异构机群下最小化非实时任务平均响应时间的调度策略

吴代贤 杨 娟 邱玉辉

(西南大学计算机与信息科学学院 重庆 400715)

摘 要 机群作为高性能计算机平台有着广阔的应用前景。由于很多情况下单一机群已经无法满足应用需求,因此分离的机群通常相互连接来建立一个多机群计算结构。分布式系统中根据系统处理任务性质的不同可分为实时任务和非实时任务两类。它们继而又可以分为有数据关联任务和无数据关联任务。本文关注的是异构机群下非实时任务流的调度,采用最小化任务平均响应时间作为目标函数。已有的策略要么不适用于异构多机群,要么没有考虑到任务传递的通信耗费,或者两者都欠缺。本文提出的任务调度策略解决了这两个问题,并且通过试验证明了 MMRT 虽然充分考虑了通信耗费,却没有明显增大系统的额外开销。

关键词 多机群,异构系统,任务调度,非实时任务,通信耗费

A Job Scheduling Policy Used to Minimize the Mean Response Time of the un-Real Time Jobs in the Multi- Heterogeneous Clusters

WU Dai-Xian YANG Juan QIU Yu-Hui

(The Faculty of Computer and Information Science, SWU, Chongqing 400715)

Abstract Clusters are more and more popular being used as the high performance computing platform. Clusters need to connected together to form a much stronger integrated computing system to fulfill the request of the applications. The jobs which are processed by the system are divided into two main kinds, refered as real-time jobs and un-real-time jobs, and each can be further divided into data related jobs and un-data related jobs. This paper address on processing of the un-real-time jobs without data relationships. And made the goal function as the minimizing the mean response time. Some of the existed polices are not suited in the multi-heterogeneous-clusters, others are lack of communication cost considering, or both. This paper propose a job scheduling policy to solve those problem without increasing the additional system cost.

Keywords Multi-clusters, Heterogeneous systems, Job scheduling, Un-real time jobs, Communication cost

1 引言及研究背景

机群作为高性能计算机平台有着广阔的应用前景。由于很多情况下单一机群已经无法满足应用需求,因此分离的机群通常相互连接来建立一个多机群计算结构[1]。机群中的资源分配作为提高机群性能的关键途径受到广大学者的关注。这里所指的资源除了指传统的处理器之外,还指带宽,存储等其他指标。针对带宽的资源分配策略主要指将有限带宽合理分配给任务,或任务流,最终实现目标函数最优的结果。如文[2]中,处理的是包交换网络中将带宽资源合理分配给任务流。文[3]中,对网络中的任务实行边(通信链路)调度。但是大部分任务调度策略中都只是将通信耗费作为一个影响因素考虑在任务调度中。

分布式系统中根据系统处理任务性质的不同可分为实时任务和非实时任务两类。实时任务通常指任务有时间 QoS 限制,如限定了完成时间的 DAG 任务流和实时系统中的实时任务流。而非实时任务则是指无时间限制的任务流,它们继而又可以分为有数据关联任务和无数据关联任务。数据关联任务指任务的执行由其前驱任务的数据所驱动,可描述为 DAG 任务流的处理,并且目标函数可用最小化整体任务响应时间来衡量[4~6,7]。而无数据关联的任务则是指任务以一定频率到达系统,且任务间相对独立。无数据关联的任务调度的目标函数通常用最小化任务平均响应时间[8~10]和均衡任务负载[11]两个指标来构建。

文[8]提出了一个静态任务分配技术,目标是在异构机群中最小化非实时任务流的任务平均响应时间。但它假设机群中为单计算节点。文[12]在异构多处理器系统中用响应时间、吞吐量和系统时间等指标对任务分配策略进行衡量。文[11]考虑在异构多处理器环境下分布式进行非实时任务流的任务均衡调度,但是分布式的 GDA 计算依赖的信息必须是准确的全局调度信息,否则很难获得稳定状态。文[10]是一个在异构多处理器系统中对非实时任务进行调度的策略,且目标函数为最小化平均响应时间。但是 MMP 没有考虑任务的通信耗费问题。文[9]提出了 OMRT 算法,在异构多机群环境中对非实时任务进行最小化任务平均响应时间的调度。同文[10]一样,它也没有考虑任务传递的通信耗费问题。

本文提出了一个在多机群结构下对非实时任务流进行最小化平均响应时间的调度策略。我们选择任务的平均响应时间来作为任务调度的参考指标。系统处理的任务类型是任务间无关联的非实时任务流。在构建的由多个异构机群组成(即机群间的计算能力各不相同)的网络模型中考虑了任务的通信耗费参数。而机群内部则由同构处理器构成(同个机群中的处理器计算能力相同)。随后我们给出了一个启发式的求解算法 MMRT(),它通过迭代的形式求解我们所构建的线性规划问题 min *imize*(R)。并且最后通过试验证明 MMRT 虽然充分考虑了通信耗费,但是却没有明显地增大任务调度

吴代贤 博士生,讲师;邱玉辉 教授,博士生导师,主要从事人工智能及计算机网络方面的研究。

的开销。

2 网络环境模型构建

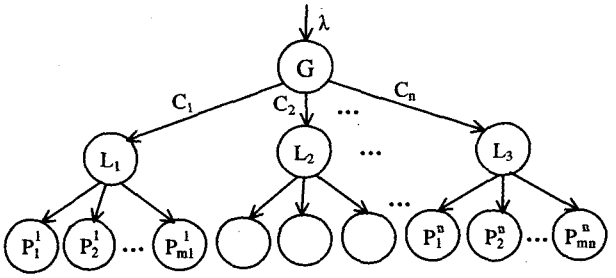

图 1 n 机群网络环境图

用一个六元组来描述异构多机群网络环境 $N=\langle G,L,P,E,C,U\rangle$，表示网络 N 由一个全局调度器 G 和 n 个本地调度器 L_i 组成，如图 1 所示。则 L 为本地调度器集合 $L=\{L_i \mid i\in[1,n]\}$。P 为处理器集合，有 $P=\{p_j^i \mid i\in[1,n],j\in[1,m_i]\}$，其中 i 表示机群 i，m_i 表示机群 i 中的处理器个数。假设 n 个机群为异构机群，即机群间处理能力各不相同，而机群内部处理器为同构处理器，则用 u_i 表示机群 i 中每个处理器的处理能力(机群 i 中每个处理器处理单个任务的时间倒数，可看作计算频率)。p_j^i 为当前机群 i 中的第 j 个处理器。$E=\{e_i \mid i\in[1,n]\}$为全局调度器 G 与各本地调度器 L_i 之间的通信链路。因为各机群在地域上各异。因此假设各通信链路 e_i 上的通信能力各不相同。令 $C=\{c_i \mid i\in[1,n]\}$表示链路 e_i 传递单位任务的时间的倒数(可看作通信频率，注意，这里假设 L_i 与 p_j^i 之间无通信耗费，这是基于同机群处理器在地域上可能都是邻近的这个事实)。$U=\{u_i \mid i\in[1,n]\}$，表示机群 i 中每个处理器的处理能力。假设 n 个机群为异构机群，即机群间处理能力各不相同，而机群内部处理器为同构处理器，则用 u_i 表示机群 i 的处理能力(机群 i 中处理器 p_j^i 处理单个任务的时间倒数，可看作 p_j^i 的计算频率)。则 m_iu_i 为机群 i 的计算能力。设 G 与 L_i 之间的通信为顺序通信，即 N 为单端口模型。且 L_i 按 c_i 的非递增顺序排列，即通信能力强的机群优先调度，这样可以避免通信瓶颈问题。

3 问题构建

假设系统处理的任务类型为非实时任务，即任务无时间 QoS 限制。则令目标函数为最小化任务平均响应时间。设 R 为任务平均响应时间，则有目标函数为 $\min imize(R)$。若任务到达频率为 λ，则希望求得一种任务分布$(\alpha_1\lambda,\alpha_2\lambda,\cdots,\alpha_n\lambda)$满足目标函数。$\alpha_i$ 为各个机群上所分配任务的比例。任务的响应时间定义为任务到达本地调度器与任务完成时间的时间间隔[9]。若 R_i 表示机群 i 上单个任务的平均响应时间，则有：

$$R=\frac{\sum_1^n R_i}{n} \tag{1}$$

由文[9]可知

$$R_i=w_i+(1/u_i) \tag{2}$$

w_i 表示机群 i 上任务的平均等待时间，$1/u_i$ 表示机群 i 上单个任务的执行时间。若$\frac{\alpha_i\cdot\lambda}{c_i}>\frac{1}{u_i}$，即 e_i 的通信能力强于机群 i 的处理能力，则任务到达时需要等待，且等待时间为

$$w_i=\frac{\sum_{j=1}^{\lceil\frac{\alpha_i\cdot\lambda\cdot u_i}{c_i}\rceil-1}(j/u_i)}{\lceil\frac{\alpha_i\cdot\lambda\cdot u_i}{c_i}\rceil} \tag{3}$$

(3)式指当新任务到达时机群 i 最坏情况为 i 上还有 km_i 个任务等待执行，则它至少需要等待 $k\cdot(1/u_i)$个时间片刻，其中 $k\in[1,\lceil\frac{\alpha_i\cdot\lambda\cdot u_i}{c_i}\rceil-1]$。若$\frac{\alpha_i\cdot\lambda}{c_i}\leqslant\frac{1}{u_i}$，则任务到达时无需等待，即 $w_i=0$，令$\frac{\alpha_i\cdot\lambda}{c_i}>\frac{1}{u_i}$的概率为 P_i，则一般情况下有

$$w_i=\frac{\sum_{j=1}^{\lceil\frac{\alpha_i\cdot\lambda\cdot u_i}{c_i}\rceil-1}(j/u_i)}{\lceil\frac{\alpha_i\cdot\lambda\cdot u_i}{c_i}\rceil}\cdot P_i \tag{4}$$

假设 T_i 为分配到机群 i 上的任务量，则

$$T_i=\alpha_i\cdot\lambda \tag{5}$$

表 1 机群 i 上的时刻定义表

$ST(T_i)$：表示从 G 传递任务量 T_i 到 L_i 的开始时刻
$FT(T_i)$：表示从 G 传递任务量 T_i 到 L_i 的结束时刻
$SE(T_i)$：表示任务量 T_i 在 L_i 上执行开始的时刻
$FE(T_i)$：表示任务量 T_i 在 L_i 上执行完成的时刻

定义如表 1 中的时刻关系：

其中

$$ST(T_i)=FT(T_{i-1}) \tag{6}$$

即 T_i 开始传输的时刻为 T_{i-1}结束传输的时刻。这是基于网络通信模型为单端口的假设，即不同 L_i 的任务传递必须是顺序的。

$$FT(T_i)=ST(T_i)+T_i/c_i \tag{7}$$

T_i 结束传输的时刻定义为开始时刻加上传输耗费的时间段。

代(6)入(7)式得：

$$FT(T_i)=f'[FT(T_1)]+T_i/c_i \tag{8}$$

即 $FT(T_i)$是一个关于 $FT(T_1)$的函数。令

$$ST(T_1)=0 \tag{9}$$

代(9)入(7)式得：$FT(T_1)=T_1/c_1$ (10)

代(10)入(8)式得：

$$FT(T_i)=f'(T_1/c_1)+T_i/c_i \tag{11}$$

$f'(T_1/c_1)$函数为将 $FT(T_i)$与 T_1/c_1 关联起来的映射函数，即 $FT(T_i)$可用 T_1/c_1 表示。因为机群 i 上任务量 T_i 的总共的执行耗费时间可用任务平均响应时间与任务量的乘积来表示，即 $R_i=\frac{FE(T_i)-SE(T_i)}{T_i}$ (12)

而机群 i 上任务执行的开始时刻为

$$SE(T_i)=ST(T_i)+1/c_i \tag{13}$$

因为机群 i 上任务的执行开始时刻与任务传递开始的时刻之间仅有一个任务的传递时间差。代(9)入(13)式得：

$$SE(T_1)=1/c_1 \tag{14}$$

同理，所有任务的执行时间与整体任务量之比就应该为整个系统的平均响应时间，即

$$R=\frac{FE(T_n)-SE(T_1)}{\lambda} \tag{15}$$

代(12)入(15)式得：

$$T_n\cdot R_n+SE(T_n)-SE(T_1)=\lambda\cdot R \tag{16}$$

代(13)(6)(8)(5)代入(16)式可得 α_1 与 α_i 之间的关联表达式，可用(17)式表示：

(下转封三)

结论 等价性检验的主要目的是在一个设计经过变换之后，穷尽地检验变换前后功能的一致性，即证明设计的变换没有产生功能的变换。比如逻辑最小化后与原电路等价，对电路结构作局部修改后需证明与原电路等价，从较高级别的描述综合为较低级别的描述，也需要证明其所实现的电路与原来的描述等价，例如行为级到寄存器传输级，寄存器传输级到门级，或者时钟树的插入、扫描链的重排序、FPGA 到 ASIC 的转换等。所以等价性检验也不只是单纯地用于组合电路的等价检验，也可用于时序电路的等价性检验。

在本文中，我们分析了基于 BDD 的组合电路的结构等价性验证方法。值的注意的是，这种方法对输入变量的排序是极其敏感的，因此容易发生状态爆炸的情况。通常在实际运用中我们都是基于 ROBDD 进行等价性验证的，并且对于有较多部分相同的电路，系统可以首先进行同构比较，发现并排除完全同构的部分，仅对不同构的部分进行验证。系统也可以对用户指定的部分电路进行验证。

参 考 文 献

1 Bryant R. Graph-Based Algorithms for Boolean Function Manipulation. IEEE Transactions on Computers, August 1986. 677～691

2 Bryant R. Symbolic Boolean Manipulation with Ordered Binary Decision Diagrams. ACM Computing Surveys, 1992, 24(3):293～318

3 Andersen H R. An introduction to Binary Decision Diagrams. HRA100497, Technical University of Denmark, Lyngby. 4. Christoph Kern and Mark R. Greenstreet. Formal Verification In Hardware Design: A Survey. ACM Transactions on Design Automation of Electronic Systems, 1999, 4(2):123～193

4 Brace K S, Rudell R L, Brayant R E. Efficient Implementation of a BDD Package. In: 27th ACM/IEEE Design Automata Conference, 1990

5 韩俊刚，杜慧敏. 数字硬件的形式化验证. 北京：北京大学出版社，2001

6 Kuehlman A, Krohm F. Equivalence Checking Using Cuts and Heaps. In: Proc. of DAC, 1997. 263～268

7 Malik S, Wang A R, Brayton R K, Vincentelli A S. Logic Verification using Binary Decision Diagrams in a Logic Synthesis Environment. In: Proc. of ICCAD, 1988. 6～9

（上接第 255 页）

$$\alpha_i = f^a(\alpha_1) \quad (17)$$

另外有限制条件 $\sum_{i=1}^{n}\alpha_i = 1$ (18)

在知道 R 和 R_i 的前提下代(17)入(18)可求出 α_i 的值。

4 最小化任务平均响应时间算法 MMRT( )(Minimize Mean Response Time)

由上述构建的问题限制条件和等式可得出线性规划问题 min *imizg*(*R*)的启发式迭代搜索算法 MMRT()。算法描述如下：

MMRT(){

Begin：令 α_i 初始值为 $\alpha_i = \frac{m_i \cdot u_i}{\sum_{i=1}^{n} m_i \cdot u_i}$，即 α_i 初值为基于权重的分配。

1：若 $\frac{\alpha_i \cdot \lambda}{c_i} > \frac{1}{u_i}$，则 $P_i = 1$，否则 $P_i = 0$

2：代 α_i，P_i 值入(4)式，求得 w_i

3：代 w_i 入(2)式求得 R_i，代入(1)式继而求得 R

4：若 R 已收敛，则结束调度，否则继续

5：代求出的 R 和 R_i 值入(17)式，求得新 α_i 的值，转入执行步骤 2}

算法 MMRT()是一个反复迭代的过程，迭代的结束条件是目标函数值的收敛，因此会存在一些震荡，如图 2 所示。

5 模拟试验

我们将 MMRT()算法的调度性能与文[9]中的 OMRT 算法进行比较，如图 2 所示，可看出 MMRT()虽然考虑了通信耗费因素，却并没有明显地增大额外开销，即调度时间并未显著增加。而 MMRT 与 OMRT 的曲线有些震荡，这是由于迭代使得算法复杂度波动的原因，因为 MMRT 的算法复杂度为 $O(n \cdot k)$，k 为不定的迭代次数。

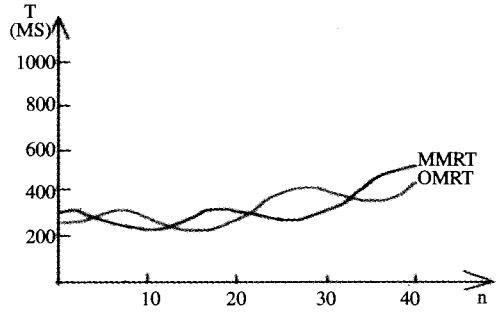

图 2 MMRT 与 OMRT 策略调度时间地比图

结论 本文提出了一个在多机群结构下对非实时任务流进行最小化平均响应时间的调度策略。我们在构建的网络模型中充分考虑了任务的通信耗费，并且通过一个启发式的求解算法 MMRT()来求解我们所构建的线性规划问题 min *imize*(*R*)。最后通过试验证明 MMRT 虽然充分考虑了通信耗费，但是却没有明显地增加任务调度的开销。在以后的工作中我们将考虑在异构多机群系统中对实时或软实时任务进行考虑通信耗费的任务调度。

参 考 文 献

1 Barreto M, Avila R, et al. The MultiCluster Model to the Integrated Use of Multiple Workstation Clusters. In: Proc. Third Workshop Personal Computer-Based Networks of Workstations, 2000. 71～80

2 Hou Y T, Panwar S S, et al. On Generalized Max-Min Rate Allocation and Distributed Convergence Algorithm for Packet Networks. IEEE Transaction on Parallel and Distributed Systems, 2004, 15(5): 401～416

3 Sinnen O, Sousa L A. Communication Contention in Task Scheduling. IEEE Transaction on Parallel and Distributed Systems, 2005, 16(6): 503～515

4 Kafil M, Ahmad I. Optimal Task Assignment in heterogeneous Distributed Computing Systems. IEEE Concurrency on Complex Distributed Systems, 1998, 6(3):42～51

5 Ranaweera S, Agrawal D P. A Task Duplication Based Scheduling Algorithm for Heterogeneous Systems. In: Proceedings of the 16th International Parallel and Distributed Processing Symposium, Florida: IEEE Computer Society Press, 2002. 445～450

6 Bansal S, Kumar P, et al. An Improved Duplication Strategy for Scheduling Precedence Constrained Graphs in Multiprocessor Systems. IEEE Transaction on Parallel and Distributed Systems, 2003, 14(6):533～544

7 Bajaj R, Agrawal D P. Improving Scheduling of Tasks in a Heterogeneous Environment. IEEE Transaction on Parallel and Distributed Systems, 2004, 15(2):107～118

8 Tang X Y, Chanson S T. Optimizing Static Job Scheduling in a Network of Heterogeneous Computers. In: Proc. 29th Int'l Conf. Parallel Processing, 2000. 373～382

9 He L, Jarvis S A, et al. Allocating Non-Real-Time andSoft Real-Time Jobs in ulticlusters. IEEE Transaction on Parallel and Distributed Systems, 2006, 17(2):99～112

10 Georgiadis L, Nikolaou C, et al. A fair workload allocation policy for heterogeneous systems. Journal of Parallel and Distributed Computing, 2004, 64(1):507～519

11 Rotaru T, Nageli H H. Dynamic load balancing by diffusion in heterogeneous systems. Journal of Parallel and Distributed Computing, 2004, 64(4):481～497

12 Chow Y C, Kohler W H. Models for Dynamic Load Balancing in Heterogeneous Multiple Processor System. IEEE Trans. Computers, 1979, 28(5):354～361