

文章编号: 1005 - 0523(2008)01 - 0085 - 04

基于随机高级 Petri 网改进型网格资源调度算法

李培松, 刘觉夫, 周 娟

(华东交通大学 信息工程学院, 江西 南昌 330013)

摘要: 网格环境下资源是分布的、异构的、动态的, 因此对网格资源管理和调度是具有挑战性的问题. 文章采用随机高级 Petri 网模型来描述网格资源调度问题, 并在此基础上改进了 Sufferage 算法. 考虑了网格调度执行前输入数据和执行结束后输出数据的存取和传输时间及调度的执行时间. 还在追求任务最小完成时间的同时考虑了用户 QoS 要求. 仿真试验结果表明, 改进型算法能够大幅提升网格平台满足用户 QoS 要求的能力.

关键词: 随机高级 Petri 网; Sufferage 算法; QoS

中图分类号: TP393

文献标识码: A

网格计算^[1]技术关心的主要问题是如何在动态的、多机构的虚拟组织中协调资源共享和协同解决问题. 由于网格中的资源具有广域分布、异构和动态的特性, 而且为不同的组织所拥有, 各组织对资源的管理机制和策略也不尽相同, 因此如何对网格中的资源进行有效的管理和调度是网格计算技术的关键内容之一.

典型的网格资源调度算法包括: 快速贪吃算法 (Fast - Greedy)、贪吃算法 (Greedy)、极小 - 极小算法 (Min - Min)、极大 - 极小算法 (Max - Min)、忍耐算法和老化算法^[2]. 由于网格环境的动态特性, 调度算法所需要的部分重要性能参数 (如节点的本地负载变化等) 是不确定和难以预测的^[3]. 因此, 传统的调度算法得到的调度结果与实际的调度结果之间存在很大的偏差. 因此首先分析了随机高级 Petri 网模型, 并对 Petri 网模型进行了改进, 可以防止随网格系统参数的增加而发生空间爆炸. 再在 Sufferage 算法的基础上, 提出了一种改进型 ASufferage (Ameliorate Sufferage) 启发式调度算法. 此算法基于随机高级 Petri 网模型, 综合考虑了执行前输入数据及执行完成后输出数据的存取和完成时间以及调度的执行时间. 在追求最小完成时间的同时兼顾了用户的 QoS 要求. 最后对算法进行了仿真试验, 并对试验结

果进行了分析.

1 网格资源调度的随机高级 Petri 网模型

Petri 网最早是由 Carl Adam Petri 博士提出, 随机高级 Petri 网 SHLPN 将指数分布的变迁实施时间引入到变迁集中, 特别适合对于并行和分布系统的模拟和分析^[4-5].

在 SHLPN 中, 任务的到达和接受任务由实践变迁来表示, 到达和服务的速率与系统的状态参数有关. 任务进入缓冲队列和共享互斥区由瞬时变迁表示, 瞬时变迁比时间变迁有更高的实施优先级. 缓冲队列由位置来表示, 允许变迁的实施条件用变迁的可实施谓词规定, 当谓词条件满足时, 变迁才能实施.

给出的随机高级 Petri 网模型如图 1 所示, 为了更好的表示各参数, 作如下约定:

(1) 网格系统中包含 m 个任务, n 个服务, 其中的第 i 个任务, 记作 R_i , 第 j 个服务记作 S_j ($1 \leq i \leq m, 1 \leq j \leq n$).

(2) 每个服务包含一个缓冲队列, S_j 的队列有标示符 q_j 表示, 其缓冲空间的容量为 b_j .

收稿日期: 2007 - 12 - 16

作者简介: 李培松 (1982 -), 男, 山东菏泽人, 硕士生, 主要研究方向: 网格计算, 分布式计算.

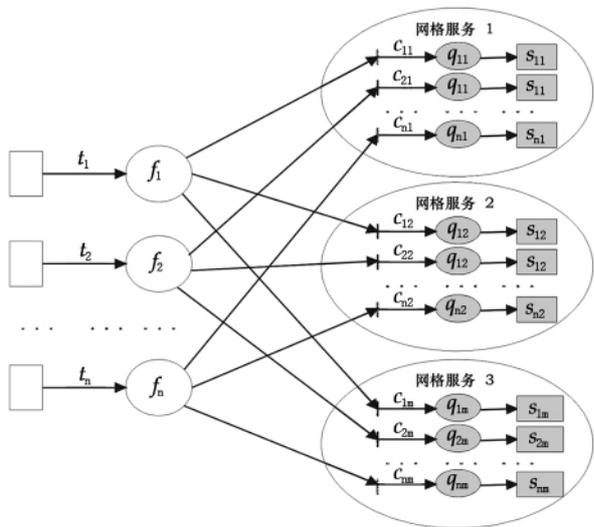


图1 网络调度随机高级 Petri 网模型图

(3) 任务的到达为泊松分布过程. 到达速率为 λ_i , 被分配到 n 个服务队列中任一队列, 当 n 个队列都满时, 任务的到达过程中断.

(4) 每个服务对于不同任务有不同的服务时间, S_j 的服务速率为 V_j , 服从独立指数分布.

在此模型中, 资源的变迁和位置的符号约定如下:

t_i 表示 R_i 到来时的时间变迁, 具有实施速率 λ_i .

f_i 表示对于任务 R_i 分配的位置.

c_{ij} 表示将任务 R_i 分配给网格服务 S_j , 即调度的执行, 由此可实施谓词和随机开关表达.

q_{ij} 表示接受任务的队列, 容量限定为 b_{ij} .

s_{ij} 表示服务 S_j 对任务 R_j 进行处理, 即服务变迁, 具有实时速率 V_j . 在图 1 中, 调度策略可由变迁 d_{ij} 的可实施谓词和实施速度加以描述.

变迁 d_{ij} 的可实施谓词可以表达为:

$$(M(q_{ij}) < b_{ij}) \wedge (\forall k, 1 \leq k \neq j \leq m, (M_i(q_j) \leq M_i(q_k)) \vee (M(q_{ik}) = b_{ik}))$$

变迁 d_{ij} 的实施概率为:

$$P(d_{ij}) = \begin{cases} \frac{1}{\|Q\|} & \text{if } j \in Q \\ q & \text{otherwise} \end{cases}$$

其中,

$$Q = \{k | M_i(q_k) = \min(M_i(q_i)), M(q_2) \wedge M_i(q_m) \text{ and } M(q_{ik}) < b_{ik}\}$$

随着网格系统的参数 m, n 和 b_{ij} 的增长, 可以把模型 A 分解为 n 个子模型 A_1, A_2, \dots, A_n , 每个子模型仅包含由时间变迁 d_{ij}, s_{ij} 和 q_{ij} . 调度策略可由 d_{ij} 和 s_{ij} 的可实施谓词和实施速率进行描述. 这样可以

防止模型的状态空间爆炸.

2 改进型网格资源调度算法

2.1 问题定义

拟解决的问题是如何将 n 个相互独立的任务 R_1, R_2, \dots, R_n 分配给网格系统中的 m 个计算资源 S_1, S_2, \dots, S_m . 任务的模式基于图 1 的随机高级 Petri 网模型. 假设任务在每个计算资源上的执行时间、输入输出数据的存储时间和传输时间是可以预测的, 任务只有获得所输入数据后才能在计算资源上执行, 并且只有在执行完成后才将结果数据一起传送到远程存储资源, 对于每个任务 R_i , 定义如下参数:

ER_{ij} : 任务 R_i 在 S_j 上的执行时间

IR_{ij} : 任务 R_i 分配到 S_j 后, 所有输入数据从相应存储资源中取出并传送到 S_j 的所需时间

OR_{ij} : 任务 R_i 分配到 S_j 后, 所有输出数据传送并保存到相应存储资源的所需时间

PR_j : S_j 上的最早空闲时间

CR_{ij} : 任务 R_i 分配到 S_j 后的完成时间

QoS_i : 任务 R_i 的 QoS 要求参数

2.2 对 Sufferage 启发式算法的改进

(1) Sufferage 算法在计算任务完成时间时只考虑了执行时间, 而忽略了任务执行开始前输入数据以及执行完成后输出数据的存取和传输时间. 在实际的网格环境中, 有时候任务数量非常大, 并且网络带宽竞争比较激烈, 把相同的任务调度到不同的计算资源上存取和传输的时间差别很大, 所以必须考虑资源调度执行前输入数据以及执行完成后输出数据的存取和传输时间对调度策略和调度性能的影响. 针对这些情况, ASufferage (Ameliorate Sufferage) 启发式算法在执行资源调度时综合考虑了执行前输入数据及执行完成后输出数据的存取和完成时间以及调度的执行时间.

(2) 资源调度过程中要尽可能保证用户的 QoS 要求. Sufferage 启发式算法的调度目标是一味追求最小的完成时间, 而没有考虑任务的 QoS 要求, 这样就会造成过多的任务达不到 QoS 要求而被抛弃. ASufferage 算法基于简化的随机高级 Petri 网模型, 考虑了任务的动态性及实时性, 即任务在提交时给出截止时间 SL_i . 如果任务 R_i 调度到资源 S_j 上获得最早完成时间为 CR_{ij} , 调度到 S_k 上获得次早完成时间为 CR_{ik} , 而且 $CR_{ij} \leq SL_i, CR_{ik} > SL_i$, 则认为如果不把资源 S_j 分配给 T_i , 则 R_i 达不到其 QoS 要求, 其

Sufferage 值最大 (设定为固定最大值 Sufferage. Max). 这种调度策略能尽可能保证用户的 QoS 需求. 减少被抛弃任务的概率. 改进型 ASufferage (Ameliorate Sufferage) 算法

- 1 输入: 请求任务 R 与网格资源 S 的信息
- 2 输出: 任务最小完成时间及被抛弃的任务数
- 3 For 所有任务 R 中的任一 R_i
- 4 For 所有资源 S
- 5 任务 R_i 分配到资源 S_j 上的完成时间 $CR_{ij} = IR_{ij} + PR_j + ER_{ij} + OR_{ij}$
- 6 Endfor
- 7 Endfor
- 8 Do 直到所有任务被分配
- 9 For 每一个任务 R_i
- 10 计算最早完成时间 CR_{ij} , 如果 $CR_{ij} > SL_i$, 则抛弃 R_i
- 11 Endfor
- 12 把所有资源标记为未分配
- 13 For 任一个 R_i
- 14 查找资源 S_j 的次早完成时间 CR_{ik}
- 15 If($CR_{ij} \leq SL_i$ and $CR_{ik} > SL_i$)
- 16 Sufferage value = Sufferage. MAX
- 17 Else Sufferage value = $CR_{ik} - CR_{ij}$
- 18 If 资源 S_j 未被调用
- 19 指派任务 R_i 到资源 S_j , 标记 S_j 为已分配
- 20 Else
- 21 如果已分配到 S_j 上的 R_s 的 Sufferage 值小于 R_i 把 R_s 标记为未分配,
- 22 把 R_i 分配给给资源 S_j
- 23 Endfor
- 24 更新标记资源的最早空闲时间 PR 和任务完成时间 CR
- 25 Enddo

3 仿真结果及性能分析

3.1 仿真试验环境及算法设置

在仿真试验中, 用网格模拟器仿真由 $n = 20$ 个计算服务提供者组成的网格系统, 研究 50 ~ 400 个独立服务请求集合的调度情况.

计算资源属性: 安全级别 PS 设置为四个级别 { poor, low, medium, high }, 每个服务的安全级别在此区间内随机分布. 资源的单位时间失效率 FP 在

区间 $[0.0001, 0.0015]$ 上随机分布.

计算任务属性: 服务安全需求级别 DS 设置与服务提供者安全级别设置相同. 每个计算任务在各个资源上的执行时间、远程存取及传输输入输出时间、截止时间均在一定范围内随机产生. 每组试验做 30 次, 结果取其平均值.

3.2 试验结果及性能分析

本试验主要是测试不同任务数量情况下两种算法的性能. 图 2 给出了一对调度效果的影响. 有图中可以看出, 随着任务数的增加, 两种算法下的 makespan 都在增加, 但 ASufferage 算法下的 makespan (最小完成时间) 明显比 Sufferage 算法下的 makespan 要小.

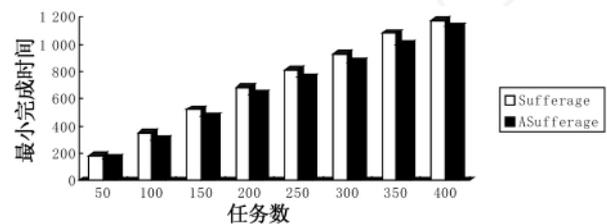


图 2 任务数量与最小完成时间图

图 3 显示了对调度效果及用户 QoS 的影响, 可以看出, 在 Sufferage 算法下随着任务数的增加, 由于竞争越来越激烈, 资源越来越小, 达不到用户 QoS 要求的任务数增加明显. 而 ASufferage 启发式算法考虑了用户的 QoS 要求, 所以随着任务数的增加, 达不到用户 QoS 要求的任务数增加平缓.

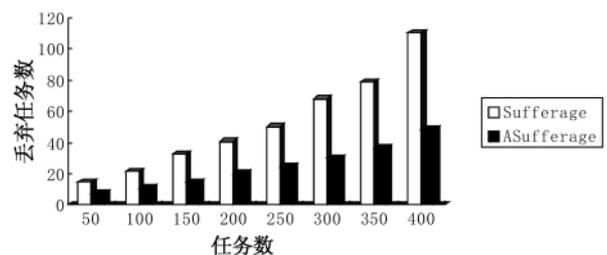


图 3 任务数量与丢弃任务数图

4 结束语

在网格环境中, 任务映射与调度算法是 NP - HARD 的, 通过分析 Petri 网模型并对其做了改进, 可以防止随网格系统参数的增加而发生空间爆炸. 并在随机高级 Petri 网模型的基础上改进了 Sufferage 算法, 改进的算法考虑了最小完成时间的同时兼顾了用户的 QoS 要求. 仿真试验结果表明, 该算法能够减小网格资源调度的最小完成时间, 随着任务数的增加, 达不到用户 QoS 要求的任务数增加平

缓改进后更适合网格实际环境 具有实际价值.

参考文献:

- [1] Foster I ,Kesselman C. The grid: blueprint for new computing infrastructure [M]. San Francisco ,CA: Morgan Kaufmann Publishers ,1999.
- [2] HE Xiao - shan ,XIAN He - sun. QoS guided Min - Min heuristic for grid task scheduling [J]. Journal of Computer and Technology. 2003 ,18(4) : 443 - 350.
- [3] R Buyya ,D Abramson ,J Giddy. Nimrod/G: an architecture for a resource management and scheduling system in a global computational grid[A]. ZHANG Xiao - xiang ,LI Guo - jie. Proceedings of the 4th International Conference and Exhibition on High Performance Computing in Asia - Pacific Region(HP - CASIA 2000) [C]. Beijing: IEEE Computer Society Press 2000. 284 - 287.
- [4] Mnaouer A B ,Riyami B A. Colored petri nets based modeling and simulation of mixed workload interaction in a non-dedicated cluste [A]. Yoshio Oyauagi. Proceedings of the 7th International Conference on High Performance Computing and Grid in Asia Pacific Region [C]. Tokyo: IEEE Computer Society Press 2004. 294 - 297.
- [5] Huang B. A new scheduling model based on extended petri net - tREM net [A]. Iverson. Proc. of the Int. Conference on Robotics and Automation [C]. San Diego ,CA: 1994. 495 - 498.

An Ameliorated Grid Resource Scheduling Algorithm Based on Random Advanced Petri Net

LI Pei - song ,LIU Jue - fu ,ZHOU Juan

(School of Information Engineering ,East China Jiaotong University ,Nanchang 330013 ,China)

Abstract: The management of resources and scheduling computations is a challenge in a grid ,because the resources are geographic distributed ,heterogeneous ,dynamic. This article uses the Random Advanced Petri Net model to describe scheduling problems and improve Sufferage Algorithm. It considers the time used to access and transfer the input/output data before/after the execution and the QoS requirements of users while pursuing the minimum completing time. The results of Ameliorate Sufferage algorithm can satisfy the QoS requirements of grid users.

Key words: random advanced petri net; sufferage algorithm; QoS

(责任编辑: 刘棉玲)