通信遅延を低減したプロセッサ間通信機構の提案

飯塚 剛[†] 佐藤裕幸[†] 落合真一[†]
伊藤隆弘[†] 斉藤成一[†]

組み込み用途における分散メモリ型並列処理システムに最適化したプロセッサ間通信機構「*QuickPass*」を提案する．*QuickPass* では，システム性能に大きく影響するメッセージ通信遅延を最小に抑えるために，シンプルでかつユーザレベルから直接制御可能なインタフェースを開発した．このインタフェースは，資源保護機能の実現と低コストで実装可能という特長も兼ね備える．通信遅延性能としては，アプリケーションプログラム間で片道 2μ 秒以下という見積り結果を得た．これは，他方式の約 4 倍の性能にあたる．

A Proposal of Latency-reduced Inter-processor Communication Mechanism

Tsuyoshi Iizuka,[†] Hiroyuki Sato,[†] Shinichi Ochiai,[†]
Takahiro Ito[†] and Seiichi Saito[†]

We propose an inter-processor communication mechanism named "*QuickPass*", which is optimized for distributed memory parallel processing systems in embedded applications. We have developed a simple interface directly controled in user level to minimize the latency of message passing. The *QuickPass* interface has additional features that the resources are fully protected and that the implementation cost remains low. The one-way communication latency between application programs is estimated less than 2 usec, which means about four times better than other communication mechanisms.

1. はじめに

近年，レーダーシステムや通信機器などの組み込み分野においても，ますます高度な処理が必要になってきている．従来このような分野では，特定の処理を高速に行う専用ハードウェアや，多数の DSP を用途に応じた形態で接続するなどして対応していた．しかし，最近では汎用マイクロプロセッサのコストパフォーマンス向上が著しく，組み込み分野においても，専用ハードウェアの置き換えやプロセッサ個数の削減による低コスト化と，より複雑な処理による高機能化を目指して，汎用マイクロプロセッサを複数接続した高性能並列処理システムに対する期待が高まりつつある．ただし，組み込み分野においては，次に示すようなニーズに対応する必要がある．

(1) **リアルタイム処理** 処理時間に上限が規定されるため，機能または負荷を明示的に分散させる必要がある．機能または負荷分散に使われるプロセッサ間通信の遅延を最小に抑えなければならない．

(2) **信頼性&可運用性** 信頼性とトラブル解析性が重要で，その実現のために保護機能が必要になる．また，万一トラブルが発生しても，システムを動作させたままで故障モジュールの切り離しと交換（活線挿抜）が必要な用途もある．

(3) **多様なシステム構成** システムで必要な各種処理を複数のモジュールに分散する．各モジュールは処理内容に適したプロセッサ，I/O，ソフトウェアなどで構成されるので，多様なシステム構成となりうる．

(4) **小型&低コスト** 各モジュールは，Compact-PCI [1] などの比較的小型の基板に実装する必要があり，かつ低コストでなければならない．

以上の点から，主に市販アプリケーションの高速処理を目的とするような汎用の SMP または分散共有メモリ型のサーバ計算機を，組み込み用途のシステムに広く適用するのは困難である．

本稿では，このようなニーズを満たす高性能な並列処理システムの実現に不可欠な，低遅延かつ低コスト

† 三菱電機株式会社情報技術総合研究所
Information Technology R&D Center, Mitsubishi Electric Corporation

のプロセッサ間通信機構「*QuickPass*」を提案し，その有効性を明らかにする．*QuickPass* は，アプリケーション間を接続する各階層において，データを「すばやく渡す」ことにより，通信遅延を最小化することを基本コンセプトとしており，通信遅延の見積りでは 2 μ 秒以下と，他方式の約 4 倍の性能を得る．

2. 関連研究とその課題

並列計算機アーキテクチャの開発には，様々な取り組みが行われている[2]．この中から，我々は 1 章で示したニーズを満たすために，高速な通信路を介したメッセージ通信によってプロセッサ間の通信を行う方式に注目した．この分野の研究における技術動向を 2.1 節から 2.4 節に示すが，組み込み分野へ適用するには依然として 2.5 節に示すような課題が残されている．

2.1 高速通信路

近年，各種ネットワーク通信路の高速化は著しく，汎用 LAN の Gigabit Ethernet や，並列処理システム専用の Myrinet[3]，cLAN[4]，RHiNET[5] などにみられるように，1 Gbit/秒クラスの伝送速度が実現されている．これらの通信路に共通することは，各ホストは Point-to-point でスイッチと接続され，送信側ホストから送出されるデータはスイッチによって受信側ホストに宛てて交換制御されることである．

2.2 高速ホストインタフェース

通信路とホストとのインタフェースは，PCI[6] などの高速なシステムバスに接続されるアダプタカードとして実装されることが多い．PCI で接続することにより，ホスト内では最大 500 MB/秒程度（64 bit@66 MHz の場合）の転送速度を得ることができる．PCI および同プロトコルの CompactPCI などは，そのデータ転送速度と汎用性からみて適切な選択と考えられる．

2.3 ユーザレベル通信

通信路とホストインタフェースが高速化する一方で，通信プロトコルの処理は従来どおり OS で実行されるため，そのオーバヘッドがメッセージ通信のボトルネックとして顕在化してきた．TCP/IP などのプロトコルを使って通信する限り，通信路とホストインタフェースの高速化の恩恵を受けられない．

この問題に対しては，OS を介すことなくユーザレベルで直接メッセージ通信を制御する方式がいくつか提案されており，文献 7)，8) では各方式に関する分析がまとめられている．主に研究目的の AM[9]，BIP[10]，FM[11]，PM[12]，VMMC-2[13]，U-Net[14] などがあげられているが，他に商用目的の VIA（Virtual Interface Architecture）[15),16)] がある．これらは通信路に上述の Gigabit Ethernet を使用するもの[17)]，Myrinet を使用するもの[10)~13),15)]，cLAN を使用するもの[16)] などに分類できる．

表 1 低遅延通信の比較
Table 1 Comparison of low-latency communication systems.

	BIP/ Myrinet	PM/ Myrinet	VIA/ cLAN	*QuickPass*
伝送速度	1.2 Gb/秒	1.2 Gb/秒	1.2 Gb/秒	2 Gb/秒
IOP*1	あり*2	あり*2	なし	なし
物理アドレス	SW*3	FW*4	Table*5	Offset*6
保護機能	なし	不完全*7	完全	完全
通信遅延時間	約 5 μ 秒	約 7.5 μ 秒	約 7.5 μ 秒	2 μ 秒以下
主な適用分野	クラスタ	クラスタ	クラスタ	組込システム
	バッチ	バッチ	バッチ	リアルタイム
対応 OS	Linux	Linux, NetBSD	Linux, WindowsNT	LynxOS, Linux

*1 アダプタ搭載 I/O 制御プロセッサ．*2 LANai プロセッサ．
*3 ホストのソフトウェアで生成．*4 IOP のファームウェアで生成．
*5 変換テーブルで変換．*6 ベースアドレスにオフセット加算で生成．
*7 ギャングスケジューリングにより補完．

これらの通信方式はライブラリの形態で実装されており，アプリケーション間の片道の通信遅延では短いメッセージの場合およそ 5〜20 μ 秒程度，スループットでは長いメッセージの場合 100 MB/秒程度の性能が得られている[7]．

2.4 低遅延通信の比較

2.3 節であげた通信方式のうち，通信遅延で 10 μ 秒以下を実現している BIP，PM，VIA との機能および性能の比較を**表 1** に示す．

(1) **BIP** 通信路に Myrinet を使い，アダプタに搭載される I/O 制御プロセッサが通信ハードウェアを制御する．ホストメモリに置かれる通信データの物理アドレスは，ホストプロセッサ上のソフトウェアで管理されるため，アダプタ側には生成機能が実装されない．保護機能もサポートされないため，マルチプログラミングに対応できない．

(2) **PM** 通信路には BIP と同様主に Myrinet を使うが，通信データの物理アドレスはアダプタ上の I/O 制御プロセッサが管理する．保護機能に関しては，PM 単体では不完全だが，ギャングスケジューリング[18)] により，通信路に接続されるシステム全体にわたる時分割制御で実現される．

(3) **VIA** アダプタ上のアドレス変換テーブルを用いて，ハードウェア的に通信データの物理アドレスを生成し，同時に保護機能も実現している．これらの機能とソフトウェアとのインタフェースは，専用のハードウェア（cLAN が一例）として実装され，アダプタには通常 I/O 制御プロセッサは搭載されない．

2.5 課 題

まず，現状の他方式では，保護機能の実装と通信遅

延の最適化を両立できていない．表 1 にあげた低遅延通信方式のうち，完全な保護機能を備えるものは現状 VIA しかなく，その通信遅延は約 7.5 μ 秒[4] であり，十分な性能とはいえない．

また，データベース処理や科学技術演算などバッチ処理を主な応用分野とする計算機クラスタが想定されており，1 章であげたリアルタイム処理や多様なシステム構成といった，組み込み分野でのニーズに対応できない．

さらに，プロセッサ間通信にかかるコスト（アダプタとスイッチの価格）の面でも，組み込み分野の製品に適用するには高すぎるため，ハードウェアの簡略化などによるコスト低減が課題になる．

3. 課題の分析

以下では，2.5 節であげた他方式における課題について分析する．

3.1 保護機能

従来一般的であった OS を介した通信では，OS が各アプリケーションからの通信要求を一元管理するので，アプリケーションおよびシステムの資源と，アダプタ上のコンテキストは保護されていた．つまり，アプリケーションが他のアプリケーションまたはシステムの資源を不正にアクセスすることは防止され，またアダプタの設定および処理が他の通信要求で破壊されることはなかった．しかし，ユーザレベル通信では OS が介在しないので，何らかの資源保護メカニズムを導入して，システムの信頼性を確保する必要がある．

しかし，BIP では上述のように保護機能がサポートされておらず，我々の要求条件を満たさない．

PM ではアダプタ上の保護機能を簡略化して，ギャングスケジューリングで補完している．ある瞬間にはシステム全体で 1 つのアプリケーションが通信路を占有するうえ，すべてのアダプタ上のコンテキスト終了を待って次のアプリケーションに切り替えるため，アプリケーション切替えに要する時間も長くなる．このため，リアルタイム性が必要な分野には適用できない．

また，VIA ではアダプタのハードウェアで複数のコンテキスト（VI）をサポートして VI 間の干渉をなくすことにより保護を実現するが，アダプタにアドレス変換とメモリ管理機能が必要となるため，処理の複雑化による通信性能劣化とコスト上昇の危険性がある．さらに，ホストメモリ上に置くディスクリプタにより通信を制御するが，スレッド間の保護に対応するには，ディスクリプタの操作時にソフトウェアで排他制御しなければならず，通信遅延の増大につながる．

3.2 通信遅延

通信遅延の要因を，送信オーバヘッド，転送処理時間，受信オーバヘッド，エラー検出によるものに大別して，各々について分析を試みる．

(1) **送信オーバヘッド** アプリケーションが通信ライブラリをコールしてから，アダプタがデータ送信処理を開始するまでの時間であり，送信バッファへのデータコピー，送信制御情報およびヘッダの生成と設定，アダプタへの送信起動などが含まれる．2.3 節であげたユーザレベル通信方式においても，データコピーを削除するゼロコピー転送方式は広く実装されているが，なお 2〜4 μ 秒程度の送信オーバヘッドが残る[7]．この削減のためには，ホストとアダプタ間の送信制御情報交換に費される時間を短縮する必要がある．送信制御情報はホストまたはアダプタのメモリ上にディスクリプタとして置く方法が一般的であり，ソフトウェアによるディスクリプタ生成と，アダプタへの転送に時間を要する．また，Myrinet のようにディスクリプタの解釈とそれに対応した制御をアダプタ上の I/O 制御プロセッサがファームウェアで実行するタイプでは，いっそう遅延が増大することになる．さらに，3.1 節で述べたように，保護機能の実装が通信遅延の増大要因になりうる．

(2) **転送処理時間** アダプタが送信起動されてから，受信側アダプタにデータが届くまでの時間であり，アダプタによる送信データリード，データバッファリング，通信路へのヘッダおよびデータの送出，スイッチによる交換処理などが含まれる．まず，ホストメモリからの送信データリードと，通信路への送出をパイプラインで並列に動作させることができれば，処理時間を短縮することが可能なはずである．しかし，一般的なネットワークの中には，通信路でのアンダーフローを防止するために，アダプタ内に送信データをある程度バッファリングしてから通信路へ送出を開始するものがある．ほかに，Myrinet 固有の問題として，送信データをいったんアダプタ内の SRAM に DMA 転送した後に，あらためて SRAM から通信路に DMA 転送しなければならず，遅延増大の要因となりうる．PM では即時送信によりこの問題の影響を軽減しているが[12]，Myrinet のような構成とする限り，このオーバヘッドをなくすことは原理的に不可能である．また，スイッチによる遅延時間は，たとえば cLAN では 0.5 μ 秒程度を要するが[4]，これはスイッチでのルーティングとバッファリングによると考えられる．カットスルー方式としてバッファリングを不要にできれば，0.1 μ 秒程度にまで低減可能であろう．

(3) **受信オーバヘッド** 受信アダプタに着信してから受信アプリケーションが検出するまでの時間であり，受信データのバッファリング，ホストメモリへのデータ転送，アプリケーションによる着信検出が含まれる．BIP，PM，VIA では，アダプタが自動的に受信データをホストメモリに DMA 転送することにより，ホストプロセッサの介入を不要にしており，受信オーバヘッド低減に役立っている．また着信検出での遅延を小さくしたい場合には，割込みではなくポーリングによる検出とするのが一般的である[7),16)]．割込み検出にすると必ず OS を介することになり，ユーザレベル通信のメリットが失われる．受信データの自動転送とポーリングによる着信検出を組み合わせ，さらに受信データのバッファリングを削除すれば，受信オーバヘッドはデータおよびステータスの DMA 転送と着信検出に要する時間にまで削減できる．これは，通常のプロセッサとホストブリッジを使用すれば 1μ 秒程度以下に抑えられるが，見方を変えれば，ホストを特殊な構成としない限りこれ以上の低減は不可能なことを意味する．

(4) **エラー検出** プロセッサ間通信で発生したエラーは，何らかの方法で検出されなければならないが，その検出/通知方法と通信性能にはある程度トレードオフの関係がある．つまり，エラー検出/通知の粒度を高めるためには，通信性能が犠牲になる傾向がある．たとえば PM では，受信終了まで CRC エラーの検出ができないとの理由で，受信データ全体をアダプタ上の SRAM にいったんバッファリングしてからホストメモリに DMA 転送するが[12)]，これは受信オーバヘッドの増大につながる．また，受信側アダプタが送信側に着信応答を返し，送信側ではその応答を待って次のデータ送信を開始する方式では，スループットが犠牲になる．このような方法をとっても，アダプタで検出できるのは通信路とそのインタフェースのエラーまでであり，プロセッサ間通信に関するすべてのエラー（受信側のシステムバスエラーやアプリケーションの暴走などを含む）をアダプタの機能だけで検出/通知するのは困難なことに注意すべきである．

3.3 コ ス ト

アダプタ上にアドレス変換，メモリ管理機能，I/O 制御プロセッサとそのメモリを備える方式では，それがコスト上昇の要因になりうる．また，スイッチの拡張性を高めるには，スイッチ間のルーティングとバッファリングが必要になり，コストの上昇と装置サイズの増大につながる危険性がある．

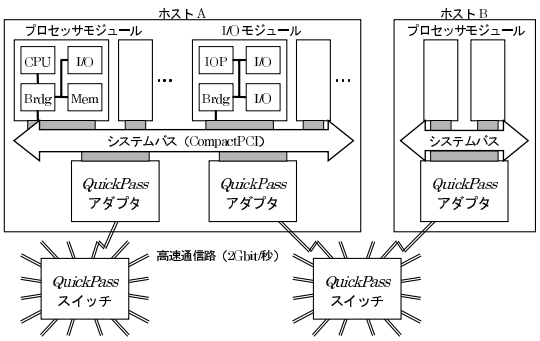

図 1 システム構成図
Fig. 1 System block diagram.

4. *QuickPass* の提案

4.1 基本コンセプト

3 章で示した課題分析をもとに，システム性能の向上と低コスト化を目指して，組み込み用途の並列処理システムに最適化したプロセッサ間通信機構「*QuickPass*」を開発する．想定するシステム構成を**図 1** に示す．

方式検討にあたっては様々なトレードオフが生じるが，以下に示す優先順位とする．

(1) **保護機能の実現** 高信頼化のためには必須
(2) **通信遅延の低減** システム性能向上に必要
(3) **小型&低コスト** 製品化の条件
(4) **汎用性と拡張性** システム設計でカバー可能

まず保護機能であるが，組み込み分野でもシステム構成はますます複雑化しており，信頼性とトラブル解析性を確保するためには必須と考える．さらに，システム性能の向上には通信遅延を低減する必要があるが，3.2 節で述べたように受信オーバヘッド削減には自ずと限界があるため，本提案では送信オーバヘッドと転送処理時間の低減にフォーカスしたい．また，組み込み用途の製品に適用するためには，小型化と低コスト化が必要なのは当然であり，アダプタとスイッチをシンプルな構成にできるよう考慮する．

一方，汎用性と拡張性に関しては，組み込みシステムに特化することで，ある程度の制限は許容可能と考える．我々がターゲットとする組み込みシステムでは，専用アプリケーションの機能/負荷分散が想定されるため，市販アプリケーションの処理性能向上を目的とするような汎用的な仕組みを導入することの優先度は低い．また，通信路も独自に定義できるため，汎用の LAN/SAN に求められる通信プロトコルや拡張性に対する優先度も低い．

以上を基本コンセプトとして開発した *QuickPass*

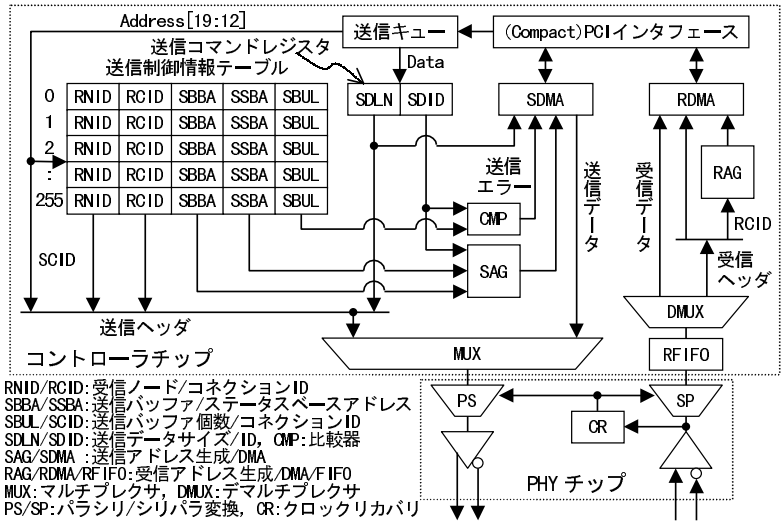

図 2 アダプタ構成図
Fig. 2 Adapter block diagram.

について以下に説明する．

4.2 構成と機能

QuickPass アダプタの構成を**図 2** に示す．下記のうち (1)〜(5) が本稿で提案する *QuickPass* のアーキテクチャにかかわる部分であり，その他は実装に依存する部分である．

(1) **送信制御情報テーブル** 送信制御とヘッダ生成に必要な，コネクション固有の静的な制御情報である送信バッファとステータスのベースアドレス（SBBA/SSBA），バッファ個数（SBUL），受信ノード ID とコネクション ID（RNID/RCID）を格納する．

(2) **送信コマンドレジスタ** メッセージごとに変化する動的な制御情報である送信データ ID（SDID）とデータサイズ（SDLN）を連結した 1 ワードを設定するレジスタで，ホストプロセッサからここへライトすることによりアダプタが送信処理を開始する．

(3) **送信パラメータ生成** 送信コマンドレジスタへのライトアドレスを送信コネクション ID（SCID）として使い送信制御情報テーブルを参照することにより，コネクション対応の制御情報を選択し制御パラメータを生成する．送信アドレス生成回路（SAG）において，送信バッファおよびステータスのベースアドレスにデータ ID をオフセットとして加えて，送信データとステータスのアドレスを生成する．このとき，比較器（CMP）では送信データ ID とバッファ個数を比較し，前者が大きい場合にはエラーと見なす．

(4) **送信制御** 送信 DMA（SDMA）は，上述の送信データアドレスから送信データを DMA リードして PHY（トランシーバ）に与えることにより通信路に送出する．送信処理が完了すると，送信ステータスアドレスに終了ステータスを DMA ライトする．

(5) **受信制御** 受信ヘッダとデータは，受信 DMA（RDMA）によりホストメモリへ DMA ライトされる．このとき，受信アドレス生成回路（RAG）がヘッダ中の受信コネクション ID（RCID）を用いて，受信データおよびステータスのアドレスを生成する．受信処理が完了すると，受信ステータスアドレスに終了ステータスを DMA ライトする．受信アプリケーションは，このステータスをポーリングして着信を検出する．

(6) **通信路** PHY がパラレルデータをシリアルデータに変換する（受信では逆）．通信路は伝送速度 2 Gbit/秒のシリアルバスで，受信側にクロックリカバリ回路（CR）を設け，受信信号からクロックを抽出して同期化制御しており，クロックの専用線は不要である．また，送信側からデータにその有効/無効を示す制御情報を重畳して伝送するので，ホストメモリからの送信データリードが間に合わない場合には，通信路にウエイト挿入が可能である．受信側からは送信側にフロー制御情報を送ることにより，受信 FIFO（RFIFO）オーバフローを防止できる．活線挿抜にも対応する．

(7) **スイッチ** 図 1 に示したように各アダプタの通信路はスイッチに接続される．スイッチは 16 ポートのクロスバースイッチであり，ノンブロッキング通信が可能である．また，カットスルー方式によってデータバッファリングなしに，ヘッダ中の受信ノード ID に対応するポートへ転送する．マルチキャスト機能も備える．

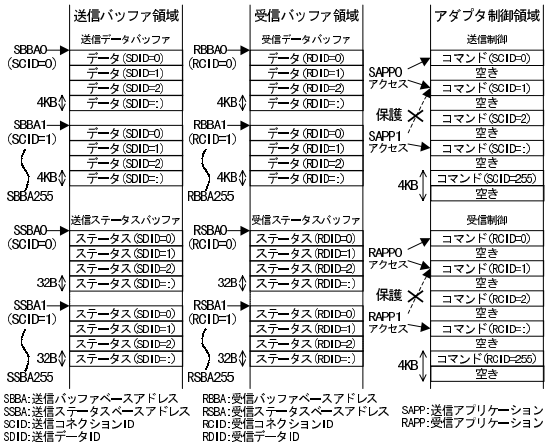

図 3 アドレスマップ
Fig. 3 Address map.

(8) **ホスト インタフェース** 4倍速(64 bit@66 MHz)までの ComapctPCI に対応しており，最大 500 MB/秒程度の転送速度を有する．これは，通信路の伝送速度 2 Gbit/秒に対してボトルネックにならない十分な性能である．アダプタは，32/64 bit と 33/66 MHz の任意の組合せの ComapctPCI 上で動作可能である．

4.3 アドレス割り付け

アドレスマップ（物理空間）の例を**図 3** に示す．

(1) **送受信バッファ領域** データバッファおよびステータスバッファ領域はホストメモリ上に割り付けられ，アダプタからの DMA 対象とするためピンダウン（物理メモリ上に常駐）される．データバッファ領域は最大 256 のコネクション対応に分割されており，各々は最大 8 K 個のページサイズ☆に整合するバッファ（コネクションあたり最大 32 MB）からなる．個々の送受信データはバッファ境界の先頭から置かれ，サイズはページサイズ以下の任意長である（ただしワード単位）．ステータスバッファ領域もコネクション対応に分割され，各々はバッファ個数のキャッシュラインサイズ☆☆に整合するステータスワードからなる．コネクションに対応するバッファおよびステータスのベースアドレス（物理）とその個数は，あらかじめ OS によりアダプタ内の制御情報テーブルに設定される．

(2) **アダプタ制御領域** コマンドレジスタは，各コネクション対応にページサイズ整合のアドレス空間を複数持ち，ホストの物理空間にマッピングされている．OS は，各アドレスをそれに対応するコネクションを使うアプリケーション（SAPP/RAPP）に割り当てる．アダプタでは，コマンドをライトされたアドレスにより送信コネクション ID（SCID）を特定する．

☆ ページサイズとして 4 KB を仮定．
☆☆ キャッシュラインサイズとして 32 B を仮定．

5. *QuickPass* の特徴

5.1 保護機能の実現

(1) **資源保護** OS は各コマンドレジスタアドレスを対応するアプリケーションに割り当てることにより，通常のメモリ管理機構を使ってコネクション間の保護を実現する．各アプリケーションは，コネクション設定時にコマンドレジスタアドレスに対するアクセス権を獲得しておけば，その後は直接アクセスできユーザレベル通信が可能となる．制御情報テーブルの設定値は OS で管理され，アダプタでデータ ID の正当性もチェックされるので，アプリケーションの設定ミスによる他アプリケーションまたはシステムの資源への不正なアクセスも防止できる．

(2) **コンテキスト保護** 送信制御にはディスクリプタが不要で，1 ワードのコマンドライト（アトミック転送）のみで制御可能である．よって，従来はアダプタ制御でのコンテキスト保護のために必要であった，PM におけるギャングスケジューリングや，VIA におけるスレッド間の排他制御といった，通信遅延性能劣化の要因となる処理を排除できる．

(3) **ホストへの条件** このような保護機能実現ためには，OS のカーネルに特別な仕組みを必要としないため（デバイスドライバと通信ライブラリの対応は必要），業界で実績のある OS をそのまま利用することが可能である．たとえば，POSIX 準拠のリアルタイム OS である LynxOS [19] や，ハードリアルタイムが不要な用途には Linux などと組み合わせることを想定している．また，図 1 に示すような疎結合マルチプロセッサ構成のホストにも適用可能である．以上のように，ホストに対して特別な条件を要求しない．

5.2 通信遅延の低減

(1) **送信遅延の最小化** 他方式でも効果が実証されているユーザレベル通信とゼロコピー転送を採用する．さらに，通信制御に必要な情報を静的なものと動的なものに分け，前者をあらかじめ OS が制御情報テーブルに設定しておくことにより，通信のたびに設定が必要な後者を最小化してディスクリプタを不要にした．これにより制御情報の生成と設定に要する時間を削減でき，さらに制御情報はすべて物理アドレスベースとして単純なオフセット加算だけとしたため，アドレス生成に要する時間も最小化できた．

(2) **転送処理時間の最小化** 通信路に対してウエイトの挿入を可能としたことで，アンダーフローへの配

慮が不要になったため，ホストメモリからの送信データリードと並行して，そのデータの到着を待つことなく，通信路に対してヘッダを送出してスイッチの経路設定を行うことができる．また，ホストメモリからリードされた送信データは，バッファリングなしに即座に送信処理され，さらにスイッチおよび受信側アダプタでもバッファリングされないので，転送処理時間が低減される．なお，通信路ではウエイト挿入を可能とするための制御情報をデータに重畳するが，これによるスループットの劣化はわずか数%☆である．この程度のスループット劣化は，上述の転送処理時間低減の効果に比較すれば，許容できるものと考える．

5.3 小型&低コスト

QuickPass アダプタは，各種通信制御を行うコントローラと，通信路との物理層インタフェースであるPHY のわずか 2 チップで構成される．他方式で必要なデータバッファ，I/O 制御プロセッサとそのメモリ，アドレス変換テーブルは不要である．制御情報テーブルのサイズは，32 Kbit☆☆であり，これを含めてコントローラの全機能は一般的な FPGA または ASIC で実現できる規模である．

また，通信路に関しては，信号線は小振幅差動のシリアル信号 × 2 対（送信と受信）の計 4 本だけであり，スイッチのピン数削減に寄与する．ComapctPCI の空きピンを利用してバックプレーンで接続することも可能で，この場合ケーブルも不要になる．スイッチも 1 チップ構成であり小型化と低コスト化に貢献する．

5.4 汎用性と拡張性

QuickPass は以上のような特長を備える一方で，汎用性と拡張性の面では多少の制限が生じるが，4.1 節に示した基本コンセプトには矛盾しない．

(1) **バッファ割付け** 送受信データおよびステータスバッファは，それぞれ物理アドレスで連続な領域にピンダウンしておく必要がある．アプリケーションから OS（デバイスドライバ）に対してバッファ領域を要求するインタフェースにおいて，この機能が必要となる．*QuickPass* で想定する一般的なマルチタスク OS を使う場合，各種アプリケーションが起動された後に，物理アドレスで連続な領域を動的に確保することは困難である．したがって，アプリケーションの動的な起動/停止を前提とするような汎用的な用途には，*QuickPass* を適用するのに限界がある．しかし，我々がターゲットとする組み込み用途では，処理内容を予測できることと，リアルタイム性能が重視されることから，潜在的に必要な最大限のバッファをシステム立ち上げ時に確保するので問題にならない．必要性が不確定な領域を予約することになるが，二次記憶を搭載しない組み込みシステムでは，いずれにしろ潜在的に必要な最大限の物理メモリを実装しておく必要があることに変わりはない．物理メモリ容量の問題は，昨今の DRAM チップの大容量化と低価格化により解決されるものと考える．

(2) **通信エラー検出** 受信側から送信側に受信確認の応答を返す機能は，処理を複雑化するのでアダプタには実装しない．コリジョンがなくメッセージロスも発生しない専用通信路であるのと，通信路でのエラー（リンクエラーと CRC エラー）はアダプタで検出してそれを搭載するホストに通知するので，受信側アダプタが送信側に応答を返す必要はない．また，仮に TCP/IP などの通信プロトコルを使ったとしても，受信側アプリケーションまで正常に到達したことを確認することはできず，完璧な着信確認のためにはアプリケーションで明示的に応答するしか術はない．アプリケーションで応答することによりエラー検出/通知の目的は達成できるので，性能劣化につながる応答機能をアダプタに実装するのは得策でない．

(3) **ノード数** 現状ではスイッチチップの限界により，ドメインを構成するノード（ホスト）数は最大 16 に限定される．ただし，ホスト内をマルチプロセッサ構成とすれば，ドメインを数十プロセッサで構成することは可能であり，ホストに複数のアダプタを搭載すれば複数ドメインと通信することもできる．また，組み込み用途ではシステム設計の面から，機能または負荷分散の範囲を局所化することでノード数をある程度限定できると考える．スイッチを多段化してその間のルーティング機能を付加すれば，ノード数を増やすことは不可能でないが，ルーティングによる通信遅延とコスト増大の危険性がある．急速な半導体技術の進歩を考えると，1，2 年後には 2，3 倍の容量のスイッチチップが実現できると予想され，現状でのスイッチ多段化への対応は得策でないと判断した．

6. 有効性の評価

6.1 通信遅延の見積り

4 章と 5 章に示した *QuickPass* について，短いメッセージ（16 B）通信での通信遅延を**図 4** のように見積もる．見積り条件として，通信路は 2 Gbit/秒，ホストインタフェースは 32 bit@66 MHz の CompactPCI

☆ データ 32 bit あたり制御情報を 2 bit 付加する場合の劣化は制御情報 2 bit ÷ データ 32 bit × 100=6%.

☆☆ ページサイズ 4 KB，物理アドレス 32 bit の場合，256 コネクション × 2 方向 × 制御情報 64 bit=32 Kbit.

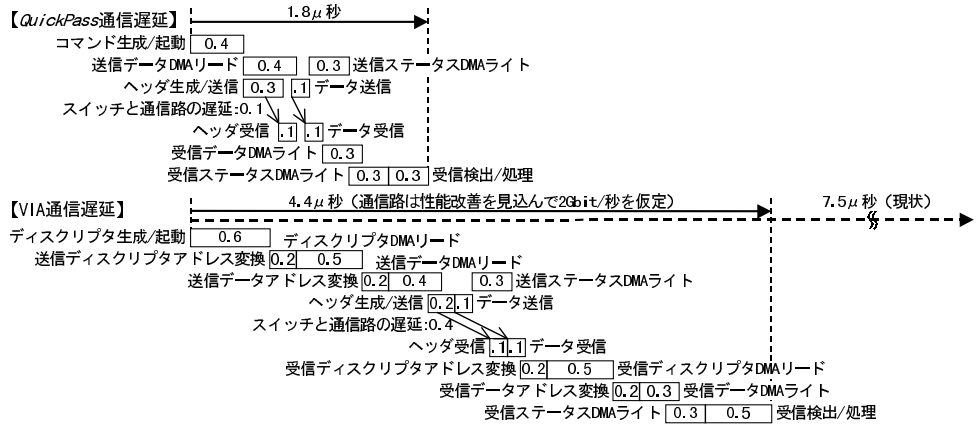

図 4 通信遅延の見積り
Fig. 4 Latency estimation.

表 2 通信遅延の内訳
Table 2 Details of latency.

処理内容	内訳	種別	遅延
コマンド生成/起動	Lib Call，コマンド生成	SW*1	0.1
	コマンド転送（1 ワード）	Bus*2	0.2
	アダプタ取り込み/起動	HW*3	0.1
送信データ DMA リード	データ転送	Bus	0.1
	メモリ制御，アクセス時間	HW	0.2
	アダプタ取り込み	HW	0.1
ヘッダ生成/送信（経路設定を含む）データ送信	ヘッダ生成と送信	HW	(0.1)*5
	通信経路設定	Line*4	(0.2)*5
	データ送信	HW	0.1
スイッチと通信路	スイッチと通信路の伝搬遅延	Line	(0.1)*5
ヘッダ受信 データ受信	受信 FIFO への取り込み	HW	(0.1)*5
	受信 FIFO への取り込み	HW	0.1
受信データ DMA ライト	データ転送	Bus	0.1
	メモリ制御，アクセス時間	HW	0.2
受信ステータス DMA ライト 受信検出/処理	ステータス転送	Bus	(0.1)*6
	メモリ制御，アクセス時間	HW	0.2
	ポーリング，受信処理	SW	0.1
	メモリ制御，アクセス時間	HW	0.2
片道遅延時間の合計			1.8μ 秒

*1 ソフトウェアによる処理時間．*2 バス転送による時間．
*3 ハードウェアによる時間．*4 通信路/スイッチによる時間．
*5 送信データ DMA リードと並列に処理．
*6 受信データ DMA ライトと並列に処理．

（250 MB/秒），ホストプロセッサ性能は 500 MIPS 程度，ハードウェアの動作は最適化されていることを仮定する．ハードウェアで想定されるステート数，バスサイクル数，ソフトウェア処理のステップ数から求めた通信遅延の内訳を**表 2** に示す．

QuickPass における通信のフローを以下に示す．

（1） **コマンド生成/起動** 送信アプリケーションは通信ライブラリをコールすることにより，送信コマンドを生成してコマンドレジスタにライト（1 ワード）し，アダプタに送信起動をかける．アダプタはコマンドを受けると，そのコマンドとあらかじめ OS により設定された送信制御情報テーブルの内容から送信制御に必要な各パラメータを生成する．

（2） **送信データ DMA リード** 次に，生成した送信データアドレスから送信データを DMA リードする．ホストブリッジはメモリから送信データをリードして，アダプタがそれを取り込む．

（3） **ヘッダ生成/送信とデータ送信** 送信データ DMA リードと並行して，ヘッダを生成して通信路に送出すると，スイッチでは受信ノードへの経路を設定する．その後，DMA リードしたデータから順次通信路に送出する．

（4） **スイッチと通信路の遅延** スイッチではヘッダおよびデータをバッファリングすることなく，受信ノードに転送する．送信ノードから受信ノードへの経路はあらかじめ設定されているので，データはワイヤスピードで転送される．

（5） **ヘッダ受信とデータ受信** 受信アダプタでは通信路からヘッダとデータを受信して受信 FIFO（RFIFO）に入れる．

（6） **受信データ DMA ライト** 受信アダプタでは，受信コネクション ID（RCID）より受信データアドレスを生成して，受信データを DMA ライトする．ホストブリッジは受信データをメモリにライトする．

（7） **受信ステータス DMA ライトと受信検出/処理** 受信ステータスを DMA ライトする．受信アプリケーションは，受信ステータスをポーリングすることによりメッセージ受信を検出して処理する．

6.2 優 位 性

QuickPass での通信遅延としては図 4 に示すように，1.8 μ 秒という見積り結果を得た（アプリケーション間の片道通信遅延）．これは，2.5 節に示したように，現状の保護機能を備えた通信方式の最高性能が 7.5 μ 秒程度であることに比較すると，約 4 倍の性能にあ

たる．

この一連の処理では，5.2 節で示したように，ディスクリプタが不要であり，送信起動処理とディスクリプタリードに必要な時間が削減される．さらに，アドレス変換テーブルの参照が不要で，またアダプタおよびスイッチでのバッファリングによる遅延もないことが特長である．

6.3 他方式との比較

VIA について見積もった通信遅延も図 4 に示す．通信路は cLAN 相当とし，伝送速度は *QuickPass* と同等の 2 Gbit/秒（つまり現状の cLAN の約 2 倍）を仮定した．この性能改善を見込んだ VIA/cLAN の通信遅延 4.4 μ 秒と比較しても，*QuickPass* は 2 倍以上の性能であり，優位性は明らかである．この見積りから，VIA/cLAN 相当では，以下に示すような遅延増大要因のあることが分かる．

(1) **ディスクリプタによる遅延** VIA では，ディスクリプタを使って通信を制御するため，その生成とスレッド間の排他制御が必要で起動に時間がかかる．また，アダプタが制御情報を獲得するには，ディスクリプタの物理アドレス生成と DMA リード（50 B 程度）が，送信および受信の両方で必要になる．

(2) **アドレス変換による遅延** アダプタ上のアドレス変換テーブルを参照して物理アドレスを生成するが，このエントリはデータおよびディスクリプタのメモリページごとに必要で，100 Mbit 程度☆の容量となり，通常は外付けの DRAM チップで実現されるため，物理アドレスの生成に時間がかかる☆☆．

(3) **逐次処理による遅延** DMA リードで得られたディスクリプタを解釈してはじめて，送信データを特定できる．したがって，ディスクリプタ DMA リードと送信データアドレス変換および送信データ DMA リードを並列に処理することができないため，遅延時間が増大する．

(4) **バッファリングによる遅延** メッセージ送信の途中で通信路にウエイトを挿入できないと，送信データをいったんバッファリングしてから通信路に送信する必要がある．またスイッチにあらかじめ転送パスが設定されない場合，ヘッダ中の宛先から受信ノードを判断して切替え制御を行うため，バッファリングと切替え制御の遅延が発生する．ただし，この遅延 0.4 μ 秒は，通信路とスイッチの実装に依存する部分である．

なお，VIA を比較対象にした理由は，汎用並列処理システムでは今後業界標準になる可能性があり，かつ，現状の低遅延通信では唯一完全な保護機能を備えるからである．3.1 節で示したように，BIP では保護機能がサポートされておらず，PM では保護機能のリアルタイム性に課題があり，どちらも我々の要求条件を満たさないと判断して，比較対象から外した．

6.4 伝送速度とデータサイズ依存性

次に，通信路の伝送速度と通信遅延の関係を見積もった結果を**図 5** に示す．アダプタとスイッチによる遅延をゼロと仮定した理想的なアダプタとスイッチによる通信遅延もあわせて示す．*QuickPass* では，今後通信路が 10 Gbit/秒まで高速化すると，通信遅延を約 1 μ 秒まで低減可能なことが分かる．一方，VIA のようにディスクリプタと大容量のアドレス変換テーブルの参照を必要とする方式では，そのオーバヘッドにより，現状の構成とする限り通信遅延を 2 μ 秒以下にすることは困難であろう．

また，データサイズと通信遅延の関係を見積もった結果を**図 6** に示す．データサイズが 1 KB 程度以下の

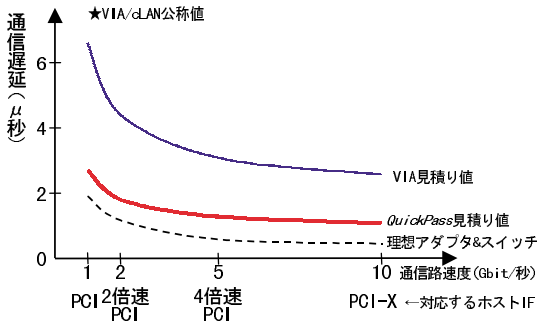

図 5 伝送速度と通信遅延の関係
Fig. 5 Latency vs. transfer speed.

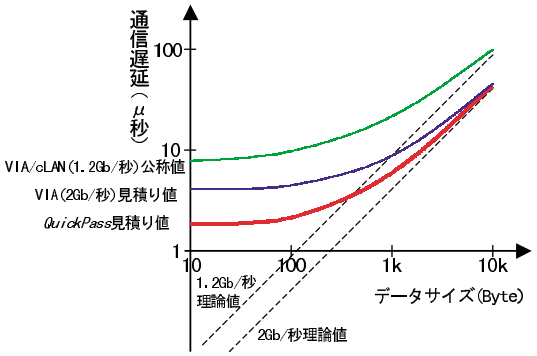

図 6 データサイズと通信遅延の関係
Fig. 6 Latency vs. data size.

☆ *QuickPass* と同等条件で，256 コネクション × 2 方向 × 8 K エントリ × (32 − 12) = 80 Mbit．ほかに各エントリに保護機能用のビットが必要．

☆☆ この遅延を削減するために TLB を持つなどの対策も考えられるが，ヒット率を高めるには構成が複雑になる．

比較的短いメッセージ通信において，*QuickPass* の優位性は顕著である．リアルタイム性が要求される並列処理システムにおいては，処理粒度を小さくすることが望ましく，結果としてデータサイズも一般に小さくなる傾向にあるため，*QuickPass* におけるこの特性は，適していると考えられる．

1 KB 程度以上のデータサイズでは他方式との差が縮まり，ほぼ理論値に収束することが分かる．データサイズが大きくなると，通信遅延における処理オーバヘッドの占める割合が減るためと考えられる．

7. まとめと今後の課題

今後の組み込みシステムに要求される処理性能に対応することを目的に，高性能な並列処理システムを構築するための技術として，低遅延のプロセッサ間通信機構「*QuickPass*」を提案した．*QuickPass* は以下にまとめた特徴を備えることにより，保護機能と低コストを実現したうえで，アプリケーション間の片道通信遅延として 1.8 μ 秒と，他方式の約 4 倍（他方式における通信路の性能改善を見込んでも 2 倍以上）の性能になることを示し，優位性を明らかにした．

(1) **ユーザレベル通信と保護** 複数のコネクションそれぞれにコマンドレジスタアドレスを対応させ，ユーザレベルで直接制御可能としながら，同時に保護機能も実現した．

(2) **制御パラメータ生成** アダプタに制御情報テーブルを備え，送信起動と同時に制御パラメータを生成する．ディスクリプタとアドレス変換は不要である．

(3) **コピー/バッファリング撤廃** 送信から受信に至るまでの経路全体（ライブラリ，アダプタ，スイッチ）で，データコピーとバッファリングを撤廃した．

(4) **組み込み用途に最適化** 汎用性/拡張性よりも低遅延/低コストを重視したシンプルなインタフェースとした．

今後は，実機へのインプリメントを行った後，通信性能の評価を行って本提案の有効性を実証したい．ノード数の拡張性（スイッチ容量増強），通信路およびホストインタフェースの高速化，光ファイバによる長距離伝送への対応が今後の課題である．また，実際の製品への適用に向けては，MPI[20] などのアプリケーションインタフェースの整備と並列処理方式の開発が必要である．

参 考 文 献

1) PCI Industrial Computer Manufacturers Group: *PICMG 2.0 CompactPCI Specification Revision 3.0* (1999).
2) 森眞一郎，富田眞治：並列計算機アーキテクトからみた計算機クラスタ，情報処理，Vol.39, No.11, pp.1073–1077 (1998).
3) Boden, N.J., Cohen, D., Felderman, R.E., Kulawik, A.E., Seitz, C.L., Seizovic, J.N. and Su, W.-K.: Myrinet – A Gigabit-per-Second Local-Aread Network, *IEEE Micro*, Vol.15, No.1, pp.29–36 (1995).
4) Giganet, Inc.: *cLAN for Linux Data Sheet* (1999).
5) 工藤知宏，山本淳二，建部修見，佐藤三久，西宏章，天野英晴，石川 裕：PC 間ネットワークによる共有アドレス空間を持つ並列処理システム，*Hokke '99*, pp.121–126 (1999).
6) PCI Special Interest Group: *PCI Local Bus Specification Revision 2.2* (1998).
7) Araki, S., Bilas, A., Dubnicki, C., Edler, J., Konishi, K. and Philbin, J.: User-Space Communication: A Quantitative Study, *Supercomputing '98*, Orlando, USA (1998).
8) Bhoedjang, R.A., Ruhl, T. and Bal, H.E.: User-Level Network Interface Protocols, *IEEE Computer*, Vol.31, No.11, pp.53–60 (1998).
9) von Eicken, T., Culler, D.E., Goldstein, S.C. and Schauser, K.E.: Active Messages: A Mechanism for Integrated Communication and Computation, *19th International Symposium on Computer Architecture*, Gold Coast, Australia, pp.256–266 (1992).
10) Prylli, L. and Tourancheau, B.: BIP: A new protocol designed for high performance networking on Myrinet, *PC-NOW Workshop*, Orlando, USA (1998).
11) Pakin, S., Lauria, M. and Chien, A.: High Performance Messaging on Workstations: Illinois Fast Messages (FM) for Myrinet, *Supercomputing '95*, San Diego, USA (1995).
12) 手塚宏史，堀 敦史，石川 裕：ワークステーションクラスタ用通信ライブラリ PM の設計と実装，*JSPP '96*, pp.41–48 (1996).
13) Dubnicki, C., Bilas, A., Chen, Y., Damianakis, S. and Li, K.: VMMC-2: Efficient Support for Reliable, Connection-Oriented Communication, *Hot Interconnects Symposium V*, Stanford, USA (1997).
14) von Eicken, T., Basu, A., Buch, V. and Vogels, W.: U-Net: A User-Level Network Interface for Parallel and Distributed Computing, *15th ACM Symposium on Operating Systems Principles*, Copper Mountain, USA (1995).
15) Buonadonna, P., Geweke, A. and Culler, D.: An Implementation and Analysis of the Virtual Interface Architecture, *Supercomputing*

'98, Orlando, USA (1998).
16) Speight, E., Abdel-Shafi, H. and Bennett, J.K.: Realizing the Performance Potential of the Virtural Interface Architecture, *The International Conference on Supercomputing* (1999).
17) 住元真治，堀　敦史，手塚宏史，原田　浩，高橋俊行，石川　裕：Gigabit Ethernet を用いた高速通信ライブラリの設計と評価，*JSPP '99*, pp.63–70 (1999).
18) 堀　敦史，手塚宏史，石川　裕：ギャングスケジューリングの PC クラスタ上での実装，*SWoPP '97*, pp.79–84 (1997).
19) LynuxWorks, Inc.: *The scalable, reliable and highly deterministic operating system for real-time and embedded applications* (1999).
20) Message Passing Interface Forum: *MPI: A Message-Passing Interface Standard* (1995).

(平成 12 年 4 月 24 日受付)
(平成 12 年 8 月 31 日採録)

飯塚　剛

1987 年東京大学工学部電子工学科卒業．同年三菱電機（株）入社．コンピュータグラフィクス，産業用計算機，通信装置の監視制御技術に関する研究開発に従事．情報技術総合研究所勤務．

佐藤　裕幸（正会員）

1982 年筑波大学第三学群情報学類卒業．同年三菱電機（株）入社．1985〜1989 年（財）新世代コンピュータ技術開発機構に出向し，逐次型および並列型推論マシンのシステムソフトウェアの研究開発に従事．現在，三菱電機（株）情報技術総合研究所にて，並列処理ソフトウェア，最適化システムの研究開発に従事．

落合　真一（正会員）

1988 年慶応義塾大学大学院理工学研究科修士課程修了．同年三菱電機（株）入社．産業用計算機向けリアルタイム OS および実時間通信技術に関する研究開発に従事．情報技術総合研究所勤務．

伊藤　隆弘

1981 年横浜国立大学電気工学科卒業．同年三菱電機（株）入社．産業用グラフィックディスプレイ，装置内インタコネクト，通信装置の監視制御技術に関する研究開発に従事．情報技術総合研究所勤務．

斉藤　成一

1973 年早稲田大学理工学部電子通信学科卒業．同年三菱電機（株）入社．コンピュータの高速化，高信頼化，高耐環境 EMC 設計等の研究開発に従事．2000 年東京農工大学大学院博士後期課程修了．工学博士．現在，三菱電機（株）情報技術総合研究所に勤務．電気学会会員．