3F–09

# 高速 I/O バス上のメモリー貫性管理機構を用いたワークステーションクラスタにおける分散共有メモリの実現

滝田 裕, 田中 英彦
{takita,tanaka}@mtl.t.u-tokyo.ac.jp
東京大学大学院工学系研究科

## 1 はじめに

現在、スケーラブルでかつ並列に実行できるプログラムの記述が容易な並列計算機として、分散共有メモリ (DSM) 型並列計算機の研究が盛んに行われている。DSM を専用ハードウェアによって実装したものは高性能であるが高価であり、ワークステーションクラスタ (WSC) 上にソフトウェアで実装したものは安価ではあるが性能がでていない。本稿では、ソフトウェアによる DSM の実装を基本とし、PCI のような高速 I/O バス上の通信インターフェースにメモリ一貫性管理機構を設けることで、WS クラスタにおいても高性能な DSM を実現する手法について述べる。

## 2 従来のソフトウェアによる DSM 実現

ソフトウェアによる DSM 実現は、各 WSC の仮想メモリ空間がプロセス内で一致するようにプロセス毎のページテーブルに手を加える仮想共有メモリという方法で行なわれる (図 1)。

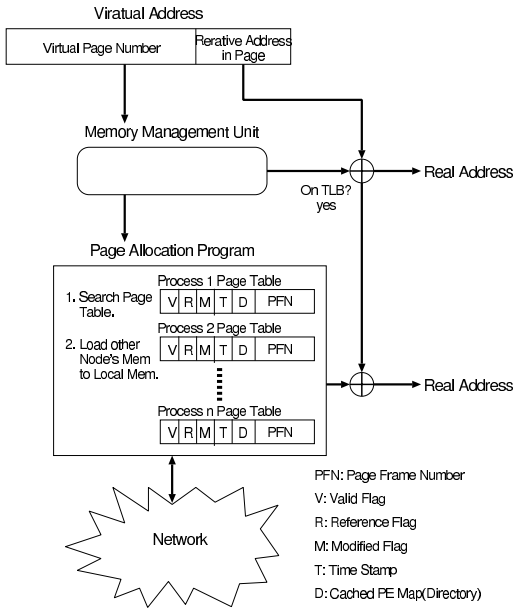


図 1: 仮想共有メモリ

この方法では仮想メモリ管理同様、一貫性管理をソフトウェア (OS) によって行うため、共有メモリへのアクセスはデータを通信するまでの処理に時間がかかり、データの共有の多い一般的な科学技術計算では性能がでない。

*Distributed Shared Memory of Workstation Clusters with Memory Coherent Mechanisms on Fast I/O bus*
Hiroshi TAKITA, Hidehiko TANAKA
Graduate School of Engineering, University of Tokyo

## 3 ノード間通信の改善

データの一貫性保持のため、データを共有するノード間で通信が行われる。通信での速度向上は、共有メモリアクセスのオーバーヘッドを減らす。通信速度を決定するものは、レイテンシ、スループット、品質の 3 点である。本節では、本稿で提案する手法が前提とする、レイテンシ、スループットに関する従来からある通信速度を向上させる技術を列挙する。

### 3.1 スループットの改善

光ファイバーのような高品質の回線を使用した場合、通信で使用するクロックを上げることで 1Gbit/sec 程度のスループットを出すことができる。実際に製品として、HIPPI, Fiber Channel, Gigabit Ethernet 等が存在している。

### 3.2 レイテンシの改善

日立の SR-2201 のリモート DMA や富士通の AP-1000 の Line sending のように、ユーザープロセスのデータ領域間でカーネル内のバッファを使用せずに直接データの受渡しを行う機構を設けることが可能になっている。これを利用することで、送信前と受信後のデータのコピーを省くことができ、通信にかかる時間を削減できる。

## 4 本提案手法

共有データの大きさは、同期変数等で使用されるキャッシュラインサイズ以下のものと、大規模な行列計算での被演算行列のようにページサイズを上回るものの主に 2 種類に分類できる。後者はユーザーレベル通信機構によって性能改善できる。本節では、前者のような少量のデータの共有時における通信のオーバーヘッド削減を目的とした、I/O バス上のメモリ一貫性管理機構の提案を行なう。

### 4.1 I/O バス上のメモリー貫性管理機構

メモリ一貫性管理コントーラは I/O バス上にあり、過去に参照された共有メモリのアドレス、メモリの内容、共有状態を表わすディレクトリで構成されるデータをキャッシュしている (図 3)。共有データを読み出す場合、以下の手順でメモリ一貫性管理コントローラにデータの要求が行われる。

1. Exclusive Owned の場合、WS のキャッシュ上にあるか調べる。
2. Exclusive Owned の場合、ページテーブルを参照しメモリ上にあるか調べる。

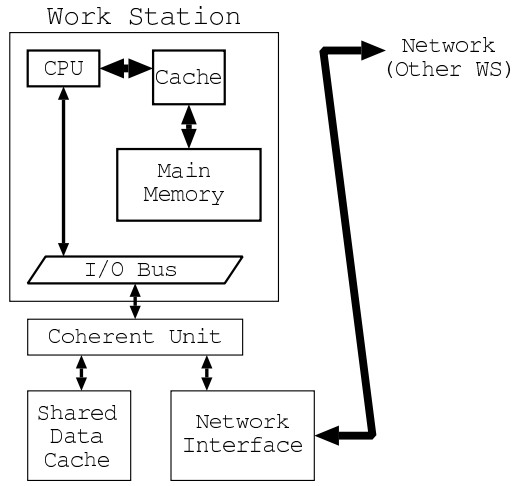


図 2: 各ノードの構成

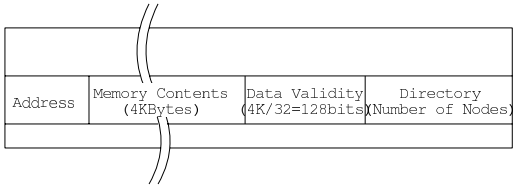


図 3: 一貫性管理コントローラのキャッシュ

3. 上記で発見されない場合とその他の場合、メモリ一貫性管理コントローラに必要なデータのアドレスを通知。
4. メモリ一貫性管理コントローラはキャッシュを参照し、読み込みを行おうとしているアドレスを含むキャッシュラインサイズのデータが有効であるか判断し、有効ならば OS 側にキャッシュのデータと、今読み出そうとしているデータ部分以外は保証しないというフラグを返す。
5. 無効なデータに対する読み出しはネットワークインターフェースへのデータ要求を行う。
6. 割り込み等によりメモリ一貫性管理コントローラからデータが渡される。

メモリ一貫性管理コントローラからネットワークインターフェースへのデータ要求は、キャッシュ上の有効データを除いた部分だけに縮約される。

同様にして、書き込みも行なわれる。書き込みで他のノードに通知される情報は、無効化を基本とする場合、書き込みがおこなわれたアドレスだけである。

### 4.2 プロトコル

ここではキャッシュの一貫性管理は無効化を基本としているが、書き換え時に更新を行うことも可能である。無効化を基本とした場合、同一キャッシュライン上に二つの共有変数があって、それぞれを使用しているノードの集合が重ならない場合に通信量の削減が可能になる。また、メッセージパッシングにおけるメッセージコンバインと同じような仕組でデータ有効フラグが働くため、通信の削減が期待できる。

## 5 関連研究

メッセージパッシングにおける通信始動のオーバーヘッドの削減が目的の Shrimp[1] ではユーザーレベル通信機構を使用した Release Consistency に基づくメモリ共有機構が実現されている。WSC ベースの DSM 実現では少量のデータを共有する場合の性能がネックになるため、多くの研究が成されている。TreadMark[2] は少量のデータ共有をコンパイル時に判別し、前もって実行コード内に同期操作命令を入れることで効率の改善を行っている。Winsconsin 大学の Wind Tunnel Project[3] で開発された Tempest[4] ではソフトウェアにより DSM を実現しているが、少量のデータ共有時の通信を削減するため、メモリの ECC 操作によるキャッシュライン単位の一貫性管理と、実行バイナリの事前チェックによる同期操作命令の挿入を行っている。Rochester 大学の Cashmere[5] は通信インターフェースをメモリ空間に配置し、少量のデータ共有時の通信オーバーヘッドを削減している。

## 6 まとめ

本稿では、高速 I/O 上でのメモリ一貫性管理機構による WSC での DSM 手法の提案を行った。本手法では、メモリの一貫性管理の多くをハードウェアによって行うことで、ソフトウェアによる一貫性管理より高速に行うことが可能になる。今後、ユーザーレベルでの通信機構と組み合わせた場合のネットワークインターフェースの詳細な検討と、シミュレーションによる定量的な評価を行う。

### 参考文献

[1] Matthias A. Blumrich, Kai Li, Richard Alpert, Cezary Dubnicki, and Edward W. Felten. Virtual memory mapped network interface for shrimp multicomputer. In *The International Symposium on Computer Architecture*. IEEE, 1994.

[2] Alan L. Cox, Sandhya Dwarkadas, Pete Keleher, and Willy Zwaenepoel. An Integrated Approach to Distributed Shared Memory. In *First International Workshop on Parallel Processing*, December 1994.

[3] Mark D. Hill, James R. Larus, and David A. Wood. Parallel Computer Research in the Wisconsin Wind Tunnel Project. In *NSF Conference on Experimental Research in Computer Systems*. NSF, June 1996.

[4] Mark D. Hill, James R. Larus, and David A. Wood. Tempest: A Substrate for Portable Parallel Programs. In *Compcon*, March 1995.

[5] Leonidas I. Kontothanass and Michael L. Scott. High performance software coherence for current and future architecture, November 1995.