計算機アーキテクチャ 119－32
（1996．8．29）

動的バリア同期管理機構をもった並列計算機MDBM/FMM

立川尚久† 大串弘一郎† 加藤亮†
松田孝史† 岩根雅彦†

汎用並列計算機において，同期と通信にかかるオーバーヘッドを減らすことは細粒度および粗粒度並列性を持つ様々なプログラムを効率良く実行する上で重要である．よって，動的バリア同期管理機構，再構成可能共有メモリ，NEWS転送を評価するためのテストベッドとして並列計算機MDBM (Multiple Dynamic Barrier Management MIMD) /FMM(Flexible Mesh-network Multi-microprocessors)を開発した．MDBM/FMMはHC(Host Computer)，NIIU(New Integrated Interface Unit)，及び32-64台のPUから構成される．動的バリア同期管理機構はプログラムの実行時に各プロセッサ間で同期をとるためのバリアグループを設定する．再構成可能共有メモリは共有メモリ構成から局所メモリ構成まで段階的に構成できる．NEWS転送はILLIAC網を用いたDMA転送である．バリアグループの変更頻度を検討した結果，変更にかかるコストとプロセッサ利用率の間でトレードオフをとりプロセスもしくはモジュールレベルでバリアグループを変更する場合に有効性が確認された．ラプラス方程式によりディリクレ問題を解く問題でNEWS転送の効果を測定した．16PU使用時の速度向上比は14.4倍となった．

Multi-microprocessor System MDBM/FMM with Dynamic Barrier Synchronization Management Mechanism

NAOHISA TATSUKAWA,† KOUICHIROU OHGUSHI,† AKIRA KATOU,†
TAKASHI MATSUDA† and MASAHIKO IWANE †

In a general purpose multi-processor system, decreasing the overhead of interprocessor communication and synchronization is important for running from fine to coarse grain size parallel programs. Therefore, MDBM/FMM(Multiple Dynamic Barrier Management MIMD/ Flexible Mesh-network Multi-microprocessors) were developed as the test bed evaluating the dynamic barrier synchronization management mechanism, the reconfigurable shared memory and the news transfer. The MDBM/FMM consists of host computer and new integrated interface unit and 32-64PUs. The dynamic barrier synchronization management mechanism specifies dynamically in execution of programs the barrier group in which the processing unit synchronize each other. The reconfigurable shared memory can configure not only local memory but also shared memory. The news transfer is direct memory access transfer using the ILLIAC IV interconnection network. We tested about the barrier group alteration frequency and the news transfer on the MDBM/FMM. From the first test, it shows that considering tradeoff both barrier group alteration cost and processor utilization is a necessity. The second test shows that the speedup ratio is 14.4 using 16PUs on the MDBM/FMM.

1. はじめに

粗粒度の並列性だけでなく細粒度の並列性を利用することができれば並列計算機の資源を有効に利用する機会を増やすことができる．細粒度並列処理においてはプロセシングユニット間の同期および通信が頻繁に行われるため，オーバーヘッドの極めて小さい同期と通信の機構が求められる．

† 九州工業大学工学部電気工学科
Department of Electrical Engineering, Faculty of Engineering,Kyushu Institute of Technology

細粒度並列処理に適した同期としてはバリア同期があげられる．これは同期を行おうとするプロセッサ群が待ち合わせを行い，すべてのプロセッサが待ち合わせ位置に到達したならば処理を再開するものである．このような同期の仕組みとしてFuzzy Barrier[1]，Elastic Barrier[2]，Ultimate Barrier[3]，SBM(Static Barrier MIMD)[4]，DBM(Dynamic Barrier MIMD)[5]等が提案されている．

通信のトポロジーをバス結合とそれ以外に大別すると，バス結合は1) 任意のノード間でアクセスレイテンシが均一であり各ノードに対するジョブの割付け方による処理効率への影響がない，2) 相対的に通信の帯域

幅が狭い，といった特徴を持つ．バス結合以外のトポロジーでは逆の性質を持つ．細粒度並列処理に適した通信機構とは，データサイズの小さい多数の通信を効率的に処理できるものでなければならない．

以上を踏まえ，次のような機能を持つテストベッド並列計算機MDBM/FMMを開発した．

MDBM/FMMは複数のバリアグループ(同期をとるプロセシングユニット組)の同期成立を同時に検出可能なバリア同期機構を持ち，バリアグループの変更は任意のプロセシングユニットから行うことができる．プロセシングユニットのメモリ空間はローカル，グループ共有，システム共有の3つに分割される．共有空間のコピーが各プロセシングユニットに置かれ，単一のバスを介してハードウェアにより一貫性が保持される．ローカル空間に対するRead/Writeアクセスおよび各共有空間に対するReadアクセスは他のプロセシングユニットとの競合なく行うことができる．さらにFMMでは特定アプリケーション向けの通信機構として，ILLIAC網による隣接プロセシングユニット間のDMA転送機構を持つ．

本稿は以下のように構成されている．第2章では基本設計を述べる．第3章でMDBM/FMMシステム構成を示す．第4章では実験について述べる．第5章でむすびとする．

2. 基本設計

2.1 動的バリア同期管理機構

多重プログラミング環境でバリア同期を用いるには1)各プロセッサからの同期要求を同時に処理する，2)複数のバリアグループの同期成立を並行して検出する，3)バリアグループを必要に応じて更新する，といった能力が必要である．

バリア同期の成立順序は各プロセスによって異なるため，バリアグループの更新をハードウェアで自動化することは困難である．また，同期成立の条件をゆるめたUltimate Barrier等のバリア型同期[3]を用いればダミーバリアの挿入によるオーバーヘッドは低く抑えることができるため，バリアグループの設定をハードウェア化することのメリットは小さい．このことから，MDBM/FMMではハードウェアで実現するのは任意のPUからバリアグループを設定できる機能にとどめ，バリアグループの更新操作はソフトウェアの責任とする．

動的バリア同期管理機構の構成を**図1**に示す．複数のバリアグループの同期検出を並行して行なうために並行した包含検索を行なうことのできるMCAMを用いる．各プロセッサは同期位置に到達するとBar命令(バリア同期要求)を実行する．この命令によりBreq信号を介してBRR(Barrier Request Register)の対応するビットを1にセットする．MCAMはBRRを検索データとして，このデータに包含される全てのエントリを並行して検索し，それらの論理和をBgrt(Barrier grant)に出力する．Bgrtの値は以下の式で表される．

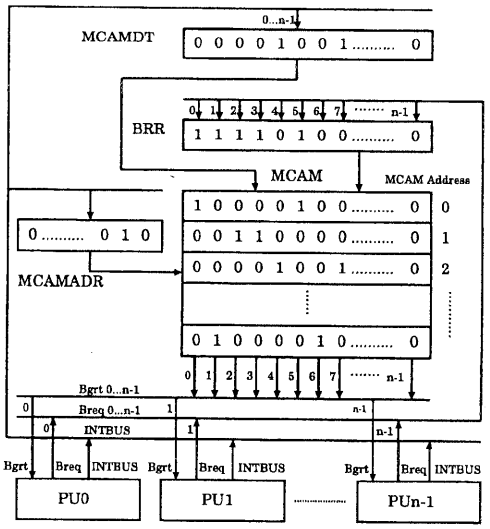

図1 動的バリア同期管理機構

$$B_{grt} = \cup\{MCAM_i \mid MCAM_i \subseteq BRR\}$$

但しiはMCAMのエントリ番号を表す．また∪は集合の全要素の論理和を行うものとする．

Bgrt信号がPUに送り返されるとともに，BRRのうち同期の成立したPUに対応するビットが0にリセットされる．

2.2 再構成可能共有メモリおよびNEWS転送

細粒度/中粒度/粗粒度並列処理を効率よく行うには，共有メモリ構成から局所メモリ構成まで段階的にメモリ構成を選択できる再構成可能なメモリが望ましい．

そこで共有メモリを個々のプロセッサに分散させて，プロセッサグループごとに任意の大きさの共通のメモリ空間，すべてのプロセッサで任意の大きさの共通のメモリ空間をとることのできる再構成可能共有メモリRSM(Reconfigurable Shared Memory)を考える．

プロセッサグループ内のすべてのプロセッサで共通なメモリ空間をグループ共有メモリ(GSM:Group Shared Memory)，システム内のすべてのプロセッサで共通なメモリ空間をシステム共有メモリ(SSM:System Shared Memory)と呼ぶ．

図2にそのRSM構成を示す．メモリへの書き込み，読みだしの性質の違いを考慮してDPM(Dual Port Memory)を利用し，LM(Local Memory)とGSMとSSMの境界を示す境界レジスタBASE(GS/SS)を用いることでメモリ空間を分割する．

MPUがメモリに書き込みを行う時，書き込み先アドレスと境界レジスタであるBASE(GS/SS)と比較する．LMへの書き込みの場合はDPMのPort_Aからデータを書き込む．GSMへの書き込みの場合はDPMのPort_Aからデータを書き込み，GPNO(グループ番

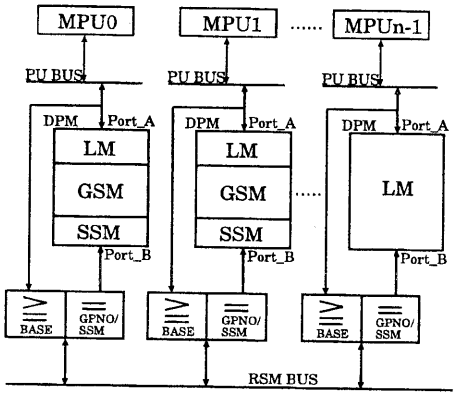

図2 RSM 構成

号), アドレス, データを RSM バスにブロードキャストする. 他の MPU は自身が持つ GPNO と RSM バス上の GPNO が一致しているなら DPM の Port_B から RSM バス上のデータを書き込む. SSM への書き込みの場合は RSM バスに GPNO をブロードキャストしないことを除いて GSM への書き込みと同様の動作をする. 各 MPU は DPM の Port_A から独立に読みだしが可能である. なお GSM および SSM の相互排除のために lock/unlock 機能を用意する.

特定アプリケーションではプロセッサグループ内プロセッサ間で大量のデータの転送が必要なことがある. このような場合は共有メモリによるプロセッサ間通信ではオーバーヘッドが大きい. そこでプロセッサグループ内隣接プロセッサ間の大量のデータ転送のために ILLIAC 網による DMA 転送機構を用意する. これを NEWS 転送と呼ぶ. NEWS 転送の方向は上 (North), 右 (East), 左 (West), 下 (South) の 4 方向を指定できる.

3. MDBM/FMM システム構成

3.1 システム概要

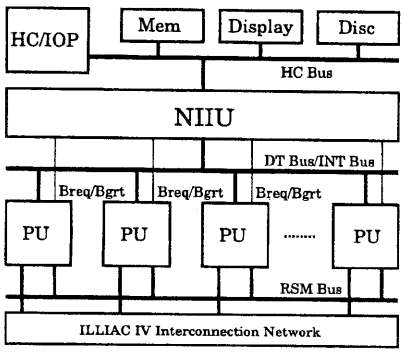

図3 MDBM/FMM システム構成

MDBM は図 3 に示すように HC(Host Computer), NIIU(New Integrated Interface Unit), 及び 32-64 台の PU(Processing Unit) から構成される. HC は 80486DX,16MB メモリ, 各種入出力装置などから構成され, 主にユーザインターフェースを提供する. PU は 8088MPU, 8087NDP, 256KB メモリ, 各種インターフェースなどから成る. NIIU は FPGA および汎用 IC により構成されている.

HC-NIIU 間は HC バス, NIIU-PU 間は DT バス, INT バス, 及びバリア同期要求信号 Breq/ バリア同期成立信号 Bgrt で接続される. PU は RSM バスで接続されている. FMM では特定アプリケーション向けの PU 間相互結合網として ILLIAC 網を備える.

3.2 NIIU 構成

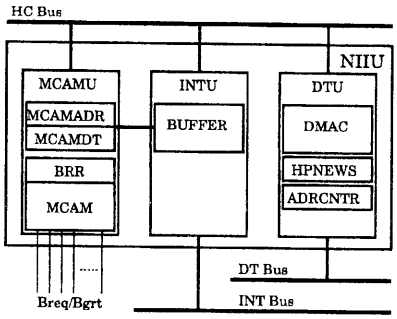

図4 NIIU 構成

NIIU は図 4 に示すように 1)HC と PU 間のマルチキャスト機能によるデータ転送, 2)PU から HC への割り込み処理, 3)PU からのバリア同期に関係する処理を行う DTU(Data Transfer Unit), INTU(Interrupt Unit), MCAMU(MCAM Unit) の 3 つのモジュールから構成される.

DTU は HC バスと DT バスを介して HC-PU 間でプログラムやデータを DMA 転送する. また隣接する PU 間の NEWS 転送の制御も行なう. PU から HC への DMA 転送は 1 対 1 でなされるが, HC からは複数 PU へマルチキャストの転送を行うことができる.

NEWS 転送では転送方向として N, E, W, S の 4 方向のいずれか 1 つを指定する. プロセッサグループ内の PU について, 指定された方向にのみ同時に同一アドレスのデータが転送される.

MCAMU に対するバリアグループ変更要求は HC と各 PU から行なうことができる. HC からは IO デバイスへのアクセスの形で直接 MCAMU に要求が送られ, PU からは INTU を通して要求が送られる. 各 PU の INT バスに対するバス使用権はデージーチェーンにより順位付けられており, 1 度に 3 バイトの任意のデータを送ることができる. PU 内のインターフェース回路が PU 番号および送信の目的を示す 1 バイトのタグを付加

する．INTU はタグに基づいて HC と MCAMU のどちらにリクエストを配送するかを判別する．

3.3 PU 構成

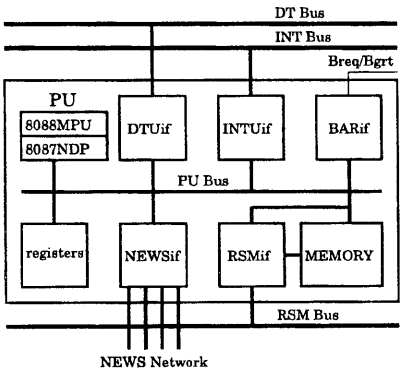

図5 PU 構成

PU は割り当てられた逐次命令列を各々独立に実行するユニットである．PU の構成を図5に示す．各種バスおよび Breq/Bgrt 信号線と PU バスとの接続にはそれぞれ専用のインターフェース回路を使用する．PU バスに対して DTUif はバスマスタであり，INTUif および BARif は IO デバイスである．NEWSif は DTUif の制御のもとで PU バスのドライブとデータ取り込みを行なう．

8088MPU が持たない命令は IO デバイスへのアクセスにより実装している．SVCC 命令はホストへの割り込み要求，GLOCK 命令，GUNLOCK 命令，SLOCK 命令，SUNLOCK 命令はグループ共有メモリ，及びシステム共有メモリの相互排除，WMCAM 命令，Bar 命令，AdBar 命令☆はバリア同期関連を表す．INTUif は，SVCC 命令，WMCAM 命令が実行された時に INTU にリクエストを送る．3 バイトのパラメータは MPU の IO 出力命令により INTUif 内のバッファに書き込んでおく．

RSMif は GSM と SSM の一致性維持を行う．共有メモリへの書き込みは RSM バスの使用権が得られるまで待たされ，その後 GPNO もしくは SSM アクセスフラグを付加して RSM バスに出力される．他の全てのプロセッサは RSM バスをスヌーピングし，自らの GPNO と一致する場合もしくは SSM アクセスフラグが付加されていた場合にはメモリの指定のアドレスに取り込む．RSM バスの調停には IEEE FutureBus のプロトコルのサブセットを使用している．

☆ Bar 命令と AdBar 命令を組み合わせることで Fuzzy Barrier を行なうことができる．命令の実行が Barrier Region へ入ることを AdBar で示し，MCAMU に対して Breq を送信する．後に Bar 命令を実行し，同期が成立していれば命令の実行を継続する．

4. 実　　験

4.1 バリアグループ変更

4.1.1 実験方針

MDBM/FMM ではバリアグループをソフトウェアで変更するため，若干のオーバーヘッドを伴う．従ってバリアグループの変更の頻度とプロセッサ利用率の間でトレードオフが必要となる．バリア型同期と最適化コードスケジューラを用いればダミーバリアによるオーバーヘッドを低く抑え，並列化による速度向上に見合った台数のプロセッサを使用するコードを生成することができるだろう．その場合バリアグループの変更はそのコストに見合った間隔で，もしくは動的に実行すべきコードが変化する場合に限って行なわれる．

マルチプログラミング環境では実行時にプロセスを特定の PU に割り当てる．このため，このプロセスレベルの動的スケジューリングにバリアグループ変更が伴う．

次に，プログラマはプログラムを複数のモジュールに分割し，与えられたプロセッサのどれにどのモジュールを担当させるかを決定することができる．プロセッサの担当するモジュールが変化する時にはバリアグループの変更を必要とする場合がある．

最後に，1 つのモジュールの内部でもダミーバリアによるオーバーヘッドを削減するためにバリアグループに変更を加えることが考えられる．

4.1.2 プロセスレベル

プロセス単位でバリアグループを変更する例として，様々な半径を持つ球の体積と表面積を求める簡単な doall ループを複数の子プロセスで実行する場合を考える．ループボディを 2 項演算の中間コードレベルで細粒度並列化し，並列度を 2 とする．ループイタレーションを 10 回アンローリングし，これを 1 つの子プロセスとする．子プロセスの実行には 2 台の PU を要するが，これを 1 つのバリアグループとして登録する．バリアグループ変更にかかる時間の子プロセスの実行時間に占める割合は理想値で 3.96% に抑えられる．実験では 2 台，4 台，8 台の PU を使用し，それぞれの場合について 80 個の子プロセスを実行した．

プロセス生成にかかるオーバーヘッドは考慮せず，WMCAM 命令と子プロセスの実行にかかる時間を測定した．2 台の PU を利用した実行時間を 1 として，4 台と 8 台の PU を利用した場合の速度向上比を図6に示す．同図において，理想値とは MCAM 命令の実行時間と共有メモリへの書き込み競合によるオーバーヘッドを除いたものである．

実測値において，4 台の PU で 1.90 倍，8 台の PU で 3.52 倍の速度向上比を得た．4 台よりも 8 台の場合が理想値に対する実測値の速度向上比の低下が大きくなっているが，この要因としては用いた例が共有メモリへの書き込みを頻繁に行なう物であったため，RSM バ

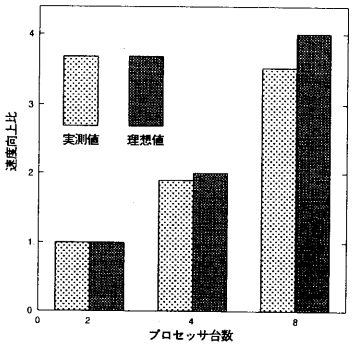

図6　速度向上比

表1　逐次コードに対する速度向上比

プログラム	並列度		
	4	3	2
1	3.24	-	-
2	1.43	1.43	1.29
3	1.35	-	1.34
4	1.95	-	-
5	2.63	-	-

スへのアクセス競合が多発したものと思われる.

4.1.3　モジュールレベル

5つの簡単なプログラム[6]を例に，逐次コードに対する速度向上比とプロセッサ利用率を考慮して，いくつかの並列度で並列化を行なった．モジュールは実行開始時にバリアグループの登録を行なう.

各プログラムの逐次コードに対する速度向上比を表1に示す.

プログラム2，3は他のプログラムと比較してより少数のPUで同程度の速度向上比を得ることができた．このような性質のモジュールが複数存在する場合には，例えば**図7**に示すように同一のバリアグループで順に実行していく方法と，それぞれに適したバリアグループに変更して実行する方法を選択することができる.

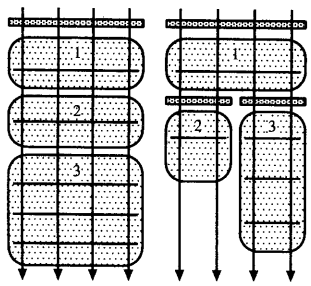

図7　モジュールの割り付け例

4.1.4　命令レベル

4.1.3で用いたプログラムについて，命令間の依存関係を満たすに必要な最小のバリアグループのみで同期をとるようにバリアグループ変更命令を加え並列化を行なった．プログラム1,3,4,5は並列度4，プログラム2は並列度3とする．但し，プログラム1はダミーバリアが存在しないためバリアグループの変更は必要無い．バリアグループを変更しない場合の実行時間を1とした場合の各プログラムの速度向上比対プロセッサ利用率を図8に示す.

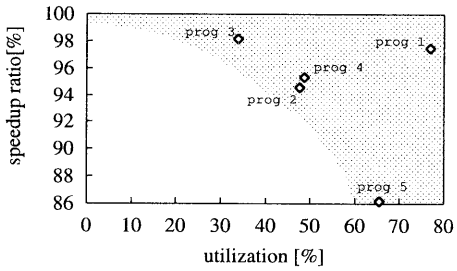

図8　プロセッサ利用率と速度向上比の関係

バリアグループ変更を行なった場合，プロセッサ利用率の高いプログラムほど実行時間が増加した．実行時間はバリアグループを頻繁に変更する程より低下する傾向があると予想されるので，モジュールのプロセッサ利用率対速度向上比は一般に図8の網掛けの領域のような分布をすると思われる.

5つのプログラムすべてでバリアグループを変更したことによる速度向上は見られなかったことから，ダミーバリアによるオーバーヘッドを削減する目的でバリアグループを変更することは効果が無いことがわかった.

4.2　NEWS転送

偏微分方程式の数値解法として用いられる差分法の一つであるラプラスの方程式でディリクレ問題(第1境界値問題)を解く場合に，データ転送にNEWS転送を使用して粗粒度の並列計算を行う．ラプラスの方程式は次のように変形できる.

$$U(x,y) = \frac{1}{4}(U(x-h,y) + U(x+h,y) + U(x,y-h) + U(x,y+h))$$

この計算は，物理学における多くの偏微分方程式，計算機上での画像処理など幅広く応用される.

実際の計算は，2つのデータ配列(u0,u1)を用意してu0の値を用いてu1の値を計算し，その後u1の値を用いてu0の値を計算する，という動作をループで繰り返す．各PUには，**図9**のように分割したデータ配列を割り当てることで並列計算を行う．PUでデータ配列の境界部分を計算する場合に，隣接するPUで計算された値が必要になるので，各PUの境界データを隣のPUへNEWS転送を用いて転送する.

今回の実験では，データ配列のサイズを128x120，150回ループ(すなわち，計算回数は300回)で，PUの台数を1，2，4，6，8，10，12，14，16台使用

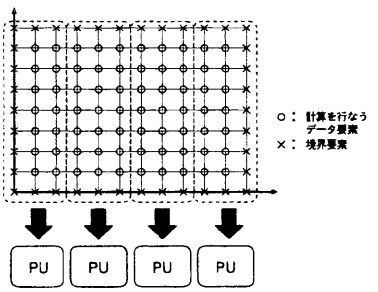

図9　データの分割

して行った．その結果を図10に示す．

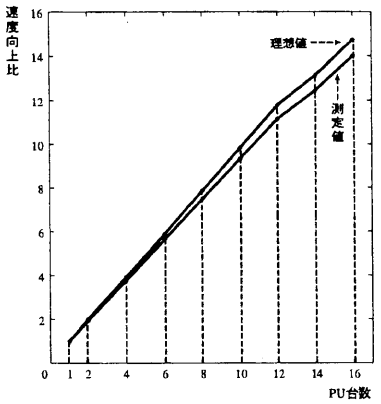

図10　速度向上比

図10でPU台数が14，16台の時に理想値，実測値共に少し下がっているのは，データを分割する際にデータ数がPU台数で割り切れないためである．図10を見てわかるように，実測結果は理想の場合とそれほど変わらない速度向上比が出ており，並列計算を行ったことによる台数効果が表れている．NEWS転送は，同じグループ内の同じ方向への転送は一括して行われ，同じグループに属するPU台数に依存しないので，今回のように転送の間隔が広く，転送量が大きいような場合には，理想値から極端に落ちずに高い速度向上比を得ることができた．FMMのNEWS転送機能は，各PUのメモリに対して直接読み書きを行うmemory-to-memoryで転送を行い，またシステム内に1台のDMAコントローラで制御することでハードウェア量を減らしているにも関わらず，今回のようなアプリケーションの場合には有効な結果を得た．

5. むすび

動的バリア同期管理機構は任意のPUからバリアグループを設定できる機構にとどめ，バリアグループの更新操作はソフトウェアの責任とした．通信機構については共有メモリ構成から局所メモリ構成まで段階的にメモリ構成を選択できる再構成可能共有メモリ，及び特定アプリケーション向けとしてNEWS転送を提案した．

バリアグループの変更頻度についての実験ではバリアグループの変更によるオーバーヘッドのために変更頻度には限界があるが，バリアグループの変更にかかるコストとプロセッサ利用率の間でトレードオフをとることでプロセスレベル，モジュールレベルにおいてバリアグループ変更による有効性が確認できた．また，ラプラス方程式でディリクレ問題を解く場合にNEWS転送を用いた結果，速度向上比は16PUで14.4倍となりその有効性が確認された．

今後の課題としてはバリアグループ変更の頻度を考慮したスケジューラの開発，および様々な特定アプリケーションによるNEWS転送の有効性の確認があげられる．

参考文献

1) Gupta, R.:The Fuzzy Barrier:A Mechanism for High Speed Synchronization of Processors,*Proc. Third Int. Conf. on ASPLOS, pp.54-63(Apr. 1986).*

2) 松本 尚:Elastic Barrier: 一般化されたバリア型同期機構, 情報処理学会論文誌, Vol. 32, No. 7, pp. 886-896 (1991).

3) 高木 浩光, 有田 隆也, 曽根 将容: 重複可能なバリア同期管理型同期のためのスケジューリングアルゴリズムとその性能, 電子情報通信学会研究会資料 CPSY91-15, pp. 91-97(1991).

4) M.T.O'Keefe and H.G.Dietz, Hardware Barrier Synchronization: Static Barrier MIMD (SBM),*1990 Int'l Conf. on Parallel Processing vol.I, pp. 5-42(1990).*

5) M.T.O'Keefe and H.G.Dietz, Hardware Barrier Synchronization: Dynamic Barrier MIMD (DBM),*1990 Int'l Conf. on Parallel Processing vol.I, pp. 43-46(1990).*

6) 岩根 雅彦, 濱田 智雄, 小島 和広, 松田 孝史: 式の分割による並列化アルゴリズム ESH, SWOPP'96(秋田)(1996)