Vol.2013-HPC-141 No.14
2013/10/1

FX10におけるパケットペーシングを用いたアプリケーションの通信性能評価

柴村 英智[1,2,a)]

概要：パケットペーシングを適用したアプリケーションの通信性能について，実システムで調査した結果を報告する．メッセージ通信時にパケットの送出間隔を積極的に制御し，通信効率を改善するパケットペーシング技術について，その効果や課題が多くのシミュレーション評価によって明らかになってきた．本研究では，実機でのパケットペーシングの有効性を実証することを目的とし，PRIMEHPC FX10 においてパケットペーシングを施したアプリケーションの通信性能を調査した．その結果，これまでのシミュレーション評価で認められてきた，パケットペーシングの有効性，ならびにメッセージ長やノード数に応じたパケットペーシング効果の向上が確認された．

キーワード：インターコネクト，FX10，パケットペーシング，シミュレーション，NSIM

Performance Evaluation of Communication Applications using Packet Pacing on Fujitsu FX10

Hidetomo Shibamura[1,2,a)]

Abstract: This paper presents a performance evaluation of communication applications using packet pacing on a real system. Packet pacing technique improves communication performance by controlling packet injection interval aggressively. The effectiveness and problems about packet pacing have been becoming clear through many simulation evaluations. In this study, to confirm the effectiveness of packet pacing in the real machine, communication applications using packet pacing were examined on a PRIMEHPC FX10 system. Then, some results which were already recognized in the past simulation: the effectiveness of packet pacing, and the improvement of the effectiveness corresponding to message length and/or node size, were demonstrated.

Keywords: Interconnect, FX10, Packet pacing, Simulation, NSIM

1. はじめに

メッセージ通信時にパケット送出間隔を積極的に制御することで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，各種の集団通信に対して適切なパケット送出間隔でペーシングを行うことにより通信の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きた．そして，それらの結果からパケットペーシングについて，以下の効果や課題が明らかになってきた．

(1) **通信アルゴリズムに応じたペーシング効果**
パケットペーシングによる通信性能の向上は，通信アルゴリズムすなわち通信パターン毎に異なる．アルゴリズムによっては，ほぼ 100%の通信帯域を利用することも可能である [1].

(2) **メッセージ長やノード数によるペーシング効果の増加**
メッセージ長（通信パケット数）やノード数（ホップ

[1] 公益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
[a)] shibamura@isit.or.jp

数）が増加するにつれて，ペーシングの効果も増加する [2]．したがって，将来の大規模インターコネクトへの活用が期待できる．

(3) **インバランスへの感受性**
通信の高速化を図るために集団通信にパケットペーシングを適用しても，通信開始時刻のインバランスや集団通信のアルゴリズムによって通信性能が大きく変化したり，場合によってはペーシングの効果をスポイルしてしまうこともある [3]．

一方，パケットペーシングを実機で活用するためには，これまでのシミュレーションによる性能評価から実機による評価へと展開し，パケットペーシングの有効性を実証しなければならない．また，インバランスをはじめとする実システムにおける課題も明確にする必要がある．

本研究では，実機におけるパケットペーシングの有効性を実証することを目的とし，既存の HPC システムにおける評価実験を行う．具体的には，パケットの送出間隔を制御できる富士通社製「PRIMEHPC FX10」（以下，FX10）を利用して，ランダムリング通信と全対全通信にパケットペーシングを適用した場合の通信性能を調査する．ランダムリング通信では，実機におけるパケットペーシングの効果を実証するとともに，メッセージ長やノード数が増加した場合におけるペーシング効果の向上について確認する．また，全対全通信では，FX10 のインターコネクトである Tofu を駆動する専用ライブラリを利用し，FX10 での実機評価，ならびにインターコネクトシミュレータ NSIM による評価結果との比較を行う．なお，本研究ではパケットペーシングを適用するプログラムは MPI で記述されているものとする．

以下，2 章では，本研究で前提とするパケットペーシングについて述べる．3 章では，実機で実行させる評価アプリケーションについて説明する．4 章では，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し，5 章でまとめる．

2. パケットペーシング

2.1 パケット転送間隔を変更可能なシステム

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）からネットワークに対して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の転送間隔時間を変更できる機能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]．

2.2 パケット間ギャップの設定

FX10 において，メッセージ通信に対してパケット間ギャップを設定する方法には 2 種類ある．

一つは，MPI ライブラリ内部の変数（MCA パラメータ）の値を一時的に変更する方法である．これは，MAC パラメータの一つである `common_tofu_packet_gap` に，1 パケットの転送にかかる時間を 8 とした場合の比率を設定する [5]．例えば，このパラメータに 16 を設定すると，先行するパケットが送信された後，2 パケット分（$16 \div 2$）の転送間隔を空けて次のパケットが送信される．したがって，メッセージの通信帯域は 1/3 に減少するが，他のパケットが転送される機会が増し，ネットワーク全体での通信効率が向上する可能性が高まる．また，このパラメータに 0 を指定した場合はパケットペーシングは行われない．なお，この方法はプログラムの実行開始時にのみ設定可能であり，プログラム全体でユニークなパケット間ギャップ値しか与えられない．すなわち，プログラム実行中の変更やメッセージ通信毎の変更はできない．

もう一つの方法は，Tofu インターコネクト向け低レベル通信機構を実装した Tofu ライブラリ [6] の使用である．このライブラリは，RDAM 通信によって Tofu インターコネクトを駆動させる通信 API を提供しており，本研究ではユーザレベル通信の一つであるワンサイド通信を使用し，パケット転送間隔を指定することで評価実験を行った．本手法では，前述の MPI ライブラリとは異なり，プログラム実行中やメッセージ通信毎にパケット間ギャップを設定することができ，積極的なパケットペーシングの制御が可能となる．

3. 評価アプリケーション

通信アプリケーションに対するパケットペーシングの効果を明らかにするために，次の 2 つの通信パターンについて評価実験を行う．

(1) ランダムリング通信
(2) 全対全通信

3.1 ランダムリング通信

ランダムリング通信は，HPC チャレンジベンチマーク [7] の通信性能測定である `b_eff` のうち，ランダムに選出した

ランクの並びでリングを形成し，隣接するプロセス同士で1対1通信を行うもの（random ordered）である．

本実験では，前述のMPIライブラリによるランダム通信プログラムを用い，MCAパラメータによってパケット間ギャップの設定変更を行う．まず，様々なメッセージ長におけるランダムリング通信について，パケット間ギャップ値を0から増加させながら利用可能な全プロセスでの実行時間を測定する．そして，ギャップ値=0（ペーシング無し）の実行時間に対して正規化したものを“ペーシング効果”とし，評価指標とする．また，ノード数を増加させながら同様の実験を行うことで，メッセージ長やノード数によるペーシング効果を評価する．

3.2 全対全通信

全対全通信を実現する様々な通信アルゴリズムがあるが，本実験ではMPICH[8]やOpenMPI[9]などの多くのMPIライブラリに実装されているpairwise exchangeアルゴリズムを用いる．

Tofuライブラリによる全対全通信プログラムを本実験では用いる．これは，MPIライブラリによる通信時の通信モードや通信方式の切替を取り除くためである．具体的には，FX10では大規模システムにおいて全体の通信性能を大きく損なわずに最適な通信を実現するために，通信時に高速型通信モードと省メモリ型通信モードの切替が適宜行われる．また，メッセージ長やホップ数に応じて，Eager通信方式とRendezvous通信方式の切替も行われる．したがって，これらの切替に起因する通信挙動の変化やオーバヘッドがパケットペーシングに影響を与える可能性がある．そこで，これらの切替を排除するために，Tofuライブラリで利用できるシンプルなワンサイド通信による全対全通信を行う．

ランダムリング通信と同様に，様々なメッセージ長やノード数についてパケット間ギャップを0から増加させながら実行時間を測定し，ペーシング効果を評価する．

4. 評価実験

4.1 ランダムリング通信

4.1.1 実行内容

MPIライブラリによるランダムリング通信プログラムを，九州大学情報基盤研究開発センターのFX10で実行した．実行時のパラメータを**表 1**に示す．本実験ではペーシングの効果を十分に確認できるよう，1ノードあたり16プロセスを割り当て，ネットワークへの負荷を高めている．

4.1.2 実行結果

図 1 (a)〜(f)に，ランダムリング通信におけるパケットペーシングの効果をノード数毎に示す．グラフの横軸はパケット間ギャップを表し，縦軸はペーシング無し（ギャップ値=0）における実行時間を1として，各ギャップ値における実行時間を正規化した値（ペーシング効果）を表す．なお，図中のグラフが下がるほどパケットペーシングの性能が良いことを示す．

表 1 ランダムリング通信の実行パラメータ

パラメータ	設定値
ノード数	24, 48, 96, 192, 384, 768
プロセス数	384, 768, 1536, 3072, 6144, 12288
パケット間ギャップ	0.0（ペーシング無）〜16.0
メッセージ長	16KiB〜4MiB

表 2 全対全通信の実行パラメータ

パラメータ	設定値
ノード数	64/768
プロセス数	64
パケット間ギャップ	0.0（ペーシング無）〜2.0
メッセージ長	512KiB，1MiB，2MiB
XYZ各軸のノードサイズ	4×2×8, 2×4×8, 4×8×2, 2×8×4, 8×4×2, 8×2×4

まず，図1（a）で示される，24ノードにおけるランダムリング通信に着目する．パケット間ギャップが0から増加するにつれて，パケットペーシングの効果が向上（グラフが下がる）していることが確認できる．そして，最もペーシング効果が高いポイント付近以降は急激に効果が低く（グラフが立ち上がる）なり始めている．すなわち，実行時の状況において最適なペーシングポイントを中心に，V字型のグラフになっている．

48ノード（図1（b））通信においても同様の傾向が確認でき，24ノード通信と比較すると，全体的にペーシング効果が高まっていることがわかる．加えて，メッセージ長の増加によっても，ペーシング効果が向上している．

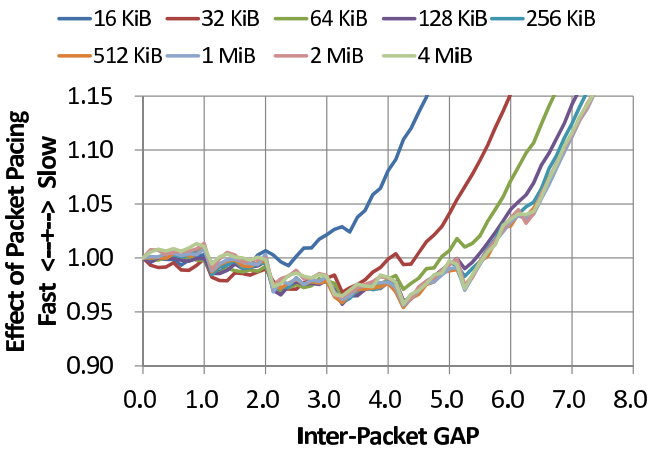
さらに，ノード数が768ノード（図1（f））まで増加した場合も同様の傾向を確認することができる．これらの結果から，実機おけるパケットペーシングの有効性が実証されたといえる．また，メッセージ長やノード数の増加に応じたパケットペーシング効果の向上も確認できた．

4.2 全対全通信

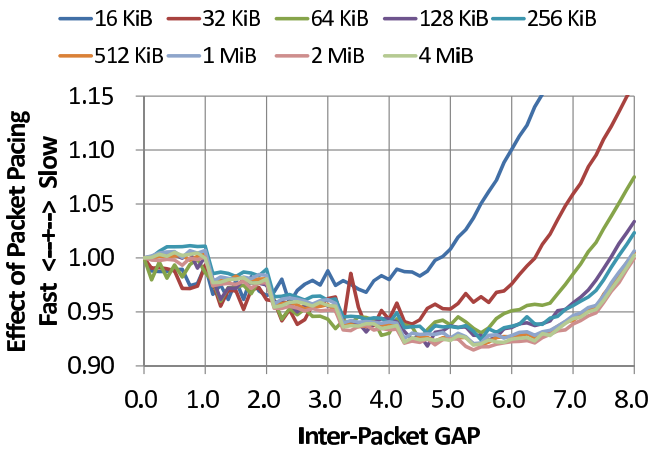
4.2.1 実行内容

Tofuライブラリによる全対全通信プログラムをFX10で実行した．実行時のパラメータを**表 2**に示す．なお，本実験では利用可能な全768ノードのうち，Tofuインターコネクトで物理的に3次元トーラス網を構成できる64ノードのみを使用し，1ノードあたり1プロセスを割り当てた．これは以下の理由によるものである．

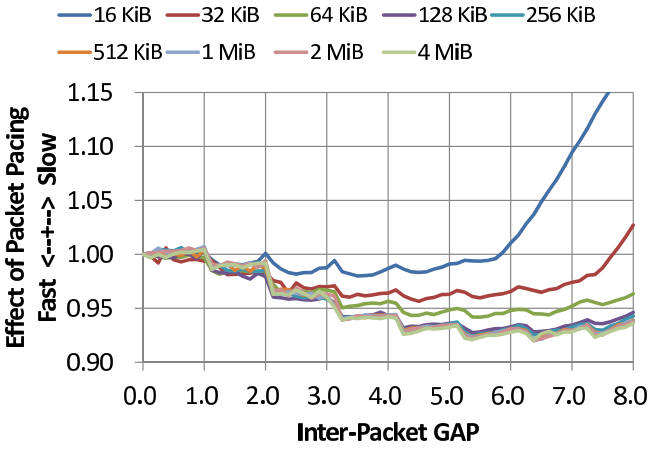
FX10ではジョブ投入時に論理的に1次元から3次元までのトーラス網を構成するようプロセス位置の形状（トポロジ）を指定できるが，実行時には6次元Tofuの物理座標（X, Y, Z, A, B, C）に割り当てられる．この際に，ト



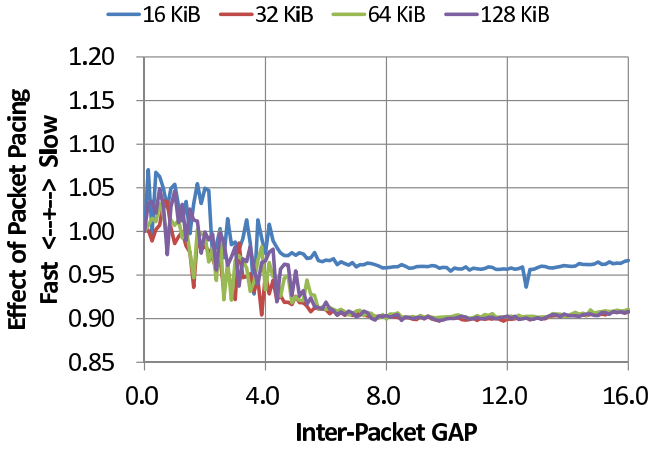
(a) 24 ノード (384 プロセス)



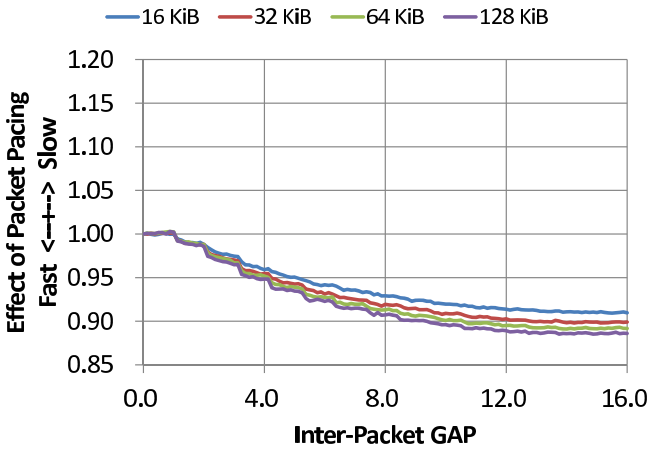
(b) 48 ノード (768 プロセス)



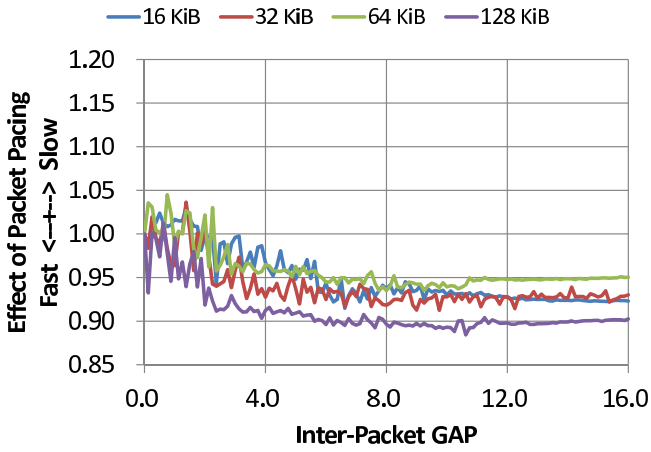
(c) 96 ノード (1,536 プロセス)



(d) 192 ノード (3,072 プロセス)



(e) 384 ノード (6,144 プロセス)



(f) 768 ノード (12,288 プロセス)

図 1 ランダムリング通信におけるパケットペーシング効果

ポロジの各次元数によっては次元軸がたたみ込まれ，他の次元軸を使ったショートカット経路が利用可能となる．その結果，論理トポロジで想定したホップ数よりも短くなる場合がある．

本研究ではパケットペーシングの有効性を確認することが目的であるため，このような物理ノードへのプロセスの配置具合によって，ホップ数が変化することは好ましくない．そこで，X，Y，Z の 3 軸のみでプロセス間通信を行う 3 次元トーラス網を構成するように，物理座標に配置されたノード中からプロセスを選択する．

具体的には，まず，ジョブ実行時に全 768 ノードを取得する．本実験環境では 8×6×16 の論理 3 次元トーラスが最大構成となる．次に，これらの物理ノードの座標で A，B，C の 3 軸がそれぞれ 0 となるノードを選ぶ．すなわち，ここで選ばれたプロセスが配置されるノードの物理座標は (X, Y, Z, 0, 0, 0) となる．以後，選ばれたノードに配置さ

れたプロセス間でのみ全対全通信を実行する．なお，本実験環境では 4×2×8 の 64 ノード構成を持つ 3 次元トーラス網となる．

また，3 次元トーラス網の各次元サイズを入れ替えた場合のペーシング効果を評価するために，64 ノードの各プロセスが持つランクとは別個に，新たな仮想ランクを割り付けた．具体的には，各次元軸の物理ノード数が 4×2×8 のトーラス網において，論理座標 X，Y，Z が（0，0，0）となるプロセス位置を定め，そのプロセスからは，4×2×8，2×4×8，4×8×2，2×8×4，8×4×2，あるいは 8×2×4 のトーラス網となるように 6 種類の仮想ランクの割り付けを行った．

4.2.2 実行結果

図 2 (a)〜(f) に，全対全通信におけるパケットペーシングの効果を，仮想ランクの割り付けの方針毎に示す．グラフの横軸はパケット間ギャップを表し，縦軸はペーシング効果を表す．なお，グラフが下がるほどペーシングの性能が良い．

図 2 の（e）と（f）のグラフから，これら 2 つの仮想ランク割り付けについては，若干のペーシング効果があるものの，（a）から（d）の 4 つ割り付けにおいては，有効なペーシング効果が認められない．これは以下の理由によるものである．

プロセスに仮想ランクを割り付ける際は論理軸 X，Y，Z の並びを変えているが，物理トーラス網のトポロジは 4×2×8 と固定である．また，FX10 のルーティングは次元順による決定的ルーティングであるため，常に X，Y，Z の順で各座標軸を経由する．ここで，物理 X 軸のサイズは 4，物理 Y 軸のサイズは 2 と小さいため，パケットペーシングの効果が全く出ない．したがって，仮想ランクの割り付けの際に X 軸や Y 軸から先に割り付けた場合には，ペーシング効果が現れない．

一方，図 2（e）と（f）のように Z 軸から割り付けた場合，物理 Z 軸のサイズは 8 であるが，pairwise exchange の通信パターンでは，メッセージが衝突するパターンが少なく，衝突が発生する場合でも高々 2 ホップ通信によるものであるため，パケット間ギャップ値を 1 としたペーシングしか効果が出ない．よって，このペーシング効果による利得も他の通信時間で平均化され，図 2（e）や（f）のようにパケット間ギャップが 1 よりも小さいポイントで通信時間が速くなっているといえる．

4.2.3 NSIM によるシミュレーション評価

前述の現象がシミュレーションでも発生するか確認するために，インターコネクトシミュレータ NSIM[10] を利用して，4×2×8 の 64 ノード 3 次元トーラス網における全対全通信を評価した．

トーラス網の諸仕様や通信性能を FX10 と同等にし，正確なシミュレーションを行うためには，NSIM に与えるパラメータを適切に設定しなければならない．本実験では，Tofu インターコネクトの基本性能は，文献 [11], [12], [13] に基づいて設定した．また，実システムでの全対全通信は Tofu ライブラリを利用しているため，MPI オーバヘッドや通信ライブラリに関わる NSIM の設定項目については，Tofu ライブラリを用いた 1 対 1 通信を別途実行し，その結果から得られた測定値を元に算出し設定ファイルを較正した．**表 3** に主要な NSIM の設定を示す．

表 3 NSIM コンフィグレーション

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	Round Robin
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	128B
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	5GB/s（単方向）
ルーティング計算時間 (RC)	3.2ns
仮想チャネル設定時間 (VA)	3.2ns
スイッチ設定時間 (SA)	3.2ns
フリット転送時間 (ST)	3.2ns

全対全通信プログラムについては，FX10 で実行した主要通信部分を NSIM の MGEN プログラムで実行させた．なお，実機では各プロセスの開始時刻にインバランスがあるため，MGEN プログラムにおいて数ナノ秒のインバランスを発生させ，シミュレーション時刻に加えた．

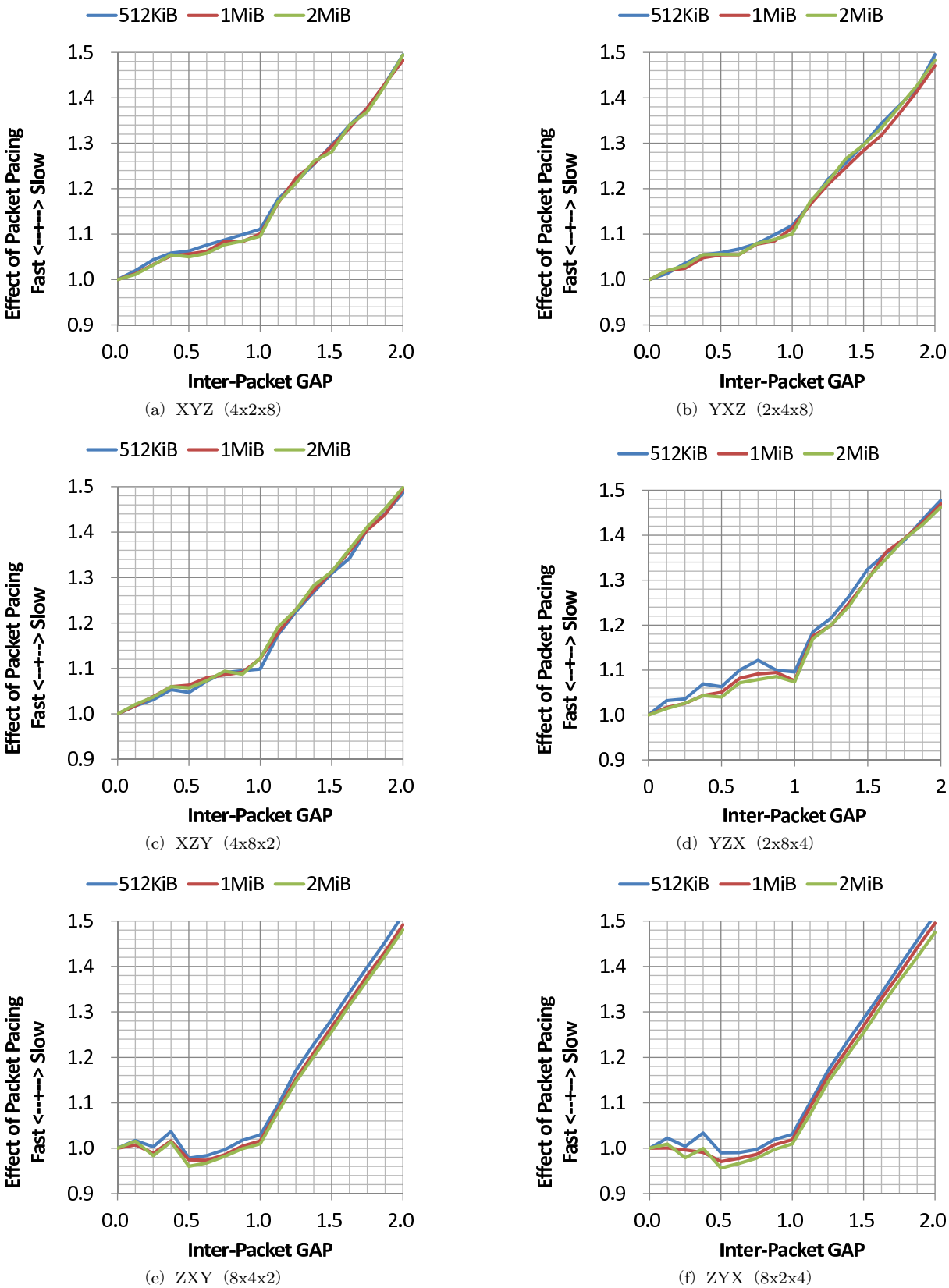
メッセージサイズが 1MiB の全対全通信を NSIM でシミュレーションした結果を**図 3** に示す．先の 2 つの評価実験と同様に，グラフの横軸はパケット間ギャップを，縦軸はペーシング効果を表す．

このグラフから，NSIM でのシミュレーション評価でもパケット間ギャップが 1 よりも小さい部分においてペーシング効果が現れており，FX10 での実機評価とほぼ同様の傾向になっていることがわかる．

以上の結果から，4×2×8 の 64 ノード 3 次元トーラス網ではノード数が小さく，各次元でパケットペーシングが作用する十分なホップ数がないため，パケットペーシングの効果が出なかったと考えられる．

5. まとめ

メッセージ通信時にパケットの送出間隔を積極的に制御し，通信効率を改善するパケットペーシング技術について，実機での有効性を実証することを目的とし，FX10 での評価実験を行った．その結果，これまでのシミュレーション評価で認められてきたパケットペーシングの有効性をはじ



(a) XYZ (4x2x8)　(b) YXZ (2x4x8)

(c) XZY (4x8x2)　(d) YZX (2x8x4)

(e) ZXY (8x4x2)　(f) ZYX (8x2x4)

図 2　ランダムリング通信におけるパケットペーシング効果

め，メッセージ長やノード数に応じたペーシング効果の向上が実際に実システム上で確認された．

64 ノードの 3 次元トーラスにおける全対全通信を評価したが，ノード規模が小さすぎたため十分なパケットペーシングの効果を発揮することができなかった．しかし，ノード数に応じたペーシング効果の向上が確認できたため，パケットペーシング技術は今後のポストペタスケール時代におけるインターコネクトの基盤技術に成り得ると考える．

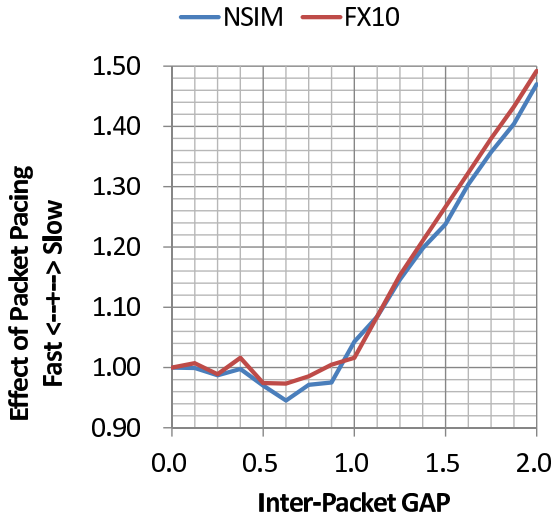


図 3 NSIM と FX10 における全対全通信のペーシング効果の比較（3 次元トーラス網：4×2×8，メッセージサイズ：1MiB）

今後は，プログラムの実行時にホップ数に応じてパケット間ギャップを動的に変える MOD ペーシング [3] について，実機上での性能評価を行う．また，今回よりもさらに大規模なノード数を持つシステムでの性能評価についても行う予定である．

謝辞 本研究を進めるにあたり日頃からご協力いただく富士通株式会社 住元真司氏，安島雄一郎氏，秋元秀行氏，三浦健一氏に感謝する．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.
[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] 柴村英智，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に対するロード／ネットワークインバランスの影響，情報処理学会研究報告，Vol.2012-HPC-133 (SWoPP2012)，pp.1–8，2012.
[4] T. Toyoshima: ICC: An interconnect controller for the tofu interconnect architecture, A Symposiumu on High Performance Chips (Hot Chips 24), 2010.
[5] Technical Computing Suite V1.0 –MPI 仕様手引書（PRIMEHPC FX10 用）–，富士通株式会社，2012.
[6] 志田直之，住元真司，宇野篤也：スーパーコンピュータ「京」の MPI と低レベル通信，FUJITSU，Vol.63，No.3，pp.299–304，2012.
[7] HPC Challenge Benchmark: http://icl.cs.utk.edu/hpcc/
[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/.
[10] 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.
[11] Y. Ajima, S. Sumimoto, and T. Shimizu: Tofu: A 6D Mesh/Torus Interconnect for Exascale Computers, Computer, Vol.42, No.11, pp.36–40, 2009.
[12] Y. Ajima, T. Inoue, S. Hiramoto, T. Shimizu, and Y. Takagi: The Tofu Interconnect, IEEE Micro, Vol.32, No.1, pp.21–31, 2012.
[13] 安島雄一郎，井上智宏，平本新哉，清水俊幸：スーパーコンピュータ「京」のインターコネクト Tofu，FUJITSU，Vol.63，No.3．pp.260-264，2012.