并行化编译中的一种集成优化方法*

孙 彤 李三立 李晓明

(清华大学计算机系 北京 100084) (哈尔滨工业大学计算机科学与工程系 哈尔滨 150001)

摘要 本文提出了一种面向分布存储器多机系统的并行化编译方法. 针对分布存储并行系统的特点,作者采用的基本优化策略是:折衷并行性与数据引用局部性;减少和隐藏通信开销. 通过对基于仿射函数的程序分解方式所导致的数据通信性质的分析,得到了适合分布存储结构特殊要求的并行性开发方法. 为了在保持并行性的前提下最小化通信数据总量,提出了基于齐次线性方程组求解的程序全局优化分解方法. 为了优化数据通信的组织,提高结点代码的效率,又提出了一种以线性不等式组作为工具的更加实用的通信优化和结点代码生成方法.

关键词 并行化编译,分布存储,优化编译,消息传送,任务划分,单程序多数据流.

尽管并行化编译方面的研究与开发已经有 20 多年的历史,但至今为止它仍然是实现实用的高性能计算的主要瓶颈. 本文给出的并行化编译方法系统地集成了多种优化技术,它将并行化过程分为若干步,在每一步完成一定的优化目标. 我们针对每一步研究了相应的优化技术.

为了使串行程序能在分布存储器多机系统上有效运行,必须首先发现并提取程序中的并行性,然后再根据开发出的并行性对程序中的计算和数据进行划分,最后分解出一组消息传送式的结点程序,它们将被分配到不同的处理机结点中并行执行. 为生成优化的结点程序,我们的编译器采用如下的处理过程:

(1)词法分析、语法分析、流分析、标量优化、循环标准化、归纳变量检测、归约识别;

(2)数据相关性分析;

(3)标量和数组扩展,结点分割(Node Splitting);

(4)对所有循环嵌套由外向内进行循环分布[1];

(5)将 DOALL 循环进行内移分布;

(6)将不包含 DOALL 循环的嵌套转换成 Ⅱ 型标准形式,以便于利用流水并行性;

(7)计算循环嵌套和数组的安放函数,减少通信数据量;

(8)优化确定虚拟阵列的分布函数,在提高负载平衡性和减少通信开销之间取得折衷;

---

* 本文研究得到国家攀登计划资金资助. 作者孙彤,1968 年生,博士,讲师,主要研究领域为并行计算机体系结构,并行操作系统,编译和语言. 李三立,1935 年生,教授,中国科学院院士,博士导师,主要研究领域并行处理,先进处理机体系结构. 李晓明,1957 年生,教授,博士导师,主要研究领域为并行计算机系统结构,并行编译,依自然法则的计算方法.

本文通讯联系人:孙彤,北京 100084,清华大学计算机系

本文 1995-12-05 收到修改稿

(9)采用通信优化和开发流水并行性等技术,生成优化的SPMD结点程序.

本文第1节讨论并行性提取问题.第2节给出一种以减少通信数据总量为出发点的程序全局优化分解方法.第3节给出一种基于线性不等式组描述的通信优化和结点代码生成方法.

1 开发循环并行性

我们的并行化编译器的目标是要生成并行性可伸缩的SPMD并行程序,这样加速比才能随着机器规模的扩展而增加,而串行程序中能够被利用的可伸缩的并行性主要隐含在循环中,尤其在多层嵌套循环中,可开发并行性的潜力是很大的.对于分布存储器的机器来说,并行性是通过循环嵌套迭代空间分解的方法来利用的.通过依次对串行程序施用循环分布以及DOALL循环内移分布和完全可排列化等变换可完成串行程序中的并行性的提取.通过循环分布可减少循环中所含语句的数量,将循环携带相关变为循环独立相关,从而一方面使尽量多的计算组织到DOALL循环中,另一方面又可生成尽量多的紧嵌套,使流水并行性得以开发.

1.1 循环分解和数据分解

在分布存储系统上,为使多个处理机在并行执行DOALL或DOACROSS[1]循环的迭代时能获得最大的加速,需要考虑负载平衡和通信代价2方面的问题.为了便于进行优化,我们将循环及数据分解过程分为2步.第1步将被分解循环的每个迭代与它所访问的数组的元素建立对应关系,具有对应关系的迭代与数组元素将被分配到相同的处理机中,通过优化地选择这种对应关系可以减少通信数据量.这种对应关系是通过将迭代和数组元素映射到一个规模不受限的虚拟处理机阵列上来建立的.这一步称为循环和数组的安放.第2步是将虚拟阵列分割映射到物理的机器上(裁剪程序中的并行性以适应物理机器的并行性),这样迭代集和数组就可按照它们在虚拟阵列上的安放方案分布到物理处理机中.这时的优化目标是负载平衡和进一步减少通信代价.这一步称为虚拟阵列分布.设Z为整数集合,N为非负整数的集合,$IS\subset Z^l$为第l层循环的迭代空间,$DS\subset Z^m$为m维数组的下标空间,$VS\subset N^n$为n维虚拟处理机空间.

定义 1.1. 第l层循环的安放函数定义为$c:IS\rightarrow VS$,$\forall\ x\in IS$,$c(x)=Cx+\gamma$.$n\times l$矩阵C称为循环安放矩阵,n维向量γ称为偏移向量.m维数组的安放函数定义为$d:DS\rightarrow VS$,$\forall\ x\in DS$,$d(x)=Dx+\delta$.$n\times m$矩阵D称为数组安放矩阵,n维向量δ称为偏移向量.语句S的迭代空间为$IS(S)$,数组A在S中的一个引用可表示为$A(f(I))$,其中I为循环变量向量,f为一个仿射函数,$f:IS(S)\rightarrow DS_A$,$\forall\ x\in IS(S)$,$f(x)=Fx+\varphi$.矩阵F称为访问矩阵,φ称为访问向量,f称为访问函数.

为了得到无通信的循环分解,循环安放函数和数组安放函数之间应满足一定的关系,对此我们有如下引理.

引理 1.1. 假设数组引用$A(f(I))$在语句S中出现,包含S的h层嵌套循环中的第l层循环为被分解循环,它的安放函数为$c(x)=Cx+\gamma$,则对$A(f(I))$的访问不引起通信当且仅当$d_A\cdot f=c^{(h-l)}$,即:$\forall\ x\in IS(S)$,$D_AFx+D_A\varphi+\delta_A=(C|0^{(h-l)})x+\gamma$,其中$c^{(h-l)}(x)=(C$

$|0^{(h-l)})x+\gamma, C|0^{(h-l)}$是将矩阵 C 增加 $h-l$ 列零向量构成的矩阵,我们称它为 S 的安放矩阵.

定义 2.2. 假设 n 维虚拟处理机空间 $VS=[0:V_1-1]\times\cdots\times[0:V_n-1]$,$n$ 维物理处理机空间 $PS=[0:p_1-1]\times\cdots\times[0:p_n-1]$. VS 的分布函数为 $dist:VS\rightarrow PS$, $dist=(\rho_1,\cdots,\rho_n)$, $\forall\ x=(x_1,\cdots,x_n)\in VS$, $dist(x)=(\rho_1(x_1),\cdots,\rho_n(x_n))$. ρ_i 称为轴分布函数. 轴分布函数可以是:$BLOCK$, $CYCLIC$, $BLOCK-CYCLIC(B)$.

1.2 DOALL 循环的内移分布

利用循环分布技术可以发掘串行程序中的 DOALL 循环,但这样得到的 DOALL 的循环体中可能含有多个语句及包含这些语句的嵌套循环. 为了增加循环和数组安放的优化潜力,希望每个 DOALL 循环包含尽量少的语句. 在文献[2]中我们证明:经过 DOALL 循环的内移与分布变换,可使每个 DOALL 循环的循环体中都只包含一个语句或者一个仅含单语句的循环嵌套,而经过这些变换并不增加通讯和同步开销. 我们称经过这些变换的程序为 I 型标准形式的程序.

1.3 不含 DOALL 的紧嵌套循环中并行性的开发

Wolf 等人[3]提出了一种开发紧循环嵌套中最大并行性的方法,该方法将提取并行性的过程分为 2 步,第 1 步将循环嵌套变换为由若干完全可排列嵌套组成的一种标准形式的循环嵌套,并且从最外层开始,使每个完全可排列嵌套的层数都尽可能地大. 这种形式的标准嵌套称为完全可排列化嵌套. 第 2 步是对这种标准形式的嵌套再施行进一步的变换来最大化并行性. 每个 n 层的完全可排列嵌套都含有 $n-1$ 维的并行性,它被片化后,可以以流水方式并行执行. 我们使用 Wolf 的方法中第 1 步的算法提取不含 DOALL 的循环嵌套中的流水并行性. 我们称经过完全可排列化后的嵌套为 II 型标准形式的嵌套.

2 程序优化分解

事实上,循环嵌套的迭代空间的分解就是将其划分为低维的一族平行的超平面,超平面之间并行执行. 为保证每个迭代都尽量访问本地数据,嵌套中访问的数组相应地也要划分成超平面,而且数组元素的超平面与迭代的超平面应按对应关系分配到相同的处理机中. n 维的空间划分为 $n-m$ 维的超平面,则称其为 m 维的划分. 如果同一数组在流相关的 2 个嵌套中出现,而它在 2 个嵌套中的划分的维数不同或维数相同但划分的方向不同时,都会导致较大的通信代价,我们称这种通信为重组通信. 事实上对于适合并行计算的问题,对迭代空间进行一维划分就足够了. 当一个嵌套的迭代空间中有多维的并行性时,我们只需选择其中的一维进行分解,这样就有多种划分方向可供选择,因而优化地选择划分方向就可减少重组通信的次数.

2.1 利用 DOALL 并行性的分解

在下文中,称 l 层嵌套中的第 l 层循环的安放函数为此嵌套的安放函数. 由于只进行一维划分,虚拟处理机阵列为一维空间,因此一个 l 层嵌套 L 的安放矩阵为 $1\times l$ 矩阵,记为 $C=(t_1,\cdots,t_l)$,一个 m 维数组 A 的安放矩阵为 $1\times m$ 矩阵,记为 $D=(s_1,\cdots,s_m)$. 对于一个 I 型标准形式的循环嵌套 L,若其中 $i_1,\cdots,i_n$ 层循环为 DOALL,则为了利用 DOALL 并行性,

$t_{i_1},\cdots,t_{i_n}$中应至少有 1 个不为 0,且其它 $t_j(j\neq i_1,\cdots,i_n)$均为 0. 要求 $t_{i_1},\cdots,t_{i_n}$不全为 0 是为了保持并行性,因此我们称这一要求为并行性条件. 若满足 $t_j\neq 0$ 的最小的 j 为 j_0,则第 j_0 层循环称为被分解循环. 设数组 A 在 L 中的引用为 $A(f(I))$,其中$\forall\ x\in IS(L)$,$f(x)=Fx+\varphi$,$F=(a_{ij})_{m\times l}$. 根据引理 1.1,将 A,L 安放到虚拟阵列中之后,$A(f(I))$为本地访问的充分必要条件是 L 的安放函数 c 与 A 的安放函数 d 满足 $d\cdot f=c$,即 $DF=C$ 且 $D\varphi+\delta=\gamma$.

定义 2.1. 若 $DF\neq C$,则称访问 $A(f(I))$所需的通信为重组通信. 若 $DF=C$但 $D\varphi+\delta\neq\gamma$,则称所需的通信为平移通信. 在程序执行过程中数组的安放矩阵改变时所需的通信也称为重组通信,而当安放矩阵不变但安放偏移改变时,也称所需通信为平移通信.

显然重组通信要比平移通信的代价高得多,因此首要的是减少重组通信的数量. $DF=C$可写作如下形式的齐次线性方程组(其中 $s_1,\cdots,s_m,t_{i_1},\cdots,t_{i_n}$为未知数):

$$\begin{cases} a_{11}s_1+a_{21}s_2+\cdots+a_{m1}s_m=0 \\ \cdots \\ a_{1i_1}s_1+a_{2i_1}s_2+\cdots a_{mi_1}s_m-t_{i_1}=0 \\ \cdots \\ a_{1i_n}s_1+a_{2i_n}s_2+\cdots+a_{mi_n}s_m-t_{i_n}=0 \\ \cdots \\ a_{11}s_1+a_{21}s_2+\cdots+a_{m1}s_m=0 \end{cases} \tag{1}$$

引理 2.1. 若方程组(1)存在满足并行性条件($t_{i_1},\cdots,t_{i_n}$不全为零)的解,则这样的解所确定的 L 和 A 的安放函数可使对 $A(f(I))$的访问为本地访问. 否则需要通过重组通信来完成访问,此时称 $A(f(I))$为本质通信引用.

引理 2.2. 设 A 在 L 中有 h 个非本质通信引用模式 $f_1,\cdots,f_h$,则:

(1)A 在 L 中是可无重组通信安放的当且仅当下面方程组 $DF_1=DF_2=\cdots=DF_h=C$ 有满足并行性条件的解. 该方程组称为定向约束方程组,其中所有方程的集合记为 $EO(A,L)$.

(2)A 在 L 中是可无通信安放的当且仅当 A 是可无重组通信安放的并且存在安放矩阵 D 使得 $D\varphi_1=D\varphi_2=\cdots=D\varphi_h$. 称 $D\varphi_1=\cdots=D\varphi_h=\gamma-\delta$ 为偏移约束方程组,记为 $ED(A,L)$.

定理 2.1. 对于 l 个循环嵌套 $L_1,\cdots,L_l$ 和它们所引用的 h 个数组,存在它们的一组安放方案使得每个数组在每个嵌套中的非本质通信引用都不引起重组通信当且仅当方程组 $\bigcup_{i=1}^{l}\ \bigcup_{A\in Arrays(L_i)} EO(A,L_i)$有满足并行性条件的解.

2.2 利用流水并行性的分解

当 I 型标准形式的程序中的一个嵌套中没有 DOALL 循环时,我们将其变换为 II 型标准形式,利用其中的流水并行性. 利用流水并行性还可为无重组通信的安放方案的选择增加自由度. 当对于一个 I 型标准形式的程序段中的嵌套及其所引用的所有数组不存在一组无重组通信的安放方案时,我们将其转换成 II 型标准形式,然后再寻找无重组通信的安放方案. 2.1 节中有关无重组通信的结论都适合于利用流水并行性的情形,差别仅在于并行性条件不同.

2.3 全局优化分解

在我们的程序分解方式下,可能导致的通信可分为 3 种类型,按照通信代价由小到大的

顺序分别为:平移通信、流水通信和重组通信.由于重组通信的代价较高,因此在我们的方法中,首先通过优化地确定嵌套和数组的安放矩阵,减少重组通信的次数.我们考虑使用流水通信来取代重组通信以减少通信代价,但这会使一些嵌套按流水并行方式进行计算,从而造成处理机空闲,这时就需要在两者间取得折衷.

在确定安放矩阵时为达到全局最优,可以考虑将程序按照流相关性划分为一些无重组通信区,并使这种区域的数量尽量少,这样在每个区域内可确定不引起重组通信的嵌套及数组安放矩阵,而重组通信仅出现在无重组通信区的交界处.因而通过选择无重组通信区的划分方案就可以最小化重组通信代价.

在划分无重组通信区时采用的方法是:首先将每个循环嵌套都作为一个无重组通信区,然后逐步合并这些无重组通信区,合并的策略是优先执行那些可消除最多通信的合并,这样可使得到的结果在某种程度上接近最优.在合并过程中,如果合并后不是无重组通信区,则我们考虑采用流水通信是否可以获得更好的性能.若不是则放弃合并,若是则这样的合并被采纳.我们引入一种称为加权安放相关图的数据结构,用于完成这样的安放矩阵计算过程.这种数据结构描述了程序中的数组和迭代空间的安放需求.通过对加权安放相关图的分割,可以得到安放通讯图,该图确定了所有数组和迭代空间的安放矩阵,我们可定义安放通讯图的值,用来近似地表明在这种安放方案之下所能获得的加速收益.实现这一方法的算法见文献[2].

在确定了安放矩阵之后,我们考虑通过优化地确定嵌套和数组的安放偏移,来最小化平移通信代价.为此引入了偏移图的概念,将问题模型化为计算最大代价的无冲突生成子图.文献[2]给出了完成这一任务的一种启发式算法.

3　通信优化和结点程序生成

一个程序中每个嵌套的安放函数和每个数组在程序中不同位置上的安放函数以及虚拟处理机阵列的分布函数,这 3 个方面的信息决定了计算和数据在物理处理机结点之间的分配,我们称它们为程序分解方案.由程序分解方案可以计算出每个嵌套的迭代空间中分配到各个结点处理机中的迭代子集以及每个数组分配到各个结点处理机中的元素子集,由此还可以确定出非本地的数组访问.在此基础上可分别计算出 SPMD 型结点计算代码和结点通信代码,然后将计算代码和通信代码合并到一起,就得到了原始的 SPMD 型结点程序.对原始的结点程序采用一些优化技术可生成有效的结点程序,从而获得更好的加速效果.

虽然程序分解方案确定了结点程序中需要通信的数据总量,但如何组织通信对于结点程序的性能仍起着很关键的作用.这可以从 2 个方面来看:

(1)由于消息发送过程中起动时间远大于在互联网中传送一个数据项所花费的时间,因此将短消息组合成长消息发送,可以减少通信开销.

(2)通信操作在结点程序中的插入位置不合适会阻碍并行性的发挥.对通信操作的插入位置和计算操作的组织进行优化,可以部分地重叠通信和计算的执行过程,隐藏通信开销.

我们的方法[2]是以线性不等式组作为编译器中基本的描述和操作单位,利用它的丰富表达力和良好的可操作性,可以对已有的通信优化方法进行重新组织,在综合其本质的基础上,形成一种系统的框架,从而简化了优化过程并促成了更进一步的优化.

3.1 结点程序生成的基本框架

我们将针对如下形式的嵌套进行讨论,其中 $l2,u2,\cdots,ln,un$ 为仿射函数,$I=(I_1,\cdots,I_n)$.

```
DO I1=l1,u1
  ...
  DO I2=l2(I1),u2(I1)
  ...
    DO In=ln(I1,...,In-1),un(I1,...,In-1)
        ...
        A(g(I))=fun(A(f(I)))
        ...
    ENDDO
  ...
  ENDDO
  ...
ENDDO
```

定义 3.1. 由迭代和处理机号组成的二元组称为迭代处理机对,计算分解关系 CR 是由迭代处理机对组成的集合,$(i,p)\in CR$ 当且仅当处理机 p 执行迭代 i,即:$CR=\{(i,p)\in IS(L)PS\mid dist\cdot c_L(i)=p\}$.

当 $dist=(\rho_1,\cdots,\rho_n)$ 中所有 ρ_j 都为 BLOCK 型函数时,CR 为下面线性不等式组的整数解 $IEQCR(i,p)=$

$$\begin{cases}\lceil vnum(j)/pmum(j)\rceil * p_j\leqslant PR_j(c(i))<\lceil vnum(j)/pnum(j)\rceil * (p_j+1) & (j=1,\ldots,m)\\ l_j(i_1,\ldots,i_{j-1})\leqslant i_j\leqslant u_j(i_1,\ldots,i_{j-1}) & (j=1,\ldots,n)\end{cases}$$

其中 $i=(i_1,\cdots,i_n)$,PR_j 为投影函数:$PR_j(x_1,\cdots,x_n)=x_j$,$p=(p_1,\cdots,p_m)$.

为了得到分配给每个处理机的所有迭代,我们根据不等式组 $IEQ_{CR}(i,p)$ 生成一个 $m+n$ 层循环,该循环按照向量 $(p_1,\cdots,p_m,i_1,\cdots,i_n)$ 的字典序扫描计算分解关系 CR 中的所有元素,其中外层的 m 层 p 循环用于枚举处理机号,内层的 n 层 i 循环用于枚举分配给每个处理机 P 的所有迭代. 这样的代码的结构如下:

```
IF (p1.ge. lp1) .AND. (p1.le. up1) .AND.
  (p2.ge. lp2(p1)) .AND. (p2.le. up2(p1)) .AND.
  ...
  (pm.ge. lpm(p1,..., pm-1)) .AND. (pm.le. upm(p1,...,pm-1)) THEN
  DO i1=li1(p),ui1(p)
    DO i2=li2(p,i1),ui2(p,i1)
      ...
      DO in=lin(p, i1,...,in-1), uin(p, i1,...,in-1)
      ...
      ENDDO
      ...
    ENDDO
  ENDDO
ENDIF
```

定义 3.2. 数据分解关系 DR 是由下标处理机对组成的集合,$(a,p)\in DR$ 当且仅当数组 A 的元素 $A(a)$ 分配到处理机 p 中,即 $DR=\{(a,p)\in DS(A)\times PS\mid dist\cdot d_A(a)=p\}$.

定义 3.3. 设数组 A 的 2 种分解方案分别为 $d_1,dist_1$,和 $d_2,dist_2$,如果在程序中的某一点要将 A 从第 1 种方案变为第 2 种方案,则定义它的位移通信集 $MM\subseteq PS\times PS\times DS(A)$,

$(p_s,p_r,a)\in MM$ 当且仅当处理机 p_s 需要将 $A(a)$ 发送到 p_r.

定理 3.1. $MM=\{(p_s,p_r,a)\in PS\times PS\times DS(A)\mid dist_1\cdot d_1(a)=p_s, dist_2\cdot d_2(a)=p_r, p_r\neq p_s\}$.

按照向量(p_s,p_r,a)的字典序扫描不等式组的解空间可以生成消息发送代码,而按照向量(p_r,p_s,a)的字典序扫描不等式组的解空间可以生成消息接收代码.

定义 3.4. 对于一个需要通信的读引用 $A(f(I))$,定义它的读通信集 $RM\subseteq PS\times IS(L)\times PS\times IS(L)\times DS(A)$,$(p_s,i_s,p_r,i_r,a)\in RM$ 当且仅当处理机 p_s 需要在迭代 i_s 中发送其本地数据 $A(a)$到处理机 p_r 中用于迭代 i_r 的计算$(f(i_r)=a)$.

定理 3.2. $RM=\{(p_s,i_s,p_r,i_r,a)\in PS\times IS(L)\times PS\times IS(L)\times DS(A)\mid (i_r,p_r)\in CR, a=f(i_r),(a,p_s)\in DR, p_s\neq p_r, i_r=i_s\}$.

按向量(p_s,i_s,p_r,i_r,a)的字典序扫描读通信集 RM 可形成消息发送循环嵌套的代码,按照向量(p_r,i_r,p_s,i_s,a)的字典序扫描读通信集 RM 则可生成消息接收代码. 类似地,我们可以定义写通信集,并由它得到针对非本地写引用的通信代码.

使用循环分割(Loop Spliting)方法将计算代码、消息发送代码和消息接收代码合并到一起就得到了原始的结点程序.

3.2 通信优化

通信优化可以从 2 个方面考虑. 一方面是加大消息的粒度,减少发送消息的总数,从而减少发送和接收消息的软件开销. 在我们的方法中通过通信聚集、通信凝聚、消息聚集、多目(Multicasting)或广播通信等优化措施来完成这一任务. 通信优化的另一方面是将消息的传送过程与本地的计算重叠执行,从而隐藏一部分通信延迟. 在我们的方法中通过消息流水、粗粒度消息流水和迭代重排序等优化措施来实现计算与通信的重叠,避免处理机空闲. 通信优化措施的具体实现方法见文献[2].

4 相关的工作及结论

本文提出了一整套实现程序全局优化的并行化编译方法. 我们首先研究了对循环嵌套的迭代空间和数组下标空间采用基于仿射函数的分解方式时所导致的数据通信的有关性质,得到了"DOALL 循环的内移和分布变换不影响通信性能"的结论. 由此提出了一种满足分布存储结构特殊要求的并行性开发方法. 然后在此基础上提出了一种以减少通信数据总量为出发点的,基于齐次线性方程组求解的程序全局优化分解方法.

Sadayappan 等人[4]提出了针对完全并行循环嵌套的超平面分解方法,我们将其推广到任意的循环嵌套上,并给出了在程序全局优化地确定分解方案的方法. Gupta 和 Banerjee 开发了一个自动确定数组静态分解的方法[5],他们基于一种网络寻径模型建立了一个代价估计系统,利用它来完成对各种可能的数组分解方案的穷举查找. 该方法中循环嵌套的分解方案是根据"拥有者计算"规则由数据分解方案决定的. 由于数据分解是全程固定不变的,因而对大型程序可能是无效的. 在我们的方法中,同一数组的分解方案可以是动态变化的,并且循环嵌套的分解方案与数组分解方案是同时确定的,不存在一个决定另一个的问题,因而可获得更好的效果. 我们的方法避免了穷举查找,适用于大型的程序. 他们的方法的不足还在

于不能考虑流水并行性.

与 Anderson 和 Lam 的方法[6]相比,我们通过解齐次线性方程组来确定安放矩阵,避免了用迭代方法计算迭代空间和数组下标空间的划分子空间以及计算矩阵伪逆的复杂计算过程.由于我们对程序中的 DOALL 循环施行了内移分布变换,所有 DOALL 循环都位于最内层的紧嵌套中,因而不必象他们的方法那样按自底向上的顺序针对每个嵌套级进行分解,并且避免了带符号常量的线性代数计算,这样一方面简化了算法,另一方面也增加了进一步减少通讯的可能性.

本文提出的以线性不等式组作为工具的通信优化和结点程序生成方法克服了 RICE 大学 Fortran D 编译器[7]的基于 RSD 的方法以及 PARADIGM[8]的基于符号集的方法处理能力不强的缺陷,使得编译器能够处理更一般的程序类型,从而更加实用化.具体表现在:(1)能处理耦合下标;(2)能处理不规整迭代空间;(3)能处理 BLOCK-CYCLIC 分布方式;(4)能实现各种重组通讯;(5)将各种通信优化方法系统地组织融合到一起,简化了优化过程,并促成了更进一步的优化.

参考文献

1 Zima H, Chapman B. Supercompilers for parallel and vector computers. Addison-Wesley Publishing Company, 1990.

2 孙彤. 数据与计算的对准——面向分布存储并行计算机编译的通信优化技术[博士论文]. 哈尔滨工业大学,1994.

3 Wolf M E, Lam M S. A loop transformation theory and an algorithm to maximize parallelism. IEEE Trans. on Para. and Distri. Systems, 1991,2(4):452～471.

4 Ramanujam J, Sadayappan P. Compile-time techniques for data distribution in distributed memory machines. IEEE Trans. on Para. and Distri. Systems, 1991,2(4):472～482.

5 Gupta M, Banerjee P. Demonstration of automatic data partitioning techniques for parallelizing compilers on multicomputers. IEEE Trans. on Para. and Distri. Systems, 1992,3(2):179～193.

6 Anderson J M, Lam M S. Global optimizations for parallelism and locality on scalable parallel machines. In: ACM SIGPLAN'93 Conference on Programming Language Design and Implementation, 1993. 112～125.

7 Tseng C W. An optimizing Fortran D compiler for MIMD distributed-memory machines [Ph. D. dissertation]. Rice University, 1993.

8 Su E, Palermo D J, Banerjee P. Automating parallelization of regular computations for distributed-memory multicomputers in the PARADIGM compiler. In: 1993 International Conference on Parallel Processing, 1993, I: 30～38.

AN INTEGRATED OPTIMIZATION SCHEME IN PARALLELIZING COMPILERS

Sun Tong Li Sanli

(*Department of Computer Science Tsinghua University Beijing* 100084)

Li Xiaoming

(*Department of Computer Science and Engineer Harbin Institute of Technology Harbin* 150001)

Abstract This paper presents a complete suit of systematic optimizing methods which

may be used in parallelizing compilers for multicomputers or computer clusters. In the compilation scheme, two strategy are adopted. One is trading off parallelism and communication cost and the other is reducing and hiding communication overhead. Through analyzing the properties of data communication required for the program partition approach based on affine functions, the authors find a method to exploit parallelism in serial programs satisfying the special requirements of distributed memory machines. In order to minimize the total of data needing to be communicated, they invent a global optimization program partition method based on solving linear equations. In order to optimize the organization of communication codes and generate more efficient node programs, they invent a more practical method based on linear inequalities to perform communication optimization and node programs generation.

Key words　Parallelizing compiler, distributed memory, optimizing compiler, message passing, task partition, SPMD.