木構造型ネットワークにおける 最適Broadcastスケジューリング

蓬来　祐一郎[†]　西田　晃[†]　小柳義夫[†]

集合通信のスケジューリングは、通信時間を大きく左右する。今回、我々は、その影響が大きいと考えられる木構造における Broadcast の最適スケジューリングを考える。この問題は、並列計算に限らず、大規模データベースのミラーリング等にも関連し、幅広い研究がある。しかし、従来の研究では、ネットワークを抽象化したりスイッチや不均一なネットワークなどのより現実的なモデルを避けていた。本研究では、このような不均一なネットワークも考慮した場合、NP 困難な問題になることを示し、最適解の探索に深さ優先探索による分枝限定法を用いた方法を提案する。その際、木構造の対称性からくる冗長性を高速な木の同型判定アルゴリズムにより省く手法を紹介し、その有効性を検証する。また実機によるテストを行い Broadcast の実行時間が大幅に削減される場合があることを示す。

Optimal Broadcast Scheduling on Tree-structured Networks

Yuichiro Hourai,[†] Akira Nishida[†] and Yoshio Oyanagi[†]

The communication time of a group communication on a specific network depends on the scheduling of the group communication. A scheduling should be suited to network structures. In this research, we focus our attention on broadcasting on networks of tree structures by point-to-point communications. It has some applications such as mirroring a huge database in minimum time. Conventional researches assume symmetric and uniform networks. However, heterogeneity of networks exists almost everywhere. First, we show that the computational complexity of a decision problem whether broadcast will finish in given time is NP-hard. Then, we struggle this hard problem by the depth-first branch and bound algorithm with the use of a fast tree isomorphism determination algorithm. The computation time of scheduling and the communication time of the optimized broadcast will be shown.

1. はじめに

ブロードキャストは、ある 1 つのノードがもつデータを、他のすべてのノードに伝える問題である。これは集団通信の中でも最も基本的な問題であり、様々な面から研究されている。

計算量の観点からは、ネットワークをグラフ $G(V,E)$ とし、初めにノード集合 $V_0(V_0 \subseteq V)$ がメッセージを持っているような状況で、それを V 全体にメッセージを伝える問題がある。このとき、枝 $(v_i, v_j) \in E$ が存在するノード v_i, v_j の間のみで通信が可能であるとし、ある 1 ステップであるノードが通信できる相手は 1 ノードのみとしたモデルでは、ネットワークが任意のグラフの場合、最適なスケジューリングを得ることは、NP 困難であることが知られている (MINIMUM BROADCAST TIME[1])。但し、このモデルでネットワークを木構造に限った場合、多項式時間アルゴリズムが存在する。しかし、このようなモデルでは、メッセージを貯えておくことができないハブのようなものが考慮されていなかった。

隣接点間の通信に限らない、Point-to-Point 通信による Broadcast の実装としては、木構造型のアルゴリズムが好まれており、MPICH[2] もこれを採用している. しかし、これはネットワーク構造を考慮したものではない。例えば、図 1 のような単純なネットワークでプロセス番号 1 のプロセスを根として MPICH の `MPI_Bcast` を実行すると、プロセス番号 0 のプロセスを根とした場合に比べ 2 倍近くまたはそれ以上の実行時間がかかる。これはネットワーク構造を考慮せず、図 2 の下図のように遅い通信路を 2 回にわたって用いるためである。

上記の問題に限らず、ネットワークを考慮せずにプロセスの番号 (rank) を付けた場合に、回避可能な場合でも、同じ通信路を競合して使うなどの問題が起こ

† 東京大学大学院情報理工学系研究科
Dept. of Computer Science, Graduate School of Information Science and Technology, University of Tokyo

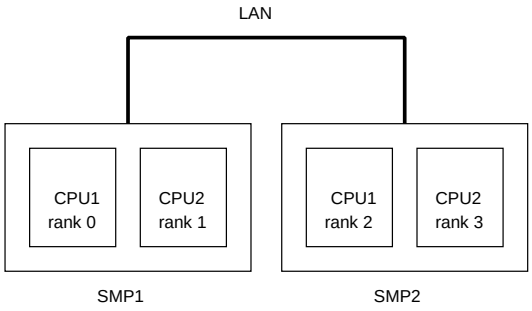


図 1 2 ノード SMP クラスタ

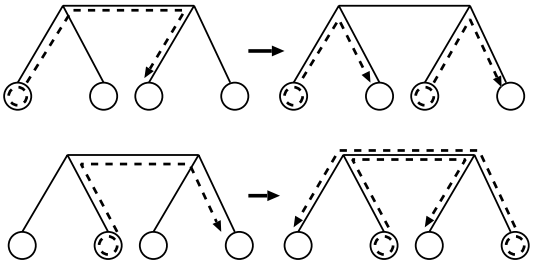


図 2 通信の競合

る場合が多々ある。

近年、グリッドの研究が活発化してきており、遠隔地との通信などボトルネックになる通信を減らすために、ネットワーク構造を考慮した MPI 実装の開発も進んでいる。[3)~5)] しかし、ネットワークを階層的に分類するに留まり、詳細なネットワークの記述にはいたっていないため、更なる性能の向上が見込まれる。また、このような研究では上位層のネットワーク性能が下位層のネットワーク性能より劣っていると仮定している場合が多い。

その他の研究としては、IP マルチキャストも上げられるが、メッセージの到達が保証されないなど、通常の並列計算等にはそのままもちいるのには向いていない点も多い。

この研究では、Point-to-Point の通信を用いて、通信性能が上下非対称であったり、構造や性能に偏りがある木構造型のネットワークで、最適なブロードキャストのスケジューリングを得ることを目的としている。

2. 通信モデル

通信に使われるネットワークのモデルは以下のように定義する。

- ネットワークはグラフ $G(V, E)$ で与えられる。
- 通信に参加するノード V_c と中継するだけのノード V_h があるとする。$(V = V_c \cup V_h, V_c \cap V_h = \phi)$
- 枝 $e = (v_i, v_j) \in E$ には、$v_i \to v_j$, $v_j \to v_i$ それぞれの向きとそれに対応したバンド幅、遅延が関連付けられている。
- 通信は、隣接するノード間のみでなくパスの存在する任意のノード間で行える。
- 1 つのノードは同時に複数のノードと通信が可能である。

ただし、通信に関する制限として、

- メッセージを持っているノードから持っていないノードへの通信のみが行われる。
- メッセージの受け取りが完了するまで送信はできない。
- 2 地点間の通信は、そのパス上の最小のバンド幅で行う。
- ある通信を行おうとしたときに、そのパス上のある枝ですでにトラフィックがあり、新しい通信が行われるとその枝のバンド幅を越える場合には、そのような通信は禁止する。

最後の制限は、2 つの通信がそのネットワークのバンド幅以上の通信を行おうとすると、他方の通信を遅れさせたり、性能が低下し、メッセージの到着時間が実装により不確定になるため、スケジューリングには適さないためである。

3. 計算の複雑さ

3-PARTITION は、$A = \{a_1 \cdots, a_{3m}\}$, $B = \sum_{i=1}^{3m} a_i/m$, $B/4 < a_i < B/2$ であるような集合 A に対して、3 つの要素を持つ m 個の集合への分割 $C_i = \{a_j, a_k, a_l\}$ $(i = 1, \cdots, m)$ $(A = \cup_i^m C_i$ かつ $\cap_i^m C_i = \phi)$ が存在するか判定する問題で、擬多項式時間アルゴリズムのない NP 完全問題である。[1)]

このインスタンスを図 3 の、ネットワークに変換し、$b_i = 1/(\frac{\lceil \log m \rceil - i}{B} + \frac{1}{\min_j a_j})$ $(i = 1, \cdots, \lceil \log m \rceil)$, $d_i = 1/(\frac{1}{\min_j a_j} - \frac{1}{a_i})$ $(i = 1, \cdots, 3m)$ とすると、ブロードキャストが $\lceil \log m \rceil / B + 1/\min a_i$ という時間で終了するか判定する問題に多項式還元可能でありこのことから、擬多項式時間アルゴリズムも、存在しない NP 困難な問題であることがわかる。これは、最適解を得るには最悪、全解探索を行うしかないことを意味する。

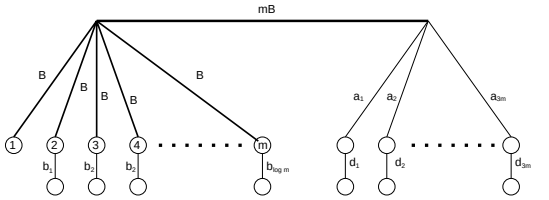


図 3 3-PARTITION からの変換。枝に付いた数字はバンド幅、最初ノード 1 だけがメッセージを持つ

この研究では、最適解を得るためにこの NP 困難な問題を解く。このとき、徐々に送信者と受信者を割り当てて行き通信にかかる時間を計算していくが、ある

ノードから見てまったく同一に見えるノード集合は、そのうちのある 1 つのノードにメッセージを送るようにスケジューリングした最適解とその他のノードに送るようにスケジューリングした最適解は、同じである。我々の研究対象となるネットワークでは、SMP や均一なクラスタなど部分的には対称な構造を含んでいることが多いため、この点を考慮するとかなりの計算を削減できることが期待できる。このような計算の削減を行うため、木の同型判定アルゴリズムを改良し、適用する。

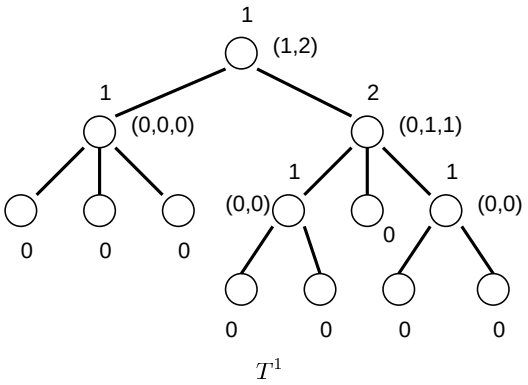
4. 木の同型判定

まず、2 つの木の判定を行うアルゴリズムは、以下のようなものである。[6),7)]

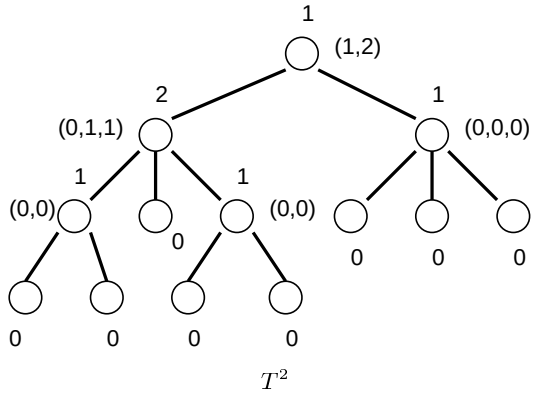
4.1 アルゴリズム

木を T^1, T^2 とし、n, d を両方の木の頂点の数と深さとする。これが一致しなければ同型でない。ノードを根からの深さでわけ、$V_0^i, \cdots, V_d^i (i \in \{1, 2\})$ とする。

(1) T^1, T^2 のすべての葉ノードに要素が番号 0 のリストをラベル付けをする。
(2) $j = d$ とする。
(3) $V_j^i (i = 1, 2)$ をラベルをもとに辞書順に従い、基数ソートする。
(4) V_j^1,V_j^2 をソートされた順でラベルを比較し異なっていたら No を出力して終了
(5) $V_j^i (i = 1, 2)$ に属する葉ノード以外に、1 から始まる新しい番号を付け直す。その際、元のラベルが一致するノードには、同じラベルを付ける。
(6) V_{j-1}^i のノードに対して、V_j に属する子のラベルを V_j のソートされた順に連結したリストとする。
(7) $j = j - 1$
(8) $j = 0$ なら終了。そうでないなら、3. へ行き手順を繰り返す。



ネットワークグラフの頂点や枝には、バンド幅等のパラメータが付いているが、上記アルゴリズムの前に、



そのようなパラメータを番号 1 から n の ID に変換しておき、更にそれをもとに木の各深さ毎に、固定長のラベルを張りつければ、ネットワーク性能の一致も考慮した同型判定が可能になる。

4.2 部分木の同型判定

上記のアルゴリズムで番号付けを行うと、ある根に対して、親子関係が決まり、ある親に対する子供同士の同型判定が自然にできるが (図 4)、親側の部分木と子供側の部分木の同型判定が行われていない (図 5)。これでは、あるノードからみた冗長性しか削減できない。

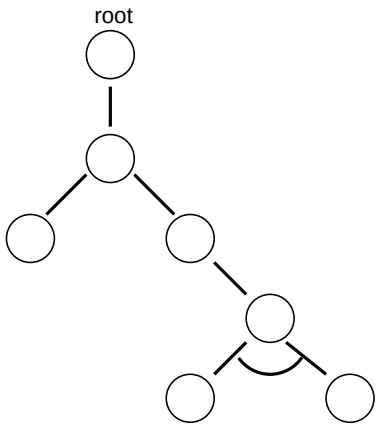


図 4 発見できる対称性。弧が同型であることを表す

では、すべての親子の組に対して同型判定を行うと計算量のオーダが増えてしまうかというと、そうではない。あるノードに対してそこに接続する部分木が対称であるには、部分木のノード数が同じでなければならないが、ある根から葉へのパスにおいては、そのような部分木は、高々1 つで、他の条件を満たす部分木とノードを共有しない。よって、判定にはノード数の線形時間がかかるだけなので、すべての親子の対称性

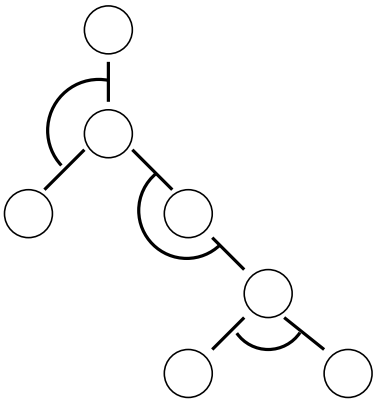


図 5 木全体の対称性

を判定するのにも全ノード数の線形時間ですむことになる。

4.3 計 算 量

上記アルゴリズムは、木の頂点の数を n とすると、ラベルにつけられる数字は、各々の深さでのノードの数を越えない。このため、基数ソートによる計算時間の合計は $O(n)$ となる。その他の部分も、各々の深さでのノード数に比例した計算しか行わないため、全体で $O(n)$ の計算時間とメモリを使い計算可能である。

5. スケジューリングアルゴリズム

スケジューリングは、深さ優先探索による分枝限定法で行い、それぞれのステップで送信可能なノードと受信ノードの組合せを生成し、スケジュールしていく。探索の順序は、時系列にしたがって探索していき、探索順序での冗長性も省く。同型判定のアルゴリズムも深さ優先探索も使用メモリが非常に少ないので、キャッシュに載った計算も可能である。

6. 実　　験

6.1 計 算 時 間

上記アルゴリズムを、実装し実行時間を計測した。使用した計算機は、CPU:Pentium4 1.8GHz 2nd-cache 256KB RAM:RDRAM 512MB OS:Linux のマシンである。まず、図 6 のような完全に対称なネットワークのモデルで最適なスケジューリングを計算した。このようなネットワークの場合、冗長性はほぼ削除できるため、通信に参加するノード数を 16 ノードとした場合、1.6 秒で最適解の探索を終了した。

実験として、図 7 のような Dual CPU のクラスタをモデル化して、ノード数を変えて、最適なスケジューリングを得るのにかかった計算時間を調べた。

NP 完全問題のため、ノードの増加による実行時間の低下は避けられないがこのモデルの場合、16cpu 程度であれば、十分実行可能な時間で計算できた。

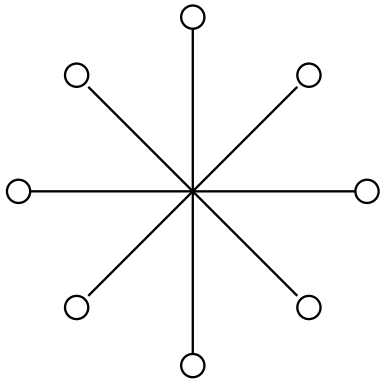


図 6 スター型ネットワーク

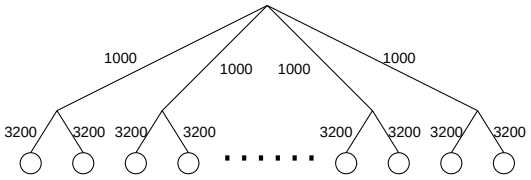


図 7 Dual CPU クラスタのモデル

表 1 Dual CPU クラスタモデルでのスケジューリング計算時間

	計算時間 (秒)
2 cpu x 4 node	0.0
2 cpu x 5 node	1.6
2 cpu x 6 node	4.8
2 cpu x 7 node	25.8
2 cpu x 8 node	2919.9

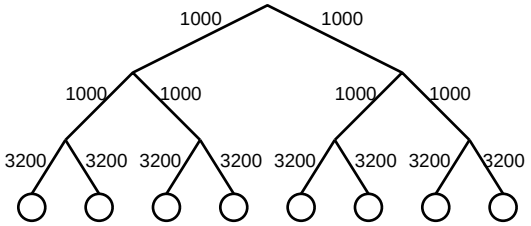


図 8 クラスタのクラスタのモデル

最後に、さらに木の階層を増やして、図 8 のような場合について計算した。

このモデルの場合、通信の参加ノードが 16 ノードのスケジューリング計算は、12 時間しても終了しなかったため打ち切った。これは、同型判定で枝刈りできる通信の組合せが減ったためである。

6.2 実機テスト

MPI[8] の集団通信関数`MPI_Bcast` とそれぞれの

表 2 クラスタのクラスタモデルでのスケジューリング計算時間

モデル	計算時間 (秒)
2 cpu x 2 node x 2 cluster	0.2
2 cpu x 3 node x 2 cluster	36.0
2 cpu x 4 node x 2 cluster	NA

表 3　1Mbyte のブロードキャスト時間 (単位は msec)

ネットワーク				MPI/SCORE		
Graph	CPUs	SMP Nodes	SMP	MPI_Bcast		最適化
Type	in SMP	in a Cluster	Clusters	root 0	root 1	
図 7	2	2	-	12.6	27.8	12.8
図 7	2	3	-	25.3	41.9	26.9
図 7	2	4	-	25.1	34.8	24.9
図 8	2	2	2	29.4	44.7	31.1
図 8	2	3	2	52.1	69.2	50.9

表 4　1Mbyte のブロードキャスト時間 (2) (単位は msec)

ネットワーク				LAM/MPI		
Graph	CPUs	SMP Nodes	SMP	MPI_Bcast		最適化
Type	in SMP	in a Cluster	Clusters	root 0	root 1	
図 7	2	2	-	30.1	29.9	22.9
図 7	2	3	-	49.0	60.3	36.6
図 7	2	4	-	62.9	81.5	36.7
図 8	2	2	2	61.9	80.5	35.7
図 8	2	3	2	111.2	118.8	49.1

`MPI_Send`, `MPI_Recv` を用い、提案アルゴリズムでスケジューリングしたものとの実行時間を比較する。実験に使用した計算機は、Gigabit ethernet で結ばれた Xeon クラスタ。

実験に用いたネットワークは、最適スケジューリング計算に用いたモデルと同等のもの。これらネットワークで MPICH の`MPI_Bcast` を用いると、根の rank を 0 とした場合には、最適なスケジュール、1 とした場合には最適にはなっていない。

MPI の実装には、MPICH/SCORE[9]、LAM/MPI[10] を用い、同じ実験を行った。それぞれの実行時間を、表 3,4 に示す。

それぞれ左側に計測したネットワーク、右側に用いた集合通信と、その実行時間が示してある。実行時間の数値は、実験を 10 回行い平均した値である。実験したシステムにおいて MPICH/SCORE では、Point-to-Point の通信が同等なはずの相手でも、通信相手によって大きく性能が異なったり、性能のゆらぎがあったため、実行時間の分散は比較的大きかった。

7. ま と め

より現実的なモデルの上で、Point-to-Point によるブロードキャストの最適化を行う手法を開発し、実験を行った。

最適なスケジューリングを求める実験では、対称性の大きいあるいは、小規模なネットワークで、最適解を得られることを示した。しかし、NP 完全問題であるため、計算量の爆発による実行時間の増加は大きかった。今後、下限計算やより多くの枝刈りなどにより、計算を削減する必要がある。

実機における実験においては、トポロジーを考慮しないブロードキャスト通信よりも実行時間を短縮可能な場合があることを示した。LAM/MPI においては、全ての場合で実行時間を短縮できた。

謝辞　なお、本研究の一部は、21 世紀 COE プログラム「情報科学技術戦略コア」超ロバスト計算原理プロジェクトのサポートによるものである。

参 考 文 献

1) Garey, M. R.: *COMPUTERS AND INTRACTABILITY, A Guide to the Theory of NP-Completeness*, W.H. Freeman (1979).
2) Gropp, W., Lusk, E., Doss, N. and Skjellum, A.: A high-performance, portable implementation of the MPI message passing interface standard, *Parallel Computing*, Vol. 22, No. 6, pp. 789–828 (1996).
3) Kielmann, T., Bal, H. E. and Gorlatch, S.: MAGPIE: MPI's collective communcation operations for clustered wide area systems., *Proc. Seventh ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP'99)*, pp. 131–140 (1999).
4) Vadhiyar, S. S., Fagg, G. E. and Dongarra, J.: Automatically Tuned Collective Communications, *SC2000* (2000).
5) Karonis, N. T., de Supinski, B. R., Foster, I., Gropp, W., Lusk, E. and Bresnahan, J.: Exploiting Hierarchy in Parallel Computer Networks to Optimize Collective Operation Performance, *Proc. 14th International Parallel and Distributed Processing Symposium (IPDPS'00)*, pp. 377–386 (2000).
6) Aho, A. V., Hopcroft, J. E. and Ullman, J. D.: *The Design and Analysis of Computer Algorithms.*, Addison-Wesley, Reading MA (1974).
7) Valiente, G.: *Algorithms on Trees and Graphs*, Springer Verlag (2002).
8) Message Passing Interface Forum: *MPI: A Message-Passing Interface Standard* (1994).
9) Sumimoto, S.: *A Study of High Performance Communication Using a Commodity Network of Parallel Computers.*, PhD Thesis, Keio University (2000).
10) Squyres, J., Lumsdaine, A., George, W., Hagedorn, J. and Devaney, J.: The Interoperable Message Passing Interface (IMPI) Extensions to LAM/MPI (2000).