基于排队论综合指标评估的动态负载均衡算法

王文博，叶庆卫，周宇，陆志华
（宁波大学信息科学与工程学院，浙江 宁波 315211）

摘　要：互联网通信、计算机集群和云环境均具有一定的复杂性和动态性，极易发生负载失衡，从而降低服务效率、增加能耗。因此，负载均衡技术成为重点研究课题。现有的负载均衡策略均是以 CPU、内存、进程等参数的占用率来评估服务器当前的负载情况，但服务器负载情况的复杂性往往使其难以得到准确评估。针对该问题，提出了一种基于排队论综合指标评估的动态负载均衡算法，首先引入排队论模型评估各服务器的实时负载情况，然后根据各服务器的负载综合指标，将输入队列中的任务逐一分配给各服务器。实验结果表明，该方法可有效平衡各服务器的负载且减少任务请求的平均等待时间。

关键词：负载均衡；排队论；性能
中图分类号：TP393
文献标识码：A
doi: 10.11959/j.issn.1000−0801.2018204

Dynamic load balancing algorithm based on queuing theory comprehensive index evaluation

WANG Wenbo, YE Qingwei, ZHOU Yu, LU Zhihua
School of Information Science and Engineering, Ningbo University, Ningbo 315211, China

Abstract: Internet communication, computer cluster and cloud environment have complex and dynamic characteristics, which can cause load imbalance easily, reduce the service efficiency and increase the energy consumption. Therefore, the load balancing technology becomes the focus of research. The existing load balancing strategy uses the occupancy of CPU, memories, processes to estimate the current load of each server. But it is hard to guarantee its accuracy. Aiming at this problem, a dynamic load balancing algorithm based on queuing theory comprehensive index evaluation was proposed. Firstly, queuing theory model was introduced to estimate the real-time load of each server, and then the tasks of input queue was distributed to each server separately according to the load comprehensive index of each server. Experimental results show that this method can balance the load of each server effectively and reduce the average waiting time of the task requests, which is of great application value.

Key words: load balancing, queuing theory, performance

收稿日期：2017−10−30；**修回日期**：2018−06−26
基金项目：国家自然科学基金资助项目（No.51675286，No.61071198）；浙江省重点科技创新团队资助项目（No.2013TD21）
Foundation Items: The National Natural Science Foundation of China (No.51675286, No.61071198), The Key Scientific and Technological Innovation Group of Zhejiang Province (No.2013TD21)

1　引言

近年来，大规模集群计算、云计算技术飞速发展。在实际应用中，经常出现负载失衡的情况[1]，使得系统利用率低，用户请求得不到快速响应，服务质量急剧下降，因此负载均衡技术也变得日益重要[2]。目前国内外已经对此进行了大量的研究工作，先后提出了各种负载均衡算法[3]。其中，有经典算法，如阿里云和亚马逊云等提供的负载均衡服务使用轮询算法、最小连接数算法等，这类算法思想简单、容易实现[4]、具有较好的实用性，但无法实时准确判断服务器的负载状况；有启发式任务调度算法，如遗传算法[5]、蚁群算法[6]、粒子群算法[7]等，这类算法与经典算法相比，负载均衡效果更好，但存在实现更复杂、速度较慢、易陷入局部最优解等缺点；排队论在负载均衡领域中也有所应用，参考文献[8]提出一种结合周期反馈原理的动态负载均衡策略，利用排队论建模，获得系统性能的计算公式。参考文献[9]提出一种在黄金时间将工作转移到空闲节点来平衡负载的策略，并通过排队论建模分析性能指标。这些基于排队论的负载均衡算法，都是通过排队论对系统性能进行分析与评价，而负载评估的方法依然与其他算法相似，并没有真正地将排队论应用于指导任务的分配。

本文从构建排队综合指标出发，提出了一种基于排队论综合指标评估的动态负载均衡算法，克服了以往算法中使用 CPU、内存占用率等参数难以与负载评估直接挂钩的难题，降低了服务器负载信息收集的复杂性，使排队论模型真正起到均衡负载的作用。首先定时统计各服务器的服务率和任务到达率，其次通过排队论建模计算各服务器的任务平均排队长度和任务平均等待时间，然后根据求得的综合指标来评估各服务器的实时负载情况，最后将下一任务分配给负载综合评估指标最小的服务器。

2　基于排队论的负载均衡调度模型

排队论是一套非常成熟的经典理论，广泛应用于人们的生产、生活等各个领域中[10]。在互联网络通信中，服务器处理客户机请求的机制本身就是一个排队理论的现实情况[11]，用户以任务的形式作为要求服务的一方，向服务系统发起请求，未能及时处理的任务在服务器上形成排队队列，服务器端如何合理分配、处理这些任务，直接影响整个系统的性能[12]。

2.1　模型建立

负载均衡调度系统模型如图 1 所示，整个系统是一个多服务器多队列排队系统，每个服务器都有一个自己的任务队列，因此单个服务器节点也是一个独立的排队系统，系统对待每一个任务请求都是平等的，从而每个服务器都可抽象为一个单服务器单队列排队模型，即 M/M/1 排队模型[13]。通过这种经典排队模型求出的平均排队长 L_{q_i} 和平均等待时

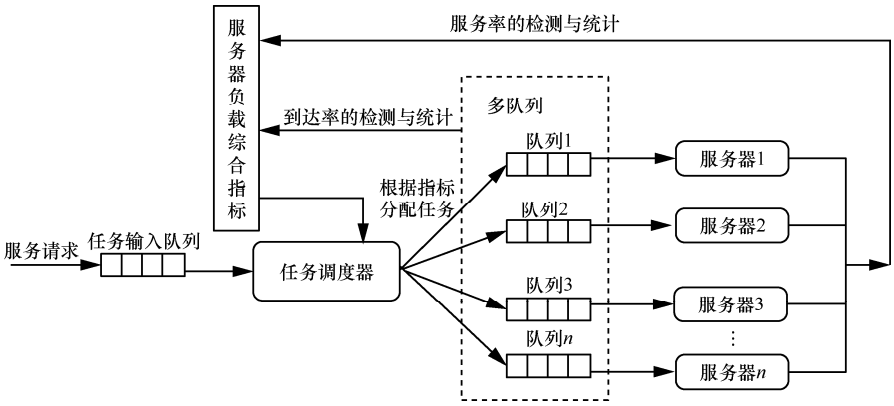

图 1　基于排队论的负载均衡任务调度模型

间 W_{q_i} 是两个重要的系统评估指标[14]，下面单看第 i 个服务器节点的 M/M/1 排队模型。

首先，在每个服务器内建立一个时间表，记录下每个任务的到达时间 arrive_time 、服务开始时间 start_time 、服务结束时间 end_time 。假设在时间 T 内有 n 个任务请求到达服务器，任务请求到达的时间间隔是一个随机变量，各个任务请求之间相互独立，因此任务请求服从泊松分布，则用户请求的平均到达率为：

$$\lambda_i = \frac{n}{T} \tag{1}$$

其次，若在时间 T 内服务器将 a~$a+k$ 个任务处理完毕，则该服务器的平均服务率为：

$$\mu_i = \frac{k+1}{\sum_{i=a}^{i=a+k}(\text{end_time}_i - \text{start_time}_i)} \tag{2}$$

根据排队理论的 Little（里特）公式，当 $\frac{\lambda_i}{\mu_i} < 1$ 时，系统是稳定的，并且会有一个平滑的分布[15]。排队论的生灭过程如图 2 所示。

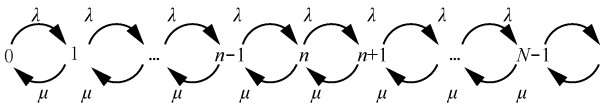

图 2　生灭状态转移

由图 2 可得，系统处于稳态时的概率方程如下：

$$\lambda P_{n-1} + \mu P_{n+1} = \lambda P_n + \mu P_n \tag{4}$$

$$\mu P_n = \lambda P_{n-1} \tag{5}$$

设 $\frac{\lambda_i}{\mu_i} = \rho_i$，由 $\sum_{k=0}^{\infty} P_k = 1$，递推可得：

$$P_0 = \frac{1}{1+\rho_i+(\rho_i)^2+...+(\rho_i)^n+...}(n=1,2,3...) \tag{6}$$

$$P_0 = 1-\rho_i \tag{7}$$

$$P_n = \rho_i^n(1-\rho_i) \tag{8}$$

其中，ρ_i 称为服务强度，是平均到达率与平均服务率的比值，即在同一段时间内平均到达的任务数与平均被服务的任务数之比。它刻画出各个服务器的繁忙程度。假定任务以简单流到达，系统容量为无限，则可求得以下几个参数来反映当前服务器的特征。

第 i 个服务器的任务平均排队长度为：

$$L_{q_i} = \sum_{n=1}^{\infty}(n-1)P_n = \sum_{n=1}^{\infty} nP_n - \sum_{n=1}^{\infty} P_n$$

$$= \frac{\rho_i^2}{1-\rho_i} = \frac{\rho_i\lambda}{\mu_i-\lambda_i}(0<\rho_i<1) \tag{9}$$

任务在服务器中的逗留时间分布函数为：

$$F(x) = 1-\mathrm{e}^{-(\mu_i-\lambda_i)x} \tag{10}$$

任务在服务器中的逗留时间 X 服从参数为 $\mu-\lambda$ 的负指数分布，因此 X 的期望值即任务平均逗留时间为：

$$W_{s_i} = \frac{1}{\mu_i-\lambda_i} \tag{11}$$

任务在系统中等待时间的期望值，等于任务在系统中逗留时间的期望值减去服务时间的期望值,因此任务平均等待时间为：

$$W_{q_i} = W_{s_i} - \frac{1}{\mu_i} = \frac{\rho_i}{u_i-\lambda_i} \tag{12}$$

2.2　调度策略与队列负载评价

服务器的平均排队长度 L_{q_i} 越短，说明该服务器中排队任务越少，但因任务的大小是不同的，只根据平均排队长度不能够完全反映服务器当前的负载情况。而平均等待时间 W_{q_i} 越短，说明该服务器处理任务的效率越高，因此在负载均衡调度系统中，使用平均排队长度和平均等待时间的加权平均值 H 作为综合评估指标来反映当前服务器的负载情况，并指导任务调度器对任务的分配。H 值越大代表当前服务器的负载越重，H 值越小代表当前服务器的负载越轻，因此任务调度器根据各服务器的 H 值，将输入队列中排在队首的任务分配到当前 H 值最小的服务器队列中，以平衡各服务器的负载。式（13）为 H 值的计算式：

$$H_i = \alpha \cdot L_{q_i} + \beta \cdot W_{q_i} \tag{13}$$

$$\alpha + \beta = 1 \tag{14}$$

那么负载综合指标的平均值 $\overline{H}$ 为：

$$\overline{H} = \sum_{i=1}^{n} H_i / n \tag{15}$$

由此可以计算出各服务器负载综合评估指标的方差值 F，方差越小，表明各服务器当前的负载均衡度越好，为：

$$F = \sum_{i=1}^{n} (H_i - \overline{H})^2 / n \tag{16}$$

2.3　算法流程

算法的总体流程如图 3 所示，主要包括两部分：一是前端任务调度器，主要任务是获取各后端服务器的负载信息，计算综合评估指标，并将任务分配给各个服务器；二是后端服务器，主要是执行队列中的任务，并收集相关信息传递给任务调度器。

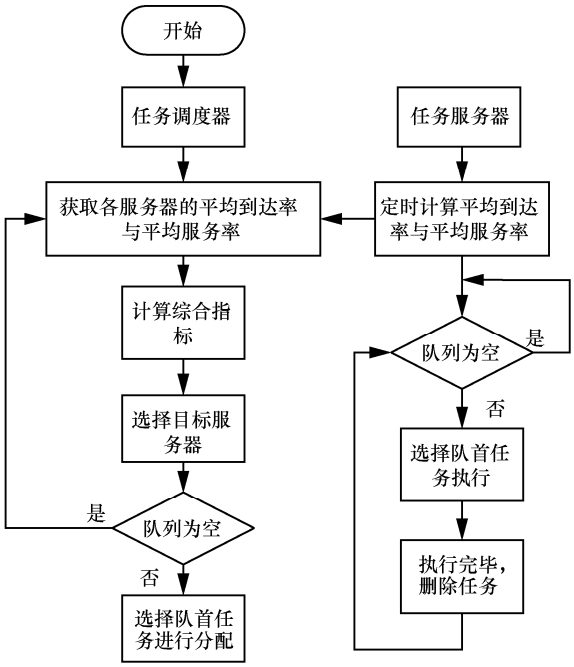

图 3　基于排队论综合指标评估的动态负载均衡算法流程

3　仿真实验与性能分析

因 MATLAB 编程简单，仿真算法容易实现，所以本文使用 MATLAB 编写仿真任务调度器。由于多个服务器需要同时运行，因此使用 C 语言编写多个可执行 exe 程序来仿真服务器。系统由 1 个任务调度器和 4 个任务服务器组成，且 4 个服务器能力相同。对轮询算法、最小连接数法和基于排队理论的动态负载均衡算法进行仿真对比和分析，经过多次实验，结果如图 4、图 5 所示。图 4 中，通过求取 3 种算法综合指标的方差来评估服务器的负载均衡度，由图 3 可知本文算法的负载均衡度明显优于最小连接数算法和轮询算法。

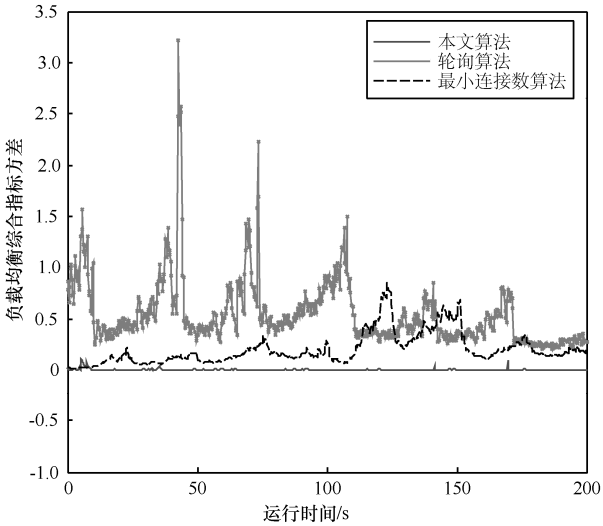

图 4　各个服务器负载均衡综合指标的方差

前文已经提到在实际应用中，用户希望得到快速的响应，因此任务的平均等待时间也是衡量一个负载均衡算法优劣的指标之一。图 5 仿真了本文算法、最小连接数算法和轮询算法的任务平均等待时间。当有新任务到达服务器时，服务器的任务到达率会相应增加，任务的平均等待时间也会随之增加，而没有新任务到达时，任务的平均等待时间会逐渐下降，因此任务的平均等待时间大约呈周期性变化。如图 5（a）所示，本文算法中各个服务器的任务平均等待时间为 0.6~3 s，4 条曲线分布较为集中，数值较小且相差不大；而图 5（b）中最小连接数算法的任务平均等待时间为 2~6 s，比本文算法更长，且 4 条任务平均

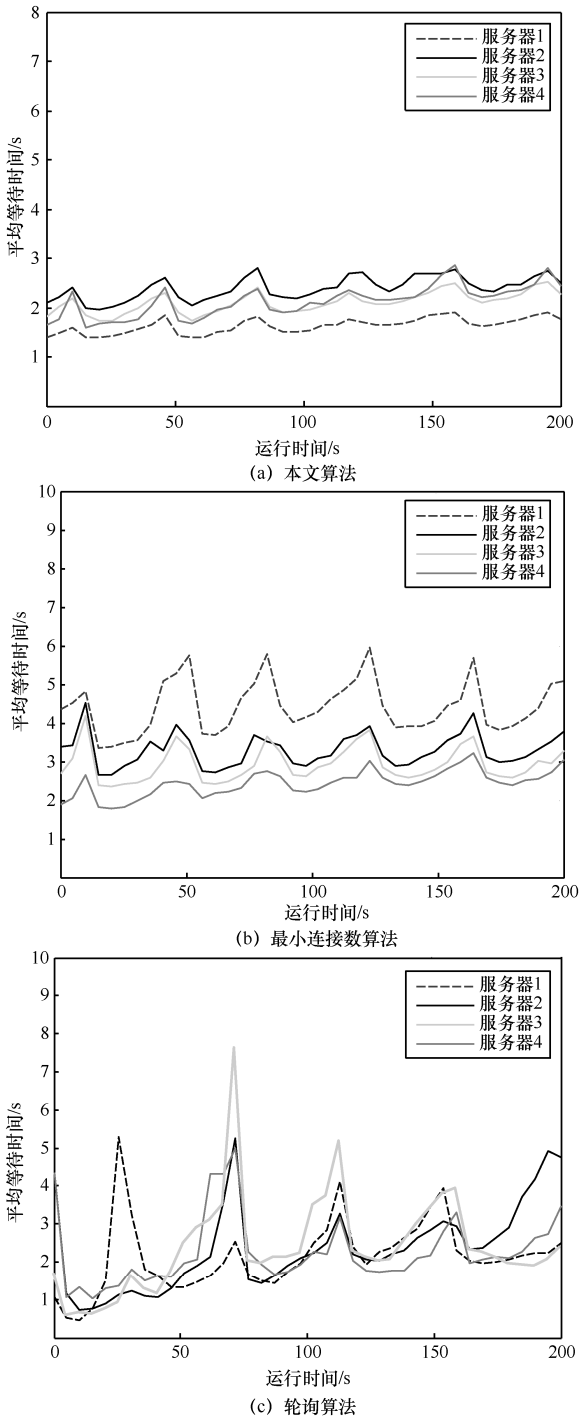

图 5　各个服务器的任务平均等待时间

等待时间曲线与本文算法相比更为分散，这可能会导致部分任务请求长时间得不到响应；图 5（c）中轮询算法的任务平均等待时间集中为 1.4~8 s，且上下浮动较大，在大部分运行时间中都比本文算法的平均等待时间长，因此本文算法效果更好。

由图 5（c）中可知，轮询算法运行到大约 70 s 时，服务器的任务平均等待时间出现很大的值，说明此时一些任务请求需要等待较长的时间才能响应，可能已有服务器出现过载情况。虽然此时 4 个服务器综合指标的方差没有相差特别大，但由于处理不同的任务需要的时间长短是不同的，因此服务器可能出现排队长度较短而任务平均等待时间较大的情况或排队长度较长而任务平均等待时间较小的情况，这也进一步印证了本文中选用平均排队长度和平均等待时间的加权平均值作为综合评估指标的合理性。

4　结束语

现有的排队论负载均衡算法，都只利用排队论建模求得的指标来评价系统性能，任务的调度策略仍然以其他算法为主。本文将排队论模型应用在计算机集群负载均衡系统中，提出了一种基于排队论综合指标评估的动态负载均衡算法。实验结果表明，该算法与轮询算法和最小连接数算法相比，有更好的负载均衡度和更短的任务平均等待时间。

本文算法具有一定的延后性，主要是指在任务平均服务率的统计部分花费较多时间，但只是毫秒级的延后，在逻辑上不存在延后性，对实验结果不会产生很大影响。本文算法仍具有一定的局限性：第一，本文算法通过仿真实验进行评测，而真实环境中会出现更多复杂不可预知的因素，因此今后将在真实集群环境中对算法的实用性和通用性进行验证测试。第二，本文算法只与轮询算法和最小连接数算法这类经典算法，在负载均衡度、平均等待时间方面进行了对比，今后将在算法速度、复杂度等方面与该领域的最新算法进行多维度、多指标的综合对比与改进。

参考文献：

[1]　李永平，邹华. 应用服务器的负载平衡技术和实现方案[J].

电信科学, 2013, 29(10): 15-18.
LI Y P, ZOU H. Load balancing technology and implementation scheme of application servers[J]. Telecommunications Science, 2013, 29(10): 15-18.

[2] DHINESH B L D, KRISHNA P V. Honey bee behavior inspired load balancing of tasks in cloud computing environments[J]. Applied Soft Computing, 2013, 13(5): 2292-2303.

[3] 孙乔, 邓仆侨, 王志强, 等. 一种基于分布式服务器集群的可扩展负载均衡策略技[J]. 电信科学, 2017, 33(9): 190-196.
SUN Q, DENG P Q, WANG Z Q, et al. A scalable load balancing strategy based on distributed server cluster[J]. Telecommunications Science, 2017, 33(9): 190-196.

[4] THAKUR A, GORAYA M S. A taxonomic survey on load balancing in cloud[J]. Journal of Network and Computer Applications, 2017(98): 43-57.

[5] 包晓安, 魏雪, 陈磊, 等. 基于 mean-variance 的服务集群负载均衡方法[J]. 电信科学, 2017, 33(1): 1-8.
BAO X A, WEI X, CHEN L, et al. Load balancing for server cluster based on mean-variance[J]. Telecommunications Science, 2017, 33(1): 1-8.

[6] 郑相全, 郭伟, 葛利嘉, 等. 一种基于跨层设计和蚁群优化的自组网负载均衡路由协议[J]. 电子学报, 2006(7): 1199-1208.
ZHENG X Q, GUO W, GE L J, et al. A load balancing routing protocol for ad hoc networks based on cross layer design and ant colony optimization[J]. Chinese Journal of Electronics, 2006(7): 1199-1208.

[7] MILANI A S, NAVIMIPOUR N J. Load balancing mechanisms and techniques in the cloud environments: systematic literature review and future trends[J]. Journal of Network and Computer Applications, 2016(71): 86-98.

[8] 陈超, 赵跃龙, 王文丰, 等. 基于反馈的改进动态负载均衡策略[J]. 计算机工程,2010, 36(14): 34-36, 39.
CHEN C, ZHAO Y L,WANG W F, et al. An improved dynamic load balancing strategy based on feedback[J]. Computer Engineering, 2010, 36(14): 34-36, 39.

[9] YIN F, JIANG C J, DENG R, et al. Grid resource management policies for load-balancing and energy-saving by vacation queuing theory[J]. Computers & Electrical Engineering, 2009, 35(6): 966-979.

[10] KUMAR M, SHARMA S C. Dynamic load balancing algorithm for balancing the workload among virtual machine in cloud computing[J]. Procedia Computer Science, 2017(115): 322-329.

[11] ZHANG Z J, FAN W G. Web server load balancing: a queueing analysis[J]. European Journal of Operational Research, 2008, 186(2): 681-693.

[12] CIARDO G, RISKA A, SMIRINI E. Equiload: a load balancing policy for clustered Web servers[J]. Performance Evaluation, 2001(46): 670-684.

[13] KO Y M, CHO Y. A distributed speed scaling and load balancing algorithm for energy efficient data centers[J]. Performance Evaluation, 2014(79): 120-133.

[14] ROSS S. Queueing theory[J]. Introduction to Probability Models, 2014: 481-558.

[15] 于国防, 王耀才, 庄立运, 等. 基于分配器队列模糊控制的集群负载平衡[J]. 计算机工程, 2008(6): 129-130, 136.
YU G F, WANG Y C, ZHUANG L Y, et al. Cluster load balancing based on fuzzy control of distributor queue[J]. Computer Engineering, 2008(6): 129-130, 136.

[作者简介]

王文博（1992−），女，宁波大学信息科学与工程学院硕士生，主要研究方向为服务器负载均衡技术等。

叶庆卫（1970−），男，博士，宁波大学信息科学与工程学院教授、硕士生导师，主要研究方向为信号检测、最优化搜索、视频识别与跟踪等。

周宇（1960−），男，宁波大学信息与工程学院教授、硕士生导师，主要研究方向为信号处理、网络与信息安全、物联网技术等。

陆志华（1983−），男，博士，宁波大学信息科学与工程学院讲师，主要研究方向为信号处理、多运动目标的实时跟踪、统计信号处理算法和应用等。