国防科技大学学报
JOURNAL OF NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY
第13卷第2期 1991年6月 Vol. 13 No. 2

多向嵌套循环的FMS并行执行模式

杨学军

（电子计算机系）

摘 要 基于小粒度并行结构的分类，提出了结构范式的概念。在此基础上，探讨了多向嵌套循环的并行执行问题，提出了父、母、子进程的概念以及面向多向嵌套循环的FMS处理机调度策略

关键词 多处理机系统，循环，结构范式，进程，调度

分类号 TP31

因为程序的90%左右运行时间花费在循环部分，所以开发循环的并行性是并行处理领域中的一个重要课题。

一般情况下，并行循环可以分为Doall、Doacross两类。在科学计算问题中，并行循环往往表现为嵌套形式。对于并行循环的非嵌套形式，目前已经有成熟的并行执行方法，即PS、SS、CS、GSS处理机调度策略。但是，对于并行循环的嵌套形式，究竟采用何种并行执行方法，仍是一个需要进一步探讨的问题。

本文着重探讨多重嵌套循环的并行执行方法。为了允许高层散列（high level spreading），我们增加了Parcase并行结构。这有利于最大限度地开发并行性，提高多处理机系统的利用率。

1 多重嵌套循环的分类及并行性描述

1.1 多重嵌套循环

首先定义多重嵌套循环，并对其进行分类。

定义 1 假设 l_1、l_2 是两个循环，如果 l_2 出现在 l_1 循环体之中，则称 l_1 包含 l_2.

记作：

$$l_1 \supset l_2.$$

形式上，可以把一个多重嵌套循环看作循环的有限集合：

$$M_l = \{l_1, l_2, \cdots\cdots, l_m\}, m > 1$$

其中，如果一个循环 l_i 不包含任何其它循环，即：$\forall\, l_j (l_j \in M_l \rightarrow \neg (l_i \supset l_j))$ 为真，则称 l_i 为最内层循环，否则称为外层循环。

1990年3月21日收稿

定义 2 对于一多重嵌套循环 $M_l=\{l_1,l_2,\cdots,l_m\}(m>2)$，如果下列条件成立：

$$\forall\, l_i\forall\, l_j\{l_i\in M_l\wedge l_j\in M_l\rightarrow[\neg(l_i\supset l_j)\rightarrow(l_j\supset l_i)]\},\ i\neq j$$

则称 l_i 为单向嵌套循环，否则称为多向嵌套循环。

1.2 嵌套深度

在一个多向嵌套循环中，一个循环 l_i 可能被多个外层循环所围绕，我们把围绕该循环的循环个数称为 l_i 的嵌套深度，并记作：d_i.

在一个多向嵌套循环中可能有多个最内层循环，我们可以按最内层循环出现的语法顺序对其进行编号，并把最内层循环的嵌套深度表示为一个数组：

$$\text{Depth：Array }[1\cdots K]\text{ of Integer }K\geqslant 1$$

1.3 循环单元

在多向嵌套循环中，每个循环均有自己的指针变量。对于一个嵌套为 d_i 的最内层循环 l_i 来说，围绕它的外层循环可按其嵌套深度排成一个序，这样围绕 l_i 的外层循环的循环指针变量可用一个向量来表示：

$$\mathbb{N}\text{：Array }[1\cdots d_i]\text{ of Integer }d_i\geqslant 1$$

并用 DIVi 表示 $\mathbb{N}[i]$的定义域。

在一个程序段 S 之中，如果符号 k 在其中出现，那么我们可以用另一符号 j 对 k 进行替换，替换之后可得一新程序段 S'，并记作：$S_{k|j}$.

定义 3 假设最内层循环 l 的嵌套深度为 d，其外层循环的指针向量为 $\mathbb{N}$，那么我们称：

$$\llcorner\ \mathbb{N}(1)|i_1,\mathbb{N}(2)|i_2,\cdots,\mathbb{N}(d)|i_d\qquad\text{其中 }i_j\in\text{DIV}(j),\ j=1,\cdots,d$$

为循环单元。

1.4 结构范式

假设 $T=\{T_1,T_2,\cdots,T_n\}$ 是一并行实体的集合。为了描述并行关系，引入三个算子：Ⅰ、Ⅱ和Ⅲ，分别称为串行算子、并行算子和并行相关算子。其含义是：

T_i Ⅰ T_j 表示 T_i 优先于 T_j 执行；

T_i Ⅱ T_j 表示 T_i 和 T_j 可并行执行；

T_i Ⅲ T_j 表示 T_i 和 T_j 可并行执行，但需加入同步指令确保数据相关关系不被破坏。

定义 假设 $T=\{T_1,T_2,\cdots,T_n\}$ 是一并行实体集合。

(1) 原子范式

我们称 $T_i\in T$ 为原子结构范式。

(2) 一个结构范式是满足下述规则的符号串：

(a) 原子范式是结构范式；

(b) 如果 T_1、T_2 是结构范式，则(T_1) Ⅰ (T_2)和(T_1) Ⅱ (T_2)是结构范式；

(c) 如果 $T_{1|i}$是结构范式，则

$$\mathop{\text{Ⅱ}}_{i=1}^{N}(T_{1|i}),\quad \mathop{\text{Ⅰ}}_{i=1}^{N}(T_{1|i}),\quad \mathop{\text{Ⅲ}}_{i=1}^{N}(T_{1|i})$$

是结构范式。

(3) T 称为上述结构范式的基。

例如：

```
Doall   I=1, N
    Parcase
        Doall   J=1, M
            H
        END   Do
    Another      case
        Doacross   K=1, M
            H
        END   Do
    END
END
```

可以表示为：

$$\prod_{i=1}^{N}\left[\left(\left(\prod_{k=1}^{M}(H_{K|k})\right)\parallel\left(\prod_{j=1}^{M}(H_{J|j})\right)\right)_{I|i}\right]$$

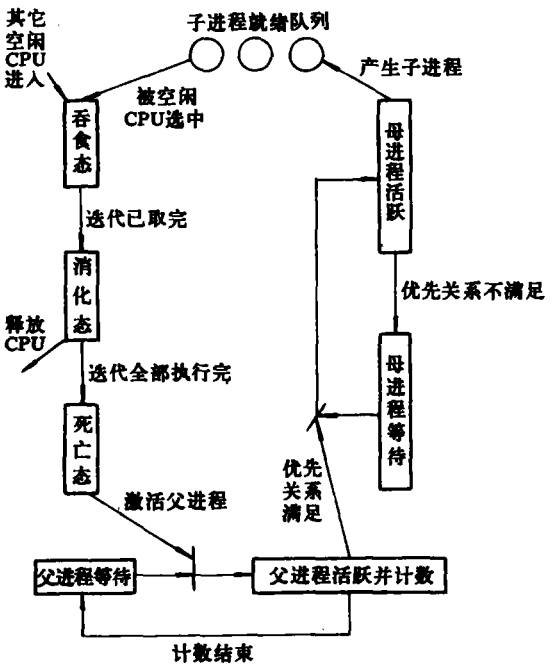

图 1　父、母、子进程的执行过程

1.5　结构范式的性质

定义　如果并行结构 PS_1，可等价变换为 PS_2，则称 $P_a(PS_1)$与 $P_a(PS_2)$等价，记为：$P_a(PS_1)\Leftrightarrow P_a(PS_2)$，其中：$P_a(PS)$表示并行结构 PS 的结构范式。

定理　假设 A、A_1、A_2、A_3、B 是结构范式，则

$$(A_1 \ \mathrm{I}\ A_2) \ \mathrm{I}\ A_3 \Leftrightarrow A_1 \ \mathrm{I}\ (A_2 \ \mathrm{I}\ A_3)$$

$$(A_1 \ \mathrm{II}\ A_2) \ \mathrm{II}\ A_3 \Leftrightarrow A_1 \ \mathrm{II}\ (A_2 \ \mathrm{II}\ A_3)$$

$$A_1 \ \mathrm{II}\ A_2 \Leftrightarrow A_2 \ \mathrm{II}\ A_1$$

$$\mathop{\mathrm{II}}_{j=1}^{N}\left(\mathop{\mathrm{I}}_{i=1}^{M}(A_{k|i,L|j})\right) \Leftrightarrow \mathop{\mathrm{I}}_{i=1}^{M}\left(\mathop{\mathrm{II}}_{j=1}^{M}(A_{k|i,L|j})\right)$$

$$\mathop{\mathrm{II}}_{j=1}^{N}\left(\mathop{\mathrm{III}}_{i=1}^{M}(A_{K|i,L|j})\right) \Leftrightarrow \mathop{\mathrm{III}}_{i=1}^{M}\left(\mathop{\mathrm{II}}_{j=1}^{N}(A_{K|i,L|j})\right)$$

$$\mathop{\mathrm{II}}_{i=1}^{N}(A_{K|i} \ \mathrm{I}\ B_{K|i}) \Rightarrow \left(\mathop{\mathrm{II}}_{i=1}^{N}A_{K|i}\right) \ \mathrm{I}\ \left(\mathop{\mathrm{II}}_{i=1}^{N}B_{K|i}\right)$$

因篇幅限制，证明从略。

2 父、母、子进程并行执行模式

多向嵌套循环的并行执行问题是一个很复杂的问题。为了清楚地阐述本文所提出的思想，我们先从基本并行执行模型入手，逐步地把基本模型演化为真正的并行执行模型。

在基本的并行执行模型中，我们把一个多向嵌套循环分解为三类进程：父、母进程和子进程。所谓子进程是可从并行执行的循环单元。所谓父、母进程是两个控制进程，其功能是产生子进程。

定义

（1）如果一个循环单元还没有被运行，则称它为静止的循环单元。

（2）如果一个循环单元中有的迭代已被运行，有的迭代还没有被运行，则称它为被吞食的循环单元。

（3）如果一个循环单元中所有的迭代都已运行，但有的迭代还没有运行完，则称它为被消化的循环单元。

（4）如果一个循环单元中所有的迭代都已运行完，则称它为死亡的循环单元。

图 1 表示父、母、子进程的执行过程及其相互联系。

系统首先启动母进程运行，母进程在运行过程中不断地产生子进程。当子进程优先关系不满足时，母进程进入等待状态。被母进程产生的子进程首先进入就绪队列，当系统中有处理机空闲时，就从队列中选取子进程运行，子进程由静止态变为吞食态。由于子进程又包含了很多小粒度任务，所以系统允许其它 CPU 进入被吞食的子进程，来并行执行同一子进程。当被吞食的子进程的迭代被取完时，子进程进入消化态，并开始释放处理机。当子进程中所有迭代均已运行完时，子进程进入死亡态，并激活父进程。父进程对已运行完的子进程进行计数，并判断其它子进程的优先关系是否满足；若满足，则激活母进程。

3 父、母、子进程的产生规则

3.1 子进程的并行执行

子进程的并行执行类似于单重循环的并行执行，一个循环单元可以变换为

Var J：Privated　I：shared {Process}

```
LL1   Test & Set 0, 0
        J=I
        I=J+1
      Clear 0, 0
        IF(J.EQ.N+1)     CALL  NEXT
        IF(J.GT.N+1)     CALL  EATER
              H
        IF(J.NE.N)  GO TO     LL1
LL2     CALL  ADVANCE
```

3.2 父、母进程的形成规则

对于循环L，如果HL是以其循环单元为基的并行范式的集合，则L的父、母进程可以通过如下规则（F(H)，M(H)表示L的父、母进程）产生：

(1) 若H∈HL是原子的，则

M(H)：I=1
　　Create(I,IV)
IV是H的外层循环指针向量。
F(H)：CNF：=CNF−1
　　IF CNF＜0　THEN　WST(NF)

(2) 若H_1，H_2∈HL且H=H_1IH_2，则

M(H)：M(H_1)
　　WST(NM)
　　M(H_2)
F(H)：F(H_1)
　　PST(NM)
　　F(H_2)

(3) 若H_1，H_2∈HL且H=H_1 Ⅱ H_2，则

M(H)：M(H_1)
　　M(H_2)
F(H)：F(H_1)
　　F(H_2)

(4) 若$H_{1|1}$∈HL，$H=\mathop{\text{I}}\limits_{1=1}^{N}(H_{1|1})$，则

M(H)：DO　I=1，N
　　修改IV
　　M(H_1)
　　END DO
F(H)：DO　I=1，N
　　F(H_1)

END DO

其中：NF、NM 分别为运行父、母进程的进程号，CNF 是一共享变量。

3.3 基本原语

为了支持 FMS 并行执行模式，我们引入了一些基本原语. 本节将介绍这些原语的功能。

NEXT：选择下一个被吞食的子进程

EATER：执行下一个被吞食的子进程

ADVANCE：将当前被执行的子进程的状态置为死亡态，并推进其它子进程的状态

Create(I,IV)：创建一个新的子进程

WST(N)：等待进程 N 发出的信号

PST(N)：向进程 N 发出信号

4. 结束语

本文基于对多重嵌套循环的规范化，提出了面向多重嵌套循环的 FMS 并行执行模式。实践证明该模式是行之有效的。

参 考 文 献

[1] Abu W, Kuck D and Lawrié D. On the Preforance Enhancement of Paging Systems Trough Program Analysis and Transformations. In IEEE Trans on Compt. 1981, C-30(5)

[2] Fang Z, Yew C, Tang T and Zhu C. Dynamic Processor Self-scheduling for General Parallel Nested Loops. ICPP87: 1~18

[3] Polychronopoulos C D. Parallel programming and compilers. by Kluwer Academic publishers, 1988

[4] 杨学军. 多任务程序并行执行技术的研究. 国防科技大学博士论文, 1990

FMS Parallel Executing Scheme for Multi-way Nested Loops

Yang Xuejun

(Department of Computers Science)

Abstract

Based on the classification of fine grain parallel structure, this paper proposes the concept of structure normal form. Based on the concept, the parallel executing problem for multi-way nested loops is investigated, father-mother-son process is discussed, and FMS processor scheduling scheme is presented for multi-way nested loops.

Key words multiprocessor system, loop, structure normal form, process, scheduling