第7卷第1期 空 军 工 程 大 学 学 报(自然科学版) Vol.7 No.1
2006年2月 JOURNAL OF AIR FORCE ENGINEERING UNIVERSITY(NATURAL SCIENCE EDITION) Feb.2006

支持资源协同分配的网格任务调度研究

刘 颖, 余侃民, 江胜荣
(空军工程大学 电讯工程学院, 陕西 西安 710077)

摘 要:提出通用的网格和任务执行模型,并以此为基础,给出一种支持资源协同分配的任务调度算法。算法通过定义临界资源的概念,改进了传统的列表调度算法。模拟实验结果表明该调度策略更符合网格计算的复杂环境,能得到较短的任务执行时间,并更好的支持不同类型资源的协同分配。

关键词:网格;资源协同分配;任务调度;列表调度
中图分类号: TP393 **文献标识码**:A **文章编号**:1009-3516(2006)01-0080-04

网格技术的出现使得在多个虚拟组织之间进行大规模的资源共享和合作成为可能[1]。网格资源协同分配就是在满足网格应用对资源的具体性能需求的同时为应用分配多个不同类型资源[2]。实际上也就是把一个应用任务集合映射到不同类型的资源上,其目标是使应用集合的整体执行时间最小。比如,在大规模并行计算应用中,执行计算任务需要同时访问多个高性能计算机,进行数据传输需访问网络元素,在交互的数据分析应用中,数据的复制需要同时访问多个存储系统,进行数据的分析与结果交互显示还要访问高性能计算机、网络元素以及显示设备。可以看出,对于网格应用,协同分配资源是很有必要的。

当应用为单一任务,且资源仅包括计算资源时,已有许多学者进行研究并提出了解决方法[3~4]。但当应用本身包含多个任务,且每个任务都要访问不同类型的资源(计算资源、通信资源、储存资源等)时,针对单一任务的解决方案就不适用了。对于多任务模式可以将其表示为有向无环图(directed acyclic graph, DAG),基于这个思想也出现了许多算法,如 Maheswaranm 等人提出的动态匹配算法[5],Iverson 等人提出的动态竞争调度资源算法[6],基于遗传算法的任务匹配[7],基于同步队列的资源协同分配[8]等。然而这些方法均考虑的是资源中仅有计算资源的情况,而 Alhusaini 等人扩展了资源类型,将其他资源如数据资源、I/O设备等也包括在待分配的资源内,提出了一个资源协同分配映射框架[9]。丁箐等人考虑任务的服务质量需求,采取重复映射的策略进行资源分配[10]。这些算法考虑多种资源的分配,但都是针对串行任务,没有考虑问题的并行性并且大多没考虑系统网络的互连拓扑结构和通信对调度的影响。实际上,在真正的网格系统中,应用任务量和资源的数量是相当庞大的,所以资源的分配问题也是一个大规模的、实时性的问题,已有的这些方法均不能满足其应用要求。列表调度[3]是一类重要的调度算法,但传统的列表调度已经不适应复杂的网格环境,不能对资源的协同分配提供很好的支持。本文对传统列表调度算法进行改进,提出一种支持协同分配的网格并行任务调度算法 SARC,通过仿真实验与传统的列表调度算法比较可以看出,该算法可以获得较短的执行时间并有效的支持资源协同分配,有较好的执行性能。

1 问题描述

1.1 系统模型

假设网格系统由 m 个资源组成,这些资源由任意的互连网络相连接。网格系统模型定义如下:Grid System $M=(E,R,B)$

收稿日期:2005-06-22
基金项目:陕西省自然科学基金资助项目(2004F14)
作者简介:刘 颖(1980-),女,江苏淮安人,博士生,主要从事网格资源管理与调度研究.

1)$E = \{E_1, E_2, \cdots, E_k\}$,网格中资源类型的集合,$k$ 是资源类型总数。

2)$R = \{R_1, R_2, \ldots, R_m\}$,网格中资源的集合,$m$ 是资源总数,每个资源定义两种属性:$R_i \cdot Type$ 和 $R_i \cdot$ Cap,分别为资源类型和资源负载量。

3)$\boldsymbol{B} = [B_{ij}]$,网络带宽矩阵,$B_{ij}$是资源 R_i 和 R_j 之间的网络带宽,$1 \leqslant i、j \leqslant m$。

1.2　任务模型

定义网格环境中并行任务执行模型 $G = (T, \boldsymbol{A}, <, D)$

1)$T = \{T_1, T_2, \cdots, T_n\}$,$T$ 是并行子任务的集合,n 是子任务的总数。

2)$\boldsymbol{A} = [A_{ij}]$,$\boldsymbol{A}$ 是 $n \times k$ 的矩阵,A_{ij}表示任务 T_i对类型为 E_j 的资源的需求关系,$1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant k$, $T_i \in T$。

3) < 是一个偏序,表示任务执行的优先约束。例如, $T_i < T_j$ 意思指 T_i 必须在 T_j 开始执行之前完成。$1 \leqslant j \leqslant n$, $T_i \in T$, $T_j \in T$。

4)$\boldsymbol{D} = [D_{ij}]$,$\boldsymbol{D}$ 是一个 $n \times n$ 维的通信数据矩阵,D_{ij}是任务 T_i 和任务 T_j 之间传输的数据量,$D_{ij} \geqslant 0$,$1 \leqslant i, j \leqslant n$, $T_i \in T$, $T_j \in T$。

1.3　资源协同分配模型

资源协同分配矩阵定义为:$\boldsymbol{S} = [S_{ij}] = [< O_{ij}, V_{ij} >]$。

O_{ij}是分配给 A_{ij}的资源,如果 $A_i \neq \varphi$、$1 \leqslant i \leqslant n$、$1 \leqslant j \leqslant k$,$Q_{ij} \in R$;如果 $A_i = \varphi$、$1 \leqslant i \leqslant n$、$1 \leqslant j \leqslant k$、$Q_{ij} = \phi$。

V_{ij}是任务 T_i 在资源 O_{ij}上的开始执行时间,$V_{ij} \in [0, \infty)$。

为了容易阐述问题,再给出如下定义:

1)TST——任务开始执行时间

任务 T_i 可以同步运行在多种资源$\{R_{i1}, R_{i2}, \cdots\}$上,这些资源最早可用时间是$\{Sta_tmi1, Sta_tmi2, \cdots\}$,则任务 T_i 开始执行时间 TST_i 定义为: $\mathrm{TST}_i = \max\{Sta\_tmi1, Sta\_tmi2, \cdots\}$,并行任务 T 的开始执行时间定义为:$\mathrm{TST} = \min\{\mathrm{TST}_1, \mathrm{TST}_2, \cdots, \mathrm{TST}_n\}$。

2)TFT——任务完成时间

任务 T_i 可以同步运行在多种资源$\{R_{i1}, R_{i2}, \cdots\}$上,在这些资源上的完成时间是$\{Fin_tmi1, Fin_tmi2, \cdots\}$,则任务 T_i 完成时间 TFT_i 定义为:$\mathrm{TFT}_i = \max\{Fin\_tmi1, Fin\_tmi2, \cdots\}$,并行任务任务完成时间 TST 定义为 $\mathrm{TFT} = \max\{\mathrm{TFT}_1, \mathrm{TFT}_2, \cdots, \mathrm{TFT}_n\}$。

3)TET——任务执行时间

假设任务 T_i 对资源类型为 E_j 的资源需求为 A_{ij},分配给 T_i 的资源为 R_l,TET_{il}是任务 T_i 在资源 R_l 上的执行时间,$T_i \in T$, $E_j \in E$, $R_l \in R$。

给定并行任务 $G = (T, \boldsymbol{A}, <, \boldsymbol{D})$和 Grid System $\boldsymbol{M} = (E, R, \boldsymbol{B})$,确定并行任务 T 到资源协同分配矩阵 $\boldsymbol{S}$ 的映射 f,$f: T \rightarrow S$, $f \in F$,F 是实际执行的映射集。算法的目标是确定函数 $f_{\min} \in F$, $f_{\min}$为并行任务最短执行时间。

给出任务调设算法——SARC。

fta - tm(T_i) :*starting tine for fask* $T_i \in T$

ready_list:list of the task that are ready to be executed

Priority(T_i) :priority of task $T_i \in T$

CR(T_i) :critical resource of task $T_i \in T$

```
BEGIN
Initialize ready_list
FOR each task T_i ∈ T ,DO
   compute blevel(T_i) using Algorithm 4
ENDFOR
REPEAT
      FOR each task T_i ∈ ready_list ,DO
         determine CR(T_i) and Sta_tm(T_i) using Algorithm1
         Priority(T_i) ←Sta_tm(T_i) - blevel(T_i)
```

ENDFOR

select the task T_{i_select} with T_i, $\overset{max}{ready\_list}$ Priority(T_i)

schedule task T_{i_select} using Algorithm 2.

update ready_list

update surfaces of resources allocated

UNTIL all tasks are scheduled

END

2 仿真实验

为了评估算法的性能,我们进行了一系列的模拟研究。执行时间是衡量调度执行性能的重要指标,也是面向性能优化的网格调度目标所在。一个调度算法能使任务的执行时间更短,那么当任务截止期缩短时,这个算法可以提供更高的调度成功率,也意味着可以提供更加高效的调度。而在任务截止期与任务调度结束之间存在的空闲时间,可以作为系数调整实际运行时间和估算时间之间的不一致,从而提高实际调度的成功率。因此,本研究中着重比较调度结果的执行时间 TET。

模拟实验采用 SimGrid[11] 开发网格调度模拟器。SimGrid 是由美国加州大学圣地亚哥分校网格研究和创新实验室(Grid Research And Innovation Laboratory)主导开发的,它的目标是为在网格环境下进行分布并行应用调度研究提供一个合适的模型和抽象(level of abstraction)并生成准确的模拟结果。

通过 SimGrid 提供的、随机生成的主机处理能力,以及网络带宽和通信延迟可以描述实际网格的计算环境;通过随机生成的数据传输量和计算量,并设置依赖关系,可以描述前面定义的任务模型和资源分配模型。为了比较 SARC 算法的性能,同时选取 HEFT 算法[12] 和 DLS 算法[13] 进行模拟实验。测试采用 SG-2.18.2 版本,在一台 Petium4 1.7 GHz 的主机上完成,每个测试均取三次模拟结果的平均值,并以 SimGrid 中的虚拟时间单位计量。当协同分配的资源数量 R 固定为 30 时,令其中计算资源、通信资源和储存资源的相对权重为 3、1、2,随着子任务数目的增加,TET 的变化曲线如图 1 所示。图 2 是并行任务中的任务数量固定为 50 时,资源数量增加时的模拟结果,其中计算资源、通信资源和储存资源的权重比例为 2、1、2。对比可见 SARC 算法较另两种算法在降低 TET 方面有明显优势且曲线平滑。

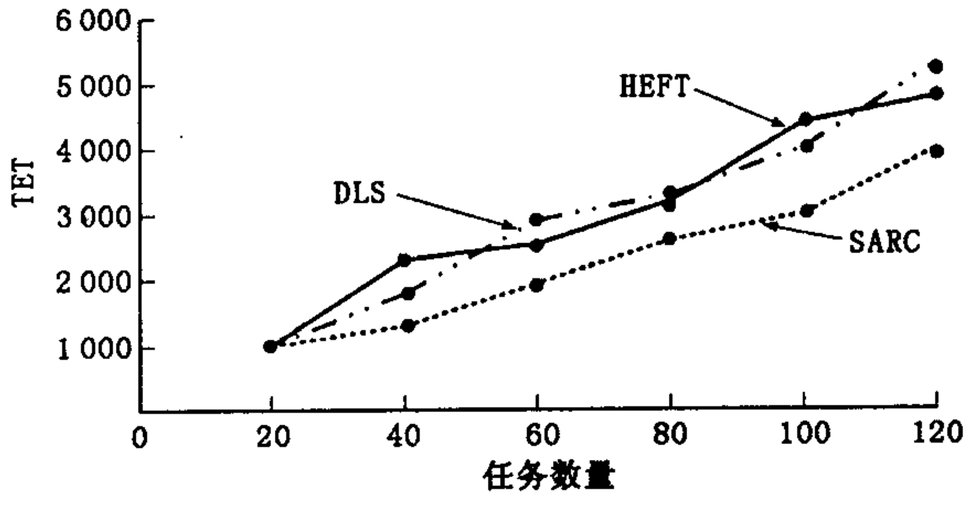

图 1 任务数量对 TET 的影响图

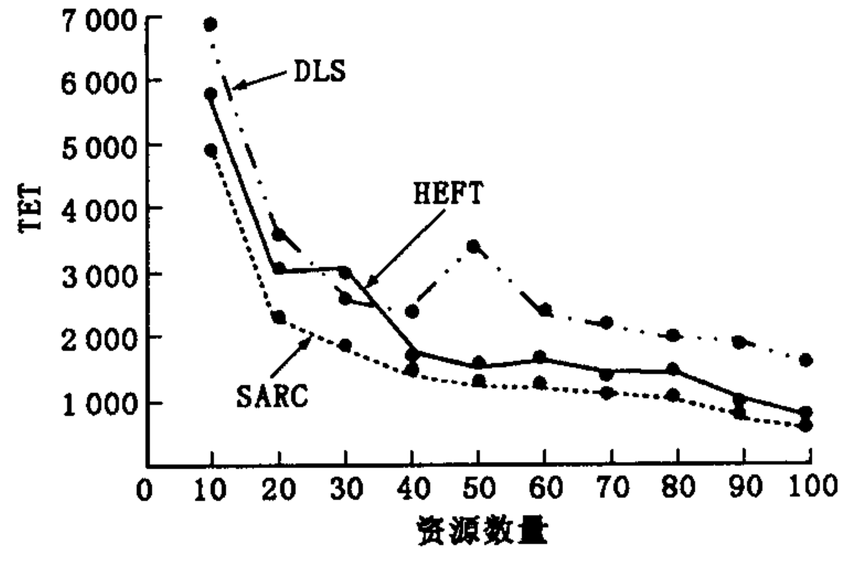

图 2 资源数量对 TET 的影响图

3 结束语

对网格环境中的资源协同分配的研究目前还处于探索阶段,本文在深入分析网格应用环境的基础上,提出了一种网格系统模型和任务执行模型。并以此为基础,提出了一个面向性能优化的网格任务调度算法 SARC,解决网格中的资源协同分配问题,该算法通过提出一个新的概念——临界资源来确定任务的最早执行时间,能够尽可能快地调度拥有最高优先级的任务,缩短任务执行时间,提高资源的利用率。模拟结果表明,SARC 算法对需要大量异构资源的网格任务可以有效地降低执行时间并支持资源的协同分配。本算法假设资源和任务都是事先已知的,即为静态的,下一步的工作是解决动态环境中资源协同分配的问题。

参考文献:

[1] Foster I, Kesselman C. The Grid: Blueprint for a Future Computing Infrastructure[M]. Morgan Kaufmann Publishers, 1999.

[2] Czajkowski K, Foster I, Karonis N, et al. Resource Management Architecture for MetacoMputing Systems[A]. the 4th Workshop on Job Scheduling Strategies for Parallel Processing[C]. Springer Verlag LNCS 1459, 1998. 62 - 82.

[3] Sih H J, Lee E A. A Compile - Time Scheduling Heuristic for Interconnection Constrained Heterogeneous Processor Architectures[J]. IEEE Transactions on Parallel and Distributed Systems, 1993, 4(2): 75 - 87.

[4] Braunt, Siegelhj, Beckn, et al. A Taxonomy Fordescribing Matching and Scheduling Heuristics for Mixed - Machines Heterogeneous Computing Systems[A]. 7thIEEE. Symposiumon Reliable Distributed Systems[C]. WestLafayette: IEEEComputerSociety, 1998. 330 - 335.

[5] Maheswaranm, Siegelhj. A Dynamic Matching and Scheduling Algorithm for Heterogeneous Computing Systems [A]. 7thIEEE Symposiumon Heterogeneous Computing Workshop(HCW'98)[C]. Orlando: IEEEComputerSociety, 1998. 57 - 69.

[6] Iversonm, Ozgunerf. Dynamic Ccompetitive Sscheduling of Mmultiple DAGs in a Ddistributed Hheterogeneous Eenvironment [A]. 7th IEEE. Symposiumon Heterogeneous Computing Workshop(HCW'98)[C]. Orlando: IEEEComputerSociety, 1998. 70 - 78.

[7] Ssanyals, Jaina, Dassk, et al. A Hhierarchical and Ddistributed Aapproach for Mmapping Llarge ASapplications to Hheterogeneous Ggrids Uusing Ggenetic Aalgorithms[A]. Proceedings of the IEEE International Conference on Cluster Computing (CLUSTER' 03)[C]. HongKong: IEEE Computer Society, 2003. 496 - 499.

[8] Azzedinf, Maheswaranm, Arnasonn. Asynchronous Co - Allocation Mechanism for Grid Computing Systems[J]. ClusterComput, 2004, 7(1): 39 - 49.

[9] Alhusainiah, Prasannavk, Raghavendracs. A Frame Work for Mapping With Resource Co - Allocation in Heterogeneous Computing Systems[A]. 9th Proceedings of Heterogeneous Computing Workshop (HCW2000)[C]. Cancun: IEEEComputerSociety, 2000. 273 - 286.

[10] 丁　箐,陈国良,顾　钧. 计算网格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.

[11] Casanova H. SimGrid - a Toolkit for the Simulation of Application Scheduling[A]. Proceedings of the 1st IEEE International Symposium on Cluster Computing and the Grid (CCGrid01)[C]. 2001. 430 - 437.

[12] Topcuoglu H, Hariri S, Wu M Y. Performance - Effective and Low - Complexity Task Scheduling for Heterogeneous Computing[J]. IEEE Trans. on Parallel and Distributed Systems, 2002, 13(3): 260 - 274.

[13] Sih G C, Lee E A. A Compile - Time Scheduling Heuristic for Interconnection - Constrained Heterogeneous Processor Architectures[J]. IEEE Trans. on Parallel and Distributed Systems, 1993, 4(2): 175 - 186.

(编辑:门向生)

Research on Grid Task Scheduling for Resource Co - allocation

LIU Ying, YU Kan - min, JIANG Sheng - rong

(The Telecommunication Engineering Institute, Air Force Engineering University, Xi'an, Shaanxi 710077, China)

Abstract: This paper presents models for computational Grids and parallel tasks running in Grid computing environments. Based on these models, a parallel task - scheduling algorithm for resource co - allocation in Grid computing environments is proposed. The algorithm is effective in identifying the critical resource of parallel tasks and scheduling a task as early as possible when all the resources required by the task are available. Simulation results show that shorter execution time and better performance can be gained by utilizing this algorithm, especially in Grid computing Environments.

Key words: grid; resources co - allocation; task scheduling; list scheduling