計算機アーキテクチャ 119−36
（1996. 8. 29）

STAFF-Linkを用いた並列分散I/Oシステムの実現とその評価

佐伯 靖[†] 高橋 淳[†]
中條拓伯[†] 金田悠紀夫[†]

内容梗概

多数のワークステーション (WS) をネットワークで接続し，並列処理環境を実現するワークステーションクラスタにおいて，1つのWSが持つディスク装置へのアクセス集中によりネットワーク上でボトルネックが生じた場合に，そのワークステーションクラスタシステム全体の処理能力が低下する．そのため，各WSのディスク装置にデータを物理的に分散させ，ディスク入出力の負荷を分散させることによって，ボトルネックを解消する並列分散I/Oシステムが必要となる．しかしながら，Ethernetで構成されたワークステーションクラスタで並列分散I/Oシステムを構築した場合，Ethernetがバス型ネットワークであるため，データサイズが大きい時やWSの台数を増やした時に通信の競合が起こり，十分な性能を得ることができない．この問題点を解決する1つの手段として，STAFF-Linkと呼ばれる高速シリアルリンクを用いて並列分散I/Oシステムを構築する方法が考えられる．STAFF-Linkは各WSをpoint-to-pointで接続できるため，Ethernetの場合に見られるような通信の競合が緩和される．本稿では，STAFF-Linkを用いた並列分散I/Oシステムを実現し，その構成と基本性能について述べる．

Implementation and Evaluation of Distributed Parallel-I/O System using STAFF-Link

Yasushi Saeki,[†] Atsushi Takahashi,[†] Hironori Nakajo[†] and Yukio Kaneda[†]

Abstract

A workstation cluster technology has spread as a parallel processing environment among end users. When the increase of access to one workstation with disk devices in workstation cluster causes the heavy network traffic, the performance of the whole workstation cluster system falls down. Therefore, we need to construct the Distributed Parallel-I/O System which relaxes the Disk I/O-bottleneck by dividing data on multiple disk device and distributing of the burden the Disk I/O. However, the much collision on the Bus-typed Ethernet causes the significant bottleneck. To solve this problem we have implemented the Distributed Parallel-I/O System using high-speed serial links called STAFF-Link (Serial Transparent Asynchronous First-in First-out) which enable to connect with the point-to-point network type. And we evaluated this system. In this paper we describe the construction and the basic performance of this system.

1. はじめに

近年，高性能ワークステーションおよび大容量磁気ディスク装置の低価格化，さらにはネットワーク技術の発達が急速に進んでいる．それらの磁気ディスク装置を備えたワークステーションをネットワークで接続し，柔軟な並列処理環境を実現するワークステーションクラスタと呼ばれる技術が普及している．また，それに伴い，ワークステーションクラスタを構成する各ワークステーション間で，メッセージ通信を行なうためのライブラリも，PVM，MPIなどが配布されている．しかしながら，ワークステーションクラスタによる並列処理環境は，I/Oアクセスが頻繁に起こるアプリケーションにおいて，ネットワークの通信性能がシステム全体の処理能力に大きく依存し，また，1つのディスクへのアクセスが集中すると，システム全体の処理能力が低下する．そのため，ワークステーションクラスタを構成する各ワークステーションの持つディ

† 神戸大学工学部情報知能工学科
Department of Computer and Systems Engineering, Faculty of Engineering, Kobe University

スク装置に，入出力データを物理的に分散させ，ディスク入出力の負荷を分散させることによって，ボトルネックを解消する並列分散I/Oシステムが必要であると考えられる．

しかしながら，LANの事実上の標準であるEthernetに代表されるような，バス型ネットワークを用いて構築した並列分散I/Oシステムにおいて，動画像や音声などの大規模マルチメディアデータを処理する場合，通信の競合が起こり，その通信容量の不足が大きな問題となる．また，ATMやFibre Channelのような高速ネットワークでワークステーションクラスタを構成し，並列分散I/Oシステムを実現するにはコストがかかるだけでなく，上位の通信プロトコルのオーバヘッドにより十分な性能を発揮することができない．これら種々の問題の解決法の1つとして，多数のpoint-to-pointリンクを用いて，並列分散入出力システムを構築する方法が考えられる[1]．そこで，我々はワークステーション間のリンクとして，高速シリアルリンクSTAFF-Link (Serial Transparent Asyncronous First-in First-out Link) を用いて，ワークステーションクラスタを構成し，並列分散I/Oシステムを実現した．この際，ワークステーションクラスタ用のメッセージ通信ライブラリを提供するソフトウェアとしてPVM[2]，その上で並列ファイルシステムを実現するソフトウェアとしてPIOUS[3][4][5]を選択した．STAFF-Linkを用いた並列分散I/Oシステムは，複数の転送路で同時に通信ができるので，ノード間通信を行なう際，ネットワーク上で通信の競合が緩和され，効率の良いデータ転送ができるという利点がある．

本稿では，まず並列分散I/Oに関する研究動向を示し，STAFF-Linkを用いた並列分散I/Oシステムの構成を述べる．その上で，Ethernetを用いた場合との比較を行なうことで，本システムの有効性を評価する．

2. 並列分散I/Oの研究動向

現在，並列処理環境において，ディスク入出力におけるボトルネックを解消するために，各方面でさまざまな研究が行なわれている．それらの中には，入出力自身を目的とする研究や，並列計算機開発プロジェクトの一部として行なわれている研究などがあり，その規模は様々である．効率の良い並列分散I/Oシステムを構築するには，次のような問題を考慮し，解決しなければならない．

- どのような形態でComputingノードとI/Oノードを接続するかというアーキテクチャレベルの問題
- 多数の入出力装置をどのように管理するかというシステムプログラムレベルの問題
- 入出力ライブラリのインタフェース，コンパイラ，言語面でのサポートをどのようにするかというユーザレベルの問題

超並列計算機や，ワークステションクラスタを用いた並列分散I/Oシステムで，現在行なわれている主なプロジェクトには，以下のようなものがある．

SIO (Argonne Scalable I/O)[6]

超並列計算機上で，その演算能力に見合うだけの効率の良い入出力機構の開発を目的としており，様々な大学，政府機関，民間企業などが参加している．このプロジェクトでは，実際のアプリケーションを動作させるのに必要なものを決定し，それをもとに新しいプログラム言語の機能，コンパイラ技術，OSのサポート，ファイル記憶機構，高性能ネットワーキングソフトウェアなどを開発する．Intel ParagonおよびIBM SP1上において実装，評価されている．

PPFS (Portable Parallel File System)[7]

イリノイ大学では，並列分散I/Oシステムの実験環境を目的として，クライアント/サーバモデルを基本とするユーザレベルの入出力ライブラリPPFSを開発した．システムの実際の物理的な入出力処理とユーザプログラムとの間にライブラリを用いることで，さまざまなデータ分散，キャッシング，プリフェッチングアルゴリズムなどを実験することができる．

HFS (HURRICANE File System)[8]

トロント大学では，共有メモリ型マルチプロセッサと，これを簡易かつ効率良く利用できるようにするソフトウェアを開発するためのNUMAchineプロジェクトが行なわれている．このプロジェクトにおいて，ファイルシステムとして，HFSが開発されている．

MPI-I/O[9]

MPI (Message Passing Interface) 環境下で並列ファイル入出力をサポートし，その機能性よりも性能を重視しており，ユーザやシステムに依存しないハイレベルな入出力インタフェースを提供している．

3. システムの概要

本システムの構成を図1に示す．I/Oノードはそれぞれ固有のディスク装置を持っており，そこに物理的に分割されたファイルを格納している．各I/Oノードには分散ファイルを管理するプロセスが存在し，Computingノードからのファイル管理やファイル入出力の要求に対して，そのプロセスが互いに通信，協調を行ない，要求を満たす．そのことにより，ユーザは分散ファイルを論理的に1つのファイルとして扱うことができる．

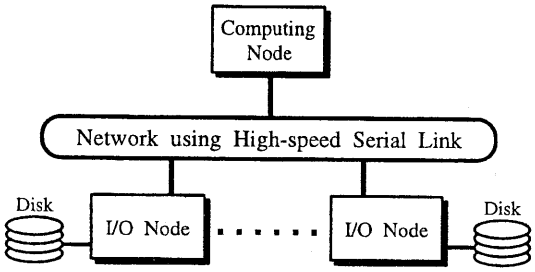

図1　並列分散I/Oシステムの構成

3.1 高速シリアルリンク STAFF-Link

STAFF-Linkは，高速シリアル通信用LSI，送信/受信用FIFOおよび通信コントローラから構成されており，通信処理の多重化による高速性(最高140Mbps)と，シリアルリンクによる柔軟性を兼ね備えている．以下に，STAFF-Linkの主な特徴を示す．

- **通信スループットの向上**
  従来のシリアル通信インタフェースでは，パラレルデータをシリアルデータに変換する際のオーバヘッドのため，通信スループットの向上が妨げられていた．一般に，シリアルリンクによるデータ通信は，1.データの書き込み，2.パラレル→シリアル変換，3.シリアル通信，4.シリアル→パラレル変換，5.データの読みとり，の5つのフェーズに分けられるが，STAFF-Linkでは，そのうち2から4のフェーズを，シリアル通信用LSIにより高速に処理し，送信側と受信側にバッファを設けて5つのフェーズをオーバラップさせることにより，通信スループットを向上させている．
- **簡易なインタフェース**
  内部の通信コントローラが，FIFOが溢れないように，自動的にフロー制御を行なうので，STAFF-Linkへのアクセスは，FIFOメモリへのアクセスと基本的には変わらない．従って，FIFOのフラグを監視することにより，可能な時にアクセスできる．
- **柔軟なネットワーク環境の構築**
  シリアルリンクのため，ポート数を比較的容易に増加でき，1台のワークステーションから多数のワークステーションへの接続が可能となる．従って，種々のトポロジーのネットワークを容易に構築できる．

3.2 実　　装

本システムは，Computingノードとして Sun の SPARCserver5，I/Oノードとして最大4台の SPARCstation5 (SS5) を用いて実装した．STAFF-Linkは，SS5の拡張バスであるSBusに接続され，4系統のリンクを持つ．また，STAFF-LinkとSBusとのインタフェースカードにはDMA (Direct Memory Access) コントローラが搭載されており，主記憶とSTAFF-LinkのFIFOメモリの間でDMA転送を行なうことができる．

また，各ノード間においてSTAFF-Linkを用いたメッセージ通信を行なうために，PVM (Parallel Virtual Machine) を移植 (STAFF-PVM[10]と呼ぶ) し，その上にPVMベースの並列分散ファイルシステムを提供するPIOUS (Parallel Input/OUtput System) を実装することにより，全体として並列分散I/Oシステムを構築する．この様子を図2に示す．

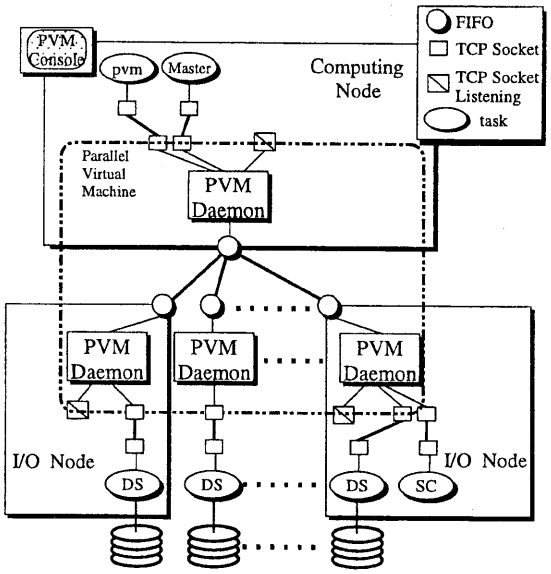

図2　PVM，PIOUSの実装図

PVMは，通信の管理を行なうPVMデーモンと通信用ライブラリから構成されるソフトウェアである．各ノードのタスクがPVMのライブラリを呼び出すと，データはそのノード上のPVMデーモンから目的となるタスクが存在するノード上のPVMデーモンへ，ネットワークを経由して適切に転送され，目的のタスクに渡される (TCP) ことで通信が完了する．また，PVMデーモンを介さずに直接タスク間で通信を行なう (TCP) モードもサポートしている．

本システム (図2) は，以下の要素から構成される．

- **サービスコーディネイター (SC)**
  open，close，chmodなどのファイル操作を行なう．
- **データサーバ (DS)**
  read，writeなどの，分散したディスクブロックの入出力制御を行なう．

SCおよびDSはPIOUSのプロセスであり，PVMのタスクの1つとして機能する．Computingノード内のタスクが，分散データへのリードアクセスとして

PIOUS の`pious_read` ライブラリを呼び出すと，そのリード命令は STAFF-Link を介して，Computing ノードの PVM デーモンから各 I/O ノードの PVM デーモンに転送される．その後各 I/O ノード内の DS に渡され，DS が各ディスクからデータを読み出し，そのデータは逆のルートを辿り，`pious_read` ライブラリを呼び出したタスクへ転送される．

3.3 STAFF-PVM の転送速度

PVM デーモンが STAFF-Link を介してデータを転送するモードとして，1バイト毎のデータの読み書きを行なうバイト転送モードと，DMA コントローラを用いた DMA 転送をサポートしている．この 2 種類の転送モードを使って，STAFF-PVM で通信を行なった場合の計測結果を図 3に示す．

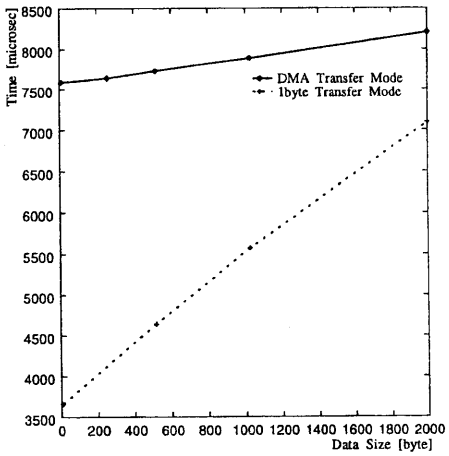

図 3　2 種類の転送モードにおける STAFF-PVM の通信時間

図 3より，バイト転送モードは少量のデータ，すなわち命令やアクノリッジなどのデータを転送するのに適していることが分かる．これは，バイト転送モードでは転送時のオーバヘッドが少ないためである．また，DMA 転送モードは DMA 転送を行なうための初期設定を行なうため，少量のデータの転送には適していない．本システムでは，一回で転送するデータのサイズを 2[Kbyte] 以下にしているので，少量のデータ転送に適しているバイト転送モードの STAFF-PVM を使用した．

4. 基本性能評価

Ethernet あるいは STAFF-Link を用いた並列分散 I/O システムにおいて，以下の基本的な実験を行ない，両者の比較評価を行なった．

まず Computing ノードのメモリにデータを格納し，そこから I/O ノードの持つローカルディスクへの書き込み (write)，および I/O ノードの持つローカルディスクから Computing ノードのメモリへの読み出し (read) に要する時間を計測した．この際，I/O ノードの台数 (1, 2, 4 台)，データサイズ (8K から 8192Kbytes) をパラメータとした．また，PIOUS のオーバヘッドを調べるために，ネットワークを経由せず Computing ノードの持つディスク装置を対象とした読み書き (Local-PIOUS) を行ない，その時間を計測し，また UNIX のシステムコールによるローカルディスクへの読み書き (System Call) も計測した．Ethernet, Local-PIOUS, System Call を用いた場合の結果を図 4, 5, また，STAFF-Link を用いた場合の結果を図 6, 7に示す．

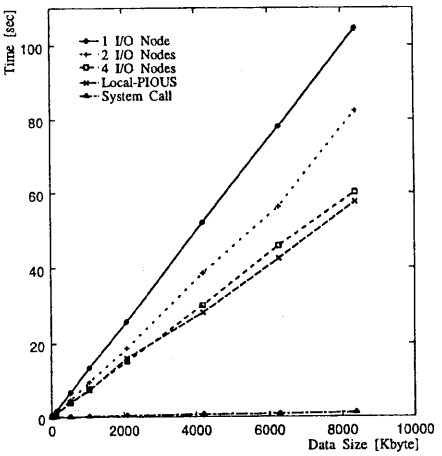

図 4　Ethernet 使用時のread に要する時間

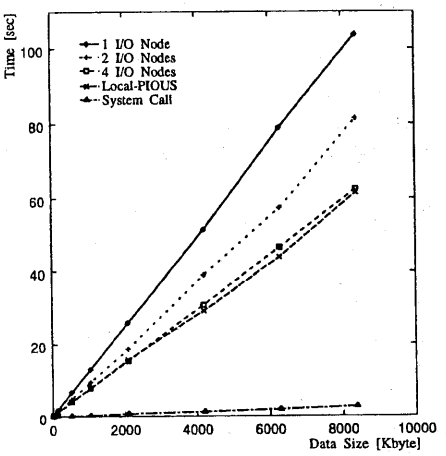

図 5　Ethernet 使用時のwrite に要する時間

図 4, 図 5より，Ethernet を用いた並列分散 I/O システムでは，データサイズにほぼ正比例して，`read` 時間，`write` 時間ともに増加していることが分かる．

次に，最小二乗法により求めたそれぞれの傾きの逆数，すなわち入出力速度に注目すると，I/O ノード数 1，2，4 台に対して，read の場合は，80.59，104.98，138.75[Kbyte/sec]，write の場合は，80.68，105.22，134.63[Kbyte/sec] となる．これらから，I/O ノードの台数が 2 倍になると，入出力速度における性能が約 1.28 倍から 1.32 倍向上されることが分かる．

さらに，Local-PIOUS と System Call のグラフから PIOUS のオーバヘッドが分かる．入出力速度に注目すると，PIOUS を使うことでread においては約 1.97%，write においては約 4.17%にまで減少している．この PIOUS の大きなオーバヘッドにも今後対応していかなくてはならない．

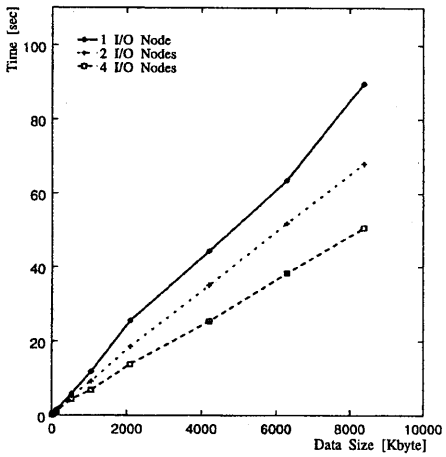

図 6　STAFF-Link 使用時のread に要する時間

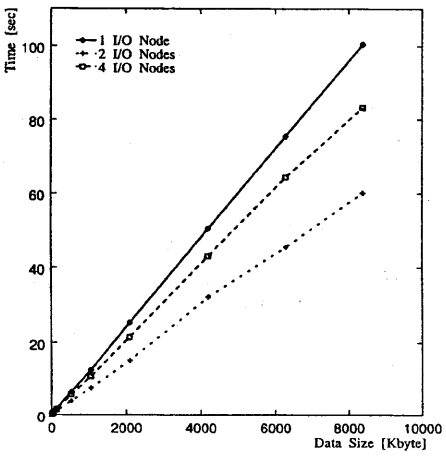

図 7　STAFF-Link 使用時のwrite に要する時間

図6，図7より，STAFF-Link を用いた並列分散I/O システムでは，read については前述した Ethernet の場合と同様のことが確認できるが，write においては，I/O ノード数を 1 台から 2 台に変化させた場合，write 速度における性能が約 1.65 倍向上するのに対し，2 台から 4 台に変化させた場合，約 0.72 倍の性能低下が見られる．この現象については，STAFF-Link を構成するハードウェアの問題，STAFF-Link の各 FIFO へのデータの読み書きを STAFF-PVM がどのように制御するかという STAFF-PVM 実装上のソフトウェアの問題，PVM を書き換えたことによる PIOUS との相性 (最小転送データサイズの違いなど) の問題など，さまざまな理由が考えられ，現在究明中である．

従って，read に注目して，2 つのシステムの比較を行なう．両システムで計測した，8192[Kbyte] のデータのread 時間を図 8に示す．

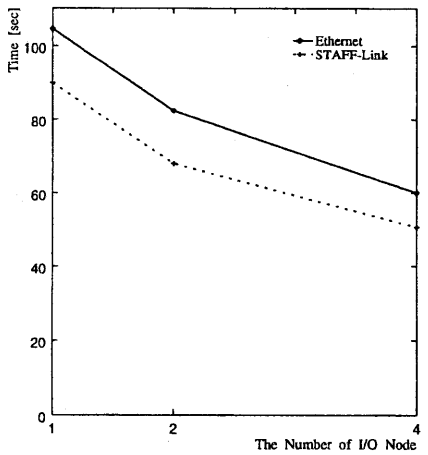

図 8　I/O ノード数に注目したread 時間

まず，I/O ノード数が同じ場合，read 速度は STAFF-Link を用いた方が約 1.20 倍速くなっている．また，I/O ノード数の増加に対するread 速度の増加率を表 1に示す．

表 1　I/O ノード数の増加に対するread 速度の増加率

ネットワーク	Ethernet	STAFF-Link
1 台から 2 台のread 速度増加率	30.2%	28.8%
2 台から 4 台のread 速度増加率	32.2%	35.2%

表1から，両者の増加率に注目すると，I/O ノード数が増えるにつれて STAFF-Link の方がread 速度の増加率が上昇している．バス型ネットワークである Ethernet は I/O ノード数が増加するに従い，ネットワークがボトルネックとなり，ある台数を越えると頭うちになり入出力の性能が低下する可能性がある．一方，STAFF-Link は各ノードが point-to-point で接続されているため，I/O ノード数が増加するに従い，ますますread 速度の増加率が上昇すると考えられる．従っ

て，I/O ノード数をさらに増加させた場合に，Ethernet を用いた場合と STAFF-Link を用いた場合の入出力における性能の差が顕著に現れるであろう．ただし，STAFF-Link を用いた並列分散 I/O システムの場合，種々のトポロジーで構成されたシステムが考えられるので，そのネットワークのトポロジーによっても，入出力の性能が変化することも考慮しなくてはならない．

5. 今後について

今回実現した並列分散 I/O システムと並行して，リンクにルーティング機能を持たせたシステムの実装を進めている．このルータボードを用いることにより，さらに多くのワークステーションを接続することができる．今後，以下のような方向で並列分散 I/O システムの実現および評価を行なっていく予定である．

DMA 転送モードの STAFF-PVM の使用

バイト転送モードではその通信時間は線形に増加し続けるので，大規模なデータが飛び交う並列分散 I/O においては，データサイズによっては無視できないほどのオーバヘッドとなる可能性がある．やはり並列分散 I/O においては，初期設定に多少時間がかかるものの，一度に多量のデータを転送できる DMA 転送モードを使用する必要性がある．

I/O ノード数の増加

本稿では I/O ノード数を最大 4 台の場合でしか計測しなかったが，Ethernet と比較を行なう際，さらに I/O ノード数を増やしたシステムで評価する必要性がある．最終的には，I/O ノード数 16 台での評価を行なう予定である．

種々のネットワークトポロジーによる性能評価

I/O ノード数を増加した場合，それらをどのようなトポロジーで接続するかということも，ホップ数やルーティングの関係で，入出力の性能に影響を及ぼすと考えられる．従って，種々のトポロジーのシステムでの評価を行なう必要性がある．

実際のアプリケーションでの評価

実際の並列処理では，入出力だけが単独で起こる可能性は極めて低い．従って，計算と入出力がオーバラップするアプリケーション，またはベンチマークプログラムでの評価を行なう必要性がある．現在，画像などの大規模データの圧縮，展開を行なうアプリケーションを用いて評価を進めている．

以上のような実装，評価を行ない，スケーラビリティ (I/O ノード数を増加させた場合に，それに見合うだけの速度向上があること) を持った並列分散 I/O システムが実現できれば，大規模な科学的計算を行ないながら，大規模なデータの入出力を行なう量子化学などの分野で応用できるであろう．

参 考 文 献

[1] 中條拓伯，松田秀雄，金田悠紀夫，“超並列計算機におけるワークステーションクラスタ・ファイルシステム，” 情報処理学会研究会報告 ARC107-24, pp.185-192 (1994)

[2] Al Geist, Adam Beguelin, Jack Dongarra, Weicheng Jiang, Robert Mancheck, Vaidy Sunderam, 村田英明 訳 “PVM3 ユーザーズガイド&リファレンスマニュアル日本語版” (1994)

[3] Steven A.Moyer, V.S.Sunderam, “PIOUS for PVM Version 1.2 User's Guide and Reference Manual,” (1995)

[4] Steven A.Moyer, V.S.Sunderam, “PIOUS: A Scalable Parallel I/O System for Distributed Computing Environments, ” Computer Science Technical Report CSTR-940302 (1994)

[5] Steven A.Moyer, V.S.Sunderam, “Characterizing Concurrency Control Performance for the PIOUS Parallel File System,” Computer Science Technical Report CSTR-950601 (1995)

[6] James T.Poole, “Preliminary survey of I/O intensive applications,” Technical Report CCFS-38, Scalable I/O Initiative, Caltech Concurrent Supercomputing Facilities, Caltech (1994)

[7] James V.Huber,Jr, “PPFS: An experimental file system for high performance parallel input/output, ” Master's thesis, Department of Computer Science, University of Illinois at Urbana Champaign (1995)

[8] Orran Krieger and Michael Stumm, “HFS: a flexible file system for large-scale multiprocessors, ” In Proceedings of the 1993 DAGS/PC Symposium, pp.6-14, Hanover, NH (1993)

[9] P.Corbett, D.Feitelson, Y.Hsu, J.Prost, M.Snir, S.Fineberg, B.Nitzberg, B.Traversat and P.Wong, “MPI-IO: A Parallel File I/O Interface for MPI, Version 0.2, ” IBM Research Report RC 19841 (87784) , IBM T.J Watson Research Center, (1994)

[10] 高橋淳，中條拓伯，小畑正貴，金田悠紀夫，“STAFF-Link を用いたワークステーションクラスタ上への PVM の実装とその評価，” ハイパフォーマンス・コンピューティング 57-1, pp.1-6 (1995)