多重出力可能なMINの命令レベルシミュレータによる評価

田辺 靖貴 †, 緑川 隆 †, 白石 大介 †, 茂野 真義 †,
金森 勇壮 †, 塙 俊博 ‡, 天野英晴 †
† 慶應大学 横浜市港北区日吉 3-14-1
‡ 東京工科大学
snail@am.ics.keio.ac.jp
May 7, 2003

Abstract

Abstract

本研究室では、SSS 型 MIN を提案し、並列計算機 SNAIL での評価を行ってきたが、ネットワークのバンド幅はプロセッサの性能の向上に伴い依然として不足することが明らかになった。そこで、まず、一般的なバンヤン網を階層的に配置し、多重出力可能にしたネットワークトポロジ PBSF を提案し、次に、キャッシュ機構を持たせるために、高速かつ効率良くキャッシュ制御パケットをマルチキャストするネットワーク MINC を提案し、これを用いた並列計算機 SNAIL–2 を実装してきた。

本論文では、PBSF、MINC の性能評価のために作られた並列計算機 SNAIL-2 の命令レベルシミュレータをもちいて、多重出力可能な MIN の評価を行なった。評価の結果、PBSF トポロジの SSS 型 MIN は通過率にすぐれ、レイテンシを抑えることにより優れた性能をしめす事がわかった。

1 はじめに

並列計算機の構成要素の中でも、プロセッサ同士を結合する結合網は性能に大きな影響を及ぼすため、システムの目的に応じて様々な方式が提案されている。その中で、数 10〜数 100 プロセッサクラスの中規模並列計算機において有効な結合網として、多段結合網 (MIN: Multistage Interconnection Network) が検討されている。

MIN は、2×2 から 8×8 程度の小さなクロスバスイッチを多段結合することにより構成され、規模拡張性に優れている。しかし、そのハードウエア量に見合った性能を得ることが難しいために、並列計算機への実装が遅れてきた。

そこで我々は、パケットを数ビット幅にシリアル化してフレームに同期させて転送することにより、高速かつ実装が容易なプロセッサ・メモリ間結合網、SSS(Simple Serial Synchronized) 型 MIN[1] を提案した。また、この SSS 型 MIN に基き、多重出力可能なネットワークトポロジである TBSF を用いて並列計算機 SNAIL[2] を実装し、評価を行ってきた。これによって、SSS 型 MIN が高い転送能力と実装効率を併せ持つことが実証された。

しかし、SNAIL で用いられた TBSF トポロジの SSS 型 MIN は、近年のプロセッサの高速化に伴い、パケットの再送による転送能力の低下、ネットワーク通過時のレイテンシが大きいことなどの問題が明らかとなった。

そこで、バンヤン網を三次元的に配置することにより通過率が高くレイテンシの小さなネットワークトポロジである PBSF(Piled Banyan Switching Fabrics)[3] を提案し、SSS 型 MIN として適応させた PBSF チップの実装を行なった。

一方共有メモリアクセスの実効レイテンシを小さくするために、MIN を用いた並列計算機にキャッシュを持たせる試みがなされた。しかしながら、従来の方式では、ディレクトリ管理に大量のメモリを必要とし、ハードウェア的に実現するのが困難であった。そこで、縮約階層ビットマップディレクトリ方式 [4] [5] を用い、高速かつ効率のよいキャッシュ制御を行なうことのできる MINC(MIN with Cache control mechanism)[6] を提案し、実装を行なった [7]。

現在 PBSF と MINC を用いたスイッチ結合型並列計算機 SNAIL–2 は、4 プロセッサで稼働しており実機による評価が行なわれている [8]。

本報告では、PBSF や MINC の、サイズや構成を変えた場合の評価を行なうため、並列システムシミュレータ構築環境 ISIS[**?**] を用いて SNAIL–2 の命令レベルシミュレータを開発した。このシミュレータにより構成やサイズを変更した場合の MIN とキャッシュ制御機構の評価を行う。

2 SSS 型 MIN

2.1 SSS 型 MIN の基本動作

SSS 型 MIN の基本構造を図 1 に示す。プロセッサからのアクセスは MIN との間のバッファにより、1〜4bit 程度にシリアル化され、フレームクロックに同期してネットワークに入力される。

各スイッチングエレメントはパケットバッファをもたず入力されたパケットのタグ情報を参照してスイッチを決定する機能だけを持つ。従ってスイッチングエレメントの構造は大変単純であり、ネットワークは全体としてシフトレジスタのような動作をする。

ネットワーク内ではパケットが投入された数クロック後からクロック毎に一段ずつスイッチの状態が決定され、MIN の入力から出力を辿る経路 (以後トレース) が形成される。スイッチに入力された二つのパケットが同一出力に向かった場合、一方のパケットは正しく転送されるが他方のパケットは正しく転送されずにデッドパケットとして扱われる。

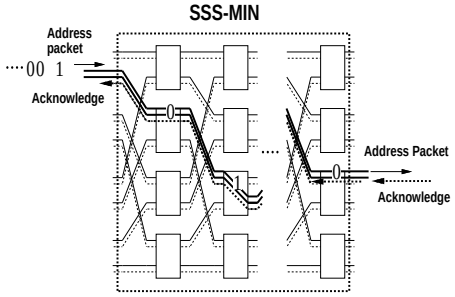


Figure 1: SSS-MIN の概観

出力側では到着したパケットが正しく転送されたかに応じて、ACK(Ackknowledg) もしくは、NAK(Not AcKnowledge) をトレースを辿って転送する。これによってパケットが正しく転送されたか、途中で衝突が発生しデットパケットになったかを入力側に通知する。NAK を受け取った入力側は次のフレームクロックに同期して再びパケットを入力することによって正しく転送されなかったパケットを再送する。

2.1.1 パイプライン化サーキットスイッチング

SSS 型 MIN では、アドレス、データ、アクノリッジの転送路は独立しており、アドレス転送によってトレースが設定されると、アクノリッジ及びデータパケット転送のトレースも決定される。

図 2 のように、フレーム i でアドレスが転送されトレースが形成されると、同時に各入力に対してアクノリッジ信号が返される。正しく転送が行なわれ、ACK が返送された場合のデータ転送はトレースを利用し、フレーム i+1 でのアドレス転送にオーバーラップして行なわれる。

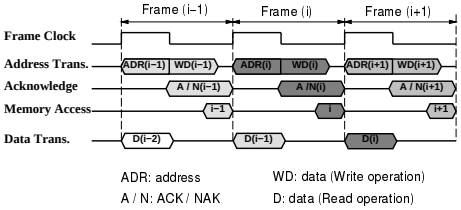


Figure 2: アドレス・データ・アクノリッジ転送タイミング

3 多重出力可能な MIN

3.1 TBSF

TBSF(Tandem Banyan Switching Fabrics) は、本来 B-ISDN(Broadband Integrated System Digital Network) で用いられる ATM(Asynchronous Transfer Mode) パケット交換用に、国内では本研究室と沖電気の共同研究により 1988 年 [9] に、海外では Tobagi らにより 1990 年 [10] に提案された網である。

TBSF は、図 3 に示すようにバンヤン網 (omega 網) を直列に接続し、各網の出口にバイパス路を設けた構造を持つ。バンヤン網を通過して目的の宛先に到着したパケットはバイパス路によりメモリモジュールに送られ、衝突により目的の宛先に到着できなかったパケットのみが次の段のバンヤン網に入力される。よって TBSF では 1 チャネルあたりの出力は直列に接続された網数の数分多重化される。本稿では、この TBSF トポロジの SSS 型 MIN によるデータ転送用ネットワーク (以降 TBSF) を後述する PBSF トポロジの SSS 型 MIN との比較のために用いる。

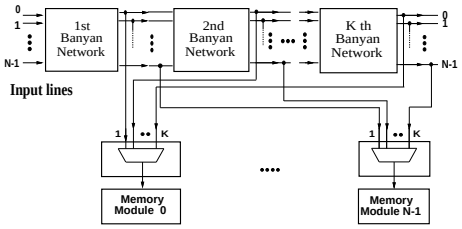


Figure 3: TBSF(Tandem Banyan Switching Fabrics)

3.2 PBSF

TBSF では、接続された各網において、パケットが衝突するまでは正しくルーチングされるにもかかわらず、それまでのルーチングの結果は次の網に対して全く貢献しない。このため直列に接続する網数に比例しネットワークレイテンシの増大が増大する。

この問題点を改良するため、図 4 に示すようにバンヤン (omega) 網を階層的に接続した構造に変更し、PBSF(Piled Banyan Switching Fabrics)[11] と名付けた。

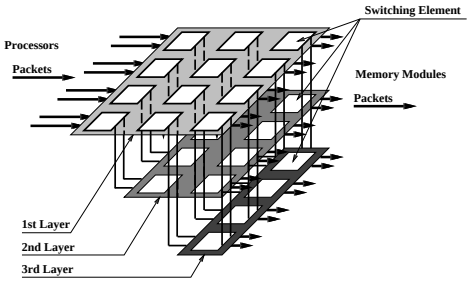


Figure 4: PBSF (Piled Banyan Switching Fabrics)

PBSF では、パケットはまず最上層のネットワークに入力され、あるスイッチエレメントで 2 つのパケットが衝突すると、片方のパケットは希望の方向に送られ衝突に敗れたパケットは一つ下の層のエレメントに送られる。2 層目以下のスイッチエレメントでは水平方向からの入力に加え上層からの入力で、最大 4 入力が 1 つの出力を競合する。この場合には 1 つは正しい出力へ、下層がまだ存在する場合は、もう 1 つのパケットを下層へ出力し、出力することができない残りのパケットはエレメント内で消滅する。いずれの場合も、パケットが消滅した場合にはプロセッサ側に NAK が返されるようになっており、消滅したパケットはネットワークインタフェースによって次のフレームで再送される。

よって PBSF は、最上層のスイッチでは 2 入力 4 出力、最下層では 4 入力 2 出力、それ以外では 4 入力 4 出力のスイッチエレメントを用いて構成される。

このため、ネットワークの出力は 1 チャネルごとに、最下層以外のスイッチエレメントは 2 出力、最下層は 1 出力となり、利用する層の数を n とすると出力は、(n - 1) * 2 + 1 で多重化される。

4 キャッシュ制御機構 MINC

4.1 縮約階層ビットマップディレクトリ方式

縮約階層ビットマップディレクトリ (RHBD: Reduced Hierarchical Bit-map Directory) 方式 [4][5] は、超並列マシン JUMP–1 のディレクトリ制御用に考案されたビットマップの縮約方式である。階層ビットマップ方式ではプロセッサ数が増えるにつれ、ディレクトリ管理に必要なメモリ量が膨大になってしまうが、RHBD 方式を用いることにより、ディレクトリに必要なメモリ量を節約することができる。

キャッシュ情報を管理するためのビットマップは各スイッチングエレメントに置かず、共有メモリに RHBD 方式により縮約されたビットマップの形で置かれる。キャッシュラインの無効化要求をマルチキャストをする際は、このビットマップに従って要求が転送する。

RHBD では、いくつかのディレクトリ縮約方式が提案されているが、今回は SM〔Single Map〕法を用いる [6]。SM 法とは、各階層ごとにその階層の全ての節のビットマップの論理和をとり、その階層の全ての節で用いる方式である。図 5 は 3 進木を用いた模式図で、**s** が送信元のプロセッサ、**d** が本来の送り先である場合 Level0 で 100、Level1 で 011、Level2 で 110 を用いることにより必要なプロセッサにパケットを送ることができる。この時、**d** の無い • は無駄なパケットを受けとる。

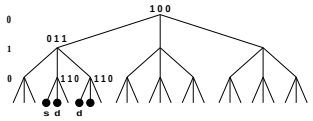


Figure 5: RHBD 方式 (SM 法)

4.2 キャッシュ制御ネットワーク

キャッシュの一貫性の保持のためのキャッシュ制御パケットは、データ転送用ネットワークとは別の SSS 型 MIN のキャッシュ制御ネットワーク (MINC) によって、RHBD 方式によって縮約されたビットマップに従いメモリモジュールから、プロセッサ側に転送される。データ転送用ネットワークは 2x2 のスイッチエレメントを用いるのに対し、キャッシュ制御ネットワークのスイッチエレメントは、4x4 のスイッチエレメントを用いて、全体のネットワークを構成する。

5 ISIS

ISIS[12] は、並列システムの性能評価、プログラム開発用シミュレータを構築するための C++言語用のライブラリツールである。

図 6 にライブラリ構成を示す。

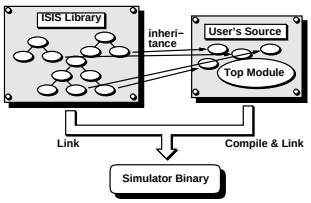


Figure 6: ISIS のライブラリ構成

ISIS では、プロセッサ、メモリ、バス等の並列計算機を構成する代表的な部品の挙動をクロック単位で記述した機能ブロック (ユニット)、機能ブロック間の接続のためのインターフェースとしてポート、送受信される情報を表わすパケットが、それぞれ基本要素としてクラスで提供されている。

ISIS を用いてシミュレータを実装する場合、必要ならばそれぞれのクラス階層から必要なクラスを取り出しその派生クラスで意図した機能を実装し新たな機能ブロックを構成し、また、ISIS で提供される機能ブロックを利用し、それぞれの機能ブロック間の接続を記述してゆくことにより、比較的容易にシミュレータを構成してゆくことができる。

6 SNAIL–2

スイッチ結合型並列計算機 SNAIL–2 は、PBSF トポロジの SSS 型 MIN と MINC の評価を目的に設計、実装された [7]。本稿では、5 節で示した命令レベルシミュ

レーションライブラリ ISIS を用いて構成された SNAIL–2 のシミュレータを利用して評価を行う。

図 7 に SNAIL–2 の構成を示す。SNAIL–2 は、実機では最大で 16 個のプロセッシングユニット (PU) と 16 個のメモリモジュール (MM) から構成され、それぞれ、PBSF トポロジの SSS 型 MIN によるデータ転送用ネットワークと、キャッシュ制御用ネットワークに用いる MINC ネットワークに接続される。

また本稿で用いるシミュレータではデータ転送用ネットワークとして、TBSF トポロジの SSS 型 MIN や、SSS 型でない通常の MIN も用いることができる。

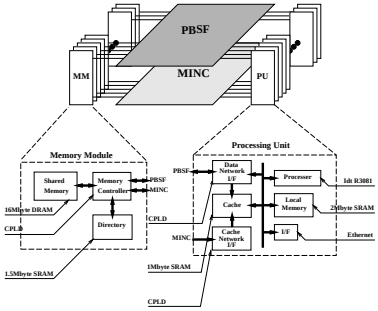


Figure 7: SNAIL–2 の構成

PU と MM 間のデータ転送、キャッシュ制御は次のようにして行われる。

- データ転送
  プロセッサから共有メモリに対するアクセスは、PU 内の PU コントローラと MM 内のメモリコントローラ間で、PBSF のネットワークを通してパケットを転送することで行われる。PU には共有メモリのキャッシュが存在し、このキャッシュを利用する場合には共有メモリアクセスのレイテンシを小さくすることができる。
- キャッシュ制御
  共有メモリへの書き込みが起こった際、PU コントローラは、PU 内のキャッシュにそのデータが存在しているのならそのデータを無効にし、PBSF インタフェースを通して MM へパケットを転送する。MM 内のメモリコントローラは、書き込みが行なわれたデータをキャッシュしている PU へ、コヒーレンス維持のためのパケットを MINC のネットワークを通して転送する。PU コントローラはこのパケットを受け取り、キャッシュにそのデータが存在しているのならそのデータを無効化する。

7 評価

SNAIL–2 の命令レベルシミュレータを用い、最大 64PU でのパフォーマンス、PBSF、TBSF データ転送用ネットワークの転送性能、キャッシュ機構の評価を行なった。

評価用アプリケーションとしては、Radix, FFT, LU の 3 アプリケーションを用い、基本的な評価条件は表 1 の通りである。また各アプリケーションのデータセットのサイズは、Radix が 131072Key、FFT が 2^{16}、LU が 192x192 である。

7.1 PBSF トポロジの SSS 型 MIN

7.1.1 台数効果

データ転送用ネットワークとして、PBSF を用いた SNAIL–2 の台数による性能向上比を図 8 に示した。

Table 1: シミュレーション環境

PU 数	1〜64
Cache	
size	256KB / PU
way 数	2-way
line size	32 byte
データ転送用ネットワーク	
レイヤ数	2-layer
フレームクロック	40 clock
Link 幅 (PU → MM)	16 bit
Link 幅 (MM → PU)	8 bit

* ただし、TBSF の場合には網数が 1 段の場合にフレームクロックを 40lock とし、2,3,4,5 段の場合にはそれぞれ、60,80,100,120 clock とした。

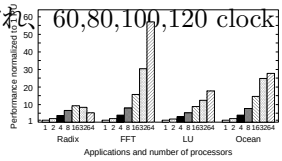
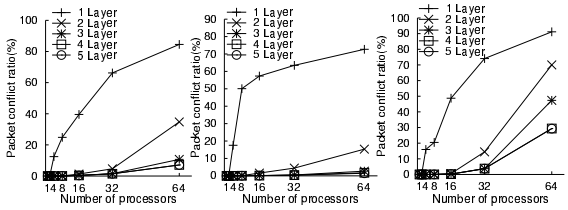


Figure 8: 台数効果 (With MINC)

結果、PBSF を用いた SNAIL–2 は、16PU 程度までは台数効果が得られ、64PU 規模でも Radix 以外のアプリケーションでは性能向上が得られ、特に FFT では 64PU 時に 1PU 時の 60 倍近い実行時間の短縮が見られ優れた性能向上が得られることがわかる。

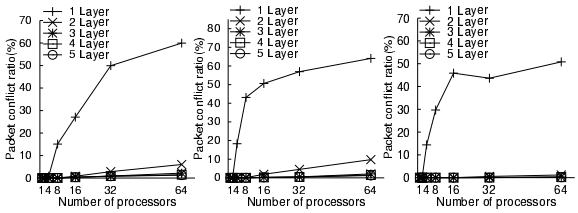
7.1.2 レイヤ数の違いによる評価

図 9 と図 10 に PBSF のレイヤ数を変化させた場合の衝突率の変化を、MINC を使用した場合としなかった場合 (キャッシュを使用した場合としなかった場合) について示した。



a) Radix b) FFT c) LU

Figure 9: 衝突率の変化 (Without MINC)



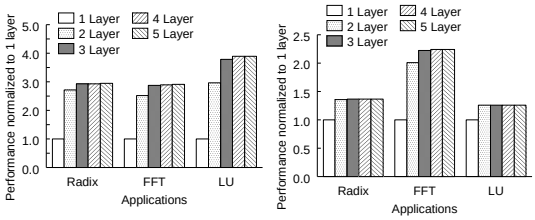
a) Radix b) FFT c) LU

Figure 10: 衝突率の変化 (With MINC)

PBSF ではレイヤ数を 2 段にすることでどのアプリケーションでもパケット衝突率を大幅に減らせることができ、3 段にした場合では 2 段の時と比較してもそれほど変化がないことが分る。

また 64PU 使用時のレイヤ数の違いによる実行時間の変化を図 11 に示した。

実行時間でもレイヤ数を 2 段にすることにより性能の向上が大きくみられ、段数をそれより増しても実行時間での性能向上はあまりないことが分る。これより、PBSF のレイヤ数は 2 段の時が最も効率がよいことがわかった。



a) Without MINC　　b) With MINC

Figure 11: レイヤ数による実行時間の変化

7.1.3 MINC による効果

キャッシュ未使用時には 2 段のレイヤ数でも 64PU 時で高い衝突率であった図 9 c) の LU でも、キャッシュを使用することにより、図 10 c) のように衝突率が大幅に減っている。

これはほかのアプリケーションにおいても同様であり、MINC を用いてキャッシュを使用することにより、ネットワークの負荷が軽減されパケットの衝突率を改善できることがわかる。

特に同期のための Spin-lock 時には、複数の PU から同一のメモリモジュールへのアクセスが集中するが、キャッシュを使用した場合は、Spin-lock 対象のラインが更新され MINC ネットワークによって各 PU でキャッシュされたラインが無効化されるまで、読み込みは各 PU 内のキャッシュから読みこむようにできるので、Spin-lock によるネットワークの混雑を回避できる。

このように、共有データのキャッシングによってネットワークへのアクセスを減らせ、さらにもっとも衝突の起きやすい同期時のネットワークの混雑が回避できるのでキャッシュ機構は非常に有効であると言える。

図 12 は、16 と 64PU 使用時にキャッシュを使わなかった場合とキャッシュを使用した場合の実行時間の向上をキャッシュ未使用時で正規化したものである。これからもキャッシュを利用することにより実行時間でも、1.1 から 2 倍の性能の向上が得られることが分る。

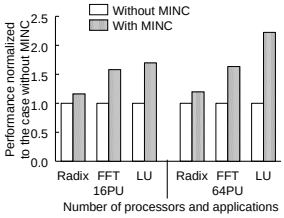
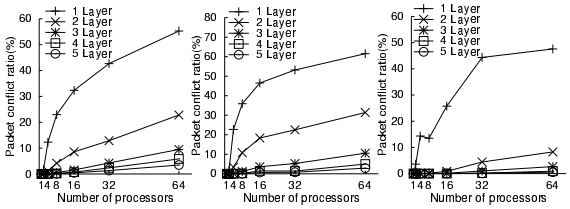


Figure 12: キャッシュによる実行時間の変化

7.2 TBSF トポロジの SSS 型 MIN

PBSF と同じように多重出力が可能な MIN である TBSF について、7.1.2 項と同じように直列に接続するバンヤン網の数の違いによる衝突率の変化を図 13 に、実行時間での性能向上を図 14 に示した。ただし、どれも MINC を用いてキャッシュを利用した場合の結果のみを示している。

TBSF の場合でも、衝突率では PBSF と同じようにバンヤン網数を 2 段にした時に最も改善している。 しかし、実行時間で見ると網数を 2 段にした場合、FFT では性能の向上が見られるが、Radix や、LU ではあまり変化がないか、むしろ性能が落ちており、網数が増えるにつれ性能は悪くなる傾向にある。



a) Radix b) FFT c) LU

Figure 13: 衝突率の変化 (With MINC)

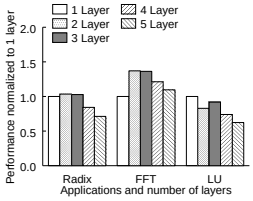


Figure 14: 実行時間の変化 (With MINC)

これは、PBSF トポロジではバンヤン網は階層的に配置されており、網数が増えてもフレームクロックの長さは変わらないのに対し、TBSF トポロジでは網が直列に接続されており、パケットが直列に接続された網を通過するのに必要なタイミングに合せフレームクロックを長くする必要があり、この結果、共有データへアクセスした際のレイテンシが増大し、パフォーマンスに影響を与えているからである。

7.3 各種データ転送用ネットワークの比較

図 15 にデータ転送用ネットワークの比較として、16 と 64PU 時のそれぞれにおいて、ワームホールルーティングを行う一般的な MIN(以後 MIN と記述する)、TBSF、PBSF トポロジによる SSS 型 MIN をデータ転送用ネットワークとして用いた場合について、それぞれの実行時間を MIN の場合で正規化して示した。ただし、TBSF と PBSF では MINC を利用しキャッシュを用いた場合についても評価を行なっているが、MIN の場合には MINC によるキャッシュ制御機構を用いることができないため評価はキャッシュを利用しない場合についてのみ行なっている。

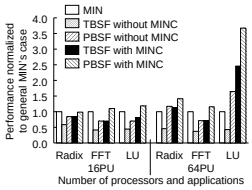
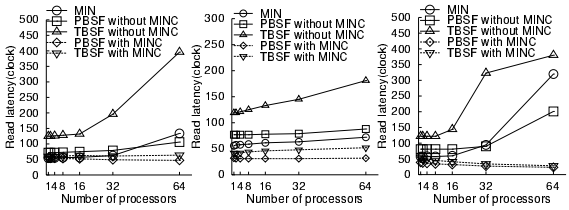


Figure 15: 実行時間の変化

これを見ると、16PU 規模では MIN と比較して、キャッシュを用いた PBSF でわずかに性能が向上しているに留まり、そのほかのキャッシュを用いていない TBSF や PBSF では MIN の方が良い性能を示すことがわかる。また 64PU 利用時では PBSF はキャッシュを利用しない場合では、MIN と比較し、FFT 以外のアプリケーションでは優れた性能をしめし、キャッシュを利用することによって、FFT でも MIN の場合よりも性能が向上しており、とくに LU では優れた性能を示すことがわかる。

図 16 はこの時の各種ネットワークを用いた場合の読み込み要求のレイテンシを比較している。 これを見ると、64PU 時で MIN より優れた性能を示すアプリ



a) Radix b) FFT c) LU

Figure 16: リードレイテンシの比較

ケーションは、64PU 時でのレイテンシが MIN と比べ優れていることがわかる。とくに LU においては、PU 数が増えるにつれて MIN のレイテンシも増えるのに対し、PBSF ではそれほどレイテンシが増加していない。またキャッシュを利用することによってレイテンシが効果的に隠蔽されていることがわかる。

7.4 レイテンシの影響

7.3 項で、64PU 規模の時のようにネットワークへの負荷が高い場合においても、PBSF は優れた転送能力を持つこと示した。しかし、1〜16PU 程度のネットワークへの負荷がそれほど高くない場合には、MIN と比較してレイテンシにおいて不利であり、実行時間でも MIN を利用した場合よりもわずかに優れるか、もしくはそれ以下である。

これは SSS 型 MIN ではネットワークが混雑していない場合でもフレーム同期のための待ち時間が必要であり、またメモリ側からの返信もフレームに同期させる必要があり、レイテンシが MIN よりも長くなってしまいこれがパフォーマンスに影響しているためである。

そこで、SSS 型 MIN は通常の MIN よりも構造がシンプルであるため高速動作するネットワークを構成することが可能であると仮定し、システム部のクロックの 2 倍のクロックをネットワーク部で用いるとして評価を行い、実行時間の比較を図 17 に、レイテンシの比較を図 18 に示した。

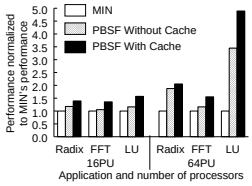
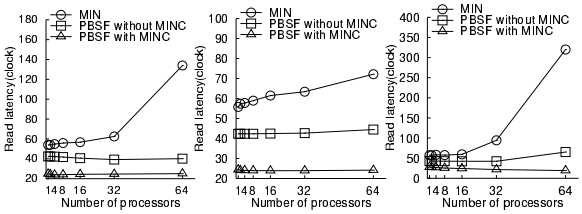


Figure 17: 実行時間の比較



a) Radix b) FFT c) LU

Figure 18: リードレイテンシの比較

このように、SSS 型 MIN では高速動作可能なスイッチにより低レイテンシなネットワークを構成しつつ、多重出力可能な PBSF トポロジを用いることによってネットワーク混雑時の転送性能を維持することによって、16PU、64PU のどちらの規模においても 通常の MIN よりも優れた性能を示すデータ転送用ネットワークを構成できることがわかる。

8 まとめ

SNAIL–2 の命令レベルシミュレータを用いて多重出力が可能な MIN および MINC を用いたキャッシュ機構の評価を行なった、

結果、PBSF トポロジの SSS 型 MIN はネットワークの混雑時でも高い転送能力を誇り、レイテンシを抑えることによってネットワークへの負荷が少ない場合でも通常の MIN と比較しても劣らないことを示した。

また MINC を用いたキャッシュ機構はネットワークの負荷を効果的に軽減し、パフォーマンスの向上に大きく貢献することを示した。

References

[1] 天野 英晴, 周 洛, 藤川 義文. “SSS(Simple Serial Synchronized) 型マルチステージネットワーク”. 情報処理学会論文誌 第 34 巻 第 5 号, pp.1134-1143, 1993.

[2] 笹原 正司, 寺田 純, 大和 純一, 塙 敏博, 天野 英晴, “SSS 型 MIN に基づくマルチプロセッサ SNAIL”. 情報処理学会論文誌 第 36 巻 第 7 号, pp.1640-1651, 1995.

[3] 塙 敏博, 天野 英晴. “多重出力可能な MIN の性能評価”. 情報処理学会論文誌 第 36 巻 第 7 号, pp.1630–1639, 1995.

[4] H.Matsumoto, T.Hiraki. “The shared memory architecture on the massively parallel processor”. Technical report of IEICE, CPSY 92-36, pp.47–55, 1992.

[5] 西村 克信, 工藤 知宏, 天野 英晴. “Pruning Cache を用いた分散共有メモリのディレクトリ構成法”. 情報処理学会論文誌 第 39 巻 第 6 号, pp.1644–1654, 1998.

[6] T.Hanawa, T.Kamei, H.Yasukawa, K.Nishimura, H.Amano. “MINC: Multistage Interconnection Network with Cache control mechanism”. IEICE Transactions on Information and Systems, Vol.E80-D, No.9, pp.863–870, 1997.

[7] 星野 智則, 緑川 隆, 天野 英晴 “キャッシュ制御機構を持つスイッチ結合型マルチプロセッサ SNAIL-2 の実装”. 電子情報通信学会コンピュータシステム研究会, CPSY99-70, pp.63–70, 1999.

[8] 白石 大介, 星野 智則, 緑川 隆, 金森 勇壮, 天野英晴 “スイッチ結合型マルチプロセッサ SNAIL–2 のデータ転送用ネットワーク PBSF の評価”. 電子情報通信学会 VLSI 設計技術研究会, 2001.

[9] 坂元, 荒井, 正木, 井上, 天野, “自己ルーティングスイッチの構成とその評価,” 信学技報 ISSE88-30 8,1988.

[10] F.A.Tobagi, T.Kwok “The Tandem Banyan Switching Fabric: a simple High-Performance Fast Packet Switch”, Proc. of INFOCOM91, 1991

[11] 天野 英晴, 藤川 義文 “マルチステージネットワーク PBSF”. 情報処理学会計算機アーキテクチャ研究報告 No.94-5, 1992.

[12] 若林 正樹, 天野 英晴, “並列計算機シミュレータの構築支援環境”. 電子情報通信学会論文誌, 2001.