

消除 VLIW 结构上的循环体间冗余流相关^{*}

容红波 汤志忠

(清华大学计算机科学与技术系 北京 100084)

E-mail: ronghb@mail.cic.tsinghua.edu.cn

摘要 数据相关是并行处理的基本依据. 该文指出, VLIW (very long instruction word) 特有的锁步性质使其数据相关性分析具有与众不同的特点. 同一体差上的流相关形成一个线性集合, 多体差上的特征流相关之间也存在包含关系. 据此, 提出一种用于 VLIW 的消除循环体间冗余流相关的方法. 该方法是完备的, 可以去除所有冗余的体间流相关, 从而减轻循环调度的负担. 文章给出判定单体差和多体差存在冗余的充分必要条件, 以及消除冗余的线性复杂度的算法. 这种方法具有普遍意义, 可作为 VLIW 上软件流水和多指令流调度的基础.

关键词 VLIW (very long instruction word), 循环调度, 数据相关, 循环, 冗余.

中图法分类号 TP338

数据相关是并行处理的基本依据. 有关数据相关性分析的成果虽然很多^[1,2], 但是针对 VLIW (very long instruction word) 这一特定结构的工作尚未见到. 目前, 在 VLIW 上进行并行循环调度, 如软件流水, 都是使用传统的基于向量机的数据相关性分析方法^[3,4], 缺乏对 VLIW 的专门研究. 而本文将说明, 由于 VLIW 结构上的特殊性, 它在数据相关方面具有与众不同的特点.

如何开发循环体间的并行性是 VLIW 研究的热点. 影响循环调度的关键因素是体间数据相关, 尤其是流相关. 本文将讨论在调度循环时如何消除冗余的体间流相关, 以降低调度的复杂性.

本文作以下几个假定.

(1) 所有处理器/功能单元以锁步方式操作. 这是 VLIW 的基本特征^[3,4].

(2) 一个周期一条长指令. 注意这个假定没有考虑多周期操作, 如浮点运算, 但这并不影响结果的正确性. 通常, 多周期操作可以用空操作弥补. 例如, 设 OP_i 到 OP_j 有体间流相关, 分别需要 T_i 和 T_j 个周期, 则可在 OP_i 和 OP_j 后分别加入 T_i-1 和 T_j-1 个空操作, 并认为 OP_i 后面的最后一个空操作与 OP_j 相关.

(3) 同一循环体中的操作具有相对固定的位置. 由于这一限制, 本文的结果只能直接用于相关图中的强连通块 (为最小化启动间距而保证优先地、紧密地调度, 次序在列表调度之前确定). 这一条件将在以后的研究有所放松.

一般地, 一个数据相关图 DDG (data dependence graph) 含有多个体差, 每个体差上又有多个流相关. 先分别消除每个体差上的冗余, 再将所有体差结合起来考虑, 进一步消除冗余, 这是本文的基本思想. 下面给出一些基本定义和结论, 其中 $A, B, A_1, A_2, \dots, A_n$ 表示流相关; OP_i 表示一个操作, i 是该操作在循环体中的序号.

1 单体差的冗余判定

定义 1. 设 A 是一个流相关, 表示 OP_i 到 OP_j 具有读写关系, 体差为 d , d 是非负整数, 记作 $A: (OP_i \rightarrow OP_j, d)$.

当 $d=0$ 时, A 为体内流相关, 否则为体间流相关. 下面提到的任何流相关, 都默认为体间流相关, 即 $d \geq 1$.

* 本文研究得到国家自然科学基金 (No. 69773028) 资助. 作者容红波, 1972 年生, 博士生, 主要研究领域为并行编译技术, 并行体系结构. 汤志忠, 1946 年生, 教授, 博士生导师, 主要研究领域为计算机并行算法, 并行编译技术, 并行体系结构.

本文通讯联系人: 容红波, 北京 100084, 清华大学计算机科学与技术系

本文 1998-12-01 收到原稿, 1999-02-01 收到修改稿

定义 2. 设有流相关 A, B , 如果只要满足 A, B 就自然得到满足, 则称 A 包含 B , 记作 $A \supseteq B$.

定义 3. 设有流相关 A, B , 如果 $A \supseteq B$, 并且 $B \supseteq A$, 则称 A, B 相等, 记作 $A = B$.

定义 4. 设有流相关 A, B , 如果 $A \supseteq B$, 但 A 不等于 B , 则称 A 真包含 B , 记作 $A \supset B$.

性质. $\supseteq, =, \supset$ 都具有传递性.

引理 1. (平移性质) 设有流相关 $A: (OP_i \rightarrow OP_j, d), B: (OP_{i'} \rightarrow OP_{j'}, d)$, 则当 $i - j = i' - j'$ 时, $A = B$.

证明: 先证 $A \supseteq B$. 用 $time(OP)$ 表示 OP 的执行时刻.

如图 1 所示, 每条竖线代表一个循环体, 竖线上的点代表操作, 操作序号自上而下递增. 下文与此相同, 不再说明.

设 A 得到满足, 即 OP_j 在 OP_i 之后执行, 则

$$time(OP_j) - time(OP_i) > 0. \quad (1)$$

由于两个体是以锁步方式、一个周期一条指令来执行的, 所以

$$time(OP_i) - time(OP_{i'}) = i - i', \quad (2)$$

$$time(OP_j) - time(OP_{j'}) = j - j'. \quad (3)$$

当 $i - j = i' - j'$ 时, $i - i' = j - j'$, 由式 (2) 和式 (3) 得 $time(OP_i) - time(OP_{i'}) = time(OP_j) - time(OP_{j'})$, 于是 $time(OP_j) = time(OP_{j'}) + time(OP_i) - time(OP_{i'})$. 由式 (1) 得 $time(OP_{j'}) > time(OP_{i'})$. 所以 $OP_{j'}$ 将在 $OP_{i'}$ 之后执行, B 也得到满足. 因此, $A \supseteq B$.

同理可证, $B \supseteq A$. 于是 $A = B$. □

在给定条件下, 图 1 中代表 A, B 的两条有向线段是平行的, 因此, 称上述性质为平移性质. 平移性质说明, 对 VLIW, 任何一个流相关 $(OP_i \rightarrow OP_j, d)$, 都可以写作 $(OP_{i-j} \rightarrow OP_0, d)$, 去掉公共符号, 可以简写为 $(i - j, d)$. 所以流相关的起始操作和终止操作的绝对位置并不重要, 重要的是它们的相对位置. 因此有定义 5.

定义 5. 流相关 $A: (OP_i \rightarrow OP_j, d)$, 可以写成 (λ, d) , 其中 $\lambda = i - j$, 称为流相关 A 的特征值.

下面推导一个重要定理.

定理 1. 设有流相关 $A: (\lambda_a, d), B: (\lambda_b, d)$, 则

(1) $A \supseteq B \Leftrightarrow \lambda_a \geq \lambda_b$; (2) $A = B \Leftrightarrow \lambda_a = \lambda_b$; (3) $A \supset B \Leftrightarrow \lambda_a > \lambda_b$.

证明: 根据平移性质, 可以将流相关 B 平移, 使之与流相关 A 的起始操作相同. 如图 2 所示, λ_a 与 λ_b 存在大于、等于和小于这 3 种关系, 分别对应于 $A \supseteq B, A = B$ 和 $B \supset A$. 证明与平移性质的证明类似, 不再赘述.

由图 2(a) 和 2(b) 有 $A \supseteq B \Leftrightarrow \lambda_a \geq \lambda_b$;

由图 2(b) 有 $A = B \Leftrightarrow \lambda_a = \lambda_b$; 由图 2(a) 有 $A \supset B \Leftrightarrow \lambda_a > \lambda_b$; □

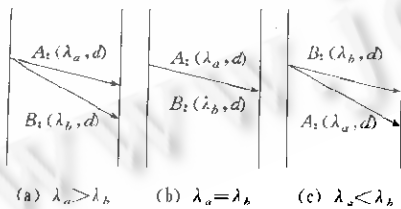


Fig. 2 Illustration of theorem 1

图2 定理1

推论. 如果 $A \supseteq B$ 不成立, 则 $B \supset A$.

证明: $A \supseteq B$ 不成立 $\Rightarrow \lambda_a \geq \lambda_b$ 不成立 $\Rightarrow \lambda_b > \lambda_a \Rightarrow B \supset A$. □

这意味着, 同一体差 d 上的所有流相关, 将按特征值的大小构成线性. 如图 3 所示, 若 $\lambda_n \geq \lambda_{n-1} \geq \dots \geq \lambda_1$, 则 $A_n \supseteq A_{n-1} \supseteq \dots \supseteq A_1$. 因此, 只要保留 A_n , 其余的都可作为冗余删掉. 我们称 A_n 为体差 d 上的特征流相关.

定义 6. 对某一体差 d , 唯一对应着一个特征值, 记为 $\lambda(d)$, 其取值如下: 当体差 d 上无流相关时, $\lambda(d) =$

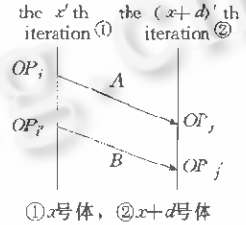


Fig. 1 Translation property
图1 平移性质

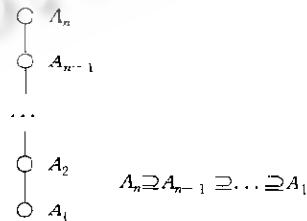


Fig. 3 Inclusion between flow dependences with the same iteration distance
图3 同一体差上的流相关的包含

$-\infty$, 否则, $\lambda(d) = \max\{\lambda_i | (\lambda_i, d) \text{ 是 } d \text{ 上的流相关}\}$. $(\lambda(d), d)$ 称为 d 上的特征流相关.

上面讨论了单体差的情况, 下面讨论多体差的情况.

2 多体差的冗余判定

定义 7. 设有流相关 $A_1, A_2, \dots, A_n, n > 1$, 如果只要满足它们就可以自然满足另一个流相关 B , 则称 A_1, A_2, \dots, A_n 真包含 B , 记作 $A_1 + A_2 + \dots + A_n \supset B$.

定义 8. 设有流相关 A_1, A_2, \dots, A_n 和 $B, n > 1, A_1 + A_2 + \dots + A_n \supset B$, 且对任一流相关 X , 如果 $A_1 + A_2 + \dots + A_n \supset X$, 则必有 $B \supset X$, 那么, 称 B 是 A_1, A_2, \dots, A_n 的最大包含流相关, 记作 $\text{MaxIncluded}(A_1, A_2, \dots, A_n) = B$.

定义 9. 设有流相关 $A_1, A_2, \dots, A_n, n > 1$, 将它们通过平移首尾相连, 则它们一定包含了另一个流相关 B, B 的起点是 A_1 的起点, B 的终点是 A_n 的终点, 称 B 为 A_1, A_2, \dots, A_n 的自然包含, 记为 $\text{NaturalIncluded}(A_1, A_2, \dots, A_n) = B$.

在有多个体差时, 即使每个体差只保留了其特征流相关, 仍可能存在冗余. 如图 4 所示, 设 A, B, C 为不同体差上的特征流相关, 则 $A + B \supset \text{NaturalIncluded}(A, B) \supset C$, 所以 C 可以删去.

最大包含流相关和自然包含流相关具有如下性质:

性质 1. $\text{MaxIncluded}(A_1, A_2, \dots, A_n) \supset \text{NaturalIncluded}(A_1, A_2, \dots, A_n)$.

性质 2. $A_1 + A_2 + \dots + A_n \supset B \Leftrightarrow \text{MaxIncluded}(A_1, A_2, \dots, A_n) \supset B$.

性质 1 和性质 2 可以由最大包含流相关和自然包含流相关的定义直接得出.

下面给出求最大包含流相关的方法.

定理 2. 设有流相关 $A_1, A_2, \dots, A_n, n > 1$, 其中 $A_i: (\lambda_i, d_i), i = 1, 2, \dots, n$, 则 $\text{MaxIncluded}(A_1, A_2, \dots, A_n) = (\sum_{i=1}^n (\lambda_i + 1) - 1, \sum_{i=1}^n d_i)$.

证明: 如图 5 所示. 通过平移, 使 A_1, A_2, \dots, A_n 首尾相接. 假设 A_1 的起点为 OP_k , 则根据特征值的定义, 其终点为 $OP_{k-\lambda_1}$, A_2 的终点为 $OP_{k-(\lambda_1+\lambda_2)}$, \dots , A_n 的终点为 $OP_{k-(\lambda_1+\lambda_2+\dots+\lambda_n)}$. 因为共涉及 n 个体间流相关, 每个相关的终点至少要滞后其起点一个周期, 所以, A_n 的终点至少要滞后 A_1 的起点 n 个周期. 那么, 由 A_n 的终点向上退 $n-1$ 个周期, 在该点处的操作 $OP_{k-(\lambda_1+\lambda_2+\dots+\lambda_n)-n+1}$ 是最早的, 必然在 A_1 的起点 OP_k 后执行的操作. 由这两点所确定的相关是 A_1, A_2, \dots, A_n 所包含的最大流相关.

所以, $\text{MaxIncluded}(A_1, A_2, \dots, A_n)$ 的特征值 $= k - [k - (\lambda_1 + \lambda_2 + \dots + \lambda_n) - n + 1] = (\lambda_1 + \lambda_2 + \dots + \lambda_n) + n - 1 = \sum_{i=1}^n (\lambda_i + 1) - 1$.

$\text{MaxIncluded}(A_1, A_2, \dots, A_n)$ 的体差显然是各 $A_i (i = 1, 2, \dots, n)$ 的体差的和. □

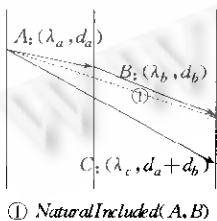


Fig. 4 Redundancy in the case of multiple iteration distances
图4 多个体差时的冗余

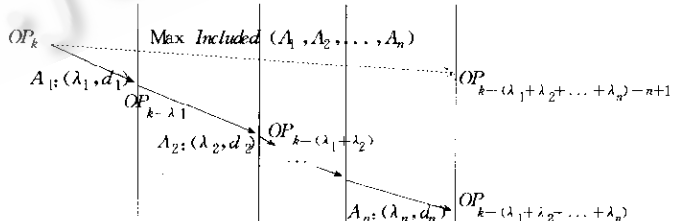


Fig. 5 Compute $\text{MaxIncluded}(A_1, A_2, \dots, A_n)$
图5 计算 $\text{MaxIncluded}(A_1, A_2, \dots, A_n)$

推论 1 (对称性). $\text{MaxIncluded}(A_1, A_2, \dots, A_n) = \text{MaxIncluded}(A_{i_1}, A_{i_2}, \dots, A_{i_n})$, 其中 i_1, i_2, \dots, i_n 是 $1, 2, \dots, n$ 的任一种排列. 即 MaxIncluded 的参数顺序无所谓.

推论 2. 给定流相关 $X: (\lambda_x, d)$ 和 $Y: (\lambda_y, d)$, 以及任意一组流相关 $A_1, A_2, \dots, A_n, n > 0$, 则

- (1) $\lambda_2 \geq \lambda_3 \Leftrightarrow \text{MaxIncluded}(X, A_1, A_2, \dots, A_n) \supseteq \text{MaxIncluded}(Y, A_1, A_2, \dots, A_n)$;
 (2) $\lambda_2 > \lambda_3 \Leftrightarrow \text{MaxIncluded}(X, A_1, A_2, \dots, A_n) \supset \text{MaxIncluded}(Y, A_1, A_2, \dots, A_n)$;
 (3) $\lambda_2 = \lambda_3 \Leftrightarrow \text{MaxIncluded}(X, A_1, A_2, \dots, A_n) = \text{MaxIncluded}(Y, A_1, A_2, \dots, A_n)$.

推论 3. 给定两组流相关 A_1, A_2, \dots, A_n 和 $B_1, B_2, \dots, B_n, n > 1$, 其中 A_i 是体差 i 上的特征流相关, B_i 是体差 i 上的任意一个流相关, $i = 1, 2, \dots, n$, 则 $\text{MaxIncluded}(A_1, A_2, \dots, A_n) \supseteq \text{MaxIncluded}(B_1, B_2, \dots, B_n)$.

消除多个体差上冗余流相关的基本思想, 就是要用若干较小体差上的流相关, 包含一个较大体差上的流相关. 因而只要保留那些较小体差上的流相关即可, 较大体差上的那个流相关就是冗余的(如图 4 所示).

对个体差 d , 我们已经在消除单体差冗余的时候将它上面的所有流相关删减到只剩下它的特征流相关, 假设为 ρ . 根据性质 2, 如果我们能找到一些流相关 A_1, A_2, \dots, A_n , 使 $\text{MaxIncluded}(A_1, A_2, \dots, A_n) \supseteq \rho$, 那么, 只要保留 A_1, \dots, A_n , 就可以删除 ρ , 从而 d 上的所有流相关都被删除.

但是, 我们要寻找的 A_1, A_2, \dots, A_n 并不唯一. 因为同一体差 d 可以分解为若干体差 d_1, d_2, \dots, d_n 之和, 这种分解是不唯一的. 每一组分解 d_1, d_2, \dots, d_n 都对应着一组特征流相关 A_1, A_2, \dots, A_n , 从而对应着一个不同的 $\text{MaxIncluded}(A_1, A_2, \dots, A_n)$.

所以, 为了判断体差 d 上的冗余是否已完全被消除, 需要找到 d 的所有分解, 从而找到所有的 $\text{MaxIncluded}(A_1, A_2, \dots, A_n)$. 这些 MaxIncluded 具有同一体差 d , 因此, 它们之间一定也可以排成线序, 我们定义它们中的最大者(包含其他所有 MaxIncluded 的那个 MaxIncluded) 为体差 d 上的特征最大包含, 记作 Δd .

下面给出多体差时的冗余判定方法.

定理 3. 给定体差 d 上的特征流相关 ρ 以及 d 上的特征最大包含 Δd , 则当且仅当 $\Delta d \supseteq \rho$ 时, 体差 d 上的所有流相关可以被删除.

证明: 所有流相关可以被删除, 即是说特征流相关 ρ 可被删除.

(1) 充分性. 当 $\Delta d \supseteq \rho$ 时, 根据 Δd 的定义, 一定存在某些流相关 A_1, A_2, \dots, A_n , 其中 $A_i: (\lambda_i, d_i), i = 1, \dots, n$, 且 $\sum_{i=1}^n d_i = d$, 使得 $\text{MaxIncluded}(A_1, A_2, \dots, A_n) \supseteq \rho$. 因此, 保留 A_1, A_2, \dots, A_n 即可, ρ 可被删除.

(2) 必要性. 由于 ρ 可被删除, 意味着存在某些流相关 A_1, A_2, \dots, A_n , 使得 $A_1 + A_2 + \dots + A_n \supset \rho$, 其中 $A_i: (\lambda_i, d_i), i = 1, \dots, n$, 且 $\sum_{i=1}^n d_i = d$. 由性质 2, 有 $\text{MaxIncluded}(A_1, A_2, \dots, A_n) \supseteq \rho$. 又由 Δd 的定义, $\Delta d \supseteq \text{MaxIncluded}(A_1, A_2, \dots, A_n)$. \supseteq 有传递性, 所以, $\Delta d \supseteq \rho$. □

因此, 在多体差的情况下, 要简化某个体差 d 上的流相关, 需要求出 Δd . 下面, 我们来推导 Δd 的公式.

定理 4. 给定体差 d , 则 $\Delta d = (\lambda, d)$, 其中 $\lambda = \max\{\lambda_1 + 1 + \lambda_2, \lambda_1 + 1 + \lambda(\Delta(d - d_1)) \mid (\lambda_1, d_1)$ 是体差 d_1 上的特征流相关, $d_1 = 1, 2, \dots, d - 1; (\lambda_2, d_2)$ 是体差 d_2 上的特征流相关, 且 $d_1 + d_2 = d\}$.

证明: 显然, Δd 的体差是 d . 现在要求特征值 λ .

由 Δd 的定义, $\lambda(\Delta d) = \max\{\lambda(\text{MaxIncluded}(A_1, A_2, \dots, A_n)) \mid A_i: (\lambda_i, d_i)$ 是流相关, $i = 1, \dots, n, n > 1$, 且 $\sum_{i=1}^n d_i = d\}$. 定理 2 的推论 3 说明, 当 A_1, A_2, \dots, A_n 都是特征流相关时, MaxIncluded 的特征值 λ 最大. 因此我们可以进一步假定 A_1, A_2, \dots, A_n 都是特征流相关.

由定理 2, $\text{MaxIncluded}(A_1, A_2, \dots, A_n) = \left(\sum_{i=1}^n (\lambda_i + 1) - 1, d \right)$. 于是, $\lambda(\text{MaxIncluded}(A_1, A_2, \dots, A_n)) = \sum_{i=1}^n (\lambda_i + 1) - 1 = (\lambda_1 + 1) + \sum_{i=2}^n (\lambda_i + 1) - 1$.

因此, $\lambda(\Delta d) = \max\left\{(\lambda_1 + 1) + \sum_{i=2}^n (\lambda_i + 1) - 1\right\}$.

当 $n = 2$ 时, $\lambda(\Delta d) = \max\{(\lambda_1 + 1) + (\lambda_2 + 1) - 1\} = \max\{(\lambda_1 + 1) + \lambda_2\}$.

当 $n > 2$ 时, $\lambda(\Delta d) = \max\{(\lambda_1 + 1) + \lambda(\text{MaxIncluded}(A_2, \dots, A_n))\} = \max\{(\lambda_1 + 1) + \max\{\lambda(\text{MaxIncluded}(A_2, \dots, A_n))\}\} = \max\{(\lambda_1 + 1) + \lambda(\Delta(d - d_1))\}$. d_1 是 d 的分解中的一部分, 所以 $d > d_1$, 于是 d_1 的取值范围是

1, 2, ..., $d-1$. 注意, 这是一个递归式.

因此, 对 n 的所有情况, 有 $\lambda = \max\{\lambda_1 + 1 + \lambda_2, \lambda_1 + 1 + \lambda(\Delta(d-d_1))\}$. □

定理 4 给出了计算 Δd 的递归方法. 为提高速度, 下面的算法将递归转化为递推.

3 算 法

首先, 我们对数据相关图上的每个体差单独考虑, 消除单体差上的冗余流相关; 然后将这些体差综合考虑, 进一步消除多体差上的冗余.

定义 $P_{n,m} = [p_{ij}]$, 其中 n 是最大体差, m 是所有体差上的流相关个数的最大值, p_{ij} 是体差 i 上的一个流相关 (ρ_{ij}, i) 的特征值. P 的第 i 行记录体差 i 上的流相关. 当体差 i 上的所有流相关都已填入 P , 但 P 的第 i 行仍未填满时, 将剩余空位都填上 $-\infty$.

令向量 $\Lambda_n = [\lambda_i]$, 其中 $\lambda_i = \max\{\rho_{ij} \mid \rho_{ij} \in P, j=1, 2, \dots, m\}$. (λ_i, i) 就是体差 i 上的特征流相关. 只保留 Λ 中的流相关, 删除其他的, 单体差上的冗余即已消除.

定义向量 $\Delta_n = [\Delta_i]$. 最初令 $\Delta_1 = -\infty$, 其余值待定.

算法如下:

```
procedure Eliminate_Redundancy
```

```
begin
```

```
    Eliminate_Redundancy_For_Each_Iteration_Distance;    {消除单体差上的冗余}
```

```
    Eliminate_Redundancy_For_Multiple_Iteration_Distances {消除多体差上的冗余}
```

```
end;
```

```
procedure Eliminate_Redundancy_For_Each_Iteration_Distance
```

```
begin
```

```
    for  $i=1$  to  $n$  do {对每个体差  $i$ , 消除冗余}
```

```
    [    max =  $P[i, 1]$ ;    index = 1;    {求体差  $i$  的特征流相关, 并删除其他的}
```

```
        for  $j=2$  to  $m$  do
```

```
            if  $P[i, j] > \text{max}$  then
```

```
                [    max =  $P[i, j]$ ;    删除  $(P[i, \text{index}], i)$  所代表的那个流相关;    index =  $j$  ]
```

```
            else
```

```
                删除  $(P[i, j], i)$  所代表的那个流相关;
```

```
                 $\Lambda[i] = \text{max}$ 
```

```
        ]
```

```
    ]
```

```
end;
```

```
procedure Eliminate_Redundancy_For_Multiple_Iteration_Distances
```

```
begin
```

```
     $\Delta[1] = -\infty$ ;
```

```
    for  $d=2$  to  $n$  do    {对每个体差  $d$  求  $\Delta d$  并删除冗余}
```

```
    [    max =  $-\infty$ ;    {按定理 4 求  $\Delta d$ }
```

```
        for  $d_1=1$  to  $d-1$  do
```

```
            [     $x = \Lambda[d_1] + 1 + \Lambda[d-d_1]$ ;
```

```
                 $y = \Lambda[d_1] + 1 + \Delta[d-d_1]$ ;
```

```
                if  $x > \text{max}$  then  $\text{max} = x$ ;
```

```
                if  $y > \text{max}$  then  $\text{max} = y$ 
```

```
            ]
```

```
         $\Delta[d] = \text{max}$ ;
```

```
    ]
```

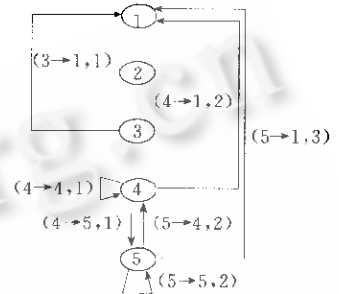
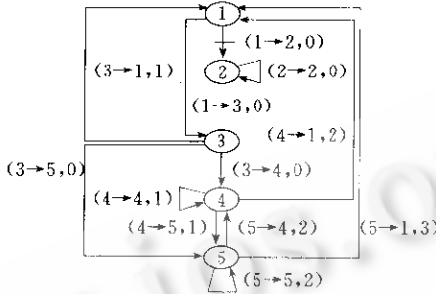
if $\Delta[d] \geq A[d]$ then 删除($A[d], d$) (按定理 3 删除冗余的特征流相关)

end.

这个算法的时间和空间复杂度都是 $O(mn)$.

最后给出一个例子. 如图 6 所示, 循环具有 6 个体间流相关, 经过冗余消除, 只剩下 1 个. 可将算法与图对照理解.

```
FOR I=1 TO N DO
1: X=A[I]+B[I]
2: B[I]=B[I]+1
3: A[I+1]=X
4: A[I+2]=A[I+1]+1
5: A[I+3]=A[I+1]-1
ENDDO
```



(a) The source program

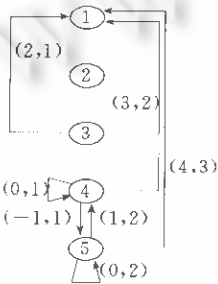
(b) The data dependence graph (DDG) with 7 loop-carried flow dependences

(c) The DDG when considering only loop-carried flow dependences

(a) 源程序

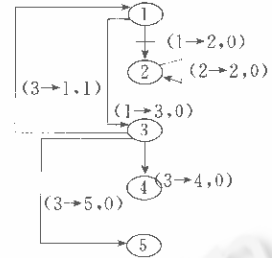
(b) 数据相关图 DDG, 含有 7 个体间流相关

(c) 只考虑体间流相关的 DDG



$$P: \begin{bmatrix} 2 & 0 & -1 \\ 3 & 1 & 0 \\ 4 & -\infty & -\infty \end{bmatrix} \quad A: \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}$$

$$\text{Initial } \Delta^{(1)}: \begin{bmatrix} -\infty \\ -\infty \\ -\infty \end{bmatrix} \quad \text{Final } \Delta^{(2)}: \begin{bmatrix} -\infty \\ 5 \\ 8 \end{bmatrix}$$



(d) Representing the flow dependences in the form of (λ, d)
(d) 用 (λ, d) 形式表示流相关

① Δ 初值, ② Δ 终值.

(e) The matrixes
(e) 矩阵

(f) The simplified DDG with only 1 loop-carried flow dependence
(f) 化简后的 DDG, 含有 1 个体间流相关

Fig. 6 An example of eliminating redundant loop-carried flow dependences
图 6 消除冗余体间流相关的例子

4 实验

随机选取几个程序, 所得实验结果见表 1.

Table 1
表 1

Program ^①	Total (intra-block) loop-carried flow dependences ^②	After redundancy removal ^③	Redundancy rate ^④ (%)
Product	5	2	60
Flower	4	1	75
Sift	5	2	60
Vander	10	2	80
Merge	13	4	69

①程序, ②(基本块内的)体间流相关总数, ③删除冗余后的体间流相关总数, ④冗余度.

其中, 冗余度是指冗余流相关占有流相关的百分比. 由表中可以看出, 平均冗余度约为 69%. 主要的原因是每个体差上有多个流相关, 但最后只保留一个. 这是由 VLIW 特有的锁步性质造成的. 本文旨在提出 VLIW 上流

相关分析的基本性质. 以上实验限于基本块内, 我们将另文介绍如何将本文结果推广到一般.

5 结束语

体间流相关是制约循环调度的根本因素, 然而并非每个体间流相关都需要考虑. 消除那些不必要的相关, 有利于减轻循环调度的负担. VLIW 特有的锁步性质导致了它在数据相关方面具有与众不同的特点. 据此, 本文提出了一种消除冗余的方法, 给出了判定单体差和多体差存在冗余的充分必要条件以及消除冗余的线性复杂度的算法.

虽然我们是以 VLIW 上的循环调度为出发点考虑的, 但是, 从文中的分析过程可以看出, 这种方法实际上是一种一般化的方法, 并不局限于循环, 故可作为 VLIW 上软件流水和多指令流调度的基础. 因此这一方法具有重要的理论和应用价值. 我们将继续深入研究这些方面的应用.

参考文献

- 1 Kai Hwang. *Advanced Computer Architecture: Parallelism, Scalability, Programmability*. New York: McGraw-Hill, Inc., 1993. 457~496
- 2 Psarris K, Kong X, Klappholz D. Direction vector I test. *IEEE Transactions on Parallel and Distributed Systems*, 1993, 4(11):1280~1290
- 3 Rau B R, Fisher J A. Instruction-Level parallel processing: history, over view and perspective. *The Journal of Supercomputing*, 1993, 1(9):9~50
- 4 Lam M. Software pipelining: an efficient scheduling technique for VLIW machines. *SIGPLAN Notices*, 1988, 23(7):318~328

Eliminate Redundant Loop-Carried Flow Dependences for VLIW Architectures

RONG Hong-bo TANG Zhi-zhong

(Department of Computer Science and Technology Tsinghua University Beijing 100084)

Abstract Data dependence is the basic guideline and constraint for parallel processing. The structural distinctiveness of VLIW (very long instruction word), i. e., the lockstep property, leads to remarkable uniqueness in its data dependence analysis, as is shown in this paper. Flow dependences with the same iteration distance comprise a linear ordered set, while characteristic flow dependences with different iteration distances may also have inclusion relationship between each other. From such discovery, a novel method is presented to eliminate redundant loop-carried flow dependences for VLIW architectures, so as to facilitate loop scheduling. The method is complete in that it gets rid of all redundancies. The sufficient and necessary preconditions to identify redundancy for single and multiple iteration distances are given, and a linear ordered algorithm is described. The method is generic, and can serve as a foundation for research on VLIW, such as software pipelining and multi-instruction stream scheduling.

Key words VLIW (very long instruction word), loop scheduling, data dependence, loop, redundancy.