

Windows NT クラスタでの類似文書検索システムの構築

入 口 浩 一[†] 岡 崎 史 裕^{††}
松 田 元 彦^{††} 佐 藤 三 久[†]

PC 8 台を利用した Windows NT クラスタによる並列類似文書検索システムは、Web 上での検索サービスの提供、大規模データの高速検索、開発時間の短縮を目的としている。構築にあたっては、新情報処理開発機構の並列分散技術 PM/Myrinet と、高速通信ミドルウェア SORB+JavaPM/Myrinet、住友金属工業株式会社の類似文書検索ライブラリを利用した。93 年から 98 年の特定分野の特許データ 14.7GB、760,489 件において、1 台の類似文書検索システムと比較した場合、検索速度は約 6 倍から 13 倍となる。

Design and Implementation of a similar sentence search system on Windows NT Cluster

HIROKAZU IRIGUCHI,[†] FUMIHIRO OKAZAKI,^{††} MOTOHIKO MATSUDA^{††}
and MITSUHISA SATO[†]

We are building a similar sentence search system on Windows NT Cluster using 8 machines. This system's goal is to realize WWW search server on very large size data, and to enable rapid prototyping of the system. We are using RWCP's PM/Myrinet, high speed middle wear SORB+JavaPM/Myrinet and Sumitomo Metal Industry's a similar sentence search library. This system achieve a good speed up from 6 to 13 times faster than single processor system for Japanese patent data from 1993 to 1998(14.7GB data size, 760,489 files).

1. はじめに

多数のユーザに同時サービスを提供するサーバシステムには、拡張性と高信頼性が必要である。クラスタ技術はこれを実現する技術として以前から利用されてきたが、プログラム設計の複雑さや価格が高価なことから、中小規模のサーバシステムに適用されることは稀であった。

ところがハード面でのプロセッサ性能の著しい進歩により、コストパフォーマンスに優れた高性能な PC クラスタが構築可能となり、サーバシステムとして注目されてきている^{1)~3)}。

そこで新情報処理開発機構では、PC クラスタのプラットホームとして、Windows NT を選択した。なぜなら広く普及していることから、いち早く低価格で高性能なデバイスや豊富なソフトウェアを使用できるからであり、もっとも普及している OS の Windows 95/98 との親和性も申し分ないためである。

一方、ソフト面では、現状、並列分散プログラムは、MPI⁴⁾、PVM⁵⁾ などメッセージ通信を利用するものや RPC、ORB、CORBA⁶⁾、HORB⁷⁾、RMI⁸⁾、DCOM⁹⁾ などリモート呼出しで記述するものが主流である。残念ながらこれらには、サーバーアプリケーションで必要とされる集団実行や耐故障の両機能をアプリケーションから透過的に記述できる機能がない。

そこで新情報処理開発機構では、ORB の抽象化されたインターフェースに集団実行と耐故障性を高めるための機能を実装するサーバーアプリケーション構築用並列分散ライブラリ (SORB: Sumikin ORB)¹⁰⁾ の開発を進めている。この SORB は、ORB のために必要なオブジェクトの直列化、リフレクションの機能を備えたオブジェクト指向言語 Java で開発した。

この SORB の対故障性機能実現時に発生する、ノード間のオブジェクトの移動によるトラフィックの増大と、クラスタ台数を増やした時のトラフィックに対応するために、Myricom 社のギガビットネットワーク Myrinet¹¹⁾ を用に開発された高速通信ライブラリ PM/Myrinet¹²⁾ を Java のマルチスレッド環境で利用可能とした、JavaPM/Myrinet¹⁰⁾ の開発を進めている。

本稿ではこれらの技術をベースとした、サーバーアプリケーションとして、類似文書検索を取り上げる。類似文

[†] 技術研究組合 新情報処理開発機構 並列分散システムパフォーマンスつくば研究室

Parallel and Distributed System Performance Laboratory,
RWCP

^{††} 技術研究組合 新情報処理開発機構 並列分散システム住友金属
Parallel and Distributed System Sumitomo Metal Laboratory,
RWCP

書検索は、キーワード検索、全文検索続く新しい検索として、大量の電子データがあふれている今日では非常に注目を集めている。しかしながら、大量のデータに対して検索を満足に行うためには、高価なサーバーを必要とするのが現実である。今、この現実を打破すべく住友金属工業株式会社の類似文書検索ライブラリを用いた、PC 8 台による Windows NT クラスタによる、並列類似文書検索システムの構築を進めている。このシステム上で、実際の 93 年から 98 年の特定分野の特許データ 14.7GB、760,489 件で、類似文書検索サーバーが 7 台と 1 台の場合の比較実験を行った。

本稿の構成は以下の通りである。2 章では我々の要素技術である JavaPM/Myrinet と、SORB、類似文書検索ライブラリについて述べる。3 章では、我々の構築した並列類似文書検索システムについて述べ、4 章で評価実験を行う。最後に 5 章でまとめと今後の課題について述べる。

2. 要素技術

この章では類似文書検索システムを構築するのに用いた我々の要素技術の JavaPM/Myrinet と、SORB、類似文書検索ライブラリについて述べる。

2.1 JavaPM/Myrinet

JavaPM は、PM/Myrinet による高速通信を Java のマルチスレッド環境の中で利用する機能である。Java のソケットクラスはファクトリパターン¹³⁾で設計されており、JavaPM はこの 1 実装として組込むことで、他の通信レイヤと同じインターフェースで利用できる。これにより、JavaPM はネットワークアプリケーション本体のプログラムを変更することなく、高速通信が可能である。

JavaPM は、Java 内でソケットエミュレート処理を行い、JNI¹⁴⁾を経由して PM ライブラリを呼出す実装にした。PM ライブラリを利用するメソッドは、Java の持つ排他制御によりマルチスレッド対応を行う。PM はメッセージの順序制御、到達確認の機能を持っている。そこで、JavaPM のソケットエミュレート処理は、1)PM の最大転送サイズの制限を超えたメッセージを処理するためのパケットの細分化と再組立て処理、2) メッセージをポート番号によりアプリケーションへ分配する処理、3)TCP フラグ回線の接続／切断の処理を行う。

2.1.1 送信処理概要

アプリケーションからの送信要求は、アプリケーションのスレッドで処理される。送信要求は、ソケットエミュレート処理した後、JNI を経由して PM の送信バッファにコピーして送信する。

PM の送信バッファが獲得できない場合には、通信相手側の受信処理が停滞していると考えられるので、CPU を他の処理で利用できるように、OS の最小単位である

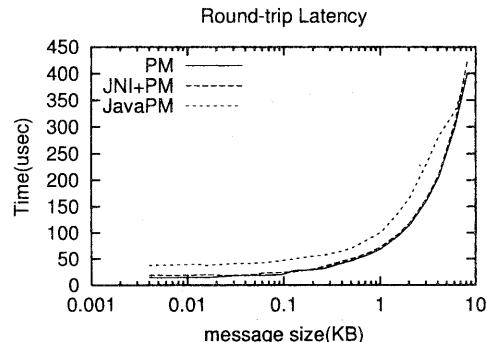


図 1 JavaPM/Myrinet のレイテンシ性能

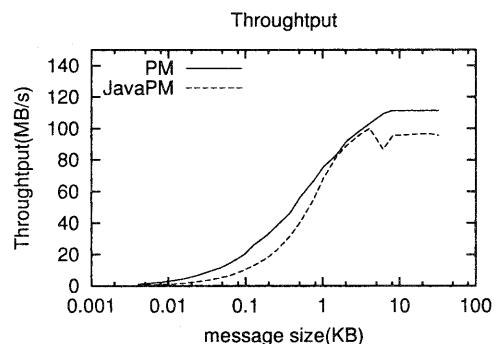


図 2 JavaPM/Myrinet のスループット性能

1msec のスリープをはさんでポーリングしている。

2.1.2 受信処理概要

PM は受信メッセージを読み出して受信バッファを開放する必要があるため、JavaPM は PM から受信メッセージを読み出す専用のスレッド（受信スレッド）を生成する。読み出したメッセージをアプリケーションに渡すまで蓄積しておくバッファ（蓄積バッファ）が必要であり、事前に 255 個準備している。未使用的蓄積バッファはキューパス（未使用キューパス）で管理し、受信メッセージを蓄積しているバッファはアプリケーション別に作成されたキューパス（ポートキューパス）で管理する。

2.1.3 基本性能

レイテンシ、スループットの性能測定には、同一の送信バッファへ転送する方法をとった。これは、Java ではオブジェクトストリームを含めてバッファリングされたストリームを使用する場合に相当する。図 1,2 で、「PM」は C 言語から直接 PM ライブラリ呼ぶ場合、「JNI+PM」は Java から JNI を経由して PM ライブラリを呼んだ場合を意味している。「JavaPM」は、受信処理がアプリケーションのスレッドで実行された場合を示している。

レイテンシは 2 ノード間でメッセージのピンポンで測定した。そのラウンドトリップ時間の計測結果を図 1 に

示す。JavaPM のメッセージ長 4byte でのレイテンシは、38usec であった。これは、PM のレイテンシ 14.8usec、受信処理／送信処理オーバヘッドの 2 倍の合計値と考えられる。

スループットは 2 ノード間のバースト転送で測定した。スループットの測定結果を図 2 に示す。6KB 時の性能悪化は、JavaPM ではヘッダとして 32bytes が追加され、PM のメッセージ長は 8224bytes となった時に PM の性能が劣化したことから、パケットサイズを 4KB に制限したことにより、メッセージがパケット 2 個に分割されたためと考えられる。

性能として、最大スループット 95MB/sec、レイテンシ 38usec である。

2.2 SORB

SORB は、Java のオブジェクト直列化機能¹⁵⁾を利用して、1) 集団実行機能、2) マイグレーション機能、3) 高信頼オブジェクト管理機能を、拡張した耐故障性を高める ORB である。現在、耐故障性の機能は実装中である。

SORB の特徴は、アプリケーションが一度ロックアップしたスタブを、JavaVM 内で一括管理するスタブ管理にある。スタブ管理により、アプリケーションから透過的にマイグレーション時の位置情報を制御できる。さらに、スタブの管理情報はキャッシュとしての役割を持ち、同一のリモートオブジェクトへのロックアップに対して高速に応答できる。同じリモートオブジェクトに対して同じスタブを共同で使用するので、スタブはマルチスレッド対応の必要がある。スタブ管理機能自身も SORB の集団実行機能を利用しておらず、SORB を利用するクライアントでは SORB サーバが動作している必要がある。

SORB のリモート呼出しへは、オブジェクト識別子・メソッド番号・引数オブジェクトを送信し、終了コード・戻りオブジェクトを受信する方法で行う。サーバ側では、登録時の名称、オブジェクト識別子、リモートオブジェクトとそのスタブを管理情報としてハッシュテーブルに登録する。

2.2.1 集団実行機能

集団実行機能は、1 回のメソッド呼出しで、複数のリモートオブジェクトのメソッドをマルチスレッドで同時に実行し、最後に全体の同期をとる方法で実装した。

ロックアップ時にホスト名の代わりにホスト名の配列から生成されるグループ識別子を指定することで集団実行スタブを獲得する。スタブは単一実行と集団実行で同じクラスを使用する。集団実行時には引数オブジェクトを各ノードにスキヤッターするメソッドと戻りオブジェクトをリダクションするメソッドが実行される。

2.2.2 マイグレーション機能

マイグレーション機能として、リモートオブジェクトを他のノードに移動可能にする機能と、他のノードの同

じ機能を持ったリモートオブジェクトに切替える機能を備える。切替機能は、スタブ管理機能で各ノードのスタブの位置情報を変更することで実装する。移動機能は次の手順で実装する。

1) 移動先での仮登録仮登録は、サーバ管理情報のリモートオブジェクトの項目を null で登録する。この状態では、オブジェクトへのロックアップは可能であるが、リモート呼出しの実行はオブジェクトが転送、初期化されるまでロックされる。

2) スタブの位置情報修正移動元サーバが各クライアントのスタブの位置情報をスタブ管理機能で移動先に修正し、移動元での登録を削除する。新しいリモート呼出しはすべて移動先に対して行われる。

3) 移動元からリモートオブジェクトの転送移動元で実行しているリモート呼出しがすべて終了した段階で、オブジェクトを移動先に転送する。

4) 移動先でのサービス開始移動先でリモートオブジェクトを登録し、初期化メソッドを呼出して移動できないフィールドを初期化する。ロックしていたリモート呼出しを実行してサービスを再開する。

2.2.3 高信頼オブジェクト管理機能

内部状態を持つリモートオブジェクトは、メソッドがリモート呼出しされた後に、クラスタ内にコピーを保存する。保存する方法は他のサーバにコピーをとるか、コピーを細分化してクラスタ全体に断片を分散して保存する。障害発生時には、このコピーを復元して他のサーバに登録した後、マイグレーションの切替機能を実行する。

2.2.4 基本性能

メモリへの byte 配列オブジェクトの転送性能による直列化性能の測定結果を図 3 に示す。図 3 で OutputStream はメモリへのオブジェクトの直列化、InputStream はメモリからの復元化、JavaPM は JavaPM を介しての直列、復元化性能である。SORB での各種配列オブジェクトの転送性能を図 4 に示す。なお、String は Array Size が示す文字列を持つ String オブジェクトである。最も高速である byte 配列オブジェクトの直列化性能に対して、JavaPM のオブジェクト転送性能は追従しない。この原因は、復元化のオーバヘッドと JavaPM の受信処理時間が加算された結果が転送性能となるからである。大きな配列に対して 1KB 単位に転送される直列化データを復元するのに平均 15.4usec、JavaPM の受信処理 15.8usec の合計値 31.2usec が処理に必要となる。性能は 32.0 MB/sec 程度と計算され、測定結果とほぼ一致する。JavaPM の処理時間によりオブジェクト転送性能は悪化するが、他のオブジェクトに対しては直列化性能が落ちるため、JavaPM の影響は小さくなる。

2.3 類似文書検索ライブラリ

類似文書検索の最大の特徴は、文書自体を検索キーと

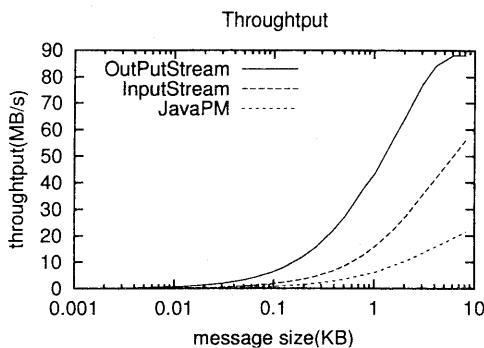


図 3 オブジェクトの直列化性能

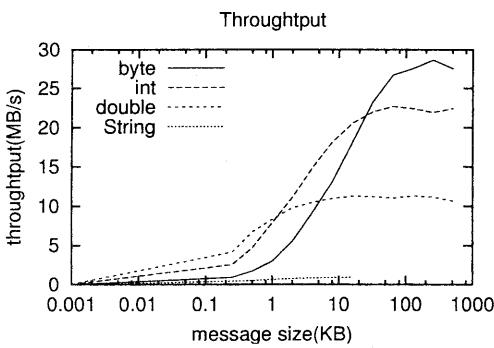


図 4 SORB のオブジェクト転送性能

することである。全文検索やキーワード検索のキーワードと論理式による検索キーとは違い、この文書自体の検索キーと検索対象文書のマッチングによって、類似文書検索は行われる。このマッチングの度合いは類似度と呼ばれ、このマッチングの仕方（類似度の計算）にはいろいろな手法がある。純粹にベクトル空間と見なすものや、計算コストを下げるために代表的なベクトルに絞るもの、上位概念でのマッチングを行うものなどがある。どの手法も、如何に計算量をへらしつつも精度が落ちないようになるか、という工夫をしている。我々が用いる並列文書検索ライブラリの知的検索ライブラリ K-Tech Search II(住友金属工業株式会社)は、独自の文書間の類似度を求めるアルゴリズムを用いて、従来のベクトル空間法などより精度の高い文書検索を実現している。1.7GBの文書データに対する平均の検索時間は5秒程度である。

また、検索時にすべての文書とマッチングを取っていたのでは、いつまでたっても終わらないので、大量の文書を扱う検索システムでは、あらかじめ、マッチングに都合の良い、文書から特徴を抽出したインデックスファイルを作成し、保存しておくのが一般的であり、このインデックスファイルのサイズも検索システムを選ぶ上で重要である。K-Tech Search II は、原文と同等のサイズであり、

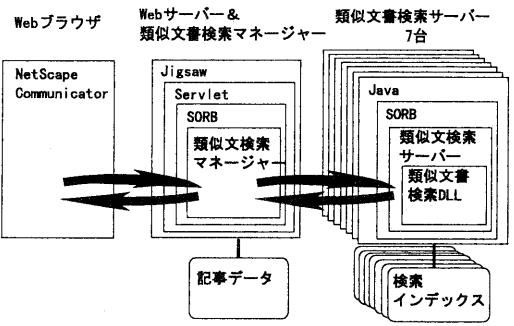


図 5 システム構成

その点では、他の全文検索システムと比べても^{16)~19)}遜色はない。インデックスファイルの作成性能は1.7GBの作成した場合に170MB/hであり、100MB作成した場合には300MB/hとなる。

3. 並列類似文書検索システム

8台のPCを、1台はWebサーバー兼類似文書検索マネージャー、残り7台を類似文書検索サーバーとして構築した。

3.1 Webサーバー兼類似文書検索マネージャー

WebサーバーとしてJigsaw²⁰⁾を利用した。JigsawはW3Cにより作成されたJavaで記述されているWebサーバーであり、JSDK(Java Servlet Developer Kit)²¹⁾を組み込むことにより、サーバー側JavaアプリケーションであるServletを実行できる。これにより、CGI等を利用する方法に比べ、Webブラウザからの検索要求を少ないシステム資源でかつ高速に扱うことができる。

類似文書検索マネージャーは、このServletとして動作し、後述の類似文書検索サーバーへの検索要求の分配と、検索結果の収集、ソートを行い、検索結果の表示とデータの表示を行う。

通信には、SORB+JavaPM/Myrinetを利用している。

3.2 類似文書検索サーバー

住友金属工業株式会社の類似文書検索ライブラリをJavaから利用するためにJNIを使っている。SORBを利用して、類似文書検索マネージャーからの検索要求に応える。また、類似文書検索ライブラリが利用する検索インデックスを保持する。

3.3 基本動作

Webブラウザ上からの検索要求は、Jigsaw上の類似文書検索マネージャーServletに渡され、SORBを利用して7台の類似文書検索サーバーに分配される。各類似文書検索サーバーで検索実行後、結果が類似文書検索マネージャーServletに集められ、ソートしてWebページを作成し表示する。

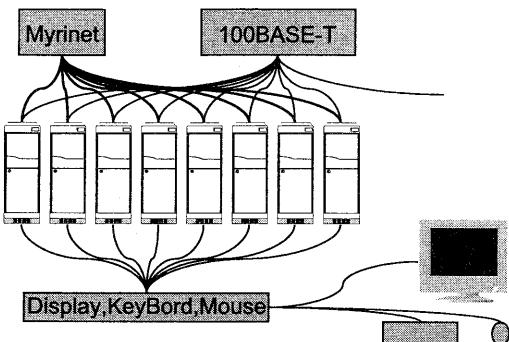


図 6 PC クラスタシステム

4. 評価実験

類似文書検索サーバーが 1 台の場合と 7 台の場合の検索速度の比較を行った。ハードウェアとソフトウェア、データを以下に示す。

4.1 ハードウェア

8 台構成のマシン構成を以下に示す。

- 400MHz Pentium II
- 128MB RAM
- SCSI HDD 9GB
- Myrinet Network card
- 100BASE-TX ethernet card

これらは、cisco fast ethernet switch と、Myrinet switch と Display switch で図 6 のように接続される。

4.2 ソフトウェア

実験に用いたソフトウェアは以下の通り。

- Windows NT 4.0 SP3
- JDK 1.2.1
- Jigsaw 2.0 beta3
- JSdk 2.0
- SORB 1.0 beta
- JavaPM/Myrinet 1.0 beta
- PM/Myrinet 1.4

4.3 データ

検索の対象として、93 年から 98 年の特定分野の公開特許のテキスト部分のみ (14.7GB, 760,489 件) を利用した。特許 1 件あたりの平均サイズは 20KByte である。そのデータを 7 台に以下の用に分割した状態が表 1 であり、a,b,c,d はそれぞれ四半期ごとを表している。

検索キーとして使う文書データは、特許データの中から無作為に 40 件選んだものである。

4.4 結果

実際に Web ブラウザ上から、40 件の検索を行った結果について、類似文書検索サーバーが 1 台の場合と 7 台の場合の検索時間を表したグラフと、1 台の場合の検索

表 1 特許データ		
年	サイズ (MB)	件数
93	352	26,448
94	400	28,333
95	379	24,606
96a	1,100	57,725
96b	1,100	57,715
96c	1,130	57,328
96d	1,290	64,707
97a	1,120	57,580
97b	1,120	57,624
97c	1,150	57,919
97d	922	45,552
98a	1,160	57,445
98b	1,200	58,015
98c	1,200	57,460
98d	1,090	52,032

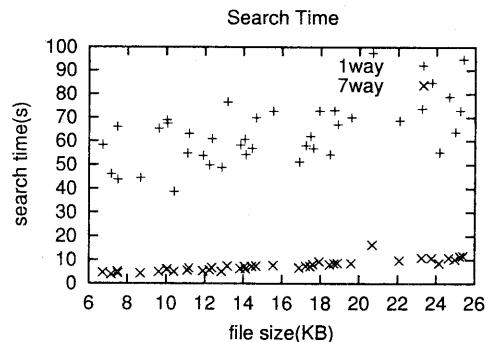


図 7 検索時間

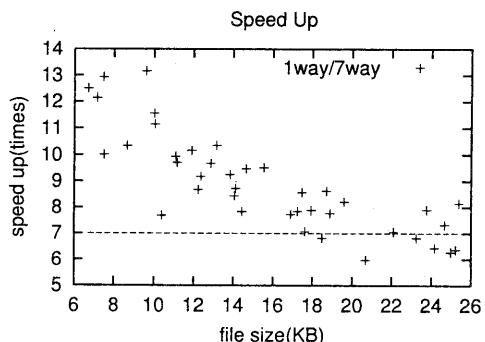


図 8 並列化の効果

時間を 7 台の場合の検索時間で割ったことにより並列化による効果を現したグラフを次に示す。並列化により検索速度は約 6 倍から 13 倍となる。

4.5 考察

並列化における速度向上が、特許文書の平均サイズの 20KB 付近まで 7 倍以上あることから、十分に満足でき

る結果が出ているといえる。

しかし、検索キーの文書のサイズが大きくなるにしたがって、速度向上割合が鈍っているのは、逆に言えば、サイズが小さいときに性能が良い(13倍)のは、現状での主記憶が128MBであることから、検索に必要なインデックスファイルに対して、ハードディスクに直接アクセスしていることが、原因であると考えられる。1台で動作させた場合には、7倍(14GB)のインデックスファイルにアクセスすることから、ほとんどWindows NTのファイルシステムのキャッシュにヒットせず、検索キーのファイルサイズが如何に関わらず、非効率的な状態で動作していることが考えられる。そして、検索キーのサイズが小さな場合には、通常の7台並列で検索しているときには、キャッシュが効率よく働き、検索速度が出ていると考えられる。検索キーのサイズが大きな場合には、キャッシュの効率が悪くなり、検索速度が落ちていると考えられる。

また、現在の検索速度自体については、並列化したとしても、3秒程度かかることから、メモリの増設やディスクの高速化、プログラムのチューニングなどによるさらなる速度向上が望まれる。

JavaPM/Myrinetを高速化のために実装し用いたが、現在の類似文書検索では、検索自体にかかる時間が大きすぎて、JavaPM/Myrinetの効果が現れてこない。しかしJava/PMは、現在実装中のSORBの対故障性機能を使ったときに、初めて有用になってくるものである。さらに、クラスタの台数を増やした場合にも有効であると考えられる。

5.まとめと今後の課題

8台のPCによるWindows NTクラスタを構築し、特許データによる性能実験を行った。現実的な検索実験により、並列化による6倍から13倍の速度向上の効果が確かめられた。

今回は、JavaPM/Myrinetのみであったが、通常のether net等の比較や、さらにデータが大きな場合(全件の公開特許データを準備中)の性能の向上を図っていく予定であり、メモリの増設による速度変化や、プログラムのチューニングを行う予定である。

課題としては、

- クラスタの進む方向性として、SMPによるクラスタが性能、値段、設置容積等の面から、期待されていることから、現在2CPUのWindows NTマシンを4台用いたクラスタ構築中であり、性能の検証を行う予定である。
- SORBの対故障性機能の実装が終わり次第、この機能によるパフォーマンスの低下や、対故障性の確認を行う予定である。

参考文献

- 1) ファーサード 訳, Mark A.Sportack : "Windows NT クラスタリング", 株式会社 プレンティスホール出版, (September 1998).
- 2) 湯浅太一・安村通見・中田登志之 編 : "bit 別冊はじめての並列プログラミング", 共立出版株式会社, (June 1998).
- 3) Beowulf project: <http://www.beowulf.org/>
- 4) The Message Passing Interface (MPI) standard, <http://www.mcs.anl.gov/mpi/>
- 5) PVM: Parallel Virtual Machine, ORNL, <http://www.epm.ornl.gov/pvm/>
- 6) CORBA(CORBA Success Stories): <http://www.corba.org/>
- 7) HORB: <http://ring.etl.go.jp/openlab/horb-j/>
- 8) Java Remote Method Invocation Specification: <http://java.sun.com/products/jdk/1.1/docs/guide/rmi/spec/rmiTOC.doc.html>, (1997)
- 9) The Distributed Component Object Model (DCOM): <http://www.microsoft.com/com/dcom.asp>
- 10) 岡崎史裕、松田元彦、入口浩一: "JavaPM/MyrinetとSORBの性能評価", 情処研報 Vol.99, No.103 pp.31-36, (December 1999).
- 11) Myrinet: <http://www.myri.com/myrinet/>
- 12) PM: High-Performance Communication Library, <http://www.rwcp.or.jp/lab/pdslab/pm/>
- 13) Erich Gammaほか: "オブジェクト指向における再利用のためのデザインパターン", ソフトバンク, 1995.
- 14) Java Technology: <http://java.sun.com/>
- 15) 小俣裕一 監訳, 豊福剛 訳, Jim Farly : "Java 分散コンピューティング", 株式会社 オライリー・ジャパン (September 1998).
- 16) 藤原鎮男 監訳, 細野公男・後藤智範・岸田和明 訳, Peter Ingwersen : "情報検索研究 認知的アプローチ", 株式会社 トッパン, (February 1995).
- 17) 原田昌紀 : "サーチエンジン徹底活用術", 株式会社 オーム社, (December 1997).
- 18) 馬場肇 : "日本語全文検索システムの構築と活用", ソフトバンク株式会社, (September 1998).
- 19) 馬場肇 : "日本語全文検索エンジンソフトウェアリスト", <http://kusastro.kyoto-u.ac.jp/~baba/wais/other-system.html>
- 20) Jigsaw The W3C's Web Server: <http://www.w3.org/Jigsaw/>
- 21) Java Servlets: <http://java.sun.com/products/servlet/>