

改进蚁群算法的云计算任务调度方法^①

谢伟增

(河南司法警官职业学院 信息技术系, 郑州 450046)

摘要: 针对蚁群算法在云计算任务调度问题求解过程存在的不足, 以找到最佳的云计算任务调度方案为目标, 提出了一种基于改进蚁群算法的云计算任务调度方法. 首先对当前云计算任务调度研究现状进行分析, 并对问题进行了具体描述, 然后采用蚁群算法对云计算任务调度问题进行求解, 并针对标准蚁群算法缺陷进行改进, 最后在 CloudSim 平台对该方法的性能进行测试. 结果表明, 改进蚁群算法可以找到较好的云计算任务调度方案, 加快云计算任务完成速度, 具有一定的实际应用价值.

关键词: 云计算技术; 任务调度问题; 蚁群优化算法; 仿真测试

Task Scheduling Method of Cloud Computing Based on Improved Ant Colony Algorithm

XIE Wei-Zeng

(Department of Information Technology, Henan Judicial Police Vocational College, Zhengzhou 450046, China)

Abstract: Aiming at the shortage of the ant colony algorithm in the solving process of cloud computing task scheduling problem, this paper presents a novel task scheduling method of cloud computing based on improved ant colony algorithm, in order to find the best cloud computing task scheduling scheme. Firstly, this paper analyzes current status of research on task scheduling in the cloud computing, and describes the problem in detail. And then ant colony algorithm is used to solve the problem of cloud computing task scheduling, and the defects of standard ant colony algorithm are improved. Finally the performance of the proposed method is tested on the CloudSim platform. The results show that the improved ant colony algorithm not only can find better scheduling scheme for cloud computing tasks, but also speed up the completion of the cloud computing tasks, which has a certain practical application value.

Key words: cloud computing technology; task scheduling problem; ant colony optimization algorithm; simulation test

1 引言

在云计算系统中, 任务种类多、规模大, 而任务调度优劣直接影响云计算的服务质量, 因此如何对云计算任务进行合理调度, 成为当前云计算系统中急需解决的问题^[1,2].

多年来, 学者们对云计算任务调度问题进行了研究, 出现了许多云计算任务调度方法^[3]. 通常情况下, 首先采用 Map/Reduce 方式对云计算任务进行处理, 将大规模任务划分成多个子任务, 然后每一个子任务通过 Map 阶段和 Reduce 阶段实现调度^[4]. 相关研究结果表明, 云计算任务调度问题属于 NP 问题, 当前主要采用启发式算法对云计算任务调度问题进行求解^[5,6]. 出

现了基于遗传算法、粒子群算法、蚁群优化算法(ant colony optimization, ACO)^[7,8]的云计算任务调度方法, 通过模拟自然界生物一些行为对问题进行求解, 保证各个节点资源的负责均衡, 尤其是蚁群算法的实际应用范围更广^[9]. 在实际应用, 蚁群优化算法还有自身缺陷, 如易陷入局部最优解, 后期收敛速度, 慢搜索时间长等^[10]. 为了标准蚁群优化算法存在的不足, 一些学者对其进行了相应的改进, 如文献[11]提出基于蚂蚁分类的蚁群算法, 但是其求解结果并不理想.

为了加快云计算任务完成时间, 提高云计算资源的利用率, 针对标准蚁群算法存在不足, 提出了改进蚁群优化算法的云计算任务调度优化方法, 并通过具

① 收稿时间:2016-08-02;收到修改稿时间:2016-10-27 [doi:10.15888/j.cnki.csa.005813]

体化实验验证分析有可行性。

2 云计算任务调度问题

当前云计算系统均采用 Map/Reduce 分布式计算模式处理海量数据，其工作流程如图 1 所示。Map/Reduce 包括 Map 和 Reduce 两个关键阶段，具体如下：

1) Map 阶段将用户提交的任务划分多个子任务，然后给予任务分配相应节点执行，执行后将结果反馈给主节点。

2) Reduce 阶段汇聚 Map 阶段各节点的结果，并将最终结果输出，产生 R 个输出文件。

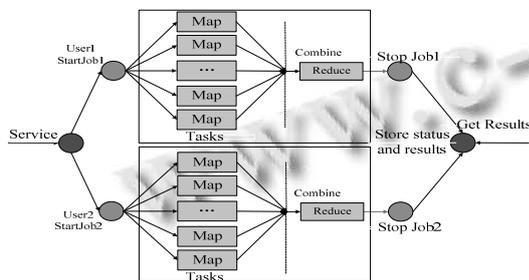


图 1 Map/Reduce 的工作示意图

子任务的数量要远远大于云计算节点数，使得每个节点均要执行多个子任务，合理对子任务分配，保证各节点的负载均衡和完成时间最短，即云计算任务调度算法要解决的问题。

3 改进蚁群算法

3.1 标准蚁群优化算法

Dorigo 等启到蚁群觅食行为的启发，提高了蚁群优化(Ant Colony Optimization, ACO)算法。设蚂蚁数量 m ，蚂蚁均从起始节点出发，它们根据一定概率转移到下一个节点，节点 i 转移到 j 的概率为：

$$p_{ij}^k(t) = \begin{cases} \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{s \in allowed_i} (\tau_{is})^\alpha (\eta_{is})^\beta} & \text{若 } j \in allowed_i \\ 0 & \text{否则} \end{cases} \quad (1)$$

式中， τ_{ij} 为边 (i,j) 上的信息素， η_{ij} 表示与边 (i,j) 相关的启发信息函数； α 和 β 为信息素因子和启发因子。

由式(1)可知，蚂蚁更加容易选择离当前节点距离较短且信息素较多的城市节点。为了防止路径上的信息素无限制累加，将使用挥发信息素的机制，以更有利于发现最优的解。当蚂蚁找到一条起点到终点的路

径后马上更新每条边的信息素，信息素更新规则如下：

$$\tau_{ij}(t+1) = \rho \tau_{ij}(t) + \Delta \tau_{ij}(t) \quad (2)$$

$$\Delta \tau_{ij}(t) = \sum_{k=1}^m \Delta \tau_{ij}^k(t)$$

式中， $\rho \in [0,1]$ 为信息素挥发系数， $\Delta \tau_{ij}^k(t)$ 为第 k 只蚂蚁在本次循环中留在边 (i,j) 上信息素的增量， $\Delta \tau_{ij}(t)$ 为所有 m 只蚂蚁本次循环中留在边 (i,j) 上信息素的增量之和，其计算方式为：

$$\Delta \tau_{ij}^k(t) = \begin{cases} \frac{Q}{L_k} & \text{若第 } k \text{ 只蚂蚁选择了 } (i, j) \\ 0 & \text{否则} \end{cases} \quad (3)$$

式中，信息素大小 Q 为常数， L_k 为第 k 只蚂蚁本次的路径总长度。

3.2 蚁群算法的改进

3.2.1 改进选择下一节点概率计算公式

改进算法中蚂蚁从当前节点 i 选择节点 j 公式为：

$$j = \begin{cases} \arg \max (\tau_{ik})^\alpha (\eta_{ik})^\beta & q \leq q_0 \\ k \in allowed_i & \\ p_{ij}^k(t) = \frac{(\tau_{ij})^\alpha (\eta_{jT})^\beta}{\sum_{s \in allowed_i} (\tau_{is})^\alpha (\eta_{iT})^\beta} & q > q_0, j \in allowed_i \end{cases} \quad (4)$$

式中， $q_0 \in (0,1)$ 是给定的参数， q 是随机数。

当 $q < q_0$ 时， j 取 $allowed_i$ 中使 $(\tau_{ik})^\alpha (\eta_{ik})^\beta$ 求得最大的 k ，能够得到使边 (i,k) 信息素最高、距离 i 最短的节点作为下一节点。

当 $q > q_0$ 时，式(1)的启发信息函数 η_{ij} 采用是当前节点 i 到 j 的距离反比函数，改进算法使用 η_{jT} 作为启发信息函数，采用节点 j 到目标节点的距离来设定期望值，其中距离目标点最近的节点的期望值最大，使得蚂蚁更加倾向于选择离目标节点更近的节点。

3.2.2 启发因子 β 的改进

标准蚁群算法的 β 值是固定的。在初始阶段，信息素值为固定，由式(4)可知，此时信息素对解的搜索没有指导的作用，随着算法的运行，信息素值会根据不同路径而不断更新，体现出全局最优解的信息，此时信息素的作用极大地提高，因此将启发因子 β 设计为一个迭代次数的减函数，即：

$$\beta(i) = b^{1/i} \quad (5)$$

式中， b 为常数， i 为当前迭代次数。

初始阶段 β 值较大，主要依赖于启发信息探索路径，以提高算法的收敛速度，随着循环次数的增加， β 值越来越小，后期依赖于信息素强度来选择路径，

充分发挥信息素的指导作用，能够充分地利用启发值和信息素来搜索路径。

3.2.3 信息素更新的改进

在蚂蚁选中边 (i,j) 后，立即按式(6)更新边 (i,j) 上的信息素：

$$\begin{aligned} \tau_{ij} &= (1-\rho)\tau_{ij} + \rho\Delta\tau_{ij} \\ \Delta\tau_{ij} &= Q/l_k \end{aligned} \quad (6)$$

式中， l_k 是指本次迭代中第 k 只蚂蚁已经走过的路径长度。

当蚂蚁选择了一条边以后适当减少该边的信息素，增加蚂蚁选择其它边的概率，当所有蚂蚁都走完一次循环之后，根据式(7)执行信息素的全局更新，从而使得最佳路径的搜索概率越来越大。

$$\begin{aligned} \tau_{ij}(t+1) &= (1-\rho)\tau_{ij}(t) + \rho\Delta\tau_{ij}(t) \\ \Delta\tau_{ij}(t) &= \begin{cases} \frac{Q}{L_{gbest}} & \text{若}(i,j) \in \text{全局最优路径} \\ 0 & \text{否则} \end{cases} \end{aligned} \quad (7)$$

式中， L_{gbest} 为目前为止得到的最优路径长度。

4 改进蚁群算法的任务调度问题求解

设定改进蚁群算法中蚂蚁访问的为一个 $N \times M$ 的点阵空间，将蚂蚁寻找最优路径的过程看作一个任务调度过程，即蚂蚁走到点 (n,m) 时就将任务 m 分配给节点 n 。每只蚂蚁在每个节点留下的信息素由该节点完成任务的时间决定。每只蚂蚁不重复经过相同的节点。设蚂蚁数为 X ，ACA算法的迭代次数为 e ，本文云计算任务调度方法工作流程如图2所示。

基于改进蚁群算法的云计算任务调度如下：

1)设置参数。设置云计算节点数 N 、任务数 M ；设定蚂蚁的搜索空间，设定迭代次数 e 、蚂蚁只数 X 、参数 α 、 β 、 ρ 等。按照任务调度序列矩阵 L 随机选择蚂蚁的起始点。

2)任务调度(蚂蚁寻找路径)并进行局部信息素更新和矩阵 L 更新，蚂蚁根据节点上信息素计算转移概率并进行移动。

3)对蚂蚁所经过路径信息素进行局部更新。

4)对所有蚂蚁完整的完成一次迭代的信息素进行全局更新。

5)若满足最大迭代次数，则输出最终任务调度分

配矩阵 A ，即最优调度方案，否则跳转到2)继续进行下一次迭代。

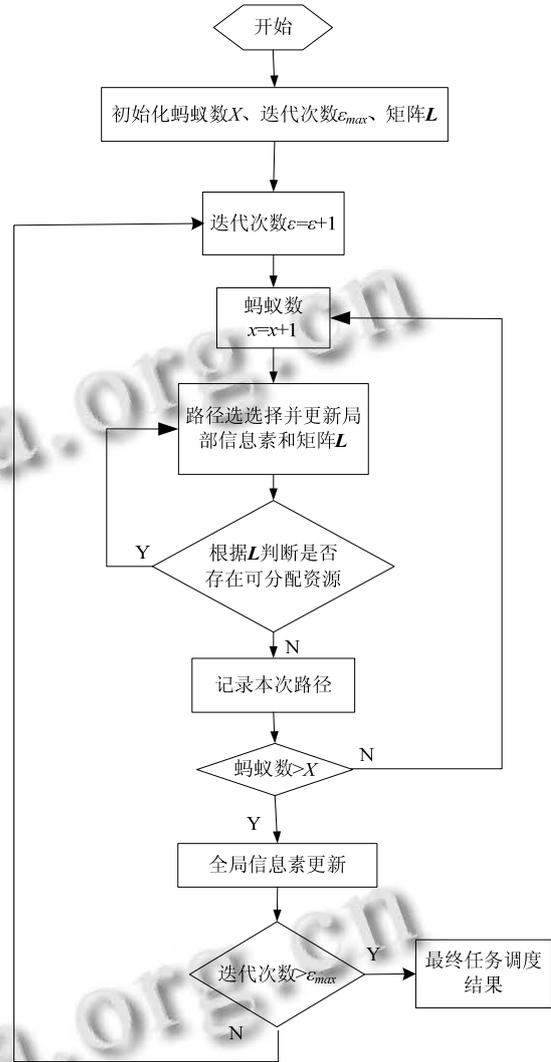


图2 改进蚁群算法的云计算任务调度流程

5 结果与分析

为了分析改进蚁群算法的云计算任务调度优化效果，在CloudSim平台进行仿真测试，选择标准蚁群算法进行对比测试，当任务数为100、节点资源为10时，标准蚁群算法、改进蚁群算法得到最优云计算任务调度优化方案的任务完成时间如图3所示。相对于标准蚁群算法，改进蚁群算法的任务完成时间更少，加快了云计算任务的完成速度，提高了云计算系统的工作效率。

当任务数为1000、节点资源为10时，标准蚁群算法、改进蚁群算法的任务完成时间如图4所示。改进

蚁群算法找到最优云计算任务调度方案,且其迭代次数少于标准蚁群算法,这是由于改进蚁群算法克服了标准蚁群算法易获得局部最优解的缺陷,缩短了云计算系统的任务调度问题求解时间,取得了更好的任务调度效果.

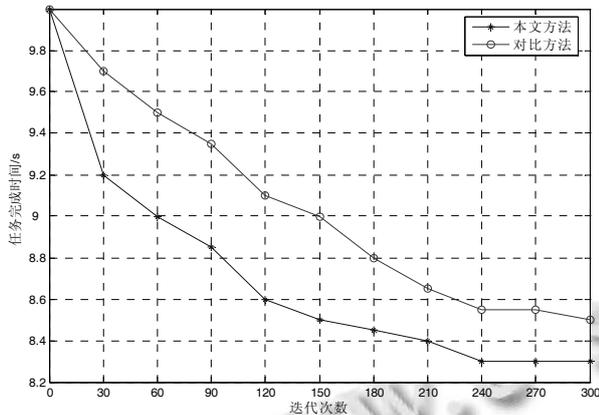


图 3 小规模任务的任务完成时间

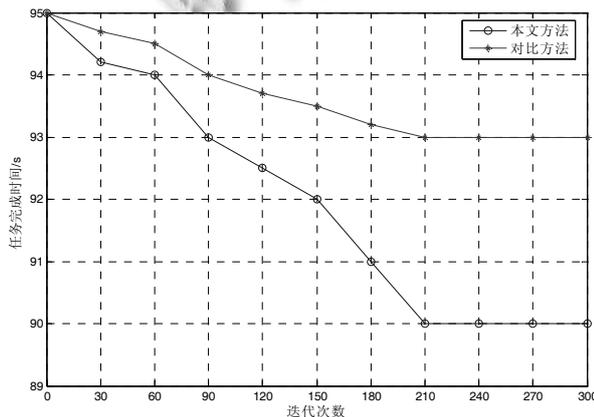


图 4 大规模任务的任务完成时间

6 结语

为了找到最佳的云计算任务调度方案,提出了一种基于改进蚁群算法的云计算任务调度方法.首先云计算任务调度问题进行了描述,然后采用改进蚁群算法对云计算任务调度问题进行求解,最后在 CloudSim 平台对该方法的性能进行测试.结果表明,改进蚁群

算法可以找到较好的云计算任务问题调度方案,加快云计算任务完成速度,是一种性能优异的云计算任务调度问题求解算法.

参考文献

- 1 Chien A, Calder B, Elbert S, et al. Entropia: Architecture and performance of an enterprise desktop grid system. *Journal of Parallel and Distributed Computing*, 2003, 63(5):597-610.
- 2 Rochwerger B, Breitgand D, Levy E, et al. The reservoir model and architecture for open federated cloud computing. *IBM Journal of Research and Development*, 2009, 53(4): 1-17.
- 3 李建锋,彭舰.云计算环境下基于改进遗传算法的任务调度算法. *计算机应用*, 2011, 31(1):184-186.
- 4 Buyya R. *Economic-based Distributed Resource Management and Scheduling for Grid Computing [Thesis]*. Melbourne: Monash University, 2002.
- 5 孙瑞锋,赵政文.基于云计算的资源调度策略. *航空计算技术*, 2010, 40(3):103-105.
- 6 左利云,左利锋.云计算中基于预先分类的调度优化算法. *计算机工程与设计*, 2012, 33(4):1357-1361.
- 7 李欢,虞慧群.云计算中基于进化算法的任务调度策略. *华东理工大学学报(自然科学版)*, 2015, 41(4):556-562.
- 8 华夏渝,郑骏,胡文心.基于云计算环境的蚁群优化计算资源分配算法. *华东师范大学学报(自然科学版)*, 2010, 11(1): 127-134.
- 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. *计算机与应用软件*, 2013, 30(1):291-294.
- 10 张焕青,张学平,王海涛,刘彦涵.基于负载均衡蚁群优化算法的云计算任务调度. *微电子学与计算机*, 2015, 32(10): 31-35.
- 11 王更生,俞云新,等.基于 DACS3 的改进蚁群算法求解 TSP 问题. *华东交通大学学报*, 2010, 27(2):57-61.