

# ATMのセル廃棄を許容するソフトウェア DSM 向け一貫性プロトコル

中村 嘉志<sup>†</sup>, 多田 好克<sup>†</sup>

近年, 計算機を高速なネットワークで接続したクラスタなどの並列処理システムがその価格性能比の良さから注目を集めている. 通常, クラスタシステムの相互接続ネットワークとしては Ethernet などの LAN が用いられることが多いが, これに ATM をあてはめるには困難が生じる. ATM はコネクション指向であることから, 並列処理におけるコネクションの維持の増大 ( $O(n^2)$  または  $O(n \log n)$  など) が台数  $n$  に対するスケーラビリティを阻害するという問題があるためである. また, ATM はセル廃棄と呼ばれる恣意的なメッセージ損失が起こることが前提のネットワークであるため, プロトコルはこれに対してある程度寛大でなければならない. 本論文では, ATM コネクションの維持数を  $O(n)$  とし, 流量制御が容易で, かつ, セル廃棄を考慮したソフトウェア DSM 指向のキャッシュ一貫性プロトコルを提案する. SPLASH2 ベンチマークを用いた実験によって, 本プロトコルの正当性と, その性能のスケーラビリティを確認した.

## A Consistency Protocol for Software Distributed Shared Memory Tolerating ATM Cell Losses

YOSHIYUKI NAKAMURA<sup>†</sup> and YOSHIKATSU TADA<sup>†</sup>

In recent year, cluster based parallel processing systems which are connected via high-speed network become popular because of good price-performance advantage. Although cluster systems generally use existing LAN such as Ethernet for interconnection network, some difficulties occur when applying ATM to these interconnection networks. The reason is mainly a problem of connection maintenance cost. This cost prevents scalability of system as a number of nodes increases. As a result, number of connections grow according to  $O(n^2)$  or  $O(n \log n)$  for number of nodes  $n$ , because ATM is connection oriented network. On the other hand, system must be tolerance against message losses, since the cell loss, that is arbitrary message losses, are occurred under ATM networks. In this paper, we propose a consistency protocol which keeps ATM connections to  $O(n)$ , makes rate control easy, and tolerates ATM cell losses. We verify validity and scalability of the proposed protocol by an experiment that used SPLASH2 benchmark.

### 1. はじめに

近年, 計算機を高速なネットワークで接続したクラスタなどの並列処理システムがその価格性能比の良さから注目を集めている. ソフトウェア分散共有メモリ (以下, ソフトウェア DSM) は, 既存のプログラミングモデルをこのようなクラスタ上での並列計算に容易に拡張可能である点で有用である. ソフトウェア DSM システムは IVY<sup>5)</sup> に始まり, これまで多数提案され

ている<sup>1),3),10)</sup>.

これまでのソフトウェア DSM システムではネットワークを IP でカプセル化したものがほとんどであり, ATM<sup>6)</sup> を直接扱ったものは提案されていない. ATM には, QoS 保証可能, LAN から WAN へのシームレスな接続性といった, 他のネットワークアーキテクチャにはない優れた特長がある. しかし, IP でカプセル化してしまうと IP の振舞いに左右されてしまうため, ATM のこれらの特長を活かすことができない. 我々は, ATM の特長のうち帯域予約可能な点に注目し, 他の通信に影響を受けない広域のソフトウェア DSM システムの構築を目指している.

ところで, ATM はコネクション指向であるため並列処理におけるコネクションの増大 ( $O(n^2)$ ) が台数  $n$  に対するスケーラビリティを阻害するという問題があ

<sup>†</sup> 電気通信大学大学院情報システム学研究科  
Graduate School of Information Systems, The University of Electro-Communications  
現在, 産業技術総合研究所  
Presently with National Institute of Advanced Industrial Science and Technology (AIST)

る．これには2つの要因をあげることができる．1つはコネクション数の問題、もう1つはコネクション維持コストの問題である．ソフトウェア DSM のような分散環境下では、あるノードは任意のノードに対して通信を行えなければならない．このため、ソフトウェア DSM にコネクション指向をナイレブにあてはめると完全結合になり、全体のコネクション数が  $O(n^2)$  で増加して現実的ではない．また、ATM はノードである端点のほかに交換機であるネットワーク自体にも帯域などの QoS 情報を持つ必要があるため、ATM の利用に際してはコネクション維持コストが無視できない．結果として、ATM では一度に多くのコネクションを保持することができず、このことがスケーラビリティ阻害の要因となる．このほかに、ATM ではセル廃棄と呼ばれる交換機による恣意的なメッセージ損失が起こることが前提になっているため、システムはこれに対してある程度寛大でなければならない．

このような問題から、ATM 上でソフトウェア DSM を構築する場合にはその特性に根差したシステム設計が必要とされる．特に、コネクション数の問題から、従来方式よりも簡素な処理機構で実現できるプロトコルが必要である．

本論文では、環状ネットワークポロジを用いて、1) 通信コネクション数の増大を  $O(n)$  で抑え、2) 流量制御が容易で、かつ、3) セル廃棄を考慮したソフトウェア DSM 指向のキャッシュ一貫性プロトコルを提案する．この提案プロトコルを載せたシステムをターゲットマシン上に構築し、SPLASH2<sup>7)</sup> ベンチマークなどの並列プログラムを用いた実験によって本プロトコルの正当性と、その性能のスケーラビリティについて論じる．

以下、本論文では、2章でシステムの概観と前提条件について説明する．3章では、一貫性プロトコルの動作を詳しく述べ、4章で実際にいくつかの並列プログラムを動作させた結果からシステムの定性的、定量的評価を行う．5章で関連研究との比較を行い、最後に6章で今後の方針とまとめを述べる．

## 2. システム構成

我々は ATM ネットワーク上で動作するソフトウェア DSM システムを構築している．本章では、このシステムの一貫性プロトコルの動作前提となるハードウェア構成と ATM で構成する環状ネットワークポロジについて述べる．

### 2.1 想定するハードウェア構成

本論文で提案するキャッシュ一貫性プロトコルは、

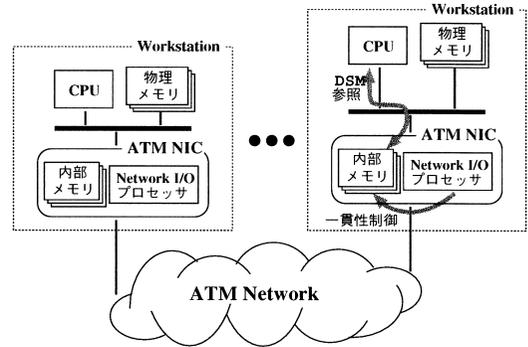


図1 想定するハードウェア構成  
Fig. 1 Structure of target hardware image.

その一貫性制御を NIC ( Network Interface Card ) 上で行うことを想定している．図1に、本システムが想定するハードウェア構成の概念を示す．キャッシュは、Home Proxy Cache<sup>8)</sup> のように I/O バスの外側の NIC の内部メモリに設け、一貫性プロトコルはこの内部メモリに対して働く．また、ソフトウェア DSM を構成する各計算ノードは ATM ネットワークに直接接続され、途中に IP ルータなどのプロトコル変換器は挟まない．

一般に、ATM のようなセルネットワークではアダプテーション層と呼ばれるプロトコル層を介してデータを扱う．セルはペイロードが極小かつ固定長であることから、アダプテーション層は上位層のユーザデータをセルに分割する処理と、分割して送られてきた複数のセルから元のデータに復元する組立処理を行う．分解/組立て処理オーバーヘッドを軽減するため ATM NIC にはこれらの処理専用の I/O プロセッサを搭載しているものがある．我々は、このプロセッサを流用し、I/O バスの外側で一貫性制御を行うことによってホスト計算機の負荷軽減を図り、計算と通信のオーバーラップを実現することを考えている．

### 2.2 環状接続ポロジと巡回型ブロードキャスト

ATM を利用するシステムでは、コネクション数を控えることが重要な課題である．通信コネクション数の増大を  $O(n)$  で抑えるため、我々は環状接続ポロジを採用し、そのうえで巡回型ブロードキャストを用いることにした．

ATM では、仮想回線 ( 以下、VC ) を確立する際に、網に対して利用する VC の特性を申告させることによってサービス品質 QoS の保証を図っている．端点であるノードと経由する交換機とがそれぞれ状態を維持する必要があるため、このことが、かえってノード数の増加に従ってシステム全体で管理する状態数が

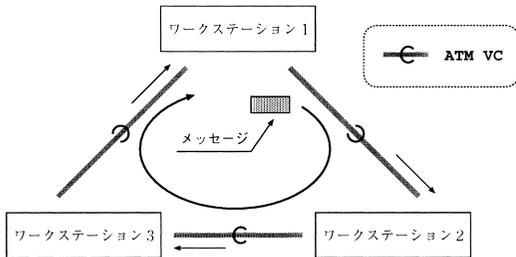


図 2 メッセージの流れの様子  
Fig. 2 Message flow.

急激に増加するという問題を生じる。すなわち、VC は有限な資源であると考える必要がある。したがって、通信コネクション数を抑えることは ATM では欠かせない要求である。

本一貫性プロトコルでは、VC は PVC (Permanent VC) 方式によってあらかじめ環状に確立されているものとする。図 2 に、各ノード間のメッセージの流れの様子を示した。

巡回型ブロードキャストの利点は、送信したメッセージの受信を送達確認 ACK に換えることができる点にある。一般に、キャッシュブロックの無効化、移動など、確実に情報の伝送を行うには送達確認 ACK を用いる必要がある。ポイント-ポイント通信でこれを行う場合、通信相手の数  $n$  に対して ACK を含めたメッセージ数は  $2n$  必要である。巡回型ブロードキャストでは、通信相手の数  $n$  に最初の送信を加えた  $n+1$  のメッセージ数で済む。

環状接続のネットワークで分散共有メモリを実現するシステムとしては、これまでにハードウェア実装の Memnet<sup>4)</sup> がある。Memnet はトークンパッシング型のネットワークを使用しており、システム上の各ノードはトークンの到達を待ってメッセージを送信する。したがって、トークンの紛失がメッセージの喪失として検知されるため、ネットワークレベルでのメッセージの再送が可能である。また、メモリに対するアクセス競合はトークンの獲得によって事前にネットワークのレベルで調停されることになる。

これに対し、我々の一貫性プロトコルは以下の点で Memnet とは異なっている。

- プロトコル内にセル損失を許す仕組みを持つ。
- メモリアクセス要求を調停する仕組みを持つ。

ATM は非同期転送がその特長である。メッセージ (セル) の紛失はネットワークレベルで検知できないため上位層で処理しなければならない。我々は、この要求に対し、信頼性のあるプロトコルスタックを ATM の上に設けるのではなく、むしろ ATM で前提となっ

ているセル廃棄を積極的に許容する方針で一貫性プロトコルを考案した。また、複数のノードから同時に出力されたメモリアクセス要求を調停する仕組みとして所有者という概念を設け、本プロトコルではこの所有者が調停を行う。

### 2.3 コンパイラサポート

一般に、ソフトウェア DSM は一貫性の崩れるメモリへの書込みを MMU のページ保護機能によってとらえ、それを基にプロトコルが動作する。対して、本プロトコルは Midway システム<sup>2)</sup> のようにコンパイラが生成するコードで書込みをとらえることが前提になっている。

### 3. セル損失を許容する一貫性プロトコル

本章で述べる一貫性プロトコルは、環状ネットワークポロジ上で巡回型ブロードキャストを利用した書込み無効化型のプロトコルである。メッセージは各々のノードで評価され、図 2 のようにネットワーク上を環状に伝搬する。また、メッセージは処理要求と同時にデータの運搬も行う。したがって、処理要求を受けたノードは別途キャッシュ内容の転送を行うのではなく、キャッシュ内容をメッセージに挿入して次のノードに送るだけでよい。

本プロトコルは、VC ごとに、1) メッセージの紛失誤りがまれに発生し、2) メッセージの重複誤りは発生せず、および、3) メッセージ順序が保存される、という ATM のネットワーク特性を前提に動作する。各 VC はあらかじめ PVC を用いて環状に接続されているものとする。環状接続により必要 VC 数は  $O(n)$  であり、資源要求は少ない。

#### 3.1 キャッシュ状態と処理要求メッセージの種類

##### 3.1.1 キャッシュ状態

本一貫性プロトコルにおいて、キャッシュブロックの状態は次の 3 種類で区分される。

- (1) valid/invalid (有効/無効)
- (2) shared/exclusive (キャッシュ複製あり/なし)
- (3) owner/copy (移動時の責任あり/なし)

(1) の区分はそのキャッシュが有効か無効かを示す。(2) の区分は他ノードにコピーがあるかどうかを示し、書込み時に他ノードのキャッシュの無効化処理が必要かどうかを表す。(3) の区分はアクセス競合時に調停の責任があるかどうかを示している。なお、owner 状態はキャッシュブロックごとにシステム内で唯一に保たれる。

##### 3.1.2 メッセージの種類

各ノードからシステムに対して処理要求を行うメッ

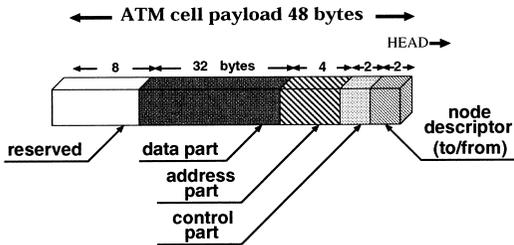


図 3 1セルで構成される処理要求メッセージの構造  
Fig. 3 Structure of request message consists of 1 cell.

セージの種類は次の 3 種類がある .

- (1) データ要求メッセージ ( R メッセージ )
- (2) 無効化要求メッセージ ( I メッセージ )
- (3) 無効化およびデータ要求メッセージ ( I&R メッセージ )

読み込み時に他ノードに対して目的アドレスのデータ要求および取得を行うには (1) のメッセージを使用する . (2) は書き込み時に他ノードに対して目的アドレスのキャッシュの無効化要求を行う . (3) は上記 (1) , (2) の 2 つの処理要求を同時に行うものである .

これらの処理要求メッセージは各々 ATM セル 1 つで構成される . メッセージは処理要求と同時にデータの運搬も行うので , 処理要求を受けたノードは別途データ伝送を行う必要がなく , メッセージ 1 つで処理が完結する . 図 3 はメッセージの構造を示したものである . メッセージを構成する ATM セルのペイロードは , ノード識別子 ( to/from ) , コントロール部 , アドレス部 , データ部の 4 つの部分からなる . ノード識別子 from には送信元ノードの論理番号が格納される . ノード識別子 to は owner 移動時に用いられる . コントロール部にはこのメッセージがどの種類のメッセージであるかを記述する . 加えて , データ部にデータが格納されているかどうかこの部分で判断できるようになっている . アドレス部には処理要求目的の共有ブロックのアドレスが示される . データ部は 32 バイトの共有ブロックを運搬する役割を果たす .

3.2 プロトコルの基本動作

3.2.1 キャッシュへの読み書き時の基本動作

キャッシュへの読み書き処理 , そしてキャッシュの状態 4 種類 ( exclusive , shared-owner , shared-copy , invalid ) により , 読み書き処理のヒット/ミス进行分类すると , リードヒット , リードミス , ダーティライト ( exclusive 状態への書き込み ) , クリーンライト ( shared-owner 状態への書き込み ) , ライトミス ( shared-copy 状態 , invalid 状態への書き込み ) の 5 つに分類される . ミス時およびクリーンライト時には ,

表 1 アクセスの分類と処理要求メッセージの種類  
Table 1 Classification of access and type of message.

キャッシュの状態	exclusive	shared	invalid
	owner		copy
読み込み	そのまま読み込む		データ要求 ( R メッセージ )
書き込み	そのまま書き込む	他ノードのキャッシュ無効化要求 ( I メッセージ )	無効化およびデータ要求 ( I&R メッセージ )

キャッシュの内容が無矛盾であることを保証するために処理要求メッセージを送出して他のノードと協調動作しなければならない . 表 1 に , 本一貫性プロトコルでのアクセスの分類とそのときのネットワークへの要求内容を示す .

処理要求は , 通常 , メッセージがネットワーク上を環状に伝搬し , 1 周して戻ってきた時点で完了する . しかし , それだけで完了しない場合もある . 処理を完了させることができるかどうかは , 要求メッセージに対して応答が行われているかどうかで判断する . たとえば , リードミス時の R メッセージでは , 回帰後にメッセージにデータが格納されているかどうかで判断することができる . 処理が完了できない場合は , 処理要求の再試行となり , このノードはシステムに対し要求メッセージの再送を行う .

3.2.2 メッセージ通過時の基本動作

本プロトコルは環状接続上で巡回型ブロードキャストを利用しているので , 処理要求ノードから送出されたメッセージは共有メモリを構成する他のノードをそれぞれ環状に伝搬し , 再び処理要求ノードへと戻る . その際 , メッセージが通過する各ノードでは , 1) 処理要求メッセージを受信し , 2) メッセージの内容を評価し , 3) キャッシュの状態の応じて必要であれば状態遷移とメッセージの操作を行い , 最後に , 4) 隣のノードへ送信する , という一連の 4 つの処理を行う . 図 4 に , メッセージ応じたキャッシュの状態遷移を示す . なお , owner 状態はそのキャッシュブロックの所有者としてつねに最新のデータを保持しており , システム内で唯一に保たれる .

3.3 セル損失の許容

3.3.1 セル損失の検出と再送

セル損失の検出はタイムアウト処理によって行う . 要求メッセージはネットワーク送出時に送信バッファに保存され , 再送に備える . 要求メッセージが環状ネットワークを 1 周して戻ってきた時点で処理は完了し ,

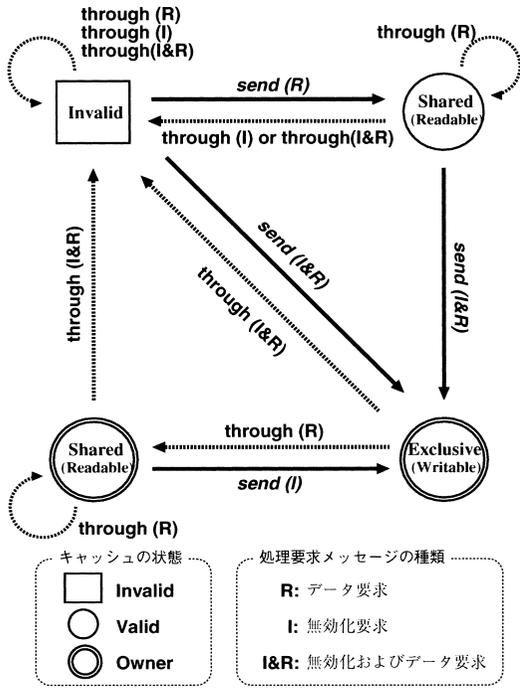


図 4 キャッシュの状態遷移  
Fig. 4 Cache state transitions.

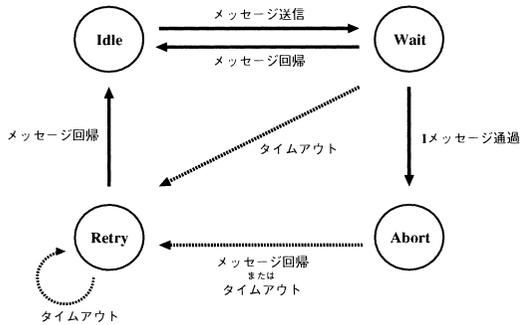


図 5 タイムアウト状態遷移  
Fig. 5 Timeout state transitions.

メッセージはバッファから破棄される。もし、一定時間内にメッセージが戻ってこなければ、そのメッセージは喪失したものと再送処理を行う。図 5 に、タイムアウトに関するノードの状態遷移を示す。

本一貫性プロトコルにおいて、セル損失が起きてても再送で済む理由は、owner 状態が最新のデータを持っており、次項で説明するようにこれが消滅しないからである。

3.3.2 アクセス競合時の動作と送達確認

本一貫性プロトコルにおいて、書き込み時においてアクセス競合が起きた場合、最初に owner 状態を持つノードに I&R メッセージを届けたノードがアクセ

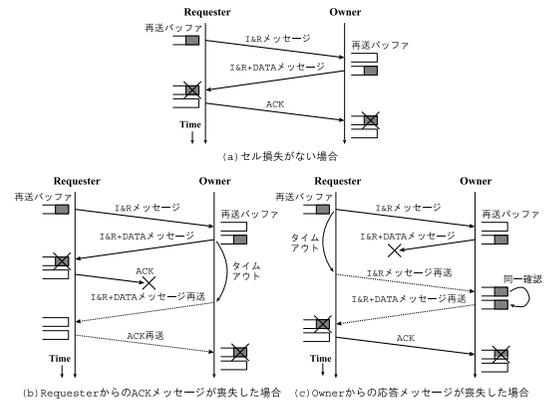


図 6 セル損失と再送  
Fig. 6 Cell loss and retransmit.

ス権を得ることができる。owner 状態はキャッシュブロックごとにシステム内で唯一に保たれており、所有者として最新のデータを保持している。アクセス権を得た I&R メッセージには最新データが挿入され、メッセージの帰後にそのノードが新たな owner となる。アクセス権を得られなかった I&R メッセージにはデータは挿入されない。このようにして owner 状態は必要に応じて動的にノード間を移動する。

この owner 状態のキャッシュブロックは、アクセス競合調停と最新のデータを保持しているという重要な役割を担っていることから、owner 状態の移動に際し、移動先から移動元へ送達確認 (ACK) メッセージを送ってメッセージ送達の確実性確保する。送達確認は、移動元から移動先にデータの挿入された I&R メッセージが届き、さらにこのメッセージが環状ネットワークを 1 周して帰した場合は送達確認とする。図 6 は、owner 状態移動に係るセル損失と再送の関係である。

移動元は I&R メッセージを受けると、自ノードが I メッセージを発行中でないことを確認してから owner 移動準備を行う。I メッセージ発行中はクリーンライト時で、自ノードが書き込みを行うため他ノードの無効化要求中であるため、アクセス権はすでにこのノードが獲得している。owner 移動準備が整ったら移動元ノードは、キャッシュの内容を I&R メッセージに挿入してそのメッセージをネットワーク上へ送出し、再送に備えてそのメッセージを送信バッファにも登録する。これ以降、送達確認を受信するまでは、この移動元ノードはこのアドレスに対するいかなる要求も受け付けない。送達確認受信後、移動元ノードはキャッシュを invalid 状態にし、再送メッセージを送信バッファから破棄する。

もし一定時間内に移動元で送達確認を受信できない

表 2 実験環境

Table 2 A platform for evaluation.

項目	WS1	WS2	WS3	WS4
CPU 主記憶	SuperSPARC 50 MHz×2 64 MB	SuperSPARC 50 MHz×2 64 MB	SuperSPARC 50 MHz 64 MB	SuperSPARC-II 75 MHz 48 MB
ATM NIC	FORE SBA-200		FORE SBA-200E	
ATM 交換機	TOSHIBA AX-1500			

場合(図6(b)),もしくは,移動先で owner 移動メッセージを受信できない場合(図6(c))は,タイマが発火して再送が行われる.前者の場合,再送に対する移動先からの送達確認か,新たな owner 移動のメッセージ通過により移動元はキャッシュを invalid 状態にすることができる.後者の場合,移動元では送達確認をもらうまで再送バッファにメッセージが残っているので,移動先からの再送メッセージと再送バッファ内のメッセージを比較して同一であることが確認できればメッセージ損失が起こる前の通常処理に戻ることができる.

#### 4. 性能評価

##### 4.1 基本アクセス時間の評価

キャッシュ一貫性制御を行うべき図1のI/OプロセッサとしてFORE SBA-200 ATM NIC上のi960プロセッサを用い,Sun SPARCstation 10と20を計算ノードとして,このNICをATM交換機により接続した環境で読み込み/書き込み性能の計測を行った.SBA-200は,NIC上に25MHzのi960CAプロセッサとホスト計算機上にマップ可能な256Kバイトのメモリを搭載している.この256Kバイトメモリの一部をキャッシュメモリとして利用した.表2に測定に用いた実験環境を示す.

3.2.1項で述べたアクセスの分類に基づき,ノード数に対する各アクセス時間を計測した結果を図7に示す.ここでは,キャッシュミスアプリケーション内の遅延時間(リードミス,ライトミス,クリーンライト)とNIC内の要求メッセージ送受信の遅延時間に区別して計測した.ライトミス時のI&Rメッセージを送信した場合,3.3.2項で述べたように要求ノードは最新のキャッシュブロック取得後に送達確認を送信しなければならないが,ここでは送達確認が届くまでの時間ではなく,キャッシュブロックを取得して読み込み待ちが解除されるまでの時間を載せている.ただし,一連のアクセスの中で送達確認の送受信は行っていない.

リードヒット,ダーティライトに要する時間は1アクセスあたり1~2 $\mu$ 秒であった.一方,ホスト計算機

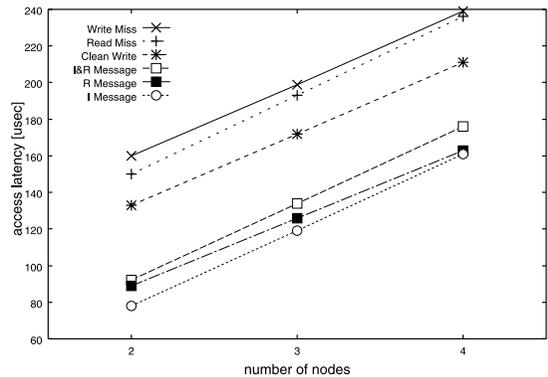


図 7 遠隔メモリアccessの遅延時間

Fig. 7 Latency of remote memory access.

上からNICメモリに直にアクセスした場合の時間は,1アクセスあたりそれぞれ,読み込みが0.9 $\mu$ 秒,書き込みが0.7 $\mu$ 秒である.したがって,読み込みで22%,書き込みで143%のオーバーヘッドが生じる.本システムでは,共有メモリに対する読み込み/書き込みコードをコンパイラが生成することが前提になっている.共有メモリへのアクセスに際しては,キャッシュ状態の確認が必要なため,その確認オーバーヘッドがキャッシュにヒット時にもアクセス時間に加算される.

図7より,ライトミス,リードミス,クリーンライトに要する時間は,ノード数に応じて線形に増加していることが分かる.そのオーバーヘッドは,ノードが1つ増えるごとに約40 $\mu$ 秒であった.また,クリーンライト,リードミス,ライトミスに要する時間は,2ノード構成の場合で150~160 $\mu$ 秒であった.クリーンライト時は,Iメッセージによって他ノードのキャッシュを無効化するだけなのでコストが低く,次いでリードミス時はRメッセージによってキャッシュ内容の送受信処理の遅延時間が増し,IメッセージとRメッセージの役割を同時に行うライトミス時が一番コストが高い.

本システムでは,ATMのコネクション数維持問題から環状接続上で巡回型ブロードキャストを利用して,ノード数に従って遠隔アクセス時間は線形に増加する.しかし,ATMは遅延変動の幅が小さく,また一貫性制御を計算ノードからNICにオフロードしたことにより安定したアクセス時間が得られる.

表 3 FFT の実行結果 (単位: 秒)  
Table 3 Execution time of FFT (sec).

ノード数	1	2	4
実行時間	0.68	0.94	0.86
高速化率	1.00	0.72	0.80

表 4 80 × 80 行列の乗算の実行結果 (単位: 秒)  
Table 4 Execution time of 80 × 80 matrix multiplication (sec).

ノード数	1	2	3	4
実行時間	4.30	2.67	3.03	1.88
高速化率	1.00	1.57	2.07	2.23

4.2 並列アプリケーションを用いた評価

SPLASH2<sup>7)</sup> の並列アプリケーションの中から FFT を用いて評価を行った。FFT は高速フーリエ変換を行うプログラムである。ここでは 2<sup>10</sup> 個のデータを用いて、逆フーリエ変換で計算結果の正当性の確認を行うモードで実行した。また、初期状態で owner はすべて、逆フーリエ変換の演算が行われるノード番号 0 に配置した。表 3 に FFT の実行時間と高速化率を示す。

実行結果から、FFT は 2 台と 4 台の間では台数効果を得たものの、両者とも 1 台よりも実行速度が遅いという結果となった。これは、本システムの実装上の問題から、データセットが大きくとれず、同期オーバーヘッドが顕在化してしまったことが原因と考えられる。ATM NIC 上には 256 K バイトのメモリ空間があるが、この上に一貫性制御プログラムとキャッシュを載せる必要があるため、共有メモリ空間が計算ノードあたり 150 K バイト程度しかとれないことがデータセットを大きくとれない要因である。

次に、80 × 80 行列 A, B, C を共有メモリ上に確保し、C = AB の乗算を行った結果を表 4 に示す。同期は行列 A, B の準備直後と演算終了後の 2 回使用した。実行結果から、台数にほぼ比例した良好な結果が得られた。これは、行列演算に対して十分なデータセットが得られたためである。また、行列 A および B に対するアクセスは読み込みのみであることからリードヒットによって通信回数が少ないこと、32 バイトキャッシュブロックによって行列 C への書き込みアクセス時にフォールスシェアリングが起きにくいためである。

4.3 セル損失についての評価

前節と同様に SPLASH2 の FFT を用いてセル損失についての評価を行った。実験は、セル損失をランダムに起こした環境下で、損失率を 0~3%まで変化させて行った。ここでの損失率は環状接続上をセルが 1 周したときの値である。また、図 7 より 4 ノード時の

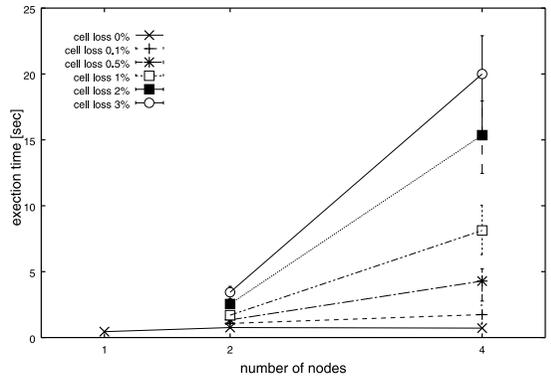


図 8 セル損失に対する FFT の実行時間の変化

Fig. 8 Variation of execution time of FFT for cell losses.

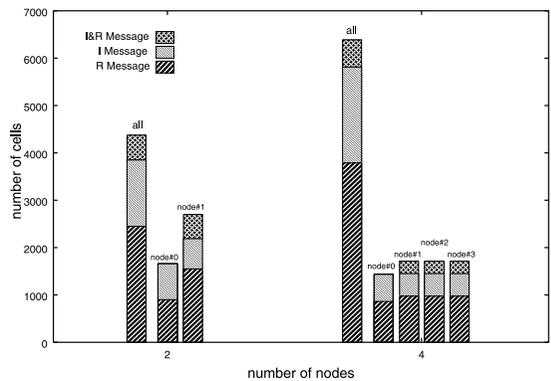


図 9 FFT のセル数

Fig. 9 Number of cells of FFT.

メッセージの遅延時間が最大 240 μ 秒であることから、再送タイムを最大遅延時間のほぼ倍の 500 μ 秒として実験を行った。すなわち、500 μ 秒以内にメッセージが帰属しなければセル損失と判断させた。図 8 にセル損失に対する FFT の実行時間の変化を示す。

図 8 より、セル損失率はノード数が増すにつれ実行時間に甚大な影響を及ぼしていることが分かる。セル損失が起きた場合、再送数が実行時間の律速条件となるが、その再送数は全体のセル数で決まるからである。図 9 にセル損失が起きない場合の FFT の通信セル数を示した。図から分かるように、2 ノードより 4 ノードの方が全体で 1.5 倍近くセル数が多くなる。したがって、同じセル損失率でも 4 ノードの方が再送数が増し、実行時間が増加するものと考えられる。

ATM では、コネクション確立時に申告した帯域を越えてセルが流れると交換機の流量制御アルゴリズムの下でセル廃棄が起こるが、予約帯域を守って流量制御をすれば広域の WAN 環境であってもセル廃棄されることはない<sup>12)</sup>。本プロトコルは送信側のコネクションが 1 本なので流量制御が容易であり、セル廃棄の起

きないように予約帯域を守ることが可能である。したがって、予約帯域内で最高の性能を引き出すことができる。万が一セル廃棄が起こった場合でも、図8から損失率0.1%以内であれば速度低下や実行時間の変動幅が小さいため実用に耐えうる範囲であるといえる。

## 5. 関連研究

ATMを用いてソフトウェアDSMを実現する際に扱わなければならない問題は、コネクション維持コストとメッセージ紛失誤りに大別される。これまでに、IVY<sup>5)</sup>をはじめとする多数のソフトウェアDSMシステムが提案されているが、この両方の問題を扱ったものは我々の知る限り提案されていない。TreadMarks<sup>1)</sup>はATMを利用して実験をしているが、AAL3/4をUDP/IPの代わりとして用いているためATMを扱っているとはいえない。

一般に、コネクション数が少ない接続トポロジとして環状接続が知られている。環状接続のネットワークでDSMを実現するシステムとして、これまでにMemnet<sup>4)</sup>がある。しかし、これは、2.2節で述べたとおりトークンパッシング型のネットワークを使用しているため、このプロトコルをATM上で直接動かすことは難しい。一方、メッセージをノード間で環状に巡回させてキャッシュの無効化を行うものとして、中條らが提案している巡回型マルチキャスト方式<sup>10)</sup>がある。これは、ディレクトリ管理されたキャッシュを持つノードの間のみで選択的に無効化要求を巡回させることにより、負荷の軽減と送達確認のメッセージ数削減によるネットワークトラフィックの緩和を実現している。しかし、無効化対象のノードは動的に変化するため、コネクション数を抑えつつ巡回型マルチキャストをATMのようなコネクション指向のネットワークの下で実現するのは困難である。

メッセージ紛失誤りを許容するものとして、高橋らが提案しているメモリアクセスプロトコル<sup>9)</sup>がある。これは、 $n$ 台のキャッシュ付きプロセッサと $m$ 台のメモリ装置をATMを用いて相互結合した並列計算機を前提とし、その上でDSMを実現するためのソフトウェアによるメモリアクセスプロトコルである。ネットワークインタフェース上に要求バッファを設け、アクセス要求ごとに状態を管理してタイムアウトによる再送を行うことでメッセージ紛失誤りを許容している。このため、通信プロトコル層を利用した場合に比べて状態数が少なく、また、状態変数の数も並行して処理するメモリ操作の数に比例するためハードウェアの簡素化が容易であるという利点がある。本提案プロトコ

ルも、紛失誤りを通信プロトコル層による送達確認処理で扱うのではなく、一貫性プロトコル側がそれを許容する点では同じである。一方、我々の方法は、全体のノード数を $n$ とすると、状態変数の数は1ノードあたり、最大 $n+1$ になる点では文献9)に劣っている。しかし、コネクション数の評価に関する明確な記述は文献9)にはなく、コネクション数は、最大 $n \times m$ 本になると考えられる。また、流量制御に関する記述もないため、実装に対する問題点が明らかではない。したがって、本提案プロトコルの方がより実践的であると考えられる。

本論文で提案したソフトウェアDSM実現プロトコルは、環状ネットワークポロジを用いて通信コネクション数を台数 $n$ に対して $O(n)$ で抑え、かつ、セル廃棄を一貫性プロトコルが許容する点で上記の実現方式とは異なる。

## 6. おわりに

本論文では、環状接続を用いてコネクション数を台数 $n$ に対して $O(n)$ で抑え、ATMのセル廃棄を許容するソフトウェアDSM向け一貫性プロトコルを提案した。さらに、SPLASH2などの並列アプリケーションを用いて実験を行い、台数効果とセル廃棄に耐性があることを確かめた。本方式における一貫性プロトコルは、ノード側から見た送信側のコネクション数は1本で済むため、ATMでは必須の流量制御の簡略化が可能である。これによって、セル廃棄の起きないように流量制御を行うことが容易であり、予約帯域内で最高の性能を引き出すことが可能である。

今後は、一貫性モデルとしてEntry Consistencyを導入し、メッセージ数の削減を図る予定である。

## 参考文献

- 1) Amza, C., Cox, A.L., Dwarkadas, S., Keleher, P., Lu H., Rajamony, R., Yu, W. and Zwaenepoel, W.: TreadMarks: Shared Memory Computing on Networks of Workstations, *IEEE Computer*, Vol.29, No.2, pp.18-28 (1996).
- 2) Bershady, B.N. and Zekauskas, M.J.: Midway: Shared Memory Parallel Programming with Entry Consistency for Distributed Memory Multiprocessors, Technical Report CMU-CS-91-170, Carnegie Mellon University (1991).
- 3) Carter, J., Bennett, J. and Zwaenepoel, W.: Techniques for Reducing Consistency-Related Communication in Distributed Shared-Memory Systems, *ACM Trans. Comput. Syst.*, Vol.13,

- No.3, pp.205–243 (1995).
- 4) Delp, G., Farber, D., Minnich, R., Smith, J. and Tam, M.: Memory as a Network Abstraction, *IEEE Network Magazine*, pp.34–41 (1991).
  - 5) Li, K. and Hudak, P.: Memory Coherence in Shared Virtual Memory Systems, *ACM Trans. Comput. Syst.*, Vol.7, No.4, pp.321–359 (1989).
  - 6) Minzer, S.E.: Broadband ISDN and Asynchronous Transfer Mode (ATM), *IEEE Communication Magazine*, Vol.27, No.9, pp.17–24 (1989).
  - 7) Woo, S.C., Ohara, M., Torrie, E., Singh, J.P. and Gupta, A.: The SPLASH-2 Programs: Characterization and Methodological Considerations, *Proc. 22nd Symposium on Computer Architecture*, pp.24–36 (1995).
  - 8) 市川明弘, 小野 航, 中條拓伯, 工藤知宏, 天野英晴: Home Proxy Cache による分散共有メモリの高速化, *情報処理学会論文誌*, Vol.40, No.5, pp.2016–2024 (1999).
  - 9) 高橋雅史, 大庭信之, 小林広明, 中村維男: 分散共有メモリ型並列計算機のためのメッセージ損失を許容するメモリアクセスプロトコル, *電子情報通信学会論文誌*, Vol.J79-D-I, No.9, pp.567–571 (1996).
  - 10) 中條拓伯, 藏前健治, 金田悠紀夫, 前川禎男: ソフトウェア DSM におけるコヒーレント・キャッシュシステムの実装と評価, *情報処理学会論文誌*, Vol.36, No.7, pp.1719–1728 (1995).
  - 11) 中村嘉志, 多田好克: ATM を前提としたソフトウェア DSM の実装法とシミュレータによる評価, *電子情報通信学会技術研究報告*, Vol.99, No.409, pp.33–39 (1999).

- 12) 松井康範, 内海秀介, 船渡大地, 中村嘉志, 成田多良, 細川達己, 徳田英幸: ATM-LAN 環境と ATM-WAN 環境の転送特性の比較, *情報処理学会研究会報告*, 95-OS-71, pp.99–104 (1995).

(平成 14 年 3 月 20 日受付)

(平成 15 年 7 月 3 日採録)



中村 嘉志 (正会員)

1994 年神奈川大学理学部情報科学科卒業。1996 年電気通信大学大学院情報システム学研究科博士前期課程修了。1997 年同専攻博士後期課程退学。同年同研究科助手を経て、現在産業技術総合研究所特別研究員。分散システムの研究に従事し、現在、情報支援システムに興味を持つ。IEEE, 電子情報通信学会各会員。



多田 好克 (正会員)

1985 年東京大学大学院工学系研究科情報工学専門課程博士課程修了。工学博士。同年電気通信大学電子情報学科着任。1992 年より電気通信大学大学院情報システム学研究科。並列・分散システムの記述法に興味を持ち、オペレーティングシステムをはじめとするシステムソフトウェアの実現法に関する研究に従事。ACM, 電子情報通信学会各会員。