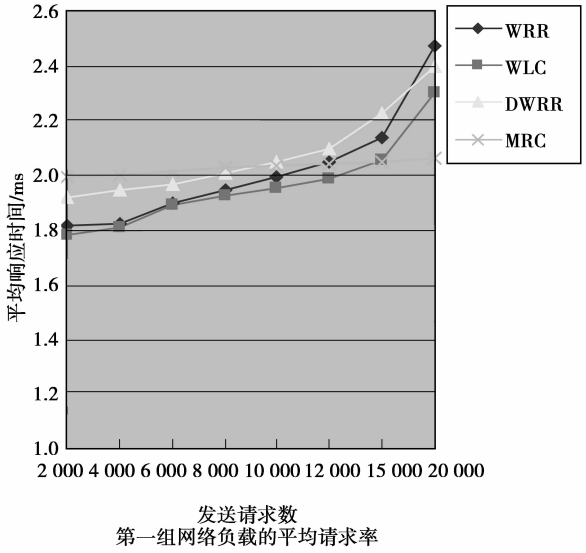

图 5　第 2 组访问负载条件下的平均请求响应时间

3　结　论

针对服务器集群系统中请求负载分配问题，提出了一种动态反馈临界加速回归分配算法。该算法通过服务器负载的等效变换更准确地反映了集群中每台服务器的当前负载状态，为均衡地分配请求负载奠定了基础；通过临界回归因子的使用有效地抑制了服务器出现 DoS 现象，避免了单台服务器性能严重下降对集群整体性能造成的负面影响，也避免了单台服务器资源闲置造成的系统浪费。给出的动态反馈负载平衡策略，根据实时反馈回来的负载信息调整服务器的权值，动态调整服务器间处理请求数的比例，从而避免服务器间的负载不平衡。实时

反馈负载动态平衡可以较好地避免集群服务器的负载倾斜,可提高集群系统的资源使用效率,从而提高集群系统的吞吐率。

参考文献

[1] MARK B, RAJKUMAR B. High performance cluser computing architatune and systems, 1999.

[2] ZHANG W S. Linux virtual server web site. http://www.linuxvirtualserver.org,2002.

[3] 郝沁汾,郝继升. WWW业务访问特性分布研究[J]. 计算机研究与发展. 2001,38(10):1172-1180.
HAO QIN-FEN, HAO JI-SHENG. Traffic access characterstic distribution research [J]. Journal of Computer Research and Development 2001,38,(10):1172 - 1180.

[4] CASSLICCHIO E, TUCCI S. Static and dynamic scheduling algorithm for scalable Web server farm[C]// In: Proceedings of the IEEE 9th Euromicro Worshop on Parallel and Distributed Processing. [s. l.]:[s. n.], 2001,369-376.

[5] BRYHNI H. A comparison of load balancing techniques for scalable Web servers[J]. IEEE Network,2001,7-8:58-64.

[6] BORZEMSKI L, GGAJEWSKI D. A load balancing system for unix-based local area networks [J]. Microprocessing and Micro-programming, 1993, 39(2):205-208.

[7] LI C H,HYEON C. Approximation algoruthms foe data distribution with load balancing of Web servers[J]. In: Proceedings of IEEE International Conference on Cluster Comuputing,2001:274-281.

[8] Load balancing in distributed systems [EB/OL]. http://www.ibr.cs.tu2bs.de/projects/load,2002.

[9] COLAJIANNI M,YU P S, DIAS D M. Analysis of task assignment policies in scalable distributed web-server systems [J]. IEEE Trans Parallel and Distributed Systems, 1998,9(6):585-600.

[10] CARDELINI V, COLAJANNI M, and YU P S. Redirection Algorithms for load sharing in distributed web-server systems[C]// Proc. 19th IEEE Int'l Conf. Distributed Computing Systems. Los Alamitos. Calif: IEEE Computer Soc. Press,1999,5:23-30.

[11] 王晋鹏,潘龙法,李降龙. LVS集群中的动态反馈调度算法[J]. 计算机工程, 2005,31(9):56-58.
WANG JIN-PENG, PAN LONG-FA, LI XIANG-LONG. Dynamic feedback scheduling algorithm in LVS [J]. Computer Engineering,2005,31.

[12] 刘安丰,陈志刚,曾志文. 一种资源负载均衡的HIJ集群启发式优化算法[J]. 小型微型计算机系统. 25(12):56-58.
LIU AN-fENG,CHEN ZHI GANG,ZENG ZHI WEN. Heuristic optimization algorithm of resource load balancing for web cluster [J]. Mini-micro Systems, 2004,25(12):123-125.

[13] 骆宗阳,董玮文,杨宇航,等. 一种具有高可用性的通用负载均衡技术[J]. 计算机工程, 2003(3):82-87.
LUO ZONG-YANG, DONG WEI-WEN, YANG YU-HANG. A kind of general load balance technology with high availability[J]. Computer Engineering,2003(02):89-93.

[14] 于磊,林宗楷,郭玉钗,等. 多服务器系统中的负载平衡与容错[J]. 系统仿真学报,2001,13(3):325-328.
YU LEI, LIN ZHONG-KAI, GUO YU-CHAI, et al. Load balancing and fault-tolerant services in multi-server system[J] . Journal of System Simulation,2001, 13(3):325-328.

(编辑 侯 湘)