計算機アーキテクチャ 125－9
（1997. 8. 20）

高速シリアルリンクを用いたDSMシステムの実装とその評価

市川明弘† 薬師神昌夫††
中條拓伯† 金田悠紀夫†

本研究室では，WSクラスタ上の並列処理環境として分散共有メモリ(DSM)システムを実装し，その評価を行なってきた．新たなハードウェアを追加する必要なく，またシステムソフトウェアをユーザレベルで記述することで，高い移植性を得ていた．

しかしながら，Ethernetを通信媒体とし，ノード間通信時にOSに対してシステムコールを発行するなど，ノード間通信時の遅延が大きくなり，システムの性能が頭うちになってしまう．そこで通信媒体をEthernetから高速シリアルリンクに変更し，通信時のオーバヘッドおよびレイテンシを削減することで，さらなる性能の向上を試みた．本論文では，ノード間の通信媒体にシリアルリンクを用いたDSMシステムの構成を述べ，その性能を評価することにより，通信媒体としてシリアルリンクを使用することがDSMシステムの性能の向上に有効であることを示す．

Implementation and Estimation of a DSM System Using High Speed Serial Links on WS Clusters

AKIHIRO ICHIKAWA,† MASAO YAKUSHIJIN,†† HIRONORI NAKAJO† and YUKIO KANEDA†

We have implemented and estimated Distributed Shared Memory system on workstation cluster. The system software is written in user context, and no additional hardware is added to construct the system. Therefore the system has a high portability. However, because Ethernet is bus-type network topology and its band-width is narrow, the latency of inter node communication is so high. Therefore we will propose to use a serial link, called STAFF-Link(Serial Transparent Asynchronous First-in First-out Link), for communication media to reduce the latency of inter node communication.

1. はじめに

LAN環境に接続されたワークステーション(WS)を有効に活用した並列処理環境として，WSクラスタの研究が行なわれている[1]．並列処理における通信方式として，メッセージ通信によるものと，共有メモリによるものがある．WSクラスタにおいてメッセージ通信を用いてプロセス間通信を行なう場合，PVM[2]やMPI[3]が広く使用されている．しかしながら，データの分割方法や，プロセス間通信のプロトコルを明確に定めなければならないなど，並列プログラムを記述するのに考慮すべきことが多い．一方，共有メモリを用いた場合，共有データにすべてのプロセスがアクセスできるのでデータ分割は必要なく，プロセス間通信で特に定めることがないなど，並列プログラムの記述が容易である．しかしながら，WSクラスタ上では物理的に共有メモリを構築することができないため，メッセージ通信を用いて仮想的な共有メモリを構築する必要がある．また，一般にWSクラスタにおいては通信遅延が大きいため，共有メモリにアクセスする際には毎回のアクセス毎にそのページを保持しているノードに要求を発行するのではなく，各ノードがキャッシュメモリを持ち，そのキャッシュメモリを介して共有メモリにアクセスすることで他ノードとの通信量を少なくし共有メモリへのアクセス遅延時間を低減させる．また，キャッシュメモリを用いた場合では各ノードが保持しているキャッシュメモリの一貫性を保つためのキャッシュコヒーレンシ制御を行なう必要がある．この方法としてはライトスルーや書き込み無効化型ライトバックなどの方式がある[4]．

本研究室では，WSクラスタ上に分散共有メモリ(DSM : Distributed Shared Memory)システムを実装し，評価を行なってきた[5]．そこでは，既存のハードウェアに新たなハードウェアを付加することなく，また，システムを構成するソフトウェアをユーザレベ

† 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
†† 大日本印刷株式会社

ルで記述することによって，高い移植性を確保している．また，キャッシュミスは UNIX のセグメンテーションフォルト時に発生するシグナル (SIGSEGV) をトラップすることで自動的に検知でき，キャッシュにヒットした場合は DSM システムの処理が介在しないことからローカルメモリへのアクセスと変わらないアクセス性能が得られている．しかしながら，通信を行なうために必要な OS に対するシステムコールによるオーバヘッドや，ノード間通信の媒体に用いている Ethernet の通信バンド幅の限界，バス型のネットワークトポロジに起因するメッセージの衝突などの要因によりノード間通信時の遅延が大きい．また，キャッシュミス時にリモートノードとの通信を行なう必要があることから，キャッシュヒット時のアクセス時間に比べてキャッシュミス時のミスペナルティが非常に大きい．そのため，DSM システムの性能を向上させるにはノード間の通信遅延を削減し，キャッシュミス時のミスペナルティを削減することが必要となる[6]．

そこで，ノード間通信にルータを搭載した高速シリアルリンクを用い，システムを構成するノード間を point-to-point で接続することで，ミスペナルティの削減を図る．また，ルータにプログラマブルプロセッサを搭載することで DSM システムの処理の一部をこのプロセッサに処理させ，要素プロセッサにかかるシステムの処理の負荷を軽減させることができる．本論文では，シリアルリンクを用いた DSM システムの構成を示し，本システムの性能評価を行ない，本システムの有効性を検証する．

2. システムの構成要素

本システムにおけるノードの構成を図 1 に示す．

本システムではノード間の接続に，本研究室で開発した高速シリアルリンクである STAFF-Link(Serial Transparent Asynchronous First-in First-out Link) を用いている．図 1 のように構成されるノードを複数接続することで，本システムは構成される．

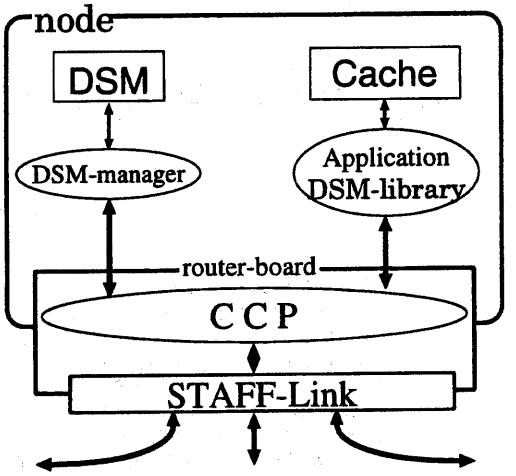
本システムでは，メモリコンシステンシモデル[7]にはシーケンシャルコンシステンシモデル[8]を，キャッシュコヒーレンシ制御方式には書き込み無効化型を採用している．また，共有メモリの管理には，分散フルマップディレクトリ方式[9]を採用している．

2.1 システムのソフトウェア構成

2.1.1 制御プロセスとライブラリ

本システムは，DSM-library，CCP(Communication Control Process)，DSM-manager の 3 つのソフトウェアで成り立っている．それぞれの機能は次の通りである．

- DSM-library
  アプリケーションに静的にリンクされ，アプリケーションからの共有メモリの確保や，ロック，アンロック，バリア同期などの要求を処理し，必要に応じて CCP を通して DSM-manager に処理を依頼する．
  アプリケーションのキャッシュミスを UNIX のセグメンテーションフォルト発生時のシグナル (SIGSEGV) をトラップすることで検知し，CCP に対してページデータや他ノードのキャッシュの無効化などの要求を発行する．また，DSM-manager からのキャッシュの無効化要求を処理するためのスレッドを生成する．
- CCP(Communication Control Process)
  DSM-library からの DSM システムに対するページデータや無効化などの要求に基づいて，適当なノードの DSM-manager に対して処理要求を発行する．このとき，DSM-library や DSM-manager から受けとった処理要求やデータなどに対してパケットの構築などの前処理も行なう．そのほか，アプリケーションからのバリア同期の要求を処理し，またパケットのルーティング制御も行なう．
- DSM-manager
  CCP を通してアプリケーションから送られたページデータや他ノードのキャッシュのコピーの無効化要求を，管理している共有メモリのディレクトリの内容を基にページデータの返送や無効化要求の発行などを行なう．



DSM: Distributed Shared Memory
CCP: Communication Control Process

図 1　STAFF-Link を用いた DSM システムのノード構成

2.1.2 本キャッシュシステムの特徴

通常，DSM システムにおいてはメモリへの平均アクセス時間を削減するために一定容量のキャッシュメモリを用意している．本システムでは，キャッシュメモリはアプリケーションの共有メモリの確保要求時に，

動的にアプリケーションのアドレス空間中に確保される．本システムで採用しているキャッシュ方式の長所，短所を挙げる．

- 長所
  キャッシュのリプレースが起こらない．
  キャッシュメモリは，共有メモリの確保要求がなされた時に動的に確保されるが，その共有メモリラインとキャッシュラインが1対1対応しているためである．よって，キャッシュのリプレースが起こらないことが保証される．
  このため，DSM-manager から返されたページデータを格納するための空きキャッシュラインを検索する必要がなくなる．また，FIFO 方式や LRU 方式といったキャッシュメモリのページ置換アルゴリズムも必要なくなる．
- 短所
  キャッシュメモリをアプリケーションのアドレス空間中に実際に確保してしまうため，アプリケーションから利用できる共有空間の大きさは，そのアプリケーションのとり得る最大の仮想空間の大きさに制限されてしまう．さらに，仮想空間のページングが起きてしまうと性能が著しく低下してしまうため，性能は搭載されている実メモリの大きさに依存する．

本システムにおけるキャッシュメモリ方式の短所は，本質的な解決方法とはいえないが，WS に搭載されている実メモリを増設することである程度回避することができる．

2.1.3 巡回型マルチキャストによる無効化

バス型のネットワークでシステムを構成している場合，ブロードキャストを行なうことで該当するページデータを保持しているノードに対して無効化要求を発行することが多い．しかしながら，本システムではノード間の接続にシリアルリンクを用いて point-to-point 型のネットワークを用いているためにブロードキャストを行なうには適さない．そこで，本システムでは無効化要求の発行に巡回型マルチキャスト方式を採用している．巡回型マルチキャスト方式とは，無効化要求を受けとるべきノードの一つ一つにパケットを巡回させながらメッセージを伝達していき，最後のメッセージを受けとったノードが要求を出したノードの DSM-manager に対してアクノリッジを返す方式である．

2.2 システムのハードウェア構成

本システムを構成する各ノードに用いたマシンのスペックを表 1 に示す．また，本システムで用いた STAFF-Link ルータボードの構成を図 2 に示す[10]．

ルータボードには，プログラマブルプロセッサとして TI 社製の DSP(TMS320C40) を搭載し，データ用メモリ (Local Data Memory: LDR)，プログラム用メモリ (Local Program Memory: LPR) をそれぞれ 2MB 搭載している．プロセッサに DSP を用いた理由としては，

- Global Bus と Local Bus の 2 本のバスがある．
- 内蔵 DMA による高速なメモリ転送が可能．
- 内蔵タイマを用いてのパフォーマンスの計測が可能．

が挙げられる．

表 1　実験で用いた WS のスペック

Workstation	Sun SPARCStation5
CPU	microSPARC II (110MHz)
OS	Solaris2.4 (SunOS5.4)
Memory	32MB

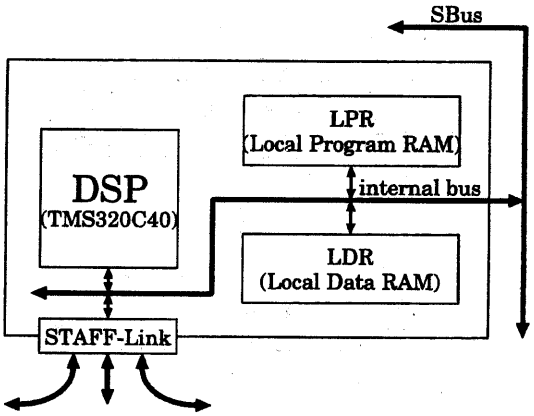


図 2　STAFF-Link ルータボードの構成概念図

WS とルータボードは SBus を介して接続されており，WS からルータボード上のデータ用メモリ，プログラム用メモリ，およびステータスフラグにアクセスすることが可能である．また，ルータボードは WS に対して割り込みを発行する機能を持ち，ステータスフラグやデータ用メモリを媒介にして WS と通信を行なうことができる．これを利用して，DSM システムの処理の一部をルータボード上の DSP に担当させることで，要素プロセッサからシステムの負荷を軽減することができる．ルータボード上では CCP が動作しているため，DSM-library や DSM-manager から見た他ノードへのインターフェースは通常のメモリと変わりがなく，また，パケットの生成やルーティング制御はルータボードが行なうので，通信インターフェースの簡略化がなされている．

また，ルータボード上には STAFF-Link のインターフェースが 3 ポート用意されている．STAFF-Link の構成を図 3 に示す．STAFF-Link では，シリアル通信専用の LSI を用いることで通信データのパラレル-シリアル変換を自動化している．また，通信路の両端に FIFO メモリを搭載することでプロセッサから見

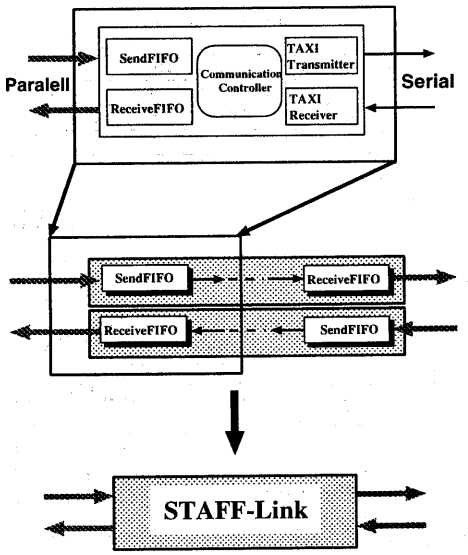


図3　STAFF-Link の構成概念図

た通信インターフェースを簡略化している．プロセッサはこの FIFO へのアクセスによってデータの送受信を行なうことができる．FIFO へのアクセスとリンクを通じた通信はオーバラップさせることができ，また，リンク間のデータ転送速度は STAFF-Link では最大で 140Mbps となり，高い通信バンド幅を確保することができる[11]．

現在のルータボードを図 4 に示す．

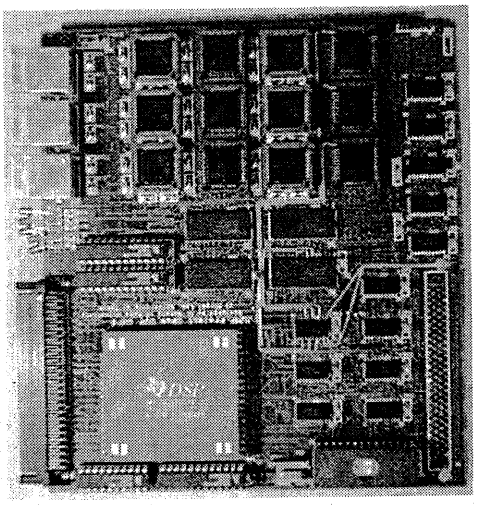


図4　ルータ PCB の実装

3. システムの性能評価

本システムにおける，共有メモリへの基本アクセス性能を共有メモリへのアクセスパターン毎に評価を行なう．システムを構成するノード数が 1 の時のシステムの性能を評価した．

3.1 キャッシュヒット時のアクセス性能

本システムにおける，キャッシュヒット時のリードアクセス，ライトアクセスの性能を表 2 に示す．

表2　キャッシュヒット時のリード／ライトアクセス性能

アクセスパターン	処理時間
リードヒット	130(ns)
ライトヒット	170(ns)

キャッシュにヒットした場合，リードアクセス，ライトアクセスに関係なく，非常に高い性能が出ている．本システムではキャッシュミスの検知を，UNIX においてセグメンテーションフォルト時に発生するシグナル (SIGSEGV) をトラップすることで行なっているため，共有メモリへの各アクセス毎にキャッシュにヒットしているかどうかを調べる必要がない．このため，キャッシュにヒットするアクセスの場合に，ローカルのメモリへのアクセスと変わりがない性能が得られる．

3.2 キャッシュミスをした場合のアクセス性能

キャッシュミスには，リードミス，ライトミス，クリーンライトの 3 パターンが存在するが，このうち，クリーンライトにはページデータの移動は伴わないので，他の 2 つと分けて考える必要がある．

3.2.1 クリーンライト時におけるアクセス性能

クリーンライトは，リードアクセスのみが許された，他のノードと共有しているキャッシュに対してライトアクセスを行なった場合である．この時，アクセスしようとしている共有メモリの同一コピーを保持しているシステム内の各ノードに対して，そのコピーを無効化するように要求することで，キャッシュのコヒーレンシを保っている．クリーンライト時におけるアクセス性能を表 3 に示す．

表3　クリーンライト時のアクセス性能

アクセスパターン	処理時間
クリーンライト	6.552(ms)

3.2.2 リードミス，ライトミス時におけるアクセス性能

自ノードのキャッシュ内にコピーが存在しない共有メモリ空間にアクセスした場合である．リードミスの場合はアクセスしようとしている共有空間のページデータのコピーを要求するだけで良いが，ライトミス

の場合はアクセスしようとしている共有空間のページデータを保持している各ノードに対して，そのコピーを無効化するように要求しなくてはならない．

リードミス，ライトミス時のアクセス性能を，表 4 に示す．

表 4　リード／ライトミス時のアクセス性能

アクセスパターン	処理時間
リードミス	7.034(ms)
ライトミス	7.280(ms)

3.3　キャッシュミス時のアクセス性能の解析

キャッシュヒット時に比べて，キャッシュミス時のミスペナルティが非常に大きく，多少アプリケーションプログラムのキャッシュヒット率が上昇しても，たいして性能は向上しない．

そこで，キャッシュミスの起こるアクセスパターンのうち，リードミスについてその処理時間の解析を行なう．

3.3.1　リードミスにおける処理時間の解析

リードミス時における処理の流れは図 5 のようになる．

1 キャッシュミストラップ：セグメンテーションフォルト時に発生するシグナル (SIGSEGV) をトラップし，ルータボードにページデータの要求を発行する．
2 ページ転送要求：ルータボードが DSM-manager にページデータの要求を発行する．
3 ページ転送：DSM-manager が要求されたページデータをルータボードに送る．
4 転送完了通知：ルータボードが DSM-library に処理の終了を通知する．
5 キャッシュへのコピー：DSM-library がルータボード上のページデータをキャッシュメモリにコピーする．

上記のそれぞれの処理に要する時間を表 5 に示す．

表 5　リードミス時の各処理に要する時間

	所要時間 (μs)
処理全体	7034
キャッシュミストラップ:1	220
ページ転送要求:2	113
ページ転送:3	625
転送完了通知:4	664
キャッシュへのコピー:5	897
合計	2519
DSM-manager への割り込み処理	1189
DSM-library への割り込み処理	2621

上記のように，実際の処理時間のみを考えると約

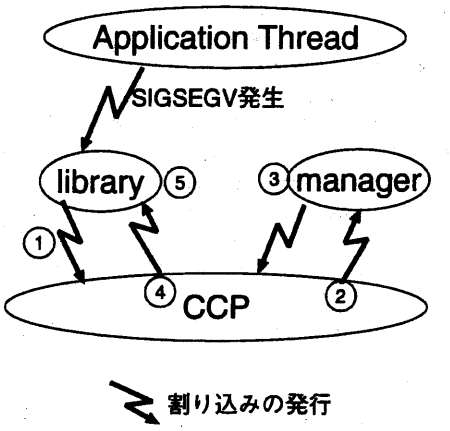


図 5　リードミス時のコヒーレンシ制御の流れ

2.5ms で処理は終るはずだが，実際には約 7ms を要する．これは，ページ転送要求を発行し，ページ転送開始を通知するために，ルータボードが DSM-manager に対して割り込みを発行することで要求の到着を通知し，リモートノードから LDR への転送を完了し，キャッシュ領域へのコピーを要求するために，ルータボードが DSM-library に対して割り込みを発行する．ライトミスの場合，リードミスと比べても，ほぼ同じ時間で各処理は終っている．これは，双方ともページデータを扱うという点で処理内容がほとんど同じだからである．クリーンライトの場合はページデータを扱わないため，ページ転送時においてページデータのコピー処理がない分処理時間が短くなる．このそれぞれの割り込み処理に約 1.2ms，2.6ms かかっており，この 2 つの割り込み処理時間で全体の処理時間のおよそ 55% を費やしている．また，これ以外にも DSM-library がルータボードに割り込みを発行して要求の発生を通知するための処理と，DSM-manager がルータボードに割り込みを発行して処理の終了を通知するための処理に約 1ms かかっており，これも含めると全体の処理時間の約 65% を占める．クリーンライトの場合，ページ転送および転送完了通知に要する時間はリードミス，ライトミスに比べて短いが，各割り込み処理に要する時間に差はないため，全体として，リードミス，ライトミスと比べて処理時間に差はなくなってしまう．

よって，キャッシュコヒーレンシ制御のための処理時間のうち，65% が DSM-library，DSM-manager，ルータボード間の同期処理に費やされており，システムの性能を向上させるには，この三者間の同期処理の改善が必要であることがわかる．

3.4　システムの性能予測

本システムを複数のノードで構成した場合の性能の予測を行なう．各ノードには 3 つのリンクがあるので，

本システムではネットワークトポロジとしてリング構造を基本とし，最も遠いノードとのバイパスリンクを追加したものを考える．

2.2 で述べたように，STAFF-Link を用いた通信では，最大で 140Mbps の通信バンド幅を確保でき，また DSP による通信用 FIFO へのアクセスとデータの転送はオーバラップさせることができる．

本システムで用いているページサイズは 4096 バイトであり，ページデータ以外のシステムの制御メッセージは 3 から 7 バイト，パケットヘッダも数バイトなので，最大パケット長は 4100 バイト強となる．よって，隣接ノード間の通信は最大で約 230μs を要する．

仮にシステムを 20 のノードで構成した場合，パケットの平均ホップ数は 3.1 となり，パケットの通信にかかる時間は約 730μs となる．このことから，ページデータを伴うコヒーレンシ制御の処理時間は現状の約 1.1 倍程度で押えられるので，ノード数の増加に対してコヒーレンシ制御の処理時間の増加は小さく押えられるものと思われる．

4. おわりに

本論文では，我々が開発を進めてきた DSM システムの性能を向上させるための手法として，ノード間の通信にシリアルリンクを用いることを提案し，現状のシステムの性能を基にその性能を予測した．その結果，ミスペナルティの半分以上が割り込みなどの同期処理に費やされていることがわかった．また，シリアルリンクを用いることで基本的な通信性能の向上が，ミスペナルティを低減させることでシステム全体の性能の向上が期待できることがわかった．

今後は STAFF-Link を用いた実装を進めて，共有メモリへのアクセス傾向の違う複数の実アプリケーションを実行することでシステムの性能を評価する予定である．

また，システムの性能が割り込みなどの同期処理に大きく依存していることから，DSM システムのプロセスの配置やハードウェアの構成などを見直す必要がある．本システムで用いたルータボードでは，ルータボードに割り込みを発行してアクセス権を DSP から獲得しなければ WS は LDR に対してアクセスできない．よって，LDR のアクセス権を獲得するためのルータボードへの割り込みが極めて短い時間間隔で発生するために，同期にかかるコストが大きくなってしまう．また，ルータボードに対して割り込みを発生させるにはシステムコールを用いなければならず，そのオーバヘッドも大きい．そこで，アクセス権にとらわれずに LDR に対してアクセスできるようにハードウェアを構成することで割り込みによる同期処理のためのコストを削減でき，現状のシステムに比べて性能が向上することが期待できる．

さらに，メモリコンシステンシモデルの緩和やキャッシュコヒーレンシ制御プロトコルの見直しなどを含めてシステム全体のソフトウェア構成を考慮することで，システムの基本的な性能を向上させる予定である．

参考文献

1) Craig C.Douglas, Timothy G. Mattson and Martin H. Shultz, "Parallel Programming Systems for Workstation Clusters," Technical Report TR-975, Yale University Depertment of Computer Science Research, August 1993.
2) Al Geist, Adam Beguelin, Jack Dongarra, Weicheng Jiang, Robert Manchek, Vaidy Sunderam, "PVM: Parallel Virtual Machine - A Users' Guide and Tutorial for Networked Parallel Computing," MIT Press, (1994).
3) W. Gropp, E. Lusk, A. Skjellum, "Using MPI:Portable Parallel Programming with the Message-Passing Interface," MIT Press, (1994).
4) 天野英晴, "並列コンピュータ," 株式会社 昭晃堂, pp.38-42, (1996).
5) 中條拓伯, 藏前健治, 金田悠紀夫, 前川禎男, "ソフトウェア DSM におけるコヒーレント・キャッシュシステムの実装と評価," 情報処理学会論文誌, 36 巻 7 号, pp.1719-1728, (1995).
6) 薬師神昌夫, 中條拓伯, 金田悠紀夫, "WS クラスタにおける機能分散ネットワークを用いた DSM システムの実装," 情報処理学会研究報告, Vol.96, No.80, pp.137-142, (1996).
7) Andrew S.Tanenbaum, 水野忠則, 鈴木健二, 宮西洋太郎, 佐藤文明 訳, "分散オペレーティングシステム," 株式会社プレンティスホール出版, pp.301-316, (1996).
8) Leslie Lamport, "How to Make a Multiprocessor Computer That Correctly Executes Multiprocess Programs," IEEE Transactions on Computers, Vol.C-28, No.9, (1979).
9) 富田眞治, "並列コンピュータ工学," 株式会社 昭晃堂, pp.146-149, (1996).
10) 吉山晃, 中野智行, 中條拓伯, 金田悠紀夫, "超並列計算機 JUMP-1 におけるディスク入出力サブシステムの実装と評価," 情報処理学会研究報告, Vol.96, No.80, pp.203-208, (1996).
11) 中條拓伯, 中野智行, 松本尚, 小畑正貴, 松田秀雄, 平木敬, 金田悠紀夫, "分散共有メモリ型超並列計算機 JUMP-1 におけるスケーラブル I/O サブシステムの構成," 情報処理学会論文誌, Vol. 37, No. 7, pp. 1429-1439, (1996).