並列ＲＤＢシステムにおける通信高速化方式

7W－4　小原　清弘　長坂　充　鍵政　豊彦　正井　一夫　宮崎　光夫
（株）日立製作所中央研究所　同ソフトウェア開発本部

1．はじめに

近年、大規模なデータベースの高速検索や高信頼性の要求、ダウンサイジングや規模に応じたスケーラブルな構成が可能などの理由により、並列計算機上で動作する、並列RDBシステムが注目されている。このような並列RDBシステムの性能を決める最も重要な要素の一つに、プロセス（プロセッサ）間通信性能がある。

プロセス間通信方式の中では、ソケットを用いたTCP/IPが最もポピュラーである。しかし、並列RDBシステム内の通信方式として採用するには一回の通信当りのプロセッサの負荷が重すぎ、システム全体の性能向上の阻害要因となってしまう。本稿では、メモリ間直接通信と呼ぶ、分散メモリ（疎結合）型並列計算機内の通信方式を用いた、並列RDBシステムの通信高速化方式を示す。最初に、分散メモリ型並列計算機内の高速なプロセス間通信方式である、メモリ間直接通信を示す。そして、メモリ間直接通信の並列RDBシステムへの適用方法を示す。

2．並列計算機および並列RDBシステムの構成

並列RDBシステムが動作するプラットフォームである分散メモリ型並列計算機は、各ノードが独立したプロセッサ、メモリやディスク等の二次記憶装置を持ち、それらのノードを高速ネットワークで接続した並列計算機である。各ノードでは独立したOSが動作する。分散メモリ型の計算機であるため、他のノードのメモリやディスクの内容を直接参照することはできない。ノード間の通信はメッセージ通信を用いる。

並列RDBシステムは、複数のノードに分散配置された、複数のサーバ（プロセス）で構成される。各サーバは互いに通信を行いながら並列に処理を進め、全体として一つのRDBシステムとして動作する。通信の内容は、処理データのパイプライン的な授受や制御の受渡しである。

3．メモリ間直接通信

メモリ間直接通信は分散メモリ型並列計算機向けに考案された、メッセージ通信方式の一種である。メモリ間直接通信の概念図を図１に示す。

メモリ間直接通信の手順は、次のようである。最初に、各ノード上の送受信プロセス共に、プロセス内の通信領域と、それに対応する物理メモリのマッピングを固定化する。マッピングを固定化した領域は、OSによるページングから除外し、物理メモリ上に常駐化させる。そして通信は、この固定マッピングしたメモリ間で、DMA転送のように、通信ハードウェアがネットワークを介し直接データ転送を行うことにより達成される。

この通信方式では、常に仮想空間が物理メモリ上にマッピングされているため、送信プロセス内の送信データがページアウトされている場合のページイン処理や、受信ノードのOSで受信データを一旦バッファリングし、受信プロセスが走行状態になるまで保存する処理が不要になる。また、送受信データのバッファリングに伴う、ユーザ空間とOS空間の間のデータコピーなどのオーバヘッドも不要になる。すなわち、「メモリ間のデータ転送」という、データ転送の本質的な部分だけを提供しているため、ネットワークのスループットにほぼ等しい高い通信性能が得られる。

一方、TCP/IP等を用いた通常の通信と比較すると、メモリ間直接通信は次のような欠点がある。

(1) プロトコルレス

TCP/IPは、フロー制御や送達確認、タイムアウトによる再送などのプロトコルを含み、信頼性の高い通信を提供している。これに対し、メモリ間直接通信は「メモリ間のデータ転送」そのものであり、これらのプロトコルを持たない。これらのプロトコル相当する処理は、より上位の、メモリ間直接通信を利用する並列RDBシステム側で実行する必要がある。

ところで、メモリ間直接通信では、フロー制御に関しては、原理的に受信バッファ不足が発生しないため必要ない。これは、受信先の通信領域が固定的に物理メモリへマッピングされているため、常時書き込みが可能となっているからである。しかし、データの破壊（転送時のビット誤り）や損失に対応し信頼性の高い通信を確保するため、送達確認や再送は必要である。

(2) 上書きの危険性

TCP/IP等の通信は、受信データをOSのカーネルが一旦バッファリングし、受信側で受信関数が呼び出されるとデータをカーネルからユーザ領域へ書き込む。このため、受信側の意図に反して、送信側の操作により受信したデータが変更されることはない。

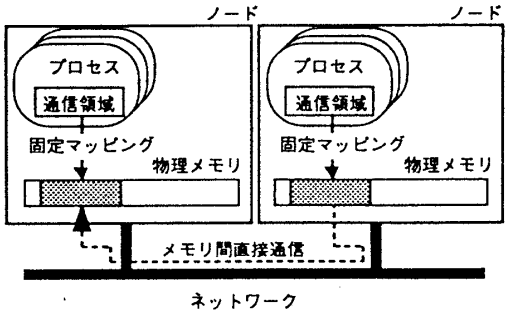

図１　メモリ間直接通信の概念図

Fast Communication Method for Parallel RDB System
Kiyohiro OBARA, Mitsuru NAGASAKA, Toyohiko KAGIMASA, Kazuo MASAI and Mitsuo MIYAZAKI
HITACHI, Ltd.

一方、メモリ間直接通信は、OSによるバッファリングを行わない「メモリ間のデータ転送」そのものであり、送信処理イコール受信領域の内容変更である。この操作は送信側の主導で行われ、通常、受信側は制御できない。したがって、複数の相手から同一通信領域に受信する場合や、送受信のタイミングの不一致等のプログラムのミスなどで、受信側が処理中のデータが、送信データで上書きされてしまう危険性がある。

4.並列RDBシステムのメモリ間直接通信の利用法

メモリ間直接通信には前章で述べたような欠点がある。しかし、通信の高速化を目指すにはメモリ間直接通信の利用が必要である。次に、これらの欠点をカバーする並列RDBシステム内でのメモリ間直接通信の利用方法を述べる。

(1) RPC形式の通信による送達確認の不要化

並列RDBシステム内で用いられる通信の形式を見ると、「処理要求—結果返却」を繰り返す通信が頻繁に用いられている。このような通信形態では、処理要求の送信に対する送達確認を結果返却で、結果返却の送信に対する送達確認を次の処理要求で代用することにより、送達確認のための通信を不要化できる。すなわち、見方を変えれば、個々の通信にはTCP/IPのような送達確認の機能は不要となる。

このような通信の性質に基づき、並列RDBシステム用の通信機能として、メモリ間直接通信を用いたRPC(Remote Procedure Call)形式の通信機能を提供する[1]。一回のRPCは、一回の「処理要求—結果返却」に対応する。メモリ間直接通信は送達確認の機能は持たないが、前述のように、送達確認を次の通信と兼用することにより、通信の送達に関する信頼性を確保する。

(2) 送受信領域の一対一対応による上書き防止

メモリ間直接通信は本質的に「メモリ間のデータ転送」である。このため、複数の相手から同時に受信する場合、ある相手からのデータを他の相手からのデータが上書きしてしまう可能性がある。図2(a)。このような状態を回避するため、送受信プロセス間で、送信領域と受信領域を一対一に対応させて用意する。図2(b)。このように、一つの送受信のコネクションに対し、独立した一組の通信領域を確保することにより、処理中のデータの上書きを防止する。

(3) ダブルバッファリング

メモリ間直接通信では、送受信は通信ハードウェアが非同期に行うため、送受信とRDBシステムの処理は並行して実行可能である。このため、各サーバにおいては、データ受信とデータ処理、データ処理とデータ送信をパイプライン的に実行できる。しかし、一つの通信相手に対し、ただ一つの通信領域だけしか用意していない場合、データ処理中の領域に新たなデータが受信され、正常なデータでの処理が行われなくなる。

これに対し、同一の通信相手に対し複数個の通信領域を確保すれば、送受信に用いる通信領域とデータ処理を行う通信領域を交互に切り替えて利用でき、送受信とデータ処理のパイプライン処理が可能となる。図3。このような通信領域の利用方法はダブルバッファリングと呼ばれる。

5.メモリ間直接通信の性能見積り

一回の通信にかかる処理量を、プロセッサの命令ステップ数換算で見てみると、その大部分は通信データのコピーとプロトコルのために費やされている。一方、メモリ間直接通信は、すでに述べたように、フロー制御や送達確認、タイムアウトによる再送などのプロトコルを持たない。またバッファリングを行わないため、それに伴うユーザ空間からOS空間へのデータコピー処理も不要である。このため、非常に高速な通信を提供できる。送信データ量4 KByteの場合、メモリ間直接通信はTCP/IPと比較して、1/4以下の処理量で一回の通信が実行できる見通しを得ている。

6.おわりに

本稿では、メモリ間直接通信と呼ぶ、分散メモリ型並列計算機内の通信方式を示した。そして、メモリ間直接通信の並列RDBシステムへの適用方法を示した。本稿で述べた通信方式は、開発中の並列RDBシステムにインプリメントし評価した。

参考文献

[1] 藤原　他：並列ＲＤＢシステムにおける通信機能の実現方式，情報処理学会第49回全国大会，7W-3，1994.

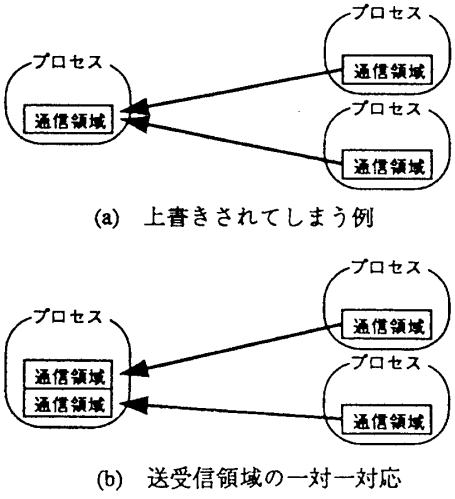

図2　送受信領域の対応

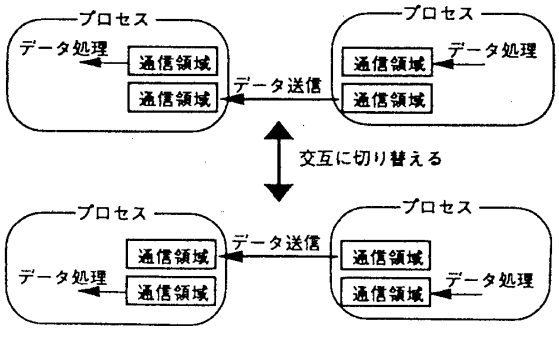

図3　ダブルバッファリング