

基于剩余负载率的动态负载均衡研究

段淮川¹, 胡 平²

(南京工业大学 信息科学与工程学院, 江苏 南京 210009)

摘 要: 为进一步提高集群系统服务的性能, 在对静态负载均衡和动态负载均衡的优缺点分析的基础上, 提出一种基于剩余负载率的动态均衡机制. 该均衡机制采用剩余负载率作为负载状态的评价标准; 针对服务器各节点性能的不同, 提出采用 BP 神经网络训练节点; 并设计了一种基于流表的静态分配策略和基于负载预测的动态分配策略相结合的任务分配策略来实现任务在集群系统各节点间的动态分配, 从而降低了服务器各节点之间任务重新调度的次数, 提高了集群系统的服务性能. 实验结果表明, 该均衡机制是可行的、有效的.

关键词: 剩余负载率; 负载均衡; BP 神经网络; 报文

中图分类号: TP393

文献标识码: A

文章编号: 1000-7180(2010)02-0141-04

Research on Load-Balance Based on Leftover Load Rate

DUAN Huai-chuan¹, HU Ping²

(College of Information Science and Engineering, Nanjing University of Technology, Nanjing 210009, China)

Abstract: A leftover Load Rate - based dynamic balance mechanism is proposed after analyzing static balance technology and dynamic balance technology on the purpose of improving the performance of cluster systems. This mechanism adopts leftover Load Rate as appraisable standard; BP neural network is used to train nodes of cluster systems because of different performance of the nodes; and a new distribution method which is based on static balance technology with a flow table and dynamic balance technology based on load - forecasting is proposed to distribute assignments to nodes of cluster systems. This distribution method reduces times of dispatching assignments between nodes, and improves the performance of cluster systems. The final simulation result suggests this balance mechanism is feasible and valid.

Key words: leftover load rate (LLR); load balance; neural network; message

1 引言

随着网络用户的增加, 当前对分布式系统的负载均衡技术的研究进一步深入. 目前, 最为广泛的方法是采用由用户浏览器、负载均衡器和 Web 服务器组成的三层结构 Web 系统, 在此基础上, 负载均衡方法分为静态均衡和动态均衡^[1]. 静态均衡通过平均分配任务给计算节点, 虽然能获得最优负载均衡结果, 但这种方法要求任务量必须是确定的值, 在现实网路中任务量是随时间的变化而变化的, 很难做到任务量确定^[2]. 动态均衡方法通过实时分析计算节点的负载状况和任务量, 选取负载最轻的节点提供服务(称为最轻负载分配模型), 动态分配任务, 相

对静态均衡技术更好的适应了分布式服务的要求^[3], 但它的缺点是, 在平均分配任务方面不如静态负载均衡, 并且未考虑到均衡服务器收集节点负载状况消耗的时间, 以及任务量和任务间的通信量的大小. 当任务量很大时, 负载均衡器对节点的繁忙计算会导致服务不及时, 而通信量很大时, 需要节点在任务调度上花费大量时间^[4].

兼顾静态负载均衡和动态负载均衡的优缺点, 文中采用剩余负载率作为负载状态的评价标准, 提出一种基于剩余负载率的负载均衡方法, 并在一个 5 节点分布式网络环境中进行仿真模拟实验. 实验结果表明, 与传统的均衡策略相比, 文中的均衡策略具有更好的均衡效果.

2 集群系统架构设计

文中对集群系统架构的设计如下所述,服务器节点和 Internet 之间设置两个 Linux 虚拟服务器(Linux Virtual Server, LVS),一台为主 LVS;另一台为备用 LVS,以防主 LVS 出现故障,以 J2EE 为运行平台,都装有负载均衡分配模块,LVS 与集群服务器之间保持不间断联系,节点服务器每隔一个很小的时间段就获取一次自身的剩余负载能力值并送给 LVS;LVS 每次接收到节点传来的数据后,计算并比较一次集群服务器的负载状态.系统处理过程如下:

- (1) 客户端向服务器提出请求;
- (2) 根据 J2EE 的有状态会话 Bean 的特性,判断请求来的 IP 地址是否在 IP 分配表中即检查服务器集群中是否已经有了对该 IP 地址请求的处理;
- (3) 若是新的 IP 地址,LVS 根据节点发送的负载状况,计算并分配服务节点;
- (4) 若是旧的 IP 地址,即同源请求,则选择原来处理该 IP 地址的服务器节点继续处理该 IP 地址的请求.

其中,为兼顾计算的复杂以及负载值的准确性,我们选取 CPU 资源和内存使用情况作为负载状态的计算因子.

3 基于剩余负载率的动态负载均衡机制

3.1 基本思想

采用剩余负载率作为节点负载的评价标准,利用 BP 神经网络学习确定节点各项硬件性能指标的权重,并根据学习结果计算各节点的剩余负载率.为避免 LVS 因提取节点负载状态而造成自身成为系统瓶颈的问题,文中采用前人的研究成果,由节点自己收集自身的负载状况,并发送给 LVS^[5],LVS 通过 BP 神经网络计算得到剩余负载率,并以剩余负载率为权值划分客户端发来的报文流,同时自动产生一个索引号,并记录在流表中,然后按剩余负载率对应分配到节点上,最后,根据索引号得到最终输出.下面讨论剩余负载率的计算和任务分配策略.

3.2 剩余负载率计算

文中提出采用剩余负载率作为负载状态的评价标准,不仅考虑到动态负载均衡对节点负载的动态预测能力和自适应能力,而且兼顾了静态负载均衡的优点,同时有效的解决了下列问题:

- (1) 提高了评价标准的准确性并缩短请求的响

应时间

剩余负载率用来标识集群中各个节点的剩余负载占总剩余负载的比率.将剩余负载率作为评价标准,用 BP 神经网络分别对各节点训练性能指标的权重后计算得到的剩余负载能力更切合实际节点的配置,可以避免服务器节点间由于硬件配置不同而导致 LVS 对当前各服务器节点的剩余负载能力做出有偏差的判断.另外,客户端请求按剩余负载率对应依次被分配到节点上,减少了请求因等待 LVS 重复收集比较各节点负载状态而花费的等待时间,大大加快了服务器响应客户端请求的速度.

(2) 准确预测各节点剩余负载的能力

问题描述如下:在某个非常短的时间片内,有若干个客户端向服务器端同时发出若干请求,负载均衡器根据预测的负载状态选择负载最小的那个节点处理请求,而在后续请求到达时,LVS 记录的还是第一个请求到达时的负载排名,那么这些在非常短的时间片内到达的请求就可能都被分配给负载最小的那个服务器节点,这就可能导致该节点由负载最小转变为负载过重,解决这个问题,传统的方法是通过节点之间的任务重新调度来实现,但是由于任务的通信量通常很大,这又会造成节点在任务调度上耗费大量资源.显然,这将成为集群服务的一个缺陷.本文采用的基于剩余负载率的任务分配策略在第一次分配任务时,就尽量控制节点在任务执行时产生新的不平衡.

剩余负载率的计算方法描述如下:设有一组服务器集合 $P = \{P_1, P_2, \dots, P_n\}$,当在非常短的时间片内有 W 个客户端请求到达时,系统预测当前各服务器节点资源状态剩余 L_i (CPU 和内存使用状况), $1 \leq i \leq n$,并传给 LVS.

$$L_i = \phi L_c + \varphi L_m \quad (1)$$

$$\phi + \varphi = 1 \quad (2)$$

其中, L_c 为 CPU 剩余; L_m 为内存剩余; ϕ 为 CPU 权重; φ 为内存权重; ϕ 和 φ 值的确定采用 BP 神经网络学习获取.

根据 Rumelhart 的误差反响传播算法设计三层神经网络拓扑结构模型^[6],其中包含输入层、中间层和输出层.设置 P 个样本集合为 $SB = \{SB_1, SB_2, \dots, SB_p\}$,第 k 个样本 $SB_k = \{L_k, T_k\}$, $L_k = (L_{ck}, L_{mk})$ 为二维输入向量; T_k 为一维输出向量;实际输出向量为 O_k ,取指标函数为:

$$E = \frac{1}{k} \sum_k (T_k - O_k)^2 \quad (3)$$

其中, E 为均方差. 网络学习规则利用梯度法求给定误差函数极小点, 训练开始时, 置各权值和阈值的初始值为小的随机数值, 输入训练样本, 计算网络实际输出, 与期望输出对比, 并由比较结果修正权值和阈值, 并判断均方差 E 是否满足精度要求, 满足则输出结果, 不满足则用修正后的权值重新训练, 直到均方误差 E 满足精度要求, 网络训练结束, 并保存权值. 算法流程图如图 1 所示.

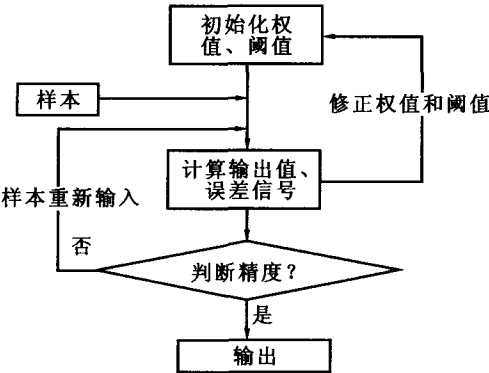


图1 算法流程图

则集群服务节点负载剩余总和为:

$$Q_i = \sum_{i=1}^n L_i \tag{4}$$

其中, Q_i 为集群服务总负载剩余. 第 i 台节点服务器剩余负载占总的剩余负载的比率, 即剩余负载率 E_i 为

$$E_i = \frac{L_i}{\sum_{i=1}^n L_i} \tag{5}$$

3.3 任务分配策略

基于 Web 的客户端请求主要是 TCP 报文流, 服务器端需要处理的也主要是 TCP 报文流. 根据这一原理, 文中采用基于流表的静态分配策略和基于动态负载预测的动态分配策略相结合的任务分配策略来实现负载的动态分配, 以求尽量减少对节点之间的任务重新调度. 流程如图 2 所示.

设节点群为 $P = \{P_1, P_2, \dots, P_n\}$, 集群处于运行状态, 某时刻输入流量为 M , 系统计算得到的剩余负载率 $E = \{E_1, E_2, \dots, E_n\}$, 对节点群的 IP 地址用剩余负载率值标记. LVS 以剩余负载率作为权值, 采用 Hash 函数 $f(x)$ 将客户端报文流剩余负载率值划分成不同大小的 n 份, 用 $\{E_1M, E_2M, \dots, E_nM\}$ 表示被划分的 n 份流量, 同时将分配结果记录到流表中. 对于每一个输入报文, 在划分的同时用 Hash 函数 $f(x)$ 计算其流表索引号, 并记录到流表中, 流表的表头构建如表 1 所示.

表 1 流表的表头结构

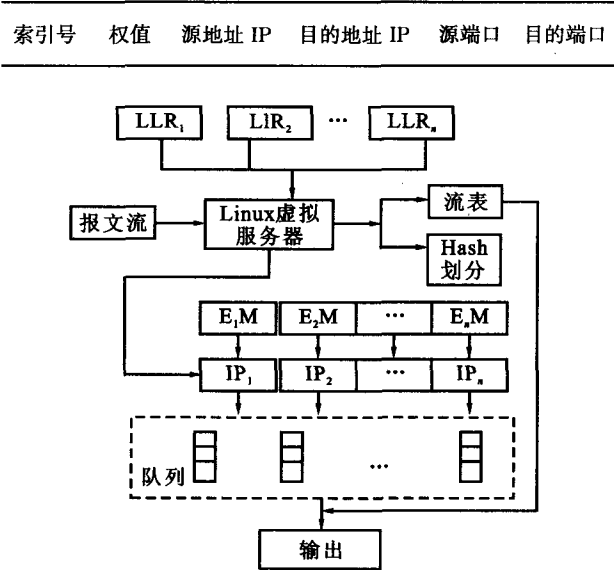


图2 任务分配流程图

LVS 将划分好的报文流分组分别按照剩余负载率值对应分配到已经标记了 IP 地址的 n 个节点上, 各个节点完成数据处理后根据索引号得到报文的最终输出节点.

4 实例测试

文中采用 MATLAB 作为样本训练工具. 测试环境中, 集群服务器为 5 台 $\{P_1, P_2, P_3, P_4, P_5\}$, LVS 为 2 台. 在各台服务器上安装测量 CPU 和内存使用状况和发送数据的模块, 并设置该模块每 5 秒钟计算一次剩余负载能力值 L_i , 同时将值发送给 LVS. 在 LVS 上安装接受数据和分析模块, 接收到各节点发送的数据就计算一次各个服务器节点的剩余负载率. 每隔 60 秒观察一次各台服务器的 CPU 和内存的工作状态. 文中的 BP 算法样本数据采用实验室数据, 训练网络, 并保存结果. 学习步长 η 取 0.3, 误差信号计算为 0.2, 最大允许误差为 0.0001 以各个服务器中的负载的均方差来衡量. 均方差公式为: $D(X) = E(X^2) - (E(X))^2$. 实验以最轻负载分配模型作为对比, 实验数据如表 2 所示. 实验结果如图 3、图 4 所示.

表 2 实验数据对比

分配策略	项目	P_1	P_2	P_3	P_4	P_5	$D(X)$
改进的算法	CPU	52%	48%	53%	36%	71%	0.0126
	内存	30%	42%	51%	28%	61%	
	负载	0.454	0.462	0.524	0.336	0.68	
最轻负载分配	CPU	35%	48%	30%	30%	96%	0.0601
	内存	26%	40%	36%	26%	90%	
	负载	0.323	0.456	0.318	0.288	0.942	

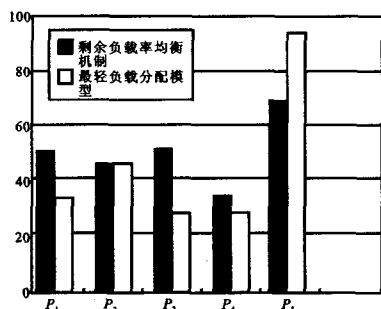


图3 60 秒时, CPU 运行状态

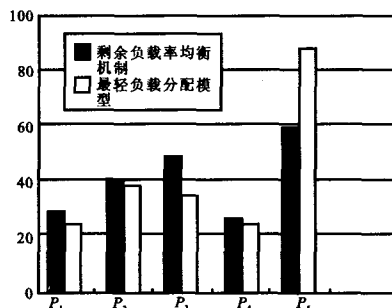


图4 60 秒时, 内存运行状态

由图3、图4可以看出,同一组集群服务器条件下,采用基于剩余负载率的均衡机制得到的各节点的均衡程度优于最轻负载分配模型的均衡策略。

5 结束语

文中提出一种基于剩余负载率的动态负载均衡

机制,在动态预测的基础上同时又引用了静态均衡技术的优点,进一步提高了集群服务各节点间任务分配的合理性,改善了系统运行的效率。

参考文献:

- [1] 余燕芳,陆军. 基于改进遗传算法的服务器端负载均衡算法[J]. 微电子学与计算机, 2007,24(7):146-148.
- [2] 熊伟,谢冬青,刘洁. 一种结构化 P2P 协议中的负载均衡方法[J]. 微电子学与计算机, 2008,25(10): 76-79.
- [3] Shivaratri N G, Krueger P, Singhal M. Load distributing for locally distributed systems[J]. IEEE Computer, 1992, 25 (12):33-45.
- [4] Cenova Z, Christensen K J. Challenges in: URL switching for implementing globally distributed web sites[C]// Proc of the Workshop on Scalable Web Services. Canada: Toronto, 2000:89-94.
- [5] 凌云,周华锋. 面向异构集群系统的动态负载均衡技术研究[J]. 计算机工程与设计, 2008, 29(12):3068-3070.
- [6] 杨建刚. 神经网络实用教程[M]. 浙江:浙江大学出版社,2003.

作者简介:

段淮川 男,(1985-),硕士.研究方向为计算机控制.

胡平 男,(1962-),硕士生导师,副教授.研究方向为过程控制.

(上接第 140 页)

参考文献:

- [1] 杨元原,马文平,白晓峰. 一种混合的 Tor 匿名系统[J]. 计算机应用研究, 2007(10):141-144.
- [2] Linux. 内核与用户空间通信之 netlink 使用方法[EB/OL]. [2007-06-18][2008-02-91]. <http://blog.cs-dn.net/zhoujunyi/archive/2007/06/18/1656759.aspx> 2008.2.1
- [3] Netfilter. Netlink 子系统[EB/OL]. <http://www.netfilter.org/2008.2.5>
- [4] 章晓明,杜春燕,陆建德. Netlink 消息通信机制的研究和在 IPsec VPN 中的应用[J]. 微电子学与计算机, 2007, 24(2):135-138.
- [5] Tsocks. The homepage of Tsocks[EB/OL]. <http://tsocks.sourceforge.net/2008.2.5>
- [6] Richard Johnsonbaugh, Martin Kalin. ANSI C 应用程序设计[M]. 杨季文,吕强,译. 北京:清华大学出版社, 2006.

作者简介:

周彦伟 男,(1986-),硕士研究生.研究方向为网络匿名.

吴振强 男,(1968-),博士,副教授.研究方向为匿名通信技术、可信计算、自适应安全.

肖梁 男,(1986-).

王宁 女,(1985-),硕士研究生.研究方向为无线网络安全.