

改进贪婪算法的云任务调度研究

王 鑫^{1,2}, 王人福¹, 蒋 华¹

(1 桂林电子科技大学 计算机与信息安全学院, 广西 桂林 541004;

2 桂林电子科技大学 海洋信息工程学院, 广西 北海 536000)

摘 要: 为了提高云计算环境中系统整体调度效率, 同时更好地模拟云计算实际应用场景, 提出一种改进贪婪算法的任务调度策略. 该算法综合考虑云环境下任务与虚拟机资源的特征, 据此改进贪婪选择策略, 使其较其他算法更好体现任务与资源之间的调度分配关系. 通过在 CloudSim 仿真平台上进行实验分析, 该算法在云环境下能够比其他常用调度算法具有更好的执行效率; 同时, 该算法较其更具均衡负载的能力, 实验结果显示任务分配的均衡度优于基本调度算法.

关键词: 云计算; 任务调度; 贪婪算法; 负载均衡

中图分类号: TP3 **文献标识码:** A **文章编号:** 1000-7180(2018)02-0109-04

Research on Cloud Task Scheduling with Improved Greedy Algorithm

WANG Xin^{1,2}, WANG Ren-fu¹, JIANG Hua¹

(1 College of Computer Science & Engineering, Guilin University of Electronic Technology, Guilin 541004, China; 2 School of Marine Information Engineering, Guilin University of Electronic Technology, Beihai 536000, China)

Abstract: For the purpose of improving the overall efficiency of the system in the cloud computing environment, and to better simulate the actual application scenarios of cloud computing, a task scheduling strategy to improve the greedy algorithm is proposed. The algorithm takes into account the characteristics of tasks and virtual machine resources in the cloud environment, and improves the greedy selection strategy so as to better reflect the mapping relationship between tasks and resources. Through the experimental analysis on the CloudSim simulation platform, the algorithm can perform better than other commonly used scheduling algorithms in the cloud environment. At the same time, the algorithm has the ability of balanced load, and the experimental results show the balance of task assignment better than common algorithms.

Key words: clouding computing; task scheduling; greedy algorithm; load balance

1 引言

云计算(Cloud Computing)是由分布式计算(Distributed Computing)、并行计算(Parallel Computing)和网格计算(Grid Computing)以及虚拟技术(Virtual Technology)发展而来的一种新型计算模型, 具有很显著的商业价值^[1].

为改进已有虚拟资源调度策略的不足与缺陷, 很多学者从不同角度对资源调度策略进行深入研究, 如文献[2]中的应用遗传算法, 对云环境下的独立任务进行有效的调度. 文献[3]针对传统资源调度算法的缺陷, 提出改进粒子群的混沌数据干扰下的云资源调度方法, 在一定程度上提高了调度效率和系统的均衡度. 文献[4]结合考虑系统整体效率和服

务质量提出一种基于贪心算法的云计算资源调度策略,提高了系统整体效率.文献[5]提出一种基于资源状态蚁群算法的任务分配策略,在算法稳定收敛方面取得不错的表现.这些算法均在某一方面取得不错的效果,但都只是考虑某个影响因素,缺乏对任务整体完成时间和资源均衡负载的综合研究.

针对已有调度策略的不足,本文通过分析云计算任务资源分配问题,综合考虑任务传输时间、资源负载状态和任务完成时间等因素,提出一种综合多维因素改进贪婪算法的资源调度策略,以期快速而高效地为虚拟机资源尽可能地均衡分配任务.仿真结果表明综合考虑多因素的改进贪婪算法具有不错的表现.

2 任务调度问题描述

资源调度是云计算研究的热点问题^[6].在进行资源调度时,不仅要考虑任务总得完成时间,还应该考虑虚拟机资源的负载均衡,因此使得资源调度成为一个多项式复杂问题.云环境的动态性、异构性等使得云计算系统中的任务调度变得非常复杂,采用何种策略能使任务调度的效率达到最高,是任务调度中的关键问题.

资源调度问题的实质在于优化调度性能,即,把任务按照设定的调度策略分发给虚拟机资源节点,以达到减少任务总的执行时间,均衡虚拟机资源运行负载,提高系统整体运行效率的目的^[7].

3 任务调度分配算法

研究云环境下任务资源调度策略,本文以 Google 提出的 Map/Reduce 编程模式为参照对象.在 Map/Reduce 框架中,作业在 Map 阶段时将多个计算任务分成许多个子任务,接着依据某种调度策略将子任务分配给若干个虚拟机资源节点,最终返回执行任务的结果^[8].在本文研究中,我们假定各子任务之间互不相关,且虚拟机之间也相互独立.

3.1 相关定义

假设某一时刻,云计算系统中存在待分配的任务 n 个,此时也有 m 个虚拟机资源可用,其中 $n > m$.为了描述方便,首先定义以下几点.

定义 1 假定虚拟机资源为 V ,其中 V 的属性有虚拟机唯一标识 ID、虚拟机网络带宽 Bandwidth、虚拟机内存大小 RAM 和虚拟机执行速度 MIPS 等,即

$$V = \{ID, Bandwidth, RAM, MIPS, \dots\} \quad (1)$$

因此,在云计算系统中所有的虚拟机资源可表示为集合 $V_m = \{v_1, v_2, \dots, v_m\}$.

定义 2 假定系统中待分配的任务为 T ,其中 T 的属性有任务唯一标识 ID、单个任务长度 Length、执行某项任务所需处理单元的数量 pesNum 及输入输出文件大小等,即

$$T = \{ID, Length, pesNum, InputSize, OutputSize, \dots\} \quad (2)$$

因此,在系统中所有的任务也可表示为集合 $T_n = \{t_1, t_2, \dots, t_n\}$.

定义 3 假定任务与虚拟机资源之间的关系为满射关系,即一个任务只允许被分配到一个虚拟机资源中,而一个虚拟机资源可以执行多项任务.因此,任务与资源的分配关系也可以表示为矩阵 A (Allocation Matrix)^[9]:

$$A = \begin{bmatrix} A_{11} & A_{12} & \dots & A_{1m} \\ A_{21} & A_{22} & \dots & A_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1} & A_{n2} & \dots & A_{nm} \end{bmatrix} \quad (3)$$

式中, A_{ij} 表示任务 t_i 与虚拟机资源 v_j 之间的映射; A_{ij} 为 0 时,说明任务 t_i 未分配给虚拟机 v_j ; A_{ij} 为 1 时,说明任务 t_i 已分配给虚拟机 v_j ; $\sum_{j=1}^m A_{ij}$ 表示虚拟机资源 v_j 执行的任务数.

定义 4 云计算系统中任务 t_i 的传输时间定义为任务从被调度开始至上传到虚拟机 v_j 的任务队列的时间记为 $trans_{ij}$,即

$$trans_{ij} = \frac{InputSize_i}{Bandwidth_j}, i \in \{1, 2, \dots, n\}, j \in \{1, 2, \dots, m\} \quad (4)$$

定义 5 在系统中,任务 t_i 在虚拟机资源 v_j 中完成执行的时间花费记为 $cons_{ij}$,即

$$cons_{ij} = \frac{Length_i}{MIPS_j}, i \in \{1, 2, \dots, n\}, j \in \{1, 2, \dots, m\} \quad (5)$$

定义 6 一个任务的执行时间为任务被调度到虚拟机任务队列的传输时间和任务在虚拟机上执行时间之和,其中,任务的等待时间即为任务 t_{i-1} 在虚拟机 v_j 上的执行时间,则有

$$totalTime_{ij} = cons_{ij} + trans_{ij}, i \in \{1, 2, \dots, n\}, j \in \{1, 2, \dots, m\} \quad (6)$$

对应于定义 3 的任务与资源的分配关系矩阵 A ,可以得到任务总耗时矩阵 T :

$$T = \begin{bmatrix} \tau_{11} & \tau_{12} & \cdots & \tau_{1m} \\ \tau_{21} & \tau_{22} & \cdots & \tau_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \tau_{n1} & \tau_{n2} & \cdots & \tau_{nm} \end{bmatrix}$$

(7)

式中, τ_{ij} 表示任务 t_i 在虚拟机 v_j 上执行时间总和, 其中 $i \in \{1, 2, \dots, n\}, j \in \{1, 2, \dots, m\}$.

3.2 算法描述与流程

贪心算法是一种简单高效的设计某些问题以求其最优解的方法,其基本思想是在求解问题时,先考虑求解出当前情况的局部最优值,再逐步趋近全局最优值,从而节省寻找最优解过程中可能的时间消耗,达到整体效率最优^[10]. 贪婪算法的关键在于贪婪策略的选择,即如何选择贪婪的对象及如何操作贪婪过程.

本文应用贪婪算法时,首先将任务按指令长度 Length 的大小升序排序,同时也将虚拟机资源按指令执行速度 MIPS 的大小升序排序;计算出任务 t_i 在资源 v_j 中执行所需时间 $\text{cons}[i][j]$,计算任务 t_i 传输至资源 v_j 上的耗时 $\text{trans}[i][j]$;将 $\tau[i][j] = \text{cons}[i][j] + \text{trans}[i][j]$ 作为贪婪对象进行迭代,为每个任务提供当前条件下分配方式最优的虚拟机资源;每次分配任务给虚拟机时,贪婪算法都尝试将任务分给虚拟机列表中编号最大的虚拟机资源,然后再进行下一步的贪婪选择操作;每次分配后,任务规模都将减少,每次的任务迭代都产生出对当前任务最合适的资源,迭代完成就可使全局任务分配方式最优.

该贪婪算法流程如下所示.

Step1:初始化任务和资源.

Step2:设定调度算法所需参数.

Step3:初始设置第一个任务分配给编号最大的虚拟机资源 $v_{\max(j)}$.

Step4:设置贪婪算法起始贪婪元素对象.

Step5:遍历任务,开始分配第 i 个任务, $i \in \{1, 2, \dots, n\}$.

Step6:计算当前已分配任务的虚拟机资源运行任务 i 时的时间消耗.

Step7:计算任务 i 在虚拟机 $v_{\max(j)}$ 上运行的时间,并在每次迭代中都以此结果为参照.

Step8:计算任务 t_i 在虚拟机资源 v_j 上的执行时间.

Step9:比较 Step7 和 Step8 的大小,从而确定如何分配虚拟机.

Step10:进行虚拟机资源负载均衡.

Step11:分配当前任务给指定虚拟机,记录当前资源负载,调转到 Step5 继续执行,直到任务分配完成.

Step12:结束任务分配过程.

4 实验与分析

为了验证本文所提调度策略的性能,使用 CloudSim 仿真平台对其进行实验. CloudSim 是澳大利亚墨尔本大学网格实验室和 Gridbus 项目组中推出的一个云计算仿真软件,支持云环境下的基础结构、虚拟引擎和资源调度等的仿真功能^[11]. 实验中用到的对比算法有 Max-min 算法、顺序分配算法、加权轮询算法,通过对实验结果的分析,从而验证本文算法的运行效果.

本文使用 CloudSim 2.1 版本,对仿真平台中和 Cloudlet、Vm 和 DataCenterBroker 类等类进行扩展. 对本文改进的调度算法进行模拟仿真实现,同时也实现 Max-min 算法、加权轮询算法和顺序分配法的调度策略以作对比. 本实验中,构建的仿真环境参数配置如表 1 所示. 其中任务数、任务长度、文件大小及虚拟机带宽和 MIPS 均是静态生成的数据,以此避免动态生成随机数对模拟算法实验性能的影响.

表 1 任务调度仿真实验基本数据配置

名称	值
任务数	10~100
任务长度	10 000~80 000
虚拟机资源数	10
虚拟机带宽	1 000~6 000
虚拟机 MIPS	100~600
虚拟机内存	1 024 MB
输入文件大小	300~1 000 MB

通过实验数据分析可以看出,在任务完成执行时间上,改进的贪婪策略、加权轮询算法和 Max-Min 算法都比顺序分配要高效得多. 从图 1 中可以看出,随着任务数量的增加,应用贪婪算法的调度策略优势较加权轮询算法和顺序分配法尤为明显. 与 Max-min 算法的对比中发现,起初任务量小时,两种算法的任务执行时间相差不大;但任务量慢慢增大后,其间的差别逐渐凸显. 这说明本文所提应用改进贪婪算法的调度策略能够适应大任务量的环境,且与其他基本调度算法相比具有不错的运行效果. 其中,该算法在任务执行完成时间上较轮询调度算

法提高明显, 相对于 Max-min 算法也有较好的表现.

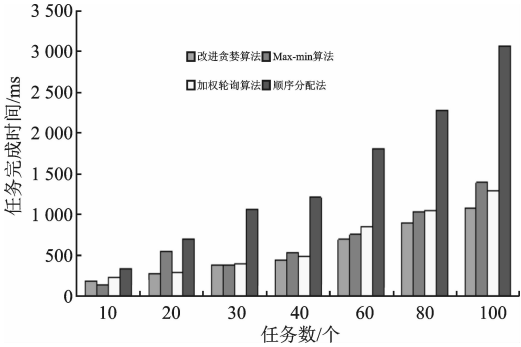


图1 任务完成执行时间比较

为了验证本文算法在任务分配均衡性与虚拟机资源负载均衡性的优势, 在实验过程中, 对各虚拟机在不同任务量下的分配任务数进行统计. 计算分析得出各算法的负载均衡度对比和负载均衡度随任务量变化而改变的趋势. 负载均衡度 β 计算公式是:

$$\beta = \sqrt{\frac{\sum_{i=1}^n (T_i - \bar{T})^2}{n-1}} \quad (8)$$

由于顺序分配算法的思想是对虚拟机资源逐个进行任务分配, 所以在负载均衡度的比较上, 只取图 2 中三种算法进行对比. 从图 2 中可以看出, 任务量为 40 时, 本文所提算法与加权轮询算法在负载均衡性上相差不大; 但随着任务量的增加, 两者在负载均衡性上的差别逐渐凸显. 与 Max-min 算法的比较中发现, 任务量小于 100 时, 该算法稍占优势; 但任务量逐步增大时, 可见本文算的法负载均衡度要优于该算法.

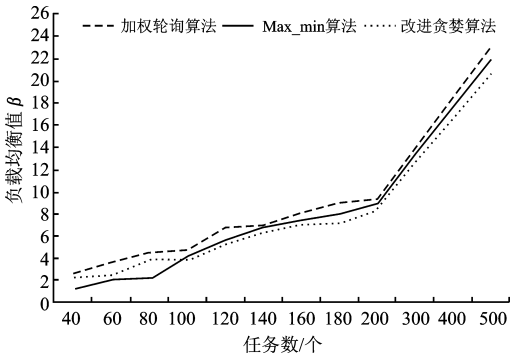


图2 负载均衡度比较

再进一步取总任务数为 180 个时, 比较应用这三种算法使各虚拟机资源承受负载的情况, 结果如图 3 所示, 其中图 3 的任务数是指在某个虚拟机上运行任务数.

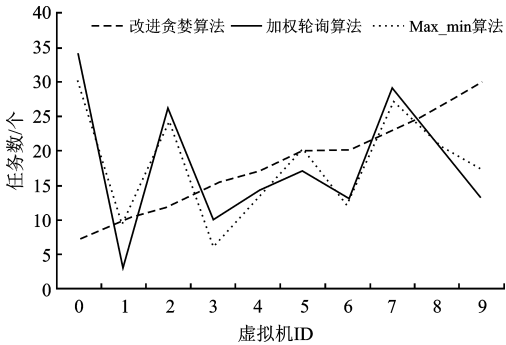


图3 虚拟机负载均衡度比较

通过图 3 可以看出各虚拟机资源承受负载情况, 其中改进贪婪算法的调度策略使得任务分配更均衡, 而且也使虚拟机资源稳定性更好; 而另外两种算法的变化曲线波动幅度大, 体现出任务分配的不均衡度高. 实验结果也显示采用本文改进算法的任务分配策略的均衡度较高于其他两种算法.

5 结束语

针对云环境下任务资源调度问题的学习探究, 本文通过综合考虑任务传输时间、资源负载状态和任务完成时间等因素而提出一种综合多维特征改进贪婪选择方案的调度策略, 并与传统基本调度算法进行比较分析. 在仿真平台上, 搭建仿真环境且实现本文算法及其比较算法, 仿真实验表明这种改进策略具有较好的任务调度效率及资源负载均衡性能. 在实际应用场景中, 影响云计算任务资源调度性能的因素有很多, 例如虚拟机资源与物理主机之间的调度策略和系统能耗等方面, 也是还需要进一步研究的内容.

参考文献:

[1] 陈康, 郑纬民. 云计算: 系统实例与研究现状[J]. 软件学报, 2009, 19(5): 1337-1348.

[2] 张春艳, 刘清林, 孟珂. 基于蚁群优化算法的云计算任务分配[J]. 计算机应用, 2012, 31(5): 1418-1420.

[3] 石峰. 云环境下资源调度效率优化研究与仿真[J]. 计算机仿真, 2016, 32(5): 363-366.

[4] 崔雪娇, 曾成, 徐占然, 等. 基于贪心算法的云计算资源调度策略[J]. 微电子学与计算机, 2016, (6): 41-43, 48.

[5] Ji Li, Longhua Feng, Shenglong Fang. An Greedy Based Job Scheduling Algorithm in Cloud Computing [J]. Journal of Software, 2014, 9(4): 1-136.

[6] 林伟伟, 齐德昱. 云计算资源调度研究综述[J]. 计算机科学, 2012, 42(10): 1-6.

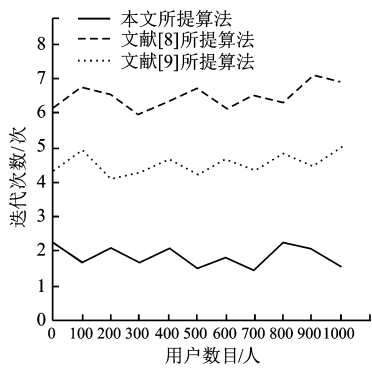


图 5 三种算法迭代次数对比

参考文献:

[1] 刘静静,智淑敏.一种传感器网络不确定感知数据挖掘方法研究[J]. 电子设计工程, 2016, 24(13):73-76.

[2] 李骏骁. 多层差异网络深度入侵数据挖掘方法研究[J]. 计算机仿真, 2015, 32(4):235-238.

[3] 张继荣,王向阳. 基于 XML 数据挖掘的 Apriori 算法的研究与改进[J]. 计算机测量与控制, 2016, 24(6): 178-180.

[4] 汤恒耀,胡志华. 基于链距离估计的非显著特征数据挖掘算法[J]. 科技通报, 2015, 31(6):142-144.

[5] 梅毅,熊婷,罗少彬. 复杂属性环境下 NoSQL 分布式

大数据挖掘方法研究[J]. 科学技术与工程, 2017, 17 (9):239-243.

[6] 侯燕,李巍,文乔农. 基于复杂网络数据流密度的增量子空间数据挖掘算法[J]. 计算机应用研究, 2015, 32(7):2023-2026.

[7] 王莉君,何政伟,冯平兴. 基于 ICA 的异常数据挖掘算法研究[J]. 电子科技大学学报, 2015, 44(2): 211-214.

[8] 唐晓东. 基于关联规则映射的生物信息网络多维数据挖掘算法[J]. 计算机应用研究, 2015, 32(6): 1614-1616.

[9] 张继福,李永红,秦啸,等. 基于 MapReduce 与相关子空间的局部离群群数据挖掘算法[J]. 软件学报, 2015, 26(5):1079-1095.

[10] 宋正江,李晓晨. 基于人工蜂群的 WSN 故障数据挖掘算法[J]. 四川大学学报(自然科学版), 2015, 52 (1):51-56.

作者简介:

王莉莉 女,(1981-),硕士研究生,讲师. 研究方向为大数据、智能信息处理. E-mail:85485579@qq. com.

(上接第 112 页)

[7] 黄俊,王庆凤,刘志勤,等. 基于资源状态蚁群算法的云计算任务分配[J]. 计算机工程与设计, 2014, 34(9): 3305-3309.

[8] 韩秋英,殷秀叶,张少辉. 云环境下基于改进遗传算法的资源调度分配算法[J]. 智能计算机与应用, 2014, 3 (4):100-102.

[9] 王登科,李忠. 基于粒子群优化与蚁群优化的云计算任务调度算法[J]. 计算机应用与软件, 2013, 29(1): 290-293.

[10] 刘鹏. 云计算[M]. 北京:电子工业出版社, 2015.

[11] 刘波,刘青凤. 基于 QoS 和效用的云计算资源调度模型[J]. 计算机测量与控制, 2014, 21(3):826-829.

作者简介:

王鑫 男,(1976-),硕士,副教授. 研究方向为无线传感器网络协议、云计算技术.

王人福(通讯作者) 男,(1993-),硕士研究生. 研究方向为云计算技术. E-mail:1946538748@qq. com.

蒋华 男,(1963-),博士,教授. 研究方向为数据库系统、信息安全.