3B－1

ディレクトリキャッシュにおける疑似フルマップシステムの定量的評価

佐藤充[*1], 三吉貴史[*2], 松本尚[*2], 平木敬[*2], 田中英彦[*1]
[*1]東京大学工学部, [*2]東京大学理学部

1 はじめに

分散共有メモリマシンでの、メモリアクセス・レイテンシ隠蔽のための技法としてキャッシュ・システムがある。並列計算機におけるキャッシュ・システムでは、メモリトランザクションにおけるコンシステンシ保持が重要な問題となる。

本発表では、そのコンシステンシ保持方式のひとつであるディレクトリ方式 [1] をとりあげ、特にその中でも疑似フルマップ [2] と呼ばれる方式を用いたシステムについてのシミュレーションの結果を報告する。

2 疑似フルマップ

擬似フルマップ方式は階層化放送機構を前提としたディレクトリ方式である。さらに、協調動作するスレッドはなるべく近傍のプロセッサ群にスケジューリングするというプロセッサ資源管理方針も仮定している。疑似フルマップ方式では、ディレクトリを階層化マップという形で保持する。階層化マップは、階層化放送機構の構造と対応し、通信距離が近いクラスタは共有ページの存在を細かく管理し、通信距離が離れるにつれて粗く (放送機能の階層単位で) 管理する。つまり、遠くのクラスタへ通信する場合は、階層化放送機構でそのクラスタと通信可能になる階層まで遡り、そこから目的地を含む部分木に向かってメッセージをブロードキャストする (図1)。

階層化放送機構を介した通信の Ack は元のメッセージが送信された経路を逆にだどって返信される。部分木へのブロードキャストであった場合は各階層で Ack のコンバイニングがなされる。末端のすぐ上のクラスタでは、すべての子どもから Ack または Dack が返った時点で、Ack をさらに上位の親に返す。これを元の部分木のルートノードまで繰り返していく。部分木のルートノードまでこの操作が終了すると、自分より上層の部分木への送信がなされた場合は、上層から Ack が返送されるのを待ち、Ack を元のメッセージ送信クラスタ方向の下層のノードに返送する。上層への送信がなされていない場合は、上層からの Ack を待つことなしに、Ack を下層のノードへ返送する。

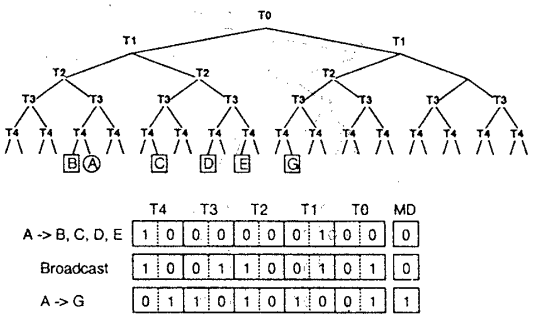

図 1: 階層化マップ

3 シミュレーション

3.1 シミュレータ

疑似フルマップを実現するためには、対象とするシステムのネットワークは階層化放送機構を持っている必要がある。今回のシミュレーションではネットワークとして図2に示すような4進木ネットワークを用いた。

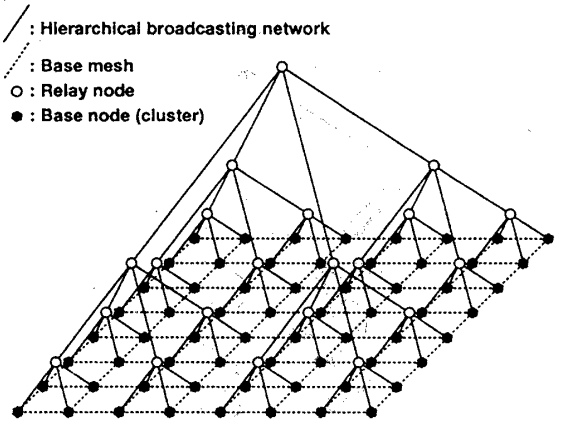

図 2: 4進木ネットワーク

シミュレータの各ノードは Router, Network Processing Unit(NPU), Access Pattern Generator(APG) から成る (図3)。APG ではアプリケーションを実行し、然るべきタイミングで NPU にノード間にまたがるメモリ

Evaluation of Pseudo Full Map System in Directory Cache
Mitsuru SATO[*1], Takashi MIYOSHI[*2],
Takashi MATSUMOTO[*2],Kei HIRAKI[*2],
Hidehiko TANAKA[*1]
[*1]Faculty of Engineering, University of Tokyo
[*2]Faculty of Science, University of Tokyo

トランザクション要求を出す。

NPU は

1. パケットの生成
2. ホームにおけるメモリトランザクションの単一性の保証
3. 中継ノードにおける Ack の収集
4. メモリ属性の更新

等を行なうユニットである。メモリトランザクションが発行されると、ホームノードの NPU において調停が行なわれ (あらかじめ単一性を保証する必要がないことが分かっている場合はこの限りではない)、実行される。階層化マップを用いた invalidate/update が実行されると、中継ノードの NPU は自分の ACK Table にエントリを生成し、定められた数の下位ノードから Ack が返ってくるのを待つ。

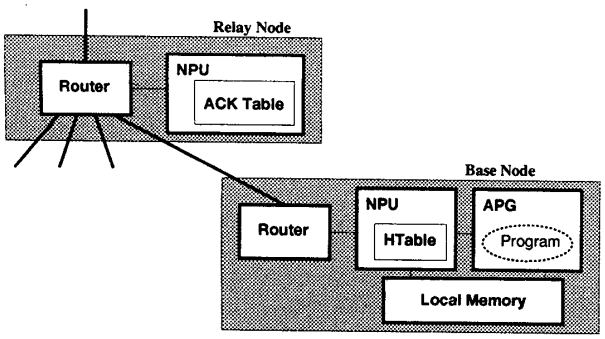

図 3: シミュレータの構造

3.2 測定

測定は、アプリケーションとして行列・ベクトル積を用いて行なった。各ノードはそれぞれ行を担当し、自分の担当行のデータを保有する。ベクトルは全員が共有して保持する。したがって、1 ループが終了する度に invalidate または update が発行される。

あらかじめプログラムが定常状態になった時のメモリの状態を実現し、本システム上でで 1 ループにかかる時間、総パケット数を測定した (表 1)。

表 1: 測定結果

台数	invalidate		update	
	clock	packet	clock	packet
4	920	1960	640	1856
8	1948	4368	1012	3712

表 1において invalidate と update で大きく差が開いているのは、パケット上でのデータの割合による影響が大きいものと考えられる。現在のシステムでは Ack パケット: 7 word、データ通信: $7+n$ word であるので、この行列・ベクトル積のような invalid になったブロックを再び共有する可能性の高いアプリケーションの場合は invalidate プロトコルはは非常に不利になる。しかし、この場合は invalidate プロトコルと update プロトコルの差は read パケット (+reply of read) + invalidate パケットと update パケットとの差に近似されるので、8 台で 600 程度の差になっている (invalidate/update は疑似フルマップによってブロードキャストされるので、単純な和にはならない)。

4 Acknowledge の収集

前述のように、疑似フルマップでは中継ノードにおいて複数の Ack をまとめて、1 つの Ack として上位ノードに送る必要がある。このため、invalidate/update のアドレスと必要な Ack の数を保持するテーブルが必要となる。本システムでは、このテーブル (ACK Table) を NIP が保有している。そのため、invalidate/update メッセージや ACK は必ず中継ノードの NIP を通過する必要があり、そこがシステムのボトルネックとなる。今回の測定では、256 ノード時、中継ノードにおいて NIP に送られるパケットの平均待ち時間は 8.3 クロック (invalidate)、7.4 クロック (update) であった。現在、この対策として Router で ACK Table をキャッシングする方法を検討中である。

5 まとめ

分散共有メモリシステムにおける疑似フルマップの性能を、シミュレータを用いて測定した結果を報告した。特に、疑似フルマップを用いた場合の invalidate プロトコルと update プロトコルの差について評価した。今後はこのシミュレータを用い、Router で ACK Table をキャッシングする手法の検討、invalidate/update プロトコルの切替え手法 [3] の評価、他のネットワーク (RDT[4]) など) を用いての測定などを行なっていく予定である。

参考文献

[1] et al D.Chaiken. Directory-based cache coherence in large-scale multiprocessor. *IEEE Computer*, Vol. 23, No. 6, pp. 49–58, 1990.

[2] 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.

[3] 松本尚. 細粒度並列実行支援機構. 情報処理学会計算機アーキテクチャ研究会報告, Vol. 12, No. 77, pp. 91–98, July 1989.

[4] 楊愚魯, 天野英晴. 超並列向きのプロセッサ結合網の提案. 情報処理学会計算機アーキテクチャ研究会報告, No. 96-20, October 1992.