

## 次世代 P2P コンテンツ交換システム - コンテンツの整合性維持と高度な情報検索の実現 -

竹辺 靖昭<sup>1</sup> 美馬 秀樹<sup>2</sup> 苫米地 英人<sup>1</sup>

### 概要

現在普及している P2P コンテンツ交換システムの多くでは、コンテンツの更新が考慮されておらず、検索機能も単純なものしか提供されていない。このため、テキストを含むコンテンツや、頻繁に更新されるコンテンツを対象とするのに適しているとは言い難い。

我々が開発中の P2P コンテンツ交換システムでは、中央サーバにおいてコンテンツのダウンロード先へのリンクを保持するデータ構造を採用し、それらのリンクを適切に管理することにより、更新されるコンテンツの改ざん防止および整合性の維持を行う。また、検索においては、クライアントでテキストの形態素解析を行い、中央サーバに転置インデックスを登録することにより分散処理による全文検索を実現する。本手法により、コンテンツ更新の頻繁な P2P 環境においても、効率的かつ速やかに最新コンテンツへのアクセスが可能となる。

本稿では、これらの手法を紹介するとともに、全文検索の負荷分散手法の予備的な実験結果について述べる。

## A Next-Generation P2P Contents Sharing System - Implementing Content Consistency Maintenance and Full-Text Search -

Yasuaki Takebe<sup>1</sup> Hideki Mima<sup>2</sup> Hideto Tomabechi<sup>1</sup>

### Abstract

One problem with current P2P (peer-to-peer) systems is that the consistency between copied contents is not guaranteed. Also, the weakness of full-text search capability in most of the popular P2P systems hinders the scalability of P2P based content sharing systems. We proposed a new generation P2P content sharing systems in which the consistency of contents in the network is maintained after updates or modifications to contents are made. Links are maintained to downloaded contents from a server and updates and modifications to the contents can be immediately detected and hence reflected on the future P2P downloads. Natural language processing including morphological analysis is performed by the P2P clients distributedly and update of inverted index on the server is concurrently conducted to provide efficient full-text search. The scheme and a preliminary experimental result are mentioned.

### 1. はじめに

P2P コンテンツ交換システムは、少ない資源で大量のコンテンツを流通させられるという優れた特徴を持っている。これを生かすことにより、P2P コンテンツ交換システムは、今後のコンテンツ交換手法の主流となることが期待できる。

しかし、実際に運用されている P2P コンテンツ交換システムでは、交換されるファイルが著作権を侵害した動画や音楽などで占められ、サービスの運用者の法的責任が問われるケースも目につく。

我々は、この問題が起こる技術的な要因として、以下のことがあると考えている。

まず、現在普及している P2P コンテンツ交換システムでは、各クライアント間で行われるコンテンツの交換を追跡するための十分な機能が備わっていないということが挙げられる。このため、サービスの運用者は、違法なコンテンツ交換を行うユーザを追跡したり、コンテンツの改ざんを確認したりすることができない。

もう一点の要因としては、現在普及している P2P コンテンツ交換システムが、ファイル名等をキーとした単純な検索機能しか提供しておらず、テキストを含むコンテンツや頻繁に更新されるコンテンツを交換するには適していないということが挙げられる。このため、P2P コンテンツ交換システムは内容が頻繁には更新されない動画や音楽のファイルの交換にしか用いられないことが多い。このため、テキストを含むコンテンツの交換にも適用でき、コンテンツの更新に際しても最新のコンテンツが検索されるような情報検索機能の実現が期待されている。

<sup>1</sup> コグニティブリサーチラボ Cognitive Research Lab.  
<http://www.crl.co.jp/> <sup>2</sup> 東京大学大学院工学系研究科 School of Engineering, University of Tokyo

こうした技術的な問題を解決するため、我々は、P2P 環境において、コンテンツの改ざんを防ぎ、実際にユーザが行ったコンテンツの交換とシステムの運用者が把握するコンテンツの交換との整合性を維持(以下ではこれを単にコンテンツ交換の整合性の維持という)する手法および情報検索の手法を開発し、成果を標準化することを目標としている[10]。

本研究で開発中のコンテンツ交換システムでは、コンテンツの更新を考慮し、コンテンツの版ごとに電子署名を保持する。コンテンツ交換の整合性を維持するために、非対称暗号鍵を用いたダウンロードプロトコルを用いる。また、コンテンツが更新されても常に最新のバージョンを取得できるように、コンテンツのオリジナルへのリンクおよびコンテンツのダウンロード先へのリンクを保持するデータ構造を用い、それらのリンクを中央サーバで管理する。

全文検索の実現のために、各クライアントにおいてテキストの形態素解析、要約処理を行い、転置インデックスの作成に必要な情報を生成し、コンテンツの更新時に中央サーバに登録する。また、検索結果をクライアントにキャッシュすることにより、中央サーバにかかる負荷を軽減する。これにより、検索サーバでの自然言語処理の負荷を各クライアントに分散し、コンテンツの更新を速やかに反映した全文検索が可能となる。

本稿では、我々が開発中の P2P コンテンツ交換システムの概要と、そのシステムで使用しているコンテンツ交換の整合性維持手法および全文検索手法を紹介する。また、検索での負荷分散手法に関する予備的な実験結果を報告する。

論文の構成は以下の通りである。第 2 節では関連研究について述べる。第 3 節では、現在開発中の P2P コンテンツ交換システムの概要について説明する。第 4 節では、コンテンツ交換の整合性を保つための手法について、第 5 節では、全文検索を実現する手法について説明する。最後にまとめと今後の課題について述べる。

## 2. 関連研究

本研究と関連する研究としては、コンテンツ交換の整合性についての研究と P2P での検索についての研究がある。

なお、本論文では、中央サーバを用いない方式の純粋な P2P システムだけでなく、Napster のように中央サーバを用いたハイブリッド型の P2P システムも含めて P2P と呼ぶ。これは、コンテンツ交換を応用として考えた場合、ハイブリッド型 P2P であっても、サービスを提供する側の帯域が少なく済むという重要な利点があるためである。

### 2.1. コンテンツ交換の整合性

P2P コンテンツ交換システムでは、コンテンツが各クライアントに保持されているため、何も対策をとらなければ悪意のあるクライアントにおいてコンテンツが改ざんできてしまう。

Napster[4]では、コンテンツの公開者が、コンテンツを中央のサーバに公開する際に、コンテンツの MD5 ハッシュ値を登録するプロトコルを用いている。また、Freenet[2]では、コンテンツのハッシュ値そのものをキーとして用いることにより、コンテンツの改ざんを防いでいる。

これらの手法により、更新が行われないと考えられる動画や音楽等のコンテンツの改ざんを防ぐことは可能だが、更新が行われるコンテンツの場合、更新する度にハッシュ値が変わるため、別のコンテンツとして扱わなければならない。Freenet では、このように更新されるコンテンツを扱うために、コンテンツのハッシュ値を内容とする間接ファイルというものを導入し、2 段階でコンテンツを取得することによってコンテンツの更新を実現している。このように、頻繁に更新が行われるコンテンツを交換するためには、コンテンツそのものの ID と、その版のハッシュ値を関連付ける仕組みが必要になると考えられる。

P2P コンテンツ交換システムにおけるもう一つの問題として、クライアント間でコンテンツを交換しても、システムの運用者がそれを追跡するのが困難であるということがあげられる。

本研究と同様に中央サーバを用いたシステムである Napster では、コンテンツのダウンロード開始時に中央サーバに対して要求を行うが、その後中央サーバはコンテンツの交換には関与しない。このプロトコルでは、ダウンロード元のクライアントが偽のデータを送っても、中央サーバからは正常にコンテンツがダウンロードできたのと同じ情報しか把握できないことが問題となる。

### 2.2. P2P での検索

P2P コンテンツ交換システムでのコンテンツの検索手法として以前から用いられているものに、中央サーバを用いるもの[4]と検索要求をブロードキャストするもの[6]がある。

中央サーバを用いる手法の問題点としては、検索サーバのスケラビリティがないことや、single point of failure による脆弱性が従来から指摘されている。また、検索要求をブロードキャストする手法の問題点としては、ノード数の増加とともに、検索要求によって帯域が大量に消費されてしまうということが挙げられる。実際に普及しているシステムでは、発見的手法によって検索要求を出す範囲を制限する工夫が行われているが、このためにネットワーク内に存在するコンテンツのすべてを検索できることが保障されない可能性が出てきてしまう。

これらの問題点を改良するものとして、分散ハッシュ

テーブル(DHT)を用いた手法が数多く提案されている[5][7]。これらの手法は非常にスケラブルであり、完全一致によるキーの検索が  $O(\log n)$  や  $O(n^2)$  ホップで可能であるという特徴を持っている。

Reynolds と Vahdat は、DHT を用いて転置インデックスを各ノードに分散して配置することによる全文検索の手法を提案している[1]。この手法では、ハッシュテーブルのキーがキーワードに、値がそのキーワードを含む文書の集合にそれぞれ対応している。文書を公開する際にクライアントは、文書中のキーワードに対応するノードに対して文書が更新されたことを通知し、転置インデックスを更新する。Reynolds らの手法は、全文検索の負荷が各ノードに分散される、転置インデックスの更新が速やかに行えることが期待できる、といった利点がある。

一方で、Reynolds らの手法には AND 検索の際にノード間で大量の検索結果が転送されるという問題点がある。Li らは P2P 全文検索に必要な帯域の大きさを見積もり、実現の困難さを指摘している[8]。

また、P2P コンテンツ交換システムにおいては、各ノードはユーザの PC であるため、可用性が低いという問題がある。このため、P2P コンテンツ交換システムにおいて各ノードにインデックスを格納すると、多重化が必要となるために Li らの見積もり以上の帯域が必要になってしまう。

以上により、現状の P2P 環境においては、インデックスを集中的に管理するための中央サーバを用いるシステム構成の方が、より現実性が高いと考えられる。

### 3. システムの構成

本研究で開発中のシステムの構成を図1に示す。本システムでは、先に述べた理由により、コンテンツの検索機能を提供するために中央サーバを用いた方式を採用する。

中央サーバでは、サービスを利用する各クライアントの公開鍵が保持される。クライアントは、中央サーバに要求することにより、サービスを利用している他のクライアントの公開鍵を取得することができる。また、中央サーバ自身も秘密鍵を保持しており、公開鍵を各クライアントに公開している。

クライアントにはユニークな ID が割り当てられている。クライアントはサービスを利用するときには、中央サーバに対して自身の IP アドレスを通知する。他のクライアントは、クライアントの ID を元に中央サーバに問い合わせることにより、そのクライアントの IP アドレスを取得することができる。

この他に、中央サーバは、コンテンツ交換の整合性維持のための機能と全文検索機能を提供する。これらについては以降の節で説明する。

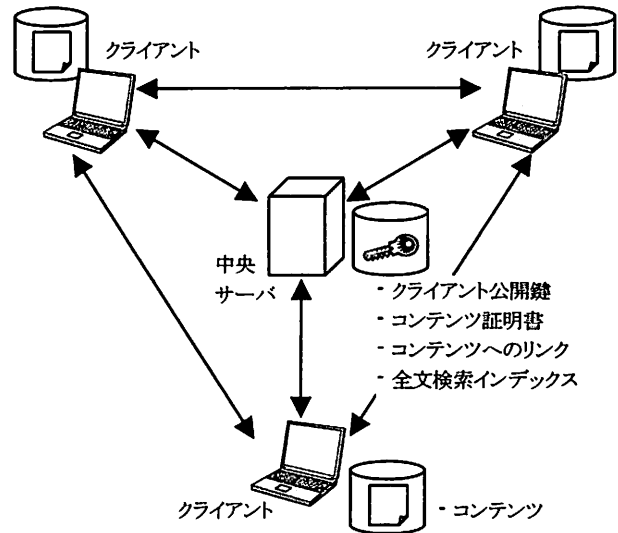


図1. 本システムの構成

## 4. コンテンツ交換の整合性維持

### 4.1. コンテンツ管理のデータ構造

本システムでは、コンテンツの公開者は、自分の秘密鍵を用いてコンテンツに電子署名を行い、コンテンツのユニークな ID とともに検索サーバに登録する。テキスト文書の場合には、クライアントにおいて形態素解析や要約処理を行い、検索キーワードを生成する。

中央サーバは、コンテンツの ID と版に対応した電子署名とを管理する。クライアントは、コンテンツの ID と版を元に中央サーバに問い合わせることにより、コンテンツの版に対応した電子署名を得ることができる。これにより、クライアントは、コンテンツが公開されてから改ざんされていないことを確認することができる。

第5節で説明するように、検索サーバより返却されるコンテンツ ID と電子署名とのペアは、他のクライアントで検索結果をキャッシュすることができる。検索サーバは、この結果が改ざんされていないことを証明するために電子署名を行う。

本システムでは、コンテンツが更新されたときに、そのコンテンツの古い版をダウンロードしたクライアントは、中央サーバに問い合わせることにより、同じ ID で最新のコンテンツを検索することができる。この手法では、同じ ID では常に最新の版しか取得できないという問題もある。Freenet のようにコンテンツのハッシュ値と間接ファイルを組み合わせれば、直接古いバージョンのハッシュ値を取得することも可能である場合がある。しかし、Freenet においても、古いバージョンは要求がなければ削除される場合もあり、常に取得できることは保証していない。本手法では、最新の版のみを対象としてプロトコルを規定し、常に取得できることを保証している。

また、本システムでは、コンテンツのダウンロード要求が特定のクライアントに集中しないように、本システムでは、最新の版をダウンロードしたクライアントのリストを管理し、ダウンロード元を分散させる手法を用いている。

この手法では、コンテンツの最新版をダウンロードしたノードの ID が、そのコンテンツ ID に対応するリストに追加される。検索サーバに対してクライアントがコンテンツのダウンロードを要求した場合には、そのリスト中から適当なノードを選択し、ダウンロード要求者に返却する。コンテンツが公開者によって更新された場合には、コンテンツに対応するノードのリストはクリアされる。

上記の手順を以下に示す。ここで、download は文書のダウンロードを要求する関数であり、nodeId はダウンロードを要求するノードの ID である。update は文書の更新を要求する関数である。getNodeId は docId の文書を保持しているノードの ID を取得する関数である。

```
nodeIdList: 文書 ID × ノード ID のリスト  
  
download(docId, nodeId) {  
  nodeIdList[docId].add(nodeId);  
  
update(docId, nodeId) {  
  nodeIdList[docId] = {nodeId} }  
  
getNodeId(docId) {  
  index = rand() * nodeIdList[docId].length;  
  return nodeIdList[docId][index];  
}
```

## 4.2. コンテンツ交換の記録

Napsterのように単純なプロトコルを用いたP2Pコンテンツ交換システムでは、サービスの運用者がコンテンツの交換を記録しようとした場合、クライアントがダウンロードの状態を偽ることによって、実際に行われたコンテンツの交換と運用者の記録との間に不整合が生じる可能性がある。

例えば、Napster では、ダウンロード開始時に中央サーバに対してダウンロード開始を要求するだけである。このため、ダウンロード元のクライアントがコンテンツを転送しなくても、中央サーバからは正常にコンテンツがダウンロードできたのと同じ情報しか把握できない。

こうした問題を防ぐために、本システムでは、中央サーバで管理されている公開鍵を利用してコンテンツのダウンロードを行うプロトコルを採用する。以下にそのプロトコルを示す。以下の説明では、ダウンロード先のクライアントを A、ダウンロード元のクライアントを B とする。

1. A から中央サーバに対してダウンロード要求を出す。中央サーバは共通暗号鍵を生成し、B に送る。
2. B は共通暗号鍵でコンテンツを暗号化し、自分の秘密鍵で署名して A に送る。
3. A はダウンロードしたデータに B の署名があるこ

とを確認した後、中央サーバに共通鍵を要求する。この要求を受けた時点で中央サーバはコンテンツがダウンロードされたという記録を行う。

4. A は共通鍵によりダウンロードしたデータを復号する。ダウンロードしたコンテンツが目的のものと同じかどうかを中央サーバにある電子署名で確認する。
5. コンテンツが目的のものとは異なっていた場合には、B の署名とともに暗号化されたデータを中央サーバに送る。中央サーバは送られたデータに B の署名があり、コンテンツと一致しないことが確認できたら 3 で行ったダウンロード記録を取り消す。

このプロトコルでは以下の性質が保証される。

- コンテンツのダウンロード元がプロトコル通りに動作していれば、ダウンロードの記録は必ず行われる
- 目的のコンテンツがダウンロードできていないにもかかわらずダウンロードの記録が行われることはない

コンテンツを持っているクライアントが、規定されたプロトコルどおりにコンテンツを共通鍵により暗号化している場合には、コンテンツをダウンロードしたクライアントは復号のために中央サーバに鍵を要求する必要があるため、コンテンツのダウンロード記録が行われる。このため 1 番目の性質が保証される。

また、コンテンツをダウンロードしたクライアントが中央サーバに鍵を要求する際には、ダウンロード元の署名がついたデータを受け取っている。もしこのデータが要求したものと異なっても、これを中央サーバに送信することにより、サーバ上のダウンロード記録を取り消すことができる。このため 2 番目の性質も保証される。

ただし、このプロトコルによっても、ダウンロード元のクライアントがプロトコル通りに暗号化を行わず、ダウンロード先のクライアントもそうしたコンテンツの送信手順に対応している場合には、中央サーバはダウンロード記録を行うことができないと考えられる。このような場合に対しては、過去の交換履歴などの情報を元にクライアントの信頼性を評価し、信頼できるクライアントをダウンロード元を選択するといった手法が考えられる。具体的な手法に関しては今後の課題である。

## 5. 全文検索

### 5.1. 検索の負荷分散方式

本方式では、中央サーバにかかる負荷を軽減するため、検索結果をクライアントにキャッシュする方式を基にする。[9]では、検索エンジンへの要求の 30~40%が繰り返されていることが報告されており、キャッシュによ

り検索処理の負荷を軽減することが期待できる。キャッシュの方式としては、ハッシュをベースとした[3]を応用した手法を用いる。クライアントが検索を行うための手順を以下に示す。

中央サーバは、長時間接続しているクライアントを、キャッシュを行うクライアントとして一定個数選び、ハッシュ値の値域を等分した区間に割り当てる。クライアントはネットワークに接続する際にこのキャッシュのリストを取得する。クライアントが検索を行うときには、検索キーワードのハッシュ値を求め、そのハッシュ値の区間に割り当てられているキャッシュに対して要求を出す。

後述の実験では、クライアントの ID および検索キーワードのハッシュ関数として SHA1 を使い、その上位のビットのみを比較することによりこの処理を実装した。

キャッシュを担当するクライアントは、もし検索キーワードに合致する検索結果を持っていなければ、検索サーバに要求を転送する。検索サーバは、検索キーワード、検索日時および検索結果に電子署名を行い、キャッシュ担当クライアントに返却する。クライアントは、電子署名および検索日時より、検索結果が改ざんされていないことを確認する。

クライアントが検索要求を出したときに、キャッシュ担当クライアントがネットワークからの離脱や過負荷状態であることにより利用できなかった場合には、検索要求を出したクライアントは自分のキャッシュのリスト上でそのキャッシュを無効のマークをつけ、次の区間に割り当てられているキャッシュに検索を要求しなおす。クライアントにおいて利用できないキャッシュが一定以上になった場合には中央サーバに要求を行い、キャッシュのリストを再取得する。

## 5.2. 負荷分散の予備的実験

検索キーワードの繰り返し回数は Zipf 分散をすることが報告されている[9]。このため、キャッシュ担当のクライアントを増やす場合、ハッシュの値域を細かく分割していくと、特定のクライアントに要求が集中する可能性がある。

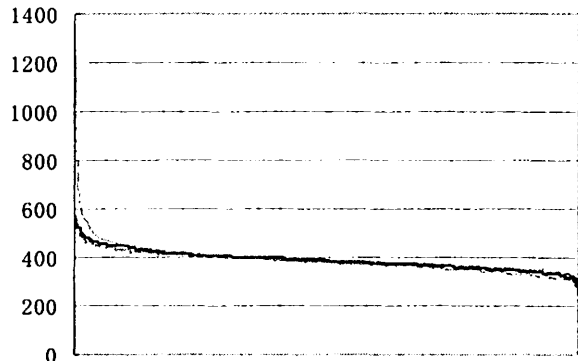
一方で、区間の分割は大きいまま、各区間を複数のクライアントで担当した場合には、その分だけキャッシュのヒット率が低下してしまう問題がある。

本研究では、それぞれの手法の得失を検証するため、予備的な実験を行った。

まず、繰り返し回数が Zipf 分散するように擬似的に検索キーワードのリストを生成した。このリストにおいては、40%のクエリが繰り返されており、また、最頻出の語は全体の 0.2%という比率になっている。これらの比率は[9]で報告されている Excite の検索トレースを元にしたものである。

このリストにおいては、すべての結果がキャッシュできると仮定すると、ヒット率は 40%になると考えられる。

このリストを用いて、(1)ハッシュの値域を 256 分割し、



— (1) 256クライアントで10万要求を処理  
 ..... (2) 1024クライアントで40万要求を処理  
 - - - - (3) 256×4クライアントで40万要求を処理

図2. 検索要求数の分布

	最大検索 要求数	キャッシュ ヒット率	平均キャッ シュ容量
(1) 256	557	20.6%	75.7
(2) 1024	1193	24.3%	61.4
(3) 256×4	578	10.6%	42.1

表1. 検索キャッシュ実験結果

1クライアントが1区間ずつを担当した状態で10万クエリを処理、(2)ハッシュの値域を1024分割し、1クライアントが1区間ずつを担当した状態で40万クエリを処理、(3)ハッシュの値域を256分割し、4クライアントで1区間を担当した状態で40万クエリを処理、という3種類の想定で、キャッシュのヒット率、必要なキャッシュ容量および各キャッシュ担当クライアントへの検索要求回数を測定した。ここで、キャッシュ容量としては、各検索キーワードの検索結果には一定のキャッシュ容量が用いられると仮定し、2回以上検索された検索キーワードの個数をカウントすることとした。

それぞれの実験の結果を図2に示す。グラフは、各キャッシュへの要求回数を多い順に並べたものである。比較が可能なように、(1)の場合のみX軸を4倍に拡大している。

(1)の状態では、キーワードのハッシュ値の値域を単純に等分しただけであるが、最も検索要求を受けるキャッシュも他の2倍以内に収まっている。これは、区間の分割が大きい場合には、高頻度な検索キーワードを含む区間に低頻度なキーワードも含まれ、効果が相殺されるためと考えられる。また、キャッシュのヒット率も高い。

(1)の状況からクライアントの接続数と検索要求回数が増加した場合に、(2)のようにハッシュ値域の分割を細かくしていくと、表1に示されるように特定のキャッシュに要求が集中してしまい、うまく負荷分散されないことがわかる。しかしこの手法では、キャッシュのヒット率は(1)の場合よりも向上している。この理由は、特定のノード以外には適度に負荷が分散されているため、過負荷に陥ったキャッシュから他のキャッシュにクライアントが

要求先を変更していくことにより、適切な効果が得られるのではないかと考えられる。現在のところ、この測定は検索要求数のみを対象としており、実際のキャッシュの負荷状況を考慮に入れてはいない。キャッシュが過負荷に陥ったときの要求先の変更による負荷分散状況の定量的な評価は今後の課題である。

また、(3)のように区間の分割をそのままにした場合、負荷分散は(1)の状況から変わらずに行われているが、キャッシュのヒット率が著しく低下してしまう。この手法でキャッシュのヒット率を向上させるためには、階層的なキャッシュなどの別の手法を組み合わせる必要があると考えられる。

また、本実験では、繰り返されるクエリの比率を一定と仮定しているが、実際には、ユーザの使用語彙には限界があるため、クエリ数の増加に伴い繰り返されるクエリの比率が高くなることが予想される。このため、キャッシュのヒット率もそれほど低下しないことが期待できる。

## 6. まとめ

本稿では、現在普及している P2P コンテンツ交換システムの問題点として、コンテンツ交換の整合性の問題と情報検索の問題を指摘し、それらを解決するための手法を提案した。また、現在我々が開発中の P2P コンテンツ交換システムの概要について説明した。

本システムでは、中央に管理サーバを設け、コンテンツ公開者の電子署名を管理することによりコンテンツの改ざんを防止する。また、非対称暗号鍵を用いたコンテンツの交換プロトコルにより、クライアント間のコンテンツのダウンロードと中央サーバでのダウンロード記録の整合性を維持することが可能になる。これに対し、現在普及している中央サーバを用いた P2P システムでは、本手法のように整合性を維持するための手法が取られていないため、ダウンロード記録に不整合が生じる可能性がある。また、中央サーバを用いない純粋な P2P システムでは、ダウンロード記録をとることは困難である。

情報検索の問題に対しては、各クライアントにおいてテキストの形態素解析、要約処理を行い、コンテンツの更新時に中央サーバに登録することによって効率的な全文検索を実現する。現在数多く研究されている分散ハッシュテーブルをベースとした手法では、部分一致検索や AND 検索などの機能の実現が困難であるが、本手法では比較的容易にこれらの機能を実現することが期待できる。一般に中央のサーバを用いた手法では、中央サーバに検索要求が集中することが懸念されるが、この問題への対応策としては、ハッシュをベースとした手法でクライアントに検索結果を分散させてキャッシュする手法を提案した。実験により、Zipf 分散する検索要求に対しても、実用上必要とされる程度の規模までは比較的公平に負荷分散が行われ、キャッシュのヒット率

も高いことが確認できた。

よって、これらの手法により、P2P コンテンツ交換システムにおいて、高度なコンテンツ交換の整合性維持と効率的な全文検索の実現が可能となる。

今後の課題としては、整合性維持手法については、複数のクライアントがプロトコルを守らない場合の対策が挙げられる。また、実際のインプリメンテーションにおいて、暗号鍵の生成等の処理の負荷が許容範囲内であるかを定量的に測定する必要があると考えられる。検索結果のキャッシュ手法については、特定のクライアントが過負荷に陥らないような負荷分散手法の研究を行う予定である。

謝辞 本研究は、総務省戦略的情報通信研究開発推進制度により、国際技術獲得方研究開発プロジェクトとして総務省の支援を受けています。

## 参考文献

- [1] P. Reynolds and A. Vahdat. Efficient Peer-to-Peer Keyword Searching. Unpublished, June 2002.
- [2] T. Hong. Freenet: A distributed anonymous information storage and retrieval system. In *ICSI Workshop on Design Issues in Anonymity and Unobservability*, 2000.
- [3] David R. Karger, Eric Lehman, Frank Thomson Leighton, Rina Panigrahy, Matthew S. Levine, and Daniel Lewin. Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the World Wide Web. In *ACM Symposium on Theory of Computing*, pages 654-663, 1997.
- [4] Napster. <http://www.napster.com/>, <http://opennap.sourceforge.net/>.
- [5] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for Internet applications. In *Proceedings of ACM SIGCOMM '01*, 2001.
- [6] Gnutella. <http://gnutella.wego.com/>.
- [7] Tylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. A scalable content-addressable network. In *Proceedings of ACM SIGCOMM '01*, 2001.
- [8] Jinyang Li, Boon Thau Loo, Joseph M. Hellerstein, M. Frans Kaashoek, David R. Karger, Robert Morris. On the Feasibility of Peer-to-Peer Web Indexing and Search. In *2nd International Workshop on Peer-to-Peer Systems*, 2003.
- [9] Yinglian Xie and David O'Hallaron. Locality in Search Engine Queries and Its Implications for Caching. *IEEE Infocom 2002*, 2002
- [10] 竹辺靖昭, 美馬秀樹, 苔米地英人. P2P コンテンツ交換システムにおけるコンテンツの整合性維持および全文検索の高度化. 第 114 回 DPS 研究会, 2003