

## Windows NT SMP クラスタでの SORB+JavaPM/Myrinet による 並列類似文書検索システムの評価

入口 浩一† 岡崎 史裕††  
松田 元彦†† 佐藤 三久†

文書間の類似度を計算することによって検索を行う類似文書検索は、一般的な全文検索と比べると検索結果を求めるための計算量がかなり多くなる。そのため、大規模なデータに対してこの類似文書検索を適用する場合には、応答時間が問題になってくる。そこで我々は類似文書検索の並列化を行い、5 台のデュアルプロセッサ PC を利用した並列類似文書検索システムを構築し、応答時間の短縮を目指した。構築にあたっては、高速通信ミドルウェア SORB+JavaPM/Myrinet と類似文書検索ライブラリ K-Tech Search II を用いた。一定の時間間隔で検索を実行する検索クライアントを作成し、応答時間の計測によって本システムの評価を行った。99 年の全分野の公開特許データ 9.7GB、382,113 件において、類似文書検索サーバー 1 台の類似文書検索システムと比較した場合、約 51 倍から 1321 倍の応答時間の向上を得られた。

### Evaluation of a Parallel Similar Document Search System on Windows NT SMP Cluster using SORB+JavaPM/Myrinet

HIROKAZU IRIGUCHI,<sup>†</sup> FUMIHIRO OKAZAKI,<sup>††</sup> MOTOHIKO MATSUDA<sup>††</sup>  
and MITSUHISA SATO<sup>†</sup>

We are building a parallel similar document search system on Windows NT SMP Cluster using 5 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 high speed middle wear SORB+JavaPM/Myrinet and Sumitomo Metal Industry's a similar document search library K-Tech Search II. This system achieve a good speed up from 51 to 1321 times faster than single processor system for Japanese patent data 1999(9.7GB data size, 382,113 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<sup>1)~3)</sup>。

さらに、最近では設置スペースや、設定の容易さから複数の CPU を持った PC に対する関心も高く、2CPU のものであれば以前に比べ安価に手に入れることが可能

となってきた。

そこでわれわれは、SMP による PC クラスタとし、プラットフォームとして、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、MPI<sup>4)</sup>、PVM<sup>5)</sup> などメッセージ通信を利用するものや RPC、ORB、CORBA<sup>6)</sup>、HORB<sup>7)</sup>、RMI<sup>8)</sup>、DCOM<sup>9)</sup> 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無い。

そこで我々は、ORB の抽象化されたインタ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アプリケーション構築用並列分散ライブラリ (SORB: Sumikin ORB)<sup>11)</sup> 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

† 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

した。

このSORBの耐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 Myrinet<sup>12)</sup>用に開発された高速通信ライブラリ PM/Myrinet<sup>13)</sup>をJavaのマルチスレッド環境で利用可能とした、JavaPM/Myrinet<sup>11)</sup>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 K-Tech Search II を用いた、PC 5台による Windows NT SMP クラスタによる、並列類似文書検索システムの構築を行った。このシステム上で、実際の99年の全分野の特許データ 9.7GB、382,113件で、類似文書検索サーバーが4台と2台、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/Myrinetは、Myricom社のギガビットネットワーク Myrinet用に開発された高速通信ライブラリ PM/MyrinetをJavaのマルチスレッド環境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<sup>14)</sup>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、JNI<sup>15)</sup>を経由してPMライブラリを呼出す実装である。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越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仮想回線の接続/切断の処理を行う。

### 2.2 SORB

SORBは、Javaのオブジェクト直列化機能<sup>16)</sup>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り、SORBを利用するクライアントではSORBサーバーが動作している必要がある。

SORBのリモート呼出しは、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。サーバー側では、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。

### 2.3 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。純粋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するか、という工夫をしている。

我々が用いる類似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。

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、計算量が多すぎ時間が掛かりすぎるので、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

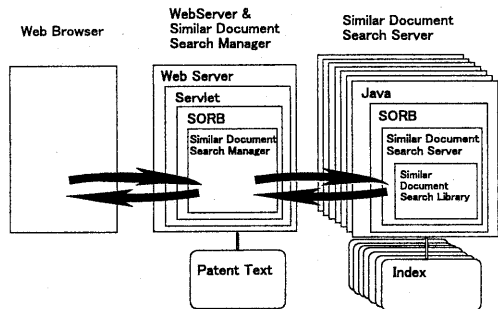


図1 システム構成

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、その点では、他の全文検索システムと比べても<sup>17)~20)</sup>遜色はない。

しかしながら、この類似文書検索ライブラリーには、検索インデックスに関して、Windows プラットホームに依存した問題点がある。それは、ファイルサイズの問題で通常の方法で扱えるのが2GB までとなっていることである。そのため、この類似文書検索ライブラリーで扱う検索インデックスは2GB が限界となる。しかし、逆にこの限界にあわせてデータ構造とアルゴリズムを考案することで、検索のスピードアップと検索インデックスの容量の少量化を行っているため、容易にこの限界を超えることができない。したがって、2GB 以上の辞書が必要な場合には、複数の検索インデックスに分割して、順次検索処理を行い最後にマージするという方法を採用している。

### 3. 並列類似文書検索システム

並列類似文書検索システムは、1台のWeb サーバー兼類似文書検索マネージャーと複数台の類似文書検索サーバーで図1に示すように構成される。本稿の実験では、4台、2台、1台の場合を取り上げている。

基本的な動作はWeb ブラウザー上からの検索要求は、Servlet の類似文書検索マネージャーに渡され、SORB を利用して複数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に集められ、ソートしてWeb ページを作成し表示する。

以下、Web サーバー兼類似文書検索マネージャーと、類似文書検索サーバーについて説明を行う。

#### 3.1 Web サーバー兼類似文書検索マネージャー

Web サーバー兼類似文書検索マネージャーは、Web ブラウザーからの検索要求を受けるWeb サーバーと、Servlet エンジン上で動作する類似文書検索マネージャー

で構成される。

Web サーバーへの検索要求は、Servlet エンジンを経由してServlet として動作している類似文書検索マネージャーへ渡される。類似文書検索マネージャーは、後述の類似文書検索サーバーへの検索要求の分配を行い、類似文書検索サーバーからの検索結果を収集し、ソートを行い、検索結果の表示を行う。

類似文書検索マネージャーと類似文書検索サーバー間の通信にはSORB を利用し、リモートオブジェクトへのルックアップの高速化し、現在実装中の耐故障性への対応を容易にする。

また、Servlet を利用することにより、CGI 等を利用する方法に比べ、Web ブラウザーからの検索要求を少ないシステム資源で高速に扱うことができ<sup>23)</sup>、さらにServlet API というインターフェースに統一されているため、容易にServlet エンジンを入れ替えることができる利点がある。

#### 3.2 類似文書検索サーバー

類似文書検索サーバーは、Windows プラットホーム上で開発された住友金属工業株式会社の類似文書検索ライブラリー K-Tech Search II をJNI を介してJava 上から利用している。

類似文書検索マネージャーからSORB を通しての検索要求に応え、類似文書検索ライブラリーを利用して検索した検索結果を返す。また、類似文書検索ライブラリーが利用する検索インデックスを保持している。

## 4. 評価実験

並列類似文書検索システムの検索要求間隔(100ms, 150ms, 200ms)における応答時間を、ネットワーク(100BASE-TX, PM/Myrinet)や検索インデックス数(1,2,4,8)、類似文書検索サーバー数(1,2,4)が違つ場合について実験評価を行った。その実験環境(ハードウェアと、ソフトウェア、データ)と実験内容、結果と考察を以下に示す。

#### 4.1 実験環境

##### ハードウェア

並列類似文書検索システムは、類似文書検索サーバー4台(表1)とWeb サーバー兼類似文書検索マネージャー1台(表2)、検索クライアント1台(表3)、の計6台で構成されており、図2に示すように、100BASE-TX Ethernet switch と Myrinet switch、Display switch で接続されている。

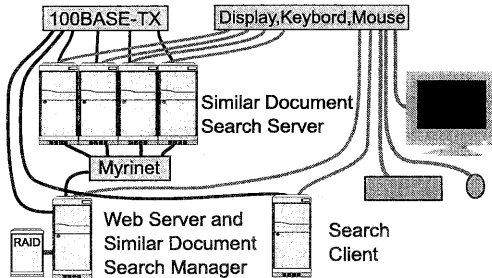


図2 PCクラスシステム

表1 類似文書検索サーバー4台

CPU	866MHz Pentium III x 2
MEMORY	1GB
HDD	36GB (SCSI Ultra160)
NIC	100BASE-TX Ethernet on board
NIC	Myrinet Network Card

表2 Webサーバー兼類似文書検索マネージャー1台

CPU	550MHz Pentium III x 2
MEMORY	1GB
HDD	9GB (SCSI U2W)
RAID	400GB (SCSI UW)
NIC	100BASE-TX Ethernet on board
NIC	Myrinet Network Card

表3 検索クライアント1台

CPU	550MHz Pentium III x 2
MEMORY	1GB
HDD	9GB (SCSI U2W)
NIC	100BASE-TX Ethernet on board

ソフトウェア

OS	Windows NT 4.0 SP5
JVM	JDK 1.3 <sup>15)</sup>
Middleware	SORB 1.0 beta JavaPM/Myrinet 1.0 beta PM/Myrinet 1.4.1
Servlet Engine	Resin 1.2.1 <sup>22)</sup>

データ

検索インデックスとして、99年の全分野の公開特許のテキスト部分(9.7GB, 382,113件)を利用した。特許テキスト1件あたりの平均サイズは26KByteであり、その特許テキストデータを8分割した状態で、検索インデックスを作成した。8つの検索インデックスのサイズと特許テキストサイズ及び件数を表4に示す。

また、実験で検索キーとして使う文書は、特許データの中から無作為に1000件選んだものである。この選んだ検索キーのサイズは全体で22MB、平均で22KBである。

4.2 実験内容

特許データの中から無作為に選んだ1000件(平均サイズ22KB)を検索キーとして、後述する検索クライアントを用いて、検索要求の間隔を100ms,150ms,200msと変

表4 検索インデックスと特許サイズ、件数

辞書名	インデックス (MB)	特許サイズ (MB)	特許件数
199900	1,154	1,130	45,234
199901	1,323	1,297	50,551
199902	1,301	1,274	49,931
199903	1,322	1,296	50,304
199904	1,444	1,318	50,243
199905	1,362	1,336	50,328
199906	1,333	1,307	50,242
199907	961	9,40	35,279

化させた場合の応答時間を計測する。

ネットワークが100BASE-TXの場合とJavaPM/Myrinetの場合において、まず、検索インデックス数を1,2,4,8と変化させた場合の単体類似文書検索サーバーの基本性能を計測を行い、つぎに、類似文書検索サーバーの台数を1,2,4と変化させた場合の並列性能の計測を行った。最後に、デュアルプロセッサのSMPクラスタであることから、2個の類似文書検索サーバーをPC上で動かした場合について計測を行った。

4.2.1 検索クライアント

任意の文書を検索キーとして、任意の時間間隔でWebサーバーに対して検索をおこなうJavaプログラムであり、Webブラウザ上からの検索動作をシミュレートしている。

本実験では、特許データの中から無作為に選んだ1000件(平均サイズ:22KB)を100ms,150ms,200msの時間間隔でWebサーバーに対して送り出す。

正確な時間間隔で検索要求を送り出すことが必要であるため、あらかじめ文書を読み込んでおくことにより、文書の読み込みに掛かる時間の影響が出ないように実装を行った。

さらにスレッドの生成時間も誤差原因となるため、あらかじめスレッドを生成して待機させておき、あらかじめ指定しておいた時間がきたら、検索要求を送り出す実装になっている。

それぞれの検索要求に対して個別にスレッドを生成させているので、メモリ使用量は多くなるが、正確な時間間隔で検索要求を出すことが可能となっている。

4.3 実験結果及び考察

4.3.1 類似文書検索サーバーの基本性能

並列化を行った時のオーバーヘッドを求めるために、8つに分割した検索インデックスを単体のサーバーに1,2,4,8と与えたときの、検索要求間隔100ms,150ms,200msにおける応答時間をそれぞれのネットワークで計測を行った。100BASE-TXでの平均応答時間を表5に、JavaPM/Myrinetでの平均応答時間を表6に、100BASE-TXの平均応答時間をJavaPM/Myrinetで割った応答時間向上度を表7に示す。

これらの表からわかることは、検索インデックス数が

表 5 平均応答時間-100BASE-TX-(ms)

インデックス数	100ms	150ms	200ms
1	1,198	354	281
2	14,026	1,047	616
4	72,134	45,571	20,790
8	816,315	790,349	765,622

表 6 平均応答時間-JavaPM/Myrinet-(ms)

インデックス数	100ms	150ms	200ms
1	560	316	256
2	13,286	944	536
4	69,748	42,948	18,560
8	803,462	779,459	755,376

表 7 平均応答時間向上度

インデックス数	100ms	150ms	200ms
1	2.14	1.12	1.09
2	1.06	1.11	1.15
4	1.03	1.06	1.12
8	1.02	1.01	1.01

4 個以上になると急激に平均応答時間がふえていくことで、システムのキャッシュが十分に効かない状態になっていると考えられる。したがって、この結果から、満足のいく応答時間が得られる組み合わせは、検索インデックスが 1 個と 2 個の場合であるといえる。

また、JavaPM/Myrinet は向上度において一つの例外をのぞいては 10%前後であり、それほど有効であるとはいえない。

#### 4.3.2 並列分散性能

検索クライアントより検索要求間隔 100ms, 150ms, 200ms で送り出した検索要求 1000 件に対して、類似文書検索サーバーを 4 台 2 台 1 台とした場合にそれぞれのネットワークにおいて実験を行った。いずれの場合も 8 個の検索インデックスを均等に分けているため、検索対象となる検索インデックスの総量は同じである。

表 8 は、100BASE-TX における平均応答時間、表 9 は、JavaPM/Myrinet における平均応答時間を示す。また、100BASE-TX の平均応答時間を JavaPM/Myrinet で割った応答時間向上度を表 10 に示す。

これらの表より、4 台で並列化を生かした類似文書検索と単一のマシンでの類似文書検索においては、平均応答時間において、100BASE-TX では 44 倍から 995 倍、JavaPM/Myrinet では 51 倍から 1321 倍の性能差があることがわかる。JavaPM/Myrinet を利用することで最大 35%平均応答時間が向上することがわかる。

また、基本性能の表と比べることによって、並列化におけるオーバーヘッドが求められることができる。表 11 は、表 8 から検索インデックスの数に応じた表 5 の値を減算した 100BASE-TX でのオーバーヘッドを表し、表 12 は、表 9 から検索インデックスの数に応じた表 6 の値を減算し

表 8 平均応答時間-100BASE-TX-(ms)

マシン数	インデックス数	100ms	150ms	200ms
4	2	18,512	1,609	769
2	4	72,093	47,363	20,417
1	8	816,315	790,349	765,622

表 9 平均応答時間-JavaPM/Myrinet-(ms)

マシン数	インデックス数	100ms	150ms	200ms
4	2	15,880	1,200	572
2	4	70,856	45,730	20,827
1	8	803,462	779,459	755,376

表 10 平均応答時間向上率

マシン数	インデックス数	100ms	150ms	200ms
4	2	1.17	1.34	1.34
2	4	1.02	1.04	0.98
1	8	1.02	1.01	1.01

表 11 並列化によるオーバーヘッド-100BASE-TX-(ms)

マシン数	インデックス数	100ms	150ms	200ms
4	2	4,486	562	153
2	4	-41	1,792	-373
1	8	0	0	0

た JavaPM/Myrinet でのオーバーヘッドを表している。応答時間に対しておおむね 10%程度のオーバーヘッドとなっており、並列化の影響はあまり出ていないといえる。

#### 4.3.3 SMP 性能

検索クライアントより検索要求間隔 100ms, 150ms, 200ms で送り出した検索要求 1000 件に対して、マシン数を 4 台 2 台 1 台とした場合に SMP を利用できるようにそれぞれ二つずつ類似文書検索サーバーを立ち上げた時の応答時間の計測を行った。いずれの場合も 8 個の検索インデックスを均等に分けているため、検索対象となる検索インデックスの総量は同じであるが、前項とは違い、それぞれ類似文書検索サーバーあたりの検索インデックス数は半分になる。

表 13 は平均応答時間を、表 14 は表 13 から検索インデックスの数に応じた表 5 の値を減算した SMP でのオーバーヘッドを表している。

予想に反しマシン数 1 台の場合には性能が悪化している。単一マシンでのそれぞれ大きなサイズのインデックスを扱っているため、ディスクキャッシュの不整合が顕著になったものと考えられる。また、マシン数 4 台の場合には性能がほとんど向上していない。これは、表 15 にあるように、接続拒否の回数が増えていることから、類似文書検索サーバー数増加によって、Servlet エンジン内の

表 12 並列化によるオーバーヘッド-JavaPM/Myrinet-(ms)

マシン数	インデックス数	100ms	150ms	200ms
4	2	2,594	256	36
2	4	1,108	2,782	2,267
1	8	0	0	0

表 13 平均応答時間 (ms)

マシン数	インデックス数	100ms	150ms	200ms
4	1	15,843	1,521	1,037
2	2	24,389	2,275	876
1	4	1,214,714	1,279,374	1,163,604

表 14 並列化+SMP 化によるオーバーヘッド (ms)

マシン数	インデックス数	100ms	150ms	200ms
4	1	13,447	813	475
2	2	10,363	1,228	260
1	4	1,142,580	1,233,803	1,142,814

表 15 接続拒否数

マシン数	100ms	150ms	200ms
4-SMP	220	110	76
4	25	13	10
2-SMP	26	13	0
2	10	4	0
1-SMP	9	9	0
1	0	0	0

オブジェクトの利用量が増加し、JavaVM のガベージコレクションに負荷をかけているためと考えられる。

オーバーヘッドという観点からも、マシン数を増やした場合と比べて、悪化しており、単純に類似文書検索サーバーを 2 台に増やすだけでは、通信量や、マージに掛かるオーバーヘッドが問題となるということがわかった。

## 5. まとめと今後の課題

5 台の PC による Windows NT SMP クラスタを構築し、特許データによる性能実験を行った。実験により、4 台による類似文書検索サーバーの並列化によって約 51 倍から 1321 倍の応答時間の向上を得られること、JavaPM/Myrinet による応答時間の向上は、最大 35% であること、単純に同一マシンで二つ類似文書検索サーバーを立ち上げるだけでは SMP の性能を十分に活かさないことがわかった。

今後の課題としては、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と、JavaVM のガベージコレクションによる性能劣化をできるだけ防ぐため Servlet 内でのデータ構造の検討を行い、メモリの消費の少ないコードへと変換を行うことが上げられる。

## 参 考 文 献

- 1) Mark A.Sportack, ファサード 訳: "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) 入口 浩一, 岡崎 史裕, 松田 元彦, 佐藤 三久: "Windows NT クラスタでの類似文書検索システムの構築", 情処研報 Vol.2000, No.21 pp.49-54, (2000).
- 11) 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).
- 12) Myrinet: <http://www.myri.com/myrinet/>
- 13) PM: High-Performance Communication Library, <http://www.rwcp.or.jp/lab/pdslab/pm/>
- 14) 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.
- 15) Java Thechnology: <http://java.sun.com/>
- 16) Jim Farly, 小俣裕一 監訳, 豊福剛 訳: "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).
- 17) Peter Ingwersen, 藤原鎮男 監訳, 細野公男・後藤 智範・岸田和明 訳: "情報検索研究 認知的アプローチ", 株式会社 トップラン, (February 1995).
- 18) 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).
- 19) 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).
- 20) 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>
- 21) Java Servlets: <http://java.sun.com/products/servlet/>
- 22) Resin servlet and JSP engine: <http://www.caucho.com/products/resin/>
- 23) Scott Ferguson: JSP vs mod\_php and mod\_perl, <http://www.caucho.com/articles/benchmark.xtp>