

一种快速网格任务调度策略^{*}

王莉¹ 窦旻² 刘宗田¹ 黄美丽¹

(上海大学计算机工程与科学学院 上海 200072)¹ (高能物理研究所计算中心网络安全组 北京 100049)²

摘要 网格任务调度目标有很多,如用户要求任务轮转时间短、花费代价小,而资源提供者希望资源利用率高,这些目标相互冲突,因此网格任务调度不仅是一个 NP 难问题,而且是一个多目标优化问题。本文根据网格环境下任务的时间相关性特点,对传统蚁群算法进行了改进,提出了一种快速网格任务调度算法。该算法不仅解决了网格调度中多目标优化问题,而且依据任务调度历史信息生成蚁群算法的初始信息素分布,提高了蚁群算法的求解速度。

关键词 网格计算,任务调度,蚁群算法,多目标优化

A Fast Algorithm for Grid Job Scheduling

WANG Li¹ DOU Min² LIU Zong-Tian¹ HUANG Mei-Li¹

(School of Computer Engineering and Science, Shanghai University, Shanghai 200072)¹

(IHEP Computing Newsletter Network Security Group, Beijing 100049)²

Abstract In a grid environment, the goals of the user and the resource provider can be conflicting, that is, the user may have the goals of minimizing turn-around time and the cost for the job, while a resource provider may have the goals of improving the throughput of the resources. So grid job scheduling is not only an NP-Hard problem, but also a multi-object optimization problem. In this paper, we propose a grid scheduler based on Ant Colony Algorithm, which satisfies the mutually conflicting goals of the users and the resource provider, as well as improves the velocity of the scheduler, by taking advantage of historical information on scheduling to form the initial information pheromone.

Keywords Grid computing, Job scheduling, Ant colony algorithm, Multi-object optimization

1 引言

二十一世纪初出现的网格技术是互联网发展的第三个里程碑,它试图实现互联网上所有资源的全面连通与协同工作,包括计算资源、存储资源、数据资源、信息资源、知识资源、专家资源等。利用网格的这种超级处理能力,能够解决以前一台高性能计算机无法解决或者不容易解决的大规模科学计算,或处理需要多种资源、多个系统和多门学科参与的重大应用研究问题,网格不但有助于解决物理、天文、生物、工程和模拟等领域的科学计算问题,同时在商业领域、社会生活和国民经济的各行各业方面具有潜在的重大影响。

网格的最终目标是实现对网格资源的优化使用,以完成用户提交的任务,并为用户提供更好的服务质量。在其实施过程中,系统必须根据任务的资源需求和网格资源的状态,对任务所要求的资源进行选择和分配,并进行任务调度和控制。因此,在网格计算中,任务调度是一项基础性研究工作。但由于任务调度本身是一个 NP 难问题,加之网格资源的动态性、分布性、异构性和自治性,使得网格下的任务调度更加复杂,以前针对传统的分布式计算的调度方法已经不能适用,因此网格环境下的任务调度研究是一项很有意义的工作。

已经有一些研究人员在任务调度算法中做了大量研究:O. H. Ibarra 等提出了 Min_min 和 Max_min 启发式算法^[1],M. Maheswaran 介绍了 Sufferage 算法^[2],Abraham 等人介绍了模拟退火等进化算法在网格任务调度中的应用^[3]。这些算法主要考虑如何缩短任务的 makespan^[4],尽可能地提高资

源的利用率和吞吐量。

但实际的网格环境中对各种资源的约束很多,要达到的调度目标也很多,如用户要求所用时间最少、花费代价最小,而网格资源提供者希望资源利用率最高,这些目标相互矛盾,对于这种多目标优化问题找到满足所有约束和目标的全局最优解是很困难的。

本文针对网格调度的多目标优化问题,对传统蚁群算法进行了改进,提出了一种基于蚁群算法的快速网格任务调度算法(Fast Ant Colony Algorithm, FACA),该算法不仅兼顾到资源提供者和用户的需求,解决了网格调度中的多目标优化问题,而且依据任务调度历史信息生成蚁群算法的初始信息素分布,提高了蚁群算法的求解速度。

2 蚁群算法的优化机理

蚁群算法(Ant Colony Algorithm, ACA)是一种受自然界生物的行为启发而产生的智能启发式算法,是从真实蚁群觅食行为的研究中产生的,由意大利学者 M. Dorigo 等最早提出^[5,6]。

生物学家研究表明,蚂蚁能够在没有任何提示下找到从其巢穴到食物源的最短路径,并且能随环境的变化而变化,适应性地搜索新的路径,产生新的选择。其根本原因是蚂蚁在寻找食物时,会在其走过的路上释放一种被称为信息素的化学物质(随着时间的推移该物质会逐渐挥发),蚂蚁可以嗅到这种信息素并且选择信息素浓度大的路径。经过某条路径的

^{*}国家自然科学基金(60575035)资助。王莉 硕士生,研究方向为网格计算、人工智能;窦旻 硕士,研究方向为网络安全、人工智能;刘宗田 教授,博导,从事网络计算、人工智能、软件工程等方面的研究;黄美丽 博士,研究方向为人工智能。

蚂蚁越多,这条路径上的信息素浓度也越大,后来蚂蚁选择该路径的概率也越高,从而形成一种正反馈机制。蚂蚁的这种“正反馈”行为能帮助它们很快找到最短觅食线路。蚁群算法就是受这种行为启发,以人工蚂蚁模拟真实蚂蚁觅食行为的分布式算法,通过信息素调整解的选取,逐步收敛到问题的全局最优解。其优点是:

- ①正反馈性,它通过信息素的不断更新达到最终收敛于最优路径上;
 - ②人工蚂蚁不是对自然界中蚂蚁行为的简单模拟,它融进了人类的智能;
 - ③具有较强的鲁棒性和分布并行性,且易于和其他方法结合;
 - ④它是一种全局优化的方法,不仅可用于求解单目标优化问题,而且可用于求解多目标优化问题;
 - ⑤可扩展性^[7]。所谓可扩展性指的是在原有有一个规模 n 的问题上求出最优解后,再增加 m 个资源,可在原有解的基础上快速找到该问题的 $n+m$ 规模的最优解。
- 但其缺点是:初期信息素匮乏,求解速度慢^[8]。

3 基于蚁群算法的快速网络任务调度策略

3.1 依据任务调度历史信息生成蚁群算法的初始信息素分布,提高求解效率

蚁群算法具有的正反馈性、协同性和隐含的并行性使其适用于分布式系统,而其具有的可扩展性及其可解决多目标优化问题的能力又使之很适合于网格环境的动态性和多目标约束的特点。

但蚁群算法有算法初期信息素匮乏的缺点,导致求解速度较慢。经观察发现,网格中提交的任务有时有一定的时间相关性,即先提交的任务在不久的将来可能会再次被提交。例如一个科学家某段时间提交了一个生成模拟数据的任务,他很可能在不久的将来再次提交这个任务。如果我们维护一个存放任务历史信息以及调度结果的表格,那么利用这些历史信息可以产生有关问题的初始信息素分布,以加快蚁群算法的运行速度(如图1所示)。

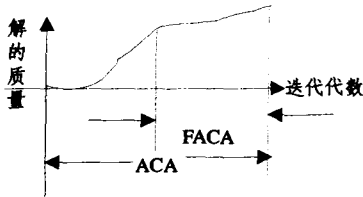


图1 FACA与ACA求解效率示意图

表1 查找表

Job	solution	used
$(j_1, j_2, \dots, j_n)$	(2185...823)	0
	(6531...752)	1
.....		
	(1583...417)	1

在查找表(如表1所示)的每条记录里,包括以下三项:任务集向量 $J=(j_1, j_2, \dots, j_n)$ 、调度结果集向量 $R=(r_1, r_2, \dots, r_n)$ 以及使用位 $used$ 。每个任务集向量 J 包括 n 个任务, j_i 代表第 i 个任务的大小。调度结果集向量中 r_i 表示第 r_i 个资源分配给对应的任务 i 。使用位 $used$ 表示最近一段时间里该

调度结果的使用次数。

对于任意两个任务集向量 $a=(a_1, a_2, \dots, a_n)$ 和 $b=(b_1, b_2, \dots, b_n)$, 用它们的夹角余弦定义二者的相似度 Sim 为:

$$Sim(a, b) = \cos(a, b) = \frac{a \cdot b}{|a| \times |b|}$$

如果相似度 Sim 大于阈值 f , 则将与该任务集对应的调度结果转化为蚁群算法的初始信息素分布, 可加快算法的求解速度。

3.2 调度目标的确定

网格环境中, 用户希望任务轮转时间短、花费代价小、而网格资源提供者希望资源利用率高等, 这些目标相互冲突。为了解决网格调度中多目标优化问题, 我们定义了一个目标函数, 不仅考虑资源提供者提高资源利用率的需求, 而且考虑了网格用户对任务轮转时间及花费代价的需求。网格任务调度问题描述如下:

设共有 n 个不相关任务, m 个计算资源, e_{ij} 表示任务 i 在资源 j 上的期望执行时间 ($1 \leq i \leq n, 1 \leq j \leq m$), r_j 表示资源 j 的可用时间, c_{ij} 表示任务 i 在资源 j 上的期望完成时间, $c_{ij} = r_j + e_{ij}$, $makespan = \max(c_{ij})$, 表示某次调度中所有任务执行完毕需用的最短时间, $T_d(i)$ 表示用户定义的第 i 个任务的截止时间, $T_{jend}(i)$ 表示第 i 个任务的轮转时间, $T_d(i)$ 表示第 i 个任务的延迟, 定义为:

$$T_d(i) = \begin{cases} T_{jend}(i) - T_d(i) & T_{jend}(i) > T_d(i) \\ 0 & \text{else} \end{cases}$$

总延迟 $T_d = \sum_i T_d(i)$, $T_c = \sum_i e_{ij} (i=1, 2, 3, \dots, n)$, 表示一次调度中所有任务的执行时间之和, $cost(i)$ 表示第 i 个任务的执行费用, 总开销 $COST = \sum_i cost(i)$, 因为任务大小不同, 为了找到一个能代表各种需求的目标函数, 需对以上各参数进行标准化, 定义三个子目标:

延迟率 $T_{dn} = T_d / (makespan * p)$, 资源空闲率 $W = 1 - T_c / (makespan * p)$, 开销率 $c = COST / (\max(cost(i)) * p)$, 分别给三个子目标 T_{dn} 、 W 及 c 赋以权值 a 、 b 和 λ (三个权值可以根据任务特点和用户需求设定), 定义总目标函数:

$$F = a * T_{dn} + b * W + \lambda * c.$$

F 越小, 则所求解越优。

3.3 基于蚁群算法的任务调度策略

设有 d 只蚂蚁, n 个任务, m 个资源, 每只蚂蚁独立工作, 1 次工作循环蚂蚁为 n 个任务选择资源, 完成 1 次任务分配, 用 N_c 来记录算法的循环数。

由于任务无先后之分, 机器无主次之分, 为使算法更具一般性, 每只蚂蚁的工作循环可随机的从不同任务和不同机器开始, 由此可得求解任务分配问题的蚁群算法如下:

//初始化。

置查找表为空

系统收集各资源信息

$N_c = 0$ // 算法循环数

$R = 0$ // 资源可用时间向量

计算 e_{ij} // 任务 i 在资源 j 上的期望执行时间

(1) 将任务向量集与查找表中的每一项比较, 计算相似度 Sim , 如 $Sim >$ 阈值 f , 则选择该任务向量对应的调度结果, 且将该向量的使用位 $used$ 加 1; 然后根据选择的优化解生成信息素初始分布。

(2) 在第 k 只蚂蚁的工作循环中, 令 $Tabu_k$ (第 k 只蚂蚁已选择的任务调度方案的集合) 为空; 由式(1)计算第 i 项任

务分配给第 j 个资源的概率 $P_{ij}^k (i=1,2,3,\dots,n)$; 选出其中最大的概率, 记为 P_{is}^k , 即由第 s 台机器来完成第 i 项任务, 然后将该选择置于第 k 只蚂蚁的 Tabu_k 中, 更新资源可用时间 τ_i , 循环直至 n 个任务选择资源完毕。

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \times [\eta_j]^\beta}{\sum_j [\tau_{ij}(t)]^\alpha \times [\eta_j]^\beta} & j \in U \\ 0 & \text{else} \end{cases} \quad (1)$$

式(1)中: 元素 $\tau_{ij} \in \tau_{n \times m}$, 代表任务 i 被分配给第 j 个资源的信息素值, 初值为常数 t ;

元素 $\eta_j \in \eta_m$, 代表第 j 个资源的性价比, 定义为该资源每秒执行的指令数(MIPS)与每秒所需费用的比值;

$P_{ij}^k \in P_{n \times m}^k$ 表示在第 k 只蚂蚁的工作循环中选择第 j 台机器完成第 i 项任务的概率, P_{ij}^k 与先前蚂蚁的工作循环中由第 j 台机器来完成第 i 项任务的信息素量 τ_{ij} 及第 j 台机器的性价比 η_j 有关,

U 为当前可用资源集合; α, β 为可调系数, α 表示先前蚂蚁的工作循环中由第 j 台机器来完成第 i 项任务产生的信息素的重要性, β 表示资源性能的重要性。

(3) 重复第 2 步, 直到 $k=d$ (d 只蚂蚁独立完成一次循环)。

(4) 所有蚂蚁完成一次工作循环, 求出当前最优解 F , 各路径上信息素根据式(2)和式(3)作调整,

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \sum_{k=1}^d \Delta \tau_{ij}^k(t) \quad (2)$$

$$\Delta \tau_{ij}^k(t) = \begin{cases} Q/F_k & \text{若 } (i,j) \in \text{Tabu} \\ 0 & \text{else} \end{cases} \quad (3)$$

式中: $\Delta \tau_{ij}^k(t)$ 表示在第 k 只蚂蚁循环中由第 j 台机器来完成第 i 项任务留下的信息素; F_k 为第 k 只蚂蚁循环中求得的最优解; ρ 表示信息素的持久性; Q 为常数。

(5) $Nc = Nc + 1$, 若 $Nc <$ 预定的迭代次数且无退化行为, 则转 2。否则将得到的最优解输出, 并存入查找表中, 如查找表已满, 则按照 LRU 策略替换。

4 实验结果

我们选用 4 个资源做了一系列实验, 每次调度 20 个子任务, 每个子任务在不同资源上的运行时间和资源之间的传输延迟是随机产生的, 任务之间不相关。算法中各参数设置如下: 查找表大小为 50, 阈值 $f=0.8$, 将查找表所得的优化解转化成信息素值是经过路径加 2, 三个权值分别设为 $a=0.7$, $b=0.2$, $\lambda=0.3$, 各路径信息素初值 τ 设为 10, 参数 $\alpha=1, \beta=1$, 轨迹更新 $\rho=0.8$, 常数 $Q=1$ 。

将本文算法与 Max_min 和 Min_min 两个常用网络任务调度算法比较, 最终实验结果如表 2 所示, 实验证明, 采用 FACA 算法进行任务调度, 时间和代价性能均优于 Max_min 和 Min_min 算法。图 3 和图 4 是 FACA 算法与传统蚁群算法用于网络任务调度的对比, 可以看出由于 FACA 算法依据任务调度历史信息生成蚁群算法的初始信息素分布, 因此比传统蚁群算法收敛快, 求解效率大大提高。

表 2

	FACA	Max_min	Min_min
makespan	417.1	461.5	482.6
费用	2591	3441	3853

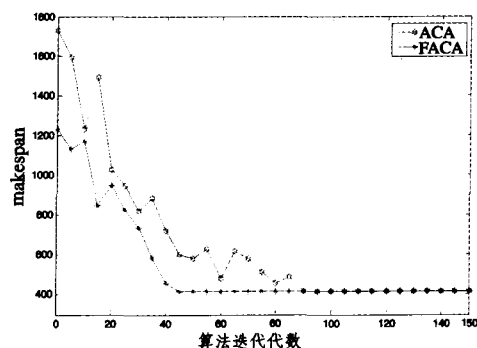


图 2 FACA 与传统蚁群算法关于 makespan 的对比

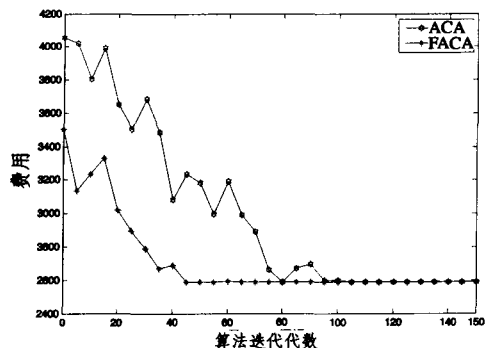


图 3 FACA 与传统蚁群算法关于费用的对比

总结 本文提出了一种基于蚁群算法的快速网络任务调度算法, 讨论了网络调度中多目标优化问题, 而且依据任务调度历史信息生成蚁群算法的初始信息素分布, 提高了蚁群算法的求解速度, 并与传统的 Max_min 和 Min_min 算法做了实验对比, 证明了该算法优于传统网络任务调度算法, 并在求解效率上优于传统蚁群算法。

由于本文对任务的特性做了假设, 只考虑了任务在资源上的执行时间和资源之间的传输延迟这两个因素, 但实际网络环境中的任务特性包括更多的因素, 因此下一步我们将考虑更多网络因素, 例如资源负载的动态变化, 并根据实时的资源信息动态调度任务。

参考文献

- 1 Ibarra O H, Kim C E. Heuristic algorithms for scheduling independent tasks on non-identical processors [J]. Journal of the ACM, 1977, 24(2): 280~289
- 2 Maheswaran M, Ali S, Siegel H J, Hensgen D, Freund R F. Dynamic mapping of a class of independent tasks onto heterogeneous computing systems [J]. Journal of Parallel and Distributed Computing, 1999, 59(2): 107~121
- 3 Abraham A, Buyya R. Nature's Heuristics for Scheduling Jobs on Computational Grids [C]. In: The 8th Int'l Conf. on Advanced Computing and Communications, Cochin, India, 2000
- 4 Pinedo M. Scheduling: Theory, Algorithms, and Systems [J]. Prentice Hall, Englewood Cliffs, NJ, 1995
- 5 Dorigo M, Gambardella, Maria L. Ant colonies for the traveling salesman problem [J]. Biosystems, 1997, 43(2): 73~81
- 6 Dorigo M, Gambardella, Maria L. Ant colony system: A cooperative learning approach to the traveling salesman problem [J]. IEEE Trans on Evolutionary Computation, 1997, 1(1): 53~66
- 7 侯向丹. 蚂蚁算法扩展性及应用研究[D]
- 8 吴庆洪, 张纪会, 徐心和. 具有变异特征的蚁群算法[J]. 计算机研究与发展, 1999, 36(10): 1240~1245