自動取得したネットワーク構成情報に基づくMPI集合通信アルゴリズムの改良

吉富 翔太[†] 斎藤 秀雄[‡] 田浦 健次朗[‡] 近山 隆[††]

[†]東京大学工学部 [‡]東京大学大学院 情報理工学系研究科 [††]東京大学大学院 新領域創成科学研究科

1 はじめに

近年グリッド環境が大規模化するのに伴い，グリッドを用いた並列分散計算の処理時間における計算機間の通信時間が占める割合が増加している．この計算処理時間を短縮するためには多数の計算機が通信に参加する集合通信を効率良く行うことが必要不可欠であり，集合通信の性能向上を目的とした多くのアルゴリズム [1, 2] が研究・開発されてきた．

しかし，これらは，大半がネットワークのトポロジーに関する情報を事前に設定した上で，利用されることを前提としている．またノードとスイッチが多段に繋がりあって構成される階層構造についてはあまり考慮をしていない．そのため，実行環境のトポロジーを知らないユーザが計算処理を行いたい場合や，負荷の少ないノードで計算させることを意図し，実行直前まで使用するノードが確定しない場合にはこれらのアルゴリズムをそのまま用いても期待した性能を得るのは難しい．また，ノードやスイッチが多段階層構造を成す場合には単純にクラスタ内やクラスタ間のノード間遅延のみを考慮して通信するアルゴリズムでは性能が低下することが考えられる．

そこで，本稿ではMPI[3]の集合通信に着目し，あらかじめ推定したネットワークトポロジーと計算機間の遅延の情報に基づき，より短時間で通信を行えるように既存の集合通信アルゴリズムを改良・拡張する．そして複数のクラスタを用いたグリッド環境において，提案した集合通信のモデルを実装し，性能の評価を行う．

2 関連研究

本章では本研究で必要になるネットワークトポロジーの推定方法ネットワークトポロジーの推定方法及び，既存の集合通信アルゴリズムに関連する研究について述べる．

ネットワークトポロジーの推定方法については，白井らはノード間で小さいメッセージの送受信を行い，その往復時間 (RTT) の測定結果だけを用いて高速かつ低負荷，さらに手動の設定を一切必要せずにネットワークトポロジーを推定する方法を提案している [4]．これより，ノード間の遅延時間の情報と，ノードとスイッチの繋がり方に関するトポロジーが得られる．

集合通信アルゴリズムに関しては，Thilo Kielmann らによるMagPIe[1]は，クラスタ内のような遅延の影響が少ない通信にはbinomial treeを用い，クラスタ間のような遅延の大きい通信にはFlat treeを作ることで，遅延の大きなノード間の通信を極力減らし，クラスタ間のバンド幅を無駄遣いせず，また大きな遅延が全体の計算時間に影響を与えないようにしている．しかし，ネットワークトポロジーや遅延がアプリケーション実行前に既知なことが前提であり，アプリケーションを実行する度に状況により利用されるクラスタやノードが変化する場合に関しては考慮されていない．

松田らによる高バンド幅なネットワーク下での集合通信の研究 [2] でもクラスタ間の遅延はクラスタ内の遅延よりも非常に大きいことを仮定し，複数ノードがクラスタ間の通信に寄与することで高バンド幅なネットワークを有効利用する手法を提案している．これも同様にトポロジーが未知の場合はこのままでは使用することができない．

加えて，クラスタ内のノードはクラスタ外から見て全て対称に扱われているが，実際はクラスタ内でもノードがいくつかのスイッチを介した多段構造になっていたり，NATやFirewallによりクラスタの部分的なノードに対してしか直接通信できない場合があるなど，クラスタ内のノードの役割は対等ではないため，ネットワークの階層構造を考慮した通信制御を行う必要がある．

3 本研究の提案手法

提案する手法は，自動的に推定されたノードとノード間の遅延及び，ノードとスイッチがどのように接続されているかというネットワークの情報を利用し，得られたトポロジー情報に応じて既存の集合通信アルゴリズムを変形・組み合わせて様々なMPIの集合通信関数を記述する．実装されるMPI集合通信は次の11種類である．

- MPI_Bcast
- MPI_Scatter, MPI_Scatterv,
- MPI_Gather, MPI_Gatherv,
- MPI_Reduce
- MPI_Alltoall,
- MPI_Allgather, MPI_Allgatherv,
- MPI_AllReduce

基本原理はトポロジーと遅延の情報が与えられたとき，様々な集合通信それぞれについて集合通信の実行時間を短縮するとされる様々なアルゴリズムを，トポロジーの形に

An Efficient MPI Collective Communication Algorithm Using Inferred Network Information
by Shota Yoshitomi[†], Hideo Saito[‡], Kenjiro Taura[‡] and Takashi Chikayama[††] (The University of Tokyo)

表1 集合通信のアルゴリズム

Bcast	Scatter	Gather	Reduce	Allgather	Alltoall	Allreduce
Ring	Ring	Ring	Ring	Ring	Ring	Ring
Flat tree	Flat tree	Flat tree	Flat tree	Flat tree	Flat tree	Flat tree
binomial	RH*	RD†	RD	RD+RH	RD+RH	RD+RH
Van de Geiijn			Rabenseifner	Bruck	Bruck	BinaryBlocks

応じて変形，組み合わせて適用することによって，ネットワークの遅延とトポロジーという観点では理想的な手順で通信を行う集合通信のモデルを設計するというものである．

3.1 推定したトポロジーデータの利用

推定によって得たトポロジーの情報から，全ノードがそれぞれどのスイッチに接続されているかが分かる．そこで，あるスイッチに接続されているノードの集合を仮想的な一つのグループと見なし，全ノードをグループ分けする．通信は，グループ間での通信と，グループ内での通信の二通りを考慮する．

3.2 アルゴリズムの選択

MPI 集合通信の関数が呼び出された後，それぞれの集合通信において，得られたトポロジーのデータと，ノード間遅延に加え，通信するメッセージサイズ，通信を行うノード数の 4 つを考慮して，集合通信のアルゴリズムを選択する．具体的には，集合通信内で実行される全てのグループ内通信とグループ間通信のそれぞれについて，表 1 に示した中のアルゴリズムの実行時間の見積もりを行う．そして，その中で実行時間が最短に見積もられたアルゴリズムを選択し，各グループ内や各グループ間の通信に適用する．

4 実装・実験

4.1 実装

提案方法を MC-MPI[5] 上に実装した．具体的には，前章に示した集合通信を既存の MPI の規格と同等の関数として実装している．またそれに先立って，アプリケーション開始時に自動的に推定されたトポロジー及び各ノード間の遅延の情報が取得されるようになっている．

4.2 実験

実験環境としては InTrigger プラットフォーム [6] において，5 つのクラスタの 112 ノードを使用した．集合通信として，MPI_Bcast, MP_Allreduce を用い，本手法と既存の手法において通信するメッセージサイズを 16byte と 128KB の二通りに変化させ，一回分の集合通信の開始から終了までの経過時間について本手法の性能評価と他手法との比較を行った．

4.3 実験結果と考察

実験結果を表 2 に示す．まず Broadcast，Allreduce のどちらも，メッセージサイズが 16byte の場合 MagPIe との明瞭な差は見られなかった．これは，推定されたトポロジーにより選択したアルゴリズムが MagPIe とほぼ同等であると考えることができる．一方でメッセージサイズが 128KB と大きなとき，特に Allreduce において経過時間は MagPIe の 90%，MC-MPI の 40% と差が生じている．これより，集合通信の中でも全対全で行うものに関しては，メッセージサイズやノード数のパラメータが大きく経過時間に影響を与えるため，トポロジーによってアルゴリズムを切り替える本手法が特に有効であると言うことができる．

表2 1回の Broadcast 及び Allreduce の所要時間

	Broadcast		Allreduce	
	16 byte	128 KB	16 byte	128 KB
OURS	36.2 ms	1.60 s	36.1 ms	2.74 s
MagPIe	35.1 ms	1.65 s	35.2 ms	3.15 s
MC-MPI	33.5 ms	1.93 s	43.3 ms	6.88 s

5 おわりに

本稿では，自動取得したネットワークトポロジーのデータを利用する，MPI 集合通信アルゴリズムの改良方法を提案した．また実際に実装を行い，複数のクラスタにまたがる環境下において，集合通信の開始から終了までの所要時間を評価し，既存の手法との比較を行った．その結果，トポロジーが既知の元で使用されるアルゴリズムと同等もしくはそれ以上の高速化が可能であることを示した．

しかし，本稿の実験では NAT や Firewall が存在する環境では現在のトポロジーの推定手法は上手く動作しないため，内部が NAT 構造になっている一部のクラスタを利用できなかった．また事前のトポロジーの推定に 2~3 秒程要しており，通信時間だけでなく全実行時間を短縮するにはこの時間を削減することが必要不可欠である．今後はトポロジーの取得手法を強化し，推定速度及び精度を高めるとともに NAT,Firewall を検知し，それらが存在する環境下でも動作するようにアルゴリズムを拡張していく．

謝辞 本研究の一部は文部科学省科学研究費補助金特定領域研究「情報爆発に対応する新 IT 基盤研究プラットフォームの構築」の助成を得て行われた．

参考文献

[1] Thilo Kielmann, Rutger F.H.Hofman, Henri E.Bal, Aske Plaat, Raoul A.F.Bhoedjang. MagPie: MPI's Collective Communication Operations for Clusterd Wide Area Systems. PPoPP'99, May 1999.

[2] 松田元彦, 石川裕, 工藤知宏, 児玉祐悦, 高野了成: グリッド上のコレクティブ通信アルゴリズム. IPSJ SIG Notes, Vol.2006, No.87 pp. 257-262(2006).

[3] The Message Passing Interface(MPI). http://www-unix.mcs.anl.gov/mpi/ .

[4] 白井達也, 田浦健次朗, 近山隆. 高速なトポロジー推定 - ネットワークを考慮した並列計算の基盤として 2007.

[5] Hideo Saito, Kenjiro Taura.: Locality-aware Connection Management and Rank Assignment for Wide-area MPI, CCGrid 2007, pp. 249-256(2007).

[6] Intrigger プラットフォーム: https://www.logos.ic.i.u-tokyo.ac.jp/intrigger/

* Recursive Halving Algorithm

† Recursive Doubling Algorithm