1J-03

トーラス/メッシュ環境のプロセスランクマッピングによる通信性能評価

根本貴大† 熊谷洋佑† 藤井昭宏† 田中輝雄†

工学院大学†

1. はじめに

近年，京コンピュータのように，大規模並列計算機のインターコネクトにメッシュ/トーラス構造を採用するケースがある．メッシュ/トーラス構造は他の代表的な構造のファット・ツリー構造と比べ，通信プロセスのハードウェア上の位置関係により通信時間が変化する[1].

本研究では，メッシュ/トーラス構造のインターコネクトを持つ東京大学の FX10(oakleaf-fx)[2]において，プロセス通信のハードウェア上の距離による通信時間の変化を確認した．また，数値シミュレーションに用いる疎行列ベクトル積(SpMV)において，通信頻度の高いプロセス集合を同じノードに割り当てるランクマップにより，通信時間の削減を図った.

2. プロセスマッピング

2.1 ネットワークトポロジ

大規模並列計算機における各演算ユニットをノード，各通信網をリンクと呼ぶことにする．ノードとリンクにより構成される形をネットワークトポロジと呼ぶ．大規模並列計算機のネットワークトポロジの例にファット・ツリー，3 次元トーラスの 2 種がある．この 2 種の特徴を表 1 に示す.

図 1 に示す 3 次元トーラスは各ノードが 3 次元方向にリンクを持ち，各ノード間通信時に経由するリンク数にばらつきが発生する．ファット・ツリーは木構造であるため，各ノード間通信時に経由するリンク数は一定である．ノード間通信の経路上のリンク数をホップ数とする.

FX10 では，6 次元メッシュ/トーラスの Tofu インターコネクト[2]をネットワークトポロジに持つ．このトポロジにおいて，各通信ノード間の距離の違いによるノード間通信時間の影響を 3.1 節に示す.

表 1 ネットワーク構成の特徴

	3 次元トーラス	ファット・ツリー
概形	トーラス	木構造
ノード間距離	個別	均一

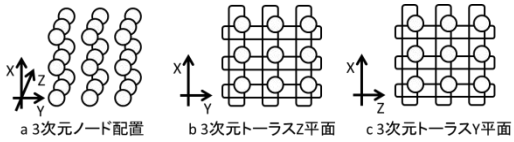

図 1 3 次元トーラスネットワークトポロジ

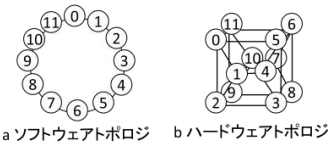

図 2 ノードの各トポロジ上の位置関係

2.2 ランクマップ

FX10 はハードウェアでは 6 次元メッシュ/トーラス，ソフトウェアでは 1・2・3 次元トーラスをネットワークトポロジに使用できる[2]．それぞれハードウェアトポロジ，ソフトウェアトポロジと呼ぶことにする．各ノードはユーザが指定したソフトウェアトポロジでのリンクを維持してハードウェアトポロジ上に配置される．図 2 は MPI プログラム実行時に 1 次元トーラスを指定したプロセスのハードウェアトポロジ上の位置の例である．プロセスのソフトウェアトポロジ上の位置をランクマップと言う[2]．3.3 節にて，FX10 上で SpMV のラックマップ改善を行い通信時間への効果を示す.

3. 実験

実験 1 では FX10 のソフトウェアトポロジ上の距離による通信時間の変化をみる.

実験 2 ではフラット MPI による SpMV 実行時にソフトウェアトポロジ上のノード毎に通信の多いプロセス同士を集約し，通信を改善したランクマッピングの通信時間への効果を示す．集約の手法

An Evaluation of Communication Performance using Process Mapping on Torus and Mesh Networks

Takahiro Nemoto†, Yosuke Kumagai†, Akihiro Fujii†, Teruo Tanaka†

† Kogakuin University

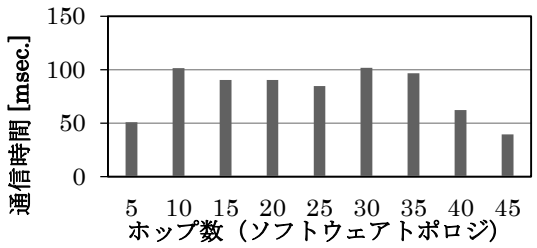

図 3　ソフトウェアトポロジ上のホップと通信時間

表 2　各トポロジ上でのホップ数と通信時間

ホップ数		通信時間 [msec.]
ソフトウェアトポロジ	ハードウェアトポロジ	
45	2	39.6
10	5	101.7

としてグラフ理論におけるグラフ分割のカットを最小化するライブラリ（ParMETIS[3]の PartKway）を利用した．SpMV 実行時の MPI 通信の通信テーブルにおける各プロセスと通信関係を頂点とエッジに対応させグラフ化する．このグラフに対し，カット後のグループ数が使用ノード数と同じになる条件でグラフ分割を行い，ノードへのプロセスの割り当てを決定する．この割り当てによりプロセス配置を改善する．計測の SpMV に用いた疎行列は 3 次元ポアソン方程式の 27 点差分問題，サイズ 200 の 3 乗である．すべての実験でソフトウェアトポロジに 1 次元トーラスを指定した．

3.1 実験 1

ここでは FX10 で 96 ノードに対しプロセスを 1 つずつ生成し，80MByte の倍精度型配列を 1 対 1 で通信しあう実験を行った．全プロセスがソフトウェアトポロジ上で同じホップ数の通信を同時に行い，開始と終了の同期を取った全体の通信時間を計測した．5~45 まで 5 刻み 9 種類のソフトウェアトポロジ上でのホップ数の結果を図 3 に示す．表 2 はソフトウェアトポロジ上でのホップ数 10 と 45 の通信時間とそれぞれのハードウェアトポロジ上でのホップ数を示す．ソフトウェアトポロジ上のホップ数の大小関係に反し，ホップ数 10 の通信時間はホップ数 45 の通信時間の 2.56 倍であった．

ハードウェアトポロジ上に配置された各プロセスの通信はソフトウェアトポロジにない最短経路のリンクを使う場合がある．図 2 における 0 と 5 のノード間通信が例である．このような経路が本実験でも使われたため，ハードウェアトポロジ上のホップ数が通信時間に影響したと考えられる．

表 3　最小カットによるランクマップ改善の効果

プロセス	マップ改善	実行時間 [μsec.]		プロセス間通信数	
		演算	通信 (削減率[%])	ノード内(b) (b/a[%])	全体 (a)
1536	無	104	194(-)	814(4%)	20718
	有	105	157(19%)	7466(36%)	
3072	無	67	181(-)	1706(4%)	43362
	有	62	165(9%)	27858(64%)	
6144	無	44	180(-)	4992(6%)	89504
	有	44	170(5%)	72146(81%)	

3.2 実験 2

ここではフラット MPI による SpMV の通信経路改善を行う．ブロック行分割の並列 SpMV の実行時間と，最小カットを用いたランクマップでソフトウェアトポロジ上でのプロセス配置の改善をした並列 SpMV の実行時間の比較を表 3 に示す．削減率はプロセス配置の改善前の通信時間から削減できた通信時間の割合である．

実行環境 1536 プロセス，96 ノードでは，ランクマップ改善の効果としてノード内通信の割合が 32%増加した．この改善により SpMV の通信時間が 19%削減された．

4. おわりに

本研究では，プロセス通信に対するハードウェアトポロジの影響と，SpMV でのソフトウェアトポロジのランクマッピング改善の効果を示した．

今回の実験 2 ではソフトウェアトポロジ上でのランクマップ改善であった．実験 1 からハードウェアトポロジのホップ数によるプロセス通信の影響を確認したので今後はハードウェアトポロジに着目したランクマップ改善の効果を検証したい．

参考文献

[1] YU, Hao, et al. Topology mapping for Blue Gene/L supercomputer. 2006 ACM/IEEE conference on Supercomputing. no.52 pp.116 (2006).

[2] 東京大学情報学基盤センタースーパーコンピューティング部門 -FX10 スーパーコンピュータシステム(oakleaf-fx), http://www.ipsj.or.jp/kenkyukai/genko.html .

[3] ParMETIS, http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview .