

IEEE1394 を用いた PC クラスタシステムのための システムソフトウェアの開発

System Softwares for the PC Cluster System Employing IEEE 1394

中山 泰一¹⁾ 兵頭 和樹²⁾ 尾崎 亮太³⁾
Yasuichi NAKAYAMA Kazuki HYODOU Ryota OZAKI

- 1) 電気通信大学情報工学科 (〒 182-8585 調布市調布ヶ丘 1-5-1 E-mail: yasu@cs.uec.ac.jp)
2) 電気通信大学情報工学科 (〒 182-8585 調布市調布ヶ丘 1-5-1 E-mail: hyoudo-k@igo.cs.uec.ac.jp)
3) 電気通信大学情報工学科 (〒 182-8585 調布市調布ヶ丘 1-5-1 E-mail: ozaki-r@igo.cs.uec.ac.jp)

ABSTRACT. In parallel cluster computing, low latency and high bandwidth network are required. It is also important that this network be cost-effective. IEEE 1394 provides a good compromise for fulfilling these requirements. Therefore, we construct a PC cluster system with IEEE 1394 which supports a 400 Mbps data transfer rate to connect nodes. In this paper, we describe the implementation and the evaluation of our systems.

1 背景

PC/AT 互換機等 (以下、PC) を何らかの通信手段によって結合し、システム全体として仮想的な並列計算機を構成する、いわゆる PC クラスタシステムに関する研究が注目されている [7]。

PC クラスタシステムに関する多くの研究において、通信手段には超高速、かつ、信頼性の高いネットワーク (Myrinet など) が用いられている。しかしながら、これらのネットワークでは特殊で高価な通信ハードウェアの使用を前提としており、必ずしも一般的であるとは言い難い。

そこで、本プロジェクトでは、通信手段に IEEE 1394 シリアルバスを用いた PC クラスタシステムを提案する。IEEE 1394 は非常に一般的な通信手段であり、これを利用することで低コストの PC クラスタシステムを構築できる。また、IEEE 1394 は、最大 400Mbps の通信帯域と非常に低遅延の通信を提供する。システムソフトウェアを整備すれば、優れた通信性能をもつ (とくに、通信遅延の小さい) システムを実現することが可能である。

まず、本プロジェクトの PC クラスタシステム (**FireCluster** と名付ける) における、システムソフトウェアの構成法について検討した。その結果、平成 12 年度において、

- MPI に準拠した汎用通信インタフェース・ライブラリ (MPICH-CF)

を開発することとした。なお、互換性のあるシステムを目指して、上記の高速通信機構と汎用通信インタフェース・ライブラリは Linux OS 上に実現した。

CF ライブラリにおいて、IEEE 1394 の遠隔メモリ書き込み機能を積極的に利用し、さらに、ユーザレベル通信やゼロコピー通信などの技術を適用して、ソフトウェア内部でのオーバーヘッドを可能な限り削減した設計により、FastEthernet を十分上回る通信性能を得た。さらに、NAS Parallel Benchmarks を用いて FastEthernet による PC クラスタシステムと比較実験を行い、高い計算能力が得られることを確認した。これらの成果は、学術論文誌などにおいても発表を行なった [4] [5] [12]。

以下、本プロジェクトの目的、成果の概要、性能評価実験の結果、本プロジェクトの現在の課題について述べる。

2 目的

本プロジェクトでは、以下の特徴を持つ PC クラスタシステムを提案し、そのためのシステムソフトウェアの開発を目的とする。

- IEEE 1394 の通信性能を最大限に活かすための高速通信機構を中心としたシステムソフトウェア (CF ライブラリ)
- 1) IEEE 1394 シリアルバスの利用
従来の PC クラスタシステムに関する多くの研究では、特殊で高価な通信ハードウェアの使用

表 1: クラスタシステムに用いられる主なネットワークとの比較

	FastEthernet	Gigabit Ethernet	Myrinet	IEEE 1394
接続	bus(or switched)	bus(or switched)	switched	bus
ケーブル長/リンク	~ 200m	~ 200m	~ 10m	~ 4.8m
最大バンド幅	100Mbps	1Gbps	1.28Gbps	400Mbps
最小片道遅延	20 μ 秒	20 μ 秒	5 μ 秒	10 μ 秒
フロー制御	×	×	○	○
プログラム可能 [§]	×	†	○	×
コスト/リンク	低	高	高	低
入手性			×	
将来性	○		—	

[§]デバイスがプログラム可能な専用のプロセッサを持っているかどうか

[†]持っているものも存在する

表 2: 市場価格の比較 (円)

	アダプタ	スイッチ	9 ノード接続時の合計
FastEthernet	1,600	14,890(16 ポート)	29,290
Gigabit Ethernet	55,000	158,000(4 ポート) × 4	1,127,000
Myrinet	130,000	550,000(16 ポート)	1,720,000
IEEE 1394	5,000	—	45,000

が前提となっている。これに対し本プロジェクトでは、IEEE 1394 高速シリアルバスを利用し、コストパフォーマンスに優れた PC クラスタシステムを実現する。

2) ソフトウェアによる高速通信機構

Myrinet などでは、通信ハードウェア自体がインテリジェントであり、それを活用することにより高速通信を確保している。これに対し本プロジェクトでは、システムソフトウェアを整備することにより高速通信を実現する。

IEEE 1394 上で IP プロトコルによる通信を実現する試みもされているが、この方式ではソフトウェア内部でのオーバーヘッドが大きく、IEEE 1394 の性能を活かした高速通信を実現することは不可能である。

そこで、本プロジェクトでは、IEEE 1394 の遠隔メモリ書き込み機能を利用した専用の通信ソフトウェアを開発する。さらに、ユーザレベル通信やゼロコピー通信などの技術を適用して、ソフトウェア内部でのオーバーヘッドを可能な限り削減する。

3) 互換性のあるシステム

2) に述べた通信機構を Linux OS 上に実現する。また、MPI に準拠した汎用通信インタフェースを用意することにより、互換性のある利用しやすいシステムを実現する。

3 高速シリアルバス規格 IEEE 1394

本章では、**FireCluster** で使用する高速シリアルバス IEEE 1394 [1] について説明する。

(1) 他の通信手段との比較

まず、クラスタシステムに用いられる他の主なネットワークと、IEEE 1394 の比較検討する。

表 1、表 2 は、ネットワーク構成、最大バンド幅、最小片道通信遅延などの技術的な特徴のほか、購入のコストや入手の容易性などについて比較したものである。コストはスイッチまでを含めた各ノード間を接続するためのホストアダプタの価格、入手性は提供しているベンダ数や、市場に出回っている製品の種類による。

Myrinet は、非常に高いバンド幅と十分低遅延の通信を提供している。しかしながら、FastEthernet などと比べると非常に高価であるなど、あまり一般

的であるとは言えない。Gigabit Ethernet は、バンド幅こそ非常に高いが、通信遅延が大きい。さらにスイッチなどを含め、やはり機器が高価である。また FastEthernet は、低コストで入手性にも優れているが、通信遅延が大きく、最大バンド幅も 100Mbps であり、性能の面で十分とは言えない。

一方 IEEE 1394 は、Myrinet などと比べると、いくつかのトレードオフはあるものの、十分低遅延で高バンド幅の通信を提供していることや、低コストで入手性に優れていることなど、様々な要求を満たしている。また、現在 800Mbps, 1.6Gbps, 3.2Gbps の規格 [8] が策定中であり、将来、ますます高速になることが期待できる。

(2) IEEE 1394 の特徴

IEEE 1394 は、低コストの汎用高速シリアルバス規格である。その通信方式は従来の I/O に基づく通信方式とは異なり、メモリ read/write 通信方式を採用している。そのアドレスモデルは、他の IEEE バスと互換性のある IEEE 1212 (CSR: Control and Status Register architecture) に基づいている。

IEEE 1394 では、二つの通信モードが提供されている。一つは、Isochronous 通信モードと呼ばれる、アプリケーションに Quality of Service (QoS) を提供する同期通信である。この通信モードでは、アプリケーションレベルでの通信帯域を保証しているが、通信の信頼性はなく、主にマルチメディアなどのストリームデータの転送に利用される。もう一つは、Asynchronous 通信モードと呼ばれる非同期の通信で、信頼性のある通信を提供している。この通信モードは、トランザクション層と呼ばれる上位層で、read, write, lock の各トランザクションを提供するために用いられている。

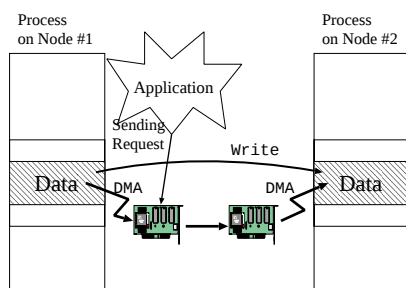


図 1: Physical Write 概念図

TI 社のリンクチップである PCILynx や、IEEE 1394 ホストコントローラの標準仕様となっている Open HCI では、Physical Write という機能が提供されている [15]。Physical Write とは、IEEE 1394 パケットヘッダの一部であるアドレス部分に、物理

メモリ空間のアドレスを指定することで、相手ノードの物理メモリに直接データを書き込む機能である。これらの処理は IEEE 1394 アダプタの DMA によって行われ、ホストプロセッサへの割り込みは起こらない (図 1)。

また、Asynchronous 通信モードのアービトレーションでは、フェアネスインターバルの概念を採用している。IEEE 1394 の各ノードは、一回のインターバル内で一回だけバス使用権を獲得し、Arbitration Reset Gap と呼ばれる一定時間のバス無使用時間を検出することで、次のフェアネスインターバルに移行する。これによって、各ノードに平等のバス使用権を与えることができる。さらに、ホットプラグ機能を採用しており、ノードの動的な抜き差しが可能となっている。

4 本プロジェクトの成果の概要

本プロジェクトでは、IEEE1394 シリアルバスを用いた PC クラスタシステム **FireCluster** を提案し、以下の 2 種類のシステムソフトウェアを設計、実現した。設計の詳細については、文献 [4] [6] [12] を参照されたい。

(1) CF ライブラリ

IEEE1394 の通信性能を最大限に活かすための高速通信機構である。

この通信機構は、Buffer-to-Buffer でのゼロコピー通信とユーザレベル通信を用いたコネクション指向の 1 対 1 通信と、ハードウェアのブロードキャスト機能を活用したグループ通信やバリア同期などの並列アプリケーションにかかせない機構を提供する。

次章の性能評価実験の結果、最小ラウンドトリップタイムが 20.2μ 秒と、Myrinet などの高速なネットワークにも劣らない、非常に低遅延の通信を実現していることを確認した。スループットに関しては、最大 250 Mbps に留まったことなど、いくつか課題を残したが、非常に低遅延の通信を実現しており、本システムは小さいデータの通信が頻繁に発生するようなアプリケーションでは十分に有効であると考えられる。

(2) MPICH-CF

MPI (Message Passing Interface) に準拠した汎用言語インタフェース・ライブラリ。MPI の代表的な実装である MPICH に手を加えることで、(1) の高速通信インタフェース CF を使えるようにしたものである。

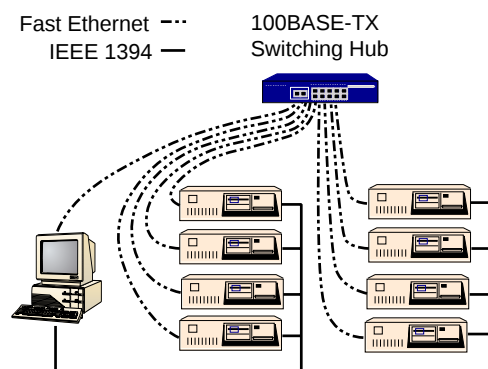


図 2: システム構成

表 3: 各計算機 ノードの仕様

プロセッサ	Intel Celeron 366MHz (FSB = 66MHz)
メモリ	64MBytes SDRAM
IEEE 1394 アダプタ	IOI-1394TTO 400Mbps (3 ポート)
Ethernet アダプタ	DEC 21140 チップ搭載 100Mbps
OS	Linux 2.0.36

次章の性能評価実験の結果、TCP/IP を利用した MPI の実装である MPICH-p4 との比較実験では、NAS Parallel Benchmarks のうち、MPICH でコンパイル可能であった 4 つのプログラム全てにおいて、MPICH-CF が MPICH-p4 を上回っている。この結果は、高速通信機構 CF と IEEE 1394 の優位性を示すものである。

5 性能評価実験

(1) システム構成

実験に使用するシステムは、8 台の計算機 ノードと 1 台のフロントエンドマシンの計 9 台の PC によって構成した (図 2)。グループ通信の信頼性を保証するためのシーケンサモジュールは、計算に参加しないフロントエンドマシンに組み込んだ。

各 PC は、低コストかつ容易に構築できるという **FireCluster** の設計思想に基づいて、一般的に入手可能な汎用のパーツのみで構成した (表 3)。また、各 ノード PC 上で動作するオペレーティングシステムとして Linux 2.0.36 を使用した。

各 ノードの接続には、3 つのポートを持つ IOI 社製の 400Mbps の IEEE 1394 ホストコントローラを用いた。このホストコントローラは、IEEE 1394 ホストコントローラの標準仕様である Open HCI に準

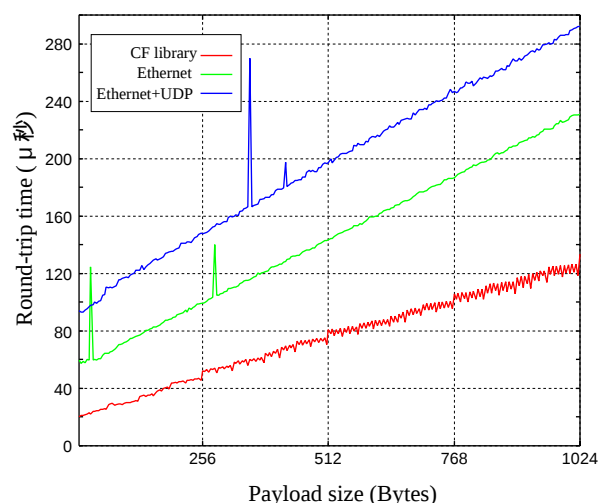


図 3: ペイロード長対ラウンドトリップタイム

拠しており、物理チップ、リンクチップ共に米 TI 社製のものが使用されている。また、各 ノードを、スイッチングハブを介して FastEthernet によっても接続した。並列アプリケーションなどの高速な通信には IEEE 1394 を使用し、ファイルシステムの共有など、システム運用に必要な通信には FastEthernet を使用する。

(2) ラウンドトリップタイム

図 3 は、ユーザから見た実効データサイズであるペイロード長に対するラウンドトリップタイム (RTT) の測定結果である。

図中の “Ethernet+UDP” の項目は、オペレーティングシステムのサービスを利用して FastEthernet 上での UDP 通信を用いて測定したものである。“Ethernet” の項目は、測定処理をドライバ中に埋め込み、カーネル呼出とデータコピーのオーバーヘッドを排除して FastEthernet の物理的な限界性能を測定したものである。なお、これらはどちらもフロー制御などは行なっておらずデータの信頼性は保証されていない。

ペイロード長が 4 Bytes の時、“Ethernet+UDP” の $94\mu\text{秒}$ 、“Ethernet” の $60\mu\text{秒}$ に対して、CF ライブラリでは $20.2\mu\text{秒}$ であった。CF ライブラリはデータの信頼性と到着順序を保証していることを考えると、FastEthernet に比べ非常に低遅延の通信を実現しているといえる。

(3) スループット

図 4 は、ペイロード長に対するスループットの測定結果である。測定は、ある ノードから別 ノードへの一方向通信によって行なった。

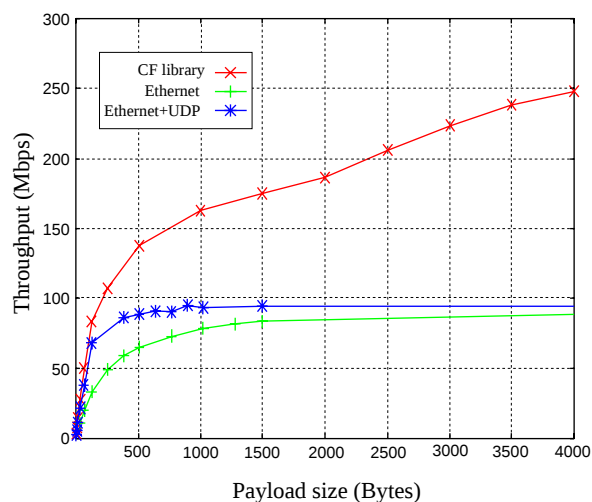


図 4: ペイロード長対スループット

CF ライブラリのスループットは最大で約 250 Mbps であり、FastEthernet の 100Mbps を大きく上回るが、今回使用したアダプタの示す 400Mbps には届いていない。これは、IEEE 1394 では通信と通信の間に Gap と呼ばれるバス無使用時間が発生することにより実効的な最大バンド幅が低下していることや、バッファ領域へのデータコピーによるオーバーヘッドが発生していることなどが原因と考えられる。

さらにスループットを改善する方法としては、ユーザレベルでのゼロコピー通信が挙げられる。これは、ユーザが利用しているメモリ領域を動的に Pin-down し、DMA 転送を可能とすることで実現する。これと併せて Pin-down cache の技術を利用すれば、効率の良い大容量通信が可能になると考えられる。

(4) グループ通信

図 5 は、マルチキャストで受信ノード数を 1 ~ 7 まで変化させた時の、最大スループットの測定結果である。表の縦軸は各受信ノードの最大スループットの平均を表している。比較として、1 対 1 通信を受信ノード数分繰り返す、マルチキャストのエミュレーションによる測定も行なった。

1 対 1 通信を用いたエミュレーションでは、受信ノード数の増加に伴い性能が劇的に劣化しているがわかる。一方、CF ライブラリで提供するマルチキャスト通信では受信ノード数に依らず 190Mbps 以上のスループットを得ている。

順序保証に BB-method[9] を用いておりオーバーヘッドがかかること、ドライバを介した受信処理によるオーバーヘッドがあることにより、1 対 1 通信ほどの性能は得られないが、受信ノード数によらず一定のス

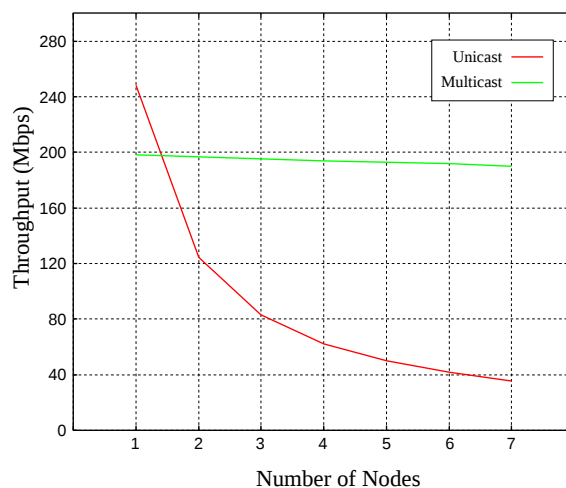


図 5: グループ通信

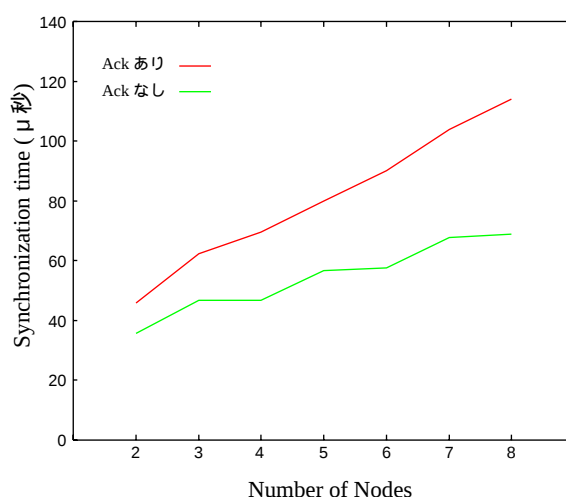


図 6: バリア同期の遅延

ループットを得ており、IEEE 1394 のブロードキャスト機能が十分に活かしているといえる。

(5) バリア同期

図 6 は、ノード数を 2 ~ 8 まで変化させたときの、バリア同期にかかる遅延時間の測定結果である。測定は、10,000 回連続した同期処理にかかる時間を測定し、1 回あたりの処理時間を算出することで行なった。Ack の“あり”、“なし”は、シーケンサからの同期完了メッセージに対して Ack メッセージを送り返すか返さないかである。“Ack あり”は、1 回の同期操作にかかる処理時間を示しており、“Ack なし”は、各ノードが同期待ちから回復するまでの遅延時間を示している。

1 回の同期操作にかかる処理時間は、シーケンサに

表 4: 通信ノード数に対する性能変化

	1 pair	2 pairs	3 pairs	4 pairs
RTT(μ s)	20.2	23.9	34.8	43.2
スループット (Mbps)	250	163	113	86

表 5: NPB による計測結果

LU Class W (Mflops/s)		
PE 数	MPICH-CF	MPICH-p4
1	41.79	40.94
2	73.72	69.69
4	139.91	116.03
8	264.91	148.39

IS Class W (Mops/s)		
PE 数	MPICH-CF	MPICH-p4
1	4.12	4.07
2	5.35	3.06
4	5.95	2.69
8	6.83	1.87

SP Class W (Mflops/s)		
PE 数	MPICH-CF	MPICH-p4
1	17.38	17.44
4	58.83	46.39

BT Class W (Mflops/s)		
PE 数	MPICH-CF	MPICH-p4
1	28.10	28.07
4	92.76	70.90

おける同期メッセージや Ack メッセージの処理の逐次化のために、線形に増加している。しかし、同期待ちから回復するまでの遅延については、IEEE 1394 のブロードキャスト機能を利用することで、ノード数増加時の遅延増加を最小限に押えることができています。

(6) ノード数増加時のスケーラビリティ

IEEE 1394 では、現在スイッチングハブのような競争を緩和するための機器は製品化されていない。このため、ノード数が増加するにつれて各ノードの実質的な通信性能が低下することが考えられる。

このことについて、一対一通信を複数組同時に行い、通信性能を測定した。測定は、最小 RTT と最大スループットの各組の平均をとることで行なった(表 4)。

最小 RTT、最大スループットともに、ノード数が増加するにつれて性能が低下しており、ノード数が増加した際のスケーラビリティは期待できない。しかしながら現在の構成では、FastEthernet 以上の性能は期待できる。特に遅延に関しては、FastEthernet より十分に良い結果を示しており、本システムは小さいデータの通信が頻繁に発生するようなアプリケーションでは十分に有効であると考えられる。

ノード数増加時の性能低下に関しては今後の課題とするが、この問題を根本的に解決するためには、スイッチの様な機器の登場が必要であると考えられる。

現在、NEC により IEEE 1394 スイッチの開発が発表されているが [11]、このスイッチの製品化により、ノード数増加時の性能低下問題が解決されることが期待できる。

(7) 実アプリケーションを用いた性能評価

実アプリケーションを用いた性能評価として、MPI によって記述された NAS Parallel Benchmarks (NPB) によるベンチマークテストを行なった。

NPB は、NASA Ames Research Center で開発されたベンチマークプログラムであり、並列計算機の実効性能を計測するために広く用いられている。

また、比較として TCP/IP を利用した MPICH の実装である MPICH-p4 を用いた計測を行なった。なお、MPICH-p4 で使用されるネットワークは FastEthernet である。

NPB の 8 つプログラムのうち IS、LU、SP、BT の 4 つのプログラムを用いて性能評価を行なった。測定結果を表 5 に示す。また、IS と LU については、プロセッサ数対性能のグラフと性能曲線を図 7 ~ 図 10 に示す。

測定結果より、4 つのベンチマークすべてにおいて MPICH-CF が MPICH-p4 を上回った。特に IS は、MPICH-p4 ではプロセッサ数が増えるにしたがい性能が低下しているのに対して、MPICH-CF ではわずかではあるが性能向上を示している。これは、IS が特に通信の多いアプリケーションであり、TCP/IP

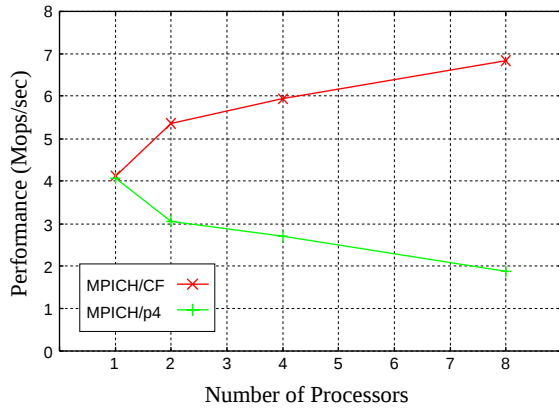


図 7: NPB (IS) のプロセッサ数対性能

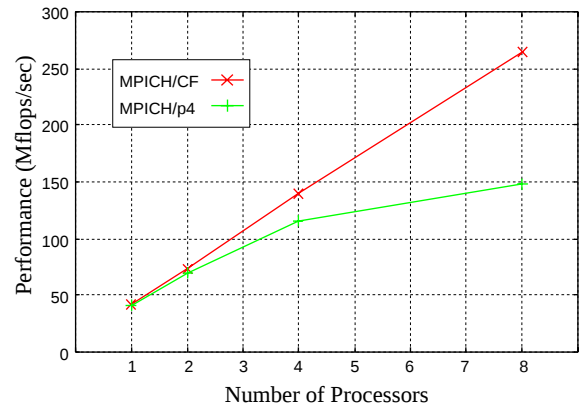


図 9: NPB (LU) のプロセッサ数対性能

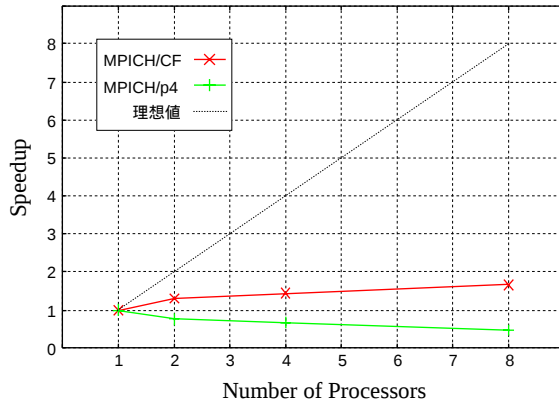


図 8: NPB (IS) の性能曲線

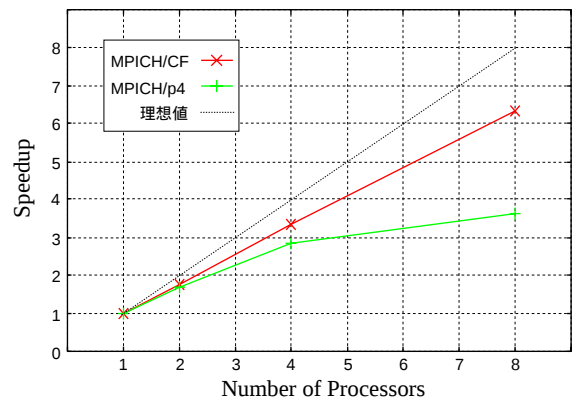


図 10: NPB (LU) の性能曲線

経由の MPICH-p4 では、プロセッサ数増加による計算性能の向上よりも通信によるオーバーヘッドの方が大きくなったためであると考えられる。一方 MPICH-CF では、IEEE 1394 と高速通信機構 CF の効果により通信のオーバーヘッドが小さく、わずかではあるが性能が向上したと考えられる。

6 関連研究

PC クラスタシステムのための通信ライブラリに関する研究として、次のものがある。

PM [14] は、Myrinet 上で信頼性のある非同期メッセージ通信とリモートメモリアクセスによるゼロコピー通信を提供する低レベル通信ライブラリで、片道通信遅延 $7.5\mu\text{s}$ 、最大スループット 113.5MB/s と非常に高速な通信を実現している。

また、住元ら [13] は、Gigabit Ethernet を用いて、クラスタコンピューティング向けの高速度通信手段と従来の TCP/IP の様なプロトコルを同時に提供する

ライブラリを設計、実現している。このライブラリは、PM と同じ API を提供し、最小ラウンドトリップタイムは $44.6\mu\text{s}$ 、最大スループットは 98.2MB/s である。

本プロジェクトと同じく IEEE 1394 を用いた PC クラスタシステムの研究として、Böszörményi ら [2] は、IEEE 1394 でのグループコミュニケーションの実現法、Isochronous 通信を用いることによる同期操作の排除などを提案している。また、それらを用いたときの、離散フーリエ変換の並列実装における性能向上予測を行なっている。しかし、実際にシステムを構築していない。

FireCluster は IEEE 1394 を用いて PC クラスタシステムを実現した最初の例と筆者らは考えている。

7 おわりに

本プロジェクトでは、IEEE 1394 高速シリアルバスを用いた PC クラスタシステム **FireCluster** を提

案し、コスト性能比に優れた高性能並列計算システムを実現した。最後に、本プロジェクトの現在の課題について述べる。

1) Linux 2.4 への対応

Linux のバージョンアップにともない、CF ライブラリの新しい Linux への対応を行なっている。現在、Linux 2.4.7 において動作している。

2) 並列アプリケーションのスケジューリング機構

複数の並列アプリケーションを効率的にスケジューリングするジョブスケジューリング機構を設計中である [10]。

3) 並列アプリケーションについての検討

並列アプリケーションについて検討中である。**FireCluster** は、とくに通信遅延について Gigabit Ethernet よりも小さいという特徴をもつため、小さなデータを頻繁にやりとりする並列アプリケーションが適している。また、動的に通信が発生する並列アプリケーションによる性能評価を考えている。

8 参加企業及び機関

本プロジェクトは実施管理組織を (株) キャンパスクリエイイトにお願いした。

9 参考文献

[1] Anderson, MindShare, Inc.: *FireWire System Architecture*, 2nd Edition, Addison-Wesley (1998).

[2] Böszörményi, L., Hölzl, G. and Pirker, E.: Parallel Cluster Computing with IEEE1394-1995, ACPC'99, LNCS 1557, pp.522-532 (1999).

[3] Gropp, W., Lusk, E., Doss, N. and Skjellum, A.: A High-Performance, Portable Implementation of the MPI Message Passing Interface Standard, *Parallel Computing*, Vol. 22, pp. 789-828 (1996).

[4] 兵頭 和樹, 中山 泰一: IEEE1394 を用いた PC クラスタシステム—通信機構の設計と評価, 情報処理学会論文誌, Vol. 41, No.SIG 8(HPS2), pp.39-47 (2000).

[5] Hyoudou, K. and Nakayama, Y.: FireCluster: PC Cluster System Employing IEEE 1394, *IEEE International Conference on Cluster Computing CLUSTER2000*, pp.363-364, Chemnitz, Germany (2000).

[6] 兵頭 和樹: IEEE 1394 を利用した PC クラスタシステム, 電気通信大学大学院電気通信学研究科情報工学専攻修士論文 (2001).

[7] 石川 裕: コモディティハードウェアを用いた並列処理技術, 情報処理, Vol.39, No8, pp.784-791 (1998).

[8] IEEE Standards Department Copyright and Permissions: P1394b Standard for a High Performance Serial Bus (Supplement), P1394b Draft 0.09 (1998).

[9] Kaashoek, M.F. and Tanenbaum, A.S.: Efficient reliable group communication for distributed system, Rapport IR-295, Faculteit Wiskunde en Informatica (1992).

[10] 小堺 康之, 兵頭 和樹, 中山 泰一: PC クラスタシステム FireCluster におけるジョブスケジューリング機構の設計, 情報処理学会第 43 回プログラミングシンポジウム論文集 (予定).

[11] NEC Laboratories: <http://www.nec.co.jp/press/ja/0008/0201.html> (2000).

[12] 尾崎 亮太, 兵頭 和樹, 中山 泰一: IEEE1394 を利用した PC クラスタシステム FireCluster における MPI の実装と評価, 並列処理シンポジウム JSP2001 論文集, pp.135-136 (2001).

[13] 住元真司, 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, 並列処理シンポジウム JSP2'99, pp.63-70 (1999).

[14] Tezuka, H., O'Carroll, F., Hori, A. and Ishikawa, Y.: Pin-down Cache: A Virtual Memory Management Technique for Zero-copy Communication, *Proc. Int'l Parallel Processing Symp.*, IEEE CS Press, Los Alamitos, Calif., pp.308-314 (1998).

[15] The Promoters of The 1394 Open HCI: *1394 Open Host Controller Interface Specification*, Release 1.00 (1997).