計算機アーキテクチャ 128－5
ハイパフォーマンス
コンピューティング 70－5
（1998．3．5）

RWC PC Cluster II の構築と性能評価

手塚宏史† 堀敦史†
Francis O'Carroll†† 石川裕†

我々は Pentium Pro 200MHz の PC 64 台を Myrinet ギガビットネットワークによって接続した PC クラスタシステム "RWC PC Cluster II" を構築し，その上にマルチユーザの並列プログラミング環境 SCore を開発している．PCC2 上の通信ライブラリ PM は通常のメッセージ転送だけでなくリモートメモリライトによるゼロコピーデータ転送をサポートしており，約 119M バイト / 秒 (8K バイトデータ) のメッセージ転送バンド幅と約 109M バイト / 秒 (同 8K バイト) のリモートメモリライトバンド幅，および約 7.5 マイクロ秒 (同 8 バイト) の通信レイテンシを持っている．また，PM のこれらの機能を用いた MPI/PM は PCC2 上で約 104M バイト (同 1M バイト) のデータ転送バンド幅と約 11 マイクロ秒 (同 8 バイト) の通信レイテンシを得ている．MPI/PM を用いた NAS パラレルベンチマークの結果によって，PCC2 の高い性能とスケーラビリティが実証された．

RWC PC Cluster II and its performance

Hiroshi Tezuka,† Atsushi Hori,† Francis O'Carroll††
and Yutaka Ishikawa†

We have built a PC cluster "RWC PC Cluster II" consisting 64 Pentium Pro 200MHz PCs connected by a Myrinet giga-bit network, and have been developing a multi-user parallel programming environment SCore on it. A communication library PM on PCC2 supports a message passing and a remote memory write using zero-copy data transfer. PM achieves 119M bytes/s(8K byte data) message passing bandwidth, 109M bytes/s(8K byte data) remote memory write bandwidth and 7.5 micro second communication latency. MPI/PM that uses these PM facilities achieves 104M bytes/s(1M byte data) data transfer bandwidth and 11 micro second communication latency on PCC2. The NAS parallel benchmark results using MPI/PM have shown PCC2's high performance and scalability.

1. はじめに

一般に市販されているワークステーション，パーソナルコンピュータの性能向上とギガビット / 秒クラスの高速なネットワークの登場によって，高性能なワークステーション /PC クラスタを容易に構築することが可能になった．このようなクラスタシステムでは構成要素に市販されている製品を用いることよって，短期間でシステムを構築することができるため，CPU の性能向上，メモリ，ハードディスクの大容量化などの技術の進歩にキャッチアップしやすいという利点を持っている．当研究室では，以前から並列計算機としてのクラスタシステムに着目し，いくつかのワークステーション /PC クラスタシステムを構築して，その上にマルチユーザ並列プログラミング環境 SCore の開発を行なってきた．

今回，我々は Pentium Pro 200MHz の PC 64 台をギガビット LAN のひとつである Myrinet[1] で接続した PC クラスタシステム RWC PC Cluster II (以下 PCC2 と呼ぶ) を構築し，その性能評価を行なった．

本稿では，まず 2 節で PCC2 のハードウェア構成を述べ，3 節でソフトウェア構成ついて述べる．次に 4 節で PCC2 の性能評価について述べ，5 節でまとめと今後の予定を述べる．

2. ハードウェア

図 1 に PCC2 の概観を示す．PCC2 はそれぞれ 32 ノードを格納したラック 2 本から構成されている．実装密度を高めるために，PC を収めるシャーシのみ特注しているが，内部の電気部品は全て市販品を用いている．PCC2 のハードウェアは a) 64 個のノード PC，b) 2

† 技術研究組合 新情報処理開発機構 つくば研究センタ
並列分散システムソフトウェアつくば研究室
Parallel & Distributed System Software Laboratory
TRC,
Real World Computing Partnership
http://www.rwcp.or.jp
†† エム・アール・アイシステムズ (株)

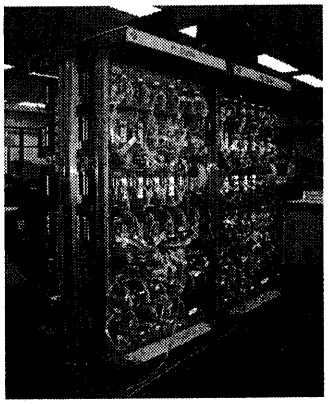

図 1　RWC PC Cluster II

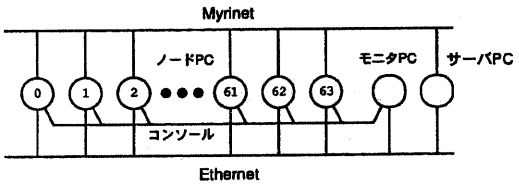

図 2　RWC PC Cluster II の構成

個のモニタ PC, c) 3 個のサーバ PC, d) Myrinet ネットワーク, e) Ethernet ネットワークから構成されている (図 2).

ノード PC は並列プログラムの計算ノードとなる PC で, 各ノード PC は, 図 3 のように, CPU カード, Ethernet カード, Myrinet カード, ハードディスクドライブから構成されている.

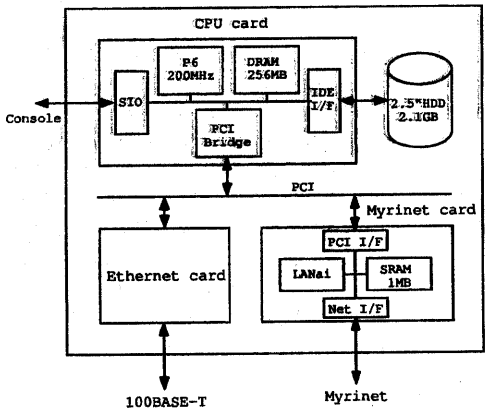

図 3　ノード PC の構成

CPU カードは PICMG[2] 規格の産業用 PC 互換 CPU カードで, Pentium Pro (200MHz 512K バイトキャッシュ), 256M バイト ECC 付き EDO DRAM, EIDE インタフェース, シリアルインタフェース, その他の I/O を搭載している. PICMG 規格の CPU カードを採用したのは, 主に実装密度と保守性を考慮したためである.

各ノード PC には 2.1G バイト (2.5 インチ IDE インタフェース) のハードディスクドライブを搭載している.

ひとつのシャーシには 2 個のノード PC が収められてひとつの電源を共有しており, 16 個のシャーシがひとつのラックに格納されて 32 ノード / ラックとなる.

モニタ PC は, 主にノード PC のシリアルコンソールとなる PC で, そのための 32 チャネルのシリアルインタフェースを持ち, Myrinet ネットワークには接続されていない. サーバ PC はキーボード / マウス, CRT ディスプレイ, 4.3G バイトのハードディスクを持った独立した PC で, 計算結果を X Window System 上に表示したり, クラスタ内の NFS サーバとなる. ノード PC とは Myrinet ネットワークで接続されている.

Myrinet は Myricom[1] 社が開発したギガビットネットワークで 160M バイト / 秒 双方向のリンクを持ち, PCC2 では並列プログラムのノード間通信ネットワークとして用いている. PCC2 の Myrinet ネットワークの構成は図 4 のようになっており, 64 ノードのクラスタ全体のバイセクションバンド幅は 5120M バイト / 秒である.

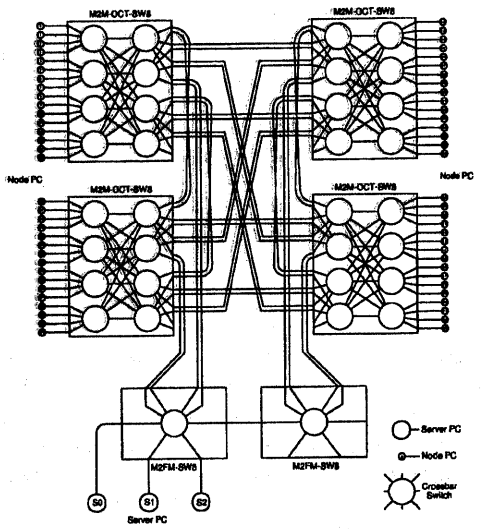

図 4　Myrinet ネットワークの構成

Myrinet ネットワークは, 各ノード / サーバ PC 内のホストインタフェースカードと, ネットワークを構成するスイッチから構成されている.

Myrinet のホストインタフェースは, プロトコルハンドリングを行なう 32 ビットの RISC プロセッサ

(LANai) と，データ転送のバッファ領域としても用いられる 1M バイトの SRAM を搭載している．LANai プロセッサ用のプログラム開発環境が Myricom 社によって提供されており，SRAM 内のプログラムを変更することによって独自のプロトコルを実装できる．

Myrinet スイッチは 8 ポートのクロスバスイッチ LSI によるスイッチで，ホストインタフェース間のルーティングに用いられる．複数のスイッチ間を接続することによってより複雑なネットワークを構成することができる．

PCC2 の Ethernet(100BASE-T) ネットワークは TCP/IP による通信に用いられる．

3. ソフトウェア

PCC2 上のマルチユーザ並列プログラミング環境 SCore は以下のものから構成されている (図 5)．

- ローカル・オペレーティングシステム
- PM 通信ライブラリ
- MPC++(MTTL)
- SCore-D マルチユーザ並列オペレーティングシステム
- MPI/PM

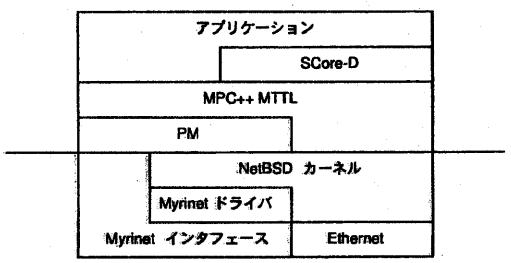

図 5 SCore の構成

以下，これらについて述べる．

3.1 ローカル・オペレーティングシステム

各 PC 上のオペレーティングシステムには，NetBSD/i386 1.2.1[3] に以下のような変更を行なったものを用いている．

- Myrinet デバイスドライバ，およびそのために必要なカーネル関数を追加．
- 256MB の主記憶をサポートするために，カーネルのロードアドレスを変更．
- バスマスタモード IDE ドライバを追加．
- いくつかのバグ修正．

ローカル・オペレーティングシステムに NetBSD を用いることで，通常用いている Unix ワークステーションと同様のソフトウェア環境を PC クラスタ上に構築することができる．

3.2 PM 通信ライブラリ

PM[4)~6)] は Myrinet 用のメッセージライブラリで，信頼性のある非同期メッセージ通信をサポートしている．PM は Myrinet インタフェースをユーザプログラムが直接アクセスしてデータ転送を行なうため，システムコール，割り込みなどのオーバヘッドがなく，低い通信レイテンシと高いデータ転送バンド幅を実現している．さらに，PM は以下のような機能を持っている．

リモートメモリアクセス
送信ノードのユーザアドレス空間のデータを受信ノードのアドレス空間にゼロコピーで直接書き込む，リモートメモリライトの機能をサポートしている．リモートメモリアクセスでは，メッセージ転送で必要となる主記憶内のデータコピーが不要であるため，高いバンド幅のデータ転送が可能となる．ゼロコピーのリモートメモリアクセスを行なうためには，ユーザデータを物理メモリにピンダウンしておく必要があるが，PM ではピンダウンによるオーバヘッドを軽減するために，ピンダウンの解除を実際に必要になるまで遅らせるピンダウンキャッシュ[7),8)] の手法を用いている．

複数の通信チャネル
複数の独立した通信チャネルを持ち，複数のプロセス / スレッドが同時に Myrinet ネットワークを使用することができる．

ネットワークコンテキストスイッチ
ギャングスケジューリングをサポートするために，各通信チャネルのコンテキストを退避 / 回復する機能をサポートしている．

通信負荷の分散
図 4 のようにスイッチ間が複数のリンクで結ばれている場合には，それぞれのリンクに通信負荷を分散させることによって，システム全体のスループットを高めることができる．

3.3 MPC++(MTTL)

MPC++[9)~11)] は C++ を並列プログラミング向けに拡張した言語で，現在の実装 (Level 0) では C++ のテンプレートライブラリを用いて，MTTL(Multi-Thread Template Library) として実装されている．MTTL はリモート関数呼び出し，リモートメモリアクセス，同期変数などの機能をサポートしている．

3.4 SCore-D

SCore-D[12)~15)] は MPC++ で記述された並列オペレーティングシステムで，Time Space Sharing Scheduling[16)] によって複数のユーザプロセスのギャングスケジューリングを行ない，PM のユーザレベル通信による高性能なマルチユーザ環境を実現している．また，SCore-D は Myrinet を介した I/O，アイドル検出[17)]，ユーザプロセス毎のロードモニタ (図 6) などをサポートしている．SCore-D はローカル・オペレーティングシステムである NetBSD 上のデーモンプロセ

スとして実装されている．

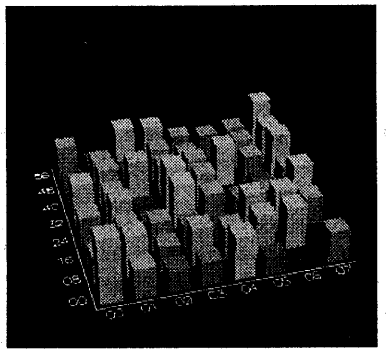

図 6 ロードモニタ

3.5 MPI/PM

MPI/PM[18] は，ポータブルな MPI の実装である MPICH[19] を PM 上に移植したものである．MPI/PM は PM のメッセージ転送機能とリモートメモリアクセス機能を用いることによって，低い通信レイテンシと高いデータ転送バンド幅を実現している．

4. 性能評価

ここでは PM および MPI/PM によるノード間通信性能の評価結果と NAS パラレルベンチマーク[20] によるクラスタシステム全体の性能評価，および SCore-D のギャングスケジューリングのオーバヘッドの評価について述べる．

4.1 PM の性能評価

PM のデータ転送のバンド幅の測定結果を図 7 に示す．この計測ではデータサイズを 4 バイトから 1M バイト*まで変えながら，1 方向のバースト転送を行なって，その経過時間と転送したバイト数からデータ転送バンド幅を求めた．ここで，"msg" は PM のメッセージバッファ間のデータ転送を行なった場合のバンド幅で，8K バイト以上のデータサイズの場合に約 119M バイト / 秒のバンド幅を得ている．また，"msg+copy" は a) 送信ノードがユーザデータ領域からメッセージバッファにコピー，b) メッセージ転送，c) 受信ノードがメッセージバッファからユーザデータ領域にコピー，を行なった場合のバンド幅で，8K バイト時に約 50M バイト / 秒である．このように PM は高いデータ転送バンド幅を持っているが，主記憶内のデータコピーが発生すると，バンド幅が低く押えられてしまうことが示されている．

次に，"rma" は PM のリモートメモリライトのバンド幅で，この場合はメッセージ転送の場合とは異なり，

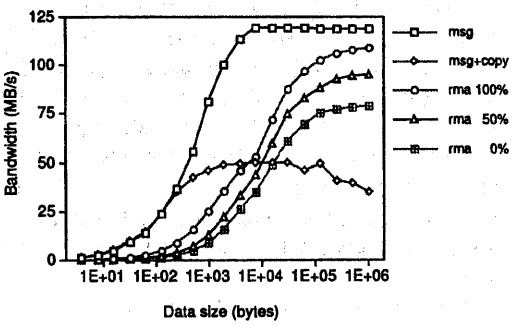

図 7 PM のデータ転送バンド幅

各データ転送開始時に，a) 送信ノードのピンダウン，b) 受信ノードへの送信要求のメッセージ転送，c) 受信ノードのピンダウン，d) 送信ノードへの送信可能通知のメッセージ転送，の手順のネゴシエーションを行なっている．また，ピンダウンキャッシュの効果を見るために，ピンダウンキャッシュのヒット率を強制的に 100%，50%，0% と変化させた場合のバンド幅を測定した．

ネゴシエーションのオーバヘッドのため，リモートメモリライトのデータ転送バンド幅は，データコピーなしのメッセージ転送の場合よりも低くなっているが，約 8K バイトよりも大きいところではデータコピーを伴うメッセージ転送の場合よりも高いバンド幅が得られている．PM のリモートメモリライトのバンド幅の最大値は，ピンダウンキャッシュのヒット率が 100% の時に約 109.4M バイト / 秒 (データサイズ 8K バイト)，ヒット率が 0% の時に約 78.8M バイト / 秒 (同 1M バイト) である．

なお，2 ノード間のピンポン転送を用いて計測した PM の一方向通信レイテンシは約 7.5 マイクロ秒 (8 バイトデータ) である．

4.2 MPI/PM の性能評価

MPI/PM のデータ転送バンド幅の測定結果を図 8 に示す．この測定ではデータサイズを 4 バイトから 1M バイトまで変えながら，送信ノードは MPI_Send() を，受信ノードは MPI_Recv() を繰り返して，その経過時間と転送したバイト数からバンド幅を計算した．また，リモートメモリライト時のピンダウンキャッシュの効果を見るために，ピンダウンキャッシュのヒット率を強制的に 100% および 0% にして計測を行なった．

データサイズが 8K バイトのところでバンド幅のグラフに段があるのは，MPI/PM ではデータサイズが 8K バイト以下の場合は PM のメッセージ転送によって，8K バイトを超える場合は PM のリモートメモリライトによってデータ転送を行なっているからである．このように MPI/PM では，データ長によって両者を使い分けることによって，全データサイズに渡って高いデータ転送バンド幅を得ている．MPI/PM のバンド幅の最大

* PM の MTU は約 8K バイトなので，それを超える大きさのデータは 8K に分割して転送している．

値はピンダウンキャッシュのヒット率が 100% の時に約 103.9M バイト / 秒 (データサイズ 1M バイト) およびヒット率が 0% の時に約 76.7M バイト / 秒 (同 1M バイト) である.

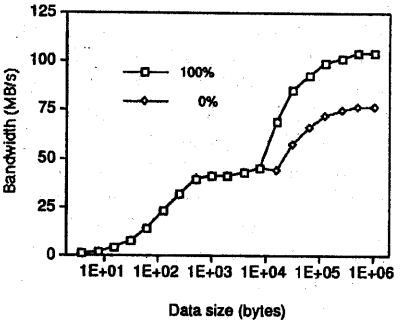

図 8 MPI/PM のバンド幅

なお, 2 ノード間のピンポン転送を用いて計測した MPI/PM の一方向通信レイテンシは約 11 マイクロ秒 (8 バイトデータ) である.

4.3 NAS パラレルベンチマーク

MPI を用いた NAS パラレルベンチマーク (NPB2.3 64 ノード) の結果 (トータル性能) を表 1 に, ノード数を変化させた場合の 1 ノード当たりの性能の変化を図 9 に示す.

表 1 NPB2.3 の結果 (64 ノード, Mops/s)

Class	BT	CG	EP	FT
A	1672.65	457.33	29.95	942.87
B	1473.02	528.94	29.95	969.80
C	1486.40	595.16	29.95	1028.33
Class	IS	LU	MG	SP
A	82.76	2101.70	1526.89	1159.33
B	76.34	1995.99	1642.17	1174.42
C	70.35	2220.03	1650.40	1122.32

図 9 の縦軸はノード数が少ない場合の 1 ノード当たりの性能との相対値である. PCC2 は各ノードに 256MB のメモリを搭載しているが, NPB2.3 Class A のいくつかのプログラムはデータ量が大きく, ノード数が少ないと実行できないため, 性能の基準となるノード数には, ベンチマークを実行可能な最小のノード数として, CG, EP, LU, SP は 1 ノード, FT, IS, MG は 2 ノード, BT は 4 ノードの値を用いた.

図 9 に示されているように, 性能に対するノード間通信の影響が大きい CG, IS を除いて, ノード数を増やすことによる 1 ノード当たりの性能の低下はほとんど見られない.

次に, システムのバイセクションバンド幅が NPB2.3 (Class A, 64 ノード) の性能に与える影響を計測した結果を図 10 に示す. この計測は PM のコンフィギュレーションファイルのスイッチ間リンクの記述をコメントアウトすることで, そのリンクを使わないようにして行なった. 特に通信性能の影響が大きい IS ではバイセクションバンド幅が低下するに従って性能も大幅に低下しており, 高性能なクラスタシステムを構築するためには, バンド幅の高いノード間通信ネットワークが必要であることが示された.

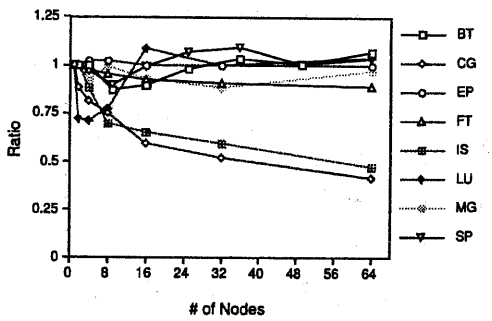

図 9 NPB2.3 スケーラビリティ

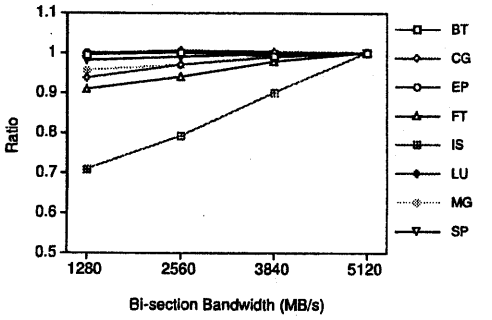

図 10 NPB2.3 バイセクションバンド幅の影響

4.4 ギャングスケジューリングのオーバヘッド

SCore-D のギャングスケジューリングのオーバヘッドの測定結果を図 11 に示す. これは, NAS パラレルベンチマークのうち, CG, EP, FT について, SCore-D の下で 100 ミリ秒ごとにギャングスケジューリングを行ないながら動かした場合の, SCore-D を用いない場合の性能に対するスローダウンの割合を計測したものである.

このように, SCore-D のギャングスケジューリングのオーバヘッドは 64 ノードの場合で約 3.3% 程度であり, 充分実用的な範囲であると考えられる.

5. まとめと今後の予定

我々は, 市販の産業用 PC(Pentium Pro 200MHz) 64 個を Myrinet ギガビットネットワークで接続した PC クラスタシステム RWC PC Cluster II を構築し

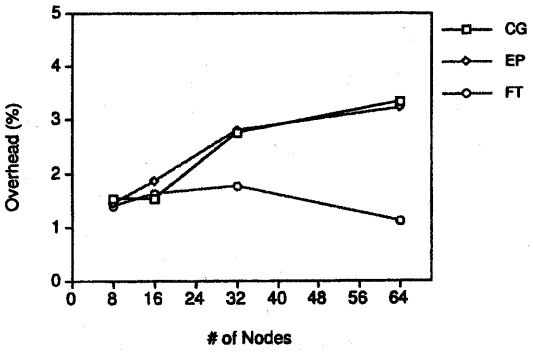

図11 ギャングスケジューリングのオーバヘッド

た.

PCC2 で用いている通信ライブラリ PM は Myrinet インタフェースを直接制御することによって，低い通信レイテンシと高いデータ転送バンド幅を実現している．MPI/PM は PM のメッセージ転送，リモートメモリライトの機能を用いることによって，同様に低い通信レイテンシと高いデータ転送バンド幅を得ている．NAS パラレルベンチマークの結果から，PCC2 は高い性能とスケーラビリティを持っていることが実証された．

現在の PCC2 は 64 ノード構成であるが，近い将来にこれを 128 ノードに拡張する予定である．また，ローカル・オペレーティングシステムについても，現在の NetBSD から，より広く用いられている Linux に切替える予定である．

参 考 文 献

1) http://www.myri.com.
2) http://www.picmg.com.
3) http://www.netbsd.org.
4) 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. 並列処理シンポジウム JSPP'96. 情報処理学会, June 1996.
5) Hiroshi Tezuka, Atsushi Hori, Yutaka Ishikawa, and Mitsuhisa Sato. PM: An Operating System Coordinated High Performance Communication Library. In *High-Performance Computing and Networking '97*, 1997.
6) http://www.rwcp.or.jp/lab/pdslab/pm/home.html.
7) 手塚, 堀, O'Carroll, 原田, 石川. ピンダウンキャッシュを用いたユーザレベルゼロコピー通信. 情報処理学会研究報告. 情報処理学会, August 1997.
8) Hiroshi Tezuka, Francis O'Carroll, Atsushi Hori, and Yutaka Ishikawa. Pin-down Cache: A Virtual Memory Management Technique for Zero-copy Communication. In *IPPS'98*, April 1998.
9) Y. Ishikawa, A. Hori, H. Konaka, M. Maeda and T. Tomokiyo. "MPC++: A Parallel Programming Language and Its Parallel Object Support". In *Proc. OOPSLA '93 Workshop on Efficient Implementation of Concurrent Object-Oriented Languages*, pp. J1–J5, 1993.
10) 石川裕, 堀敦史, 小中裕喜, 前田宗則, 友清孝志. 並列プログラミング言語 MPC++ の実現. 並列処理シンポジウム JSPP'94, pp. 105–112, 1994.
11) Yutaka Ishikawa. Multi Thread Template Library – MPC++ Version 2.0 Level 0 Document –. Technical Report TR–96012, RWC, September 1996. *This technial report is obtained via http://www.rwcp.or.jp/lab/mpslab/mpc++/mpc++.html.*
12) A. Hori, H. Tezuka, Y. Ishikawa, N. Soda, H. Konaka, and M. Maeda. Implementation of Gang-Scheduling on Workstation Cluster. In D. G. Feitelson and L. Rudolph, editors, *IPPS'96 Workshop on Job Scheduling Strategies for Parallel Processing*, Vol. 1162 of *Lecture Notes in Computer Science*, pp. 76–83. Springer-Verlag, April 1996.
13) 堀敦史, 手塚宏史, 石川裕, 曽田哲之, 小中裕喜, 前田宗則. 並列プログラム実行環境のワークステーションクラスタ上での実装. 並列処理シンポジウム JSPP'96. 情報処理学会, June 1996.
14) 堀敦史, 手塚宏史, 石川裕, 曽田哲之, 原田浩, 古田敦, 山田努, 岡靖裕. ワークステーションクラスタにおける並列プログラミング環境の実現. システムソフトウェアとオペレーティングシステム研究会資料, 96-OS-73, pp. 121–126. 情報処理学会, August 1996.
15) Atsushi Hori, Hiroshi Tezuka, and Yutaka Ishikawa. User-level Parallel Operating System for Clustered Commodity Computers. In *Cluster Computing Conference '97*, March 1997.
16) 堀敦史, 石川裕, 小中裕喜, 前田宗則, 友清孝志. 超並列オペレーティングシステムにおけるスケジューリング方式の提案. システムソフトウェアとオペレーティング・システム研究会資料, 94-OS-63, pp. 25–32. 情報処理学会, March 1994.
17) Atsushi Hori, Hiroshi Tezuka, and Yutaka Ishikawa. Global State Detection using Network Preemption. In *IPPS'97 Workshop on Job Scheduling Strategies for Parallel Processing*, April 1997.
18) Francis B. O'Carroll, Atsushi Hori, Hiroshi Tezuka, Yutaka Ishikawa, and Satoshi Matsuoka. Performance of MPI on Workstation/PC Clusters using Myrinet. In *Cluster Computing Conference 1997*, 1997.
19) http://www.mcs.anl.gov/home/lusk/mpich/index.html.
20) http://science.nas.nasa.gov/Software/NPB/.