Vol.2012-HPC-135 No.2
2012/8/1

パケットペーシングを用いた集団通信に対するロード／ネットワークインバランスの影響

柴村 英智[1,2,a)]　三輪 英樹[3,b)]　三吉 郁夫[3,c)]　井上 弘士[4,d)]

概要：本稿では，ロードインバランスやネットワークインバランスに起因する通信開始時刻のインバランスが，パケットペーシングを用いた集団通信の実行に与える影響をシミュレーションによって評価する．3 次元トーラス網ならびに 2 次元トーラス網を対象に，インバランスの付加や MOD ペーシングを適用した様々な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 を用いた評価を行った．その結果，集団通信のアルゴリズムによってインバランスの感受性が異なることがわかった．また，集団通信に対するペーシングの有効性を確認するとともに，メッセージサイズやノード数の増加に応じて実行時間の高速化率も向上することがわかった．さらに，ペーシングを適用した集団通信にインバランスが及ぼす影響を評価した結果，アルゴリズムによっては，わずかなインバランスが加わることで実行時間が大幅に増加し，ペーシングの効果を損なう場合があることが明らかになった．

キーワード：インターコネクト，インバランス，パケットペーシング，シミュレーション，NSIM

Influence of Load/Network Imbalances on Collective Communication using Packet Pacing

Hidetomo Shibamura[1,2,a)]　Hideki Miwa[3,b)]　Ikuo Miyoshi[3,c)]　Koji Inoue[4,d)]

Abstract: This paper investigates influence of imbalance of communication start times caused by load and/or network imbalances on collective communication using packet pacing. Several simulations of collective communication are carried out with various imbalances and MOD pacing on 3D and 2D tori by using an interconnect simulator NSIM. As the result, we confirmed that the collective communication algorithms have different and respective sensitivities of imbalance. Moreover, the packet pacing improves speedup ratio of execution further by increasing message size and/or number of nodes, thus the effectiveness of the packet pacing is cleared. However, few collective communications using the pacing may spoil the effectiveness by small imbalance.

Keywords: Interconnect, Imbalance, Packet pacing, Simulation, NSIM

[1] 財団法人九州先端科学技術研究所
Institute of Systems, Information Technologies and Nanotechnologies
[2] 独立行政法人科学技術振興機構，CREST
Japan Science and Technology Agency, CREST
[3] 富士通株式会社
Fujitsu Ltd.
[4] 九州大学
Kyushu University
[a)] shibamura@isit.or.jp
[b)] miwa.hideki@jp.fujitsu.com
[c)] miyoshi.ikuo@jp.fujitsu.com
[d)] inoue@ait.kyushu-u.ac.jp

1. はじめに

我々は，メッセージ通信においてパケット送出間隔を制御し通信効率を改善する，インターコネクト向けのパケットペーシング技術について研究を行っている．これまでに，バースト的なトラフィックとなる各種の集団通信について，適切なパケット送出間隔でペーシングを行うことで高速化が図れることをシミュレーション評価で確認している [1], [2]．これらのシミュレーションでは，パケットペー

シングの基本的な能力を明らかにするために，評価対象とするインターコネクトの基本仕様には現実的なパラメータを用い，プロセスの振る舞いも理想的とするために集団通信を担うプロセスは同時刻に開始するものとしている．したがって，パケットペーシングの実践的な利用に向けて，各プロセスの開始時刻を不均衡とした評価が必要である．

現在の主流であるメッセージパッシングモデルに基づく並列システムでは，各プロセスにおける負荷バランスの不均衡（ロードインバランス），ならびにインターコネクトにおける不等距離通信や通信混雑による通信遅延の不均衡（ネットワークインバランス）によって，各プロセスの演算開始時刻や通信開始時刻に差異が生じる．これらに起因して，通信待ち合わせの多い集団通信では，通信開始時刻のインバランスによって実行性能が大きく変化することが過去の研究から明らかになっている [3]．

本研究では，パケットペーシングを適用した集団通信に対して，ロードインバランスやネットワークインバランス（以下，これらに起因する演算や通信の開始時刻の不均衡をインバランスと総称）が与える影響を明らかにすることを目的とする．具体的には，まず，インバランスを定量的に表すインバランス係数を定める．次に，この係数の差異が集団通信の実行に与える影響についてシミュレーション評価を行う．ここでは，集団通信アルゴリズム，トポロジ，ノード数，およびメッセージサイズを様々に変えて評価する．そして，MOD ペーシングと呼ぶ，同時期に通信リンクを経由するメッセージ数に基づいたペーシング手法を集団通信に適用した場合の有効性について調査する．最後に，MOD ペーシングを用いた集団通信に各種のインバランスを加え，インバランスがパケットペーシングによる集団通信に及ぼす影響を総合的に評価する．なお，OS ジッタといった動的にプロセスの負荷に影響を与える変動は今後の課題とし，本研究では考慮しない．

以下，2 章では，評価実験に向けたインバランスの定量化の方法を定める．3 章では，MOD ペーシングと呼ぶパケットペーシング手法について概説する．4 章では，集団通信に対して各種のインバランスやパケットペーシングを適用した評価実験，ならびにそ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. インバランスの定量化

本研究で用いるインバランスの大きさを定量的に表現するためにインバランス係数（imbalance factor）を定める．

文献 [3] では，インバランス係数（集団通信開始時刻のぶれ）を，実行プロセスがプログラム中の集団通信に到着した時刻（集団通信を開始しようとする時刻）とすべての実行プロセスの平均到着時刻との差（インバランス時間）を，隣接ノードとの 1 メッセージの平均通信時間で正規化したもので表現している．具体的には，あるノードに到着

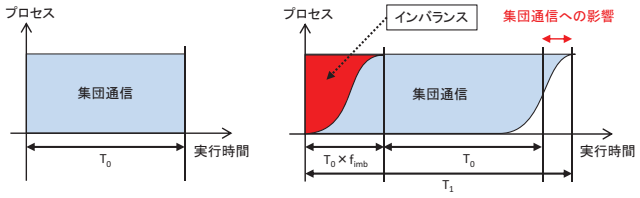


図 1 集団通信へのインバランスの挿入
Fig. 1 Insertion of imbalances into a collective communication.

したメッセージのインバランス係数が 0 の場合は，平均到着時刻にメッセージが到着することを表す．また，インバランス係数が 10 の場合は，平均到着時刻から 10 メッセージ分の通信時間だけ速く，もしくは遅れてメッセージが到着する．換言すれば，インバランスによってメッセージの到着が遅れた場合，それまでに高々 10 メッセージを送信できる機会を持つことになる．

一方，様々な集団通信アルゴリズムを一様に評価するためには，各集団通信の実行時間に対するインバランス時間の相対的な割合で比較しなければならない．これは，プロセス数の変化やアルゴリズムの差異によって集団通信の実行時間も変わるためである．そこで，本研究で扱うインバランス係数（f_{imb}）は，インバランスを単純に表現するために，**図 1** 左に示す集団通信の実行時間 T_0 の百分率とする．

集団通信を行う各プロセスには，図 1 右のように T_0 の $f_{imb}(\%)$ を幅としたインバランスを挿入し，その際の実行時間 T_1 を求めることで当該インバランスによる集団通信への影響を評価する．すなわち，インバランスの挿入による実行時間の増加割合がインバランス係数よりも大きいほど，その集団通信アルゴリズムや実行環境はインバランスの影響を受けやすい．一方，場合によってはインバランスによって通信混雑が緩和し，実行時間の増加割合がインバランス係数を下回る（実行が速くなる）こともある．

3. 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．また，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ものとする．

このようなパケット送出機構を搭載したスーパーコンピュータには，理研の「京」や富士通社製「PRIMEHPC FX10」がある．これらに搭載されている Tofu インターコネクトのルータチップ（ICC）では，トーラス網のような不等距離網での通信において広域的な公正性（global

fairness）をパケットの調停時に保つよう，転送パケット間のギャップを設定し，ネットワークへのパケットの投入率を制御することが可能となっている [4].

集団通信を高速化する最適なパケット間ギャップ値の導出手法として，**MOD（Message Overlap Degree）ペーシング**を用いる．これは，多くの集団通信アルゴリズムは 1 対 1 のメッセージ送受信が複数の通信ステップから成ることと，同一ステップ時に通信リンクを経由するメッセージ数の上限に着目した方法である [5]．具体的には，ある通信ステップ時にメッセージが経由するすべてリンクについて，オーバラップするメッセージの重複数を求め，この最大重複数 − 1 を当該ステップにおけるパケット間ギャップとするものである．なお，メッセージの重複数は，当該システムのルーティングアルゴリズムを用いることで容易に算出することが可能である．したがって，本手法では通信ステップ毎に新たなパケット間ギャップ値を必要とするが，現実的には集団通信の開始時あるいはプロセスの初期化時（MPI_Init など）に，各ステップにおけるギャップ値を事前に算出しておくことで，実行時のオーバヘッドを省くことができる．

従来のパケット間ギャップ値の導出には，大量のシミュレーションによるパラメータスタディから得られた最適値を用いたり [1]，各通信ステップにおける最大ホップ数を用いたホップペーシング [2] を用いていた．しかし，本 MOD ペーシングは，前者と比較して低い処理コストで，かつ，後者と比較して効率の良いパケット間ギャップ値を得ることが可能である．

4. 評価実験

4.1 実験内容

本実験では次の 3 項目について評価を行った．

(1) 集団通信に対するインバランスの影響
(2) 集団通信に対する MOD ペーシングの有効性
(3) MOD ペーシングを適用した集団通信に対するインバランスの影響

これらの実験にはインターコネクトシミュレータ NSIM[6] を用い，インバランス係数，トポロジ，ノード数，メッセージサイズ，パケットペーシングの有無を変えながら，各種の集団通信アルゴリズムについて実行時間を測定した．

4.2 評価対象システム

評価対象システムは，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 1** にまとめる．なお，これら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．

評価項目については，インバランス係数 (f_{imb}) を 0%から 10%までの 2%刻みとし，トポロジは 3 次元トーラス網と 2 次元トーラス網とした．また，ノード数は 128 から 1024 ノードまでとし，メッセージサイズは 16KiB から 128KiB とした．なお，パケット転送時にパケットヘッダが付加されると半端な最終パケットのサイズによってインバランスが発生するため，本実験ではパケットヘッダも含めたパケット長の倍数にアライメントを取っている．したがって，以降で示すメッセージサイズはパケットヘッダ長も含まれていることに注意されたい．集団通信アルゴリズムには，一般的な MPI 通信ライブラリに実装されている全対全通信系のアルゴリズムのうち，pairwise-exchange (以下，pwx)，ring，simple spread (ssprd)，bruck，ならびに butterlfy (btfly) を用いた [7], [8], [9].

表 1 評価対象システムの仕様
Table 1 Specification of an evaluation target system.

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

4.3 インバランスの与え方

所望するインバランス係数で評価を行うために，各プロセスの開始時刻を適切に設定しなければならない．本実験では，評価対象プログラムで集団通信関数が呼び出される前に，MGEN_Comp 関数と呼ぶ NSIM のライブラリによって，指定された時間を NSIM の仮想時刻に加算する方式を採った．すなわち，プロセス毎に設定された時間だけシミュレーション時間が経過すると集団通信を開始する．

また，インバランス係数に応じた各プロセスのインバランス値を生成するプログラムを別途作成した．これは，インバランス係数とノード数を入力とし，正規乱数によって正規分布に従ったインバランス値をノード数分だけ出力す

表 2　実験 1 のシミュレーションパラメータ
Table 2　Simulation parameters for experiment 1.

パラメータ	設定値
評価対象アルゴリズム	pwx, ring, ssprd, bruck, btfly
トポロジ	3 次元, 2 次元トーラス網
ノード数	128, 256, 512, 1024
メッセージサイズ	16KiB, 32KiB, 64KiB, 128KiB
通信設定	ゼロコピー通信あり
	ランデブー通信なし
インバランス係数（f_{imb}）	0%, 2%, 4%, 6%, 8%, 10%

るものである．正規乱数の生成には一般的な Box-Muller 法を用いた．指定されたインバランス係数に応じたプロセス開始時刻を本プログラムで生成し，インバランス値を昇順に各プロセスの開始時刻に与えるものとした．

4.4　パケット間ギャップの導出

第 3 章で述べた MOD ペーシングを NSIM で行うために，評価対象プログラムで集団通信を開始する前に，各通信ステップのパケット間ギャップ値をすべて導出している．

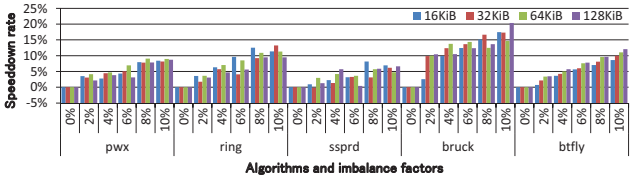
まず，ある通信ステップにおけるメッセージ通信すべてについて，評価対象システムのルーティングアルゴリズムに基づき，通信リンクを経由する度にそのリンクにおけるメッセージの重複数をカウントする．次に，全リンクで最も大きな重複数を求め，その 最大重複数 − 1 を当該通信ステップにおけるパケット間ギャップ値とする．以上をすべての通信ステップについて行い，ステップ数分のパケット間ギャップ値をプロセス終了時まで保持する．なお，パケット間ギャップを求めるコストは 0 としており，評価指標とする実行時間には含まれない．

4.5　実験 1：集団通信に対するインバランスの影響

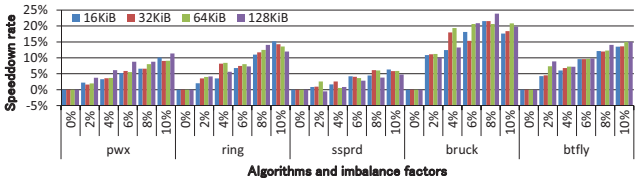
本実験では，インバランスが集団通信の実行に与える影響について評価する．**表 2** に示す評価パラメータに基づき，NSIM によって評価対象システムにおける各集団通信アルゴリズムの実行時間をインバランス係数毎に測定した．

理想的な集団通信の開始が行われた場合，すなわち，インバランス係数が 0%の実行時間を基準に，係数増加に対する実行時間増加の割合を，3 次元トーラス網について**図 2** (a) ～ (d) に，2 次元トーラス網については図 2 (e) ～ (h) に示す．横軸は集団通信アルゴリズムの種類と与えたインバランス係数であり，縦軸は実行時間増加の割合である．また，メッセージサイズ毎にそれぞれの結果を示している．なお，1,024 ノードにおける ssprd は一部シミュレーションに非常に時間を要しており割愛する．

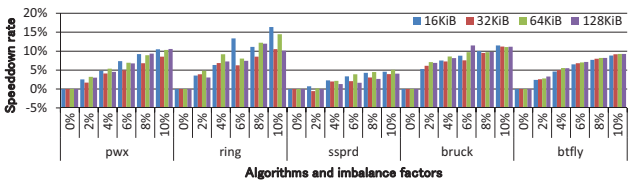
図 2 (a) ～ (d) の 3 次元トーラス網では，各アルゴリズムはインバランスの増加にともない集団通信の実行時間が概ね増加している．ただし，メッセージサイズによっては大きく増加の割合が異なる場合がある．これは，当該アルゴリズムがインバランスの影響を受けやすいことを意味する．そのような点では，btfly はメッセージサイズ，ノード数，インバランス係数に関わらず一定の割合で実行時間が



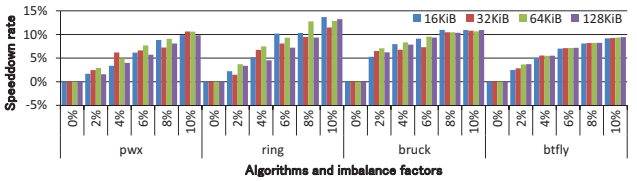
(a) 3 次元トーラス網，128 ノード（8×4×4）



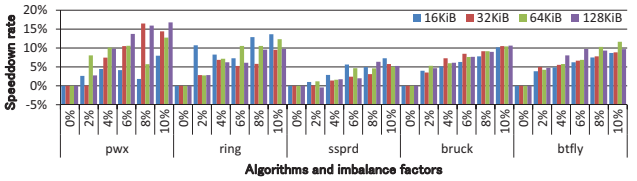
(b) 3 次元トーラス網，256 ノード（8×8×4）



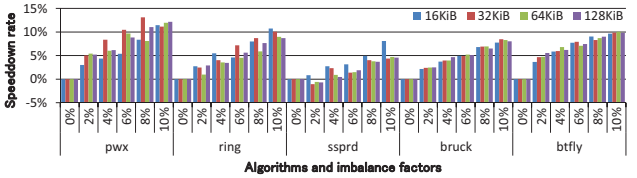
(c) 3 次元トーラス網，512 ノード（8×8×8）



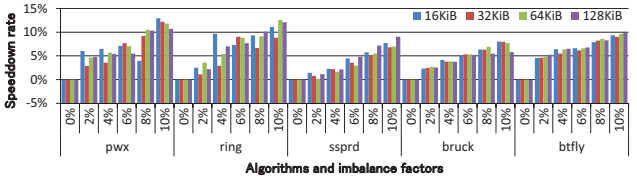
(d) 3 次元トーラス網，1024 ノード（16×8×8）



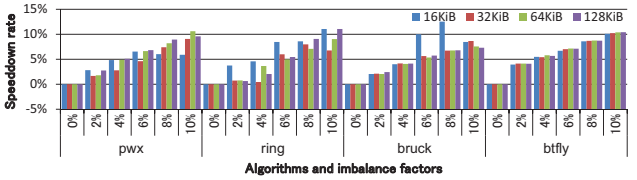
(e) 2 次元トーラス網，128 ノード（16×8）



(f) 2 次元トーラス網，256 ノード（16×16）



(g) 2 次元トーラス網，512 ノード（32×16）



(h) 2 次元トーラス網，1024 ノード（32×32）

図 2　インバランス係数に対する実行時間増加の割合
Fig. 2　Speeddown rates for each imbalance factor.

表 3 実験 2 のシミュレーションパラメータ
Table 3 Simulation parameters for experiment 2.

パラメータ	設定値
評価対象アルゴリズム	pwx, ring, ssprd, bruck, btfly
トポロジ	3 次元, 2 次元トーラス網
ノード数	128, 256, 512, 1024
メッセージサイズ	16KiB, 32KiB, 64KiB, 128KiB
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	MOD ペーシング

増加しており，ノード数が大きくなるとその増加率はインバランス係数よりも低くなっているため，インバランスの影響が低いと言える．また，ssprd は，場合によってはインバランスが加わることにより実行時間が速くなっている（図 2（b），(c)）．ssprd は集団通信の開始とともに一斉に全てのプロセスに対してメッセージを送信するため，インバランスによる通信開始の遅延が通信混雑の緩和を促しているためである．したがって，ssprd は全アルゴリズムの中でも最も実行時間の増加が少なくなっている．

図 2（e）～（h）の 2 次元トーラスにおいても 3 次元トーラス網と同様に，インバランスの増加に応じて集団通信の実行時間が増加する傾向にある．しかし，pwx と ring はノード数が小さいほど 3 次元トーラス網よりもインバランスの変化にともなう実行時間の増加率の変動が大きい．これらは，ステップ数の多いロックステップ通信を行うためメッセージの待ち合わせが多く，ノード次数が小さくルーティングの自由度が低い 2 次元トーラス網では 3 次元トーラス網よりもホップ数が大きくなるため，インバランスによる通信遅延の増加が通信順序を乱すためと考えられる．

以上の結果から，全対全通信系の集団通信は，アルゴリズムをはじめ，トポロジ，メッセージサイズ，ノード数に応じてインバランスの感受性が異なることがわかった．

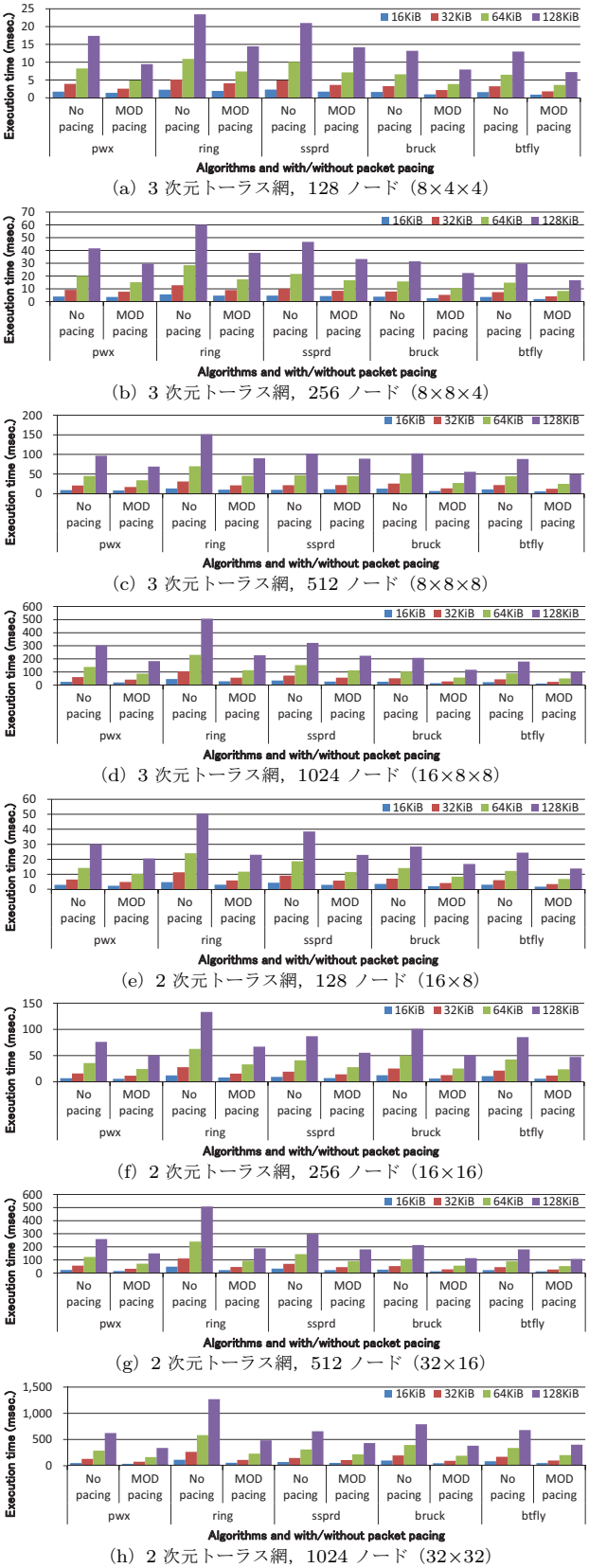
4.6 実験 2：集団通信に対する MOD ペーシングの有効性

本実験では，集団通信に MOD ペーシングを適用した場合の有効性について調査する．**表 3** に示すパラメータに基づき，NSIM によって MOD ペーシングを適用した集団通信アルゴリズムの実行時間を測定した．

ペーシングの有無による各アルゴリズムの実行時間を**図 3** に示す．横軸は集団通信アルゴリズムの種類とペーシングの有無（No pacing：無，MOD pacing：有）であり，縦軸は実行時間である．また，ペーシングによる各アルゴリズムの速度向上率を**図 4** に示す．横軸はメッセージサイズ，縦軸はペーシングによる速度向上率である．

図 3 の全グラフから，トポロジ，ノード数，メッセージサイズに依らず，各アルゴリズムに MOD ペーシングを適用することで実行が速くなっていることが確認できる．

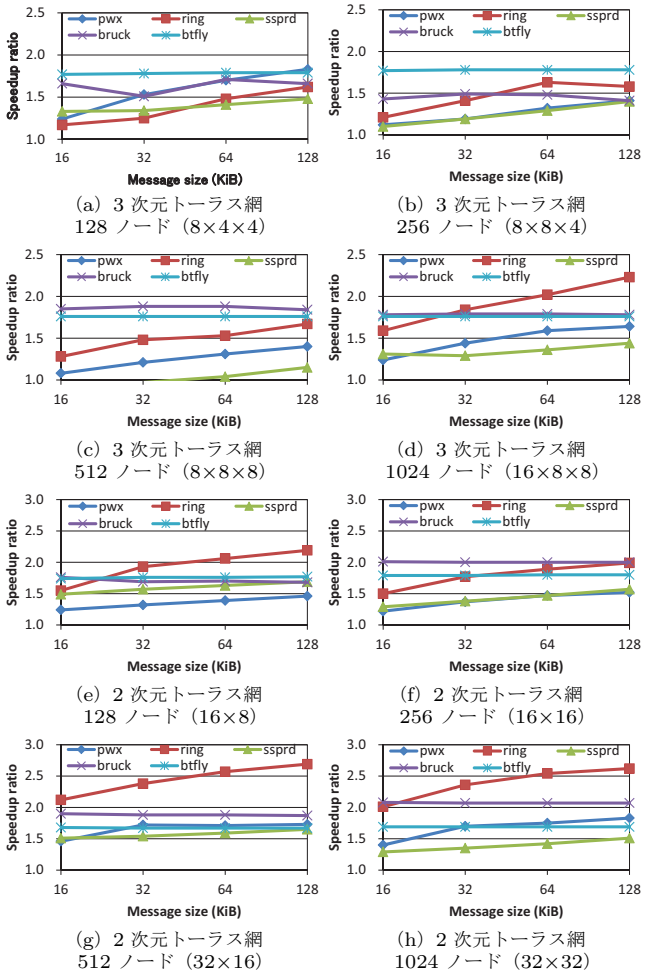
ノード数が小さい場合は bruck や btfly の実行速度が速



（a）3 次元トーラス網，128 ノード（8×4×4）
（b）3 次元トーラス網，256 ノード（8×8×4）
（c）3 次元トーラス網，512 ノード（8×8×8）
（d）3 次元トーラス網，1024 ノード（16×8×8）
（e）2 次元トーラス網，128 ノード（16×8）
（f）2 次元トーラス網，256 ノード（16×16）
（g）2 次元トーラス網，512 ノード（32×16）
（h）2 次元トーラス網，1024 ノード（32×32）

図 3 MOD ペーシングの有無による各アルゴリズムの実行時間
Fig. 3 Execution times with/without MOD pacing.

いが，図 3（h）の 2 次元トーラス網 1,024 ノードでは，pwx の方が速くなっている．これは，図 4（e）～（h）から，ノード数の増加に対する実行速度の向上率が bruck や btfly



(a) 3 次元トーラス網 128 ノード (8×4×4)
(b) 3 次元トーラス網 256 ノード (8×8×4)
(c) 3 次元トーラス網 512 ノード (8×8×8)
(d) 3 次元トーラス網 1024 ノード (16×8×8)
(e) 2 次元トーラス網 128 ノード (16×8)
(f) 2 次元トーラス網 256 ノード (16×16)
(g) 2 次元トーラス網 512 ノード (32×16)
(h) 2 次元トーラス網 1024 ノード (32×32)

図 4 MOD ペーシングによる各アルゴリズムの速度向上率
Fig. 4 Speedup ratios with MOD pacing for each algorithms.

よりも pwx の方が大きいためであると言える．すなわち，ノード数がさらに大きくなるにつれてに速度向上率はさらに増加すると予想される．同様に，ssprd を除く他のアルゴリズムもノード数が増加すると徐々にペーシングの効果が向上していることが確認できる．

一方，メッセージサイズについては，bruck と btfly を除き，3 次元，2 次元トーラス網においてサイズが大きくなるにつれてペーシングの効果が向上していることがわかる．

以上のことから，全対全通信系の集団通信に対するペーシングの有効性を確認するとともに，アルゴリズムによってはメッセージサイズやノード数の増加に応じて実行時間の高速化率も向上することがわかった．

4.7 実験 3：MOD ペーシングを適用した集団通信対するインバランスの影響

本実験では，MOD ペーシングを用いた集団通信に各種のインバランス係数を与え，インバランスがペーシングを適用した集団通信に及ぼす影響を評価する．実験に用いたパラメータを**表 4** に示す．

インバランス挿入とパケットペーシングによる各アルゴリズムの実行時間を，3 次元トーラス網について**図 5** に，2 次元トーラス網については**図 6** に示す．それぞれ，横軸は集団通信アルゴリズムの種類，ペーシングの有無，インバランス係数であり，縦軸は集団通信の実行時間である．また，メッセージサイズ毎にそれぞれの結果を示している．

表 4 実験 3 のシミュレーションパラメータ
Table 4 Simulation parameters for experiment 3.

パラメータ	設定値
評価対象アルゴリズム	pwx, ring, ssprd, bruck, btfly
トポロジ	3 次元, 2 次元トーラス網
ノード数	128, 256, 512, 1024
メッセージサイズ	16KiB, 32KiB, 64KiB, 128KiB
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	MOD ペーシング
インバランス係数 (f_{imb})	0%, 2%, 4%, 6%, 8%, 10%

図 5 (a) ～ (d) から，pwx と ring については様々なインバランスが与えられてもペーシングの効果が現れており，ノード数が大きくなるにつれてその効果も増加していることが確認できる．一方，ssprd，bruck，btfly は，インバランスが無い場合には十分なペーシングの効果があるが，インバランスが加わることでその効果は大きく抑制されていることがわかる．また，ssprd はノード数が大きくなることでペーシング前よりも実行速度が悪化している．

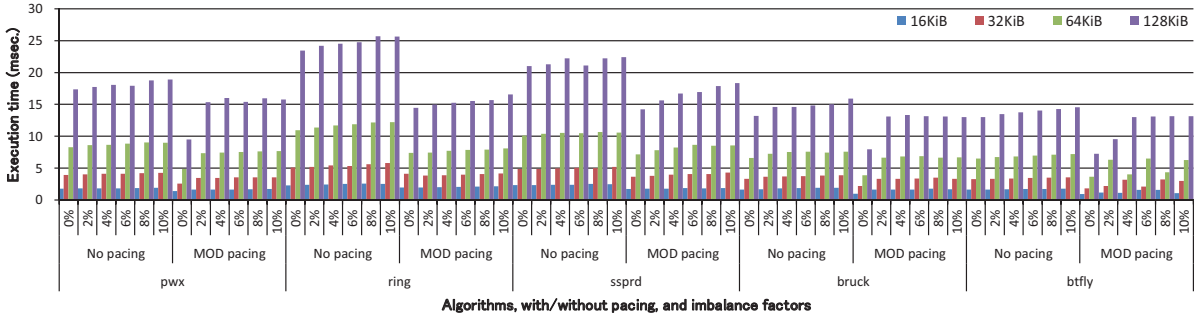
2 次元トーラス網についても，図 6 (a) ～ (d) から前述と同じ傾向が見られる．しかし，1,024 ノードの btfly については，インバランス係数が変化しても持続したペーシングの効果が出ている．

以上の結果から，アルゴリズムによっては，わずかなインバランスが加わることで実行時間が大幅に増加し，ペーシングの効果を損なう場合があることがわかった．

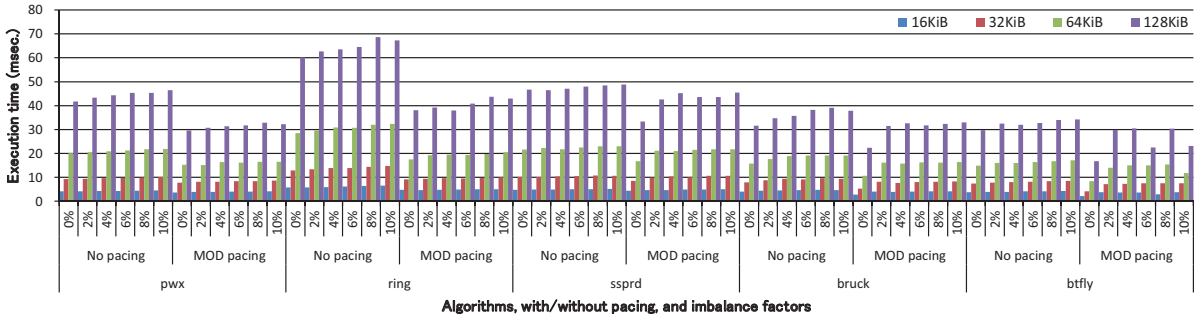
5. まとめ

本稿では，ロードインバランスやネットワークインバランスに起因する通信開始時刻のインバランスが，パケットペーシングを用いた集団通信の実行に与える影響をシミュレーションによって評価した．

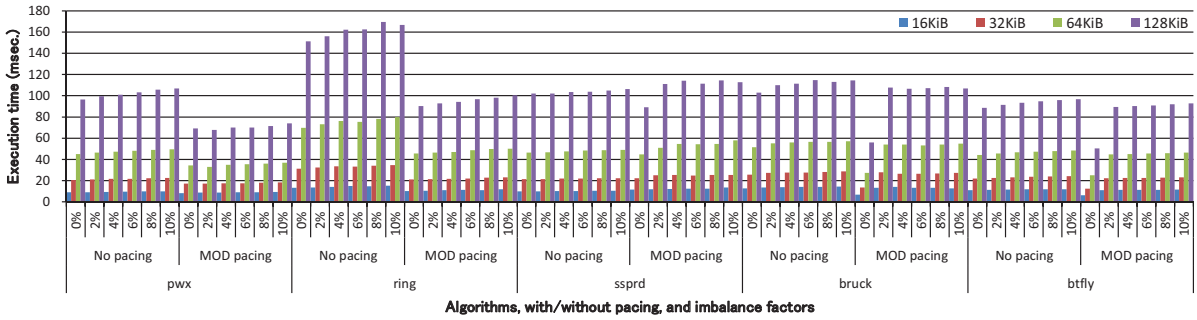
3 次元ならびに 2 次元トーラス網を対象に，様々なインバランスを集団通信に与えた結果，集団通信のアルゴリズムによってインバランスの感受性が異なることがわかった．また，MOD ペーシングを適用した集団通信の実行性能について評価した結果，MOD ペーシングの有効性を確認した．そして，メッセージサイズやノード数が増加すると概ねペーシングの効果が向上するわかった．これは，大規模システムにおけるパケットペーシングの有用性を示すものと考えられる．さらに，MOD ペーシングを用いた集団通信に各種のインバランスを加え，インバランスがペーシングを適用した集団通信に及ぼす影響を評価した結果，



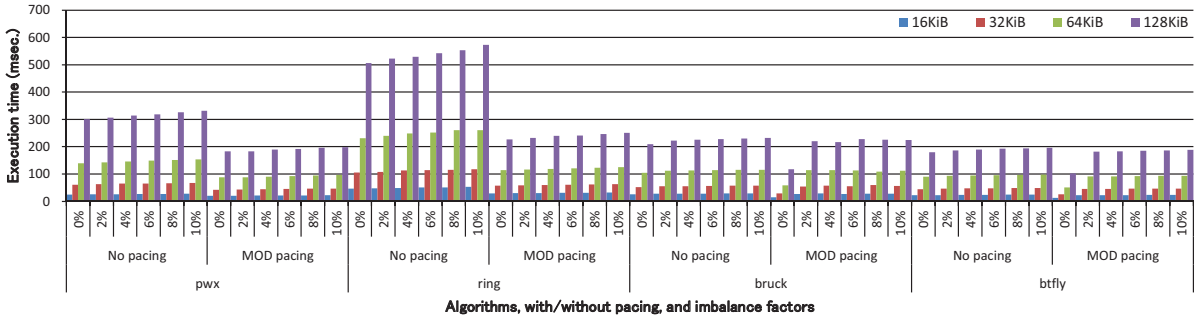
(a) 3 次元トーラス網，128 ノード（8×4×4）



(b) 3 次元トーラス網，256 ノード（8×8×4）



(c) 3 次元トーラス網，512 ノード（8×8×8）



(d) 3 次元トーラス網，1024 ノード（16×8×8）

図 5 インバランス挿入とパケットペーシングによる各アルゴリズムの実行時間（3 次元トーラス網）

Fig. 5 Execution times with imbalance factor and packet pacing for each algorithm (3D tori).

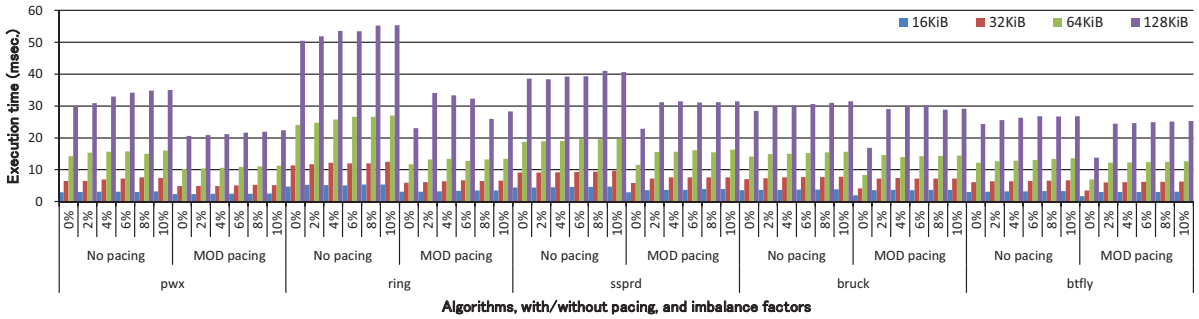
アルゴリズムによってはインバランスが加わることで実行時間が大幅に増加し，ペーシングの効果を損なう場合があることが明らかになった，

今後は，これらの現象について詳細解析を進めるとともに，さらにノード規模を大きくした場合について評価する．また，OS ジッタを踏まえた評価や高次元ネットワークを対象とした実験を行う予定である．

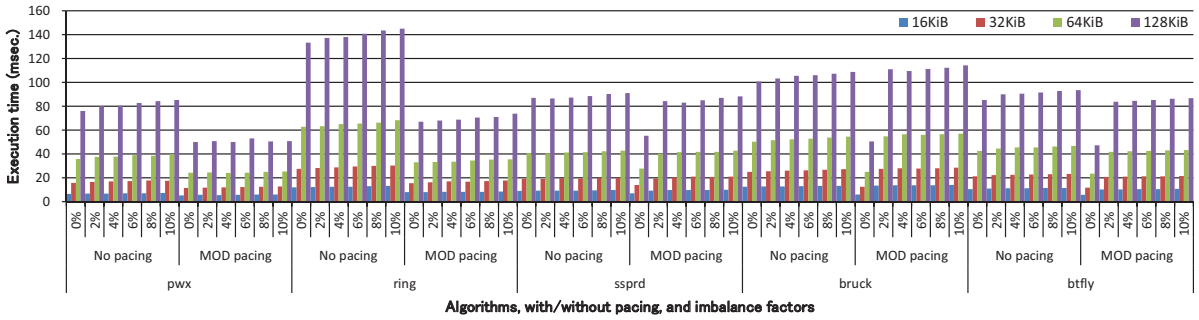
謝辞 本研究は，科学技術振興機構（JST）戦略的創造研究推進事業（CREST）における研究領域「ポストペタスケール高性能計算に資するシステムソフトウェア技術の創出」研究課題「省メモリ技術と動的最適化技術によるスケーラブル通信ライブラリの開発」によるものである．実験結果の一部は，九州大学情報基盤研究開発センターの研究用計算機システムを用いて取得したことを付記する．

参考文献

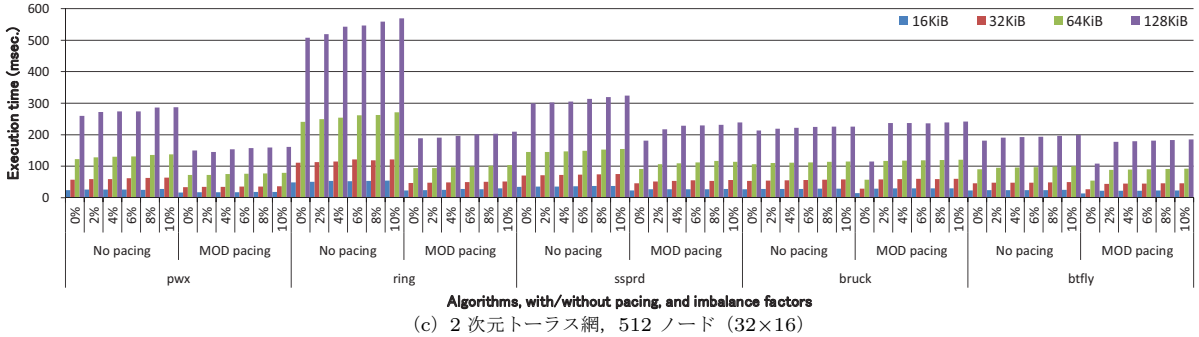
[1] 柴村 英智，三輪 英樹，薄田 竜太郎，平尾 智也，安島 雄一郎，三吉 郁夫，清水俊幸，石畑 宏明，井上 弘士：パケットペーシングによる全対全通信の最適化とシミュレーション評価，情報処理学会論文誌：コンピューティングシステム，Vol.4，No.3，pp.56–65，2011.



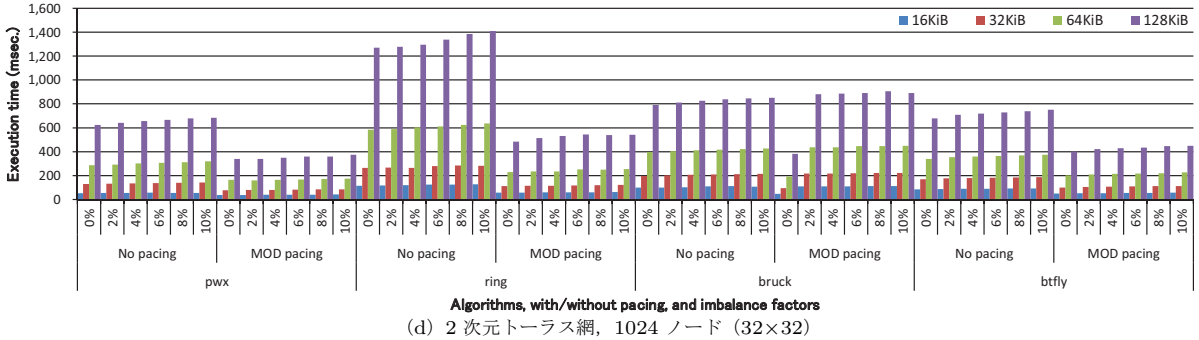
(a) 2 次元トーラス網，128 ノード（16×8）



(b) 2 次元トーラス網，256 ノード（16×16）



(c) 2 次元トーラス網，512 ノード（32×16）



(d) 2 次元トーラス網，1024 ノード（32×32）

図 6 インバランス挿入とパケットペーシングによる各アルゴリズムの実行時間（2 次元トーラス網）

Fig. 6 Execution times with imbalance factor and packet pacing for each algorithm (2D tori).

[2] 柴村 英智, 薄田 竜太郎, 三輪 英樹, 三吉 郁夫, 井上 弘士: 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, 情報処理学会研究報告, Vol.2011-HPC-130 (SWoPP2011)，pp.1–9，2011.

[3] A. Faraj, P. Patarasuk, and X. Yuan: A study of process arrival patterns for mpi collective operations, Proc. 21th ACM Intl. Conf. on Supercomputing (ICS07), 2007.

[4] T. Toyoshima: ICC: An interconnect controller for the tofu interconnect architecture, A Symposiumu on High Performance Chips (Hot Chips 24), 2010.

[5] 吉田 匡兵，柴村 英智，井上 弘士，村上 和彰：全対全通信向けパケットペーシングにおける送信間隔の導出手法，情報処理学会第 74 回全国大会，1K-6，2011.

[6] H. Miwa, R. Susukita, H. Shibamura, T. Hirao, J. Maki, M. Yoshida, T. Kando, Y. Ajima, I. Miyoshi, T. Shimizu, Y. Oinaga, H. Ando, Y. Inadomi, K. Inoue, M. Aoyagi, and K. Murakami: NSIM: An interconnection network simulator for extreme-scale parallel computers, IEICE Trans. Inf.&Syst., Vol.E94-D, No.12, pp.2298–2308, 2011.

[7] J. Bruck, C. Ho, S. Kipnis, E. Upfal, and D. Weathersby: Efficient algorithms for all-to-all communications in multiport message-passing systems, IEEE Trans. Parallel and Distributed Systems, Vol.8, No.11, pp.1143–1156, 1997.

[8] MPICH2: High-performance and Widely Portable MPI, http://www.mcs.anl.gov/research/projects/mpich2/.

[9] OpenMPI: Open Source High Performance Computing, http://www.open-mpi.org/.