計算機アーキテクチャ　147−11
ハイパフォーマンスコンピューティング　89−11
（2002．3．7）

境界データ同期型領域分割法を用いたPCクラスタ並列処理のための通信ライブラリの設計と評価

中井 甚輔†, 阿部 眞‡, 布目 淳†, 平田 博章†, 柴山 潔†

† 京都工芸繊維大学 工芸学部 電子情報工学科
〒606-8585 京都市左京区松ヶ崎御所海道町
‡ 株式会社 村田製作所 技術開発本部 第4開発グループ
〒617-8555 京都府長岡京市天神2-26-10

本稿では「境界データ同期型領域分割法」向きの通信ライブラリを設計し, ギガビットイーサネットでパーソナルコンピュータ (PC) をピアツーピア接続したPCクラスタ上で評価を行う. カーネルアドレス空間にアプリケーションの解析データ領域を確保し, カーネルメモリマップ機能でユーザアドレス空間に対応づけることで, ゼロコピー通信の実現を図る. 本稿のゼロコピー通信ライブラリを用いたアプリケーションをPCクラスタ上で実行した評価結果より, TCP/IPに比べ通信処理時間が約50%短縮されることを確認した.

Design and Evaluation of the Communication Library for a Parallel Processing using the Domain Decomposition Method with Boundary Data Synchronization on PC Cluster

Jinsuke Nakai†, Shin Abe‡, Atsushi Nunome†,
Hiroaki Hirata†, and Kiyoshi Shibayama†

† Dept. of Electronics and Information Science, Kyoto Institute of Technology
Matsugasaki, Sakyo-ku, Kyoto, 606-8585 JAPAN
‡ Development Group IV, Research & Development Division, Murata Manufacturing Co., Ltd.
26-10, Tenjin 2-Chome, Nagaokakyo, Kyoto, 617-8555 JAPAN

This paper presents the high performance communication library for the Domain Decomposition Method with Boundary Data Synchronization (DDM-BDS) on PC's which are connected directly to their neighbour nodes by Gigabit Ethernet. Mapping in-kernel data region to the user address space enables the zero copy communication. We implement and estimate this communication library on our PC cluster. Compared with TCP/IP, our evaluation results show that our library reduces the communication time by about 50%.

1　はじめに

著しい技術の進歩により, パーソナルコンピュータ (PC) の計算性能, 通信性能が向上してきたことで, 複数のPCをネットワークで接続した仮想的な並列計算機であるPCクラスタの応用範囲が広がっている. 1997年にFast Ethernetを用いたPCクラスタ上で電磁界解析を並列処理する方式について報告したが [1], 電子部品等の設計現場でも, 設計対象となる構造が年々複雑となり, 3次元解析を実用的な時間で行うことはできなくなってきた.

近年, Gigabit Ethernetの普及により通信装置の

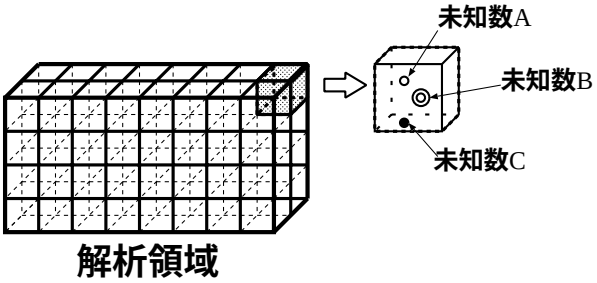

図 1: 解析領域と未知数

性能がさらに向上した. しかし, TCP/IP を用いた通信の場合, プロセッサによるメモリコピー処理等がオーバヘッドになり, Gigabit Ethernet のハードウェアの性能を引き出すことができない.

そこで我々は, プログラムを,「解析対象となる領域を複数に分割し, 並列に解析するとともに, 分割した領域の境界のデータの同期を取る」ことで計算できるアプリケーションを対象に GigabitEthernet を用いたゼロコピー通信ライブラリを設計し PC クラスタに実装した.

2 境界データ同期型領域分割法

電磁場等の場の数値解析において, 図 1に示すように解析対象となる解析領域内に多数の未知数 (以下「解析データ」と呼ぶ) を離散的に配置し, 解析データ間の関係式を用いて解析を行う手法がある. 中でも, 図 2に示すように境界領域をオーバーラップさせて, 解析領域を複数に分割し, 分割した各領域毎に PC を割当てて並列計算を行う手法を領域分割法と呼ぶ. 境界領域にある解析データ (以下, 境界データ) の計算可否は, 表 1に示すように 3 通りのケースが考えられる. PC 間で境界データの同期をとり, 各 PC では計算ができない境界データの整合をとる手法を, 本稿では「境界データ同期型領域分割法 (**DDM-BDS** : Domain Decomposition Method with Boundary Data Syncronization) 」と呼ぶことにする. また, DDM-BDS を用いるプログラムを「DDM-BDS プログラム」と呼び, 境界データのうち PC 間でデータの同期をとる際に送受信されるデータを「境界更新データ」と呼ぶことにする. 図 3に 3 台の PC 上で DDM-BDS を用いた場合のアルゴリズム例を示す.

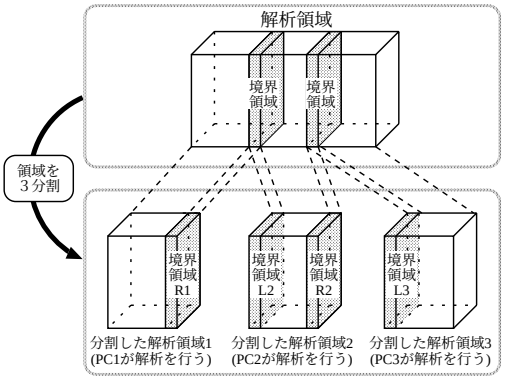

図 2: 領域分割と境界

表 1: 境界領域と計算可否

	境界 R	境界 L	境界以外
未知数 A	不可	可	可
未知数 B	可	不可	可
未知数 C	可	可	可

3 通信ライブラリの設計

本稿で開発した境界データ同期型通信ライブラリ (以後 BDS ライブラリ) を用いると, ゼロコピー通信を行うことができる. 本節では, DDM-BDS 用 PC クラスタの構成および BDS ライブラリにおけるゼロコピー通信の実現方法について述べる.

3.1 PC クラスタのネットワーク構成

我々は, DDM-BDS プログラムを実行する環境として 1 次元に PC を連続して接続した PC クラスタを想定している. BDS ライブラリを実装する PC クラスタの例として, 図 4に 3 台の PC から成る PC クラスタのネットワーク構成を示す. Gigabit Ethernet はハードウェアレベルでメッセージ転送

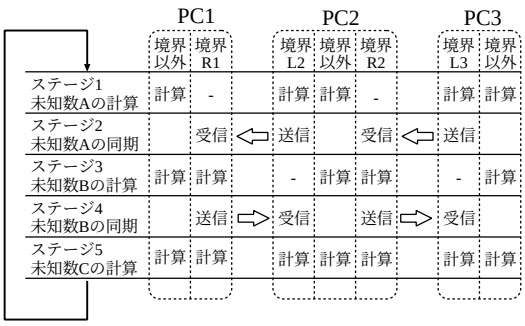

図 3: DDM-BDS を用いた並列処理アルゴリズム例

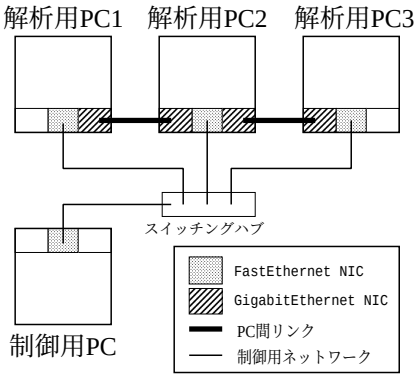

図 4: PC クラスタのネットワーク構成

の信頼性を保証しない. そこで, データ転送の信頼性の保証を実現するために, データ転送時の衝突が起こらないよう PC 間にピアツーピアのリンクを用意した. このリンクは境界領域のデータを通信するために利用する. プログラム実行の監視等の制御用メッセージ通信には, スイッチングハブ経由の制御用ネットワークを用いる.

3.2 ゼロコピー通信方式の実現

本方式では, NIC が境界領域の境界更新データを直接 DMA 転送できるように, カーネルアドレス空間に解析領域を確保する. この解析領域をユーザアドレス空間にメモリマッピングすることで, ゼロコピー通信を実現している.

ディスクリプタの設定: 図 5に示すように, 境界更新データテーブルを作成し, 境界更新データの物理アドレスとサイズを登録する[1]. そして, ネットワークドライバの送信用と受信用のそれぞれのディスクリプタに, 境界更新データテーブルの物理アドレスとサイズを設定する. NIC はこのディスクリプタを参照して DMA 転送を行い, これにより, 境界領域と NIC の間で境界更新データを DMA 転送できる. DDM-BDS では, ある境界更新データの物理アドレスとサイズは常に同じであり, PC 間リンクを用いるため, 通信処理を行うタイミングも通信相手も固定される. そのため, 境界更新データテーブルに格納する物理アドレス, サイズは一度指定すると以後変更する必要はない.

[1]送信用と受信用の境界更新データテーブルを作成し, 送信する境界更新データと受信する境界更新データのそれぞれの物理アドレスとサイズを別々のテーブルに登録する.

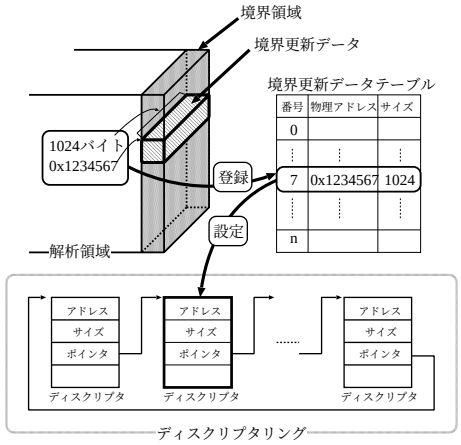

図 5: 境界更新データテーブルとディスクリプタ

境界更新データの整合性確認機能: DDM-BDS プログラムの実行を開始する時に, 初期化処理として通信を行う PC 間で境界更新データテーブルの照合を行う. これにより, 境界更新データのサイズと個数の整合性を確認することができる.

データリンク層のヘッダ配置: NIC がネットワークに送出するデータにはデータリンク層のヘッダが埋め込まれている必要がある. そこで境界領域にヘッダを格納するための領域 (以後ヘッダ領域) を確保する. 一方多くのユーザアプリケーションは, 3 次元の解析データ領域を連続した 3 次元配列の形でアクセスする. このため, 図 6のように, 解析領域に連続する位置に 16 バイト[2] のヘッダ領域を確保する.

解析領域のメモリマッピング: カーネルアドレス空間に存在する特定のメモリ領域をユーザアドレス空間に対応付けるカーネルメモリマッピング機能[3]を用いて, 解析領域をユーザアドレス空間にメモリマッピングする. これによりユーザ/カーネルアドレス空間の間のメモリコピーのオーバヘッドを削減できる (図 7).

4 評価

図 8のように, 1 次元に PC が連続して接続されている PC クラスタを 2 つの PC を用いて疑似的に

[2]イーサヘッダ 14 バイトと BDS ライブラリで用いるヘッダ 2 バイトの合計である.

[3]この機能の実装には, PM [3] のカーネルメモリマッピングの実装方法を参考にした.

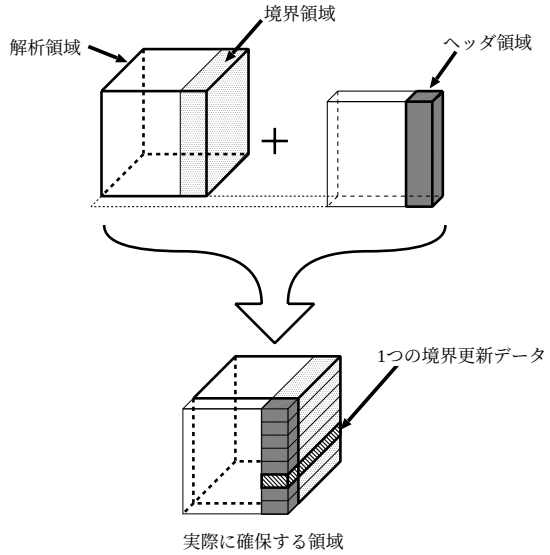

図 6: ヘッダ領域の確保

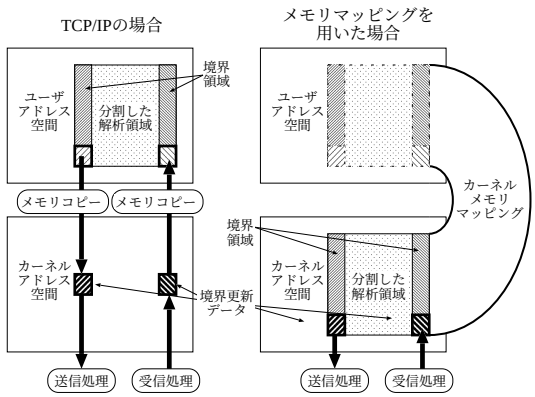

図 7: 解析領域のメモリマッピング

構築することができる. そこで 2 枚の NIC を装着した 2 つの PC を用いて評価を行った. 評価に用いたシステムの構成を図 9 と表 2に示す.

4.1 netperf による通信性能評価

まず, NIC の一般的な通信性能評価に利用されている netperf ベンチマークプログラム [2] を利用し, Gigabit Ethernet NIC の通信性能を評価した. 図 10に測定結果を示す. グラフの横軸は, 通信 1 回あたりの送受信データサイズ, 縦軸はスループットである. MTU (Maximum Transmission Unit) を 9000 バイトに, 通信時間を 30 秒に設定し測定を行った. 送信データサイズが 8400 バイトの時に最大の性能結果が出ており, その時のスループットは, 約 989Mbps である.

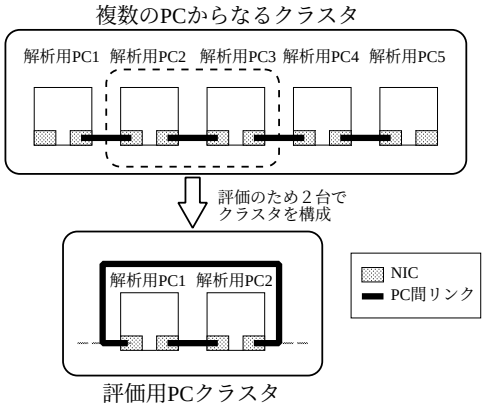

図 8: 2 台の PC による評価用クラスタの作成

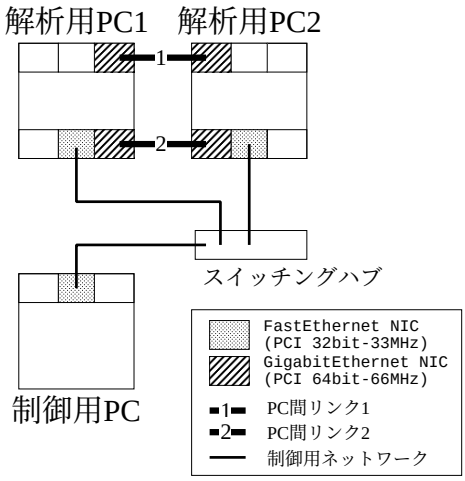

図 9: 評価に用いた PC クラスタの構成

4.2 BDS ライブラリの通信性能の評価

DDM-BDS プログラムで用いられる片方向, 双方向, 交互方向の通信例を図 11に示す. これらの 3 つの通信条件において, TCP/IP と BDS 通信ライブラリのそれぞれを使用し, 使用する PC 間リンクの数を変え[4], データ転送性能の比較を行う. 同じ条件で比較が行えるように, 測定には BDS と TCP/IP の両方に対応できるプログラムを用いた.

片方向データ転送測定: PC1 から PC2 へ送信データサイズのデータを 10 個×1 万回送信する[5]. PC2 で最初のデータの受信が完了してから最後のデータの受信が完了するまでの時間を測定した. 測定結果を図 12に示す. 測定結果から,

[4]図の PC 間リンク 1 のみを使用する 1 通信路と, 図の PC 間リンク 1,2 共に使用する 2 通信路の 2 つの条件で測定する.

[5]境界領域に境界更新データが 10 個存在し, 1 万回境界データの同期を取るアルゴリズムを想定している.

表 2: 評価に用いた PC の構成

解析用 PC	
Processor	PentiumIII-Xeon 1GHz
Memory	128MB
PCI bus	64bit-66MHz, 32bit-33MHz
制御用 PC	
Processor	PentiumPro 200MHz
Memory	192MB
PCI bus	32bit-33MHz
PC 間リンク 1,2	
NIC	Gigabit Ethernet (SysKonnect SK-9821/SK-NET GE-T)
接続方法	直接接続
制御用ネットワーク	
NIC	Fast Ethernet (DEC DE500A)
接続方法	スイッチングハブを経由
ソフトウェア	
OS	RedHat Linux6.2(Kernel 2.2.16) + BDS ライブラリ
プログラム	1)Netperf 2.1 2)DDM-BDS サンプルプログラム

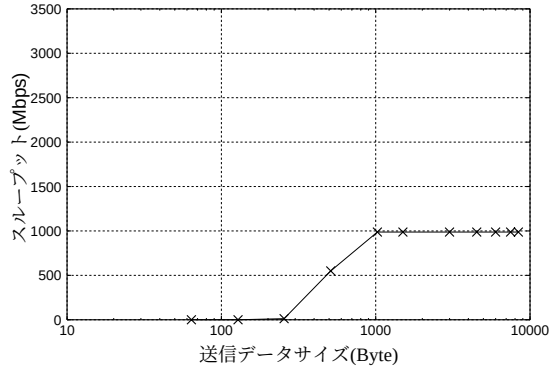

図 10: netperf ベンチマークプログラムの測定結果

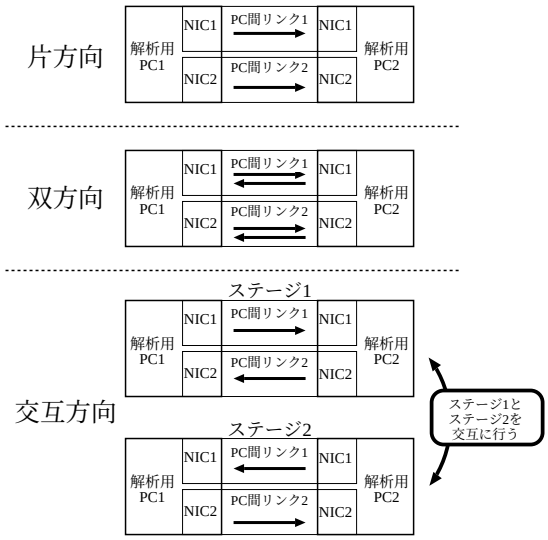

図 11: DDM-BDS 通信例

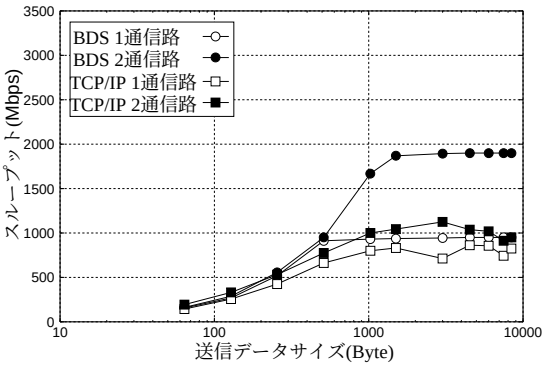

図 12: BDS 通信性能測定結果 (片方向)

TCP/IP と比べ BDS のデータ転送性能は 1 通信路では送信データサイズ 512 バイトで約 1.4 倍, 2 通信路では送信データサイズ 7500 バイトで約 2.0 倍となっており, ゼロコピー通信によるスループットの向上が確認できる. 送信データサイズが 256 バイト以下の場合, TCP/IP は複数の送信データを 1 つにまとめて送信するため TCP/IP と BDS の差が少なくなる.

双方向データ転送測定: PC1 から PC2 へ送信データサイズのデータを 10 個×1 万回送信する. 同時に PC2 から PC1 へ送信データサイズのデータを 10 個×1 万回送信する. TCP/IP では送受信の両方の処理で完了確認を 1 万回行っているが, BDS では行っていない. PC2 で最初のデータの受信が完了してから最後のデータの受信が完了するまでの時間を測定した. 測定結果を図 13に示す. 送信データサイズ 4500 バイト以上での BDS の通信性能は, TCP/IP に比べ 1 通信路で約 1.6 倍, 2 通信路では約 3.5 倍である. BDS では 送信データが 3000 バイト以上では 約 2.9Gbps のスループットが出ている.

交互方向データ転送測定: 1 通信路の場合, PC1 から PC2 に送信データサイズのデータを 10 回送信 (ステージ 1) した後, PC2 から PC1 に 同じサイズのデータを 10 回送信 (ステージ 2) する処理を 1 万回繰り返す. 2 通信路の場合, 1 通信路の処理と同時に, もの一つの PC 間リンクを使い逆方向の送信処理を行う. PC2 で最初のデータの受信が完了してから, 最後のデータの受信が完了するまでの時間を測定した. 測定結果を図 14に示す. 測定結果から, 1 通信路の BDS では, 送信データサイズの増加にともな

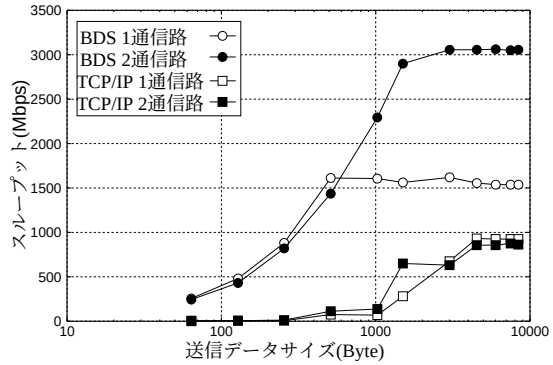

図 13: BDS 通信性能測定結果 (双方向)

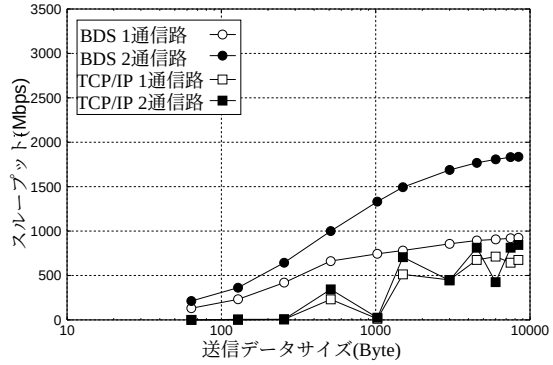

図 14: BDS 通信性能測定結果 (交互方向)

いデータ転送性能が向上している. 2 通信路では, 送信データサイズが 1500 バイト以上の時, BDS のデータ転送性能は TCP/IP の約 2.2 倍である.

4.3 DDM-BDS サンプルプログラム評価

2 節で示した DDM-BDS アルゴリズムに基づくプログラムを作成し評価を行った. プログラム実行中の計算処理と境界領域の同期処理の, それぞれの処理に要した時間の合計を測定した. 図 15に測定結果を示す. 結果より, BDS の計算時間は TCP/IP に比べ 約 15%増加し, 同期時間は半分以下に短縮された. 結果, プログラムの実行時間は 14%減少した. BDS 通信ライブラリを用いた時に, 計算時間が増加するのは, メモリマッピングした解析領域を用いて計算を行ったためである.

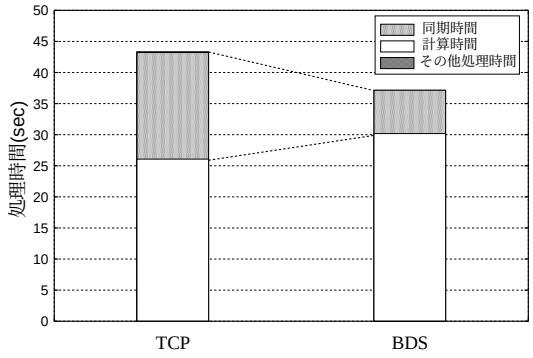

図 15: DDM-BDS サンプルプログラム実行結果

5 おわりに

本稿では, DDM-BDS を用いた PC クラスタ並列処理のための BDS 通信ライブラリを設計した. また, 実際に Gigabit Ethernet により 2 通信路で直結接続した 2 台の PC に BDS 通信ライブラリを実装した. PC クラスタ上で DDM-BDS を用いたプログラムを実行し評価を行った結果, TCP/IP 通信機構と比較して, 本稿で提案するゼロコピー通信機構では通信時間が約半分に短縮されることを確認した. 今後は, BDS を用いた場合の計算時間の増加の原因について, 調査を行い改善する予定である.

参考文献

[1] 服部 準, 阿部 眞, 韓青: “格子状 PC クラスタにおける３次元分散 FDTD 法の実装と評価”, 情報処理学会研究報告 HPC, Vol.69, No.6, 情報処理学会, pp.31-36, 1997.

[2] http://www.netperf.org/

[3] 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, 41-48, 1996.