浙江科技学院学报,第 21 卷第 1 期,2009 年 3 月
Journal of Zhejiang University of Science and Technology
Vol.21 No.1, Mar. 2009

基于集群的 Web 服务器负载均衡算法研究

郑　祺
(浙江科技学院 信息与电子工程学院,杭州 310023)

摘　要:集群技术为 Web 服务带来了新的解决方案。针对传统负载均衡算法的一些不足,提出了一种临界区加速递减权值的动态请求负载均衡算法,通过负载权值的等效变换来简化算法,最大限度满足系统最大吞吐率,减少系统响应时间。测试表明,算法达到了较好的负载均衡效果,明显优于传统算法。

关键词:集群;负载均衡;动态反馈;权值
中图分类号:TP393　　**文献标识码**:A　　**文章编号**:1671-8798(2009)01-0015-04

Research on load balancing algorithm of Web server based on cluster

ZHENG Qi
(School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China)

Abstract: The cluster technology brings a new solution for the Web service. Focusing on some shortcomings in the conventional algorithms, the author presents dynamic load balancing algorithm with multiplicative decrease in critical area. To simplify the distribution of the load balance, increase the throughput rate and decrease the response time of the system, an equivalent load-alternant was used. The test result shows that this algorithm improves the efficiency of the servers more effectively compared with traditional algorithms.

Key words: cluster; load balancing; dynamic feedback; weight

近年来,随着 Internet 的飞速发展以及电子商务、多媒体技术等的广泛应用,网络服务器的访问量大大增加,这对 Web 服务器的处理能力提出了严峻的挑战,使提高服务器性能及数据的处理能力已经成为一个急需解决的问题。集群服务器以其高可扩展性、高可靠性和高性价比,为 Web 服务器系统带来了新的解决方案[1]。

Web 服务器集群系统一般由局域网内通过高速网络连接的一组通用服务器(这些服务器可以是同构的,也可以是异构的,通常称为节点服务器)构成,这个系统对外相当于一台高性能的 Web 服务器。一般由一台特殊的服务器(请求分配服务器,也可称控制器)接受请求,并按照某种策略动态地分配到各个节点服务器上处理。请求分配服务器可以在 HTTP,DNS,TCP,IP 等不同层次上实现[2]。负载均衡是集群系统正常工作的核心部分,其主要目的是把任务合理地分配到集群的各个节点,使各节点均衡地负载,以实现整个系统的均衡负载,保证系统

收稿日期:2008-09-16
作者简介:郑　祺(1974—　),男,江苏无锡人,讲师,硕士,主要从事计算机网络并行处理研究。

的处理能力和服务质量[3]。

1 传统负载均衡算法

目前经常使用的负载均衡算法包括两大类：一类是静态负载均衡算法如转轮算法，它不考虑后台服务器执行的负载情况，仅按照预先设定的决策来进行任务的分配；另一类是动态负载均衡算法，它根据系统当前的负载情况动态分发请求，通常使用的算法有最小请求优先算法和最小期望等待时间算法。动态负载均衡算法的问题在于：一段时间内将所有新到达的请求消息发送给请求量最小或者最小期望等待时间的后台服务器，如果在这段时间内新到达的请求比较多则反而会破坏负载均衡[4]。

采用基于负反馈机制的动态负载均衡算法能够通过周期性的反馈信息不断调整用户请求分布的比例，充分考虑到每个节点的负载水平和响应能力，尽可能避免因节点负载不均衡而引起的部分节点过载的情况，从而有效提高集群的整体性能。

大多数基于参数的加权负载分布策略和选择加权比例算法通常是根据 CPU 利用率、内存利用率、存活的请求连接数、磁盘 I/O 频度以及响应时间等参数来计算服务器的负载权值，然后根据负载权值来决定请求分配方案。这种算法的最大难度在于需要为每个因素选择一个合适的计算系数，如果系数选取不当，很可能适得其反[5]。

本文的后续部分将提出一种算法相对简单的临界区加速递减权值的动态请求负载均衡算法，以避免一般动态均衡算法不断读取后台负载造成系统网络流量过大、算法复杂、负载均衡器本身成为系统瓶颈的缺陷。

2 动态负载均衡算法

算法设计思想：为了保证控制器(Director)的转发效率，控制器不检查每个请求任务的类型，只周期性地监测各节点服务器(Node Server)的负载状况，通过反馈节点服务器的响应时间，控制器等效计算出每个节点服务器当前的负载权值，并根据实际负载状况进行相应调整，然后再采用加权轮转法对用户请求进行调度，此外为尽量避免或减少突发请求情况对集群性能的影响，考虑进一步引入输入参数来动态调整节点的负载权值。

2.1 权值计算

节点的初始权值计算主要考虑以下参数：CPU 处理速度、内存容量、系统 I/O 速率和网络带宽等，节点初始权值的计算公式如下：

$$\mathrm{Load}(N_i)=R_1\times K_i\times\frac{L_{\mathrm{CPU}}(N_i)}{\mathrm{BASE}_{\mathrm{CPU}}}+R_2\times\frac{L_{\mathrm{M}}(N_i)}{\mathrm{BASE}_{\mathrm{M}}}+R_3\times\frac{L_{\mathrm{I/O}}(N_i)}{\mathrm{BASE}_{\mathrm{I/O}}}+R_4\times\frac{L_{\mathrm{NET}}(N_i)}{\mathrm{BASE}_{\mathrm{NET}}}\quad(1)$$

式(1)中：$L_f(N_i)$ 表示节点 N_i 当前某一参数的值，式(1)中依次表示为：CPU 处理速度、内存容量、系统 I/O 速率和节点的网络带宽；K_i 为处理器系数(根据节点的处理器数量确定)；BASE_f 是某一参数的基准值，依次为基准 CPU 处理速度、基准内存容量、基准 I/O 速率和基准网络带宽，这些基准值可根据集群系统的实际情况统计确定。R_i 是为每个参数设定的一个可调常量系数，用以区分各个参数的重要程度，可以根据系统实际测试进行设置，以期达到最佳状态，其中 $\sum R_i=1$。一组典型设置可采用：(0.3，0.2，0，0.5)[6]。

考虑到采用动态反馈算法确定权值需要周期性地对节点进行参数采集、计算，这需要消耗一定的系统及网络资源且存在计算系数选择困难的问题，因此考虑采用一种相对简单的方法来进行节点的权值等效变换，以达到提高集群整体效率的目标。

通过对单台服务器的响应状况(如图 1)和响应时间(如图 2)[5]进行观察，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突然下降为零并会持续一段时间。这是系统软件为防止死机而采取的一种措施，也叫“假死”现象，它会对服务器的性能产生严重影响，因此必须避免这种情况的发生。当服务器出现“假死”现象时，其响应时间会急剧增加，会超过 $T_{\max}$。这时可以认为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 2 还可看出，只有一个最基本的请求任务加载到空载服务器时，服务器的请求响应时间为 T_{bas}，如果忽略此请求的影响，则 T_{bas} 就是服务器维持运转的基本负载[5]。可将($T_{\max}-T_{\mathrm{bas}}$)定义为服务器对外提供的服务能力，设 T_{now} 为某台服务器当前的响应时间，则可用($T_{\max}-T_{\mathrm{now}}$)来表示服务器当前的剩余服务能力。由此可以得出节点当前权值的等效变换：

$$W(N_i)=\mathrm{Load}(N_i)\times\frac{T^i_{\max}-T^i_{\mathrm{now}}}{T^i_{\max}-T^i_{\mathrm{bas}}},\quad\begin{cases}T^i_{\mathrm{now}}\leqslant T^i_{\mathrm{bas}},T^i_{\mathrm{now}}=T^i_{\mathrm{bas}}\\T^i_{\mathrm{now}}\geqslant T^i_{\max},T^i_{\mathrm{now}}=T^i_{\max}\end{cases}\quad(2)$$

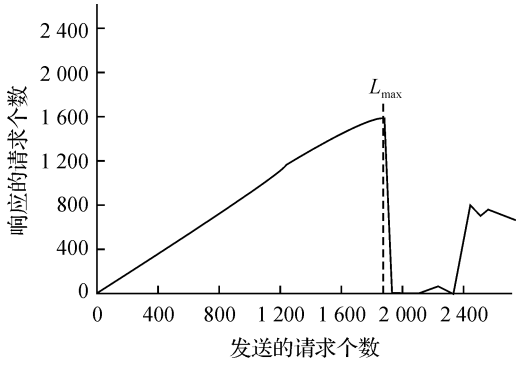

图 1　发送的请求个数与响应的请求个数的关系

Fig .1　The relationship between the number of request and response

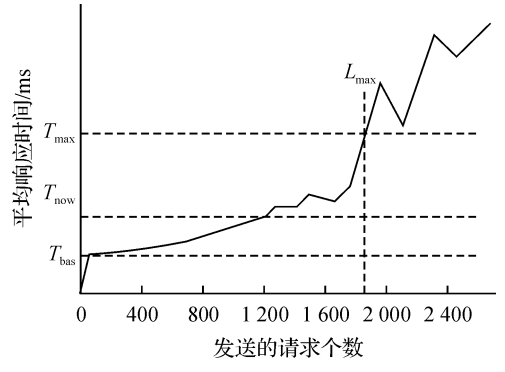

图 2　发送的请求个数与平均响应时间的关系

Fig .2　The relationship between the number of request and the average response time

2 .2　控制器调度算法

控制器根据各节点的反馈信息可计算出各节点的当前权值,然后再采用加权轮转法对用户请求进行调度,即根据节点权值的大小按照轮转的方式将用户请求分配到各节点去,权值越高的节点分配到的用户请求就越多,每个节点所分配到的用户请求数是按其权值占权值总和的比例来确定的。

2 .3　权值的动态调整

从图 2 可看出,当请求负载到达一定量时,对应的响应时间的变化会出现抖动现象,这表明系统将进入饱和状态,将此区域定义为系统进入饱和状态前的临界区,从响应时间上看为 T_{cri}～T_{max} 区间[5]。为达到抑制服务器进入饱和状态的目的,可通过适当降低进入临界区的服务器权值来减轻下阶段的工作负载。具体调整公式如下:

$$\text{Weight}(N_i)=W(N_i)\times K,$$

$$K=\begin{cases}1, T^i_{now}\leqslant T^i_{cri}\\ \alpha\times\dfrac{T^i_{max}-T^i_{now}}{T^i_{max}-T^i_{cri}}, T^i_{now}>T^i_{cri}\end{cases}\qquad(3)$$

K 是调节系数,只有当服务器负载进入临界区后才产生作用,系统负载越重则 K 值越小,这样可以使临界深度大的服务器加快递减速度,以避免其进入饱和状态。α 值可用来调节递减速度,通常取 0 .5。

在实际应用中,管理员可根据实际情况给每个节点服务器设定一个阀值,当 Weight 值小于阀值时,可认为该节点可能已进入“假死”状态,此时将不再向该节点分配请求,并同时向管理员发送过载警告。

3　性能测试

衡量负载均衡算法有两个重要评价指标:平均吞吐量和平均响应时间。为此,使用 NAT 方式构造了一个 LVS 集群[7],集群内包括 1 台控制器和 3 台节点服务器,配置分别为单 CPU 服务器、双 CPU 服务器、单 CPU 服务器和普通 PC;另使用 2 台 PC 作为客户端对集群进行模拟加压测试,使用的工具是 Linux 下的 Httperf,请求事件流为符合负指数分布的泊松事件流,服务器端用于测试的负载文件由长静态网页和多媒体文件组成。选择的对比测试算法为加权轮转法(WRR)[8]和加权最小连接数法(WLC)[8]。性能对比测试结果如图 3 和图 4 所示。

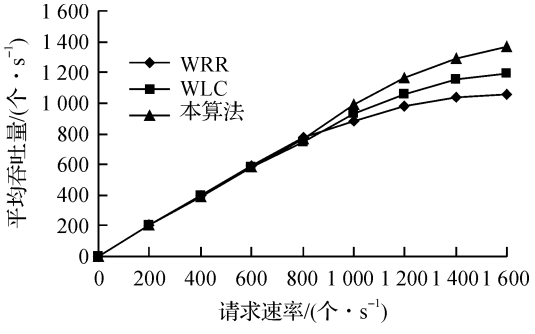

图 3　平均吞吐量对比

Fig .3　The comparison of average throughput

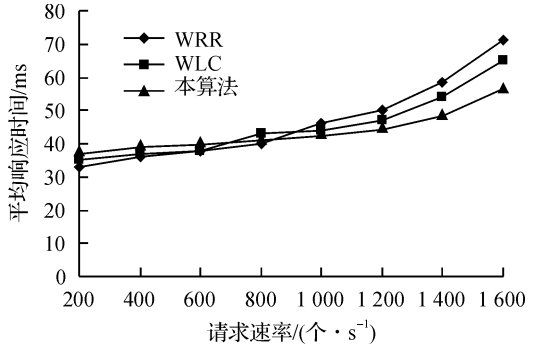

图 4　平均响应时间对比

Fig .4　The comparison of average response time

从测试结果可以看出,本算法无论在平均吞吐量以及平均相应时间方面均优于其他两种对比算法,达到了较好的负载均衡效果,可有效提高集群的整体性能。

4 结 语

集群负载均衡策略是提高集群整体性能的关键,考虑到用户请求间的差异性和集群内部节点服务器间性能的差异性,通过简单的等效计算节点权值和对进入临界区的节点采取加速递减权值的策略来合理分配用户请求,以平衡节点的负载,充分利用各节点的资源。通过测试表明,本算法基本可行并明显优于常用的加权轮转法和加权最小连接数法,可以有效提高集群的性能,尤其是在集群内部节点服务器性能不一的情况下,可以取得较为有效的负载均衡分配效果。

参考文献:

[1] CHEN Li-chuan,CHOI Hyeon Ah. Approximation algorithms for data distribution with load balance of Web servers[C]//Proceedings of IEEE International Conference on Cluster Computing,Washington DC: IEEE Computer Society Press,2001:274-281.

[2] 魏利蜂,左明,王志晓,等.一个基于集群的Web服务器负载平衡模型[J].计算机工程,2005,31(10):116-118.

[3] 田绍亮,左明,吴绍伟.一种改进的基于动态反馈的负载均衡算法[J].计算机工程与设计,2007,28(3):572-573.

[4] 张吴,廖建新,朱晓民.增强型动态反馈随机分发负载均衡算法[J].计算机工程,2007,33(4):97-99.

[5] 郭成城,晏蒲柳.一种异构Web服务器集群动态负载均衡算法[J].计算机学报,2005,28(2):179-184.

[6] 章文嵩.Linux服务器集群系统(四)[EB/OL].[2008-05-20] http://www.ibm.com/developerWorks/cn/linux/cluster/lvs/part4/index.htm1.

[7] 马双良,张英敏,宋丽君.基于LVS和计算任务的实时集群负载均衡方法[J].计算机工程与设计,2007,28(20):4934-4937.

[8] COLAJANNI M, CARDELLINI V, YU P S. Dynamic Load Balancing in Geographically Distributed Heterogeneous Web Servers[C]//Proceedings of 18th IEEE International Conference on Distributed Computing System (ICDCS 1998),Washington DC: IEEE Computer Society Press,1998:295-303.