ループ間DOACROSS方式の並列計算機EM-4上での評価

2B－1

山名早人　佐藤三久　児玉祐悦　坂根広史　坂井修一†　山口喜教
電子技術総合研究所　†新情報処理開発機構

1．まえがき

従来，Doall型以外のループを並列計算機上で実行する方式としてDoacross[Cytr86]やPipelining[PaKL80]が提案されている．しかし，これらの方式は，元々，密結合型の並列計算機を対象としたものであり，メッセージ通信によりプロセッサ間のデータ交換を行う粗結合型の並列計算機では，十分な処理性能を引き出すことができない．これは，以下に述べる問題によるものである．ここで，ループの繰り返し回数をNとする．

・Doacrossでは，プロセッサ間の通信ディレイδが(N-1)回分，全体の実行時間に加算されるため，δが十分に小さくないと処理速度の向上が得られない．

・Pipeliningでは，各文の実行時間Tsが(N-1)回分，全体の実行時間に加算されるため，Tsが十分に小さくないと処理速度の向上が得られない．

粗結合型の並列計算機では，メッセージ通信によりプロセッサ間のデータ交換を行うため，δを小さくすることが困難である．また，Tsには，他のプロセッサ間でのデータの入出力時間が含まれるため，Tsを小さくすることも困難である．

これに対して，本報告で提案するループ間Doacrossは，プロセッサ間の通信ディレイδが全体の実行時間に与える影響，及び，各文の実行時間Tsが全体の実行時間に与える影響を小さくする方式である．本方式を用いることによって，粗結合型の並列計算機上で高速なループの実行が可能となる．本方式を並列計算機EM-4[SYHK89]上において，評価した結果，Doacrossの1.9倍，Pipeliningの1.5倍の処理速度向上を確認した．

2．対象ループ

πブロック[BCKT79]に分割されたDoallループ[Poly88](π_p)とDoserialループ[Poly88](π_s)との間に，($\pi_s\ \delta_f\ \pi_p$)あるいは($\pi_p\ \delta_f\ \pi_s$)のデータ依存関係があるとき，これら2つのループを融合してできたループを対象とする．ただし，δ_fはフロー依存を表す．

($\pi_s\ \delta_f\ \pi_p$)や($\pi_p\ \delta_f\ \pi_s$)の関係にあるループは，π_sとπ_pを別々のループとして実行することも可能であるので，別々のループとして実行した場合との比較は次節で行う．

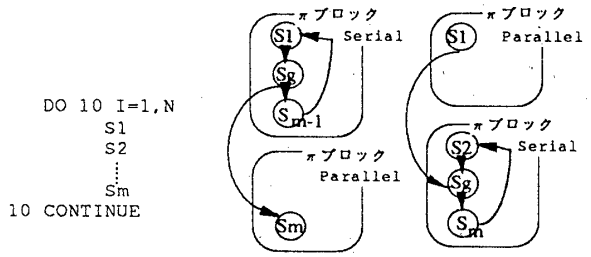


図1　ループ間Doacross対象ループ

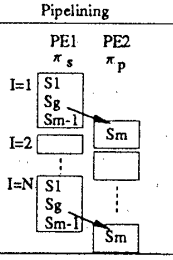
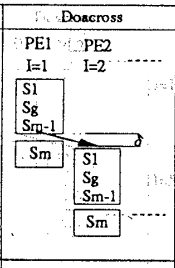
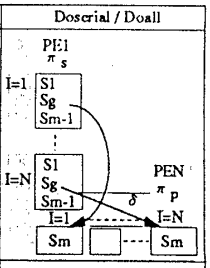
図1に，対象ループを示す．図1に示すように，融合後のループの総文数はm（Doallループ(π_p)が1文，Doserial型ループ(π_s)がm-1文），繰り返し回数はNであるとする．

Doall型ループ(π_p)の文数が1であるのは，πブロックの定義（データ依存関係のある文集合中，最大のデータ依存サイクルを持つ文集合に分割したものをπブロックと定義）より，πブロックであるDoallループは，1文になるからである．また，($\pi_s\ \delta_f\ \pi_p$)や($\pi_p\ \delta_f\ \pi_s$)におけるフロー依存は，単一のフロー依存であるとする．すなわち，ある1つの配列データについてのみフロー依存しているものとする．

3. 従来の実行方式適用時の実行時間

図1に示したループをPipelining，Doacross，そしてπ_sとπ_pをそれぞれDoserial, Doallとして実行した場合の実行時間を比較する．なお以下の議論では，図1に示した2種類の依存関係($\pi_s\ \delta_f\ \pi_p$)及び($\pi_p\ \delta_f\ \pi_s$)の内，($\pi_s\ \delta_f\ \pi_p$)の場合について実行時間を求める(表1)．($\pi_p\ \delta_f\ \pi_s$)についても同様にして求めることができるが，ここでは結果のみを表1の下欄に示す．図1に示すように，π_s内にm-1個の文，π_p内に1個の文が存在するとする．また，π_sからπ_pへのデータ依存は，文Sg∈π_s (1≦g≦m-1) から文Smに対しての1つのフロー依存であるとする．また，文Siの実行開始から文Sjの実行終了までの実行時間をT(Si,Sj)で表すとする．ここで，Doserial／Doallによる実行については，十分な数のPEが存在するとして実行時間を求めている．また，π_sからπ_pへのデータ通信は，π_sでデータが定義された時点で直ちに行われるものとした．

表1　従来の実行方式適用時の実行時間

Pipelining	Doacross	Doserial / Doall
		
●T(S1,Sm-1)≧T(Sm,Sm)時 T(S1,Sm-1)*(N-1)+T(S1,Sg) +Max(T(Sg,Sm-1), δ+T(Sm,Sm)) ●T(S1,Sm-1)＜T(Sm,Sm)時 T(S1,Sg)+δ+T(Sm,Sm)*N	(T(S1,Sm-1)+δ)*(N-1) +T(S1,Sm)	T(S1,Sm-1)*N +T(Sm,Sm) +Max(0,δ-T(Sg,Sm-1))

※($\pi_p\ \delta_f\ \pi_s$)の時の実行時間は，上式において，T(S1,Sm-1)をT(S2,Sm)，T(S1,Sg)をT(Sg,Sm)，T(Sg,Sm-1)をT(S1,Sg)，T(Sm,Sm)をT(S1,S1)で置き換える．

表1に示した実行時間は，文の実行時間を単にT(Si,Sj)で記述しているが，実計算機上でこれらの方式を実現する場合，T(Si,Sj)は，次の3種類の時間から構成される．

(1)加算等の演算時間(te)
(2)配列のアドレス計算時間及び配列データのロード／ストア時間(ta)
(3)他プロセッサへのデータ出力／入力に伴う処理時間(tc)

ループの実行時間は，これら3つの時間te, ta, tcと通信

An Evaluation of Doacross among Loops on the EM-4 Multiprocessor
Hayato YAMANA, Mitsuhisa SATO, Yuetsu KODAMA, Hirohumi SAKANE, Shuichi SAKAI†, Yoshinori YAMAGUCHI
Electrotechnical Laboratory　†Real World Computing Partnership

ディレイδの4つのパラメータに影響を受ける．表1に示した3種類の実行方式を用いた場合について，上記に示した4項目が全体の実行時間に与える影響を比較し，表2に示す．ただし，表2では，簡単のため，T(S1,Sm-1)≧T(Sm,Sm)と仮定する．

表2　te, ta, tc, δが実行時間に与える影響

	Pipelining	Doacross	Doserial / Doall
演算 t_e	O(π_s内演算数×N)	O(π_s内演算数×N)	O(π_s内演算数×N)
アドレス計算 t_a	O(π_s内の配列数×N)	none	O(π_s内の配列数×N)
通信準備 t_c	O(N)	O(π_s内の送受信数×N)	O(N)
通信ディレイ δ	O(1)	O(N)	O(1)

まず，Doacross時のtaを見かけ上noneとすることができる理由を示す．Doacross実行では，表1におけるT(S1,Sm-1)の実行時間が全体の実行時間を決定する主要因となる．従って，できる限りT(S1,Sm-1)を小さくすべきである．ここで，S1～Sm-1実行時には予め定められたアドレス，あるいはレジスタに一時的にデータをストアし，後に(Smを実行する際に)，ストアアドレスを計算しストアを行う．これにより，見かけ上，taの時間をnoneにできる．

表2より，PipeliningやDoserial / Doallによる実行時間は，δの影響をほとんど受けないが，逆に，taによる実行時間がそのまま全体の実行時間に加算される．これに対して，Doacrossによる実行時間は，δの影響が大きいが，逆に，taによる実行時間の増大を回避できる．

このように従来の実行方式は，各々，一長一短を持つ．

4．ループ間Doacross方式

ループ間Doacrossは，πブロックに分割された，Doallループと Serialループの間のデータ依存関係に着目し，これらのループ間にフロー依存が存在する時，これらを融合し一つのループとする．そして，融合されたループをk個のイテレーション毎に部分ループ化し，部分ループを1つのプロセッサに割り当てる．これにより，表2のtaをDoacrossと同様にnoneにできると共に，tcをO(N／k)，δをO(N／k)にすることができる．図2に変換例を示す．

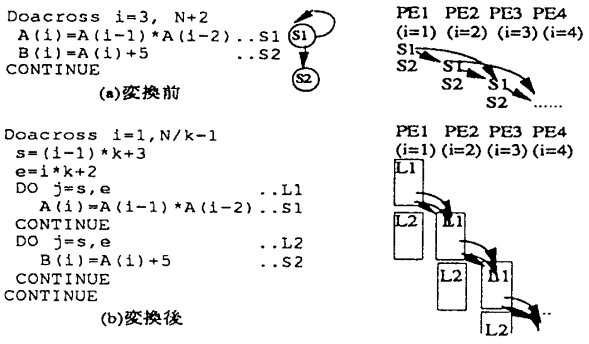


図2　ループ間Doacross方式

図2は，(π_s δ π_p)の依存関係にあるループを融合後，インデックスk個単位(部分ループ化指数と呼ぶ)で内部の2文を部分ループ化したものである．部分ループ化できる条件は，全てのインデックス集合Iに属する任意の部分集合{i}で各々のπブロックを分割した際に，分割された部分πブロック間のデータ依存関係が，分割前後で変わらないことである．例えば，図2の例では，部分ループ化後も(L1 δ_f L2)となりデータ依存関係が保たれる．ループ間Doacross化後の実行時間は，図1の例を用いると，

T(S1,Sm-1)×N＋δ×N／k＋k×T(Sm,Sm)

となり，表1のDoacrossの実行時間に比較して，δによる実行時間の増大が，1／kになり，最後の項T(m,m)の係数がk倍になっていることがわかる．すなわち，kが増大するにつれ，ループ間Doacrossの実行時間は減少し，その後増大する．これは，k×T(S1,Sm)の項がkの増加と共に増加するからである．従って，ループ間Doacrossを用いて実行時間を短縮するためには，kの決定が重要な要素となる．

5. ループ間Doacrossの評価

並列計算機EM-4[SYHK89]上で，図2のループを用いて本方式の評価を行った．プログラムは，アセンブラで直接記述し，繰り返し回数N=64とした．各々の実行方式で図2のプログラムを実行した時の実行時間をkをパラメータとして図3に示す．使用したPE台数は次の通りである．Doacrossが4台，Pipeliningが2台，Doserial / DoallのDoserial部分が1台，Doall部分が64台，ループ間Doacrossでは，k=16までが4台，k=32の時2台，k=64の時1台である．

図3に示すように，kの値によって，ループ間Doacross時の実行時間が変化することがわかる．本例の場合，k=16(PE4台)の時に実行時間が最小となり，Doacrossの1.9倍，Pipeliningの1.5倍の処理速度向上を確認できる．また，Doacrossの実行時間は，PE1台で逐次に実行する場合に比較して3%程度増大している．これは，Doacrossでは，通信ディレイδがNに比例して実行時間に加算されることにより，Doacrossの効果を消滅させてしまったためである．

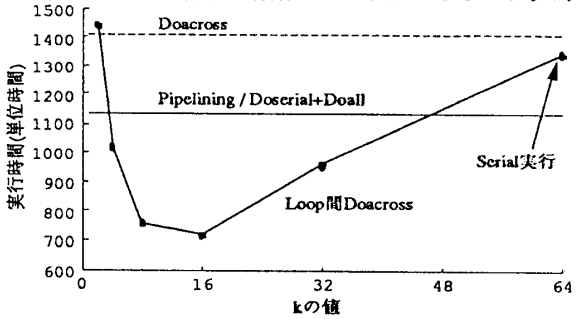


図3　各実行方式による実行時間比較

6．まとめ

本報告では，ループ間Doacrossを提案し，通信ディレイδ及び各文の実行時間Tsが全体の実行時間に与える影響を小さくし，相結合型の並列計算機上でのループの実行時間を短縮できることを示した．サンプルプログラムによる評価の結果，Doacrossの1.9倍，Pipeliningの1.5倍の処理速度向上を確認した．

今後は，対象とする並列計算機が持つパラメータを元に，最適なkをコンパイル段階で決定する方法を考えていく予定である．

謝辞

本研究を遂行するにあたり御指導，御討論いただいた太田情報アーキテクチャ部部長ならびに計算機方式研究室の同僚諸氏に感謝いたします．

文献

[BCKT79] U.Banerjee, S.Chen, D.J.Kuck, R.A.Towl. "Time and Parallel Processor Bounds for Frortan-like Loops," IEEE Trans. on Comp., C-28, 9, 660-670 (1979).

[Cytr86] Ron Cytron:"Doacross: Beyond Vectrization for Multiprocessors", Proc. of Int. Conf. on Parallel Processing '86,pp.836-844(1986).

[PaKL80] D.A.Padua, D.J.Kuck, D.H.Lawrie. "High-Speed Multiprocessors and Compilation Techniques," IEEE Trans. on Comp., C-29, 9, 763-776 (1980).

[Poly88] C.D.Polychronopoulos:"Parallel Programming and Compilers," Kluwer Academic Publishers (1988).

[SYHK89] S.Sakai,Y.Yamaguchi,K.Hiraki,Y.Kodama,T.Yuba:"An Architecture of a Dataflow Single Chip Processor", Proc. of 16th Ann. Symp. on Comp. Arch., pp.46-53(1989)