

基于强化蚁群算法的任务 DAG 在线 网格集群资源调度

袁秀丽, 赵连胜

(包头师范学院 信息科学与技术学院, 内蒙古 包头 014030)

摘要: 网格集群资源调度是一个 NP 难题, 而现有的调度方法通常具有任务调度效率低和负载不均衡的问题, 由此设计了一种基于强化学习算法和蚁群算法融合的协同依赖型任务调度方法; 首先对基于 DAG 的网格集群协同调度数学模型进行了定义, 然后, 采用改进的一步 TD 算法即 Q-Learning 算法实现集群资源的初始分配, 从而得到最优调度方案以及对应的 Q 值, 在此基础上提出一种改进的蚁群算法实现网格集群资源到任务分配的进一步优化, 将 Q-Learning 算法得到的分配方案的 Q 值用于初始化蚁群路径中的信息素, 以避免蚁群的盲目搜索, 同时将 Q 值引入路径概率函数中使得蚂蚁具有启发式的搜索能力, 从而获得协同依赖多任务集群调度的最终方案; 在 Gridsim 环境下进行仿真试验, 结果表明文中方法能有效地实现网格集群调度, 且较其它方法具有任务调度效率高、CPU 利用率高和负载均衡的优点, 具有较大的优越性。

关键词: Q 学习; 集群调度; 资源分配; 蚁群算法

On-line Grid Cluster Resource Scheduling Based on Task DAG and Reinforcement Ant Colony Algorithm

Yuan Xiuli, Zhao Liansheng

(College of Information Science and Technology, Baotou Teachers' College, Baotou 014030, China)

Abstract: Grid cluster resource scheduling is a NP problem, the given grid cluster resource scheduling method has the long scheduling time and unbalance system load, a cooperative dependent task scheduling method based on reinforcement learning and parallel ant colony algorithm is proposed. Firstly, the scheduling goal model based on DAG model is defined, then the improved one step TD algorithm such as Q-learning is used to allocate the task resource, and saving the Q value of scheduling schema. Then an improved ant colony algorithm is introduced to realize the allocation of task to the resource node. The Q value obtained from the Q-learning algorithm is used to initialize the pheromone of the route to avoid the search of the ant in ant colony. The Q value is also considered into the probability function to make it has the heuristic ability. The experiment is operated in the Gridsim environment, the result shows the method in this paper can realize the cooperative dependent task cluster scheduling, and compared with the other methods, it has the less task scheduling time and high load balance level, therefore, it is a feasible scheduling method suitable for grid environment with big priority.

Keywords: Q learning; cluster scheduling; resource allocation; ant colony algorithm

0 引言

网格任务资源调度已被证明是一个 NP 难题^[1], 同时它是网格计算研究的重要内容^[2-3]。网格计算资源的共享是通过将地理上分散的计算资源、存储资源、数据资源和信息资源采用高速网络相互连接而实现^[4-5]。

网格任务资源调度可以定义为^[6]: 将实时请求的网格任务分配到合适的网格计算资源上, 使得任务的执行满足一定的目标如执行效率足够高或负载均衡等。

经典的网格调度方法主要包括: Min-min 算法^[7]和 Max-min 算法^[8]。近年来, 一些研究工作开始通过智能优化算法或经济模型用于来解决网格资源调度问题, 如文献 [9] 采用改进的蚁群算法实现资源分配的优化, 在全局信息素和局部信息素的更新中加入 QoS 约束。文献 [10] 为了解决 Min-min 调

度算法的负载不均衡问题, 设计了一种优化的 OPT-Min-min 调度算法。文献 [11] 设计了一种混合组合的拍卖模式, 将信任度引入定价调整中, 实现资源分配的优化, 使得同一节点可以分别扮演资源提供者者和资源需求者的角色。

上述网格集群调度算法均能实现任务资源调度, 但仍然存在在线分配实时性和分配效率不高的问题, 为了解决该问题, 文中提出了一种基于 Q-Learning 算法和蚁群算法相互融合的网络资源在线任务分配算法。

1 网络 DAG 依赖任务模型

1.1 DAG 形式化模型定义

网络依赖型任务模型可以通过图论中的无环有向图 (directed acyclic graph, DAG), 一个网络任务依赖型有向无环图 DAG 可描述为:

网络任务依赖型有向无环图 DAG 可形式化地描述为五元组 $DAG = \{T, E, A, t_{in}, t_{out}\}$:

1) $T = \{t_1, t_2, \dots, t_n\}$ 表示 n 个任务具有依赖关系组成的任务集合;

2) $E = \{(t_i, t_j) | 1 \leq i, j \leq n\}$ 为任务之间关系构成的集合,

收稿日期: 2014-08-09; 修回日期: 2014-09-08。

基金项目: 内蒙古自然科学基金(2014MS0618)。

作者简介: 袁秀丽(1972-), 女, 副教授, 硕士, 内蒙古锡盟人, 主要从事信息技术教育应用方向的研究。

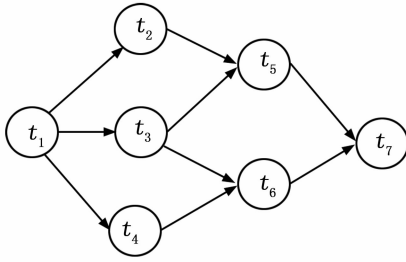


图 1 依赖型任务对应的 DAG

也称为任务依赖集, 当任务之间存在着依赖关系时, 即在对应的 DAG 图中存在对应关系的边。有序对 (t_i, t_j) 表明任务 t_j 和 t_i 执行的先后顺序关系, 即任务 t_j 需等待 t_i 执行完成后才能执行。

3) $A = \{c_i, d_{ij}\}$ 为任务的属性集, c_i 表示任务 t_i 对应的指令数, d_{ij} 每条边的数据传输量表示为

4) t_{in} 和 t_{out} 为任务集的唯一入口任务和出口任务。

1.2 任务最早完成时间

由于依赖关系 (t_i, t_j) 存在, c_{ij} 为数据量, 任务 $t_i \in T$ 所需的数据 t_j 传输到所在节点 $PE(t_i)$ 所需的数据传输时间 t_{ij} : 当所需数据在同一节点上, 则所需时间为 0; 否则为数据量与两节点通信带宽的比值, 可以表示为:

$$t_{ij} = \begin{cases} 0 & PE(t_i) = PE(t_j) \\ c_{ij}/bw(PE(t_i), PE(t_j)) & \text{其它} \end{cases} \quad (1)$$

对于任意任务 $t_i \in T$, 其最早开始时间 $start_i(t_i, PE(t_i))$ 是其所有前驱节点全部执行完成和其分配节点 $PE(t_i)$ 的所有任务均执行完的完成时间的最大值, 记为 $start_i(t_i, PE(t_i))$:

$$start_i(t_i, PE(t_i)) = \max\{start_j + et_j(t_k), start_{PE} + et_{PE}\} \quad (2)$$

在式 (2) 中, $et_j(t_k)$ 表示其任意前驱任务 t_k 的执行时间。

$et_j(t_i)$ 表示任务 $t_i \in T$ 在节点 PE_j 上的执行时间:

$$et_j(t_i) = \frac{m_i}{spe(j)} \quad (3)$$

在式 (3) 中, c_i 为任务 t_i 的计算量, $spe(j)$ 为节点 PE_j 的处理速度。

由上述可知, 任务在节点上的最早完成时间 $finish_i(t_i, PE(t_i))$ 即将任务 t_i 分配到处理机 $PE(t_i)$ 的最早完成时间可以通过下式计算:

$$finish_i(t_i, PE(t_i)) = start_i + et_j(t_i) \quad (4)$$

2 基于 Q-learning 的网格资源分配

2.1 Q-learning 算法

Q-learning 算法是由 Watkins 等人在 1989 年提出一种时间差分算法, 其更新图可以描述为:

在图 2 中, 黑色圆圈表动作, 无色空心圆圈代表状态, 在该状态下可能有几个不同的可执行动作, Q-Learning 算法的思想就是: 通过选择这个状态中的最好的动作 (使得 Q 值最大的动作) 作为 TD 更新中的下一状态 Q 值。

Q-Learning 算法的更新方程可以描述为:

$$Q(x_t, u_t) \leftarrow Q(x_t, u_t) + \alpha[r_{t+1} + \gamma \max_{u'} Q(x_{t+1}, u') - Q(x_t, u_t)] \quad (5)$$

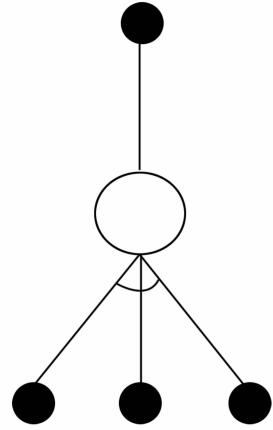


图 2 Q-learning 的更新图

在式 (5) 中, x_t 表示当前状态, u_t 表示当前状态 x_t 下采取的动作, r_{t+1} 表示当前状态动作下的立即回报, x_{t+1} 表示当前状态动作下转到的下一个状态, u' 为在下一个状态 x_{t+1} 选择的能使得 Q 值最大的动作。

2.2 基于 Q-learning 算法的网格资源分配

为了实现对网格集群资源任务分配, 采用 Q-learning 算法获得初始的分配方案, 其原理为: 将待分配的任务作为当前状态, 资源到任务的分配作为可执行的动作, 以最大化长期奖赏即每个分配方案的 Q 值为目标, 每完成一次任务分配, Agent 获得值为 0 的立即奖赏, 当 Agent 完成最后一个任务分配时, 并且优于所有已有方案时, 立即奖赏为 10, 否则为 0。

假设所有任务构成状态空间为 X , 任务到某网格节点的映射构成动作空间 U , 状态动作空间 $X \times U$, 对于状态动作空间中的所有状态动作对 $\forall (x, u) \in X \times U$, 则基于 Q-learning 算法的网格集群资源分配算法可以描述如下。

算法 1: 基于 Q-learning 算法的网格集群资源分配算法。

1) 初始化: 任意状态动作对 Q 值 $Q_0(x, u)$, 最大迭代次数 T , 最大情节数 ET , 每个情节中的最大步数 ETS , 探索因子 ϵ , 折扣因子 γ , 学习率 α ;

2) 当前状态为 x ;

3) 当前情节数 $et = 1$;

4) 当前步数 $s = 0$;

5) 在当前状态 x 下, 采取行为策略 ϵ -greedy 选择动作 u 并执行, 获得下一个状态 x' 和立即奖赏 r ;

6) 在状态 x' 下, 根据公式 (5) 选择最大 TD error 的动作 u' 执行;

7) 根据式 (5) 计算 Agent 在 (x, u) 处的 Q 值;

8) 对探索因子 ϵ 和学习率进行衰减, 并将下一个状态赋给当前状态 $x \leftarrow x'$;

9) 根据行为策略 ϵ -greedy 选择下一个状态 x' 下的动作 u' , 并将其赋给下一个执行的动作 $u \leftarrow u'$;

10) 当前步数 $s = s + 1$, 并判断是否达到最大时间步 ETS ;

如果当前情节结束, 则转入 4) 继续执行;

否则转入 (11);

11) 当前情节数 $et = et + 1$, 并判断是否达到最大情

节数:

如果达到最大情节数 ET , 则并转入 3) 继续执行;

否则转入 12);

12) 算法迭代次数 $t = t + 1$, 并判断是否达到最大迭代次数:

如果当前迭代次数小于 T , 则转入 2) 继续执行;

否则算法结束, 根据学习的 Q 值获取最优的任务动作分配方案。

3 基于并行蚁群算法的网格资源分配

3.1 经典蚁群算法

蚁群算法 (ant colony optimization, ACO) 由意大利学者 M. Dorigo 等人于 1992 年的博士论文中提出的一种图中优化路径的机率型算法, 其主要灵感来自蚂蚁在寻找食物的过程中发现最短路径的行为, 自然界的蚂蚁个体通过相互协作在路径上留下一一种挥发性的化学物质即信息素来实现蚁群蚂蚁个体之间的相互协调, 蚁群通过信息素形成正反馈, 使得路径上的蚂蚁都能根据信息素的分布寻求到一条到达目标的最优路径, 已先后用于旅行商 TSP 问题和资源二次分配问题等最优优化问题中。

3.2 基于 Q-learning 和并行蚁群算法的网格集群调度

基于 Q-learning 算法和并行蚁群算法的网格集群调度的算法原理为: 将 Q-learning 算法获得的各任务的 Q 值作为任务资源分配方案的初始化信息素值, 然后采用多个蚂蚁并行运行的蚁群算法来寻求任务资源的最优分配方案。

假设任务队列集 $TaskS$ 、集群节点集 P , 蚁群 Ant 其运行过程可以描述为算法 2 所示:

算法 2: 基于 Q-learning 和并行蚁群算法的网格集群调度。

初始化, 蚁群中的蚂蚁数量 K , 任务缓存队列 Que , 权值 α 和 β , 信息素挥发率 η , 调节因子 μ , 算法最大迭代次数 T , 目标权重因子 a 和 b ;

1) 将任务网格集群调度的方案编码为:

$$c = \{c_1, c_2, \dots, c_n\} \quad (6)$$

其中, c 为 $1 \times n$ 的向量, 向量的每个分量值对应了分配给该任务的资源号;

2) 运行算法 1 获取最优的任务资源分配方案, 并采用其 Q 值初始化分配方案的 Q 值;

3) 当前迭代次数 $t = 1$;

4) 根据式 (4) 计算所有任务的最早完成时间, 为了考虑负载均衡因素, 在蚁群的目标函数中加入负载均衡离差:

$$F = a \frac{1}{\text{finish}_i(t_i, PE(t_i))} + b \frac{1}{\varphi} \quad (7)$$

$$\varphi = \sqrt{\frac{\sum_{j=1}^m (LB_j - \bar{LB}_j)^2}{m-1}} \quad (8)$$

在式 (7) 和 (8) 中, a 和 b 为目标权重因子, 满足 $a + b = 1$, φ 为负载均衡离差因子, LB_j 为资源节点 j 上分配的任务数, \bar{LB}_j 节点 j 上的负载任务数 LB_j 的均值。

5) 根据式 (7) 计算的 F 值对所有任务排序, 并将其加入到任务等待队列 Que 中;

6) 蚁群中的 K 只蚂蚁从 Que 中选取个 K 个待执行任务:

如果 Que 中的任务数量小于 K , 则派出与队列中任务数相同的蚂蚁数;

7) 集群中的所有资源节点进行信息素更新;

$$\tau_{ij}(t+1) = (1-\rho) * \tau_{ij}(t) + \rho \Delta \tau_{ij}(t) \quad (9)$$

在式 (9) 中, 信息素的局部增量 $\Delta \tau_{ij}(t)$ 可以通过下式:

$$\Delta \tau_{ij}(t) = \begin{cases} \frac{1}{com_{ij}} & ij \in r \\ 0 & else \end{cases} \quad (10)$$

在式 (10) 中, com_{ij} 为路径 ij 的通信时间。

8) 蚂蚁的为其携带的任务分配集群资源节点;

$$p_{ijk}(t) = \begin{cases} \frac{\tau_{ij}(t)^a \cdot Q_{ij}^\beta}{\sum_s \tau_{is}(t)^a \cdot Q_{ij}^\beta} & s \in N_i^k \\ 0 & others \end{cases} \quad (11)$$

在式 (11) 中, Q_{ij}^β 为将当前任务从第 i 个资源节点迁移到第 j 个资源节点的 Q 值, 根据算法 1 可以获得, $\tau_{ij}(t)$ 为路径 ij 上的信息素。

9) 对局部信息素按进行更新;

$$\tau_i(t+1) = (1-\eta)\tau_i(t) - \mu q_i^F(t) \quad (12)$$

在公式 (12) 中, $q_i^F(t)$ 为集群资源节点的处理能力。

10) 判断是否有新任务到来, 如果有新任务到来, 将其加入到 Que 中;

11) 判断 Que 的长度是否为空,

如果为空, 如果为空, 转入 12);

否则保存当前 K 个任务的调度方案, 并转入 4) 继续运行;

12) 当前迭代次数 $t = t + 1$, 并判断当前迭代次数 t 的值:

如果其等于最大迭代次数 T , 则算法结束, 则输出所有任务对应的节点调度方案;

否则, 转入 4) 继续运行。

4 实验分析

为了验证文中方法的正确性, 在网格计算仿真环境 Grid-Sim 下对文中方法进行验证, 实验参数为: 网格中的集群数为 6, 各集群的节点数分别为 50、100、80、70、80、90, 每个节点的处理器数为 4~8, 用户数为 1~50, 用户的任务数为 0~20。

算法 1 参数设置为: 状态动作对 Q 值 $Q_0(x, u) = 0$, 最大迭代次数 $T = 100$, 最大情节数 $ET = 200$, 每个情节中的最大步数 $ETS = 100$, 探索因子 $\epsilon = 0.01$, 折扣因子 $\gamma = 0.8$, 学习率 $\alpha = 0.1$;

算法 2 参数设置如下: 蚁群中的蚂蚁数量 $K = 50$, 算法最大迭代次数 $T = 50$, 信息素挥发率 $\eta = 0.3$, 调节因子 $\mu = 0.8$, 权值 α 和 β 分别为 0.6 和 0.4,

调节因子 $\mu = 0.6$, 权值 α 和 β 分别为 0.6 和 0.4, 最大迭代次数设置为 50 次, 信息素的挥发率 $\eta = 0.4$, 调节因子 $\mu = 2$, 目标权重因子 a 和 b 分别为 0.45 和 0.55。

为了比较文中方法较其它方法的性能改进程度, 将文中方法与文献 [9] 中的基于 QoS 的方法—QoS-Ant 方法和文献 [10] 中的 OPT-Min-Min 方法进行比较, 从所有任务的最晚任务最早完成时间和负载均衡度两个方面进行比较。

两种方法对应的最晚任务最早完成时间如图 3 所示。

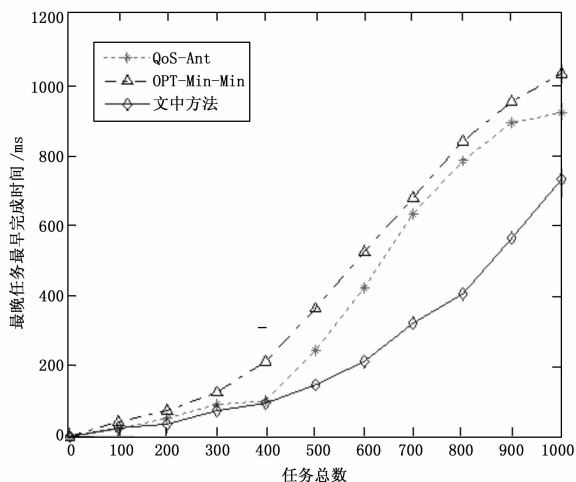


图 3 最晚任务最早完成时间对比

从图 3 中可以看出，当用户任务数从 0 增加到 1 000 个的过程中，文中方法的最晚任务的最早完成时间都远低于 QoS-Ant 方法和 OPT-Min-Min 方法，QoS-Ant 方法得到的最晚任务的最早完成时间略小于 OPT-Min-Min 方法，3 种方法的平均最晚任务的最早完成时间分别为 378.18、440.18 和 236.63，文中方法较 QoS-Ant 方法和 OPT-Min-Min 方法分别低 37.43% 和 46.24%，显然文中方法的调度效率更高。

3 种方法得到的网络资源的负载均衡离差如图 4 所示。

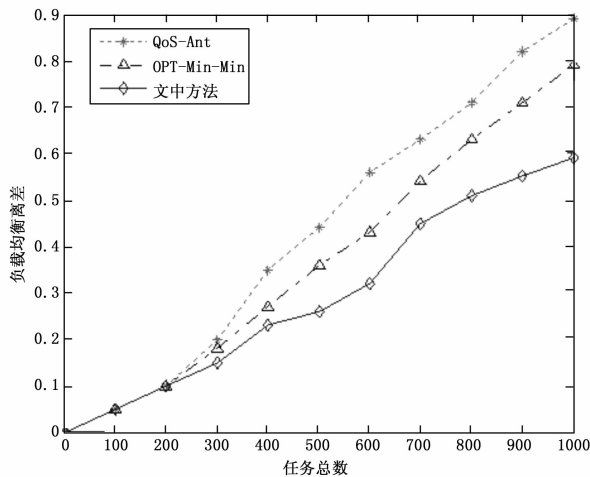


图 4 负载均衡离差

从图 4 中可以看出，3 种方法得到的负载均衡离差随着任务总数的增加均呈增加趋势，3 种方法在任务数小于 200 时负载均衡离差基本相同，但文中方法在任务数大于 200 后，其负载均衡离差就远低于 QoS-Ant 方法和 OPT-Min-Min 方法，且在 700 个任务以后增加趋势变缓。这是因为在任务数较少时，各节点的任务等待队列中任务个数均较少，因此，负载均衡程度较高，而在后期，由于任务数过多，因此，各节点的等待队

列都较长，因此，负载均衡离差改变较少，文中方法、QoS-Ant 方法和 OPT-Min-Min 方法的平均负载均衡离差分别为 0.291 8、0.431 8 和 0.369 1，显然，文中方法具有更好的负载均衡能力。

5 结语

网格计算是一种能通过共享源协同工作的形式为用户提供服务的分布式计算模式，网格集群资源调度是网格计算研究的重要内容，但以往方法往往具有任务执行时间长和负载均衡能力不佳的缺点，因此，文中，提出了一种基于 Q-learning 算法和并行蚁群算法的网格资源分配算法。首先采用 Q-learning 算法实现网格任务资源分配，然后采用并行的蚁群算法进行寻优，得到最终的任务资源分配。通过仿真实验证明，文中方法具有任务调度效率高和负载均衡能力强的优点，且能处理在线任务请求，具有很强的实用性。

参考文献:

- [1] Chervenak A, Foster C I, Kesselman C, et al. The data grid: Towards an architecture for the distributed management and analysis of largescientific datasets [J]. Journal of Network and Computer Applications, 2000, 23 (3): 187-200.
- [2] 李明楚, 许雷, 孙伟峰, 等. 基于非完全信息博弈的网格资源分配模型 [J]. 软件学报, 2012, 23 (2): 428-438.
- [3] Kwok Y K, Song S S, Hwang K. Selfish grid computing: Game-theoretic modeling and NAS performance results [A]. In: Proc. of the IEEE Int'l Symp. on Cluster Computing and the Grid [C]. Washington: IEEE Computer Society Press, 2005: 1143-1150.
- [4] Li J Y, Qiu M K, Ming Z, et al. Online optimization for scheduling preemptable tasks on IaaS cloud systems [J]. Journal of Parallel and Distributed Computing, 2012, 72 (2): 666-677.
- [5] 刘媛, 马晓雷, 刘元安, 等. 基于含虚拟卖家组合双向拍卖的网格资源管理 [J]. 吉林大学学报 (工学版), 2012, 5 (42): 1302-1308.
- [6] Kamalam G K, Murali Bhaskaran V. New enhanced heuristic min mean scheduling algorithm for scheduling meta-tasks on heterogeneous grid environment [J]. European Journal of Scientific Research, 2012, 70 (3): 423-430.
- [7] Braun T D, Siegel H J, Beck N. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems [J]. Journal of Parallel and distributed computing, 2001, 61 (6): 810-837.
- [8] Moreno R J. Scheduling and resource management techniques in dynamics grid environment [A]. The 8th Int'l Conf on Advanced computing and communications (ADCOM 2000) [C]. Cochín, 2000.
- [9] 穆瑞辉, 苗国义. 基于 QOS 约束的启发式网格资源分配算法研究 [J]. 计算机测量与控制, 2012, 20 (12): 3357-3360.
- [10] 张忠平, 温利娟. OPT-Min-Min: 基于 Min-Min 网格资源调度算法的优化 [J]. 小型微型计算机系统, 2014, 35 (7): 1573-1577.
- [11] 肖迎春, 王汉武, 李梦雄. 基于混合组合双向拍卖的网格资源分配方案 [J]. 计算机科学, 2014, 41 (5): 150-172.