

基于负载权值的负载均衡算法

张玉芳, 魏钦磊[†], 赵 膺

(重庆大学 计算机学院, 重庆 400030)

摘要: 为解决服务器集群负载分配不均的问题, 综合考虑节点负载和节点性能信息, 提出了基于负载权值的动态反馈负载均衡算法。利用负载权值选择分配负载的节点集合, 保证性能高的节点分配到较多的负载; 引入负载差值计算节点分配负载的概率, 使得负载的分布更加均匀; 通过负载增量及负载修正保持系统的稳定性。使用 OPNET 仿真软件进行测试, 结果表明该算法能有效提高负载均衡效率, 有较好的负载均衡效果。

关键词: 动态反馈; 负载差值; 负载权值; 负载增量; 负载均衡效率; OPNET

中图分类号: TP393; TP301.6 **文献标志码:** A **文章编号:** 1001-3695(2012)12-4711-03

doi:10.3969/j.issn.1001-3695.2012.12.080

Load balancing algorithm based on load weights

ZHANG Yu-fang, WEI Qin-lei[†], ZHAO Ying

(College of Computer Science, Chongqing University, Chongqing 400030, China)

Abstract: In order to solve the problem of uneven distribution of load in server cluster, considering the load and performance information of nodes, this paper presented a load balancing algorithm with dynamic feedback based on the load weights. Using load weights to help choosing nodes for load distribution, the algorithm assured that high performance nodes got more load. And it brought load difference to calculate the probabilities of those nodes to make the distribution of load uniformly. By using load increment and load amendment, the system become stability. Using the OPNET simulation software for testing, the results show that the algorithm can availably improve the efficiency of load balancing and has good load balance result.

Key words: dynamic feedback; load difference; load weights; load increment; efficiency of load balancing; OPNET

随着互联网技术的发展和网络服务的普及, 网络业务量和数据流量呈几何级数增长, 如 Facebook 用户每天公开分享 40 亿条信息, 包括状态更新、图片等。网络拥塞和服务超载成为网络服务面临的严峻问题。网络服务器集群则提供了一种高性能、低成本的网络服务解决方案, 以其可扩展性强、管理维护方便、处理能力突出等优点得到了国内外研究机构的关注和迅速发展^[1]。

集群(cluster)是一组相互独立、通过高速网络互连且以单一系统的模式加以管理的计算机。集群的内部结构对客户端是透明的, 客户端只能感觉到一个高性能的服务器在提供服务。集群系统中的一个重要问题就是负载均衡(load balancing), 其目标是根据各服务器性能来分配与其匹配的任务, 最大限度发挥集群的优势来提供更好的网络服务质量, 充分利用服务器的各种资源, 最小化应用的执行时间, 属最优化的理论范畴^[2]。

负载均衡可以直接基于硬件产品, 也可以采用软件来实现^[3]。硬件实现需要在服务器和外部网络间安装负载均衡器, 集群系统的处理能力和均衡性能较强, 但它只是从网络层来判断数据流量, 无法有效掌握服务器的实时状态, 灵活性不够。软件实现主要依靠负载均衡算法来指导任务的分配, 能更好地监视集群系统的状况, 性价比比较高, 但均衡能力受服务器

本身性能的影响, 因此对负载均衡算法的复杂度要求较高。

1 问题的提出

现有的负载均衡算法主要分为静态和动态^[4,5]两类。静态负载均衡算法以固定的概率分配任务, 不考虑服务器的状态信息, 如轮转算法、加权轮转算法、目标地址散列法、源地址散列法等; 动态负载均衡算法以服务器的实时负载状态信息来决定任务的分配, 如最小连接法、加权最小连接法、基于位置的最小连接法、带复制的基于位置的最小链接法等。一个较简单的动态均衡算法是最小连接法^[6], 每次分配请求时计算各服务器的当前连接数, 将请求分配到链接数最小的服务器。

在实际应用中, 影响客户访问频率、节点网络负载的因素很多, 且难以预测; 在系统长时间运行的情况下, 节点负载也无法得到及时修正, 会发生倾斜, 均衡效果往往不能令人满意。因此很多学者对负载均衡算法进行研究, 提出了很多改进的动态负载均衡算法, 如国防科技大学的刘健等人提出了基于动态反馈的负载均衡算法^[7], 该算法考虑服务节点的性能与服务节点实际负载两方面的因素, 用来指导任务的分配, 并通过动态反馈机制及时修正各节点的负载, 保证系统长时间平稳运行。虽然该算法考虑了节点的性能和实际的负载, 但并没有充分应用, 如在选取分配任务的服务器节点时没有考虑节点性能这一重要因素, 但是节点负载轻重跟节点负载与节点性能都有关系, 单独考虑一种因素会使得算法性能降低。

本文提出了一种基于负载权值的动态反馈负载均衡算法,

收稿日期: 2012-04-06; 修回日期: 2012-05-15

作者简介: 张玉芳(1965-), 女, 上海人, 教授, 主要研究方向为数据挖掘、网络入侵检测、并行计算; 魏钦磊(1985-), 男(通信作者), 山东曹县人, 硕士研究生, 主要研究方向为负载均衡、计算机网络与通信(weiqinlei@gmail.com); 赵膺(1967-), 男, 博士研究生, 主要研究方向为数据挖掘、网络安全。

综合利用节点的处理能力和实际负载等因素指导任务的分配,使得所有节点各尽其能,充分发挥集群系统的优势,保证系统在长时间运行的情况下不会发生较大的倾斜。

2 基于负载权值的算法

2.1 节点性能和负载的计算

集群系统中的节点 $S = (S_1, S_2, \dots, S_n)$ 往往是异构的,因此在任务分配时必须根据不同节点的性能区别对待,以达到能者多劳、负载均衡的效果。节点 S_i 的性能 $C(S_i)$ 主要从 CPU 数量 n 、CPU 频率 $C(C_i)$ 、内存容量 $C(M_i)$ 、磁盘 I/O 速率 $C(D_i)$ 、网络吞吐量 $C(N_i)$ 等指标来考虑,使用式(1)来计算。

$$C(S_i) = k_1 \times n \times C(C_i) + k_2 \times C(M_i) + k_3 \times C(D_i) + k_4 \times C(N_i) \quad i=0,1,\dots,n-1, \sum k=1 \quad (1)$$

节点 S_i 的负载 $L(S_i)$, 主要从 CPU 占用率 $L(C_i)$ 、内存占用率 $L(M_i)$ 、磁盘 I/O 占用率 $L(D_i)$ 、网络带宽占用率 $L(N_i)$ 等指标来考虑,使用式(2)来计算。

$$L(S_i) = k_1 \times L(C_i) + k_2 \times L(M_i) + k_3 \times L(D_i) + k_4 \times L(N_i) \quad i=0,1,\dots,n-1, \sum k=1 \quad (2)$$

其中: k 是各项指标的权值参数,反映不同类型的服务对各个指标的影响程度,如 FTP 服务主要对节点的网络吞吐量、硬盘 I/O 要求较大,而 HTTP 服务则侧重于节点 CPU 运算速度和内存。 k 在文献[3]中设置如下: $k = (0.2, 0.3, 0, 0.5)$, 在实际的应用过程中,该参数可以根据系统运行情况进行调节,以期达到更佳的效果。

2.2 负载权值和负载差值

节点的负载权值 $W(S_i)$ 定义为节点负载 $L(S_i)$ 与节点性能 $C(S_i)$ 的比值,采用式(3)来计算。权值越大,说明节点负载越重,从而可以根据负载权值的大小来决定任务的分配。

$$W(S_i) = L(S_i) / C(S_i) \quad i=0,1,\dots,n-1 \quad (3)$$

节点的负载差值 $\Delta L(S_i)$ 是指所有节点负载权值的最大值 $W_{\max}$ 与该节点的负载权值之差乘以该节点性能,采用式(4)来计算。

$$\Delta L(S_i) = (W_{\max} - W(S_i)) \times C(S_i) \quad i=0,1,\dots,n-1 \quad (4)$$

2.3 负载增量及节点负载修正

由于系统存在更新周期,在更新周期内无法获取各个节点的实际负载,需要引入负载增量,对节点的负载进行预测。负载增量 δ 是指某种类型的一个请求分配到某节点上,对该节点带来的负载量采用式(5)来计算,它与请求服务的类型相关,所以对于不同的服务要计算各自的负载增量。

$$\delta = L(S) / n \quad (5)$$

其中: $L(S)$ 为计算负载增量用到的节点负载大小, n 为该节点的请求数。为了提高系统的稳定性,负载增量 δ 在系统运行过程中自动调整。

在更新周期的间隔内,对新分配任务的节点负载要加上增量值,对有任务完成的节点负载要减去增量值,这样才能更真实地反映各个节点的实时负载,但这需要对各个节点任务的完成情况进行监测,必将增加服务节点的负担。为了避免这一现象的产生,本算法不实时检测节点在更新周期内完成的任务,而是将此因素考虑到负载增量的修正中去,如果节点在更新周期的间隔内完成的任务数大于新分配的任务数,那么节点的负载将减少,负载增量修正后的值也会减小,从而弥补节点有任务完成时对节点负载造成的影响,达到同样降低节点负载的效果。因此,负载修正采用式(6)来计算,其中 $C(S)$ 为计算负载增量时用到的节点性能。

$$L(S_i) = L(S_i) + \delta \times C(S) / C(S_i) \quad i=0,1,\dots,n-1 \quad (6)$$

2.4 基于负载权值的算法

算法设计思想:在节点的负载和运行状态的基础上,综合考虑节点的性能参数,通过负载权值指导新任务到来时备选节点集合的选取,通过负载差值计算备选节点集合中各节点分配任务的概率,利用负载增量等信息对节点负载进行修正,较好地解决负载分配不均的问题,提高负载均衡效率,增强集群系统的稳定性和灵活性。基于负载权值的动态反馈负载均衡算法描述如下:

a) 设定一个阈值 ε 。

b) 每到来一个新的任务请求,选取节点候选集 J 。首先选取节点 S_m , 其满足条件:

$$W(S_m) = \min \{ W(S_i) \} \quad i=0,1,\dots,n-1 \quad (7)$$

如果其他任意节点 S_i 满足条件:

$$W(S_i) < W(S_m) + \varepsilon \quad i=0,1,\dots,n-1 \quad (8)$$

则将节点 S_i 加入到集合 J 中。

c) 计算候选集合 J 中各个节点分配任务的概率 $P(S_k)$:

$$P(S_k) = \Delta L(S_k) / \sum_{i=0}^{n-1} \Delta L(S_i) \quad k=0,1,\dots,n-1, S_i \in J, S_k \in J \quad (9)$$

d) 根据集合 J 中各个节点的概率进行随机概率分配,将任务分配到 J 中的一个节点上,并使用式(6)修正该节点的负载。

3 算法性能分析

为了验证算法的性能,本文使用 OPNET Modeler 仿真软件进行实验,并与其他算法的效果对比。OPNET Modeler 提供了三层建模机制,分别在进程层、节点层和网络层进行由下到上的建模,同时在仿真的过程中它采用了离散事件驱动的模拟机理,能够准确地分析复杂网络的性能和行为,进行数据采集和统计^[8]。实验环境如图 1 所示。

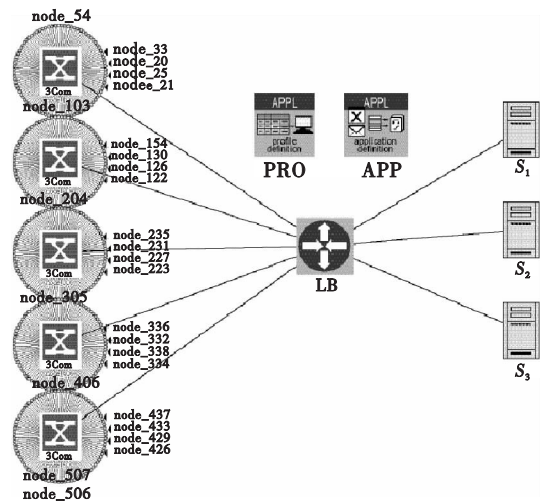


图1 实验网络拓扑图

为了检验算法对异构节点集群系统的均衡效果,集群使用三台性能不同的服务器组成,节点性能比例为 1:1.5:1.9,并提供 FTP 服务;客户端由 500 个节点组成,统一向负载均衡器发生请求;建立三个场景,分别对最小连接法(Connection 算法)^[6]、国防科技大学刘健等人改进的基于动态反馈的负载均衡算法(MTN 算法)^[7]和本文提出的基于负载权值的动态负载均衡算法(Weight 算法)进行模拟仿真。为方便计算,取参数 $k = (0.2, 0.3, 0, 0.5)$,更新周期为 20 s,仿真时间为 30 min。实验结果如图 2~4 所示。由图可以看出,最小连接法不能区分不同服务器的性能来分配任务,使得三个节点分配的任务相当;基于动态反馈的负载均衡算法虽然有了一定的改进,但是区分的效果不够明显;本文提出的算法则能更好地根据不同服务器的性能分配相应的任务,各服务的负载之比最接近于性能

之比,达到了预期的效果。

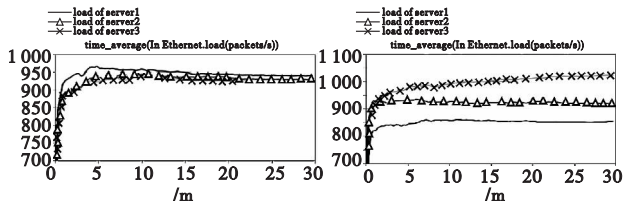


图2 Connection算法实验结果

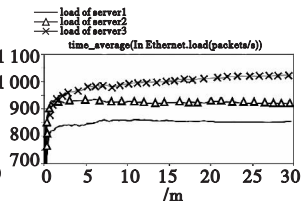


图3 MTN算法实验结果

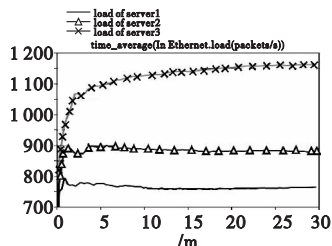


图4 Weight算法实验结果

负载均衡算法最终的目的是充分利用节点负载和性能的信息,将任务分配到集群节点上,使得各个节点的负载与其性能成正比^[9,10],为此引入了均衡效率对比三种算法的性能。均衡效率 $H(S_i)$ 是指节点实际的平均负载与按照算法分配理论上的平均负载的比值。为了简化计算步骤,取集群中的第一个节点作为参考系,使用式(10)计算。

$$H(S_i) = \frac{L(S_i)/L(S_1)}{C(S_i)/C(S_1)} \quad i=0,1,\dots,n-1 \quad (10)$$

由均衡效率定义可知,比值越接近于 100% 算法性能越好。三种算法的均衡效率结果如表 1 所示。

表 1 三种负载均衡算法的均衡效率对比

算法	服务器 S_1	服务器 S_2	服务器 S_3
Weight	100%	77.04%	77.39%
MTN	100%	72.42%	61.63%
Connection	100%	65.51%	51.44%

据表 1 可得,最小连接法的均衡效率最低,性能最高的节点均衡效率仅 51.44%,均衡效果最不理想;基于动态反馈的负载均衡算法根据不同节点的性能计算分配任务的概率,均衡效果有了较大的改进,但性能最高的节点均衡效率仅 61.63%;本文提出的算法,通过综合考虑节点的性能和负载等因素指导任务的分配,使得负载分配更加均匀、合理,各个节点分配的负载比例最接近于节点性能的比例,均衡效率达到 77.39%,相对于基于动态反馈的负

载均衡算法提高了 15.76%,均衡效果有了明显的改善。

4 结束语

负载均衡算法是负载均衡集群的核心部分,优秀的均衡算法能发挥集群的优势,充分利用集群的各种资源,为用户提供高质量的服务。通过分析常用负载均衡算法的优缺点,本文提出了一种基于负载权值的动态反馈负载均衡算法,该算法的主要特点是:充分应用节点的实际负载和性能两个因素,指导任务的分配,通过负载修正保证系统的稳定性。实验表明,该算法能更好地提高均衡效率,增强负载均衡集群的稳定性。

当然,本文提出的算法也存在不足,例如使用修正负载增量来抵消节点完成任务对负载造成的影响,有一定的滞后性。另外,均衡效率虽然有了一定的提高,但与理论上存在差距。在以后的实验中,将不断完善节点负载的分配和修正机制,进一步提高算法的均衡效率。

参考文献:

- [1] 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.
- [2] 陈伟,张玉芳,熊忠阳. 动态反馈的异构集群负载均衡算法的实现[J]. 重庆大学学报,2010,33(2):73-78.
- [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] 胡志刚,张艳平. 基于目标约束的分层动态负载均衡算法[J]. 计算机应用研究,2011,28(3):1105-1107.
- [5] BRYHNI H. A comparison of load balancing techniques for scalable Web servers[J]. IEEE Network,2000,14(4):58-64.
- [6] 买京京,龚红艳,宋纯贺. 集群系统中的动态反馈负载均衡策略[J]. 计算机工程,2008,34(16):114-115.
- [7] 刘健,徐磊,张维明. 基于动态反馈的负载均衡算法[J]. 计算机工程与科学,2003,25(5):65-68.
- [8] 操惊雷,周建国,秦磊华. 基于 OPNET 的网络压力仿真[J]. 计算机工程,2009,35(23):115-117.
- [9] SOKLIC M E. Simulation of load balancing algorithms: a comparative study[C]//Proc of ACM SIGCSE Bulletin. New York: ACM Press, 2002:138-141.
- [10] ALTMAN E, AYESTA U, PRABHU B J. Load balancing in processor sharing systems[C]//Proc of the 3rd International Conference on Performance Evaluation Methodologies and Tools. Belgium: ICST, 2008:35-48.

(上接第 4710 页)

- [4] LU Xing, DUAN Hai-xin, LI Xing. Identification of P2P traffic based on the content redistribution characteristic[C]//Proc of International Symposium on Communications and Information Technologies. [S. l.]: IEEE, 2007:596-601.
- [5] MATSUDA T, NAKAMURA F, WAKAHARA Y, et al. Traffic features fit for P2P discrimination[C]//Proc of the 6th Asia-Pacific Symposium on Information and Telecommunication Technologies. 2005:230-235.
- [6] BARTLETT G, HEIDEMANN J, PAPADOPOULOS C. Inherent behaviors for on-line detection of peer-to-peer file sharing[C]//Proc of IEEE Global Internet Symposium. Alaska: IEEE, 2007: 55-60.
- [7] ERMAN D, ILIE D, POPESCU A. BitTorrent traffic characteristics[C]//Proc of International Multi-Conference on Computing in the Global Information Technology. Washington DC: IEEE Computer Society, 2006: 42-49.
- [8] LLORET J, DIAZ J R, JIMENEZ J M, et al. Public domain P2P file-sharing networks measurements and modeling[C]//Proc of International Conference on Internet Surveillance and Protection. Washington DC: IEEE Computer Society, 2006: 10-16.
- [9] CHENG W Q, GONG J, DING W. Identifying BT-like P2P traffic by

the discreteness of remote hosts[C]//Proc of the 32nd IEEE Local Computer Networks. Washington DC: IEEE Computer Society, 2007: 237-238.

- [10] CONSTANTINOU F, MAVROMMATIS P. Identifying known and unknown peer-to-peer traffic[C]//Proc of the 5th IEEE International Symposium on Network Computing and Applications. Washington DC: IEEE Computer Society, 2006:93-102.
- [11] CHEN Zhen-xiang, YANG Bo, CHEN Yue-hui, et al. Online hybrid traffic classifier for peer-to-peer systems based on network processors[J]. Applied Soft Computing, 2009,9(2):685-694.
- [12] 杨国良. 国际 VoIP 流量特征分析[J]. 电信科学, 2007,57(6): 7-16.
- [13] CAMPBELL W M, SINGER E, TORRES-CARRASQUILLO P A, et al. Language recognition with support vector machines[C]//Proc of IEEE Odyssey: The Speaker and Language Recognition Workshop. 2004: 41-44.
- [14] 邓乃杨, 田英杰. 支持向量机: 理论、算法与拓展[M]. 北京: 科学出版社, 2009: 32-94.
- [15] WITTEN I H, FRANK E. Data mining: practical machine learning tools and techniques[M]. 2nd ed. San Francisco: Elsevier Inc, 2005.