

# P2P 型コンテンツ検索システムのための効率的な Top-k 検索処理手法

松 波 秀 和<sup>†</sup> 寺 田 努<sup>††</sup> 西 尾 章 治 郎<sup>†</sup>

近年, P2P 型ネットワークを利用したコンテンツ共有への注目が高まっている. このようなシステムでは一般にフラッディングを用いて検索クエリを拡散させるため, 検索結果の件数が多い場合にクエリ応答が大量のトラフィックを発生させる. そこで本研究では, P2P 型ネットワークにおける効率的な Top-k クエリの処理手法を提案する. 提案手法では, ユーザが上位の検索結果しか必要としない場合が多いことに着目し, クエリ応答を抑制することでトラフィックを削減している. さらに, 本稿ではシミュレーション評価により, 提案手法の有効性を明らかにする.

## An Efficient Top-k Query Processing Method for P2P-based Contents Retrieval Systems

HIDEKAZU MATSUNAMI,<sup>†</sup> TSUTOMU TERADA<sup>††</sup> and SHOJIRO NISHIO<sup>†</sup>

Recently, there has been an increasing interest in the content sharing on peer-to-peer (P2P) networks. Since such a system employs a flooding mechanism for queries and because each peer returns many search results, the response to a query causes heavy traffic. Therefore, we propose a new efficient query processing method for top-k queries on P2P networks. We focus on the fact that users usually need search results only with a higher score. Our method reduces the reply traffic by controlling the number of query replies. Moreover, we evaluate the proposed method by simulation studies.

### 1. はじめに

近年, ネットワーク環境の整備により, WWW (World Wide Web) 上で公開されているウェブコンテンツの量が飛躍的な速度で増加しつつある. ユーザは Yahoo! や Google といった検索サービスを利用することにより, これらの膨大なコンテンツの中から必要な情報を効率良く探し出すことができる.

一方, 近年 P2P (Peer to Peer) 型のネットワークを利用したコンテンツ共有への注目が高まっている. クライアントがサーバからサービスの提供を受けるサーバ/クライアント型のシステムとは異なり, P2P 型のシステムでは, ピアと呼ばれるコンピュータ同士が相互に接続し, 平等な関係の下で直接リソースやサービスをやりとりする. サーバ/クライアント型のシステムではシステム提供者が利用者の増加に応じて

システムやネットワークを拡張する必要があるのに対し, P2P 型のシステムでは特定のコンピュータに負荷が集中しないためシステム提供者の負担が小さい.

ここで, P2P 型のシステムを用いてウェブコンテンツの検索を行うことを考える. サーバ/クライアント型のシステムでは, 検索キーワードに対してサーバが適切な検索結果を一定件数返信する仕組みとなっている. そのため, P2P 型コンテンツ検索システムに比べると総トラフィック量は小さい. 一方, 多くの P2P 型コンテンツ検索・共有システムでは, 各コンテンツについて検索条件に合致するかどうかのみを判断しているため, キーワード指定によるコンテンツの検索を実行すると, 検索結果が多くなり, キーワードへの関連度が高いコンテンツを得ることは難しい.

そこで本研究では, P2P 型コンテンツ検索・共有システムにおいてコンテンツとキーワードの関係からスコアが求まる環境を想定し, スコアが上位  $k$  個の検索結果を得る, いわゆる Top-k クエリのための効率的な問合せ処理手法を提案する. 提案する問合せ手法は, すべての問合せ結果を確実に取得する方法のほか, すべての検索結果が得られることを保証しないが, 十分に高い再現率を得ながらトラフィックおよび検索所要

<sup>†</sup> 大阪大学大学院情報科学研究科マルチメディア工学専攻  
Department of Multimedia Engineering, Graduate  
School of Information Science and Technology, Osaka  
University

<sup>††</sup> 大阪大学サイバーメディアセンター  
Cybermedia Center, Osaka University

時間を削減する手法を含む。後者の手法は、各ピアにおいてクエリ発行ピアからのホップ数に応じて返信するクエリ応答の数を抑制することでトラフィックの削減と高い再現率を実現している。また、本研究ではシミュレーション評価を行い、提案手法の有用性を明らかにする。

以下、2章で関連研究について述べ、3章で本研究における提案手法について説明する。4章では、3章で提案した手法についてシミュレーションによる性能評価を行い、最後に5章で本研究のまとめを行う。

## 2. 関連研究

中央サーバを利用しないピア P2P 型ネットワークのシステムは、大きく分けて、ネットワークのトポロジやコンテンツ配置が規定されている構造型検索トポロジを用いたものと、非構造型検索トポロジを用いたものに分けられる。

構造型検索トポロジを用いたシステムとしては CAN<sup>4)</sup>、Chord<sup>6)</sup>、Pastry<sup>5)</sup>、Tapestry<sup>9)</sup> があげられる。これらのシステムでは、コンテンツのファイル名やキーワードにハッシュ関数を適用することによりハッシュ値を算出し、このハッシュ値から分散ハッシュテーブルを利用してコンテンツ配置を決定する。コンテンツ検索時には、検索キーワードにハッシュ関数を適用し、ここで求まるハッシュ値からコンテンツが存在するピアを求められるため、コンテンツ検索時のネットワーク負荷が小さいという特徴がある。1つのコンテンツに対してヒットするキーワードの数が多い全文検索システムにこれらのシステムを適用すると、多くのキーワードに対するハッシュ値を算出し、多くのピアにコンテンツまたはそのコンテンツを示すインデックス情報を配置する必要があるため、構造型検索トポロジを用いたシステムは全文検索システムには不向きである。

pSearch<sup>7)</sup> では、CAN をベースとしてコンテンツ中の重要なキーワードを用いるアルゴリズムと、semantic vector を用いて同じようなキーワードを含むコンテンツをハッシュ空間上の近い位置に配置するアルゴリズムにより、効率的な全文検索を実現している。

非構造型検索トポロジを用いたシステムとしては、Gnutella や Freenet<sup>3)</sup> などがあげられる。これらのシステムでは、ネットワーク構成やコンテンツ配置に特別な制約がないため、ピアどうして検索前の情報交換する必要がないという利点があり、実用化されている P2P システムのほとんどはこの形である。しかし、精度の高い検索結果を得るためには多くのピアにクエ

リを送信する必要があるため、構造型検索トポロジを用いたシステムよりもネットワーク負荷は増大する。

Top-k クエリを Gnutella 型の P2P ネットワーク上で実現しているシステムとして Kalnis らのシステム<sup>2)</sup> があげられる。この手法では、検索クエリにクエリ応答の要求数  $k$  を含めてフラッディングする。この検索クエリを受信した各ピアは検索クエリで指定した条件によりコンテンツを順位付けし、上位  $k$  個のコンテンツについてのクエリ応答を返信する。また、各ピアにおける負荷を抑制する必要がある場合には、クエリの転送を凍結したり、TTL を強制的に減少させたりする。しかし、この手法ではクエリ応答自体のサイズが非常に小さい環境を想定しており、各ピアが返信するクエリ応答はすべてクエリ発行ピアまで転送される。本研究ではクエリ応答に数百バイトのメタデータを含むことを想定しており、この手法を適用するとクエリ応答によるトラフィックが増大する。

また、同様に Balke らは 1 回のクエリ送信に対して 1 個の応答を受信し、Top-k 検索の結果を得る手法を提案している<sup>1)</sup>。この手法では、Top-k 検索における 1 個の結果に対して 1 個のクエリが必要となるため、 $k$  が大きい場合にはクエリ遅延が大きくなるが、クエリ応答によるトラフィックを抑制できる点で優れている。

## 3. 提案手法

本研究では、Gnutella 型のネットワークを用いるコンテンツ検索システム上における Top-k 検索処理手法として、Fixed-k Query 手法、Reduce-k Query 手法、Delayed Fixed-k Query 手法、および Delayed Reduce-k Query 手法を提案する。

### 3.1 Fixed-k Query 手法

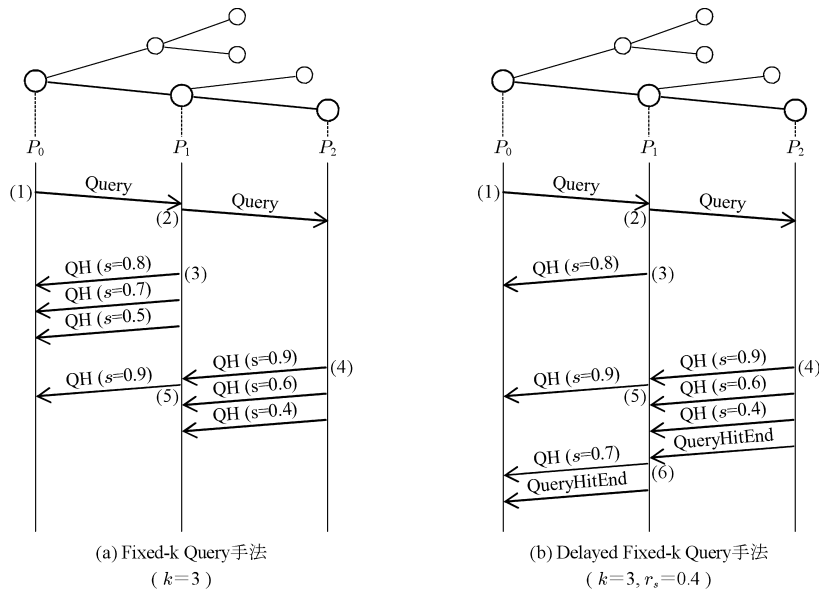
Fixed-k Query 手法は、各ピアにおいて上位  $k$  位以内のスコアを持つコンテンツを求めるクエリを単純にフラッディングする手法である。

本手法におけるコンテンツ検索の手順を、図 1 の (a) を用いて説明する。

(1) コンテンツ検索を行うクエリ発行ピア  $P_0$  は、ユーザやシステムの検索要求にあわせて以下の情報を含めた検索クエリを生成し、この検索クエリをすべての隣接ピアに送信する。

- クエリ ID
- コンテンツの検索条件
- クエリ応答要求数  $k$
- クエリの生存時間 (TTL)

クエリ ID は、ネットワーク上においてクエリ



(凡例)

Query: 検索クエリ QH( $s=x$ ): クエリ応答 (スコア:  $x$ ) QueryHitEnd: クエリ応答終了

図 1 Fixed-k Query 手法および Delayed Fixed-k Query 手法における検索手順

Fig. 1 The query-process for Fixed-k query method and Delayed fixed-k query method.

を一意に識別する ID である。同時に、 $P_0$  自身が保持しているコンテンツを対象としたコンテンツ検索を行う。提案手法において、コンテンツ検索とは、各ピアが保持しているコンテンツを対象として検索条件に合致するコンテンツを検索し、同時にその検索条件とコンテンツからスコアを算出することである。

- (2) 検索クエリを受信したピア  $P_1$  は、まず同じクエリ ID を持つクエリを過去に受信していないか検索し、受信したことがある場合は循環クエリとして受信したクエリを破棄し、以降の処理を行わない。同じクエリ ID を持つクエリを受信したことがない場合は、クエリの TTL を 1 つ減少させ、この TTL が 1 以上の場合にクエリ送信元ピアを除くすべての隣接ピアにクエリを転送する。
- (3) 検索クエリを受信したピア  $P_1$  はコンテンツ検索を行い、上位  $k$  ( $= 3$ ) 位以内のスコアを持つ各コンテンツについてクエリ応答を生成し、 $P_0$  に返信する。このクエリ応答には、以下の情報を含める。

- クエリ ID
- クエリ応答ピア ID
- コンテンツ識別子
- スコア

- そのほか、ユーザが閲覧するコンテンツを選択するために必要な情報

クエリ応答ピア ID はコンテンツ本体取得時にコンテンツが見つかったピアにメッセージを送るために利用する ID である。

- (4) (3) と同様にピア  $P_2$  でもコンテンツ検索を行い、クエリ応答を  $P_1$  に返信する。
- (5) クエリ応答を受信したピア  $P_1$  は、過去に自ピアで見つかったコンテンツと過去に受信したクエリ応答のスコアをあわせて順位付けを行い、ここで受信したクエリ応答のスコアが上位  $k$  位以内に入る場合に限りそのクエリ応答をクエリ送信元ピアに転送する。

Fixed-k Query 手法を利用することにより、クエリ発行ピアではネットワークやピアの故障が発生しない限りクエリ到達範囲内における Top-k コンテンツの一覧を確実に取得できる。

### 3.2 Reduce-k Query 手法

Fixed-k Query 手法はクエリ到達範囲内における Top-k コンテンツの一覧を確実に取得できる一方、Top-k に入る可能性があるすべてのコンテンツが返信の対象となるため、大量のクエリ応答がネットワーク上を流れてしまう。

そこで、トラフィックを削減するため、Reduce-k Query 手法を提案する。Reduce-k Query 手法では、

クエリ発行ピアにおいて Top-k コンテンツに入る可能性が高いコンテンツについてのみ返信の対象とし、クエリ発行ピア  $P_Q$  からのホップ数が大きくなるにつれて、またクエリ発行ピアから各ピアまでの経路上のピアにおける隣接ピア数が大きくなるにつれてクエリ応答の個数を小さくする。

各ピアにおいて返信または転送するクエリ応答の個数の決定方法を説明する。  $P_Q$  から  $i$  ホップ離れたピアにおけるクエリ応答返信数を  $k_i$  とする。各ピアは、クエリ転送時に自ピアにおけるクエリ応答返信数  $k_i$  と、検索クエリのパラメータとして指定するクエリ応答返信数減少余裕率  $r_m$ 、自ピアの検索クエリ転送先となるピア数  $n_{p_i}$  を利用してクエリ転送先ピアにおけるクエリ応答返信数  $k_{i+1}$  を決定する。ただし、 $k_i$  の初期値である  $k_0$  はシステムが設定することとした。

$k_{i+1}$  の値は、まず仮の値  $k'_{i+1}$  を

$$k'_{i+1} = \left\lfloor \frac{k_i r_m}{n_{p_i}} + 0.5 \right\rfloor \quad (1)$$

としたうえで、次式により決定する。

$$k_{i+1} = \begin{cases} k_i & (k_i \leq k'_{i+1}) \\ k'_{i+1} & (2 \leq k'_{i+1} < k_i) \\ 2 & (k'_{i+1} < 2) \end{cases} \quad (2)$$

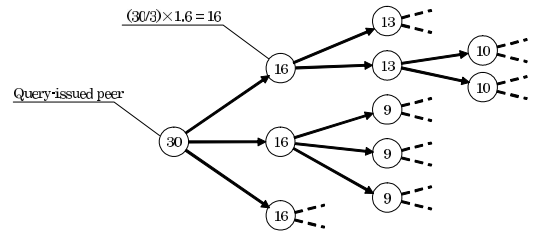
ここで、 $\lfloor \cdot \rfloor$  は  $\lfloor \cdot \rfloor$  と  $\lceil \cdot \rceil$  で囲まれた値を超えない最大の整数を意味する記号であり、四捨五入するために 0.5 を加算している。

各隣接ピアから受信するクエリ応答のうち自ピアにおいて転送対象となるものの数は隣接ピアの数に反比例すると考えられる。その数は、仮に各隣接ピアから受信するクエリ応答のスコア分布が均等であるとする、各ピアが送信するクエリ応答のうち自ピアが転送するクエリ応答に含まれるものの数は、自ピアが転送したクエリの数で自ピアにおけるクエリ応答返信数で割ったもの  $(k_i/n_{p_i})$  になる。しかし、隣接ピア数の不一致やコンテンツ分布の偏りを考慮すると、高い再現率を実現するためにはクエリ応答返信数に余裕を持たせる必要がある。ここで、本研究では  $k_i/n_{p_i}$  にクエリ応答返信数減少余裕率  $r_m$  をかけた値を用いることとした。各ピアにおけるクエリ応答送信数の決定方法の例を、図 2 に示す。

Reduce-k Query 手法の検索動作は、クエリ応答数の変化を除いて Fixed-k Query 手法と同等である。

### 3.3 Delayed Fixed-k Query 手法および Delayed Reduce-k Query 手法

Fixed-k Query 手法および Reduce-k Query 手法では、各ピアは問合せやクエリ応答を受信した段階



( $k = k_0 = 30$ ,  $r_m = 1.6$  の場合、図中の数字は  $k_i$ )

図 2 各ピアにおいて返信するクエリ応答数

Fig. 2 The number of query-replies for each peer.

で返信（他のピアから受信したクエリ応答の転送を含む）すべきクエリ応答をただちに返信する。そのため、結果的に Top-k に入らないクエリ応答も返信されることとなる。そこで、各ピアでクエリ応答の返信をすべての隣接ピアからのクエリ応答受信が完了するまで遅延させ、スコアが送信すべき順位から外れるクエリ応答を返信しないことによりクエリ応答返信数を抑制する手法を提案する。Fixed-k Query 手法に本方法を適用したものを Delayed Fixed-k Query 手法、Reduce-k Query 手法に適用したものを Delayed Reduce-k Query 手法と呼ぶ。

クエリ応答の受信完了を検出するため、各ピアはクエリ応答の返信完了後、および循環クエリとしてクエリを破棄する場合にクエリ応答終了メッセージをクエリ送信元ピアに送信する。また、検索クエリにはタイムアウト時間  $t_o$  を設定し、クエリ応答の受信が  $t_o$  の間途絶えた場合は、クエリ応答が終了したと判断する。さらに、検索にかかる時間を短縮するため、各ピアにおいて保持するクエリ応答のうちスコアが特に上位であるものについては、コンテンツ検索またはクエリ応答受信が終了した時点においてただちに返信する。その対象となるクエリ応答の最低順位  $N_s$  は、各ピアにおいて送信すべきスコアの順位  $k_i$  (Delayed Fixed-k Query 手法においては  $k_i = k$ ) のうち、一定割合  $r_s$  とし、

$$N_{s_i} = \lfloor k_i r_s \rfloor \quad (3)$$

とする。ここで、パラメータ  $r_s$  をクエリ応答即時送信率と定義する。

Delayed Fixed-k Query 手法の手順について、図 1 の (b) を用いて説明する。Delayed Reduce-k Query 手法の手順は、各ピアが返信するクエリ応答のスコアの最低順位が異なる以外は Delayed Fixed-k Query 手法と同じである。

- (1) クエリ発行ピア  $P_0$  は、生成した検索クエリを全隣接ピアに送信する。
- (2) 検索クエリを受信した  $P_1$  は、Fixed-k Query

手法の場合と同様に、検索クエリを  $P_0$  以外の全隣接ピアに転送する。

- (3)  $P_1$  はコンテンツ検索を行い、上位  $N_s (= 1)$  個のコンテンツについてのクエリ応答のみを  $P_0$  に返信する。
- (4)  $P_2$  はコンテンツ検索を行う。ここで、 $P_2$  はクエリ転送を行っていないピアであり、他のピアからクエリ応答を受信しないので  $k (= 3)$  個のクエリ応答を  $P_1$  へただちに返信する。これに続いて、クエリ応答終了メッセージを  $P_1$  に送信する。
- (5)  $P_1$  が受信したクエリ応答のうち 1 個のスコアは上位  $N_{s_i} (= 1)$  位以内であるため、該当するクエリ応答をただちに  $P_0$  へ転送する。
- (6)  $P_1$  は、クエリを転送した先のピアのうちすべてのピアからのクエリ応答終了メッセージを受信することにより、受信すべきすべてのクエリ応答を受信したと判断できる。この段階で、上位  $k (= 3)$  位以内のスコアを持つクエリ応答のうち、これまでに返信していないものについて  $P_0$  に返信し、最後にクエリ応答終了メッセージを  $P_0$  に送信する。

#### 4. 性能評価

提案手法の有効性を明らかにするため、コンテンツ検索にかかる時間、再現率、およびコンテンツ検索に必要なトラフィック量について、シミュレータを用いて性能評価を行った。

##### 4.1 シミュレーション環境

本稿では、P2P ネットワークを用いたウェブコンテンツ全文検索システムを想定し、シミュレーション環境を設定する。各シミュレーションは 5 回ずつ繰り返し、結果はそれらの平均をとることとした。

##### 4.1.1 ネットワークのトポロジ

Gnutella における各ノードの隣接ノード数は、文献 8) の先行研究によりべき法則 (Power Law) の性質に従うことが示されている。本稿におけるシミュレータでは、トポロジの生成に PLRG (Power-Law Random Graph) を利用し、各ピアの隣接ピア数がべき法則に従うこととする。PLRG では、rank exponent  $R$  と、最大隣接ピア数  $w_{\max}$  という 2 つのパラメータを使用し、ネットワークの総ピア数を  $N_p$  とするとピア番号  $j$  における隣接ピア数  $d_j$  は次式で与えられる。

$$d_j = \lfloor w_{\max} \cdot j^R \rfloor \quad (1 \leq j \leq N_p) \quad (4)$$

PLRG を用いてトポロジを生成することにより、各ピアの隣接ピア数が一定とした場合に比べると他のピア

に比べてトラフィックの多いピアが存在することとなる。また、クエリ発行ピアからのホップ数が大きいピアにおいてもクエリ応答返信数が大きくなるピアが存在するが、該当するピア数は多くないため、隣接ピア数を固定したようなグラフを用いた場合でも同様の評価結果となると予想できる。

##### 4.1.2 コンテンツの配布

P2P ネットワークにおけるコンテンツ分布は Zipf 分布に基づくことが知られている<sup>10)</sup>。そこで本研究では、実環境に近い環境におけるシミュレーション評価を行うため、文献 10) における観測結果を用いて、Zipf 係数  $\alpha = -0.9$  とした Zipf 分布に基づいてコンテンツをランダムに各ピアに配布することとした。シミュレーション環境中の総ピア数  $N_p$  が 10,000 の場合における人気順位とコンテンツ配布先のピア数の関係を図 3 に示す。

##### 4.1.3 コンテンツの検索

本研究におけるシミュレータでは、コンテンツ検索処理におけるスコアの計算に、乱数を用いて生成された 31 ビットの整数で表現されるコンテンツ ID を用いる。各コンテンツのスコアは図 4 に示すように、問合せに対してある特定のコンテンツが高いスコア  $2r+1$

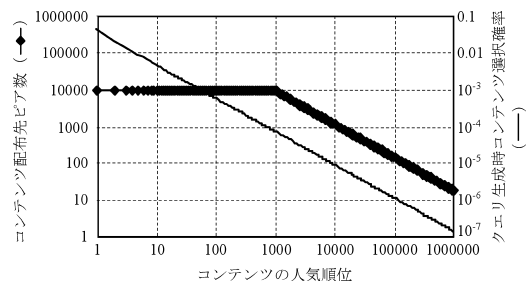


図 3 コンテンツ配布先ピア数とコンテンツ選択確率

Fig. 3 Number of peers vs. probability of contents selection.

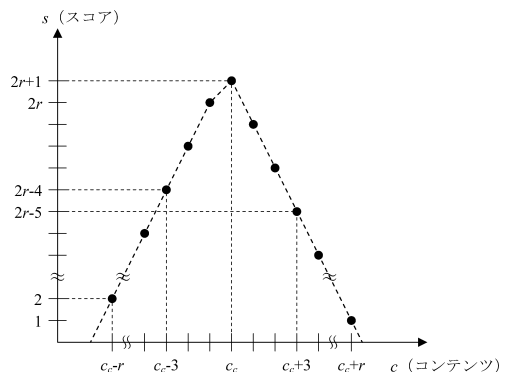


図 4 スコアの算出例

Fig. 4 An example of calculating score.

を持ち、そこから範囲  $r$  のコンテンツが線形に減少するスコアを持つこととした。クエリに含まれる検索条件はコンテンツ ID の中心値  $c_c$  と、範囲  $r$  で構成することとした。 $c_c$  は、図 3 に示すコンテンツ選択確率に従って、ネットワーク中に存在するコンテンツの中から選択する。 $r$  は、検索時に全コンテンツのうち検索条件にヒットするコンテンツの割合であるヒット率  $r_h$  およびコンテンツ ID の最大値  $c_{\max} = 2^{31} - 1$  から、

$$r = c_{\max} \times r_h \quad (5)$$

とした。このヒット率  $r_h$  は、実際の環境においてはクエリごとに様々な値になると考えられるが、本研究ではヒット率の違いによる性能評価を行うため、すべてのクエリについて同じ値とする。

コンテンツ ID が  $c$  のコンテンツのスコア  $s$  は、 $c$  および  $c_c$ 、 $r$  から、次式のとおり求めることとした。

$$s = \begin{cases} 2(r - |c - c_c|) + 1 & (c \geq c_c) \\ 2(r - |c - c_c|) + 2 & (c < c_c) \end{cases} \quad (6)$$

これにより、 $c$  が  $c_c$  に近いほど  $s$  は大きくなり、範囲  $r$  内で同じスコアを持つコンテンツが存在しなくなる。スコアの算出例を、図 4 に示す。たとえば、 $c = c_c - 3$  の場合、式 (6) より、 $s = 2r - 4$  となる。 $s > 0$  の場合にコンテンツ ID が  $c$  であるコンテンツは検索クエリにヒットしたこととする。

また、コンテンツはクエリ応答と同様に目的のピアから P2P ネットワークを通じて順次ピア伝いに転送されることとしている。クエリ発行ピアは検索終了の 10 秒後に最低 1 個、最大で 10 個のコンテンツを取得する。ここでは、スコアが最も高いコンテンツは必ず取得することとし、スコアが 2 位以下のコンテンツはランダムに選択することとした。

#### 4.1.4 評価項目

本稿では、検索所要時間、再現率、および送信メッセージ量の 3 点について評価を行う。

検索所要時間は、クエリ発行ピアにおいてクエリを発行してから最後に上位  $k$  位以内に入るスコアを持つクエリ応答を受信するまでの時間とする。

再現率は、検索クエリを受信したピアすべてのコンテンツのうち上位  $k$  位以内のコンテンツに関するクエリ応答について、クエリ発行ピアで得られた割合を示す。

送信メッセージ量は、シミュレーション時間中に各ピアが送信したコンテンツ検索に関するメッセージのデータ量の平均である。コンテンツ取得に関わるトラフィックは含まないこととする。

#### 4.2 $k_0$ とクエリ応答返信数減少余裕率の影響

まず、予備の評価として、 $k = 30$  の場合において Delayed Reduce-k Query 手法を適用する場合における  $k_0$  および  $r_m$  のパラメータの値の設定がクエリ応答返信数および再現率に与える影響を評価した。 $k_0$  は 30, 50, 100 とし、 $r_m$  を 1.2 と 4 の間で変化させた。これ以外のパラメータは表 1 に示したとおりである。

再現率と送信メッセージ量の関係を図 5 に示す。 $r_m$  を大きくするほど再現率は高くなるが、メッセージ量は増大する。グラフより再現率が 90%~95%、送信メッセージ量が 2~4 MB 程度の領域でグラフが折れ曲がっていることから、この付近の値を取るように  $r_m$  を設定すると、再現率と送信メッセージ量の双方について適切な値が得られるといえる。また、 $k_0$  が大き

表 1 シミュレーション評価におけるパラメータ

Table 1 Simulation parameters.

| シミュレーション環境       |           |      |
|------------------|-----------|------|
| 総ピア数 $N_p$       | 10,000    |      |
| $R$              | -0.4      |      |
| $w_{\max}$       | 100       |      |
| コンテンツの種類数        | 1,000,000 |      |
| 各ピアでの平均クエリ発生間隔   | 1,000     | 秒    |
| 通信速度             | 2         | Mbps |
| ピア内コンテンツ検索にかかる時間 | 0.1       | 秒    |

#### 各メッセージのサイズ

|               |     |     |
|---------------|-----|-----|
| 検索クエリ         | 140 | バイト |
| クエリ応答         | 640 | バイト |
| クエリ応答終了メッセージ  | 64  | バイト |
| コンテンツ本体 (平均値) | 40  | kB  |

#### 検索条件

|                     |      |
|---------------------|------|
| TTL                 | 5    |
| クエリ応答要求数 $k$        | 30   |
| $k_0$               | 100  |
| クエリ応答返信数減少余裕率 $r_m$ | 1.8  |
| ヒット率 $r_h$          | 0.3% |
| クエリ応答タイムアウト時間 $t_o$ | 0.5  |
| 即時送信率 $r_s$         | 10%  |

(本文中で異なる値を示している場合、その値を優先する)

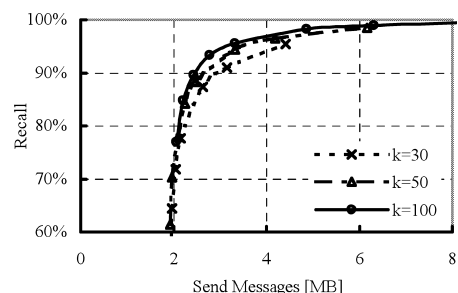


図 5 再現率の変化に対するピアあたりクエリ応答の送信数  
Fig. 5 Recall vs. query-replies.

表 2  $k_0$  および  $r_m$  と、再現率および送信メッセージ量の関係  
Table 2  $k_0$ ,  $r_m$ , recall and messages.

| $k_0$ | $r_m$ | 再現率   | 送信メッセージ量  |
|-------|-------|-------|-----------|
| 30    | 2.6   | 87.4% | 2.63 [MB] |
| 50    | 2     | 88.3% | 2.48 [MB] |
| 100   | 1.6   | 89.6% | 2.43 [MB] |
| 100   | 1.8   | 93.3% | 2.78 [MB] |

いほど同等の再現率を得るよう  $r_m$  を設定した場合における送信メッセージ量が小さく、効率的な検索が可能となることが分かる。再現率が 87% から 90% 程度になる場合におけるパラメータと評価結果を表 2 に示す。

#### 4.3 ヒット率を変化させた場合の評価

ヒット率を変化させたときの、送信メッセージ量と再現率の変化について、提案手法である Fixed-k Query 手法 (F), Reduce-k Query 手法 (R), Delayed Fixed-k Query 手法 (DF), Delayed Reduce-k Query 手法 (DR), および Balke 氏提案の手法 (B) との間で比較した結果を図 6 に示す。

Fixed-k Query 手法は全般に再現率が高く保たれているが、ヒット率が高くなるにつれてメッセージのデータ量が多くなり、最大で Delayed Reduce-k Query 手法の 10 倍近くに達している。ヒット率が 0.001 以上の場合においては、再現率が 99.5% 程度となっているが、この原因はピアにおけるデータ転送が間に合わなくなったためである。

これに対して、Reduce-k Query 手法では、送信メッセージ量を Fixed-k query 手法に比べて最大で 69% 削減した。また、Delayed Reduce-k Query 手法を用いることにより、送信メッセージ量をさらに約 16% 削減した。しかし、Reduce-k Query 手法および Delayed Reduce-k Query 手法では、再現率が他の手法に比べて低く、Reduce-k Query 手法で約 96%, Delayed Reduce-k Query 手法で約 93% である。この再現率は  $k_0$  および  $r_m$  の設定により変化するが、4.2 節で述べたとおり、再現率を向上させた場合はメッセージ量が増大するため、メッセージ量と再現率の双方を考慮して  $k_0$  および  $r_m$  を適切に設定する必要がある。

最後に、Balke らの手法では、再現率が 100% であり、クエリ到達範囲内における Top-k コンテンツについてのクエリ応答を確実に取得できている。メッセージのデータ量はヒット率が高い場合においても比較した手法の間で最小である。しかし、1 つの検索結果を確定させるごとにクエリを送信し、その応答を待つ必要があるため、検索所要時間が Delayed Reduce-k Query 手法の 13 倍程度と大きくなっている。

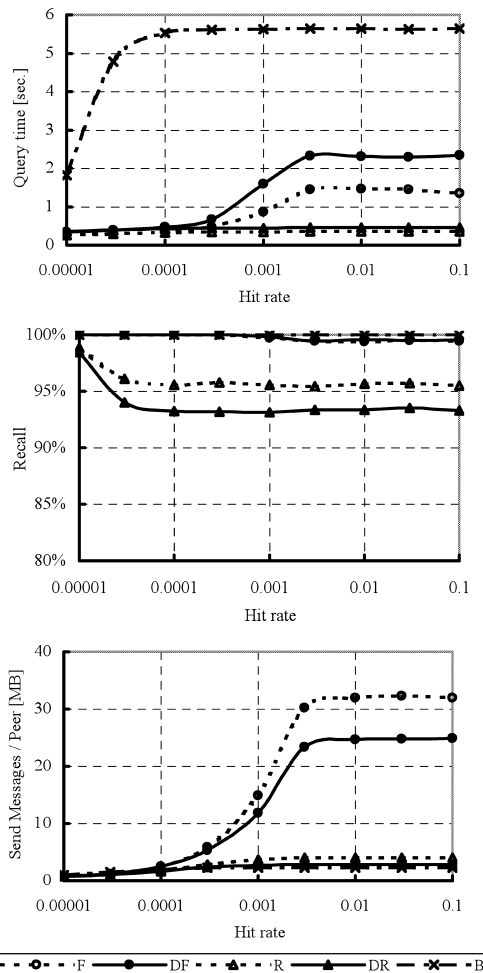


図 6 ヒット率の変化に対する所要時間、再現率、送信メッセージ量  
Fig. 6 Hit-rate vs. query-time, recall and message.

提案手法の間では、メッセージ量の多い Fixed-k Query 手法および Delayed Fixed-k Query 手法の検索所要時間が大きくなっている。また、各ピアがクエリ応答の受信完了を待ってクエリ応答を返信するため、Reduce-k Query 手法より Delayed Reduce-k Query 手法、Fixed-k Query 手法より Delayed Fixed-k Query 手法の方が検索所要時間が大きくなっている。

#### 4.4 応答要求数 ( $k$ ) を変化させた場合の評価

クエリ応答要求数 ( $k$ ) を 5, 10, 30, 50 に変化した場合の、メッセージ量、再現率、および検索所要時間の変化を図 7 に示す。ここで、Reduce-k Query 手法および Delayed Reduce-k Query 手法における  $k_0$  は  $k$  に比例させることとする。このため、 $k$  が 5 のとき  $k_0$  は 17,  $k$  が 10 のとき  $k_0$  は 34,  $k$  が 50 のとき  $k_0$  は 167 となる。

いずれの手法でも  $k$  を増加させるとメッセージ量

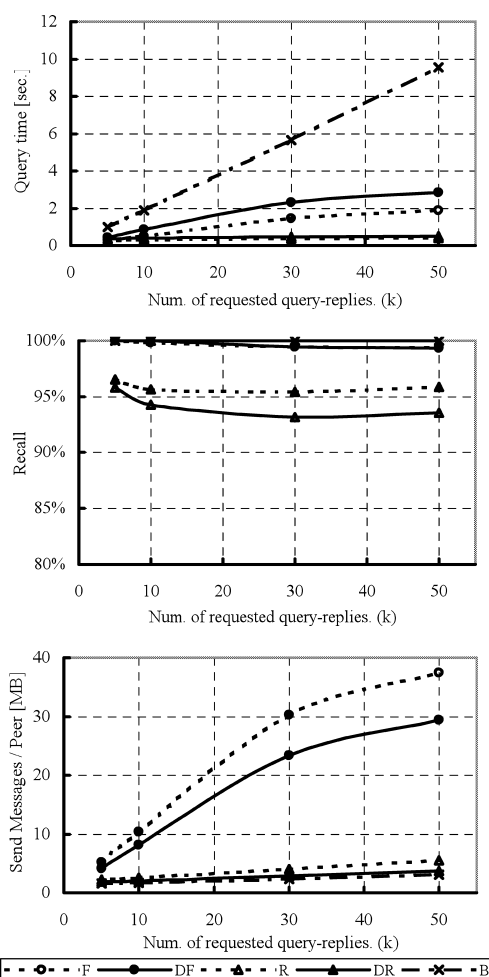


図 7  $k$  の変化に対する所要時間, 再現率, 送信メッセージ量  
Fig. 7  $k$  vs. query-time, recall, and message.

および検索所要時間は増大する。しかし, Delayed Reduce-k Query 手法では, 所要時間の増大が小さく, またメッセージ量の増加量も Balke らの手法とほぼ同等である。なお, 再現率は,  $k$  を変化させても大きな変化はみられず, この点から, Delayed Reduce-k Query 手法により, 特に  $k$  を大きくした場合における所要時間およびメッセージ量の増加が抑制できることから, メタデータを眺めながら必要なコンテンツを選択するような場合に, 参照するメタデータの数をもくしても効率的である。

#### 4.5 考 察

本稿では, 4つの手法を提案した。まず, Delayed のつかない Fixed-k Query 手法および Reduce-k Query 手法と, Delayed Fixed-k Query 手法および Delayed Reduce-k Query 手法の間で比較すると, 前者では各ピアにおいてクエリ応答の受信を待たずに返信を行う

ため, トラフィックは大きくなるが比較的早い時間で検索結果が得られ, 後者はトラフィックを削減できる一方で検索結果を得るまでに時間がかかるという特徴がある。

本稿においては, メッセージの不到達や大きな遅延がなく, かつピアの参加や退出が発生しない環境においてシミュレーションを行ったが, 実際のネットワークではメッセージの不到達や大きな遅延, ピアの参加や退出が発生する。Delayed のつく手法では, ピアの退出を検知できない場合や応答に時間がかかる場合にクエリ応答の返信を待ち続けると, クエリ発行ピアにおいて検索結果を取得するまでの時間が増大するという問題がある。そのため, ネットワークが不安定な環境においては, 現時点では Delayed のつかない Fixed-k Query 手法および Reduce-k Query 手法の方が適しているといえるが, Delayed のつく手法についても, 検索クエリが到達したことを確認できないピアやクエリ応答をまったく返信しないピアからのクエリ応答を待たずに返信することにより所要時間を小さくすることは可能である。

さらに, ピアの参加や退出が頻繁に発生する環境では, 検索クエリを受信後, 返信すべきクエリ応答の全部または一部を返信することなく退出するピアも発生すると考えられる。この場合, 現在の手法では再現率が大きく低下するため, そのような環境では Delayed のつかない Fixed-k Query 手法または Reduce-k Query 手法を用いるべきである。また, ピア切断時に P2P ネットワークの再構築を行い, 新たな接続を用いてのクエリ応答を返信可能とするなど, この問題を解決する手法の提案は今後の課題である。

また, Fixed-k Query 手法および Delayed Fixed-k Query 手法では, Reduce-k Query 手法および Delayed Reduce-k Query 手法に比べてトラフィックは増大するがクエリ到達範囲内における Top-k コンテンツに関するクエリ応答を確実に取得可能なため, ヒットするコンテンツ数自体が少ない環境や, 少数の検索結果でよいが確実に検索結果を得たい場合などでは, これらの手法が適している。一方, Reduce-k Query 手法および Delayed Reduce-k Query 手法は, 再現率を 100% とすることができず, 特に検索結果が一部のピアに固まっている場合に再現率が低下する反面, Fixed-k Query 手法および Delayed Fixed-k Query 手法に比べると大幅な検索トラフィックの抑制が可能である。そのため, (Delayed) Reduce-k Query 手法は多くのコンテンツが検索クエリにヒットする環境において, クエリ到達範囲内における Top-k クエリ応答



のうちすべてを確実に取得する必要はないがトラフィックを減少させる必要がある場合に適しており、クエリ応答返信数減少余裕率  $r_m$  を増減させることによりトラフィックの抑制と再現率の向上のうちいずれを優先させるかを選択できる。

最後に、Balke らの手法は、所要時間は長い、トラフィックを小さく抑えたうえでクエリ到達範囲内における Top-k コンテンツに関するクエリ応答を確実に取得できる。そのため、検索所要時間が長くても問題なく、かつネットワークが比較的安定している場合には Balke らの手法が適しているといえる。

## 5. おわりに

本研究では、P2P ネットワーク上における Top-k クエリを効率的に行う手法を提案した。また、シミュレーションにより提案方式の性能評価を行った。その結果から、各ピアにおけるクエリ応答の送信数は、提案手法を利用することにより一定の値以下に抑えることができることを示した。さらに、提案手法のうち Delayed Reduce-k Query 手法では、Fixed-k Query 手法と比べてトラフィックを大きく削減できることを示した。さらに、Balke らが提案する手法と比べても、メッセージ量は増加するものの、検索所要時間を削減できることを示した。

今後は、各ピアが保持するコンテンツがジャンルなどによって偏っている環境を考慮して手法の改良を行い、より精度の高いシミュレーションを行う予定である。また、コンテンツのスコアが一意に算出できない場合の対処や、同一コンテンツの複製を多数発見した場合を考慮することにより、より精度の高い検索を実現する手法を検討する。

**謝辞** 本研究の一部は、文部科学省 21 世紀 COE プログラム「ネットワーク共生環境を築く情報技術の創出」、および基盤研究 (A) (17200006)、基盤研究 (B) (2) (15300033) の研究助成によるものである。ここに記して謝意を表す。

## 参 考 文 献

- 1) Balke, W.-T., Nejdl, W., Siberski, W. and Thaden, U.: Progressive Distributed Top-k Retrieval in Peer-to-Peer Networks, *The 21st International Conference on Data Engineering (ICDE 2005)*, pp.174-185 (2005).
- 2) Kalnis, P., Ng, W., Ooi, B. and Tan, K.-L.: Answering Similarity Queries in Peer-to-Peer Networks, *International World Wide Web Conference*, pp.482-483 (2004).

- 3) larke, I., Sandberg, O., Wiley, B. and Hong, T.: Freenet: A Distributed Anonymous Information Storage and Retrieval System, *ICSI Workshop on Design Issues in Anonymity and Unobservability*, pp.46-66 (2000).
- 4) Ratnasamy, S., Francis, P., Handley, M., Karp, R. and Shenker, S.: A Scalable Content-Addressable Network, *2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pp.161-172 (2001).
- 5) Rowstron, A. and Druschel, P.: Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems, *Middleware 2001*, pp.329-350 (2001).
- 6) Stoica, I., Morris, R., Karger, D., Kaashoek, M.F. and Balakrishnan, H.: Chord: A Scalable Peer-to-Peer Lookup Service for Internet Application, *SIGCOMM'01*, pp.149-160 (2001).
- 7) Tang, C., Xu, Z. and Mahalingam, M.: pSearch: Information Retrieval in Structured Overlays, *ACM SIGCOMM Computer Communications Review*, Vol.33, pp.89-94 (2003).
- 8) Zeinalipour-Yatzi, D. and Folias, T.: A Quantitative Analysis of the Gnutella Network Traffic, Technical report, Dept. of Computer Science, University of California (2002).
- 9) Zhao, B., Kubiawicz, J. and Joseph, A.: Tapestry: An Infrastructure for Wide-area Fault-tolerant Location and Routing, Technical report, U. C. Berkeley Technical Report USB//CSD-01-1141 (2001).
- 10) 亀井 聡, 森 達哉, 大井恵太, 木村卓巳: P2P ファイル共有ネットワークの現状, 第 14 回インターネット技術第 163 委員会研究会 (2003). <http://www.itrc.net/report/meet14/NGN/kamei.pdf>

(平成 17 年 11 月 28 日受付)

(平成 18 年 6 月 1 日採録)

## 松波 秀和



2004 年大阪大学工学部電子情報エネルギー工学科卒業。2006 年同大学大学院情報科学研究科博士前期課程修了。在学中は P2P ネットワークの研究に従事。現在、デジタルプロセス株式会社所属。

**寺田 努 (正会員)**

1997 年大阪大学工学部情報システム工学科卒業。1999 年同大学大学院工学研究科博士前期課程修了。2000 年同大学院工学研究科博士後期課程退学。同年より大阪大学サイバーメディアセンター助手。2005 年より同講師。現在に至る。2002 年より同大学院情報科学研究科マルチメディア工学専攻助手, 2005 年より同講師を併任。2004 年より特定非営利活動法人ウェアラブルコンピュータ研究開発機構理事, 2005 年より同機構事務局長を兼務。工学博士。アクティブデータベース, ウェアラブルコンピューティング, ユビキタスコンピューティングの研究に従事。IEEE, 電子情報通信学会, 日本データベース学会の各会員。

**西尾章治郎 (正会員)**

1975 年京都大学工学部数理工学科卒業。1980 年同大学大学院工学研究科博士後期課程修了。工学博士。京都大学工学部助手, 大阪大学基礎工学部および情報処理教育センター助教授, 大阪大学大学院工学研究科情報システム工学専攻教授を経て, 2002 年より同大学院情報科学研究科マルチメディア工学専攻教授となり, 現在に至る。2000 年より大阪大学サイバーメディアセンター長, 2003 年より大阪大学大学院情報科学研究科長を併任。この間, カナダ・ウォータールー大学, ビクトリア大学客員。データベース, マルチメディアシステムの研究に従事。現在, Data & Knowledge Engineering 等の論文誌編集委員。本会理事を歴任。電子情報通信学会フェローを含め, ACM, IEEE 等, 8 学会の会員。