第三章数据依赖关系

数据依赖

前情提要:

对于一条等式,如a=b+c;我们称集合$OUT=\{a\},IN=\{b+c\}$

数据依赖类型

对于语句S,T,如果存在下述条件之一,则称语句T,依赖于语句S,记为$S\delta T$

> - 流依赖:$S\delta^f T$ 若$x\in OUT(S)$,且$x\in IN(T)$,且T使用S计算出的x的值,则T流依赖于S,**(先写后读)**,下面是一个实例:
>
> > **a**=b+c;
> > d=**a**+c;
>
> - 反依赖:$S\delta^a T$,若$x\in IN(S)$,且$x\in OUT(T)$,但S使用x先于T对x的赋值,则T反依赖于S**(先读后写)**,下面同样是一个实例:
>
> > c=**a**+b
> > **a**=d+e
>
> - 输出依赖:$S\delta^o T$,若$x\in OUT(S)$且$x\in OUT(T)$,但S较T之前对x进行定值,**先后赋值**,下面也是一个实例:
>
> > **a**=b+c
> > **a**=e+f

数据依赖充要条件:

语句S,T在循环L中,对于变量$s\in S(i),v\in T(j)$,满足下列条件时则称存在数据依赖:

> - 1.u,v至少一个为输出变量
> - 2.u,v表示用一个存储单元M
> - 3.在L的顺序执行中$S(i)$先于$T(i)$
> - 4.在顺序执行中 $S(i),T(j)$**之间没有对M的写操作**

依赖距离&依赖向量

依赖距离向量

令$\alpha=(\alpha_1,\alpha_2,...,\alpha_n),\beta=(\beta_1,\beta_2,...,\beta_n)$假定$\alpha,\beta$存在数据相关性则我们另一依赖距离向量:

$$D=(D_1,D_2,...,D_n)=\beta-\alpha$$

相关距离向量指明在同一存储单元的两次访问之间循环迭代的实际距离,它们对开发并行性或优化存储器层次结构时起到指引作用。

依赖方向向量

$$d_i=\begin{cases}\alpha_i<\beta_i & <\text{或}1\\ \alpha_i=\beta_i & =\text{或}0\\ \alpha_i>\beta_i & >\text{或}-1\end{cases}$$

相关方向向量对计算循环体间相关性十分有用，其相关性是通过**相关方向向量不是"="号**的外层循环传递的;

做题方法

首先摸清楚语句执行顺序,通常情况下i+a,a越大越先执行,反之i-a,a越大越迟执行.
列出所有元组,后标记对应语句标识符(如S,T),以及读还是写,如下表

标识符	读/写	下标
S	r	I+4,J-2
S	r	I,J+3
T	w	I,J
S	r	I-2,J+1

然后另先执行的为i_1,j_1,后执行的为i_2,j_2,根据r/w判断依赖关系
做个等式移项使得一边只有i_2,j_2即可得到对应依赖距离向量以及方向向量,如第3项和第4项组成依赖关系:

$$\begin{cases}i_1=i_2-2\\ j_1=j_2+1\end{cases}=>\begin{cases}i_2=i_1+2\\ j_2=j_1-1\end{cases}=>(2,-1)=>(1,-1)$$

循环向量化

可向量化循环的充要条件:

对于循环L=(L1,L2,. . ., Lm)其最内层循环Lm可向量化当且仅当：Lm中任意两个语句S和T，

> （1） 当S<T时，不存在方向向量为(0,0,…,1)的S对T的依赖关系，$T\delta S$；（三种依赖关系中任何一种）
> （2） 当S = T时，不存在方向向量为(0,0,…,1)的S对T的**流依赖关系**，$T\delta^f S$；

换言之，**最内层循环**中如果不存在与语句词法**顺序相反的依赖关系(反映在方向向量里就是最内层循环变量的方向为1)**（即反向依赖），则最内层循环可向量化。

循环并行化

可循环并行化的充要条件:

循环L=(L1,L2,. . ., Lm)中内层循环Lk可并行化当且仅当循环L中不存在层次为k的依赖关系，即**不存在方向向量**为（0,…,0,1k,,…,)(含k-1个前导零）的依赖关系（即由K层携带的依赖关系）。
PS:循环方向向量每一位均不为1

循环变换

想法:将一个大循环拆分为若干相关小循环,保证数据一致性
操作步骤:画出数据依赖图,将强连通子图进行凝聚,凝聚后的节点放在同一个拆分后的循环中进行计算,其他的之后再算,拆分后的可以使用并行化或者向量化

语句重排

不改变数据依赖图的情况下,对同一个循环的语句重新排列,使得对于某些情况下可以改善程序并行基础

循环置换

充要条件:设P是m*m的置换矩阵,p是每行每列有且仅有一个元素为1的矩阵,由循环L生成,那么对于一个循环是合法的,当且仅当每个**方向向量**与该矩阵乘积均大于0

循环逆转

例如从原来i++变为i--,该变换合法的一个条件为变换之后方向向量均为正向量,则称变换前后循环等价

MPI 编程

```
MPI_Send(
    void* data,             //数据指针,表示数据存哪
    int count,              //表示存多少个数据,配合上面这个指针使用
    MPI_Datatype datatype,  //接受数据类型,如果为int 则为MPI_INT,接受数据为char 则为MPI_CHAR
    int destination,        //目标id
    int tag,                //接受节点的标记需一致才能接受
    MPI_Comm communicator)  //通信交流,暂不明确目前设置为MPI_COMM_WORLD

MPI_Recv(
    void* data,             //数据指针,表示数据存哪
    int count,              //表示存多少个数据,配合上面这个指针使用
    MPI_Datatype datatype,  //接受数据类型,如果为int 则为MPI_INT,接受数据为char 则为MPI_CHAR
    int source,             //表示谁接受
    int tag,                //接受节点的标记需一致才能接受
    MPI_Comm communicator,  //通信交流,暂不明确目前设置为MPI_COMM_WORLD
    MPI_Status* status)     //提供接受msg的信息


#pragma omp parallel
{
    int id = omp_get_thread_num();
    if(id == 0) printf("123");
    #pragma omp barrier
    if(id == 1) printf("456");
    #pragma omp barrier
    if(id == 0) printf("789");
}
```