

# メッセージフローに基づくネットワークシミュレータ MFS の評価

矢崎 俊志<sup>†</sup> 石畑 宏明<sup>††</sup>

本論文では、筆者らが通信アルゴリズムの評価を目的として提案したフローベースシミュレータ Message Flow Simulator (MFS) のより汎用的な利用可能性を示すため、既存のパケットベースシミュレータ Booksim を用いて様々なネットワークトポロジと通信パターンで MFS を比較評価した結果を述べる。筆者らはこれまで、MFS がパケットベースシミュレータ BigSimulator より短時間で全対全通信アルゴリズムを評価可能であることを示した。また、メッセージが相互接続網を通過する時間のみを評価可能な Booksim を用いて、Fattree ネットワーク上のランダム通信シミュレーションによる比較評価を行ってきた。本稿では新たに MFS と Booksim のシミュレーション結果の差がスイッチで行われるアービトレーションの影響により生じることを示した。このことから、通信の平均ホップ数が少ないトポロジのネットワークの評価や、近距離のノードと通信を頻繁に行う並列プログラムの通信シミュレーションに MFS が利用できる可能性を示した。1 万ノード以上の大規模なネットワークについて、全ノードが、10 パケットをランダムな宛先に送るシミュレーションを実行した。この時 MFS は Booksim の 1%~2% の実行時間とメモリ使用量でシミュレーションを実行した。

## An Evaluation of Message-flow-based Network Simulator

SYUNJI YAZAKI<sup>†</sup> and HIROAKI ISHIHATA<sup>††</sup>

This paper showed evaluation results of Message Flow Simulator (MFS), which is a flow-based network simulator for large-scale parallel computer. The evaluation performed based on comparison of simulation result on various topology and communication pattern to show capabilities of application. MFS performed the simulation for evaluation of all-to-all communication algorithms faster than BigSimulator which is a packet-based network simulator embedded in a parallel computer simulator. We have evaluated MFS by using Booksim which is a packet-based network simulator and it evaluates communication time. In this paper, we showed that MFS gives different result with Booksim due to effect of arbitration in the router. The result showed that MFS can be used in case of evaluation of networks which achieve low average hop count, and parallel programs which generate communication with low average hop counts. MFS performs simulation with less run-time and memory usage when the number of nodes was over 10,000. Run-time and memory usage of MFS were from 1% to 2% by those of Booksim.

### 1. はじめに

#### 1.1 背景

多数の計算ノードを相互接続網で接続した超並列計算機上で効率の良い並列計算を実現するためには、ノード間通信の効率化が重要な課題である。これは、並列計算における通信時間の増大が並列化効率を低下させる要因となるためである。

ノード間で高い通信効率を実現する相互接続網の実装コストは、一般に、接続するノード数および、最大

通信性能を決定づけるバイセクションバンド幅に依存して増大する。実際の並列計算機においては、通信効率とコストのトレードオフにより、様々な構成の相互接続網が用いられている。比較的小規模の並列計算機においては、クロスバで相互接続網を構成することができた<sup>1)</sup>。数千ノードの接続には、よりコストの低い Fattree トポロジの相互接続網を採用している例がある。数十万ノード規模の並列計算機では、さらに、Mesh, Torus, バンド幅の小さい Fattree, またはこれらを組み合わせた不均一なトポロジを選択せざるを得ない<sup>2)~4)</sup>。このようなトポロジで構成された相互接続網は、バイセクションバンド幅が狭く、通信経路の競合が起きやすい。そのため、通信経路の偏りによってホットスポットも発生しやすく、通信路をまんべん

<sup>†</sup> 電気通信大学  
The University of Electro-Communications  
<sup>††</sup> 東京工科大学  
Tokyo University of Technology

なく効率的に利用することが難しい。

並列プログラムの開発者は限られた通信路を効率的に使うため、送受信のタイミングや順番を工夫した様々な通信アルゴリズムを用いる。通信競合が起きやすいトポロジを持つ近年の大規模並列計算機向けには、特定のトポロジや通信パターンに合わせた最適な通信アルゴリズムが考案されている。これらの通信アルゴリズムの多くは複雑であり、その評価を解析的に行うことが難しい。

従来、大規模ネットワークを対象とした通信アルゴリズムの評価は、実行に多くの時間を必要とする通信シミュレーションの結果に基づいて行われてきた。並列計算機の通信シミュレーションに使われるネットワークシミュレータは、本来、相互接続網の評価や通信時間の精密な予測を目的として開発されたものが多い。そのため、通信をパケットやフリット単位で詳細にモデル化したパケットベースシミュレータを用いるのが主流である。パケットベースシミュレータの実行には通信を行うノード数や通信されるパケット数などに比例した時間がかかる。

より効率良く通信アルゴリズムを評価することを目指して、筆者らは Message Flow Simulator (MFS) を開発した<sup>5),6)</sup>。MFS は、並列プログラムの開発者が考える抽象度の高い通信モデルを実装したものである。MFS は、通信をパケットやフリットのように粒の動きとしてとらえるのではなく、流体の流れ（フロー）として抽象化し、その競合度を算出することで通信アルゴリズムを評価するフローベースシミュレータである。MFS の大きな特徴は、より大規模な通信シミュレーションに対応した高い拡張性と、様々な通信パタンのシミュレーションを短時間で実行できる高速性である。一方で、実機に近いという意味での精度はパケットベースシミュレータと比較すると低い。

筆者らは、文献<sup>5)</sup>で、MFS が既存のパケットベースシミュレータ BigSimulator<sup>7),8)</sup> より短時間で通信アルゴリズムを評価することが可能であることを示した。この結果は、2次元 Torus トポロジを持つ相互接続網における全対全通信アルゴリズムを対象としたものであり、その他のトポロジや通信パターンに対する評価は行われていなかった。また、BigSimulator は、MFS と異なり、メッセージの流れだけでなくノードで行われる通信の初期化や終了に関わる処理もシミュレーション結果に含める。このようなノードでの処理時間を除外するため、BigSimulator による測定においては、通信されるメッセージのサイズを大きくした場合と小さくした場合の差を通信時間として比較を

行った。しかしこの方法では、ノードでの処理のうち、メッセージサイズに比例して大きくなる処理時間の影響を完全に排除することが難しかった。

筆者らは文献<sup>6)</sup>において、メッセージが相互接続網を流れる時間のみを評価することが可能なパケットベースシミュレータ Booksim を用いて、Fattree トポロジで構成される相互接続網上のランダム通信におけるシミュレーション結果の比較を行った。この比較では、Booksim のシミュレーションにおいてパケットを構成するフリット数を大きくすることで、結果が MFS に近くなることを示した。これは、1パケットあたりのフリット数の増加が、より流体モデルに近い粒度の細かな通信を行った結果に近くなるためであると考えられる。

## 1.2 目的

本論文では、通信アルゴリズムの評価だけでなく、相互接続網の評価など、MFS のより汎用的な利用法における有用性を示すことを目的として、既存のパケットベースシミュレータと MFS を比較評価し、MFS の利用可能性を議論する。本稿では、メッセージが相互接続網を流れる時間のみを評価対象とするため、文献<sup>5)</sup>で用いた BigSimulator ではなく、文献<sup>6)</sup>で用いた Booksim を比較に用いる。また、文献<sup>6)</sup>では行われていなかった Mesh や Torus トポロジで構成される相互接続網やランダム以外の通信パターンも評価に用いる。

本論文では、第2節で関連研究を引用し、並列計算機向けネットワークシミュレータについて述べる。第3節では Booksim による統計的な相互接続網評価手法を元に、MFS と Booksim を比較評価する。第4節では、MFS と Booksim による大規模ネットワークのシミュレーション結果を比較する。最後に第5節でまとめる。

## 2. 並列計算機向け通信シミュレータ

本節では、まず、並列計算機を対象とした通信シミュレータについて、パケットベースおよびフローベース方式についてシミュレーションモデルを説明する。続いて、3節以降の評価に用いるシミュレータについて述べる。

### 2.1 シミュレーションモデル

パケットベースおよびフローベースのシミュレーションのモデルを図1に示す。パケットベースシミュレーションは、図1の左側に示すように、通信を構成するパケットやフリットを粒としてとらえ、この単位でシミュレーションを行う。このモデルは、実際に行われ

る通信をハードウェア側からの視点でモデル化したものであると言える。これらの多くは、並列計算機の相互接続網上で発生する輻輳とその影響をハードウェアに近いレベルで精密に調べるために用いられる場合が多い<sup>7)~13)</sup>。また、並列計算機上で実行される並列プログラムの実行時間予測に用いられた例もある<sup>14),15)</sup>。

フローベースシミュレーションは、図1の右側に示すように、通信を連続体の流れ(フロー)として扱い、その流量が通信経路の競合により制限されるというモデルに基づくシミュレーションを行う。このモデルは、通信をソフトウェア側からの視点で抽象化したものであると言える。この方式は、パケットやフリットの振る舞いを個々に再現しないため、シミュレーションを高速に実行することができる。フローベース方式は通信経路競合の度合いを高速に評価することができるため、通信アルゴリズムの評価などに適している。

## 2.2 Booksim

パケットベースシミュレータの実装として Booksim がある。Booksim は、Stanford 大学の Dally らによって開発されたシミュレータである<sup>16)</sup>。このシミュレータでは、通信はフリット単位で行われる。Booksim は、サイクル単位でシミュレーションが進行するサイクルベースシミュレーションである。通信を詳細に表現するため、通信に関わるスイッチ(ルータ)のバッファや Virtual Channel (VC), アービトレーション機構などもモデル化されている。シミュレーションを実装するには、相互接続網のトポロジ、サイズ、スイッチのバッファサイズ、VC の数、アービトレーションアルゴリズム、1パケットを構成するフリット数、通信パターンなどを指定することができる。

Booksim は、フリットが相互接続網を通過する平均的な割合や、平均的なレイテンシを統計的に見積もることができる。これにより、相互接続網の通信性能を評価する。フリットの平均通過率(Average accepted rate)は相互接続網に投入されたフリットのうち、平

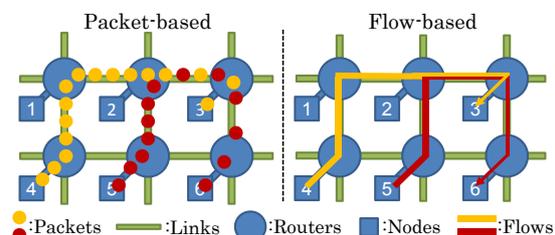


図1 パケットベースおよびフローベースのシミュレーションモデル。

Fig. 1 Simulation models of Packet-based and Flow-based methods.

均的に何割のフリットが相互接続網を通過できたかを表す値であり、これは、最大スループットに対する実効スループットの割合(Average Throughput Rate, ATR)に相当する。

Booksim では、相互接続網に投入されるフリットの多さを Injection Rate として設定することができる。相互接続網の最大通信能力を評価するためには、相互接続網がフリットで飽和した状態で評価を行う必要がある。ただし、通信がデッドロックしている状態は除外する。よって、Booksim による相互接続網の統計的手法による通信性能評価において、シミュレーションの開始直後および終了直前の通信は測定から除外される。開始直後および終了直前は、相互接続網内にフリットがあまり存在しない。この状態での測定はスループットやレイテンシを過度に高く見積もる恐れがある。

## 2.3 Message Flow Simulator (MFS)

MFS は、並列計算機上で行われる通信の最適化を目指し、通信アルゴリズムを評価する目的で開発された。MFS はフローの重なりによってその流量が制限されるモデルに基づいて、通信に必要な時間を計算する。計算の手順を次に示す。より詳細なアルゴリズムは文献<sup>5)</sup>で述べられている。

- (1) 与えられたトポロジと通信パターンで、全ノードペア間のフローと、その経路のリストを作る
- (2) 相互接続網中の全通信路について、各通信路を通過するフローの重なるの数を数える
- (3) フローの重なるの数から、各通信路を通るフロー1つあたりが利用できるバンド幅を算出する。算出において、バンド幅は通信路を共有する全フローで等しく分けられる。重なるフローの数が多ければ、その分だけ1つのフローが利用できるバンド幅は小さくなる
- (4) 算出した各フローのバンド幅と各メッセージサイズから、各メッセージの通信に必要な時間を求める
- (5) 求めた時間中の最小値を次に進むシミュレーション時間とし、その分だけシミュレーションを進める
- (6) 上記の処理を、与えられた通信パターンに含まれる全通信が完了するまで繰り返す

MFS は手順(5)で決定される時間単位でシミュレーション内の時間を進める。したがって、通信の重なりが一樣である場合、大規模なネットワークであってもシミュレーションは短時間で終了する。一方、通信の重なりが一樣でない場合、最も混雑している通信路

を通るフローが利用出来るバンド幅に合わせてシミュレーションが進行するためシミュレーション実行時間は長くなるという性質を持つ。

### 3. 統計的手法による結果比較

#### 3.1 方法

ここでは、第2節で述べた MFS および Booksim で測定したスループットに基づいて、両シミュレータのシミュレーション結果を比較する。シミュレーションの実行時間は、両シミュレータ共に測定時間が十分に長くなるように設定した。

第2.2節で述べたように、Booksim は統計的な手法に基づいて測定した Average Throughput Rate (ATR) と平均レイテンシを用いて、相互接続網を評価することができる。一方、MFS のシミュレーションモデルはフローに基づくものであり、あるノードでのメッセージ送信時刻とそのメッセージを受信するノードでのメッセージ受信時刻は同じである。よって、正確なレイテンシを得ることが難しい。

Booksim による測定では、Injection Rate を 1 とし、フリットを絶え間なく相互接続網に投入することで、相互接続網をフリットで飽和させた状態で実効スループットの測定を行う。

MFS による測定もこの条件に合わせる。MFS での測定においては、シミュレーション終了時刻前に少なくとも1つのノードが通信を完了した時点までを測定時間とした。MFS のシミュレーションモデルでは、通信の送信時刻と受信時刻は同じである。よって、相互接続網はシミュレーション開始直後に飽和状態になる。ただし、シミュレーション終了時刻の少し前では、通信が完了したノードが徐々に増える。この時、相互接続網は飽和している状態とは言えない。

MFS においては、各ノードごとに最大スループットに対する実効スループットの割合を平均したものを評価に用いる。これは、Booksim が見積もる Average Throughput Rate に近い値である。トポロジには2段の Fattree および、2次元の Mesh と Torus を用いる。それぞれネットワークの大きさを  $k$ -ary 2-tree または  $k$ -ary 2-cube で表現する。Fattree については、フルバイセクションバンド幅を持つ構成とする。

Fattree トポロジにおけるルーティングは静的に行う。Mesh, Torus トポロジについては、Dimension order による X-Y routing を用いる。Booksim では同じ距離の経路が複数ある場合、どの経路を選ぶかはパケットごとにランダムで決定される。MFS ではこの場合でも静的に定義された決まりにしたがって経路

を選択する。

Booksim において通信経路が競合した場合、アービトレーションはラウンドロビンで行う。MFS は、通信路の物理バンド幅をその通信路を共有する通信の数で公平に分割することで実効バンド幅を求めている。これは相互接続網全体で公平なアービトレーションにより全通信の公平性が保たれることを意味する。

比較にあたり、Booksim にあらかじめ実装されている通信パターンの中から Uniform, Transpose, Tornado を用いた。各通信パターンの詳細は文献<sup>16)</sup>で述べられているが、ここでも簡単に説明する。

Uniform はメッセージの宛先ノード番号をランダムに決める通信パターンであり、相互接続網の様々な統計的評価に用いられる。

Transpose は  $b$ -bit で表現された受信ノード番号  $d$  の  $i$ -bit 目を、送信ノード番号  $s$  の  $i$ -bit 目から  $d_i = s_{i+b/2 \bmod b}$  で求める通信パターンである。ただし、 $0 \leq i < b$  である。このパターンは1本の対角線を中心として鏡面対称位置のノードをペアとし、通信を行う。よって、対角線から遠いノードほど、ペアとなるノードへの通信距離が長くなる。この通信パターンは、行列の変換や並び替えを行う際に現れる。Booksim の実装においては、この通信パターンは全ノード数が2のべきである場合にのみ利用可能である。

Tornado は  $k$ -ary  $n$ -cube のように、 $k$ -ary  $n$ -digit で表現されるトポロジにおいて、受信ノード番号  $d$  の  $x$  桁目の値を、送信ノード番号  $s$  の  $x$  桁目の値から  $d_x = s_x + (\lceil k/2 \rceil - 1) \bmod k$  で求める通信パターンである。2次元の Mesh または Torus トポロジ (2-ary  $n$ -cube) においては、第1次元 ( $x$  次元) 目は1桁目、2次元目 ( $y$  次元) は2桁目として表現される。このパターンは、全てのノードペアが長い距離の通信を行うため、Mesh や Torus トポロジの相互接続網に高い負荷をかける。

実際の相互接続網の構成には様々な選択がありえるが、ここでは次のような構成を用いる。VC の数は通信に必要な最小数とする。Fattree, Mesh トポロジにおいては、VC の数を1とする。Torus トポロジについては、デッドロックを回避するため、VC の数を2とする。VC ごとのバッファは2フリット分の容量を持つものとする。また、ここでは1パケットを構成するフリット数 (Flit per Packet, FPP) を変えた場合の変化をみるため、FPP を 10, 40, 80, 100 とした場合について測定を行う。

#### 3.2 Uniform 通信による比較

図2, 図3, 図4に Fattree, Mesh, Torus トポロ

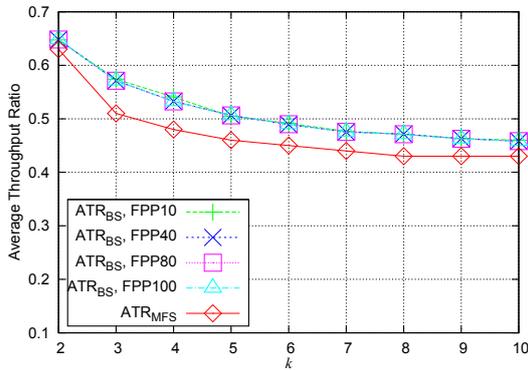


図 2 ATR の比較 (Uniform, Fattree)  
Fig. 2 Comparison of ATR (Uniform, Fattree).

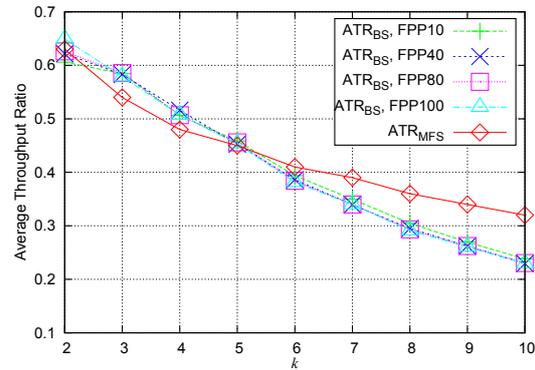


図 4 ATR の比較 (Uniform, Torus)  
Fig. 4 Comparison of ATR (Uniform, Torus).

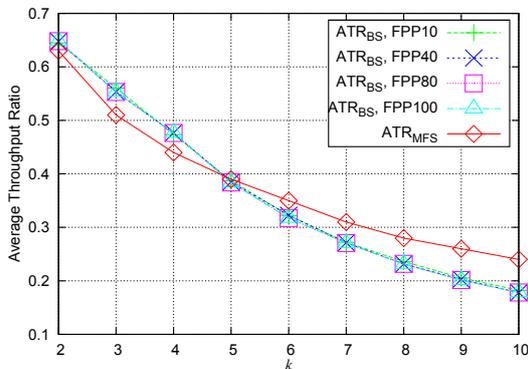


図 3 ATR の比較 (Uniform, Mesh)  
Fig. 3 Comparison of ATR (Uniform, Mesh).

ジにおける Uniform 通信のシミュレーション結果を示す。図中のグラフは横軸がネットワークの大きさ  $k$ 、縦軸が最大スループットに対する実効スループットの割合の平均 (ATR) を示している。図中  $ATR_{MFS}$  と  $ATR_{BS}$  はそれぞれ MFS と Booksim の ATR を意味する。FPP を 10, 40, 80, 100 とした場合のグラフはそれぞれ FPP10, FPP40, FPP80, FPP100 として示されている。

図 2, 図 3, 図 4 から, Uniform 通信においては, すべてのトポロジにおいて, MFS と Booksim の結果には差が生じている。この原因について考察する。

まず, Fattree に関する図 2 をみると, 図中の MFS と Booksim のグラフは,  $k$  の増加にしたがって緩やかに下がると同じ傾向を示している。

MFS および Booksim の ATR に差がある原因として, アービトレーションの影響が考えられる。MFS は, 全通信に対して相互接続網全体で公平なアービトレーションが行われるというモデルでシミュレーション

を行う。一方, Booksim では, パケットはスイッチごとにラウンドロビンによって調停される。よって, 通信経路上に多数のスイッチがある場合, その通信はそれだけ多くの調停を受けることになる。調停は各スイッチ単独では公平に行われる。しかし, 調停を受ける回数は通信によって様々であるため, 多くの調停を受けた通信とそうでない通信があるスイッチで合流し, それらの通信がさらに調停を受けると, 2 つの通信に対する調停は個々のスイッチでは公平に行われるが, 相互接続網全体で見ると公平であるとは言えなくなる。Booksim のシミュレーションにおいては, スイッチをホップする度にこの調停が行われるため, 通信ごとのホップ数にばらつきがある場合に, このような差が生じると考える。また, 今回の実験ではバッファの容量を 2 フリット分としたが, これも Booksim と MFS の差に影響を与えていると考える。

図 2 と図 3 に示す Fattree と Mesh トポロジの場合については, 全体を通して FPP10 から FPP100 のグラフに差はほとんど無い。図 4 に示す Torus トポロジの場合に着目すると,  $k = 2$  においては, FPP10 と FPP100 の点が FPP40 と FPP80 の点より若干離れた位置にある。また, FPP10 のグラフ全体が若干ながら他のグラフ全体と比較して離れている。

以降の実験では平均に近い代表値として FPP40 の値を比較に用いる。これは,  $FPP = 10, 100$  とした場合の測定値が他の場合より若干離れる場合があるという結果を踏まえたものである。また, Booksim によるシミュレーション実行時間はフリット数の増加によって長くなる。そのため,  $FPP = 80$  とするよりも,  $FPP = 40$  とする方が効率的に評価を行うことができる。実験では,  $FPP = 80$  と  $FPP = 40$  の結果はほぼ同じであるため, シミュレーション実行時間

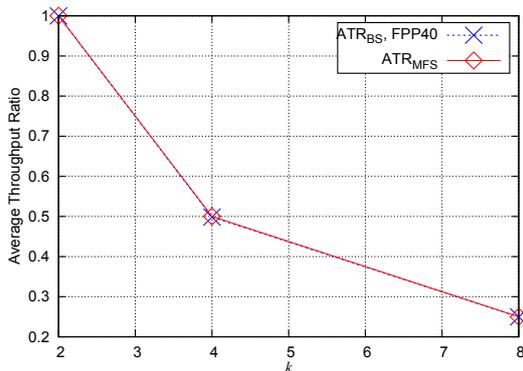


図 5 ATR の比較 (Transpose, Fattree)  
Fig. 5 Comparison of ATR (Transpose, Fattree).

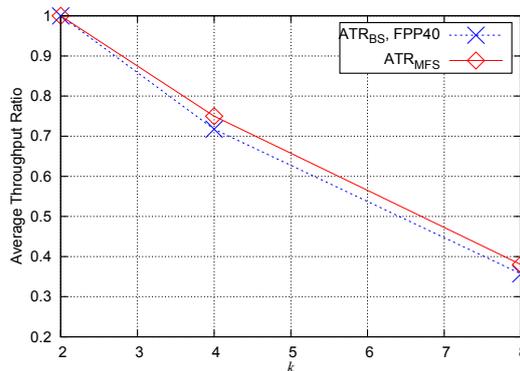


図 7 ATR の比較 (Transpose, Torus)  
Fig. 7 Comparison of ATR (Transpose, Torus).

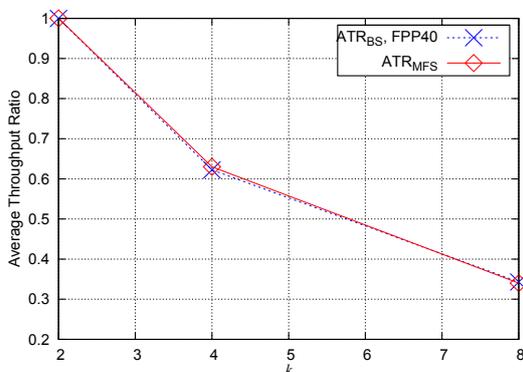


図 6 ATR の比較 (Transpose, Mesh)  
Fig. 6 Comparison of ATR (Transpose, Mesh).

がより少ない  $FPP = 40$  を用いる。

### 3.3 Transpose 通信による比較

図 5, 図 6, 図 7 に Fattree, Mesh, Torus トポロジにおける Transpose 通信のシミュレーション結果を示す。図 5 と図 6 に示す Fattree および Mesh トポロジに関しては, MFS と Booksim の ATR はほぼ一致している。Uniform 通信と異なり, Transpose 通信では特定のノードペアで規則的な通信を繰り返す。また, Fattree は全ノードペアで平均的にホップ数が少なく, なおかつそのばらつきも少ない。

Mesh トポロジにおいては, 対角線を中心として鏡面对称の位置あるノードどうしでペアが作られる。よって, 対角線から離れるほどノードペアの数が少なくなるため, やはり平均的なホップ数は少ない。よって, この場合も MFS と Booksim の差は近くなる。

図 7 に示す Torus トポロジに関するグラフを見ると, 若干の差がみられる。Torus トポロジは, Mesh トポロジと同じ理由からノードペアの平均ホップ数は

小さい。3.1 節でも述べられているが, Booksim は Dimension order の X-Y routing において, 同距離の経路が複数あると, その中からランダムで 1 つの経路を選び通信を行う。Torus トポロジにおいて  $k$  が偶数である場合は右回りまたは左回り, ないしは上まわりまたは下回りと, 少なくとも 2 通りの経路選択があり得る。Booksim はこの経路選択をパケットごとに行うため, 同じノードペアどうしの通信であっても経路が常に同じとは限らない。一方で MFS はこの経路選択を静的な決まりにしたがって決定するため, このような場合でも常に同じ経路で通信を行う。この違いが値の差に影響を与えていると考える。この時,  $k = 4$  における MFS と Booksim の ATR の差は 0.037 であった。よって, この影響は, MFS の ATR に対して 4.93% ( $=0.037/0.750$ ) 程度であり, 小さいものであると言える。ただし, この差には VC のバッファによる影響も含まれていると考える。

### 3.4 Tornado 通信による比較

図 8, 図 9 に Mesh, Torus トポロジにおける Tornado 通信のシミュレーション結果を示す。図中に  $\times$  で示されたグラフはこれまでと同じように, VC の数を 1 および 2 とした場合の Mesh および Torus トポロジにおける測定結果を示す。+ で示されたグラフは, 比較のため Virtual Output Queue (VOQ) 方式を再現するように Booksim のパラメータを設定した上で同様の測定を行った結果である。VOQ 方式は, 各スイッチの VC をノード数分用意し, 宛先ごとに専用の VC を用いる方法である。これにより, 相互接続網全体で通信が公平に調停される状態に近くなる。VC のバッファサイズはこれまでと同じように 2 フリット分とした。FPP も同様に 40 とした。

Fattree ネットワーク上の Tornado 通信は  $k$  がど

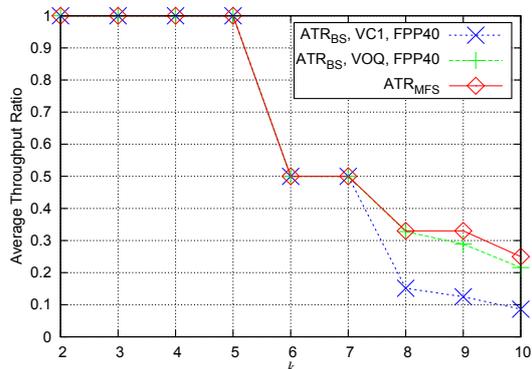


図 8 ATR の比較 (Tornado, Mesh)

Fig. 8 Comparison of ATR (Tornado, Mesh).

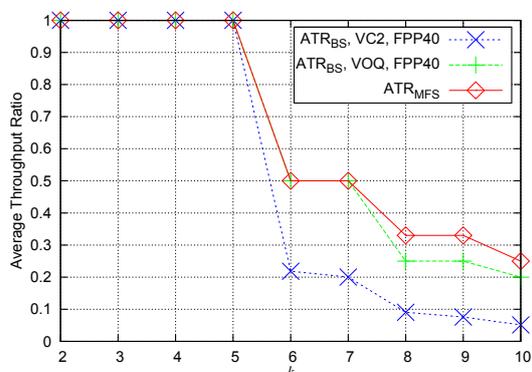


図 9 ATR の比較 (Tornado, Torus)

Fig. 9 Comparison of ATR (Tornado, Torus).

のような値でも通信の競合が起きない。実際にシミュレーションでも同様の結果を得たため、今回の比較からは除く。

図 8 に示す VC1 のグラフと図 9 に示す VC2 のグラフをに着目する。  $k \leq 5$  の場合、MFS と Booksim 共に ATR は 1 となっている。実際に Tornado 通信では  $k \leq 5$  の場合、通信の競合は起らない。

$k = 6, 7$  ではどの経路でも 2 個の通信が重なる。MFS のモデルでは、この時の ATR は 0.5 になる。図 8 に示す VC1 の時の Booksim のグラフは、MFS のグラフとほぼ一致している。一方、図 9 に示す VC2 の時の Booksim のグラフでは、この区間で MFS の ATR に対して約 0.3 の差がある。

$k \geq 8$  では、Mesh, Torus トポロジの場合、両方とも MFS と Booksim の ATR に差がある。この差の原因として、Uniform 通信の場合と同様にアービトレーションの影響が考えられる。Tornado 通信は、全てのノードペアが長い距離の通信を行うため、全

ノードペアの平均ホップ数が大きい。よって、Mesh や Torus トポロジにおいては通信経路の競合が多く発生し、アービトレーションによる差がさらに拡大すると考える。

図 8, 図 9 中に+で示す VOQ の時の Booksim のグラフは、Mesh, Torus トポロジそれぞれに示す VC1 および VC2 の場合の Booksim のグラフと比較して、より MFS のグラフに近くなっている。これは VOQ により相互接続網全体で通信が比較的公平に調停されるようになったためである。なお、Mesh トポロジにおいては  $k \geq 9$ , Torus トポロジにおいては  $k \geq 8$  でまだ MFS と Booksim (VOQ) のグラフに差がみられる。この差は、3.3 節で議論した、ルーティングの違いやバッファの影響によるものであると考える。

これまでに示した MFS と Booksim の比較から、MFS の利用範囲について考察する。MFS と Booksim のシミュレーション結果の差は、実験結果よりスイッチで行われるアービトレーションの影響により生じると考える。よって、MFS はアービトレーションをあまり行わないような通信のシミュレーションに利用できる。これには、本来通信競合があまり起きないような通信だけでなく、競合は頻繁に起きるが通信の平均ホップ数が少ないような通信のシミュレーションもこれに該当する。具体的な利用例としては、Fattree や多次元 Mesh・Torus 系トポロジのように、通信の平均ホップ数が少ないネットワークの評価、または差分法演算などのように近距離のノードと頻繁に通信を行う並列プログラムの通信シミュレーションなどが考えられる。

## 4. 大規模ネットワークにおける通信シミュレーション

### 4.1 方法

ここでは、大規模ネットワークにおいて、MFS および Booksim で通信シミュレーションを行い、そのシミュレーション結果とシミュレーション実行時間およびシミュレーションによるメモリ使用量を比較する。

シミュレーションに用いる計算機の CPU は一般のデスクトップ PC で使用されている Intel Core i7 X980, 3.33GHz であり、メモリの容量は 12GB である。トポロジには Fattree を用いる。Booksim のシミュレーションにおいては、VC の数は 2、各 VC のバッファサイズを 10 フリット分とした。FPP は 3.2 節での議論を踏まえて 40 とする。MFS のシミュレーションにおいては、1 メッセージが Booksim の 1 パケット分になるようにパラメータを与えた。通信パタ

ンは各ノードがランダムに決定する宛先に対して 10 個のメッセージを送るものとする。

シミュレーション結果から得られる通信時間の比較においては、Booksim の 1 パケットを MFS の 1 メッセージとし、この大きさを 1 Kbit と置いた。また、MFS, Booksim 共に、相互接続網中の全ての通信路の物理バンド幅を 1 Gbps とした。MFS の時間単位  $UT_{MFS}$  はメッセージサイズと物理バンド幅で  $UT_{MFS}=1 [\mu s]$  ( $= 1 \times 10^3 / 1 \times 10^9 [s]$ ) と定義される。Booksim は、1 フリットが 1 つの通信経路を伝わる時間を 1 サイクルと定義しているため、1 サイクルの時間  $UT_{BS}$  は 1 パケットあたりのフリット数を  $FPP$  とすると、物理バンド幅から  $UT_{BS} = 1/FPP [\mu s]$  と求めることができる。いま、 $FPP = 40$  なので、 $UT_{BS}=0.025 [\mu s]$  である。

メモリ使用量の比較には Linux のメモリフットプリントを top コマンドにより観測した値を用いる。Booksim はプログラム中で動的なメモリ確保と開放を多数繰り返す実装となっているため、正確なメモリ使用量を測定することは難しい。よって、今回はこの手法により、おおよその値の測定した。

## 4.2 結 果

MFS と Booksim のシミュレーション結果を表 1 にまとめる。表中の VCT は仮想通信時間 (Virtual Communication Time) の略であり、シミュレーションにより見積もられた通信時間を示す。Run-time はシミュレーションの実行に要した時間を示す。Memory はシミュレーションで使用したメモリ量を示す。

まず、VCT の値を比較する。MFS の VCT は Booksim の 0.935 ~ 1.112 倍であり、MFS と Booksim のシミュレーション結果は近くなった。

次に Run-time を比較する。MFS は Booksim の 0.020 ~ 0.010 倍程度の時間でシミュレーションを完了している。Memory についても、MFS は Booksim の 0.021 ~ 0.010 倍程度である。MFS については、 $k=24$  以上のより大規模なネットワークについてもシミュレーションを実行したが、いずれの場合もシミュレーション実行時間とメモリ使用量は小さく抑えられている。このことから、シミュレーションの実行速度および使用リソースの点から、MFS のスケーラビリティが高いことがわかる。今回の実験では、数万ノードの通信シミュレーションを、一般のデスクトップ PC 上で実行することができた。

## 5. おわりに

本論文では、筆者らが提案した Message Flow Sim-

ulator (MFS) を評価した結果について述べた。MFS は、通信を流体の流れ (フロー) として抽象化し、その流量に基づいて通信シミュレーションを行うモデルを採用している。これは、通信をソフトウェア側の視点から抽象化したものである。

本論文では、通信アルゴリズムの評価だけでなく、相互接続網の評価など、MFS のより汎用的な用途への利用可能性を示すことを目的として、MFS を既存のパケットベースシミュレータの Booksim と比較した。比較においては、両シミュレータで Uniform, Transpose, Tornado 通信と Fattree, Mesh, Torus トポロジの組み合わせでシミュレーションを行い、その結果得られた Average Throughput Rate (ATR) を元に、Booksim と MFS の違いを比較した。

比較結果から、MFS と Booksim のシミュレーション結果の差は、複数の通信がスイッチを同時に通過する際に行われるアービトレーションの影響により生じる可能性を示した。このことから、Fattree や多次元 Mesh・Torus 系トポロジのように、通信の平均ホップ数が少ない相互接続網の評価、または差分法演算などのように近距離のノードと頻りに通信を行う並列プログラムの通信シミュレーションなどにおいては、MFS が利用可能であるという知見を得た。

全てのノードが 10 メッセージ (パケット) をランダムな宛先に送るシミュレーションを実行し、その結果を比較した。測定には Intel Core i7 X980 (3.33GHz) と 12GB のメモリを搭載した計算機を用いた。MFS が見積った通信時間 VCT が Booksim に近いことを確認した上で、シミュレーション実行時間とシミュレーション中のメモリ使用量を比較すると、MFS は Booksim の 0.020 ~ 0.010 倍程度の実行時間とメモリ使用量でシミュレーションを実行可能であることがわかった。MFS については、1 万ノード以上の大規模なネットワークについてもシミュレーションを実行した。その結果、いずれの場合もシミュレーション実行時間とメモリ使用量は小さく抑えられており、シミュレーション実行速度と使用メモリ量の点から、MFS のスケーラビリティが高いことを示した。

今後は、Booksim とのシミュレーション結果の差を生む原因となったアービトレーションを再現できるよう、MFS を機能拡張することで、MFS の利用範囲をさらに広げることが課題である。

## 謝 辞

本研究を進めるにあたりご協力いただいた富士通株式会社次世代テクニカルコンピューティング開発本部

表 1  $k$ -ary 3-tree における MFS と Booksim のシミュレーション結果比較  
Table 1 Comparison of simulation results performed by MFS and Booksim on  $k$ -ary 3-tree networks.

k	# of nodes	MFS			Booksim			MFS/Booksim		
		VCT [ $\mu$ s]	Run-time [s]	Memory [MB]	VCT [ $\mu$ s]	Run-time [s]	Memory [MB]	VCT	Run-time	Memory
16	4,096	32.82	2.9	27	35.10	142.4	1300	0.935	0.020	0.021
18	5,832	33.68	4.3	39	34.00	257.2	2300	0.991	0.017	0.017
20	8,000	35.18	6.1	54	31.65	473.2	4200	1.112	0.013	0.013
22	10,648	34.51	8.7	71	34.50	855.0	7200	1.000	0.010	0.010
26	17,576	35.38	21.5	118	-	-	-	-	-	-
30	27,000	34.75	42.0	182	-	-	-	-	-	-
34	46,656	34.84	67.0	263	-	-	-	-	-	-

の追永勇次氏, 清水俊幸氏に深謝します. 本研究は, 九州大学情報基盤研究センターの研究用計算機システム, 電気通信大学の情報基盤センター教育用計算システムを利用して行われました. 本研究の一部は科研費 (22500052) の助成を受けたものです.

### 参 考 文 献

- 1) <http://www.jamstec.go.jp/esc/index.en.html>.
- 2) Hoisie, A. et al.: A Performance Comparison Through Benchmarking and Modeling of Three Leading Supercomputers: Blue Gene/L, Red Storm, and Purple, *SC '06: Proc. of the 2006 ACM/IEEE conference on Supercomputing*, p. 3 (2006).
- 3) Alam, S. R. et al.: Cray XT4: an early evaluation for petascale scientific simulation, *SC '07: Proc. of the 2007 ACM/IEEE conference on Supercomputing*, New York, NY, USA, ACM, pp. 1–12 (2007).
- 4) Barker, K. J. et al.: Entering the petaflop era: the architecture and performance of Roadrunner, *SC '08: Proc. of the 2008 ACM/IEEE conference on Supercomputing*, Piscataway, NJ, USA, IEEE Press, pp. 1–11 (2008).
- 5) 矢崎俊志, 石畑宏明: 通信アルゴリズム評価用メッセージフローシミュレータの開発, 情報処理学会論文誌 コンピューティングシステム (ACS), Vol. 3, No. 2, pp. 88–98 (2010).
- 6) Syunji, Y. and Hiroaki, I.: Message Flow Simulator for Evaluating Communication Algorithms, *Proc. of The 9th IASTED international Conference on Parallel and Distributed Computing and Networks 2010*, pp. 291–298 (2010).
- 7) <http://charm.cs.uiuc.edu/research/bignetsim/>.
- 8) Choudhury, N. et al.: Scaling an optimistic parallel simulation of large-scale interconnection networks, *Proc. for WSC '05, Winter Simulation Conference*, pp. 591–600 (2005).
- 9) Ang, B. S. et al.: Micro-architectures of high performance, multi-user system areanetwork interface cards, *Proc. of IPDPS 2000*, pp.13–20 (2000).
- 10) 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 D-I, Vol. J84-D-I, pp. 247–256 (2001).
- 11) Boku, T. et al.: INSPIRE: A general-purpose network simulator generating system for massively parallel processors, *Proc. of PERMEAN95*, pp. 24–33 (1999).
- 12) Wilmarth, T. L. et al.: Performance Prediction Using Simulation of Large-Scale Interconnection Networks in POSE, *Proc. of PADS '05*, Washington, DC, USA, IEEE Computer Society, pp. 109–118 (2005).
- 13) Wilmarth, T.L. et al.: Performance Prediction Using Simulation of Large-Scale Interconnection Networks in POSE, *PADS '05: Proc. of the 19th Workshop on Principles of Advanced and Distributed Simulation*, Washington, DC, USA, IEEE Computer Society, pp. 109–118 (2005).
- 14) 久保田和人ほか: 大規模データ並列プログラムの性能予測手法と NPB 2.3 の性能評価, 情報処理学会論文誌, Vol. 40, pp. 2293–2303 (1999).
- 15) Susukita, R. et al.: Performance Prediction of Large-scale Parallel System and Application using Macro-level Simulation, *SC08: Proc. of The International Conference for High Performance Computing, Networking, Storage and Analysis* (2008).
- 16) James, D. W. and Brian, T.: *Principles and Practices of Interconnection Networks*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2003).