計算機アーキテクチャ　105－12
ハイパフォーマンス
コンピューティング　50－12
（１９９４．　３．　１１）

マルチコンピュータにおける
ノード間高速通信アーキテクチャの検討

村山秀樹　吉澤　聡　相本　毅　井内秀則　村瀬彰一　林　剛久
日立製作所　中央研究所

ネットワークの高性能化に伴い、メッセージ通信のボトルネックがネットワークハードウエアからソフトウエア処理のオーバヘッドに移行している。マルチコンピーティングでは、システムスループット向上のためにメッセージ通信のオーバヘッドを低く押さえることの重要度が高まっている。通信処理ソフトの性能評価を行い、データコピーレス化、割り込み回数の低減を行なうことが性能改善効果が高いことが分かった。ここでは、仮想メモリ機構を持つマシン上でデータコピーレス化を行なうためのDLA(Dynamic Link Allocation)方式と、割り込み回数の低減を行なうために有効なラストパケット割り込み機構を提案した。

Designing High Performance Communication Mechanism for Multi Computer Systems

Hideki Murayama　Satoshi Yoshizawa　Takeshi Aimoto　Hidenori Inouchi　Shooichi Murase　Takehisa Hayashi
Hitachi, Ltd. Central Research Laboratory

With the performance of computer networks growing, the bottle neck of communication has shifted from the performance of network hardware to the overhead of software. In multicomputer systems, the efficiency of communication is important. We analyzed the behavior of communication software and we designed two hardware mechanisms for eliminating software bottle necks for multi processing with virtual memory systems. One is data copy-less communication mechanism (DLA:Dynamic Link Allocation) and the other is last packet interruption mechanism.

1. はじめに

現在、ローカルエリア・ネットワークの普及にともない、単体のWSではできない処理を複数のマシン上で並列に処理する分散処理システムが普及しつつある。光ファイバ等の高速ネットワーク媒体の出現により、ネットワーク層でのレイテンシ・スループットが飛躍的に向上している。このため、従来は、適用範囲外であった処理に対しても分散処理が適用できる可能性が高まって来ている。

マルチコンピーティングでは、システムスループット向上のめたにメッセージ通信のオーバヘッドを低く押さえることが重要である。メッセージ通信のソフトオーバヘッド低減のために通信ソフトウエア処理の性能評価を行ない、ボトルネックの解析を行なった。以下、マルチコンピュータにおいて高速な通信を行うためのハードウエア付加機構の検討について論じる。

2. マルチコンピュータシステム

2.1 システム構成

本研究の対象となるマルチコンピュータシステムの構成例を図2.1に示す。

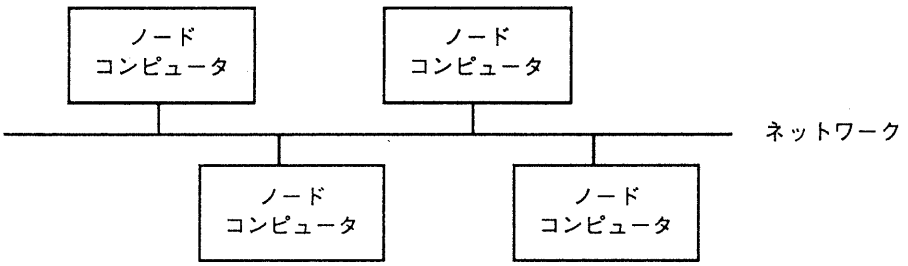

図2.1 マルチコンピュータシステム

マルチコンピュータシステムは、複数のノードコンピュータがネットワークにて結合して構成され、並列処理を行うシステムである。各ノードコンピュータは、ネットワークを介してメッセージ通信を行いながら並列処理を進めていく。マルチコンピュータシステムでは、リソース集中によるボトルネックの発生を防ぐために、リソースを分散して保持する。リソースが分散しているので、マルチコンピュータシステムでは、必要に応じてオーナーに対してリソースの授受のためのメッセージ通信が頻発する。従って、メッセージ通信のソフトウエアオーバヘッドを低減することがシステムスループット向上のために重要である。

2.2 通信処理ソフトウエア性能評価

メッセージ通信に使用されているプロトコルとして、ソケットインタフェースを持つTCP/IP、UDP/IPが主流を占めている。ネットワークの高性能化と低ソフトウエアオーバヘッドの要求から、プロトコルによるチェックサム処理を削減するオプションを持つUDPプロトコルが実現されている。現在、チェックサム処理を削除したUDPプロトコルは、実用に供されているプロトコルの中で最もソフトウエアオーバヘッドが低いものの一つである。ここでは、チェックサム処理を削除したUDPプロトコルをベースとして低ソフトウエアオーバヘッドな通信方式を検討する。まず、UDP/IP(チェックサムオフ)の性能評価を行い、内部処理の比率を解析した。この結果を図2.2に示す(データサイズ4 Kbyte)。図に示すように、処理の大半をデータコピー処理が占めている。以下、データコピーレス化を中心としてソフトウエアオーバヘッド低減のための高速化方式の検討を行う。

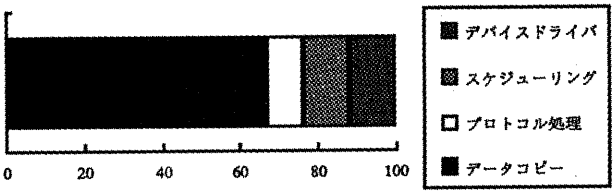

図2.2　UDP/IPにおける処理比率(チェックサムオフ)

3　データコピーレス処理方式

図2.3に示したように、通信ソフトウエア処理のプロセッサオーバヘッドの最たる部分はデータコピー処理である。以下、データコピーレス化による高速化方式を検討する。データコピー処理が必要な理由は、送信側と受信側で異なる。以下、送信側と受信側を分けてデータコピー処理の要因とデータコピー削減方式の提案を示す。

3.1　送信側のデータコピーレス化

UDPに於て送信側の処理でコピーが必要となる要因は、以下の通りである。

(1) 送出するデータが主記憶常駐領域に存在することが必要であるため。

(2) プロセスがブロックされずに動作できる非同期動作を行うため。

(1)は、ネットワークアダプタが、送信動作を行なうときにデータが主記憶内に存在しなければ、DMA処理によるデータ転送ができないためである。データ送出処理を行なう対象のデータが主記憶内に常駐することを保証するために、ページアウトされる可能性のあるユーザ空間からOS内部のページアウト対象外の主記憶常駐領域にコピーする処理を行う必要がある。この問題を解決するためには、データ領域を主記憶常駐させるための機構を追加することが必要である。

(2)の非同期動作を行うためには、データ領域の保護のためにデータコピーをなくすことができない。従って、データコピーレス化のためには、同期動作を行なう必要がある。現状のAPIに変更を加えないで同期動作を実現するためには、センドシステムコールの終了をデータ送信処理の終了まで待たせる同期ブロッキング型の採用が考えられる。しかし、同期ブロッキング型プロセスがブロックされてしまうことにより、並列動作が犠牲に可能性がある。そこで、同期動作を行うが、プロセスがブロックすることなく動作を行う同期ノンブロック型の新規APIを採用することを検討する。同期ブロッキング型と同期ノンブロッキング型の得失を表3.1に示す。

表3.1　非同期型および同期型の得失

項目	同期ブロッキング型	同期ノンブロッキング型
動作	・センドシステムコールからの復帰を送信処理の終了まで待たせる。	・センドシステムコールは送信処理の終了を待たずに復帰する。
長所	・データコピー処理が不要。 ・APIは変更不要。	・データコピー処理が不要。 ・アプリケーションの非同期動作が可能。
短所	・アプリケーションの非同期動作が不可能。	・新規APIが必要となる。 (送信終了を確かめるシステムコール成功後にセンドデータ領域が再利用可能となる。)

3.2 受信側のデータコピーレス化

UDPに於て受信側の処理でコピーが必要となる要因は、以下の通りである。

(1) データが到着する領域が主記憶常駐でなければならない。

(2) レシーブシステムコールにて受信アドレスが通知される。

(1)は、ネットワークアダプタが、受信動作を行なうときにデータ領域が主記憶常駐領域内に存在しなければ、DMA処理によるデータ転送ができないためである。従って、(1)を解決するためには、受信側のバッファ領域がデータの到着に先だって主記憶に常駐していることが必須である。(2)は、レシーブシステムコール発行がデータの到着以前という保証がないためにデータの到着時に転送先が不定ことに起因する。(2)を解決するためには現状のAPIを変更することが必要である。

(1),(2)を解決するためには、以下に示すAPIの変更が必要である。(1) データの到着に先行して受信領域のアドレスを通信ソフトに通知するAPI(受信領域のプリアサイン)を追加する。(2) 受信領域のプリアサイン処理中では、受信領域を主記憶常駐領域とする。

上記に示したAPIを追加することにより、受信処理の基本動作は以下のようになる。

(1) コネクションをオープンする。

(2) 受信領域のプリアサインを行う。

(3) レシーブシステムコールの発行。

(4) 次の受信に備えて、受信領域の開放を行う。

(5) 以降、(3),(4)を繰り返す。

(2)はデータの到着に先行することが必要であるために(1)処理の内部で行うことが望ましい。上記に示したAPIの追加を行うことにより、受信処理のデータコピーレス化を図ることができる。APIは、対象とするユーザによって様々なバリエーションが考えられるので、ここでは詳細については言及しない。[1]

以下、ソフトウエアは上記のAPIの基本要件を満たして動作すると仮定した場合に、データコピーレス通信を行うためのハードウエア付加機構の要件とその方式検討について述べる。

4. コピーレス通信サポート機構

コピーレス通信をサポートするためのハードウエア付加機構の要件を以下に示す。コピーレス通信を行うためには、パケットの受信側のデータ領域を特定する情報に従ってデータを主記憶内にDMA転送する(ハードウエアデマルチプレクス)機能を持つことが必須である。しかし、マルチプログラミング環境を持つ仮想記憶マシンでの実現を考慮すると以下の問題も同時に解決しなければならない。

(1) あるプロセスのバグによって、他のプロセスにダメージを与えない。(プロテクション)

(2) 仮想アドレス連続の領域(物理アドレスで連続が保証されない)に対する転送を行う。

(1)は、データコピーレス化により、ネットワークからのデータは直接ユーザの受信領域に転送される。プロテクション機構がなければ、送信側の設定の誤りにより、別のユーザ領域やシステム領域の内容を破壊する可能性がある。したがって、ハードウエアによるセキュリティチェックを行う機構を持つことが必要である。

(2)は、仮想記憶マシンは、主記憶領域をページを単位として管理している。そのため、ユーザの要求する受信領域は、仮想アドレス空間での連続領域であることしか保証されない。したがって、仮想アドレス空間での連続領域は、主記憶常駐された実アドレス空間ではページ単位に分割された非連続の領域となる。以下、プロテクション機構と物理非連続領域へのデータを転送機能を持つコピーレス通信サポート機構の方式を提案する。

4.1 DLA方式の提案

上記に示した問題を解決するためのハードウエア付加機構としてDLA(Dynamic Link Allocation)方式を提案する。DLA方式の特徴は以下の通りである。

(1) データコピー処理をなくすためにパケットを受信プロセスの領域に直接転送する。
(2) プロセス間の保護を実現するためハードウエアで、セキュリティーキーチェックを行う。
(3) セキュリティーキー及びDLAポート数の制約を緩和するために主記憶上に制御情報テーブルを持つ。
(4) データ受信バッファメモリの管理は、受信側で行う(ノード間の独立性を保つ)。

以下、DLA方式の構成とその制御動作を示す。

4.2 DLA方式ハードウエア構成

DLA方式ハードウエア構成を図4.1に示し、動作概要を以下に示す。

(1)受信バッファのプリアサイン時に、DLAPortテーブルの要素、アドレス変換テーブルを初期化し、対応するDLAポートのデータ領域を主記憶常駐領域にする。
(2) 送信側は、受信アドレスを決定するための情報(Port ID)とプロテクションのためのセキュリティキーを付加したパケットを送出する。
(3) 受信側ハードウエアは、Port IDによってDLA Portテーブルから対応するエントリを取り出す。
(4) パケットに付加されたセキュリティキーと(3)で取り出した内容のセキュリティキーを比較することでプロテクションチェックを行う。
(5) プロテクションチェックが成功したら、アドレス変換テーブルの内容にしたがって、受信側の物理アドレスを決定してDMA処理により受信側の領域に直接データ転送を行う。

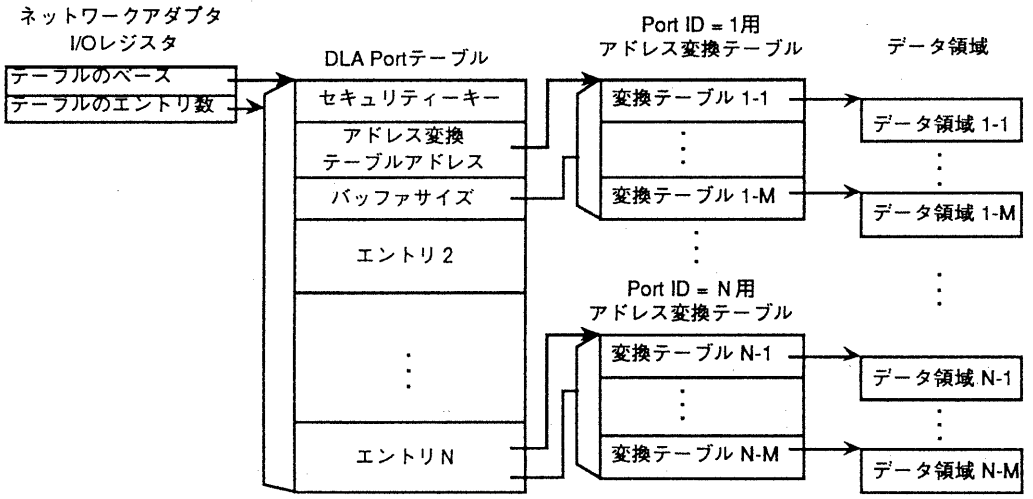

図4.1 DLA方式ハードウエア構成

以下、プロテクション機構及び仮想アドレスへの転送機構の詳細を示す。

4.2.1 プロテクション機構の実現

プロテクションは、以下の処理によって行なう。

(1) 送受信で通信領域に対するアクセス権を示すキーを定める。
(2) 送信側では、キーを付加してパケットを転送する。
(3) 受信側では、到着時にパケット内のキーと通信領域のキーをチェックする。

ハードウエア付加機構がなければ、割込処理で上記処理を行なうことになる。これは、ソフトウエアオーバヘッドの低減という目的に反する。したがって、プロテクションのチェックを行うハードウエア機構として付加した。

4.2.2 仮想アドレスへの転送機構

仮想メモリ管理機構を持つシステムでは、仮想アドレス空間での連続領域が物理アドレス空間では連続の領域であることが保証されない。仮想アドレスで連続な領域とプロセッサの物理アドレス空間の対応を図4.2に示す。

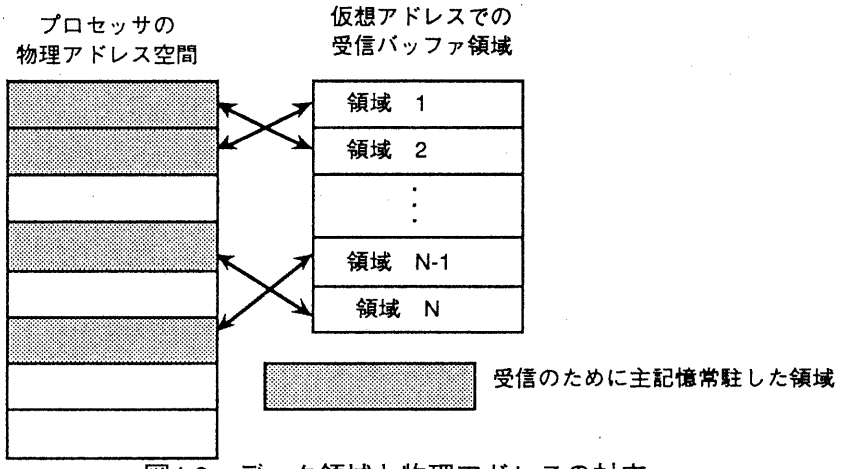

図4.2 データ領域と物理アドレスの対応

図4.2に示すように受信バッファ領域は、仮想アドレス空間での連続領域となっているが、プロセッサの物理アドレス空間では、非連続な領域となっている。物理アドレス空間で非連続な領域にデータを転送するためには、受信バッファ内での位置をプロセッサの物理アドレス空間に変換するためのアドレス変換機構を持つことが必要である。そこで、到着したパケットを転送するアドレスを取得する際に仮想アドレスを物理アドレスに変換する機構を設けた。

仮想アドレスで連続な受信領域に対する転送は、以下の動作によって行なうことができる。パケットを主記憶に転送する前に、アドレス変換テーブルの情報によって受信の物理アドレスを決定する。次に、決定された受信の物理アドレスに対してDMA処理によるデータ転送を行なう。上記の処理により、物理非連続なデータ受信領域(仮想アドレスでは連続な領域)に対する転送を行うことができる。

5. 割込回数の削減

DLA方式の提案によって、コピーレス通信を行なうためのハードウエア付加機構について述べた。コピーレス通信によって、コピー処理が不要となると次に削減対象とするソフトウエア処理はスケジューリングのオーバヘッドである。スケジューリング処理を低減するために、スレッド等を用いると効果がある[2]。さらに、ここではハードウエア付加機構によって割込回数を削減し、スケジューリング処理のオーバヘッドを削減することを検討する。以下、本ハードウエア付加機構について示す。

5.1 ラストパケット割り込み機構

受信側では、データの到着がレシーブシステムコールに先行していた場合には割込を発行する必要がない。割込は、レシーブシステムコールをデータの到着に先行して発行してブロックされたプロセスを起動するために必要である。従って、ブロックされているプロセスがなければ割り込みを発行する必要がない。即ち、ブロックされているプロセスの存在を示す状態を保持することによって割込の発行を制御できる。

また、高スループットを得るために大パケットによる通信を行うと、他のパケットが割り込むことができない。パケットが送信できるまでの最大時間の延長によってレイテンシが悪化する。この問題を解決するためにレイテンシを増大させない程度のサイズでパケットを送信して、データ領域の最後が到着したときに割り込みを発行するハードウエア付加機構が考えられる。緊急のパケットを送信したいノードコンピュータは、データ全体を一括送信して割込を発行させることでレイテンシを低くおさえることができる。

上記に述べた、ブロックされているプロセスの存在状況を管理する機構と、データ領域の最後のパケット到着にて割り込みを発行する機構を実現するハードウエア付加機構としてラストパケット割り込み機構を提案する。図5.1にラストパケット割り込み機構を示す。

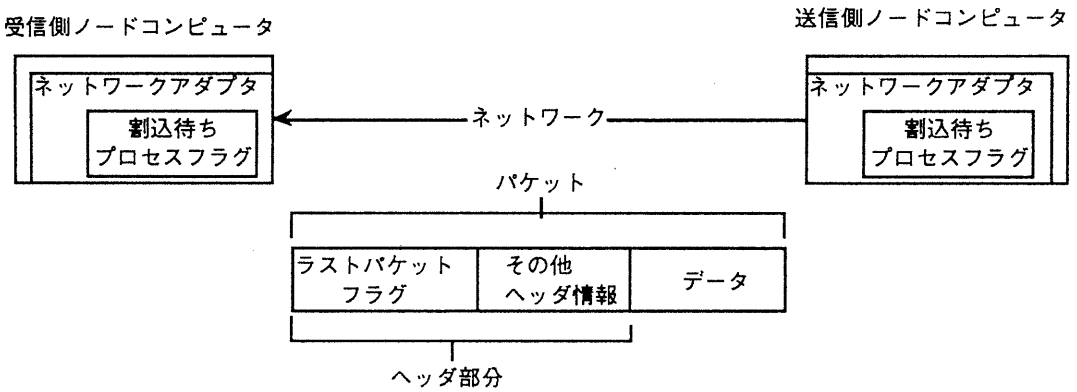

図5.1 ラストパケット割り込み機構の方式概要

以下、ラストパケット割り込み機構の動作概要を示す。

受信側のノードコンピュータのレシーブシステムコールがデータの到着に先行していない場合は、割込を発行させる必要はない。

レシーブシステムコールが先行し、プロセスがブロックされている場合の動作を以下に示す。

(1) 受信プロセスのブロック時に、ネットワークアダプタの割込待ちプロセスフラグをオンにする。
(2) 送信側では、通信の際の最後のパケットにラストパケットフラグをオンにする。
(3) 受信側では、ネットワークアダプタの割り込み待ちプロセスフラグがオンである状態とラストパケットフラグがオンである状態のAND条件によって、割り込みの発行を制御する。

上記の動作をすることにより、受信側は大量データを複数のパケットに分けて受信している間に、別のノードからの緊急パケットの到着が割り込んで先行して受信できるために低レイテンシを犠牲にすることなく高スループット通信を実現することができる。

6. 効果

DLA方式を採用した場合のUDP/IP(チェックサムオフ)とのソフトオーバヘッド比推定を図6に示す。縦軸は、UDP/IPの4kbyte転送時のソフトオーバヘッドを1とした相対値である。

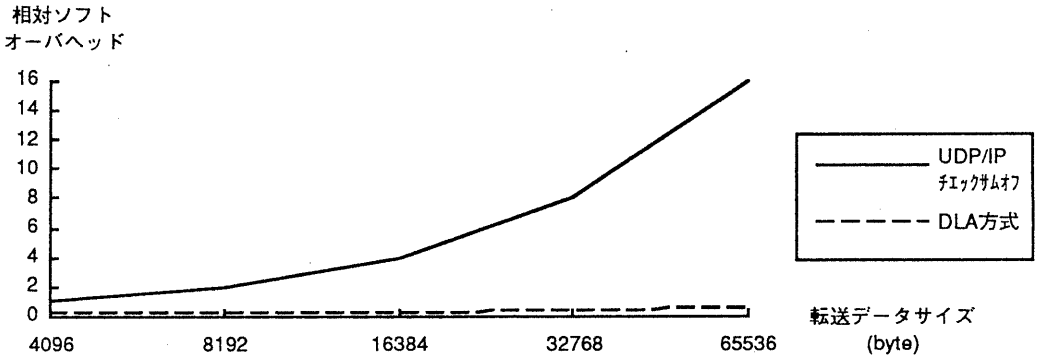

図6 ソフトオーバヘッド比

ラストパケット割込により、パケット到着時に割込が発行される割合の低減と高スループット要求時にパケットサイズが変化しないので、レイテンシは増加しない。ラストパケット割込に関する定量的な評価は、システム全体の動作によって変化するためにここでは、議論しない。

7. おわりに

通信ソフトウエアの低オーバヘッド化を目的として、マルチプロセス動作し仮想メモリ機構を持つマシン上でデータコピーレス通信を行う際に有効なハードウエア付加機構として、DLA(Dynamic Link Allocation)方式とラストパケット割り込み機構を提案した。今後は、シミュレーション等により、DLA方式の有効性と実現可能性を検証し、更に様々なユーザプログラムへの適用を考慮したコピーレス通信を行うためのAPI方式の検討を行う予定である。

8. 参考文献

[1] J.J.Dongarra etal. A Proposal for a user-level,message passing interface in a distributed memory environment Technical Report TM-12231,Oak Ridge National Laboratory, June 1993

[2] 前川他、分散オペレーティングシステム、共立出版、1991