ハイパフォーマンス
コンピューティング 81－4
（２０００．６．２）

PC クラスタ *Maestro* 上の メッセージパッシングライブラリ MMP の提案と評価

山際伸一[*] クンカセーム プーシット[**] 和田耕一[***]

論文概要

我々は，PC クラスタにおける通信について注目し，従来の中・広域ネットワークにおける(1)通信プロトコルソフトウェア，(2)デバイスハンドラ，(3)ネットワークハードウェアそれぞれに再検討を加え，最適化されたネットワークを有する PC クラスタの開発を進めている．このうち (3)ネットワークハードウェアに関してはリンク層におけるバースト転送と送信単位の縮小化による通信の多重化を図った PC クラスタ向けネットワークである Maestro ネットワークを開発した．

本論文では(1)通信プロトコルソフトウェアと(2)デバイスハンドラに最適化を施したメッセージパッシングライブラリ MMP の構成と評価について述べる．さらに，MMP を Maestro ネットワークで接続された PC クラスタに実装する．通信実験により MMP に適用した高速化技法のうち，フラグメンテーションの最適化と，通信インターフェースによる自律的な受信領域確保が有効であることを示す．さらに，アプリケーションプログラムを用いて本システムの全体評価を行い，良好な結果が得られたことを示す．

Design and Implementation of Message Passing Library on *Maestro* PC Cluster

SHINICHI YAMAGIWA[*], PUSIT KULKASEM[**] and KOICHI WADA[*]

Abstract

PC cluster's interconnection network has been based on the technology available in conventional WAN or LAN. Focusing on the communication characteristics of cluster computing, we pursue the performance optimization of PC clusters. We have developed *Maestro* network that reduces the overhead of conventional network hardware by using two new optimizations: the message pipelining by minimization of the transfer unit, and transferring the units with burst.

This paper describes the design and implementation of the message-passing library called MMP implemented on the PC cluster connected with *Maestro* network. MMP includes two new optimizations in the layer of protocol software and in the layer of device handler: the optimization of fragmentation and the autonomous memory allocation of a received message by the network interface. The paper also evaluates the performance of MMP as well as the overall performance of the PC cluster connected with *Maestro* network.

1 はじめに

マイクロプロセッサの価格･性能比の向上に伴い，PC を計算要素とした PC クラスタ環境における並列計算が注目され，高性能なシステムとアプリケーションに関する研究が各所で進められている[2)7)]．

しかしながら，PC クラスタでは，多くの場合，中・広域通信のための通信プロトコルとネットワークハードウェアを用いて構築された環境が多く，並列計算での通信に求められる低いレイテンシ，かつ高いスループットを実現するのは困難である[4)]．

通信路におけるオーバヘッドをもたらす要因として，(a)通信プロトコルソフトウェア，(b)デバイスハンドラ，(c)ネットワークハードウェア，におけるオーバヘッドが挙げられる．これらのオーバヘッドを総合的に検討し，それぞれに対して改善を行うことが，高性能 PC クラスタの実現に不可欠である．

我々は，これらのうちネットワークハードウェアのオーバヘッドを改善した PC クラスタ用ネットワークを開発した．本ネットワークを Maestro ネットワークと呼ぶ[8)]．Maestro ネットワークは，PC に接続される通信インターフェースと，8 つの通信インターフェースを IEEE1394 で接続するスイッチボックスから構成され，リンク層における，(1)バースト転送と，(2)送信単位の縮小による通信の多重化を特徴とする AGC(Adaptive Granularity Control)プロトコルを実装するリンクレイヤコントローラ MLC(Maestro Link Controller)を搭載している．

本論文では，上述の(a)通信プロトコルソフトウェア，および，(b)デバイスハンドラに関するオーバヘッドを改善するメッセージパッシングライブラリ MMP を設計し，Maestro ネットワークに実装を行い，

*筑波大学工学研究科, Institute of Information Sciences and Electronics, University of Tsukuba
**ブラパー大学理学部情報科学科, Department of computer science, faculty of science, Burapha University, Thailand
***筑波大学電子・情報工学系, Institute of Information Sciences and Electronics, University of Tsukuba

Maestro ネットワークの全体性能を評価する．

MMP には新しい高速化技法として，(i)送受信におけるフラグメンテーションの最適化，(ii)通信インターフェースによる自律的な PC 内送受信領域の確保を適用する．これらの高速化技法と MMP の設計と実装について述べ評価を行う．

以降，2 章では従来の PC クラスタにおける通信とオーバヘッド要因について議論する．3 章では，Maestro ネットワークについて説明し，4 章では，MMP の設計と実装についてのべる．最後に評価実験を行い，本論文のまとめを述べる．

2 PC クラスタにおける従来の通信と遅延要因

高い性能を維持する PC クラスタの構築に必要な通信特性として，細粒度通信での低いレイテンシと，行列計算等における大量のデータ通信での高いスループットを維持することが必要になる．

また，並列計算における通信最適化として，通信と計算との重ね合わせも重要である．この最適化には PC での送信データの用意と通信インターフェースでの送信操作が逐次的になることを回避するため，通信インターフェースが PC と独立に動作することが要求される．

これらの通信特性への要求の一方で，従来の PC クラスタでは通信路でのオーバヘッドが問題になり，高い性能を得るのが困難である．このオーバヘッドに関する議論を以下に行う．

(1) 通信プロトコルソフトウェアのオーバヘッド

PC 内で行われる処理のうち，大きなオーバヘッド要因となりうるのは通信データのコピー操作である．広域で用いられる TCP/IP では，送受信関数の内部で複数回のコピーが行われている [4]．このコピー操作を削減することが高速通信を実現するためには必要である．

(2) デバイスハンドラのオーバヘッド

デバイスハンドラで問題となるのは，送受信完了に伴う割り込み操作と送信時の不適切なフラグメント長を用いた通信操作のパイプライン化である．

前者の割り込み操作は，PC の割り込み処理ハンドラが起動されるレイテンシが大きいだけでなく，それに伴うアプリケーションプログラムの中断が問題となる．

一方，後者について，従来のネットワークでは PC での送信データの準備と PC から NI への転送操作の多重化を図るフラグメンテーションが行われている．しかし，最適なフラグメント長はネットワークハードウェアの性能やバッファサイズに依存する．したがって，送受信操作の多重化を図り，高い通信スループットを得るには，フラグメント長を最適化することが重要である．

(3) ネットワークハードウェアのオーバヘッド

従来のネットワークハードウェアでは，一度の送信機会に一つの送信単位（例えば，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．

スケーラブルな性能を維持する PC クラスタを構築するためには，上述の(1)(2)(3)におけるオーバヘッドを十分，考慮した最適化を行う必要がある．そこで，我々はこれら 3 つのオーバヘッドの削減を行うための最適化について検討を行った．

(3)に関しては，我々が開発した Maestro ネットワークが効果的な改善を行っている．そこで，Maestro ネットワークへのメッセージパッシングライブラリを構築することにより，(1)(2)(3)のすべてに対する最適化を試みる．

次節では，Maestro ネットワークについて説明を行い，Maestro ネットワークの構成上，メッセージパッシングに効率よく適合する機能について述べる．

3 Maestro ネットワーク

3.1 高速化技法

我々は，Maestro ネットワークに以下の 2 つのクラスタコンピューティング向けの高速化技法を適用している．

- リンク層におけるバースト転送の実現

通信インターフェースの送信時における物理層への調停回数の増大によるオーバヘッドを回避するため，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．物理層により取得した送信機会毎に，リンク層での最小送信単位の整数倍のデータを一括送信する．したがって，バーストにより，長いメッセージの送信時間が短縮され，スループットが向上する．このバーストを伴う送信操作のことをネットワークバーストとよぶことにする．

- リンク層における送信単位の縮小

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，全てのデータが PC から通信インターフェースに転

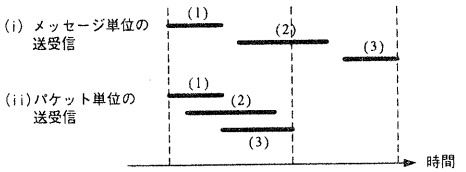

図 1　送信単位縮小による通信の多重化

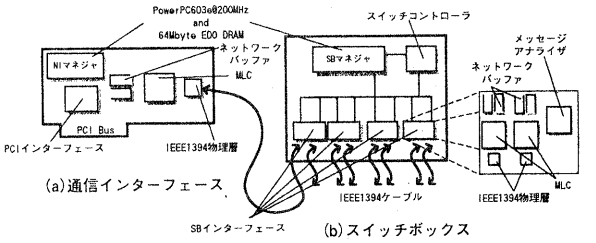

図 2　Maestro ネットワーク

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縮小化を行う．この小さい送信単位のことをパケットとよぶことにする．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，リンク層における通信の多重化が行われる．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．

3.2 Maestro ネットワークの構成

図 2 に示すように，Maestro ネットワークは，各 PC に PCI バスを介して接続される通信インターフェース，および，IEEE1394[3]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 は DMA コントローラを内蔵した PCI インターフェース[5]，マイクロプロセッサ PowerPC603e (200MHz)と 64Mbyte の EDO DRAM を搭載した NI マネジャ，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 物理層から成る．

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 プロトコルは半二重通信媒体において双方向通信を実現し，3.1 節で述べた 2 つの高速化技法を実現する．すなわち，AGC プロトコルはパケット単位での送受信にネットワークバーストを併用することにより，通信手順の多重化と通信媒体の利用率を向上させ，高い実効スループットを得る．

一方，SB は，PowerPC603e(200MHz)と 64Mbyte の EDO DRAM を搭載した SB マネジャ，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 インターフェース 4 対から構成される．

各 SB インターフェースは，1 つのメッセージアナライザ，二対の 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の種類を解析し，メッセージヘッダ部分のみを SB マネジャに転送する．

SB マネジャは，メッセージアナライザからのメッセージヘッダを解釈し，メッセージの転送先を決定した後，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 インターフェース上のネットワークバッファ間で 400Mbps 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 インター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，これを利用して選択的に複数の SB インターフェース内のネットワークバッファにメッセージ転送を行う．

以上に，Maestro ネットワークに用いられている高速化技法と，その全体構成について述べた．

Maestro ネットワークの機能のうち，NI と PC 間のデータ転送を行う機能（DMA，PCI インターフェースによる直接アクセス）と，NI の自律動作可能な構成が，通信プロトコルソフトウェアとデバイスハンドラによるオーバヘッドを改善する手段となる．これらの手段を効率よく利用するための高速化技法を用いたメッセージパッシングライブラリ MMP の設計と実装について次節に述べる．

4　MMP の設計と実装

4.1 高速化技法

(1)フラグメンテーションの最適化

コピー操作は通信レイテンシの増大要因になりうるが，異なるレイヤ間でのコピー操作は処理のオー

バラップを期待でき，スループットを増大できる．たとえば，送信時における(i)PC と通信インターフェース間，(ii)送信側と受信側の通信インターフェース間，(iii)受信側通信インターフェースと PC 間，での転送の一部がオーバラップできる．

しかしながら，(i)(ii)(iii)のそれぞれの操作で交換されるデータの大きさはネットワークの特性により異なるため，一意に決定できない．たとえば，ネットワークのピークスループットの大小に関わらず，断片が大きすぎると PC から通信インターフェースへのコピー操作の完了待ち時間が長くなり，逆に，断片が小さすぎると NI における送受信処理の起動が多発し，ネットワークの使用効率が低下する．そこで，大きなデータの送信に関しては，実験による最適なフラグメント長を導入し，データ転送手順をパイプライン化する．

(2)通信インターフェースの自律的な受信領域確保

通信インターフェースが送受信完了したことを PC に割り込みを用いて通知する方法では，PC の計算を中断させる問題があることを 2 節に述べた．我々は，この問題を改善するために，メッセージ長の受信領域を PC 内メモリに通信インターフェースが自律的に確保し，転送することを提案する．つまり，送受信メッセージ領域の確保を PC と通信インターフェースがそれぞれ独立に行うことにより，PC で行われる計算と，通信インターフェースでの処理を重ね合わせることが可能となる．

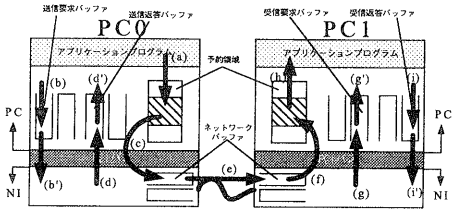

図 3　MMP ライブラリの送受信操作

4.2 Maestro ネットワークへの実装

MMP は PC 内のメモリで一度コピーを行うことにより，PC と NI での送受信データ転送操作を多重化する．さらに，MMP の送受信関数はノンブロッキングで実行される．MMP ライブラリは 3 つの送受信関数 MP_send(), MP_receive(), MP_wait()と，同期関数 MP_barrier()から構成される．

MP_send()は，ノンブロックの送信をおこなう関数である．MP_send()を実行すると，(1)送信データのフラグメンテーションを行い，(2)各フラグメント長の領域を PC 内メモリの一部に確保し，(3)各フラグメントをその領域にコピーし，(4)NI にフラグメント毎の送信要求を発行する．(1)の PC 内メモリの一部とは，OS が起動する前に予約する巨大な連続領域であり，予約領域とよぶことにする．その予約領域は，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，OS のメモリ管理の対象外となる [6]．また，フラグメント長として，Maestro ネットワークで最適な 4048byte を用いている．このフラグメント長に関しては，実験的に最適な長さを決定する．6 章で示す実験において，この長さにおける通信性能の比較を行い，最適なフラグメント長の決定に関する言及を行う．MP_receive()は NI が予約領域に書き込んだフラグメントを，アプリケーションプログラムの計算領域にコピーし，連続したデータに再構成する．MP_receive()はノンブロックで実行され，MP_wait()で，その完了が確認される．MP_barrier()は，同期を行うために用いられる関数である．

4.3 MMP ライブラリを用いた通信の流れ

本節では，以上で議論を行ってきた MMP ライブラリを用いた送受信操作を説明する．

図 3 に示すように，PC0 と PC1 の間で，通信を行う場合を考える．図では簡単のため SB は示していない．PC0 が，PC1 に対して 8000byte のデータを送信すると仮定する．

まず，送信側の PC0 では，アプリケーションプログラム中で送信すべきデータが MP_send()に渡される．MP_send()は，そのデータを一時的に格納する領域を予約領域に確保する．このとき，フラグメント長は 4048byte なので，8000byte のデータは，4048byte と 3952byte の 2 つに分割される．その後，MP_send()は，分割したそれぞれのフラグメントに対し，(1)予約領域コピーを行い（図 3(a)），(2)NI に送信要求を発行する（図 3 (b)）．この(1)(2)の操作を，全てのフラグメントが完了するまで繰り返す．(1)のコピー操作が PC で行われているとき，NI は，要求を読み出し（図 3 (b')），予約領域からフラグメントをネットワークバッファへと転送する（図 3 (c)）．この PC でのコピー操作と NI での転送操作がオーバラップし，高いスループットを実現できる．

NI は，ネットワークバッファへの転送が完了すると，予約領域に確保されたフラグメント領域の解放を PC に許すための返答を送信返答バッファへと書き込む（図 3 (d)）．PC はそれを読み出し，フラグメント領域の解放を行う（図 3 (d')）．

フラグメントは SB でルート解析が行われ，PC1 へと転送される（図 3 (e)）．その後，PC1 の NI は受

信操作を起動する．NIはフラグメントのための受信領域中に自律的に確保し，PC0における送信手順と逆の順で転送する（図 3 (f)(g)(g')）．この受信領域の自律的な配置操作により，NIはPCに割り込みを用いずにフラグメントの受信を要求するので，PCでの計算に影響を与えない．

PC1では，MP_wait()が，複数のフラグメントをMP_receive()により受信要求した量の連続データに再構成する（図 3 (h)）．ここでは2つのフラグメントが再構成され8000byteのデータを作成する．PC1はフラグメント毎に完了のための返答を受信返答バッファに書き込む（図 3 (i')）．この返答はNIにより読み出され，予約領域に確保されたフラグメント領域の解放を行う．

PC1の受信操作に関しても，アプリケーションプログラム中でのデータ再構成（図 3(h)）と，NIからPCへのフラグメントの転送（図 3(f)）が最適なフラグメント長でパイプライン化され，通信の手順を多重化している．

5 評価

上述のMMPに採用した高速化技法を評価するために，基本性能とフラグメンテーションの最適化に関する通信実験を行う．これらの実験には，2台のPC間でデータを送受信し，受信側から返される4byteの返答を送信側が受け取るまでのRound Trip Timeを用いる．さらに，通信インターフェースの自律的な受信領域確保と，Maestroネットワークの全体性能評価を行う．以上の実験環境を表 1に示す．

5.1 基本性能

図 4にMMPとMaestroネットワークの基本性能についての比較を示す．Maestroネットワークの基本性能とは，一方のNIがPCの予約領域にあるデータを読み出し，もう一方のNIがPCの予約領域にそれを書き込んだのち，送信側NIが受信側NIからの受信確認を受け取るまでの時間である．

MMPのネットワーク利用率は，4byte転送の時，最悪で，Maestroネットワークの基本性能の30%である．しかし，大きなデータに対しては，高速化技法が有効に働き，Maestroネットワークの基本性能の96%を達成している．

5.2 フラグメンテーションの最適化

この実験では，送受信時に行われるフラグメンテーションにおけるフラグメント長を変更し，送信操作のパイプライン化の効果を考察する．

図 5にフラグメント長を，我々の用いた4048byte，短い場合の223byte，長い場合の1048528byte，の3つの比較を示す．

スループットについて，我々の用いた最適なフラグメント長である4048byteに比べ，短い場合については48%，長い場合については80%しか達成できない．この実験から，送受信操作をパイプライン化するためのフラグメント長はシステム毎に固有の最適値があることがわかる．すなわち，高い実効スループットを得るためには実験的なフラグメント長の決定が重要であることがわかる．

6 アプリケーション

通信インターフェースの自律的な受信領域確保と共に，Maestroネットワークに用いられている高速化技法の総合的な評価を行うため，NAS Parallel Benchmarkから，IS，CGの評価を行う．ベンチマークの問題サイズは，Class Aを用いる．Maestroネットワークと比較するネットワークとして100BASE-TX Ethernet上でのMPIを用いる．

ISは整数列に対しRadix Sortingを行う． 図 6にISの結果を示す．1台実行の時間が2台の実行時間よりも短いのは，それぞれのネットワークの通信スループットがボトルネックになっているからである．グラフから，Maestroネットワークは Ethernetの性能より，最高で約 1.4 倍の性能を得ている．これは，最適なフラグメンテーションを用いたパイプライン化により，大量のデータ転送における安定したスループットを供給できているからである．

CG は正値対称な大規模疎行列の最小固有値を求めるための共役勾配法を実行する．CG ではノンブロック通信を用いたバタフライ転送が行われる．図 7にCGの結果を示す．Maestroネットワークでは受信時にNIが独立動作し，PCにおける計算との重ね合わせが実現できているため，実行時間が短縮できたと考えられる．一方，Ethernet では，受信時に通信インターフェースからPCへの割り込みが発生するため，PCでの計算が中断されるために，通信と計算が逐次的になり，8台時には4台時に比して実行時間の増加が見られる．

7 おわりに

我々は，リンク層におけるバースト転送と送信単位の縮小化による通信の多重化を図ったPCクラスタ向けネットワークを有するMaestroクラスタシステムを開発している．

本論文ではMaestroネットワーク上に実装したメッセージパッシングライブラリMMPの構成と評価について述べた．MMPの実装にあたって通信プロトコルソフトウェアとデバイスハンドラに施した最適化について詳しく述べた．

MMP を用いた通信実験により，我々の提案する高速化技法のうち，(1)フラグメンテーションの最適化と，(2)通信インターフェースによる自律的な受信領域確保が有効であることを示した．さらに，本システムの全体評価を行うため，NAS Parallel Benchmark から IS，CG を用いて評価を行い，良好な結果が得られたことを示した．

表 1　実験環境

PC	Pentium II Cerelon 400MHz SDRAM 128Mbyte(内，32Mbyte は予約領域)
OS	Linux kernel ver. 2.0.36
MPI	MPICH implementation Ver. 1.1.1

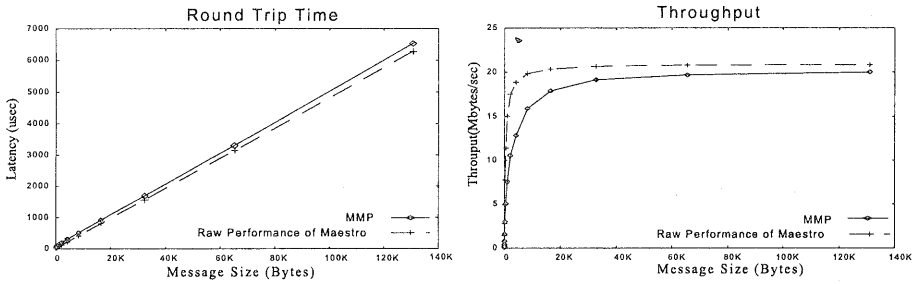

図 4　基本性能

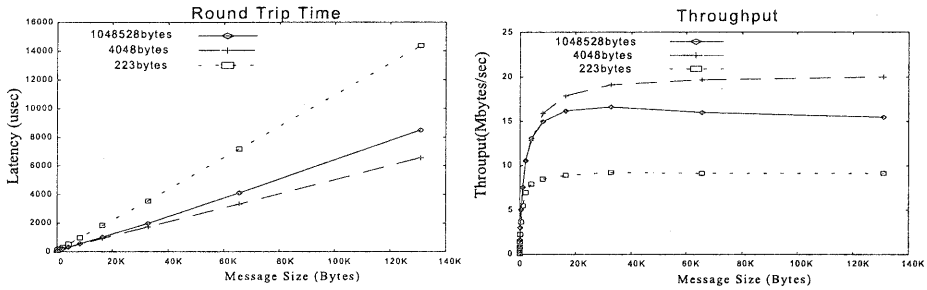

図 5　フラグメンテーションの最適化

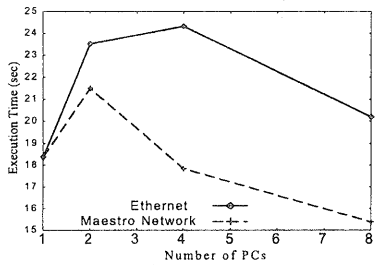

図 6　IS の結果

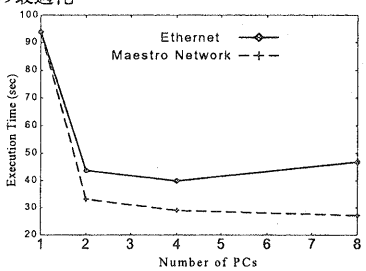

図 7　CG の結果

参　考　文　献

1) D. Bailey et. al., *THE NAS PARALLEL BENCHMARKS*, RNR Technical Report RNR-94-007, 1994

2) Rajkumar Buyya , *High Performance Cluster Computing: Architectures and Systems*, Prentice Hall.

3) http://www.1394ta.org

4) V. Karamcheti and A. Chien, *Software Overhead in Messaging Layers: Where Does the Time Go?*, Proceedings of International Conference on Architectural Support of Programming Languages and Operating Systems (ASPLOS-VI), 1994.

5) PLX Technology, *PCI9060 Data Sheet VERSION1.2*, December 1995.

6) Alessandro Rubini, Andy Oram, *Linux Device Drivers*, Chapter 13, O'Reilly & Associates.

7) IEEE Task Force on Cluster Computing, http://www.dgs.monash.edu.au/~rajkumar/tfcc/

8) Shinichi Yamagiwa, Masaaki Ono, Takeshi Yamazaki, Pusit Kulkasem and Koichi Wada, *Maestro-Link: A High Performance Interconnect for PC Cluster*, LNCS 1482 Field Programmable Logic and Applications, 1998.