

# 一种多层网络下动态负载均衡算法

刘立帮, 黄刚

(南京邮电大学 计算机学院, 江苏 南京 210003)

**摘 要:**分布式系统由若干个独立的节点组成,一些节点由于接收到大量请求而过载,还有一些节点却负担较少的请求任务。通过负载均衡技术可以使节点间的负载分配更加合理,最大化利用服务器集群的处理能力,达到扩展服务器集群的带宽和增加吞吐量,加强网络数据处理能力,提高网络的灵活性和可用性的目的。传统的集中式负载均衡方案采用静态负载均衡算法,由控制器全权负责任务分配。它的优点是功耗低而且稳定性强,缺点则是负载均衡效果不是最佳,总体处理速度较慢,中央控制器节点由于负担重容易成为系统瓶颈。同时,它的系统扩展在大规模集群中表现差。相比之下完全分布式方案是可扩展的,由于所有节点既是处理节点,也是分发器,而调度器只负责任务调度,从而减轻了控制器的负担,避免成为系统瓶颈。提出了一种异构分布式计算系统集群的负载均衡策略。该算法采集各个节点 CPU 使用率、存储器使用率两个系统参数,以决定各节点的工作量。同时,设计两层结构,解决全局通信负担较重的问题。仿真结果表明,该算法有效提高了负载均衡的效率。

**关键词:**集群负载均衡;分布式系统;异构网络;节点虚拟化

**中图分类号:**TP301.6

**文献标识码:**A

**文章编号:**1673-629X(2017)02-0051-05

doi:10.3969/j.issn.1673-629X.2017.02.012

## A Dynamic Load Balancing Algorithm in a Multi-level Network

LIU Li-bang, HUANG Gang

(School of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

**Abstract:** A distributed system consists of several independent nodes, in which some nodes may be overloaded due to massive requests arrivals, and another some are idle without any requests. Load balancing techniques can be used to effectively distribute the load between nodes to reach the purpose of extending bandwidth of server clusters, increasing its throughput, enhancing network data processing capability, improving network flexibility and availability. Traditional centralized load balancing adopts static load balancing algorithm, solely responsible for the tasks assigned by the controller. Its advantage is low power consumption and high stability, and its disadvantage is not the best in the load balancing effect and slow overall processing speed. Due to the heavy burden, the central controller node can easily become a bottleneck. At the same time, its system scalability is poor, with bad performance in large scale cluster. By contrast, a fully distributed solution is scalable, because all nodes are both processing nodes and the dispatcher, while the load scheduler is only task scheduling, thereby reducing the burden on the controller to prevent it from becoming a system bottleneck. A heterogeneous distributed computing systems in the cluster load balancing strategy is proposed. The algorithm requires the CPU usage and memory usage to determine the workload of each node. At the same time, two-level structure is designed to solve the problem of heavier global communications burden. Simulation results show that the algorithm can effectively improve the efficiency of load balancing.

**Key words:** cluster load balancing; distributed systems; heterogeneous network; node utilization

## 0 引言

随着用户需求的增长和云计算的日益普及,以及 Internet 上广泛的分布式应用对服务质量(QoS)需求的增长,各种服务应用对网络所能提供的 QoS 提出了更高的要求。与此同时,企业间的竞争也要求服务提

供商能提供更加快速、稳定的服务<sup>[1]</sup>。通过负载均衡(Load Balance)技术可以使节点间的负载分配更加合理,最大化利用服务器集群的处理能力,达到扩展服务器集群的带宽和增加吞吐量,加强网络数据处理能力,提高网络的灵活性和可用性的目的。服务器集群通过负载均衡技术可以将大量的数据流量和并发访问均衡

收稿日期:2016-03-30

修回日期:2016-07-05

网络出版时间:2017-01-10

基金项目:国家自然科学基金资助项目(61171053);南京邮电大学基金(SG1107)

作者简介:刘立帮(1988-),男,研究方向为云计算与虚拟化应用;黄刚,教授,研究方向为计算机软件理论及应用。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20170110.1019.046.html>

分摊到多个节点分别进行处理,以减少用户等待和用户响应时间,并将单个负载较重节点的运算分摊到多个节点进行并行处理,最大化地发挥集群的运算和服务能力。

## 1 概述

分布式系统是若干独立计算机的集合,主要目的是使用户能够方便地访问、获取远程资源,这些资源包括计算机、存储设备、数据、文件、Web 网页以及网络。因为在提供这些资源的系统中的服务节点存在巨大差异,所以,讨论分布式环境下的负载均衡问题首先要考虑系统中各个计算机之间的差别,以及计算机之间通信方式的差别。

进一步讲,环境的动态网络拓扑也并非固定。这项工作的重点是设计一种考虑分布式环境的动态、可扩展性和异构性的负载均衡算法。

研究人员对于分布式计算系统负载均衡问题进行了深入研究,对于该问题通常有两种方法:

(1)静态负载均衡方法。它取决于静态信息,例如 CPU 能力、内存性能等来做出负载平衡决策。

(2)动态负载均衡策略<sup>[2-3]</sup>。该方法尝试获取系统的当前状态,并以此为依据做出决策,因此,能够进一步提高系统性能。

传统的负载均衡算法,比如随机法、轮转调度算法<sup>[4]</sup>(Round-Robin scheduling, RR)、最小连接数法(Least-Connection scheduling, LC)等,也有一些基于这些传统算法改进而来的算法,这些算法在海量的请求压力和复杂的异构网络环境下显得力不从心。

DLB 算法被提出<sup>[5-12]</sup>以监视资源利用率,即, CPU、内存磁盘、I/O 等,然后计算它们的加权作为服务器的负载。有研究者在此基础上引入了反馈,动态调整每个资源的权重;提出了模糊控制理论将服务器负载分成不同的层次。但是这些算法仍没有解决一个问题:在监控的时间间隔之间,节点性能发生改变,而此时服务器管理节点并没有及时获知,导致请求被优先分配到管理节点认为性能较优越的服务器节点上,因此产生负载不均的情况。而如果片面缩短 DLB 算法中的时间间隔,将会大大增加系统开销。

传统的分布式系统模型中的每个节点都可以作为一个发送器或接收器。一个节点作为发送方,如果当时的工作量级大于它的容量。然后,尝试与邻国销售当前的负载平衡其负载。为了使系统的工作原理有效,每个节点需要具备网络中所有其他节点的知识。这是该系统的最大缺点。发送器/接收器通过发送广播消息到所有其他节点,并期望一个响应于发现的节点。创建时会涉及所有节点一定的通信开销。

为了解决以上问题,降低负载平衡过程的复杂性,并兼顾异质性和可扩展性,文中尝试使用两级策略,以平衡节点之间的工作量负载。异构系统被划分为若干个簇,而每个簇由若干个节点组成。

负载均衡的一个重要因素是在负载均衡算法中使用的负载指标。在 DLB 算法的基础上,提出的算法使用节点的处理器和内存利用率、节点中排队的作业数量以及节点的处理能力三元组负荷指标。通过所使用的负载量度计算几个现有负载尺度,提供关于一个节点的负载的更多信息。

整个负载均衡分为两个过程。首先,系统负载会以各个簇的处理能力为标准,将作业均衡分配至各个簇,簇依据各个节点的处理能力将作业均衡分配至节点。当系统开始处理工作以后,各个节点依据当前负载计算得出的负载率,将其传递至簇管理节点,并为自己设定负载率阈值。如果超出阈值,则重新计算负载率向管理节点反馈。此时管理节点为此节点更新权值,继续分配任务。如果节点超载,则在向管理节点更新权值的同时尝试将作业分给簇内负载较轻的节点,如果不成功则尝试向其他簇寻求负载较轻的节点。

仿真结果表明,提出的方法明显地减少了作业的平均响应时间。

## 2 系统模型与问题定义

考虑  $N$  个异质节点  $P_1, P_2, \dots, P_N$  通过通信网络连接。每个节点具有一些计算设备和本地存储器。对于分布式系统,假定作业在任何节点都按泊松率  $\lambda_i$  到达。而作业的服务时间服从指数分布,即每个作业的完成时间之间没有依赖关系。作业被假定为独立的,可以在任何节点上执行。作业的连接时间长短各异,任务大小不同,在这里认为每一种类型的作业在所有作业中的分布是均匀的。因此,每个节点作为一个  $M/M/1$  队列。节点接收作业请求并将其存入缓冲区,按照先来先服务的策略完成所有作业。该模型中的网络被组织成多个小型集群。每个小型集群都有指定为群集管理器的节点,它负责收集集群内普通节点的负载信息,然后根据信息为子节点计算权值,并按照权值为其分配作业。

当某一集群中的节点出现负载不均衡时,该节点向其管理节点(CM)反馈。CM 接收到反馈后,尝试在本集群中进行负载均衡。而当 CM 发现本集群中所有节点均产生过载时,尝试向主节点反馈,并由主节点进行全局负载均衡。全局平衡是通过平衡相邻集群之间的负载来实现。这种结构分散了负载平衡过程,尽量减少通信开销,所以它可以扩展到大系统。

分层负载均衡系统模型如图 1 所示。

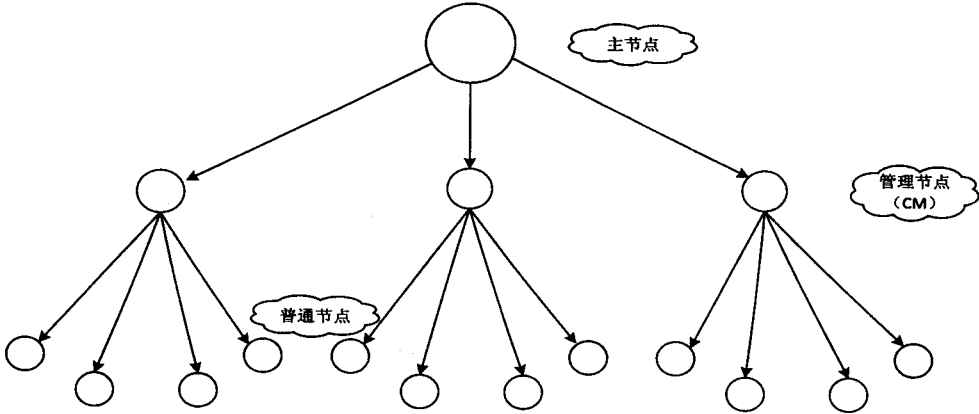


图1 分层负载均衡系统模型图

每个节点负责:

- (1)保持自身的工作负荷信息。
- (2)发送该工作负载信息到CM。
- (3)执行负载均衡通过其集群管理器的决定。

除了节点基本功能,CM 还负责:

- (1)接收并保存其他提交的工作节点负荷信息。
- (2)决定何时启动本地负载均衡。
- (3)向每个工作节点发送负载均衡决定。
- (4)决定发起全局负载均衡。

为了进行负载平衡,每个管理节点负责维护子节点信息表。参数如表1所示。

表1 参数表

参数	意义
$N$	虚拟机节点个数
CM	集群管理节点
$P_i$	第 $i$ 个节点
$T_e$	时间间隔
$N_{idle}$	节点剩余性能
$W_i$	节点权值
$R_{idle}$	内存剩余利用率
$CPU_{idle}$	CPU 剩余利用率
$N_a$	连接数

3 集群负载均衡策略(CLB)

3.1 CLB 策略概述

集群中所有节点获取自身性能参数信息,并把信息发送给CM,CM统计本组节点权值信息,计算出本组集群节点剩余权值信息,并将信息填入节点信息表。然后,CM按照节点信息表,将作业分配至各个节点。同时,CM计算本组节点负载均值,并提升10%作为超载阈值。如果有节点负载超过此阈值,则视为此节点过载,并启动局部负载均衡策略,将此节点作业转发至其他非过载节点处理。设定负载率80%为全局阈值,如果本组所有节点负载均超过80%,则认为本组集群

过载,启动全局负载均衡策略,将本组集群中各节点的作业转发至其他组集群,同时,主节点减少向本集群转发作业。

负载均衡过程开始从下级接收负载更新消息。在 $T_e$ 每个周期的时间间隔称为估计间隔,每个处理器 $P_i$ 系统计算其负载信息的参数,然后将信息发送到集群管理器。

负载均衡策略包括以下阶段:

- (1)在每个 $T_e$ ,每个节点获取自身信息,并发送负荷指标到集群管理器CM。
- (2)CM从本集群中的各个节点处接收信息,计算并产生转发表。同时,判断本集群是否处于饱和状态。
- (3)如果集群已经饱和,则CM启动全局负载均衡。否则检查稳定性标准,以确定是否需要本地负载均衡。
- (4)CM启动本地负载均衡。由CM协调集群内节点,把作业从超载节点转移到欠载节点。重复这个过程,直到该集群达到平衡。

3.2 CLB 集群负载均衡过程

3.2.1 工作节点性能参数获取

节点的负载情况在算法中起到至关重要的作用<sup>[13-14]</sup>。文中采集各节点CPU性能、内存性能、CPU利用率、内存利用率等多项指标。其中,CPU性能、内存性能指标为常量。

- (1)CPU处理能力:该算法中,通过调用系统接口获取服务器的CPU参数,包括核心数量和主频等,并为其设定权值。用 $C_i$ 表示,范围为0~1。
- (2)内存参数:同样,获取计算机的物理内存和虚拟内存的大小,以便设置内存权值。用 $M_i$ 表示,范围为0~1。
- (3)节点的可用性由节点CPU剩余和内存剩余决定,如果节点的这两个指标参数值较高,则反映此节点是空闲节点,反之,则是繁忙节点。这里由CPU利用率( $CPU_{rate}$ )和内存利用率( $R_{rate}$ )计算节点的性能剩

余指标  $W_{(i)(idle)}$ 。

(4) 对任何服务节点  $P_i$  来讲, 假设所有作业到达的时间间隔服从泊松分布, 则节点总连接数服从强度为  $\lambda$  的泊松过程, 即单位时间内节点的连接总数为  $\lambda$ , 也即认为每个作业到达时间间隔为  $1/\lambda$ 。节点响应时间是从节点接收请求到返回服务信息所花费的总时间, 平均节点响应时间在衡量节点负载时更有意义。类似的, 认为在单位时间内, 节点响应的请求数  $\mu$  为节点响应时间的倒数, 即节点响应时间为  $1/\mu$ 。所有工作被认为是独立的。

(5) 需要采集系统连接数  $N_i$  和系统响应时间  $T_i$  来评价该算法与其他算法孰优孰劣。系统连接数是在时刻  $T$  时, 节点所建立的连接数, 反映节点当前的负载情况。系统响应时间是从服务器接收到请求至响应请求所花费时间的总量, 反映服务器在处理作业时的效率。

### 3.2.2 工作负载评估

在评估节点工作负载时, 需要建立权值衡量函数模型。对于在集群中建立节点服务器工作负载的数学模型, Watts 和 Taylor<sup>[3]</sup> 通过研究证明: 使用线性加权法可以定量描述服务器负载量的有效性。故文中采用该方法建立改进算法的数学模型。

线性加权法<sup>[15]</sup>的理论思路是: 各个度量指标在总目标数值中所占的重要程度是不同的, 那么就可以根据其各自的重要性分别为它们设定系数, 并将这些带有系数的度量指标值相加, 最终得到总目标的值。这样, 在多个指标共同作用且作用力不同的情况下, 得到的目标值能够很好地反映实际情况。

因此, 可以得出改进的基于多衡量指标的负载均衡衡量函数:

$$u(f) = \sum_{i=1}^n \omega_i f_i \quad (1)$$

其中,  $\omega$  为权值系数;  $f$  为权值。

第  $i$  台服务器的剩余权值为:

$$W_{(i)(idle)} = \omega_{(i)(C)} * C_i * CPU_{idle} + \omega_{(i)(R)} * M_i * R_{idle} \quad (2)$$

其中,  $CPU_{idle}$  是 CPU 剩余利用率,  $CPU_{idle} = 1 - CPU_{rate}$ ;  $R_{idle}$  是内存剩余利用率,  $R_{idle} = 1 - R_{rate}$ 。

### 3.2.3 集群负载均衡

在这个过程中, CM 节点获取集群内节点的负载信息, 并根据节点性能剩余利用率来判断本集群负载情况。在集群中, 有的节点出现过载, 有点则会出现欠载的情况, 具体可以分为高负载、中等负载和欠载。需要将作业从高负载节点转发至欠载节点进行处理。

### 3.2.4 全局负载均衡

在某一集群中会出现负载不均衡的情况, 同样的

情况在不同的集群间依然不可避免。依照负载情况, 集群可以分为饱和集群(超载集群)、非饱和集群(平均负载集群和欠载集群)。当一个集群出现饱和的情况时, 它尝试向主节点报告状态, 请求减轻负载。此时, 主节点启动策略, 询问各集群管理节点集群是否饱和, 并按照非饱和集群的剩余利用率权值, 将作业转发至这些非饱和集群。

## 4 实验结果

为了验证该算法的可行性, 搭建了基于 Windows Server 2008 的虚拟机集群, 并编写程序进行模拟实验。

实验平台环境由 16 台物理主机(CPU 四核八线程, 主频 2.4 GHz, 内存 32 GB)、2 台千兆路由器以及若干台客户机组成。利用这些硬件设备搭建虚拟机集群, 为了使环境更贴近实际, 为各个虚拟机分配不同的资源, 具体见表 2。

表 2 虚拟机配置表

类型	虚拟服务器配置类型	节点数量
1	内存 2 GB, CPU 单核 2.4 GHz	12
2	内存 4 GB, CPU 单核 2.4 GHz	20
3	内存 8 GB, CPU 单核 2.4 GHz	8
4	内存 16 GB, CPU 双核 2.4 GHz	4
5	内存 32 GB, CPU 四核 2.4 GHz	2

在集群中, 把性能较优秀的虚拟机设置为各个簇的管理节点, 其他节点设为普通节点。在模拟程序中, 采用 Visual Studio 开发环境, C++ 语言编写代码, 并部署至环境中。

负载均衡前后各类型节点访问量分别如图 2 和图 3 所示。

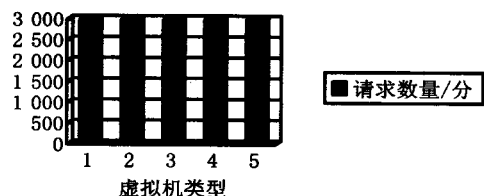


图 2 负载均衡前各类型节点访问量

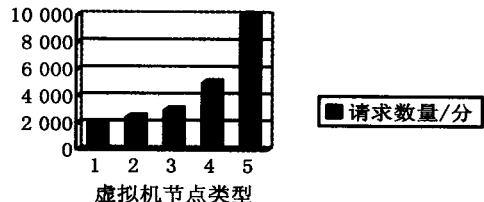


图 3 负载均衡后各类型节点访问量

## 5 结束语

在考虑异构环境的情况下, 提出了分布式系统两层结构节点自适应虚拟机集群系统负载均衡算法。算

法主要通过节点权值和负载率来评估节点性能,并进行资源的合理配置。而当请求间差异过大导致资源分配不均时,该算法能在代价较小的情况下解决问题。

在下一步的工作中,将考虑引入软件定义网络(SDN),尝试将集群网络负载均衡算法在新的网络环境中进行验证。另外,算法在提高整体响应速度的同时并没有侧重点,对于单个请求来说,算法给出的响应时间并非最优解,将在这方面做进一步研究。

#### 参考文献:

- [1] Yang G. Dynamic load-balancing in iSCSI systems based on a feedback control mechanism [C]//Proceedings of the 2011 fifth international conference on management of e-commerce and e-government. [s. l.]: IEEE Computer Society, 2011: 187-192.
  - [2] Saxena S, Khan M Z, Singh R. Performance analysis in distributed system of dynamic load balancing using fuzzy logic [C]//2012 spring congress on engineering and technology. [s. l.]: IEEE, 2012: 1-5.
  - [3] Watts J, Taylor S. A practical approach to dynamic load balancing [J]. IEEE Transactions on Parallel & Distributed Systems, 1998, 9(3): 235-248.
  - [4] Chatterjee M, Setua S K. A new clustered load balancing approach for distributed systems [C]//Third international conference on computer, communication, control and information technology. [s. l.]: IEEE, 2015: 1-7.
  - [5] Zhang L, Li X P, Yuan S. A content-based dynamic load-balancing algorithm for heterogeneous web server cluster [J]. Computer Science & Information Systems, 2010, 7(1): 153-162.
  - [6] 李 坤. 基于动态反馈机制的服务器负载均衡算法研究 [J]. 电子科技, 2015, 28(9): 45-49.
  - [7] Kaur S, Singh J, Kumar K, et al. Round-robin based load balancing in software defined networking [C]//2nd international conference on computing for sustainable global development. [s. l.]: [s. n.], 2015.
  - [8] Jain D, Jain S C. Load balancing real-time periodic task scheduling algorithm for multiprocessor environment [C]//International conference on circuit, power and computing technologies. [s. l.]: IEEE, 2015.
  - [9] 袁 刚. 基于服务分类的预测型动态负载均衡调度算法 [J]. 计算机技术与发展, 2015, 25(6): 96-100.
  - [10] 杨 洋, 杨家海, 王 会, 等. IP 网络时延敏感型业务流自适应负载均衡算法 [J]. 通信学报, 2015, 36(3): 131-141.
  - [11] 童瑞霞. 基于动态反馈机制的集群负载均衡算法研究 [D]. 武汉: 武汉理工大学, 2011.
  - [12] 冯秀玲. 云计算环境下的负载均衡算法的研究与设计 [D]. 北京: 北京邮电大学, 2012.
  - [13] 叶春森. 企业云计算服务的商业价值创造研究 [D]. 合肥: 合肥工业大学, 2014.
  - [14] 马华伟, 吴晓莉. 云计算环境下虚拟机负载均衡问题研究 [J]. 微电子学与计算机, 2014, 31(4): 10-12.
  - [15] 康承昆, 刘晓洁. 一种基于多衡量指标的 HDFS 负载均衡算法 [J]. 四川大学学报: 自然科学版, 2014, 51(6): 1163-1169.
- 
- (上接第 50 页)
- 用, 2015, 35(3): 668-674.
  - [5] Kennedy J. Particle swarm optimization [M]//Encyclopedia of machine learning. Boston, MA: Springer, 2010: 760-766.
  - [6] Liang J J, Qin A K, Suganthan P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions [J]. IEEE Transactions on Evolutionary Computation, 2006, 10(3): 281-295.
  - [7] Lynn N, Suganthan P N. Comprehensive learning particle swarm optimizer with guidance vector selection [C]//Proceedings of the 2013 IEEE symposium on swarm intelligence. Singapore: IEEE, 2013: 80-84.
  - [8] 许 骏, 许晓东. 一种群体智能融合算法及其在应急设施选址的应用 [J]. 计算机工程与科学, 2014, 36(4): 667-673.
  - [9] Dewerra D, Hertz A. Tabu search technique—a tutorial and an application to neural networks [J]. Operations Research, 1989, 11(3): 131-141.
  - [10] Ostermeier H. Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation [C]//Proceedings of the 1996 IEEE international conference on evolutionary computation. Nagoya: IEEE, 1996: 312-317.
  - [11] Wang Y, Li B, Weise T, et al. Self-adaptive learning based particle swarm optimization [J]. Information Sciences, 2011, 181(20): 4515-4538.
  - [12] 潘全科, 朱剑英. 一类解决 Job Shop 问题的禁忌搜索算法 [J]. 中国机械工程, 2006, 17(5): 536-539.
  - [13] Liang J J, Qin A K, Suganthan P N, et al. Evaluation of comprehensive learning particle swarm optimizer [J]. Lecture Notes in Computer Science, 2004, 3316: 230-235.
  - [14] 蔡昭权, 黄 翰. 自适应变异综合学习粒子群优化算法 [J]. 计算机工程, 2009, 35(7): 170-171.
  - [15] 刘关俊. 基于粒子群算法的移动机器人路径规划研究 [D]. 长沙: 中南大学, 2007.
  - [16] Liang J J, Suganthan P N. Adaptive comprehensive learning particle swarm optimizer with history learning [C]//Simulated evolution and learning. Berlin: Springer-Verlag, 2006: 213-220.