第 6 卷第 7 期
1995 年 7 月

软 件 学 报
JOURNAL OF SOFTWARE

Vol. 6, No. 7
Jul. 1995

基于任务图的一种并行程序设计方法(Ⅱ)
——选择拓扑结构*

吴巧泉 沈 平 张德富

(南京大学计算机科学系,南京 210093)

摘要 本文探讨了基于任务图的并行程序设计方法的后半部分——按照任务图选择拓扑结构和将并行算法映射到并行结构上.

关键词 任务图,并行程序设计方法,映射.

文献[1]提出一种基于任务图的并行程序设计方法.该方法首先分析欲解的问题,产生数据流程图,并以此设计出表示并行算法的任务图;然后根据任务图选择合适的系统拓扑结构,把算法映射到结构上并使之匹配;最后编程,调试.文献[1]已对此方法中的前半部分进行了讨论.本文重点探讨它的后半部分.

任务图设计完成后,如何为并行算法选择一个适合的拓扑结构是很重要的.本文提出一种根据并行算法特征来选择适合的系统拓扑结构(简称为重构)的方法.

我们讨论的可重构并行计算机是可以通过硬件开关或软件指令来重新构造系统的拓扑结构.它们有如下几点假定:

(1)系统由一定数目的同构处理机组成.

(2)系统可调整成共享内存,或私有内存,或二者兼而有之.

(3)系统可分为多个独立的不同大小的子系统.对于单个算法的执行,本假设只意味可选择处理机数来适合算法需要;而对于由多个算法组成的一个作业的执行,本假设意味着这几个算法可分别在各个子系统上同时执行.

(4)系统及各个子系统都能执行 SIMD 或 MIMD 操作,并能在此二种模式间动态转换.

(5)系统及各个子系统都有一可调整的互连网络,能提供各种或一定数目的通信结构.

实际上,以上假定的可重系统是现有各种可重构系统的一个集合,除了异构并行计算机和数据流机没有包括在内,其余的基本上都包含了.

一旦算法和结构确定了,余下的问题就是把算法映射到结构上.实际的可重构系统往往只是本文假设的可重构系统的一个子集,不同的可重构系统其映射算法不同.本文针对

* 本文 1993-05-24 收到,1994-02-01 定稿

本研究得到国家 863 高技术计划资助.作者吴巧泉,1968 年生,工程师,主要研究领域为并行处理与电力自动化.沈平,1957 年生,工程师,主要研究领域为计算机网络与并行处理.张德富,1937 年生,教授,主要研究领域为计算机软件,并行处理与分布式计算.

本文通讯联系人:张德富,南京 210093,南京大学计算机科学系

Transputer 可重构系统提出一种重构和映射的策略.我们曾按照这种策略完成了一些并行程序的设计,说明此策略是可行、有效的.

1 并行算法并行计算机结构之间的关系

并行算法与并行计算机结构的关系远比串行算法与串行计算机结构的关系密切而又复杂.并行算法直接依赖于并行计算机结构,不同类型的并行计算机结构,在其上所使用的并行算法也不相同,并且所欲解决的应用问题也不完全相同.并行算法的结构只有与并行计算机结构匹配,系统才能高效运行.因此人们非常重视研究并行算法与并行计算机结构的关系,并在此基础上探讨将并行算法映射和划分到并行计算机结构上,以便引导人们科学、系统地研制基于可重构系统的并行程序设计方法,而不是象现在按设计者的经验,人为地进行并行程序设计.可是到目前为止还没有找到并行算法与并行计算机结构之间的完全对应关系,只有一些不完全的信息供有关研究者参考.文献[2]提出了一种根据并行算法的某些重要特征选择适合的并行计算机结构的总的指导思想.

为了更好地挖掘并行计算机的潜力,必须深入了解并行算法和并行计算机结构之间的关系,哪些是并行算法的主要特征,哪些是并行计算机结构的主要特征.

在分析一系列已有的并行算法的基础上,表1总结了并行算法特征与并行计算机结构特征之间的关系.表中"1"表明两个特征之间很可能具有强烈的依赖关系;"2"表明次之;"3"更次之.还有其他的依赖关系,但对于大部分算法,这些已能刻划算法与结构的关系.表中的模式指 SIMD 或 MIMD 或流水线.

表1 并行算法特征与并行计算机结构特征之间的关系

依赖关系	处理机数	存贮组织	内存容量	模式	网络	同步	处理机能力	数据类型	寻址模式	数据结构	I/O
并行性类型	3	2	3	1		3					
数据粒度		1	2	1	3					2	
任务粒度		1		1	1	1					
并行度	1	2		2							
内存	2	2	1								
一致性		2		1	3						
同步		2		1	2	1					
数据依赖关系		2		3	1						
基本操作		2			2		1				
数据类型					3			1	2		1
数据结构		2		3	2				3	1	
I/O		3	3	2	2						1

2 根据算法特征选择合适的系统结构

表1表示的算法特征可以从任务图的任务流程图中分析得到.下面逐条分析算法特征所适合的并行计算机结构,如存贮器组织、需要的处理机数、操作模式和网络构造等.

2.1 并行特性

并行特性与并行性类型、算法和数据的划分有关,它有以下3个特征.

(1)数据并行性/功能并行性

在任务图中,若并行执行的各个任务功能相同,只是输入数据不同,那么这个算法的并行性类型为数据并行性;若并行执行的各个任务功能不同,那么该算法的并行性类型为功能并行性.并行性类型影响数据的分布,进程到处理机的分配和采用的并行模式(有SIMD/MIMD/流水线3种).功能并行性常采用MIMD模式,而数据并行性常采用SIMD或流水线模式.对于数据并行性,所需的处理机数与数据集大小有一定的比例关系,通常处理机数较大;而基于功能并行性的算法则常采用以十计而非以千计的处理机数.

(2)数据粒度

数据粒度作为基本单元处理的数据项大小,从任务图及任务流程图中可看出其大小.它与数据分布、通信要求、处理机能力和存贮规模有关.细数据粒度算法常适合用具有私有内存的SIMD或流水线模式来实现.数据粒度对总的内存容量不作要求,但对每个处理机的内存容量有要求.另外,数据粒度还指明一个数据项通信所需的带宽.

(3)任务粒度

任务粒度是可单独或并行执行的任务大小.它是决定同步频率的一个重要因素,并且影响到并行模式的选择、任务到处理机的分配、存贮器组织、通信要求以及负载均衡等.细粒度任务算法需要频繁的同步,如果可能,最好采用SIMD或流水线模式、私有内存组织和高速互连网络结构.粗粒度任务算法,由于计算量较大,通信量较小,一般建议采用MIMD模式.

2.2 并行度

并行度即为可并行执行的任务数,从任务图上即可看出,它与数据粒度和任务粒度有关.它影响到并行计算机规模的选择以及可获得的最大加速.此外,并行度还与并行模式和内存组织有关.

2.3 操作的一致性

从任务流程图的分析中可以知道操作是否一致,如果执行的操作是一致的,则采用SIMD或流水线模式比较合适.一致性常与数据并行性有关,它有几个层次,取决于操作对象的粒度,有的高层不一致,而低层却一致;有的却正好相反.如果执行的操作不一致,则采用MIMD较合适,并采用使各处理机的负载尽量均衡的策略.

2.4 同步要求

除任务粒度所隐含的同步要求外,任务优先约束也包含着同步要求.任务中的每次通信都可能需要一次同步.同步要求将影响进程到处理机的分配以及算法各个部分的调度.

2.5 数据依赖关系

算法的数据依赖关系体现在任务图中任务的通信关系.它对数据分配模式和通信结构起决定作用,并且对决定采用私有内存还是共享内存组织也有很大作用.数据依赖关系的处理实际上是一个图同构问题,因为算法的数据依赖关系和系统的通信拓扑结构都可以用有向图来表示.

对于某一算法的数据依赖图(即为任务通信图),要确定其适合的通信结构来加以映射是不容易的,但现有的关于算法到结构映射的经验可以加以利用.方法是把许多这样的经验

组合成一个映射库,库中包含已知的数据依赖图及其对应的系统拓扑结构和两者的映射关系(可能有多个).映射过程的第一步是确定问题的大小,因为有些算法,如对一个 3×3 窗口绕时邻接点的操作,其数据依赖图独立于问题大小,计算任意一个输出象素所需要的数据与输入图象的大小无关.对于这样的依赖图,库中可用基本的数据依赖模式来表示.而另一类算法,如 FFT,其数据依赖图就与问题大小有关:一个 16 点 FFT 算法需要 16 个输入值来计算一个输出值;而一个 1024 点 FFT,需要 1024 个输入值,其数据依赖图是规则的.但对于一个具体算法所需的通信结构则与问题大小有关.对于这一类依赖图.库中就不能用一个简单的模式来表示,而用数据依赖图的推导规则来表示.映射过程的第二步是处理通信拓扑结构,把算法数据依赖图中的结点映射到系统通信拓扑结构图中的结点,这一步是通过把算法的数据依赖图与库中的数据依赖图搜索匹配得到结果.这个方法的具体算法如下:

(1)确定算法的数据依赖图是与问题的大小有关还是与大小无关.

(2)如果算法的数据依赖图是与问题的大小有关的,则确定输入问题的大小 S.

(3)对库中每一个数据依赖图进行下列处理:

(i)如果该数据依赖图是与大小有关的,则根据库中对应于该数据依赖图的推导规则产生一个大小为 S 的模板图.

(ii)比较该数据依赖图与算法的数据依赖图是否同构.

(4)一旦找到一个同构图,比如说为图 P,则采用库中存贮的对应于 P 的映射信息,该信息可能包含算法结点(任务)对结构结点(处理机)的适当分配和问题大小与结构大小一致的处理策略等.

(5)若没找到同构图,则采用图转换技术或最近似的匹配技术,求出近似的同构图.

刻划数据依赖关系是解决算法映射到结构问题的一个重要部分,上述方法较有效.

2.6 基本操作

算法的基本操作从任务流程图中可得出,它决定所需的处理机能力,在某种程度上也决定了一些通信要求.它对网络和存储器组织也有影响.

2.7 数据类型和精确度

数据类型和数据精确度与单个处理器能力和对内存组织要求直接有关,同时也隐含对通信带宽的要求.

2.8 数据结构

许多算法有一个自然的数据结构.如果一个系统结构,能支持算法的存取模式,以挖掘数据结构的规则性,并能允许数据结构各部分之间相互作用,则算法在此系统结构上运行效率较高.

最后须对以上根据算法特征选择的结构加以综合分析,得出一个较合适的系统结构.

3 任务图到结构的映射

上一段的讨论实质上是选择一系统拓扑结构,余下的问题就是讨论如何把算法映射到一定的结构,对于这个问题,已经提出很多策略,但这些策略只是对某些情况比较适合,目前要提出通用、高效的解决方法还很困难,有些策略虽对任何情况都能应用,但效率不高.本文将提出一个较方便的方法,该方法设有一个图论库和一组理论性算法及策略,根据所处理

的问题的算法,采用不同的映射与划分策略.

首先判断一下算法的通信结构,如果是常见的通信结构(例如树、超立方体等)或者是已有研究的通信结构,则只要从库中取出划分和映射模式,库中事先存放有关这一类经验证明很有效的划分映射模式.这一步具体算法在2.5节已描述.

如果不是以上类型的通信结构,则判断其是否是特殊类型的结构,如果是,则采用针对该类型结构设计特殊映射算法.如果再不是,则可采用某些通用的算法,如文献[3]描述的算法,或采用某些启发式算法.

具体实现时,输入的是任务图和系统的拓扑结构图,及其他有用信息,输出的是两者的映射关系.首先由于我们设计出的任务图可能具有循环关系,这与某些映射算法的条件不符,因而先要将它转化为非循环图(理论上,如一个任务图包含循环,那么它一定可以转化为非循环图,见文献[4]);其次,由于我们设计的任务图具有并行和串行两类不同性质的通信,所以得修改某些算法,在对任务图划分时,尽量把具有串行关系的任务(具有串行通信关系,以"→"表示)分在一簇.

4 实 现

4.1 实现环境(与文献[1]相同)

4.2 基于NCD-1系统的重构和映射策略

前面我们的讨论是基于一种理想的可重构系统,实际的可重构系统只是其中的一个子集.比如NCD-1可重构系统,除了通信拓扑结构外,其余均不可调整,况且因为每个Transputer只有4条通信链路,所以并非各种通信结构都构造.针对这种特殊情况,本文提出一种重构和映射策略.

对于NCD-1系统的重构,我们只根据算法的数据依赖关系.它表现于任务图中即为任务之间的通信关系.所以本文的重构和映射策略,输入的是任务图和Transputer数,输出的是Transputer间的通拓扑结构和任务图对此结构的映射.本策略的目标是使各处理机(Transputer)负载均衡和每个处理机的4条链路充分利用,其策略的具体步骤如下:

(1)检查Transputer数(设为P个)和任务数(设为t个),若$t>p$,则把任务图中的任务分成P簇,尽量使得各簇的执行时间接近,并尽可能把关系密切通信量大的任务分为一簇.若$t\ll p$,则一个任务一簇.

(2)依次检查每簇,如其与其它簇通信线超过4根,则对通信量小的通们线加屏蔽标志,直到剩下4根通信线为止.

(3)根据分簇屏蔽后的任务图,构造NCD-1的通信拓扑结构:对于每一簇分配一个Transputer,并根据簇与簇之间未被屏蔽的通信关系,连接Transputer结点的间通信链路.

(4)把那些标志屏蔽的通信线映射到Transputer结点链路上.

4.3 实 例

下面以解线性代数方程组(扩充高斯消去法)为例作简要说明.

·问题描述:

线性代数方程组$AX=y$,式中A为常数矩阵,X为解向量,y为常数向量.

·算法设计:

对于一外 Transputer，任务图如图 1，由于该任务较简单，所以不需要细化，对于 2 个 Transputer，类似于上例，扩展了其数据并行性，任务图如图 2，对于 4 个、8 个 Transputer 也作类似的进一步细化，算法运算结果如表 2.

图1 解线性方程组初始任务图

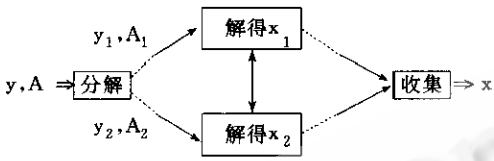

图2 解线性方程组第二级任务图

表 2 用扩充高斯消去法解方程组的运算结果(104 元线性方程组)

算法	ESEQGAUSS	PEGAUSS2	PEGAUSS4	PEGAUSS8
处理机数	1	2	4	8
S	1	1.98	3.92	7.50
E	1	0.99	0.98	0.93
T(秒)	16.5	8.3	4.2	2.2

参考文献

1 张德富，吴巧泉. 基于任务图的一种并行程序设计方法(Ⅰ)——任务图的设计. 软件学报，1995，6(4)：379－384.

2 Jamieson L H. Characterizing parallel algorithms. The MIT Press，1987. 65－100.

3 Chaudhary V，Aggarwal J K. Generalized mapping of parallel algorithms onto parallel architectures. Proceeding of the 1990 International Conference on Parallel Processing，1990(2)：137－141.

4 Bare J L. A survey of some theoretical aspects multiprocessing. ACM Computing Surveys，1983. 31－80.

A METHOD OF PARALLEL PROGRAM DESIGN BASED ON TASK GRAPH(Ⅱ)——CHOOSE TOPOLOGICAL STRUCTURE

Wu Qiaoquan Shen Ping Zhang Defu

(*Department of Computer Science, Nanjing University, Nanjing* 210093)

Abstract This paper has presented a method of parallel program design based on task graph and discussed design of task graph. In this paper, the authors develop letter half of the method for choosing topological structure by task graph and mapping parallel algorithm to parallel architecture.

Key words Task graph, method of parallel program design, mapping.