

# 多维 QoS 约束云任务调度研究

任金霞, 钟小康

(江西理工大学 电气工程与自动化学院, 江西 赣州 341000)

**摘 要:** 云任务调度是目前研究的热点, 为了在给定用户满意度的前提下缩短任务完成时间, 建立了具有用户满意度约束的云任务调度数学模型, 提出一种多维 QoS 约束的改进模拟退火云任务调度算法对模型进行求解. 以任务完成时间为目标, 引入多维 QoS 约束时间贪心策略产生初解, 实行简单模拟退火过程, 始终处于用户 QoS 约束下搜索最佳分配方案. 仿真实验表明, 该调度策略能够保证用户满意度的同时缩短任务完成时间, 是一种用户和云服务提供商同时兼顾的有效调度策略.

**关键词:** 云计算; 服务质量; 贪心策略; 模拟退火算法

**中图分类号:** TP393

**文献标识码:** A

**文章编号:** 1000-7180(2018)07-0097-04

## Cloud Task Scheduling Research with Multidimensional QoS Constraints

REN Jin-xia, ZHONG Xiao-kang

(College of Electrical Engineering & Automation, Jiangxi University of  
Science and Technology, Ganzhou 341000, China)

**Abstract:** Cloud task scheduling is a hotspot of current research, in order to shorten the task completion time on the premise of a given user satisfaction. Mathematical model of Cloud task scheduling with user satisfaction constraint is established. And an improved simulated annealing cloud task scheduling algorithm with multi-dimension QoS constraints is proposed to solve the model in the paper. The task completion time as the goal, time greedy strategy of multi-dimension QoS constraints is introduced to produce initial solution. The algorithm adopt simple simulated annealing process to search the best scheme under the user's QoS constraints. Simulation experiments show that the scheduling policy to ensure customer satisfaction and shorten task completion time. It is a kind of effective scheduling strategy that consider both users and providers of cloud services.

**Key words:** cloud computing ; quality of service; greedy strategy; simulated annealing algorithm

### 1 引言

近年来,随着互联网相关技术的飞速发展,云计算作为一种新兴的商业计算模型出现于公众的视野. 在云计算环境中,用户通过客户端进入云计算系统,根据自身的需求获取计算能力、存储空间和信息服务. 当用户向云计算系统提交任务请求后,调度中心根据用户需求将合适的资源分配给用户. 随着用户数量的增加,需求的多样化和资源池的变化,使得

云计算的任务调度成为一个富有挑战性的问题. 很多研究学者使用经典算法或启发式智能算法来解决这个具有 NP 难属性的问题,例如贪心算法<sup>[1]</sup>、遗传算法<sup>[2]</sup>、粒子群算法<sup>[3]</sup>、模拟退火算法<sup>[4]</sup>、蚁群算法等<sup>[5]</sup>. 目前各相关研究文献中关注最为普遍的是任务完成时间,而用户 QoS 需求的关注则偏少. 本文根据云计算的特点,以任务完成时间为目标,建立具有多维用户 QoS 约束的云任务调度数学模型,再利用融合贪心策略的模拟退火算法进行求解,得出最

佳分配方案.

## 2 云任务调度模型

如图 1 所示,云任务调度的主要过程为:用户通过互联网向云计算系统提交任务请求,预处理器接收到任务请求后对其进行分类、排序等预处理(可分为计算型、储存型和传输型等任务);任务调度器收到处理后的任务集选择合适的数据中心,通过算法按要求得出从任务到资源的最佳分配方案;数据中心根据分配方案将任务绑定到指定的资源上执行;数据中心监测器收集资源的全部信息并反馈给任务调度器.

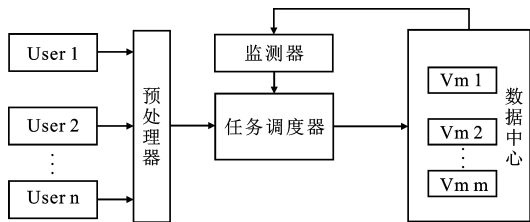


图 1 云任务调度主要流程图

在整个云任务调度过程中,始终困扰云服务提供商的问题是:如何得到既能最大程度满足用户所有 QoS 需求又不降低系统吞吐量的分配方案?要想解决这个难题需要将这个 NP 难问题转化为数学问题,然后对其求解,这也是本文研究的主要内容.

### 2.1 云任务调度数学模型

在本文中, $n$  表示任务数, $m$  表示资源数, $i \in \{1, 2, \dots, n\}$ ,  $j \in \{1, 2, \dots, m\}$ ,  $C$  表示分配矩阵, $n$  行  $m$  列,  $c_{ij}$  为该矩阵第  $i$  行  $j$  列元素;  $T$  表示任务完成时间矩阵,  $n$  行  $m$  列,  $t_{ij}$  为该矩阵第  $i$  行  $j$  列元素, 表示任务  $i$  在资源  $j$  上执行的完成时间;  $S$  表示用户满意度矩阵,  $n$  行  $m$  列,  $s_{ij}$  为该矩阵第  $i$  行  $j$  列元素, 表示任务  $i$  在资源  $j$  上执行的用户满意度.

建立具有用户满意度约束的云任务调度数学模型,表述如下:

$$\min f = \max \left\{ \sum_{j=1}^m (C^{\circ} T)_{ij} \right\} \quad (1)$$

约束条件为:

$$c_{ij} = \begin{cases} 1 & i \rightarrow j \\ 0 & \text{其它} \end{cases} \quad (2)$$

$$\sum_{j=1}^m c_{ij} = 1 \quad i = 1, 2, \dots, n \quad (3)$$

$$\sum_{i=1}^n \sum_{j=1}^m (C^{\circ} S)_{ij} / n > \beta / 100 \quad (4)$$

式中,  $C^{\circ} T$  表示矩阵  $C$  与  $T$  的 Hadamard 积<sup>[6]</sup>, 其结

果还是一个  $n$  行  $m$  列的矩阵;  $C^{\circ} T$  的第  $i$  行  $j$  列的元素  $(C^{\circ} T)_{ij} = c_{ij} * t_{ij}$ . 式(1)为目标函数,求使得任务完成时间最小的分配矩阵  $C$ ,即可得出任务—资源最佳分配方案. 式(2)要求分配矩阵的元素值为 0 或 1,若  $c_{ij} = 1$ ,则表示给任务  $i$  分配资源  $j$ . 式(3)要求每个任务都必须执行,且一个任务只能分配一个资源,不允许重复分配. 式(4)要求分配方案的用户满意度为  $\beta\%$  以上,  $\beta \in (0, 100]$  为满意度系数,服务商可自行选择.

### 2.2 与模型相关的属性及参数

为表述方便,假定本文讨论的任务集中每个任务之间相互独立,用户满意度仅考虑用户的三维 QoS 需求,即 QoS 时间需求、QoS 性能需求和 QoS 费用需求. 根据上述的数学模型,在执行任务的资源中与调度相关的参数有计算能力  $mips$  (百万条指令/s)和与计算能力成正比的资源价格  $price$  (元/百万条指令);任务作为调度的对象,在其属性中与调度相关的参数有任务长度  $tLong$  (百万条指令)、期望资源性能  $tProperty$ 、期望费用  $tCost$  (元)和期望完成时间  $tTime$  (s).

对于任务的执行而言,任务完成时间主要考虑是计算的时间,忽略任务的通信时间,则有:

$$t_{ij} = tLong_i / mips_j \quad (5)$$

式中,  $tLong_i$  表示任务  $i$  的长度;  $mips_j$  表示资源  $j$  的计算能力. 由式(5)可计算出任务完成时间矩阵  $T$ .

**定义 1** 用户 QoS 时间需求满意度矩阵为  $H$ , 第  $i$  行  $j$  列元素  $h_{ij}$  计算公式如下:

$$h_{ij} = \begin{cases} 1 & t_{ij} \leq tTime_i \\ 0 & t_{ij} > tTime_i \end{cases} \quad (6)$$

**定义 2** 用户 QoS 性能需求满意度矩阵为  $P$ , 第  $i$  行  $j$  列元素  $p_{ij}$  计算公式如下:

$$p_{ij} = \begin{cases} 1 & mips_j \geq tProperty_i \\ 0 & mips_j < tProperty_i \end{cases} \quad (7)$$

**定义 3** 用户 QoS 费用需求满意度矩阵为  $E$ , 第  $i$  行  $j$  列元素  $e_{ij}$  计算公式如下:

$$e_{ij} = \begin{cases} 1 & tLong_i * price_j \leq tCost_i \\ 0 & tLong_i * price_j > tCost_i \end{cases} \quad (8)$$

由以上分析及定义可分别求出矩阵  $H$ 、 $P$  和  $E$ , 其元素为 1 表示满足该维 QoS 需求,为 0 则表示不满足. 根据下式(9)可计算出用户满意度矩阵  $S$ , 其元素值最小为 0%, 最大为 100%, 表示满足用户所有 QoS 需求的程度.

$$S = (H + P + E) / 3 \quad (9)$$

3 云任务调度算法

由前一章节对调度数学模型的分析可知,可用启发式智能优化算法对模型求最优解.而模拟退火算法作为一种模仿固体退火过程的智能优化算法,利用 Metropolis 接受准则并适当地控制温度的下降过程实现模拟退火,理论上能够求出全局最优解.传统的模拟退火算法虽然原理简单、使用灵活,但它在云任务调度运用上求解时间较长.本文利用改进的模拟退火算法求解云任务调度问题,从全局到局部全面搜索最优解.

3.1 编码解码与目标函数

本文采用实数直接编码方式,用整数型一维数组  $x[n]$  对解进行进行编码,数组中各元素值从集合  $\{0,1,\cdots,m-1\}$  中取.解码规则按下式(10)可求出分配矩阵  $C$  中  $n$  个元素的取值,然后把其余元素置为 0.

$$c_{ix[i-1]} = 1 \quad i = 1,2,\cdots,n \quad (10)$$

目标函数为式(1),在解码完成后可按目标函数计算得到该分配方案的目标值,算法在迭代寻优过程中以该值为依据.例如:有 4 个任务,3 个资源,最优解为  $x[4] = \{2,3,1,2\}$ ,解码得  $c_{12} = 1, c_{23} = 1, c_{31} = 1, c_{42} = 1$ ,则分配矩阵  $C$  为:

$$C = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

3.2 用户 QoS 约束时间贪心策略

在经典的云任务调度算法中贪心算法是一种操作简单,运行速度快的调度策略.它的思想是每次分配都使得该阶段任务完成时间最短,达到对时间跨度的贪心.本文把用户 QoS 需求融入到贪心算法中,在用户 QoS 需求约束下对时间跨度实行贪心策略,所得分配方案在任务完成时间上不是最短的,但在用户满意度方面是最高的.因此,把它作为算法的初始值对其进行进一步寻优能够更快的找到最优解.用户 QoS 约束时间贪心策略如下.

Step1:按式(5)计算任务完成时间矩阵  $C$ .

Step2:按式(9)计算用户满意度矩阵  $S$ .

Step3:对任务进行资源分配,在矩阵  $S$  该任务所对应的行中找出值为 1 的元素,如果有多个值为 1 的元素再根据矩阵  $T$  计算选出使得当前任务完成时间最小的那个元素,则该任务分配的虚拟机序号就为选中元素的列号,并从任务集中删除该任务,进

入下一个任务分配.

Step4:检查任务集是否为空,如果为否,则转入 Step3;如果为是,则输出任务集所对应的分配方案,算法结束.

3.3 改进模拟退火云任务调度算法流程

根据云计算特性结合本文建立的云任务调度数学模型及 3.2 节的分析,对传统模拟退火算法作出改进.

(1)利用 3.2 节所述的用户 QoS 约束时间贪心策略获取初始解,加快收敛速度.

(2)借鉴遗传算法的交叉操作,使新解由上一代的解随机选取两个元素相互交叉产生,加强局部搜索能力.

(3)在用 Metropolis 准则判断是否接受新解前按式(4)的规定增加用户满意度判断,保证了所得解的用户满意度.算法的目标函数值由式(1)计算得出.算法基本流程如图 2 所示.

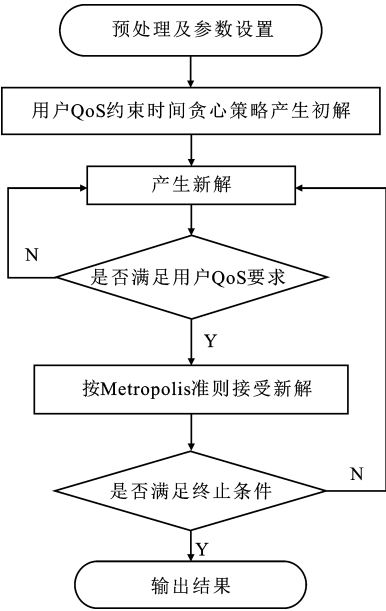


图 2 改进模拟退火云任务调度算法基本流程图

4 仿真实验

本文采用澳大利亚墨尔本大学 Rajkumar Buyya 教授领导团队开发的云仿真器 CloudSim 3.0<sup>[7]</sup>来进行仿真实验.用 Java 编程语言在 MyEclipse10.7 软件下对 CloudSim 平台的 Datacenter-Broker 类进行扩展,实现根据本文算法定义的云任务调度策略的仿真实验,通过对本文算法与传统的模拟退火算法和比较实验中使用较普遍的 Min-Min 算法相比较来验证本文算法执行效率和执行效果的

有效性.

实验中,本文算法初始温度设为 50℃,冷却系数为 0.999,β 设为 99,终止条件为小于 1℃或连续 50 次迭代没有接受新解.传统模拟退火算法与本文算法一致.任务和资源的相关参数按要求随机产生.

4.1 评价指标

云计算系统进行任务调度对云服务提供商来说应保证系统调度的效率及吞吐量,对用户来说应保证服务质量.本文仿真实验设置三个评价指标,第一个为各调度算法从开始到结束的运行时间,可通过电脑获取;第二个为任务完成时间,由式(1)计算得出;第三个为用户满意度 satis,计算公式如下:

$$satis = \sum_{i=1}^n \sum_{j=1}^m (C^S S)_{ij} / n \tag{11}$$

4.2 实验设计

资源数为 100 台虚拟机不变,任务数由 100 开始依次增加 100 个,每组实验重复 10 次取平均值,记录三种算法的执行时间、任务完成时间和用户满意度.如下图所示:图 3 为三种算法执行时间的比较;图 4 为三种算法任务完成时间的比较;图 5 为三种算法用户满意度的比较.

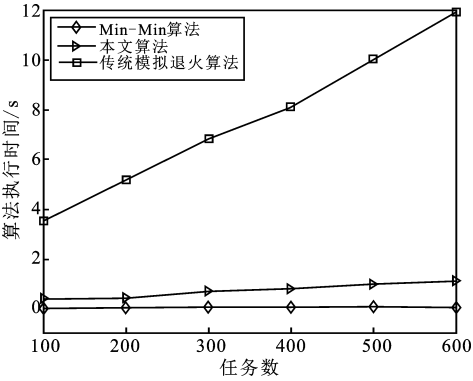


图 3 三种算法执行时间的比较

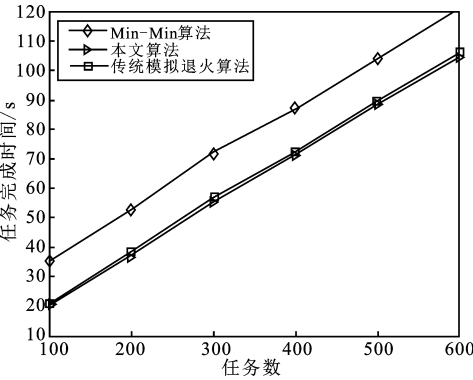


图 4 三种算法任务完成时间的比较

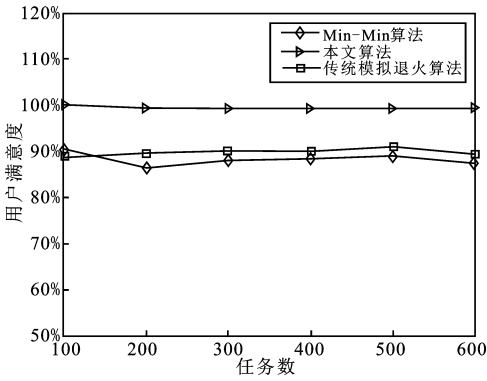


图 5 三种算法用户满意度的比较

4.3 结果分析

从图 3 中可以看出 Min-Min 算法执行速度最快,本文算法的执行速度也非常快,且两者随任务数的增加执行时间增长速度都很缓慢,对比改进前的传统模拟退火算法可知本文算法取得了很大的提高.从图 4 中又可以看出本文算法在任务完成时间方面比 Min-Min 算法好近 25%,与改进前基本相同.同时,图 5 可以看出本文算法在用户满意度方面是三种算法中最好的,基本保持在 100%,而 Min-Min 算法和传统模拟退火算法则在 90%左右,提高了 10%.综合算法执行时间、任务完成时间和用户满意度三个方面来看,本文算法比改进前提高了算法执行效率和用户满意度,任务完成时间上还能保证,同时,在可接受范围内牺牲很小的执行时间取得了比 Min-Min 算法好的任务完成时间和用户满意度.所以,从各方面考虑本文算法是解决云任务调度问题的一种行之有效的方法,能够运用于复杂的实际云环境中.

5 结束语

本文针对云计算任务调度问题以多维用户 QoS 需求为前提,建立了一种具有用户满意度约束的云任务调度数学模型,采用一种融合了 QoS 约束时间贪心策略的改进模拟退火云任务调度算法对该模型求解.通过仿真实验证明本文方法能够提高执行效率和用户满意度并缩短任务完成时间,是一种同时兼顾云服务提供商和用户的有效调度策略.

参考文献:

[1] 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,2016,33(6):41-44.  
[2] 盛小东,李强,刘昭昭.云环境下基于模板遗传算法的任务调度方法[J].计算机应用,2016;36(3):633-636.

搜索能力较强的算子来提升蜻蜓算法的整体性能将是下一步关注的重点;另为如何考虑将改进的蜻蜓算法应用在组合优化问题上也是下一个要研究的重点.

参考文献:

[1] Mirjalili S. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems[J]. Neural Computing & Applications, 2016, 27(4): 1053-1073.

[2] Abdel-Basset M, Luo Q, Miao F, et al. Solving 0-1 Knapsack Problems by Binary Dragonfly Algorithm [C]// International Conference on Intelligent Computing. Liverpool, United Kingdom: Springer, 2017: 491-502.

[3] Tharwat A, Gabel T, Hassanien A E. Parameter Optimization of Support Vector Machine Using Dragonfly Algorithm[C]// Proceedings of the International Conference on Advanced Intelligent Systems and Informatics. Cairo, Egypt: Springer, 2017: 309-319.

[4] Sree R K S, Murugan S. Memory based Hybrid Dragonfly Algorithm for Numerical Optimization Problems [J]. Expert Systems with Applications, 2017, 83(1): 63-78.

[5] 吴伟民,吴汪洋,林志毅,李泽熊,方典禹. 基于增强个体信息交流的蜻蜓算法[J]. 计算机工程与应用, 2017, 53(4): 10-14.

[6] Rainer Storn, Kenneth Price. Differential Evolution-A Simple and Efficient Heuristic for Global Optimization over Continuous space[J]. Journal of Global Optimization, 1997, 11(4): 341-359.

作者简介:

**赵齐辉** 男, (1993-), 硕士研究生. 研究方向为群智能算法的改进与应用. E-mail: 845714323@qq. com.

**杜兆宏** 男, (1987-), 硕士. 研究方向为智能计算, 个性化学习.

**刘 升** 男, (1966-), 教授, 博士. 研究方向为智能计算、机器学习.

(上接第 100 页)

[3] 冯小靖,潘郁. 云计算环境下的 DPSO 资源负载均衡算法[J]. 计算机工程与应用, 2013, 49(6): 105-108.

[4] 余盛季,魏恺明,李强等. 云计算环境下基于时间和可靠性的调度策略[J]. 计算机应用研究, 2016, 33(9): 2673-2678.

[5] 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, 32(5): 1418-1420.

[6] 杨明,刘先忠. 矩阵论(第二版)[M]. 武汉: 华中科技大学出版社, 2005: 136-140.

[7] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Belo-

glazov, et al. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms [J]. Software: Practice and Experience, 2011, 41: 23-50.

作者简介:

**任金霞** 女, (1970-), 副教授. 研究方向为智能控制、智能优化算法.

**钟小康(通讯作者)** 男, (1988-), 硕士研究生. 研究方向为云计算, 智能优化. E-mail: zhongxiaokang88@163. com.