

P2P 型ウェブコンテンツ共有における相関性を考慮した キャッシングシステムの実現

中村 聡史[†] 塚本 昌彦^{††} 西尾章治郎^{††}

[†] 大阪大学大学院工学研究科 〒 565-0871 大阪府吹田市山田丘 2-1

^{††} 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 2-1

E-mail: [†]nakamura@ise.eng.osaka-u.ac.jp, ^{††}{tuka,nishio}@ist.osaka-u.ac.jp

あらまし P2P ネットワークを利用したウェブコンテンツ共有では、コンテンツ取得における待ち時間短縮が重要な課題となる。このコンテンツ取得における問合せ時間を短縮するため、本稿ではコンテンツ間の相関性を考慮したコンテンツキャッシングシステムを実現し、評価実験によりシステムの有用性を明らかにする。

キーワード P2P, ファイル共有, ウェブコンテンツ, 相関性, キャッシング

On realizing a caching system for P2P web contents sharing system

Satoshi NAKAMURA[†], Masahiko TSUKAMOTO^{††}, and Shojiro NISHIO^{††}

[†] Graduate School of Engineering, Osaka University Yamadaoka 2-1, Suita-shi, Osaka, 565-0871 Japan

^{††} Graduate School of Information Science and Technology, Osaka University Yamadaoka 2-1, Suita-shi, Osaka, 565-0871 Japan

E-mail: [†]nakamura@ise.eng.osaka-u.ac.jp, ^{††}{tuka,nishio}@ist.osaka-u.ac.jp

Abstract In a web contents sharing system on P2P network, it is important to shorten the waiting time for contents acquisition. In order to achieve this, in this paper, we realized a contents caching system which uses the correlation between contents to make the caching efficient, and make the usefulness of a system clear by some experiments.

Key words P2P, file sharing system, web contents, correlation, caching

1. ま え が き

近年、コンピュータの低価格化・高性能化、ネットワークインフラの整備などにより、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。その数は、JP ドメインに所属するサーバで公開されているコンテンツだけでも約 2 億 7421 万ファイルにのぼり、コンテンツの総サイズは 10 テラバイトに達すると推計されている [8]。ユーザは検索サービスなどを利用することにより、これら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、ユーザがオフライン状態にあるときや、サーバやネットワークが不調な時はコンテンツに辿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ものであり、ユーザが必要とするときにはコンテンツが別のものになっている可能性もある。さらに、コンテンツの数があまりに多すぎるため、新しいコンテンツに対する検索などは困難である。

こうした問題を解決するため、本研究グループではウェブコンテンツ共有システムを提案・実装している [1]。ウェブコンテンツ共有システムでは、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 [5] のような PureP2P 型のネットワークで共有する。ウェブコンテンツを保存する際に、取得元の URL や、ウェブページのタイトル、ファイルサイズ、更新日時、取得日時などのメタ情報もまとめて保存しておくことで、メタ情報を利用した柔軟な検索を可能とする。共有されるコンテンツの種類は多岐にわたり、すでに公開されていない過去のコンテンツなども共有できるため、ユーザにとって有用なシステムであるといえる。また、ユーザがコンテンツを保存したときにそのコンテンツが検索対象となるため、新しいコンテンツに対する検索も可能となる。さらに、同じ興味をもつユーザ同士でコンテンツ共有ネットワークを構成することにより、効果的にコンテンツ検索および共有ができる。このウェブコンテンツ共有システムから構成される P2P ネットワークをウェブコンテンツ共有ネットワークと呼んでいる。ウェブコンテンツ共有ネットワークではバケツリレー式にコンテンツを転送する [2] ため、コンテンツ取得までの待ち時間の短縮が重要な課題となる。

コンテンツ取得における待ち時間を短縮するには、各ピアにおいてコンテンツ間の相関性を考慮してコンテンツをキャッシ

ングすることが有効であると考えられる。本研究グループでは、コンテンツの取得要求を受けた際、コンテンツ保持ピアにおいてもコンテンツの解析を行い、関連性の高いコンテンツをコンテンツ要求ピアに向けて送出することでコンテンツ取得における待ち時間を短縮する、能動的キャッシング手法を実現している[3]。能動的キャッシング手法を利用することによりコンテンツ取得における待ち時間は短縮されるが、この手法は問合せ時に場当たり的に動作するものであるうえ、各ピアが独自にコンテンツをキャッシュしているため、ネットワーク全体を俯瞰した際に効果的であるとは限らない。

そこで本研究では、ウェブコンテンツ共有システムにおけるコンテンツ間の関連性を考慮したキャッシングについて考察を行い、コンテンツ要求における待ち時間を短縮する分散キャッシング手法を提案、実装する。分散キャッシング手法では、相互接続するピア同士がコンテンツの関連性などを考慮し、効果的にキャッシュを分散配置することで、コンテンツ取得要求におけるコンテンツ要求ピアからコンテンツ保持ピアまでのホップ数を低減する。また実験により、これまでに実現した能動的キャッシング手法と、分散キャッシング手法の有用性について考察を行う。

以下、2章ではコンテンツ共有の特徴について述べ、3章ではウェブコンテンツ共有システムにおけるキャッシング手法の提案を行う。4章ではキャッシングシステムの実装について述べ、5章では実験によるシステムの有用性の評価を行う。6章で実現したキャッシング手法に関する考察を行い、最後に7章にてまとめと今後の課題について述べる。

2. コンテンツ共有

2.1 コンテンツ間の関連性

ユーザがブラウザを利用して WWW 上にあるウェブコンテンツを閲覧する場合、ウェブコンテンツと、そのウェブコンテンツ内に含まれる画像やスタイルシートなどのインラインオブジェクトは必ず連続してアクセスされる。また、そのウェブコンテンツからリンクとして接続されている他のウェブコンテンツは、連続してアクセスされる可能性が高い。さらに、ディレクトリサービスなどで同一のカテゴリに分類されているようなウェブコンテンツや、検索エンジンを利用して同一キーワードで取得されるウェブコンテンツも連続してアクセスされる可能性が高い。

以上のように、あるコンテンツ閲覧システムにおいて複数のコンテンツが存在し、それらのコンテンツが連続してアクセスされる場合、コンテンツ間に関連性があるという。この関連の度合いを相関度と呼び、アクセスされる頻度が高ければ高いほど相関度は高いものとする。

コンテンツ共有システムにおいて、コンテンツ間の関連性は大きく分けて以下のように分類される。

従属関係： コンテンツ A がコンテンツ B をコンテンツの一部として含む場合、コンテンツ B はコンテンツ A に従属しているという。主コンテンツがアクセスされた際、従属しているコンテンツは必ず連続してアクセスされる。

接続関係： コンテンツ A からコンテンツ B への接続が存在する場合、コンテンツ B はコンテンツ A に対して接続関係にあるといえる。主コンテンツがアクセスされた後、接続関係にあるコンテンツは連続してアクセスされる可能性が高い。

類似関係： コンテンツ A とコンテンツ B がある条件下において同一カテゴリに分類される場合、コンテンツ A とコンテンツ B は類似関係にあるといえる。主コンテンツがアクセスされた後、類似関係にあるコンテンツは連続してアクセスされる可能性が高い。

無関係： コンテンツ A とコンテンツ B の間に関係が存在しない場合、コンテンツ A とコンテンツ B は無関係であるといえる。無関係にあるコンテンツが連続してアクセスされる可能性は低い。

このコンテンツ間の関連性をウェブコンテンツ共有に適用すると、コンテンツとコンテンツ内部に含まれるインラインオブジェクトとの関係は従属関係に、コンテンツとコンテンツからリンクされているコンテンツとの関係は接続関係に、同一のクエリで検索されるコンテンツ間の関係は類似関係に、まったく関係のないコンテンツ間の関係は無関係に分類される。ここで分類した関連性は、コンテンツの種類や問合せの内容により強弱が異なるが、本稿では簡単のため、従属関係にあるコンテンツ間の相関度が最も高く、接続関係および類似関係にあるコンテンツ間の相関度は中程度、無関係のコンテンツ間の相関度はないものとして考える。

2.2 コンテンツ交換手法

P2P 型のコンテンツ共有におけるコンテンツのやり取り方式には、直接交換方式と間接交換方式が存在する。

直接交換方式とは、コンテンツ要求ピアからコンテンツ保持ピアに対してコネクションを確立し、確立したコネクションを利用して直接コンテンツを取得するものである。この方式は Gnutella [5] や Napster [6] などの P2P ファイル共有システムに利用されており、P2P コンテンツ共有における一般的なコンテンツ交換手法であるといえる。直接交換方式では、P2P ネットワークから見ると 1 ホップでコンテンツをやり取りできるため、一般にコンテンツ取得における待ち時間が短くなる。一方で、コンテンツ要求ピアからコンテンツ保持ピアに対して直接コネクションを確立できない場合はコンテンツを取得できないうえ、コンテンツを保持するピアに負荷が集中するという問題がある。

間接交換方式とは、コンテンツ要求ピアとコンテンツ保持ピアの間に複数のピアが存在し、コンテンツ問合せを行った経路を利用して、コンテンツ保持ピアからコンテンツ要求ピアに対して順次バケツリレー式にコンテンツを転送するものである。この方式は、Winny [7] などのファイル共有システムで利用されている。間接交換方式では、一部の性能の良いピアが多数のピアと相互接続し、人気のあるコンテンツをキャッシュしておくことにより、コンテンツを保持するピアに負荷が集中することを防ぐことができる。また、直接コネクションを確立する必要がないため、コネクション確立に関する問題もない。一方、間接交換方式では、コンテンツ要求ピアとコンテンツ保持ピアの間のホップ数が大きくなるため、コンテンツ取得における待ち

時間が長くなってしまいう問題がある．このコンテンツ要求ピアとコンテンツ保持ピアのホップ数が減少すると，コンテンツ取得までの待ち時間も短くなるため，ネットワーク上でコンテンツを効果的にキャッシングすることが重要な課題となる．

一般的な P2P ファイル共有システムでは，音楽ファイルなどファイルサイズが大ききものが共有されている．ウェブコンテンツ共有システムでは，共有されるファイルのサイズがネットワーク帯域に比べ非常に小さく，共有されるファイルの数も膨大であるという特徴がある．また，コンテンツ交換時に相関度の高いコンテンツに対して連続的にコンテンツ要求が発生するうえ，人気のあるコンテンツを保持しているピアにはコンテンツ取得要求が集中する．そのため，ウェブコンテンツ共有システムでは，コンテンツ共有に間接交換方式をとっている [2]．間接交換方式をとり，中継ピアにおいてコンテンツをキャッシュすることにより，コンテンツ保持ピアへ負荷が集中してしまう問題を防ぐことが可能となっている．

2.3 能動的キャッシング

ウェブコンテンツ共有ネットワークで共有されるウェブコンテンツは，それ自身である HTML ページの他に，CSS などのスタイルシート，JPEG や PNG といった画像などのインラインオブジェクトから構成されている．このインラインオブジェクトはウェブコンテンツの閲覧において欠かせないものである．ウェブコンテンツ共有ネットワークにおいてコンテンツを取得する場合には，まずコンテンツの母体となる HTML ページを要求し，HTML ページを取得する．取得した HTML ページを解析した後，順次インラインオブジェクトを要求し，取得するという手順が必要になる．そのため，インラインオブジェクトの量が多い場合には要求および応答の数も増加してしまい，結果としてコンテンツ取得における待ち時間が長くなってしまいう．また，コンテンツ要求ピアとコンテンツ保持ピアの間のホップ数が大きい場合には，何度も応答が行き来する必要があるため，コンテンツ中継ピアの負荷も増大してしまいう．ウェブコンテンツ共有ネットワークにおいては，コンテンツ要求ピアからコンテンツ保持ピアまでのホップ数がコンテンツ取得の待ち時間に大きく影響する．そのため，相関度の高いコンテンツをコンテンツ要求ピアの近くのピアでキャッシュしておくことにより，コンテンツを取得するまでの待ち時間を大幅に短縮することができる．

本研究グループで実現している能動的キャッシング手法では，コンテンツ保持ピアがコンテンツ取得要求を受信したときに，要求されたコンテンツをコンテンツ要求ピアに向け送信すると同時に，コンテンツの内容を解析する．解析により従属関係にあるインラインオブジェクトや接続関係にあるリンク先のコンテンツをリストアップし，その相関度とコンテンツ要求ピアから保持ピアまでのホップ数 (*Hops*) をもとに，コンテンツをコンテンツ要求ピアに向け転送する．このとき，コンテンツの転送回数 (*TTL*) は下記の式により与えられる．

$$TTL = Hops \times \text{相関度}$$

このやり取りにより，連続してコンテンツにアクセスすると予

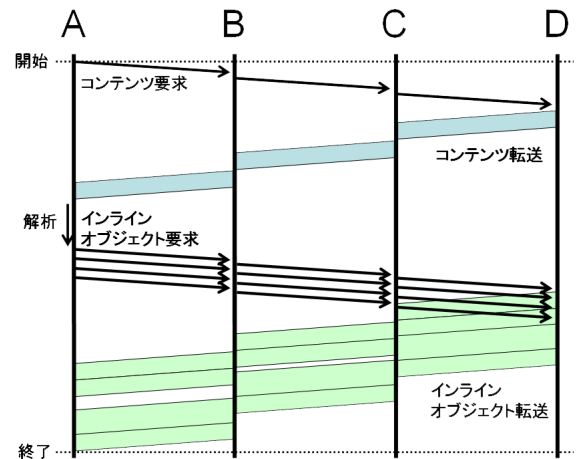


図 1 コンテンツ取得要求

Fig. 1 An example of figure caption.

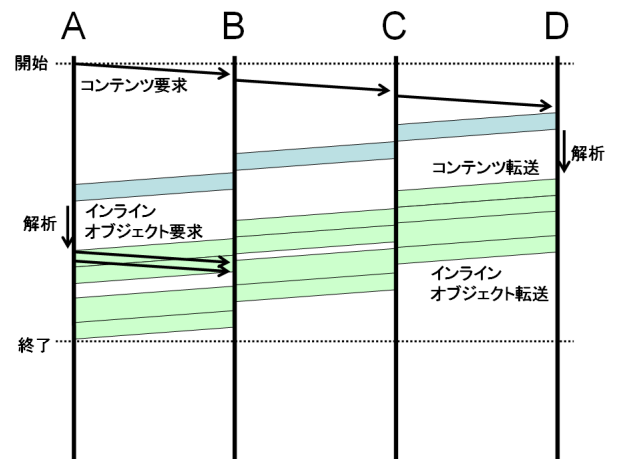


図 2 能動的キャッシングを利用したコンテンツ取得

Fig. 2 An example of figure caption.

想されるコンテンツ要求ピアの近くで，対応するコンテンツをキャッシュすることが可能となり，コンテンツ取得にかかる待ち時間を短縮できると考えられる．なお，コンテンツの相関度は 0 から 1 までの値で与えられるが，従属関係の場合は 1，接続関係の場合および類似関係の場合は 0.25，無関係のときは 0 としている．

図 1 は，ウェブコンテンツ共有ネットワークにおいて，コンテンツ取得要求ピア A から，ピア B，ピア C を中継して，コンテンツ保持ピア D に対してコンテンツ取得要求を行っている様子である．一方，図 2 は，コンテンツ取得の際に能動的キャッシング手法を利用している様子である．図 1 では，まず始めに母体となるコンテンツを要求し，コンテンツを取得した後，コンテンツを解析し，インラインオブジェクトを要求している．インラインオブジェクトは一気に要求され，コンテンツ保持ピアから順次コンテンツが転送されている．図 2 でも同様に，母体となるコンテンツを要求し，引き続きインラインオブジェクトを要求している．このときに，コンテンツを保持するピア D でもコンテンツの解析を行い，関連するインラインオブジェクトを転送しているため，ピア B-ピア C 間，ピア C-ピア D 間

の無駄なやり取りを省くことに成功しており、結果としてコンテンツ取得までの待ち時間が短縮されている。

能動的キャッシング手法を利用することにより、コンテンツ取得までの待ち時間を短縮することはできているが、コンテンツ要求が発生したときに始めて動作するため、後手を踏むことが多くなってしまふ。また、次から次へとコンテンツがキャッシュされるため、前にキャッシュされたコンテンツが破棄されやすくなる。さらに、コンテンツの経路となるピアが同一のコンテンツをキャッシュしていくため、隣接するピア単位でキャッシュの内容を比較したときに、キャッシュの内容が似たようなものになってしまう。結果として、経路単位で見たときのキャッシュに過度の偏りが発生してしまい、非効率であると考えられる。

3. 分散キャッシング手法

ウェブコンテンツ共有システムのようにバケツリレー式にコンテンツを交換するコンテンツ共有システムでは、コンテンツ要求ピアからコンテンツ保持ピアまでのホップ数を低減することが、コンテンツ取得における待ち時間短縮において有効となる。ホップ数の低減にはコンテンツのキャッシュが有効であるが、全てのコンテンツを各ピアがキャッシュすることは困難であるため、コンテンツを極力分散配置することが重要となる。しかし、ウェブコンテンツ共有ネットワークでは階層のないP2P型のネットワークを利用しているうえ、共有されるコンテンツの数が膨大すぎるため、コンテンツをトップダウン式に分散配置することはできない。また、ウェブコンテンツ共有ネットワークではコンテンツが転送される経路に同一のコンテンツがキャッシュされていくため、経路単位で比較した場合、キャッシュに無駄が多くなってしまふ。

自ピアのキャッシュ可能サイズが限界に達し、キャッシュ中からコンテンツを削除する必要が発生した場合、コンテンツ取得要求に必要なホップ数を極力増加させないようにコンテンツを削除することが好ましい。提案する分散キャッシング手法では、コンテンツ取得要求に必要なホップ数を極力増加させないようにコンテンツに優先度を設定し、キャッシュ可能サイズの限界近くまでコンテンツを保存すると、優先度の低いコンテンツから順に破棄していく。

キャッシュ中に蓄えられたコンテンツの優先度は、下記に挙げる各種条件により与えられる。

- 最後にアクセスされてからの経過時間
- 他のコンテンツとの相関度
- 隣接ピアでの保存状態
- コンテンツのサイズ

転送されてきたコンテンツを受信し、キャッシュに保存したときや、コンテンツに対する検索問合せがあったとき、コンテンツに対する取得要求があったときは、そのコンテンツが必要とされていると考えられる。必要とされているコンテンツは引き続きアクセスされる可能性が高い。そこで、最後にアクセスされてからの時間が短いコンテンツの優先度を高く設定し、最後にアクセスされてから長い間アクセスされていないコンテンツは優先度を低く設定する。この優先度の設定により、あまり

アクセスされていないコンテンツを効果的に破棄し、アクセスされやすいコンテンツをキャッシュ中に残すことが可能となる。

コンテンツ間に強い相関関係のあるウェブコンテンツ共有システムでは、コンテンツが連続的にアクセスされる可能性が高い。特に、従属関係にあるコンテンツは、必ず連続してアクセスされるため、主となるコンテンツと従属するコンテンツは合わせて保持しておくことが好ましい。主となるコンテンツが削除される場合は、従属するコンテンツを削除しても大きな問題にはならないと考えられるが、他に従属するコンテンツは、複数の主コンテンツをもつ可能性がある。ここで、すべての主コンテンツのアクセス確率を同一とした場合、多数の主コンテンツと従属関係にあるコンテンツはアクセスされる可能性が高いと考えられる。つまり、他コンテンツとの相関度の強いコンテンツの優先度を高く設定し、キャッシュに長く残すことにより、コンテンツ取得要求時における待ち時間を短縮できる。

ウェブコンテンツ共有システムにおけるローカルディスクへのコンテンツ保存スタイルとしては、ユーザがデータとして明示的にローカルディスクに保存するタイプと、中継時にキャッシュとして一時的に保存するタイプがある。ユーザが明示的にコンテンツをローカルディスクに保存している場合は、ユーザが削除処理を行わない限りコンテンツが消えてしまうことはないが、キャッシュに保存されているコンテンツは、キャッシュの状態に応じてシステムに自動的に削除されてしまふ。つまり、自ピアと隣接するピアでデータとして保存されているコンテンツは、そのピアを操作するユーザが明示的にそのコンテンツを削除するか、そのピアがネットワークから退出しない限りは、自ピアからアクセス可能である。一方、隣接するピアでキャッシュ中に保存されているコンテンツは、キャッシュ可能サイズなどの都合により、自動的に削除される可能性があるため、常にアクセス可能であるとは限らない。もし、自ピアにおいてキャッシュに保存されているコンテンツが、隣接するピアでデータまたはキャッシュとして保存されている場合は、キャッシュからそのコンテンツを削除しても、コンテンツ取得要求におけるホップ数は最悪で1しか増加しない。そこで、自ピアでキャッシュされているコンテンツが、隣接するピアで保存されている場合は優先度を低く、隣接するピアで保存されていない場合は優先度を高く設定することにより、コンテンツ取得要求における待ち時間を短縮できると考えられる。このとき、コンテンツを保持している隣接ピアの数が多ければ多いほど優先度は低く、少なければ少ないほど優先度は高く設定することが好ましい。また、コンテンツがデータとして保存されているか、キャッシュとして保存されているかによって、コンテンツのアクセス可能性は異なる。そこで、データとして保存されていれば優先度を低く、キャッシュに保存されている場合は優先度を高く設定することが考えられる。一方、隣接ピアでの保存状態のチェックは、負荷の増大を招く可能性がある。

ウェブコンテンツ共有システムで共有されるコンテンツは、WWW上で公開されているコンテンツを想定しているため、コンテンツの種類はテキストから画像、音声、映像など多岐にわたる。また、共有されるコンテンツのサイズも、コンテンツの

種類や内容によってさまざまである。例えば、HTML に代表されるテキストコンテンツのサイズは、10 キロバイト程度のものが一般的で、大きくとも 100 キロバイト程度である。一方、JPEG に代表される写真などの画像ファイルは 60~100 キロバイト程度のものが一般的だが、数百キロバイトに達するものも多々ある。また、WAV や MP3 などといった音声や音楽ファイルは、1~5 メガバイト程度と他のコンテンツに比べファイルサイズが非常に大きい。ここで、キャッシュ可能サイズが 100 メガバイトとしたとき、10 キロバイトのコンテンツはキャッシュに 10000 個保持できるが、1 メガバイトのコンテンツはキャッシュに 100 個しか保持することができない。全てのコンテンツが同じ確率でアクセスされるとすると、ファイルサイズの大きなコンテンツよりは、ファイルサイズの小さいコンテンツを多数キャッシュに保持する方が、そのピアにおけるキャッシュヒット率は高くなる。つまり、単純にキャッシュヒット率を向上させるためには、ファイルサイズの小さなコンテンツをキャッシュに優先的に残す方がよいと考えられる。ただし、すべてのピアがファイルサイズの小さなコンテンツをキャッシュに優先的に残すような戦略をとると、ファイルサイズの大きなコンテンツはどこでもキャッシュされにくくなる。また、このファイルサイズの大きなコンテンツは、ファイルサイズの小さなコンテンツに比べ、ネットワークを介したコンテンツ転送にも余計に時間がかかってしまう。結果として、ネットワーク全体で見たときのパフォーマンスが低下する可能性がある。そこで、ネットワークの帯域が狭いときにはファイルサイズの大きなコンテンツを優先的に保持し、キャッシュ可能サイズに余裕がないときにはファイルサイズの小さなコンテンツを優先的に保持するなどして、各ピアでファイルサイズに関するコンテンツの優先度のつけ方を変更する機構が必要となると考えられる。

以上のように、キャッシュ中におけるコンテンツの優先度は、最後にアクセスされてからの経過時間、他コンテンツとの相関度、隣接ピアでの保存状態、コンテンツのサイズなどにより決定される。しかし、膨大なコンテンツが存在するウェブコンテンツ共有システムでは、これらすべての条件を考慮して随時優先度の低いコンテンツを計算するのは困難である。

ここで、コンテンツ取得要求や能動的キャッシングにおいてコンテンツが転送される際、等しい相関度をもつコンテンツは、コンテンツ保持ピアから同じホップ数だけ転送される。コンテンツが転送される場合は、コンテンツが転送されたホップ数 ($Hops$)、あて先までのホップ数 (TTL) が指定されてコンテンツが送信されるため、その値を利用してキャッシュ中のコンテンツの保持や削除を判断することで、コンテンツを分散配置できると考えられる。

例えば、コンテンツを中継する際、コンテンツを保持しているピアからのホップ数が 1 の場合、コンテンツは隣接ピアでデータまたはキャッシュとして保存されていたと考えられる。そのため、 $Hops$ が 1 の時は該当するコンテンツを優先的にキャッシュに残留させる必要はないと考えられる。ホップ数が 1 のときにコンテンツを受信したピアがそのコンテンツを優先

的にキャッシュに残さないという戦略をとる場合、ホップ数が 2 のときにコンテンツを受信したピアはそのコンテンツをキャッシュに残すことが好ましい。同様に、奇数ホップ目のピアではコンテンツを積極的に保持せず、偶数ホップ目ではコンテンツを積極的に保持するようにすると、効果的にキャッシュの分散配置が可能になると考えられる。一方、コンテンツ取得要求ピアからのホップ数 (TTL) が 1 の場合は、コンテンツ取得要求ピアでコンテンツが保持される可能性が高いといえる。そのため、 $Hops$ の時と同様、 TTL が偶数のときは該当するコンテンツをキャッシュに残すようにし、 TTL が奇数のときはコンテンツをキャッシュに残すのを優先しないことにより、コンテンツを分散配置できると考えられる。 $Hops$ または TTL をベースとしたキャッシングにより、従属するコンテンツなど相関性のあるコンテンツは同じピアに保持されやすくなるため、効果的に分散キャッシュできると考えられる。

以上の仕組みにより、隣接ピアでのコンテンツの保存状態を予測し、結果的に隣接するピア同士で協調しながらコンテンツを保持することが可能となるため、隣接ピアごとにキャッシュが最適化されると考えられる。

4. 実 装

分散キャッシング手法では、キャッシュ可能サイズの限界に達すると、キャッシュ中に含まれるコンテンツの優先度の低いコンテンツから順に削除する。ウェブコンテンツ共有システムで共有されるコンテンツのサイズは、他のコンテンツ共有システムで共有されるコンテンツのサイズに比べ非常に小さい (簡易調査によると平均で 10KBytes 程度)。つまり、キャッシュ可能サイズを 100MBytes 程度に制限しても、キャッシュされるコンテンツの数は 1 万に達する。キャッシュシステムは、優先度の低いコンテンツを即座に求め、該当するコンテンツをキャッシュから迅速に削除する仕組みが必要となる。なお、本プロトタイプシステムの実装においては、コンテンツの優先度について、最後にアクセスされてからの経過時間と、他コンテンツとの相関度について考慮する。また、転送回数やホップ数をベースとした優先度の設定により、隣接ピアでの保存状態についても考慮する。

そこで本研究では、キャッシュ中のすべてのコンテンツを、最後にアクセスされてからの経過時間をベースとしたリスト形式 (キャッシュリスト) で保持する。キャッシュリストには、コンテンツ自体の情報 (URL や最終更新日、サイズなど) のほかに、タイムスタンプ、繰り返し回数という値を用意する。キャッシュリストには先頭から追加し、末尾から削除するものとする。コンテンツ自体は、URL と日時をベースとしたキャッシュフォルダに保存し、保存時やアクセス時に、ファイルシステムにおいてコンテンツのファイルの最終更新日を設定する。また、キャッシュ可能サイズとキャッシュ使用サイズを保持する。なお、タイムスタンプというのは該当するコンテンツがアクセス (受信, 送信, 保存など) された時刻を意味する。また、繰り返し回数というのは、コンテンツのキャッシュ中における優先度をベースにした値で、コンテンツがキャッシュリストの末尾に

達し、削除対象となったときに、再度キャッシュの先頭への追加を許可するかどうかを判断するための値である。この繰り返し回数と経過時間をベースとしたリスト形式のキャッシュリストを利用することで、相関性などを考慮した優先度付きの分散コンテンツキャッシュシステムを実現する。

繰り返し回数は、コンテンツ検索問合せ時やコンテンツ取得要求時、コンテンツ転送時に、コンテンツ保持ピアからのホップ数 ($Hops$)、コンテンツ要求ピアまでのホップ数 (TTL) をもとにして設定する。ここで、 $Hops$ による方法と、 TTL による方法を併用すると、 $Hops$ が奇数で TTL が偶数のときなどにどちらを優先するべきかという問題が発生する。コンテンツ保持ピアとコンテンツ要求ピアを比較すると、コンテンツ保持ピアの方がコンテンツをデータとして保存している可能性が高いと考えられるため、ここでは $Hops$ を優先してコンテンツに繰り返し回数を与える。一方、コンテンツ保持ピアからコンテンツ取得要求ピアまでのホップ数が大きいとき、中間のピアでコンテンツを長く保持することが好ましいと考えられる。 $Hops$ と TTL の和により、中継ピアにおいてコンテンツ保持ピアからコンテンツ取得要求ピアまでのホップ数が求まるため、 $Hops$ と TTL の和が大きく、中間地点などにピアが存在するとき、コンテンツを長く残留させるようにする。ここでは特に、 $Hops$ や TTL が 0 の場合コンテンツの繰り返し回数を 2 に、 $Hops$ および TTL が 1 のときは繰り返し回数を 0 に設定する。また、中間地点などにコンテンツを長く残留させるため、 $Hops$ と TTL の和が 8 以上で、 $Hops$ が 4 の倍数かつ TTL が 4 以上のとき、繰り返し回数を 2 に設定する。上記の条件を満たさず、 $Hops$ が偶数の時には繰り返し回数を 1 に、奇数のときは繰り返し回数を 0 と設定する。

コンテンツを取得または送信すると、同一のコンテンツがキャッシュ中に含まれているかどうかを判定する。同一のコンテンツがキャッシュに含まれていない場合は、コンテンツをキャッシュフォルダに格納した後、キャッシュリストの先頭にコンテンツの情報を格納する。このとき、キャッシュリストのタイムスタンプとキャッシュ中のファイルの最終更新日を同一のものに設定する。また、キャッシュの繰り返し回数も $Hops$ や TTL をもとにして同様にセットする。さらに、キャッシュ使用サイズに追加したコンテンツ (ファイル) のサイズを加算する。すでに同一のコンテンツがキャッシュに含まれている場合は、コンテンツの最終更新日を現在時刻に設定し、キャッシュリストの先頭にコンテンツの情報を格納する。コンテンツに対する取得要求や検索問合せが発生したときには、キャッシュ中のコンテンツの最終更新日を変更し、キャッシュリストの先頭にそのコンテンツの情報を追加する。

一方、コンテンツ追加時にキャッシュ可能サイズとキャッシュ使用サイズを比較し、キャッシュを限界近くまで使用していないかを調べる。キャッシュ可能サイズにキャッシュ使用サイズが近づくと、新しいコンテンツを保存するための余裕がなくなるため、キャッシュリストから削除するコンテンツを決定し、キャッシュフォルダからコンテンツを削除する。なお、キャッシュ使用サイズがキャッシュ可能サイズを超えないようにする

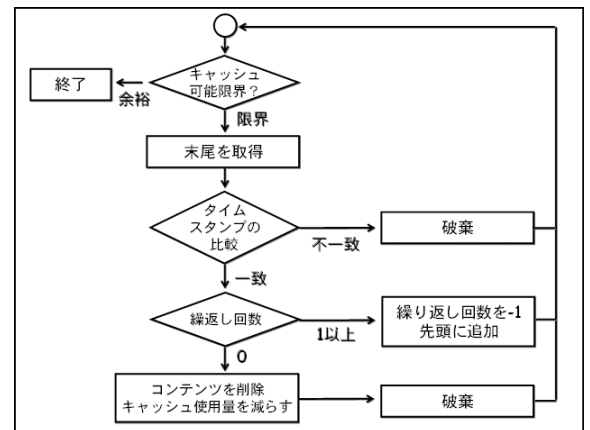


図3 キャッシュからのコンテンツ削除処理

Fig. 3 The flowchart for delete some contents from cache.

ため、キャッシュ使用サイズがキャッシュ可能サイズの 95% に達したとき、キャッシュ削除試行を実施する。この割合をキャッシュ削除開始割合と呼ぶ。この割合はユーザごとに設定できるようにする。

実際にキャッシュを削除する試行は図3の通りである。まず始めにキャッシュリストの末尾にあるコンテンツ情報をリストから取り出し、コンテンツのタイムスタンプと、そのコンテンツに対応するキャッシュフォルダ中のファイルの最終更新日を比較する。タイムスタンプとファイルの最終更新日が一致しない時はこのキャッシュ情報を破棄する。一方、コンテンツの最終更新日とタイムスタンプが一致する場合は、コンテンツ情報に含まれる繰り返し回数をチェックし、繰り返し回数が 1 以上の場合は、繰り返し回数を 1 つ減らしてキャッシュリストの先頭にそのキャッシュ情報を追加する。繰り返し回数が 0 の場合は、キャッシュフォルダからコンテンツを削除し、キャッシュ使用サイズをコンテンツサイズ分減らす。また、コンテンツのキャッシュ情報も破棄する。この試行を、キャッシュ使用サイズがキャッシュ可能サイズの 90% を下回るまで繰り返す。この割合をキャッシュ削除停止割合と呼ぶ。この割合についてもユーザごとに設定できるようにするが、削除開始割合と削除停止割合の割合が近ければ近いほど、キャッシュ削除試行は頻繁に発生するため、ピアでの計算量が増加する可能性がある。

本実装では、キャッシュリスト中に、同一コンテンツに関する複数のキャッシュ情報が登録されるが、前に登録されたキャッシュ情報のタイムスタンプと、コンテンツの最終更新日が異なるため、そのキャッシュ情報を処理するときは無視する。このような実装により、キャッシュリストの無駄な走査が不要となり、ピアにおける負荷を低減できると考えられる。

なお、コンテンツ取得要求ピアにおいて、取得したコンテンツをユーザがデータとして保存した場合は、該当するコンテンツをキャッシュから削除する。

5. 実測実験

能動的キャッシング手法と、分散キャッシング手法を利用したプロトタイプシステムの有用性を示すため、実機を用いた簡

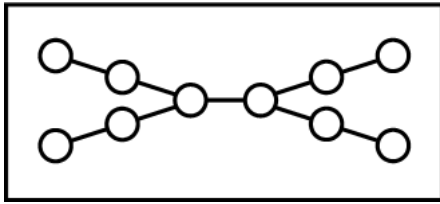


図4 実験で利用したネットワークトポロジ
Fig. 4 The network topology for experiment.

易実測実験を実施した。

ここでは、コンテンツを中継する際にコンテンツをキャッシュに保存し、一定期間を経過後キャッシュから削除する従来手法と、能動的キャッシング手法のみを利用した場合、分散キャッシング手法のみを利用した場合、能動的キャッシング手法と分散キャッシング手法両者を併用した場合について比較を行う。それぞれの手法について、実機を用いたネットワークを利用して、キャッシュ可能サイズを変化させながらコンテンツ取得における待ち時間がどのように変化するかを計測する。

実験では、10台のピアを用意し、図4のようなトポロジのP2Pネットワークを構成する。また、コンテンツとして4,000個のウェブページ（ファイル数は約28,000、総サイズは約480MBytes）を用意し、10台の各ピアに他と重複しない400個のウェブページと、他と重複する適当に選ばれた100個のウェブページをデータとして保存しておく（合計500ページ、ファイル数は平均で約3,700、サイズは平均で約65MBytes）。

各ピアはコンテンツ問合せに利用するため、すべてのコンテンツのURLを保持しておく。各ピアは、30秒ごとに問合せを行うタイミングをランダムに決定し（0秒～10秒の間で決定）、ランダムに取得するコンテンツをあらかじめ用意しておいたURLリストの中から選択し、コンテンツの取得を行う。なお、取得対象となるコンテンツを、データとして保持している場合は、改めて別のコンテンツを選択する。コンテンツ取得の前には、実際にURLを利用したコンテンツの検索問合せを行い、最初に返事が返ってきたピアに対してコンテンツ取得要求を送信する。本実験では、コンテンツ取得要求を送信してから、インラインオブジェクトも含めたウェブページを完全に受信できるまでの時間を、各ピアにおいて計測する。この試行を1時間にわたり実施する。

時間の経過によるコンテンツ取得の待ち時間の変化は、図5と図6の通りである。図の横軸は時間で、縦軸はすべてのピアにおけるコンテンツ取得待ち時間の平均を表している。図5はキャッシュ可能サイズを10MBytesに、図6はキャッシュ可能サイズを20MBytesに制限しているものである。図より明らかのように、コンテンツが随時キャッシュされていくことにより、コンテンツ取得における待ち時間が短縮されている。また、キャッシュ可能サイズに余裕があるときに比べ、キャッシュ可能サイズに余裕がないときは、あまり良い結果が出ていないことがわかる。各手法を比較すると、従来手法に比べ、能動的キャッシング手法を利用するとコンテンツ取得における待ち時間が大幅に短縮できていることがわかる。これは、ホップ数の

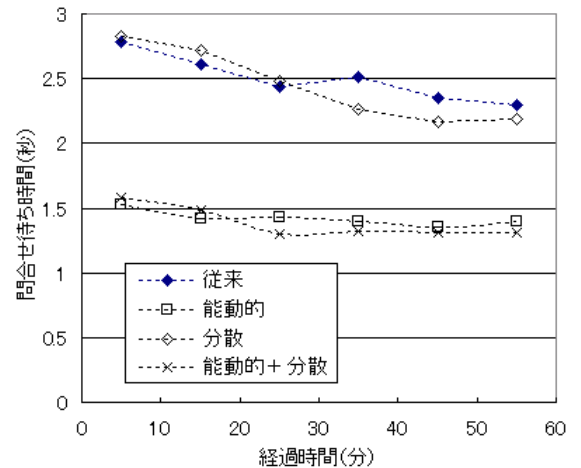


図5 実験結果（キャッシュ可能サイズ10MBytes）
Fig. 5 The results of experiment. (10MBytes)

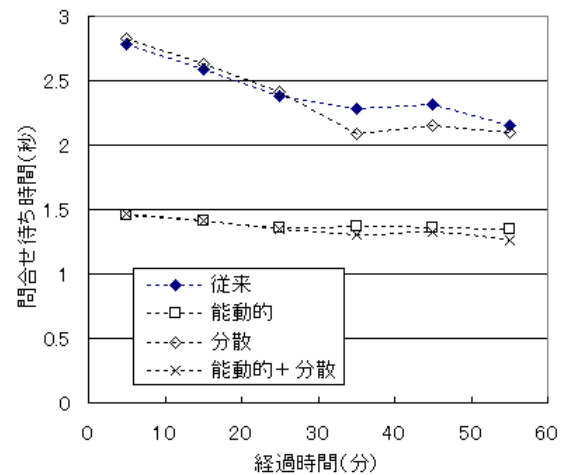


図6 実験結果（キャッシュ可能サイズ20MBytes）
Fig. 6 The results of experiment. (20MBytes)

低減がかなり効率よく働いているためと考えられる。一方、分散キャッシング手法については、従来手法に比べ、わずかに良い結果が得られていることがわかる。また、従来手法や能動的キャッシング手法がキャッシュサイズの変化により影響を受けているのに比べ、分散キャッシング手法ではキャッシュサイズの変化にあまり影響を受けていないことがわかる。

6. 考 察

実測実験の結果より、ウェブコンテンツ共有ネットワークにおけるコンテンツ取得の待ち時間短縮に、コンテンツの能動的キャッシングが有効であることが示された。一方、分散キャッシング手法の有用性についてはあまり明らかにすることができなかった。これは、分散キャッシング手法は、コンテンツ取得要求ピアからコンテンツ保持ピアまでのホップ数が長く、長時間での運用により効力を発揮するのに対し、今回実施した実験では、都合上大きなネットワークを運用できず、コンテンツ取得要求ピアからコンテンツ保持ピアまでのホップ数があまり長くならなかったこと、時間的な都合からあまり多くのコンテンツ

について試すことができなかつたことなどが理由として考えられる。また、あらかじめ用意したコンテンツの量が多くないことや、キャッシュ可能サイズも比較的小さく設定されていたことも分散キャッシング手法にとって不利だったのではないかと考えられる。今後の実験では、能動的キャッシング手法と、分散キャッシング手法に狙いを絞り、長期的な実験を行うことで、それぞれの特徴について明らかにする予定である。

今回の実験では、各ピアが独自にコンテンツ取得要求を発生するため、一部ピアにコンテンツ取得要求や、コンテンツ転送といったタスクが集中することがあった。しかし、集中するタスクも、今回実験で用いたような 30 秒に 1 回コンテンツ取得要求を行う程度であれば、各ピアが処理可能な範囲であったため、特に大きな問題とはならなかつた。ユーザがウェブコンテンツ共有ネットワークを利用してコンテンツ検索を行う頻度は、普段検索エンジンなどを利用してコンテンツ検索を行う頻度と同等であると考えられるため、30 秒に 1 回程度というのは十分に高い頻度であると考えられる。しかし、今回は 10 ピア程度の小規模なネットワークで運用実験を行ったので特に問題にはなかつたが、ネットワークの規模が 100 台、1,000 台、10,000 台と増えていくと、負荷はさらに増大する可能性がある。ネットワークの規模の変化による性能の変化については、シミュレーション実験などによりある程度明らかにできるが、シミュレーションでは理想的な値しか出てこない可能性があるため、実機による運用実験を行うことが重要であると考えられる。そこで、今後は実現したウェブコンテンツ共有システムをインターネット上で配布し、ボランティアを募ることにより、実機を用いた有用性の診断を行う予定である。

一方、今回の実験では URL 以外でのコンテンツ検索を行っていないため、類似関係にあるコンテンツの転送を行っていない。このとき、従属関係にあるコンテンツは、必ず同時にアクセスされるため問題はないが、接続関係にあるコンテンツや、類似関係にあるコンテンツを転送し、そのコンテンツが利用されなかつた場合、無駄にトラフィックを増やしてしまうことに繋がる。この問題については、コンテンツの相関度をより細かく設定することで対応できると考えられる。例えば、同じコンテンツから接続関係にあるコンテンツでも、同サイトにあるコンテンツに比べ、他サイトにあるコンテンツへのアクセスは一般的に低いといえる。ウェブアクセスの特性を考慮し、コンテンツの相関度をコンテンツ内のリンクの数や、リンク先の情報、リンクの表示位置により計算することにより、無駄なコンテンツの転送を低減できると考えられる。コンテンツ相関度の計算については今後の課題とする。

これまで実現してきたウェブコンテンツ共有システムでは、P2P 型ネットワークを作成する際に、ピア間の物理的なネットワークにおける距離を考慮していない。物理ネットワークの距離は、P2P のファイル交換において大きく性能に関わる。ピア間のネットワーク距離が遠い場合は、ピア間のネットワーク距離が近い場合に比べ、問合せなどにおいて余計に時間がかかる。もし、あるネットワーク距離が短い二つのピアが、それぞれから見てネットワーク距離の遠いピアを介して接続している場合、

問合せなどにおける時間の無駄になってしまう。そこで、今後は、ウェブコンテンツ共有ネットワークを構成する際に、ピア間の RTT (Round Trip Time) などを計算し、ネットワーク距離の近いピアがなるべく集中するような最適なネットワークを構築する予定である。

7. ま と め

本稿では、ウェブコンテンツ共有システムを利用したコンテンツ共有におけるコンテンツ取得の待ち時間を短縮するため、コンテンツ間の相関性を考慮した分散キャッシング手法を提案し、キャッシング手法を用いたシステムを実現した。また、これまでに実現した能動的キャッシング手法と、本手法を利用した実測実験を実施することにより、実際にウェブコンテンツ共有ネットワークにおけるコンテンツ取得までの時間を短縮できていることを明らかにした。今回実現したキャッシングシステムでは、各ピアにおける計算処理量や、ピア同士でやり取りするトラフィックが増加してしまう。しかし、実測実験からも明らかのように、その計算負荷やトラフィックの増加は大きな問題にはならないと考えられる。実現したコンテンツ共有システムは有効であると考えられる。

今回行った実測実験では、小規模なネットワークについてしか調査を行うことができなかつた。今後は、実測実験の結果を元にモデルを作成し、大規模な P2P 環境におけるシミュレーションを実施する予定である。また、大規模な P2P 環境における実測実験も行う予定である。

文 献

- [1] 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,” 情報処理学会論文誌, Vol. 45, No. 1, 2004.
- [2] 中村聡史, 塚本昌彦, 西尾章治郎, “ウェブ共有システムにおけるコンテンツ流通制御のための監視機構,” DICOMO2003 論文集, Vol. 2003, No. 9, pp. 725–728, 2003.
- [3] 中村聡史, 塚本昌彦, 西尾章治郎, “ウェブ共有環境におけるコンテンツ間の相関性を考慮したキャッシングについて,” 情報処理学会研究報告 (データベースシステム研究会 2004-DBS-132・放送コンピューティング研究グループ 2004 - BCCgr-7 合同研究発表会), Vol. 2004, No. 3, pp. 66–72, 2004.
- [4] larke, I., Wiley, B., and Hong, T. W., “Freenet: a distributed anonymous information storage and retrieval system,” *Proc. ICSI Workshop on Design Issues in Anonymity and Unobservability*, 2000.
- [5] “Gnutella,” <http://www.gnutella.com/>.
- [6] Roxio, Inc.: “Napster,” <http://www.napster.com/>.
- [7] “Winny,” <http://www.geocities.co.jp/SiliconValley/2949/>.
- [8] 総務省, “平成 15 年版情報通信白書,” 2003.