B-027

2 次元メッシュネットワーク上での全対全通信アルゴリズム性能比較

Performance Comparison of All-to-All Communication Algorithms on 2-dimensional Mesh Network

高上 治之† 鈴木 悠太郎† 矢崎 俊志† 石畑 宏明†

Haruyuki Takaue Yutaro Suzuki Syunji Yazaki Hiroaki Ishihata

1. はじめに

近年では、大規模並列計算機のノード数の増加に伴い、メッシュ、トーラス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。現実に、BlueGene/L などの大規模並列計算機では、3 次元トーラスをベースとした、メッシュ・トーラス系の通信ネットワークが採用されている。このようなネットワークトポロジでは、ネットワーク上で通信メッセージを送信する際に衝突が起きやすく、通信性能が悪化する。このため、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。

2. 対象とするトポロジ

2.1 メッシュネットワーク

本論文で扱う、メッシュネットワークについて述べる。メッシュネットワークでは、2 次元格子状に接続された各ルータにノードを 1 つ接続する構成をとる。ルータ間は双方向のリンクで接続されている。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。メッシュを構成する全リンクは同一のバンド幅を持つものとする。また、ノードとルータ間も双方向の通信路で接続され、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。

ルーティングは Dimension-order ルーティングを行う。この方式は、メッセージを送信元ノードからまず X 軸方向に送り、次に Y 軸方向に送って、宛先ノードに到達させる方式である。ネットワーク中の全リンクについて、複数のメッセージが 1 つのリンクを共有していれば、その経路のバンド幅はフェアに等分される。ノードは複数の通信コントローラを持ち、各ノードの通信ソフトウェアは、FIFO 順に空いている送信コントローラを使用する。

3. 全対全通信アルゴリズム

3.1 従来のアルゴリズム

堀江ら[1]は n 次元トーラスネットワーク上でコンテンションの影響も考慮した最適な全対全通信アルゴリズムを、提案している。このアルゴリズムは、ノード間の 1 対 1 通信をグループ化し、ノード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たものである。

こ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、全ノード間でフェーズを合わせる事が必須となり、これが大きなオーバーヘッドとなる。

3.2 提案するアルゴリズム

全対全通信は、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を送信する通信パターンである。

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(ノード数－1)回の送信を FIFO 順で並行動作する通信コントローラに割り当てていく。このため、どれも同一方向への送信が連続する。同一方向への連続送信では、複数の通信を重ねても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。

そこで、同一方向への複数の通信を重ねないよう、送信する宛先の順序を工夫し、ノードに接続されたネットワークのリンクの使用率を向上させることを考える。提案するアルゴリズムでは、待ち時間を考慮しない。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良い方式をとっている。同時送信数を 2 とした時、常にすべてのリンクの使用率が 100%となるように、スケジューリングする。

3.3 奇数サイズの正方形 2 次元メッシュ

まず、一辺のサイズ N が奇数である 2 次元メッシュについて考える。各ノードは自分から見た相対位置のノードを (i, j) で表す。ただし、$i, j \in \{-(N-1)/2, \ldots, +(N-1)/2\}$ とする。提案するアルゴリズムを図 1 に示す。$send(i, j)$ は、自分の位置から x 方向へ i、y 方向へ j の位置にあるノードへのデータ送信を指す。

step1

```
1: for i = 1 to ⌊(N−1)/2⌋ do
2:   send(i,0); send(0,i);
3:   send(−i,0); send(0,−i);
4: end for
```

step2

```
1: for i = 1 to ⌊(N−1)/2⌋ do
2:   send(i,i); send(−i,−i);
3:   send(−i,i); send(i,−i);
4: end for
```

step3

```
1: for i = 1 to ⌊(N−1)/2⌋ do
2:   for j = 1 to ⌊(N−1)/2⌋ do
3:     if (i ≠ j)
4:       send(i,j); send(−j,−i);
5:       send(−j,i); send(i,−j);
6:     end if
7:   end for
8: end for
```

step4

```
1: L ← N/2;
2: for i = 1 to ⌊(N−1)/2⌋ do
3:   send(L,i); send(−i,L);
4:   send(L,−i); send(−i,L);
5: end for
6: send(L,0);
7: send(0,L);
8: send(L,L);
```

図 1　提案アルゴリズム

各ノードが、このアルゴリズムをそれぞれ実行することで全対全通信を行う。通信は自分の位置から、$+x$ 方向、$+y$ 方向、$-x$ 方向、$-y$ 方向の順に行う。このように送信することにより、同時送信数を増やした際、各リンクに流れるメッセージが公平に共有して流れるため、効率よくリンクを使用できる。本方式は、同時送信数を 2 とした時、全てのリンクの使用率を 100%とすることができる。よって、以降は全てのノードが 2 つの通信を同時に行うことを前提とする。

† 東京工科大学　Tokyo University of Technology

step1 では、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のみ、送信を行う。step2 では、対角線上にあるノードに対してのみ、送信を行う。step3 において、それ以外の位置にあるノードに対して、送信を行う。送信する際、同時に送信する 2 つのメッセージのうち、1 つ目のメッセージの x 方向への距離と、2 つ目のメッセージの x 方向への距離の和、および 1 つ目のメッセージの y 方向への距離と、2 つ目のメッセージの y 方向への距離の和が等しくなる時、重なりは距離分だけとなり最少となる。このようにすれば、メッセージの重なりが最少になる。

3.4 偶数サイズの正方形 2 次元メッシュ

一辺のサイズ N が偶数である、2 次元メッシュについて考える。まず、このネットワーク内にある最大の奇数サイズのネットワークを step1、step2、step3 と順に処理する。その後、step4 にて余った各行について、奇数サイズの時と同様の考え方で、送信するメッセージの 1 つ目のメッセージの x 方向の距離と、2 つ目のメッセージの x 方向の距離の和、および 1 つめのメッセージの y 方向の距離と、2 つ目のメッセージの y 方向の距離の和が等しくになるように組み合わせて送信を行う。

4. アルゴリズムの性能

4.1 メッシュでの全対全通信時間の下限値

1 辺が N の 2 次元メッシュネットワークを使用して全対全通信を行った場合の通信時間の下限値 T_m を求める。通信時間は、通信のボトルネックとなる経路を通る通信の回数に比例する為、この回数を求めれば良い。
まず、2 次元メッシュネットワークは、X 方向、Y 方向のどこで分割してもリンクの数は変わらないため、今は N を偶数に限定し、Y 方向で等分して考える。左半分に属している $N^2/2$ 個の各ノードが、右半分に属している $N^2/2$ 個のノードにそれぞれ通信を行う為、右側から左側に行われる通信の数は、全体で $N^4/4(=N^2/2\times N^2/2)$ になる。この通信が分割線上の N 本のリンクを通過する為、$N^3/4$ となる。奇数の場合も考慮すると、式(1) が下限値となる。

$$T_m = \left\lfloor \frac{N}{2} \right\rfloor \left\lceil \frac{N}{2} \right\rceil N \qquad (1)$$

4.2 提案したアルゴリズムの性能

本方式において送信数を 1 とした場合の通信時間 T_{o1} を考える。リンクのバンド幅を 1 とし、各ノードはサイズ 1 のメッセージを残りのノードに送るものとすると、T_{o1} は、

$$\begin{aligned} T_{o1} &= \{(\sum_{i=1}^{S} i)\times 4\} + \{(\sum_{i=1}^{S} 2i)\times 2\} + \{(\sum_{i=1}^{S}\sum_{j=i+1}^{S} j)\times 8\} \\ &= \frac{4S(S+1)(2S+1)}{3} \end{aligned} \qquad (2)$$

となる。ただし、奇数の時 $S=(N-1)/2$ とし、偶数の時、$S=(N/2)-1$、$L=N/2$ とする。この式は、次のように求めることが出来る。

まず、奇数の時を考える。距離 i に送信する場合を考えると、通信時間は i かかる。そのため、提案アルゴリズムの step1 において必要な通信時間は、式(2)の第一項で求められる。同様に step2 の通信時間も式(2)の第二項で求められる。送信数を 1 とした場合、送信順は任意であるため step3 においてかかる通信時間は、遠い方になるため、式(2)の第三項で求められる。式(2)の S に $(N-1)/2$ を代入し、奇数の場合の通信時間 T_{o1} を求めると、$\{N(N+1)(N-1)\}/3$ となる。

同様にして偶数の時を考える。step4 においてかかる通信時間は、$\{(\sum_{i=1}^{S} L)\times 4\}+L\times 2+L$ であり、これを式(2)に加算すれば、偶数の場合の通信時間が求められる。さらに、$S=(N/2)-1$ より、偶数の場合の通信時間を求めると、$\{N(2N^2+1)\}/6$ となる。

4.3 提案アルゴリズムにおける理論値

同時送信数を 2 とした時の、本方式による通信時間の理論値を求める。

まず、奇数の場合を考える。step1 における通信時間は式(3)の第一項となり、送信数 1 の時と比べ、1/2 の通信時間である。step2 に関しては、送信数 1 の時と変わらないが、step3 における通信時間は式(3)の第三項となり、送信数 1 の時と比べ、3/4 の通信時間である。式(3)の S に $(N-1)/2$ を代入し、奇数の場合の通信時間を求めると、$\{N(N+1)(N-1)\}/4$ となる。これは式(1)と、等しくなっており、同時送信数 2 とした時、本方式は下限値を引き出している事が分かる。

同様にして同時送信数 2 とした時の偶数の場合を考える。step4 においてかかる通信時間は、$[\{\sum_{i=1}^{S}(L+i)\}\times 2]+L+L$ であり、この値を式(3)に加算すれば、偶数の場合の通信時間が求められる。さらに、$S=(N/2)-1$ より、偶数の場合の通信時間を求めると、$N^3/4$ となる。式(1)と比較すると、偶数の場合も本方式は下限値を引き出している事が分かる。また、偶数の場合も奇数の場合も送信数 1 の時と比べ、約 3/4 の通信時間になっている事が分かる。

$$\begin{aligned} T_{o2} &= \{(\sum_{i=1}^{S} i)\times 2\} + \{(\sum_{i=1}^{S} 2i)\times 2\} \\ &\quad + \{(\sum_{i=1}^{S}\sum_{j=1}^{S}(i+j)\times 2 - (\sum_{i=1}^{S} 2i)\times 2\} = S(S+1)(2S+1) \end{aligned} \qquad (3)$$

5. 結論

メッシュネットワークと複数の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を提案した。提案したアルゴリズムでは、複数のメッセージを同時に送受信し、その送信方向を全経路の使用率が一定になるものを組み合わせることによりリンクの使用効率を向上させている。このアルゴリズムにより、最大限の性能を引き出すことが出来ることを示した。

今後の課題として、まず 3 次元メッシュ・トーラスへの拡張を行う。次に、$N\times N$ の正方形 2 次元メッシュではないケースについて提案したアルゴリズムを拡張し、考えていく。

参考文献

[1]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, Apr. 1993.