ハイパフォーマンス
コンピューティング 93−14
（２００３．３．１２）

WAN 上の複数クラスタによる単一 MPI アプリケーションの性能評価

松田元彦[†] 石川 裕[††] 工藤知宏[†††]

2 つのクラスタ計算機が WAN 等の高遅延・高バンド幅のネットワークで接続されている環境を想定して，評価環境を構築する．そのために，クラスタ内のノードの一部をブリッジとして使用しネットワークに遅延を加えるソフトウェアを実装する．ブリッジとして用いる遅延ノード上では，パケットを一旦メモリに書き出した後，一定時間後に送信するプログラムを実行させる．本環境では，遅延時間とともに遅延ノードの個数を変えることによりバンド幅も変更可能である．本評価環境を用いて，高遅延の下，遅延とバンド幅を変化させた場合の MPI アプリケーションの挙動を観測する．評価対象としては NAS Parallel Benchmarks を使用する．遅延時間は，メトロポリタン地域で想定される 0ms ～4ms の範囲を選んだ．実験の結果は，この程度の遅延であればクラスタ間を接続することによって多少なりとも性能向上が得られることを示す．また，ベンチマーク毎の遅延とバンド幅による影響の受け方の違いをグラフとして示す．

MPI Application Performance Estimation on Clusters Connected over WAN

MOTOHIKO MATSUDA ,[†] YUTAKA ISHIKAWA [††] and TOMOHIRO KUDOH[†††]

For evaluating the behavior of two clusters connected via a WAN-like network, an emulation system for a network with large latency is developed. The system uses compute nodes as bridges of the network, which add a latency to packets passing them. The bridge nodes run a forwarding program, which stores in-coming packets in the memory and then transmits them later. This software-based system enables to vary bandwidth as well as latency, by changing the number of the bridge nodes employed. This system is used to observe the behavior of MPI applications, specifically the NAS Parallel Benchmarks, under the large latency varying from 0 milliseconds to 4 milliseconds. The benchmarks show that connecting two clusters has some merit under this range of latency. The benchmarks also show that each program reveals different sensitivity to the latency and the bandwidth.

1. はじめに

グリッド上の通信を使って，地域あるいは広域ネットワークで MPI[1] のアプリケーションを動かすことのできる MPICH-G2[2] といったシステムが現れてきた．既にキャンパス・ネットワーク内などでは複数のクラスタ計算機を利用できる環境が存在する．また，広域ネットワーク・インフラの充実は顕著である．東京近辺のメトロポリタン地域を例にとると，つくば–東京間では遅延時間として片道 3ms～4ms 程度が観測される．この程度の遅延時間の範囲で，複数クラスタをつなぐシステムが実用的かどうかは興味あるところである．しかし，こういった環境下で MPI アプリケーションを評価した報告は少ない [3][4]．

そこで，論文 7) ではこのような環境を想定して高遅延なネットワーク環境におけるクラスタ計算機の性能評価を行なった．そこでは 1000Base-T イーサネット上で NasPar ベンチマーク [5][6] の性能測定を行ない，MPICH-SCore (PM/Ethernet)，MPICH-G2，MPICH-P4 という複数の MPI の実装を比較した．遅延は Linux 上のネットワーク・エミュレータを利用して人工的な遅延の挿入している．NIC を 2 枚挿した PC を遅延ノードとしてスイッチ間に配置し，そこでエミュレータを動作させた．

しかし，このような実験形態では動作挙動の明確さの点で問題が残る．通信に TCP/IP 等のプロトコルが介在するが，それ自体がチューニングの対象になっており挙動が明確さに欠ける．特に現在の Linux では，使用されるカーネルのバージョンによってイーサネットや TCP/IP の性能が大きく変動することが知られている．その他にも，簡単にクラスタ間のバンド幅を変化させることができない問題もある．近い将来 WAN では 10 Gbps 程度が利用可能になると予想されるが，その実験にはクラスタ間に複数リンクを使用するネットワークを構築してやる必要がある．

そこでもっと構成要素が明確で構成変更が容易な環

† キヤノンシステムソリューションズ
Canon System Solutions, Inc.
†† 東京大学 大学院情報理工学系研究科
University of Tokyo
††† 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology

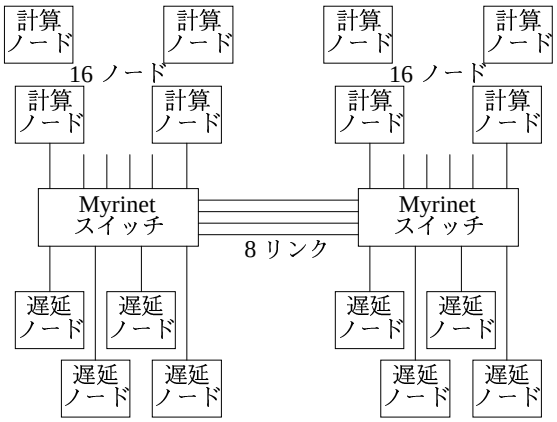

図 1 システム構成

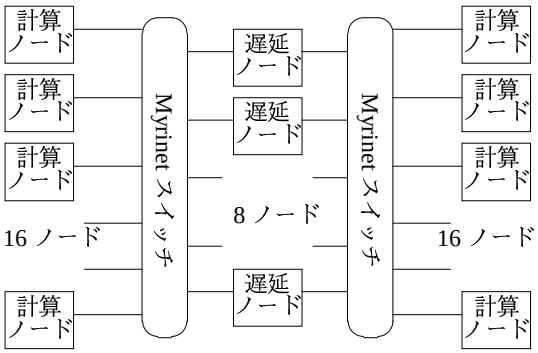

図 2 論理的なノード間の接続

境として，SCore クラスタ・システム [8] の計算ノードを遅延ノードとして使用する環境を構築することにした．ネットワークの遅延機構をクラスタ・ノード上で実現することは自然な発想である．クラスタ間のバンド幅の変更は，遅延ノードの個数を変更することで可能である．この場合，変更はノード上のソフトウェアだけであり，複数の遅延ノードを設定するのは容易である．また，SCore で使用される通信レイヤである PM[9] および Myrinet は，イーサネット，TCP/IP 等と比較すると構成要素が簡潔であり性能に与える影響も予測し易いと考えられる．

ただし，本来ならば遅延とバンド幅を独立に変化させるのが理想的な環境であるが，残念ながらそうはならなかった．まず，遅延ノードのハードウェア上の制約によって，遅延ノードを通過するバンド幅がリンク・ハードウェア性能の半分になっている．また，システム的な制約によりバンド幅が遅延の影響を大きく受けることが分かった．SCore クラスタ上の PM/Myrinet は通信バッファサイズが小さく，大きな遅延下ではバンド幅が制限される．さらに PM/Myrinet の特徴であるゼロコピー通信では通信にハンドシェークが必要であり，それに伴う遅延によりバンド幅が制限される．

以下では，まず，クラスタシステムに遅延を挿入する実験システムについて説明する．その実験システムの基本性能として，遅延を変化させた場合の通信スループット・バンド幅の変化を示す．次に，この実験システムを使って NarPar ベンチマークを計測する．各ベンチマークにおいて，遅延とバンド幅の両方を変化させた場合の性能の変化をスケーラビリティとして示す．続いて，実験システムの問題点に関する考察とまとめを述べる．

2. 遅延を挿入する実験環境

2.1 システム構成概要

まず，32 台構成の SCore クラスタを元に，8 台のノードを追加して 40 台構成のクラスタを構築した．追加したノードは元の 32 台と全く同一構成である．その後 8 台をクラスタ構成情報から削除してソフトウェア的に切り離し，遅延ノードとした．**図 1** にシステム構成を図示する．

このクラスタを 16+16 の 2 つのクラスタと考え，クラスタ間をまたぐ通信の全てが遅延ノードを経由するようにルーティング情報を設定する．つまり，論理的なノード間の接続は**図 2** のようになる．遅延ノードでは，受信したパケットを一定時間をおいて再送信を行なう．経由する遅延ノードは，行き先ノードに従ってラウンド・ロビンで静的に割り当てる．

2.2 通信ライブラリの変更

PM/Myrinet は，SCore クラスタ・システムの Myrinet を使う高速な通信レイヤである．これに人工的な遅延を挿入する．通信プロトコルは一切変更しない．

実験を簡単にするため，通信ライブラリの変更は最小限にとどめる．今回，変更するのは，通信ライブラリ中のルーティング情報を構成ファイルから読み出す部分だけである．さらに SCore 環境では，ルーティング情報の読み出しは `scored` と呼ばれるデーモン・プログラムが行ない，デーモンからアプリケーションに通知されるようになっている．このため，ルーティング情報の置き換えはデーモンを変更するだけで良く，アプリケーションは全く変更する必要がない．

Myrinet のハードウェアは，ソース・ルーティングを用いている．各パケットのヘッダとしてルート情報が付加されている．ルート情報はバイト列で，各バイトはスイッチでのフォワーディングする方向を指示する．スイッチではフォワーディング毎にルート情報を 1 バイト削除する．PM/Myrinet ではスイッチの接続状態を構成情報ファイルに保持しており，それに従って静的にルートを決定している．

今回は，このルート情報を変更し，遅延ノードを経由するように設定する．パケットのヘッダとして，遅延ノードへのルート情報と遅延ノードから目的ノード

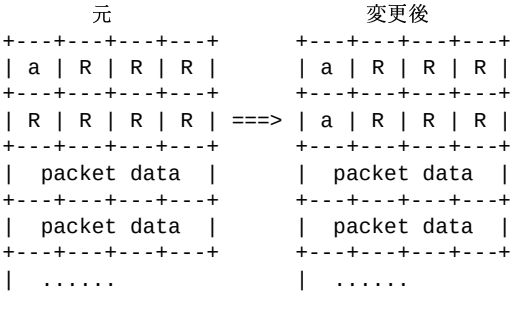

図 3 パケット・ヘッダの変更

へのルート情報の二つを挿入する．

PM/Myrinet のパケット・ヘッダには，目的ノード番号が入っていない．そのため，なんらかの方法で目的ノードを知らせる必要がある．このように二つのルート情報を持たせることで，PM/Myrinet 自体のパケット・ヘッダ等の形式を全く変更する必要がなくなった．また，遅延ノードでの処理が非常に簡単になる．パケットが遅延ノードに達した時点で，ヘッダには目的ノードへのルートが先頭にあるので，そのまま再送を行なうだけで良い．これはまた，通信プロトコルに関する一切の情報を必要としないので PM/Myrinet に依存する部分はなく，汎用のツールとして用いることが容易である．

図 3 にパケットのヘッダ情報を図示する．デフォールトの PM/Myrinet の設定では，ルーティングは 7 ホップまで可能になっている．これを変更して，3 ホップまでのルートを 2 個配置する．実験に用いるクラスタ構成では，各クラスタは 32 ポートのスイッチ (M2M-OCT-SW8) で接続される．この構成では，3 ホップ以下ですべての遅延ノードへのルートがカバーできるので，PM/Myrinet のデフォールトの設定のままで使用できる．ルートのホップ数の上限はコンパイル時に指定されるので，これを越える構成では PM/Myrinet ライブラリの再コンパイルが必要である．

2.3 遅延ノードプログラム

遅延ノードでは，受信したパケットを一旦ノードのメインメモリに保持し，指定した時間をタイマで待ってから再送信する．受信したパケットは全て一旦メモリに書き込み，リンク・リストとして保持する．受信したデータはそのまま何の変更もせず，一定時間後に送信する．

この処理は，全て Myrinet の NIC 上のプロセッサで実行される．ノード上のホスト・プロセッサでは，ページングを禁止するメモリのロック (ピンダウン) 処理のみを行なう．ホスト・プロセッサはノードの初期化とピンダウン処理を行なった後はアイドル状態の無限ループに入る．

プログラム自体は，PM/Myrinet のリモートメモリ参照機能を変更して作成した．主な変更としては，リモートメモリ参照にリンク・リストとして保持するコードを追加するだけである．Myrinet の NIC 上のプロセッサは 0.5μsec 毎にカウントされるタイマを持っており，遅延時間はこのタイマで計測している．

メモリは 10MB 程度をピンダウンして使用している．これは今回の実験の遅延量には十分な量である．

2.4 システム仕様

計算ノードおよび遅延ノードのプロセッサは Pentium III 933 MHz，メモリは 1 GB，OS は Linux 2.4.18 ベースである．プロセッサボードは 2 CPU 構成であるが，実験では 1 CPU しか使用していない．

実験環境で使用する Myrinet は数世代前の製品にあたる，Myricom 社製 Myrinet を使用した NIC カードは，33 MHz 32-bit PCI のカードである．NIC 上のプロセッサは LANai4.2 (クロック 33 MHz) である．リンク速度は 160 MB/s の SAN ケーブル仕様である．スイッチは 32 ポートの M2M-OCT-SW8 を 2 つ使用している．通信リンクの速度は 160 MB/s であるが，通信速度は PCI バスの上限 133 MB/s で押えられており，アプリケーションからの実測値はほぼ 120 MB/s である．

SCore クラスタ･システムは SCore 5.0，MPI は SCore 附属の MPICH-1.2.0 を使用した．Fortran コンパイラは Intel IFC 6.0，コンパイル・オプションは `-O3` を使用した．

3. バンド幅性能評価

3.1 遅延プログラムの性能

アプリケーションの性能評価を行なう前に，予備評価として遅延ノードを経由する場合のバンド幅を計測した．比較として，遅延のない PM/Myrinet 直接の場合の値も示す．MPI における通信遅延時間は，0ms 遅延の場合は 100.0 μsec である．一方，遅延なしの場合 23.4 μsec である．これはデータサイズ 8 byte 時のラウンド・トリップ値である．MPI におけるピークバンド幅は，0ms 遅延の場合は 53.5 MB/s である．一方，遅延なし場合 120 MB/s である．

遅延ノードのバンド幅性能は 50 MB/s を少し上回る程度であり，遅延を入れない場合の半分程度の性能しかない．原因は PCI のバンド幅による制限である．今回，33 MHz 32-bit PCI の Myrinet を使用しているが，PCI バスのバンド幅は最大 133 MB/s である．パケット毎にメモリへの書き込みと読み出しを行なうため PCI バスを 2 回使用する．よってパケットをフォワードする場合のバンド幅はこの半分の値である 66 MB/s で押えられることになる．

3.2 通信オプション

MPI の実装では，一定のデータサイズを超える場合，

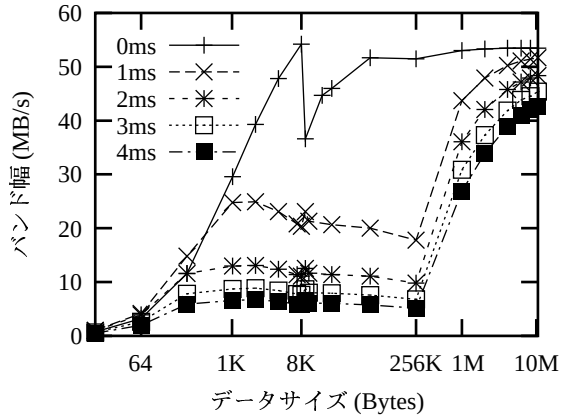

図 4 遅延に対するバンド幅の変化

送信側と受信側で `MPI_Send` と `MPI_Recv` の呼び出しが揃ってから通信するランデブー処理を行なう．これはデータのバッファへのコピーを避けるためであるが，ランデブー処理には通信によるハンドシェイクが必要である．加えて，SCore 上の MPI の実装ではゼロコピー通信としてリモートメモリ参照を使ってさらにコピーを減らす通信が可能である．この場合，ランデブー処理の後，リモートメモリ参照が行なわれる．

以下の実験では，遅延のない場合にはゼロコピー通信オプションを指定した．具体的には，オプション `mpi_zerocopy=on` を指定した．この場合，16 KB を超える通信でリモートメモリ参照が使用される．遅延のある場合は (遅延 0ms の場合も含め)，ゼロコピー通信とランデブーを行なうメッセージサイズを指定した．具体的には，オプション `mpi_eager=262144,mpi_zerocopy=on` を指定した．これにより 256 KB を超える通信でリモートメモリ参照が使用される．この 256 KB という値は，バンド幅の実測によりメッセージ通信とゼロコピー通信がこのあたりのサイズでクロスオーバーすることを観測して決定した．

3.3 遅延によるバンド幅の変化

図 4 に遅延時間を変えた場合のバンド幅の変化を示す．これは送信側は `MPI_Send` を，受信側は `MPI_Recv` をバースト的に繰り返して測定したものである．

データサイズ 8 KB のところでのバンド幅の低下は PM/Myrinet のパケットサイズ (MTU) によるものと考えられる．ここでの低下は遅延のない場合 (PM/Myrinet 直接) でも観測される．データサイズ 256 KB での屈折は，上記に説明したメッセージ通信からリモートメモリ参照へのプロトコルの切替えに伴うものである．

PM/Myrinet では，メッセージ送受信のバッファとして合計 128 KB 程度が割り当てられる．このサイズでは 1ms においても遅延をカバーするには不足すると考えられるが，実際大きくバンド幅が低下している．

また，MPI における PM/Myrinet のゼロコピー通信は立上りが悪い．ゼロコピー通信を行う場合，データ転送を含めて 3 往復の通信が行なわれる．これらは `MPI_Send` の繰返しを行なう場合オーバーラップされない．まず，MPI の実装ベースである MPICH では，デバイスレイヤである ch レイヤにおいてランデブー処理として 1 往復のハンドシェイクが行なわれる．続いて SCore 上の MPI 実装として，ページングを禁止するピンダウン処理のためのハンドシェイクが 1 往復，ゼロコピー通信をリモートメモリ読み出しによって行なうので 1 往復が必要になっている．

4. NasPar 性能評価

4.1 評価方法

MPI アプリケーションの遅延とバンド幅に対する性能変化を見るために，NAS Parallel Benchmarks (NPB2.3) を使用してそれぞれの性能評価を行なった．データサイズはすべてクラス B を使用した．データはベンチマークの表示のうち「Mop/s total」値を比較している．

遅延時間は 0ms から 4ms と変化させた．また参考として，さらに大きい遅延時間として 10ms，20ms の結果も併せて示した．バンド幅は遅延ノード数を変化させることによって変更している．遅延ノードとして 8 ノード用意しており 1，2，4，8 とノード数を変化させて測定した．グラフではそれぞれバンド幅として，1×50 MB/s，2×50 MB/s，4×50 MB/s，8×50 MB/s と表示している．

グラフは，16 ノードの時の性能に対する比 (スケーラビリティ) を表示している．つまり，値 1 がクラスタ間を接続するメリットがない状態，値 2 が理想的なスケーラビリティが得られた状態にあたる．

グラフ上ではスケーラビリティの値 1 のところを水平な線で示している．また，遅延を加えない場合，つまり PM/Myrinet で 32 ノードの場合の値も水平な線で示した．この値としては，クラスタ間に 4 リンクを使った場合を採った．先に示したように，遅延ノードは約 50 MB/s 強のバンド幅を持っている．8 ノードでは 400 MB/s 程度であるので，これに相当する 4 リンクの値を遅延なしの場合とみなした．

4.2 EP

EP の結果を示すグラフは省略した．EP は Embarrassingly Parallel なベンチマークであり，遅延にもバンド幅にも影響されない．すべてのケースについてスケーラビリティは 2 である．

4.3 CG と MG

図 5，**図 6** に CG と MG の結果を示す．これらは遅延のない場合は割りに良いスケーラビリティを示すが，遅延が大きくなるにともない性能が低下する．遅延時間 4ms 程度が限界となっている．同時にバンド幅の影

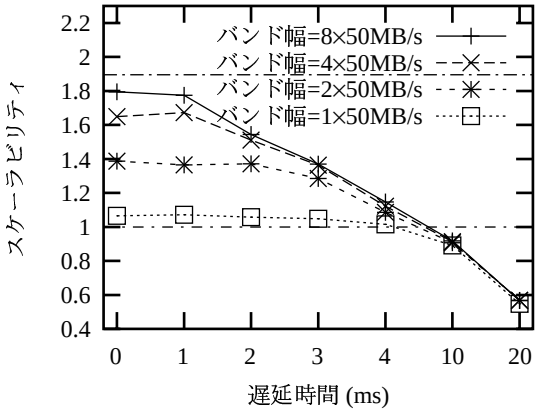

図 5 CG における遅延に対するスケーラビリティの変化

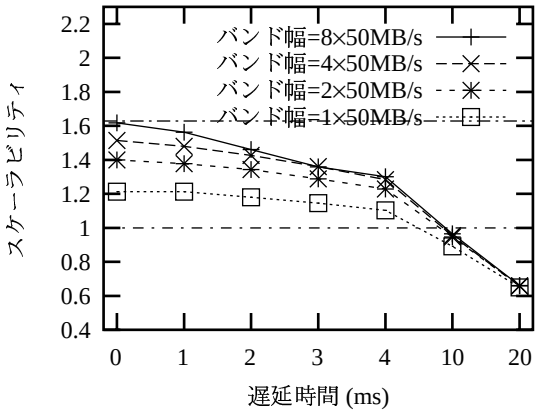

図 6 MG における遅延に対するスケーラビリティの変化

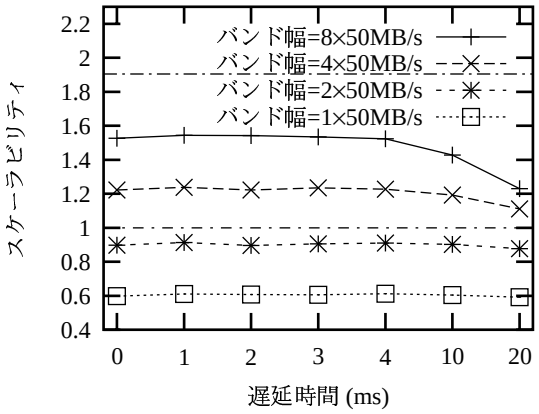

図 7 FT における遅延に対するスケーラビリティの変化

響も受けていることが分かる．今回の実験では，CG や MG のように遅延とバンド幅の双方に影響を受ける場合，支配的なのが遅延なのかバンド幅なのかあるいは双方なのか分離できない．

4.4 FT

図 7 に FT の結果を示す．FT ではバンド幅による

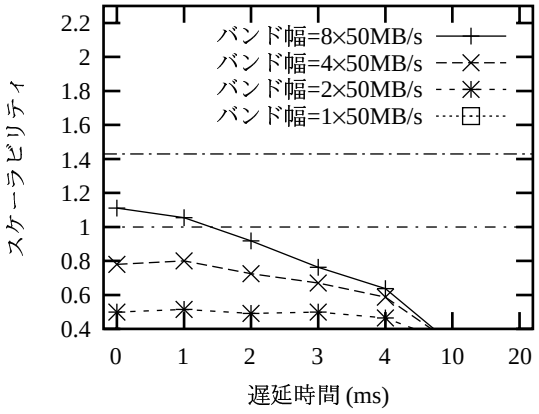

図 8 IS における遅延に対するスケーラビリティの変化

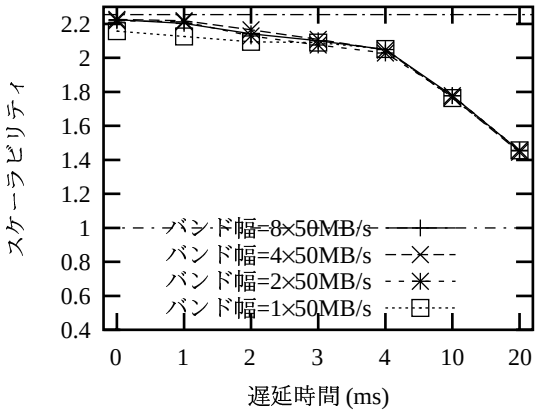

図 9 LU における遅延に対するスケーラビリティの変化

影響が大きい．ここでは示していないが，遅延を入れない場合の結果も同じような傾向を示す．一方，遅延にはそれほど影響されていないことが分かる．

4.5 IS

図 8 に IS の結果を示す．バンド幅 1×50 MB/s の場合の結果はグラフから外れており表示されていない．IS では遅延がない場合もスケーラビリティが良くない．

IS は全対全通信を使っており，バンド幅の影響を受けることが良く知られている．一方，論文 7) の結果では，性能は良くないが (スケーラビリティの値で 0.5) 遅延の影響はあまり受けていないことが観測されている．今回の実験環境では，遅延に従ってバンド幅が低下するので，遅延にも影響を受ける結果になっていることも考えられる．

4.6 LU

図 9 に LU の結果を示す．LU はとても性能が良い．遅延を隠蔽するコーディングがなされており，それが効果的であると考えられる [10]．また，バンド幅の影響も小さい．さらに遅延が小さい場合にスーパーリニアな結果になっている．

5. 実験システムの問題点

高遅延・高バンド幅な環境でのベンチマークが目的であったが，実験システムはハードウェア上の制約によりバンド幅が半分であった．しかし，高遅延下でベンチマークの評価を行なうには，遅延システムを改善するだけでは問題は解決しない．アプリケーションの動作挙動について議論するためには，通信レイヤ全体が高遅延に対応している必要がある．そのためには，MPI の実装を含めて作り直す必要がある．

例えば，クラスタの通信レイヤは低遅延を想定しており，通信バッファが小さいことが問題である．MPI の実装については，ハンドシェイク，送受信の逐次化，低レベルでの逐次化などといった点が問題になると考えられる．ハンドシェイクの問題ついては，バンド幅の性能評価のセクションで通信が 3 往復行なわれている点を説明した．特にランデブー処理は，デバイスレイヤの抽象化によるコストである．送受信の逐次化の問題は，ランデブー処理にも関係するが複数の通信処理がオーバーラップされないことである．低レベルでの逐次化の問題は，送信がパケット単位でなくデータサイズ分を単位に行なわれることである．この場合，通信は一定の相手に対して連続的に行なわれる．

6. おわりに

遅延とともにバンド幅を変化させた場合の NasPar ベンチマークの挙動を見ることができた．ただし，遅延とバンド幅が独立ではないので結果を見る場合に注意が必要である．

この実験のために，クラスタ内のノード自体を利用して遅延を挿入するシステムを開発した．遅延時間を自由に設定できること，バンド幅を十分大きくとること，かつ通信リンク数を変えることによりバンド幅も変更できることをシステムの目的にした．ただし，クラスタ用の通信レイヤは遅延が小さいことを仮定しており，遅延が大きいところではバンド幅が制約されてしまい遅延とバンド幅を独立に変更できる環境にはならなかった．

今回はクラスタの通信プロトコルは一切変更せず，簡単なパラメータについて調整しただけである．プロトコル上の問題など理想的な状態とは言えないが，遅延の大きいネットワークでの MPI アプリケーションの性能をある程度確かめることができた．

遅延時間が 0ms から 4ms の間では，NasPar ベンチマークはそれほど悲観的でない結果を示した．基準が同一でないので比較は不可能だが，遅延時間よりもバンド幅の影響を大きく受ける傾向も分かった．この結果から，グリッドのような高遅延環境でも EP 以外の MPI アプリケーションが実行される可能性が十分ある．この実験結果を，高遅延環境でも性能低下の小さい MPI 通信ライブラリを開発するのに役立てて行きたい．

謝　辞

本研究にあたり議論に参加いただいた株式会社 SRA 平野基孝氏に感謝します．

本研究の一部は，経済産業省の委託事業「ネットワークコンピューティング技術の開発」による．

参考文献

1) Message Passing Interface Forum. *MPI: A Message-Passing Interface Standard, May 5, 1994.* University of Tennessee, Knoxville, Report CS-94-230, 1994.
2) MPICH-G2. http://www3.niu.edu/mpi/
3) 辻田祐一, 山岸信寛, 木村和幸, 大谷孝之, 鶴岡信彦，藤田直行．機種計算機間 MPI 通信ライブラリ Stampi による VPN を介した SCore クラスタシステム間通信性能評価．情報処理学会研究会 2002-HPC-92-3, October 2002.
4) Atsuko Takefusa, Satoshi Matsuoka, Hirotaka Ogawa, Hidemoto Nakada, Hiromitsu Takagi, Mitsuhisa Sato, Satoshi Sekiguchi, and Umpei Nagashima. Multi-client LAN/WAN Performance Analysis of Ninf: a High-Performance Global Computing. SC'97, IEEE, 1997.
5) William Saphir, Rob Van der Wijngaart, Alex Woo, and Maurice Yarrow. New Implementations and Results for the NAS Parallel Benchmarks 2. 8th SIAM Conference on Parallel Processing for Scientific Computing, March 1997.
6) NAS Parallel Benchmarks. http://science.nas.nasa.gov/Software/NPB
7) Motohiko Matsuda, Yutaka Ishikawa, and Tomohiro Kudoh. Evaluation of MPI Implementations on Grid-connected Clusters using an Emulated WAN Environment. The 3rd IEEE/ACM Intl. Symp. on Cluster Computing and the Grid (CCGrid 2003), May 2003 (to appear).
8) SCore and PC Cluster Consortium. http://www.pccluster.org
9) Toshiyuki Takahashi, Shinji Sumimoto, Atsushi Hori, Hiroshi Harada, and Yutaka Ishikawa. PM2: High Performance Communication Middleware for Heterogeneous Network Environments. SC'00, IEEE, 2000.
10) Maurice Yarrow and Rob Van der Wijngaart. Communication Improvement for the LU NAS Parallel Benchmark: A Model for Efficient Parallel Relaxation Schemes. Tech. Rep. NAS-97-032, NASA Ames Research Center, November 1997.