Vol.2011-HPC-130 No.50
2011/7/28

パケットペーシングを用いた集団通信アルゴリズムのシミュレーション評価

柴村英智[†1]　薄田竜太郎[†1]　三輪英樹[†2]
三吉郁夫[†2]　井上弘士[†3]

本稿では，パケットペーシングを適用した各種の集団通信アルゴリズムの性能評価について述べる．2 次元トーラス網ならびに 3 次元トーラス網を対象に，パケットペーシングを適用した集団通信の実行性能について，インターコネクトシミュレータ NSIM を用いて定量的な評価を行った．その結果，一般的な集団通信にパケットペーシングを適用した場合の有効性を確認するとともに，アルゴリズム，トポロジ，ノード数，メッセージサイズに応じて，集団通信を最適に実行するパケットの送出間隔は異なることが分かった．また，通信ホップ数に応じたパケット間ギャップを設定することで，アルゴリズムによっては大幅な実行時間の短縮が達成されることを確認した．

Simulation Evaluation of Collective Communication Using Packet Pacing

Hidetomo Shibamura,[†1] Ryutaro Susukita,[†1]
Hideki Miwa,[†2] Ikuo Miyoshi[†2] and Koji Inoue [†3]

This paper presents quantitative performance evaluations of collective communication algorithms on 2D and 3D torus networks to investigate the effectiveness of packet pacing techniques. The execution performances of traditional algorithms with packet pacing are evaluated by using an interconnection network simulator NSIM. As the result, we confirmed the improvement of the execution performances of collective communications by packet pacing. Furthermore, the packet pacing according to communication hops works in some collective communications.

1. はじめに

並列処理における集団通信は，並列プログラムを実行する全プロセスが参加し，1 対全あるいは全対全のプロセス間でデータの分配・収集を定型的に行う．この集団通信は多くの科学技術計算には欠かせないものとなっており，効率良く動作させるためには，並列計算機のインターコネクト（相互結合網）の構成に配慮し，通信データが特定のネットワーク部分に極力集中しない通信アルゴリズムを用いることが重要である．しかし，これまでに様々な工夫に基づいた集団通信アルゴリズムが提案されているが，計算機システムやプログラムの特性によって実行性能が異なるため，それらの優越を決定的に定めることは難しい．そこで，近年の並列処理ライブラリでは，通信データのサイズやシステムの運用状況などによって適切なアルゴリズムを選択する実装が採られることが多い[1),2)]．

一方，インターネット環境を基盤とするマルチメディア配信やグリッドコンピューティングの分野では，複数のバースト的なトラフィックによる性能の低下を解決するために，パケットの送出間隔を制御しトラフィックを平滑化する，ペーシングに関する研究や評価実験が盛んに行われている[3),4)]．このようなペーシング技術は，コモディティネットワークにおいて非常に高いネットワーク利用効率を達成しているが，並列計算機のインターコネクトへの応用はほとんどない．そこで我々は，将来のスーパーコンピューティングに向けた要素技術として，インターコネクトへのパケットペーシングの適用実験を行っている．過去の研究では，ルータに 4 つの NIC（通信コントローラ）を搭載した 2 次元トーラス網を対象とし，同時送受信機構を活用する全体全通信アルゴリズムにパケットペーシングを適用しその有効性を確認した[5)]．

本研究では，一般的な集団通信アルゴリズムに対してパケットペーシングが与える影響を調査することを目的とする．通常のメッセージ通信と比較して高トラフィックとなる集団通信にパケットペーシングを適用し，通信混雑や輻輳を抑制することで全体の実行性能の向上をねらう．具体的には，単一 NIC をルータに持つ一般的なトーラス網を対象に，既存の集団通信アルゴリズムに対して様々なパケットの送出間隔でペーシングを行う．そして，ア

†1 財団法人九州先端科学技術研究所
Institute of Systems, Information Technologies and Nanotechnologies
†2 富士通株式会社
Fujitsu Ltd.
†3 九州大学
Kyushu University

ルゴリズムごとにみられる通信性能の変化をシミュレーションによって定量的に調査する．また，ホップ数に応じたペーシング制御を行い，その有効性を確認する．なお，近年の大規模 HPC システムはトーラス系のインターコネクトを採用するものが多く，運用時のプロセス配置なども考慮し，2 次元ならびに 3 次元トーラス網を評価の対象する．

以下，2 章では，本研究で評価対象とする集団通信アルゴリズムについて述べる．3 章では，評価に用いるインターコネクトシミュレータ NSIM について概説する．4 章では，集団通信アルゴリズムの比較やパケットペーシングによる評価実験の結果について議論する．そして，5 章でまとめと今後の課題について述べる．

2. 集団通信アルゴリズム

本研究では，集団通信の中でも多くの並列処理ライブラリで利用されている，Alltoall, Allgather，Allreduce といった全てのプロセスが参加する多対多通信について焦点を絞る．これらの集団通信は本質的に自プロセスのデータを他のすべてのプロセス群に送信することである．したがって，ネットワーク上で通信メッセージの衝突が無い理想的なネットワーク環境下では一様の通信時間（理想実行時間）となる．そこで，集団通信の種類には捕らわれず，集団通信を実現するアルゴリズムの種類に着目する．

評価対象とする集団通信アルゴリズムには，様々な並列処理ライブラリにおいて広く利用されている以下の 5 つを用いる．なお，本稿で用いる略称を括弧内に示す．

- Pairwise exchange（pwx）
- Ring（ring）
- Simple spread（ssprd）※拡散型
- Simple spread（ssprd2）※集約型
- Bruck（bruck）

Pairwise exchange 常に 2 つのプロセスが通信ペアを組んで通信を行い，実行ステップごとにペアを変えていく方式である．すべてのプロセスとペアを組むため，プロセスが p 個の場合には $p-1$ ステップを要する．i ステップ目において送信ノード j は，受信ノード $j \oplus i$（排他的論理和）とペアを組む．したがって，すべてのプロセスと通信を行うために，プロセスの総数は 2 べきでなければならない．

Ring 前述の pairwaise exchange と同様に実行ステップごとにペアを変え，すべてのプロセスとペアを組む．よって，p プロセスの場合には $p-1$ ステップを要する．i ステップ目では，送信ノード j は，受信ノード $(j+i) \mod p$ とペアを組む．したがって，ネットワーク全体で環状（ring）の通信路が形成され，一つのプロセスにデータが集中しない．また，送信が終わるとノード $(j-i) \mod p$ からのメッセージを待つため，全体でロックステップ実行が行われ，同期をとりながら通信が行われる．

Simple spread（拡散型） 各プロセスが他のすべてのプロセスに対して送信を順次行う．一般的には，送信ノード i から，$i+1, i+2, i+3, \ldots, (i+p-1) \mod p$ の順にメッセージを送信する．全てのプロセスがいっせいに送信を開始し，また，一つのプロセスに対して同時に複数のメッセージが送られるため，メッセージの衝突が発生しやすくネットワークに高い負荷をかける．

Simple spread（集約型） 前述の拡散型と若干異なる通信順序を採っており，各プロセスは順に他のすべてのプロセスからデータを受信するものである．すなわち，拡散型ではステップごとに一つのプロセスから複数のプロセスに対して送信が行われるのに対して，集約型では一つのプロセスに対していっせいに送信される．

Bruck Bruck アルゴリズム[6] は，複数の受信ノードに対するデータをまとめて送信することで，実行ステップの減少を図ったものである．k ステップ目で，送信ノード i は受信ノード $(i+2^k) \mod p$ にメッセージを送り，$(i-2^k) \mod p$ からメッセージを受け取る．したがって，全体では $\lceil \lg p \rceil$ ステップとなるが，プロセス数が大きくなるにつれて一回の送受信データ量が増加するため，全体での総通信量は大幅に増加する．

以上の集団通信にパケットペーシングを適用した場合の評価指標として，各アルゴリズムによるプログラムの実行時間だけでなく，理想的なネットワークにおける多対多の集団通信に要する時間，すなわち全体全通信の理想実行時間も用いる．そこで，まず，$x \times y$ の 2 次元トーラス網，ならびに $x \times y \times z$ の 3 次元トーラス網における全対全通信の理想実行時間 (T_{ideal}) を求める．これは，インターコネクトのバイセクションリンクを通過する総データ量 Q とバイセクションバンド幅 BW_{bisect} から以下の式で求められる．

$$T_{ideal} = \frac{Q}{BW_{bisect}}$$

ここで，1 プロセスに送信するメッセージサイズを L_{mesg}，単方向のリンクバンド幅を BW_{link} とすると，2 次元トーラス網における Q と BW_{bisect} は，それぞれ，

$$\begin{aligned} Q &= \left(\left\lfloor \frac{x}{2} \right\rfloor y \left\lceil \frac{x}{2} \right\rceil y + \left\lceil \frac{x}{2} \right\rceil y \left\lfloor \frac{x}{2} \right\rfloor y \right) L_{mesg} \\ &= 2y^2 \left\lfloor \frac{x}{2} \right\rfloor \left\lceil \frac{x}{2} \right\rceil L_{mesg}, \\ BW_{bisect} &= 4y\, BW_{link} \end{aligned}$$

となり，理想実行時間は次式で求められる．

$$T_{ideal_{2DT}} = \frac{y}{2}\left\lfloor\frac{x}{2}\right\rfloor\left\lceil\frac{x}{2}\right\rceil\frac{L_{mesg}}{BW_{link}}$$

また，3 次元トーラス網での理想実行時間は同様に，

$$\begin{aligned} Q &= \left(\left\lfloor\frac{x}{2}\right\rfloor yz\left\lceil\frac{x}{2}\right\rceil yz + \left\lceil\frac{x}{2}\right\rceil yz\left\lfloor\frac{x}{2}\right\rfloor yz\right)L_{mesg} \\ &= 2y^2z^2\left\lfloor\frac{x}{2}\right\rfloor\left\lceil\frac{x}{2}\right\rceil L_{mesg}, \end{aligned}$$

$$B_{bisect} = 4yz\,B_{link}$$

となり，理想実行時間は次式となる．

$$T_{ideal_{3DT}} = \frac{yz}{2}\left\lfloor\frac{x}{2}\right\rfloor\left\lceil\frac{x}{2}\right\rceil\frac{L_{mesg}}{B_{link}}$$

なお，実際の通信時には，パケットヘッダの転送オーバヘッドも考慮しなければならない．たとえば，リンクバンド幅が 4GB/sec.，パケットサイズが 2KiB，パケットヘッダサイズが 32KiB の場合に，512KiB のメッセージサイズで集団通信を行うと，1 つのプロセスに送信するメッセージサイズは

$$\begin{aligned} L_{mesg} &= \text{メッセージサイズ} + \text{パケット数} \times \text{パケットヘッダ長} \\ &= 512\text{KiB} + \left\lceil\frac{512\text{KiB}}{2\text{KiB} - 32\text{B}}\right\rceil \times 32\text{B} \end{aligned}$$

となるため，理想実行時間は，8×8 の 2 次元トーラス網では 8.522ms，$4 \times 4 \times 4$ の 3 次元トーラス網では 4.261ms となる．本稿で扱う集団通信のメッセージサイズごとの理想実行時間を**表 1** にまとめる．

表 1 2 次元/3 次元トーラス網における集団通信の理想実行時間（単位：msec.）

	32KiB	64KiB	128KiB	256KiB	512KiB
2DT-8x8	0.532	1.065	2.130	4.261	8.522
2DT-16x8	2.130	4.260	8.522	17.043	34.087
2DT-16x16	4.260	8.520	17.043	34.087	68.174
2DT-32x16	17.039	34.079	68.174	136.348	272.695
2DT-32x32	34.079	68.157	136.348	272.695	545.391
3DT-4x4x4	0.266	0.532	1.065	2.130	4.261
3DT-8x4x4	1.065	2.130	4.261	8.522	17.043
3DT-8x8x4	2.130	4.260	8.522	17.043	34.087
3DT-8x8x8	4.260	8.520	17.043	34.087	68.174
3DT-16x8x8	17.039	34.079	68.174	136.348	272.695

※パケットサイズ：2KiB，パケットヘッダサイズ：32B
リンクバンド幅：4GB/sec.

3. NSIM：インターコネクトシミュレータ

我々は NSIM と呼ぶインターコネクトシミュレータを開発している[7]．NSIM は，次世代インターコネクトの性能評価を目的とし，設計・開発現場での実践的な利用を念頭に，現実的な時間内でシミュレーションを完了することを目指している．

これまでに開発されてきたシミュレータの多くは，具体的なアプリケーションの通信パターンを入力とするため，実際の並列計算機から取得した通信ログ，もしくは人工的に生成した通信パターンを利用することが多い．この手法は，実際に発生した通信事象に基づいた忠実なシミュレーションができる．一方，インターコネクトのトポロジや通信性能の差異によってプログラムの挙動が動的に変化する場合は，正確なシミュレーションを行うことが難しい．また，大規模インターコネクトのシミュレーション時には，相当する通信ログの取得限界問題が障壁となる[8),9)]．

そこで我々は，アプリケーションの振る舞いに忠実に従ったシミュレーションを行うために，NSIM では「MGEN プログラム」と呼ぶ MPI 互換のインタフェース（MGEN API）で記述されたプログラムを入力とし，実際にそのプログラムを駆動することによってオンデマンドで通信パターンを生成している．一般的な MPI 通信ライブラリと同様に，利用者は MGEN ライブラリを用いてプログラムを記述し実行することで，所望のアプリケーションとインターコネクトのシミュレーションを行うことができる．また，インターコネクトに関する詳細仕様を設定ファイルで与えるため，実機のみならず新規のインターコネクトの性能予測ツールとしても利用できる．

MGEN ライブラリは，基本的な送受信通信プリミティブをはじめ，非同期通信や基本的な集団通信を提供しており，ランデブー通信，ゼロコピー通信といったオプションもサポートしている．また，将来の技術研究に向けた NSIM 専用の高レベル API を用意しており，複数 NIC による同時送受信や，パケットペーシングのためのパケット間ギャップの設定ができる．さらに，実システムに即した評価を行うために，OS ジッタやロードインバランスを踏まえたシミュレーションが可能である．なお，シミュレーションでは通信による実際のデータの授受は行われない．NSIM はインターコネクトの性能評価を目的としており，MGEN ライブラリは NSIM への通信事象を生成するための枠組みゆえである．

NSIM は，MGEN プログラムの予測実行時間（MGEN 実行時間），リンクバンド幅，リ

表 2 評価対象システムの仕様一覧

パラメータ	設定値	パラメータ	設定値
ルーティング方式	次元順+dateline	MTU	2KiB
パケット転送方式	VCT	パケット長	32B〜2KiB(MTU)
フロー制御方式	クレジットベース	パケットヘッダ長	32B
ノード間リンクバンド幅	4GB/s（単方向）	仮想チャネル数	2
ルーティング計算時間 (RC)	4ns	仮想チャネルバッファ	8KiB (MTU×4)
仮想チャネル設定時間 (VA)	4ns	フリット長	16B
スイッチ設定時間 (SA)	4ns	NIC 数	1
フリット転送時間 (ST)	4ns	DMA 転送レート	16GB/s
スイッチ遅延時間	78ns	メモリバンド幅	16GB/s
ケーブル遅延時間	10ns	MPI 関数処理オーバヘッド	200ns

ンクビジー率，バッファ利用率，総転送データ量といった統計情報を出力し，輻輳解析に役立つ支援機構も備えている．また，データ解析のための処理スクリプトやツールを取り揃えている．その一つに，NSIM が出力した通信ログを，MPICH2[2] の MPE ライブラリに含まれる Jumpshot の入力ファイル形式に変換するツールがあり，MGEN プログラム実行時の通信状況を可視化できる．

NSIM の具体的なメカニズムや実装などの詳細については，文献 7) を参照されたい．また，NSIM は一層の精度向上や機能充実を計るとともに，今後のスーパーコンピュータの研究開発へ貢献するために，シミュレーション実行サービスを TaaS（Tools as a Service）と呼ぶクラウド環境によって公開している．詳細については，文献 10)，11) を参照されたい．

4. 評価実験

本章では，NSIM を用いたシミュレーションによる評価実験について述べる．まず，様々な集団通信アルゴリズムに対してパケットの送出間隔を変えたパケットペーシングを行い，ペーシングの有効性を調査する．次に，各アルゴリズムについて，メッセージの送信時にホップ数に応じたペーシング制御を行いその効果を確認する．

4.1 評価対象システム

集団通信プログラムが動作するシステムは，近年のスーパーコンピュータ相当の機能・性能を有するものを想定する．本評価実験で仮定する評価対象システムのインターコネクトの仕様を**表 2** にまとめる．なお，これらのは，現在の主流となる技術・性能を反映しているが実システムは存在しない．

4.2 パケットペーシング

本研究で用いるパケットペーシング機構は，ハードウェア実装によって実現されていることを前提とする．メッセージの送信手続きが開始され，ルータに搭載された NIC（通信コントローラ）からパケットを送出する際に，パケット長の転送に要する時間を基準とした非送出期間（以下，**パケット間ギャップ**：inter-packet gap）を設ける．ここで，パケット送出時に n パケット分のリンク転送に要する時間だけ待たせる場合を，パケット間ギャップ $= n$（ただし，$n \geq 0$）とする．パケット間ギャップが 0 の場合，パケットは連続して送出される．

NSIM でパケットペーシングを用いたシミュレーションを行う場合，パケット間ギャップの値は MGEN ライブラリの一つである高レベル API 関数で設定する．NSIM はパケット間ギャップの刻み幅を自由に設定することができる．例えば，刻み幅を 1/8 に設定すると最小でパケット長の 1/8 の長さのパケット間ギャップを挿入することができ，精密なペーシングのシミュレーションも可能である．

4.3 実験 1：様々なパケット間ギャップ値による評価

本実験では，集団通信に対するパケットペーシングの有効性を調査する．評価対象アルゴリズムには，2 章で述べた pwx，ring，ssprd，ssprd2，ならびに bruck を用いる．具体的には，パケット間ギャップ値を，0（ペーシングなし）から各トポロジの最大ホップ数まで 1.0 刻みで変化させ，それぞれのギャップ値における集団通信の実行時間を調査する．最大ホップ数は，$x \times y$ の 2 次元トーラス網では $(x+y)/2$，$x \times y \times z$ の 3 次元トーラス網では $(x+y+z)/2$ で求められる．

ギャップ値が 0 から大きくなるにつれてネットワークの通信混雑が緩和されるため，実行時間が減少しペーシングの効果が現れると考えられる．一方，あるギャップ値を境界として大きくなるとパケットの送出間隔が空きすぎることによって逆に実行時間が増加することも容易に予想される．実験に用いた評価パラメータを**表 3** に示す．

4.4 実験 1：結果と考察

各パケット間ギャップ値に対する集団通信の実行時間について，トポロジとノード数ごとにプロットしたものを**図 1** から**図 10** に示す．横軸はパケット間ギャップ値であり，大きくなるにつれて次のパケットを送出するまでの時間が長くなる．縦軸は当該ギャップ値でペーシングした場合の実行時間である．また，各プロットはアルゴリズムとメッセージサイズで分類される．

まず，全ての図からパケット間ギャップが 0 から大きくなるにつれて実行時間が減少していることが分かる．パケットの送出頻度が減少しネットワーク上でのパケットの衝突が緩和

Vol.2011-HPC-130 No.50
2011/7/28

表 3 実験 1 の評価パラメータ

パラメータ	設定値
評価対象アルゴリズム	pwx, ring, ssprd, ssprd2, bruck
トポロジ	2 次元トーラス網，3 次元トーラス網
ノード数	64, 128, 256, 512, 1024
メッセージサイズ	32KiB, 64KiB, 128KiB, 256KiB, 512KiB
通信設定	ゼロコピー通信あり．ランデブー通信なし．
パケット間ギャップ	1〜（各トポロジ・ノードサイズにおける最大ホップ数） 刻み幅：1.0

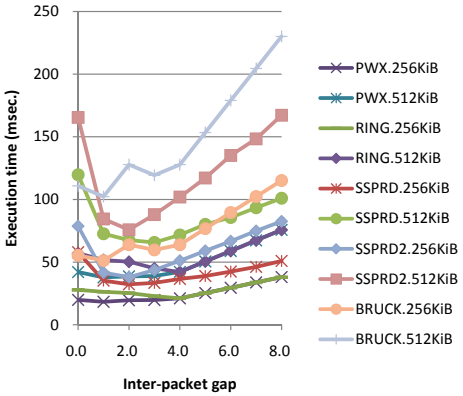

図 1 各パケット間ギャップ値における実行時間（2DT，64 ノード）

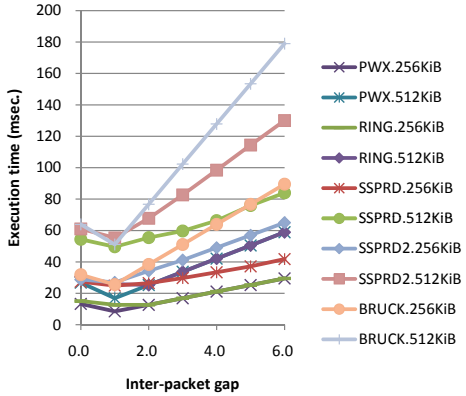

図 2 各パケット間ギャップ値における実行時間（3DT，64 ノード）

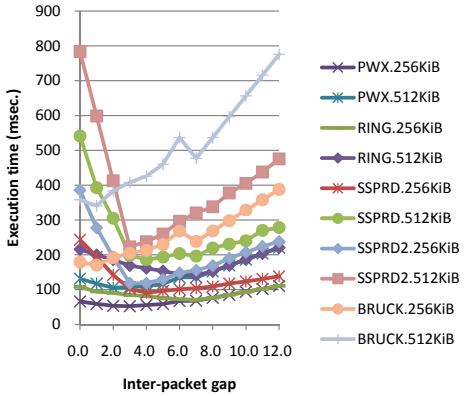

図 3 各パケット間ギャップ値における実行時間（2DT，128 ノード）

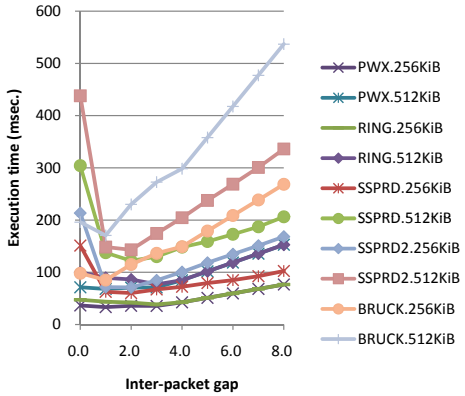

図 4 各パケット間ギャップ値における実行時間（3DT，128 ノード）

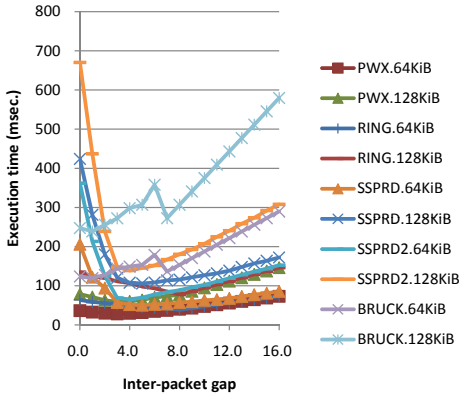

図 5 各パケット間ギャップ値における実行時間（2DT，256 ノード）

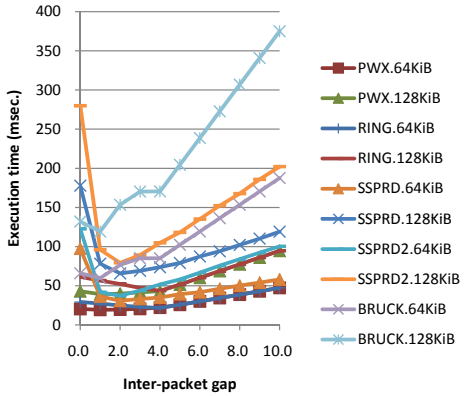

図 6 各パケット間ギャップ値における実行時間（3DT，256 ノード）

され，全体の通信が円滑になったためである．一方，あるパケット間ギャップ値から実行時間が増加している．これは，パケットの送信間隔が大きくなりすぎ，ペーシングの効果が薄れたためである．また，ペーシングの効果が最大となるギャップ値は，アルゴリズムの差異だけでなく，メッセージサイズやノード数の差異によっても異なることが確認できる．

次にアルゴリズムとメッセージサイズに着目する．すべてのトポロジとノード数の場合について，ペーシングを行わない場合（パケット間ギャップ値 = 0）は，pwx が最も実行時間が短く，次いで ring，bruck，ssprd，ssprd2 の順に長くなっている．これらの中で，トポロジやノード数が変わっても pwx，ring，bruck は傾向の大きな変化は見られないが，ssprd,ssprd2 は実行時間が急激に増加している．これは，集団通信が始まるとともに全プロセスがいっせいにメッセージを連続して送信するという Simple spread アルゴリズムの特徴を良く表している．

一方，ペーシングを適用した場合，bruck には効果が低く実行時間が大幅に増加している．これは 2 章で述べたように，プロセス数が大きくなるにつれて全体での総通信量が大幅に増加するため，ペーシングの効果をスポイルしていると考えられる．pwx，ring については大きな変化が見られないが，ssprd と ssprd2 は急激な性能向上が確認できる．特にノード

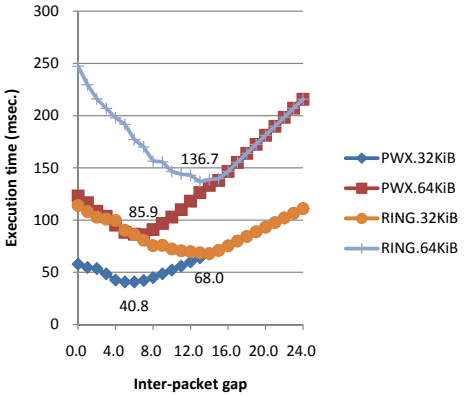

図 7 各パケット間ギャップ値における実行時間（2DT，512 ノード）

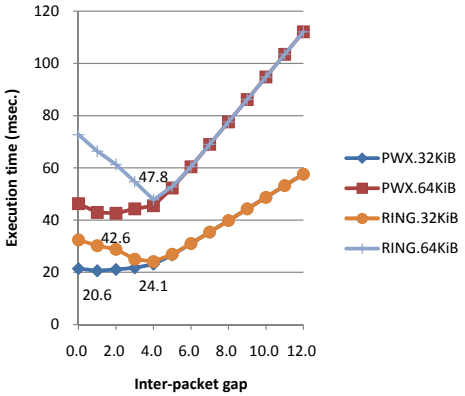

図 8 各パケット間ギャップ値における実行時間（3DT，512 ノード）

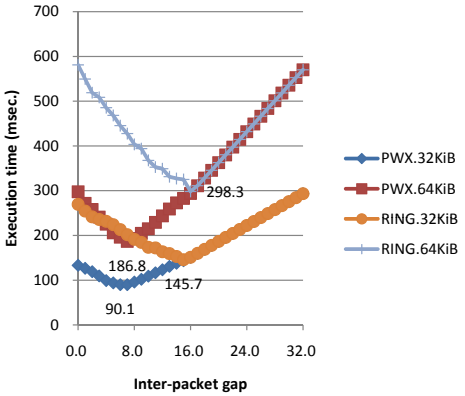

図 9 各パケット間ギャップ値における実行時間（2DT，1,024 ノード）

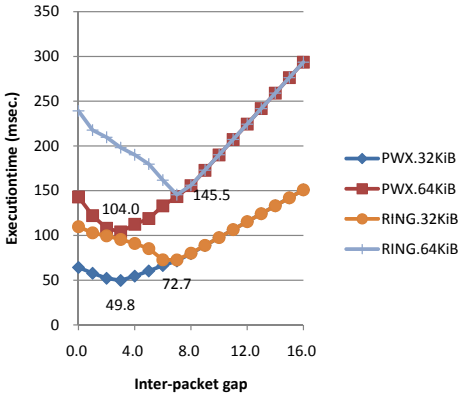

図 10 各パケット間ギャップ値における実行時間（3DT，1,024 ノード）

数が大きくなると buruck を上回り，**図 5** や**図 6** からは，メッセージサイズが小さい場合は ring に近い性能を示すことが分かる．これは一度に大量のパケットがネットワークに投入されることをペーシングによって抑止するためである．

評価に用いたアルゴリズムのうち，他のものよりも実行時間が短い pwx と ring を 512 ノードと 1024 ノードで実行した結果を**図 7** から図 10 に示す．なお，図中には，ペーシングを適用し各アルゴリズムで最も実行時間が短くなったポイントについてその実行時間を示している．これらの図から，ノード数やメッセージサイズが大きくなると pwx や ring でもペーシングの効果があることがわかる．1,024 ノードにおける pwx の実行時間は，2 次元および 3 次元トーラス網でそれぞれ 90.1ms，49.8ms であるが，表 1 に示した理想実行時間と比較すると 2.6 倍から 2.9 倍遅い．すなわち，インターコネクトが持つ転送能力の 3 割程度しか利用していないことになる．

表 4 実験 2 の評価パラメータ

パラメータ	設定値
評価対象アルゴリズム	pwx, ring, ssprd, ssprd2, bruck
ノード数	64, 128, 256, 512, 1024
メッセージサイズ	32KiB, 64KiB, 128KiB, 256KiB, 512KiB
通信設定	ゼロコピー通信あり．ランデブー通信なし．
パケット間ギャップ	メッセージのホップ数 −1

以上の実験から，一般的な集団通信にパケットペーシングを適用した場合の有効性が確認できた．また，アルゴリズムをはじめ，トポロジ，ノード数，メッセージサイズに応じて，集団通信を最適に実行するパケット間ギャップ値は異なることが確認できた．また，各ノードサイズで最も実行時間が短い pwx であっても，十分にインターコネクトの性能を引き出していないことが分かった．

4.5 実験 2：ホップ数によるパケットペーシング

ルータに 1 つの NIC を搭載する場合，送信あるいは受信は 1 つしかできない．よって，トーラス網において集団通信を行う場合，経路上のリンクを共有するメッセージ数は高々ホップ数と同じである．したがって，理論的にはパケットの送出時に（（ホップ数 −1）× パケット長）の転送に要するパケット間ギャップを設けることで，他のパケットと干渉することなく交互にリンクを利用することが可能である．

そこで，本実験では各集団通信アルゴリズムによってメッセージを送信する際に，パケット間ギャップ値をホップ数に応じて決定するペーシング（以下，ホップペーシング）を行い，その効果を明らかにする．

表 4 に本実験に用いたパラメータを示す．

4.6 実験 2：結果と考察

図 11 から**図 14** に，ホップペーシングを適用した集団通信の実行時間をノードサイズ別に示す．図中の 2 本のグラフは左からそれぞれホップペーシングを適用しない場合と適用し

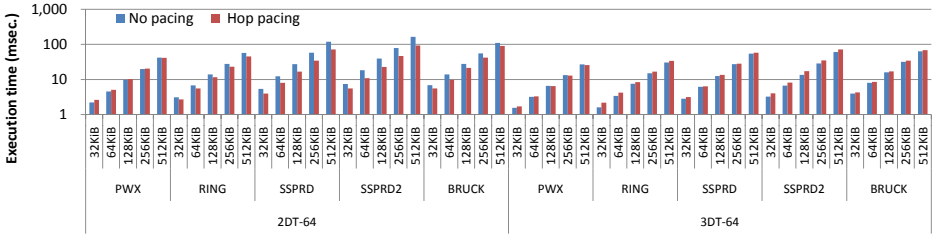

図 11 ホップペーシングによる実行時間（64 ノード）

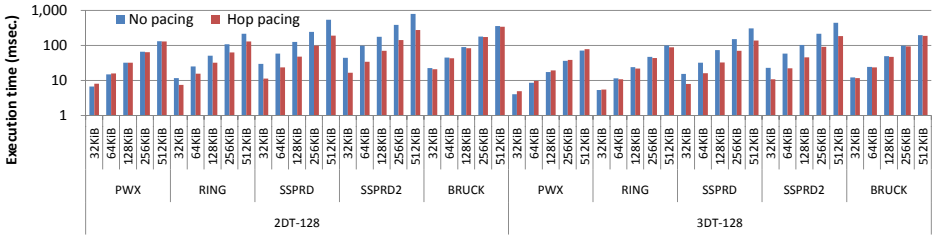

図 12 ホップペーシングによる実行時間（128 ノード）

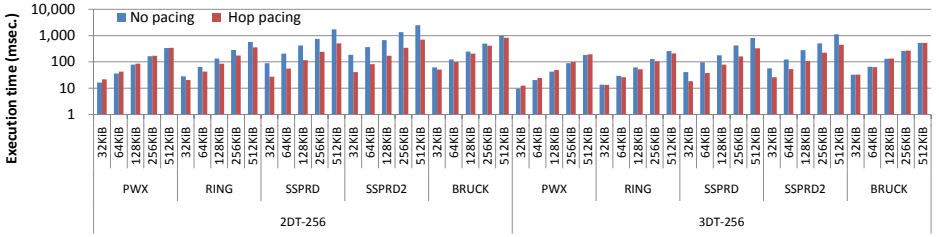

図 13 ホップペーシングによる実行時間（256 ノード）

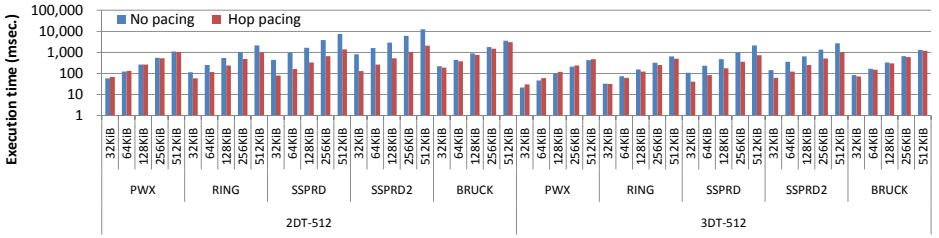

図 14 ホップペーシングによる実行時間（512 ノード）

た場合である．これらの図から，pwx 全般と 3DT-64 以外ではホップペーシングの効果が確認できる．特に，ssprd と ssprd2 の速度向上が大きく，ノード数が大きくなるにつれてパケットペーシングの効果が顕著になっている．

これらの変化を詳細に確認するために，**図 15** から**図 18** にホップペーシングによる速度向上率をノードごとに示す．これらの図から，どのノード規模においても pwx は有効な速度向上がほとんど見られない．また，ring はノード数が大きくなると悪化していることがわかる．これは，pwx，ring，bruck では，通信パターンによっては経路上での衝突が全くないステップもあるため，不用なペーシングにより逆に通信性能が悪くなった可能性が考えられる．しかし，メッセージサイズが大きくなると衝突が発生する時間が長くなり，ペーシングの効果が現れてくるといえる．

一方，ssprd と ssprd はノード規模が大きくなるにつれてペーシングの効果が増していることが分かる．これは，4.4 章で述べたように，一度に大量のパケットがネットワークに投入されることをペーシングによって抑止する作用が大きいためと考えられる．

以上の実験から，通信ホップ数に応じたペーシングは，衝突を引き起こす転送パケットがネットワーク上に大量にある場合やメッセージサイズが大きい場合に効果的であるといえる．

5. ま と め

本稿では，一般的な集団通信アルゴリズムに対してパケットペーシングが与える影響を調査することを目的とし，2 次元トーラス網，ならびに 3 次元トーラス網での集団通信アルゴリズムの性能評価を行った．各集団通信に対してパケットペーシングを適用しシミュレーションによってその有効性を示した．また，通信ホップ数に応じたパケット間ギャップを設定することで，アルゴリズムによっては大幅な実行時間の短縮が達成されることを確認した．

実システムで利用するには OS ジッタや計算負荷の不均衡を考慮した評価も必要でなる．これについては重要な課題とする．今後は，さらにノード規模を大きくし，集団通信だけでなく様々な実践的なアプリケーションについても評価を行う予定である．

謝辞 本研究を進めるにあたりご協力いただく九州大学村上和彰教授，青柳睦教授，南里豪志准教授に感謝する．また，本研究の実験結果の一部は，九州大学情報基盤研究開発センターの研究用計算機システムを用いて取得したことを付記する．

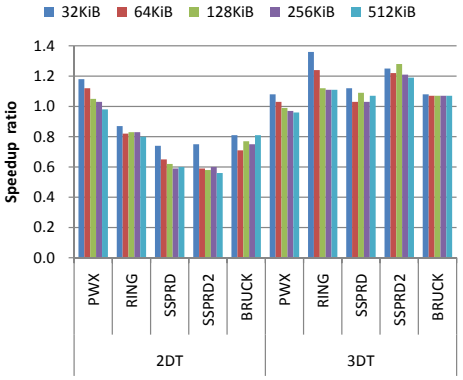

図 15 ホップペーシングによる速度向上率（64 ノード）

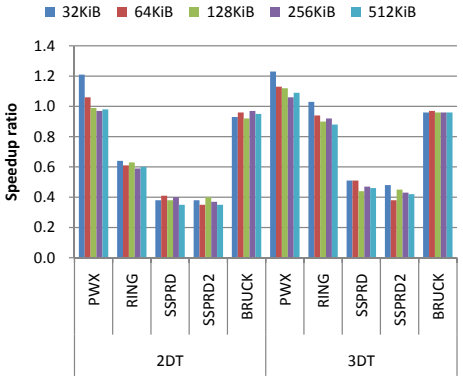

図 16 ホップペーシングによる速度向上率（128 ノード）

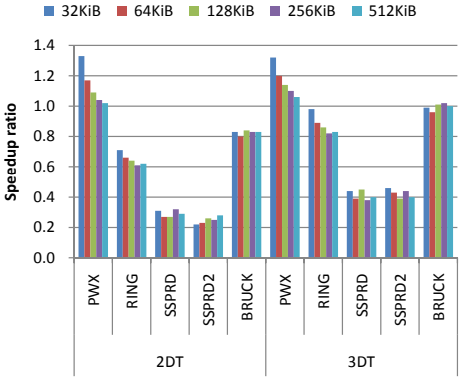

図 17 ホップペーシングによる速度向上率（256 ノード）

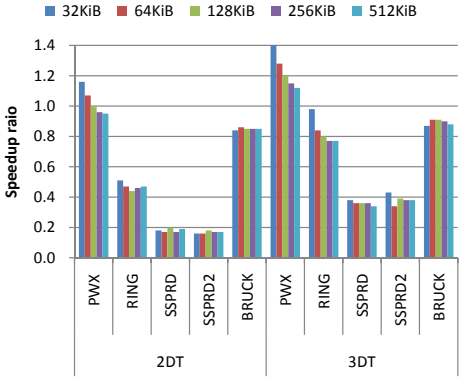

図 18 ホップペーシングによる速度向上率（512 ノード）

参 考 文 献

1) Faraj, A., Xin, Y. and Lowenthal, D.: STAR-MPI: Self Tuned Adaptive Routines for MPI Collective Operations, *Proc. 20th ACM Intl. Conf. on Supercomputing (ICS06)* (2006).
2) MPICH2: http://www.mcs.anl.gov/research/ projects/mpich2/.
3) Aggarwal, A., Savage, A. and Anderson, T.: Understanding the Performance of TCP Pacing, *IEEE INFOCOM2000 Conference on Computer Communications*, pp.1157–1165 (2000).
4) 高野了成，工藤知宏，児玉祐悦，松田元彦，石川裕，岡崎史裕：ギャップパケットを用いたソフトウェアによる精密ペーシング方式，情報処理学会論文誌コンピューティングシステム， Vol.47, No.SIG7, pp.194–206 (2006).
5) 柴村英智，三輪英樹，薄田竜太郎，平尾智也，安島雄一郎，三吉郁夫，清水俊幸，石畑宏明，井上弘士：パケットペーシングによる全対全通信の最適化とシミュレーション評価，情報処理学会論文誌：コンピューティングシステム， Vol.4, No.3, pp.56–65 (2011).
6) Bruck, J., Ho, C., Kipnis, S., Upfal, E. and Weathersby, D.: Efficient Algorithms for All-to-all Communications in Multiport Message-Passing Systems, *IEEE Trans. Parallel and Distributed Systems, 8(11)*, pp.1143–1156 (1997).
7) 三輪英樹，薄田竜太郎，柴村英智，平尾智也，眞木淳，稲富雄一，井上弘士，安島雄一郎，三吉郁夫，清水俊幸，安藤壽茂：NSIM:将来の大規模相互結合網を対象とした通信シミュレータの開発，情報処理学会研究報告 2010-HPC-125，pp.1–9 (2010).
8) 井上弘士，薄田竜太郎，安藤壽茂，石附茂，小松秀実，稲富雄一，本田宏明，山村周史，柴村英智，于雲青，青柳睦，木村康則，村上和彰：大規模システム評価環境 PSI-SIM: 数千個のマルチコア・プロセッサを搭載したペタスケールコンピュータの性能予測，情報処理学会研究報告，Vol.2008, No.39, pp.51–56 (2008).
9) Susukita, R., Ando, H., Aoyagi, M., Honda, H., Inadomi, Y., Inoue, K., Ishizuki, S., Kimura, Y., Komatsu, H., Kurokawa, M., Murakami, K.J., Shibamura, H., Yamamura, S. and Yu, Y.: Performance Prediction of Large-scale Parallel System and Application using Macro-level Simulation, *Proc. Intl. Conf. Supercomputing (SC2008)* (2008).
10) 柴村英智，薄田竜太郎，平尾智也，吉田真，神戸隆行，三輪英樹，三吉郁夫，井上弘士，村上和彰：クラウド環境による OpenNSIM インターコネクトシミュレーションサービス，情報処理学会研究報告 2010-ARC-192,2010-HPC-128，pp.1–9 (2010).
11) OpenNSIM: https://ngarch.isit.or.jp/taas/opennsim/.