行政院國家科學委員會補助專題研究計畫成果報告

允許非鄰近處理機相連的陣列上資料相衝問題之研究

On the Data Collision Problem of the Processor Array with Non-neighboring Connections

計畫類別：4個別型計畫　　□整合型計畫
計畫編號：NSC 89－2213－E－009－139
執行期間：89 年 8 月 1 日至 90 年 7 月 31 日

計畫主持人：蔡中川
共同主持人：

本成果報告包括以下應繳交之附件：
□赴國外出差或研習心得報告一份
□赴大陸地區出差或研習心得報告一份
□出席國際學術會議心得報告及發表之論文各一份
□國際合作研究計畫國外研究報告書一份

執行單位：國立交通大學資訊工程系

中　華　民　國 90 年 8 月　　　日

行政院國家科學委員會專題研究計劃成果報告

允許非鄰近處理機相連的陣列上資料相衝問題之研究

On the Data Collision Problem of the Processor Array with Non-neighboring Connections

計劃編號：NSC 89-2213-E-009-139
執行期限：89 **年** 8 **月** 1 **日至** 90 **年** 7 **月** 31 **日**
主持人：蔡中川　　交通大學資訊工程系
計劃參與人員：黃為霖、張博揚、邱威傑　交通大學資訊工程系

一、中文摘要

當使用空時映射將均勻相依演算法映射至處理機陣列時，我們必須檢查通道相衝是否發生，以確保映射的正確性。在本計劃中，我們探討允許非鄰近處理機互連之陣列模式上的通道相衝問題。我們的研究成果為提出一新的通道相衝檢查方法。此方法不但適用於鄰近處理機互連之陣列模式，也適用於非鄰近處理機互連之陣列模式。此外，我們也比較了我們的方法與過去的通道相衝檢查方法之間的關係。我們證明了過去之方法皆為我們方法的一個特例。

關鍵詞：通道相衝、非鄰近互連、處理機陣列模式、空時映射、均勻相依演算法

Abstract

When space-time mapping a uniform dependence algorithm into a processor array, link conflicts should be checked to ensure the validity of the mapping. In this project, we have studied the problem of checking link conflicts on the processor array model with non-neighboring connections. A unified method for solving this problem has been proposed. This method is applicable not only to the array model with neighboring connections but also to the array model with non-neighboring connections. Relations between our method and previous methods were also studied. We have shown that all previous methods are special cases of our method.

Keywords: link conflict, non-neighboring connection, processor array model, space-time mapping, uniform dependence algorithm

二、緣由與目的

均勻相依演算法(uniform dependence algorithms) **是一種在科學計算中常見的演算法，如矩陣相乘、**LU **分解、代數路徑問題…等等問題的演算法**[8,9]**；此類型的演算法，一般可由凸形計算定義域**(index set)**與有限數量的常數相依向量**(dependence vectors)**所表示**[6,13,14]**。在將** n **維均勻相依演算法映射至** k ($k<n-1$)**維處理機陣列**(processor array)**的過程中，我們必須避免發生通道相衝**(link conflict)**的情況，以確保均勻相依演算法能正確的在處**

理機陣列上執行。所謂的通道相衝意指兩個以上的資料同時出現在兩處理機間的同一連接通道上。

在通道相衝問題的研究上，最早由 Lee 和 Kedem[10,11]將通道相衝的檢查程序以數學式子正式地表示出來。然而，他們的方法需要列舉計算定義域內 $O(N^{2n})$對計算點(N 為計算定義域之大小)，因此其時間複雜度太高。為了提升檢查通道相衝之效率，在 Lee 和 Kedem 之後，又有 Ganapathy 和 Wah[2,3]、Xue[15-18]、Ke 和 Tsay[7]分別基於輸出入空間(I/O spaces)與混合整數線性規劃(mixed integer linear programming)之觀念，提出了時間複雜度較低的通道相衝檢查方法。然而，上述文獻所提出之通道相衝檢查方法，皆只是針對一簡單的處理機陣列模式所設計：他們皆限制了每條資料通道必須是連接鄰近處理機的資料通道(neighboring link)，也就是說，代表每一資料通道的向量，其內的元素必須是互質。這種對於資料通道的限制雖然簡化了通道相衝問題之討論，但相對地，卻也增加了通道相衝發生之機會[16]，因而限制了處理機陣列上所能得到的空時映射之最佳執行時間[1]。採用允許非鄰近處理機互連之陣列模式(簡稱非鄰近互連陣列模式)能避免上述缺點，但由於非鄰近互連陣列模式上之通道相衝問題較為複雜，因此過去尚未出現適用於非鄰近互連陣列模式上的通道相衝檢查方法。

在本計劃中，我們探討非鄰近互連陣列模式上的通道相衝問題。我們的目的是設計出一個適用於非鄰近互連陣列模式之通道相衝檢查方法，而且，我們希望此方法還必須同時適用於過去所使用的處理機陣列模式(即鄰近互連陣列模式)。另外，我們也探討我們的方法與過去各方法之間的關係。

三、結果與討論

本節簡要描述本計劃之研究成果。更詳細的內容可參考[4]。

3.1 通道相衝發生之充要條件

我們提出了虛擬計算點(virtual nodes)之觀念來處理通道相衝問題。令 J 表一均勻相依演算法之計算定義域，D 表所有相依向量之集合，則虛擬計算點定義為下列形式之分數點(rational vector)：

$$\mathring{\sim} = \mathring{j} + \frac{z}{\cent(\overrightarrow{\ln_i})}\mathring{d}_i, \quad \text{where } \mathring{j} \in J, \mathring{d}_i \in D, z \in \text{w}。$$

在上式中，$\cent(\overrightarrow{ln_i})$為對應於 d_i 之資料通道 $\overrightarrow{ln_i}$ 的連結數(linking number)，其定義為：

$$\cent(\overrightarrow{ln_i}) = \frac{\gcd(\mathrm{S}d_i)}{\gcd(\overrightarrow{ln_i})},$$

其中 S 為空間映射矩陣。我們發現，不同之陣列模式對應到不同連結數所區隔之虛擬計算點。藉由考慮不同連結數所區隔之虛擬計算點，我們可以處理各種陣列模式上的通道相衝問題。經由仔細地研究通道相衝與虛擬計算點之關係，我們推導出同時適用於非鄰近互連與鄰近互連陣列模式上的通道相衝檢查條件：

定理 1 ***空時映射矩陣 T 在資料通道 $\overrightarrow{ln_i}$ 上發生通道相衝若且唯若存在兩個有效(valid)虛擬計算點***

$$\mathring{\sim} = \mathring{j}_1 + \frac{z_1}{\cent(\overrightarrow{ln_i})}\mathring{d}_i, \quad \mathring{\ } = \mathring{j}_2 + \frac{z_2}{\cent(\overrightarrow{ln_i})}\mathring{d}_i,$$

滿足 $j_1 \neq j_2 + zd_i, \forall z \in \text{w}$, ***且使得*** $T\breve{\sim} = T\ \ $。

3.2 通道相衝之檢查

利用定理 1 與使用 Hermite normal form[5,12]之觀念，我們證明了檢查通道相衝等價於檢查一 nJk 維凸形多面體(convex polytope) P 內之整數點。Ke 和 Tsay[7]已發表了一個可檢查 P 內之整數點的方法；他們的方法需要列舉 $O((2N)^{nJk})$個整數

點。我們提出了一個改進的方法來減少 Ke 和 Tsay 之方法所需列舉的整數點數。在我們的方法中，我們利用 Fourier-Motzkin elimination [19]將 P 投影成 $nJkJ1$ 維之凸形多面體 P'，並證明了通道相衝之檢查亦可經由檢查 P 內之整數點達成。因此，我們的方法只需列舉 $O((2N)^{nJkJ1})$ 個整數點；此結果較 Ke 和 Tsay 為佳。

我們的方法可簡要描述成下列四個步驟：

步驟 1：利用 Hermite normal form 演算法計算 P。

步驟 2：利用 Fourier-Motzkin elimination 演算法計算 P 在$[1,0,\ldots,0]^T$ 方向上的投影 P'。

步驟 3：利用線性規劃(linear programming)計算包圍 P 之最小邊界盒(bounding box) C。

步驟 4：列舉 C 內所有整數點，檢查其是否在 P 內且引起通道相衝。

3.3 與其它方法之關係

為了增進對不同之通道相衝檢查方式的瞭解，我們研究了我們的方法與過去各方法之間的關係。我們的結果為證明了過去文獻所提出之所有方法皆為我們的方法之一個特例(即定理 1 中 $¢(\vec{ln}_i) = \mathrm{gcd}(\mathrm{S}d_i)$ 之特殊情況)。

3.4 討論

本計劃中，我們利用虛擬計算點之觀念成功地解決了非鄰近互連陣列模式上之通道相衝問題。我們認為此觀念也可應用至其它更複雜的陣列模式。故本計劃之結果對於未來其它更複雜之陣列模式上的通道相衝問題，提供了一個研究上的參考。

四、計劃成果自評

我們已達到本計劃之預期目標。我們提出了適用於非鄰近互連陣列模式之新通道相衝檢查方法，也證明了新方法之正確性與複雜度。我們也比較了我們的方法與過去各方法之間的關係，並證明了過去方法皆為我們方法的一個特例。

五、參考文獻

[1] J. A. B. Fortes, B. W. Wah, W. Shang, and K. N. Ganapathy. *Algorithm-Specific Parallel Processing with Linear Processor Arrays*, volume 33 of *Advances in Computers*. Academic Press, 1994.

[2] K. N. Ganapathy and B. W. Wah. Synthesizing optimal lower dimensional processor arrays. *In Proc. Int. Conf. Parallel Processing*, pages III.96–III.103, 1992.

[3] K. N. Ganapathy and B. W. Wah. Optimal synthesis of algorithm-specific lower-dimensional processor arrays. *IEEE Trans. Parallel and Distri. Sys.*, 7(3):274–287, 1996.

[4] W. L. Huang. *A virtual node approach to checking link conflicts in the mapping of dependence graphs into processor arrays*. Master thesis, National Chiao Tung University, Taiwan, R.O.C., 2001.

[5] R. Kannan and A. Bachem. Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM Journal on Computing*, 8(4):499–507, 1979.

[6] R. M. Karp, R. E. Miller, and S. Winograd. The organization of computations for uniform recurrence equations. *J. of ACM*, 14(3):563–590, July 1967.

[7] J. Y. Ke and J. C. Tsay. An approach to checking link conflicts in the mapping of uniform dependence algorithms into lower dimensional processor arrays. *IEEE Trans. Comput.*, 48(7):732–737, July 1999.

[8] H. T. Kung. Why systolic architectures? *Computer*, 15(1):37–46, 1982.

[9] S. Y. Kung. *VLSI Array Processor*. Prentice-Hall Int. , Englewood Cliffs, NJ, 1988.

[10] P. Z. Lee and Z. M. Kedem. Synthesizing linear array algorithms from nested for loop algorithms.

IEEE Trans. Comput., C-37(12):1578–1598, December 1988.

[11] P. Z. Lee and Z. M. Kedem. Mapping nested loop algorithms into multidimensional systolic arrays. *IEEE Trans. Parallel and Distri. Sys.*, 1(1):64–76, January 1990.

[12] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, 1986.

[13] W. Shang and J. A. B. Fortes. On time mapping of uniform dependence algorithms into lower dimensional processor arrays. *IEEE Trans. Parallel and Distri. Sys.*, 3(3):350–363, May 1992.

[14] W. Shang and J. A. B. Fortes. Time optimal linear schedules for algorithms with uniform dependencies. *IEEE Trans. Comput.*, 40(6):723–742, Jun 1991.

[15] J. Xue. A new formulation of mapping conditions for the synthesis of linear systolic arrays. In *Int. Conf. on Application Specific Array Processors*, pages 297–308, 1993.

[16] J. Xue. A unified approach to checking data link and computational conflicts in the design of algorithm-specific processor arrays. Technical Report 94-100, Dep. Mathematics, Statistics and Computing Science, The University of New England, Australia, 1994.

[17] J. Xue. Closed-form mapping conditions for the synthesis of linear processor arrays. *Journal of VLSI Signal Processing*, (10):181–199, 1995.

[18] J. Xue and P. Lenders. Avoiding data link and computational conflicts in mapping nested loop algorithms to lower-dimensional processor arrays. In *International Conference on Parallel and Distributed Systems*, pages 567–572, 1994.

[19] G. M. Ziegler. *Lectures on Polytopes*. Springer-Verlag, 1995.