計算機アーキテクチャ 116－6
（1996. 1. 26）

一般化されたコンバイニング機構

田中 清史　　松本 尚　　対木 潤　　平木 敬

東京大学大学院理学系研究科情報科学専攻
〒113 東京都文京区本郷 7-3-1
Email: {tanaka, tm, tsuiki, hiraki}@is.s.u-tokyo.ac.jp

大規模並列計算機システムにおいて効率の良い大域／細粒度処理を行うためには、ネットワーク上の通信量をできる限り削減することが必要である。従来の通信メッセージ(リクエスト)のコンバイニング技法は、ネットワーク上で複数の同一メッセージが衝突した場合にそれらを一つにすることで通信量を減らすことを目的としていたが、その限定された機能のために効率化に限界があった。本稿ではネットワークのスイッチングノードにおける到着条件、演算機能、マッチング条件の３項目を一般化することにより、コンバイニングの成功率を最大限に高める一般化されたコンバイニングを提案する。更にこの一般化されたコンバイニング機構を実現する高機能結合網を並列計算機プロトタイプお茶の水５号上で実装する。

Generalized Combining

Kiyofumi Tanaka　　Takashi Matsumoto　　Jun Tsuiki　　Kei Hiraki

Department of Information Science, Faculty of Science, the University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113, Japan.
Email: {tanaka, tm, tsuiki, hiraki}@is.s.u-tokyo.ac.jp

On a large-scale parallel computer system, the number of messages through an interconnection network should be reduced for the efficient large-area/fine-grained operation. Existing combining techniques aim at decreasing the number of requests by combining identical requests, which meet in the network, into a single request. But the efficiency is limited because of their restricted functions. In the paper, we propose *the generalized combining* which consists of three generalizations on switching nodes; that is, generalization of arrival requirement, generalization of processing function and generalization of matching requirement. This improves the success rate of matching. On OCHANOMIZ-5, a parallel computer prototype, we implement a high functional network that has the power of the generalized combining.

1 はじめに

大規模な分散共有メモリ型並列計算機システムには、本質的に局所性のない処理(プロセッサ間の通信、同期、メモリ管理など)が存在する。これらはプロセッサ間相互結合網を介して行なわれるが、結合網通信は高速な処理を行う要素プロセッサのリモートデータ要求に対して高レイテンシである。完全結合のネットワークでは、このデータ到着の遅延が低く押えられるが、大規模システムにこのような完全結合のネットワークを適用するのはハードウェアのコストや技術の面から現実的でない。従って、規模の大きなシステムには通信レイテンシのより大きい他のネットワークを採用せざるをえない。この場合、近傍による局所性を利用した最適化により大域通信量を押えることが効果的であるが、共有メモリ管理などではなおネットワークレイテンシよりも粒度の小さい大域/細粒度処理が存在し、並列化による効果を得ることが困難である。

これらの問題点の解決を目的として、ネットワークのノードにスイッチング(単純な通信)以外の機能を搭載した機能ネットワークによって通信量を削減することが提案されてきた[1]。すなわち通信メッセージのコンバイニングによって通信量を減らすネットワークであるが、それらは限定された機能のために効率化に限界があった。

本稿では、ネットワーク上の通信量削減における従来のコンバイニングを一般化することにより、コンバイニングネットワークに付随した問題点を除去し、可能な限り通信量を減らす方式を提案する。更にこの一般化されたコンバイニング機構を実現する高機能ネットワークのお茶の水5号上における実装を述べる。

2 背景

共有メモリをサポートする大規模並列計算機では、同一共有変数(もしくは同一メモリブロック)へのアクセスが、リクエスト先のメモリバンクやネットワーク内部のスイッチングノードで頻繁に衝突する可能性がある。これは"ホットスポット"と呼ばれる現象の大きな原因となり、ネットワークの飽和を引き起こす。この現象はプロセッサ台数に比例して顕著になり、パフォーマンスに大きく影響する。このアクセス競合を軽減する方法として、複数の同一メモリブロックへのリードリクエストがネットワーク内のスイッチングノードで衝突したとき、それらを一つにコンバイニングするネットワークが考案された。

NYUのUltracomputer[2]やIBMのRP3[3]でこのようなコンバイニングネットワークが提案されたが、これらは各スイッチングノードでのコンバイニング数が2に限定され、またコンバイニングのためのキューの長さも固定されたものであった。このように限定された機能の下では、コンバイニング可能なリクエスト同士の到着時間のずれ(リクエストが同時にノードに到着することは一般に保証されず、リクエスト同士の到着時間のずれがキューの長さを越える場合はコンバイニングが成立しない)、あるいはノード上での該当リクエストの数が可能なコンバイニング数を超過したことによるコンバイニング成功率の低下を招く。この場合、リクエストが衝突するのは該当メモリブロックのHOMEノードの近傍のみに限られ、全体としては各スイッチングノードのコンバイニングキューの通過時間によるレイテンシの増大が顕著になる。

Philip Bitar[4]はこの到着時間のずれによるコンバイニング率の低下を緩和するために、スイッチングノード上での待ち時間をコンバイニングウィンドウによって設定し、その時間内は先行するリクエストをノード内で待機させることが効果的であることを指摘した。またJUMP-1[5]では、分散共有メモリ管理のためのAcknowledgeメッセージの回収において、全てのAckメッセージが該当ノードに到着するまで待ち、そろった時点でそれらのコンバイニングを行う。この場合到着するメッセージ数が既知であるので、待つことによる余分なオーバーヘッドは生じない。

以上のものはメモリアクセスに関するものに限定されているが、一般に大域/細粒度処理はこの限りではなく、効率的な並列処理のためにはバリア同期やシストリックアレイ的なリダクション数値演算が必要となる。単一機能では対処可能な大域/細粒度処理に限界があるが、コンバイニング機能を一般化することによってこの限界を除去できる。次節で一般化されたコンバイニングを提案する。

3 一般化されたコンバイニング

3.1 3つの一般化

一般化されたコンバイニングは、以下の3項目の一般化からなる。

1. 到着条件の一般化
2. 演算機能の一般化
3. マッチング条件の一般化

到着条件の一般化

リクエスト(メッセージ)のノード上での待ち合わせ時間(コンバイニングが成立するための、先行するメッセージと後続するメッセージとの最大許容到着時間差)はゼロから無限大の間で設定可能(可変)とする。ゼロの場合はメッセージはノード上で待機することなく通過する。よってコンバイニングは行われない。無限大の場合は対象となるメッセージが全て到着するまでノード上で待機する。(予測可能なコンバイニングのためのもの。)この一般化により、待ち時間がコンバイニングキューの固定長に制限されるといった制約は受けなくなる。

演算機能の一般化

ネットワークの内部スイッチングノードは到着するメッセージに対してあらゆる演算機能を持つ。これにより、単

に同一アドレスへの複数のリードリクエスト(メッセージ)をコンバイニングするのみでなく、Fetch&Add[2]などの同期不可分命令やバリア同期、ネットワーク数値演算などの演算が可能となる。

マッチング条件の一般化

マッチング条件の一般化は次の2つからなる。

a) 任意数のコンバイニング

コンバイニングの対象となるメッセージの数を指定可能とする。これにより任意数のオペランドのリダクション演算などが可能となる。ただし、待ち合わせ時間を越えた場合は指定した数のコンバイニングは成立しない。このため指定した数のコンバイニングを必ず成立させたい場合は待ち合わせ時間を無限大に設定する必要がある。

b) 任意のマッチング key(ID)

任意のマッチング Key を指定可能とする。これにより、マッチングの対象となる Key は従来のメモリアドレス空間のみに制限されない。またコンバイニングの対象(相手)となるメッセージの ID(異なる ID 同士のコンバイニング)が指定可能となる。

3.2 特徴

一般化されたコンバイニングを実現するネットワークは、細粒度データフローを内部ノードに置いたものと類似しており、大域／細粒度処理に適している。データフローと異なる点として次のことが挙げられる。

- 不完全マッチングの存在

  待ち時間を越えた場合はコンバイニングは成立しない。メモリアクセスのリクエスト同士のコンバイニングなどは静的に予測不可能なため、この不完全マッチングが存在する。

- データの溢れ

  ノード上に到着して待機するメッセージ数は静的に予測不可能である。従って特に設定した待ち合わせ時間、コンバイニング数が大きい場合に待ちデータ数のメモリ量に対する爆発が起こる可能性がある。

- 多種多様な演算器

  厳密なデータフローは1つのノードにつき、単一命令演算を担当する。一般化されたコンバイニングでは、各ノード上で到着メッセージの種類により異なる複数の演算機能を持つ。

データの溢れについては、何らかの方法で対処する必要がある。現実的には大規模並列システムは NUMA 型であり、溢れを起こしたスイッチングノードに最も近い要素プロセッサが存在する。溢れが起こったことを検出した場合、そのプロセッサに割り込みによって処理を依頼することで対処可能である。

3.3 機能例

ここで一般化されたコンバイニングを実現するネットワークがサポート可能であり、超並列計算機システムの効率的処理のために必要な機能の例をいくつか挙げる。

分散共有メモリにおけるリモートメモリアクセスに伴う Acknowledge メッセージのコンバイニング

分散共有メモリでは、クラスタ間参照のコストを削減するために、リモートのデータをキャッシュすることが重要である。この場合、コンシステンシ管理のために Acknowledge メッセージ(Ack)が必要であるが、トランザクションの全ての要求先からの Ack を順次に処理すると通信のレイテンシが増大する原因となる。このことから Ack メッセージをコンバイニングによって効率良く回収[5]してネットワークの負荷を軽減することが要求される。この場合、到着する Ack の数はノードでマルチキャストした時点で既知であるので、コンバイニング数をその数にして、かつ待ち時間を無限大に設定し、回収を行うことが可能である。

リダクション数値演算機能

プロセッサ間で頻繁にデータの通信が必要な高並列計算において、ネットワークの通信速度の限界により台数効果が得られない場合がある。逆に通信データを利用してネットワークがシストリックアレイ的にリダクション数値演算を行うことにより計算の高速化が期待できる。ネットワークのスイッチングノードにおいて、待ち合わせ時間を無限大に設定し、演算機能を一般化することによって、あらゆるリダクション演算が可能となる。

階層化された Elastic Barrier

Elastic Barrier[6]は、並列実行の乱れが実行時間の遅れとして具体化するのを elastic 動作によって最大限に防止する外乱に強い静的バリア機構である。階層化された Elastic Barrier[7]は Elastic Barrier を大規模並列システムに拡張したものであり、多重発行した同期コマンドを制御する同期コントローラと、ハードウェアによるバリアのコンバイニングツリーによって実現される。このコンバイニングは、バリア同期の性質から待ち時間は無限大、マッチング Key はプロセッサグループ ID である。

メモリベース Elastic Barrier

共有メモリとコンバイニングネットワークによるメモリベース Elastic Barrier は、上記のハードウェアによる Elastic Barrier よりもオーバーヘッドが大きくなるが、ハードウェアによる場合のようなグループ構成に起因するスケジューリングの制約を受けない。よってより多重のグループ構成が可能となる。待ち時間は無限大、マッチング Key はグループと同期ポイントに対応して割り当てられた共有変数のアドレスである。

同期不可分命令のコンバイニング

演算機能の一般化により、Fetch&Add や Test&Set などの同期不可分命令のコンバイニングが可能となる。ただし到着条件の一般化により待ち時間を設定することによって、Gottlieb[2] の方法よりも計画的なコンバイニングが可能となり、成功率を上げることができる。

リモートメモリアクセスのリクエストコンバイニング

同一アドレスへのリードリクエストのコンバイニングは、ネットワーク上の通信量を減らすことができ、ホットスポットの軽減になる。リプライされたデータは、コンバイニングを行ったノードにおいてリクエスト元にマルチキャストされる。このコンバイニングにおいて、マッチング Key はメモリアドレスである。

4 お茶の水 5 号

4.1 設計方針

お茶の水 5 号 (OCHANOMIZ-5: Omnipotent Concurrency-Handling Architecture with Novel OptiMIZers-5) は、平木研究室における並列処理プロジェクト (お茶の水プロジェクト [8]) の第 5 号プロトタイプ計算機であり、スケーラブル並列計算機システムの性能評価および並列計算支援機構の評価を目的としている。主に、プロセッサベース同期機構、メモリベース同期機構、ハードウェアによる分散共有メモリ [9]、および一般化されたコンバイニングを実現する高機能ネットワークの評価のためのテストベッドとして位置付けられる。

4.2 全体構成

図 1にお茶の水 5 号の全体構成を示す。処理の局所性を利用した高速化、システムのスケーラビリティの確保という観点から階層構造を採用している。また、プロセッサとメモリの間のバンド幅の確保のために、主記憶は分散メモリ実装となっている。現在のところ、図のように 4 つのクラスタ基板およびそれらの間の結合網を実現するためのネットワーク基板からなっている。なお、ホストコンピュータと接続し、外部との入出力を行なう。

4.3 クラスタ内構成

クラスタ内には、2 つの PE(要素プロセッサ) が実装される。PE として SuperSPARC+ を使用する。これらがクラスタ内で共有バス (MBus： SPARC 専用バス) 共有メモリ型マルチプロセッサを構成する。2 次キャッシュの容量は PE 当たり 1Mbyte で、クラスタ内主記憶は 16Mbyte の DRAM モジュール 2 つからなる。その他に、メモリコントローラ、共有バスの使用権を調停するアービター、ネットワークへのインターフェースのそれぞれに、Xilinx 社の FPGA を使用している。

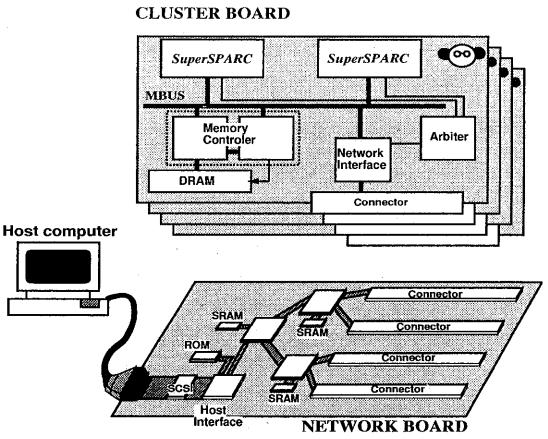

図 1: お茶の水 5 号の構成

4.4 クラスタ間ネットワーク

お茶の水 5 号のネットワークは 2 進木階層構造の形態をとり、各内部スイッチングノードに FPGA を使用することにより、一般化されたコンバイニングを実現する高機能結合網を構成する。ネットワークノード間のデータの転送幅は 8 ビットずつ両方向にあり、各ノードにはそれぞれ、マッチングメモリといった機能を実現するために使用される SRAM が外部に用意されている。

5 一般化されたコンバイニングの実現

お茶の水 5 号のクラスタ間ツリー状ネットワークは、一般化されたコンバイニング (ただしハードウェア量の削減のためその部分的実現) によって次の機能を実現する。

- インバリデートリクエスト及び Ack メッセージのコンバイニング
- 階層化された Elastic Barrier
- メモリベース Elastic Barrier
- Fetch&Add、Test&Set のコンバイニング
- リードリクエストコンバイニング
- リダクション数値演算

上の機能において、インバリデートリクエスト及び Ack メッセージ、リードリクエストのコンバイニングについてはキャッシング方式の共有メモリブロックに対して行う。メモリベース Elasitc Barrier、同期不可分命令のコンバイニング及びリダクション数値演算はノンキャッシュな共有メモリ空間の変数に対して行う。なお、階層化された Elastic Barrier についてはそのためのハードウェアを付加して実現する。

到着条件の一般化を実現

待ち時間は4段階(レベル0～レベル3)で設定可能とする。これは通信パケット内の2ビットで指定する。レベル0は待ち時間ゼロ、レベル3は待ち時間無限大である。レベル2はスイッチングノード上でリクエストを一定時間待たせることを意味する。実際の待ち時間はスイッチングノードの再構成により可変である。レベル1については、待ち時間はゼロであるがコンバイニングユニットを通過し、リクエストならウェイトバッファー内にエントリをセット(又は更新)、リプライならばコンバイニングの有無(リプライ先のアドレス)を調べてからエントリをクリアする。ウェイトバッファー内に該当エントリがある間は後続する同種のリクエストはコンバイニングされる。これは予測不可能なコンバイニング(お茶の水5号ではリードリクエスト及びインバリデートリクエストのコンバイニング)に対する、低オーバーヘッドかつ到着時間の差が最も緩和された方法である。

再構成による演算機能の一般化の実現

全ての演算機能を同時に実現するのはハードウェアの制約から困難である。再コンフィグレーション可能なFPGAを使用して、アプリケーションの高速実行に最も適した演算機能を再構成によって実現する。

マッチング条件の一般化の部分的実現

お茶の水5号においてネットワーク構造は2段の2進木であることから、3つ以上のメッセージのコンバイニングの可能性は低いことが予測される。このことからコンバイニング数を2以下に制限する。よってコンバイニング数の指定は待ち時間の指定(level-0 or others)で置き換え可能である。

マッチングKeyとして、階層化されたElastic Barrierの場合はプロセッサID、その他の場合は物理メモリアドレス空間を使用する。

ネットワークノードの構成

ネットワークの内部ノードは、コンバイニング機能を実現するために以下の要素から構成される。(図2)

スイッチ要素およびそのコントローラ(Router)

ノードの親、2つの子、及びコンバイニングユニットの間で4×4クロスバースイッチを構成する。

外部メモリで構成するマッチングストア

レベル1～3用のウェイトバッファー、及びレベル2用の待ち合わせキューのためのメモリ空間を外部のSRAMによって確保する。ウェイトバッファーは1つのマッチングキーについて1エントリが割り当てられ、エントリを更新することによりマッチングストアとして使用する。なお、オーバーヘッド削減のためハッシュ表を構成して高速探索を行う。

演算器とコントローラ

スイッチからコンバイニングユニットに入ってくるデータをレジスタにセットし、レジスタ-メモリ演算を行う。再構成により異なる演算を可能とする。

溢れ検出回路

各ハッシュキーに割り当てられた空間ごとに溢れをチェックする。溢れを検出した場合は、割り込みパケットを生成し、近傍のプロセッサへ送る。

階層コントローラ(HC)、グループレジスタ

階層化されたElastic Barrierを実現するために、階層コントローラと同期グループ(マッチングキー)を指定するグループレジスタを持つ。

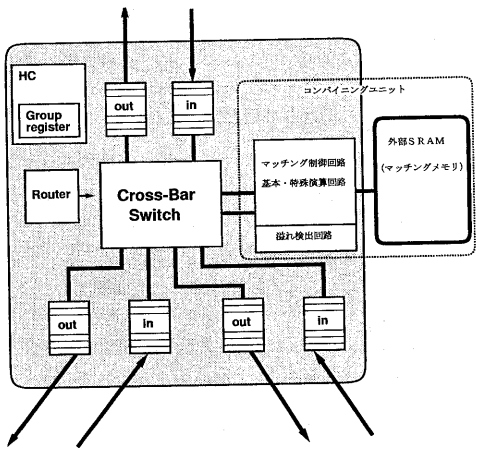

図2: ネットワーク内部ノードのブロック図

階層化放送機構

お茶の水5号はハードウェアによる分散共有メモリをサポートする。ネットワークはそのキャッシュコンシステンシ管理における効率化支援のために、階層化された放送(ブロードキャスト、マルチキャスト)機能を持つ。ネットワークは2進木なので、スイッチングノードにおいて放送はオン、オフの2種類に限られる。

6 コンバイニングの処理例

お茶の水5号のツリーネットワークによるコンバイニング処理の例として、分散共有メモリ[9]のキャッシュコンシステンシ管理のためのコヒーレントインバリデートトランザクションの処理を取り上げる。このリクエストは、クラスタ内のCPU内蔵キャッシュまたは2次キャッシュがキャッシュしているexclusiveでないブロックにCPUが書き込みを行う際に発行される。なお、お茶の水5号では

ブロックの共有情報の一つとして、そのブロックのホームノードが共有最大距離(そのブロックを共有する最も離れたクラスタまでの距離)を保持する。

以下で共有メモリ空間のshared状態のブロックに対するインバリデートリクエストのネットワーク内トランザクション処理を述べる。

1. CPUが共有ブロックへのコヒーレントインバリデートを発行した場合、そのクラスタノードは該当ブロックのホームノードへレベル1のインバリデートリクエストを発行する。

2. スイッチングノードにこのリクエストが到着したとき、ウェイトバッファーを探索する。先行する同じリクエストがなかったらウェイトバッファーにエントリを格納し、(レベル1であることより)リクエストを要求先にフォワードする。既に先行リクエストがあった場合、後から到着した方のリクエストはそこで放棄される。(これはコンバイニングと考えることができ、ホームノードにおけるリクエストFIFOキューでの同様の処理を除去できる。)

3. ホームノードはこのインバリデートリクエストを受け取ると、該当エリア(共有距離内)にレベル0のインバリデートリクエストを発行(放送)する。

4. スイッチングノードにこのインバリデートリクエストが到着したとき、該当エリアにマルチキャスト(ただし共有エリアのルートならば他方の子ノードへのみ送信し、ウェイトバッファーにレベル3のダミーAckエントリを挿入)する。

5. このリクエストを受け取ったクラスタノードは、クラスタ内でインバリデート処理を行い、レベル3のAckメッセージを返送する。実際にはそのブロックコピーが存在しないクラスタや最初の要求元クラスタは、ダミーAckを返送する。

6. スイッチングノードにこのAckが到着したとき、ウェイトバッファーを探索し、先行Ackがなければエントリを挿入する。先行Ackがあればホームノードの方向へフォワードし、エントリをクリアする。

7. ホームノードは該当Ackが返送されてきたら、最初の要求元にレベル1のAckメッセージをリプライとして返送する。

8. スイッチングノードにこのAckが到着したとき、エントリをクリアし、要求元へフォワードする。

9. 要求元にAckが到着し、トランザクションが終了する。

7 おわりに

本研究において、大規模並列計算機上で効率のよい大域／細粒度処理を実現する一般化されたコンバイニングを提案した。これはネットワークのスイッチングノードにおける、到着条件、演算機能、およびマッチング条件の3項目の一般化からなる。更に、一般化されたコンバイニングを実現する高機能結合網を設計し、お茶の水5号上に実装した。今後の課題としては、実機上で実用レベルの規模のプログラムを動かし、性能評価を行なう。

謝辞

本研究は通商産業省RWCプロジェクトの一環として行われた。なお、お茶の水5号の開発に当たり、協力していただいた日本サンマイクロシステムズ株式会社ならびにデジタルテクノロジー株式会社に深く感謝の意を表します。

参考文献

[1] Herbert Sullivan, Theodore. R. Bashkow, David Klappholz : A Large Scale Homogeneous Fully Distributed Parallel Machine. Proc. Fourth Symposium on Computer Architecture. pp.105–124 (1977).

[2] Allan Gottlieb, Ralph Grishman, Clyde P. Kruskal, Kevin P. McAuleffe, Larry Rudolph, and Marc Snir: The NYU Ultracomputer-Designing an MIMD Shared Memory Parallel Computer. IEEE Transactions on Computers, 32(2):175-189 (February 1983).

[3] Gregory F.Pfister and V.Alan Norton : "Hot Spot" Contention and Combining in Multistage Interconnection Networks. IEEE Transactions on Computers, 34(10):943-948 (October 1985).

[4] Philip Bitar : Combining Windows: The Key to Managing MIMD Combining Trees. The workwhop on Scalable Shared Memory Multiprocessors. (May 1990)

[5] 松本尚, 平木敬: 超並列計算機上の共有メモリアーキテクチャ. 電子情報通信学会技術研究報告, CPSY92-26. pp.47–55 (August 1992).

[6] 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 (July 1991).

[7] 松本 尚, 平木 敬 : 拡張されたSnoopy Spin Waitと階層化されたElastic Barrier. 第47回情報処理学全国大会講演論文集 (4), pp.43–44 (October 1993).

[8] 平木 敬, 松本 尚, 稲垣 達氏, 大津 金光, 戸塚 米太郎, 中里 学 : 細粒度並列計算機お茶の水1号 — 基本構想 —. 第47回情報処理学会全国大会講演論文集 (6), pp.55–56 (October 1993).

[9] 対木潤, 田中清史, 松本尚, 平木敬 : スケーラブル並列計算機プロトタイプ : お茶の水5号. 情報処理学会研究報告, ARC Vol.95, No80, pp.25–32 (August 1995).