木構造型ネットワークにおける最適ブロードキャストスケジューリング

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．従来の研究ではネットワークを抽象化し，ハブ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．しかし，グリッドコンピューティングへの関心や分散データベースなどの需要の増加とともにこの問題の重要性が増してきている．そこで本研究において，スケジューリングの影響が大きいと考えられる木構造におけるブロードキャストの最適スケジューリングを考える．まず，不均一なネットワークを考慮した場合，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，その有効性を示す．また実機によるテストを行い，汎用的なMPI実装のブロードキャスト関数MPI_Bcastと比較し，ブロードキャスト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 communications. Schedules should be suited to network structures. Conventional researches have assumed symmetric and uniform networks, and have neglected some practical network properties. However, heterogeneity of networks exists almost everywhere, and recent growth of grid computing increases the importance of this challenging task. In this research, we focus our attention on broadcast scheduling on networks of tree topology by one-to-one communications. Since trees have no multiple paths between any two nodes and multiple communication pairs can share a communication line, the broadcast time is sensitive on schedules. First, we propose a network and communication model and show the computational complexity of broadcast scheduling. Then, we propose an efficient algorithm to solve this hard problem by the depth-first branch-and-bound algorithm with the use of a fast tree isomorphism determination algorithm. Our experiments show the efficiency and effectiveness of our algorithm. We also show that the communication times of the optimized broadcast are greatly superior to built-in MPI_Bcast in some cases.

1. はじめに

ブロードキャストは，1つもしくは複数のノードが共通に持つデータを通信に参加する他のすべてのノードに伝え，共有させる問題である．これは集団通信の中でも最も基本的な問題であり，様々な面から研究がなされているが，近年の並列分散環境の発展やグリッド環境など計算機環境の変化などにより，研究の重要性が増してきている．

ブロードキャストの計算量に関する研究は古くから行われており，ネットワークをグラフで表しデータを持ったノードは1ステップで隣接する1ノードのみと通信可能とするモデル（MINIMUM BROADCAST TIME[1]）において，ネットワークに制限がない場合，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．ただし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．このようなネットワークや通信のモデルでは，現在のネットワークの実状とはほど遠く，リンクごとの通信性能の違いやハブのようなノードも考慮されていなかった．

このように一般には最適解を求めることが難しいため，なんらかの方法で集団通信を行わなければならない．隣接点間の通信に限らない，1対1通信によるBroadcastの実装としては，様々なものが提案されて

† 東京大学大学院情報理工学系研究科
Department of Computer Science, The University of Tokyo
†† 科学技術振興機構 CREST
CREST, Japan Science and Technology Agency

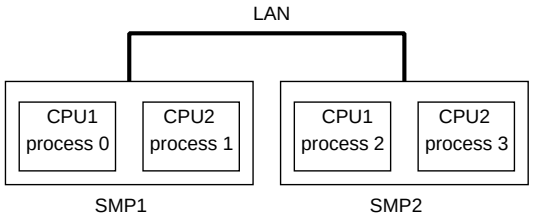

図 1 2 ノード SMP クラスタ
Fig. 1 Two-node dual processor cluster.

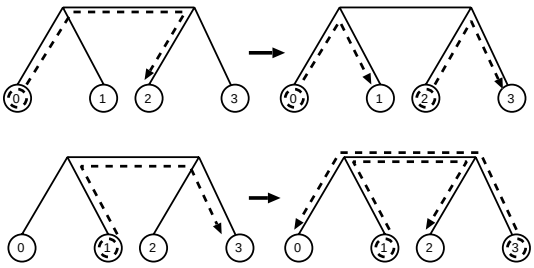

図 2 通信の競合
Fig. 2 Difference in communication load.

いる[2),3)]．しかし，これらの多くは，ネットワークを限定したものや，あるいはネットワーク構造をまったく考慮せず，組合せ的に通信回数などが最小となるように作られたものである．よって，これらのアルゴリズムによって行われる通信がネットワーク構造に適していない場合が多々ある．たとえば，**図 1** のような単純なネットワークで，図のようにそれぞれのノードのプロセスに番号が振られたとする．このとき，プロセス番号 1 のプロセスを根として binary tree ベースのブロードキャストを実行すると，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場合がある．これはネットワーク構造を考慮せずに，プロセス番号 0 のプロセスを根としたブロードキャストスケジュールに使われる通信相手の番号を単純に 1 つずつずらして利用し，**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からである．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番号が付けられた場合にも，同じ通信路を競合して使うなどの問題が生じる．

また，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したり，大きな通信遅延を考慮したりするために，ネットワーク構造を考慮した MPI 実装の開発も行われている[4)~6)]．たとえば，Vadhiyar[5)] らの方法では，ネットワークを抽象化し階層的に木構造で記述し，上位の階層で代表ノードどうしのみで集団通信を行い，下位層では再び再帰的に集団通信を行い，上位層での通信を削減している．しかし，この方法では，ネットワークを階層的に分類するにとどまり，詳細なネットワークの記述にはいたっていないため，上位層が下位層より，ネットワーク性能が優れている場合や，木構造がバランスされていない場合などには，十分にネットワーク性能をいかしきれない．また，ネットワーク性能が一様でないだけにとどまらず，近年では，通信性能が上下で非対称性を持った通信サービスが普及している．今後，これらネットワークを用いた P2P アプリケーションや分散計算環境などにおいても集団通信の利用が期待されることから，この影響は無視できない．また，メッセージを多数のノードに送信するその他の研究としては，IP マルチキャストもあげられるが，メッセージの到達が保証されないなど，通常の並列計算などにはそのまま用いるのには向いていない点も多い．またブロードキャストは通信パターンを逆にすることで集団通信の Reduce に変換できるが，マルチキャストではこれが行えない．

以上の考察から，本研究では，ネットワークのモデルをより現実に近いものとし，通信を信頼性のある 1 対 1 通信のみを用いることを考える．このような複雑なモデルを許容した最適化問題は，ネットワーク構造を木構造に限定しても，NP 困難な問題となり，多項式時間で最適解が計算可能なアルゴリズムの存在は期待できない．そこで最適解を求めるには最悪全解探索となるアルゴリズムを考えることとなるが，SMP などのように，ネットワークは部分的に対称性を多く含んでいる場合も多い．そのような場合には，同型なノードへの通信の順序を入れ換えても最適解はまったく同じとなり，下限計算などを用いてもそのままでは，枝刈りすらうまくいかないことが容易に予想される．そこで，まず，我々は，そのような冗長性を効率良く省く手法を開発し，実験によりその効果を検証する．また，既存の集団通信ライブラリのブロードキャストとその 1 対 1 通信によりスケジュールされたブロードキャストの実行時間を比較しその有効性を示す．

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，本研究においては G は木構造であるとする．
- 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，$v_i \to v_j$，$v_j \to v_i$ それぞれの向きに対応したバンド幅（> 0）および

遅延（≥ 0）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．
- メッセージの受信側は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超える場合には，そのような通信はバンド幅が確保できるまで延期する．

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．

2.1 ネットワーク状態

ネットワークの状態として，ノードごとに，すでにデータを持っているかどうかのフラグ，それぞれの枝に対して，予約されているバンド幅を時間ごとに保持する．また，スケジューリングを時系列に従って行うために，最後にスケジュールした通信の送信開始時刻を保持する．つまり，スケジューリングの順番による冗長性を省くため，送信開始時刻がネットワーク状態に記録された時刻より過去になる通信はスケジュールしない．

回線の状態

回線には，最大バンド幅と遅延が関連付けられており，b_i をバンド幅，d_i を回線の通信遅延とし，通信路は，$(b_1, d_1), (b_2, d_2), \cdots, (b_n, d_n)$ というようなパラメータ列を持つ．このパスを使用したデータサイズ D の通信は，バンド幅は $b = \min b_i$ で行われ，i 番目の回線は，$\sum_{j=1}^{i} d_i$ から D/b の時間，b のバンド幅の利用が予約されるものとする．たとえば，何も通信が起こっていない状態で**図3**の左端のノードから右端のノードへデータサイズ D の通信を行う場合には，左端のノードが通信を開始してから $d_1 + d_2 + d_3$ 単位時間後から右端のノードは受信を始め，$d_1 + d_2 + d_3 + D/b$ 単位時間後に受信が完了するものとする．

回線については，本研究の実験では，全二重の通信が可能な枝に対しては，2つの異なるネットワーク状態を保持し，半二重の通信のみが可能な枝に対しては，1つのネットワーク状態のみを保持することで，これらのモデルを表現している．その他，回線は同時に1つの通信しか行えない場合や，スイッチが1対1の通信しか扱えない場合など，様々なネットワークの制約が考えられるが，これらはスケジューリングの際に禁止事項として容易に制限が可能なため，本研究で詳細は扱わない．

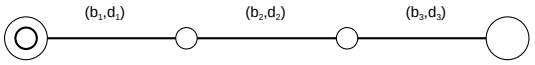

図3 通信モデル
Fig. 3 Communication model.

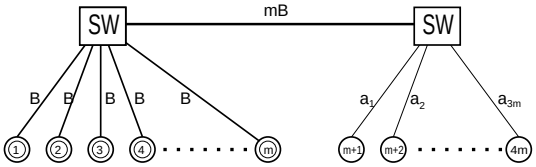

図4 3-PARTITION からの変換．枝に付いた数字はバンド幅．最初ノード 1 から m だけがデータを持つ
Fig. 4 Transformation from 3-PARTITION.

3. 計算の複雑さ

3-PARTITION [1] 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]．

3-PARTITION のインスタンスを**図4**の構造を持ったネットワークに変換する．図4において，枝に関連付けられた記号はそれぞれのバンド幅を表し，二重丸で囲まれた1から m と番号付けられたノードは初期状態でデータをすでに持っているものとし，$m+1$ から $4m$ と番号付けられたノードは，これからデータを受信するべきノードとする．また，通信遅延はすべてのノード間で0とする．このときデータサイズ1の最小のブロードキャスト時間は，3-PARTITON の解が Yes のときに限り，ID1 から m のそれぞれのノードが ID$m+1$ から $4m$ のノードから3つのノードを適切に選ぶことにより，$1/\min_i a_i$ という時間で終了する．つまり，3-PARTITION の任意のインスタンスはこのモデルでの MINIMUM BROADCAST TIME のインスタンスへ多項式還元可能である．また，多少複雑になるため詳細は省くが，3-PARTITION から1

つのノードからのブロードキャストの問題への還元できることが示せる．

このことから，$P \neq NP$ の仮定のもとで，多項式時間アルゴリズムも，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分かる．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．

4. 木の同型判定

この研究では，近似解ではなく最適解を得るために NP 困難な問題を解く．このとき，木構造型の探索により，徐々に送信者と受信者をスケジュールしていき，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．また，このような点を考慮しないと，最適解が複数存在するため，単純な枝刈りなどの手法では問題となる．このような冗長性のある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．

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[7),8)]．

4.1 木の同型判定アルゴリズム

木を T^1，T^2 とし，n，d_{max} 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．また，ノードを根からの深さで分け，それぞれの木 T^i の深さ d であるノードの集合を V_d^i $(i \in \{1,2\}, 0 \leq d \leq d_{max})$ 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する．
(2) $j = d_{max}$ 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 をソートされた順にラベルを比較し，異なっていたら No を出力して終了．
(5) $V_j^i (i = 1,2)$ に属するノードに，1 から昇順に新しい番号を付け直す．その際，ラベルが同一のノードには，同じラベルを付ける．
(6) V_{j-1}^i に属する葉ノード以外のラベルを，その子のラベルをソートされた順に連結したリストとする．

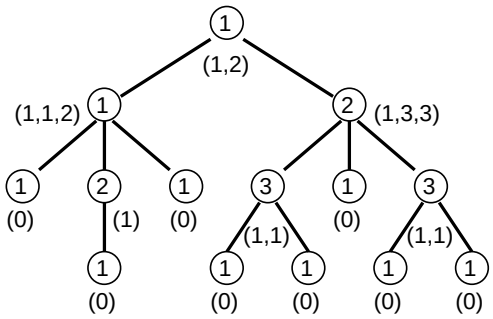

図 5 T^1
Fig. 5 T^1.

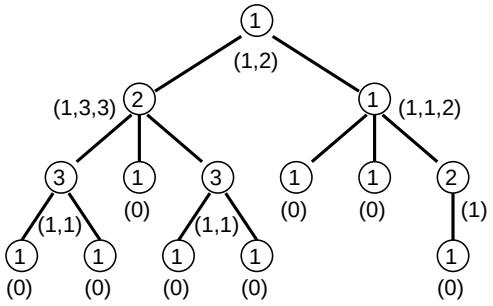

図 6 T^2
Fig. 6 T^2.

(7) $j = j - 1$.
(8) $j = 0$ なら終了．そうでないなら，ステップ (3) へ行き手順を繰り返す．

簡単な例を**図 5**，**図 6** に示す．仮ラベルとして付けられるリストは '(','(' と ',' で表されており，最終的なラベルはノード内に書かれている．2 つの木の根ノードが一致するため，これらは同型である．また，同じ深さで，同じラベルを持つノードは，そこを根として持つ部分木が同型で，ラベルが異なるノードどうしは，部分木も異なることが分かる．

ネットワークグラフの頂点や枝には，バンド幅などのパラメータが付いているが，上記アルゴリズム実行の前に，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，深さごとにラベルを付けなおす処理を行えば，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 7** 左），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 7 右）．これでは，あるノードからみた冗長性しか削減できない．

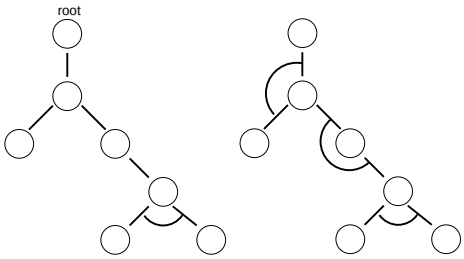

図 7 ある根から見た対称性（左）と木全体の対称性（右）．弧が同型であることを表す
Fig. 7 Isomorphisms from a root (left) and isomorphisms from any nodes.

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，そうはならない．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，たかだか 1 つで，他の条件を満たす部分木とノードを共有しない．このことから，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，ラベルに付けられる数字は，各々の深さでのノードの数を超えない．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．

4.4 受信側候補削減

冗長性の削減においては，通信状態も考慮されなければならない．しかし，通信状態も含めてすべて比較することは，効率的でない．そこで，我々の手法では，メッセージをすでに持っているノードには，固有の ID のラベルを，持っていないノードには，それとは異なる別のラベルを付ける．これによって，すでに通信が行われたノードを持つ部分木は，他の部分木と同型とならないようにする．線形時間で同型判定を行うには，ラベルはある根に対して同じ深さを持つノード数のオーダを超えてはいけないが，これはノードの深さとラベルを関連付け，基数ソートをし，新たにラベル付けを行うことにより，解決できる．**図 8** に同型の例を示す．

4.5 送信側候補削減

送信側候補削減においては，受信側候補の削減においてメッセージを持っていないノードに固有 ID のラベルを付ける点で異なるほかに，あるノードの子孫がすべて同型である場合にのみ，その中で最も早く送信可能なノードのみが通信を行う．これは，同型性を利用した部分最適解の選択による枝刈りであり，受信側の削減とは，根拠が多少異なる．**図 9** に，この基準による同型の例を示す．

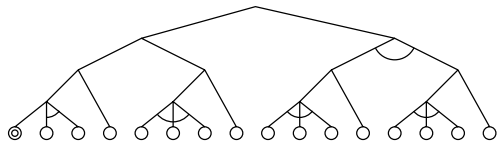

図 8 受信側候補の同型判定の例．二重丸が送信ノード．弧で結ばれた部分木のノードは同型なため冗長性を省ける
Fig. 8 An example of reducing receiver candidates.

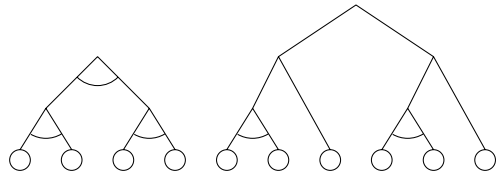

図 9 送信側候補の同型判定の例．弧で結ばれた部分木中のノードから 1 つずつ最適な候補を選ぶ
Fig. 9 An example of reducing sender candidates.

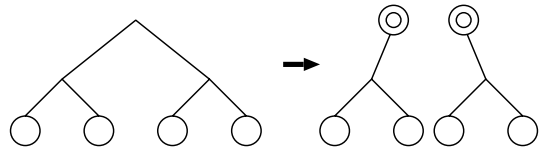

図 10 部分最適解の計算
Fig. 10 Decomposition for local optimum.

4.6 下 限 計 算

小規模なネットワークに対しては，最小のブロードキャスト時間は高速に，計算可能である．まだ，メッセージを受け取っていない部分木のブロードキャスト時間を全体のブロードキャスト時間の下限値に利用する．このために，部分木の最小ブロードキャスト時間を計算するが，切断された部分は，仮に通信ノードとしてここを根とするブロードキャスト時間を計算する（**図 10**）．これを，分解可能な部分木について計算する．

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う．以下に，再帰的に計算する手順を示す．

(1) 現在のネットワーク状態 $\mathcal{N}$ を受け取る．
(2) $\mathcal{N}$ で最後の通信が終わる時刻が現在の最適解より遅ければ枝刈りをする．また，下限計算が利用できる場合には，枝刈りを行う．

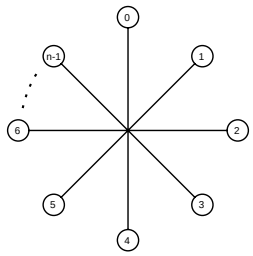

図 11 スター型ネットワーク
Fig. 11 Networks with star topology.

(3) $\mathcal{N}$ から受信側ノード集合 V_r と送信側ノード集合 V_s を計算する.
(4) 同型判定により冗長性を省いた受信送信ペアの集合 $P = \{(s,r)|s \in V_s, r \in V_r\}$ を計算.
(5) P を優先度を元にソートする.
(6) for $(s,r) \in P$
  (a) ノード s からノード r に通信を行った場合のネットワーク状態 $\mathcal{N}_{new}$ を計算.
  (b) 再帰的に $\mathcal{N}_{new}$ から始まる最適 Broadcast のスケジューリングを計算.
  (c) 得られたスケジューリング時間を現在の最適解と比較し,更新.
(7) 最適なスケジューリングを返す.

探索の順序は,時系列に従って探索していく.また探索の順序の優先度においては,現在メッセージを持つノードから遠いノードへの通信,通信の開始可能時間が早い通信,近いノードへの通信を優先した.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,小さなネットワークモデルでは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.計算には,CPU: Xeon 2.4 GHz × 2,RAM: 512 MB,OS: Linux Redhat9 の 1CPU を利用し,ネットワークモデルのパラメータは実測値をもとに決めた値を使用した.また,ブロードキャストされるデータを 1 Mbyte とした.

まず,**図 11** 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同型判定を用いた場合と用いない場合それぞれに計算した(**表 1**).SCut on/off で前述した送信側候補の削減を行ったか行わなかったかを示し,同様に RCut on/off で受信側候補の削減を行ったか行わなかったかを示してある.冗長性をまったく削減しない場合,無数に最適解が存在するため,あっという間に計算量が爆発する.しかし,冗長性削減を行うと,ほぼすべての無駄な計算を省けるため,一瞬で解くことが可能となることが分かる.

表 1 図 11 のモデルでのスケジューリング計算時間
Table 1 Calculation times for the models, Fig. 11.

ネットワーク	計算時間 (秒)			
Nodes	SCut off RCut off	SCut on RCut off	SCut off RCut on	SCut on RCut on
8	0.7176	0.1675	0.0008	0.0005
10	124.0832	13.8438	0.0016	0.0007
12	2890.3923	307.8589	0.0019	0.0009
14	>1 hour	>1 hour	0.0111	0.0010
16	>1 hour	>1 hour	0.1094	0.0012

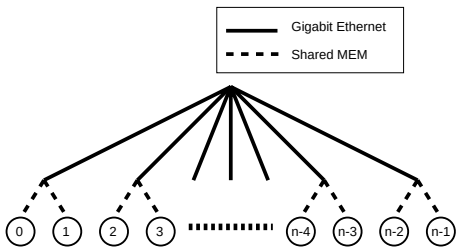

図 12 Dual CPU クラスタのモデル
Fig. 12 Models for dual-CPU clusters.

表 2 図 12 のモデルでのスケジューリング計算時間
Table 2 Calculation times for the models, Fig. 12.

ネットワーク	計算時間 (秒)			
Nodes	SCut off RCut off	SCut on RCut off	SCut off RCut on	SCut on RCut on
2 × 4	0.0858	0.1287	0.0056	0.0075
2 × 5	43.7440	41.5854	0.2105	0.1990
2 × 6	426.8857	224.3249	0.2975	0.1819
2 × 7	2577.0724	1468.8743	0.2674	0.1602
2 × 8	>1 hour	>1 hour	1.8953	0.7604

次に,**図 12** 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(**表 2**).このネットワークでは下限計算が有効になるが,最小のブロードキャスト時間が同じ場合でも,ネットワークが小さいほど得られる下限値も小さくなるため,ネットワークサイズと計算時間に相関がない場合がある.また,同じ通信ノード数でも,スター型ネットワークの場合より計算時間が短いケースがあるが,これは,スター型ネットワークより最適解に近い解の数が少ないため,枝刈りが有効に働くためであると考えられる.

さらに木の階層を増やして,**図 13** のような 2 つの Dual CPU のクラスタをつなげた場合について計算した(**表 3**).このモデルの場合においても,送信側候補,受信側候補両方の冗長性の削減をした場合には,

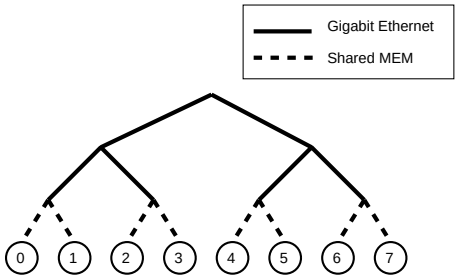

図 13 クラスタのクラスタモデル
Fig. 13 Models for clusters of two uniform clusters.

表 3 図 13 のモデルでのスケジューリング計算時間
Table 3 Calculation times for the models, Fig. 13.

ネットワーク	計算時間 (秒)			
Nodes	SCut off RCut off	SCut on RCut off	SCut off RCut on	SCut on RCut on
$2 \times 2 \times 2$	0.1002	0.1474	0.0152	0.0196
$2 \times 3 \times 2$	199.6090	136.8981	1.0716	0.7729
$2 \times 4 \times 2$	>1 hour	>1 hour	8.4765	4.0481

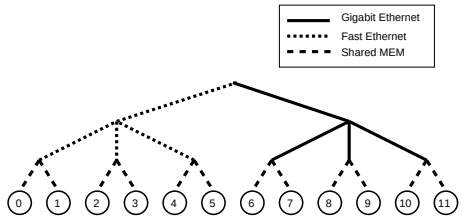

図 14 異なるクラスタのクラスタモデル
Fig. 14 Models for clusters of two nonuniform clusters.

表 4 図 14 のモデルでのスケジューリング計算時間
Table 4 Calculation times for the models, Fig. 14.

ネットワーク	計算時間 (秒)			
Nodes	SCut off RCut off	SCut on RCut off	SCut off RCut on	SCut on RCut on
$2 \times 2 \times (1+1)$	0.1463	0.1818	0.0229	0.0267
$2 \times 3 \times (1+1)$	79.3955	16.0693	0.9041	0.2406
$2 \times 4 \times (1+1)$	>1 hour	>1 hour	5546.7	146.1595

十分効率的な時間で解が得られている．

最後に，さらに対称性をくずして，ネットワーク性能の異なる 2 つのクラスタをつないだ場合（**図 14**）について計算した（**表 4**）．このネットワークでは，Fast ethernet 側のネットワークの通信時間が支配的で，Gigabit ethernet 側のスケジュールを多少変えても通信時間がさほど変わらない．このため枝刈りの効率が悪く，計算時間が大幅に増加していると考えられる．得られたこのタイプの解の例として，12 ノード問題の場合（図 14）の最適スケジュールは，$0 \to 6$，$0 \to 2$，$6 \to 8$，$6 \to 4$，$8 \to 10$，$8 \to 9$，$10 \to 11$，$8 \to 7$，$0 \to 1$，$2 \to 3$，$4 \to 5$ の順番に通信を行うスケジュールとなる．

表 5 1 Mbyte のブロードキャスト時間（1）（単位は msec）
Table 5 Broadcast times (1) (in msec).

ネットワーク				MPICH/SCORE		
Graph Type	CPUs	SMPs	Clusters	`MPI_Bcast` root 0	`MPI_Bcast` root 1	最適化
図 12	4	2	1	12.8	30.3	12.7
図 12	6	3	1	26.5	41.9	26.3
図 12	8	4	1	23.4	34.4	24.9
図 13	8	4	2	29.2	41.8	31.6
図 13	12	6	2	43.8	66.7	44.3
図 13	16	8	2	45.9	61.8	45.1
図 14	8	4	1+1	178.8	333.8	180.6
図 14	12	6	1+1	320.0	463.6	191.4
図 14	16	8	1+1	319.6	453.1	270.8

表 6 1 Mbyte のブロードキャスト時間（2）（単位は msec）
Table 6 Broadcast times (2) (in msec).

ネットワーク				LAM/MPI		
Graph Type	CPUs	SMPs	Clusters	`MPI_Bcast` root 0	`MPI_Bcast` root 1	最適化
図 12	4	2	1	12.2	13.0	12.3
図 12	6	3	1	20.7	22.2	23.1
図 12	8	4	1	29.2	30.8	21.6
図 13	8	4	2	29.5	31.2	22.0
図 13	12	6	2	46.5	48.6	32.8
図 13	16	8	2	55.2	59.4	32.4
図 14	8	2+2	1+1	259.1	269.2	174.5
図 14	12	3+3	1+1	429.4	443.2	185.6
図 14	16	4+4	1+1	514.4	531.7	260.1

6.2 実機によるブロードキャスト実行時間計測

MPI[9] の集団通信関数 `MPI_Bcast` とそれぞれの `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．`MPI_Bcast` では，ブロードキャストのルートを MPI のプロセス ID が 0 のものと，プロセス ID が 1 のものそれぞれ実験を行った．実験に使用した計算機は，Gigabit ethernet で結ばれた dual Xeon 2.0 GHz のクラスタと dual Xeon 2.4 GHz のクラスタ．NIC は，前者のクラスタが Broadcom NetXtreme BCM5700（Gigabit ethernet）と 3Com 3c905C-TX（Fast ethernet）で，後者のクラスタがオンボードの Broadcom NetXtreme BCM5703X（Gigabit ethernet）を 2 つずつ持つ．

これらのマシンがスイッチングハブに接続され，そのネットワークの接続形態が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になるようにし，SMP クラスタ，クラスタのクラスタに対して，利用台数を変えて実験を行った．

実験に用いた MPI の実装は，MPICH/SCORE-5.4.0[10]，LAM/MPI-7.0[11] である．それぞれのブロードキャストの実行時間を，**表 5**，**表 6** に示す．同期をとった後に時刻計測を開始し，ブロードキャストを行い再び同期をとった後に時間を計測し，その時間の差を通信時間とした．

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．

MPICH の実装は，データサイズが大きい場合，データをいったん分割し，ツリーベースの実装による scatter をした後に，recursive doubling algorithm を用いた allgather を行い，1 ノードのデータの送信量を減らす工夫をしている[12]．しかし，木構造のように通信どうしが互いに強く干渉してしまうネットワークや，非均一なネットワークでは理想性能が出ないため，分割しない場合と同等の性能になっている．また，根ノードが 0 以外の場合には，ノード数を法として ID を循環させているため，通信の衝突が起こりやすくなっている．LAM/MPI での実装は，5 ノード以上では binary tree ベースのアルゴリズムが用いられている[11]．

この通信時間の計測実験においては，LAM を用いたすべての場合において，スケジューリング計算で得られた理想的な通信時間とほぼ同等の通信時間となったが，SCore を用いた場合，一部で性能が得られていない場合があった．

7. まとめと考察

バンド幅やスイッチなどを考慮した，より現実的なネットワークモデルのうえで，1 対 1 通信によるブロードキャストのスケジューリングを最適化する方法を示し，効率的な冗長性削減手法を開発し，実験を行った．

最適なスケジューリングを提案手法により求める実験では，対称性の大きいネットワークでは，冗長性削減により大幅に探索時間を短縮できることを示し，また，比較的ノード数の少ないネットワークにおいては，提案手法のみでも，十分短い時間で最適解を得られることを示した．しかしながら，この最適化問題は NP 困難な問題であるため，計算量の爆発による実行時間の増加は大きく，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．ただし，最適解自体は探索のごく初期段階で発見されており，大部分の計算時間は最適性の確認となっている．

実機における通信時間を計測した実験においては，ブロードキャストのルートを変えてブロードキャストを行ったり，ネットワーク構造を変えて，最適化されたスケジューリングとの比較を行った．これらの実験において最適化されたスケジューリングによるブロードキャスト通信は，トポロジーや通信性能を考慮しない実行時間が大幅に短縮可能な場合があることを示した．また，MPICH/SCore と LAM/MPI において MPI のライブラリ関数 `MPI_Bcast` 比較し，MPICH/SCore では，対称的なネットワークにおいて，適切なプロセスを根としたブロードキャストを行った場合には，比較的良い結果が得られたが，根のプロセスを変えると大幅に性能が悪くなることを示し，LAM/MPI ではノード数が増加するにつれ最適解と比較し，大幅に遅くなることを示した．

最後に，提案手法の問題点として，ネットワークが木構造を持つことを利用して，効率的な同型判定を行い冗長性を削減しているが，同様の手法を用いて同型判定を一般のグラフで行うことは，非常に難しいことが知られている．また，ノード間に複数の通信経路がありうる場合には，通信経路が静的に決まっていないと，経路も同時に考慮しなければならず，さらに問題が複雑になり，そのまま拡張することは適当でない．このため，ヒューリスティックスや近似アルゴリズムの開発が重要な研究課題となってくると思われる．

謝辞 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，および科学技術振興機構 CREST 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) Al-Dubai, A.Y., Ould-Khaoua, M. and Mackenzie, L.: A Scalable Plane-Based Broadcast Algorithm for 3D-Mesh Networks, *11th Euromicro Conference on Parallel, Distributed and Network-Based Processing*, pp.149–156 (2003).
3) Chen, Y.-S., Juang, T.-Y. and Shen, Y.-Y.: Multi-Node Broadcasting in an Arrangement Graph Using Multiple Spanning Trees, *IEEE ICPADS'2000*, pp.213–220 (2000).
4) Kielmann, T., Bal, H.E. and Gorlatch, S.: MAGPIE: MPI's collective communcation operations for clustered wide area systems., *Proc. 7th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming* (*PPoPP'99*), pp.131–140 (1999).
5) Vadhiyar, S.S., Fagg, G.E. and Dongarra, J.:

Automatically Tuned Collective Communications, *SC2000* (2000).
6) 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).
7) Aho, A.V., Hopcroft, J.E. and Ullman, J.D.: *The Design and Analysis of Computer Algorithms.*, Addison-Wesley, Reading MA (1974).
8) Valiente, G.: *Algorithms on Trees and Graphs*, Springer Verlag (2002).
9) Message Passing Interface Forum: *MPI: A Message-Passing Interface Standard* (1994).
10) Sumimoto, S.: A Study of High Performance Communication Using a Commodity Network of Parallel Computers, Ph.D. Thesis, Keio University (2000).
11) Squyres, J., Lumsdaine, A., George, W., Hagedorn, J. and Devaney, J.: The Interoperable Message Passing Interface (IMPI) Extensions to LAM/MPI (2000).
12) 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).

(平成 15 年 7 月 31 日受付)
(平成 15 年 12 月 9 日採録)

蓬来祐一郎（学生会員）

1977 年生．2000 年東京大学理学部情報科学科卒業．2002 年同大学大学院理学系研究科情報科学専攻修士課程修了．現在，同大学院情報理工学系研究科コンピュータ科学専攻博士課程在学中．

西田　晃（正会員）

1970 年生．1995 年東京大学理学部情報科学科卒業．1998 年同大学大学院理学系研究科情報科学専攻博士課程修了．理学博士．同年より東京大学大学院理学系研究科情報科学専攻助手．2002 年より科学技術振興事業団戦略的創造研究推進事業「シミュレーション技術の革新と実用化基盤の構築」領域研究代表者を兼務．反復解法，特に大規模固有値解法と並列数値処理の研究に従事．ACM，IEEE，SIAM，日本応用数理学会，日本ソフトウェア科学会各会員．

小柳　義夫（正会員）

1943 年生．1966 年東京大学理学部物理学科卒業．1971 年同大学大学院理学系研究科物理学専門課程修了，理学博士．同年同大学助手．高エネルギー物理学研究所理論部門助手，筑波大学電子情報工学系講師，助教授，教授を経て，1991 年東京大学理学部情報科学科教授．並列処理，数値解析，計算物理学に関する研究に従事．特に，偏微分方程式の高速並列解法，最小二乗法の数値計算，乱数やモンテカルロ法に興味を持つ．物理学会，日本統計学会，応用統計学会，計算機統計学会，応用数理学会等各会員．