

クラスタ向けネットワークアーキテクチャとプロトコルの提案 —*Maestro* ネットワークの開発と性能評価

山 際 伸 一[†] 福 田 宗 弘^{††} 和 田 耕 一^{††}

マイクロプロセッサの劇的な性能の向上にともない、価格/性能比が優れた並列計算システムとしてPCクラスタを利用する傾向が高まっている。しかしながら、従来の汎用ネットワークハードウェアと通信プロトコルで構成されるPCクラスタは、通信オーバヘッドが大きく、内在する性能を引き出すことが困難である。独立したシステムとしてのPCクラスタの特性を考慮すると、通信の最適化が可能である。本論文では、クラスタコンピューティング向けの通信手順最適化として、ネットワークハードウェアのリンク層におけるバースト転送と通信単位の小粒度化による通信の多重化を提案している。これらの高速化技法を実現するリンク制御プロトコル AGC (Adaptive Granularity Control) プロトコルを提案し、リンクレイヤコントローラ MLC (Maestro Link Controller) への実装について述べている。さらに、MLCを用いて構築した*Maestro* ネットワークの通信実験により、本最適化技法が通信性能の向上に有効であることを示している。

Network Architecture and Communication Protocol for Cluster Computers —Development and Performance Evaluation of *Maestro* Network

SHINICHI YAMAGIWA,[†] MUNEHIRO FUKUDA^{††} and KOICHI WADA^{††}

The emergence of high-performance microprocessors has made it attractive to use a PC cluster as a parallel computing system with an excellent cost/performance ratio. Most existing PC clusters have used WAN or LAN-oriented hardware products and protocols due to their market availability. This however loses possibilities of improving the intra-cluster communication, which can be further optimized for the use within a geographically small area. We have achieved such performance optimization from the network hardware point of view. This paper presents two optimization techniques for cluster computing: the burst and pipelined transfer of minimized transfer units, implemented at hardware link layer. We propose a link control protocol realizing these two techniques, called AGC (Adaptive Granularity Control) protocol. This paper describes its implementation in a link control hardware, referred to as MLC (Maestro Link Controller) and demonstrates the efficiency of our proposed optimization techniques through the experiments on the *Maestro* network constructed with MLC.

1. はじめに

PCの価格/性能比の向上にともない、汎用ネットワークで複数のPCを接続したPCクラスタによる、並列・分散コンピューティングに注目が集まっている^{11),12)}。多くの場合、PCクラスタは、広域ネットワーク向けのネットワークハードウェアと通信プロト

コルを用いて構成されている。たとえば、ネットワークハードウェアについてはEthernetが、通信プロトコルについてはTCP/IPが主流となっている。しかし、これらは広域の通信を前提としているため、PCクラスタに適用した場合、通信におけるオーバヘッドが大きく、内在する性能を十分に引き出すことは難しい¹⁷⁾。

性能向上を妨げる要因としては、(1) 通信プロトコルソフトウェア、(2) デバイスハンドラ、および、(3) ネットワークハードウェアがあげられる。高性能を実現するには、PCクラスタの構成上の特性、すなわち、地理的に限定された環境に構築される点を考慮して、これらの最適化を図る必要がある。

† 筑波大学工学研究科

Doctoral Program in Engineering, University of Tsukuba

†† 筑波大学電子・情報工学系

Institute of Information Sciences and Electronics, University of Tsukuba

上記(1)の通信プロトコルソフトウェアに関しては、現在までに、FM¹⁴⁾、PM³⁾、BIP⁵⁾等の研究で性能向上策が提案されている。本研究では、(3)のネットワークハードウェアに特に注目して、クラスタコンピューティング向けのリンク制御プロトコル AGC(Adaptive Granularity Control)プロトコルを設計し、本プロトコルで制御される Maestro ネットワークを構築した。本論文では、AGCプロトコルの設計、および、Maestro ネットワークの構築について論ずる。さらに、Maestro ネットワークの評価結果を示し、AGCプロトコルに取り入れた高速化技法である、リンク層における(1)バースト転送と(2)送信単位の小粒度化による通信の多重化が、それぞれ、送信遅延の短縮と送信手順の多重化によるスループットの増大に関与し、クラスタコンピューティング向けの通信に対して有効であることを示す。

以下、本論文では2章において、PC クラスタにおける通信形態の特徴を述べ、その問題点を指摘する。3章では、それらの問題点の解決方法を提案し、Maestro ネットワークの構築について述べる。4章では、提案する方法の有効性について、実験を用いて議論する。最後に、本論文のまとめと今後の課題について述べる。

2. 従来の通信

2.1 クラスタコンピューティングにおける通信

PC クラスタでの通信形態は、並列処理における情報交換の形態を反映したものとなる。このような通信の特徴として、同期に必要な少量データの頻繁な交換と、行列計算等における大量のデータ転送、があげられる。

プロセッサ間の同期は、少量データの送信・受信の組で行われる。このとき、通信の最小単位が同期に要するデータ量より大きい場合、通信レイテンシの増大を招く。すなわち、PC クラスタにおける通信機能として、同期等で要求される小粒度の通信を低レイテンシで実現できることが求められる。

一方、大量のデータ転送に関しては、広域ネットワークのための通信プロトコルでは転送データが複数個の単位に分けられ、複数回にわたって送受信される。この複数回の送受信操作にともなうオーバヘッドは、スループット低下の一因となっている。並列計算でしばしば必要となる大量のデータ転送に対応して、適切な粒度でのバースト転送が効率良く行えることも、クラスタ向けネットワークに必要とされる機能である。

以上に加えて、ルーティングにおいても、クラスタは広域ネットワークと異なる特徴を持つ。たとえば、

クラスタでは固定的な計算機を対象とし、地理的に限定された範囲でルーティングを行うため、広域ネットワークのルーティングにおける IP (Internet Protocol) アドレスから MAC (Media Access Control) アドレスを導出する過程は不要である。すなわち、ルーティングに対しても、冗長な処理を排して最適化されたクラスタ向けルーティングが必要である。

2.2 従来の通信機能と遅延要因

本節では、PC クラスタにおける通信オーバヘッドの個々の要因について述べる。

(a) 通信プロトコルソフトウェア

通信プロトコルソフトウェアの問題は、アプリケーションプログラムに、PC が受信するメッセージを渡すまでのデータのコピー回数である。コピー回数の増加により、通信レイテンシが増大する。さらに、コピー操作を OS カーネル内で行うと、OS の実行モードをユーザモードからカーネルモードへ移行する必要がある。したがって、粒度の小さい通信を多量に行うと、この実行モード移行のためのオーバヘッドにより通信レイテンシが増大する。

これらの問題に対しては FM¹⁴⁾、PM³⁾、BIP⁵⁾ 等のプロジェクトで性能改善策が提案されている。これらの通信プロトコルソフトウェアでは、アプリケーションプログラムからネットワークハードウェアに送信データが渡されるまで、コピー処理をともなわない 0 コピー通信を行っている。

(b) デバイスハンドラ

PC のデバイスハンドラでの問題として、それによって確保される転送データ用の領域が不連続になることがあげられる。たとえば、デバイスハンドラは、送受信データのための領域として、(1) カーネル空間のセグメント (数百 byte から数 Kbyte)、または、(2) メモリページ (4 Kbyte 固定または 8 Kbyte 固定)、を複数確保する。確保されたセグメント、または、メモリページからなる領域は、仮想アドレス空間では連続であるが、物理アドレス空間で連続している保証はない。したがって、数十 Kbyte の長いメッセージは、物理メモリ上の不連続領域に分散配置される可能性がある。分散配置された場合、PC 上のメモリ領域からネットワークハードウェアに DMA 転送する際に、(1) 仮想アドレスから物理アドレスを求め転送開始アドレスとし、(2) セグメント、またはページの境界までの DMA 転送を起動する、といった一連の操作が繰り返し必要となる。この仮想-物理アドレス変換、および DMA 転送の起動操作のオーバヘッドが、スループットの低下を

招く。

上述のような、分散領域に対する転送にともなうオーバヘッドを削減するには、大きな連続領域を確保できる機能がデバイスハンドラに必要である。

(c) リンク層

(i) フレーミングにともなうオーバヘッド

アプリケーションプログラムから送信を要求したデータを、リンク層でフレーミングする例として Ethernet を考える。Ethernet では、最大転送データ長が 1500 byte に規定され、これに 36 byte のヘッダとフッタが付加される¹³⁾。したがって、送信データの大きさを P_n byte とすると、 $36/(P_n + 36)\%$ がリンク層で認識される情報である。プロセッサ間の同期等に使われる最小データの長さを 4 byte と仮定すると、 $36/(4+36)\% = 90\%$ がヘッダとフッタの情報となる。このように、クラスタ環境の特性を考えると広域ネットワークを前提とした付加情報は冗長で、高性能化のために、クラスタに適したフレーミングを設計しなければならない。

(ii) 粒度の小さいメッセージにともなうオーバヘッド

多くのネットワークハードウェアでは、送受信の際、PCとの間でのデータ転送に DMA 転送が用いられる。しかし、少量のデータを転送する場合、DMA 転送を起動するオーバヘッドの転送全体に占める割合が大きくなる。この問題に対しても、ネットワークハードウェアに少量のデータ転送を高速に行える機能を持たせ、転送データ量によって、DMA 転送と適応的に使い分けることで対応できる。

(iii) non-burst 転送によるオーバヘッド

従来のネットワークハードウェアでは、一度の送信機会に 1 つの送信単位（たとえば、パケット）のみを送信する。このため、小さいデータを複数回送信するときには、それらの総データ量が一度の送信機会で送信できる量であっても、その回数分の送受信操作を繰り返す。通信媒体の利用率を高めるには、ネットワークハードウェアのリンク層が、一度の送信機会で複数の送信単位を一括して送出できることが重要である。

(d) 物理層

マルチキャストにともなうオーバヘッド

従来のマルチキャストでは、すべての PC がメッセージを受け取り、該当しない PC はこれを消去するか、または、該当する複数の PC に対して、

その台数分の通信を繰り返す。前者は、Ethernet のような、キャリアセンシティブなメディアで使用される。しかしながら、該当者以外も受信して、それを消去するオーバヘッドが発生する。後者は、送信側が受信側の数だけ、メッセージをコピーして送らなければならない。このようなオーバヘッドを回避するには、物理層において、メッセージの選択的なコピーを行う機能が必要である。

これらの各層における遅延要因のうち、近年のクラスタコンピューティングに関する研究では (a) 通信プロトコルソフトウェアを中心として行われているものが多い。(b) デバイスハンドラに関しては、Myrinet⁸⁾、U-Net¹⁵⁾ 等において積極的に改善が試みられている。我々は、(c) リンク層、(d) 物理層に注目し、その高速化技法を検討した。次章で、本技法の詳細と、それを使った AGC (Adaptive Granularity Control) プロトコル、および、AGC プロトコルを実装したリンクコントローラ MLC について説明する。

3. クラスタ内通信の高速化と実装

3.1 クラスタ内通信の高速化技法

以下に、前節で考察を行った事柄について性能改善を実現する方法を述べる。

(a) リンク層におけるバースト転送の実現

物理層が、通信媒体へのデータ送信権を獲得するための調停について考える。調停回数が増加すると、通信媒体の利用効率が低下し、高いスループットを得ることが困難となる。そこで、我々はリンク層におけるバースト転送を実現することを考える。すなわち、物理層により取得した送信機会ごとに、リンク層における最小送信単位の整数倍のデータを一括して送信する機能を実現する。本機能により、長いメッセージの送信時間を短縮し、スループットを向上させることが可能となる。以降では、このリンク層におけるバーストをともなう送信操作をネットワークバーストと呼ぶことにする。

(b) リンク層における送信単位の小粒度化による通信の多重化

従来の通信インターフェースでは、リンク層での送受信はメッセージを単位として行われる。この場合、メッセージ全体が PC から通信インターフェースに転送されるまで次の送信処理がブロックされるので、(1) 送信側 PC から通信インターフェースへの転送、(2) 送信側および受信側の通信インターフェース間の通信、(3) 受信側の通信インターフェースから PC への転送が逐次的になる。この様子を図 1 (i) に示す。図 1 での各実線

上の番号は、前述の処理の番号に該当する。

この通信の逐次化を回避するために、我々は、リンク層での送信単位の小粒度化により通信を多重化する。この小さい送信単位のことをパケットと呼ぶことにする。通信インターフェースは、PCから転送されてきた最初のパケットをリンク層へ送信すると同時に、PCから後続のパケットを受け付ける。受信側でも同様に、到着したパケットを順次、PCへと転送する。これにより、PCから通信インターフェースへの転送操作と、リンク層での送信操作、および、受信側でのリンク層からPCへの受信操作を多重化することができる。図1(ii)は、この高速化技法を用いた場合の時間の短縮を表している。すなわち、通信が多重化され、通信スループットの向上が期待できる。

3.2 Maestro ネットワークの実装

本節では、前述した高速化技法を適用したMaestro ネットワークの全体構成とその動作概要について述べ、AGC (Adaptive Granularity Control) プロトコルの詳細と、それを実現したリンクレイヤコントローラ MLC (Maestro Link Controller) について論ずる。

3.2.1 全体構成

図2に示すように、Maestro ネットワークは、各PCにPCIバス⁹⁾を介して接続される通信インターフェース、および、IEEE1394⁴⁾を介して各通信インターフェースからのメッセージを受信し、転送を行うスイッチボックス

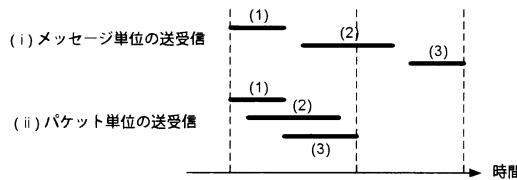


図1 送信単位の小粒度化による通信の多重化
Fig. 1 Conventional vs. pipelined transfer.

スから構成される。以降、特に断らない限り、Maestro ネットワークの通信インターフェースとスイッチボックスを、それぞれ、NI (Network Interface) および、SB (Switch Box) と略す。NIから送信されるメッセージは、図3に示すフォーマットに従って、ヘッダと单一または複数のパケットに変換される。単一パケットを有するメッセージをType0メッセージ、複数パケットを有するメッセージをType1メッセージとして区別する。ネットワークバーストでは、1個以上のパケットが一括転送される。メッセージヘッダはルーティングのための情報を保持する領域で、SBによって解釈される。図中のPktCounterはメッセージを構成するパケット数を保持する領域である。

(1) NIの構成

NIはDMAコントローラを内蔵したPCIインターフェース¹⁰⁾、マイクロプロセッサPowerPC603e (200MHz)⁷⁾と64MbyteのEDO DRAMを搭載したNIマネージャ、送信、および、受信用のネットワークバッファ、AGCプロトコルを実装したMLC、および、200Mbps IEEE1394物理層からなる。

このうち、PCIインターフェースは、NIがPCと通信するための仲立ちを担うだけでなく、内蔵のDMAコントローラを使って、PCのメモリとNIマネージャの

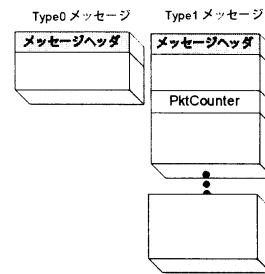


図3 SBで認識可能なメッセージフォーマット
Fig. 3 Message formats handled by SB.

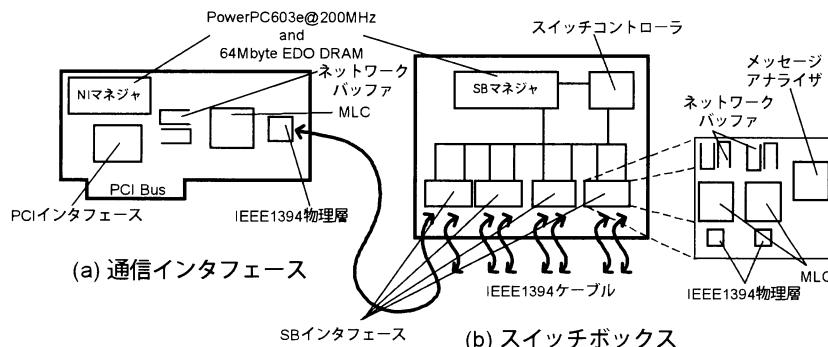


図2 Maestro ネットワーク
Fig. 2 Maestro network.

DRAM、および、ネットワークバッファとの間のデータ転送を行う機能を持つ。DMA 設定のオーバヘッドが顕著に表れる程度の小さなデータを転送する場合は、NI マネージャ内の PowerPC が転送を代行する。さらに、NI では、NI マネージャ内の DRAM を、PCI インタフェースを介して PC から直接アクセスできるように設定できる。

MLCには、ネットワークのフロー制御と IEEE1394 物理層への物理的な転送手順の実現を行う AGC プロトコルが実装される。AGC プロトコルは、通信機会の公平化とネットワークバースト、および、メッセージよりも短いパケット単位での送信操作による通信の多重化を実現する。このプロトコルと MLC の実装についての説明は次項で行う。

(2) SB の構成

SB は、PowerPC603e (200 MHz) と 64 Mbyte の EDO DRAM を搭載した SB マネージャ、DMA コントローラを内蔵したスイッチコントローラ、異なる 2 つの NI からの通信を受理する SB インタフェース 4 対から構成される。

各 SB インタフェースは、1 つのメッセージアナライザ、2 対の MLC、それに付随するネットワークバッファ、および、IEEE1394 物理層を搭載している。この MLC と IEEE1394 物理層については、NI に搭載したものと同一のものを用いる。メッセージアナライザは、NI から転送されるメッセージが Type0、Type1 のどちらであるかを解析する。この解析のために、メッセージアナライザは、メッセージヘッダ部分のみをその内部 FIFO バッファに保存する。この内部 FIFO バッファは、ネットワークバッファに格納可能なメッセージ数分のヘッダを保存できる領域を持っているため、MLC との間でのフロー制御を行う必要はない。

SB マネージャは、メッセージアナライザからのメッセージヘッダを解析し、メッセージの転送先を決定する。複数の NI から SB にメッセージが転送されてきた場合、それらのメッセージヘッダ部分が、それぞれの MLC に対応するメッセージアナライザによって抽出される。SB マネージャは、これら 4 つのメッセージアナライザがメッセージのヘッダ部分を抽出したかどうかをラウンドロビンで巡回しながら調べる。SB マネージャは、ヘッダからメッセージの転送先を決定し、スイッチコントローラに転送を要求する。

スイッチコントローラは、転送要求を受け取り、異なる SB インタフェース上のネットワークバッファ間で 400 Mbps の転送速度で DMA 転送を行う。図 2 (b) に示すように、4 つの SB インタフェース間はバス結合

されており、スイッチコントローラは、同時に複数の SB インタフェース上のネットワークバッファにメッセージを転送することができる。このスイッチ機構により、前述の従来のネットワークにおける問題点のうち、マルチキャストにともなう遅延要因を削減できる。さらに、スイッチコントローラは、SB マネージャで作成されたメッセージを上述のバスを用いて NI に送信する機能を持つ。この機能により、高速な NI 間の同期を実現する。

現在、8 台の PC の接続を対象としたシステムを稼働させている。PC 台数を増加させる場合についての対応策としては、SB を木構造に接続したネットワークトポジをを考えている。このときのデッドロックについては、NI、SB におけるネットワークバッファの送信・受信それぞれのチャネルが分離されているので発生しない。

3.2.2 Adaptive Granularity Control プロトコル

IEEE1394 のような半二重通信媒体では、その双方が同時に送信を行うことはできない。そこで、我々は、半二重通信媒体にも対処できるクラスタコンピューティング向けの新しい通信プロトコル AGC (Adaptive Granularity Control) プロトコルを提案する。

AGC プロトコルは point-to-point の通信路を規定し、図 4 に示すような入出力を想定している。プロトコルを実装するモジュールは、通信媒体の物理層とネットワークバッファに接続される。ネットワークバッファは、FIFO バッファとして構成され、通信クライアント側からメッセージを単位として書き込み・読み出しが行われる。一方、通信媒体の物理層側からはメッセージより小さなパケット単位での送受信が行われる。

AGC プロトコルの特徴は、(1) パケット単位で行われる通信手順、(2) ネットワークバーストを可能にするフロー制御、(3) 公平な送信権獲得制御による双方向通信、(4) 複数チャネルの実現、の 4 点である。

(1) パケット単位で行われる通信手順

AGC プロトコルでは、送信操作をパケットを単位として行う。これにより、前述の高速化技法のうち、リンク層で送信単位を小粒度化することによる通信の多重化を実現できる。

(2) ネットワークバーストを可能にするフロー制御

AGC プロトコルでは、受信側のバッファの空き容量を表す情報を送信側に知らせておくことにより、ネットワークバーストを行う。ネットワークバースト時に一括送出する単位をフレームと呼ぶ。フレームは 0 個以上のパケットからなる。受信側バッファ

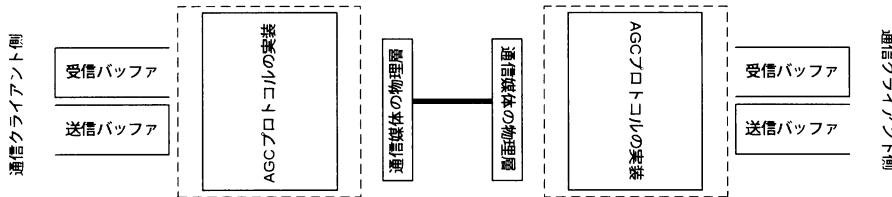


図 4 AGC プロトコル周辺の構成
Fig. 4 Hardware configuration supported for AGC protocol.

の空き容量情報を Credit と呼ぶ。Credit は受信可能なパケット数で表され、送信ごとに、リンクの他端に通知される。

(3) 公平な送信権獲得制御による双方向通信

半二重通信媒体では、一方のリンク端に送信権が集中することがあり、送信権を得られないリンク端の送信操作がブロックされることがある。このように、片方のリンク端が連続的に送信権を獲得すると、もう一方からの通信レイテンシは増大する。AGC プロトコルではこのような状況を回避するために、送信権獲得を完全に公平に制御する。リンク端は送信を行った後、必ず受信を行う。しかし、この規則は、送信と受信の組が成立しないとデッドロックを引き起こす。これに対処するために、各送信機会に返信すべきパケットがない場合には、パケットを含まない空フレームを返信する。このフレームのことをヌルフレームと呼ぶ。一方、パケットを含むフレームのことをデータフレームと呼ぶ。

(4) 複数チャネルの実現

ネットワークバッファは、複数のチャネルで構成され、チャネルごとに送信のための優先順位を設ける。送信機会ごとにたかだか 1 つのチャネルに書き込まれた、パケットが送信の対象となる。

以上の特徴において、(1) は通信レイテンシを低減し、(2) は通信媒体の使用効率を上げる効果があり、(3) は通信媒体の送信権のリンク両端へ均等な配分を行う。また、(4) については、通信媒体の物理的な機能追加をすることなく、通信の多重化を可能とする。

3.2.3 Maestro Link Controller の実装

MLC (Maestro Link Controller) は、AGC プロトコルを実装し、ネットワークバッファのフロー制御と IEEE1394 物理層の制御を行う。

MLC は FPGA で実装し、Altera MAX7256-7²⁾ を用いた。

以下に、実装する MLC の仕様を示す。

- 最大転送能力 200 Mbps の IEEE1394 物理層、データバス幅は 4 bit
- 送受信バッファは各々 2 チャンネル (Ch0, Ch1),

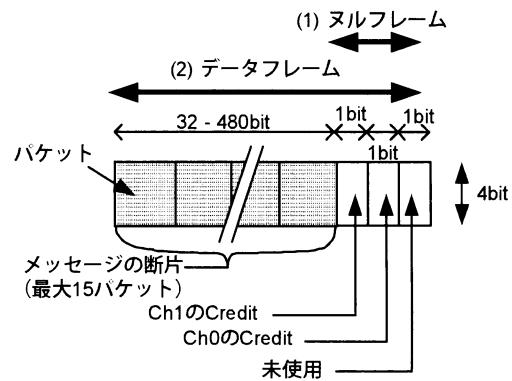


図 5 MLC におけるフレームフォーマット
Fig. 5 Frame format handled by MLC.

各容量 2 Kbyte

- パケット長は 16 byte
- 送受信バッファのバス幅 16 bit

この仕様を満たす (1) ヌルフレームと、(2) データフレームのフォーマットを図 5 に示す。通信レイテンシは、データフレームのヘッダを除くパケット数に比例する。このため、通信レイテンシを縮小するためには、そのパケット長を可能な限り短縮することが理想的である。しかし、極度に小さくすると、ネットワークバッファの各チャネルのアドレッシングに使用するアドレスレジスタの幅を大きくしなければならず、ハードウェア量が増加する。本 MLC では、我々が用いた FPGA で実装可能な 16 byte をパケット長とした。

また、図 5 に示すように、ヌルフレーム、および、データフレームは各チャネルの Credit を含む。IEEE1394 物理層に合わせ、並列に送ることができるデータ幅は 4 bit とした。各 Credit は 4 bit 幅 × 1 bit 長で構成され、パケット数を保持する。最大 15 パケットのバーストが可能である。データフレームは 32 × 4 bit (16 byte) から 480 × 4 bit (240 byte) まで可変であり、メッセージの断片を 16 byte の倍数で一度に送信することができる。

3.2.4 通信の流れ

Maestro ネットワークにおける通信の流れを、一方

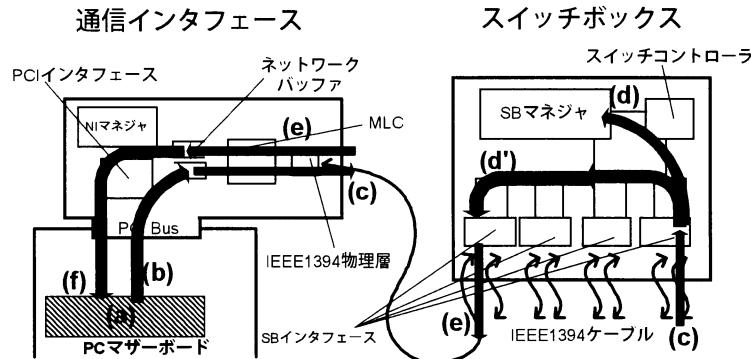


図 6 Maestro ネットワークにおける通信の流れ

Fig. 6 Message flow.

の PC メモリから他方の PC メモリに、メッセージを転送する例を用いて説明する（なお、この例は、PC メモリ間 DMA 転送として、4 章の性能評価において再度使用する）。

最初に、PC のメモリに 300 byte のメッセージが用意されたとする（図 6(a)）。このメモリは、OS が起動する前に予約するメッセージ転送用の連続領域であり、予約メモリと呼ぶことにする。予約メモリは、デバイスドライバによってアプリケーションプログラムの一部にマップされる¹⁾。用意されたメッセージは、PCI インタフェース内の DMA コントローラ、または、NI マネージャの PowerPC が直接ネットワークバッファに転送する（図 6(b)）。ネットワークバッファへのメッセージ書き込みが 16 byte を超えると、ただちに、MLC によりデータフレームに変換され、SB へと転送される（図 6(c)）。このとき、MLC におけるネットワークバーストの最大長は 240 byte であるため、最少で 2 度の送信機会に分割される。この分割の間に SB 側が送信権を獲得できるので、SB 側のネットワークバッファに送信すべきパケットがある場合、SB の MLC が送信を行う。これにより、片方のリンク端による長時間のバスの占有、偏った送信の続行を解消できる。

SB では、NI から受信したメッセージのメッセージヘッダが SB マネージャに渡される。SB マネージャはメッセージヘッダを解析し、スイッチコントローラにメッセージ転送を要求する（図 6(d)）。このとき、マルチキャスト要求がメッセージヘッダに指示されていると、送信元から複数の送信先にスイッチコントローラが DMA 転送を行う（図 6(d')）。この DMA 転送は送信先の数だけ転送を起動するのではなく、1 つのネットワークバッファから、複数の送信先ネットワークバッファへ同時転送を行う。

SB のネットワークバッファにメッセージのコピーが行われると、MLC により、NI に送信される（図 6(e)）。メッセージを受け取った NI の MLC は、送信処理の反対の順序で処理を行い（図 6(f)）、PC の予約メモリへと書き込みを行う。このとき、MLC のフレーム単位ごとの受信機能が働いて、受信側 NI のネットワークバッファから PC 上のメモリへの転送のうち、先頭 16 byte 以降のメッセージは、MLC のネットワークバッファへの書き込み処理とオーバラップされて処理できる。

以上、本 MLC は、我々の提案する 2 つの高速化技法、すなわち、リンク層におけるバースト転送、および、送信単位の小粒度化による通信の多重化を実現していることを述べた。

4. 性能評価

本章では、Maestro ネットワークの 1) 基本性能、2) リンク層におけるバースト転送の効果、3) リンク層における送信単位の小粒度化による通信の多重化の効果、に関する評価を示す。すべての実験には 3.2.4 項で説明した PC メモリ間の DMA 転送を用いる。この実験での時間計測は、NI 上の PowerPC のタイムベースレジスタをタイマとして用い、送信側が DMA コントローラによりメッセージをネットワークバッファへ転送する直前から、受信側が Acknowledge として送信側に 16 byte の Type0 メッセージを返し、これを受信側が受け取るまでの時間とする。以下の実験結果では、この時間を Remote DMA Latency として表示している。

4.1 Maestro ネットワークの基本性能

(1) MLC 間の基本性能

MLC 間の転送レイテンシは 100 MHz のロジックアナライザで計測した。16 byte 構成のパケット 1 個から

なる Type0 メッセージを転送したときの各区間でのタイミングチャートを図 7 に示す。送信側の時間 t_1 は、MLC による送信権獲得に必要な最大時間を示している。受信側の時間 t_4 は、MLC による送信権獲得のための最小時間を示している。送信権を獲得する時間は、送信権を放棄した直後に、送信バッファに 16 byte 以上のメッセージの一部が書き込まれたとき、最大となる。送信権獲得時間には $1.12 \mu\text{sec}$ の差があるが、この時間を考慮に入れて約 $8 \mu\text{sec}$ ($t_1 + t_2 + t_3 + t_4$) で受信側 MLC に到達できる。これは、IEEE1394 標準リンク層における転送遅延の最悪値である $125 \mu\text{sec}$ に比べ、約 $1/15$ である。

図 8 に PC メモリ間の DMA 転送についてのレイテンシとスループットのグラフを示す。最大スループットは約 20 Mbytes/sec であった。これは、IEEE1394 の 200 Mbps 物理層でのピーク性能の 80% を達成しており、十分な効果をあげているのが分かる。

また、MLC の双方向通信性能を評価するため、PC

メモリ間の DMA 転送を 2つの NI で同時にを行い、メッセージが交差する場合のスループットについて考察を行う。実験は、SB からの同期メッセージを受信した直後に、2つの NI が一方の NI から、もう一方の NI へ PC メモリ間 DMA 転送を開始する。時間は、1つの NI のみで行った場合と同一の区間で計測し、長い時間がかかった NI の結果を採用する。

この実験において、片方の NI における最大スループットは 10 Mbytes/sec であった。これは、上述の 1 つの NI から PC メモリ間転送を行った場合の 50% である。

(2) SB の基本性能

SB でのメッセージスイッチに要する時間は、図 7 の $t_2 + t_3$ である。

SB マネジャによって、メッセージアナライザの FIFO バッファからメッセージヘッダが読み出されながら、スイッチコントローラへ転送要求が書き込まれるまでの時間が t_2 であり、ルーティングに要する時間である。

また、 t_3 はスイッチコントローラにおいてメッセージを転送している時間である。この時間には、スイッチコントローラが転送要求を解釈し、ネットワークバッファ間の DMA 転送を行う時間が含まれる。スイッチコントローラが転送要求を解釈する時間については、 40 nsec と小さい。このとき、スイッチコントローラは、Type0 メッセージを DMA 転送する。このスループットは 400 Mbps であり、MLC の最大通信スループットの 2.5 倍である。

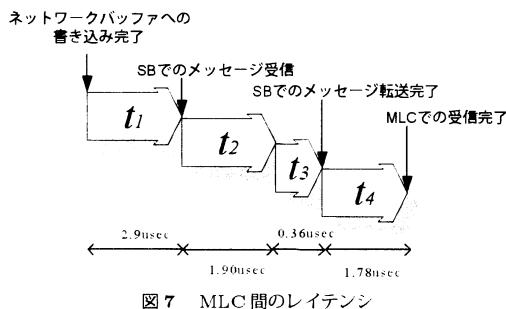


Fig. 7 Inter-MLCs communication latency.

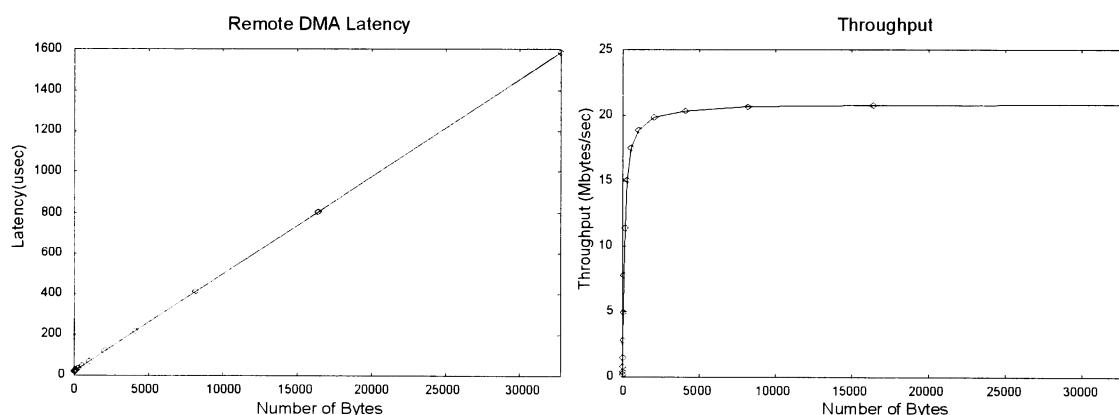


図 8 Maestro ネットワークの基本性能
Fig. 8 Basic performance of Maestro network.

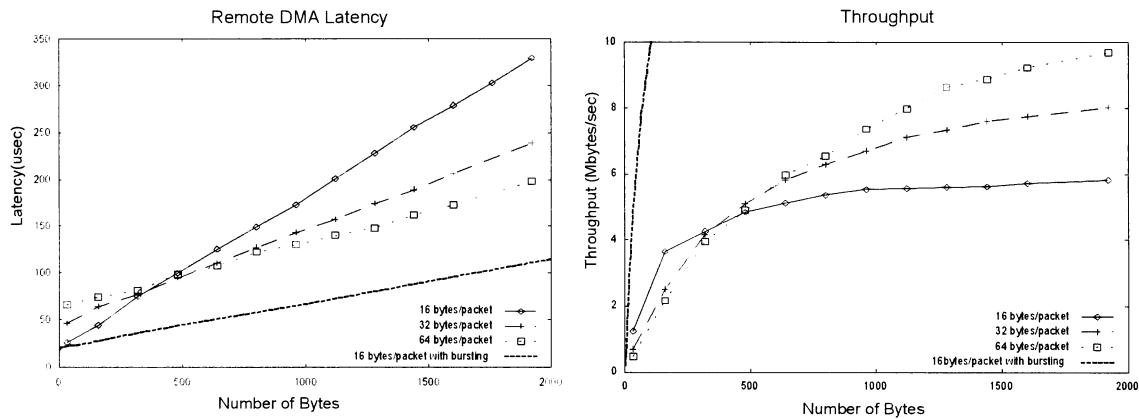


図 9 リンク層におけるバースト転送の効果

Fig. 9 Effect by network burst.

4.2 リンク層におけるバースト転送の効果

図 9 は、ネットワークバーストを用いない場合について、パケット長を 16 byte, 32 byte, 64 byte に固定した場合と、我々が行ったネットワークバーストによる高速化との比較を表している。

バーストを行わず、パケット長を増加させるとスループットは向上する。最大スループットは、パケット長を 16 byte から 64 byte にすることにより 1.6 倍以上改善されるが、逆にレイテンシは 3 倍以上増加する。

4.3 送信単位の小粒度化による通信の多重化の効果

ネットワークバッファに書き込んだメッセージが、メッセージを単位として物理層に送信される場合と、パケットを単位として行われる場合の比較を図 10 に示す。この図では、メッセージを単位とした場合を Message-based Transfer、パケットを単位とした場合を Packet-based Transfer として表している。

Maestro ネットワークにおけるメッセージ単位の PC メモリ間 DMA 転送の時間計測に関しては、PC から送信側 NI のネットワークバッファにメッセージ全体の書き込みが完了した後、MLC による通信を開始する。これに対して、受信側 NI では、メッセージ全体がネットワークバッファに到着した後、PC に転送する。メッセージを単位とする場合、図 10 の Remote DMA Latency は、送信側の PC から NI への DMA 転送の時間と、送信側 NI から受信側 NI への転送時間、および、受信側の NI から PC までの時間をそれぞれ別に採取し、これらを足し合わせた時間とする。Maestro ネットワークにおける NI のネットワークバッファの最大容量が 2 Kbyte なので、16 byte から 2 Kbyte までのメッセージ長で実験した。

パケット単位の通信の場合、送信側 NI の DMA 転送の完了を待たずに、受信側にメッセージヘッダが渡

され、PC への DMA 転送が開始される。これにより、送信・受信それぞれの通信操作が多重化され、メッセージ単位の転送方法に比べて通信レイテンシが低く抑えられ、かつ、スループットが増大している。我々のパケット単位による方法では、メッセージ長が 16 byte 付近であれば、メッセージ単位による転送との差はないものの、メッセージ長の増大に比例して、送信と受信のオーバラップ時間が長くなる。実験から、メッセージ単位による転送に比べて、最大で 8 Mbytes/sec スループットが向上している。

4.4 結果の考察と議論

以下に評価結果をまとめ、考察を加える。

(1) 基本性能

8 μsec の低レイテンシと、通信媒体のピーク性能の 80% を達成するスループットを実験により示した。半二重通信媒体である IEEE1394 を用いながら、双方通信を実現しつつ、高いスループットが維持できていることを確認した。

また、MLC の双方通信性能に関する実験では、单方向の場合の 50% のスループットが実現できることを確認した。さらに、この実験の結果から、SB のオーバヘッドである SB マネジャでのルート解析が MLC による通信操作と多重化されていることと、スイッチコントローラの DMA 転送によるバス占有時間は非常に短いことが分かる。

(2) リンク層におけるバースト転送の効果

本実験により、ネットワークバーストが通信媒体の使用効率を向上させることができることを実証した。Maestro ネットワークでは半二重通信媒体を使用しているにもかかわらず、高いネットワーク使用効率を示した。これは、バースト転送によるところが大きい。

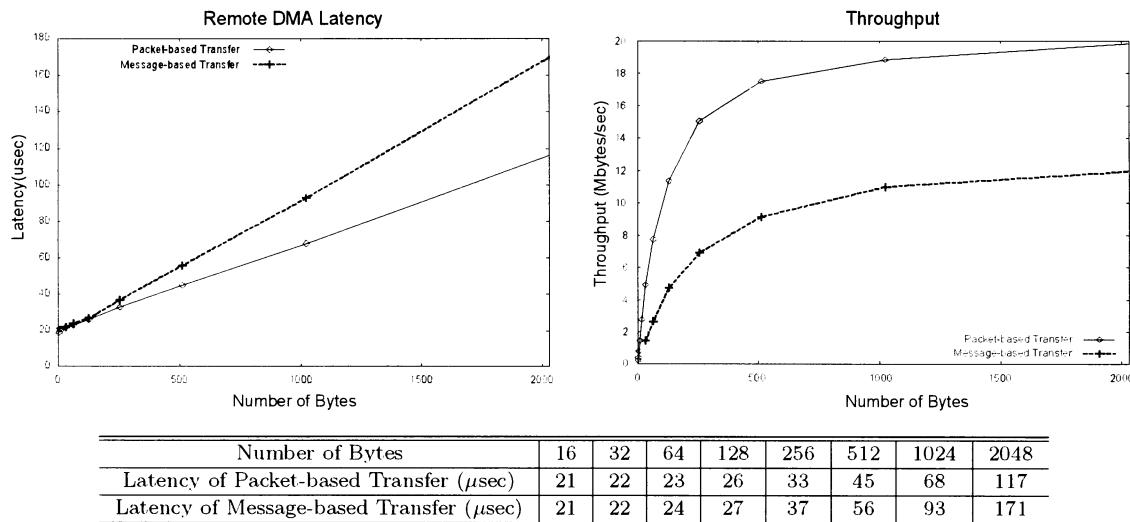


図 10 送信単位の縮小による通信の多重化の効果
Fig. 10 Effect by pipelined transfer of minimized transfer unit.

(3) 送信単位の小粒度化による通信の多重化の効果

Maestro ネットワークで行っているパケット単位の転送の場合、メッセージを単位とする場合に比べて約 8 Mbytes/sec の性能向上を示した。この実験から、送信単位を小粒度化し、送信操作と受信操作を多重化させることができ、長いメッセージに対して高いスループットを維持する有効策であることが分かる。

5. 関連研究

Maestro ネットワークに適用した高速化技法と性能に関して関連研究と比較する。

(1) リンク層におけるバースト転送の実現

TCP 等で用いられているスライディングウィンドウは、フラグメント化された断片を非同期的に送信することによりスループットを向上させている。しかし、フラグメント化された断片は、IP、Ethernet デバイスハンドラへと渡され、ヘッダ、フッタが付加されたメッセージとして構成されるので、メッセージ長の増大、レイテンシの増大につながっている。また、Myrinet では、通信媒体に “B” bit を規定し、STOP and GO によるフロー制御を行い、バースト転送を可能にしている。Myrinet では 8 bit ごとに送信を行うが、STOP を受け取るまで連続送信可能である。しかし、STOP and GO によるフロー制御アルゴリズムでは、バッファのチャネルごとに専用線を必要とするので、チャネル数の増加にともなって “B” bit を増加させる必要がある。

これに対し、AGC プロトコルでは、ネットワークバッファに用意された 1 つのチャネルからメッセージ

ジがパケットに細分化され、バースト転送されている途中で、異なるチャネルのメッセージに属するパケットが転送に割り込んでも構わない。したがって、通信媒体が頻繁にアイドル状態に陥ることはない。また、AGC プロトコルで規定するチャネルは、通信媒体に専用線を規定しないので、ネットワークバッファ量に比例する数のチャネルを構成できる。

(2) リンク層における送信単位の小粒度化による通信の多重化

TCP で行っているフラグメントーションが、我々の提案する方法と似た操作を行っているが、フラグメント化された後の操作が異なる。TCP ではフラグメント化の後、それぞれをメッセージとして扱うため、送受信操作におけるオーバヘッドが大きい。また、多くのシステムでは TCP の実現は PC で行われ、通信インターフェースからの割り込み制御で送受信操作を起動する。このため、PC 内での割り込みハンドラとデバイスハンドラによって生ずるソフトウェアオーバヘッドが非常に大きくなる。

Myrinet を用いた FM2.0⁶⁾では、送受信の開始・終了を担う関数と、メッセージの断片を送受信する関数を規定している。これらを用いて、送受信操作のパイプライン化を図っている。しかし、Myrinet は、送信時に、通信インターフェース上の SRAM にデータを一時格納しなければならないので、16 byte 程度の非常に小さな断片では PC と通信インターフェースとの DMA 転送オーバヘッドが大きくなり、実効スループットが低下する。また、Myrinet では、メッセージの先頭にルーティング情報と、最後に tail 情報を付加

表1 リンクレイヤ間レイテンシの内訳
Table 1 Breakdown of latency at link layer.

規格名 (メッセージ長 byte)	NI → SW (μsec)	SW (μsec)	SW → NI (μsec)	合計 (μsec)
Gigabit Ethernet (14 byte)	1.30	5.00	2.50	8.70
Myrinet (15 byte)	0.67	1.40	0.67	2.74
Maestro ネットワーク (16 byte)	1.78	2.26	1.78	5.82

する必要がある。これに加えて、通信インターフェース上から通信媒体への送信 DMA 転送は、各メッセージの最後で区切られるので、あるメッセージの末尾部分と次のメッセージの先頭部分を一度の DMA 転送で送信することができず、さらにスルーパットを低下させている。

一方、我々の提案した手法では、通信インターフェース上でメッセージのフラグメンテーションと転送処理を行うため、TCP のフラグメンテーションや、FM2.0 の方法と比べて、通信媒体のアイドル時間を削減することができる。

したがって、我々が提案した 2 つの高速化技法は、メッセージの基本的な操作面において、TCP のフラグメンテーション、Myrinet のチャネルの複数化、および、FM2.0 のパイプライン転送と関連しているものの、リンク層と物理層における通信性能の改善に一層重点を置き、システム全体の性能向上を図っているといえる。

(3) 性能

リンクレイヤ間転送のレイテンシについて、Maestro ネットワークの性能と、ギガビットの通信能力を持つネットワークである Gigabit Ethernet、および Myrinet とを比較した。

表1に3つのネットワークデバイスを用いた場合のリンクレイヤ間のレイテンシ（単位は μsec）の内訳を示す。それぞれのネットワークが扱うパケット構成が異なり、等量の転送については比較が困難であるため、14 byte から 16 byte と転送量に相違がある。Gigabit Ethernet については 14 byte からなるヘッダ部のみであり、Myrinet については 12 byte データに 3 byte からなるヘッダ部が付加されている。Maestro ネットワークに関しては、Type0 メッセージについて示しており、8 byte データに 8 byte のヘッダ部により構成されている。また、各ネットワークの通信インターフェースを NI、スイッチを SW として表している。NI → SW は通信インターフェース上のメッセージをスイッチに転送するまでの時間を表している。また、SW → NI はその反対の転送遅延を表す。SW における数値は、スイッチがメッセージをルーティングし、転送するまでの時間を表している。

Gigabit Ethernet については文献 19) の Essential 社の通信インターフェースと Extreme 社のスイッチに関する報告を、Myrinet については文献 18) の報告を参考にし、算出した。

Gigabit Ethernet では、スイッチのレイテンシが Maestro ネットワークの約 2 倍になっている。Myrinet については、通信インターフェース上の LANai プロセッサが通信に特化した命令を実行できるため、Maestro ネットワークの約半分のリンクレイヤ間レイテンシを実現している。

一方、スルーパットについては、文献 19) をもとに、Gigabit Ethernet を用いて 1500 byte のメッセージを PC メモリ間 DMA 転送した場合のピーク性能に対する実効性能の割合を求めた。スイッチを介した場合の実効スルーパットを文献 19) における実験結果のグラフとスイッチのレイテンシの報告より求めると、約 313 Mbps であり、ピーク性能である 1 Gbps の約 31% である。また、Myrinet を用いた場合の PC メモリ間 DMA 転送について、スイッチを介した構成での実測を行い、同様なネットワーク利用率を求めた。1500 byte を PC メモリ間 DMA 転送を行った場合、そのスルーパットは 113 Mbps であり、単方向のピーク性能である 640 Mbps の 18% である。Maestro ネットワークにおける同様の実験では、通信媒体のピーク性能の 68% を実現している。すなわち、Maestro ネットワークでは、短いメッセージであっても、ピーク性能に対して十分に高い実効性能が達成できており、ネットワークバーストと、パケット単位の送受信による通信手順の多重化が実効性能の向上に有効であるといえる。

6. おわりに

本論文では、従来の PC クラスタに使用されている通信技法の問題点を列挙し、特に、リンク層と物理層のレベルから通信ハードウェアアーキテクチャの改善を試みた。その改善方法として、リンク層においてメッセージをより短いパケットに分割し通信を多重化させ、さらに、複数のパケットをバースト転送する方法を提案した。これら 2 つの高速化技法に、半二重通信用の公平な送信権の配分方法を加え、AGC プロト

コルとして規定し、これを MLC に実装した。MLC を搭載した PC 用通信インターフェースとスイッチボックスからなる Maestro ネットワークを構築し、性能評価実験を行うことにより、我々の高速化技法が PC クラスタ向けネットワークの最適化に貢献していることを示した。

Maestro ネットワークで得られた成果は、point-to-point の通信路を構成できれば、他の通信媒体にも適応できる。その例としては、USB (Universal Serial Bus)¹⁶⁾があげられる。また、400 Mbps および、800 Mbps の次世代 IEEE1394 物理層を用いることにより、MLC で実現できる最高スループットはそれぞれ、320 Mbps, 640 Mbps となる。それにともない、ネットワークバーストの Credit が大きくできるため、一度の送信機会で転送できるパケット数が多くなり、通信の多重化がより効率良く行われる。したがって、小さなメッセージでも MLC で可能な最高スループットに漸近することが予想できる。

また、Maestro ネットワークの IEEE1394 物理層の高速化にともなう MLC の変更としては、データバス幅の拡大のみである。データバス幅が拡大すると 4 bit の物理層から 16 bit 幅のネットワークバッファのデータ幅に変換する必要がなくなるので、通信インターフェースをより小さな回路規模で構成できる。

今後の課題として、本論文で提案した高速化技法を利用する 1) ユーザレベル通信ライブラリと 2) 共有メモリライブラリの実装と評価、および、3) IEEE1394 の次世代物理層と光媒体への AGC プロトコルの適用と性能評価を考えている。

謝辞 Maestro ネットワークの実装にあたり、多大なる助言と協力をいただいた筑波大学電子・情報工学系技官小野雅晃氏に感謝を申し上げます。

参考文献

- 1) Rubini, A. and Oram, A.: *Linux Device Drivers, Chapter 13*, O'Reilly & Associates (1998).
- 2) Altera Corporation: *1998 Data Book* (1998).
- 3) Tezuka, H., Hori, A., Ishikawa, Y. and Sato, M.: PM: An Operating System Coordinated High Performance Communication Library, *High-Performance Computing and Networking, Lecture Notes in Computer Science*, Vol.1225, pp.708-717, Springer-Verlag (1997).
- 4) IEEE Standard Department: *IEEE Standard for a High Performance Serial Bus* (1994). <http://www.1394ta.org>.
- 5) Prylli, L. and Tourancheau, B.: BIP: A new protocol designed for high performance networking on Myrinet, *Workshop PC-NOW, IPPS/SPDP98*, Orlando, USA, Elsevier Science Publishers (1998).
- 6) Lauria, M., Pakin, S. and Chien, A.A.: Efficient Layering for High Speed Communication: Fast Messages 2.x, *Proc. 7th High Performance Distributed Computing (HPDC7) Conference*, Chicago, Illinois (1998).
- 7) Motorola: MPC603e & EC603e RISC Microprocessors Users Manual (1997). <http://www.mot.com>.
- 8) Boden, N.J., Cohen, D., Felderman, R., Kulawik, A.E., Sietz, C.L., Seizovic, J.N. and Su, W.-K.: Myrinet – A Gigabit-per-Second Local-Area Network, *IEEE Micro*, Vol.15, No.1 (1995).
- 9) PCI Special Interest Group: *PCI Local Bus Specification, Rev. 2.1* (1995).
- 10) PLX Technology Inc.: *PCI9060 Data Sheet VERSION1.2* (1995).
- 11) Buyya, R.: *High Performance Cluster Computing: Architectures and Systems*, Prentice Hall (1999).
- 12) Buyya, R.: *High Performance Cluster Computing: Programming and Applications*, Prentice Hall (1999).
- 13) Breyer, R. and Riley, S.: *Switched and Fast Ethernet, Second Edition*, Ziff Davis Press (1996).
- 14) Pakin, S., Karamchetti, V. and Chien, A.A.: Fast Messages (FM): Efficient, Portable Communication for Workstation Clusters and Massively-Parallel Processors, *IEEE Concurrency*, Vol.5, No.2, pp.60-73 (1997).
- 15) von Eicken, T., Basu, A. and Vogels, W.: U-Net: A User Level Network Interface for Parallel and Distributed Computing, *Fifteenth ACM Symposium on Operating Systems Principles*, pp.40-53 (1995).
- 16) USB Implementers Forum Inc.: *Universal Serial Bus Revision 1.1 specification*. <http://www.usb.org/>.
- 17) Karamchetti, V. and Chien, A.: Software Overhead in Messaging Layers: Where Does the Time Go?, *Proc. International Conference on Architectural Support of Programming Languages and Operating Systems (ASPLOS-VI)*, pp.526-531 (1994).
- 18) 松田元彦、手塚宏史、田中良夫、久保田和人、安藤 誠、佐藤三久：SMP クラスタ向けネットワーク・インターフェース上 AM 通信、情報処理学会計算機アーキテクチャ研究会資料、Vol.97, No.125, pp.55-60 (1997).

- 19) 住元真司, 石川 裕, 堀 敦史, 手塚宏史, 原田 浩, 高橋俊行: Gigabit Ethernet を用いた高速通信ライブラリの設計、情報処理学会ハイバフォーマンスコンピューティング研究会資料、Vol.72, No.19, pp.109-114 (1998).

(平成 12 年 2 月 9 日受付)
(平成 12 年 6 月 2 日採録)



山際 伸一 (学生会員)

1974 年生。1997 年筑波大学第三学群情報学類卒業。1999 年同大学大学院工学研究科博士前期課程修了。修士(工学)。現在、同大学大学院工学研究科博士後期課程在学中。並列・分散処理、および、クラスタ計算機に関する研究に従事。電子情報通信学会学生会員。



福田 宗弘 (正会員)

1963 年生。1986 年筑波大学第三学群情報学類卒業。1988 年同大修士課程理工学研究科修了。工学修士。同年日本アイー・ビー・エム(株)東京基礎研究所勤務。1995 年カリフォルニア大学アーバイン校 (University of California, Irvine), MS 授与。1997 年同大博士課程修了。Ph.D. 同大助手。1998 年筑波大学電子・情報工学系講師。モバイルエージェント、分散シミュレーションの研究に従事。



和田 耕一 (正会員)

1956 年生。1984 年神戸大学大学院システム科学専攻博士課程修了。同年、同大学大学院自然科学研究科助手。1987 年筑波大学電子・情報工学系講師。助教授を経て 1999 年教授。現在に至る。1992~1993 年カナダ、ビクトリア大学客員研究員。並列・分散処理とコンピュータアーキテクチャ、マルチメディア情報処理に関する研究に従事。学術博士。IEEE, ACM 等会員。