

P2P 型コンテンツ検索システムにおける効率的な Top-k 検索処理手法

松波 秀和[†] 寺田 努^{††} 西尾章治郎[†]

[†] 大阪大学大学院情報科学研究科 〒 565-0871 大阪府吹田市山田丘 1-5

^{††} 大阪大学サイバーメディアセンター 〒 567-0047 大阪府茨木市美穂ヶ丘 5-1

E-mail: [†]{matunami.hidekazu,nishio}@ist.osaka-u.ac.jp, ^{††}tsutomu@cmc.osaka-u.ac.jp

あらまし 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. キーワード P2P ネットワーク, 情報検索, ネットワーク応用, 性能評価

An Efficient Top-k Query Processing Method on a P2P-based Contents Retrieval System

Hidekazu MATSUNAMI[†], Tsutomu TERADA^{††}, and Shojiro NISHIO[†]

[†] Graduate School of Information Science and Technology, Osaka University Yamadaoka 1-5, Suita-shi, Osaka, 565-0871 Japan

^{††} Cybermedia Center, Osaka University Mihogaoka 5-1, Ibaraki-shi, Osaka, 567-0047 Japan

E-mail: [†]{matunami.hidekazu,nishio}@ist.osaka-u.ac.jp, ^{††}tsutomu@cmc.osaka-u.ac.jp

Abstract Recently, there has been increasing interest on research of the contents sharing on peer-to-peer (P2P) network. Since such a system employs flooding for a query, the replies for the query may bring about heavy traffic in the case each peer replies many search results. Therefore, we propose a new efficient query processing method for top-k query on P2P network. We focus that users usually need search results only which have higher score. We reduce the reply traffic by controlling the number of query replies. Moreover, we show the availability of proposing method by simulation studies.

Key words P2P network, information retrieval, network application, performance evaluation

1. はじめに

近年, ネットワークインフラの整備により, 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! [12] や Google [3] 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よく探し出すことができる. しかし, サーバやネットワークが不調なときは検索結果が得られてもコンテンツにたどり着くことができない. また, コンテンツは随時更新されていくためユーザが必要なコンテンツが別のもので置き換えられている可能性もある.

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に接続し, サーバから一方的に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, コンピュータ同士が相互に接続し平等な関係の下でお互いに直接リソースやサービスをやり取りする. P2P 型のネットワークを用いたコンテンツ共有により, P2P ネットワーク上の多数のコンピュータが保持しているコンテンツに対する検索および取得が可能となる.

筆者らの研究グループでは, これまで各ユーザが WWW 上から収集したウェブコンテンツを P2P 型のネットワークを用いて相互に共有し, 共有コンテンツに対する検索および閲覧を可能とするウェブコンテンツ共有システムを提案し, プロトタイプシステムの実装を行ってきた [7]. ウェブコンテンツ共有システムでは, 各ユーザが WWW 上から収集し, ローカルディスクに保存しておいたウェブコンテンツを, Freenet [4] や Gnutella [2] のようなピア P2P 型のネットワークを用いて共有する.

一般的に, Google などにおけるウェブコンテンツの検索では, キーワードを指定した全文検索を行う. 近年の検索エンジ

ンの改良により、少数のキーワードでも精度のよい検索が可能であるため、多くのユーザは比較的単純なキーワードしか指定しない。また、ウェブコンテンツのサイズは Gnutella などと主に共有されている音楽ファイルや映像ファイルに比べると小さく、ウェブコンテンツ共有システムでは各ピアに大量のコンテンツが格納されていると考えられる。そのため、1回のクエリに対して膨大な数のウェブコンテンツが検索条件に合致し、大量のクエリ応答トラフィックが発生する。しかし、大量のクエリ応答を受信したユーザが閲覧できるコンテンツの数は限られるため、クエリ発行ピアでは上位数個から数十個程度のコンテンツについてのクエリ応答を受信できれば十分であると言える。

そこで本研究では、P2P型のコンテンツ検索・共有システムにおいて各コンテンツとキーワードとの関連度により算出されるスコアを利用し、上位 k 個の検索結果を得る、いわゆる Top-k クエリを実現するための効率的な手法を提案する。本稿で提案する手法では、クエリ発行ピアからのホップ数に応じてピアに要求するクエリ応答の数を抑制することにより、再現率を確保しつつ、クエリ応答のトラフィックを大幅に減少させる。また、シミュレーション評価により他の手法との比較を行い、提案手法の有用性について考察する。

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、5章では4章における評価を踏まえて提案手法の拡張について述べる。最後に6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、大きく分けて構造化検索トポロジを用いたものと非構造化検索トポロジを用いたものに分けられる。

構造化検索トポロジを用いたシステムとして CAN [8]、Chord [10]、Pastry [9]、Tapestry [13] などが提案されている。これらのシステムでは、分散ハッシュテーブル (DHT) を利用してハッシュ空間におけるコンテンツのキーの配置を厳密に決定する。キーの算出にはファイル名やコンテンツのキーワードを用いる。これにより、コンテンツ検索時にキーを算出し、キーが存在するピアを求め、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が、1つのコンテンツにヒットするキーワード数が多い全文検索にこれらのシステムは向きである。

pSearch [11] というシステムでは 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 pVSM (Peer-to-peer Vector Space Model) 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 pLSI (Peer-to-peer Latent Semantic Indexing) アルゴリズムの利用により、効率的な全文検索を実現している。

非構造化検索トポロジを用いたシステムとしては、Gnutella や Freenet などが挙げられる。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、ピア同士で検索前の情報交換が不要という利点がある。しかし、良い

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、構造化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 [5] 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、各ピアは検索クエリの転送時に過去に受信した検索クエリと照合し、検索条件が似ている場合には検索クエリをこれ以上転送せず、過去に受信した検索クエリの検索結果を利用する。本研究で提案する手法ではこの手法においてフラッディングされるクエリに対する応答に注目し、さらなる検索の効率化を実現している。

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 [1]。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

本章では、本研究で提案した3つの Top-k クエリ処理手法について述べる。

3.1 Simple Top-k 手法

この手法は、単純に各ピアに対して k 個のクエリ応答を要求するクエリをフラッディングし、各ピアが k 個のコンテンツに関するクエリ応答を返信する手法であり、Kalnis らのシステム [5] をもとに各ピアにおいて最終的な検索結果に入り得ないクエリ応答を受信したときにクエリ応答を破棄する手法である。

まず、コンテンツ検索を行うクエリ発行ピアは、検索クエリに一意のクエリ ID を与え、検索クエリを自ピアが直接接続している隣接ピアの全てに送信する。検索クエリには以下の情報を含める。

- クエリ ID
- コンテンツの検索条件
- TTL (クエリの生存時間)
- 受信したいクエリ応答の数 k

これと同時に、クエリ発行ピアは自ピアで保持しているコンテンツを対象として検索を行い、検索クエリに含まれる検索条件に合致するコンテンツのうち上位 k 位以内のスコアをもつコンテンツについてクエリ応答を作成し、自ピアの検索結果リスト L に保持する。

検索クエリを受信した各ピアは、クエリを転送するとともに、自ピアで保持しているコンテンツを対象として検索を行う。その手順は、以下の通りである。

- (1) 受信したクエリと同じクエリ ID をもつクエリを過去に受信している場合は、循環クエリとして受信したクエリを破棄する。
- (2) クエリの TTL を1つ減らす。

(3) TTL が 1 以上のときは、クエリ送信元ピアを除く全ての隣接ピアにクエリを転送する。

(4) 自ピアで保持しているコンテンツを対象としてクエリに含まれる検索条件に合致するコンテンツを検索し、これらのうち上位 k 位以内のスコアをもつコンテンツについてクエリ応答を作成する。クエリ応答は以下の情報を含み、1 個のコンテンツについて 1 個作成する。

- クエリ ID
- クエリ応答ピア ID
- コンテンツのスコア
- コンテンツのメタデータ

クエリ応答ピア ID は、コンテンツが見つかったピアを識別するためのもので、同じピアで見つかったコンテンツに関するクエリ応答には同じ ID を与える。コンテンツのメタデータにはタイトルや URL、本文の要約といった、ユーザが閲覧したいコンテンツを選択し取得する上で必要な情報を含む。

(5) 生成したクエリ応答を検索結果リスト L に保存する。 L では、クエリ応答をスコアが高い順に保持する。

(6) L に含まれるクエリ応答をクエリ送信元ピアに返信する。

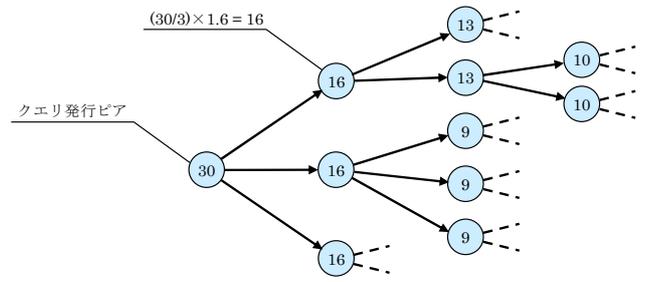
検索クエリ送信先となるピアからクエリ応答を受信したピアでは、受信したクエリ応答に含まれるスコアが検索結果一覧 L において k 位以内に入る場合は、受信したクエリ応答を L に挿入し、 L に含まれるクエリ応答のうち k 位以内から外れたクエリ応答を削除するとともに、このクエリ応答を受信したピアがクエリ発行ピアでなければ受信したクエリ応答を自ピアの検索クエリ送信元ピアに転送する。 L におけるスコアが k 位以内に入らないクエリ応答は破棄する。

この手法を利用すると、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。しかし、各ピアにおける Top- k コンテンツのうち下位のコンテンツが最終結果で k 位以内に入る可能性は低く、クエリ発行ピアにおいて不要なクエリ応答がネットワーク上に大量に流れると考えられる。

3.2 Reduce-k Query 手法

Simple Top- k 手法における問題を解決するため、Reduce- k Query 手法を提案する。この手法では、クエリ発行ピアからのホップ数が大きいピアにおいて、Top- k コンテンツのうち下位のコンテンツがクエリ発行ピアにおいて k 位以内に入る可能性が低いという観点から、各ピアが返信するクエリ応答の個数をクエリ発行ピアからのホップ数に応じて減少させる。

ここで、クエリ発行ピアから i ホップ離れたピアにおけるクエリ応答送信数を k_i とする。また、 $k_0 = k$ である。クエリ発行ピアから i ホップ離れたピアにおいて、検索クエリのパラメータとして指定するクエリ応答送信減少数余裕率 r_m と自ピアの検索クエリ転送先となるピア数 n_{p_i} 、および自ピアにおけるクエリ応答送信数 k_i を利用して、クエリ発行ピアから i ホップ離れたピアのクエリ転送先ピアにおけるクエリ応答送信数 k'_{i+1} を求める。まず k'_{i+1} の仮の値 k'_{i+1} を



($k = 30, r_m = 1.6$ の場合。图中的数字は k_i 。)

図1 各ピアにおいて返信するクエリ応答数

Fig.1 The number of query-replies for each peer.

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

とする。ここで、 $\lfloor \cdot \rfloor$ は $\lfloor \cdot \rfloor$ 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。 $r_m = 1$ とした場合、理論的にはこのピアではクエリ転送先ピアから合計で k_i 個のクエリ応答を受信することとなり、クエリ発行ピアでは k 個程度受信することとなる。しかし、検索クエリの検索条件に合致するコンテンツのうち、最終的に上位 k 個に入るコンテンツが各ピアで同じ数だけ見つかるとは限らないため、 $r_m > 1$ とすることにより再現率を向上させる。ただし、 $k'_{i+1} = 0$ となるピアも存在するなど、式 (1) に従うと各ピアにおける送信すべきクエリ応答数が少なすぎる場合もあるため、クエリ応答送信数 k_{i+1} は、

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

とする。

動作例を図 1 を用いて説明する。まず、クエリ発行ピアでは $k_0 = k$ から式 (2) を利用して各隣接ピアに要求するクエリ応答の数 k_1 を計算し、その値を含めたクエリを各隣接ピアに送信する。同時に、クエリ発行ピア自身で保持しているコンテンツから検索条件に合致するコンテンツを検索する。クエリ発行ピアから検索クエリを受信したピアは、さらに k_2 を計算し、検索クエリを各隣接ピアに転送する。また、スコアが上位 k_1 個のクエリ応答を検索結果一覧 L に保持し、クエリ発行ピアに返信する。また、隣接ピアからクエリ応答を受信すると、受信したクエリ応答のスコアが検索結果 L において上位 k_1 位以内に入る場合、そのクエリ応答を L に保持し、クエリ応答を転送する。以上の手順を各ピアで繰り返すことにより、各ピアでは少なくとも各ピアで受信したクエリ応答および自ピアにおいて検索したコンテンツから生成したクエリ応答のうち、スコアが上位 k_i 個のクエリ応答について上位のピアにクエリ応答を送信することとなる。

3.3 Delayed Reduce-k Query 手法

Reduce- k Query 手法と比べてさらにクエリ応答の送信数を減らすため、各ピアでクエリ応答の送信を一定時間遅延させ、その間に上位 k 位以内から外れたコンテンツについてはクエリ

応答を送信しないことによりクエリ応答の送信数を減少させる Delayed Reduce-k Query 手法を提案する。

Delayed Reduce-k Query 手法では、転送した検索クエリの TTL が T であるピアにおいて、最後のクエリ応答受信後、またはピア内におけるコンテンツ検索終了後指定した遅延時間 t_{1T} だけ待ってクエリ応答を送信する。 t_{1T} は、

$$t_{1T} = t_0 + t_1 T \quad (3)$$

とする。ただし、 t_0 および t_1 の値は検索クエリにおいて指定する。

ところで、 L に含まれるクエリ応答のうちスコアが特に上位のものについては、クエリ発行ピアにおいて最終的な検索結果に含まれる可能性が高いと考えられる。そこで、検索結果が得られたもしくはクエリ応答を受信した時点で上位 N_{s_i} 位以内に入るコンテンツについて、クエリ応答を直ちに送信する。

各ピアにおけるクエリ応答即時送信数 N_{s_i} の計算方法として、最大値法と加算法の 2 種類を提案する。最大値法では、

$$N_{s_i} = \max(\lfloor k_i r_s \rfloor, n_s) \quad (4)$$

とし、加算法では、

$$N_{s_i} = \lfloor k_i r_s + n_s \rfloor \quad (5)$$

とする。クエリ応答即時送信率 r_s 、およびクエリ応答即時送信数 n_s はクエリ送信時にパラメータとして指定する。 k_i はそのピアにおけるクエリ応答送信数である。

また、各ピアにおいて全てのクエリ応答の受信を完了したことを認識できるように、各ピアは送信する必要があるクエリ応答を全て送信後にクエリ応答終了メッセージをクエリ応答に加えて送信する。各ピアでは、全ての検索クエリ送信先ピアからクエリ応答終了メッセージを受け取った場合には、送信していないクエリ応答を直ちに送信し、最後にクエリ応答終了メッセージを送信する。

4. 性能評価

本研究で提案した手法について、シミュレータによる性能評価を行った。

4.1 想定環境

本稿では、TTL を 5 とした P2P ネットワークを想定している。

4.1.1 ネットワークのトポロジ

本稿におけるシミュレータでは、まずネットワーク上の総ピア数である n_p 個のピアを図 2 の太線のようにリング状に接続する。つまり、各ピアに $0, 1, \dots, n_p - 1$ の番号を付けたとすると、 i 番と $i + 1$ 番 ($0 \leq i < n_p - 1$)、 0 番と $(n_p - 1)$ 番のピアを接続する。

続いて、各ピアは図 2 の細線のようにピアを 1 個ランダムに選択して接続する。ここでは、ランダムに選択したピアが自分であった場合、およびすでに接続されたピアであった場合には接続しないため、各ピアの隣接ピア数の平均は、約 3.95 となる。本研究では、シミュレーション実行中にピアの参加や退出、およびネットワーク構成の変更は行われなかった。

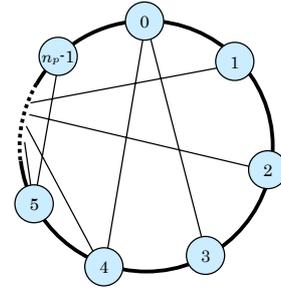


図 2 シミュレーションにおけるピアの接続状態

Fig. 2 The connection model among peers for simulation.

4.1.2 コンテンツの配布

本稿におけるシミュレータでは、簡単のためコンテンツを 31 ビットの整数とし、この整数の範囲を検索条件として指定することにより検索を行う。また、Top-k クエリにおいて必要となるスコアもこの整数とする。この上で、P2P ネットワークにおけるコンテンツ分布が Zipf 分布に基づくことが知られているため [6]、シミュレータでは Zipf 分布に基づいてコンテンツを配布する。

各ピアでは、まずピアあたりのコンテンツ数 N_p に応じてコンテンツストレージ配列を生成する。そして、シミュレータはネットワーク上に存在するコンテンツ数として指定した $N = 100N_p$ 個のコンテンツを、乱数により生成する。コンテンツ生成と同時に、各々のコンテンツをランダムに選択した 1 個のピアに配布することにより、ここで生成した各コンテンツが最低 1 個は P2P ネットワーク中に存在するようにする。

続いて、コンテンツを各ピアに Zipf 分布を利用して配布する。まず、確率変数の和について

$$\int_{0.5}^{N_c+0.5} Kx^\alpha dx = 1 \quad (6)$$

を満たすように K を決定するため、

$$K = \frac{\alpha + 1}{(N + 0.5)^{\alpha+1}} - 0.5^{\alpha+1} \quad (7)$$

とする。ただし、 $\alpha = -0.9$ とする。続いて、用意した N 個のコンテンツのうち n 番目のコンテンツを各ピアのストレージに順次配布する。この n は

$$\int_{0.5}^{n+0.5} Kx^\alpha dx = R \quad (8)$$

(ただし、 R は 0 以上 1 未満の乱数)

を満たすように求めると、

$$n = \left\lceil \alpha+1 \sqrt[\alpha+1]{\frac{R(\alpha+1)}{K} + \left(\frac{1}{2}\right)^{\alpha+1}} + 0.5 \right\rceil \quad (9)$$

となる。このようにして、すべてのピアのストレージが満杯になるまでコンテンツを配布する。また、コンテンツ配布時には、ピアに同一のコンテンツが複数保存されないよう考慮する。

4.1.3 クエリの発行

シミュレータでは、クエリ発行イベントが発生すると、クエ

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.

繰り返し回数		5
シミュレーション時間		1000
ネットワーク中のピア数	n_p	10000
ネットワーク中のコンテンツ	N_c	1000000
ピア内コンテンツ数	N_p	10000
ピアあたりクエリ発生間隔	t_{iq}	1000
TTL	T	5
ヒット率	r_h	0.1%
クエリ応答要求数	k	100
クエリ応答送信減少数余裕率	r_m	1.5
クエリ応答即時送信数	N_{s_i}	$[0.1k_i]$
送信待ち時間	t_{l_T}	$0.05 + 0.002T$

り発行ピアをランダムに選択し検索クエリを発行する．クエリ発行イベントの平均発生間隔 t_i は、ネットワーク上のピア数を n_p 、各ピアにおけるクエリ発行間隔を t_{iq} として

$$t_i = \frac{t_{iq}}{n_p} \quad (10)$$

とし、クエリ発行イベントの発生間隔を t_i を平均とする正規乱数とした．これにより、各ピアにおけるクエリ発行間隔は平均 t_{iq} 秒となる．

4.1.2 項で示したようにコンテンツを 31 ビットの整数としていることを利用して、クエリの検索条件は 31 ビットの整数の範囲（開始値 s_s と終了値 s_e ）とした．開始値 s_s は 0 と $2^{31} - 1$ の間の乱数とし、終了値 s_e は、シミュレーションのパラメータとして指定するヒット率を r_h とし、

$$s_e = (s_s + 2^{31} \times r_h) \bmod 2^{31} \quad (11)$$

により求める．ただし、 $A \bmod B$ は、 A を B で除算したときの剰余である． $s_s \leq s_e$ のときは、 $s_s \leq c \leq s_e$ となる全ての c が、 $s_e < s_s$ のときは、 $c \leq s_e, s_s \leq c$ となる全ての c が検索条件に合致するコンテンツとなる．その c の値が小さいコンテンツをスコアの低いコンテンツとする．

4.1.4 通信速度

各ピアの通信速度は上り・下りともに 512000 [ビット/秒] とし、メッセージの送信遅延時間を、 $0.001 + 0.01 \times \text{ExpRnd}$ [秒] とした．ただし、ExpRnd は平均 1 の指数乱数とする．

メッセージ長は、クエリ応答にコンテンツの 256 バイト程度の要約を含むことを想定し、検索クエリメッセージを 140 バイト、クエリ応答メッセージを 640 バイトと設定した．

4.2 k とクエリ応答送信減少数余裕率の影響

まず、上位 30 個のクエリ応答を得る Delayed Reduce-k Query 手法のシミュレーションを行い、クエリ応答要求数 k およびクエリ応答送信減少数余裕率 r_m がクエリ応答送信数および再現率に与える影響を評価した． k は 30, 50, 100 とし、クエリ応答送信減少数余裕率 r_m を 1.1 ~ 6 の間で変化させた．これ以外のパラメータは表 1 に示した通りである．

その結果得られたクエリ応答送信数、およびクエリ到達範囲内における上位 30 個のコンテンツに関するクエリ応答が得ら

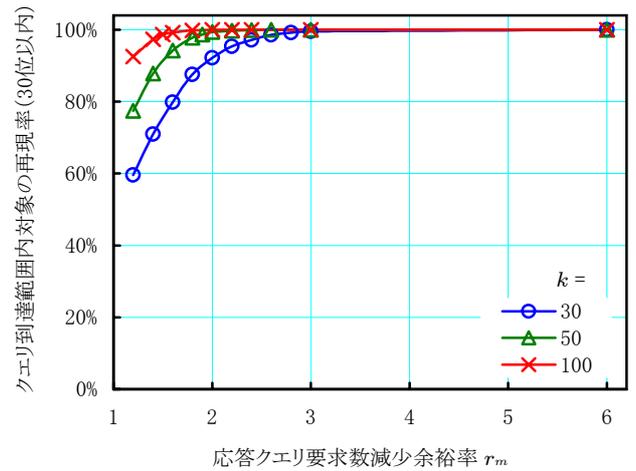
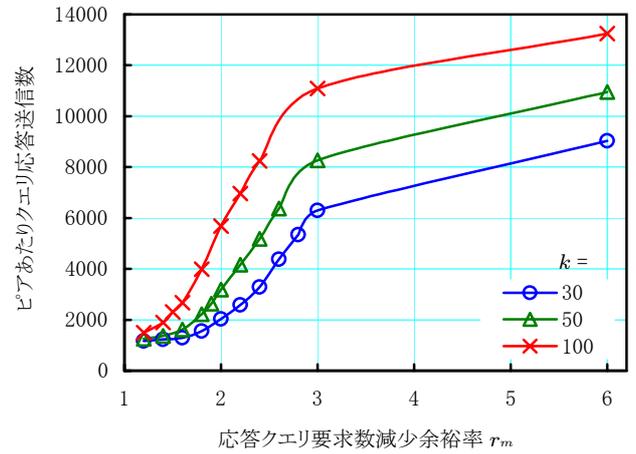


図 3 k と r_m の変化に対するピアあたりのクエリ応答送信数と再現率

Fig. 3 k and r_m vs. query-replies and recall.

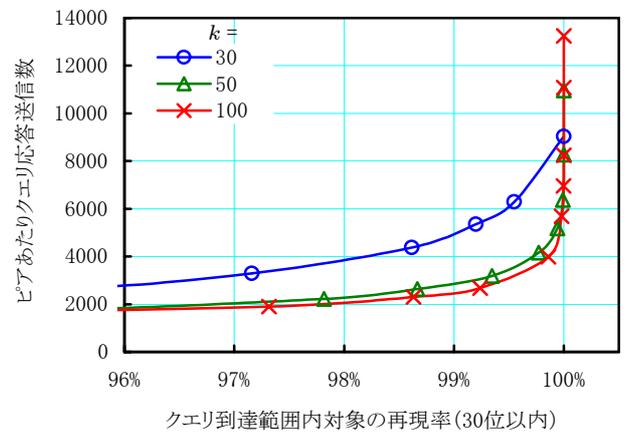


図 4 再現率の変化に対するピアあたりクエリ応答の送信数

Fig. 4 Recall vs. query-replies.

れたかどうかを示す再現率を図 3 に示す．ネットワーク上の全コンテンツに対する再現率は、図 3 に示した値の 85% 程度である．クエリ応答要求数 k が大きいほど、また r_m が大きいほどクエリ応答送信数は大きくなる．また、再現率は r_m が増加するにつれて増加する．

図 4 では再現率とピアあたりのクエリ応答送信数の関係を示す．このグラフを見ると、 k が大きいほど同じ再現率を得ると

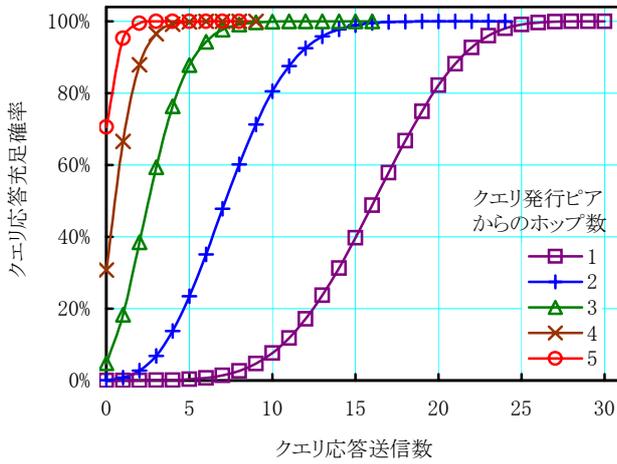


図5 クエリ応答充足確率

Fig. 5 Rate of satisfying query-replies.

表2 クエリ応答必要送信数と実際のクエリ応答送信数

クエリ発行ピアからのホップ数	1	2	3	4	5	
クエリ応答必要送信数 (充足確率 99%)	25	16	8	4	2	
実際のクエリ 応答送信数	$k = 30, r_m = 2.6$	20	17	15	13	11
	$k = 50, r_m = 1.9$	24	15	10	6	4
	$k = 100, r_m = 1.5$	38	19	10	5	3

(実際のクエリ応答送信数は、全ピアの隣接ピア数が4の場合)

きのクエリ応答送信数が減少していることから、パラメータの設定上は k を大きくすると効率的な検索が可能であると言える。98%以上の再現率を確保するために必要な値は、 $k = 30$ のときは $r_m \geq 2.6$ 、 $k = 50$ のときは $r_m \geq 1.9$ 、 $k = 100$ のときは $r_m \geq 1.5$ である。

続いて、 $k = 30$ として Simple Top-k 手法を用いた場合のシミュレーション結果を利用し、各ピアが送信したクエリ応答のうち何個が最終的な検索結果に含まれたかを図5に示す。縦軸は、クエリ発行ピアにおいて最終の検索結果を求める際に横軸で示した数を超えてクエリ応答を送信する必要のない確率であるクエリ充足確率を示す。図5より、クエリ充足確率を99%にするために送信する必要のあるクエリ応答の数は、表2に示す通りとなる。

この値と、表2に示した隣接ピア数が全ピアで4の場合における実際のクエリ応答送信数を比較すると、同じ再現率を得る場合においては、 $k = 30$ の場合と比較して $k = 50$ や $k = 100$ の場合のほうが実際のクエリ応答送信数はクエリ応答必要送信数に近い値となっている。また、 $k = 100$ の場合は $k = 50$ の場合に比べてクエリ発行ピアからのホップ数が4、および5の場合において実際のクエリ応答送信数が1減少しただけであり、ホップ数が1および2の場合においては大幅に増加しているにもかかわらずピアあたりのクエリ応答送信数は約12.5%減少している。このことから、クエリ発行ピアから離れたピアにおけるクエリ応答送信数を減少させると、クエリ発行ピアに近いピアにおけるクエリ応答送信数を増加させたとしても、ネットワーク全体においてピアあたりのクエリ応答送信数は減少することがわかる。

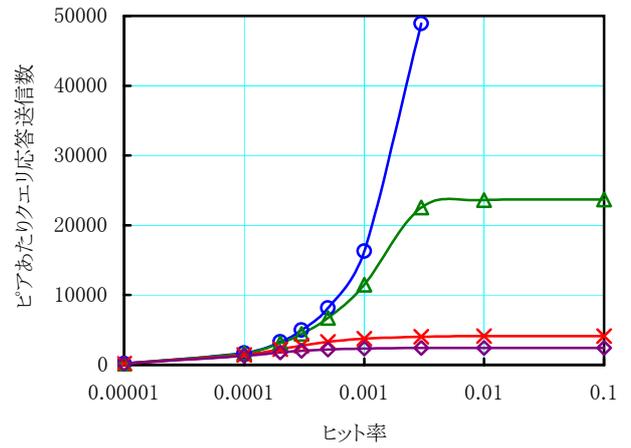


図6 ヒット率の変化に対するピアあたりクエリ応答送信数

Fig. 6 Hit-rate vs. query-replies.

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、応答送信数と再現率の変化について、Top-kを指定しない全応答手法、Simple Top-k手法、Reduce-k Query手法、およびDelayed Reduce-k Query手法の間で比較する。Simple Top-k手法では、 $k = 30$ とした。Reduce-k Query手法、およびDelayed Reduce-k Query手法においては、4.2節における実験結果より $k = 100, r_m = 1.5$ とした。また、Delayed Reduce-k Query手法では、即時送信率 $r_s = 0.1$ 、クエリ応答遅延 $t_{l_T} = 0.01 + 0.002T$ とした。ここに示さないパラメータは表1に示した通りとした。

再現率は、全応答およびSimple Top-k手法の場合で85.1%程度であるのに対して、Reduce-k Query手法の場合で平均84.6%、Delayed Reduce-k Query手法の場合でも平均84.3%であることから、十分な値であると言える。

ピアあたりクエリ応答送信数の平均値を図6に示す。ヒット率が0.003のとき、Simple Top-k手法では、全応答手法の場合の46%に抑制できた。全応答手法の場合はヒット率に比例して応答クエリ送信数が増加するため、ヒット率の高いクエリが発行された場合には、クエリ応答のトラフィックが極端に増大する。しかし、Simple Top-k手法、および本稿における提案手法を用いると、ヒット率が高い場合においてもクエリ応答送信数は一定の値以上には増加しない。このことから、Top-kクエリによるクエリ応答送信数の抑制は有用であるといえる。

しかし、Simple Top-k手法ではヒット率が0.1のとき、各ピアは1秒あたり平均で23.7個のクエリ応答を送信している。一方、Reduce-k Query手法ではヒット率が0.1のときでも各ピアにおける1秒あたりのクエリ応答送信数は平均で4.1個と、Simple Top-k手法の場合に比べて83%の減少を実現した。さらに、Delayed Reduce-k Query手法ではヒット率が0.1の場合で1秒あたり2.4個と、Reduce-k Query手法と比べて41%の削減を実現した。

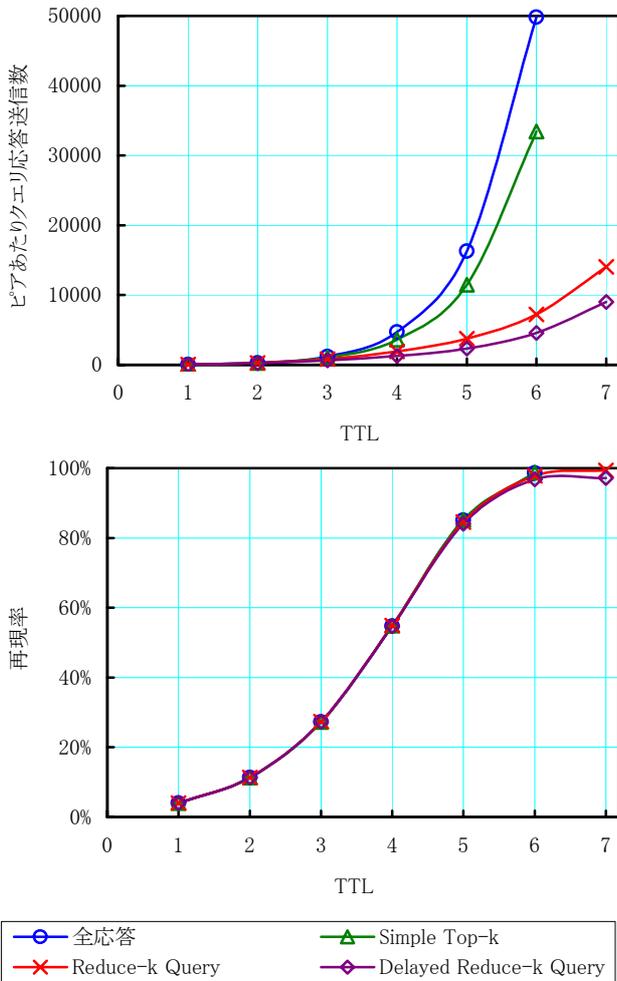


図7 TTL の変化に対するピアあたりクエリ応答送信数および再現率
Fig. 7 TTL vs. query-replies and recall.

4.4 TTL の影響

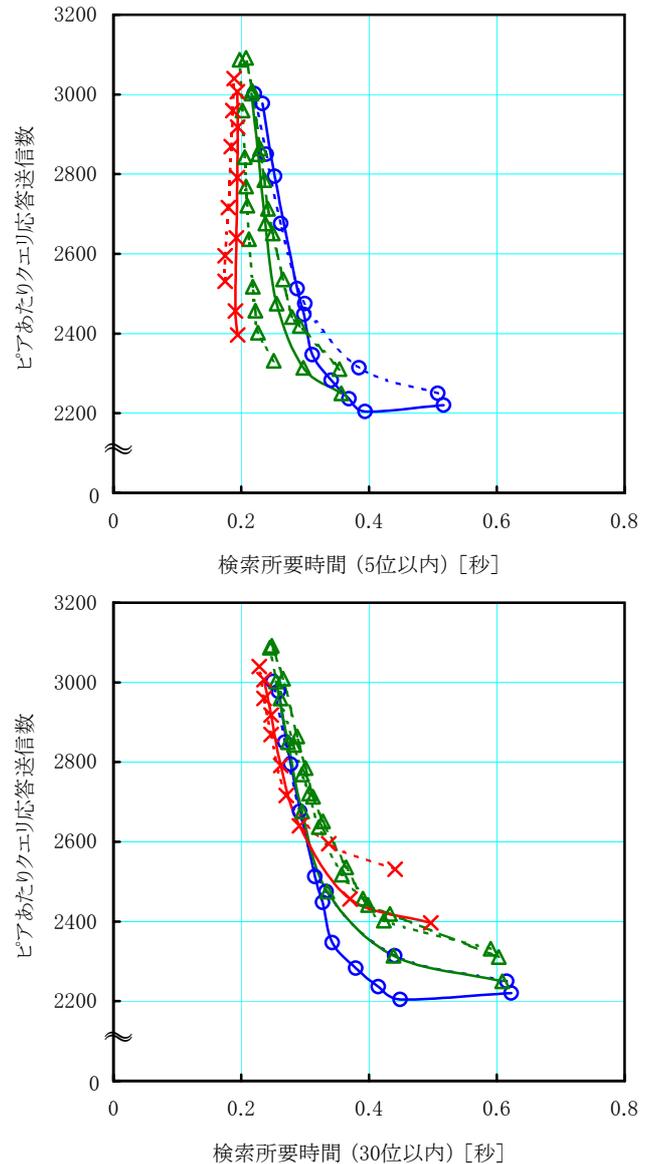
さらに、TTL の値がクエリ応答送信数および再現率に与える影響を評価した。4.3 節で示した 4 つの手法について、ヒット率 $r_h = 0.001$ とし、TTL を 1 から 7 まで変化させた。その結果のピアあたりクエリ応答数およびネットワーク中に存在する全コンテンツを対象とした再現率を図 7 に示す。

ピアあたりクエリ応答送信数は TTL が増加すると増加する。しかし、全応答手法や Simple Top-k 手法では TTL が 1 増加するごとに 3 倍程度に増加するのに対し、Reduce-k Query 手法、および Delayed Reduce-k Query 手法においては TTL が 1 増加するごとの増加が 2 倍程度に抑えられている。

再現率は、TTL を増加させると増加するが、手法間の差はほとんどない。また、TTL が 6 と 7 の場合での差も小さい。このことから、Reduce-k Query 手法や Delayed Reduce-k Query 手法を用い、適切な TTL を設定することにより、Simple Top-k 手法を用いた場合に比べて高い再現率をより低いネットワーク負荷で実現可能である。

4.5 クエリ応答送信遅延時間の影響

Delayed Reduce-k Query 手法において、送信待ち時間および即時送信数の設定とネットワークへの負荷および検索所要時間



[凡例]

r_s	$n_s = 0$	$n_s = 1$	
		最大値法	加算法
0	—○—	—○—	---○---
0.1	—△—	---△---	---△---
0.3	—×—	---	---×---

図8 検索所要時間の変化に対するピアあたりクエリ応答送信数
Fig. 8 Turn around time vs. query-replies.

の関係を評価した。クエリ応答即時送信数 N_i は式 (4), (5) により求める。 $r_s = 0, 0.1, 0.3$ とし、以下の 3 つの手法について評価した。

- $n_s = 0$
- 最大値法 ($n_s = 1$)
- 加算法 ($n_s = 1$)

ただし、最大値法は $r_s = 0.1$ の場合のみ評価している。また、送信待ち時間は、 $0.001 \sim 0.1$ の値に $0.002T$ を加算したものとした。ただし、 T は各ピアにおける TTL である。上記以外の

パラメータは表 1 の値を採用した。

シミュレーション結果として、5 位以上、および 30 位以上のコンテンツに関するクエリ応答が得られるまでの検索所要時間を横軸、ピアあたりのクエリ応答送信数を縦軸にとったグラフを図 8 に示す。左下にいくほど、クエリ応答の送信数が少なく、また所要時間も短くなるため良い結果であると言える。

その結果、上位 5 個のコンテンツに関するクエリ応答が得られるまでの時間は、同程度のクエリ応答送信数の場合、即時応答数を大きく設定する方が短くなっていることがわかる。しかし、上位 30 個となると、即時応答数を小さく設定する方が短くなる。

また、 $n_s = 0$ の場合、最大値法、および加算法の 3 つの手法間で比較すると、多くの場合で $n_s = 0$ とした場合が性能は最も良くなっていることがわかる。以上の原因として、定数で設定する場合には TTL が 0 や 1 のピアにおいて 1 個でも即時送信数を増やすと結果が大きくなっていることが挙げられる。

5. 提案手法の拡張

実環境において提案方式を運用する場合、現状では様々な問題が発生する。本章では、それらの問題に対応するための、提案方式の拡張について考察する。

5.1 確実な Top-k コンテンツの取得

提案手法では、平均で検索クエリ到達範囲内の 98% 程度の Top-k コンテンツについての応答クエリを受信できている。しかし、実際には 90% 未満の応答クエリしか受信できていないケースが 1.8% 存在する。検索クエリ発行ピアから離れた特定のピアにキーワードに関連するコンテンツが大量に配置されている場合、それらのコンテンツに関する応答クエリが少数しか返信されないため、再現率が大きく悪化している。

より確実に Top-k コンテンツに関するクエリを取得するための手法として、クエリ発行ピアにおいてコンテンツに関する応答クエリを確実に受信できていないことを検知し、その場合に特定のピアに再度クエリ応答を要求する手法が考えられる。この手法については今後詳細を検討し、評価を行う予定である。

5.2 他の検索効率化技術の利用

現在の提案手法では、コンテンツ検索時に検索結果のキャッシュを利用してない。過去の検索結果に関するキャッシュを利用すると、クエリのトラフィックを抑制が可能である。さらに、絞込み検索を行うクエリについてもキャッシュに含まれる送信元ピアを参照することによりクエリの転送先を制限できると考えられる。

また、ウェブコンテンツ検索では各ユーザが過去に行ったキーワードに近いキーワードを利用して検索を行うことが多い。そのため、求めるクエリ応答を返信するピアがそれらのクエリの間で同じような組合せとなる可能性が高い。そこで、これらのピアに優先的に検索クエリを送信することにより、検索クエリやクエリ応答によるトラフィックを抑制できると考えられる。

6. 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、Simple Top-k 手法と比べてヒット率が 0.1% の場合で各ピアにおけるクエリ応答の送信数を 73% 削減できることを示した。

ここで、コンテンツ同士の関連や高い頻度で検索キーワードとして利用されるキーワード、あるいはユーザが絞込み検索を行う状況などを考慮することにより、より効率の良い問い合わせ処理が可能になると考えられる。さらに、実際の P2P ネットワークでは、ピアの参加・退出も頻繁に発生する。

今後は、上記であげたような性質をシミュレータに盛り込み、より精度の高いシミュレーションを行う。さらに、実際に提案手法を用いた P2P ウェブコンテンツ検索システムを実装し、多数のユーザによる利用評価を行う予定である。

謝辞 本研究は、科学研究費補助金(基盤研究(B)(2))「大規模な仮想空間システムを構築する放送型サイバースペースに関する研究」(プロジェクト番号:15300033) および、文部科学省 21 世紀 COE プログラム(研究拠点形成費補助金)の研究助成によるものである。ここに記して謝意を表す。

文 献

- [1] W.-T. Balke, W. Nejdl, W. Siberski, and U. Thaden, "Progressive Distributed Top-k Retrieval in Peer-to-Peer Networks," Technical Report of Hannover University, July 2004.
- [2] "Gnutella," <http://www.gnutella.com/>.
- [3] "Google," <http://www.google.com/>.
- [4] I. Iarke, O. Sandberg, B. Wiley, and T.W. Hong, "Freenet: A Distributed Anonymous Information Storage and Retrieval System," *Proc. ICSI Workshop on Design Issues in Anonymity and Unobservability*, pp.46-66, July 2000.
- [5] P. Kalnis, W.S. Ng, B.C. Ooi, and K.-L. Tan, "Answering Similarity Queries in Peer-to-Peer Networks," *Proc. International World Wide Web Conference*, pp.482-483, May 2004.
- [6] 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, <http://www.itrc.net/report/meet14/NGN/kamei.pdf>, Nov. 2003.
- [7] 中村聡史, 塚本昌彦, 西尾章治郎, "コンテンツ流通制御を考慮したウェブコンテンツ共有システムの実現," *情処学論*, vol.45, no.1, pp.74-83, Jan. 2004.
- [8] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable Content-Addressable Network," *Proc. SIGCOMM'01*, pp. 161-172, San Diego, California, United States, Aug. 2001.
- [9] A. Rowstron, and P. Druschel, "Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems," *Proc. Middleware 2001*, pp. 329-350, Heidelberg, Germany, Nov. 2001.
- [10] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Application," *Proc. SIGCOMM'01*, pp. 149-160, San Diego, California, United States, Aug. 2001.
- [11] C. Tang, Z. Xu, and M. Mahalingam, "pSearch: Information Retrieval in Structured Overlays", *ACM SIGCOMM Computer Communications Review*, vol.33, no.1, pp.89-94, Jan. 2003.
- [12] "Yahoo! Japan," <http://www.yahoo.co.jp/>.
- [13] B.Y. Zhao, J. Kubiatowicz, and A.D. Joseph, "Tapestry: An Infrastructure for Wide-area Fault-tolerant Location and Routing," U. C. Berkeley Technical Report USB/CSD-01-1141, 2001.