

P2P ネットワークにおける コンテンツの人気度を反映した複製配置

川崎 陽平[†] 佐藤 崇[†] 吉田 紀彦[†]

[†] 埼玉大学大学院理工学研究科

要旨 ピアツーピア (P2P) ネットワークの内でも、検索パケットのブロードキャストを行う Pure P2P ネットワークでは、過大なトラフィックが問題となっている。そこでこれを改善する目的で、コンテンツのアクセス頻度、すなわち「人気度」を考慮したコンテンツ複製の配置方式を提案する。具体的には、インデクス情報配置と協調させたコンテンツ複製配置の調整を核に、ノード内複製の LRU に基づく置換、検索パス上の複製位置の簡便な決定法も組み合わせることで方式を構築し、シミュレーション実験によって、従来の方式よりも総パケット数、ストレージ利用率とも優れていることを示した。

Popularity-based Content Replication in Peer-to-Peer Network

Yohei Kawasaki[†] Takashi Sato[†] Norihiko Yoshida[†]

[†] Graduate School of Science and Engineering, Saitama University

Abstract Among peer-to-peer (P2P) networks, pure P2P networks broadcast search packets, and cause excessive network traffic in particular. Addressing this issue, we propose a new strategy for content replication which prioritizes popular and attracting contents. Our strategy is based on replication adjustment, being cooperated with index caching, as well as LRU-based content replacement, and a more effective replica placement method than well-known probabilistic ones. We also present some experiment results to show our strategy is better than other ones in regards to network traffic and storage consumption.

1 はじめに

ピアツーピア (Peer-to-Peer, P2P) ネットワークは、互いに対等な立場にある不特定多数のノードから構成される、情報コンテンツの分散共有・配信のシステムである。P2P ネットワークでは負荷が各ノードに分散される反面、コンテンツが各ノードに分散しているため、利用者が求めるコンテンツの検索に工夫が必要となる。

検索方法としては、コンテンツの所在情報 (インデクス) を保持するインデクスサーバを用いる Napster [1] などの Hybrid P2P (Centralized P2P ともいう)、ネットワーク内で検索パケットのブロードキャスト (flooding) を行う Gnutella [2] などの Pure P2P (Decentralized

Unstructured P2P ともいう)、ネットワーク上に分散ハッシュ表 (Distributed Hash Table, DHT) を構成してコンテンツを高速に特定する Chord [3] などの方式 (Decentralized Structured P2P ともいう) などがある。

Hybrid P2P と DHT 型 P2P は検索が効率的であるが、Hybrid P2P ではインデクスサーバの負荷集中や障害への脆弱性が問題となり、DHT 型 P2P はネットワークの厳密な構造化を必要とする。一方で Pure P2P は、検索パケットの指数関数的な増加によるネットワーク負荷や、検索パケットの到達範囲しか検索できないことが問題となるが、容易にネットワークを構成できて耐故障性にも優れることなどから、現実には Pure P2P を全面的または部分的に取り入れたネットワークが多用されて

いる。

Pure P2P の上記の問題に対する有効な改善策の一つに、コンテンツの複製をネットワーク内に分散させて配置することがあり [4, 5]、すでに幾つかの研究もなされている。しかし、コンテンツ複製配置の研究で、「人気度」まで考慮しているものは少ない。一般にコンテンツのアクセス頻度、すなわち人気度は一様ではなく、必ず偏りがある。どのコンテンツも一様に複製配置の対象として扱うのではなく、人気度の高いコンテンツほど多くの複製を配置するという、人気度を考慮した複製配置によって、無駄な複製の配置を抑制するとともに、全体としての検索効率をさらに向上させることができるはずである。

本研究では、このような人気度を考慮した複製配置を Pure P2P ネットワーク上で分散的に実現する、スケラブルで実装の容易な方式を目指している [6]。以下、まず 2 で関連研究を概観する。そして、3 で本方式を説明し、4 で実験結果を示す。5 はまとめである。

2 コンテンツ複製配置

P2P ネットワークにおけるコンテンツ複製配置は、一般に検索が成功するまでのパケット数の減少、および検索成功率の向上をもたらす、したがって、ネットワーク負荷の削減、および検索の効率化が実現できる。これらの効果は、Pure P2P において特に顕著となる。

一般に Pure P2P では、あるノード（以下、Requester と呼ぶ）が検索を開始すると、検索パケットを自らの全ての隣接ノードに送り、受け取ったノードはそれを自らの全ての隣接ノードに転送する。そして、目的のコンテンツを保持するノード（以下、Holder と呼ぶ）に行き当たったら検索が成功する（ヒットするという）。そして、その検索パスを逆に辿って、コンテンツの所在を Requester に通知する。ここで、無限に探索が続くことを避けるために、検索パケットには転送（ホップ）の最大値が指定されており、転送の度に減っていく。

極論すれば、コンテンツの複製をすべてのノードに配置すれば、検索のための通信は不要となり、検索コストも含む参照コスト 0、検索成功率 100% という理想的な結果になる。しかし、ノードのストレージ資源は有限であることから、現実には適用できない。これを図式的に簡単に示したものが図 1 であり、1 つのコンテンツの複

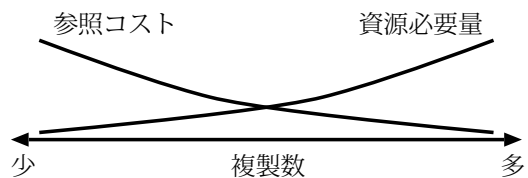


図 1 複製数と参照コスト・資源必要量の関係

製数に応じた参照コストとストレージ必要量の関係を示している。複製の個数を参照コストとストレージ必要量の 2 点から捉えたと、どこかに最適点が存在するはずである。

このように、コンテンツ複製配置においては、複製の個数と配置、すなわちどのノードに複製を配置するかが重要である。それを決定する方式について、すでに提案されているものを以下にまとめる。

(1) Owner Replication

検索がヒットした時に、Requester にだけ複製を配置する方式である。検索 1 回あたりに配置される複製が高々 1 個であるため、単純で、かつネットワーク負荷などのコストが最小ではあるが、複製がネットワーク内に広まるために十分な時間を要する [5]。

(2) Path Replication

Requester から Holder に至る検索パス上の全てのノードに複製を配置する方式である。1 度に複数の複製が配置されるため、コンテンツは広まりやすいが、必要なストレージやネットワーク資源が大きくなる [5]。

(3) Path Random Replication

検索パス上の各ノードにおいて、あらかじめ定められた配置確率に基づいて複製を配置する方式である。(2) Path Replication の改良となっているが、適切な配置確率を決定する必要がある [7]。

(4) Random Replication

検索パス上のノード数と同数の複製を、P2P ネットワーク上からランダムに選出したノードに対して配置する方式である。配置ノードをランダムに選出するためのアルゴリズムの詳細が定められておらず、余分なコストをかけずにノードの選出ができるかは定かでない [5]。

以上の他に、複製配置問題をネットワークの色分け問題に還元して解く方法も提案されているが [8]、ネットワーク全体のトポロジーをどのノードも把握できない P2P ネットワークに適用するのは難しいと考えられる。

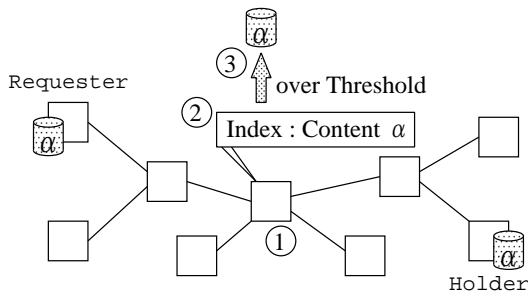


図2 複製配置までの手順

3 コンテンツの人気度に基づく複製配置

P2P ネットワーク上で、各ノードが検索・参照するコンテンツは、ネットワーク上の全てのコンテンツに対して一様ではなく、変動はあるにせよ、偏りがあるのが普通である。この偏りは、コンテンツの人気・不人気を意味する。

より人気の高いコンテンツにより多くの複製を配置することは、参照頻度の高いコンテンツについて検索成功率を向上させるとともに、目的のコンテンツへより少ないホップ数で到達しやすくするため、トラフィックの削減も期待できる。一方で、人気の低いコンテンツの複製数を抑制することは、ストレージ資源の消費を抑えることになる。

前節で概観した関連研究は、コンテンツの人気度を考慮することなく、一度でも検索があれば全て複製を配置するものがほとんどで、人気度の低いコンテンツについて、複製を生成するためのトラフィック (Owner Replication を除く) やストレージ消費に無駄が生じる。

数少ない例外の一つとして、Square-Root Replication [4] は、コンテンツへの要求回数に基づいて他コンテンツとの相対的な人気度の平方根に比例する個数の複製を配置する方式で、検索の効率化をもたらすことが示されている。Path Replication ないし Random Replication で Square-Root Replication を近似的に実装して行った実験 [5] では、全体の検索トラフィックを大幅に削減できることが確認されている。しかし、コンテンツの相対的な人気度を求めるためにネットワーク全体を把握する必要があり、そのままの実装が困難で、このような近似的な実装を採用する以外は現実的ではないと考える。

以上から、本研究で提案する複製配置の方式を、次の

ような基本方針に基づいて構築した。

- 基本的には関連研究と同様に、検索パス上に複製を配置することを考えるが、検索の度に即座に配置することはせず、配置するか否かを人気度に従って判断する。
- 各ノードのストレージは有限であることから、ノードがどのコンテンツの複製を保持するかも、人気度に従って判断する。
- Path Random Replication や Random Replication に見られるような確率的な配置は行わず、より簡便な方法を用いる。

以下に、コンテンツ人気度の分布に関する考察を述べた上で、人気度に基づく複製配置方式を説明する。図2はその概要を示したものである。

(1) コンテンツの人気度

一般にウェブ上の様々なコンテンツについて、人気度の評価として各々の参照回数を測定し、全コンテンツの結果をランク付けして並べると、いわゆる「べき乗則」(Power-Law) が成り立つことが明らかになっている \ citeweb-access。この結果は P2P ネットワーク上のコンテンツについても同様に成り立つと考えるのが自然である。

ただし、ネットワーク全体の状況をどのノードも把握できないことから、全コンテンツの人気分布が「べき乗則」に従うという性質を積極的に活用することは難しい。本研究では、人気度に偏りがあるはずという前提の裏付けとして考えるにとどめる。

(2) 複製配置の位置決定

複製配置の位置決定には、その作業に大きなノード負荷やネットワーク負荷を要しない簡便な、かつその後の検索を効率化できるような方法を用いる。具体的には、Requester は見つかったコンテンツを必ず取得することから、Requester に複製を置く Owner Replication を援用すると同時に、それに併せて、Requester から Holder からも最も離れた、検索パス上の中央のノードを複製配置位置とする (図2-①)。ただし、後述のように、すぐには置かない。

例えばその検索パス上で考えるとして、再びいずれかのノードが同じ検索を行った場合、1つの複製をどこに置くのが最適かを図3に示す。このように、中央に複製

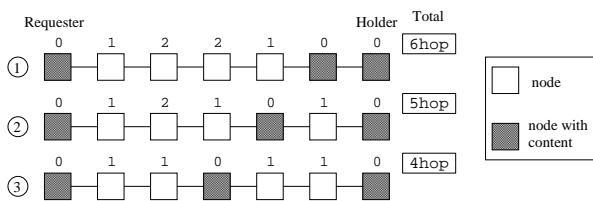


図3 検索パス上でのコンテンツアクセスホップ数

表1 インデクス情報の構造

検索キー	コンテンツ検索のためのキー
所在ノード (1)	コンテンツが所在するノードのアドレス情報
所在ノード (2)	同上 (Requester が複製配置をしたときに追加)
参照回数	このインデクスの参照回数

を配置すれば、検索ホップ数の合計、したがって期待値を最小にすることができる。これは、1つの複製を検索パス上に確率的に配置するよりも効果的である。

この検索パス上の中央ノードの検出は、次のように行う。検索パケットはノードを通過するごとにホップ数を1ずつ計数していき、Holderに到達した時の値を H_q とする。ヒットパケットが検索パスを逆に辿ってRequesterへと返る過程で、 $H_q/2$ (切上げか切捨てかは実装に依存)に位置するノードが中央ノードとなる。

(3) 複製の(仮)配置

上記のように複製配置の位置を決定したとして、そのノードに複製を即座には配置しない。一度検索があったからといって、その後に検索がないようでは、複製を配置するのは無駄になる。そこで、まずはインデクス、すなわちそのコンテンツをどのノードが保持しているかの所在情報だけを配置する。具体的には、HolderおよびRequesterから通知を受けることによってRequesterの情報を配置する。

インデクス情報は具体的には、検索キーワードとコンテンツ所在の対であって、コンテンツそのものに比べて概ね非常に小さく、配置や保持に要するコストは無視できる。その具体的な構造を表1に示す。

インデクス情報をネットワーク上に意図的に配置して検索を効率化することについては、以前から研究を進めてきており [10]、本研究でもそれを考慮している。ここで配置するインデクス情報は、それだけでも検索の効率化に寄与するとともに、仮配置として複製配置の予定位

表2 ノードのパラメータ設定

ノード数	2000
検索パケット TTL	5
初期コンテンツ最大保持数	60
ストレージ容量	100
複製配置閾値	15

表3 人気度ごとのコンテンツ数と検索要求率

人気度	検索要求率 (%)	コンテンツ数 (個)
(Low) 1	5	2500
2	10	1000
3	20	400
4	25	150
(High) 5	40	50

置を示す意味も持つ。このようにインデクス情報とコンテンツ複製の配置を連携させるところに、本研究の最大の特徴がある。

このインデクス情報にはカウンタを付加し、その後のインデクス参照回数を計る。そして、回数がある一定の閾値を越えたら初めて、実際のコンテンツの複製を配置する。インデクス参照回数が多いことは人気があること、およびそのノードがネットワーク内の要所的な位置にあることを意味し、以上の手順によって人気度の高いコンテンツを優先的に配置することができる。参照回数が少ない、すなわち人気のないコンテンツの複製は配置されないため、ストレージ資源の消費を抑えることができる。

なお、そのノードはコンテンツのインデクス情報としてHolderとRequesterの2つを持つので、参照に対してはラウンドロビンで交互のインデクスを返答として返すことにより、コンテンツへの参照を分散させる。

(4) 複製の置換え

ストレージが消費され一杯となったノードが、新たにコンテンツのオリジナルないし複製を保持しようとする際には、LRU (Least Recently Used) の原理で、最も古くかつ最も参照の少ない複製を破棄する。

4 実験と評価

提案方式の効果を実証するために、作成したシミュレータ上に仮想 P2P ネットワークを構築して実験を行った。

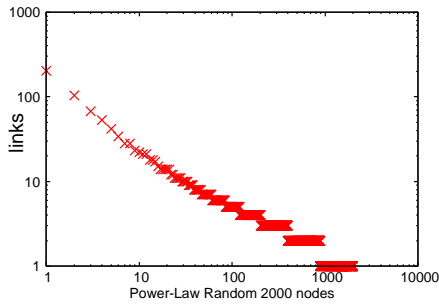


図4 リンク数分布の降順ソート

4.1 設定

各ノードには性能や回線容量に差は無いものとした。ノード数やノードの各パラメータの設定を表2に示す。

(1) コンテンツと検索の扱い

実験開始時に、各ノードには60個を上限としてランダムにコンテンツを持たせた。それらはストレージが複製で一杯になった場合でも破棄の対象としない。また、インデクスは各ノードとも1,000個を上限に保持できるものとし、溢れる場合にはFIFOで破棄することとした。

コンテンツの種類は計4,100個を用意した。コンテンツの人気度が3で述べたべき乗則にできるだけ近づけられるよう、人気度に5段階を設け、表3に示すように、それぞれに検索要求率を設定することによって、人気の差を作り出した。

いずれの実験においても、2,000台のノードからランダムに選んだ1台に1回の検索を行わせることを1回の試行とし、この試行を合計10万回実行した。結果は2,000回ごとの推移を示す。

(2) ネットワーク構成の扱い

実際のP2Pネットワークのトポロジー構成は、「べき乗則」の性質を持つことが報告されている[11]。すなわち、少数のノードが大多数のリンクを持ち、大半のノードはリンクが少ない。そこで、特に4.3に示す比較実験では、そのような構成を作成した。

実験におけるノード数2,000台について、縦軸を各ノードのリンク数として横軸はリンク数の降順にソートしたグラフが図4に示すようになるように、ノードとリンク数の分布を定めた。そして、ノード同士の接続はランダムであるPower-Law Random (PLR) トポロジーとした。

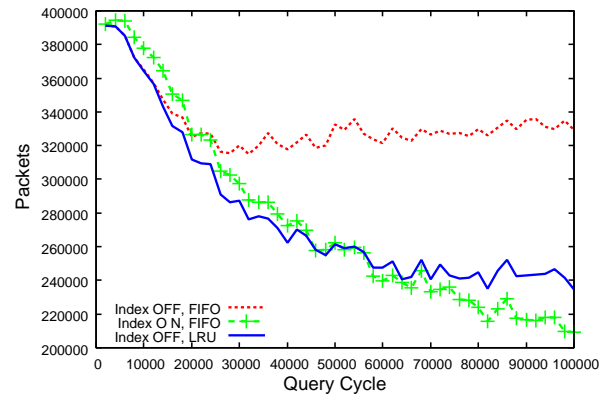


図5 総パケット数の推移

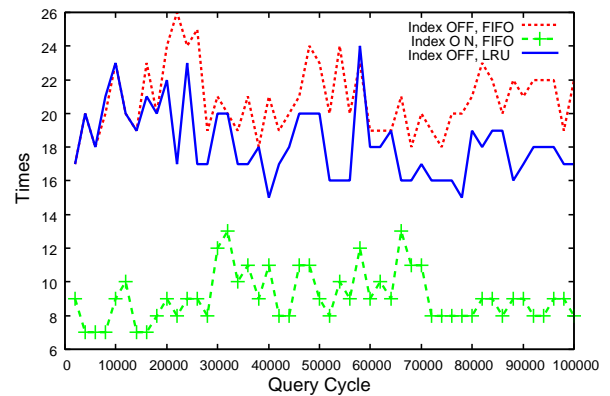


図6 コンテンツ提供最大回数の推移

4.2 実験1 - 提案方式における要素技術の効果

提案方式は3で述べたように、インデクス情報配置と連携させた複製配置の調整、およびLRUに基づくノード内の複製の置換、この2つの要素技術を協調させている。そこで、それぞれ単独の効果を明らかにするため、次の3つのケースを用意して比較を行った。なお、この実験ではネットワーク構成を単純化し、各ノードが2から4のリンク数でランダムに接続する構成とした。

- (1) Index OFF, FIFO 複製配置の調整もLRU置換も行わない。複製は即座に配置し、ノード内置換もFIFOとする。
- (2) Index ON, FIFO 複製配置の調整は行うが、LRU置換は行わない。
- (3) Index OFF, LRU 複製配置の調整は行わないが、LRU置換は行う。

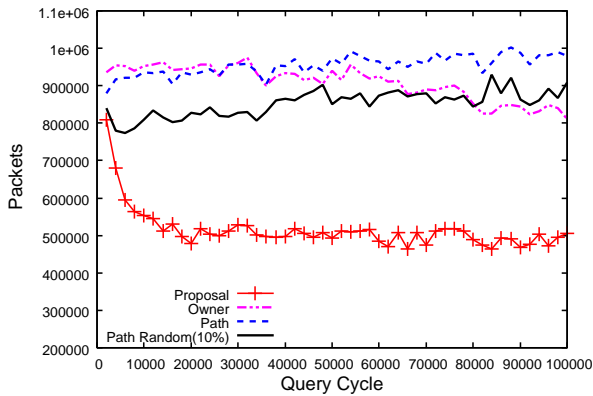


図7 各方式の総パケット数の推移

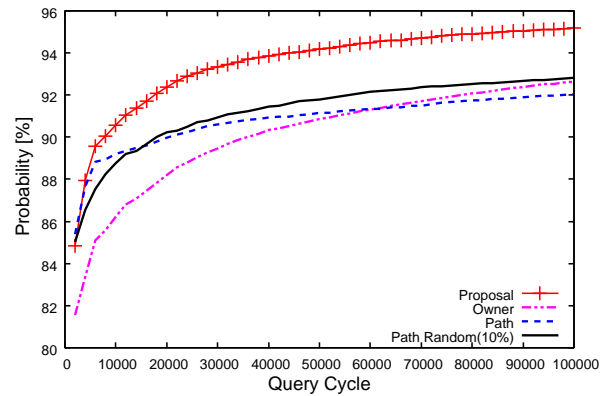


図9 各方式の検索成功率の推移

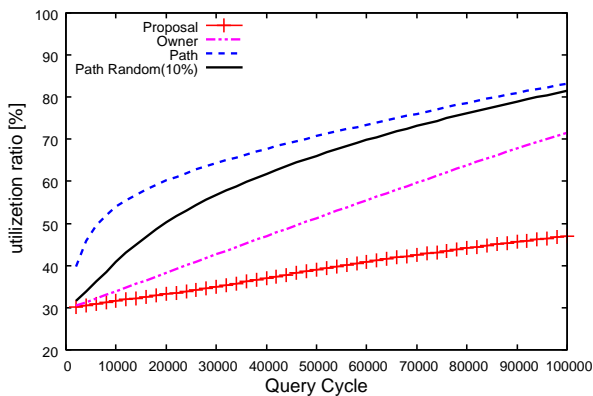


図8 各方式のストレージ利用率の推移

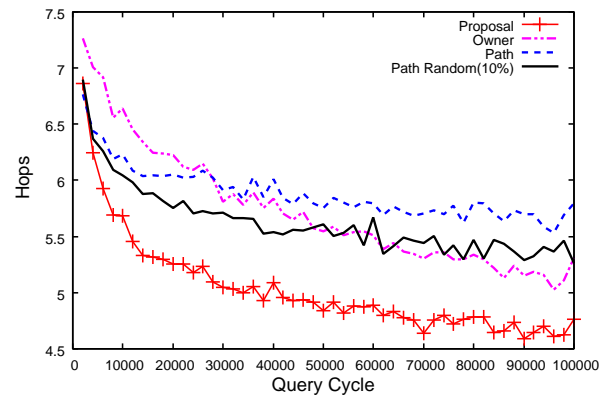


図10 各方式の平均ホップ数の推移

提案方式そのものの組み合わせとなる Index ON, LRU のケースでは, Index ON の効果により, 実験の範囲内では複製の置換が各ノードで頻発することはない, LRU 置換の効果が強く現れなかった. そのため, 得られた結果が (2) のケースとほぼ同一となったことから, 実験 1 では Index ON, LRU の結果は載せていない.

図 5 はクエリーパケットと検索ヒットパケットを合計したネットワーク内の総パケット数, すなわちトラフィックの時間的推移である. 各パケットがノードを通過した際にカウントした. (2) “Index ON, FIFO”, (3) “Index OFF, LRU” と, (1) “Index OFF, FIFO” より総パケット数の削減が達成できているのは明らかである.

子細に見ると, (2) のほうが検索 20,000 回付近までは総パケット数が多くなっているが, これは, インデクス情報が多数のノードで検索にヒットし, 結果として, ヒットパケットが増えたことによる. そこを越えると, 検索が

ヒットするまでのホップ数の減少が顕著になって, 逆転する. 一方で (3) は, ノード内置換が頻繁になってきたとみられる 12,000 回付近から違いが大きくなっている.

図 6 はコンテンツ提供最大回数, すなわち各ノードでコンテンツを提供した回数を計数して全ノード中の最大値を求めた結果の時間的推移である. この値が大きいことは, あるノードで負荷が突出していることを表す.

(3) の (1) からの違いはさほど顕著ではないが, (2) の違いは明白である. (1) に比べて, 最大値が 26 から 13 に抑えられており, これはインデクス情報の保持ノードでのラウンドロビンによって, コンテンツへの参照が一部に集中せずに分散されたことを意味する. なお, 図中には表れていないが, (2) では検索成功率が最も高く, コンテンツ提供回数も最も多いことを確認している.

4.3 実験 2 - 他の方式との比較

次に, 本研究での提案方式を他の方式と比較した. ネットワーク構成は, 先に述べた PLR トポロジーである.

- (1) Proposal 提案方式 .
- (2) Owner Owner Replication .
- (3) Path Path Replication .
- (4) Path Random Path Random Replication (配置確率 10%).

図 7 は各方式の総パケット数の推移の比較である . 提案方式のみが実験開始直後から急激な減少を見せ , 約 500,000 パケットで安定したが , その他の方式は終始開始直後のパケット数から大きな変動を現さなかった . PLR でなくランダムトポロジーを用いた実験では , 提案方式を含む全ての方式で総パケット数が減少していることから , ここでの結果は PLR トポロジーの影響が現れたものと考えられる . PLR トポロジーでは検索パケットがリンク数の多い , いわゆるハブのようなノードに集中しやすく , 特に多くの複製を配置する Path Replication と Path Random Replication では , ハブにおいて複製の置き換えが頻繁に起きる . ゆえに , FIFO による置換を行なう他の方式のほうが , LRU 置換を行い , インデクス情報においても検索がヒットする提案方式よりもハブにおける検索成功率が低く , 検索パケットをさらに拡散させてしまう .

また , 図 8 はストレージ利用率 , すなわちオリジナルも複製も含めたコンテンツの合計を全ノードの総ストレージ容量で割った値の推移の比較である . ここでも提案方式が最も低い値を示している . このように , 提案方式は , より少ないストレージ利用率でより少ないトラフィックを達成している . ストレージを消費していたずらに多くの複製を配置するのでは , 複製の生成に多量のデータ転送を必要とし , その意味でもコンテンツの人気を考慮することの効果は大きい .

総パケット数 , すなわちトラフィックには , 検索成功率と平均ホップ数も影響する . 図 9 は検索成功率の推移の比較 , 図 10 は平均ホップ数 , すなわち検索が成功した際のホップ数の全検索についての平均値 (複数ノードでヒットした場合は小さいほうを採用する) の推移の比較である . 前者は高いほど , 後者は低いほど , flooding による検索パケットの拡散を削減するため , トラフィック削減につながる . いずれも , 提案方式が最も優れた値を示している .

5 おわりに

本研究では P2P , 特に Pure P2P ネットワークにおける過大なトラフィックの問題を改善し , 検索効率を向上させる目的で , コンテンツの複製配置を取り上げた . そして , 従来に関連研究が , コンテンツのアクセス頻度 , すなわち「人気度」を考慮せず , 頻度の多寡にかかわらずに複製を配置するのに対して , 人気度を考慮した方式を提案した . そして , より人気の高いコンテンツにより多くの複製を配置することで , 参照頻度の高いコンテンツについて検索成功率を向上させるとともに , 検索ホップ数を減らすことでトラフィックの削減も果たし , 一方で , 人気の低いコンテンツの複製数を抑制することで , ストレージ資源の消費を抑制した .

具体的には , インデクス情報配置と協調させたコンテンツ複製配置の調整を核に , ノード内複製の LRU に基づく置換 , 検索パス上の複製位置の簡便な決定法も組み合わせることで方式を構築し , シミュレーション実験によって , 従来方式よりも , 総パケット数 , ストレージ利用率 , 検索成功率 , 平均ホップ数 , いずれについても優れていることを示した .

残された課題としては , 第 1 に , 複製配置の契機となるインデクス参照回数の閾値をいかに最適に決定するかがある . 第 2 に , 検索がヒットしたパス以外では「flooding」による過大なトラフィックの問題が依然として残っているので , その改善がある . 後者について , 現在は , 検索パケットの TTL を変化させて徐々に検索範囲を拡大していく Expanding Ring [5] の適用が有効であろうと考えている .

本研究の成果は , インターネット上のサーバに一時的に極度のアクセスが集中する Flash Crowds 現象に対抗すべく , 我々が別に研究を進めている P2P を応用した適応型 CDN とも関連があり [12, 13] , Flash Crowds でアクセス集中の対象となるコンテンツを複製拡散させる方式などへの展開についても , 検討を進めている .

謝辞

本研究の一部は科学研究費補助金 , 基盤研究 (B) 17300012 の支援を受けている .

参考文献

- [1] Napster website, <http://www.napster.com/>.
- [2] Gnutella website, <http://www.gnutella.com/>.
- [3] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, “Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications”, Proc. ACM SIGCOMM 2001, 2001.
- [4] E. Cohen and S. Shenker, “Replication Strategies in Unstructured Peer-to-Peer Networks”, Proc. ACM SIGCOMM 2002, 2002.
- [5] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, “Search and Replication in Unstructured Peer-to-Peer Networks”, Proc. 16th ACM Int’l Conf. on Supercomputing, 2002.
- [6] 川崎陽平, “ピアツーピアシステムにおけるコンテンツ複製配置の最適化”, 埼玉大学卒業論文, 2005.
- [7] 丸田大輔, 山本寛, 尾家祐二, “P2P ネットワークにおけるストレージ負荷分散実現のための複製配置手法”, 電子情報通信学会 技術研究報告, NS2003-319, IN2003-274, pp.131-136, 2004.
- [8] B.-J. Ko, and D. Rubenstein, “ Distributed Self-Stabilizing Placement of Replicated Resources in Emerging Networks”, IEEE/ACM Transactions on Networking, Vol.13, No.3, pp.476-487, 2005.
- [9] 佐藤進也, 風間一洋, 清水奨, “アクセス履歴を使用した Web サーバの状態の推定”, <http://www.ingrid.org/w3conf-japan/97/sato/paper.html> .
- [10] 吉田紀彦, 内田良隆, 檜崎修二, 瀬川淳一, 下川俊彦, “インデクスサーバの自律形成によるピアツーピアシステムの動的効率化”, 電子情報通信学会論文誌, Vol.J86-B, No.8, pp.1445-1453, 2003.
- [11] L. A. Adamic, R. M. Lukose, A. R. Puniyani and B. A. Huberman, “Search in Power-Law Networks”, Physical Review E, Vol.64, pp.46135-46143, 2001.
- [12] C. Pan, M. Atajanov, T. Shimokawa, and N. Yoshida, “Design of Adaptive Network against Flash Crowds”, 電子情報通信学会/情報処理学会 情報科学技術レターズ, Vol.3, pp.323-326, 2004.
- [13] C. Pan, M. Atajanov, T. Shimokawa, and N. Yoshida, “Flash Crowds Alleviation via Dynamic Adaptive Network”, インターネットコンファレンス 2004 論文集, pp.21-28, 2004.