

スケーラビリティのある WWW 並列全文検索システム構築法の提案と評価

速水賢史[†] 竹野浩^{††} 永瀬智哉[†]
藤本典幸[†] 萩原兼一[†]

World Wide Web 上で情報を効率的に探し出すために使われる WWW 全文検索システムは、一般に複数の計算機を用いて構成されている。WWW 並列全文検索システムには、増大し続ける World Wide Web 上の文書数に対応するため、文書情報保有能力と検索要求処理能力におけるスケーラビリティが必要となる。並列全文検索システムでは、インデックスの計算機間での配置法が検索要求処理能力に影響を及ぼす。このインデックスの計算機間での配置法について、文書分散手法と単語分散手法の2つをスケーラビリティの点で評価し、効率的なシステムの構築法を検討する。

Proposal and Evaluation of Scalable WWW Parallel Search Engine Designing Methods

SATOSHI HAYAMI,[†] HIROSHI TAKENO,^{††} TOMOYA NAGASE,[†]
NORIYUKI FUJIMOTO[†] and KEN-ICHI HAGIHARA[†]

WWW search engines, the system for finding preferable information on World Wide Web, are generally implemented on NOW(Network of Workstations). WWW search engines are required to be scalable at the number of managed documents and the throughput of processing queries. Efficiency of processing queries depends on how to partition index among workstations. In this paper, we evaluate two partitioning methods, document partitioning and term partitioning, and investigate designing a scalable WWW parallel search engine.

1. はじめに

World Wide Web (以下 WWW と呼ぶ) において公開される文書 (以下 WWW 文書と呼ぶ) を網羅的に収集し、収集した文書に対して全文検索を行う WWW 全文検索システムが存在する。Altavista¹⁾, goo²⁾ などに代表されるこのシステムに対して、ユーザは検索要求を送信することで、その検索要求に該当する WWW 文書の URL を得ることができる。この WWW 全文検索システムは、近年の急速な WWW の普及による WWW 文書の増大³⁾⁴⁾ に対応するため一般的に複数の計算機を用いた WWW 並列全文検索システムとなっている。WWW 並列全文検索システムを構築する場合、今後も WWW 文書が増大し続けることを考慮して、システムに計算機を追加することによって情報収集能力を効率良く向上させ、WWW 文書の増加に対応出来るような形で構築するのが望ましい。この場合、増大した検索対象 WWW 文書に対応するため、同時に計算機の追加による検索要求処理能力を維持できる必要がある。本稿では、情報収集能力に関してスケーラビリティのある WWW

並列検索システムにおいて、検索要求処理能力を高めることを目標とする。そこで、検索に用いるデータベースであるインデックスの分散手法が検索応答時間、検索応答のスループットに影響を及ぼすことに着目する。本稿では、文書分散手法、単語分散手法の2つのインデックス分散手法について性能の定式化による検索応答能力の比較を行い、効率的な検索システム構築法を検討する。

2. 並列型全文検索システム

2.1 WWW 全文検索

WWW 文書の集合 D に対して検索式 q が指定されたとき、 D の文書のうち q によって表される検索要求条件に適合する文書集合 $D(q)$ を求めることを全文検索という。ここで、検索式 q を以下の式で定義する。

- 単語 w は検索式である
 - q_1 および q_2 が検索式ならば
(q_1 and q_2), (q_1 or q_2), (q_1 not q_2) は検索式である
- また、 q 中に表れる単語の集合を $g(q)$ とおく。

$D(q)$ は以下の式で再帰的に定義される。

$$\begin{aligned} D(w) & \text{ は } w \text{ を含む文書の集合} \\ D(q_1 \text{ and } q_2) & = D(q_1) \cap D(q_2) \\ D(q_1 \text{ not } q_2) & = D(q_1) - D(q_2) \\ D(q_1 \text{ or } q_2) & = D(q_1) \cup D(q_2) \end{aligned}$$

[†] 大阪大学 大学院基礎工学研究科
Graduate School of Engineering Science, Osaka University
^{††} NTT サイバソリューション研究所
NTT CyberSolution Laboratories

WWW 全文検索システムでは文書にスコアと呼ばれる実数値をつけるスコアリングを行う⁵⁾。ここで使われるスコアとは、ある文書 d が検索式 q によって表される検索要求条件にどれほど適合した情報を持っているかということを表す値である。もっとも一般的なスコアリングに **TF/IDF** が存在する。**TF**(Term Frequency) 値とは文書 d での語 w の出現頻度から算出する値であり、**IDF**(Inverse Document Frequency) 値とは文書集合 D での語 w の出現頻度から算出する値である⁵⁾。以降では文書 $d \in D$ の w による TF 値を $T_f(w, d)$ 、文書集合 D での語 w の IDF 値を $I_{df}(D, w)$ とする。複数単語からなる検索式のスコアを、以下のように再帰的に定義する。○は 2 項演算子である。

$$S_c(D, q, d) = T_f(w, d) \circ I_{df}(D, w) \quad (q = w) \quad (1)$$

$$S_c(D, q, d) = 0 \quad (d \notin D(q)) \quad (2)$$

$$S_c(D, (q_1 \text{ and } q_2), d) = S_c(D, q_1, d) \times S_c(D, q_2, d) \quad (3)$$

$$S_c(D, (q_1 \text{ or } q_2), d) = S_c(D, q_1, d) + S_c(D, q_2, d) \quad (4)$$

$$S_c(D, (q_1 \text{ not } q_2), d) = \begin{cases} S_c(D, q_1, d) & (S_c(D, q_2, d) = 0) \\ 0 & (S_c(D, q_2, d) \neq 0) \end{cases} \quad (5)$$

検索応答 $A(D, q)$ は、 $\{d | d \in D(q)\}$ を $S_c(D, q, d)$ により降順に整理させたものとし、この時の d の順位を $rank(D, q, d)$ とする。

2.2 インデックス

WWW 全文検索システムでは、保有している文書集合 D_s を必ずしもそのまま記憶しておく必要はない。任意の WWW 文書 d は d の所有者の WWW サーバに記憶されており、 d の所在である URL が分かれば d を取得することができるからである。

実際の WWW 全文検索システムでは、実用的な検索速度を得るために、 d と URL を構造化して記憶している。これをインデックスと呼ぶ⁵⁾⁶⁾。

2.2.1 インデックスの構造

検索応答に用いる文書情報を $info(d)$ 、 D に含まれる文書に出現するユニークな単語全ての集合を $W(D)$ で表す。一般に $info(d)$ は必須となる URL の他に d の収集日時やヘッドライン等が含まれる。検索応答 $A(D, q)$ は $D(q)$ ではなく $\{info(d) | d \in D(q)\}$ がスコアで整理したものと再定義する。語 w に関する情報の集合 $I(D(w), w)$ を以下のように定義する。

$$I(D(w), w) = \{ \langle w, T_f(w, d), info(d) \rangle | d \in D(w) \}$$

全文検索システムが記憶しておくべき情報は、以下の $I(D, W(D))$ である。

$$I(D, W(D)) = \{ I(D(w), w) | w \in W(D) \}$$

インデックスは、主に図 1 の構造をした 3 つのファイルで構成される。 $add(w)$ は w の出現情報ファイルでの

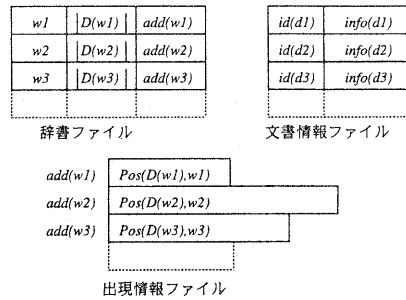


図 1 インデックスの構造

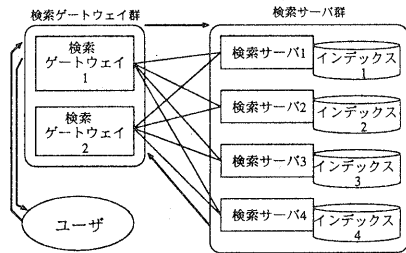


図 2 検索システムの構成

格納位置を指す。また、 $D(w)$ を以下の式で定義される $Pos(D(w), w)$ により記憶する。

$$Pos(D(w), w) = \{ \langle id(d), T_f(w, d) \rangle | d \in D(w) \}$$

$id(d)$ は、全ての $d(d \in D)$ 間でユニークに割り振られた識別番号であり、文書情報ファイルを用いて $id(d)$ より $info(d)$ を特定する。

2.3 WWW 並列全文検索システムの全体像

WWW 並列全文検索システム (以降単に検索システムと呼ぶ) では大きく分けて、情報収集、インデックス作成、全文検索の 3 つの処理をおこなっている。

2.3.1 情報収集、インデックス作成

各計算機は、共通 URL データベースより URL 集合を得てそれにより WWW 文書を収集する。共通 URL データベースはシステム内の計算機間で重複した文書を収集することを避けるため使われる。この処理を繰り返すことで大量の WWW 文書を収集し、2.2 節で述べたインデックスに整形する。

2.3.2 検索要求応答処理

検索システムでの検索応答処理は、インデックス検索処理をおこなう検索サーバと、ユーザとの検索要求や検索結果の受け渡しをする検索ゲートウェイから構成される。ある 1 台の計算機が検索ゲートウェイと検索サーバの役割を兼ねることもある。本稿で提案する検索システムでは情報収集およびインデックス作成を行った全計算機を検索サーバとする。検索ゲートウェイは検索サーバとは別の 1 台以上の計算機を用意する。(図 2)

以降では情報収集処理及びインデックス作成をおこな

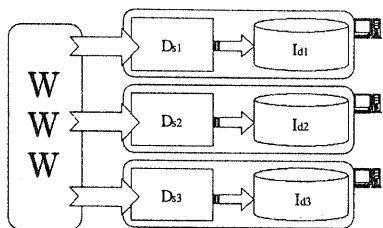


図3 文書分散によるインデックス分割

う検索サーバの台数を n 台とし、それぞれの計算機を P_i ($1 \leq i \leq n$) で表す。また、検索ゲートウェイの集合を検索ゲートウェイ群、検索サーバの集合を検索サーバ群と呼ぶ。ユーザからの検索要求は検索ゲートウェイ群の1台が受け取る。検索ゲートウェイは検索要求を解析し、検索サーバ群に検索命令を送信する。検索サーバ群はインデックスより検索要求に含まれる単語の文書情報を抽出し、検索ゲートウェイに送信する。検索ゲートウェイでは検索サーバから得た単語情報を統合し検索応答を作成し、ユーザに送信する。

3. インデックスの分割方法と検索手順

文書分散手法⁶⁾⁷⁾と単語分散手法⁶⁾⁸⁾の2つのインデックス分散手法と、インデックスの検索処理について述べる。

3.1 インデックス分割手法

文書分散手法

情報収集処理により検索システムの得た文書集合 D_s を、計算機間で $D_{s1}, D_{s2}, \dots, D_{sn}$ の互いに素な n 個の部分集合に分割する。

任意の計算機 P_i は D_{si} より

$$Id_i = I(D_{si}, W(D_{si}))$$

で定義されるインデックスを作成し(図3)、検索要求に対して検索サーバとしてこのインデックスを検索する。

単語分散処理

単語分散処理では、 $W(D_s)$ に含まれる語 w のハッシュ値を用い、 $W(D_s)$ を $W_1(D_s), W_2(D_s), \dots, W_n(D_s)$ の互いに素な n 個の部分集合に分割する。ここで $W_i(D_s)$ は以下のように定義する。

$$W_i(D_s) = \{w | w \in W(D_s), i = h(w)\}$$

ハッシュ値 $h(w)$ ($1 \leq h(w) \leq n$) を用いて、任意の語 w を $W_{h(w)}(D_s)$ に割り振る。

任意の計算機 P_i では、 $W_i(D_s)$ より

$$Iw_i = I(D_s, W_i(D_s))$$

で定義されるインデックスを作成し、計算機 P_i に配置する(図4)。 P_i は検索要求に対して検索サーバとしてこのインデックスを検索する。

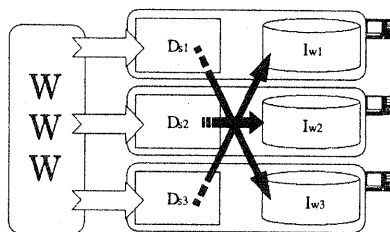


図4 単語分散によるインデックス分割

3.2 検索処理

3.2.1 検索処理の手順

検索処理とは q に対して文書集合 D より構築されたインデックスを用いて $A(D, q)$ を得ることである。

以降、 q の検索をするために参照する $Pos(d, q)$ の総数を検索件数と呼ぶ、 q の検索件数は、以下の $R(D, q)$ で表される。

$$R(D, q) = \sum_{w \in g(q)} |D(w)| \propto |D| \quad (6)$$

また、検索により得られた q に適合する文書の総数をヒット件数と呼ぶ。 q のヒット件数は、以下の $H(q)$ で表される。

$$H(D, q) = |\{d | S_c(D, q, d) \neq 0\}| \propto |D| \quad (7)$$

検索の手順を以下に示す。

- (1) $w \in g(q)$ に (a)~(c) の処理を行う。
 - (a) 辞書ファイルを用いて $add(w)$ 、 $|D(w)|$ を得る。 $|D(w)|$ より $I_{df}(D, w)$ を計算する。
 - (b) $add(w)$ より、出現情報ファイルを用いて $Pos(D(w), w)$ を得る。
 - (c) $\forall d \in D(w)$ に関して $T_f(d, w)$ と $I_{df}(D, w)$ より $S_c(D, w, d)$ を求める。
- (2) 式(1)~(5)に従い $S_c(D, q, d)$ を得る。 $S_c(D, q, d)$ が0であるものを除外して $Pos(D(q), q)$ とする。
- (3) $Pos(D(q), q)$ を $S_c(D, q, d)$ でソートする。
- (4) $Pos(D(q), q)$ について、文書情報ファイルを用いて $info(d)$ を得る。

検索処理(1-a)は2分探索であるため $\log(D(w))$ に比例した負荷であり実際はほぼ定数時間となる⁵⁾。検索処理(1-b)はメモリ上に $Pos(D(w), w)$ を載せる処理であり、 $|D(w)|$ に比例した負荷、検索処理(1-c)は $|D(w)|$ 回演算を繰り返すので $|D(w)|$ に比例した負荷となる⁵⁾。式6の定義より、検索要求全体での検索処理(1-b)、検索処理(1-c)の負荷の合計は $R(D, q)$ に比例したものである⁵⁾。

検索処理(2)は、 $R(D, q)$ に比例した負荷となる⁵⁾。

検索処理(3)はソートであり、要素数は $H(D, q)$ であるため負荷は $H(D, q) \log(H(D, q))$ に比例する。

検索処理(4)は、 $id(d)$ から $info(d)$ を引く処理を $H(D, q)$ 件行うため、負荷は $H(D, q)$ に比例する。

表 1 検索要求応答の各処理

WDH		WDD	
処理の主体	処理の内容	処理の主体	処理の内容
GW		検索要求解析, 検索命令作成	
GW, サーバ i	検索命令通信	GW, 全サーバ	検索命令通信 $\{I_{df}(D, w) w \in q\}$ の計算
サーバ i	検索処理 (1)	全サーバ	検索処理 (1) 検索処理 (2) 検索処理 (3)
GW, サーバ i	$\{Pos(d, w) d \in D(q)\}$ の通信	GW, 全サーバ	$\{Pos(d, q) 1 \leq rank(D_{si}, q, d) \leq 1000, 1 \leq i \leq n\}$ の通信
GW	検索処理 (2) 検索処理 (3) 検索処理 (4)	GW	$\{A(D_{si}, q) 1 \leq i \leq n\}$ のマージソート 検索処理 (4)
GW		検索応答の送信完了	

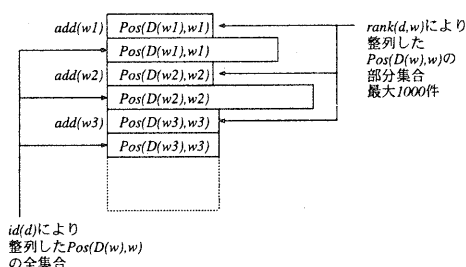


図 5 改良を加えた出現情報ファイル

3.2.2 ユーザの特性を考慮したインデックス検索改良

WWW 検索ではユーザの傾向より、検索応答はスコアの上位 1000 件で十分であり、実際には高々上位 100 件しか必要ではないということが分かっている⁹⁾。

これを考慮して、3.2.1 節の (3) では全体ではなく $\{d | 1 \leq rank(D, q, d) \leq 1000\}$ のみをソートするよう変更する。(4) では $\{d | 1 \leq rank(D, q, d) \leq 100\}$ について $info(d)$ を得て応答し、 $\{d | 100 \leq rank(D, q, d) \leq 1000\}$ はユーザからの追加要求を受けてから $info(d)$ を得るよう変更する。また、WWW 検索では q 全体に占める 1 単語からなる q の確率が非常に高い⁶⁾。1 単語での検索ではインデックス作成時点でスコアによるソートを先に行っておくことで、検索段階でのソートを無くすことが出来る。インデックス作成時点でスコアによるソートを行った出現情報ファイルを図 5 に示す。それぞれ変更後の (1-b)', (3)', (4)' は、以下ようになる。

(1-b)' $q = w$ ならば $rank(D, d, q)$ により整列した

$Pos(D(q), q)$ 最大上位 1000 件を得る。

$q \neq w$ ならば $id(d)$ により整列した $Pos(D(q), q)$ を得る。

(3)' $q \neq w$ ならば $S_c(D, q, d)$ によるソートを行う。

(4)' $\{d | 1 \leq rank(D, d, q) \leq 100\}$ について、文書情報ファイルを用いて $info(d)$ を得る。

このとき、 $\{d | 1 \leq rank(D, d, q) \leq 100\}$ を特定することが出来るのは検索ゲートウェイであるので、全体の文書情報ファイルは検索ゲートウェイに配置する。

4. 各分散手法での検索の相違

文書分散でのインデックス分割、単語分散でのインデックス分割のそれぞれの場合について、検索における各処理は表 1 のようになる。表中の GW は検索ゲートウェイ、全サーバは $P_i (1 \leq i \leq n)$ 、サーバ i は検索式 q に対して $\{P_i | i = h(w), w \in q\}$ で表される検索サーバである。

4.1 文書分散での検索

4.1.1 検索手順

文書分散した検索システムでの検索を以下のように行う (図 6)。文書分散では、全検索サーバが $Pos(D_s(w), w)$ を分割して保持しており、検索ゲートウェイは全検索サーバに検索命令を送る。任意の P_i で検索処理 (1-c) を行うためには各単語の IDF 値 $\{I_{df}(D_s, w) | w \in g(q)\}$ が必要となる、IDF 値を求めるため、全 P_i は検索処理 (1-a) により得た $\{D_{si}(w) | w \in g(q)\}$ を検索ゲートウェイに送信する。検索ゲートウェイは IDF 値を計算し、検索サーバ群へ送信する。IDF 値を受信後、各 P_i は検索処理 (1-b)', (1-c), (2), (3)' をおこない得られた $\{Pos(d, q) | 1 \leq rank(D_{si}, d, q) \leq 1000\}$ を検索ゲートウェイに送信する。

検索ゲートウェイで受信するのはスコアで整列した n 個の $Pos(d, q)$ の集合である、検索ゲートウェイではこれらの集合に対してスコアによるマージソートをおこない、 $\{Pos(d, q) | 1 \leq rank(D_s, d, q) \leq 1000\}$ を得る。これについて検索処理 (4)' を行う。

4.1.2 計算時間の推定

検索ゲートウェイと検索サーバ群の通信路の性能を、スタートアップレイテンシ L 、伝送速度 V_i とする。ここでスタートアップレイテンシとは通信の前処理にかかる時間である。

検索要求の解析にかかる時間は検索要求中に含む単語数、演算子の種類により異なるが、検索処理時間に比べて無視できる大きさである。検索命令の送信は、おなじ検索命令を全検索サーバに送る処理であり、検索命令作成の時間 T_{g1} は 1 つの検索命令を作成する時間を定数

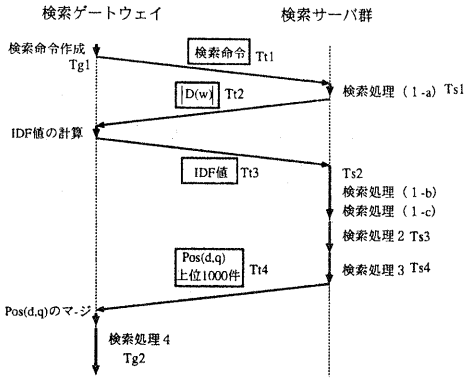


図6 文書分散での検索手順

時間 M として $T_{g1} = M$ である。検索命令のメッセージサイズを m とおくと一台あたりの通信時間は $L + m/V_i$ となる、 m は V_i と比較して非常に小さい値であるので一台あたりの通信時間は L と表せる。検索ゲートウェイから検索サーバ群への通信時間 $T_{t1} = nL$ となる。同様に、検索ゲートウェイからの $|D(w)|$ の通信時間 $T_{t2} = nL$ 、検索ゲートウェイからの IDF 値の通信時間 $T_{t3} = nL$ である。IDF 値の計算も単純な演算であり、かかる時間は無視できる。

各 P_i について、3.2.1 節より検索処理 (1-a) は 2 分探索であり処理時間 T_{s1} はほぼ定数 c となり、検索処理 (1-b), (1-c) にかかる時間 T_{s2} 、検索処理 (2) にかかる時間 T_{s3} はそれぞれ $R(D_{si}, q)$ に比例する。 $R(D_{si}, q)$ は表 2 のようにほぼ均等に分散する⁸⁾ ため、 $R(D_{si}, q) = R(D_s, q)/n$ であり、比例係数を a_1, a_2 とおくと、 $T_{s2} = a_1 R(D_s, q)/n$ 、 $T_{s3} = a_2 R(D_s, q)/n$ である。検索処理 (3)' に要する時間 T_{s4} は 3.2.1 節で述べたとおり $H(D_{si}, q)$ に比例する。 $R(D_i, q)$ と同様に $H(D_{si}, q) = H(D_s, q)/n$ であるため、比例係数を a_3 とおき、 $a_3 H(D_s, q)/n$ で表す。

P_i は $Pos(D, q)$ を最大 1000 件しか送信しないので (3.2.2 節)、 $Pos(d, q)$ ひとつのサイズを p とおくと、 $Pos(D, q)$ の通信時間は

$L + \{p \times \min(H(D_s, q)/n, 1000)\}/V_i$ である。全検索サーバでの通信時間 T_{t4} は

$$T_{t4} = nL + \min(H(D_s, q), 1000n)p/V_i \quad (8)$$

検索処理 (4)' を行うのは最大 100 件であり (3.2.2 節)、負荷は件数に比例する (3.2.1 節) ので、1 件あたりの負荷を a_4 とすると、検索処理 (4)' 全体によりかかる時間 T_{g2} は、

$$T_{g2} = \min(100, H(D_s, q))a_4 \quad (9)$$

4.1.3 検索応答速度、検索負荷の定式化

検索応答時間および検索負荷を定式化する。

検索応答時間 T_d は次のようになる

表 2 検索件数の変化

	$R(D_i, national)$	$R(D_i, review)$	$H(D_i, q)$
P_1	7926	5879	1356
P_2	8142	6254	1322
P_3	7821	6121	1292
P_4	8309	5768	1429
P_5	8398	5889	1432
P_6	9039	6326	1525
P_7	8124	6143	1415
P_8	8152	5988	1418

$$T_d = T_{g1} + T_{g2} + \sum_{j=1}^4 T_{tj} + \sum_{k=1}^4 T_{sk}$$

4.1.2 節で推定した値と次の A, B を用いて表す。

$$A = \min(H(D_s, q), 1000n)p/V_i \quad (10)$$

$$B = \min(100, H(D_s, q))a_4 \quad (11)$$

$$T_d = \frac{(a_1 + a_2)R(D_s, q) + a_3H(D_s, q)}{n} + 4nL + M + c + A + B \quad (12)$$

1 検索要求あたり検索ゲートウェイの計算時間 L_{dg} は

$$L_{dg} = T_{g1} + T_{g2} + \sum_{j=1}^4 T_{tj} = 4nL + M + A + B \quad (13)$$

検索システム全体では、検索サーバの処理は n 台で並行に行うため 1 検索要求あたり検索サーバ群の計算時間 L_{ds} は、

$$L_{ds} = n \sum_{i=1}^4 T_{si} + \sum_{j=1}^4 T_{tj} = (c+4L)n + (a_1+a_2)R(D_s, q) + a_3H(D_s, q) + A \quad (14)$$

4.2 単語分散での検索

4.2.1 検索手順

単語分散した検索システムでの検索を以下のように行う (図 7)。単語分散では、 $P_i (i = h(w))$ が $Pos(D(w), w)$ を保持している。検索ゲートウェイは $\{P_i | i = h(w), w \in g(q)\}$ に検索命令を送る。

$\{I_{df}(D_s, w) | w \in g(q)\}$ は各 $P_i (i = h(w))$ で計算することができる。 P_i は検索処理 (1-a), (1-b)', (1-c) を行い得られた $Pos(D(w), w)$ を検索ゲートウェイに送信し、検索ゲートウェイで処理 (2), (3)', (4)' を行う。

4.2.2 計算時間の推定

q に含まれる w の数を $|g(q)|$ で表す。

検索要求の解析は文書分散同様無視できる。検索命令の通信 T_{t5} は、異なる検索命令を $|g(q)|$ 台の検索サーバに送る処理であり、検索命令を作成する時間 $T_{g3} = |g(q)|M$ 、 $T_{t5} = |g(q)|L$ である。 P_i での検索処理 (1) にかかる時間 T_{s5} は、単語分散では処理の対象となる要素数が

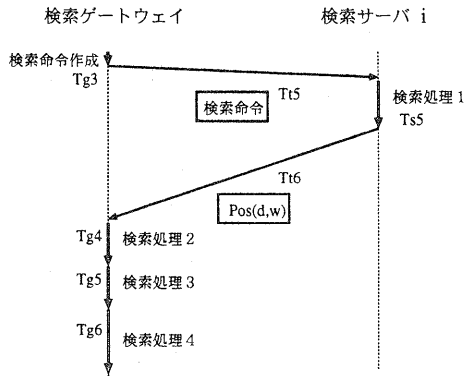


図 7 単語分散での検索手順

$R(D_s, w)$ であるので計算時間は $c + a_1 R(D_s, w)$ である。検索サーバ群全体の T_{s5} の計算時間は、

$$T_{s5} = \sum_{w \in g(q)} (c + a_1 R(D_s, w)) = |g(q)|c + a_1 R(D_s, q)$$

検索サーバ群からの通信時間 T_{t6} は

$$\begin{aligned} T_{t6} &= \sum_{w \in g(q)} (L + pR(D_s, w)) \\ &= |g(q)|L + pR(D_s, q)/V_i \end{aligned} \quad (15)$$

である。検索処理 (2), (3)' の計算時間 T_{g4} , T_{g5} も文書分散時と同様であるが、処理の対象となる要素数が $R(D_s, q)$, $H(D_s, q)$ であるので、それぞれ $a_2 R(D_s, q)$, $a_3 H(D_s, q)$ となる。検索処理 (4)' の計算時間 T_{g6} は文書分散と同じである。

4.2.3 検索応答速度, 検索負荷の定式化

検索応答時間 T_w は

$$T_w = \sum_{i=3}^6 T_{gi} + T_{s5} + T_{t5} + T_{t6}$$

で表され、4.2.2 節で推定した値と式 10 を用いて

$$\begin{aligned} T_w &= (a_1 + a_2 + p/V_i)R(D_s, q) + a_3 H(D_s, q) \\ &\quad + |g(q)|(2L + M + c) + a_3 H(D_s, q) + B \end{aligned} \quad (16)$$

1 検索要求あたり検索ゲートウェイの計算時間 L_{wg} は

$$\begin{aligned} L_{wg} &= \sum_{i=3}^6 T_{gi} + T_{t5} + T_{t6} \\ &= |g(q)|(2L + M) + (a_2 + p/V_i)R(D_s, q) \\ &\quad + a_3 H(D_s, q) + B \end{aligned} \quad (17)$$

1 検索要求あたり検索サーバ群の計算時間 L_{ws} は、

$$\begin{aligned} L_{ws} &= T_{s5} + T_{t5} + T_{t6} \\ &= |g(q)|(2L + c) + (a_1 + p/V_i)R(D_s, q) \end{aligned} \quad (18)$$

のようになる。

5. 評価実験

5.1 実験環境

情報収集用計算機及び検索サーバ群として Pentiu-

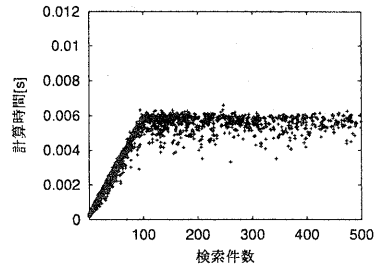


図 8 検索処理 (4) の計算時間 T_{g2} , T_{g6}

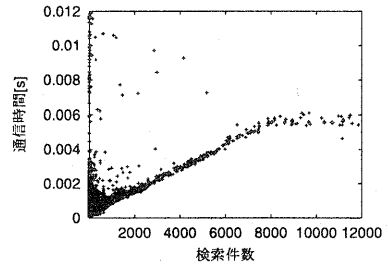


図 9 文書分散での通信時間 T_{t4}

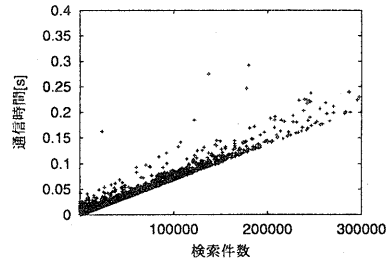


図 10 単語分散での通信時間 T_{t6}

mII 350MHz の CPU, メモリ 512MB, ハードディスク UW - SCSI9.1GB から構成される計算機 8 台を 100Mbps のイーサネットスイッチングハブで接続した NOW(Network Of Workstations) を利用した。

また、検索ゲートウェイとして PentiumII 450MHz の CPU, メモリ 128MB, ハードディスク W - SCSI4.1GB で構成される検索ゲートウェイ 1 台を利用した。

検索応答速度, 検索負荷の測定には, WWW 検索システム Metacrawler¹⁰⁾ に実際にユーザより送られた検索要求 1 万件を用いた。

5.2 負荷推定の検証実験

4.1.2 節, 4.2.2 節で推定した検索負荷の定式化について実験による検証をおこなう。

図 8 は全ての検索要求について検索処理 (4) の計算時間 T_{g2} , T_{g6} と $H(D_s, q)$ の関係を表したものである。計算時間は式 9 で表されるように、100 件までは件数に比

例し、それ以上では一定の値をとる。

図 9 は、全ての検索要求について文書分散における検索ゲートウェイでの検索サーバからの結果受信にかかる時間 T_{d4} と、ヒット件数との関係を表した物である。実験環境での検索サーバは 8 台であり $n = 8$ 、通信時間は検索件数が 1000n 以下では検索件数に比例し、それ以上では一定の値であり、式 8 によく一致している。

図 10 は、全ての検索要求について単語分散における検索ゲートウェイでの検索サーバからの結果受信にかかった時間 T_{d6} と、検索件数との関係を表したものである。検索件数に比例する通信時間となっており、式 15 を満たしていると言える。

5.3 スケーラビリティの考察

検索サーバ 1 台あたりがインデックスに保持する WWW 文書の件数を G とすると、検索システム全体の文書数 $|D_s| = nG$ である。式 6,7 より、 $R(D_s, q), H(D_s, q)$ は nG に比例する、 $R(D_s, q), H(D_s, q)$ を nG を用いて、

$$R(D_s, q) = f_r(q)nG, H(D_s, q) = f_h(q)nG \quad (19)$$

と表す。 $f_r(q), f_h(q)$ は WWW 文書中の q の出現頻度であり、 q 固有の定数である。

式 19 を用いて式 12 を変形する。

$$T_d = Dn + E + B$$

$$D = 4L + \min(f_h G, 1000)p/V_t \quad (20)$$

$$E = \{(a_1 + a_2)f_r(q) + a_3 f_h(q)\} G + c + M$$

同様に式 16 を変形する。

$$T_w = D'n + E' + B$$

$$D' = \{(a_1 + a_2 + p/V_t)f_r(q) + a_3 f_h(q)\} G \quad (21)$$

$$E' = |g(q)|(2L + M + c)$$

文書分散では、全ての検索要求について、検索命令の通信、 I_{DF} 値の計算、各検索サーバからの結果の受信において検索ゲートウェイは検索サーバと通信をしなければならず(式 20)、検索応答時間について台数の増加により検索応答速度は低下する。

単語分散では、検索処理 (1), (2), (3) の負荷および結果の通信量は n に比例し(式 21)、 n に関係なくこれらの負荷は検索サーバ $|g(q)|$ 台と検索ゲートウェイにかかるため、台数の増加により検索応答時間は低下する。

文書分散と単語分散を比較した場合、文書分散では検索サーバ全体に大部分の負荷が分散されているため、台数の増加による検索応答速度の低下は緩やかになっており(図 11, 12)、検索応答速度の面でスケラビリティが高い。単語分散では全ての負荷が検索サーバ $|g(q)|$ 台と検索ゲートウェイにかかるため、台数の増加にほぼ比例して検索応答時間は増大する。(図 13, 14)

式 19 を用いて式 13 を変形する。

$$L_{dg} = Fn + B$$

$$F = 4L + M + \min(f_h G, 1000)p/V_t$$

同様に式 17 を変形する。

$$L_{wg} = F'n + B + |g(q)|(2L + M)$$

$$F' = (a_2 + p/V_t)f_r(q) + a_3 f_h(q)$$

表 3 はスループットを検索要求あたりの計算時間、通信量の合計の逆数として求めたものである。検索サーバの台数が増えることにより、文書分散での検索ゲートウェイの負荷 L_{dg} は検索応答時間と同様に、通信数の増加にともない増加しスループットは低下するが、全体に占める通信の負荷の割合が小さく、低下が抑えられておりスケラビリティがあるといえる。単語分散での検索ゲートウェイの負荷 L_{wg} は、通信量、計算量ともに検索サーバの台数に比例する。これらが検索ゲートウェイの負荷の大部分を占めているため、検索スループットは検索サーバ台数に反比例して低下する。このため、検索サーバ数に比例した数の検索ゲートウェイが必要となり、単語分散は文書分散と比較してスケラビリティ面で劣っている。

式 19 を用いて式 14 を変形する。

$$L_{ds} = Hn$$

$$H = 4L + \min(f_h G, 1000)p/V_t + c + \{(a_1 + a_2)f_r(q) + a_3 f_h(q)\} G$$

同様に式 18 を変形する。

$$L_{ws} = (a_1 + p/V_t)f_r(q)n + |g(q)|(2L + c)$$

文書分散の検索サーバ群の負荷 L_{ds} 、単語分散の検索サーバ群の負荷 L_{ws} は、ともに $O(n)$ の負荷であり、 n 台の検索サーバに負荷が分散されるため、検索サーバ群のスループットは n に依存せずほぼ一定となり、どちらの手法にもスケラビリティがあると言える。(表 3)

5.4 両分散手法の併用

検索応答時間のスケラビリティの面で考えると、検索サーバ台数に依存が少ない文書分散が勝ると考えられる。検索スループットは、文書分散では全検索サーバとの通信が必要であるため検索ゲートウェイのスループットは落ち、検索サーバ群のスループットも一定となり、単語分散では検索ゲートウェイと特定の検索サーバに負荷を集中させることができるが、その負荷は検索サーバ台数に比例して増大するため検索ゲートウェイのスループットは低下し検索サーバ群のスループットも一定となる。

総合的に判断すると、計算機環境により差は出るが、文書分散のほうが単語分散よりスケラビリティがあると言える。しかし、検索要求が 1 単語のみから構成される場合、単語分散では検索負荷が検索サーバ台数に関わらず一定以下(最大 1000 件の通信、計算)であるため上記の問題点はおこらない。

そのため、1 単語検索要求は単語分散されたインデックス検索、複数単語検索処理は文書分散されたインデックス検索と 2 つの手法を併用することによって、負荷を軽減しスループットを高める事が出来る。表 3 の併用手法は、文書分散されたインデックスで複数単語検索を行い単語分散されたインデックスで 1 単語検索を行った場合に検索要求あたりの計算時間、通信量の合計の逆数で

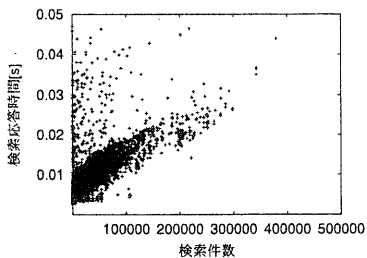


図 11 検索応答時間 (文書分散, 検索サーバ 4 台)

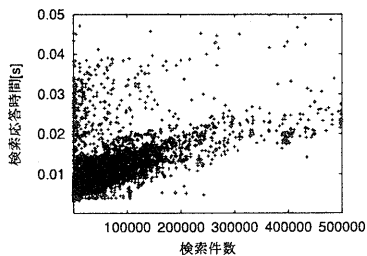


図 12 検索応答時間 (文書分散, 検索サーバ 8 台)

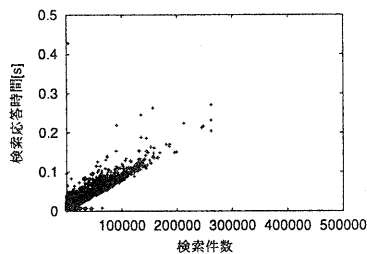


図 13 検索応答時間 (単語分散, 検索サーバ 4 台)

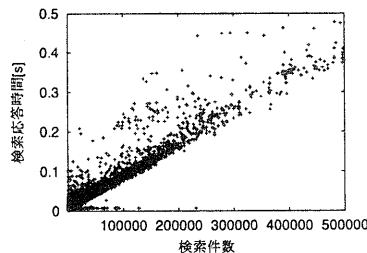


図 14 検索応答時間 (単語分散, 検索サーバ 8 台)

求められるスループットである。単語分散手法を取り入れる事で文書分散手法の性能を向上させる事が出来ると考えられる。

6. まとめ

検索システムのインデックス分散方法によるスケールビリティの変化について、文書分散手法と単語分散手法について負荷の推定による定式化を行い、実際にそれぞれの分散を行う検索システムを実装し評価実験により推

表 3 検索スループット

検索サーバ群	検索ゲートウェイ 1 台のスループット	検索サーバ群のスループット
文書分散, 4 台	187.70[件/秒]	343.37 [件/秒]
単語分散, 4 台	52.28[件/秒]	177.03 [件/秒]
文書分散, 8 台	143.90[件/秒]	341.47 [件/秒]
単語分散, 8 台	25.26[件/秒]	171.17 [件/秒]
併用手法, 8 台	157.24[件/秒]	374.40 [件/秒]

定の正当性の検証を行った。

負荷を定式化することで、文書分散手法と単語分散手法の特性を考察した。また、単語分散と文書分散を併用する事により、さらに負荷を軽減できることを示した。

今後の課題として、複数単語検索において単語分散と文書分散を併用する事による負荷の軽減を検討する事が挙げられる。複数単語検索においても検索件数が小さいならば単語分散手法が勝り、検索件数が大きい場合は文書分散手法が勝ると考えられるが、その境界を調べ、性能の向上率を調べることが必要と考えられる。

謝辞 本研究は一部平成 11~12 年度文部省科学研究費補助金・基盤研究 (C) (11680357) および P D C (並列・分散処理研究推進機構) の補助による。

参考文献

- 1) AltaVista : <http://www.altavista.com/>
- 2) goo : <http://www.goo.ne.jp/>
- 3) Lawrence, S. and Lee Giles, C. : "Searching the World Wide Web", Science, No.280, pp.98-100 (1998)
- 4) 内田斉, 宮沢浩, 大岩寛 : "わが国の WWW コンテンツ量の推計"
<http://www.a-brain.com/project/WebCount/index.htm>
- 5) Frakes, William B. and Baeza-Yates, R. : "Information Retrieval : data structure and algorithms" Prentice Hall (1992)
- 6) Baeza-Yates, R. and Ribeiro-Neto, B. : "Modern Information Retrieval" Addison-Wesley (1999)
- 7) Cahoon, B. and McKinley, Kathryn S. : "Performance Evaluation of a Distributed Architecture for Information Retrieval", 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.110-118 (1996)
- 8) 澤田雅人, 竹野浩, 藤本典幸, 萩原兼一 : "並列全文検索システム構築のための手法の提案とその評価", 情報処理学会研究報告, アルゴリズム 71-1, pp.1-8 (2000)
- 9) Silverstein, C., Henzinger, M., Marais, H. and Moricz, M. : "Analysis of a very large web search engine query log", SIGIR Forum, 33-3 (1999)
- 10) metacrawler :
<http://www.metacrawler.com/>