

基于 VIKOR 和用户 QoS 优先级的 云计算自适应算法^①

程 曦¹, 宋铁成²

1. 四川文理学院, 康养产业学院, 四川 达州 635000; 2. 重庆邮电大学 通信与信息工程学院, 重庆 400065

摘要: 针对传统调度算法不能提供一种在考虑负载均衡的同时满足用户服务质量(Quality of Service, QoS)的自适应方法进行了研究, 提出一种基于改进多准则妥协解排序法(VIKOR)和用户 QoS 优先级的云计算自适应调度算法. 该算法首先使用最优-最差多准则决策为每个评估标准或用户 QoS 优先级分配重要性权重, 然后使用 VIKOR 评估和确定每个用户请求的等级并进行排序, 最后自适应任务调度根据虚拟机的负载率来分配工作负载以保持动态负载均衡. 实验结果表明: 本文方法在 MaKespan、吞吐量 and 虚拟机利用率性能方面均有明显改善.

关键词: 云计算; 多准则妥协解排序法; 自适应调度; 最优-最差准则; 服务质量

中图分类号: TP393

文献标志码: A

文章编号: 1000-5471(2021)07-0109-05

近年来, 随着互联网的发展, 云计算为用户服务增加了一种新的模式, 允许用户随时随地以按次付费的方式访问信息技术服务^[1-2]. 但云环境的特征不同于其他环境的特征, 云计算的资源是异构的且分布在很大的区域, 操作环境和这些资源使用的工具不同. 云中的资源通过动态通信层相互关联, 并向用户提供多种服务和计费策略, 云计算的其他特征包括按需自助服务、广域网接入、共享资源池和快速弹性^[3-5]. 由于这些特性, 满足云资源峰值需求的动态负载均衡成为现代高性能计算系统面临的挑战. 云计算中的负载均衡通过将多个客户端的请求提供给服务器组来减少此问题, 从而最大程度地减少了客户端的等待时间^[6].

云计算负载均衡使数据中心可以避免虚拟机(Virtual Machine, VM)中的过载/欠载^[7], 负载均衡算法需要具有以下特征: 成本效益, 即负载均衡算法以较低的成本提供更好的性能; 可扩展性和灵活性, 即负载均衡算法必须具有改变大小的能力, 并且必须对分布式系统中的变化可行; 优先级, 即必须在执行负载均衡之前确定所有工作的优先级^[8-9]. 负载均衡方法主要分为 3 类: 静态负载均衡、动态负载均衡和混合负载均衡^[10]. 静态负载均衡方法通过先到先得策略将任务分配给资源, 此方法使用内存和处理器功能容量之类的信息. 动态负载均衡方法是利用当前系统的状态将任务分配给资源的遗传算法^[11]. 混合负载均衡方法利用静态和动态负载均衡方法将任务分配给资源.

文献[12]提出一种用于在云环境中实现负载均衡的萤火虫算法和改进粒子群优化算法的混合算法, 分别用于制定、改进资源有效利用率和任务响应时间等重要指标, 该算法在全局搜索性能和收敛速度上都有明显的提高. 文献[13]提出一种基于模糊逻辑和蚁群优化的混合算法来改善云计算负载均衡, 该算法采用田口实验设计来确定蚁群优化参数的最佳值, 并定义模糊模块来评估信息素值提高计算时间, 该算法可以改善云计算架构中的负载均衡, 减少相应的时间. 文献[14]提出一种云环境下基于混合启发式和元启发式

① 收稿日期: 2020-07-01

基金项目: 国家自然科学基金项目(61702065); 四川省教育信息化应用与发展研究中心项目(JYXX18-030).

作者简介: 程 曦, 硕士, 讲师, 主要从事计算机应用技术研究.

技术的负载均衡优化算法,用于负载分布均匀的情况下优化虚拟机的利用率.该算法基于启发式技术和元启发式算法的混合,实现其在制造时间和成本方面的最佳性能.

在研究了现有云计算负载均衡的基础上,为了在考虑负载均衡的同时满足用户服务质量要求,本文提出一种基于改进多准则妥协排序法(VIKOR)和用户服务质量(QoS)优先级的云计算自适应调度算法.该算法使用最优-最差多准则决策从用户角度为每个评估标准或用户 QoS 优先级分配重要性权重,VIKOR 通过最大化团体利益和最小化个人负面影响对用户要求进行排序.根据虚拟机的速度和内存能力,设计自适应任务调度方法将用户请求从全局优先级队列分发到属于相应虚拟机的输出队列.当服务器的容量发生超时变化时,该算法可以产生最优的吞吐量.实验结果表明,本文算法能有效提高 MaKespan、吐量出和虚拟机利用率等性能.

1 基于 VIKOR 和用户 QoS 优先级的云计算自适应调度

本文提出的协议首先将用户所有请求放入一个输入队列,然后根据用户需求,采用最优-最差和 VIKOR 混合方法估计用户请求的相对重要性/权重并进行排序,根据 VM 的速度和内存容量来分配工作负载,当服务器的容量发生超时变化时,该算法可以产生最优的吞吐量.具体流程图如图 1 所示.

1.1 从用户角度评估标准权重

由于在决策过程中不同的最终用户具有不同的准则,因此本文使用最优-最差多准则决策为每个评估标准分配重要性权重.该方法考虑两个成对比较向量,并允许评估每个准则相对于输入队列中用户请求的权重.

最优-最差多准则决策的步骤如下:最终用户依赖专家组的判断,并针对其请求的 QoS 参数初始化一组首选标准 $[c_1, c_2, \dots, c_n]$,然后从用户角度分析和确定最优(最理想/最重要)和最差(最不理想/最不重要)标准.

相对于其他准则,确定最佳准则的优先级.这是根据从 1 到 9 的标准进行的. a_{Bi} 值表示最佳准则 B 相对于准则 i 的优先级.然后根据最差准则确定准则的优先级,也是根据从 1 到 9 的标准进行的. a_{iW} 值表示标准 i 相对于最差标准 W 的优先级.最后计算准则权重 $(W_1, \dots, W_n)$,其中所有 i 的最大绝对差 $\left| \frac{W_B}{W_i} - a_{Bi} \right|$ 和 $\left| \frac{W_i}{W_W} - a_{iW} \right|$ 最小化.计算准则权重的最小-最大模型表示为

$$\min_i \max \left\{ \left| \frac{W_B}{W_i} - a_{Bi} \right|, \left| \frac{W_i}{W_W} - a_{iW} \right| \right\} \quad (1)$$

式(1)中 $\sum_i W_i = 1, W_i \geq 0$. W_B 表示最佳准则 B 的权重, W_i 表示准则 i 的权重, W_W 表示最差标准 W 的权重.通过求解上述问题,得到优选准则的最优权重.

1.2 评估和确定每个用户请求的等级

VIKOR 方法是一种良好的多属性决策方法,它通过最大化群体利益和最小化负面影响,使用折衷解决方案得到决策者接受.折衷排序根据折衷编程方法中用作聚合函数 L_{pi} 的排序来计算.假设有 m 个备选方案,表示为 $A_1, A_2, \dots, A_m$, A_i 的第 j 方面的排名由 f_{ij} 表示,该 f_{ij} 指定备选方案 A_i 的第 j 标准函数的

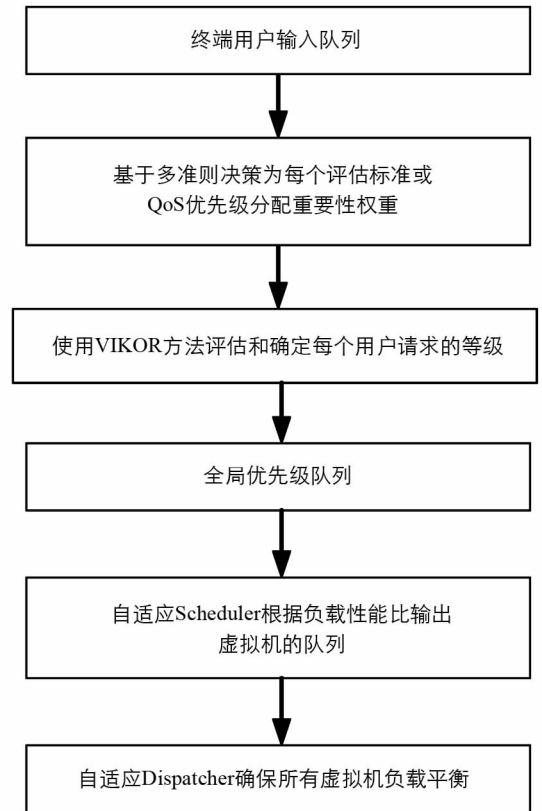


图 1 本文提出的算法流程图

值. 如果考虑 k 个准则, 那么跃变度量可以表示为

$$L_{pi} = \left\{ \sum_{j=1}^k \left[\frac{f_j^* - f_{ij}}{f_j^* - f_j^-} \right]^p \right\}^{1/p}$$

(2)

其中, $1 \leq p \leq \infty$; $i = 1, 2, \dots, m$ 为全局准则的指数, f_j^* 为正最优解, f_j^- 为负最优解.

VIKOR 方法评估有两个规则: 多数规则和反对规则.

1.3 自适应调度将请求分发到属于相应 VM 的输出队列

为了实现云计算环境的动态自适应, 满足不同任务的 QoS 要求, 本文提出一种基于阈值动态负载均衡算法的自适应 scheduler, 将请求分发到属于相应 VM 的输出队列. 如果自适应 scheduler 应用预先定义的时间间隔并以任何间隔将请求分发到相应的 VM 输出队列, 那么确定最佳间隔大小将变得困难. 对于较短的预定义时间间隔, 自适应 scheduler 组件和自适应 dispatchers 之间信息交换可能比长的预定义时间间隔提供更多的网络开销. 此外, 当预定时间间隔太长时, 自适应 dispatchers 对 VM 获取信息提供的精度较低, 从而导致负载不平衡问题. 为了克服这些缺点, 确定一个阈值来与自适应调度器 VM 负载进行比较. 如果 VM 的负载大于阈值, 则自适应 dispatcher 将负载信息发送给自适应 scheduler.

2 实验结果与分析

在本文中, 所有实验均是在一台配置为 Intel Core i7、3.4 GHz CPU 和 4 GB 内存的 Microsoft Windows 10 平台上进行, 使用云计算仿真平台 CloudSim 3.0.3 对本文提出的自适应任务调度算法进行评估. CloudSim 是基于离散事件模型, 用 Java 语言开发的仿真应用系统, 具有跨平台部署等特点.

为了评价本文自适应任务调度算法的性能, 采用吞吐量、MaKespan 和 VM 利用率等指标进行评估. 吞吐量是指虚拟机以最大速率每单位时间执行的用户任务数; MaKespan 是指处理一组提交给系统的任务的完成时间; VM 利用率是指 CPU 使用率的百分比, 其值介于 0 到 1 之间. 将本文提出的算法与基于滑动窗口的自学习自适应负载均衡器(SSAL)算法进行分析比对. SSAL 是一种将时间逻辑划分为固定大小的间隔, 并将任务请求分配给在每个间隔具有最高性能的 VM 的负载均衡算法.

表 1 给出了在 VM 固定为 100 的情况下, 本文算法和 SSAL 在输入任务量不同时的平均 MaKespan^[15]. 由表 1 可以看出, MaKespan 随着输入任务量的增加而增加, 本文方法的 MaKespan 性能比 SSAL 好 3.7%. 这是因为本文方法自适应 scheduler 根据性能负载比对所有 VM 进行排序, 并选择相应的输出队列给相应的 VM, 因此它可以满足要求.

表 1 输入任务数不同时各算法的 MaKespan ms

任务	200	300	400	500	600	700	800
SSAL	179.1	218.4	268.8	308.1	319.1	352.7	379.2
本文	148.2	201.5	232.3	263.1	275.5	281.1	304.8

表 2 给出了在虚拟机(VM)固定为 100 的情况下, 本文算法和 SSAL 在输入任务量不同时的吞吐量. 由表 2 可以看出, 吞吐量随着输入任务量的增加而增加, 本文算法的吞吐量高于 SSAL. 这是因为本文方法在自适应 dispatcher 的 VM 负载上设置阈值, 将 VM 的负载率发送给自适应 scheduler, 而不是周期性地更新负载信息, 然后将用户的任务请求分配给低负载、高性能的 VM, 从而产生最佳吞吐量.

表 2 输入任务数不同时各算法的吞吐量 请求 · s⁻¹

任务	200	300	400	500	600	700	800
SSAL	15.72	16.87	17.77	19.04	21.42	22.94	23.22
本文	22.43	24.21	25.22	26.74	27.39	29.40	32.78

表 3 给出了在虚拟机(VM)固定为 100 的情况下, 本文算法和 SSAL 在输入任务量不同时的 VM 利用率. 由表 3 可以看出, 与 SSAL 相比本文方法提高了 VM 的利用率, 这是因为本文提出的方法基于 VM 当前负载和性能来分配请求, 从而使 VM 有较高的利用率.

表 3 输入任务数不同时各算法的 VM 利用率

任务	200	300	400	500	600	700	800
SSAL	0.522	0.540	0.599	0.602	0.614	0.653	0.690
本文	0.581	0.613	0.652	0.692	0.692	0.749	0.755

表 4 给出了在任务数量固定为 500 的情况下, 本文算法和 SSAL 算法在 VM 数量不同时的 MaKespan. 由表 4 可以看出, 本文算法具有更高的效率, 这是因为本文算法在全局优先级队列中将请求分配给适当的 VM, 基于 VM 最近的评估能力, 它使每个任务都具有时间效率.

表 4 VM 数量不同时各算法的 MaKespanms

VM 数量	20	40	80	120	160	200
SSAL	264.6	232.6	215.9	170.4	126.7	110.9
本文	345.7	332.2	273.3	213.5	189.2	169.9

3 结 语

针对云计算中的多准则决策问题, 本文提出一种基于 VIKOR 和用户 QoS 优先级的云计算自适应调度算法, 该算法执行自适应负载均衡和排序, 并选择合适的任务分配操作来保证服务质量, 使用最优-最差多准则决策从用户角度为每个评估标准或用户 QoS 优先级分配重要性权重, VIKOR 通过最大化团体利益和最小化个人负面影响对用户请求进行排序, 自适应任务调度器根据每个 VM 的负载率将 VM 分配给请求, 并试图强制所有 VM 几乎同时实现任务请求. 当服务器的容量发生超时变化时, 该算法可以产生最优的吞吐量. 实验结果表明, 本文算法在 MaKespan、吞吐量、虚拟机利用率方面均有明显的提高. 未来的工作是进一步完善算法模型, 使其更加切合实际应用场景, 并针对特定的使用场景, 对算法进行针对性的优化.

参考文献:

[1] ISMAYILOV G, TOPCUOGLU H R. Neural Network Based Multi-Objective Evolutionary Algorithm for Dynamic Workflow Scheduling in Cloud Computing [J]. Future Generation Computer Systems, 2020, 102: 307-322.

[2] 陶 勇, 沈济南. 基于动态关键路径的云工作流调度算法 [J]. 计算机应用研究, 2018, 35(5): 1500-1505.

[3] KHORSAND R, GHOBAEI-ARANI M, RAMEZANPOUR M. A Self-Learning Fuzzy Approach for Proactive Resource Provisioning in Cloud Environment [J]. Software: Practice and Experience, 2019, 49(11): 1618-1642.

[4] GHOBAEI-ARANI M, KHORSAND R, RAMEZANPOUR M. An Autonomous Resource Provisioning Framework for Massively Multiplayer Online Games in Cloud Environment [J]. Journal of Network and Computer Applications, 2019, 142: 76-97.

[5] MISHRA S K, SAHOO B, PARIDA P P. Load Balancing in Cloud Computing: a Big Picture [J]. Journal of King Saud University - Computer and Information Sciences, 2020, 32(2): 149-158.

[6] KHORSAND R, GHOBAEI-ARANI M, RAMEZANPOUR M. FAHP Approach for Autonomic Resource Provisioning of Multitier Applications in Cloud Computing Environments [J]. Software: Practice and Experience, 2018, 48(12): 2147-2173.

[7] XU X L, ZHANG X Y, KHAN M, et al. A Balanced Virtual Machine Scheduling Method for Energy-Performance Trade-Offs in Cyber-Physical Cloud Systems [J]. Future Generation Computer Systems, 2020, 105: 789-799.

[8] SAFARI M, KHORSAND R. Energy-Aware Scheduling Algorithm for Time-Constrained Workflow Tasks in DVFS-Enabled Cloud Environment [J]. Simulation Modelling Practice and Theory, 2018, 87: 311-326.

[9] NAYAK S C, TRIPATHY C. Deadline Based Task Scheduling Using Multi-Criteria Decision-Making in Cloud Environment [J]. Ain Shams Engineering Journal, 2018, 9(4): 3315-3324.

[10] 陈 暄, 王大伟, 王常亮, 等. 基于 IFWA-ABC 的云计算资源任务调度算法的研究 [J]. 计算机应用研究, 2019, 36(10): 3022-3026.

[11] SAFARI M, KHORSAND R. PL-DVFS: Combining Power-Aware List-Based Scheduling Algorithm with DVFS Technique for Real-Time Tasks in Cloud Computing [J]. The Journal of Supercomputing, 2018, 74(10): 5578-5600.

[12] GOLCHI M M, SARAEIAN S, HEYDARI M. A Hybrid of Firefly and Improved Particle Swarm Optimization Algorithms for Load Balancing in Cloud Environments: Performance Evaluation [J]. Computer Networks, 2019, 162: 1-15.

[13] RAGMANI A, ELOMRI A, ABGHOOR N, et al. An improved Hybrid Fuzzy-Ant Colony Algorithm Applied to Load Balancing in Cloud Computing Environment [J]. Procedia Computer Science, 2019, 151: 519-526.

[14] KAUR A, KAUR B. Load Balancing Optimization Based on Hybrid Heuristic-Metaheuristic Techniques in Cloud Environment [J]. Journal of King Saud University-Computer and Information Sciences, 2019, 2(10): 1-12.

[15] CHEN Xing, WANG Haijiang, MA Yun, et al. Self-Adaptive Resource Allocation for Cloud-Based Software Services Based on Iterative QoS Prediction Model [J]. Future Generation Computer Systems, 2020, 105: 287-296.

Cloud Computing Adaptive Algorithm Based on VIKOR and User QoS Priority

CHENG Xi¹, SONG Tie-cheng²

1. College of Health Care Industry, Sichuan University of Arts and Science, Dazhou Sichuan 635000, China;
2. College of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Abstract: Aiming at the fact that traditional scheduling algorithms cannot provide an adaptive method that satisfies user quality of service (QoS) while considering load balancing, an adaptive scheduling algorithm has been proposed in this paper for Cloud Computing Based on vlsekriterijumska optimizacijai kompromisno resenje (VIKOR) and user QoS priority. In the algorithm, the optimal worst multi criteria decision is first used to assign importance weight to each evaluation standard or user QoS priority, then VIKOR used to evaluate and determine the level of each user's request and sort it. Finally, adaptive task scheduling allocates workload according to the load rate of virtual machine to maintain dynamic load balance. The experimental results show that the performance of MaKespan, throughput and virtual machine utilization are significantly improved.

Key words: cloud computing; vlsekriterijumska optimizacijai kompromisno resenje (VIKOR); adaptive scheduling; best-worst multi (BWM); quality of service

责任编辑 夏 娟