計算機アーキテクチャ 73－4
（1988. 10. 31, 11. 1）

並列処理システム－晴－の ソフトウェアシミュレータによる評価

山名早人 萩原 孝 草野義博 村岡洋一

早稲田大学 理工学部

－晴－の機能レベルのソフトウェアシミュレータを用いて、データフロー実行における待ち合わせ記憶のバンク化と－晴－の大域的メモリである共有メモリについて性能評価を行う。

データフロー実行では、待ち合わせ記憶が性能上の隘路となり所期の性能を達成することが困難となる。この問題に対して、待ち合わせ記憶のバンク化を行い処理性能の向上を図る。また－晴－では共有メモリに、マルチリードメモリを採用している。このため、要素プロセッサ数の増加に伴い莫大なハードウェア量が必要となる。そこで、このハードウェア量削減の可能性をさぐるために、要素プロセッサ数とポート数及びバンク数の関係についてシミュレータにより検討を行う。

Evaluation of a Simulated Parallel Processing System -Harray-

(in Japanese)

Hayato YAMANA, Takashi HAGIWARA,
Yoshihiro KUSANO and Yoichi MURAOKA

School of Science and Engineering, Waseda University
3-4-1 Okubo, Shinjuku, Tokyo, 169 JAPAN

The perpose of this paper is to evaluate a banked matching memory unit and a global memory unit of the -Harray- system by software simulation.

A matching memory unit is a bottleneck in the data flow execution. The bottleneck resulting from complicated data handling is cleared by banking its unit in the -Harray- system.

The global memory unit in the -Harray- system is constructed with multi-access memory scheme. The scheme make the amount of the hardware large. Therfore, the relation between the amount (the number of port and bank) and the performance is investigated to reduce the amount of the hardware.

１．はじめに

科学技術計算の高速実行に対する要求は、計算機の処理性能が向上するにつれて、ますます増大している。こうした要求に応えるため、プロセッサを多数台並べて並列処理を行い、高速処理を実現しようとするマルチプロセッサシステムの研究がさかんに行われている。その中でもデータフローコンピュータはプログラムに潜在する並列性を自然に抽出し、かつ並列処理を自然な形で実現出来ることから、注目されている。しかし、実現に際して（１）並列性を明示的に制御することが出来ないので、並列度の爆発による資源の枯渇が発生する、（２）データバケットの一致を調べる、いわゆる待ち合わせ記憶の性能がシステム全体の処理性能に大きな影響を及ぼすといった問題があることも知られている。

また、マルチプロセッサシステムにおいては、各々のプロセッサ間の結合方式も重要な問題となる。結合方式には大別してメモリ共有方式とメッセージ交換方式がある〔１〕。メモリ共有方式では、各プロセッサ間で共有のメモリ領域を持つことが出来るため、配列に代表される大域的データをＰＥ間で共有出来る。しかし、プロセッサ数が増加するに伴いアクセス競合も増大するといった問題がある。これに対してメッセージ交換方式では、プロセッサ間通信をＩ／Ｏポートを通してメッセージ通信の形で行うので、メモリアクセス競合といった問題が生じない。また、データが流れていくという意味から、データフローコンピュータとの整合性もよい。

これらの背景のもとに、並列処理システム－晴－では、プログラムの並列性を自然に引き出し高並列実行を行うために、前述のデータフロー方式を取り入れている。そして先に述べたデータフローにおける問題を解決するため、ＣＤフロー方式（Controlled Data Flow）〔２〕を提案している。ＣＤフロー方式はコントロールフローとデータフローを融合させた制御方式であり、マクロブロックと呼ぶ処理単位内でデータフロー実行を、マクロブロック間ではコントロールフロー制御を行い、明示的に並列性を制御する。また、待ち合わせ記憶の高速化のためにWMのバンク化を提案している〔３〕。

また、プロセッサ結合方式には、データフローコンピュータとの整合性のよいメッセージ交換方式を用いる。しかし、科学技術計算では大規模な配列等の構造データを扱うことが多く、この方式だけでは十分といえない。そこで、局所的なデータはメッセージ交換により通信し、大域的なデータは共有メモリを用いる方式を取り入れている。また、この共有メモリには、アクセス競合を避けるためにマルチリードメモリを採用し、さらに書き込み時のアクセス競合を軽減するためにバンク化を行う。これまでに、このような共有メモリに対する評価はいくつか行われているが、書き込み・読み出しのアクセス比率をパラメータとしたもの〔４〕〔５〕〔６〕が大半であり、実際のプログラムを使用したもの〔７〕は、ほとんどない。また、各プロセッサが完全にアクセス競合なしに読み込みを可能とし、書き込み時の競合も低く抑えるためには、ハードウェア量が莫大なものとなる。

本稿では、－晴－の機能レベルのソフトウェアシミュレータＨＳＳＶ２(Harray Software Simulator Version 2) を用いて、実際のプログラムを動作させ、（１）待ち合わせ記憶のバンク化の評価を行いその有効性を示す。次に（２）共有メモリのポート数及びバンク数と要素プロセッサ数との関係を調べ、共有メモリにおけるハードウェア量削減の可能性をさぐる。

以下、第２章では－晴－の概要を説明する。第３章ではソフトウェアシミュレータについて述べ、第４章でシミュレーション結果を示す。

２．並列処理システム－晴－

本章では、並列処理システム－晴－の概要を述べ、今回の評価対象である待ち合わせ記憶（WM:Waiting Memory）と共有データメモリ（GM:Global Memory ）について詳しく説明する。

２．１．－晴－の実行方式

－晴－では、ＣＤフロー方式（図１）を採用している。この方式は、プログラム全体をいくつかのマクロブロックと呼ぶタスクに分割し、マクロブロック内ではデータフロー実行をマクロブロック間ではコントロールフロー制御を行うものである。各マクロブロックは、その内部の並列度に応じて、プロセッサ群へ割り付けられる。例えば、並列度の高いベクトル演算などを含むマクロブロックは多数のプロセッサに分散され、並列に処理される。

このＣＤフロー方式によって、並列性を陽に秩序付けることができ、資源管理が容易になる。またＩＦ文等によって起こる条件分岐先の先行計算をマクロブロック単位で積極的に行うことが出来る。すなわち、プログラムの制御の流れに沿ってマクロブロックを複製・融合させ、各コントロールフロー毎にプロセッサを割り付け高速化を図るフローグラフ展開〔８〕がこの一例である。

２．２．－晴－の構成

－晴－は図２に示すように６つの大きなユニットからなり、プログラムのコンパイルや初期データのロード及び結果出力のため、ホストコンピュータと接続される。

集中制御機構(GCU:Global Control Unit)は、－晴－システム全体を制御する機構であり、各要素プロセッサ(PE:Processing Element)に対し、マクロブロック実行

制御命令を出す。この命令は、起動放送ネットワーク(BCN:BroadCast Network)を通じて各ＰＥに放送される。ＰＥは、ＧＣＵから の命令に従ってマクロブロックの実行を開始し、その終了を同期メモリ(SYM:SYnchronize Memory)に報告する。ＳＹＭは、複数のＰＥ間同期を高速に実行するためのものであり、ＧＣＵはＳＹＭの同期完了報告をチェックしながら制御を続ける。

ＰＥ間のデータ交換のためには通信ネットワーク(DCN:Data Communication Network)を用意している。また、配列等の大域的なデータは共有データメモリ(GM:Global Memory)に格納し処理を行う。

ＰＥの構成を図３に示す。ＰＥはＧＣＵからのマクロブロック実行命令を起動放送制御機構(BCU:BroadCast Control Unit)が受信することにより実行を開始する。まず、データメモリ(DM:Data Memory)内にある起動マクロブロックに属するデータ（以後パケットと呼ぶ）を待ち合わせ記憶(WM:Waiting Memory)に転送する。WMではパケットの待ち合わせを行い、対となったパケットを実行機構(EX:EXecution unit)に送る。ＥＸでは、演算 を実行すると共に新しいパケットを作成し、スイッチ機構(SW:SWitching unit)へ送る。 ＳＷはＤＣＮとのインタフェースであり、他ＰＥ宛のパケットを処理する。データメモリ制御機構(DMC:Data Memory Controller)は、現在実行中のマクロブロックに属するパケットをWMへ、その他のパケットをデータメモリ（Data Memory）へ格納する。これにより、WMには、現在実行中のマクロブロックに属するパケットのみが存在することになり、WMの負荷を軽くすることが出来る。

２．３．WMの構成

データフロー実行では、WM（待ち合わせ記憶）の高速化がシステム全体の高速化において重要なポイントとなる。この理由は次の２点である。

（１）連想検索処理が必要であり、現在のハードウェア技術では、大容量の高速連想機能の実現は困難である。

（２）データフローでは、次にどの命令が発火するかは不確定であり、フォンノイマンコンピュータにみられるレジスタを使用出来ない。このため、メモリを直接アクセスすることになり高速な処理が望めない。

まず（１）の問題点に関しては、マクロブロック単位で実行する－晴－の方式によって、WMに溜まるパケット量を少なくすることが出来る。このため、WMの少容

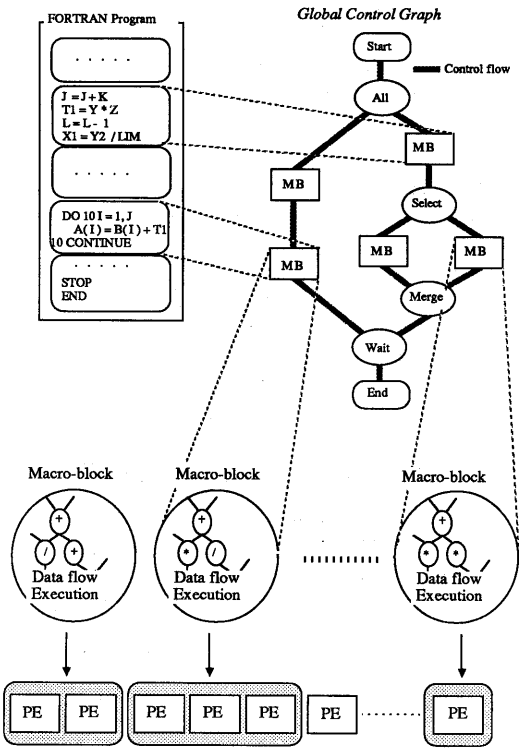


図１Controlled Data flowの概念

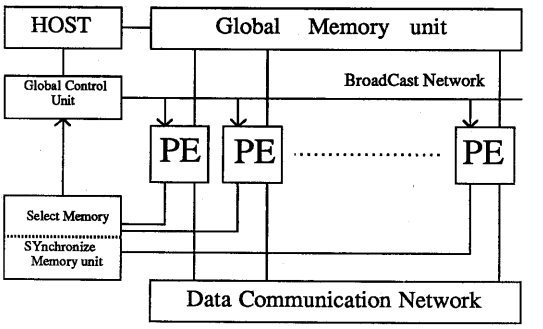


図2 -晴-の全体構成

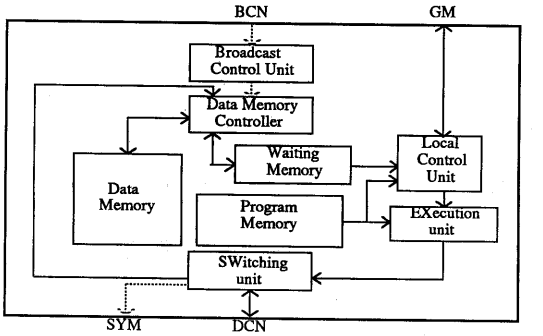


図3 PEの構成

量化が可能となる。これは、実行に際して現在必要とされないパケット（現在実行中でないマクロブロックのパケット）をＤＭへ格納し、ＷＭにはすぐに必要なパケットのみを置いているからである。ＷＭの具体的な容量に関しては、現在検討を行っている段階であり、別途報告する予定である。

（２）の問題点に関しては、ＷＭのバンク化によって対処する。（以後、共有メモリのバンク化と区別するために、ＷＭのバンクのことをＷＭバンクと呼ぶことにする）ＷＭのバンク化とは、図４に示すようにＷＭに入って来るパケットを各ＷＭバンクに分散し、スループット向上を図る方法である。このようにＷＭをバンク化した場合、各ＷＭバンクへの負荷を出来る限り等しくすることが、ＷＭのバンク化効率を上げるために必要となる。すなわち、入力パケットの各ＷＭバンクへの分散方法が重要となる。

この分散方法を決定するにあたり考慮すべき点は、各ＷＭバンクが独立に動作出来るようにパケットを分散することである。すなわち、対となるパケットは、いずれも同一のＷＭバンクに割り付け、そのＷＭバンク内で検索処理が完結するようにしなければならない。この点を考慮し、パケットの行き先ノード番号の下位２ビット（４バンク時）によってバンクを決定する方法を用いて、これまで考察して来た〔３〕〔９〕。この方法によると、対となるパケット同士は、必ず同一の行き先ノード番号を持っているため、同一のバンクに分散される。

しかし、この方法で分散を行うとノード番号の付け方による負荷のかたよりの影響を大きく受ける。特にカラーを用いたプログラムの場合には、カラーの異なった同一ノード行きのパケットが存在する為、同一ＷＭバンクに負荷が集中しやすく、ＷＭバンク化の効率が低下すると考えられる。そこで、カラーを考慮した、ＷＭバンクへの分散方法を提案する。

カラーを考慮したＷＭへの分散は、以下の手順で行う。

（１）行き先ノード番号とカラーの下位２ビットの和をとり、それをＷＭバンク番号（０～３）とする。〔４バンク化時〕

（２）スティッキーパケットはカラーを無視して待ち合わせを行うため、全ＷＭバンクに対してコピーを送る。

この方法により、同じノードを行き先とするパケットもカラーが異なれば違ったＷＭバンクへ割り振られる。これにより、カラー使用時のノード番号の付け方による負荷のかたよりの影響を小さくすることが出来ると考える。

２．４．ＧＭの構成

科学技術計算では、大規模な配列等を利用したものが多い。その為、プロセッサ間結合方式としてメッセージ交換方式のみを採用した場合には、通信のトラフィックが大きくなり、配列等のデータを効率的に扱うことが出来ない。そこで、局所的なデータ通信にはメッセージ交換を用い、大域的なデータ通信には共有メモリを用いる方式を－晴－では取り入れる。しかし、共有メモリの形態をとった場合には、複数プロセッサからのアクセス競合が問題となる。そこで、このアクセス競合を回避する解決法の１つとしてマルチリードメモリを採用する。マルチリードメモリの実現方法には、１つのメモリに複数のポートを設けたマルチポートメモリと、同一の内容をもった複数のメモリモジュールを用意し、同時読み出しを可能とするものがある。

－晴－では、後者の方法を用いる。図５に－晴－の共有データメモリＧＭの構成を示す。図５では、同一の内容をもったメモリモジュールを各ＰＥが持ち、読み出しのアクセス競合をなくしている。しかし、書き込み時には、全メモリモジュールに対して書き込みを行い、メモリモジュール間でのデータの同一性を保証をする。このため、書き込み時にはアクセス競合が起こる。これを軽減する為に－晴－ではバンク化を行い、異なったバンク間での並列書き込みを可能とする。

しかし、このようなメモリ形態をとると、

（１）同一バンクへのデータ書き込みに対する調停が必要となる。

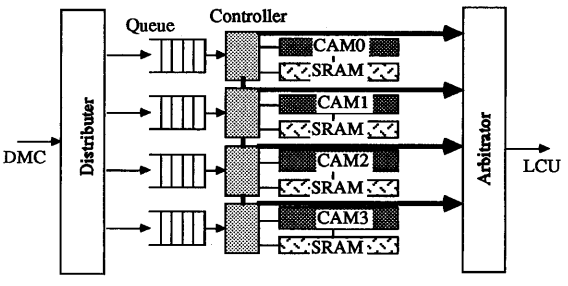


図４ WM構成(4バンク化時)

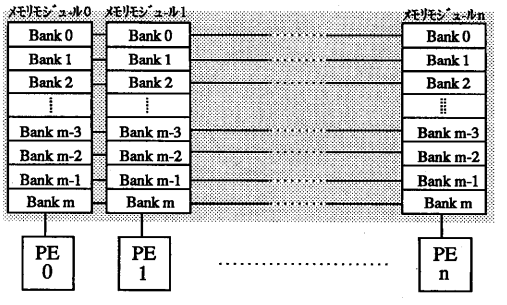


図5 GMの構成

（２）メモリモジュール数及びメモリモジュール間の接続数が膨大になるため、特に実装面で問題がある。

等が挙げられる。

これらの欠点はプロセッサ数が増加するに伴って、より困難となるため、性能をある程度犠牲にしてもハードウェアの簡略化を図ることが必要である。このため、第４章において－晴－の共有データメモリＧＭのポート数およびバンク数とプロセッサ数との関係を調べ、ハードウェア量削減の可能性をさぐる。

３．ソフトウェアシミュレータ

３．１．シミュレータの概要

－晴－のアーキテクチャの妥当性を検証するため、第２章で述べた－晴－の構成を機能レベルで実現するソフトウェアシミュレータＨＳＳＶ２(Harray Software Simulator Version 2)を作成した。このシミュレータは、Micro-VAXII/GPX 上の Modula-2 言語で記述され、ソースコードで約14,000行である。入力は、－晴－の機械語で記述されたプログラムである。このプログラムは、通常のＦＯＲＴＲＡＮで書かれたプログラムをデータフローグラフに直した後、これを－晴－のアセンブラで記述し直し、ＨＩＡＳ（Harray Integrated Assembler）により機械語に翻訳したものである。

このＨＳＳＶ２の主要パラメータを表１に示す。現在、各プロセッサ間の通信ネットワークＤＣＮについては検討中であり、シミュレータでは暫定的にマルチバスを想定している。したがって、今回のシミュレーションでは、ＰＥ間での通信にＤＣＮを使用せず、全てＧＭを介して通信を行っている。

表1 HSSV2の主要パラメータ

項　　目	設 定 値
システムクロック	10 MHz
PE内モジュール間 パケット通信速度	200ns/Packet
ネットワーク(DCN)通信速度	3MPackets/s
演算時間（除算）	300 ns
（上記以外）	1500 ns
待ち合わせ記憶(WM)	
処理時間(平均)	700 ns/Packet
バンク数	4/PE
ALU数	2/PE
メモリ・アクセス時間	
Program / Data Memory	100 ns
Global Memory	400 ns

３．２．実行プログラム

科学技術計算においては、大規模な非対称行列を係数行列に持つ連立一次方程式を扱う場合が多い。そこで、連立一次方程式解法に使用されるプログラムの中から３つを取り挙げ解析の対象とする。連立一次方程式の解法には大きく分けて、直接法と反復法がある。直接法から（ａ）ＬＵ分解と（ｂ）ガウスの消去法を、反復法から（ｃ）双対共役勾配（ＢＣＧ）法を取り挙げる。各プログラム共に、２×２、４×４、８×８、１６×１６の行列を扱った場合についてシミュレーションを行う。

４．シミュレーション結果

４．１．WMのバンク化の評価

WMのバンク化により処理速度がどの程度向上するかを調べるために、１台のＰＥを用いてシミュレーションを行う。そして、WMバンク化しない時の実行時間を基準（＝１）とした時の処理速度比を求める。プログラムは、（ａ）ＬＵ分解、（ｂ）ガウスの消去法、（ｃ）ＢＣＧ法について、２×２及び８×８行列を実行させる。

また、各WMバンクへの分散方法は、

Ｎ）ノード番号の下位ビットをそのままWMバンク番号とするこれまでの方法。

Ｎ＋Ｃ）ノード番号とカラー番号の下位ビット同士の和をとり、それをWMバンク番号とする方法。

の２種類を用いる。図６にこの結果を示す。

まず、８×８行列を実行した時の結果について考察する。８×８行列を実行した時、（ａ）（ｂ）（ｃ）いずれの結果においても、カラーを考慮した分散方法の方が、ノード番号のみで分散する方法に比較して効率がよい。また、WMバンク数が２の時と４の時とを比較すると、WMバンク数が２の時の方がこの傾向が顕著に表れており、WMバンク数４の時は、どちらの分散方法を用いた場合でもほぼ同じ効率を示している。これは、プログラムの性質によるものと考えることが出来る。実際に、WMをバンク化せずにWMの処理速度を４倍に設定してシミュレーションを行ったところ、３種類のプログラムについていずれも、４バンク化した時の値とほぼ一致した。WMの処理速度を４倍にした結果と４バンク化した時の結果がほぼ等しいということは、プログラムの性質（並列度の変化等）によって４バンク化した時に、これ以上の高速化が望めないことを示す。すなわち、４バンク化時にはパケットの分散方法よりも、プログラムの性質によって高速化が妨げられているといえる。

次に２×２行列を実行した場合の結果について考察す

る。この場合、（ａ）（ｂ）のプログラムでは、カラーを考慮しない分散方法の方が、カラーを考慮した方法に比較して性能がよい。また、（ｃ）のプログラムの場合においても、カラーを考慮した場合としない場合での差がほとんど見られない。これは、２×２行列の場合、並列度が小さいために、カラーを考慮した分散方法を行うことによるオーバーヘッドが表れるためだと考える。このオーバーヘッドはスティッキーパケットをWMが全バンクにコピーして渡すことによって生じる。

最後に２×２行列と８×８行列の結果を比較すると８×８行列の方が処理性能がよい。これは、プログラムの並列度が増すことにより、WMのバンクが有効に利用されることに起因する。（Ｃ）のＢＣＧ法では、２×２と８×８の時の性能の差が、他の２つのプログラムに比較して小さい。これは、ＢＣＧ法は処理の途中でN^2 に比例して並列度が出る部分もあるが、総和計算のために逐次に実行しなければならない部分もあり、平均すると並列度があまり増加しないことに起因している。

以上の結果より、カラーを考慮したパケットの分散方法は、考慮しない方法に比較して、プログラムの並列度があまり小さくない場合において効率のよい分散を実現出来ることがわかる。また、４バンク化した時には、分散方法よりもプログラムの性質によって処理性能が抑えられるという傾向があることがわかる。したがって、WMのバンク化を考えた場合、２バンク化の時はカラーを考慮した分散方法により１．７倍前後の処理性能向上が見込まれるが、４バンク化した時は、プログラムの並列度が十分になければコストパフォーマンスがよくないことがわかる。

４．２．GMの評価

マルチリードメモリに関する諸特性のうち、パフォーマンスに大きな影響を与えるものに、メモリへのポート数（メモリモジュール数）、バンク数、メモリモジュール間でのデータ転送時間等が挙げられる。ここでは、この内のポート数とバンク数についてＰＥ数との関係を調べ、一晴一の共有データメモリであるGMのハードウェア量削減の可能性を調べる。

ポート数とは、一度にメモリにアクセス出来るＰＥの数である。図５では、各ＰＥが１つのポート（メモリモジュール）を持っており、ＰＥ数＝ポート数である。このポート数を少なくした時の性能に与える影響を調べる。

バンク数とは、書き込みバンク数のことであり、同一バンクに対して書き込みの出来るＰＥ数は、同一時期には１台に限られる。ただし、読み出しは、そのバンクに対しての書き込みが行われていなければ、バンク数に関係なく全てのＰＥがアクセス可能であるとする。また、配列の各バンクへの分割は、配列の添え字番号をバンク数で割った余りでグループ化した。ここでは、このバン

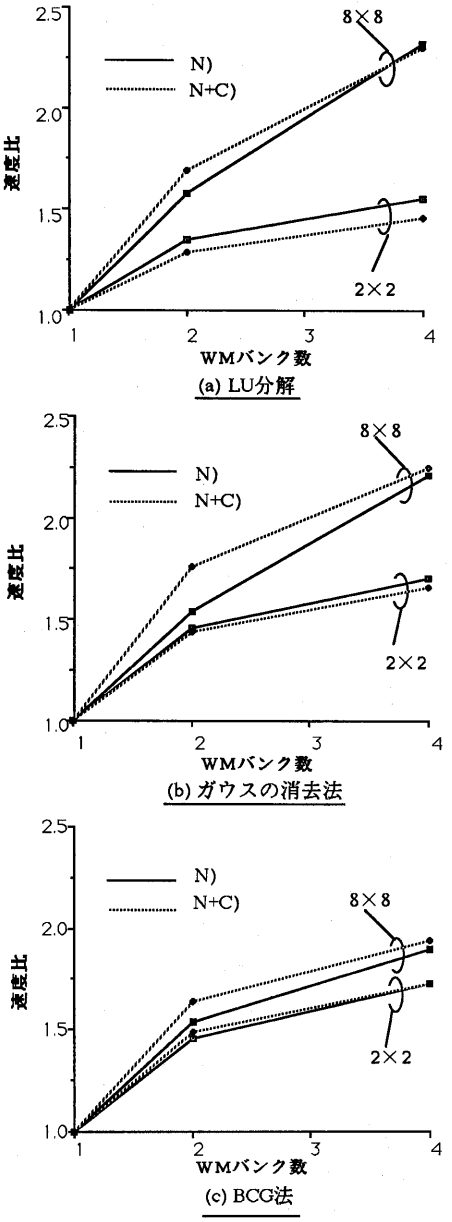


図6 WMバンク化による処理速度向上

N)　ノード番号による分散方法
N+C)ノード番号とカラーによる分散方法
(WMバンク化しない時を基準とした速度比を示す)

ク数を変化させた時の性能に与える影響を調べる。

メモリモジュール間でのデータ転送時間は、今回は考慮せず０と仮定した。しかし、現実には０にすることは不可能であり、モジュール数の増加に伴い、モジュール間での遅延が生じるので、今後検討していく。

ａ）ポート数の影響

結果を図７に示す。横軸に、ＰＥ数で正規化したポート数をとり、縦軸には、ポート数＝ＰＥ数とした時の実行時間を基準とした時の速度比をとる。また、この時バンク数の影響を排除するために、バンク数は無限大（１ワード１バンク）と仮定した。また、ポート数による影響はプログラムの並列度にも依存しているため、ＰＥ数とプログラムの最大並列度の比をパラメータとした。

この結果より、（ａ）（ｂ）（ｃ）いずれのプログラムにおいても、ポート数がＰＥ数の２分の１以上であれば、処理速度にほとんど影響を与えないことがわかる。また、プログラムの最大並列度にＰＥ数が近づくにつれ、ポート数はさらに少なくてよいことを、これらの結果は示している。これは、ＰＥ数が増大すると１ＰＥあたりのメモリアクセス量が減少するため、全てのＰＥが同時にポートを要求する確率が減少することに起因していると考える。

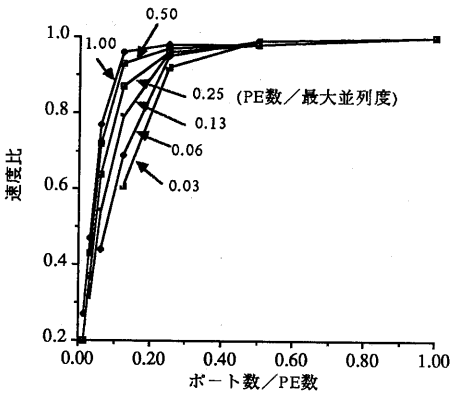
ｂ）バンク数の影響

結果を図８に示す。横軸に、ＰＥ数で正規化したバンク数をとり、縦軸には、バンク数を無限とした時の実行時間を基準とした時の速度比をとる。また、この時のポート数による影響を排除するために、ポート数は無限大と仮定した。また、ＰＥ数が８，１６，３２台の場合について、行列が８×８、１６×１６の場合についての結果を示す。

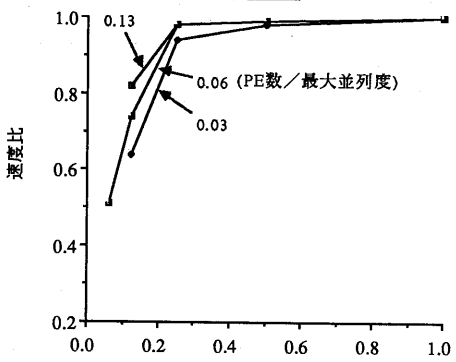
この結果より、バンク数がＰＥ数の２分の１以上であれば、処理速度にほとんど影響を与えないことがわかる。また、１割程度の処理速度低下を許せば、バンク数はＰＥ数の４分の１程度でもよいことがわかる。

ポート数とバンク数が処理速度に与える影響を比較すると、明らかにポート数のほうが大きな影響を与えることがわかる。一般にメモリへの書き込みと読み出しの比率は、読み出しの方が大きいことを考えると、書き込み時のアクセス競合を軽減させるバンク数の増加よりも、読み込み時のアクセス競合を軽減させるポート数の増加の方が重要であることを示している。

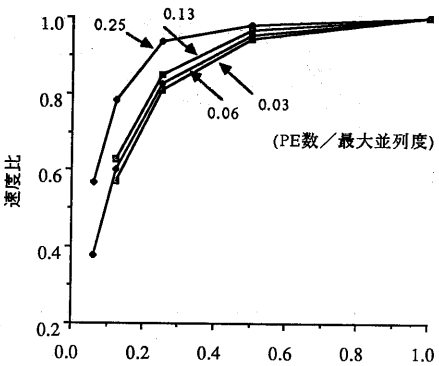
以上の結果より、ポート数はＰＥ数の２分の１、バンク数もＰＥ数の２分の１までであれば、処理速度に与える影響は無視出来ることがわかった。



(a) LU分解



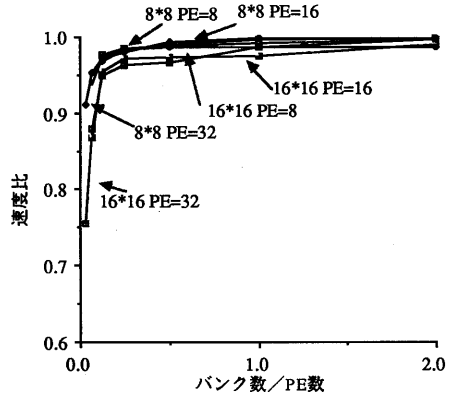
(b) ガウスの消去法



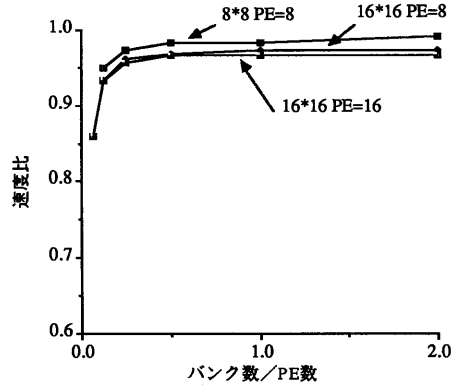
(c) BCG法

図７ ポート数の影響

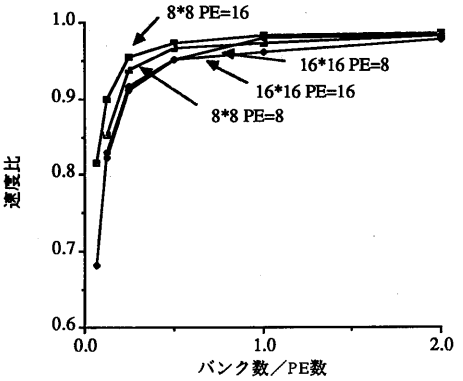
(バンク数=∞,ポート数=∞とした時を基準とした速度比を示す)



(a) LU分解



(b) ガウスの消去法



(c) BCG法

図８バンク数の影響

(バンク数-∞,ポート数-∞とした時を基準とした速度比を示す)

５．おわりに

本稿では、－晴－の機能レベルのソフトウェアシミュレータを用いて、（１）待ち合わせ記憶WMのバンク化と（２）共有データメモリGMの評価を行った。その結果、WMのバンク化については、バンク化時のパケット分散の方法として、カラーを考慮した方法が考慮しない方法に比較して効率がよいことを確認した。また、共有データメモリの評価においては、ポート数・バンク数共にＰＥ数の２分の１までに削減しても処理速度にほとんど影響を与えないことを確認した。

今後は、さらに多くのプログラムについてこれらの結果を検証すると共に、このソフトウエアシミュレータを用いて－晴－の性能評価を進めていく予定である。

謝辞

シミュレータ作成及び本研究の遂行にあたり御支援いただいた本研究室の樋口和広氏に感謝いたします。

参考文献

〔１〕富田：〝並列計算機構成論〟，昭晃堂，PP.195-240，(1986)

〔２〕H.Yamana, T.Marushima, T.Hagiwara, Y.Muraoka : "System Architecture of Parallel Processing System -Harray-", Proc of Int. Conf. on Supercomputing, pp.76-89, (1988)

〔３〕山名，丸島，草野，村岡：〝並列処理システム－晴－の要素プロセッサ構成〟，情処研報，Vol.88, No.19, 88-CA-70-4, (1988)

〔４〕久野：〝競合問題の解決を目差すメモリ構成法－TRANSPARENT MEMORY〟，信学論文誌，Vol.69-D,No.2, pp.128-135, (1986)

〔５〕内堀，古谷，西田：〝マルチリード・メモリの動作解析〟，第２７回情処全大，5N-2, (1982)

〔６〕天野，地川，吉田，相磯：〝マルチリード・メモリを用いた並列計算機の性能解析〟,信学論文誌，Vol.J67-D, No.9, pp.1028-1035, (1984)

〔７〕曽和，上村：〝データフローコンピュータＤＦＮＲ－２のシミュレーションによる性能評価〟，信学論文誌，Vol.J69-D, No.7, pp.1054-1065, (1986)

〔８〕萩原，山名，村岡：〝並列処理システム－晴－の並列化コンパイラの設計について〟，信学技報，CPSY88-32, PP.43-48, (1988)

〔９〕山名，草野，萩原，村岡：〝並列処理システム－晴－の待ち合わせ記憶構成〟，第３７回情処全大，1N-2, (1988)