100BASE-TX 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

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。我々はこの目標を満たす高速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、100BASE-TX 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、既存オペレーティングシステムの UDP/IP および TCP/IP 上で同様の性能測定を行い、デバイスドライバからユーザインタフェースまで一貫した設計し最適化した我々の提案方式との性能比較を行う。

1　はじめに

並列計算機も実用化時代に入り、多くの商用マシンが開発され実務に供されている。しかし、多くのマシンは複数の独立したジョブを高速に処理するサーバ機として使用されており、汎用の並列計算環境つまりマルチジョブ／マルチユーザ環境における高効率の並列処理はまだ実用レベルではない。一方、LAN 用ネットワークの高速化に伴って、高速ネットワークで複数台のマシンを結合したワークステーションクラスタ（Network of Workstations: NOW）やサーバ機クラスタが注目を集めるようになってきている。現状ではまだ汎用の LAN 用ネットワークの性能が十分でない面があるが、データベース共有やファイル共有の分散処理程度であれば、並列計算機に取って代わりそうな勢いである。

汎用並列計算機やワークステーションクラスタにおける次なるチャレンジとしては、現在の分散処理環境と同等の汎用環境を維持しつつ、スケーラビリティと並列処理による高速性を安価に実現することである。汎用性と並列処理性能の高さから集中共有メモリやハードウェア分散共有メモリを持つ商用並列計算機の発売が相次いでいる。しかし、専用マシンは量産効果がなく非常に高価である。やはり、NOW のような量産高価を活かせる方向で汎用高性能並列計算機を開発する必要がある。

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を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 の概要と高速実装技術を簡単に説明し、汎用の通信機構である 100BASE-TX (Fast Ethernet)[4] および 10BASE-T (Ethernet)[5] 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。その報告過程において、既存 OS 上の UDP/IP および TCP/IP を使った通信方式と性能比較し我々の MBCF 方式の有効性を示す。

2　ソフトウェアメモリベース通信

2.1　MBCF の概要

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

---

*Performance of Memory-Based Communication Facilities Using Fast Ethernet (100BASE-TX).
Takashi MATSUMOTO and Kei HIRAKI,
Department of Information Science, University of Tokyo.

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**)[6] 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。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。これに対して松本が出した回答が MBCF である。

MBCF は MBP の動作と機能を付加ハードウェアなしにソフトウェアのみで実現している。このため、通信同期の端緒は単なる load/store 命令ではなく、通信相手（タスク／プロセス）の論理的な識別子と通信相手における操作対象論理アドレスを含んだ通信パケットをユーザレベルで構成し、MBCF 送信用のシステムコールを実行する。システムコール内で汎用の通信ハードウェア（100BASE-TX インタフェース等）を利用して送信する。受信側は受信割り込みルーチン内でパケット内の情報を利用して通信先タスクの操作対象アドレスを直接操作する（返答が必要であれば返答を送信する）。ユーザはメモリを介して通信を行うため、ノード間の同期を低コストの Snoopy Spin wait（実行フロー切替えオプション付きのスピンウェイト）で行うことが可能となる。

2.2 MBCF の高速実装技術

MBCF は以下に述べるソフトウェア技巧と最新アーキテクチャを駆使して、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。本高速実装技術には、ソフトウェア的な方式選択の側面と最新プロセッサアーキテクチャを活用している側面がある。必ずしも技術毎に厳密な区分ができるわけではないが、二つ側面に分けて記述する。

2.2.1 ソフトウェア技巧

- **論理アドレスによる通信相手空間の直接操作**
  固定されたキューを介したデータ通信を行わないため、通信データのコピー回数ならびにキュー操作を大幅に削減できる。逆に、キュー構造が必要な場合はメモリの任意の位置にキューを設定することができる（Memory-Based FIFO）。
- **高性能プロセッサのローカル処理の高速性**
  キャッシュミスが少なくなるように最適化されたプログラムを用意して MBP の機能をメインプロセッサによって実現すれば、処理オーバヘッドは大きくない。
- **軽い送信専用システムコール**
  既存の OS（UNIX 等）の通信用システムコールはオーバヘッドが大きい。その理由は、通信パケットのコピー回数の多さと、通信と無関係な処理を行っているからである。SSS–CORE の MBCF 実装では ether の通信保証に必要な最低限の一回のコピーで実装されており、通信と無関係な処理はまったくなされていない。また、専用システムコールでノンブロッキングであることが保証し、プロセッサコンテクストの退避／復旧も最低量（システムコール内で使用する資源のみ）で済ませている。
- **軽い専用受信割り込みルーチン**
  送信専用システムコールと同様に、余分な処理を一切せず MBCF に特化した受信割り込みルーチンを用意した。また、割り込み処理ルーチンの共通化といった高速化を阻害するプログラミング手法は一切排除した。
- **多機能かつ機能固定の受信割り込みルーチン**
  受信割り込みルーチン内つまりカーネルモード内で処理を行うために、ユーザプログラムを受信ルーチンとして使用することを許していない。保護と仮想化の下でユーザプログラムを受信ルーチンとして使うためには余分なコピーの発生が避けられない。

2.2.2 汎用高性能プロセッサアーキテクチャの活用

- **複数コンテクストの混在できる TLB**
  最近の高性能プロセッサの TLB はコンテクスト識別子を含んでおり、複数のコンテクストが混在できる。このため、異なるアドレス空間（コンテクスト）を操作する MBCF コマンドが割り込みで実行される際に、多くとも MBCF で操作するアドレスの TLB をセットするだけで済み、TLB の内容はほとんど影響を受けないため割り込み後の実行速度も低下しない。
- **軽いアドレス空間切替えハードウェア**
  現在のコンテクスト識別子を更新する CPU 命令は 1 命令であり、実行クロック数も小さい。

- **カーネル内ユーザ権限メモリアクセス**
  カーネル内でユーザ権限でメモリアクセスが低コストで可能な機能が最近のプロセッサには搭載されている。この機能により余分なコードを付加せずにカーネルルーチン内でユーザの替わりにメモリアクセスが可能になる。
- **ページエイリアス機能**
  ページエイリアス機能と前出の複数コンテクストの混在できる TLB を活用することにより、完全なアドレス範囲チェックをソフトウェアによって行わなくても、MBP と同レベルのメモリ保護を行うことができる [3]。
- **物理アドレスタグを持つプロセッサキャッシュ**
  ページエリアスを利用する場合に、コンシステンシを保つために必要な条件である。また、各ノードのワークステーションがマルチプロセッサ構成になっていてもキャッシュコンシステンシを保つことができる。

なお、後者の機能の全部が満たされないプロセッサでも、若干の性能低下はあるが、不足部分をソフトウェアエミュレーションすることで MBCF を実装することは可能である。

MBCF の定性的な議論は文献 [2] を、100BASE-TX 版および 10BASE-T 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　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、10BASE-T 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、100BASE-TX 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[7, 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。

3.1　100BASE-TX 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。開始と終了時の二回の時計読み出しオーバヘッドにより、計測した値は約 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。round-trip 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送信し、対象メモリを読み出し後、データをパケットで返送し、要求元に指定されたメモリ領域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。round-trip 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。round-trip 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 100BASE-TX による MBCF の Round-trip タイムを示す。

性能比較として、表 1の内容と、同一ハードウェアに対して SunOS4.1.4 の TCP/IP（図中 TCP100/SunOS）および UDP/IP（図中 UDP100/SunOS）ソケットライブラリを用いた場合の Round-trip タイムと転送パケットサイズの関係を、図 1に示す。ただし、TCP/IP のソ

表 1: 100BASE-TX 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

表 2: 10BASE-T 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

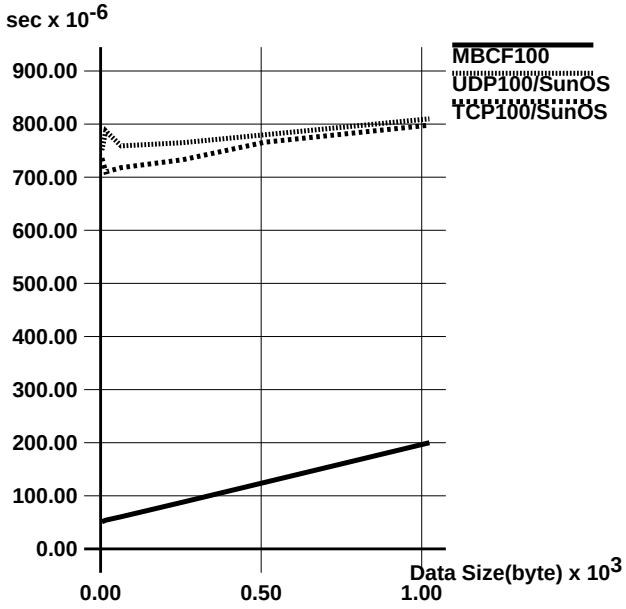

図 1: Round-trip タイムの比較

表 3: 100BASE-TX 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

ケットには TCP_NODELAY オプションを付加し、ストリーム通信を基本とする TCP/IP が細粒度通信実験において不利にならないようにした。UDP/IP のケースでは、UDP/IP のみではパケットの転送保証および順序保証が行なわれないため、UDP/IP 上に MBCF のパケット転送プロトコルを実装し測定した。64byte 以下のデータ長のパケットでは、前出の高速実装技法を駆使した SSS-CORE 上の MBCF は、SunOS のソケットライブラリを使った通信に比べ、10 分の 1 以下のレイテンシで通信が可能である。また、UDP100/SunOS および TCP100/SunOS のグラフではパケット内のデータサイズ増加によるオーバヘッドの増加が、キャッシュ配置等による測定誤差と同程度となりはっきりしない。

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

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、10BASE-T の場合も、送信ルーチンが 100BASE-TX 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 4に 100BASE-TX と 10BASE-T 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[1]。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（100BASE-TX: 12.5Mbyte/s, 10BASE-T: 1.25Mbyte/s）を使い切ることに成功している。

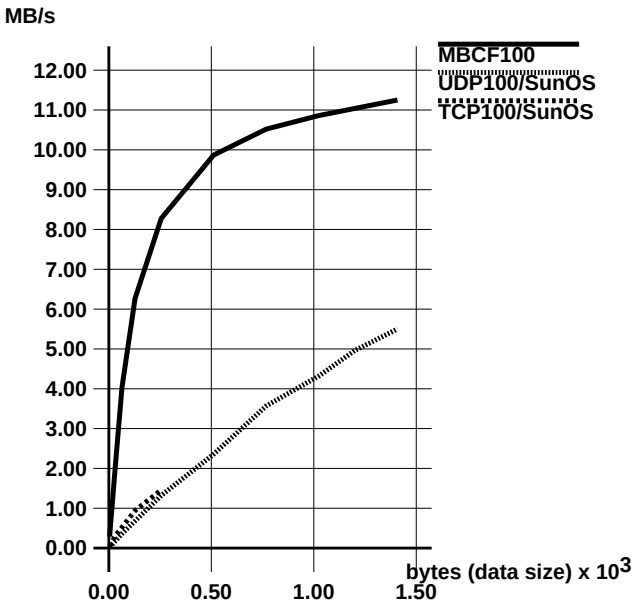

図 2: Peak bandwidth の比較

比較のため、Round-trip タイムと同様に SunOS 4.1.4 上の TCP/IP と UDP/IP についても MBCF と同じ Peak

[1]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。

表 4: 100BASE-TX/10BASE-T のメモリベース通信の Peak bandwidth

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

bandwidth を測定する実験を行った。UDP は前述のように MBCF の通信プロトコルを使って（プログラムのプロトコル部分を流用して）、通信保証を行った。100BASE-TX 上の MBCF のバンド幅性能と共に測定結果を図 2に示す。TCP/IP の測定がパケットデータ長が 32byte で途切れているのは、今回の測定の様に少々の通信競合は無視して eager に送信をするような場合には、TCP/IP の slow start や競合時の速度減速プロトコルが災いして、大幅に性能が低くなるためである。つまり、今回の実験のデータ送信は一方向であるが、割合は少ないが Acknowledge が返送されるのため、64byte 以上では通信競合によって測定不能な程性能が低下した。MBCF と UDP100 を比較すると、データ長が短い場合は MBCF の性能の方が圧倒的に高い。100BASE-TX の性能の上限のため、データ長が 1000byte を越える辺りで MBCF は性能グラフが飽和傾向にあるが、UDP100/SunOS ではデータ長を大きくしても 100BASE-TX の性能上限の半分の性能も出すことができない。

3.2 ロジアナによる基本性能の厳密測定

100BASE-TX 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

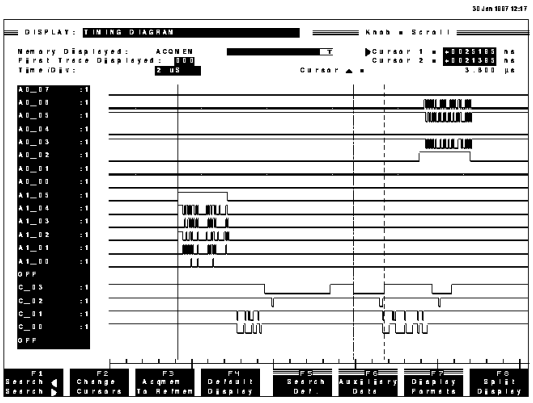

図 3: 4byte 遠隔書き込み時の送信側信号波形

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 100BASE-TX 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。

3.3 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[2]、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 100BASE-

[2] 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。

TX よりも大幅に高い。SSAM[8] 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。

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 [9] から引用し、表中の MBCF および SP–2 以外の性能値は文献 [8] 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 100BASE-TX 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 100BASE-TX 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。

3.4 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）[3]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。10BASE-T による MBCF を使ったこの実験は文献 [1] においても報告しているが、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、100BASE-TX のみではなく 10BASE-T の MBCF の性能についても再度測定し掲載する。

[3] 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。

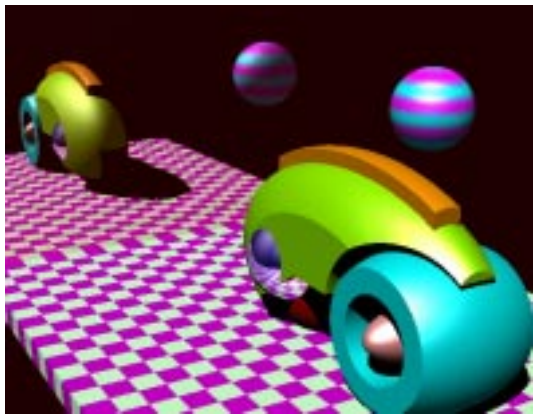

図 4: 測定用レイトレ画像（実際は 256 色）

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（100BASE-TX MBCF では 5 台のうち 1 台、10BASE-T 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。10BASE-T 版 MBCF による実験結果を表 6、100BASE-TX 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 10BASE-T 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

表 6: 10BASE-T 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: 100BASE-TX 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

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、 10BASE-T および 100BASE-TX 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 10BASE-T 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、 10BASE-T 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、 10BASE-T 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。

100BASE-TX 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 100BASE-TX 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、 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ようとした。 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、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。しかし、オーバヘッドを必要以上に増やさないため転送保証を行わないと、 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り測定不可能であった。

そこで、基本性能の節でも説明した Ethernet 用の MBCF の転送保証プロトコルを UDP 上に移植して、 MBCF と機能的に同等な UDP100/SunOS (100BASE-TX) および UDP10/SunOS (10BASE-T) を用いた並列レイトレの性能評価に切替えた。

10BASE-T による UDP10/SunOS の性能を表 8 に、 100BASE-TX による UDP100/SunOS の性能を表 9に示す。これらの実験では、全般に 10BASE-T 版 UDP10/SunOS の方が 僅かであるが 100BASE-TX 版 UDP100/SunOS よりも性能が高い。しかし、いづれの場合も並列効果は 2 台以上では得られず、 2 台以上の並列処理のすべてにおいて実行時間に 1 秒内外のバラつきが見られた。表 8および表 9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–

表 8: 10BASE-T UDP10/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 9: 100BASE-TX UDP100/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

CORE に比べて大幅に性能が悪く、 MBCF 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。

4 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 10BASE-T ならびに 100BASE-TX 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。100BASE-TX 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、優るとも劣らないものである。

さらに、 MBCF と同等の機能を既存オペレーティングシステム上の UDP/IP 上に実現して性能比較を行ったところ、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 に能力が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て検討を加え、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 100BASE-TX 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われた。 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技術情報を提供していただいている日本サン・マイクロシステムズ株式会社、ならびに研究開発環境を良好に保ってくれている東京大学平木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] IEEE: IEEE Std 802.3u-1995 CSMA/CD Access Method, Type 100BASE-T. IEEE, New York (October 1995).

[5] ISO/IEC: ISO/IEC 8802-3: 1996 (ANSI/IEEE Std 802.3, 1996 Edition) CSMA/CD. IEEE, New York (July 1996).

[6] 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).

[7] 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).

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

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