Vol.2010-HPC-126 No.15
2010/8/3

Mesh・Torus ネットワーク上での最適全対全通信アルゴリズムの評価

高上治之[†1]　矢崎俊志[†2]　安島雄一郎[†3]
清水俊幸[†3]　石畑宏明[†1]

本論文では，筆者らが提案した Mesh・Torus ネットワーク上での全対全通信アルゴリズム A2AT について，フリットレベルのネットワークシミュレータである Booksim を用いて通信性能の評価について報告する．A2AT ではバーチャルチャネルを宛先ノード数分だけ用意し，全メッセージ間で公平なアービトレーションとした場合，理論値に対し，Torus では平均約 0.97 倍，Mesh では平均約 1.25 倍の通信時間がかかった．一方，より現実のマシンに近いパラメータのケースでは，理論値に対し平均約 1.11 倍の通信時間であった．既存の全対全通信アルゴリズム A2AND と比較すると，差が 1.29 倍〜1.90 倍とネットワークの構成が大きくなるほど，差が広がり A2AT の方が優位であった．各ノードからの送信数を 2，4 と増やした場合でも，送信数 1 のときと比べ，Mesh では平均約 1.13 倍早く，Torus では平均約 1.35 倍早く通信が完了した．

Evaluation of Optimal All-to-All Communication Algorithm on Mesh Network and Torus Network

Haruyuki Takaue,[†1] Syunji Yazaki,[†2]
Yuichiro Ajima,[†3] Toshiyuki Shimizu[†3]
and Hiroaki Ishihata[†1]

We evaluate the performance of previously proposed all-to-all communication algorithm (A2AT) by using Booksim, a flit level simulator. A2AT attains 0.97 to 1.25 times of theoretical communication time under nearly ideal condition that the routers in the simulation model have enough number of virtual channels.

The case when the number of virtual channels is limited to two and the routers have adequate size of buffer, this is more practical condition in actual system, A2AT achieves 1.11 times of theoretical communication time. We also show that A2AT is 1.29 to 1.90 times better than existing algorithm and its difference increases when the network size becomes large. When the numbers of concurrent message transfer are 2 and 4, A2AT performs 1.13 to 1.35 times faster than that of 1.

1. はじめに

ネットワークへの通信負荷が高い通信パターンとして全対全通信がある．全対全通信とは，全てのノードが他の全てのノードに対して，それぞれ異なった内容のメッセージを送信する通信パターンである．この通信パターンは，行列の転置，FFT など多くのアプリケーションで頻繁に使用される．近年の大規模並列計算機では，ノード数の増加に伴い，Mesh・Torus などのネットワークトポロジが用いられる事が多くなってきた．このようなネットワークを持つシステムでは，ネットワークのバンド幅を最大限に引き出す通信アルゴリズムの実現が重要となる．

特定のトポロジのネットワークで，最適な全対全通信アルゴリズムが提案されている．Scott ら[1] は，Hypercube および Mesh 上での最適な全対全通信アルゴリズムを提案している．堀江ら[2]，Yu-Chee ら[3] はそれを n 次元 Torus に拡張したものを提案している．これらのアルゴリズムは，ノード間の 1 対 1 通信をグループ化し，グループごとに通信するフェーズと休むフェーズとを組み合わせることにより，通信ネットワークのバイセクションバンド幅を 100%使用するように通信の順序をスケジューリングしている．これにより，理論的な下限の通信時間で全対全通信を行うことができる．

これらの手法の問題点として，個々のノードがメッセージの送信タイミングを同期しなければならない点が挙げられる．このアルゴリズムを実際に利用しようとすると，全ノードの同期が必須となり，これが大きなオーバーヘッドとなる．

BlueGene/L では，専用の全対全通信アルゴリズムが用いられている．この方法では，Adaptive Routing と宛先のランダム化を組み合わせることにより，100%に近い通信性能を実現している[4]．Adaptive Routing の実装が必要であり，実装が複雑となる．また，各次元のサイズが異なる Torus での通信性能を改善する方式も提案されている[5]．

†1 東京工科大学
Tokyo University of Technology
†2 電気通信大学 情報基盤センター
Information Technology Center, The University of Electro-Communications
†3 富士通株式会社
FUJITSU, LIMITED.

Mesh・Torus のように，ノードが複数のリンクをもつ場合には，複数の通信コントローラを並行動作させ，リンクを同時に使用することにより通信性能を向上することができる．最近の並列計算機は，このように 1 つのノードに複数の通信コントローラを持ち，複数のメッセージを同時に送信できるものが増えている．Bruck ら[6]，Tipparaju ら[7]，Ajima ら[8] は，そのようなシステムについて報告している．

筆者らは 2 次元の Mesh および Torus ネットワークにおいて，複数のメッセージを同時送信可能な通信コントローラをもつシステム向けに，全対全通信を最適に実行するアルゴリズム A2AT[9] を提案した．A2AT では，従来の方法のように各ノードが通信フェーズ毎に同期をとる必要はない．個々のノードは複数のメッセージを並行して送信するだけでよい．ネットワーク上での競合の影響を考慮するにあたり，ネットワーク中の全リンクについて，メッセージは公平にリンクのバンド幅を共有して流れるものと仮定した．

アービトレーションの違いや，バッファリングなどの影響は通信性能にも影響を与える．現在の多くのシステムではルータ内のメッセージをラウンドロビンで選択し，各ポートで公平に出力する実装が主流である．このようなルーターを縦続接続したネットワークでは，各ノードが同時にメッセージを送った場合，あとから合流したメッセージの方が優先度が高くなる．このため，同時に同サイズのメッセージを同じタイミングで送信した場合でも，各ノードで送信完了時間にばらつきが大きくなる．この影響を以降，グローバルアンフェアと呼ぶ．最近のハードウェアでは，ホップ数に応じてアービトレーションの優先度を変えることでグローバルフェアネスを実現した実装がある[10]．これは，遠くからきたパケットほど優先度をあげることで，各ノードで送信完了時間を公平とするようなアービトレーション方法である．アルゴリズムの提案時には，このモデルの違いによる影響については評価されていなかった．また，ルーター上でのパケットのバッファリングの影響も考慮しておらず，この点においても現実のマシン上での通信性能との乖離が懸念される．

本論文では，多くのシステムで採用されているルーター内でのラウンドロビンによるアービトレーションの影響，バッファリングの影響も考慮したシミュレーションモデルによる A2AT の性能評価を行う．2 章で対象とするネットワークについて述べ，3 章で A2AT について述べる．4 章では A2AT と既存のアルゴリズムとの全対全通信時間を比較し，5 章でまとめる．

2. 全対全通信

2.1 対象とするネットワーク

A2AT では，2 次元格子状に接続された各ルータにノードを 1 つ接続する構成の 2 次元 Mesh・Torus ネットワークを対象とする．ルータ間は双方向のリンクで接続されている．Mesh・Torus を構成する全リンクは同一のバンド幅を持ち，各リンクは同時に双方向の通信を行うことが可能なものとする．

メッセージの中継は，ワームホール方式，またはバーチャルカットスルー方式を想定している．ルーティングは Dimension-order ルーティングとする．この方式は，XY 平面上にある 2 次元 Mesh・Torus ネットワークにおいて，まずメッセージを送信元ノードから X 軸方向に送り，次に Y 軸方向に送る方式である．

直接法による全対全通信アルゴリズムとして，MPICH[11] に使用されている SimpleSpread 法 (以降，A2A) がある．直接法は通信途中でメッセージの蓄積・結合を行わず，バンド幅がボトルネックとなるような，メッセージサイズが大きい通信に用いられる．このため，直接法による通信の効率化においては，バンド幅を最大限に引き出す通信アルゴリズムの実現がより重要となる．

各ノードは，**図 1** のように，送信メッセージが格納された 1 つの送信キューに対し，複数の通信コントローラを持つ．ノードとルータは双方向の通信路で接続され，ルータ間リンクのバンド幅に対して十分に大きいバンド幅を持つものとする．各ノードは，全対全通信に必要な $(N-1)$ 回の送信を FIFO 順に並行動作する通信コントローラに割り当てていく．

2.2 既存の全対全通信アルゴリズム（A2AND）

A2A は，以下のように宛先を 1 個ずつ順に変えて送信する．

```
for i = 1    to    N − 1 do
  send((myid + i) mod N);
end for
```

ここで，N は全ノード数，$myid$ は並列処理における自分の Rank，send(t) は，ノード t へのメッセージ送信を示す．

通信ネットワークが Mesh や Torus である場合は，1 次元で順に送受信位置を移動させていく A2A の他に，d 次元座標上の相対位置を順次移動させる方法もありうる．この方法を以降，A2AND と呼ぶ．x 方向のサイズが N_x，y 方向のサイズが N_y の 2 次元 Mesh・Torus

Vol.2010-HPC-126 No.15
2010/8/3

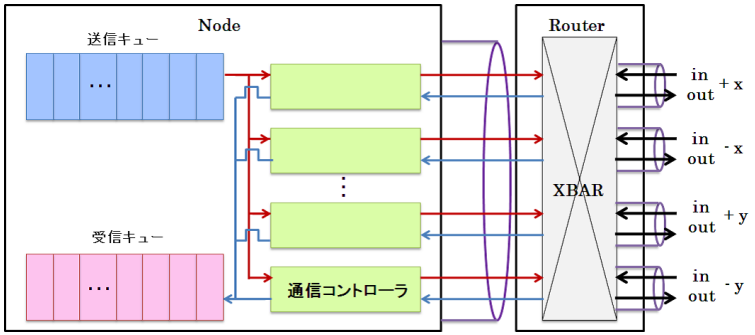

図 1 Node 内，Router 内の構成

では，送信先のノードの相対位置を 2 次元座標 (x, y) で示す．各ノードは $0 \leq x \leq N_x - 1$，$0 \leq y \leq N_y - 1$ の範囲に以下のようにメッセージを送る．

for $x = 0$ to $N_x - 1$，$y = 0$ to $N_y - 1$ **do**
  if $x \neq 0$ and $y \neq 0$ **then**
    send$((myidx + x)$ **mod** N_x，$(myidy + y)$ **mod** $N_y)$;
  end if
end for

3. アルゴリズム（A2AT）

提案した通信アルゴリズム A2AT では，ノードに接続されたリンクを有効活用することを考え，1 つのノードから行われる同一方向への複数の通信を重ねないよう，通信をスケジューリングする．MPICH で使用されている既存のアルゴリズムでは，どれも同一方向への送信が連続し，その方向のリンクのバンド幅がネックとなって全体のバンド幅を有効に利用できない．通信コントローラを複数用いるメリットを活かすため，全対全通信に必要な $(N-1)$ 回の送信を FIFO 順に並行動作する通信コントローラに割り当てていく．並行動作する通信コントローラの数を，以降 NCT とする．

一辺のサイズ N が奇数である 2 次元 Mesh において，各ノードは自分から見て，x 方向に i，y 方向に j の位置にあるノードを相対位置 (i, j) で表す．各ノードが，自分の位置をネットワークの原点として送信先を考えているため，送信範囲は，$-(N-1)/2 \leq i, j \leq +(N-1)/2$ となる．

3.1 奇数サイズの正方形 2 次元 Mesh

本論文では，奇数サイズの正方形 2 次元 Mesh および Torus ネットワーク上での A2AT の性能評価のみを行う．まず，奇数サイズの正方形 2 次元 Mesh の場合について述べ，次に Torus への拡張の順に述べる．奇数サイズの正方形 2 次元 Mesh・Torus では 2 step で全対全通信を行う．他のパターンについては文献[9] で述べられている．Mesh では NCT を 2，Torus では NCT を 4 とすることを前提とし，常に全方向のリンクを使用するように通信をスケジューリングしている．**図 2** に，Step 1，Step 2 の処理を示す．

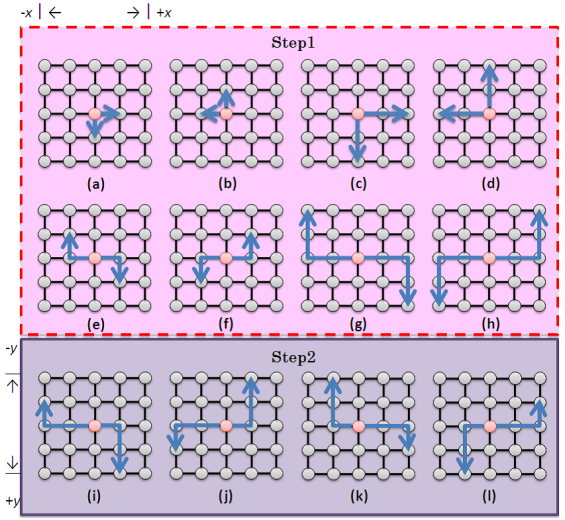

図 2 奇数サイズ正方形 2 次元 Mesh

Step 1

for $i = 1$ to $\frac{N-1}{2}$ **do**
  send$(i, 0)$; send$(0, i)$; {(a)(c)}
  send$(-i, 0)$; send$(0, -i)$; {(b)(d)}
  send(i, i); send$(-i, -i)$; {(e)(g)}

send$(i,-i)$; send$(-i,i)$; {(f)(h)}
end for

Step 1 では，まず送信元ノードは x 方向および, y 方向の軸上にあるノードに対して，**図 2**(a)〜(d) のように，x 軸上にあるノードへのメッセージの送信と y 軸上にあるノードへのメッセージの送信を組み合わせて行う．全ノードが同時にこの操作を行う事により，すべてのリンクを使用し，送信を行うことができる．

次に，送信元ノードは対角線上にあるノードに対して，**図 2**(e)〜(h) のように，x 方向, y 方向の各リンクには，ともに最大 $2i$ 個のメッセージが流れるように組み合わせて送信を行う．この場合も，すべてのリンクを使用し，送信を行うことができる．

Step 2

for $i = 1$ to $\frac{N-1}{2}$ **do**
  for $j = 1$ to i **do**
    send(i,j); send$(-j,-i)$; {(i)(k)}
    send(j,i); send$(-i,-j)$; {(i)(k)}
    send$(i,-j)$; send$(-j,i)$; {(j)(l)}
    send$(j,-i)$; send$(-i,j)$; {(j)(l)}
  end for
end for

Step 2 では，Step 1 で送信が完了した以外の位置にあるノードに対して，**図 2**(i)〜(l) のように，x 方向，y 方向の各リンクには，ともに最大 $(i+j)$ 個のメッセージが流れるように組み合わせて送信を行う．この場合も，すべてのリンクを使用し，送信を行うことができる．

3.2 2 次元 Torus への拡張

2 次元 Torus ネットワークについても 2 次元 Mesh ネットワークと同様に考えることができる．常に全方向のリンクを使用するために，通信コントローラを 4 個使い，$+x$ 方向, $-x$ 方向, $+y$ 方向, $-y$ 方向の 4 方向を組み合わせて同時に送信を行う．

一辺のサイズ N が奇数である正方形 2 次元 Torus ネットワークでは，奇数サイズの正方形 2 次元 Mesh ネットワークのアルゴリズムを，NCT を 4 として処理することにより，常に 4 方向のリンクを使用することが出来る．これにより，各リンクに流れるメッセージ数が均一となる．

3.3 A2AT の通信時間

まず，NCT を 1 のときを考える．A2AT は，ネットワーク中の全リンクについて，メッセージは公平にリンクのバンド幅を共有して流れると仮定した．

すべてのノードが自分の位置から (i,j) の位置にあるノードへメッセージを送信したとき，x 方向の各リンクに流れる最大のメッセージ数は i 個となり，y 方向の各リンクに流れる最大のメッセージ数は j 個となる．各リンクのバンド幅は，各リンクに同時に流れるメッセージ数により公平に等分されるため，x 方向のバンド幅は $1/i$，y 方向のバンド幅は $1/j$ となる．各ノードが享受できるバンド幅は経路の最少のバンド幅で律速されるので，各ノードは，$1/max(i,j)$ のバンド幅でメッセージを送信することが出来る．バンド幅のみに着目し，レイテンシを無視したときの理論値 TV_{NCT1} は，$N(N+1)(N-1)/3$ となる．

NCT を 2 としたとき，Mesh では理論的な下限の通信時間 LB_{Mesh} となる．Torus では，各フェーズの通信時間は NCT を 1 としたときの 1/2 となるため，理論値 TV_{NCT2} は TV_{NCT1} の 1/2 となる．NCT を 4 としたとき，Torus では理論的な下限の通信時間 LB_{Torus} となる．A2AT の理論的な通信時間を，**表 1** にまとめる．

表 1 A2AT の理論的な通信時間

	Mesh	Torus
$NCT = 1$	TV_{NCT1} = N(N+1)(N-1)/3	
$NCT = 2$	LB_{Mesh} = N(N+1)(N-1)/4	TV_{NCT2} = N(N+1)(N-1)/6
$NCT = 4$	-	LB_{Torus} = N(N+1)(N-1)/8

4. アルゴリズムの性能評価

4.1 Booksim

Booksim は，Stanford 大学の Dally らによって作られた，サイクルベースのフリットレベルのシミュレータである[12]．Booksim では，他のネットワークシミュレータ同様，特定の通信パターンを流し，定常状態のネットワークでスループットやレイテンシを測定する．メッセージ単位でルーティングを行い，フリット単位でフロー制御を行う．メッセージがノードからネットワークに流れるまでに 1 サイクル，ノードの受信処理に 1 サイクルかかる．また，ルーターを 1 つ通過するのにも 1 サイクルかかる．

本アルゴリズムの評価を行うため，Booksim の拡張を行った．シミュレーション可能な通信パターンとして，全対全通信アルゴリズムである，A2A，A2AND，および A2AT を

追加した．全対全通信の最後のメッセージの送信で各ノードの送信を停止させ，最も遅い最後のメッセージの到着でシミュレーションを停止させることで，全対全通信の通信時間を測れるようにした．

後述する通信時間のばらつきの影響を見るために，各ノードは他ノードからのメッセージの受信完了後，目的のメッセージの送信を開始するローカル同期モードを実装した．この方法は同期のためのオーバヘッドが小さく，ブロードキャストなどで使用されるリングアルゴリズムなどで使われている．また，ルーターからノードへのポートを増やすことで，同時送信数 NCT を増やせるように実装した．

4.2 実験方法

提案した全対全通信アルゴリズム A2AT について，既存の全対全通信アルゴリズム A2AND とのシミュレーション結果から得られた通信時間を比較した．シミュレーションの条件を**表 2** に示す．

表 2 シミュレーション条件

ネットワーク	k-ary 2-cube/mesh(k = 5，7，9，11，13，15，17).
通信パターン	A2AND，A2AT.
パケット長	1 パケット，100 フリット.
メッセージ長	1 メッセージ，1 パケット.
スイッチング	ワームホール方式，バーチャルカットスルー方式.
バッファリング	フリット単位でバッファリング.
アービトレーション	ラウンドロビンによるパケットごとのアービトレーション.
ルーティング	Dimension-order ルーティング.
バーチャルチャネル（VC）	物理チャネルあたり 2 個，ノード数個.
バッファサイズ	VC チャネル 1 個あたり，20 フリット分.

Torus の場合，通信のデッドロックを回避するために 2 つのバーチャルチャネル (VC) を用いる．通信が境界線である dateline を超えるものと超えないものとで別々の VC を使用するように，VC を切り替える．Mesh の場合，VC が 1 つでもデッドロックが発生しないが，Torus と条件を合わせるために VC を 2 つとし，動的に空いている VC を優先して使う．

4.3 シミュレーション結果

4.3.1 VC が十分な数ある場合

A2AT の通信性能を確認するため，アルゴリズム考案時に想定した状況に近い形である各ノードが VC を宛先ノード数個持ち，バッファサイズを小さくしてシミュレーションを行った．この方法では，各ノードは宛先ごとに異なった VC を使いメッセージを送信する．各ルーターでのアービトレーションはすべての VC に対して公平に行われる．よって，メッセージの通過ホップ数に依存せず，全メッセージで公平なアービトレーションとなる．

バッファリングの影響をなくすため，バッファサイズを最小限である 2 とした．この場合の結果を**図 3** に示す．横軸はネットワークの 1 辺のサイズ，縦軸は NCT を 1 としたときの理論値 TV_{NCT1} に対する比率である．

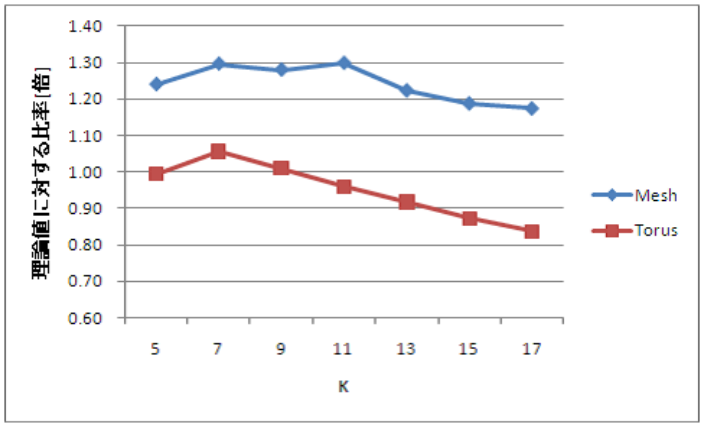

図 3 VC = 宛先ノード数

NCT を 1 としたとき A2AT では，いずれのネットワークサイズでも理論値に対する比率の変化は小さかった．NCT が 1 のとき，A2AT と A2AND の理論値は同じであり，A2AND は A2AT とほぼ同様のシミュレーション結果が得られた．

Torus では理論値 TV_{NCT1} と比較し，平均約 0.97 倍の時間で通信が完了している．理論値と異なるのは，ルーターを通過するのに時間がかかりレイテンシが伸びた点があげられる．また，A2AT 考案時に考慮していなかったバッファリングの影響により通信遅延が軽減した点もあげられる．バッファリングの影響については，4.3.3 節にて述べる．

Mesh では理論値と比較し，平均約 1.25 倍の時間で通信が完了している．Mesh では，上記の 2 点に加え右行きと左行きのメッセージでホップ数が異なることによる影響があげられる．A2AT では，同じタイミングで送信を行った通信は，同じタイミングで終わることを前提としている．この前提が，ホップ数が異なることによる影響で崩れてしまった．送信

が終わるタイミングが異なるため，異なるフェーズ間でのメッセージの重なりが発生した．送信が早く終わったノードでは，次のフェーズの通信を行うため，通信が遅れているフェーズと経路が重なり，想定外のメッセージ同士が重なった．この影響で，各ノードの各フェーズでの送信時間にばらつきが生じ，通信時間が長くなり，理論値との差が広がった．

4.3.2 VC を 2 個とした場合

各ノードが VC を 2 個持つ場合についてシミュレーションを行った．これは，実際のシステムに多く使われている構成である．この結果を**図 4** に示す．

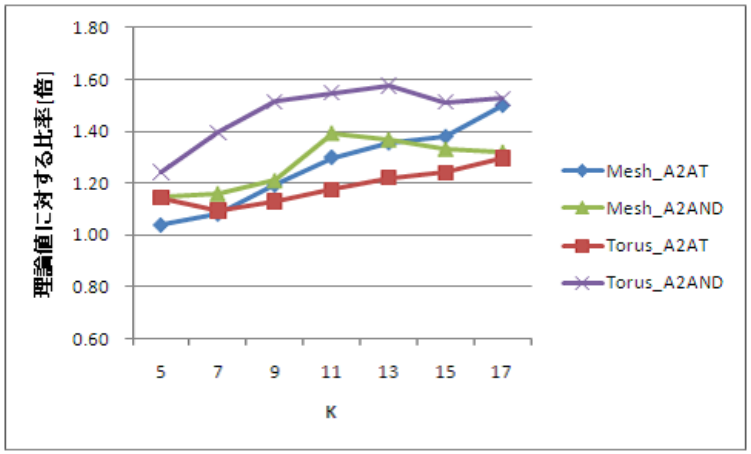

図 4 VC = 2，NCT = 1，ローカルに同期なし

Torus では理論値と比較し，平均約 1.19 倍の時間で通信が完了している．VC を 2 個とした場合，VC を十分に持つ場合のような全メッセージに対して公平なアービトレーションは行われない．そのため，グローバルアンフェアの影響により通信時間が長くなった．A2AT は，ネットワークサイズが大きくなると，Mesh ネットワーク上でメッセージを送った場合，A2AND よりも通信が遅い結果となった．

グローバルアンフェアの影響による，各ノードでのばらつきを抑えるために，ローカル同期モードでシミュレーションを行った．結果を**図 5** に示す． ローカルに送受信の同期を取ることにより，A2AT は A2AND に比べ Mesh では平均約 1.18 倍早く，Torus では平均約 1.67 倍早く通信が完了している．TV_{NCT1} と比較し，平均約 1.11 倍の時間で通信が完了している．ローカルに送受信の同期を取ったことにより，各ノードで異なったフェーズ間でのメッセージの重なりのない前提とした通信に近づき，グローバルアンフェアの影響が小さくなったためである．

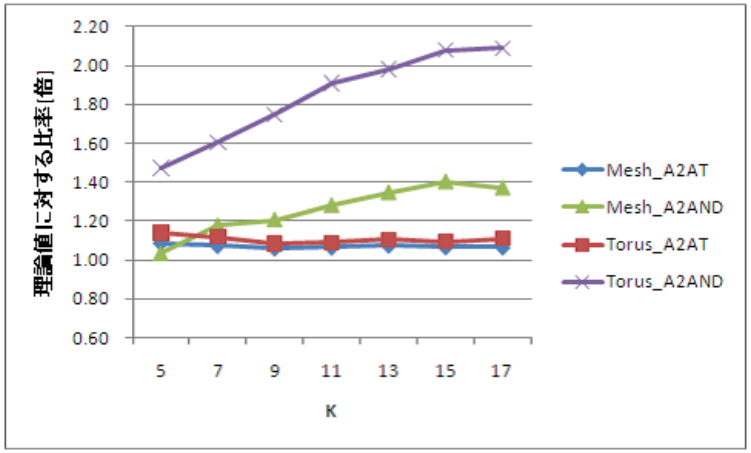

図 5 VC = 2，NCT = 1，ローカル同期モード

A2AND では，同一方向への通信が連続する性質上，同一方向のメッセージの重なりが多くなる．ネットワークサイズが大きくなるほど，経路上での重なりも多くなり，グローバルアンフェアの影響も大きくなる．そのため，ローカル同期モードでも待ち時間が長くなり，通信時間が長くなった．

NCT を 1 としたとき，ローカル同期モードで通信を行った場合，A2AT は，A2AND と比較し約 1.29 倍〜1.90 倍早く，ネットワークサイズが大きくなるほど，優位であった．

4.3.3 バッファリングの影響

図 3 に示したように，A2AT は Torus ネットワーク上でローカルに送受信の同期を取らなかった場合，理論値 TV_{NCT1} の値よりも通信が早く終わることがある．ルーター内のバッファリングにより，通信ネットワークの使用効率が上がったためと考えられる．この影響を調べるために，VC を 2 個，ネットワークの 1 辺のサイズを 11，17 とし，バッファサイズを変更し，1 パケット 100 フリットとしシミュレーションを行った．結果を**図 6** に示す．横軸はバッファサイズ，縦軸は TV_{NCT1} の通信時間に対する比率である． バッファサイズが大きくなるに従い，通信時間が短くなり，TV_{NCT1} の値よりも通信が早く終わっている．こ

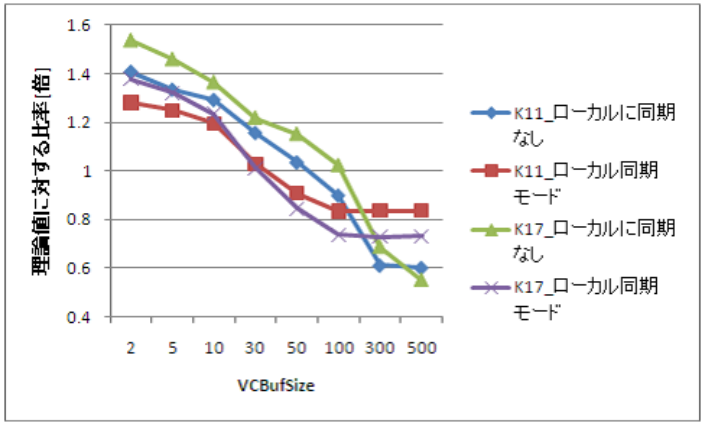

図 6 バッファリングの影響

れはパケット長に対し，十分に大きなバッファを持っているためである．

ローカル同期モードとローカルに同期を取らなかった場合を比較すると，バッファサイズが 200 を超えたところで，ローカルに同期を取らない場合の方がローカル同期モードよりも早くなる．

ローカル同期モードでは，各ノードは各フェーズでの他ノードからのメッセージの受信完了を待つため，ローカルに同期を取らなかった場合と異なり，各フェーズ間でフリットを送信するのに間隔ができる．そのため，バッファに格納されるフリット数が少なくなり，バッファサイズが大きくなるにつれ取らなかった場合に比べ通信時間は長くなる．

1 辺のサイズが 11 のとき，ローカル同期モードとローカルに同期を取らなかった場合で通信時間に平均約 10%の差があった．それに対し，1 辺のサイズが 17 のときでは通信時間に平均約 20%の差があった．ネットワークサイズが大きくなると経路上でのメッセージの重なりが多くなり，グローバルアンフェアの影響も大きくなる．そのため，ネットワークサイズが大きくなるほどローカル同期モードとローカルに同期を取らなかった場合とで差が広がる．

4.3.4 NCT を増やした場合

1 パケットを 100 フリット，1 メッセージを 1 パケット，バッファサイズは 200 フリット分とし，*NCT* を増やした場合についてもシミュレーションを行った．*NCT* を 2，4 としたときのシミュレーション結果を**図 7** に示す．

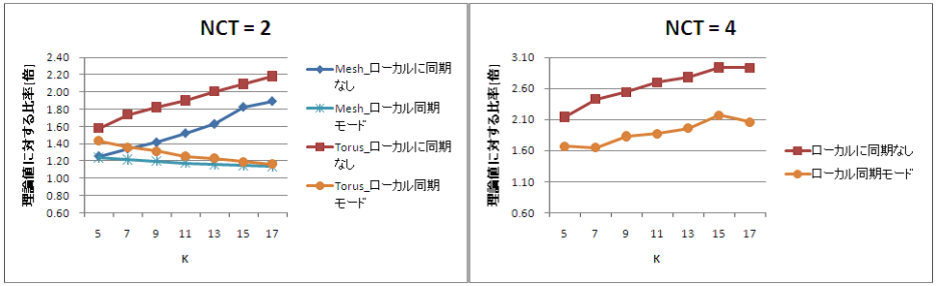

図 7 NCT = 2，4，VCBuf = 200

A2AT が *NCT* を 2 としたときの理論値と比較して，Mesh では平均して約 1.18 倍，Torus では平均して約 1.28 倍の時間で通信が完了している．

NCT を 2 としたとき A2AT は，*NCT* を 1 のときと比べ通信時間が長くなる場合がある．*NCT* を増やした場合，同時に複数の宛先から受信する．そのため，グローバルアンフェアに加え，ノード側で受信待ちによるノードごとのばらつきが大きくなったためである．ローカル同期モードで通信を行い，ノードごとのばらつきを小さくすると，Mesh，Torus ともに改善がみられた．*NCT* が 1 のときと比べ，Mesh では平均して約 1.13 倍早く，Torus では平均して約 1.35 倍早く通信が完了した．

NCT を 2 としたとき，ローカル同期モードで通信を行った場合，理論値に対し平均約 1.23 倍であり，この場合でも A2AT ではネットワークサイズによらず安定した通信時間で完了しており，有用であるといえる．

NCT を 4 としたとき，*NCT* を 2 としたときの通信時間と比較して，平均して約 0.93 倍となり通信性能の向上が得られなかった．*NCT* が 2，4 のどちらの場合でも，特定のフリットの tail が宛先が空いているにもかかわらず，宛先ノードに到達できないケースが見られた．この影響により，ノードごとのばらつき，各フェーズでの待ち時間が長くなり，理想とした通信時間と差が大きく，通信性能の向上が小さかった．宛先ノードに到達できないことに関して，原因を究明中であり今後の課題となる．

NCT を 4 としたときは，*NCT* の増加に従った通信性能の向上は得られなかったが，*NCT* を 2 としたときよりもネットワークの 1 辺サイズが 9 より小さい場合では，平均約 1.06 倍の早さで通信が完了する．ネットワークサイズが小さい場合では，*NCT* を 4 とした方が通信時間が短くなった．

Vol.2010-HPC-126 No.15
2010/8/3

5. ま と め

以前提案した全対全通信アルゴリズム A2AT について，フリットレベルのネットワークシミュレータである Booksim を用いて通信性能評価を行った．A2AT では NCT が 1 であるとき，VC が十分な数あれば Torus では理論値 TV_{NCT1} に対して，平均約 0.97 倍の通信時間となった．Mesh では右行きと左行きのメッセージでホップ数が異なる影響で，グローバルアンフェアの影響が大きくなり，平均約 1.25 倍の通信時間となった．

VC を 2 個持つ，実際のシステムに多く使われている構成では，グローバルアンフェアとなるため，VC が十分な数ある場合と比べ，通信時間は長くなった．グローバルアンフェアの影響による各ノードでのばらつきを抑えるため，ローカル同期モードでシミュレーションを行った．その結果，異なったフェーズとの重なりにより起こる通信遅延が小さくなり，TV_{NCT1} と比較し，平均約 1.11 倍の時間で通信が完了した．A2AND では同一方向への通信が連続するため，競合の影響による待ち時間が長くなり，ネットワークサイズが大きくなると，A2AT との差も広がった．

バッファリングの影響も，バッファサイズが 2 パケット分よりも小さい場合は，ローカル同期モードの方が通信時間が早い．ネットワークのサイズが大きくなるほど，ローカル同期モードとローカルに同期を取らなかった場合で差が大きく，バッファサイズが小さい場合は，ローカル同期モードで通信を行った方が良いといえる．バッファサイズが 2 パケット分よりも大きな場合では，ローカルに同期を取らない方が早かった．

A2AT は，VC の数が少なくバッファサイズも小さい場合，Mesh，Torus ともにローカル同期モードで通信を行うことで，理論値に近い値を取る．バッファサイズが 2 パケット分よりも大きい場合，Torus ではバッファリングの影響により，通信遅延の軽減の効果が大きく，ローカルに同期を取らない場合の方が通信が早く終わる．

NCT を増やした場合でも，ローカル同期モードで通信を行った場合，NCT が 1 の場合よりも通信が早く完了する．ネットワークのサイズが大きくなるほど，ローカルに同期を取った場合と取らなかった場合で差が大きく，バッファサイズが小さくネットワークのサイズが大きい場合では，すべての場合においてローカルに同期を取ったほうが通信が早く終わる．しかし，特定のフリットの tail が宛先が空いていても，宛先ノードに到達できない影響により，理論値との差が大きかった．今後の課題として，特定のフリットが宛先ノードに到達できない原因を究明し，NCT の増加に従った通信性能の向上を実現する．

本研究の一部は，九州大学情報基盤センターの研究用計算機システムを利用して行われました．

参 考 文 献

1) Scott, D.S.: Efficient all-to-all communication patterns in hypercube and mesh topologies, *6th Distributed Memory Computing Conference*, pp.398–403 (1991).
2) 堀江健志， 林憲一：トーラスネットワークにおける最適全対全通信方式，情報処理学会論文誌， Vol.34, No.4, pp.628–637 (19930415).
3) Tseng, Y.-C. and Gupta, S.: All-to-all personalized communication in a wormhole-routed torus, IEEE Trans. Parallel and Distributed Systems， Vol.7, No.5, pp. 498–505 (1996).
4) Almási, G., Heidelberger, P., Archer, C.J., Martorell, X., Erway, C.C., Moreira, J.E., Steinmacher-Burow, B. and Zheng, Y.: Optimization of MPI collective communication on BlueGene/L systems, ICS '05: Proc. 19th annual international conference on Supercomputing，New York, NY, USA, ACM, pp.253–262 (2005).
5) Kumar, S., Sabharwal, Y., Garg, R. and Heidelberger, P.: Optimization of All-to-All Communication on the Blue Gene/L Supercomputer, *37th International Conference on Parallel Processing*, pp.320–329 (2008).
6) Bruck, J., Ho, C.-T., Kipnis, S. and Weathersby, D.: Efficient algorithms for all-to-all communications in multi-port message-passing systems, *SPAA '94: Proc. 6th annual ACM symposium on Parallel algorithms and architectures*, New York, NY, USA, ACM, pp.298–309 (1994).
7) Tipparaju, V. and Nieplocha, J.: Optimizing All-to-All Collective Communication by Exploiting Concurrency in Modern Networks, SC '05: Proc. 2005 ACM/IEEE conference on Supercomputing，Washington, DC, USA, IEEE Computer Society, p.46 (2005).
8) Ajima, Y., Sumimoto, S. and Shimizu, T.: Tofu: A 6D Mesh/Torus Interconnect for Exascale Computers, *Computer*, Vol.42, No.11, pp.36–40 (2009).
9) 高上治之，矢崎俊志，安島雄一郎，清水俊幸，石畑宏明：2 次元 Mesh ネットワーク・Torus ネットワーク上での最適全対全通信アルゴリズム，情報処理学会論文誌コンピューティングシステム， Vol.3, No.2, pp.88–98 (2010).
10) Abts, D. and Weisser, D.: Age-based packet arbitration in large-radix k-ary n-cubes, SC '07: Proc. 2007 ACM/IEEE conference on Supercomputing，New York, NY, USA, ACM, pp.1–11 (2007).
11) : MPICH. http://www.mcs.anl.gov/research/projects/mpi/.
12) Dally, W. and Towles, B.: *Principles and Practices of Interconnection Networks*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2003).