数理解析研究所講究録
613巻 1987年 60-73

60

高並列コンピュータと数値アルゴリズム

── PAXにおける最近の経験 ──

筑波大・構造　星野　力 (Tsutomu Hoshino)

並列コンピュータのアーキテクチュアに関しては，従来よりいくつかのアプローチ（二陣営対立? ）がある．例えばSIMDかMIMDか，PIPELINEかARRAYか，命令フローかデータフローか，低並列度か高並列度か，といったアプローチがあるが，プログラミングとアルゴリズムに決定的な差異を生じさせるのは，分散メモリ構成か集中メモリ構成かという点である．図1．集中型はどのプロセッサからどの変数へも自由にアクセスでき，ユーザはあまりアーキテクチュアを意識しなくてもよいが，アクセス集中によるネットワークの遅れや飽和など技術的問題が多い．分散型はローカルメモリへのアクセスは効率よく出来るが，他のプロセッサ／メモリへのアクセスにはアーキテクチュアを意識せねばならない．

技術の趨勢としては，VLSI技術の進歩により，ハードウェアの価格が低下し，分散できる仕事は分散したほうが経済性

がよくなってきている．汎用的な最高速のプロセッサを用いる方式は，一部のどうしてもそれを必要とするユーザに限られ，パーソナルコンピュータやワークステーションに見られるような中速のプロセッサを，分散結合する方式にとってかわられるであろう．台数＝並列度も，現在の10程度から100以上，さらにVLSI技術の許すまで拡大し，間もなく高並列時代がやってくるであろう。では，高並列時代の数値アルゴリズムはどうあるべきか?

1．コンシステントな並列アルゴリズム

まず大規模計算のための並列アルゴリズムは，Lambiotte, Jr.の意味においてコンシステント(Consistent)でなければならない．プロセッサ台数を一定とし，問題のサイズNに関して，並列処理アルゴリズムの演算回数と逐次処理のそれがオーダ的に等しいとき，並列処理アルゴリズムはコンシステントであるといわれる．(J.Lambiotte Jr., The Solution of Tridiagonal Linear Systems on the CDC STAR-100 Computer, ACM Trans.on Mathematical Software, Vol.1, No.4, Dec. 1975.)

すなわち

$$O(T_1(N)) = O(T_P(N)),$$

ただし

$T_1(N)$: 1台のPU(Processing Unit)による逐次処理時間,

$T_P(N)$: P台のPUによる並列処理時間,

N : 問題の規模,

である. コンシステントでない並列アルゴリズムは, 大規模計算($N \to \infty$)において逐次処理よりも遅くなり実用的でない.

例として各PUに分散したデータの並列加算(Cascade Sum)を考えよう.

逐次処理では $N-1$,

逐次Cascade Sumは $N\log_2 N$,

並列処理(Non-consistent Cascade Sum)は

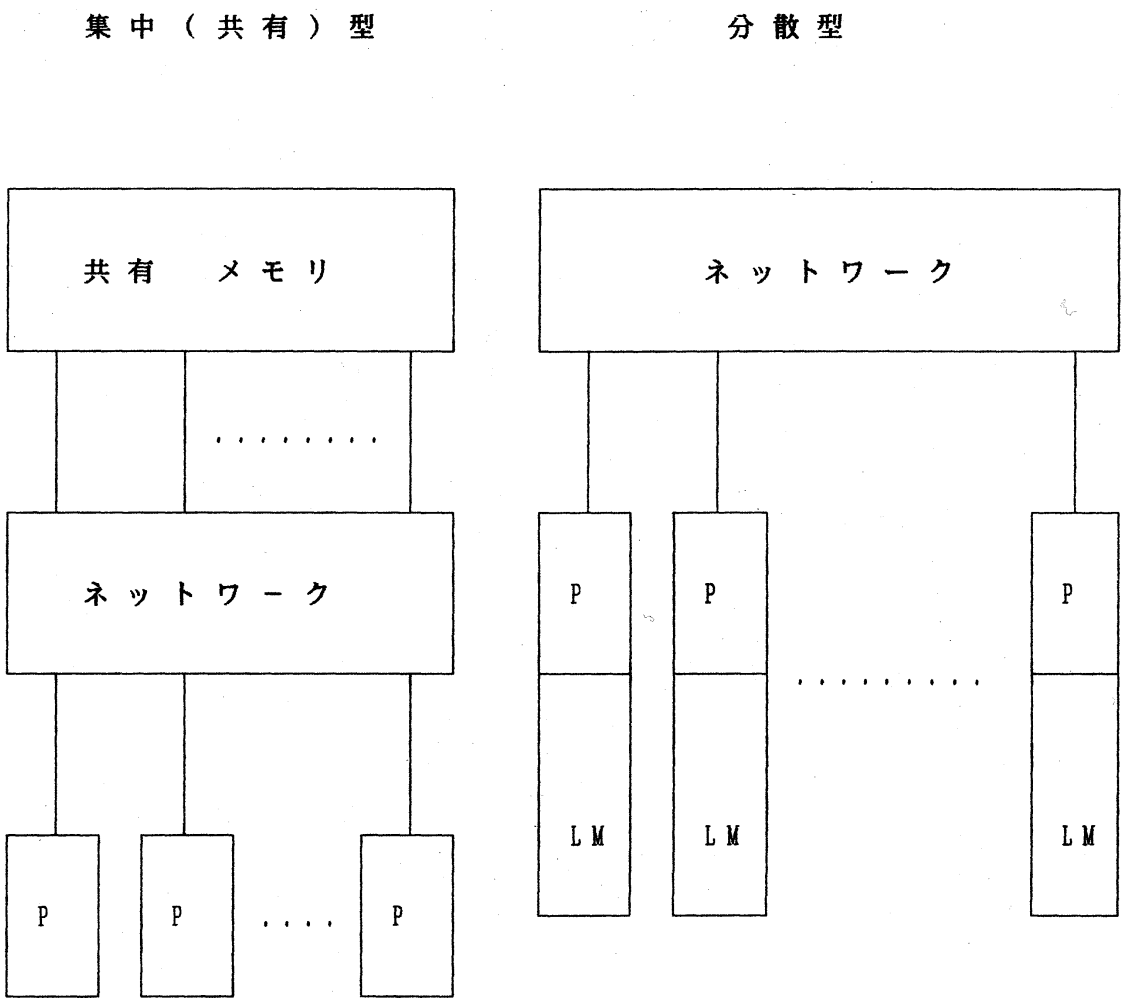
$$(N\log_2 N)/P + N - 1,$$

並列処理(Consistent Cascade Sum)は

$$N/P - 1 + \log_2 P + P - 1,$$

となり並列加算アルゴリズムはコンシステントに実行できる. PAXの場合, 前提として(データ数N)/(PU台数P)≧1であり, N/P - 1はPU中のSerial Sum(部分和), $\log_2$ PはP個のデータのP台のPUによるCascade Partial Sumすなわち $((P \log_2 P)/P)$, P - 1はPU間データ転送である. 加算+の代わりに, 任意の2項演算*, >, <, などにも拡張できる.

図 1　集中型と分散型並列計算機アーキテクチュア.



P　：　プロセッサ

LM　：　ローカル　メモリ

2. 並列処理による高速化

並列処理により高速化する条件とは何であろうか. 逐次処理時間のステップ数がaNであるようなアルゴリズムがあったとしよう. これをP台のPU上で並列化したとき, b/Pの複雑度をもつ部分をf(N)ステップにわったて処理しなければならないとしよう. 図2. このとき, 並列処理が逐次処理を越えるためには, Speedup Sが $S = aN/(bf(N)/P) > 1$

したがって, マシン並列度 $P > bf(N)/(aN)$

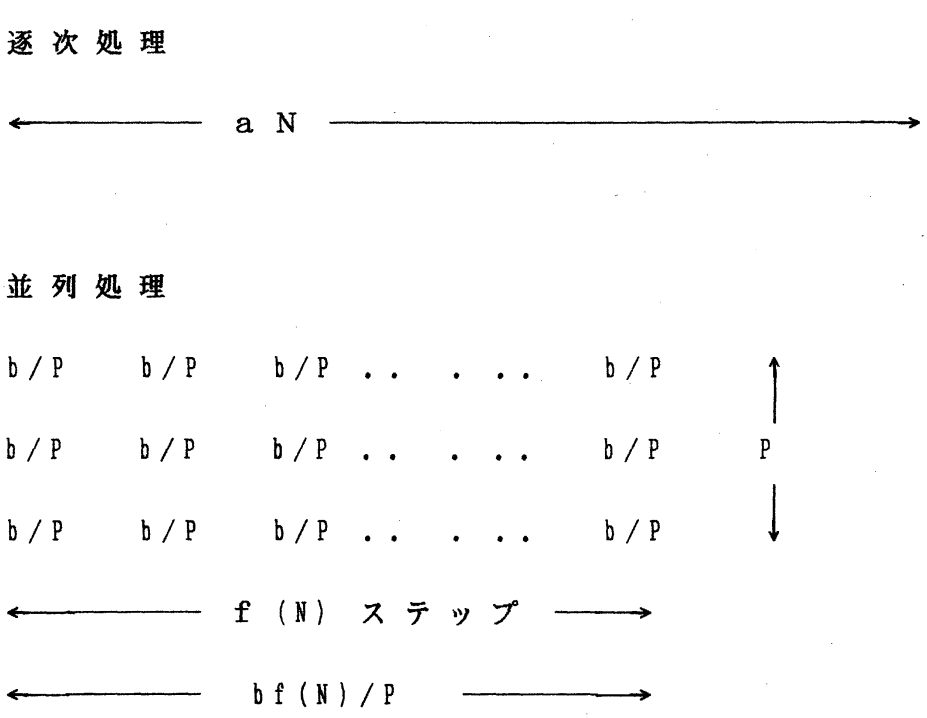
が必要である。ここに最終的に実現する並列度は, アルゴリズムのもつ並列度とマシンが実現する並列度の小さいほうで決まることに注意せねばならない.

先ほどのCascade 加算の例では, $a=b=1$, $f(N)=N\log_2 N$, であり $\log_2 N$ は並列処理アルゴリズム故の冗長度と解釈できる。$N=128$ のとき冗長度 $\log_2 N=7$ は, パイプラインベクトルマシンの持つ並列度, すなわちパイプラインの段数とほぼ同程度である。マシン並列度が低いことがベクトル計算機でCyclic Reductionなどのアルゴリズムの速度が上がらない真の理由であり, アルゴリズムのせいではない. 現にPAXではCyclic Reductionは十分実用的である.

一般に並列アルゴリズムには逐次処理には存在しない冗長

度を含み，これを圧倒するマシンの並列度がないかぎり，並列処理により速度が向上することは期待できない．

図２　逐次処理アルゴリズムの並列化．



3．三項対角方程式と高速POISSON SOLVER

PAXにおいて成功したコンシステントなアルゴリズムの一例として，二次元Poisson方程式求解ルーチン FAGECR(1)がある．[佐藤，星野，並列ポアソンソルバーFAGECR，情報処理学会第33回全国大会2X-7]．FAGECR(Fourier Analysis Gaussian

Elimination Cyclic Reduction with 1 stage Block CR)は，HockneyのFACR(L)をPAX上に実装したものである．現在FAGECR(1)はPAXに実装されたPOISSON SOLVERのうちで最高速であり，128X128点問題を約1秒（効率88.7%）で解く．これはDOUBLE FFT法の約3倍，対称SOR法でpreconditionしたpCG法の約12倍の速度である．

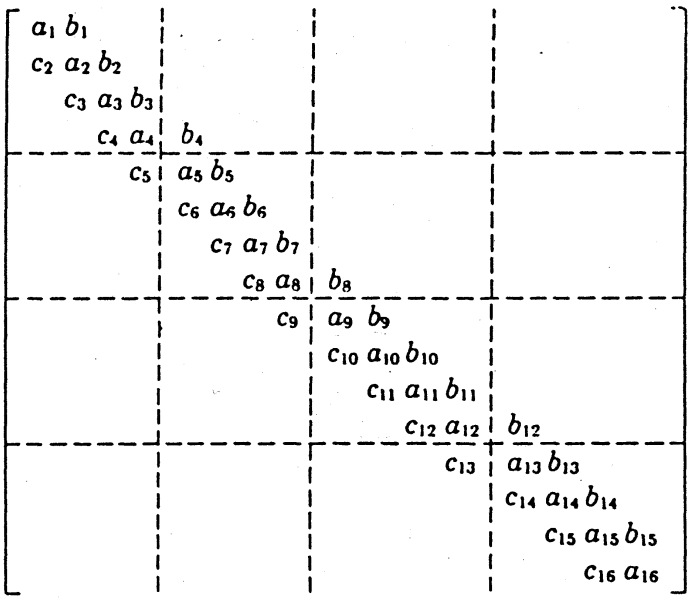
この解法の一番内側のループにおいて，三項対角一次連立方程式をリング接続されたPUアレイで解く必要があり，並列Cyclic ReductionとGauss消去を組み合わせたGECR法を使っている．GECR法は偏微分方程式のADI法をPAXに実装したときに用いたものと同じものである．[上村，星野，並列計算機PAXによるADI法の実行と評価，情報処理学会論文誌，第26巻第1号，昭和60年1月，19-24頁]．GECR法はNavier-Stokes方程式の陰解法Beam-Warming法をPAXで実行したときにも効率のよい並列化をもたらした．[Tsutomu Hoshino, Takeshi Kamimura, Toshihiro Iida, and Tomonori Shirakawa, Parallel ADI Scheme Using GECR(Gauss-Elimination-Cyclic-Reduction) Method and Implementation of Navier-Stokes Equation in the PAX Computer, Proc. 1985 International Conf. on Parallel Processing, IEEE (1985) pp. 426-433.]

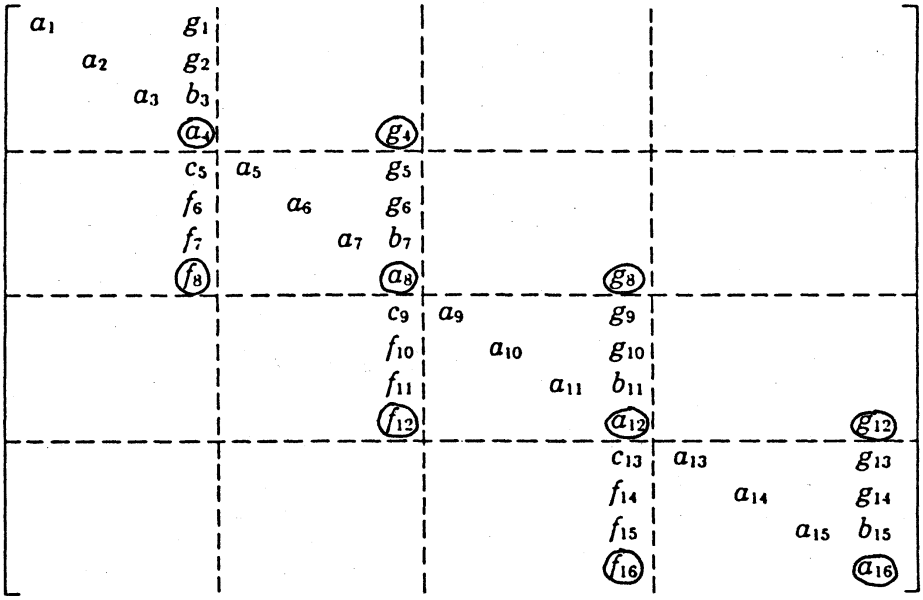
GECR法は，三項対角方程式をいくつかのブロックに分割し

て，各ブロック内部で内点のGAUSS消去を行い残りの境界点間の三項対角式を得，これをCyclic Reduction(PARACR)で解き，消去した点は各ブロックにおいて後退代入により求めるものである．PAXではブロックはPUに対応させるのが自然である．詳しい内容はこれらの論文を参照願うとして，ここではWangによる三項対角方程式の並列アルゴリズムとの関係を述べる．

Wangはベクトルおよびパラレルプロセッサ（ただし遅れなしの共有メモリ，または転送オーバヘッドなしのプロセッサ間結合を持つ無限個のプロセッサ）上で有効な並列解法を提案している．[H.H. Wang, A Parallel Method for Tridiagonal Equations, ACM Transactions on Mathematical Software, Vol.7, No.2,(June 1981) pp.170-183]．まず図３上のような三項対角行列があったとしよう．これを４個のブロックに分割し，(1)各ブロックでGAUSS消去を行うと，図３下のようなひげ(fill-in)が生ずるが，(2)各ひげを順次消去していくと対角項のみになり前進消去が完成する．(3)後退代入は各ブロックでおこなう．(1)と(3)のステージにおける並列度またはベクトル長は分割ブロック数であり，(2)のそれはひげの長さである．両者は反比例の関係にあり（分割数とひげの

図 3　三項対角行列のブロックGAUSS消去





長さの積は行列のサイズ＝一定である），結局最適な分割はNの平方根となる．ところで図３下の丸で囲った要素に注目すると，これらはまた三項対角行列をなしているから，これを逐次的に解くこともできる．[J.S. Kowalik, S.P. Kumar, Parallel Algorithms for Recurrence and Tridiagonal Equations, Parallel MIMD Computation, Kowalik ed., pp. 295-307, MIT Press (1985)]．

PAXのGECR法は，複雑な構造物の解析手法の一つである部分構造法からヒントを得てブロックを部分構造として捕らえる．このとき上記のブロック内部のGAUSS消去は部分構造の内点消去である．図３下の丸で囲った要素は部分構造の境界点にあたる．境界点に関する三項対角方程式はCyclic Reductionで解くのでこの部分のオーダはWangやKowalikの逐次処理よりは低い．最高次のオーダは内点GAUSS消去であるからこのアルゴリズムはコンシステントである．（Wangの方法もKowalikの方法もコンシステントである．）

ところで内点が１個になったら，このGECR法はただのCR法になってしまう．しかし流体解析のように大規模計算のニーズが大きい分野では当分の間，内点数は十分大きいと期待できる．このようにして，三項対角方程式は高並列コンピュータにとってネックではなくなったので，Navier-Stokes方程式

の解法に内在する高並列度を高並列コンピュータにより解き放つことができる．NS方程式は非線形移流項の存在のため，まず線形化し（これは各点したがって各プロセッサで並列処理できる），連立方程式であるため方程式間で対角化（これも各プロセッサごとに並列処理できる）せねばならない．問題のサイズが大きくなるとこの部分を高並列処理できる高並列コンピュータが，現在主流になりつつあるマルチスーパコンピュータのような低並列度のコンピュータに優ることになるであろう．

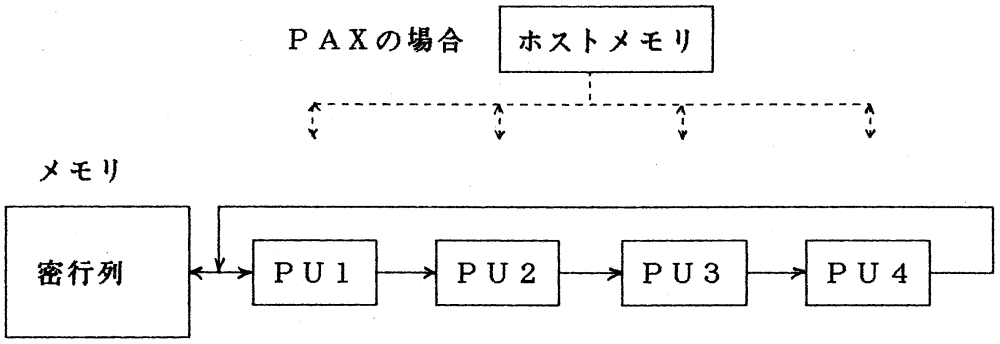
目下このFAGECR(1)をFAGECR(L)へ拡張中である．二次元物理空間を同じく二次元PUアレイ上へ写像すると，二つの方向へデータ転送が生じオーバヘッドが大きくなることが予想される．メモリさえ十分にあれば，むしろ大規模計算を想定して一次元PUアレイへ写像したほうがよいだろう．これは目下作業中である．

4．帯行列連立一次方程式のSYSTOLIC的解法．

帯行列連立一次方程式はリング状に結合されたPUアレイにより効率よく解けることはいまや常識である．［シストリックアレイのためのアルゴリズムは多く発表されているが，例え

ば A.Sameh, On Some Parallel Algorithms on a Ring of Processors, Compter Physics Communications, Vol.37 (1985) pp.159-166]. PAXもリングアレイであるからこれらのアルゴリズムは実装できる. 上記論文は密行列の解法例で, 図5にN=9の場合を示す.

図5　リングアレイによる密行列のGAUSS消去.



タイムチャート

	PU1	PU2		
	1			
	21			
	31	2		
	41	32		
	51	42	3	
	61	52	43	
	71	62	53	4
	81	72	63	54
	91	82	73	64
	◆1	92	83	74
↓	65	◆2	93	84
	75	−−	◆3	94
時間	85	76	−−	◆4
	95	86	−−	−−

1：1行目のLOADING
21：1行をpivotとする
2行目の消去.
◆1：1行目のSTORE

各行列要素単位に操作が終れば次のPUへ送られる．この解法はもちろん帯行列にも適用できて，約半帯幅の75%の並列度が期待できる．目下これも電力系統の安定度解析に用いることを想定して作業中である．

5．おわりに

今までのPAX開発を通じて経験したことから，科学技術計算のための並列計算機のアーキテクチュアとアルゴリズムについて，現時点では以下のような見解を持っている．

1．　やはり，アルゴリズムの冗長性を打ち消すために高並列でなければならない．（ハードウェア技術もその方向を向いている．）

2．　分散メモリ構成ではデータ転送がネックにないかねないので，高速のプロセッサ間通信が必要である．（最近出現しているパッケト通信疎結合方式は数値計算には使えないのではないかと懸念される．集中メモリ方式で高並列にするには従来より困難があり，最近もNETWORKの飽和等の悲観論もある．しかしまだ結論は出ていない．）

今後の課題は多いが，現状は人手不足で既製の線形アルゴリズムの並列化に追われている．いつも叫ばれているように，

並列処理特有の新アルゴリズムとそれに適した新アーキテクチュアの研究がなされねばならない．また単に数値アルゴリズムだけを考えることはもはや不可能で，計算機のアーキテクチュアとの接点すなわち，データやタスクをどのプロセッサとメモリに置くかという写像（マッピング）が重要である．

最後に一言．高並列アルゴリズム研究は外国とくにアメリカにおいて盛んで，多くの並列コンピュータがマーケットにでてきており，これらは並列計算の研究のためのTOOLとして買われている．一方日本では極めて対照的に，研究的な意味ではただ退屈なだけのベクトルスーパコンピュータの研究発表が学会を賑わしている．

われわれのPAXコンピュータは新しいハードウェアを設計中であり，これが完成すると古いPAX-128は，ユーザがいないので多分廃棄することになろう．今まで外部のユーザで唯一PAXを使って共同研究をしたのは，日本人ではなく米国ロスアラモス研究所のユーザであった．日本のユーザは体制的な（例えば確立されたメインフレームやスーパーコンピュータしか相手にしないといった）思考方法を改めないと，日本の数値技術はいつまでたっても後進性を脱却できないであろう．