トポロジを考慮したノード間通信方式の提案

大野 善之† 星 宗王‡ 加納 健‡
NEC システムプラットフォーム研究所†
NEC HPC事業部‡

1.はじめに

近年，並列計算機システムは大規模化しており，システムが占める面積自体も大きくなり，ノード間のケーブル長による通信レイテンシはますます大きくなっている．そのため，多数の計算機が通信に参加する集合通信におけるノード間の同期処理のオーバーヘッドが大きくなっている．

そこで本研究では，通信ステップ間の同期処理を行うことなく，集合通信の時間を短縮するための方式を提案する．

2.関連研究

集合通信を短時間で効率良く行うことを目的とした通信方式の研究がなされている．

Chanら[1]は，各種集合通信の基本アルゴリズムであるMinimum-Spanning Tree (MST)アルゴリズムとBucketアルゴリズムのN次元トーラス/メッシュ網への適用法を提案している．Chanらの方式を使うと，N次元トーラス/メッシュ網においてMSTアルゴリズムやBucketアルゴリズムの各ステップで通信パス競合が発生しないようにできる．

また，Suhら[2]は，N次元トーラス/メッシュ網におけるAlltoAll通信のアルゴリズムを提案している．Suhらの方式では，複数のノードに対するデータを組み替えて転送することでステップ数を少なくし，通信の起動にかかる時間分のコストを削減している．

上記アルゴリズムを含む多くの集合通信アルゴリズムでは，通信を複数のステップに分け，各ステップにおいて通信パス競合が発生しないように送信順序を工夫している．しかし，異なる通信ステップが同時に実行された場合には通信パス競合によって性能が低下するため，通信ステップ毎にノード間で同期をとる必要がある．この同期にかかる時間のために，集合通信にかかる時間が長くなるという問題がある．

3.提案方式

本研究では，ステップ間の同期処理を行うことなく，既存アルゴリズムの通信パタンで集合通信を行うための方式を提案する．提案方式は，(1)通信開始時刻を統一する同期と(2)通信ステップ間の待ち合わせからなる．1ステップ目において通信開始時刻を統一し，以降の各通信ステップでは，各ノードの送信データサイズとノード間レイテンシを考慮することで通信を待ちあわせることで次ステップの通信開始時刻を統一する(図1)．

以下，本提案方式の詳細を説明する．

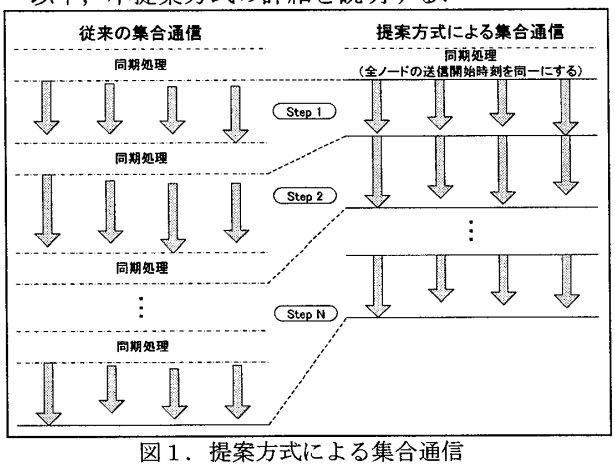

図1．提案方式による集合通信

3.1 通信開始時刻統一用同期処理

本節では，1ステップ目の通信時刻を統一する同期処理について述べる．本同期処理は，各ノード間のレイテンシを各ノードが把握できていることを前提とする．

本同期方式は，全ノード間で同期パケットを送受信することで同期をとる方式である．各ノードでは，1クロックに1ずつ減算されるタイマ，および，受信した同期パケット数を記録するカウンタを保持しており，同期開始時は0に初期化されている．

各ノードは，同期パケットを受信すると，カウンタ値とタイマを更新する．カウンタ値は1だけインクリメントし，タイマ値は，同期パケットの送信元ノードからの通信レイテンシが最大であるノードに同期パケットが到達するまでの時間と，現在のタイマ値の大きい側に更新する．このように，同期パケットを送受信してカウンタとタイマを更新していき，カウンタ値=全ノード数−1，タイマ=0 になったときに同期処理を完了する．これにより，同期処理の完了が最も遅いノードの完了時刻を知ることができ，全てのノードがその時刻に集合通信の開始時刻を統一することができる．

A Proposal of Inter-node Communication based on Topology
† Yoshiyuki Ohno, System Platforms Research Laboratories, NEC Corporation
‡ Noritaka Hoshi and Yasushi Kanoh, HPC Division, NEC Corporation

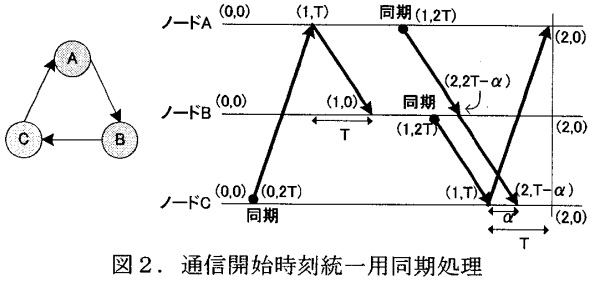

図2．通信開始時刻統一用同期処理

3.2 通信ステップ間の待ち合わせ

前節の同期方式により，1ステップ目の通信開始時刻を統一することができる．以降のステップにおいては，ステップ毎で最も遅く送受信が完了する時刻を推定し，推定した時刻まで各ノードが次ステップの送信開始を待つことで，2ステップ目以降の通信開始時刻を統一することができる．

通信時刻の推定には，同じステップの全てのノードの通信についての送信データサイズとノード間のレイテンシを用い，送信側の送信開始時から受信側の受信完了時までの時間を計算すればよい．

4.評価

提案方式の効果をシミュレーションにより評価した．評価には，Suh らの AlltoAll 通信アルゴリズム[2]を用い，3次元トーラス網で AlltoAll 通信を行う場合の総通信時間を計測した．

4.1 トーラス網における同期

トーラス網では，全ノード間で同期パケットを送受信して同期をとるのではなく，次元ごとに同期をとるほうが効率がよい．これは，本提案方式の同期方式にも応用できる．各次元のノード群との同期処理の完了時刻を統一し，次の次元の同期処理の開始時刻をそのノード群で統一することで，並列計算機システム全体の同期処理の完了時刻を統一することができる．

4.2 AlltoAll 通信アルゴリズム

本評価で採用した AlltoAll 通信アルゴリズムの概略を説明する．本アルゴリズムでは，N 次元トーラスで接続されたノードを 4^N ノードずつのノードグループに分割し，前半のフェーズでノードグループ間におけるデータの送受信を行い，後半のフェーズでノードグループ内のデータの送受信を行う．各フェーズの通信は複数のステップに分割でき，同一ステップにおいては通信パスの競合が発生しないようになっている．

4.3 評価モデル

本評価では，16×16×16 ノードが 3 次元トーラス網で接続された並列計算機を想定する．ノード間のケーブルや SerDes など各通信用回路を含めた通信レイテンシは X 方向，Y 方向，Z 方向の順に 21[ns]，25[ns]，70[ns]とする．また，隣接するノード間のスループットは 16[GB/s]であるとする．

4.4 結果と考察

本評価では，ステップ毎に同期をとる方式(従来方式1)，通信開始時のみ同期をとる方式(従来方式2)，提案方式の3種類のシミュレーションを行った．シミュレーション結果を図3に示す．

転送データサイズに関わらず，従来方式1よりも提案方式のほうが AlltoAll 通信にかかる時間は約 24[μs]短くなり，提案方式によって同期処理にかかる時間を削減できていることがわかる．

また，従来方式2と比較しても，提案方式のほうが AlltoAll 通信の時間は短くなっている．通信ステップ毎の同期をとらないことにより発生する通信衝突の影響が大きく，提案方式による通信ステップ毎の待ち合わせの時間コストのほうが小さいことがわかる．また，従来方式2では，各ノードの転送データサイズが不均一な場合に，性能が大きく劣化するが，提案方式はそのような場合でも性能低下は小さくなると考えられる．

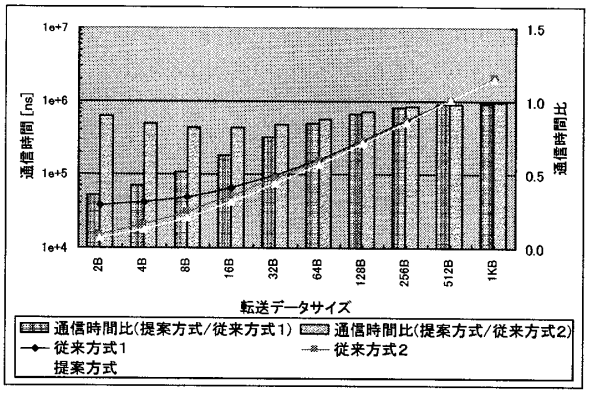

図3．シミュレーション結果

5.おわりに

本稿では，通信ステップ間の同期処理とることなく集合通信を実施するために，通信開始時刻を統一する同期および通信ステップ間の待ち合わせのための方式を提案した．また，シミュレーション評価により，通信ステップ間で同期処理をとっていた従来の集合通信と比較し，集合通信にかかる時間が短くなることを示した．

参考文献

[1] E. Chan *et al.*: Collective Communication on Architectures that Support Simultaneous Communication over Multiple Links, In Proc. of PPoPP 2006, Pages 2-11, 2006.

[2] Y.J. Suh *et al.*: All-to-All personalized communication in multidimensional torus and mesh networks, IEEE Transactions on Parallel and Distributed Systems, Volume 12, Issue 1, Pages 38-59, 2001.