並列ベンチマークによるSNAILの評価

藤原　崇　　鬼頭　宏幸　　山口　喜弘　　塙 敏博　　亀井 貴之　　米田 卓司
山本　淳二　　天野 英晴

慶應義塾大学 理工学部

SSS(Simple Serial Synchronized)型MIN(Multistage Interconnection Network)は単純なエレメントからなる通過率の高いMINに対して同期式にパケットを入力して交換することにより、高い性能価格比を実現するプロセッサ-メモリ間結合網である。我々は、SSS型MIN用に高い通過率を持つTBSF(Tandem Banyan Switching Fabrics)チップを開発し、マルチプロセッサプロトタイプSNAILを実装した。本論文では、共有メモリ型並列計算機の開発、評価用のベンチマークプログラム集であるSPLASHのうちいくつかを利用して、SNAILおよびネットワークの評価を行った。アプリケーションによってはMINの混雑はかなり激しくなり、パケットの衝突率は16PUにおいて45%に達したが、16PUにより約7倍から12倍の台数効果が得られた。

和文キーワード　　並列計算機、SPLASH、SNAIL、MIN

The Evaluation of SNAIL with Parallel Benchmark Programs

T. Fujiwara　　Y. Kitoh　　Y. Yamaguchi　　T. Hanawa　　T. Kamei　　T. Komeda
J. Yamamoto　　H. Amano

Keio University

Simple Serial Synchronized (SSS) Multistage Interconncetion Network(MIN) is a high performance but low cost processor-memory network. Synchronized bit-serial communication simplifies the structure/control of a switching element, and also solves the pin-limitation problem. An LSI chip for the SSS-MIN based on a high bandwidth connection topology called the TBSF(Tandem Banyan Switching Fabrics) were developed and now a multiprocessor prototype SNAIL using these chips is available. In this paper, we evalutate SNAIL and its network using some programs from SPLASH, which is the collection of benchmark programs for multiprocessors with shared memory. Although severe network congestion which causes 45% conflict ratio at maximum is observed, 7-12 times performance improvement is achieved with 16 processors.

英文 key words　　multiprocessor,SPLASH,SNAIL,MIN

1 はじめに

Multistage Interconnection Network(MIN) は、接続可能な最大プロセッサ数の制限の厳しいバス結合型と比べて複数のメモリモジュールからのデータの同時転送が可能であることから、数十から数百プロセッサ規模のマルチプロセッサにおけるプロセッサ・メモリ間結合網として注目されている [1][2]。

しかし、従来のマルチプロセッサ用に提案された MIN は、スイッチングエレメントの多機能化やピンネック等により、実装に困難な点が多く、MIN 結合型並列計算機のコストを引き上げ、開発を遅らせる原因になっていた。

そこで我々は、パケットをシリアル化、同期入力化する事によってスイッチングエレメントを単純化し、ハードウェアコストを減少させかつ高速なネットワークを構成できる SSS アーキテクチャを提唱し [3][4]、ネットワークチップの実装を行って来た [5]。また、この SSS アーキテクチャの動作上の特徴をいかすことにより、簡単にメッセージコンバインの実装が可能である。

我々は、SSS 型 MIN のプロトタイプチップを実装し、これを用いたマルチプロッセッサプロトタイプ S-NAIL(SSS Network Architecture ImpLementation)[6][7] を開発した。SNAIL における SSS 型 MIN の性能は n-Queen 等の簡単なプログラムにより評価されてきたが [7]、大きなサイズの実際的なアプリケーションプログラムにおける評価は行なわれていなかった。本報告ではこの SNAIL 上に、並列ベンチマークプログラム集 S-PLASH のいくつかを実装することにより、SSS 型 MIN の実際的な評価を行った。

2 SSS 型 MIN

2.1 SSS 型 MIN の基本構造

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 との間のバッファで 1-4bit 程度にシリアル化され、B-ISDN 等で用いられている ATM 用交換システム同様、フレームクロックと呼ばれる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、フレームクロックから生成されるタイミングで入力されたパケットのタグ情報を参照し、それに従ってスイッチを決定する機能しか持たない。さらにスイッチングエレメント内にはパケットバッファは設けられていないため、各エレメントは 1bit 分の記憶しか行わない。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交換機能を持ったシフトレジスタのように動作する。そして全ステージを通過する分の遅延の後、パケットの先頭が出口から出力される。

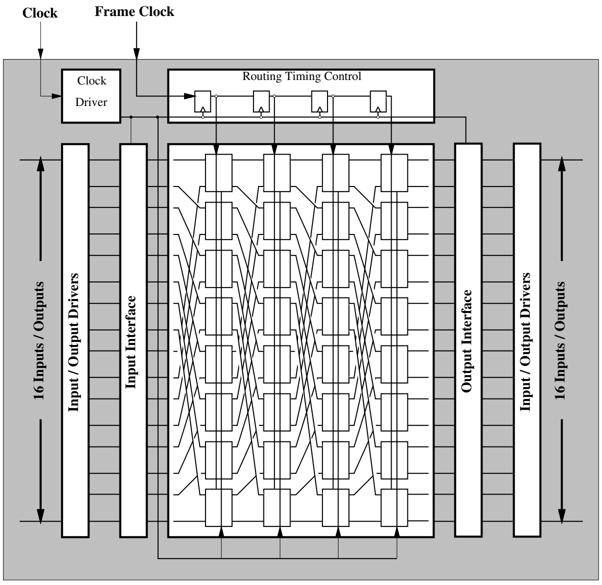

図 1: SSS 型 MIN の構造

SSS 型 MIN では、パケットバッファを持たない為、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い方向に転送されるが、他方のパケットは正しく転送されない。この様な場合、パケットに用意されたコンフリクトビットがセットされ、以後デッドパケットとして扱われる。

ネットワーク内ではフレームクロックの立上りの数クロック後、1 クロック毎に一段ずつスイッチの状態が決定される。そして段数分のクロックの遅延の後にネットワーク全スイッチの状態が決定し、ネットワークの入力から出力への論理的なパス (以下トレースと呼ぶ) が設定され、パケットの先頭が出口より出力される。これと同時に、出力側で有効なパケットが到着したかどうかを調べ、ACK(ACKnowledge) もしくは NAK(NotAcKnowledge) をプロセッサ側に返す。プロセッサ側が NAK を受け取った場合、次のフレームでパケットを再送しなくてはならない。

2.2 高速化手法

SSS 型 MIN は基本的にシリアル転送を行うため、ビットパラレル転送の場合と比較し、転送時間の点で不利である。この点を克服するため、シリアル転送 ·フレーム同期転送の長所をいかした以下のような高速化がなされている。なお、これらの手法はすでに実装が行われた SSS 型 TBSF チップで実装されている。

2.2.1 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、アクノリッジおよびデータパケット転送のトレースも決定される。図 2に示す通り、これらの転送はオーバーラップして行われる。図中、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。正しく転送が行われ、ACK が返送された場合、データの転送はこのトレースを利用してフレーム $i+1$ でのアドレス転送にオーバラップして行われる。

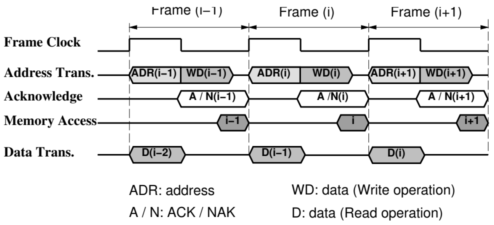

図 2: アドレス ·データ ·アクノリッジの転送タイミング

2.2.2 ステージホッピング

アドレスパケットのルーティングタグにより一度トレースが確定されてしまえば、アドレスパケットの残りとメモリに対する書き込み時のデータは既に設定されたトレースをたどって行けばよい。 従って、ルーティングタグ以降のアドレスパケット、データパケット、アクノリッジ信号は各ステージでレジスタに格納されることなく、これらを飛び越してチップ内のマルチプレクサおよび配線による遅延のみで出力に送られる。

2.2.3 メッセージコンバイン機能

メッセージコンバイン機能は同一宛先のパケットをエレメント内で結合し、1 つのパケットとして扱う事により、Hot Spot へのアクセス集中による Tree Saturation を回避する方法である。この機能の効果はシミュレーション等で確認されているが、従来型の MIN でこの機能を実現するためには 6 倍から 32 倍のハードウェアを要するため [8]、実現が困難であった。

しかし、SSS 型 MIN ではシリアル転送の特徴を生かして簡単に実現できる。SSS 型 MIN のスイッチングエレメントはパケットバッファを持たないため、コンバインができるかどうかは入力されたパケット 2 つのみを比較すればよい。さらに、パケットはシリアル化されて入力されるため、比較器は数 bit 幅のものでよい。

3 SNAIL の構成

プロセッサとローカルメモリからなる 16 個の PU は SSS 型 MIN を介して共有メモリモジュール 16 個と接続されている。(図 3) ただ 1 つのプロセッサのみ Ethernet インターフェースを持ち、ホストのワークステーションと結合されている。又、パフォーマンスモニタのシステムは、SNAIL と独立に動作し、MIN の両端を監視している。全てのシステムは、単一システムクロック (50MHz) と、フレームクロック (50/16=3.125MHz) を共有し、同期動作する。

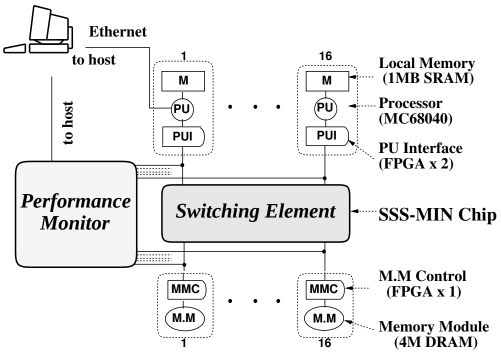

図 3: SNAIL の構成

SNAIL は絶対性能よりも柔軟性を重視した設計になっている。PU-MIN インターフェースとメモリコントローラは Xilinx 社の FPGA XC3090 を用いて実装している。FPGA の動作速度は速くはないが、コンフィグレーショ

ンデータを書き換える事により、ハードウェアを変更する事ができる。

ここで、プロセッサ部、メモリモジュール、パフォーマンスモニタ部について簡単述べる。

- プロセッサ部

  プロセッサ部の仕様を表1に示す。

  表 1: プロセッサ部の仕様

Processer	MC68040(25MHz)
Local Memory	1Mbyte/each
Processer Controller	FPGA:XC3090
Packet Decoder	FPGA:XC3090

  プロセッサ部は4プロセッサが1枚のボードに実装されている。全てのプロセッサうち、1つがマスタプロセッサとなり、マスタプロセッサはEthernetを介してホストのワークステーションと接続されている。マスタプロセッサにはモニタプログラムが搭載されており、ワークステーションからプログラムをロードすることができる。

- メモリモジュール

  メモリモジュールの仕様を表2に示す。

  表 2: メモリモジュールの仕様

16 Shared Memory Modules	4MByte/each (64MByte/all)
Memory Controller	FPGA:XC3090

  メモリコントローラは、MINとメモリ間のパラレル/シリアル変換と、ブロック転送、メモリリフレッシュ等の制御をしている。

  MIN内での衝突がない場合の共有メモリへのアクセス時間を表3に示す。ローカルメモリは80nsecでアクセスが可能なので共有メモリへのアクセスはローカルメモリの約4倍かかる。

  共有メモリ16モジュールとネットワークは合わせて1つのボードに実装されている。このボードに4プロセッサを含むボード4枚を差すことにより、16プロセッサのSNAILが構成される。

表 3: メモリアクセス時間

Frame period	320nsec
32 bit data read/synchronization	360nsec
1 line block transfer	800nsec

- パフォーマンスモニタ部

  一般的に、MINを用いたマルチプロセッサは、共有バス型マルチプロセッサに比べると、メモリモジュールが多数あるので、交信の全体を把握するのが困難である。しかし、SSS型MINは基本的にビットシリアル通信で、しかも全パケットが同期して入出力されるため、MINを用いた他のシステムに比べ、システムモニタの実装が容易である。このシステムは、簡単なCPU(MC68000)とRS232Cを装備する事により、SNAIL本体とは独立にホストと交信する事ができる。MINの入口と出口からパケットをモニタする事により、総パケット数、衝突による転送失敗パケット数等の情報を得る事ができる。

3.1 SSS-TBSFチップのSNAILへの実装

SSS-TBSFチップ1つで、16プロセッサ、16メモリモジュールを接続する事ができる。SSS型MIN 1チップではアドレス3ビット、データ2ビットパラレルの転送路が確保できる。しかし、プロセッサのパフォーマンスと比較すると、ネットワーク通過に伴うレイテンシとメモリアクセスにかかる時間を隠蔽できないので充分ではない。そこで、SNAILではビットスライスにチップを4つ用い、アドレス幅12ビット、データ幅8ビットにして、レイテンシを改善している。チップ4つを用いる事により、1フレームの長さがプロセッサクロックで16クロックとなる。

3.2 ハードウェア同期機構

Fetch&DecやTest&Setによる同期は、SSS型MINのスイッチの優先度が固定のためにスタベーションが起こる可能性あり、優先度の強さに応じて同期変数のポーリング間隔を変えている。そのため、最適な同期機構とはいえない。そこで、ハードウェアによる同期機構をサポートした。

4　評価用アプリケーション

本研究での実装、評価の対象となるアプリケーションとして共有メモリ型マルチプロセッサシステムのベンチマークプログラム集である SPLASH のうち MP3D、Water、Cholesky、Barnes-Hut を選んだ。

4.1　SPLASH とは

並列アプリケーションの実行により並列計算機の評価を行なう場合、アプリケーションの選択は重要である。いわゆる"toy プログラム" では実際的な性能を評価することができず、ともすれば誤った評価が得られる可能性がある。そこでよりリアルなアプリケーションを提供するためにできたのが SPLASH(Stanford ParalleL Applications for SHared-memory (SPLASH)[9] である。

4.2　プログラムの内容

以下に、今回評価に用いた MP3D、Water、Cholesky、Barnes-Hut のプログラムの内容を述べる。

4.2.1　MP3D

モンテカルロ法により希薄な流体の流れをシミュレーションするプログラムである。直方体のトンネルのなかに障害物を配置し、そのトンネルを超音速で粒子が通過する際の粒子の動きをシミュレートする。

主要な共有データは直方体の空間を分割した小空間の 3 次元配列である。各タイムステップについて、5 種類の処理が順番に行なわれ、各処理の終了時にバリア同期がとられる。

4.2.2　Water

このプログラムでは n 個の水分子の液体状態での分子内および分子間の相互作用と水分子のポテンシャルを計算する。

計算はユーザが指定したタイムステップまで行なわれる。立方体中の水分子の動きのニュートン方程式を Gear の sixth-order predictor-corrector 法を用いて解く。

4.2.3　Cholesky

このプログラムは、supernodal elimination 法 [10] を用いて疎行列の Cholesky 分解を行なう。Cholesky 分解の ordering, symbolic factorization, numerical factorization の 3 ステップのうち、numerical factorization を並列化して測定対象としている。共有メモリに配置される主なデータ構造は疎行列そのものである。データ量を圧縮するため、疎行列は非ゼロ要素のみを持つ配列と、その行列中での位置を表す情報を格納した配列によって表現されている。supernode と呼ばれる複数列の塊を単位として行列の要素の変更を行なう。プロセッサは共有メモリ上のタスクキューからタスクを取り出して実行する。同一の列の要素の変更とタスクキューからのタスクの取り出し時にはロックが用いられる。

4.2.4　Barnes-Hut

このプログラムは、Barnes-Hut というアルゴリズムを用いて、重力場における body からなるシステムの発展をシミュレーションしている。Barnes-Hut アルゴリズムは、3 次元空間の階層 8 分木表現をもとにしている。ここでは、全ての body が点としてモデル化され、システム内において、全ての body は他の body と相互作用が働いている。

プログラムは、ユーザが指定したタイムステップだけ実行され、それぞれのステップでは全ての body にかかる力を計算している。実行時間の大部分は、この部分の計算に費される。

5　アプリケーションによる性能評価

5.1　評価条件

今回の評価は 16 プロセッサ、16 メモリモジュールで行った。

プロセッサ間の同期は、ハードウェアによる同期機構を用いている。各アプリケーションは、初期データを各プロセッサのローカルメモリ、共有メモリに配置した後、並列に実行する。実行時間はこの並列実行部分のみを測定する。また、パフォーマンスモニタによる測定も同様に並列実行部分のみを測定した。

5.1.1　MP3D

入力ファイルとして、MP3D に添付されいる test.geom をジオメトリファイルとした。これは、$x = 14, y = 24, z = 7$ のセルからなる空間で粒子数 500 個、50 ステップで実行した。

5.1.2 Water

入力ファイルとして、Water に添付されている LWT12 を用いた。詳しくは以下の通りである。

```
TEMPERATURE                =   298.00 K
DENSITY                    =  0.99800 G/C.C.
NUMBER OF MOLECULES        =       64
TIME STEP                  = 1.50e-16 SEC
ORDER USED TO SOLVE F=MA   =        6
NO. OF TIME STEPS          =        2
FREQUENCY OF DATA SAVING   =       -1
FREQUENCY TO WRITE RST FILE=     3000
SPHERICAL CUTOFF RADIUS    =   6.2128 ANGSTROM
```

5.1.3 Cholesky

入力ファイルとして Cholesky に添付されている lsh-p.O ファイルを用いた。これは、3466 の正方行列で非ゼロ要素の数は 10215、スーパーノードの数は 1846 という行列である。

5.1.4 Barnes-Hut

入力したパラメータは以下の通りである。

```
nbody(number of particle)          = 1024
seed(random number generator seed) = 123
dtime(integration time-step)       = 0.025
eps(usual potential softening)     = 0.05
tol(cell subdivision tolerence)    = 1.0
fcells(cell allocation parameter)  = 2.0
tstop(time to stop integration)    = 0.075
dtout(data output interval)        = 0.25
```

またパーティションには、コストゾーンを用いた。

5.2 評価項目

SNAIL に接続されるパフォーマンスモニタにより以下のものが計測可能である。

- 実行時間

  実行時間を計測する事により、プロセッサ数の増加による台数効果を評価する事ができる。

- ネットワーク通過パケット数 (OUT)

  ネットワークを通過する総パケットの数を表す。これは、衝突してデッドパケットとなったパケットも含んでいる。これにより、ネットワークの混雑度を評価する事ができる。

- 衝突パケット数 (CF)

  衝突パケットを計測する事で、入力されたパケットのうち、どれくらいの割合でパケットが衝突しているかが分かり、パフォーマンスへの影響を評価する事ができる。

- プロセッサの共有メモリアクセス回数 (IN)

  全プロセッサが発行した共有メモリへのアクセス回数を表す。本来なら、この値はネットワーク通過パケット数から衝突パケット数を引いたものと同じであるが、コンバインされたパケット数を計測するには、入力されたパケット数を数えなければならない。ここでコンバインされたパケット数の計算方法を示す。

  $$CB = IN + CF - OUT$$

  コンバインされたパケット数を計算する事により、全パケットのうち、どのくらいのパケットがコンバインしたのか、衝突したパケットのうち、どのくらいのパケットがコンバインしたのかを評価する事ができる。

5.3 SSS 型 MIN と SNAIL の評価結果

5.3.1 台数効果

各アプリケーションの、プロセッサ数の増加による台数効果を図 4に示す。この図は、1 プロセッサで並列にプログラムを実行した場合の実行時間を基準にしている。MP3D、Water、Barnes-Hut はほぼ直線的に性能が向上していることがわかる。プロセッサ数の増加に伴って、理想直線から外れていくのは、同期待ち時間の占める割合が増加するためである。特に Cholesky は他の 3 つのアプリケーションに比べると直線的な台数効果が得られていない。これは、それらのアプリケーションに比べ同期待ち時間、特にロックによる待ち時間が長くなるためである。また、SNAIL は 2 次キャシュを実装していないので、プロセッサ数が増えるとネットワークが混雑することもあげられる。(図 5参照)

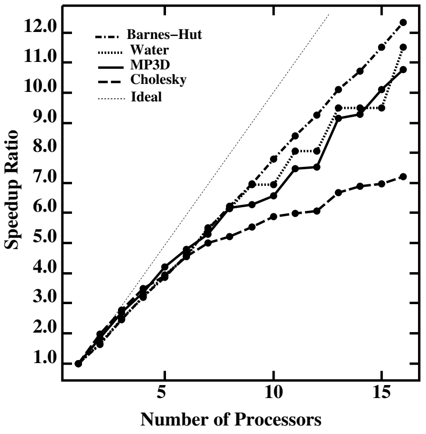

図 4: 台数効果

5.3.2　ネットワーク利用率

ネットワークの負荷がどのくらいであるかを確認するため、各アプリケーションにおけるネットワーク利用率 (フレームがパケットにより使用される割合:すなわちMIN に対する負荷の割合) をパフォーマンスモニタにより測定した。

図5より MP3D、Water、Barnes-Hut のネットワーク利用率はそれぞれ、25%、12%、8%でありプロセッサ数が増えてもそれほど変化していない。しかし、Cholesky についてはプロセッサ数が増えるに従い、ネットワークの利用率も高くなっている事がわかる。特にプロセッサ数が 13 台以上になると利用率は著しく高くなり、台数効果に影響が出てくる。ネットワークの利用率が高くなるのは、衝突したパケットが再送されるためと考えられる。

5.3.3　衝突率

衝突率とは、測定時にネットワークを通過した全パケットに対する衝突したパケット数の割合を表す (図 6)。全てのアプリケーションで、プロセッサの増加にともない衝突率も高くなっている事が分かる。また、ネットワークの利用率が高いもの程、衝突率も高くなっている。パケットの衝突率が高くなるとパケットの再送が多くなるため、性能向上の低下につながる。

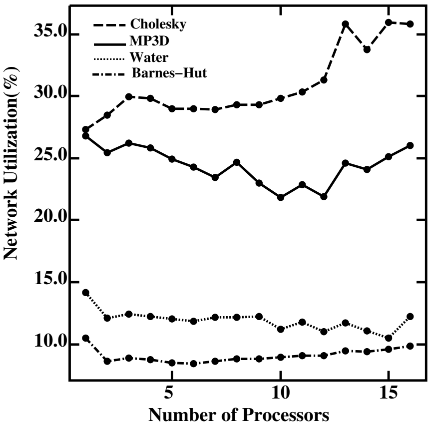

図 5: ネットワーク利用率

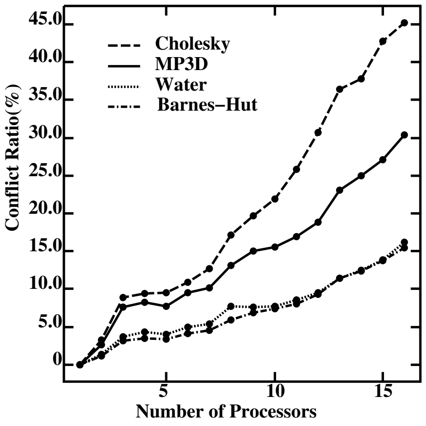

図 6: 衝突率

5.4　メッセージコンバインの効果

メッセージコンバインの行なわれたパケット数の全パケットに占める割合をコンバイン率と呼ぶ。今回実装したアプリケーションにおいて、コンバイン率を測定した結果を図 7に示す。コンバイン率は、MP3D, Water においては 16PU でも 3%程度に留まっているが、Choleskey においては 9%を超えている。しかし、コンバイン機能を用いた場合と用いない場合の実行時間は、ほとんど変化が見られなかった。これは、Choleskey におけるコンバインがほとんど同期操作の busy waiting 時を行なわれているためであると考えられる。この場合、コンバインにより結合網の混雑が低減されても、全体の実行時間は負荷が大きく計算の遅いプロセッサによって律速され

るため、実行時間は低減されない。

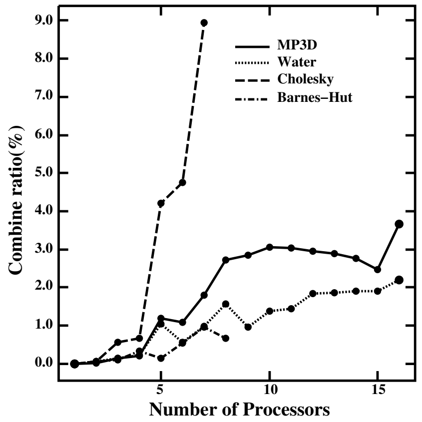

図 7: コンバイン率

現在の SNAIL の性能と MIN の性能を考えるとプログラムの最適化をぎりぎりまで行なわなければコンバインの効果は現れないであろう。より高速な CPU を用いた場合や、プロセッサ数が更に多い場合には効果が期待できる。

6 結論

並列アプリケーション集 SPLASH 中の MP3D, Water, Choleskey, Barnes-Hut により、SNAIL の結合網の評価を行なった。

結合網に対する負荷はアプリケーションにより異なり、最も頻繁な交信を要する Cholesky では結合網の混雑はかなり激しくなり、衝突率は最大 45%、ネットワーク利用率は最大 35% 程度になった。このことは、プロセッサ数が大きい範囲では性能向上に影響を及ぼしたが、それでも 16PU で 7 倍程度の性能向上を達成することができた。プロセッサ間交信の少ない Water や Barnes-Hut では良好な性能向上が観測された。一方で、メッセージコンバインは、3%から 10%程度のコンバイン率があるものの、実際の速度向上は観測できなかった。

評価を通じて、SNAIL に用いた SSS 型 MIN は、アプリケーションによっては若干混雑が激しくなる事があっても、プロセッサの性能、規模とよくバランスがとれていることがわかる。これらの評価をもとに、最近の高性能マイクロプロセッサを PU に用いた場合に必要とされる SSS 型 MIN の性能について解析し、後継機を開発していく予定である。

参考文献

[1] J.Konicek, et al,: *The Organization of the Cedar System*, Proc. of 1991 Int. Conf. Parallel Processing, (1991).

[2] G.F.Phister, et al.: *The IBM Research Parallel Processor Prototype (RP3): Introduction and architecture*, Proc. of 1985 Int. Conf. Parallel Processing, (1985).

[3] Amano, H., Zhou, L. and Gaye, K.:SSS(Simple Serial Synchronized)-MIN:A Novel Multi Stage Interconnection Architecture for Multiprocessors, *Proc. of the IFIP 12th World Computer Congress*, Vol.I, pp.571-577 (Sept. 1992).

[4] 天野, 周, 藤川: 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 Vol.34, No.5, pp. 1134-1143, (1993).

[5] 周洛, 天野英晴: SSS-MIN 用プロトタイプチップの実装, 情報処理学会第 44 回全国大会論文集,6-103,104 (1992).

[6] M.Sasahara, et al: *SNAIL:a multiprocessor based on the Simple Serial Synchronized multistage interconnection network architecture*, Proc. of International Conference on Parallel Processing, pp.I117-120, Aug. 1994

[7] 笹原, 寺田, 大和, 塙, 天野: SSS-MIN アーキテクチャに基づくマルチプロセッサプロトタイプ SNAIL, 情報処理学会論文誌, 第 36 巻 第 7 号 (1995).

[8] G.F.Phister and V.A.Norton : *Hot Spot Contention and Combining in Multistage Interconnection Networks*, IEEE Trans. on Comput. vol. c-34, No.10, (1985).

[9] Jaswinder Pal Singh, Wolf-Dietrich Weber and Anoop Gupta,“SPLASH:Stanford Parallel Apprications for Shared-Mamory”,Computer System Laboratory Stanford University, CA 94305

[10] E.Rothberg and A.Gupta,“Techniques for improving the performancs of sparse factorization on multiprocessor workstations,”Proceedings of Supercomputing ’90,November,1990.