密結合型並列計算機における配列の配置に関する研究

1Q-1

小池　稔　大川　善邦

（大阪大学　工学部　電子制御機械工学科）

1．　はじめに

従来，並列化コンパイラのコンパイリング手法に関する研究は数多くなされてきた．それらはプロセッサのタスク配置に関する問題が主流であった．[1]これに対して複数のタスク（またはプロセッサ）からアクセスされる配列データを，並列計算機のデータメモリにどの様に組み合わせて配置するか，という問題がある．組合せを考える理由は次の通りである．

① １つのタスクがアクセスするデータメモリの個数が少ない方が，アクセスの対象となるメモリへの接続切り換え回数が少なくなる．

② １つのデータメモリに配置される配列データの個数が少ない方が各タスクからのアクセス競合の可能性を軽減することができる．

そこで本研究ではソースプログラムから組合せ配置の為の情報を獲得し，次にそれに基づいた配列の組合せ配置の方法を提案し，更にその有効性に対する検証を行う．今回は後者の配置方法について報告する．

2．　配列の組合せ配置方法

2．1　環境

本研究では次のような環境を仮定している．

① データメモリを備えた密結合型並列計算機上で実行されるプログラムをコンパイルする．

② データメモリ群とプロセッサ群との接続形態は完全結合網である．

③ 各配列に対する各タスクのアクセス状況（変数のバイト長，アクセス回数，アクセス頻度，配列の大きさ）は予め調査されている．

2．2　記号の説明

ここでは次節以降で用いる記号について説明を行う．

a	：配列の個数	（a＞1）
t	：タスクの個数	（t＞1）
m	：データメモリの個数	（m≧1）
g	：変数のグループ分け個数	（g≦a）
Aj	：配列	（1≦j≦a）
Tr	：タスク	（1≦r≦t）
Mi	：データメモリ	（1≦i≦m）

Cjr：AjがTrにアクセスする回数（例：表１）

$Sjk=\sum_{r}|Cjr-Ckr|$

：配列の相互疎遠度（例：表２）

$Xj=\sum_{r}Cjr$

：Cjr表の横合計：Ajへのアクセス回数

Xmax：Xjの最大値（表１の場合X＝２０）

$\Delta x=max\{(Xj+Xk)-Xmax,\ 0\}$

：AjとAkを組み合わせたときの総演算時間の変化量の代表値（演算時間がメモリアクセス回数に比例すると仮定する．）

$Yr=min\{Cjr,\ Ckr\}$

$\Delta y=-\{\sum_{r}Yr\}$

：接続切り換え回数変化の代表値

2．3　組合せ配置アルゴリズムの概要

組合せ配置アルゴリズムの概要は以下の通りである．

① m＝１の時，全ての配列をM1に配置して終了する．

② m≧２の時，g＝a－１（≧m）とする．

③ Cjr表からSjk表を作成する．

④ Sjk表中から値が小さい組合せから順に並べる．

⑤ AjとAkを組み合わせて配置したときのΔx，Δyを求める．

⑥ Δx毎，Δy毎に小さい組合せから順位をつける．

⑦ Δx，Δyの各順位の和をとり，その値が最も小さい組合せを採用する．

⑧ g＝mならば終了する．

⑨ Aj，Akに対して，Cjr←Cjr＋Ckr，g←g－１として③へ戻る．

3．例題とシミュレーションによる検証

3．1　例題

表１に示すアクセス状況に対して，２．３のアルゴリズムを適用すると，以下のような結果が得られる．

① g＝４のとき，Sjk は表２の通りである．表２によって選ばれた組合せに対するΔx，Δy及びその順位を表３に示す．これにより，A1＋A2 を採用する．

以下同様にして

② g＝３のとき，A3＋A5を採用する．

③ g＝２のとき，（A1＋A2）＋A4を採用する．

以上をまとめると各gでの組合せは

Ⅰ g＝４のとき・・（A1＋A2），A3，A4，A5

Ⅱ g＝３のとき・・（A1＋A2），（A3＋A5），A4

Ⅲ g＝４のとき・・（A1＋A2＋A4），（A3＋A5）

3．2シミュレーションにおける仮定

① あるタスクTrの，あるメモリMiに対するアクセスを１回の処理とする．

Array Allocation Method on Tightly Coupled Multiprocessors
Minoru Koike,Yoshikuni Okawa
Osaka University

Cjr		Tr					Xj
		1	2	3	4	5	
Aj	1	4	3	0	0	0	7
	2	2	1	0	0	0	3
	3	0	0	8	5	7	20
	4	0	0	4	9	1	14
	5	0	0	3	2	6	11

表１　Ｃjr表の例

Sjk		1			Ak
	2	4	2		
Aj	3	27	23	3	
	4	21	17	14	4
	5	18	14	9	13

表２　Ｓjk表の例

A						Δx		Δy	順位
j	k	Sjk	Xj	Xk	Xj+Xk	Δx	順位	Δy	順位	の和
1	2	4	7	3	10	0	1	-3	4	5
3	5	9	20	11	31	11	4	-12	1	5
4	5	13	14	11	25	5	3	-6	3	6
3	4	14	20	14	34	14	5	-10	2	7
2	5	14	3	11	14	0	1	0	5	6

表３　表１の例による結果

番号	組合せ	番号	組合せ	番号	組合せ	番号	組合せ
0	A12345	13	A11344	26	A11145	39	A12112
1	A11345	14	A12125	27	A11315	40	A12121
2	A12145	15	A12142	28	A11341	41	A12211
3	A12315	16	A12144	29	A12115	42	A12221
4	A12341	17	A12215	30	A12141	43	A12212
5	A12245	18	A12312	31	A12311	44	A12122
6	A12325	19	A12313	32	A12225	45	A11333
7	A12342	20	A12241	33	A12242	46	A11115
8	A12335	21	A12321	34	A12322	47	A11141
9	A12343	22	A12331	35	A12333	48	A11311
10	A12344	23	A12244	36	A11144	49	A12111
11	A11335	24	A12323	37	A11313	50	A12222
12	A11343	25	A12332	38	A11331	51	A11111

表４　配列の組合せ番号表
（書式はＡＡ1A2A3A4A5となっている．Ａj＝Ａk（j≠k）となっていれば同じデータメモリに配置する．例"A12313"の場合：Ａ1とＡ4，Ａ3とＡ5をそれぞれ組み合わせることを示す．）

② メモリアクセスの直後にＴrは内部演算を行う．

③ Ｍiにアクセスするタスク（Ｔr）が，以前にＭiをアクセスしたタスク（Ｔp）と異なる場合（r≠q），またはＴrがアクセスするメモリ（Ｍi）が，以前にＴrがアクセスしたメモリ（Ｍh）と異なる場合（i≠h）は接続切り換えを行う．

④ Ｃjr表内の数字はＴrのＭiへのアクセス回数とし，１回の処理を行う毎にＣjr←Ｃjr－１とする．表内の数字が全て０になったときシミュレーションを終了する．

３．３　シミュレーションとの比較

図１に総演算時間の変化量，図２に接続切り換え回数の変化量を示した．２．３節のアルゴリズムによるΔｘ，Δｙの変化と，定性的には一致している部分が多い．

４．　おわりに

現在，実際のプログラムに対して２．３節の組合せアルゴリズムを適用し，妥当な組合せが得られるかを検討している．今後は組合せアルゴリズムを相互結合網用からマルチバス用への修正や，組合せのためのソースプログラムからの情報読み取りのアルゴリズムについても検討する予定である．

〔参考文献〕

1）坂井修一：並列計算機におけるスケジューリングと負荷分散，情報処理，Vol.27,No.9,pp.1031～1036（1986）．

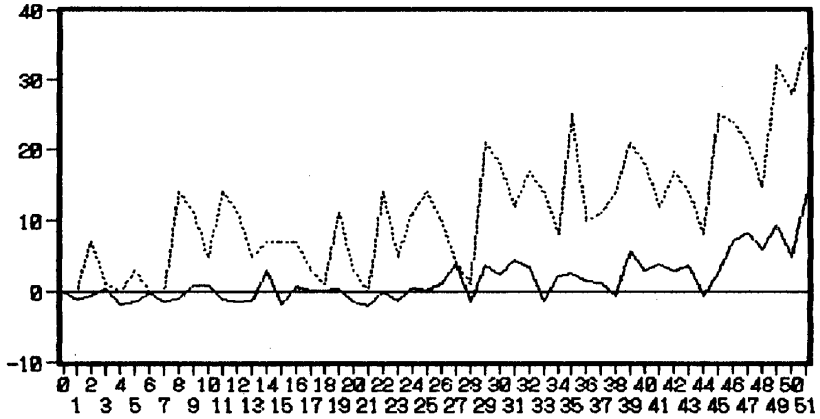

図１　総演算時間の変化量
（横軸：配列の組合せ番号，縦軸：総演算時間の変化量）
（上側点線：Δｘ，下側実線：シミュレーションによる結果）

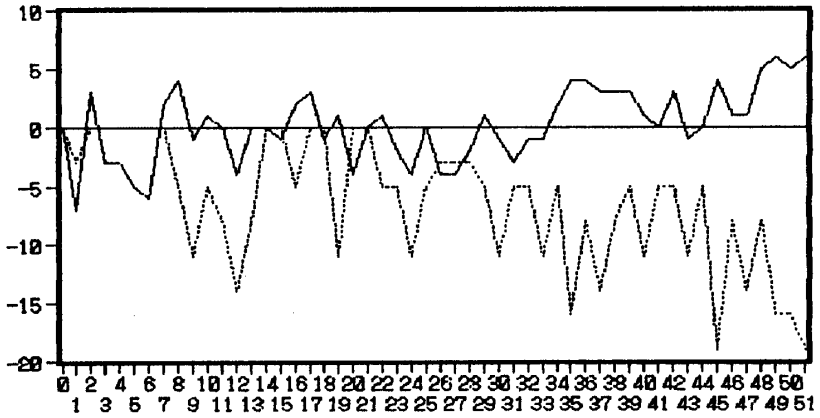

図２　接続切り換え回数の変化量
（横軸：配列の組合せ番号，縦軸：接続切り換え回数の変化量）
（下側点線：Δｙ，上側実線：シミュレーションによる結果）