分散型 WWW ロボットによる WWW 情報収集

山名早人 †1, 田村健人 †2, 河野浩之 †3, 亀井聡 †3, 原田昌紀 †4,
西村英樹 †5, 浅井勇夫 †6, 楠本博之 †7, 篠田陽一 †8, 村岡洋一 †9

†1 電子技術総合研究所 情報アークテクチャ部
〒 305-8568 茨城県つくば市梅園 1-1-4 Tel:0298-54-5955, E-Mail:yamana@etl.go.jp
†2 日本 IBM 東京基礎研究所， †3 京都大学大学院 工学研究科， †4 東京大学大学院 総合文化研究科
†5 シャープ (株) 技術本部， †6 大阪府立大学 工学部， †7 慶應義塾大学 環境情報学部
†8 北陸先端科学技術大学院大学 情報科学研究科， †9 早稲田大学 理工学部

あらまし

WWW の急速な普及に伴い、インターネット上には、現在 200 万台を越える WWW サーバが存在する。これらの WWW サーバから発信される情報を検索するためには、これらの情報を収集しデータベースを構築する必要がある。そこで、本稿では、 WWW サーバ上のデータを高速に収集する手法として、分散型 WWW ロボットを提案する。現在、国内の 8 個所に分散型 WWW ロボットを設置し実験を行っており、最終的には、日本国内の WWW サーバ上にあるデータを 24 時間以内に収集することを目標としている。予備評価の結果、 4 つの WWW ロボットを用いることにより、 1 つの WWW ロボットを用いた場合に比較し、 5.8 ~ 9.7 倍の速度向上が得られることを確認した。また、 n 台の分散型 WWW ロボットを用いることにより、最大 $(2.8 \times n)$ 倍程度の速度向上が期待できることがわかった。

WWW Information Collection with Distributed WWW Robots

Hayato YAMANA†1, Kent TAMURA†2, Hiroyuki KAWANO†3, Satoshi KAMEI†3,
Masanori HARADA†4, Hideki NISHIMURA†5, Isao ASAI†6, Hiroyuki KUSUMOTO†7,
Yoichi SHINODA†8, Yoichi MURAOKA†9

†1 Electrotechnical Laboratory, Computer Science Div.
1-1-4 Umezono Tsukuba-city, Ibaraki-pref, 305-8568 Tel:0298-54-5955, E-Mail:yamana@etl.go.jp
†2 IBM Tokyo Research Laboratory, †3 Kyoto University, †4 University of Tokyo,
†5 Sharp Corporation, †6 Osaka Prefecture University, †7 Keio University,
†8 Japan Advanced Institute of Science and Technology, †9 Waseda University

Abstract

Currently, the number of World-Wide Web servers becomes over two millions because of the rapid spred of WWW. The documents on the WWW servers must be collected to make a database to search them. In this paper, we propose distributed WWW robots to collect the documents quickly. Our final goal is to collect all of the documents on the WWW in Japan within one day. Currently, eight distributed WWW Robots are running in Japan. The experimental results show that we are able to gain 5.8 to 9.7 times speedup when four distributed WWW robots are placed at different places in comparison with when only one WWW robot is used. We also expects that we are able to gain about $(2.8 \times n)$ times speedup at most when we use n WWW robots to collect the documents.

1 まえがき

本稿では、複数のWWW(World Wide Web)ロボットを協調動作させることにより、WWWサーバ上のデータを高速に収集する分散型WWWロボットを提案すると共に、これまでの実験結果について報告する。

WWWは、Mosaicが開発されて以来、インターネット上での情報提供及び情報収集の手段として世界中で利用されている。しかし、WWW自体は、複数のWWWサーバから特定の情報を検索するプロトコルを持たないため、WWW上のデータを対象とした検索を行うためには、既存のWWWアーキテクチャ上で検索システムを構築しなければならない。このようなシステムの代表例として、AltaVista [1]やHotBot [2]等が挙げられる。これらは、HTTPクライアントの一種であるWWWロボット [3]と呼ばれるソフトウェアを用いてWWWサーバ上のデータを収集し、収集したデータに対する検索を提供している。WWWロボットは、次々とWWWサーバにアクセスし、HTMLファイルを自動的に取得する。そして、収集したHTMLファイルに書かれたリンクを辿り、別のWWWページを取得するという一連の動作を繰り返すことにより、WWW上のデータを収集している。

現在のWWWロボットは、AltaVistaやHotBotをはじめとする各々の検索システムが個別に動かしており、世界中に約160のWWWロボットが存在する [4]。そして、一つのWWWサーバに対して数十のWWWロボットが訪れているのが現状である。このような状況は、ネットワークトラフィック及びWWWサーバに与える負荷の増大をもたらしている。また、WWWサーバ数は、指数関数的に増大しており(1998年3月現在約208万台 [5])、WWWロボットでの収集時間も指数関数的に増大している。例えば、AltaVistaであるサイトを検索したところ、96年7月16日作成のページがインデックス化されたのは98年1月27日であった。このように、最新情報に対する検索が困難な状況となってきている。

このような問題を解決する方法として、これまでに、Harvest [6]が提案されている。Harvestでは、Gathererと呼ぶWWWロボットをネットワーク上の複数個所に配置し分散収集をすることができる。また、Brokerと呼ぶインタフェースを用いることにより、複数のGathererが収集したデータに対する検索を可能としている。しかし、分散収集時のWWWサーバの割り当ては、静的にしか設定することができず、分散を自動的に行うための仕組みを持たない。

これに対して本論文では、複数のWWWロボットが、互いに重複しないWWWサーバのデータを協調して短時間で収集するための分散型WWWロボットを提案する。本論文で提案する分散型WWWロボットは、担当するWWWサーバを自動的に決定する機能を持ち、動的にWWWサーバの担当を変更することができる。

以下では、従来のWWWロボットの問題を示すと共に、分散型WWWロボットを提案し、日本国内のWWWサーバを対象に8個所で分散収集を行っている状況について報告する。

2 従来のWWWロボットの問題

WWWロボットは、次のような問題を持つ。

① ネットワークとWWWサーバの負荷問題

現在のWWWロボットは、検索システム毎に別々に動作しているため、一つのWWWサーバに対して数十のWWWロボットが訪れている。このような多数のWWWロボットによる同一WWWサーバへのアクセスは、ネットワーク及びWWWサーバに大きな負荷を与えている。

② WWWページ取得時間増大の問題

WWWサーバ数は、図1に示すように指数関数的に増大しており、全てのデータを一ヶ所に収集し検索システムを提供することは、困難になりつつある。実際に、1996年の秋以降、商用検索システムの収集URL数の増加が鈍化し、現在は約8,000万URLで頭打ちになっている [7]。これは、WWWロボットを動作させるホスト及び回線容量がボトルネックになっているためである。例えば、早稲田大学で開発した検索システム「千里眼 [8]」のWWWロボットは、50万ページの収集に約1ヶ月を要している。

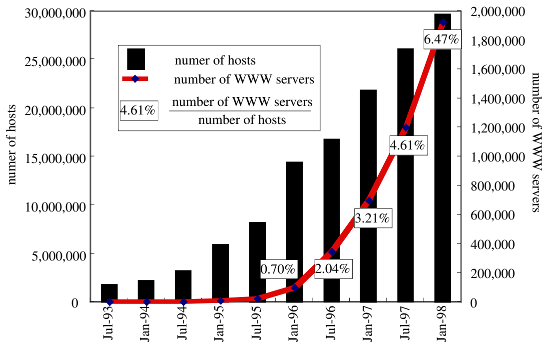

図 1: インターネットに接続されるコンピュータ台数と WWW サーバ台数の推移 ([5] [9] のデータに基づき作成)

③ 最新情報の網羅性欠如問題

WWW ページは頻繁に更新されるため、最新情報に対する検索を実現するには、WWW ロボットで再度収集して、検索システムの索引を作り直す必要がある。しかし、現在の WWW は、サーバ側からページの更新をクライアントに知らせる機能をもたず、WWW ロボットが、一つ一つのページにアクセスしてみるまでは、更新されているかどうかが分からない。このため、常に最新のデータを検索対象とするには、既に収集されたページについて、更新チェックを頻繁に行わなければならない。例えば、Infoseek [10] では、既に収集したページに対して、1ヶ月後に更新チェックを行い、該当ページが更新されていれば、次は 2 週間後に更新チェックをするといったように、更新チェック間隔を自動的に変える方式をとっているが、根本的な解決策ではない。

3 分散型 WWW ロボット

前節で挙げた問題点を解決するため、複数の WWW ロボットを協調動作させ、互いに重複しない WWW サーバのデータを収集することにより、WWW データの網羅的な収集を高速に行う。さらに、WWW ロボットを動作させている多くのサイトと協調することにより、現在のように数十の WWW ロボットが同一の WWW サーバを訪れることのないようにすることを目指す。

分散型 WWW ロボットは、図 2に示すように、全体を管理する Public Robot Server Manager(PRSM) と個々の WWW ロボットである Public Robot Server(PRS) から構成される。PRSM は、PRS に対して担当 WWW サーバの分配や、各 WWW サーバと PRS 間との距離計測を指示する。一方、PRS で新規に発見された WWW サーバや距離計測の結果は、PRSM に送られる。このようにして、PRS は PRSM からの指示に基づき各々互いに重複しない WWW サーバを担当し WWW データを収集する [11]。

収集されたデータは、最終的に図中の Search Service Server に再配布することにより、検索サービスのためのインデックス作成を行う。現在の HTTP/1.1 準拠のサーバは、1 回の接続で複数のデータを転送する Keep-Alive 機能を持っているが、複数のデータをまとめて転送する機能は持たない。このため、各 PRS で収集したデータを Search Service Server に再配布する際のオーバヘッドを小さくするため、複数のデータを 1 回の接続でまとめて圧縮 (Info-zip を利用) して送る機能、及び、Search Service Server からの要求に応じてデータ中の必要な部分のみ (例:URL や更新日時の指定による指定) を送る機能を PRS に持たせる。

分散型 WWW ロボットにおいては、PRS と WWW サーバ間のデータ転送速度が、全体の収集時間を決定する大きな要因となる。このため、PRS の分担では、以下に示すように、PRS と WWW サーバ間のサイト間距離を考慮した分散を行う。

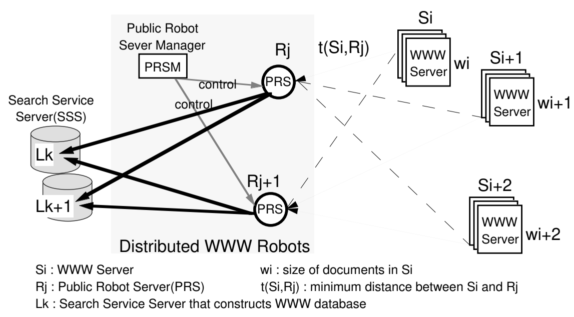

図 2: 分散型 WWW ロボットの構成

① モデル及びコスト定義 [12]

図 2に示すモデルを用い PRS(R_j) が WWW

サーバ (S_i) のデータを収集する収集コスト、及び、PRS(R_j) が Search Service Server (L_k) へデータを再配布する再配布コストを定義する。これらのコストは、S_i のドキュメント量を w_i とすると、w_i とサイト間距離 ($t(S_i, R_j)$) との積として定義できる。したがって、WWW サーバ S_i のドキュメントを PRS R_j が取得する場合、発生する収集コストは、$w_i t(S_i, R_j)$ となる。また、PRS R_j で収集した WWW サーバ S_i のデータを Search Service Server L_k へ再配布するコストは、再配布時のデータ圧縮率を K とすると、$K w_i t(R_j, L_k)$ となる。

② ネットワーク負荷と分散収集時間 [12]

全 WWW データ ($S_i(i = 1..n)$) を PRS ($R_j(j = 1..m)$) で収集し、Search Service Server($L_k(k = 1..l)$) へ全収集データを再配布する際に必要な収集コストと再配布コストの総和は、各 PRS R_j の担当する WWW サーバ群を集合 V_j で表すと、

$$\sum_{j=1}^{m} \sum_{S_i \in \{\boldsymbol{V_j}\}} \{w_i \times t(S_i, R_j) + K \times w_i \times \sum_{k=1}^{l} t(R_j, L_k)\} \quad (3.1)$$

となる。これはネットワークに与える総負荷を示している。また、複数の PRS の内、律速となる PRS が全体の実行時間を決定するため、分散収集時間は、以下の式で表すことができる。

$$\max_{j=1,\ldots,m} \Big(\sum_{S_i \in \{\boldsymbol{V_j}\}} \{w_i \times t(S_i, R_j) + K \times w_i \times \sum_{k=1}^{l} t(R_j, L_k)\}\Big) \quad (3.2)$$

③ ネットワーク負荷及び分散収集時間の最小化 [12] 分散収集時間を最小化するために、まず、式 (3.1) のネットワーク負荷を最小にする $V_j(j = 1..m)$ を与えるアルゴリズム (アルゴリズム **1**) を考え、次に分散収集時間を最小化するアルゴリズム (アルゴリズム **2**) を考える。

A) 欲張り法 (*greedy method*)
– アルゴリズム 1

式 (3.1) の最小化を行う分割は、欲張り法を用いることで、厳密解を得ることができる。

1. サイト S_i を選択。
2. すべての R_j について収集コストと再配布コストを計算し、これらの和が最小となる j を決定。
3. 集合$\boldsymbol{V_j}$に S_i を加える。
4. 1,2,3 をすべての i について繰り返す。
5. $\boldsymbol{V}$j ($j = 1, \ldots, m$) を出力。

B) 近似解導出法 – アルゴリズム 2

アルゴリズム 1 による分割後、律速となる PRS への負荷を他へ分散する。この際、「コストが最も大きい PRS の分担を減らし」、「その変更による全体のコストの増加が最も少なくなる」ように分担を変更し式 (3.2) を最小化するための近似解を得る。

1. アルゴリズム 1 を実行。
2. 収集コストと再配布コストの和が最も大きい PRS を選択。
3. 2 の PRS が分担している WWW サーバのうち、他の PRS へ移動した際、全体の合計コスト (式 (3.1)) の増加が最も少ない WWW サーバを選択して移動。
4. 初期状態で、収集コストと再配布コストの和が最小だった PRS のコストが最大になるまで 2,3 を繰り返す。
5. $\boldsymbol{V}$j ($j = 1, \ldots, m$) を出力。

4 分散型 WWW ロボットを用いた WWW 情報収集実験

3節で述べた分散型 WWW ロボットを実装し実験した結果を以下に示す。

4.1 サイト間距離 $t(S_i, R_j)$ の決定

サイト間距離 $t(S_i, R_j)$ として、ping が利用可能かどうかを調べるため以下の実験を行った。

4.1.1 ping 最小応答時間と HTTP 転送時間

ping の応答時間の最小値と HTTP 転送時間との間の相関関係を調べる実験を行った。京大（ kyoto-u.ac.jp ）、早大（ waseda.ac.jp ）、東大 (u-tokyo.ac.jp) のホストから、他の 3 つの WWW サーバ（ www.win.or.jp, www.crl.go.jp, www.kagoshima-u.ac.jp) に対し、 1996 年 11 月第 5 週の平日午前 3 時頃に以下の測定を行った [11]。

- 測定項目
  - ping 最小応答時間（ 1032 byte パケット 10 個の最小値 ）から求めた転送速度
  - HTTP により 50 ページを転送した際の平均転送速度

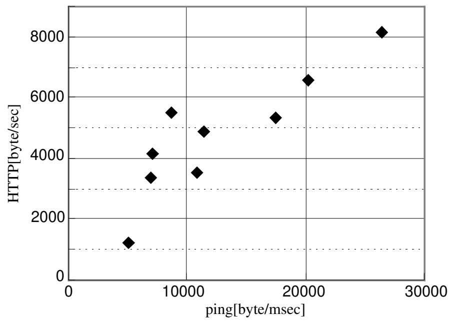

図 3: ping の最小応答時間から求めたデータ転送速度と HTTP 転送速度との相関関係

測定結果を図 3に示す。図 3に示すように ping 最小応答時間から求めたデータ転送速度と HTTP による平均転送速度の間に相関関係（比例）があることがわかる。つまり、 ping 最小応答時間と HTTP 転送時間の間に比例関係が成立することを示す。なお、 ping 応答時間の平均値を用いずに最小値を用いたのは、時間及び曜日による変動が最も小さいからである。

4.1.2 HTTP 転送速度の予測

4.1.1から、 ping 最小応答時間と HTTP 転送速度の間にほぼ比例関係が成立することがわかった。この関係を用いて転送速度を予測する手法について検討する。

1998 年 3 月 17 日午前 2 時 ~ 午前 9 時にかけて、電総研 (etl.go.jp) から、 (A)www.japic.or.jp、(B)www.crl.go.jp、 (C)www.miti.go.jp、(D)www.kagoshima-u.ac.jp の 4 つの WWW サーバに対して以下の測定を行った。

- 測定項目
  - ping 応答時間（ 64/1024/2048 byte パケット 10 個の最小値 ）
  - HTTP により 100 ページを転送した際の転送時間

表 1に パケットサイズを 64 / 1024 / 2048 byte と変化させた時の ping 最小応答時間と、これらの ping 最小応答時間の関係から計算によって求めた 電総研と (A) ~ (D) 間の往復 Latency と往復 Latency を 0 と仮定した際の最大 Throughput を示す。また、表 2に HTTP により転送されたファイル数、転送時間、総ドキュメント量を示す。なお、転送時にエラーとなったファイルは除外した。

表 1: ping 最小応答時間, Latency, Throughput

WWW Server	ping 最小応答時間 (packet size[byte])			往復 Latency	最大 Throughput
	64	1024	2048		
	[ms]	[ms]	[ms]	[ms]	[byte/s]
(A)	20	40	55	17.7	54,949
(B)	18	45	61	15.2	44,716
(C)	29	57	74	26.1	42,729
(D)	75	103	117	72.3	45,773

表 2: HTTP での転送ファイル数、平均サイズ、ドキュメント量、転送時間

WWW Server	ファイル数	平均サイズ [byte]	ドキュメント量 [byte]	転送時間 [ms]
(A)	99	2811	278,259	21,677
(B)	76	3158	240,006	14,455
(C)	98	4526	443,501	31,491
(D)	97	2152	208,778	46,278

次に、 4.1.1の結果より、転送時間が ping 最小応答時間に比例すると仮定して、以下の式 (4.1) で近

似し予測転送時間を算出する。

$$予測転送時間 = \frac{(ping最小応答時間)}{(pingパケットサイズ)} \times (ドキュメント量) \times (補正パラメータa) \quad (4.1)$$

表3に予測転送時間と実転送時間との誤差を示す。表3によれば、pingのパケットサイズとして64byteを使った場合が、最も実転送時間に近い予測転送時間を得ることができることがわかる。一方、表1の往復Latencyと最大Throughputから、実際に転送されたファイル数を考慮して転送時間を計算した場合[1]の実転送時間との誤差は、11.3%であった。

以上より、HTTPによる転送時間は、パケットサイズを変化させた複数のpingによる応答時間から求められる往復Latencyと最大Throughputを用いて計算することにより、実転送時間に近い値を得ることができる。しかし、64byteのpingの最小応答時間のみを用いて予測した場合でも、実転送時間との誤差を14%程度におさえられることがわかる。従って、サイト間距離を短時間で得るために、以下の実験では、サイト間距離としてpingの最小応答時間を用いることにした。

なお、HTTP転送時間がパケットサイズの小さいping最小応答時間に比例する理由は、HTTP転送において、最大Throughput、すなわちネットワークのバンド幅よりもLatencyの影響が大きいためだと考えられる。実際に、$((latency) \times (file数) \times (補正パラメータc))$により予測した場合でも誤差を22%におさえることができた。なお、厳密な検討は今後の課題である。

4.2 予備評価

WWWロボットの分散による収集及び再配布時間の短縮を調べるために、アルゴリズム1及びアルゴリズム2を用いて計算により予備評価を行った。予備評価では、第1ドメインがjpであり、かつ第

[1]以下の式により算出した。

$$予測転送時間 = (latency) \times (file数) \times (補正パラメータa) + \frac{(ドキュメント量)}{(throughput)}$$

表 3: 予測転送時間と実転送時間の誤差

pingのパケットサイズ	WWW Serverの実転送時間と予測転送時間との誤差[a]	補正パラメータaの値[b]
64	14.1%	0.19
1024	19.1%	1.37
2048	20.3%	2.02

[a]WWW Server(A) ~ (D)について各々の実転送時間を基準とし、予測転送時間との誤差を求め、それらを平均した値。

[b]誤差を最小にする値を補正パラメータaとして採用。

1 ~ 3ドメインを同じくするWWWサーバ群を一つの S_i (例:*.waseda.ac.jp)とし、これまでに、「RCAAU – mondou [13]」「千里眼[8]」「ODIN [14]」が収集している6980の異なるドメインを対象に、PRSを東大、早大、電総研、及びシャープの4個所に配置することを想定して評価した。また、収集したデータの再配布先であるSearch Service Server L_i は、PRSと同じサイトにあると仮定した。つまり、4つのPRSで集めたデータを最終的に4つのSearch Service Serverに再配布する。各 S_i のドキュメント量は、実際に測定してみるまではわからないが、これまでの「RCAAU – mondou」「千里眼」「ODIN」のロボットの動作記録を用いることにより推定し総ドキュメント量は約669MBとなった。圧縮率 K については、実際に収集したデータを $tar+gzip$ で圧縮し転送を行った場合の時間を計測し、$K = 0.2$ と設定した。サイト間距離 $t(S_i, R_j)$ は、$ping$ により、50byteのパケット5個を投げた時の往復時間の最小値を用いた。

図4に結果を示す。図中のトータルコストは、PRSが担当する全WWWサーバのデータを収集し再配布を終えるまでの予測時間(hour)を示す。ただし、計算では式(4.1)の補正パラメータとして表3の0.19を用いた。なお、トータルコストは、ネットワークトラフィック増大等により変動することが考えられる。ただし、4.1より、ping最小応答時間とHTTP転送速度との間に比例関係があることが分かっているため、アルゴリズム間の比較をするための値としては、利用可能と判断できる。

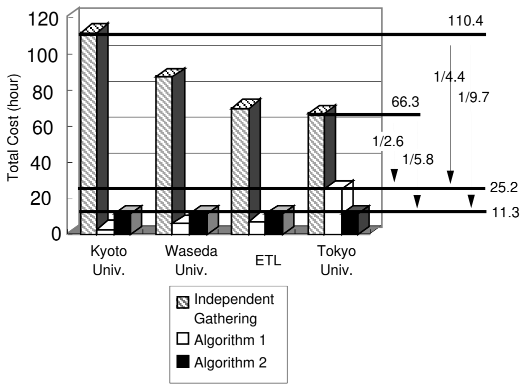

図 4: 予備評価

図4に示すように、一箇所のPRSが全データ収集を担当し、再配分も行うという集中型（Independent Gathering）での収集を行った場合のトータルコストは京大が最も大きく、東大が最も小さくなった。これは、対象となった6980ドメインのWWWサーバに対するサイト間距離が東大が最も近いことを示す。また、アルゴリズム1によって、ネットワークの負荷が最小となるような分割を行うと、集中型に比べ、2.6(東大を基準) ~ 4.4(京大を基準) 倍の高速化が得られることがわかる。さらに、アルゴリズム2により、ネットワーク的に有利な東大に集中している負荷を分散し、トータルコストの均等化を行うと、アルゴリズム1の2.2倍の高速化ができる。

以上より、WWWロボットに分散協調の機能を設け、各ロボットの設置場所に応じて収集の分担を変更する場合、4台のPRS使用時、集中型に比べて5.8 ~ 9.7倍の速度向上が得られることが予備評価からわかった。

4.3 分散型 WWW ロボットによる実験状況

PRS及びPRSMの実装は、主にJava、一部をC言語で記述した。分散型ロボットにおけるインプリメントでは、PRSとWWWサーバ間のサイト距離の値として、*ping* を30分おきに48回行ないその最小値を $t(S_i, R_j)$ として採用した。つまり、24時間中、最もサイト間距離が短くなる時の値を利用している。これは、ネットワークの構成が変わらない限り時間変動を持たず、かつ、その値がHTTPによる転送速度とほぼ比例の関係を持つことが4.1で確認されたためである。

ドキュメント量 (w_i) は事前には分からないため、初期値は1とし、PRSが収集した総ドキュメント量が判明した段階で、定期的 (現状では1ヶ月毎) に再配置する仕様とした。また、第一ドメインが jp であるサイトを対象にデータ収集を行う。

1997年7月より32個の Si (32ドメイン) を初期値として設定し、早大、シャープ、東大、京大、ASCII、電総研、北陸先端大、慶大の8個所でPRSを動作させている。1997年1月14日時点で、13320ドメイン (第1 ~ 第3レベルのドメインが同じドメインを1つのドメインと数える) を発見し、pingによるサイト間距離計測を継続中である。表4に各PRSでサイト間距離計測が終了したドメイン数を示す。いくつかのPRSにおいて、

表 4: pingによるサイト距離計測済ドメイン数

Waseda U.	Sharp	Tokyo U.	Kyoto U.
1964	1456	520	347
ASCII	ETL	JAIST	Keio U.
298	2622	26	331

動作が不安定であったたため、現時点では、全てのドメインについてのサイト間距離の計測が終了していない。しかし、早大、シャープ、東大、電総研の4つのPRSにおいてサイト間距離の計測が終了している78ドメインについて、データを調べたところ、サイト間距離は、これら4つのPRS間で1.1 ~ 10.9倍、平均2.8倍の差が計測された (図5)。なお、PRSと S_i が同一の場合には、最大27倍もの差 (東大ドメインを東大のPRSが担当する場合とシャープのPRSが担当する場合) が計測された。この結果からわかるように、各PRSがネットワーク的に近いドメインを対象として分散収集すれば、データ転送のオーバヘッド削減により、PRS台数以上の高速化が得られると考えられる。

つまり、4.1の「サイト間距離 (ping最小応答値) がHTTP転送速度との間に比例関係が存在する」という結果を用いると、WWWロボットを n 台に分散させた時、最大 $(2.8 \times n)$ 倍の高速化が得られると予想できる。

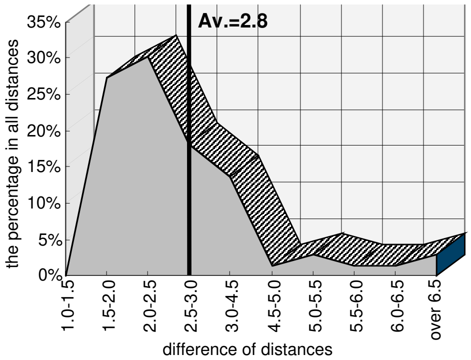

図 5: サイト間距離の分布

5 おわりに

本論文では、著者らが共同して行っている分散型 WWW ロボット実験の動作状況について報告した。当面の目標として日本国内の WWW サーバ上の全データを 24 時間以内に収集可能な分散型ロボットの構築を目指して、実験を続けている。

既存の情報検索サービスは、検索対象のドキュメントおよびサービスの両方の面からスケールしなくなってきており、情報検索サービスの分散化技術の開発が急務なことは明らかである。情報検索システムの分散化の方式には、分散型 WWW ロボットを使った分散収集と既存の集中検索を組み合わせる方式と、検索能力自体を分散サーバ内に収容する完全分散システムが考えられる。

後者の方式としては、これまでに Ingrid [15] が提案されている。 Ingrid は、情報カテゴリの関連性に基づく情報間のトポロジを事前に構築しておき、そのトポロジを前提とした分散探索を行うシステムであり、データを分散させたまま検索する方式をとる。しかし、全 WWW サーバにこうした機能を持たせることは困難である。したがって、現存の WWW サーバ全体を対象とした検索を提供するためには、分散型 WWW ロボットによって収集した一定の WWW データを単位として、それらを分散させたまま検索を実現するシステムを考えていく必要がある。

今後は、サイト間距離として ping を用いることの正当性を厳密に検証すると共に、サイト間距離として、前回の WWW ロボット巡回時の HTTP 転送時間の利用も考えていく。また、当面の目標である 24 時間以内のデータ収集を達成を目指すと共に、次のステップとして、検索能力を PRS 内に収容した分散検索についても検討を行っていく予定である。

謝辞

本研究を遂行するにあたり、御討論いただいた電子技術総合研究所情報アーキテクチャ部並列アーキテクチャラボ（ラボリーダ: 山口喜教）の皆さんに感謝いたします。

参考文献

[1] AltaVista, http://www.altavista.digital.com/

[2] HotBot, http://www.hotbot.com/

[3] The Web Robots Pages, http://info.webcrawler.com / mak / projects / robots / robots.html

[4] Database of Web Robots, http://info.webcrawler.com / mak / projects/ robots / active/ html / index.html

[5] Netcraft-Web-Server-Survey, http://www.netcraft.co.uk/survey/

[6] C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber and Michael F.Schwartz., "*The Harvest Information Discovery and Access System*", Computer Networks and ISDN Systems, Vol.28, pp.119-125 (1995) (http://harvest.transarc.com/)

[7] 山名, "*WWW* 情報検索の現状", コンピュータソフトウェア, Vol.14, No.5, pp.67–74, 1997

[8] 千里眼, http://senrigan.ascii.co.jp/

[9] Internet Domain Survey, http://www.nw.com/zone/WWW/top.html

[10] Infoseek, http://www.infoseek.com/

[11] 田村, "分散 *WWW* ロボットに関する研究", 早稲田大学修士論文 (1997.3)

[12] S.Kamei,H.Kawano,T.Hasegawa, "*Effectiveness of Cooperative Resource Collecting Robots for Web Search Engines*", Proc. of Pacrim'97, Vol.1, pp.410–413, 1997

[13] RCAAU – mondou –, http://www.kuamp.kyoto-u.ac.jp/labs/infocom/mondou/

[14] ODIN, http://kichijiro.c.u-tokyo.ac.jp/odin/

[15] P.Francis, 神林, 佐藤, 清水: "次世代情報検索インフラストラクチャ *Ingrid*", NTT R & D, Vol.45, No.2, pp.159-165 (1996)