14-IS01

シミュレーションによる大規模並列プログラムへのパケットペーシングの適用と有効性の検証

柴村英智（公益財団法人九州先端科学技術研究所）

概要　実践的な大規模並列プログラムの通信部分に対して，アプリケーション毎に適切な間隔でパケットを送出するパケットペーシングを適用することでネットワークの輻輳を抑制し，全体の通信時間を短縮できることをシミュレーションによって明らかにする．本研究では，全球雲解像モデル NICAM の主要通信部分についてインターコネクト・シミュレーションを通じた輻輳状況の可視化・解析を行った．その結果，多くのメッセージが通過する通信リンクを発端として輻輳が発生することが明らかになった．そこで，パケットの送出間隔を適切に制御するパケットペーシングを適用することによって，ほぼ完全に輻輳の発生を抑制することができ，理論計算値に近い通信性能を達成できる可能性があること確認した．

1.　共同研究に関する情報

(1) 共同研究を実施した拠点名

九州大学 情報基盤研究開発センター

(2) 共同研究分野

□　超大規模数値計算系応用分野

□　超大規模データ処理系応用分野

□　超大容量ネットワーク技術分野

■　超大規模情報システム関連研究分野

(3) 参加研究者の役割分担

柴村英智【九州先端科学技術研究所】

①実践的な大規模並列プログラムの選定・取得

②シミュレーション評価に向けた準備

③大規模インターコネクト･シミュレーションの実施

④シミュレーション結果の解析・評価

南里豪志【九州大学】

①実践的な大規模並列プログラムの選定・取得

③大規模インターコネクト･シミュレーションの実施

④シミュレーション結果の解析・評価

眞木淳【九州先端科学技術研究所】

②シミュレーション評価に向けた準備

④シミュレーション結果の解析・評価

2.　研究の目的と意義

研究の目的

実践的な大規模並列プログラムの通信部分に対してパケットペーシング（後述）を適用し，通信時間を短縮できることをシミュレーションによって明らかにすることである．これは，通信全体の高速化を図ることによってプログラム全体の実行時間を短縮することにつながる．

パケットペーシングとは，通常は切れ目無く連続して送出されるパケットを所定の間隔（パケット間ギャップ）を空けて断続的に送出することである．通信経路上で異なるメッセージのパケットが衝突する場合に，パケットペーシングを適用すると他方のパケット間ギャップのタイミングで自パケットの転送を行うことができ，通信衝突が解消された円滑な通信となる．その結果，ルータにおけるパケットの待ち時間や転送再開までの遅延が大幅に無くなり，通信全体の高速化が達成される．

このようなパケットペーシング技術の研究として，これまで申込者らは全対全通信をはじめとする集団通信を評価対象とし，シミュレーション評価や実機での検証実験を経てパケットペーシングの有効性を実証してきた．本研究では，評価対象を主流のスーパーコンピュータ上で走行されるアプリケーションと定め，特に実用的な大規模並列プログラムに対してシミュレーションによる通信ボトルネックの解析を行い，パケットペーシングを施した際の通信の高速化，延いてはプログラム全体の高速化の可能性を探る．

研究の意義

これまでのパケットペーシングに関する研究で得られた知見の中で注目すべき点は，通信衝突の頻度が高いほど，換言すると，通信メッセージサイズが大きく通信ホップが長くなるほどパケットペーシングの効果が増すことである．これは大量の計算ノードを利用する近年の大規模並列プログラムに対して通信高速化の観点から大きく貢献できることを意味する．

また，パケットペーシングはノード間通信のみならず他の通信ネットワークでの活用が考えられる．その一つは，メニーコア化が進むプロセッサ内のNoC(Network on Chip)への適用である．現在のノード間通信における問題と同様，コア数が増加するに従いコア間通信における遅延や通信衝突が問題になる日は近いであろう．もう一つのネットワークとして多数のストレージや外部入出力装置を接続する HPC 向けの I/O ネットワークへの適用も有効であると考える．

以上のように，パケットペーシングは，ポストペタ／エクサ時代の大規模通信における実用的な技術として幅広い応用が期待できる．本来パケットペーシングは，インターネットのような中長距離のオープンネットワークでの QoS を提供するための技術であるが，これをインターコネクトのような短距離かつクローズドなネットワークで積極的に活用する事例は課題代表者らの知る限り初めてであり，本研究が当該分野に果たす役割は大きい．

3. 当拠点公募型共同研究として実施した意義

実際の計算機運用に携わる拠点スタッフと連携することで,インターコネクトシミュレータ NSIM による大規模シミュレーションの実行に終始せず,シミュレーションで想定するインターコネクトの仕様設定やシステム構成に関する現実的なシミュレーションパラメータを決定するために綿密な情報交換ができる点に大きな意義がある．また，シミュレーション結果の分析についても，運用面からみた実践的な判断が期待できる．

4. 前年度までに得られた研究成果の概要

なし（※新規課題のため）

5. 今年度の研究成果の詳細

5.1 評価対象プログラムの MGEN プログラム化

本研究で評価対象とする実践的な大規模並列プログラムとして，全球雲解像モデル NICAM（http://fiber-miniapp.github.io/）を選定した．このコードから主要通信部を抽出し，インターコネクトシミュレータ NSIM に与える MGEN プログラムを作成した. NSIM の入出力ファイルを図 1 に示す．NSIM ではアプリケーションの振る舞いに従ったシミュレーションを行うために，MGEN プログラムと呼ぶMPI相当のCプログラム(図 2)を駆動させ，シミュレーション中にオンデマンドで通信イベントを生成している．したがって，実際のMPIプログラムの通信パターンに則したインターコネクトのシミュレーションができる．ただし，シミュレーションではメッセージデータの授受は行わないため，受信メッセージの内容に応じて通信パターンが変化するようなプログラムのシミュレーションについては対象外としている．

現在，基本的な 1 対 1 の同期・非同期通信をはじめ，ランデブー通信，ゼロコピー通信をサポートしており，基本的な集団通信は MGEN プログラムとして別途提供している．また，インターコネクトの詳細仕様をインターコネクト・コンフィグレーションファイルと呼ぶ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．そして，MPI プロセスと物理ノードの対応を任意に与えられるように，ランク・ノード変換テーブルファイルを入力とする．

NSIMはシミュレーション終了後,性能情報ファイル，統計情報ファイル，通信履歴ファイルを出力する（図 3)．性能情報ファイルは，シミュレーションによって得られた評価対象インターコネクトの性能情報(MGENプログラムの予測実行時間，総転送パケット数，総転送データ量，実効バンド幅，リンクスループット，リンクビジー率など)

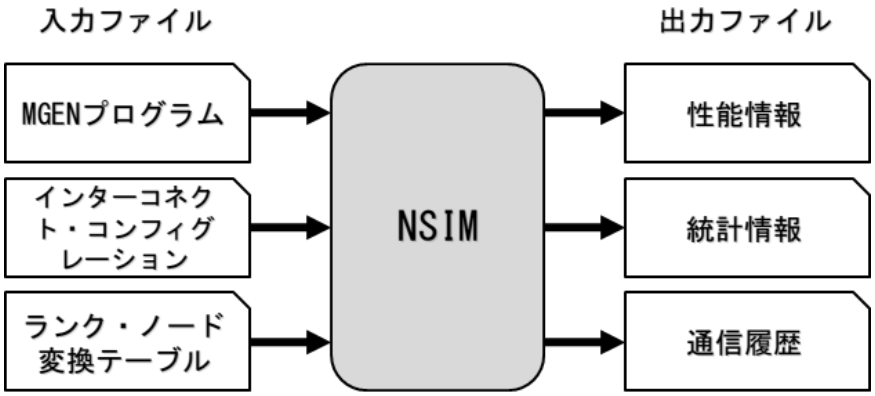


図 1　NSIM の入出力ファイル

```
#include <stdio.h>
#include "mgen.h"

int MGEN_Main( int argc, char *argv[] )
{
   int myrank, mysize, i, src, dst, msg_size=1024;
   MGEN_Request req[2];
   MGEN_Status stat[2];
   MGEN_Comm_rank(MGEN_COMM_WORLD, &myrank);
   MGEN_Comm_size(MGEN_COMM_WORLD, &mysize);   // mysize must be a power of 2

// Pairwise exchange
   for ( i=1; i<mysize; i++ ) {
      dst = src = myrank ^ i;
      MGEN_Irecv(NULL, msg_size, MGEN_BYTE, src, 0, MGEN_COMM_WORLD, &req[0]);
      MGEN_Isend(NULL, msg_size, MGEN_BYTE, dst, 0, MGEN_COMM_WORLD, &req[1]);
      MGEN_Comp(Computation overhead);
      MGEN_Wait(&req[1], &stat[1]);
      MGEN_Wait(&req[0], &stat[0]);
   }
   return (0);
}
```

図 2　MGEN プログラム例

を含む．また，統計情報ファイルは，リンクスループット，仮想チャネルバッファの利用率，通信レイテンシ，ネットワークレイテンシなどの詳細な統計情報である．そして，シミュレーションされた通信イベントの詳しい履歴を通信履歴ファイルに出力する．NSIM は並列離散事象シミュレーションモデルに基づき，MPI で実装しており，多くの並列処理プラットフォームで実行可能である．

NICAM では有限体積法による離散化を行い，水平方向の格子としては正 20 面体格子を用いる．通信に関しては幾つかの control volume をまとめて“region”とし，region と MPI ランク（プロセス）が 1 対 1 に対応する．

NICAM における正 20 面体格子の展開図を図 4 に示す．隣接する 2 つの面（3 角形）を併せ 1 つのダイヤとした 10 個のダイヤ（X=0～9）から成っており，最小規模の並列実行（10 プロセス）ではダイヤ自身が region となる．矢印はデータ分割における通信パターンを表す．

1 つの region は計算領域であり，さらに再帰的に 4 分割していくことで並列性を高める工夫をしている．この再帰の繰り返し回数は Rlevel-n と定義されている．したがって，各 region を図 4 のように 4 つのサブ region に分割し，再帰的に n 回分割を繰り返すと region の数は全体として 10×4^n となる．本研究では 2 種類の規模のシミュレー

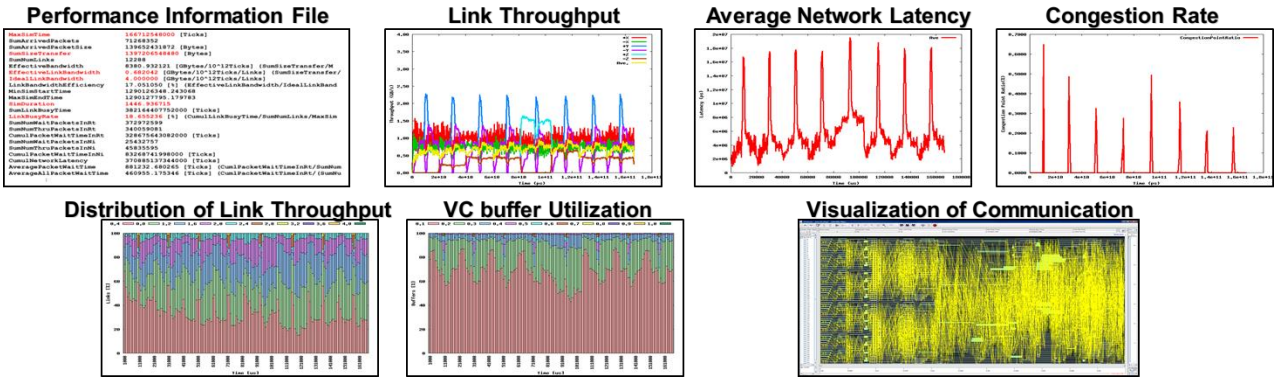


図 3　NSIM が出力する性能情報ファイル，統計情報ファイル，通信履歴ファイル

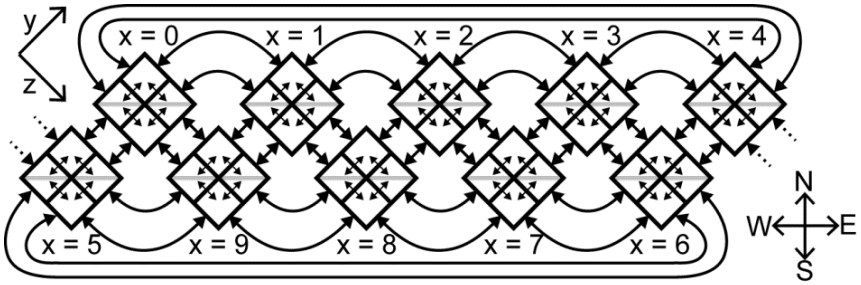


図 4　正 20 面体の展開と通信パターン（rlevel=1）

ションを実施した．一つは rlevel=2 とした小規模実行モデル（160 プロセス）であり，もう一つは rlevel=7 とした大規模実行モデル（163,840 プロセス）である．分割された正 20 面体格子内での region の位置を指定するため, 論理的な region 座標を（x, y, z）と表す．大規模実行モデルでは 4 プロセス／ノードでの実行とし，図 5 のように隣接する 2×2 の 4 プロセスを 1 ノードに割り当てた. ここで, ノード座標を指定するために(x, y', z') と表す.

NICAM の主要な通信は各 region と隣接する 4 つの region との間のデータ交換であり，これは対応するプロセス間の非同期な 1 対 1 通信となる．また，ノード内通信はノード間通信よりも速いため通信時間は隠蔽できるものとし, 本研究ではノード間通信のみを扱う.

以上のような NICAM の主要通信を模擬する通信パターンプログラム（MGEN プログラム）を作成した．この MGEN プログラムでは，NSIM の制約のため，ノード内の 4 プロセスを 1 プロセスで置き換え，4 方向への通信を 2 回ずつ行うことにより，4 プロセス分の通信をシミュレーションするものとした.

評価対象インターコネクトのトポロジは 3 次元トーラス網（X, Y, Z）とし，NICAM における論理ノード座標 (x, y', z') から物理ノード座標 (X, Y, Z）へのマッピング方法を図 6 に示す.

このマッピングにより，小規模実行モデルでは図 4 中のダイヤ内に加えてダイヤ間の通信の一部が隣接通信となる．それ以外のダイヤ間の通信経路は他の通信経路と重なり合うことになり，1 つのリンクを共有する通信経路の数（共有数）は最大で 3 となる．また，マッピングされるノードの形状は，10×2×2（論理ノード座標）から 5×4×2（物理ノード座標）となる（図 6（a））.

同様に，大規模実行モデルではノードの形状は 10×64×64（論理ノード座標）から 40×32×32（物理ノード座標）となり（図 6（b））, ダイヤ内の隣接通信の大部分はマッピング後も隣接通信として残る．それ以外の通信はトーラスリンクがある X 方向の面上を 1 ないしは 2 ホップ通過する.

北側のダイヤ同士および南側のダイヤ同士の一

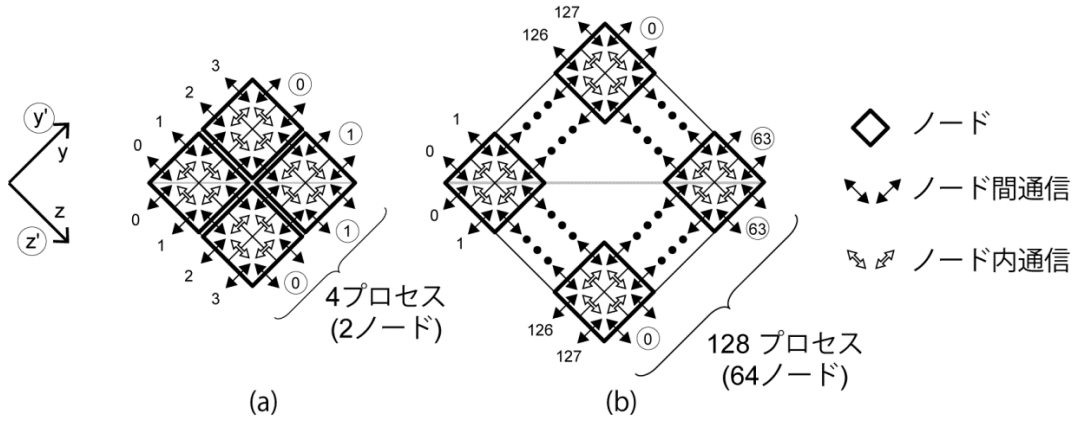


図 5　実行モデル毎のノード内通信とノード間通信
(a) 小規模実行モデル　(b) 大規模実行モデル

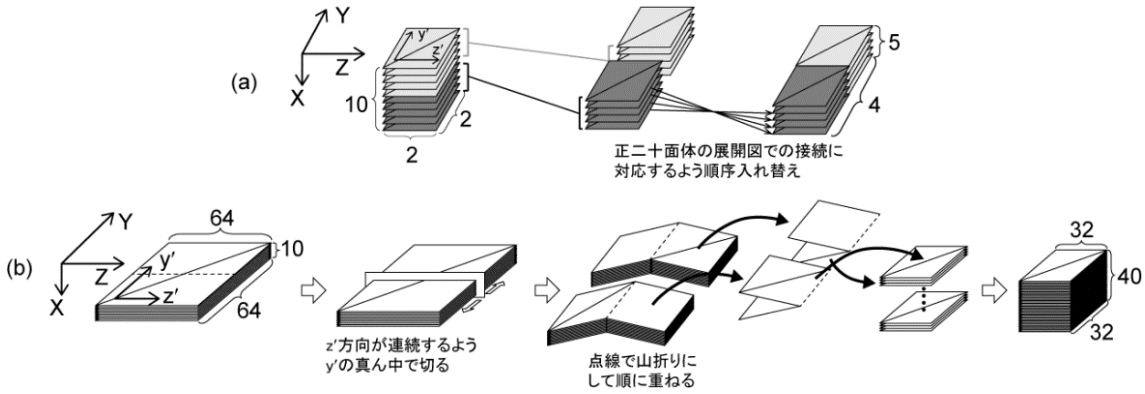


図 6　実行モデル毎のノードマッピング（概要）

部の通信では YZ 面上で多くの経路がダイヤ型の角に位置するノード（北側では Y=31, Z=0，南側では Y=0, Z=0）を通過し，それらが共有するリンクで共有数が 17 と最大になる．なお，どの経路でも X 方向へはトーラスリンクがある 4 つの面上（Y=0，Y=31 の 2 つの XZ 面上，ならびに Z=0, Z=31 の 2 つの XY 面上）だけを通過し，内部（0 < Y, Z < 31）では X 方向のデータ移動は存在しない．

5.2 パケットペーシングによる NICAM 通信の高速化

パケットペーシングは，通信時のパケット送出間隔を適切に設定することで，通信衝突を抑制し効率の良い通信を実現するものである．これまでの研究から，ホップ数やメッセージ長が大きくなるほど通信衝突を抑制する効果が高いことが明らかになっており，大規模インターコネクトへの活用が期待できる．一方で，通信パターン（アルゴリズム）によっては，通信開始時刻のばらつき（以下，通信開始時刻のインバランス）により，ペーシングが効かない場合もある．

本研究では，プロセス配置をノードの次元昇順にプロセスを配置するノーマルマップを基準とし，パケットペーシングを適用した場合について通信性能の推定を行った．また，実機で生じる通信開始時刻のインバランスがパケットペーシングにもたらす影響についても評価を行った．

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，

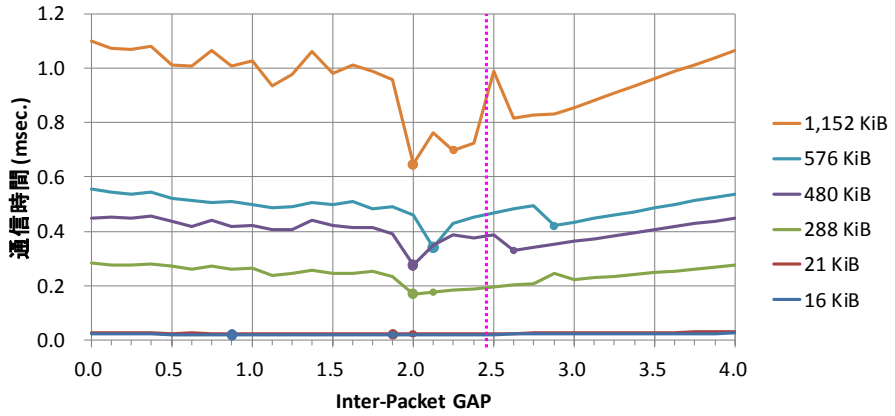


図 7　パケット間ギャップに対する通信時間の変化（小規模実行モデル）

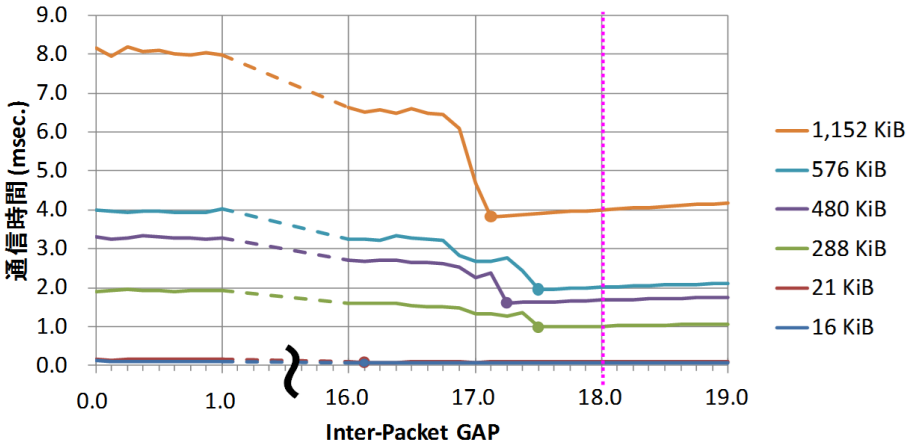


図 8　パケット間ギャップに対する通信時間の変化（大規模実行モデル）

パケット間ギャップ=n（ただし，n≧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 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．

NSIM は，MPI 通信に相当する関数群以外に，パケット間ギャップを明示的に設定しパケットペーシングをともなうメッセージ通信を行う，独自の低レベル通信関数を備えている．この低レベル通信関数を用いて評価対象アプリケーションにパケットペーシングを適用した通信性能推定を行う．

評価対象とするインターコネクトは 3 次元トーラス網とし，NICAM 通信の実行モデルは小規模実行モデルと大規模実行モデルとする．これらの実行モデルについて，パケット間ギャップ（パケットの送信間隔）を 0.0（パケットペーシング無し）から変化させながら，各データサイズにおける通信時間が最小となるギャップ値を調査した．

小規模実行モデルにおける推定通信時間を図 7 に示す．各データサイズともパケット間ギャップが大きくなるにつれて通信時間が徐々に減少していることがわかる．そして，パケット間ギャップが 2.0 付近において，各通信時間が最小になっている．ここで，各データサイズにおける最小通信時間から全通信時間を積算すると，パケットペーシングによって約 40%の通信時間を削減できることがわかった．さらにパケット間ギャップを大きくすると無駄な非通信時間が増えるため，通信時間はギャップ値に比例して増加している．

同様に，大規模実行モデルにおける推定通信時間を図 8 に示す．各データサイズともパケット間ギャップが増加するにしたがって徐々に通信時間が減少している．さらにパケット間ギャップが増加すると無駄な非通信時間が多くなるため全通信時間はギャップ値に比例して増加する．この増加傾向は小規模実行モデルのものと比較すると穏やかであるため，パケット間ギャップを数パケット分大きめに設定しても全通信時間は大きく変化しない．したがって，多数のノードを使う実行モデルでは，十分なパケットペーシングの効果を発揮できるといえる．なお，大規模実行モデルにおいて全通信時間を最小にするパケット間ギャップ値は 17.125 であった．

5.3 インバランスの影響調査

実際のシステムでは OS ジッタや負荷の不均衡によって通信開始時刻のインバランスが発生する．これまでのパケットペーシングに関する研究から，通信の高速化を図るため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通信アルゴリズムによっては通信性能が大きく変化し，場合によってはペーシングの効果を損なう可能性もあることがわかっている．そこで，実システムでの実行を想定した，NICAM 通信に対するインバランスの影響を調査した．具体的には，各プロセスでの通信開始前に様々な空白時間を加え，パケットペーシングを適用した場合の通信時間の変化を観測した．

通信開始前に加える空白時間は，プロセス毎にインバランス係数(f_{imb})を越えない値を乱数によって生成した．インバランス係数を 0 秒（インバランス無し），100 ナノ秒，1 マイクロ秒，10 マイクロ秒，100 マイクロ秒，1 ミリ秒とした場合の空白時間，ならびに，参考のためにバリア同期による空白時間を用いた．なお，バリア同期はソフトウェア実装とし，同期アルゴリズムには dissemination を用いた．

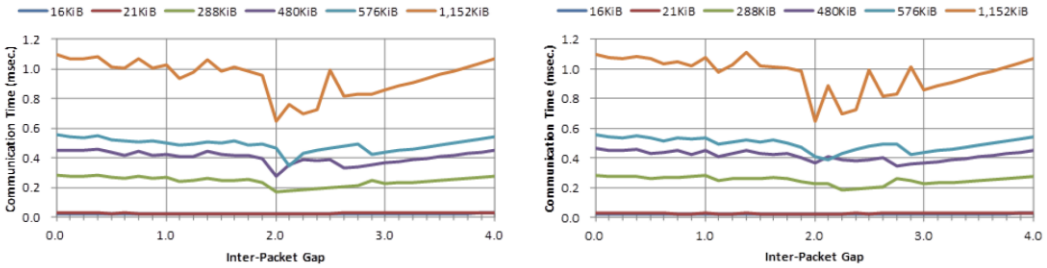
図 10（a）～（g）に，小規模実行モデルにおいてインバランス値を変化させた場合の通信時間の推移を示す．データサイズが 1,152KiB 程度と大きい場合は，引き続くパケット数が多くパケットペーシングの効果が効きやすいため，インバランス係数が 0 から 100 マイクロ秒まで大きく変化してもパケット間ギャップが 2.0 付近で最小の通信時間を保っている．しかし，データサイズが小さくなるにつれてインバランスの影響を受けやすくなり，最適なパケット間ギャップ値は増加傾向になっていることが確認できる．さらに，インバランス係数を 1 ミリ秒と大きくしすぎると，パケットペーシングの効果が全く出ていない．

一方，大規模実行モデルにおいてインバランス値を変化させた場合の通信時間の推移を図 9（a）～(g) に示す．小規模実行モデルと比較して，ノード規模が大きい大規模実行モデルではホップ数も大きいため，すべてのデータサイズについてインバランスに左右されず，十分なパケットペーシングの効果が発揮されている．

以上のことから，NICAM 通信では，ノード規模が小さい場合はインバランスの影響を受けやすいため最適なパケット間ギャップを適宜導出することは容易ではない．一方，ノード規模が大きい場合はインバランスの影響を受けにくく，理論的なパケット間ギャップから若干大きめの値を設定することで十分なパケットペーシングの効果を得ることができる．今後の課題として，この適切なパケット間ギャップ値を導出する手法の開発が重要である．

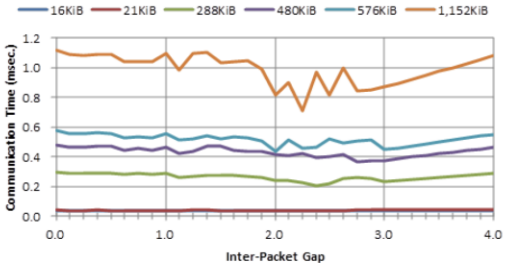
これまでの結果をまとめ，NICAM 通信についての通信時間の推定結果を整理する．

小規模実行モデルにおける推定結果を図 11 に示す．最左のバーは，机上計算による通信時間（理論値）である．残る 4 つのバーは NSIM によるシミュレーション結果によるものであり，パケットペーシングの有無，インバランスの有無により 4 種となる．それぞれのバーは，データサイズ毎の通信時間を積算しており，この図から全通信時間は 288KiB 以上の通信によって支配されていることがわかる．

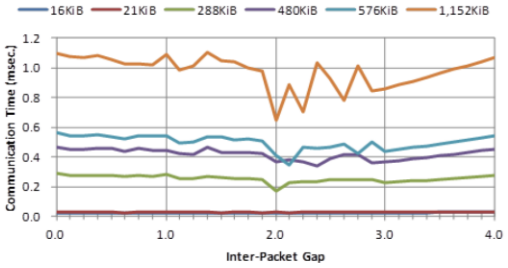


(a) インバランス 0nsec（無し）

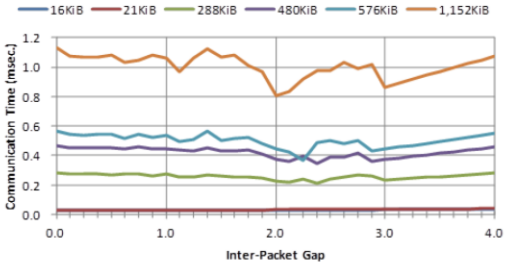
(b) インバランス 100nsec



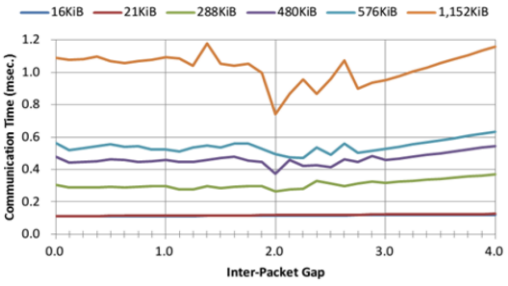
(c) インバランス 186.4nsec（SW バリア）



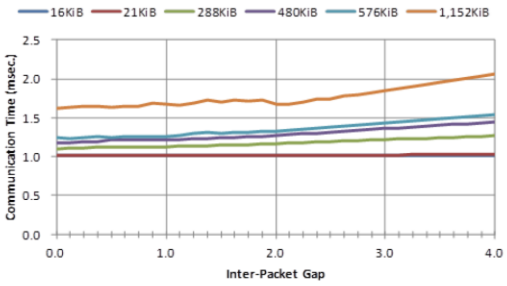
(d) インバランス 1usec



(e) インバランス 10usec



(f) インバランス 100usec

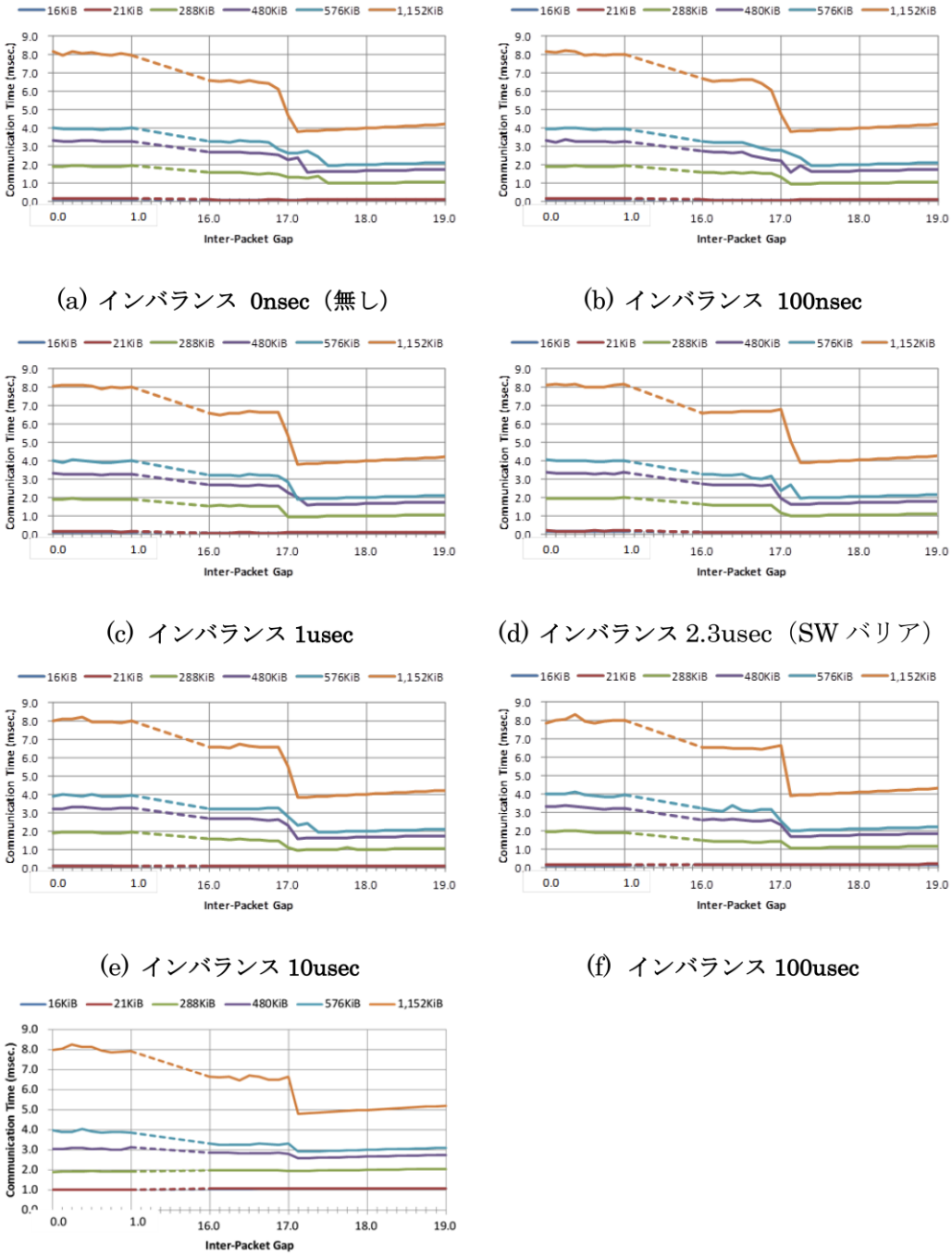


(g) インバランス 1msec

図 9　各インバランス係数における通信時間の推移（大規模実行モデル）

机上計算値を基点とし，NSIM によって通信衝突のオーバヘッドを考慮した通信時間と比較すると，281.7 ミリ秒（机上計算）から，638.1 ミリ秒（インバランス無し）もしくは 654.2 ミリ秒（インバランス有り）へと約 2.27 倍の通信時間の増加となった．これらに対し，パケットペーシングを適用することによって，それぞれ 385.9 ミリ秒，440.9 ミリ秒とそれぞれ 60%から 67%に通信時間を短縮できる可能性がある．

同様に，大規模実行モデルの推定結果を図 12 に示す．小規模実行モデルの結果と同様に，通信衝突を考慮した場合，理論値から約 2.2 倍の通信時間の増加となった．そして，パケットペーシングを適用することで約 49%と大幅な短縮が可能となり，理論値に近い通信性能を達成できる見込みがあることがわかった．



(a) インバランス 0nsec（無し）　(b) インバランス 100nsec

(c) インバランス 1usec　(d) インバランス 2.3usec（SW バリア）

(e) インバランス 10usec　(f) インバランス 100usec

(g) インバランス 1msec

図 10　各インバランス係数における通信時間の推移（小規模実行モデル）

6.　今年度の進捗状況と今後の展望

本課題では，当所の計画にもとづき実践的なアプリケーションの通信部分にパケットペーシングを適用し，通信時間全体を短縮できることをシミュレーションによって明らかにした．一方で，実アプリケーションから主要な通信パターンを抽出し，NSIM 向けの MGEN プログラムを生成する作業コストが大きく，種々のアプリケーションの通信パターンを取得するには至らなかった．

今後は，主要通信パターンの導出手法を見直すとともに，NSIM によるアプリケーションの具体的な実行時間を推定する手法を確立することが急務の課題である．

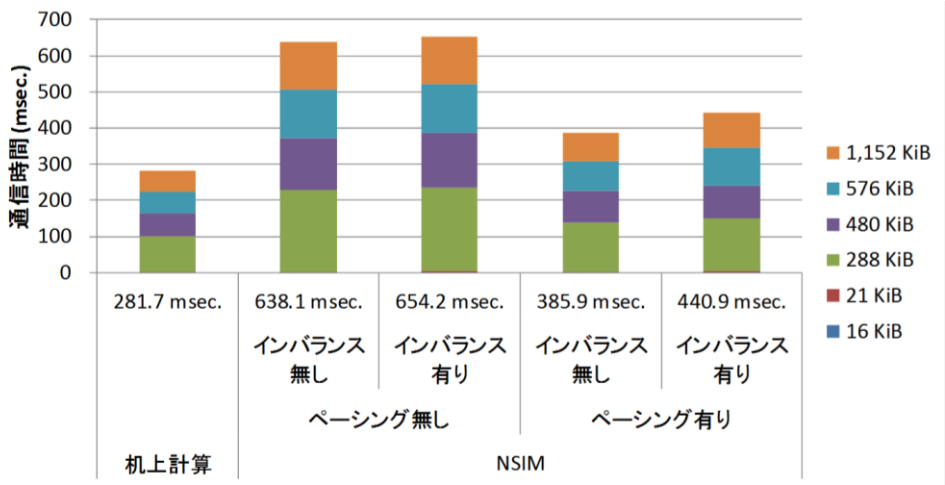


図 11 NICAM 通信における推定通信時間（小規模実行モデル）

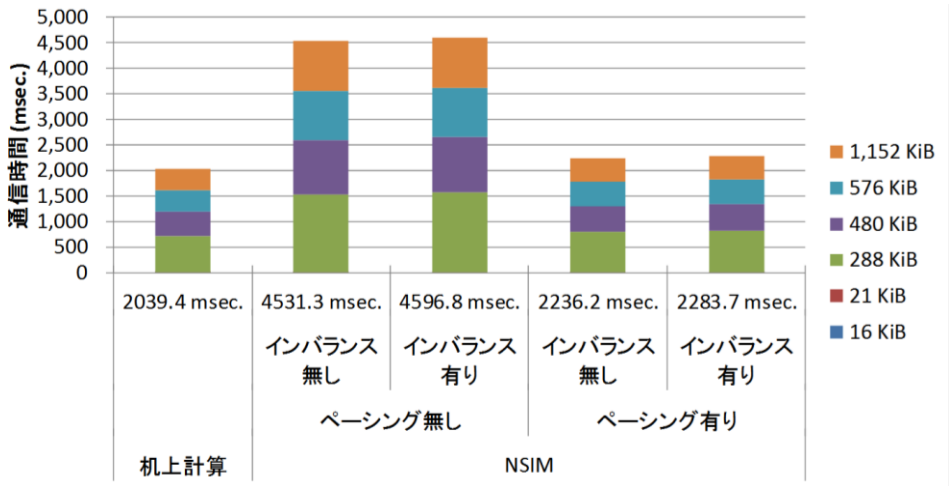


図 12 NICAM 通信における推定通信時間（大規模実行モデル）

7. 研究成果リスト

(1) 学術論文

該当なし

(2) 国際会議プロシーディングス

該当なし

(3) 国際会議発表

(3-1) Hidetomo Shibamura: Active Packet Pacing as a Congestion Avoidance Technique toward Extreme Scale Interconnect, International Supercomputing Conference 2014 (ISC'14), Poster at HPC in Asia Poster Session, 2014.

(3-2) Hidetomo Shibamura: Active Packet Pacing as a Congestion Avoidance Technique in Interconnection Network, International Conference on Parallel Computing 2015 (ParCo2015), Sep. 2015. (*to appear*)

(4) 国内会議発表

該当なし

(5) その他（特許，プレス発表，著書等）

該当なし