100BaseTX によるメモリベース通信の性能評価

松本 尚　　平木 敬

東京大学 大学院理学系研究科 情報科学専攻
〒 113 東京都文京区本郷 7–3–1
Email: tm@is.s.u-tokyo.ac.jp

あ ら ま し　汎用並列分散システムでは、効率の良い実行環境を実現するためにノード間の高速かつ保護され仮想化されたユーザレベル通信および同期のサポートが不可欠である。我々はこの目的に叶う高速なユーザレベル通信同期として、他ノードのメモリ空間内のデータを直接読み書きするソフトウェアメモリベース通信を考案し開発している。本稿では、 100baseTX を用いたメモリベース通信のパケットフォーマットを解説し、その基本性能を性能テストプログラムとロジックアナライザによる観測で明らかにする。そして、並列アプリケーションにメモリベース通信を利用した場合の性能を、並列レイトレーシングを題材にプロセッサ台数や通信粒度をパラメータとして変化させることで明らかにする。最後に、比較のために既存オペレーティングシステムの UDP/IP 通信上にメモリベース通信方式で同様の性能測定を行い、ハードウェアドライバから実現した本来の方式との性能差を調べる。

キーワード　メモリベース通信, ユーザレベル高速通信, ワークステーションクラスタ, 汎用並列分散処理環境

Performance of Memory-Based Communication Facilities Using Fast Ethernet (100BaseTX)

Takashi MATSUMOTO　　Kei HIRAKI

Department of Information Science, Faculty of Science, University of Tokyo
7-3-1 Hongo Bunkyo-ku Tokyo 113, Japan
Email: tm@is.s.u-tokyo.ac.jp

Abstract　In general-purpose parallel and distributed systems, performance of the protected and virtualized user-level communications/synchronizations is the most crucial issue to realize efficient execution environments. We proposed a novel high-speed user-level communications/synchronizations scheme "Memory-Based Communication Facilities (MBCF)" suitable for the general-purpose system with off-the-shelf communication-hardware. This paper describes packet formats of the MBCF with 100baseTX communication interfaces. Next, the paper shows basic performance of the MBCF/100baseTX using test programs and a logic analyzer which measures wave forms of 100baseTX interface. Finally, we develop another MBCF interface on UDP/IP in conventional operating systems and compare the performance of our original MBCF with that of the MBCF/UDP.

key words　Memory-Based Communication facilities, User-Level Communication, Network of Workstations

1 はじめに

マイクロエレクトロニクス技術の急速な進展に伴い、ワークステーションを高速ネットワークで結合したワークステーションクラスタ（Network of Workstations: NOW）や多数の CPU を搭載した分散メモリ型並列計算機が容易に実現可能になり、従来メインフレーム、すなわち大型 / 超大型計算機システムだけが実現可能であった応用分野に対しても適用の可能性が高まってきた。これらの分散メモリ型の並列分散計算環境を大型 / 超大型計算機システムとして利用するためには汎用性（マルチユーザ・マルチジョブ）の導入が不可欠である。しかしながら、汎用性導入による保護機能の実現や実資源の仮想化は、並列分散処理の高効率化と相容れない要素があり、並列分散処理の大幅な性能低下の原因となる。

並列処理および分散協調処理の最大の特徴はプロセッサおよびノード間の通信と同期であり、高速なユーザレベル通信同期なしには効率の良い並列実行環境はあり得ない。そして、このユーザレベル通信同期は保護と仮想化の要件十分に満足しつつ高速に実装される必要がある。我々はユーザレベル通信専用ハードウェアを用いることなしに、これらの条件を満たすソフトウェアメモリベース通信 (**MBCF**: Memory-Based Communication Facilities) [1, 2, 3] を考案した。

本稿では、 MBCF の特徴を簡単に説明し、過去の公表内容から若干変更があったパケットフォーマットを述べ、汎用の通信機構である 100baseTX (Fast Ethernet) および 10baseT (Ethernet) を用いて実装された MBCF の基本通信性能とアプリケーション（並列レイトレーシング）における性能について報告する。

2 MBCF の特徴

集中共有メモリを持つ並列計算機では、プロセッサは共有メモリ領域への通常のメモリアクセスで通信同期を行う。ユーザプログラムはマッピングされたページにしかアクセスできないため、ページ管理機構によってジョブ間の不当干渉を排除することが可能である。つまり、ユーザレベルの通信（同期）を通常のメモリの load/store で実現しており、保護に関してはプロセッサのメモリ保護機構の方式がそのまま流用可能である。

しかし、集中共有メモリ型並列計算機は集中共有メモリへのアクセスがボトルネックとなり、プロセッサ台数の大規模なものを製造することが困難である。そこで、松本は従来のページ管理機構を遠隔メモリアクセスに拡張した Memory-Based Processor (**MBP**)[4] を考案した。 MBP を持つ分散メモリ実装の並列計算機やワークステーションクラスタ（**NOW**: Network of Workstations）では、集中共有メモリ型計算機と同様に通常のメモリアクセスとして高速かつ保護され仮想化されたユーザレベル通信同期が実現できる。

しかるに、 MBP タイプのハードウェア付加機構は現時点において一般的ではなく、ソフトウェアの助力なしに主記憶を大容量キャッシュとして流用するためには主記憶に付加的なタグ情報を持たせる必要がある[1]。また、 MBP は主要素プロセッサのメモリアクセス動作と密に協調して働くため、 MBP の実装はプロセッサのメモリ周りの実装に依存してしまう可能性がある。

これらの理由から、汎用並列分散オペレーティングシステムの開発に当たって、我々は集中共有メモリや MBP と同様な通信同期ハードウェアを仮定しない分散メモリ実装の並列計算機環境（NOW を含む）において実現可能な、高速かつ保護され仮想化されたユーザ通信 / ユーザ同期を考案する必要にせまられた。これに対して我々が出した回答が MBCF である。

MBCF は以下のような最新技術やソフトウェア技巧を用いて、基本的に MBP の動作と機能を高速ソフトウェアエミュレーションで実現している。

- 高性能プロセッサのローカル（キャッシュヒット）処理の高速性
- 余分な処理をしない軽い送信専用システムコール
- 余分な処理をしない受信割り込みルーチン
- コンテクスト識別子を含む複数コンテクストの混在できる TLB
- 軽いアドレス空間切替えハードウェア
- ページエイリアス機能
- 物理アドレスタグを持つプロセッサキャッシュ
- 論理アドレスによる通信相手空間の直接操作

MBCF の定性的な議論は文献 [2] を、 100baseTX 版および 10baseT 版の MBCF の実装技術に関する詳細は文献 [3] を参照されたい。なお、専用ハードウェアの不要な高性能分散共有メモリシステム「非対称分散共有メモリ（**ADSM**）[2]」にも MBCF が使用され、 MBCF と同じ方針が採用されている。

3 MBCF のパケットフォーマット

文献 [3] で公開したパケットフォーマットに関して、ユーザレベルのパケットコンバイニング（パケットマージ）の性能を改善するために、若干の変更が施されフォーマットは現在以下のようになっている。

図 1に MBCF_WRITE を行うイーサパケットを示す。便宜上、パケットの先頭を 0 番地として byte 単位のアドレスによってパケット内の位置を示すことにする。 0 番地から 19 番地までは Ether 上の IP パケットと同形式をしている。ただし、 Packet Type として IP 等の他のプロトコルと衝突しないものを使用し、 Source Physical Node ID は IP アドレスではなく MBCF 用の独自のノード ID を採用している。 20 番地は空いている。 21 番地から 23 番地ま

[1] false sharing の問題を緩和するために必要である。

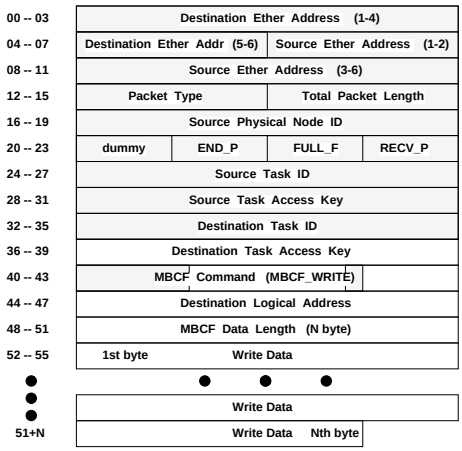

図 1: 単一 MBCF_WRITE のパケット

での 3 個の 1byte 長の制御データは通信到着保証と FIFO 性保証と Acknowledge 制御のために使用されている。24 番地、 28 番地にはそれぞれパケット送信元のタスク ID とアクセスキーが格納され、 32 番地にはパケットの宛先のタスク ID が格納される。これら、 0 番地から 35 番地までのパケットの内容は必ずカーネルレベルのルーチンによってセットアップされる。宛先のタスク ID と宛先ノード ID およびその ether address はユーザが指定した宛先の論理タスク番号からオペレーティングシステム内において表引きで求められる。 36 番地から 51 番地までのデータはユーザによる直接パケット生成時にはユーザレベルでセットアップされる。ただし、 40 番地の MBCF Command の上位ビットはパケット再送制御等に使用されるため、カーネルルーチンで正しく再設定される。図中のハッチ部分はカーネルが値を設定する領域であり、ハッチが施されていない部分はユーザが値を自由に設定できる。 36 番地には宛先タスクのアドレス空間へのアクセスキーが格納され、 40 番地には MBCF の機能を示すコマンドが書かれる[2]。 44 番地に宛先タスク内の書き込みを開始する論理アドレスが格納され、 48 番地に書き込みデータ長が格納される。 52 番地からは実際の書き込みデータ続く。保護のため、アクセスキーが間違っている場合や宛先論理アドレスが宛先タスクにおいてアクセス可能になっていない場合は単に通信 (この場合は書き込み) が失敗するだけではなく、セキュリティ保持のためのペナルティが発信元等に課される場合もある。

図 2に MBCF パケットの他の実例を示す。 0 番地から 35 番地までは図 1の MBCF_WRITE パケットと同一であるため省略してある。図 2(a) には遠隔ノード上の複数データを不可分に SWAP するコマンド「MBCF_SWAP」のバリエーションの一つである「MBCF_SWAP_STAT」を示す。 MBCF_SWAP_STAT はこの処理が終了したことを発信元タスクのフラグ変数 (56 番地にポインタ) に 1 を書き込むことで通知する。また、 SWAP で遠隔操作対象のタスク内のデータを返送するためのポインタ (送信元タスク内の論理アドレス) が 52 番地に含まれている。コマンドバリエーションを示すことは本稿の主旨ではないので、全コマンドの詳細な説明は他稿に譲る。図 2(b) には同一タスク宛の複数の MBCF コマンドが 1 パケットのまとめられた形式のパケットを示す。図では同一タスクへの 1byte と 4byte の二つの MBCF_WRITE を 1 パケットにコンバイニングして (まとめて) いる。複数の MBCF Command が含まれていても再送制御やフロー制御等は 40 番地の最初の MBCF Command の上位ビットのみで行うため、任意の MBCF コマンドが一つのパケットにまとめられるわけではない。複数の MBCF コマンドがまとめられる (コンバイニング可能) かどうかは、返送パケットの有無や返送パケットのサイズおよびマルチキャスト動作の経路等で決まる。

[2]MBCF コマンドと宛先タスク ID および宛先アクセスキーのパケット内位置が以前の発表から変更されている。

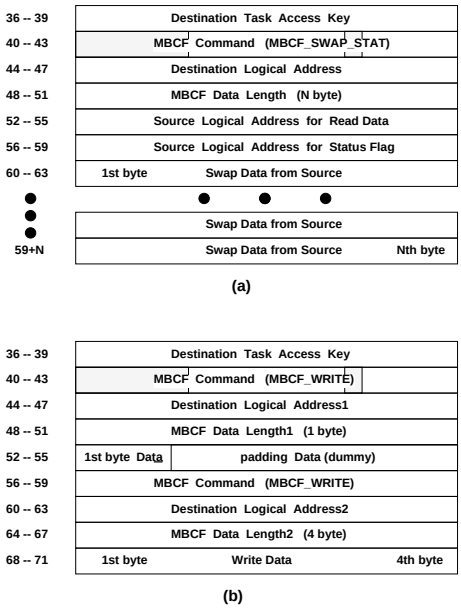

図 2: MBCF パケットのバリエーション

MBCF の発信用の軽い専用システムコールには二種類ある。両者とも、送信先は論理タスク番号で渡され、 35 番地以前のパケット領域はカーネルによって設定される。一つのシステムコールは宛先タスクの操作対象論理アドレス等のパラメータと パケット内データ (MBCF_WRITE の書き込みデータ等) へのポインタが構造体で渡されるものである。そして、もう一つはユーザがパケットのユーザ設定領域 (図 2に相当する 36 番地以降) を完全に組み立ててから組み立てたパケットへのポインタをシステムコールに渡す。前者はユーザがすでに使っているメモリ領域をデータとして転送する場合に、ポインタ渡しできるのでユーザモード内におけるコピーが省略できる。ただし、複数の MBCF コマンドをコンバイニングする場合はシステムコール内のパケット組み立てのための処理量が増える。後者は、パケット組み立てが完全にユーザ (コンパイラ、ユーザモードのランタイムを含む) に任されているので、

MBCF コマンドのコンバイニング等が自由に行える。

現在の 10baseT または 100baseTX による MBCF の実装では再送のためパケットのコピーをノード内に残す必要があるため、二種類のシステムコール共にデータのカーネル内におけるパケットデータのコピーが避けられない[3]（ユーザに再送バッファの管理まで任せればコピー不要となるが、これではユーザ側の処理が非常に繁雑）。ハードウェア的にデータ転送が保証されていれば、後者のユーザによるパケット組み立てのシステムコールでは、カーネル内におけるデータコピーが完全に省略できる。

4 性能評価

本稿で述べる MBCF の評価は以下の環境で行った。使用した NOW 環境は Axil 320 model8.1.1 (Sun SS20 互換機, 85MHz SuperSPARC CPU × 1) を 8 台、10baseT のハブで接続している。この 8 台のうちの 5 台は Sun Microsystems 社製の Fast Ethernet SBus Adapter 2.0 を追加して、100baseTX のハブで Fast Ethernet 接続されている。オペレーティングシステムは MBCF および ADSM のテストベッドとして開発された汎用超並列超分散オペレーティングシステム SSS–CORE[5, 1] Ver.1.0 を使用した。SSS–CORE Ver.1.0 にはこれまでに公表した MBCF の機能（保護やセキュリティ面を含む）がフルスペック（Memory-Based FIFO, Memory-Based Signal 等を含む）で実装されている。

4.1 100baseTX 版 MBCF の基本性能

MBCF の基本性能に関して 2 ノード間の通信性能測定プログラムを使用して評価を行った。

時間は 0.5μsec 単位の時計で性能測定プログラム内でソフトウェア的に計測した。ただし、この時計の読み出し 1 回に約 1.2μsec のオーバヘッドがハードウェア構成上かかるため、計測した値は開始と終了時の二回の時計読み出しにより、約 1.2μsec だけ大きな値になっている。なお、いくつかのケースについてはロジックアナライザを使用した厳密な波形測定に基づく時間測定を後に示す。

基本性能は Round-trip タイム、送信システムコールのオーバヘッド、Peak Bandwidth で示す。Round-trip タイムの表においては三種類の MBCF コマンドの種類別に性能を示す。これらの表における各通信コマンドの機能と測定条件は以下の通りである。

- **MBCF_WRITE**

  通信要求時に data を運び、対象メモリに書き込み後、書き込み完了を要求元のタスクに通知する。時間の測定は通信要求のシステムコールの直前から書き込み完了のフラグをスピンウェイトで検知するまでである。

- **MBCF_READ**

  通信要求時にアドレス・コマンド情報のみを運び、対象メモリを読み出し後、データをパケットで転送し指定されたバッファに格納の後に読み出し完了のステータスを指定アドレスに書き込む。時間の測定は通信要求のシステムコールの直前から読み出し完了のフラグをスピンウェイトで検知するまでである。

- **MBCF_SWAP**

  通信要求時に data を運び、対象メモリから古い data を読み出し後、運んで来た data を書き込み、読み出した data をパケットで転送し指定されたバッファに格納の後に SWAP 完了のステータスを指定アドレスに書き込む。時間の測定は通信要求のシステムコールの直前から SWAP 完了のフラグをスピンウェイトで検知するまでである。

表 1に 100baseTX による MBCF の Round-trip タイムを示す。

表 1: 100baseTX による MBCF の Round-trip タイム

data size (byte) コマンド種別	4	16	64	256	1024
MBCF_WRITE (μs)	51	54	60.5	88	200
MBCF_READ (μs)	51	54.5	61	88	200.5
MBCF_SWAP (μs)	51.5	58	71.5	125.5	351

参考までに表 2に 10baseT によるメモリベース通信の Round-trip タイムを示す。

表 2: 10baseT による MBCF の Round-trip タイム

data size (byte) コマンド種別	4	1024
MBCF_WRITE (μs)	213	1080
MBCF_READ (μs)	228	1090
MBCF_SWAP (μs)	229	1930

次に、ソフトウェアで測定した送信時オーバヘッドの表 3を示す。測定は遠隔書き込み時のシステムコール呼び出し直前から呼び出しから戻るまでを前出の 0.5μsec の時計で計測した。なお、10baseT の場合も、送信ルーチンが 100baseTX と通信ハードウェアレジスタの操作部分以外同じであるため、本オーバヘッドはほとんど同じである。

表 3: 100baseTX による MBCF の送信オーバヘッド

data size (byte)	4	16	64	256	1024
送信コスト (μs)	5	5.5	6	8.5	20

[3]MBCF 現実装ではこのコピーを最小限の 1 回に抑えている。

表 4: 100baseTX/10baseT のメモリベース通信の Peak bandwidth

data size (byte)	4	16	64	256	1024	1408
100baseTX (Mbyte/s)	0.29	1.06	4.03	8.28	10.86	11.24
10baseT (Mbyte/s)	0.04	0.17	0.48	0.89	1.13	1.17

表 4に 100baseTX と 10baseT のメモリベース通信のピーク転送性能を示す。測定は MBCF_WRITE をデータサイズを変えながら実測した。表中の値は Ethernet のヘッダーや MBCF のプロトコルデータを省いた遠隔書き込みの転送データの正味サイズだけから計算した値である。 Round-trip タイムの測定ではないので、通信ごとの操作完了フラグの返送は行っていない[4]。転送は同一アドレスに対してバースト状に行い、ネットワークが過負荷状態にならないように 16 転送毎に完了フラグを返送させてチェックするアクノリッジによって流量を調節した。ユーザレベルの保護され仮想化された通信方式としては良好な値であり、ほとんど生のハードウェア性能 (100baseTX: 12.5Mbyte/s, 10baseT: 1.25Mbyte/s) を使い切ることに成功している。

4.2 ロジアナによる基本性能の厳密測定

100baseTX による MBCF は非常に高速であるため、アクセスに 1.2μsec を要する 0.5μsec 刻みのタイマでは正確な測定は難しい。そこでワークステーションと Fast Ethernet board にロジックアナライザを接続して波形の計測を行い、正確な時間を求めた。キャッシュのヒット状況によってコストが変動する。以下の測定値はキャッシュエントリのスラッシングが発生しない状況での値である。システムコールならびに受信ルーチンのオーバヘッドの測定はプロセッサチップ (SuperSPARC-II) にある supervise アクセスピンを測定することで行った。

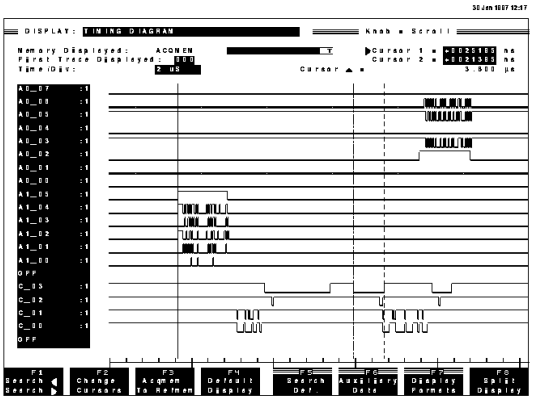

図 3: 4byte 遠隔書き込み時の送信側信号波形

図 3にメモリベース通信の成否を返答するオプション付きの 4byte MBCF_WRITE 時のロジック信号波形を示す。この例では成否返答オプション付きの遠隔書き込みが終了する度に、同じアドレスへの同じメモリベース通信を繰り返している。この信号波形は遠隔書き込み要求側の波形であり、信号 A0_03 から A0_06 は送信データを A1_01 から A1_04 は送信側の受信データ (ステータスの返答パケット到着) を示している。 C_03 は low で supervisor アクセスモードに居ることを示す。図 3の C_03 の最初の low 期間はステータスの返送による受信割り込み処理区間 (8.0μsec) を示す。 2 回目の low 期間が遠隔書き込み送信のためのシステムコール期間を示す。この図では **3.80μsec** である、他の通信箇所では **3.40μsec** も観測された。ソフトウェアによるタイマ測定では、 5.0μsec を下回ることがないのは計時カウンタの読み出しオーバヘッドによるものである。なお、 3 回目の C_03 の low 区間は Fast Ethernet Adapter の送信終了割り込みによる処理期間である。

同一条件において、送受信つまり書き込み要求から成否返答パケットの処理終了までの時間を測定した。測定値は **50.0μsec** 近辺に集中しており、最良値は **48.40μsec** であった。この値は 100baseTX の MBCF の Round-trip タイムに他ならない。

同じく 4byte MBCF_WRITE を成否返答オプションなしの条件を用いて、受信側のオーバヘッドを測定した。キャッシュミスの発生する初回のアクセスを除き、 **6.40μsec** 程度に観測値が集中する。この値が受信割り込みルーチンのオーバヘッド値である。

4.3 MBCF と MPP の通信性能比較

以下に、高並列計算機 (MPP) のソフトウェアを含んだ通信性能を参考に表 5に掲げる。なお、これらは保護および仮想化の度合が MBCF と比べて低く[5]、機能的にも大幅に劣るので、本来は定性的側面から比較対象から除外されるべきものである。

表中の SSAM と MBCF 以外は専用通信ハードウェアを持つ並列計算機システムであり、通信ハードウェア自体の性能は今回の MBCF 実装が使用した 100baseTX よりも大幅に高い。 SSAM[6] は MBCF と同様にワークステーションクラスタベース (ただしネットワークは 156Mbps ATM) のソフトウェアによる通信機構である。ただし、SSAM はパケットの到着保証と順序保証のプロトコルを省略しているため、実用化時にはこの値よりも悪くなることが予想される。

[4]完了フラグの返送は MBCF コマンドのオプションの一つである。

[5] ノードを跨るギャングスケジューリングが強制されたり、通信ネットワークが一つのアプリケーションに占有されたりする。

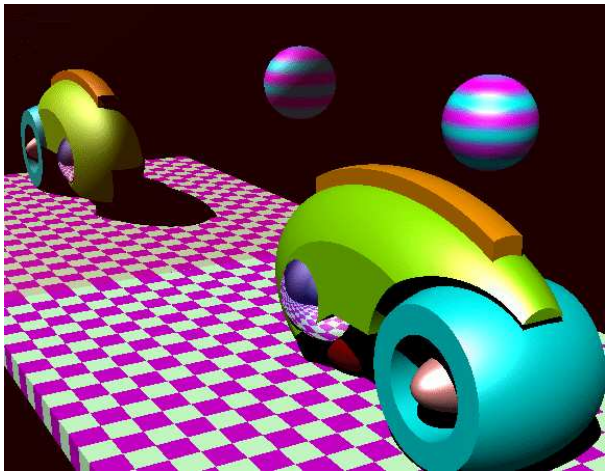

図 4: 測定用レイトレ画像 (実際は 256 色)

表中の SP–2 の二つのエントリは MPL/udp が UDP/IP 上に作られたユーザ通信インタフェースを使用した場合の性能、 MPL/p が一つのアプリケーションが高速通信ネットワークを占有する通信インタフェースを使用した場合の性能である。保護や汎用性の側面を考慮すると MBCF と比較すべき数値は MPL/udp の方である。

表 5中の MBCF の Round-trip タイムはロジックアナライザによる厳密測定の値を採用している。

SP–2 の性能値は文献 [7] から引用し、表中の MBCF および SP–2 以外の性能値は文献 [6] から引用した。

表 5: 通信性能比較 (MBCF vs MPPs)

Machine	Peak band (Mbytes/s)	Round-trip latency(μs)
SP–1 + MPL/p	8.3	56
Paragon + NX	7.3	44
CM–5 + Active Message	10.0	12
SP–2 + MPL/udp	10.8	554.0
SP–2 + MPL/p	35.5	78.0
SS20 cluster + SSAM	7.5	52
SS20 cluster + MBCF	**11.2**	**49**

我々の MBCF は CM–5 上の Active Message (AM) に Round-trip タイムで劣っているが、 MBCF が一切特殊ハードウェアを必要としないこと、 AM/CM–5 が仮想化や保護に十分に対応していないことを考慮すれば、我々の MBCF の方式および 100baseTX 上の MBCF の実装が非常に優れていることが判る。現在、 MBCF の通信能力 (Peak bandwidth) は 100baseTX の性能によって上限が規定されているため、通信ハードウェアとして他の MPP と同程度に高速なものを使用すれば、さらに大幅な性能アップが期待できる。

4.4 アプリケーションによる性能測定

並列レイトレーシングプログラム (以下レイトレ)[6]を使って、 SSS–CORE 上で複数台数のワークステーションによって並列計算を行い、 MBCF_WRITE で 1 台のワークステーションにフレームバッファ表示を行う実験を行った。 10baseT による MBCF を使ったこの実験は文献 [1] で報告しているが、送受信ルーチンの完成度ならびに前回報告時点より MBCF の通信保証や保護仮想化機能が強化されているため、 100baseTX のみではなく 10baseT の MBCF の性能についても再度測定し本稿に掲載する。

実験に使用した 3D ソリッドモデルの完成時の絵を図 4に示す。この絵を 576 × 450 の解像度で並列計算を行った。並列計算の方法はサイクリックに N ピクセルずつを割り当て、 N ピクセルのカラー値 (R,G,B 各 8bit) を計算した後、 dither 変換を掛けた N ピクセル分の Nbyte データを 1 パケットとして表示ノードのフレームバッファに MBCF_WRITE で直接書き込む (100baseTX MBCF では 5 台のうち 1 台、 10baseT MBCF では 8 台のうち 1 台は表示専用とした)。ただし、各プロセッサ (各ワークステーション) はすべて 576 × 450 回のループを回り、ピクセルが自分の担当領域かどうか実行時に判断し、担当外であればスキップしている。上記条件で並列レイトレを実行し、完全に画像生成が終了するまでの全計算時間を計測した。時間は 100msec 単位で計測した。 10baseT 版 MBCF による実験結果を表 6、 100baseTX 版 MBCF による実験結果を表 7に示す。

本実験における 1 ピクセル当たりの計算時間は平均約 70μsec(ピクセル上の絵の複雑さによって大幅に変化) である。オーバヘッドによる実行遅延がなければ、各ノードはピクセル当たり計算時間の転送サイズ倍の時間間隔でパケットを表示ノードに送り、表示ノードには 1 ピクセル当たり計算時間のサイズ倍を台数で割った間隔でパケットが到着し処理が要求される。並列レイトレは並列アプリケーションとしては構造が非常に単純であるが、上記のように転送データのサイズと計算台数を調節することでコンピュテーションインテンシブな振舞からコミュニケーションインテンシブな状況まで自由に調整することができる。

表 6でアスタリスクが付いている項目は 10baseT Ethernet の通信競合により大幅にプロセッサごとの実行時間が変動し処理全体の実行時間も安定しない (項目にもよるが結果に± 0.3sec 程度の幅が存在する) 実験項目である。表内には 3 回測定した平均が記入されている。なお、 10baseT および 100baseTX 共に転送データサイズが 32byte 以下のパケットではヘッダーやダミー等も含めて 76 バイト分のパケットが毎回通信されている。例えば、 4byte パケットの通信では 4.92Mbyte のデータが送られる計算となる。これから 10baseT 4byte/6 台の Ethernet の転送レートは 677Kbyte/s に達し、 CSMA/CD 方式として通信量が飽和状態である。他のアスタリスクの項目も同様である。つまり、 10baseT の MBCF では 1 台の 1 パケットのデータ量が 4byte 以下になるとパケット数増加に

[6]北海道大学の山本強先生のプログラムを C 言語で書き直し並列化した物を使用している。

表 6: 10baseT MBCF: 転送サイズと台数による並列レイトレ計算時間

サイズ (byte)	64	32	16	8	4	2	1
1 台時間 (sec)	16.90	16.97	17.12	17.39	17.96	19.26	24.63
2 台時間 (sec)	8.67	8.75	8.83	8.94	9.26	12.96	*23.43
3 台時間 (sec)	6.41	6.07	6.07	6.10	7.85	*12.93	*23.16
4 台時間 (sec)	4.55	4.60	4.65	4.83	*7.48	*12.95	*23.54
5 台時間 (sec)	3.73	3.74	3.77	*4.28	*7.34	*12.86	*23.46
6 台時間 (sec)	3.43	3.44	3.32	*4.26	*7.27	*13.04	*23.59
7 台時間 (sec)	2.79	2.80	2.88	*4.16	*7.44	*13.06	*23.65

表 7: 100baseTX MBCF: 転送サイズと台数による並列レイトレ計算時間

サイズ (byte)	64	32	16	8	4	2	1
1 台時間 (sec)	16.87	16.93	17.03	17.23	17.64	18.42	20.02
2 台時間 (sec)	8.65	8.73	8.78	8.82	9.00	9.37	10.24
3 台時間 (sec)	6.40	6.05	6.03	6.02	6.15	6.39	6.88
4 台時間 (sec)	4.55	4.59	4.62	4.67	4.72	4.91	5.29

よる送信コストの増加を大幅に上回る実行時間の増加が発生する。これは通信網の飽和によって、全計算時間がピクセルの計算時間ではなく、通信の転送時間の総和で規定されるようになったためである。なお、保護と仮想化の度合は高まり、通信到着が完全に保証されるようになったにも関わらず、 MBCF ルーチンの完成度の向上により、10baseT 版　MBCF の 1 台でデータサイズが小さい場合（つまり送信または受信オーバヘッドがそのまま全計算時間に反映する場合）は、計算時間が以前の報告 [1]（例えば 1byte/1 台: 37.7sec ）に比べて大幅に改善している。

100baseTX 版 MBCF（表 7 ）では転送能力が大幅に改善されるため、転送単位が 4byte 以下になっても台数に従ったほぼリニアな性能アップが得られている。ちなみに 1byte/4 台時の 100baseTX のデータ転送レートはヘッダー等を含めて 3.72Mbyte/s に達している。表示ノードが処理するパケットレートは 約 49,000packet/s に達している。

比較的大きな粒度の通信においても、台数に関してリニアスピードアップに達していないのは、各プロセッサが担当以外のピクセルに対して若干の処理 (自分の担当かどうかの判定) を行っていることと、 Ethernet 通信の衝突に起因するものである。

並列レイトレを用いて、 UNIX において軽いと言われている UDP を使ったソケット通信による性能を参考のために測定した。このために UDP のソケットで通信するプログラムに前記レイトレを書き換えたものを用意した。表示プログラムは SunOS 4.1.4 上の X11R6 で表示し (ただし XFlush はしない) 、計算プログラムは SunOS 4.1.4 上で動かした。使用マシン環境は SSS–CORE 用の環境の OS を交換してまったく同じにした。アプリケーションのコンパイル条件も同一のコンパイラを用い、最適化オプション (O4) も同一にした。表 8に実験結果を示す。測定は実時間で行い、 UDP 通信の送信側は送信バッファに書き込めない場合は書き込めるまで待つようにプログラムを作った[7]。しかし、オーバヘッドを必要以上に増やさないため、それ以上の転送の保証を表 8の測定においては行わなかった。そのため、 2 台以上の並列計算実行では通信の衝突により大幅にピクセルを取りこぼす結果となった。計算結果が正しく表示される場合 (1 台でレイトレ実行し 1 台で表示した場合) のみを表に示している。ただし、アスタリスクをつけた 100baseTX の UDP による通信の 1byte 単位の転送のケースでは、 1-to-1 の転送にもかかわらずピクセルの値が少なからず表示ノードまで届かなかった。

UDP による通信のレイトレと MBCF の 1 台のケースのレイトレと比較すると、すべてのケースにおいて MBCF の性能が上回っている。特に 8byte 単位以下といった細粒度のデータ転送では UDP による通信のオーバヘッドが実行時間に大きく悪影響を与えている。

単純な UDP 通信では、ピクセルの取りこぼしが問題となり、並列実行時の性能測定が不可能であるため、 Ethernet 用の MBCF の転送保証プロトコルを UDP/SunOS 上に移植して、 MBCF/udp-sunOS を作成した。低レベルの送受信ルーチンを UDP/IP のライブラリ / システムコールに置き換えて、他の上位プロトコルのプログラムは MBCF/100baseTX をそのまま使用した。

10baseT による MBCF/udp-sunOS の性能を表 9 に、100baseTX による MBCF/udp-sunOS の性能を表 10に示す。これらの実験では、全般に 10baseT 版 MBCF/udp-sunOS の方が 僅かであるが 100baseTX 版 MBCF/udp-sunOS よりも性能が高い。並列効果は 2 台以上では得られず、 2 台以上の並列処理のすべてで、実行時間に 1 秒内外のバラつきが見られた。表 9および表 10では、各条件において 2 回測定して良い (小さい) 方の値を採用した。MBCF/SSS–CORE に比べて大幅に性能が悪く、 MBCF

[7] この同期を行わないと 1 台で計算して 1 台で表示する状況でもピクセルが大幅に欠落する。

表 8: UDP 通信による並列レイトレ計算時間 (1 台計算 & 1 台表示、比較用)

転送サイズ (byte)	64	32	16	8	4	2	1
10baseT 版 UDP (sec)	17.6	18.5	19.5	22.2	29.7	41.5	68.4
100baseTX 版 UDP (sec)	17.3	17.9	18.8	20.7	24.7	32.6	*47.7

表 9: 10baseT MBCF/udp-sunOS: 並列レイトレ計算時間 (比較用)

サイズ (byte)	64	32	16	8	4	2	1
1 台時間 (sec)	18.04	19.25	21.50	26.94	36.91	56.74	96.33
2 台時間 (sec)	10.34	11.61	13.35	19.13	28.11	45.67	76.14
3 台時間 (sec)	10.90	11.61	13.15	19.18	28.42	47.39	78.79
4 台時間 (sec)	9.80	11.30	14.44	19.04	30.62	46.08	75.18

表 10: 100baseTX MBCF/udp-sunOS: 並列レイトレ計算時間 (比較用)

サイズ (byte)	64	32	16	8	4	2	1
1 台時間 (sec)	18.19	20.69	22.12	27.36	38.68	59.31	99.45
2 台時間 (sec)	10.78	12.04	14.16	19.07	28.33	48.13	77.10
3 台時間 (sec)	11.81	11.85	13.88	18.30	26.54	46.26	80.41
4 台時間 (sec)	10.81	11.45	13.60	18.37	27.31	45.79	75.96

の高速実装には低レベルドライバとプロトコルの一貫性が重要であることがわかる。

5 おわりに

保護と仮想化の徹底した高速ユーザレベル通信であるソフトウェアメモリベース通信 (**MBCF**) を 10baseT ならびに 100baseTX を使って、そのサポートオペレーティングシステム SSS–CORE と共に実装した。 100baseTX 版 MBCF の基本性能を測定したところ、 Peak bandwidth が **11.2**Mbyte/s、 Round-trip latency が 49μs であった。これらの値は、 MBCF の現実装より大幅に転送能力の高い通信ハードウェアを持ち仮想化や保護のレベルが低い並列計算機のユーザレベル通信能力と比べて、優とも劣らないものである。

さらに、 MBCF を既存オペレーティングシステム上の UDP/IP 上に実現して並列アプリケーションによる性能比較を行ったところ、粒度の小さな通信では既存オペレーティングシステムの通信オーバヘッドで SSS–CORE 上の MBCF に大きく及ばないことが明らかになった。 MBCF のようなユーザレベル通信を高速に実現するためには、ハードウェアレベルのドライバからユーザインタフェースまで一貫したポリシで開発することが重要である。

今回、我々が使用した SS20 タイプのワークステーションはすでに一世代古いタイプのマシンとなっている。そこで、現在最新鋭のワークステーションへの MBCF ならびに SSS–CORE の移植作業を進めている。これと並行して 100baseTX より高速なファイバチャネルインタフェースや Gigabit Ethernet を使った MBCF の実装も進める予定である。これらの研究開発作業が進めば、レイテンシは現在の 3 分の 1 から 5 分の 1、ピーク転送能力は 5 倍から 10 倍に性能アップが可能となる予定である。

謝辞

本研究は情報処理振興事業会 (IPA) が実施している独創的情報技術育成事業の一環として行なった。 MBCF を実装するベースとなった SSS–CORE の共同開発者の株式会社アックスの渦原茂氏、 Sun ワークステーションのハードウェア情報を提供していただいた日本サン・マイクロシステムズ株式会社および株式会社物産マイクロエレクトロニクス、ならびに研究室の研究開発環境を良好に保ってくれている東大平木研研究室の構成員各位に心より感謝いたします。

参考文献

[1] 松本 尚, 平木 敬: 汎用超並列オペレーティングシステム: SSS–CORE — ワークステーションクラスタにおける実現 —. 情報処理学会研究報告 96–OS–73, 情報処理学会, Vol.96, No.79, pp.115–120 (August 1996).

[2] 松本, 駒嵐, 渦原, 平木: メモリベース通信による非対称分散共有メモリ. コンピュータシステムシンポジウム論文集, 情報処理学会 pp.37–44 (November 1996).

[3] 松本 尚, 平木 敬: 汎用並列オペレーティングシステムにおける資源保護と仮想化. 情報処理学会研究報告 97–OS–75, 情報処理学会, Vol.97, No.56, pp.37–42 (June 1997).

[4] 松本 尚, 平木 敬: Memory-Based Processor による分散共有メモリ. 並列処理シンポジウム JSPP '93 論文集, pp.245–252 (May 1993).

[5] 松本 尚, 平木 敬: 汎用並列オペレーティングシステム SSS–CORE の資源管理方式. 日本ソフトウェア科学会第 11 回大会論文集, pp.13–16 (October 1994).

[6] T. von Eicken, A. Basu, and V. Buch: Low-Latency Communication Over ATM Networks Using Active Messages. *IEEE Micro*, pp.46–53 (February 1995).

[7] M. Snir et al.: The Communication software and parallel environment of IBM SP2 *IBM Systems Journal*, Vol. 34, No. 2, (1995).