Vol.2013-HPC-140 No.17
2013/8/1

RDMA スケジューリングによる MPI 通信の高速化

畑中 正行[1]　堀 敦史[1]　石川 裕[2,1]

概要：本稿では，RDMA コマンド・スケジューラ RSched の京 ToFu インターコネクトでの性能妥当性を示す．RSched は MPI 集団または永続通信の呼出しによって与えられる複数の通信要求に対し，複数の RDMA エンジンとネットワーク・リンクをもつインターコネクト上で，最適な RDMA コマンド列を生成する．評価にあたっては，実際の京の気象・気候アプリケーションからの袖通信パターンを使い，隣接 4 方向及び 8 方向の複合通信で比較用のスケジューリング・アルゴリズムの中で最も高速であることを確認した．

1. はじめに

以前の我々の研究 [1] では，京コンピュータのインターコネクトである ToFu 上での RDMA 転送では MPI 実装が宛先ノードだけでなく，宛先ノードの受信 RDMA エンジンも指定することが必須であり，適切に指定しないと性能が出せないという問題を解決するために，RDMA コマンド・スケジューラ RSched を開発しその有効性を示した．

しかしながら，RSched で使われるスケジューリング・アルゴリズムが実際の ToFu インターコネクトの性能を十分に引き出しているかは明らかでなかった．実際，複数の RDMA エンジンやネットワーク・リンクをもつインターコネクトに対し，一つの通信パターンを構成する複雑な通信要求（RDMA コマンド列）をどのように割り当てれば通信時間を最小化できるかは自明ではない．

RDMA スケジューリングは，その場その場で個別の通信要求を処理するのではなく，集団通信または永続通信の呼出しによって獲得できる複数の通信要求から通信パターンを推測し，インターコネクトの機能や特性に合わせて，適切なコマンド列を生成するためのものである．

本稿では，袖通信を例に，基本の RDMA 転送レベルまで分解して，RSched のスケジューリング・アルゴリズムの妥当性を検証した．検証のために，実際に京コンピュータで使われている気象・気候アプリケーションの袖通信パターンに基づくベンチマーク・プログラムを使用した．この評価の結果，ステンシル計算の袖通信の範囲では RSched で使われているアルゴリズムが隣接 4 方向及び 8 方向の複合通信で比較用のスケジューリング・アルゴリズムの中で最も高速であることを確認し，また，オリジナルの Eager / Rendezvous プロトコルに比べ、約 2 倍の性能改善結果が得られた．

2. 関連研究

伝統的な集団通信の高速化については，インターコネクトの特性に応じてアルゴリズム及び実装に関し多くの研究がある [2], [3] ．しかしながら，隣接通信のような基本パターンでさえ，伝統的な集団通信から外れた通信パターンは，潜在的な性能改善の余地があるにも関わらず，取り残されてきた．

最新の MPI-3 仕様 [4] では，隣接通信の集団操作のために，`MPI_Neighbor_allgather` 及び `MPI_Neighbor_alltoall` プリミティブ（とそれらの亜種）が追加されているが，最近のスーパーコンピュータに採用されているインターコネクト上でそうした通信をどうすれば高速化できるのかの研究は始まったばかりである．実際にも，MPI-3 仕様をサポートするスーパーコンピュータ・システムは現時点では少なく，著者らは、多くのシステムがサポートする MPI-2 仕様上の MPI 永続通信 プリミティブを使って `MPI_Neighbor_*` プリミティブを代替する MPI の実装を京コンピュータ及び東京大学 FX 10 システム上で開発してきた [1].

Gropp と Thakur は MPI RMA 通信（1 方向通信; `MPI_Put`/`MPI_Get`）の性能を袖通信 (隣接 4/8 方向) ベンチマークを使って評価した [5]．RMA は RDMA 操作に対する自然な拡張であり，インターコネクトのハードウェアによっては `MPI_Isend` や `MPI_Irecv` よりも低オーバーヘッドになりうる．しかしながら，通信パターンを知る機

[1] 理化学研究所
RIKEN
[2] 東京大学
University of Tokyo

© 2013 Information Processing Society of Japan

会はほとんどなく，全体として最適化したり，資源の競合を抑制することはそれ自身では困難である．

Hoefler と Schneider は，MPI-3 仕様の隣接集団通信のためのいくつかの最適化手法を提案した [6] ．通信スケジューリングのために DAG ベースの汎用のアルゴリズムを提案しているが，これは送受信要求の中に依存関係があることを前提に導入されており，依存関係が生じた原因の一つは袖通信での斜め方向の通信を不要にするために [7] の二段階転送を採用したためだと考えられる．この二段階転送は，通信相手からの転送を待って自側の転送を開始しなければならないため，高度な通信オフロード機構がない限り，RDMA エンジンに対する通信要求の突放し実行は困難であり，結果として計算と通信のオーバラッピングを阻害する要因となるおそれが高い．

Kumar らは，永続通信で一度 RDMA コマンド・リストを作成したら，後の再利用のためにリストを RDMA キューに保存する最適化手法を提案した [8] ．これにより，関数及び RDMA アクセスのオーバヘッドを最小化できる．この手法は Blue Gene に過度に依存しているが，RDMA コマンド・リストとして一括して管理することの有用性を示している．

我々の知る限り，永続通信における複数の通信要求を，複数の RDMA エンジン上にスケジュールすることによる最適化の可能性指摘した論文は多くはない．本稿の焦点は，通信パターンとして一括して与えられた通信要求のセット全体の通信レイテンシを最小化するために，RDMA コマンド・スケジューラ RSched がとり得るアプローチを明確化することである．

3. 設計と実装

3.1 ToFu について

3.1.1 ToFu ICC

各計算ノードは ICC (InterConnect Controller) と呼ばれる，ToFu インターコネクトを実装した LSI に接続される．ICC は大きく，(1) ToFu Network Router (TNR), (2) ToFu Network Interface (TNI), (3) ToFu Barrier Interface (TBI) から構成される．ルータ部の TNR には 10 本の物理リンクが接続され，それぞれ 5 GB/s (× 双方向) のリンク速度をもつ．RDMA エンジン部の TNI は 4 基あり，全体で同時に 4 送信+4 受信可能である．

3.2 評価用スケジューリング・アルゴリズム

この節では，アルゴリズム評価で用いる 2 つの RDMA コマンドのスケジューリング・アルゴリズム RoundRobin 及び BLbased について説明する．

3.2.1 RoundRobin アルゴリズム

項 3.1.1 で述べたように，ToFu では計算ノードあたり 4 基の RDMA エンジンをもつ．RoundRobin アルゴリズムは，RDMA-put 転送要求をラウンド・ロビンで利用可能な RDMA エンジンに順番にキューする．図 1 は，6 つの RDMA コマンドを 4 基の RDMA エンジンにスケジュールする場合のこのアルゴリズムの動作を説明している．

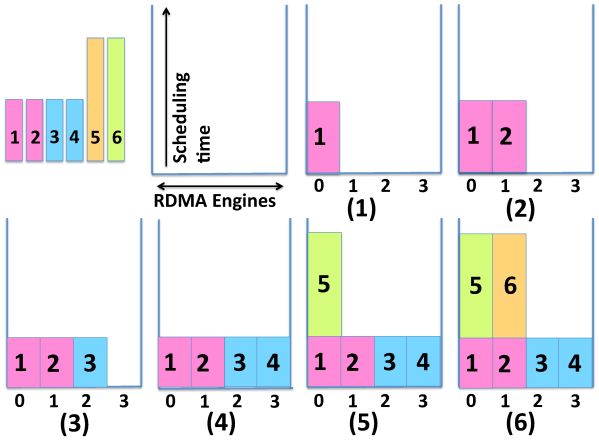

図 1 RoundRobin スケジューリング・アルゴリズム

長方形はスケジューリングしようとしている RDMA-put コマンド，長方形の色は宛先の隣接方向（専有するネットワークリンク），長方形の高さは通信時間（ネットワーク・リンクを専有する時間），長方形の幅は，専有する RDMA エンジンの数を示す．

この方式は RDMA-put コマンドの処理順序によって性能は様々である．図 1 のように，詰込みが甘く全体の通信時間が長くなったり，異なる RDMA エンジンを使って同じ時刻帯に同じ方向（同じネットワーク・リンクを専有）に転送するために競合が発生する可能性がある．

評価では，この RoundRobin アルゴリズムの 3 つのバリエーション RoundRobin(1), RoundRobin(2), 及び RoundRobin(4) を使用する．

RoundRobin(1) は転送時に 1 基の RDMA エンジンしか使わない．これにより，すべての通信要求が 1 つの RDMA エンジンのコマンド要求 FIFO キューにつながれ，通信に参加するすべてのプロセスがある時点で 1 方向かつ（恐らくは）同じ方向への転送しか行われない．よって，このアルゴリズムでは送信元を絞り込むことによりリンク競合を回避する目的のアルゴリズムである．

RoundRobin(2) 及び RoundRobin(4) はネットワーク・リンク競合を意図的に行うことが可能である．RoundRobin(2) は同時に 2 基の RDMA エンジンをしか使わないが，RoundRobin(4) は 4 基のエンジンすべてを使用する．

3.2.2 BLbased アルゴリズム

RSched は集団または永続通信の呼出しによって与えられる複数の通信要求を，複数の RDMA エンジンとネットワーク・リンクをもつインターコネクトに対し，最適

© 2013 Information Processing Society of Japan

な RDMA コマンド列を生成する RDMA コマンド・スケジューラである [1] .

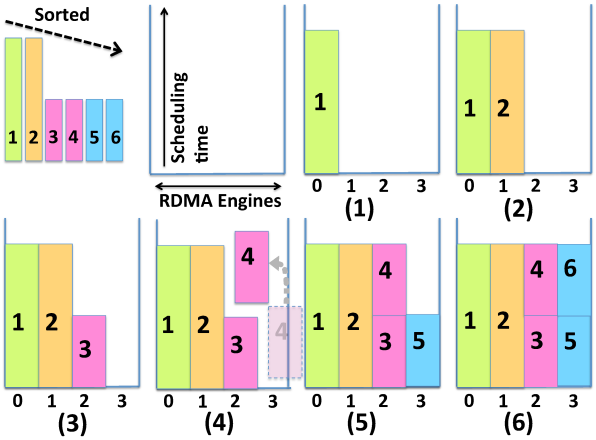

図 2 BLbased スケジューリング・アルゴリズム

RSched では，BLbased と呼ばれる Bottom-Left 発見的アルゴリズム [9] を使って，第一キーとして転送長（降順）、及び第二キーとして方向でソートされた RDMA-put 転送要求を，すべての要求の完了が最短になるよう， RDMA エンジンに詰込む．その際に，同じリンクを使う要求を同時にスケジュールしない，宛先での受信 RDMA エンジンの競合を避ける等の Bottom-Left に対する追加の制約を満足するよう，スケジュールする．その様子を図 2 に示す．基本的には最も底の空き地のうち最も左に埋めてゆく単純な方法であるが，図 2 (4) のように，リンク競合が起きうる同じ方向（図中は同じ色）の割り当てを抑制する追加の制約をもつ．

評価では，この BLbased アルゴリズムの 3 つのバリエーション BLbased(1), BLbased(2), 及び BLbased(4) を使用する．これらはそれぞれ，斜め方向（北西、北東、南西、南東）の通信時に，同時に使用する RDMA エンジンを 1 基，2 基，4 基に制限する．東西南北方向には制限しない．

4. 評価

4.1 評価用袖通信パターン

袖通信は，ステンシル・プログラムで主要な通信パターンであるが，多くのバリエーションが存在する．例えば，気象・気候アプリケーション SCALE-LES3 [10] では袖通信は主要な通信パターンであり，問題規模に応じてグリッドサイズを使い分けている．

表 1 は SCALE-LES3 での水平方向のグリッドを 16×16 に固定し，鉛直方向のグリッドを 60 ～ 872 まで変化させた場合の、袖通信の通信パターンを示している．袖領域 が 2 のため，東西方向の転送長が $(2+k+2) \times 2 \times i \times 8$ [B], 南北方向が $(2+k+2) \times j \times 8$, 斜め方向が $(2+k+2) \times 2 \times 8$ である．この通信パターンは 14 個の RDMA-put コマンドから構成される．

表 1 SCALE-LES3 性能評価用 Grid Size での転送サイズ

グリッドサイズ	東西 2 方向 ×1 送信	南北 2 方向 ×2 送信	斜め 4 方向 ×2 送信
$k \times j \times i$	転送長 [B]	転送長 [B]	転送長 [B]
60 × 16 × 16	16,384	8,192	1,024
70 × 16 × 16	18,944	9,472	1,184
80 × 16 × 16	21,504	10,752	1,344
109 × 16 × 16	28,928	14,464	1,808
218 × 16 × 16	56,832	28,416	3,552
327 × 16 × 16	84,736	42,368	5,296
436 × 16 × 16	112,640	56,320	7,040
545 × 16 × 16	140,544	70,272	8,784
654 × 16 × 16	168,448	84,224	10,528
763 × 16 × 16	196,352	98,176	12,272
872 × 16 × 16	224,256	112,128	14,016

性能測定は，48 プロセスの二次元循環プロセス・トポロジー上で，41,000 回の袖領域の交換を行なう．各交換では `MPI_Startall` と `MPI_Waitall` が呼出される．動作は擬似集団通信モードで，`MPI_Startall` の最初と，`MPI_Waitall` の最後で，内部的に `MPI_Barrier` による通信同期が実行され，バリア同期とバリア同期の間で，RoundRobin または BLbased スケジューリング・アルゴリズムで生成した RDMA-put コマンド列を実行し，要求完了キューに対し発行されたすべての要求の完了を待ち合わせる．

以降の節の通信性能のグラフ中の X 軸は，SCALE-LES3 における鉛直方向のグリッド数で，通信サイズではない．Y 軸は 41,000 回の交換での 1 回あたりの平均通信時間であり，平均の `MPI_Startall` と `MPI_Waitall` にかかった時間を秒で表している．但し，その時間にはスケジューリングに要した時間は含まれない．

4.2 東西方向の通信

表 1 の東西方向の転送要求を，RoundRobin(1), (2), 及び (4) と BLbased(1), (2), 及び (4) の計 6 種類のスケジューリング・アルゴリズムで評価する．図 3 は，各アルゴリズムでの当該転送要求のスケジューリング結果を示す．2 転送要求しかないことと，BLbased アルゴリズムは東西方向には同じ振舞いをするため，RoundRobin(1) を除き，結果はすべて同じである．

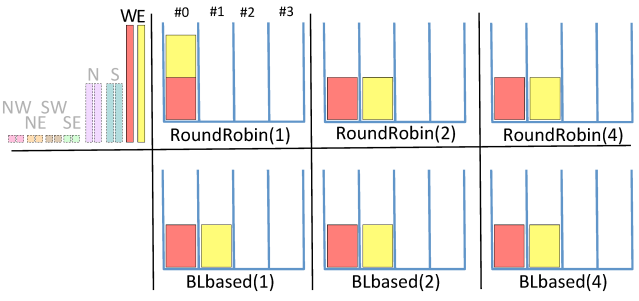

図 3 SCALE-LES3 東西方向通信スケジューリング結果

© 2013 Information Processing Society of Japan

節 4.1 で説明した測定環境の元での東西方向の単体性能測定結果を図 4 に示す。

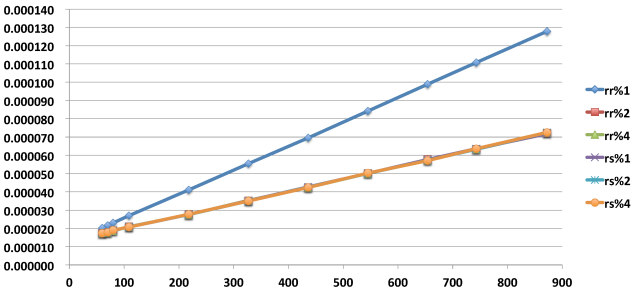

図 4 東西方向通信性能
ここで rr%1 は RoundRobin(1), rr%2 は RoundRobin(2), rr%4 は RoundRobin(4), rs%1 は BLbased(1), rs%2 は BLbased(2), rs%4 は BLbased(4) の短縮名である

図 4 のスケジューリングの結果どおり，RoundRobin(1)（図中の *rr%1*）を除き，他のスケジューリング・アルゴリズムすべては同じ性能であった．転送サイズに比例しており，RoundRobin(1) は他に比べ，オーバヘッド分を除いた正味の転送時間で約 2 倍の時間がかかっており，RDMA エンジン 1 基で 2 転送要求をシリアライズしたスケジュール意図どおりの結果が得られた．しかしながら，2 つの要求を 2 基の RDMA エンジンで分散した測定値を見ると，横軸 872 のときの通信時間は 72 us であり，図から約 18 us のオーバーヘッド分を差し引いた 54 us が横軸 60 からの純粋な転送時間の増分とすると，$(224,256-16,384)[B] \div 54[us] \approx 3.8[GB/s]$ で，実効効率を加味しても，ToFu のリンク速度 5 [GB/s] を下回るため，今後調査が必要である．

4.3 南北方向の通信

表 1 の南北方向の転送要求を，前項と同じ方法で評価した．図 5 は，各アルゴリズムでの当該転送要求のスケジューリング結果を示す．BLbased アルゴリズムは南北方向には同じ振舞いをし，同じ方向（図中では同じ色）の複数の要求を同時刻帯にスケジュールしないアルゴリズムである．RoundRobin アルゴリズムは，スケジューリング結果が入力の順序に左右されるが，今回は比較のために意図的に異なる RDMA エンジンから同じ方向の転送要求が同時にスケジュールされるよう入力を選んだ．

南北方向の単体性能測定結果を図 6 に示す。測定条件は東西方向の測定と同じである．

この測定では，リンクが競合するように，同時刻帯に同じ方向の意図的な転送要求列を処理した RoundRobin(4) が最も速かった．また，RoundRobin(2) が RoundRobin(4) のちょうど 2 倍遅く，RoundRobin(1) が BLbased の 2 倍遅かった．横軸 872 のときの RoundRobin(4) の通信時間

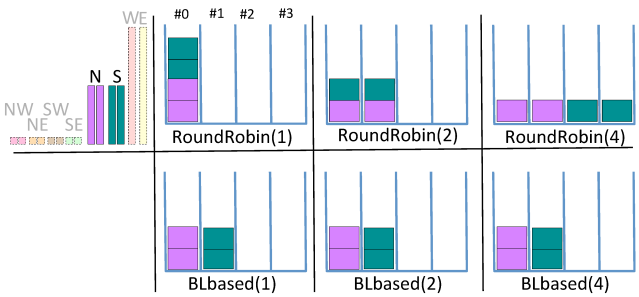

図 5 南北方向通信スケジューリング結果

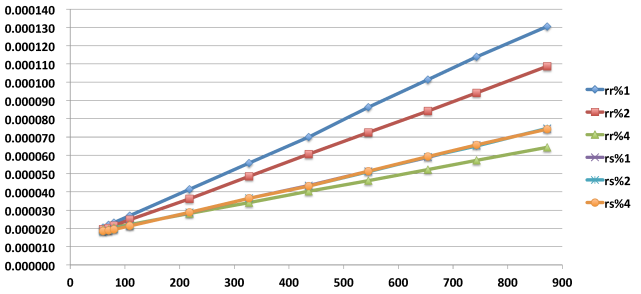

図 6 南北方向通信性能

は 64 us であり，図から約 18 us のオーバーヘッド分を差し引いた 46 us が $112,128[B] \div 46[us] \approx 2.4[GB/s]$ である．2 つの要求が同じリンクを共有して（競合して）いるとすれば，リンク自体は 2 倍の 4.8 [GB/s] であり，この値は理論値の 5 [GB/s] に対して妥当である．それに対し，BLbased は $(112,128 * 2)[B] \div (74-18)[us] \approx 4.0[GB/s]$ しか出ていないため，東西方向同様調査が必要である．

4.4 東西南北方向の通信

表 1 の東西及び南北の 4 方向の転送要求を同じ方法で評価した．図 7 は，各アルゴリズムでの当該転送要求のスケジューリング結果を示す．BLbased はこの通信パターンでは詰込み処理により，RoundRobin より全体の通信時間をスケジューリング上は最小化できる．

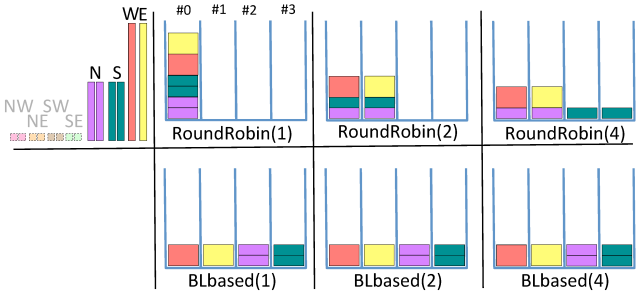

図 7 東西南北方向通信スケジューリング結果

東西南北 4 方向の単体性能測定結果を図 8 に示す。測定条件はこれまでの測定と同じである．BLbased(4) は横軸 872 の場合，図 4 より東西方向 72 us，図 6 より南北方向が 74 us であったが，東西南北 4 方向では 86 us になり，

$(112, 128*2) \div (86-23)[us] \approx 3.5[GB/s]$ になった．86 us まで低下したのは，同時 4 方向に転送したため，何らかのボトルネックに達したためと考えられる．しかしながら，南北方向の RoundRobin(2) の 108 us（872 グリッド）よりは短く，4 方向の転送ではリンク競合起こさない方が有利である．また，既存の Eager / Rendezvous プロトコル（図 8 の *orig*）よりも約 2 倍高速である．

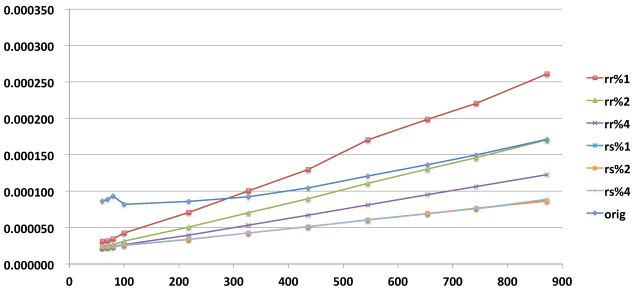

図 8 東西南北方向通信性能

4.5 斜め方向の通信

表 1 の斜め方向の転送要求を同じ方法で評価した．図 9 は，各アルゴリズムでの当該転送要求のスケジューリング結果を示す．BLbased は斜め方向の場合に，使用する RDMA エンジンを各アルゴリズムで変更する．RoundRobin と BLbased の違いは，BLbased が同時刻帯に同じ方向の転送要求をスケジュールしないようにすることである．

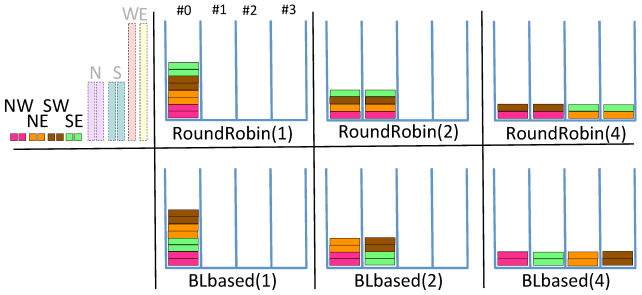

図 9 斜め方向通信スケジューリング結果

斜め方向の単体性能測定結果を図 6 に示す。測定条件はこれまでの測定と同じである．

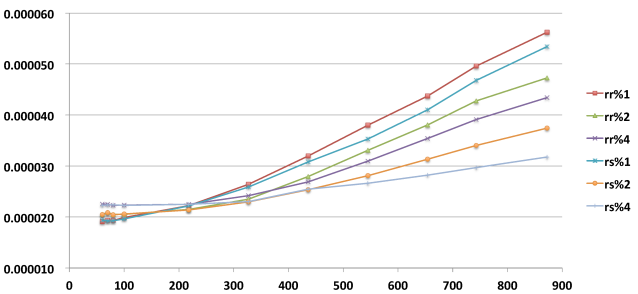

図 10 斜め方向通信性能

斜め方向の場合，ToFu の場合，物理トポロジーとランクのマッピングに依存するが同時 2 方向なら競合なしに転送できる可能性があるが，同時 4 方向に転送する場合，高い確率で競合が発生する．測定結果は，BLbased(4) が最も速く、以下 BLbased(2), RoundRobin(4), 及び RoundRobin(2) の順だった．元々リンク競合のある RoundRobin は遅かった．

4.6 全方向の通信

表 1 の全方位の 8 方向の転送要求を同じ方法で評価した．図 11 は，各アルゴリズムでの当該転送要求のスケジューリング結果を示す．斜め方向の転送が，BLbased では転送要求のメッセージ長によるソートにより，転送が後になるのに対し，RoundRobin は要求順序どおりに最初にスケジュールされる点以外は，これまでと同じである．

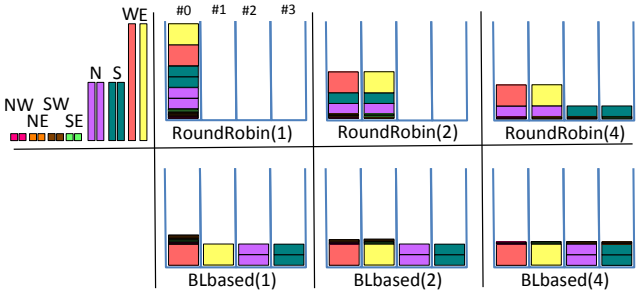

図 11 全方向通信スケジューリング結果

全 8 方向の性能測定結果を図 12 に示す。基本的に，東西南北方向と同様に傾向であり，南北方向の 1/8 のサイズの斜め転送が上乗せされたと言える。BLbased (1), (2), 及び (4) の 3 種類に関しては，転送サイズが大きい部分では BLbased(4) で十分であると判断できる．また，既存の Eager / Rendezvous プロトコル（図 12 の *orig*）よりも BLbased(4) は約 2 倍高速である．

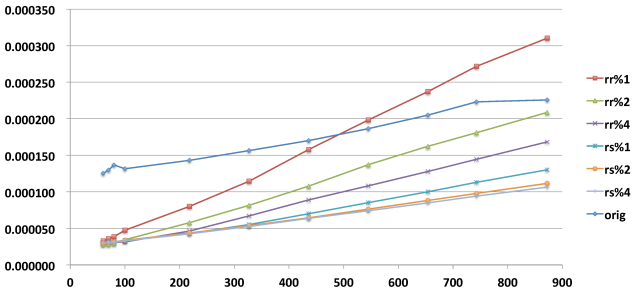

図 12 全方向通信性能

5. 議論

図 13 は，斜め方向の通信の図 10 を拡大したものである．斜め方向の通信は南北方向に対し 1/8 で，鉛直グリッド 60 の場合，転送サイズは 1 KiB になる．ToFu の場合，

© 2013 Information Processing Society of Japan

各 RDMA エンジンに対し一つの要求キューがあり，複数の RDMA を使用する場合，各キューに書き込む時間差が生じる．図 13 のように，転送サイズが小さい場合は，RDMA エンジンへのアクセス・オーバーヘッドが無視できず，使用する RDMA エンジンの数によって全体の通信時間が変化すると考えられる．

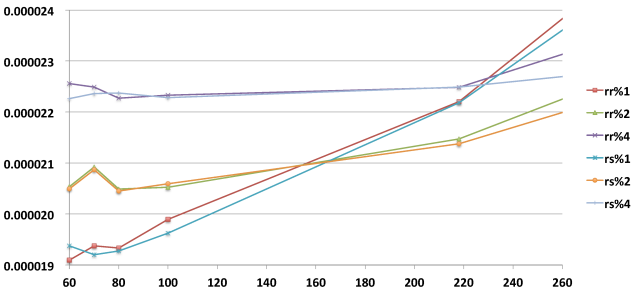

図 13 斜め方向通信性能（拡大図）

よって，汎用な RDMA コマンド・スケジューリングを考慮する際，RDMA エンジンへのアクセス・オーバーヘッドを加味した，スケジューリング・アルゴリズムが必要になる．例えば，図 14 のような単純な仕組も考えられる．つまり，スケジューリングのための詰込みの初期値として，事前にオーバヘッド分（図 14 から約 1 us 程度）を擬似の要求として割り当てておくことで，小サイズに対応したスケジュールを実現できる．しかしながら，より正確な通信時間の予測が必要になる上，現在 BLbased で使用しているソート処理はメッセージ長の降順であり，小さな隙間を最初に埋めるのには適していない．小サイズのスケジューリング・アルゴリズムは今後の検討課題である．

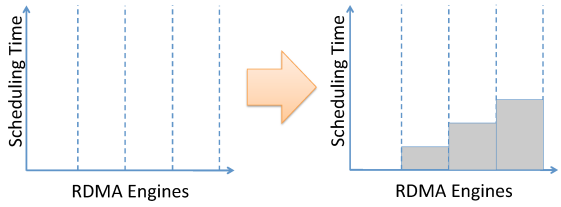

図 14 RDMA オーバーヘッド対応スケジューリング案

6. 結論

ステンシル計算の袖通信の範囲では BLbased は隣接 4 方向及び 8 方向の複合通信で比較用のスケジューリング・アルゴリズムの中で最も高速であることを確認した．また，オリジナルの Eager / Rendezvous プロトコルに比べ、約 2 倍の性能が得られた．しかしながら，競合をさけるより，競合した場合でも却って性能がよいケースもあり，また小サイズでの通信で過渡的な振舞いについても今後調査・検討してゆく予定である．

また今回検討した，ステンシル計算では隣接通信が基本であり，ToFu のルーティングの詳細な検討は不要であった．ルーティングが意味をもつ長距離の通信が発生するようなアプリケーションへの対応は今後の検討課題である．より複雑な通信パターンに対応するため，アルゴリズムに改良を加えてゆく．

参考文献

[1] Hatanaka, M., Hori, A. and Ishikawa, Y.: Optimization of MPI Persistent Communication, *EuroMPI* (2013). (to appear).
[2] Adachi, T., Shida, N., Miura, K., Sumimoto, S., Uno, A., Kurokawa, M., Shoji, F. and Yokokawa, M.: The design of ultra scalable MPI collective communication on the K computer, *Comput. Sci.*, Vol. 28, No. 2-3, pp. 147–155 (online), DOI: 10.1007/s00450-012-0211-7 (2013).
[3] 松本　幸, 安達知也, 住元真司, 南里豪志, 曽我武史, 宇野篤也, 黒川原佳, 庄司文由, 横川三津夫：MPI_Allreduce の「京」上での実装と評価，情報処理学会論文誌コンピューティングシステム（ACS），Vol. 5, No. 5, pp. 152–162 (2012).
[4] Message Passing Interface Forum: MPI: A Message-Passing Interface Standard, Version 3.0, Technical report (2012).
[5] Gropp, W. D. and Thakur, R.: Revealing the performance of MPI RMA Implementations, *EuroPVM/MPI* (2007).
[6] Hoefler, T. and Schneider, T.: Optimization principles for collective neighborhood communications, *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '12, Los Alamitos, CA, USA, IEEE Computer Society Press, pp. 98:1–98:10 (2012).
[7] Palmer, B. and Nieplocha, J.: Efficient Algorithms for Ghost Cell Updates on Two Classes of MPP Architectures, *International Conference on Parallel and Distributed Computing Systems, PDCS 2002, November 4-6, 2002, Cambridge, USA*, IASTED/ACTA Press, pp. 192–197 (2002).
[8] Kumar, S., Heidelberger, P., Chen, D. and Hines, M.: Optimization of applications with non-blocking neighborhood collectives via multisends on the Blue Gene/P supercomputer, *24th IEEE International Symposium on Parallel and Distributed Processing, IPDPS 2019*, pp. 1–11 (online), DOI: 10.1109/IPDPS.2010.5470407 (2010).
[9] Baker, B., Coffman Jr., E. and Rivest, R.: Orthogonal Packings in Two Dimensions, *SIAM Journal on Computing*, Vol. 9, No. 4, pp. 846–855 (1980).
[10] Sato, Y., Yashiro, H., Nishizawa, S., Miyamoto, Y. and Tomita, H.: Development of SCALE-LES3 model and numerical simulations of shallow clouds by the model, *The Second International Workshop on Nonhydrostatic Numerical Models*, Sendai, Japan, pp. 209–226 (2012).

© 2013 Information Processing Society of Japan