2N–05　アドレス変換機能を持つネットワークインターフェイス – メモリベース通信の性能測定 –

國澤 亮太　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

1　はじめに

我々は汎用超並列オペレーティングシステム上でのリモートメモリアクセス手段としてメモリベース通信 [2] を提案してきた。既存のハードウェアを用いる際にはソフトウェアの介在により保護されたメモリアクセスを提供するが、本稿ではネットワークインターフェイスカード (NIC) にアドレス変換機能の一部を持たせることにより削減できるオーバーヘッドを考察する。

2　メモリベース通信

分散共有メモリ型並列計算機においてマルチユーザ / マルチジョブ環境を提供するためのハードウェアとして MBP(Memory-Based Processor) [1] が考案された。ユーザプロセスは共有空間をページ単位で自分のアドレス空間にマップし、 MBP にはその対応関係が知らされている。 MBP はプロセッサのデータアクセスを監視し、共有メモリ空間へのアクセスでかつ自分がローカルに持っていない領域ならば、アクセス先のアドレスをネットワークアドレスに変換してリモートメモリアクセス通信を発行する。

しかし、プログラム中のどの変数を共有空間に割り当てるかはすでに分かっているので、共有空間へのアクセスならばそれを検知してリモートメモリアクセスのためのコードを生成することは可能である。その考えに基づき、 MBP のようなメモリアクセスを監視するハードウェアでなく、リモートメモリアクセスの場合にはユーザプログラムで通信を起動する非対称分散共有メモリシステムが提案された [3]。この時使用されるメモリベース通信は MBP をソフトウェアで実装したものであり、アクセス保護と一貫性管理のプロトコルはページ単位で実装する。通信相手先プロセスの仮想アドレスを指定したリモートメモリアクセスである。ユーザのバッファを直接指定することでコピーの回数が削減できる。

Network Interface card with address translation buffer – evaluation of memory based communication

Ryota KUNISAWA, Takashi MATSUMOTO, Kei HIRAKI

Department of Information Science, Faculty of Science, University of Tokyo

kunisawa@is.s.u-tokyo.ac.jp

3　ネットワークインターフェイス

我々は高速ネットワークを構築する物理層として光シリアル通信に注目し、 Fibre Channel を用いたスイッチングネットワークを構築中である。我々が使用している 1 ギガビットの Fibre Channel トランシーバは送信ポートと受信ポートを持ち、それぞれ 850 メガビット毎秒でデータを転送できる。ネットワークインターフェイスカードは SBus で実装し、 Sun の SPARC Station 20 で使用している。

NIC には 15 エントリの TLB を実装した。 TLB があふれた場合は OS に割り込みをかけ、ソフトウェアによりページ変換をおこなう。 TLB の入れ換えアルゴリズムをソフトウェアによって実現すると OS はアプリケーションにとって最適な置換アルゴリズムを方法を選ぶこと容易になるが、今回は一番古く置換された物を追い出す方法で実装した。

4　SunOS でのメモリベース通信の実装

メモリベース通信のうち、特に非対称分散共有メモリシステムの構築に必要なリモートメモリライトを SunOS 上のデバイスドライバとして実装した。

4.1　送信

SunOS において、ユーザのバッファをデバイスドライバがアクセスできるのは、`read` または`write` システムコールを発行した後に、バッファをカーネルのアドレス空間にマップした時点からである。リモートメモリライトを送信する場合は`write` システムコールを発行した時点でデータが用意されているので、 SunOS の`write` システムコール中でユーザバッファをマップし、送信が終ればアンマップする。`write` システムコールでは送信側の仮想アドレスとデータの大きさしか指定できないので、`write` に先だって`ioctl` システムコールを用いてデバイスドライハに相手先プロセスとその仮想アドレスを知らせる。

4.2　受信

リモートメモリライトを受信した側では、通常の場合ユーザのバッファはマップされていない。ユーザの受

信バッファをマップできるのはreadシステムコールであるが、このセマンティクスはメモリベース通信のリモートメモリライトとは相容れない。readはすでに存在しているデータを読み込むためのシステムコールであり、readを発行した時点でカーネル内のバッファにあるデータをユーザのバッファにコピーするからである。また、SunOSのページ管理機構をNICのページ管理のために使用することはできない。そのため、SunOSでの実装においては通信デバイス毎に受信バッファをあらかじめカーネル内に用意して、常に物理メモリに存在させた。

ユーザプロセスは通信デバイスに対してmmapシステムコールを用いて受信バッファをプロセスのアドレス空間に張り付けることができ、ユーザプロセスは、他人からリモートメモリライトでアクセスされる変数はその範囲のアドレスから割り付ける必要がある。

5 評価

評価は1対1通信で行なった。送信、受信側ともSunのSS20 (50MHzのMbus、25MHzのSBus)を用いている。図1は横軸にメモリベース通信で転送するデータのサイズ、縦軸に転送に費やされた時間をとっている。実線がTLBを実装したNIC、点線がTLBを実装しないNICでの所要時間である。TLBを実装した方は一度プログラムを走らせてNIC上のTLBがすべて有効になった状態から実験を開始し、どちらも1024回データを採取した。データのばらつきはほとんど存在しなかったが、OSの予期せぬ処理が入ってデータ転送に費やされた時間が正確に計測できない場合があった。そのため、一旦データを取ったのち平均値より2000μS以上高い値を捨てて再び平均値を取った。

データサイズが15ページを超えるとTLBは常にフラッシュされるので、グラフは一致する。データサイズが15ページより小さい場所でもグラフが一致してしまっていて、TLBの効果が現れていない。ロジックアナライザを用いた計測によればTLBにヒットした場合(ページエントリの実装方法にもよるが)約27μSのオーバーヘッドが削減できていた。表のグラフの傾きは転送のスループット(約17MB/s)であり、これは1ページを転送するのに約250μS費やしている計算となる。現在のNICのハードウェア性能ではデータ量が1ページに収まっている場合の転送レートは約24MB/s(1ページを163μSで転送)なので、OSによるオーバーヘッドが大きくて効果が現れていない、またOSによるI/Oのスケジューリングによって効果が見えないと考えられる。

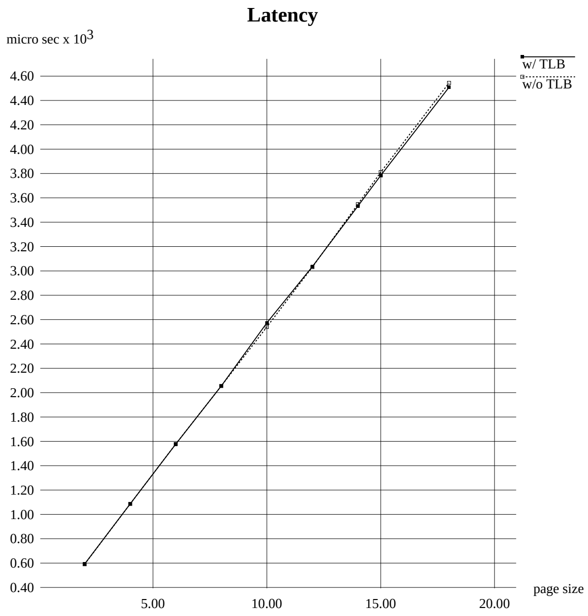

図 1: データサイズと転送時間

6 終りに

ネットワークが十分に速くなった現在、ユーザレベルでの高速な通信を実現するにはOSのオーバヘッドもデータ転送にかかる時間に比べて無視できるほど小さいことが必要であり、またユーザのバッファをデバイスドライバからアクセスできるようにする機能が必要である。我々が現在開発中の汎用超並列オペレーティングシステム*SSS–CORE*ではこの両方の機能を持っていて既存のハードウェアでも軽いメモリベース通信を実現しており、今後は*SSS–CORE*上で本稿のNICを活用していく。

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われた。

参考文献

[1] 松本 尚, 平木 敬, “超並列計算機上の共有メモリアーキテクチャ”, 信技報, CPSY92-96, pp.47–55 (August 1992)

[2] 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*のメモリベース通信機能”, 情報処理学会 第53回全国大会 (September 1996)

[3] 松本 尚, 駒嵐 丈人, 渦原 茂, 平木 敬, “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, pp.37-44 (November 1996)