

程序自动并行化过程中的任务划分^{*}

冯百明, 丁晓宁, 肖 骊, 况正谦, 康继昌

(西北工业大学 计算机科学与工程系, 陕西 西安 710072)

摘 要:任务划分是程序并行化过程中的重要一环,它的优劣直接影响并行程序的正确性。本文以基于数组元素局部相关的计算为基础,讨论了将串行程序所描述的任务划分为多个子任务的方法,提出了编译程序静态识别与并行程序动态判断相结合的方法,成功地实现了对多个串行程序的正确并行化,该方法降低了对串行程序书写格式的要求,提高了程序自动并行化的程度。

关 键 词:串行程序; 并行程序; 任务划分; 分布式计算

中图分类号: TP302 文献标识码: A 文章编号: 1000-2758(2000)02-0298-04

近年来计算机工作者在自动并行化的基础理论和实用方法两个方面进行了大量的研究工作^[1],并出现了一些实际系统 Tiny、Polaris、PFA、AFT 等,这些系统大都面向并行计算机系统。程序自动并行化技术目前还很不完善^[2]。由于网络技术的发展和 PVM、EXPRESS、MPI 等分布并行计算环境的完善,人们开始关注网络环境下的并行计算。针对已广泛使用的 PVM 环境,作者进行了从源程序到源程序的自动并行化研究,已实现了一个用于银河—Ⅲ 计算机上的程序自动并行化工具 NPUPAR,已交付用户使用,达到了用户的要求。NPUPAR 面向流场计算类问题,该类计算问题的特点是对多维网格点进行迭代求解,程序中表现为对多个多维数组的迭代计算。计算数组中一个元素的值,仅与该元素近邻空间的某些元素有关(称这种相关特性为局部相关),其中作用在网格上的 DO 循环语句,可不分析其数据相关性而看作 DOALL 循环^[3,4]。NPUPAR 适用于并行计算机系统,也适用于网络环境。

任务划分是实现程序并行化的基础,任务划分策略的好坏直接影响到负载平衡性、通信复杂度、任务间的依赖性以及任务间的同步方式和同步频繁程度等。将串行程序描述的任务划分成多个可并行运行的子任务是实现并行化编译的关键。本文针对基于数组分块并行计算的程序并行化中碰到的任务划

分问题,在数据分布^[5,6]的基础上讨论将串行源程序描述的任务划分为多个子任务的方法。

1 串行任务的并行划分

1.1 定义和说明

NPUPAR 中的任务划分遵从“拥有者写”原则,并行化编译时,对数组写访问作为划分任务的依据,按照该数组的分布方式,将计算任务划分到该数组元素分布到的相应节点机上,在计算的过程中,如果用到本节点机上不存在的元素,要在计算之前通信,把所需要的元素从其它节点机上取过来,这样,每个节点机在计算过程中对数组的访问总是局部的。

定义 1 如果一个参数是常数、常量或者是通过扫描源程序可以得到其值(数值)的变量,该参数是静态可知的。

定义 2 串行程序描述的任务为总任务,简称为任务。分配到各个节点机上的任务称之为子任务,每一个子任务都有一个唯一识别号 ME_i 。

定义 3 串行程序中的数组 $A(*, N_i, *)$ 按 N_i 所在维分布到 P_i 个节点机上的子数组定义为 $A(*, \text{part}_i, *)$, part_i 由分布策略确定,一般 $\text{part}_i = \text{INT}(ME_i/P_i) + 1$ 。

■ 收稿日期: 1998-06-26

作者簡介: 冯百明(1966—),男,甘肃省武威市人,西北工业大学博士,现西北师范大学讲师,主要从事并行/分布计算及自动并行化技术等的研究。

并行程序采用 SPMD 程序模式,多维划分的情况下,为了使用方便,各任务除了 ME 之外,还有一个与 ME 等效的多元组 $(ME_1, ME_2, \dots, ME_n)$ 用来标记 ME 的各维坐标,维数的关系同 C 语言中多维数组的定义。数组分布的策略采用分块策略^[7],例如,把数组 $A(N_1, N_2)$ 分布到 $P(P = P_1 \times P_2)$ 个节点机上,在第一维下标上把 A 分为 P_1 块,在第二维下标上把 A 分为 P_2 块, $BLOCK_{ij}$ 块的范围为 $A(LIMIT(i): LIMIT(i + 1) - 1, LIMIT(j): LIMIT(j + 1) - 1)$,其中的 $LIMIT(*)$ 是根据分配策略构造的函数, $LIMIT(i)$ 表示第 i 块与第 $i + 1$ 块之间的界限。第 i 块的相应下标范围为半闭半开域 $[LIMIT(i), LIMIT(i + 1))$ 。并行程序中的数组名同原串行程序中的数组名是相同的,只是其大小成了原串行程序的 $1/P$ 。

1.2 在循环体中的任务划分

循环是科学计算程序中大量使用的语句,串行程序中的并行资源大都用循环语句描述^[8],已经有许多关于循环语句的并行化的研究成果^[4],讨论的对象都是型如式(1)的循环。

$$\begin{aligned} &DO\ I = 1, N \\ &\dots \\ &ENDDO \end{aligned} \tag{1}$$

研究中发现,有不少循环的步长不是 1,初值也不是 1,这些循环不容易或者不可能变换为式(1)所示的形式,尽管这种形式的循环所占的比例不大,但它们是程序并行化中不可避免的问题。作者将讨论 $DO\ I = init, term, step$ 形式的循环,式(1)的循环也包括在讨论之内,是一种特殊的形式。为了问题的一般性和普遍性,以多重循环中的其中一层循环式(2)来讨论,这样,可将问题方便地推广到多维的情况。 DO 语句中各参数的下标 i 表示该 DO 语句在多重循环中所在的深度。

$$\begin{aligned} &DO \dots \\ &\quad DO\ I_i = init_i, term_i, step_i \\ &\quad \quad DO \dots \\ &\quad \quad \quad A(*, I_i, *) = \dots \\ &\quad \quad ENDDO \\ &\quad ENDDO \\ &ENDDO \end{aligned}$$

现分三种情况讨论。

1.2.1 循环范围静态可知,循环步长为正的情形

将该循环所作用的数组维划分为 P_i 块, $ME_i (0 \leq i < P_i)$ 为任务的识别号,可直接确定该层循环

可划分为哪些子任务,各 ME_i 的范围。用 $LIMIT(t) \leq init_i < LIMIT(t + 1)$ 确定 $init_i$ 所在任务号 t 。得到 $step_i = 1$ 时各任务的循环初值函数(3)式。

$$begin(ME_i) = \begin{cases} part_i + 1 & ME_i < t \\ init - LIMIT(t) & ME_i = t \\ 1 & ME_i > t \end{cases} \tag{3}$$

如果循环步长 $step_i > 1$,则有

$$begin(ME_i) = \begin{cases} part_i + 1 & ME_i < t \\ init_i - LIMIT(t) & ME_i = t \\ step_i + ((init_i - LIMIT(t)) \bmod step_i) & ME_i > t \end{cases} \tag{3'}$$

用 $LIMIT(t) \leq term_i < LIMIT(t + 1)$ 确定 $term_i$ 所在的任务号 t 。得到循环终值函数

$$end(ME_i) = \begin{cases} part_i & ME_i < t \\ term_i - LIMIT(t) & ME_i = t \\ 0 & ME_i > t \end{cases} \tag{4}$$

1.2.2 循环范围静态可知,循环步长为负的情形

任务号的确定同 1.2.1 的内容, $|step_i| = 1$ 时各任务的循环初值函数

$$begin(ME_i) = \begin{cases} part_i & ME_i < t \\ init - LIMIT(t) & ME_i = t \\ 0 & ME_i > t \end{cases} \tag{5}$$

如果循环步长 $|step_i| > 1$,则有

$$begin(ME_i) = \begin{cases} LIMIT(i + 1) - ((LIMIT(i + 1) - init_i) \bmod |step_i|) & ME_i < t \\ init_i - LIMIT(t) & ME_i = t \\ 0 & ME_i > t \end{cases} \tag{5'}$$

确定了 $term_i$ 所在的任务号 t ,可得到各任务的循环终值函数

$$end(ME_i) = \begin{cases} part_i + 1 & ME_i < t \\ term_i - LIMIT(t) & ME_i = t \\ 1 & ME_i > t \end{cases} \tag{6}$$

1.2.3 循环范围静态不可知的情形

由于循环范围静态不可知,所以作者采用由编译程序扫描判断,生成合适的程序源代码插入到目标程序中去的方法,具体的判断到程序运行的过程中由并行程序去判断,以便确定是否该执行本循环、循环范围。

1.3 在循环体之外的任务划分

1.3.1 数组下标为常数或常量

数组下标为常数或常量的数组元素在串行程序

中的书写形式一般为 $A(*,n,*)$, 其中的 n 可以是常数、常量以及由常量和常数构成的表达式, 所以下标 n 的值从串行程序中是静态可知的, 计算该数组元素的任务号是静态确定的。采用如下的不等式确定任务号 t , 在并行程序中还要实现下标平移。

$LIMIT(t) \leq n < LIMIT(t+1)$, 并行程序中的新下标为 $n' = n - LIMIT(t)$

一条语句 $A(*,n,*) = \dots$ 并行化后得到的结果为

```
IF (ME.EQ. t) THEN
  A(*,n',*) = ...
ENDIF
```

1.3.2 数组下标静态不可知

数组 $A(*,V,*)$ 的下标 V 静态不可知的情况有 V 是一个变量或者是包含有变量的表达式。编译程序无法确定 V 的值, 因此要向并行程序中插入源代码。如:

$$A(*,V,*) = \dots$$

V 为第 i 个下标。在并行程序中转换为如下的 IF 控制语句, 其中 $V - LIMIT(ME)$ 为平移下标的计算表达式。

```
IF ((LIMIT(ME).LE.V).AND.(V.LE.
LIMIT(ME+1)))THEN
  A(*,V-LIMIT(ME),*) = ...
ENDIF
```

2 实际算例及并行化结果

下面给出了一个实际算例的并行化结果, 由于串行程序长度超过了 3000 行, 限于篇幅作者这里只选择一些有代表意义的语句段。该问题的数组为 $U(99,41,13)$, 划分方式为二维划分, 第一维下标上被分为 4 块, 第二维下标不划分, 第三维下标上被分为 3 块, 整个任务被分成 12 个子任务。MEI 的取值为 0,1,2,3, MEK 的取值为 0,1,2。

串行程序:

```
I=ID-1
DO J=2,JD-2
  DO K=2,KD-2
    U(I,J,K)=...
  END DO
END DO
```

并行化结果

```
IF(MEI.EQ.3)THEN
```

```
I=23
DO J=2,JD-2
  IF(MEK.EQ.0)THEN
    KBEGIN=2
  ELSE
    KBEGIN=1
  ENDIF
  IF (MEK.EQ.2)THEN
    KEND=2
  ELSE
    KEND=KTERM
  ENDIF
  DO K=KBEGIN,KEND
    U(I,J,K)=...
  END DO
END DO
ENDIF
```

采用该划分方法, 不但能够保证并行程序的正确性, 而且有利于提高并行程序的性能。经实际算例验证, 在 10M 以太网网络环境下, 双机并行效率达到了 87.3%, 四机和八机效率分别达到了 83.5% 和 80.1%。另外, 利用该方法可分解大规模计算问题, 使单机无法计算的程序由多个节点机共同计算。对于单机无法计算的问题, 实际测算的结果是: 四机对双机的相对效率为 97.7%, 八机对双机的相对效率为 96.1%。

3 结 论

本文中提出的划分方法首次采用了编译程序与目标程序相结合的策略, 减少了对输入程序的变换, 为最终实现完全自动并行化编译做了有效的尝试。该方法充分发挥编译程序的潜力, 尽可能地把能由编译程序完成的处理由编译程序完成, 同时结合同并行程序的能力, 把一些静态不可知的判断变换为源语句插入并行程序中, 由目标程序实现动态判断。使用该方法能把符合要求的串行程序并行化, 得到高质量的并行程序源代码。其正确性和效果已经得到了验证与串行源程序提供者的肯定。

致谢 研究过程中, 得到了国防科技大学分布式与并行处理国家重点实验室的协助, 提供了实验设备与测算环境和部分计算实例, 尤其得到了王正华主任、赵文涛博士后、杨晓辉博士的帮助, 在此表示感谢。

参考文献:

[1] Eigenmann R, Hoeflinger J, Padua D D. On the Automatic Parallelization of the Perfect Benchmarks. IEEE Transaction on Parallel and Distributed System, 1998, (1):5~22

[2] 朱传琪,臧斌宇,陈 彤. 程序自动并行化系统. 软件学报,1996,(5): 180~185

[3] Ramaswamy S, Sapatneker S, Banerjee P. A Framework for Exploiting Task and Data Parallelism on Distributed Memory Multicomputers. IEEE Transaction on Parallel and Distributed System, 1997, (11): 1098~1114

[4] Chen D, Yew P. On Effective Execution of Nonuniform DOACROSS Loops. IEEE Transaction on Parallel and Distributed System, 1996(5): 463~476

[5] Gupta M, Banerjee P. Demonstration of Automatic Data Partitioning Techniques for Parallelizing Compilers on Multicomputers. IEEE Transaction on Parallel and Distributed System, 1992, (2): 179~193

[6] Lee D Z. Efficient Algorithms for Data Distribution on Distributed Memory Parallel Computers. IEEE Transaction on Parallel and Distributed System, 1997, (8):825~838

[7] 郭柯榕,唐新春. 基于多层循环并行化的负载平衡优化. 国防科技大学学报, 1997, (10):40~45

[8] Yang Y, Jin C, Zhang X. Adaptively Scheduling Parallel Loops in Distributed Shared-Memory Systems. IEEE Transaction on Parallel and Distributed System, 1997(1): 70~81

On Task Partitioning in Automatic Parallelization

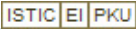
Feng Baiming, Ding Xiaoning, Xiao Li, Kuang Zhengqian, Kang Jichang

(Department of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an 710072)

Abstract: Task partitioning has great influence on the performance of the resulting parallel programs in automatic parallelization. We developed a software system——NPUPAR, for converting serial programs to parallel ones in CFD. In this paper, we present the method for task partitioning in NPUPAR. In section 1, we described loop partitioning and non-loop partitioning in detail. We proposed a method to take advantage of both static analysis and dynamic decision when loop domain or array subscripts is uncertain. Such a method greatly extended the scope of serial programs to be converted. An example is given in section 2. It shows that our method for task partitioning is effective.

Key words: task partitioning, parallel program, automatic parallelization

(编辑: 蔺西亚)

作者: [冯百明](#), [丁晓宁](#), [肖骊](#), [况正谦](#), [康继昌](#), [Feng Baiming](#), [Ding Xiaoning](#),
[Xiao Li](#), [Kuang Zhengqian](#), [Kang Jichang](#)
作者单位: [西北工业大学, 计算机科学与工程系, 陕西, 西安, 710072](#)
刊名: [西北工业大学学报](#) 
英文刊名: [JOURNAL OF NORTHWESTERN POLYTECHNICAL UNIVERSITY](#)
年, 卷(期): 2000, 18(2)

参考文献(8条)

1. [Eigenmann R;Hoeftlinger J;Padua D D](#) [On the Automatic Parallelization of the Perfect Benchmarks](#)[外文期刊] 1998(01)
2. [朱传琪;臧斌宇;陈彤](#) [程序自动并行化系统](#) 1996(05)
3. [Ramaswamy S;Sapatnekar S;Banerjee P A](#) [A Framework for Exploiting Task and Data Parallelism on Distributed Memory Multicomputers](#)[外文期刊] 1997(11)
4. [Chen D;Yew P](#) [On Effective Execution of Nonuniform DOACROSS Loops](#)[外文期刊] 1996(05)
5. [Gupta M;Banerjee P](#) [Demonstration of Automatic Data Partitioning Techniques for Parallelizing Compilers on Multicomputers](#)[外文期刊] 1992(02)
6. [Lee D Z](#) [Efficient Algorithms for Data Distribution on Distributed Memory Parallel Computers](#)[外文期刊] 1997(08)
7. [郭柯榕;唐新春](#) [基于多层循环并行化的负载平衡优化](#)[期刊论文]-[国防科技大学学报](#) 1997(10)
8. [Yang Y;Jin C;Zhang X](#) [Adaptively Scheduling Parallel Loops in Distributed Shared-Memory Systems](#)[外文期刊] 1997(01)

本文读者也读过(9条)

1. [丁丁](#), [罗四维](#), [高瞻](#), [DING Ding](#), [LUO Si-wei](#), [GAO Zhan](#) [一种基于任务划分的启发式网格调度策略](#)[期刊论文]-[北京交通大学学报\(自然科学版\)](#) 2007, 31(5)
2. [龚心宇](#), [于勤](#), [李晓明](#), [陈道蓄](#), [谢立](#) [JAPS中的任务划分与粒度控制](#)[期刊论文]-[小型微型计算机系统](#) 2002, 23(2)
3. [邓彬](#), [王宏力](#), [邓方林](#) [基于复杂连续系统并行仿真的自动任务划分](#)[期刊论文]-[计算机工程与设计](#) 2000, 21(2)
4. [李俊平](#), [薛海燕](#) [关系规范化理论及非规范化设计在数据库中的运用](#)[期刊论文]-[计算机与数字工程](#) 2005, 33(1)
5. [沈勤华](#), [SHEN Qin-hua](#) [可扩展的自动并行化编译系统](#)[期刊论文]-[计算机工程](#) 2009, 35(8)
6. [张春明](#), [孙豁然](#), [王恩德](#), [赵志刚](#), [范铁生](#), [Zhang Chunming](#), [Sun Huoran](#), [WANG Ende](#), [Zhao Zhigang](#), [Fan Tiesheng](#) [矿产资源信息系统空间属性数据建模方法](#)[期刊论文]-[金属矿山](#) 2005(6)
7. [姚放吾](#), [卢昭材](#), [YAO Fang-wu](#), [LU Zhao-cai](#) [基于权重可变免疫算法的动态可重构任务划分](#)[期刊论文]-[计算机技术与发展](#) 2009, 19(7)
8. [孙昌言](#) [BC范式的判定定理及关系的规范化方法](#)[期刊论文]-[同济大学学报\(自然科学版\)](#) 2002, 30(6)
9. [江文毅](#) [串行程序并行化的探讨](#)[会议论文]-1999

引用本文格式: [冯百明](#), [丁晓宁](#), [肖骊](#), [况正谦](#), [康继昌](#), [Feng Baiming](#), [Ding Xiaoning](#), [Xiao Li](#), [Kuang Zhengqian](#), [Kang Jichang](#) [程序自动并行化过程中的任务划分](#)[期刊论文]-[西北工业大学学报](#) 2000(2)