

网格环境中动态资源映射算法的研究

余建桥 廖剑伟

(西南大学计算机与信息科学学院 重庆 400715)

摘要 提出了基于任务加权相容图的网格环境中资源动态映射算法,通过在任务的执行过程不断将任务分配到相应的资源中,并动态调整加权相容图直至所有任务完成。通过仿真实验比较,该算法不论是在任务数不断变化的情况下,还是资源不断变化的情况下都能达到较好的效果。

关键词 网格计算,动态资源映射,非计算资源

Study on Dynamic Resource Mapping in Grid Computing

YU Jian-qiao LIAO Jian-wei

(Faculty of Computer and Information Science, Southwest University, Chongqing 400715, China)

Abstract This paper proposed a dynamic resource mapping algorithm based on weighted compatible graph (WCG) in grid environment. This method finishes the mapping through dynamic adjusting WCG after the completion of mapped task. Simulation shows that this method will achieve good performance not only in changing task scenario but also in changing resource scenario.

Keywords Grid computing, Dynamic resource mapping, Non-computing resource

1 引言

网格从广义上说是一个广域异构资源的有机集合体,提供透明的远程访问、资源共享、分布计算等功能。资源映射的任务是为网格应用程序中的任务分配合适的资源,并调度这些任务按照一定的顺序完成,使得资源利用率和应用程序的执行时间都达到最佳的效果。因此,对于提交到网格系统中的大量应用任务,如何快速有效地协同分配可用的资源,使得整体执行时间最短,同时提高可用资源的利用率,已成为网格在实用化过程中的一个非常核心的问题。

目前提出的资源映射方法多是在任务执行之前就进行资源的映射,即静态分配,而且大部分算法只是考虑了计算资源。实际上,在真正的网格系统中,应用任务量与资源的种类和数量是相当庞大的,资源节点的加入和退出也是相当频繁且不具有可预测性,所以资源的分配问题是一个大规模的、动态性的问题,静态分配算法不能满足其动态性要求。另外,尽管有不少的动态算法被提出,也仅仅是考虑计算资源而没有关心非计算资源。

Alhusaini 算法考虑的资源类型已扩展到数据资源、I/O 设备等,在 Alhusaini 算法^[1,2]的基础上,同时关注计算资源和非计算资源的动态资源调度,在任务执行过程中随着任务的完成而不断动态地调整任务间的资源依赖关系,直至完成整个网格应用。

2 静态分配资源及 Alhusaini 算法

2.1 任务相容图及调度资源算法

假设有 m 台计算设备 M 和 r 种非计算资源 R 一起构成网格系统,其中 $M = \{m_1, m_2, \dots, m_m\}$, $R = \{r_1, r_2, \dots, r_r\}$ 。在该网格环境中运行的网格应用程序 T , T 分解为 n 个子任务,即 $T = \{T_1, T_2, \dots, T_n\}$ 。

定义 1 用 $\text{Exec}(T_i, m_j)$ 表示机器 m_j 上完成 T_i 的计算代价,包括通信时间和计算时间。

定义 2 当任务 T_i 所需要资源与 T_j 所需的资源的交集不为空时,表示任务 T_i 和 T_j 不相容。

定义 3 给定的一组任务 $T = \{T_1, T_2, \dots, T_n\}$ 和它们的资源需求,相容图表示为 $G = (T, E)$,其中 T 是任务的集合; E 是一组边 $e(i, j)$ 的集合,每条边连接了 T 中的两个任务。

$e(i, j)$ 表现了任务 T_i 和任务 T_j 之间的数据依赖性,只有当任务 T_i 和任务 T_j 是不相容时,才存在 $e(i, j)$ 。

定义 4 加权相容图(WCG) $W = (T, E, C)$,其中 T 表示任务集; E 是一组边的集合; C 是权值集合。

定义 5 独立集合^[3]是任务相容图 G 的子集,子集中任意两个节点都是不相邻的。具有最大节点数目的独立集合称为最大独立集合。

网格应用可以分解为多个任务并表示为有向无环图(DAG),并构建任务相容图。基于该思想也产生了许多算法,如 Maheswaran 等提出动态匹配和调度资源的算法^[4]; Alhusaini 等提出了一个资源协同分配映射框架^[1,2]; Iverson 等人提出了动态竞争的调度资源算法、基于遗传算法的任务匹配^[5]、基于同步队列的资源协同分配^[6]; Yang 提出了基于策略的网格资源映射^[7]; 兼顾任务的服务质量需求,丁等采取重复映射的策略进行资源分配^[11]。

2.2 Alhusaini 算法

上述的大部分方法考虑资源仅有计算资源的情况,而 Alhusaini 等扩展地考虑资源类型,将数据、I/O 设备等其他资源也包括在待分配的资源内。Alhusaini 算法基础是在任务加权相容图的基础上构建最大独立集合进行任务资源的映射。图 1 给出任务加权相容图的例子。那么存在多个独立集合 $\{T_0, T_3\}$, $\{T_0, T_1, T_5\}$, $\{T_0, T_5, T_6\}$ 等等,如果不考虑任务权值,后面两个集合是该任务加权相容图中最大的独立集合。

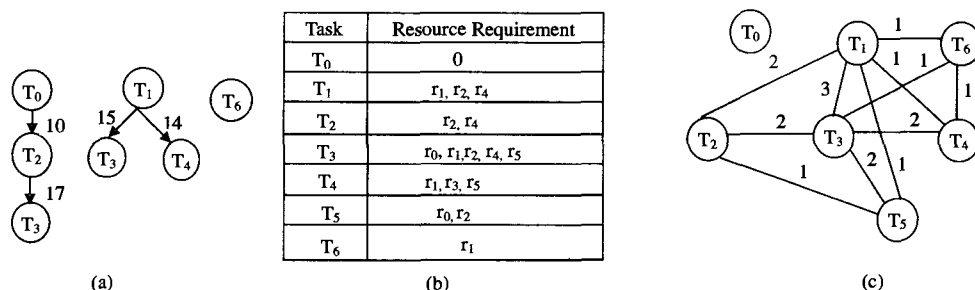


图1 (a) DAG例子, (b) 图a中任务的资源需求, (c) 加权相容图

Alhusaini 算法的基本思想是当一个任务在运行阶段不需要再使用资源 r_k 时, 则释放该资源。每个映射事件都会以下面的过程完成: 一个多个任务的集合 S 被选择执行时, 首先选择处于该集中调度顺序最优先的就绪任务开始执行, 直到 S 中的所有任务都按照调度次序逐一完成。对于每个任务 T_i , 它在映射事件发生时, 首先完成自身所需时间最短的机器 mb (称之为最优机器)。寻找一个最优机器是通过以下公式进行机器的比较, 只要条件式满足时, 那么就说明任务 T_i 可以在机器 mb 上面执行。

$$\text{Exec}(T_i, mb) \leq \text{Exec}(T_i, mj) + \Delta \text{Exec}(T_i, mj)$$

其中 Δ 是从 0% 到 100% 的值, 表示任务 T_i 在机器 mb 机器上执行的可能性。

Alhusaini 算法的特点是通过尽早释放资源, 在信息汇集阶段把有向无环图中的任务 T_i 随机分配到系统中的某台机器中。然而在网格环境中, 任务资源的变化都是不可确定的, 而且网格应用本身的任务数也有可能是不断变化的, 因此 Alhusaini 算法不能够适应网格环境的动态性要求。

3 动态资源映射算法

动态资源映射算法思想是在任务加权相容图(WCG)的基础上构建最大独立集合来完成资源映射并调度任务执行, 直至完成整个网格应用。首先, 算法根据 WCG 中生成最大独立集合, 完成最大独立集合中的所有任务的资源映射; 然后, 当这些任务执行完成以后, 为尚未进行资源映射的任务重新建构 WCG, 直至所有的任务都执行完成。

算法描述:

Begin

1. 利用已就绪任务集合READY生成加权任务相容图WCG

2. 选择整数 P , 使得 $2^P P < N$, 其中 N 为任务总数

3. While (没有完成资源映射任务集合不为空) do:

4. 调整WCG

5. 将所有就绪任务加入到集合READY

6. 将不需要非计算资源的任务加入到集合M

7. 在READY集合中选择权值最大的任务 t 加入到关键任务集合K中

8. 将与任务 t 不相容的任务从READY移出

9. While (READY中任务数 $> P$) do:

10. 根据WCG构建最大独立集合S

11. While (集合S不为空)

12. if (集合K不为空)

13. 将K集合中的权值最大的任务 t 映射到相应的资源中去

14. 将任务 t 从集合K中移除

15. else

16. 将S集合中的任务 t 映射到相应的资源中去

17. 任务 t 执行结束

18. 从集合S中移除任务 t

19. End(while)

20. End(while)

21. End(while)

22. End

图2 算法关键部分描述

图2 给出算法该阶段关键步骤的描述, 其中处于 READY 集合中的任务状态为就绪状态, 表示该任务所有的前驱任务都已经完成, 而且它所需要的所有资源都能满足。

算法中最主要的工作就是构建 WCG, 并在 WCG 的基础上选择最大的独立集合 S 。构建最大独立集合 S 是算法的关键, 具体实现如下:

当任务 T_i 不需要任何非计算资源且当它处于就绪状态时, 直接将该任务加入最大独立集合 S 中。

选择一个关键任务集 K , 该集合中的任务数是一个小于 P (P 满足 $2^P P$ 小于任务总数) 的常数。如果选择较多的关键任务, 就必须为这些关键任务预分配更多的非计算资源。因此, 与关键任务共享非计算资源的其它任务就会受到资源限制不能进入就绪状态, 则把这些任务从最大独立集合中删除。

在从 READY 集合中选择最大独立集合的时候, 必须首先选择标注了不需要任务非计算资源的任务, 这样可以有效减少从 READY 集合中构建最大独立集合的时间复杂度, 因此可以将这些任务直接加入到最大独立集合中去。

当某个任务在 READY 集合中具有最大权值, 那么将它选择加入到最大独立集合中。之所以这样做, 是因为通常在就绪状态中权值最大的任务的后继任务比较多, 而这些后继任务也有可能要使用 T_i 所需要的资源, 那么 T_i 越早完成, 就可以从就绪集合中选择更多的任务去执行, 这样可以减少构建最大独立子集的时间复杂度。

当任务 T_i 的映射事件发生以后, 根据该原则选择其它的关键任务, 直至关键任务数达到 C 。选择了关键任务以后, 还需要从 READY 集合中选择其它任务, 加入到最大独立集合 S 中。

4 时间复杂度分析与仿真

4.1 算法时间复杂度分析

该算法把任务分派到能在最短时间内完成该任务的机器, 避免了在任务数和资源数不断变化的情况下调度时间的增加。算法关键伪代码的第一步——资源映射操作是基于 WCG 完成的, 构建 WCG 的时间复杂度是 $O(N^2 R)$ [11]; 算法其它步骤的主要工作就是在 WCG 的基础上构建最大独立集合, 而该操作的时间复杂度是 $O(NR)$, 其中 N 是网格应用分解的任务数, R 表示非计算资源的数目。因此可以得出, 该算法完成资源映射的时间复杂度是 $O(N^2 R)$ 。综上所述, 整个算法的时间复杂度是 $O(N^2 R)$ 。

4.2 实验与仿真

利用 Simgrid Toolkit 开发了调度模拟器, 以更好地验证动态资源映射算法的效能。Simgrid Toolkit 提供了一系列核

(下转第 205 页)

- [2] Guo Huiping, Georganas N D. A Novel Approach to Digital Image Watermarking Based on a Generalized Secret Sharing Scheme. *Multimedia Systems*, 2003, 9(3): 249-260
- [3] Guo Huiping, Georganas N D. Jointly Verifying Ownershi Pof an Image Using Digital Watermarking. *Multimedia Tools and Applications*, 2005, 27(3): 323-349
- [4] 刘连山, 李人厚, 高琦. 视频数字水印技术综述. *计算机辅助设计与图形学学报*, 2005, 17(3): 379-386
- [5] 向辉. 图形数据数字水印技术. *系统仿真学报*, 2002, 14(12): 1649-1651
- [6] 蒋天发, 周熠, 何秉姣, 等. 基于感知数字水印对音频信息稳健性影响的研究. *武汉大学学报(工学版)*, 2004, 37(6): 93-96
- [7] Brassil J T, Low S H, Maxemchuk N F. Copyright Protection for the Electronic Distribution of Text Documents//*Proceedings of the IEEE*. 1999, 87: 1181-1196

- [8] 李媛媛, 许录平. 用于矢量地图版权保护的数字水印. *西安电子科技大学学报(自然科学版)*, 2004, 31(5): 719-723
- [9] 王勋, 林海, 鲍虎军. 一种鲁棒的矢量地图数字水印算法. *计算机辅助设计与图形学学报*, 2004, 16(10): 1377-1381
- [10] Cox I J, Miller M L, Bloom J A. 数字水印. 王颖, 黄志蓓, 译. 北京: 电子工业出版社, 2003
- [11] Makova K D, Ramsay M, Jenkins T, et al. Human DNA sequence variation in a 6.6-kb region containing the melanocortin 1 receptor promoter. *Genetics*, 2001, 158: 1253-1268
- [12] Harris E E, Hey J X. Chromosome evidence for ancient human histories. *PNAS, USA*, 2000, 96: 3320-3324
- [13] Yao Y G, Nie L, Harpending H, et al. Genetic relationshi Pof Chinese ethnic populations revealed by mtDNA sequence diversity. *Am J Phys Anthropol*, 2002, 118: 63-76

(上接第 198 页)

心函数,用以建立特定应用领域的模拟器。这些函数包括对处理机、存储等资源的描述,以及对数据传输、计算和依赖关系等任务的描述,同时提供了诸如调度、预测、时间和模拟的功能函数。通过随机生成的主机处理能力、存储资源和通信延迟可以描述实际的异构式分布式环境^[4];测试在一台酷睿双核 T2300(1.66GHz)、内存 512M 的主机上完成,每个测试结果均取 3 次模拟的平均值。

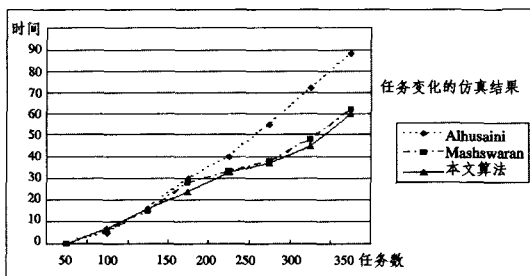


图 3 任务数变化时的仿真结果

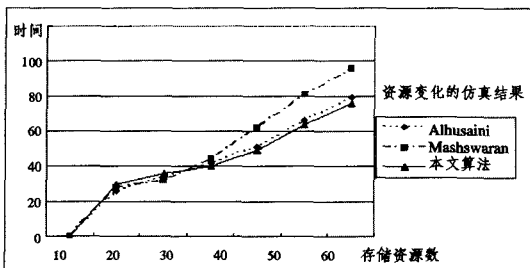


图 4 存储资源数变化时的仿真结果

图 3 显示了将变化的任务数映射到 50 个计算资源和 25 个存储资源的模拟结果。图 4 是固定任务中的任务数量为 200、计算资源为 50 个时变化存储资源数量的模拟结果,其中时间单位是以 Simgrid Toolkit 中的虚拟时间单位计量。

从仿真结果中可以看出所提出的动态资源映射算法比 Alhusaini 算法和 Maheswaran 算法总体性能更为稳定。原因是该算法在任务执行阶段完成的资源映射,更好地解决了 Alhusaini 算法在任务运行之前就进行资源映射,无法更为准确地确定资源在运行阶段是否可用,而导致的任务不能按照调度时间正常完成;另外,虽然 Maheswaran 是动态调度算法,在任务数变化的情况下可以达到较好的资源映射效果,但是该算法仅仅考虑了计算资源而没有考虑非计算资源,因此与所提出的动态资源映射算法比较,在存储资源数变化的仿真实验中明显要差。

结束语 Alhusaini 算法采用基于任务相容加权图进行资源映射。然而该算法在构建最大独立集合时并没有区分关键任务,而且所有的映射操作都是在任务执行之前完成,因此达不到实际的最佳效果。而动态资源映射算法采用最大权值的关键任务优先,而其它任务随机选取的原则相结合的方法,一方面尽可能达到资源映射的要求,另一方面也大大降低了时间复杂度和减少资源映射所需的时间。

动态资源映射算法在任务执行阶段利用加权函数进行任务的资源映射。通过仿真实验发现,与其它资源映射算法相比,它可以更有效地减少调度时间;从效率来说,算法时间复杂度是 $O(N^2R)$,相对于其它资源映射算法来说具有较低的时间复杂度。

参考文献

- [1] Alhusaini A H, Prasanna V K, Raghavendra C S. A framework for mapping with resource co-allocation in heterogeneous computing system[C]//9th Heterogeneous Computing Workshop. 2000:273-286
- [2] Alhusaini A H, Prasanna V K, Raghavendra C S. Run-Time Adaptation for grid environments[C]//Proceedings 15th International Parallel and Distributed Processing Symposium. 2001: 864-874
- [3] Christofides N. Graph Theory: an Algorithmic Approach [M]. London: Academic press, 1975
- [4] Maheswaran M, Siegel H J. A dynamic matching and scheduling algorithm for heterogeneous computing systems [A]//7th IEEE Symposium on Heterogeneous Computing Workshop(HCW'98) [C]. Orlando: IEEE Computer Society, 1998: 57-69
- [5] Casanova H. Simgrid—A toolkit for the simulation of application scheduling[C]//Proceedings of the 1st IEEE International Symposium on Cluster Computing and the Grid (CCGrid'01). 2001
- [6] Azzedin F, Maheswaran M, Arnason N. A synchronous co-allocation mechanism for grid computing systems. *Cluster Computing* [J], 2004, 7 (1): 39-49
- [7] Yang Juan, Bai Yun, Qiu Yuhui. A decentralized resource allocation policy in minigrid[J]. *Future Generation Computer Systems*, 2006
- [8] Topcuoglu H, Hariri S, Wu M Y. Performance-effective and low-complexity task scheduling for heterogeneous computing. *IEEE Trans. on Parallel and Distributed System*, 2002, 13: 260-274
- [9] Rehn C. Dynamic mapping of cooperating tasks to nodes in a distributed system [J]. *Future Generation Computer Systems*, 2006: 35-45
- [10] Wang L Z, Cai W T, Lee B S, et al. Resource co-allocation for parallel tasks in computational grids [A]//Proceedings of the International Workshop on Challenges of Large Applications in Distributed Environments [C]. Seattle: IEEE Computer Society, 2003: 88-95
- [11] 丁箭, 陈国良, 顾钧. 计算网络环境下的一个统一的资源映射策略[J]. *软件学报*, 2002, 13 (7): 1303-1308
- [12] 刘丽, 杨扬, 田志民. 网格计算环境下资源联合分配的映射策略与机制[J]. *计算机工程*, 2005, 16(31): 130-133