論文の内容の要旨

論文題目　A Study on Memory-Based Communications and Synchronization in Distributed-Memory Systems
分散メモリ環境におけるメモリベース通信同期に関する研究

氏名　松本尚

共有メモリモデルにおける記述力はメッセージパッシングを基本とするプログラミングモデルと同じである。言い替えると、どちらのモデルでも他方をエミュレートすることが可能である。しかし、性能の観点から見ると事情は異なっている。メモリはコンピュータにとって最も基本的な構成要素の一つであり、メモリ管理機構によって保護され仮想化された状態で、プロセッサが直接アクセスが可能である。このため、通信や同期を行うサブシステムにおいて、メモリに関する情報が使用できることが高速化および多機能化にとって非常に重要である。この事実は通信同期がハードウェア実装されているかソフトウェア実装されているかに依存しない。もう少し具体的に述べると、通信先の最終目的地のバッファアドレスを通信サブシステムが認識していれば、通信サブシステム内で直接その領域へのデータ転送を済ませてしまうことが可能になる。この転送はメモリへの転送であるため、プロセッサのメモリ管理機構と同様の保護機構を使用することにより、保護と仮想化も実現可能である。以上の原理に基づいた三種類のメモリベース通信同期機構を考案し、その内の二種類は実装技術の研究と実用性の評価のために実際に開発を行った。

最初の機構は Memory-Based Processor (**MBP**) と呼ばれる細粒度処理用のコプロセッサである。MBP はメインメモリ内に実装され、通信同期のような局所性を活かせない処理を担当する。一方、MBP を補完する相手としてメインプロセッサには大きなレジスタセットとキャッシュによって局所性を活用する従来型高性能マイクロプロセッサが使用される。MBP の起動はメインプロセッサからメモリへ出力されるメモリ操作トランザクションであり、共有された領域へのトランザクションは MBP 内の hardwired 回路によってコンシステンシ維持操作がなされる。このコンシステンシ維持に付随してメモリを介したノード間通信がなされる。MBP はネットワーク上に仮想化された論理アドレス空間（ネットワークアドレス空間）を張っており、ノード内の物理アドレスからネットワークアドレスに変換する C-TLB と、ネットワークアドレスからノード内の物理アドレスに変換する N-TLB をアドレス変換の高速化のために内蔵している。また、これらの TLB 内のエントリをオペレーティングシステムによって適正に管理することによって、ページ単位でメモリベース通信同期に対して保護を掛けることができる。この二段階のアドレス変換方式により、各ノードは特定のネットワークアドレスに対して自由な物理アドレスを割り当てることが可能になり、ネットワークアドレスを利用にしたキャッシングがノード内のメインメモリを流用して可能となった。MBP が考案されるまでは、ハードウェアベースの分散共有メモリはシステム全体で一意である物理アドレスを使用し、遠隔メモリのキャッシュ用に特別なメモリ、つまり本来の物理アドレスをタグとして保持する機構がついたハッシュメモリが用意されていた。MBP の考案発表以降は、Reflective Memory や Memory Channel といった類似のアドレス変換機構を持った機構が実用化されていった。しかし、Reflective Memory や Memory Channel は更新型のプロトコルしか持たず、プログラムが持つ局所性を破壊してしまうという重大な欠点があった。これに対して、MBP は考案当初から、複数のプロトコルを適材適所で使い分けることが想定されており、一般的な無効化型プロトコルや単純

な遠隔書き込みおよび遠隔読み出しといった操作も可能に設計された。これらのプロトコルをページ単位に指定することが可能であり、変数やデータの性質によってプロトコルを使い分けることが可能となっていた。MBP はページ単位に共有を管理することと先述のアドレス変換方式によって、メインメモリを効率良く遠隔キャッシュに利用することを可能にしている。しかし、この方式では、メインプロセッサによる粒度の細かいメモリアクセスと無効化型プロトコルの相乗効果によってページ単位の無効化が頻発する危険性がある。この危険を回避するために、ページ単位で共有の管理を行うが、データの移動や無効化はメインプロセッサのキャッシュブロック単位で行う方式を確立させた。つまり、メインメモリにはキャッシュブロック単位で状態を示すタグが付与される。アドレス情報や共有情報はページ単位でよいため、キャッシュブロック単位でこれらすべての情報を持つ MBP 考案以前のマルチプロセッサよりも、タグのメモリ量は大幅に少ない。

MBP は超並列計算機の構成要素として考案されたため、MBP の研究開発においてもう一つ重要な技術が考案された。それは、「階層マルチキャストと Ack コンバイニング」と呼ばれる技術である。分散共有メモリではメモリを介して同期を取るためメモリの操作順序が同期の基本となる。このために、どのメモリ操作までが確実に終了しているかがメインプロセッサから確認できる必要がある。そこで、コヒーレンス維持のために発生するマルチキャスト通信において、通信先の各ノードから Acknowledge-message（Ack）をマルチキャスト要求元に返送する必要がある。マルチキャスト先が多くなればなるほど、要求元には Ack の返送が集中して Ack の回収で大きなオーバヘッドが発生してしまう。この問題に関して、通信路に tree 構造を埋め込んで tree の各経路選択場所において階層的に Ack を回収することで、マルチキャスト要求元への Ack メッセージの集中を防ぐことに成功した。この技術は実際の市販並列マシンにも採用されている。

MBP の次に研究開発されたのが、Memory-Based Communication Facility (**MBCF**) である。MBCF は、当初 MBP の機能をオペレーティングシステムの層で効率良くエミュレーションできないかと考えて開発が開始された。しかし、開発を進める上で、MBCF が純粋なソフトウェア実装であることが大きなメリットとなることが判明してきた。通常のパーソナルコンピュータやワークステーションに ethernet を装着するだけで、MBCF システムの計算ノードにすることが可能である。そして、ethernet は通信粒度が 1500byte 程度の中粒度まで許されるため、データ転送の効率を高めることができる。MBP は他のハードウェア分散共有メモリと同様にメインプロセッサのメモリ操作で直接起動されていたため、キャッシュブロックまたはそれより小さい細粒度での通信やコヒーレンス処理を余儀なくされていた。これに対して、MBCF は 1 パケット以下の任意サイズのメモリ操作を可能なメモリベース通信同期として定義され、MBP の遠隔メモリ操作をいわゆる遠隔メモリ DMA に拡張したものとなっている。つまり、MBP の通信粒度を大きくしたものが MBCF である。MBP と MBCF を比較すると、MBCF はメインプロセッサ内蔵の TLB を N-TLB として流用し、MBCF 送信時のパラメータに通信相手の論理アドレスを直接指定することにより C-TLB を省略可能にしたものであることが判る。MBCF の実装技術としても、最新マイクロプロセッサのアーキテクチャ的な特徴を利用した高速化とプログラミングエンジニアリング的な高速化を組み合わせて、保護と仮想化を保ったまま非常に高速なユーザタスク間通信を可能としている。ethernet 上への実装では、100BSAE-TX と SPARCstation 20 の組合せで、最大 11.93MB/sec のデータ転送能力と 24.5μsec の片道レイテンシを達成した。また、1000BASE-SX と ULTRA 60 組合せでは、最大 80.92MB/sec のデータ転送能力と 9.6μsec の片道レイテンシを達成した。最大転送能力が 1Gbit/sec (125MB/sec) に及ばないのは ULTRA 60 のハードウェア的な制約から来るものであり、MBCF のソフトウェアオーバヘッドのみがボトルネックであれば wire speed が十分達成できることを確認した。さらに、通信先の論理アドレスを指定することにより、FIFO 操作や SIGNAL 操作といった高機能操作が論理アドレスで仮想化されて実現される。

図 1 に ULTRA60 と 1000BASE-SX を使用した環境における *SSS–CORE* OS 上の MBCF と Solaris 上の TCP/IP の最大転送レートを転送データサイズを変えながら計ったグラフを示す。図 2 に同じ環境において *SSS–CORE* OS 上の MBCF と Solaris 上の TCP/IP の片道遅延時間を転送データサイズを変えながら計ったグラフを示す。なお、TCP/IP の測定では片道遅延時間の測定では TCP_NODELAY オプションを付加し、転送レートの測定では同オプションを外して少しでも TCP/IP の性能が向上するように優遇してある。しかし、結果は図の通り、遅延時間で十

倍、転送レートで小さなパケットサイズでは格段の差が開いている。*SSS-CORE* は筆者が研究開発中の MBCF を基本通信同期機構として持つ汎用スケーラブルオペレーティングシステムである。

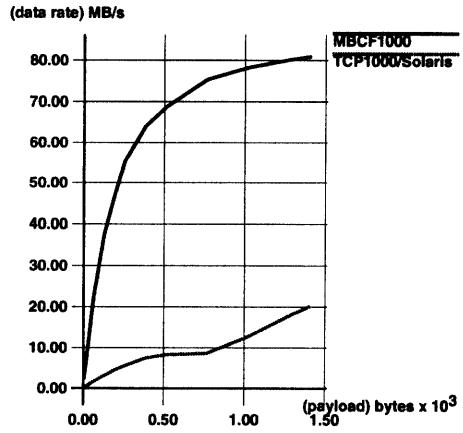

図 1 MBCF と TCP/Solaris の最大転送レート

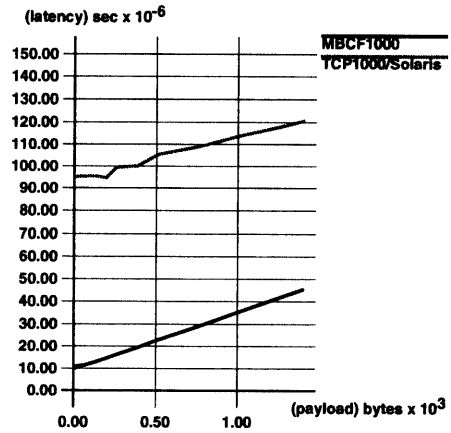

図 2 MBCF と TCP/Solaris の片道遅延時間

MBCF システムによってワークステーションクラスタや PC クラスタに論理的な共有アドレス空間を張ることが可能となり、この空間の下で高速にユーザタスク間通信同期が可能となった。しかし、共有メモリで記述された並列プログラムを効率良く実行するためには、単に共有メモリとしてシステム全体のメモリがアクセスできるだけでは不十分であり、遠隔ノード内のデータが自分のノードにキャッシュする能力、いわゆるソフトウェア分散共有メモリの実現が不可欠である。ソフトウェア分散共有メモリでは、K. Li. によって 1988 年に発表されたメモリ管理機構を流用して、共有データへの書き込みやキャッシュミスの検出をページトラップで行う方法が一般的であった。この方式が開発された当初は、密結合マルチプロセッサの並列プログラムを分散メモリマシンでそのまま走らせようという狙いがあったため、プロセッサの load/store を必要に応じてトラップして通信やコヒーレンス維持動作を実現する K. Li. の手法は画期的なものであった。専用ハードウェアをまったく必要としない点も好評を博していた。しかし、ページトラップ発生後の処理オーバヘッドが大く、複数の共有メモリ操作を同時に行わないと並列処理による性能向上が困難であることが判明してきた。そこで、ハードウェア分散共有メモリの研究活発化と同時に研究が進んだ緩和されたメモリコンシステンシモデルがソフトウェア分散共有メモリにも導入されるようになった。緩和されたメモリコンシステンシモデルでは、共有メモリアクセスをデータ転送のためのアクセスと同期のためのアクセスに分離して、データ転送のためのアクセス間には順序関係を課さずに、複数のメモリトランザクションの同時実行を可能とする。緩和されたメモリコンシステンシモデルの導入と、トラップルーチンの改良でソフトウェア分散共有メモリ上の並列プログラムの性能は大幅に改善した。ただし、この緩和されたメモリコンシステンシモデルの導入はそれまでの大前提、つまり密結合マルチプロセッサとバイナリコンパチビリティのある分散共有メモリマシンを開発するという前提を崩すことになったのである。分散メモリ実装の並列計算機に標準アーキテクチャが存在しないことも、バイナリコンパチビリティが無意味であることをサポートする事実である。これらの事実から導かれる結論として、ページトラップの流用というソフトウェア分散共有メモリの常識を打ち破った新しいソフトウェア分散共有メモリ実現のための方法論を考案した。つまり、ユーザレベルでキャッシュエミュレーションを行うコードをアプリケーションに最適化コンパイラによって挿入し、元のプログラムに対応するコードとキャッシュエミュレーション用に挿入されたコードを一緒にして徹底的な最適化を施すのである。特に最適化として、緩和されたメモリコンシステンシモデルを活用して、複数の細粒度コンシステンシ維持操作をより粒度の大きな通信に置き換えて通信回数を減らす最適化が非常に重要である。また、キャッシュのタグのチェック等も冗長なチェックは削除できる。連続した複数の共有メモリアクセスを解析することによって、多くのプログラムでは通信粒度を大きく改善できることが判った。実際に、MBCF と新方式を適用したソフトウェア分散共有メモリである UDSM/ADSM に対応した最適化コンパイラ RCOP が、筆者の共同研究者（丹羽淳平、稲垣達氏）の手によって研究開発された。RCOP は、今までワークステーションクラスタではスピードアップが難しい（並列処理を行うと性能が大幅ダウンする）と言われてい

たSPLASH-2のRadixやFFTといったプログラムの並列実行による性能向上に成功した。RCOPの成功の後、同種の最適化コンパイラの開発が多く見受けられるようになった。Shastaというバイナリトランスレータが RCOPと対比されるが、Shastaは通信回数や通信量を大幅に削減することを目的としていない点が最大の差であり、そのためShastaは最適化の余地が少ないバイナリトランスレータという形態を採っている。さらに言えば、Shastaの開発はMemory Channelを採用して無効化型プロトコルをソフトウェアエミュレーションによって実現せざるを得ないという事情からなされたものであり、Memory Channelは細粒度低レイテンシで大容量通信が可能な専用通信機構である。RCOPは汎用のLANカードで接続されたワークステーションクラスタ上で効率良く共有メモリ並列プログラムを実行するために開発された最適化コンパイラであり、ソースコードを手続き間解析を含む解析テクニックで徹底的に調べて、通信の回数と量を減らし、通信粒度を可能な限り大きくする。もちろん、本質的に細粒度通信を要求するようなアプリケーションに対しては我々の方式やRCOPは無力かもしれないが、SPLASH-2ベンチマークで見る限り我々のアプローチは大きな成果を上げている。

通常のNICハードウェアを前提とした場合、MBCFは定性的に見て、メッセージパッシング型のインタフェースよりも通信同期システムのインタフェースとして優れている。通信相手のアドレス情報を活かして、コピー回数を減らしたり、特定の領域に単なる書き込み以上の高レベルの操作を施したりすることが可能となる。また、ワークステーション上のActive Message (AM) であるSparcStation Active Message (SSAM) は保護された高速通信がユーザレベルで可能な点で、MBCFと対比され得る数少ない通信方式の一つである。AMは命令ポインタを通信パケット内に持ち、パケットの受信を含む操作自体をその命令ポインタを含むユーザルーチンで行う通信方式である。AMでは受信はユーザが自由にカスタマイズしたルーチンで実現される。しかし、この受信ルーチンに何ら制約がないため、この受信ルーチンにカーネルモードで実行している割り込みルーチン内から制御を受け渡すことができず、UNIX signalのように間接的に受信先のタスクが制御権を持った時に、ユーザモードで実行権を渡す必要がある。この間接的な受信ルーチンの起動がオーバヘッドになり、SSAMは本質的にMBCFよりもオーバヘッドが大きい。MBCFは提供する操作の種類はいくらでも新たに開発可能であるが、大きなコストのかかる処理は実装しないというルールに従って操作を増やしていくため、受信割り込み内で操作を直接行うことが可能である。

MBCFは実装技術を駆使して可能な限り軽量な高速通信として実装されている。しかし、メインプロセッサがプロトコル処理とデータコピー（最小限度の送受信で一回ずつ）を行っている。通信が十分に高速になり、例えばgigabit ethernet等で比較的粒度の小さなMBCF通信が多発した場合は、メインプロセッサがMBCF通信の処理で忙殺され本来の処理ができない事態も考えられる。さらに、セキュリティやプライバシーのために暗号通信が普及した場合は、大きな処理負荷をメインプロセッサに掛けかねない。そこで、通信処理や通信に伴う暗号処理を肩代りするネットワークインタフェースアーキテクチャMemory-Based Processor-II (**MBP2**) をMBCFに基づいて考案し、プロトタイプMBP2Pの開発を行った。開発期間の短縮のためMBP2Pは、マルチチップ構成を採り、組み込みプロセッサとgigabit ethernetのmedia access controllerにそれぞれ市販品であるSPARCliteとXMAC2を使用し、全体の制御用とデータパス用に大規模field programmable gate arrayを採用した。MBP2PはTLBを内蔵しており、ユーザタスクから直接DMAによってパケットを送信することが可能であり、受信時は直接ユーザタスク内のメモリにパケットの内容を反映させることができる。当初の予定通り、大きなサイズのデータ転送ではメインプロセッサの負荷を大幅に軽減することが可能となった。しかし、メインプロセッサのソフトウェアで実現されたMBCFよりもレイテンシは大幅に悪化した。この原因は非力な組み込みマイクロプロセッサにある。メインプロセッサと同程度のクロック周波数で走行していても、組み込みプロセッサは、二次キャッシュの不在、命令キャッシュの容量不足、ライトスルー方式のデータキャッシュ、スヌープ機構の不在という致命的な速度低下要因を複数持っており、メインプロセッサの数分の一程度の実力しかないことが明らかになった。この失敗を教訓に、MBP2アーキテクチャのような用途に使用可能な汎用組み込みマイクロプロセッサCasablancaを共同研究者（田中清文）と研究開発している。ただし、レイテンシのみを取り上げれば、最新の高性能マイクロプロセッサは非常に高速であり、その高速性を活用できるMBCFの方が将来的にもMBP2に優る可能性が大きい。ただし、MBP2は内部に暗号回路等を持つことにより、メインプロセッサの負荷軽減目的には大きな威力を発揮すると考えられる。