3F−10 仮想メモリ空間にマップされたDSMインターフェースを用いた仮想共有メモリの支援

渡辺 正泰 , 田中英彦
東京大学大学院 工学系研究科

1 はじめに

今日に至るまで多くの共有メモリ並列計算機が研究/開発され、ハードウェア実装による共有メモリ計算機に関しては有望な商用機が開発される迄になった。それに対し、主にワークステーションクラスタを対象としたソフトウェア仮想共有メモリ機構 (SVM) はまだ開発途上にある。本稿では、ハードウェア実装による共有メモリ機構から、その機能の一部を比較的開発の容易なI/O装置としてワークステーションに付加し、ソフトウェア仮想共有メモリを支援することを考え、その構想について報告する。

2 ソフトウェアSVM機構

ソフトウェアSVM機構とは、MMUを利用し共有メモリ空間へのアクセスを監視し、ページ単位で分散共有メモリ間の一貫性を保持することで、仮想的に分散共有メモリを実現する機構である。ソフトウェアSVM機構には、ハードウェア実装による分散共有メモリに対し、

- 一般のワークステーションで安価に実現できる。
- 問題に合わせ、プロトコルを自由に設定することが可能。

などの利点があるが、同時に

- ソフトウェアで管理することによるオーバーヘッド。
- 低速なネットワークを利用することによる大きなリモートアクセスコスト。
- 一貫性保持単位が大きいことによる、false sharingの発生。

などの問題も抱えている。

3 ソフトウェアSVMのハードウェア支援

ソフトウェアSVM機構の問題点に対し、比較的小規模なハードウェアの追加を行なうことでその問題を解決する試みが行なわれている。主に次の2つの手法が試みられている。

1: 高速な通信インターフェースを利用する。

2: 一貫性保持単位を小さくする機構を設ける。

1は通信に必要なコストを削減する試みで、自然な考えと言える。例えば商用インターフェースではMyrinetなどのGigabitLANが注目されている。研究では通信インターフェースを仮想メモリ空間にマップすることで、ユーザーレベルで通信インターフェースをアクセスすることを可能にし、ソフトウェアオーバーヘッドを削減する手法が注目されている。SHRIMP[1], DECのmemory channelを利用したCASHEMERe[3]等が挙げられる。

2は、通常ページ単位 (4〜16Kbytes) で行なわれる一貫性保持をキャッシュライン単位 (32〜128bytes) 程度まで小さくする手法である。Tempest[2] では次の2つの手法が試みられている

- メインメモリのECCビットを転用し、キャッシュライン単位で割り込みを起こす。
- メモリバス上にメモリトランザクションを監視するハードウェアを追加し、キャッシュライン単位で割り込みを起こす。

この手法はfalse sharingを減らす上で非常に有効であり、特に後者は更にネットワークインターフェースを追加することで大きな効果を挙げている。しかし、メモリバスを用いたインターフェースは汎用性を欠き、またその開発は比較的難しく、今後は今まで以上に開発が困難な状況になることが予想される。

4 メモリマップDSMインターフェース

一方、ハードウェア実装による分散共有メモリの分野ではディレクトリキャッシュを用いたCC-NUMA分散共有メモリ機構が研究/開発され、SGIのOrigin2000などの商用機も登場している。Jump-1でも同様の研究が行なわれ、その一環であるJump-1/3[4]ではワークステーションクラスタに分散共有メモリを付加するインターフェースが開発されている。

Jump-1/3では、ワークステーションのI/Oバス上に通信とプロトコル処理を行なうプロセッサを持つボード (DPSボード) を置き、ボード上に配置された分散共有メモリの一貫性管理はボード自身が行なう。各ワークステーションはDPSボード上の共有メモリを自身の仮想メモリ空間にマップして利用する。

Software SVM support with VM-mapped DSM interface

Masahiro WATANABE and Hidehiko TANAKA
Graduate school of Engineering, The University of Tokyo

また、共有単位はページ単位であるが、一貫性の保持はキャッシュライン単位で行なわれる。したがってfalse sharingの発生は一般のハードウェアによる共有メモリと同等に押えられる。

5 DSMインターフェースの、ソフトウェアSVMへの利用

Jump-1/3のDPSボードのようなDSMインターフェースは、本来SVMを支援する目的で作成されたものではないが、ここではソフトウェアSVMを支援するハードウェアとしてとらえる。このようなDSMインターフェースは次の問題を解決できると考えられる。

- 一貫性保持などのプロトコル処理と通信処理はDPSボードによって行なわれ、ソフトウェアオーバーヘッドが軽減される。
- 一貫性保持単位はキャッシュライン単位であり、false sharingの問題が低減される。
- 細粒度アクセスを、SVMと比べて低レイテンシで実行できる。

しかし、次の様な問題も存在する。

- I/Oバス上に存在するため、プロセッサキャッシュが有効に利用できない。
- 同様に、ワークステーション側のメモリを共有メモリとして利用できない。
- I/Oバスはメモリバスに比べ遅く、大規模なデータに対し十分なバンド幅を提供できない。

そこで、DSMインターフェースをSVM支援ハードウエアとしてとらえ、上記の問題の解決を図る。原理的なアイディアは次のようなものである。

一貫性管理は原則としてソフトウェアによりページ単位で行なわれるが、複数のノードから同時にアクセスを受けるページ、すなわちfalse sharingを起しがちなページに関してはDSMインターフェース側に移動し、DSMインターフェースで管理する。同期変数領域など、細粒度アクセスが主で低レイテンシ性が重要なページに関しても同様に扱う。

逆にRead onlyであるページ、配列等の一貫性管理単位が大きくても良いデータが主なページ、排他的アクセス権を持つページについてはローカルメモリに移動し、ソフトウェアによる管理を行なう。

これにより、ハードウェア実装のDSMの長所であるfalse sharingの低減、細粒度アクセス時のオーバーヘッドの低減と、プロセッサキャッシュ、及びローカルメモリの有効利用が実現できる可能性がある。

6 必要となる技術

上記の機構を実現するためには、DSMインターフェース側に

- DSMインターフェース間での一貫性保持機構
- ローカルメモリ間のページ単位転送機構
- DSMインターフェース間でのページ単位転送機構
- ソフトウェアSVMのための低レイテンシ通信機構

ソフトウェアSVMシステム側に、

- 通常のページ単位一貫性保持機構
- ローカルメモリ-DSMインターフェース間一貫性保持機構
- DSMインターフェース、またはローカルメモリに移動するページを選択する機構

以上の技術が必要だと考えられる。DSMインターフェースについてはCC-NUMA型分散共有メモリ機構の成果とメモリマップ通信インターフェース技術が利用できると思われる。

ソフトウェアSVMシステム側では、DSMインターフェース/ローカルメモリに移動するページを選択する機構が最も重要な課題になるものと考えられる。

7 まとめ

本稿では、ソフトウェアSVM機構のメモリマップDSMインターフェースを用いた支援について、利用する技術とその構想に関しての報告を行なった。今後は前提とするDSMインターフェース、SVMシステムの詳細な検討と、シミュレーションによる数値評価を行なう予定である。

参考文献

[1] M. Blumrich, K. Li, R. Alpert, C. Dubnicki, E. Felten, and J. Sandberg. Virtual memory mapped network interface for the SHRIMP Multicomputer. In *Proceedings of the 21st Annual Internatinal Symposium on Computer Architecture*, pp. 142–153, 1994.

[2] Mark D. Hills, James R. Larus, and David A. Wood. Tempest: A substrate for portable parallel programs. In *In COMPCON '95*, pp. 327–332, 1995.

[3] L. Lontothanassis, G. Hunt, R. Stets, N. Hardavellas, M. Cierniak, S. Parthasarathy, W. Meirs, S. Dwarkadas, and M. Scott. VM-Based shared memory on low-latency, remote-memory-access networks. In *Proceedings of the 21st Annual Internatinal Symposium on Computer Architecture*, 1997.

[4] 安生健一朗, 中條拓伯, 小野航, 工藤知宏, 山本淳二, 西宏章, 木透徹, 天野英晴. 分散共有メモリを持つWSクラスタ:JUMP-1/3. 並列処理シンポジウムJSPP'97論文集, pp. 321–328, 1997.