

P2P ネットワークにおけるデータの使用頻度を考慮した 木構造に基づく複製更新伝播

渡辺 俊貴[†] 神崎 映光[†] 原 隆浩[†] 西尾章治郎[†]

[†] 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
大阪府吹田市山田丘 1-5

E-mail: †{watanabe.toshiki,kanzaki,hara,nishio}@ist.osaka-u.ac.jp

あらまし 近年の計算機の高性能化やネットワークインフラの発達により, Peer-to-Peer(P2P) ネットワークへの関心が高まっている. P2P ネットワークでは, 検索効率やデータ可用性の向上, 負荷分散のためにデータを複製し, ネットワーク上の複数のピアに配置することが有効である. しかし, データに更新が発生した場合, 複製を所持するピアに対して, 即座に更新情報を通知する必要がある. 筆者らの研究グループではこれまでに, 負荷分散と遅延減少を目的として木構造に基づく複製更新伝播法を提案している. この手法では, データのオリジナルを所持するピアを根とし, 複製を所持するその他のピアを内部節点とした n 分木の更新伝播木を構成し, この更新伝播木に沿って更新データを伝播する. ここで, 複製を所持する全てのピアに対して, 常に同等に更新データを伝播すると, 更新伝播時の負荷が増大してしまう. そこで本稿では, これまでに提案した更新伝播法を拡張し, 各ピアのデータのアクセス頻度に応じて伝播させるデータを変更する手法を提案する. 提案手法では, データのアクセス頻度の高いピアには更新データを伝播させ, データのアクセス頻度の低いピアには, 複製が古くなったことのみを通知する小さなメッセージを送信する. これにより, 更新データを伝播させるピア数を削減し, 更新伝播時の負荷や遅延を抑えることができる.

キーワード P2P, 複製, 更新伝播, アクセス頻度

Update Propagation Based on Tree Structure Considering Data Access Frequency in Peer-to-Peer Networks

Toshiki WATANABE[†], Akimitsu KANZAKI[†], Takahiro HARA[†], and Shojiro NISHIO[†]

[†] Dept. of Multimedia Eng. Grad. Sch. of Information Science and Technology, Osaka Univ.
1-5 Yamadaoka, Suita, Osaka

E-mail: †{watanabe.toshiki,kanzaki,hara,nishio}@ist.osaka-u.ac.jp

Abstract Recently, there has been increasing interest in research of data sharing in peer-to-peer networks because of high performance computers and advance of network infrastructure. In a P2P network, it is effective to replicate data items on multiple peers for efficient data retrieval, improving data availability, and load balancing. In such an environment, a data update occurred on a particular peer should be immediately propagated to other peers holding its replicas. In our previous work, we have proposed a novel update propagation strategy using a tree structure for delay reduction and load balancing. This strategy creates an n -ary tree whose root is the owner of the original data and the other nodes are peers holding its replicas, and propagates the update information according to the tree. However, since this strategy propagates the update information to all the replica holders every time an update occurs, the load of update propagation becomes large. In this paper, we extend our previous strategy to selectively propagate each update information considering the data access frequency of each peer. The extended strategy propagates the updated data to peers which frequently access the data, whereas, only a small message informing that the replica has become stale is propagated to peers which rarely access the data. This approach reduces the number of peers that receive the updated data of large volume, and reduces the load and the delay of update propagation.

Key words P2P, replica, update propagation, access frequency

1. まえがき

近年、計算機の高性能化やネットワークのブロードバンド化により、Peer-to-Peer(P2P) ネットワークを用いたデータ共有に関する研究が注目されている。ピア同士が対等な接続形態を取り、互いにサービスを提供しあう P2P ネットワークでは、検索効率やデータ可用性の向上、負荷分散のためにデータを複製し、ネットワーク上の複数のピアに配置することが有効である [2], [3]。一方、リアルタイムの天気予報やニュース速報、分散 Web コンテンツなどの分散ファイルシステムサービスでは、複数のピアで共有しているデータに更新が発生する環境が想定される。データに更新が発生した場合、複製を所持するピアに対して、即座に更新情報を通知する必要がある。筆者らの研究グループではこれまでに、負荷分散と遅延減少を目的として木構造に基づく複製更新伝播法を提案している [9]。この手法では、データのオリジナルを所持するピア (オリジナルノード) を根とし、複製を所持するその他のピアを内部節点とした n 分木の論理ネットワーク (更新伝播木) を構成し、この更新伝播木に沿って更新データを伝播する。

ここで、複製を所持するピアの中には、頻繁にその複製にアクセスするピアと、あまりその複製にアクセスしないピアが存在することが考えられる。これまでに提案した手法は、これらのピアに同等に更新データを伝播するため、あまり最新のデータを必要としていないピアにまで更新データを伝播し、余分な負荷が発生してしまう。

そこで本稿では、筆者らが文献 [9] において提案した複製更新伝播法を拡張し、各ピアのデータへのアクセス頻度に応じて更新データを伝播させるピアを変更する手法を提案する。提案手法では、データへのアクセス頻度が高いピアにのみ更新データを伝播させ、データのアクセス頻度の低いピアには、データが古くなったことを通知する小さなメッセージ (無効化情報) を送信する。これにより、更新データを伝播させるピア数を削減し、更新伝播時の負荷や遅延を抑えることができる。

以下では、2 章で関連研究について述べ、3 章で、本稿で提案する複製更新伝播法について説明する。4 章で提案手法の性能評価を行い、最後に 5 章で本稿のまとめと今後の課題について述べる。

2. 関連研究

P2P ネットワークサービスに関する研究分野では、データの複製を配置する研究は盛んに行われているが、複数のピアで共有されるデータに更新が発生する環境を考慮した研究は、あまり行われていない。しかし実環境では、データに更新が発生することが一般的である。このような環境では、複製を所持するピアに対する更新データの通知が遅れると、更新前の古いデータにアクセスしてしまう可能性が高くなる。そのため、複製を所持する全てのピアに対して、即座に更新データを通知する機構が必要となる。以下では、従来研究における複製更新伝播法について、その概要と問題点について述べる。

2.1 確率的な更新伝播法

Datta らは、P2P ネットワーク上の複製を所持するピアに対して、確率に基づいて更新データを伝播させる手法を提案している [3]。この手法では、複製を所持する各ピアは、同じ複製を所持するピアの情報をランダムに保持する。データを更新したピアは、同じデータの複製を所持するピアのうちのいくつかを隣接ピアとし、それらのピアに対してある確率 (更新伝播率) で更新データを伝播する。一度更新を伝播したピアは、部分リストと呼ばれるリストに加えられ、更新データと一緒に隣接ピアに通知される。更新データを受け取ったピアは、リストに存在しない隣接ピアに対して、更新伝播率に基づいて更新データを伝播する。この更新伝播率の値は、更新発生元ピアからの論理ネットワーク上のホップ数が大きくなるにつれて小さくなる。つまり、更新データが多くのピアに伝わるにつれて、隣接ピアに更新データを送る確率が低くなる。しかしこの手法では、全てのピアに更新が伝播される保証がない。さらに、1 つのピアに対して複数の経路を通して更新が伝播されることがあり、余分な負荷が発生する可能性がある。

2.2 チェイン構造を用いた更新伝播法

Wang らは、複製を所持するピアを一直線のチェイン上に配置する論理ネットワークを形成することにより、更新伝播時のオーバーヘッドを抑えつつ、ネットワーク耐性の向上を実現する手法を提案している [8]。この手法では、複製を所持する各ピアは、左右 m 個ずつのピア情報 (IP アドレスなど) を自身の調査ノードとして保持する。チェイン上のピアで更新が発生した場合、更新発生元ピアは、左右 m 個の調査ノードに更新データを送信する。それぞれの方向の m 個の調査ノードのうち、最も遠くにあるネットワーク上に存在する (オンライン) ノードを更新伝播責任ノードとし、更新データが送られてきた方向とは反対方向の調査ノードに対して、同様に更新データを伝播し、次の更新伝播責任ノードを決定する。以上の操作を、左右両方向のピアに対して繰り返すことにより、チェイン上の全てのオンラインノードに更新データを伝えることができる。しかしこの手法では、複製を所持するピアを一直線上に並べるため、ピア数が増えると、全てのピアに更新データを伝播するまでの遅延時間が長くなってしまふ。また、更新データを多くのピアに送信するピアと全く送信しないピアが現れ、更新伝播時の負荷の偏りが大きくなってしまふ。なお、文献 [8] では、任意のピアが更新を発生させる環境を想定しているのに対し、本論文で提案する手法では、オリジナルノードのみが更新を発生させる環境を想定している。

2.3 木構造を用いたマルチキャスト

P2P ネットワークにおいて、木構造を用いたマルチキャストに関する研究も盛んに行われている [4], [6], [7]。これらの研究では、データが木に沿って伝播される点で提案手法と類似している。しかし、これらの研究では、木に参加する各ピアが多くの情報を管理し、複雑な計算を実行することで、よりバランスの取れた最適な木を構築、維持することを目的としている。このような手法は、木構造維持に必要なコストが大きくなるため、複製が頻繁に置き換えられ、更新データを伝播させる環境には

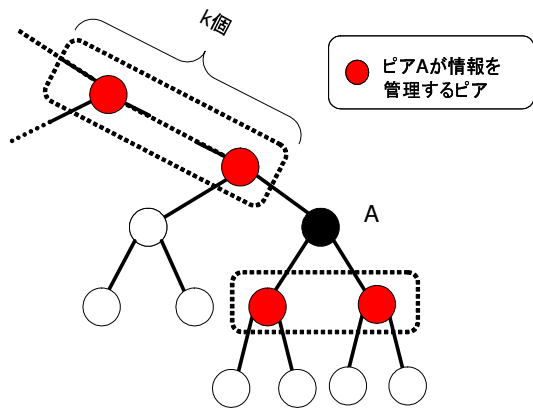


図 1 更新伝播木上のピアの管理情報

適していない。一方、提案手法では、各ピアが少数の情報を管理し、単純な操作を行うことで、可能な限りバランスの取れた木を構築、維持している。

2.4 木構造に基づく複製更新伝播法

筆者らの研究グループではこれまでに、更新伝播時の負荷分散と遅延減少、および障害耐性向上を目的とし、木構造に基づく複製更新伝播法を提案した [9]。この手法では、検索に用いるネットワークとは別に、オリジナルノードを根とし、複製を所持するピアを内部節点とした n 分木の論理ネットワーク (更新伝播木) を形成し、この更新伝播木に沿って更新データを伝播する。これにより、更新伝播時の負荷分散と遅延減少を両立する。また、図 1 のように、更新伝播木に参加する各ピアが、親の方向に対して k 個分の先祖ノード、および、子ノードの情報 (IP アドレスなど) を管理し、新規ピアの更新伝播木上での参加位置の決定や、複製削除などに伴うピアの脱退時における更新伝播木の再構成を自律分散的に行う。

この手法では、ネットワーク障害や機器の故障などによる、周辺ピアへの通知のない退出 (不当な退出) を考慮している。一般に、ピアがネットワークから不当に退出した場合、更新伝播木の修復を行うことができず、更新データを全てのピアに伝播することができなくなる可能性がある。更新伝播木が分断した場合に、検索用の論理ネットワークを用いてクエリをフラッディングさせ、データ (複製) を所持するピアを発見し、更新伝播木に再参加させる方法も考えられるが、大きなトラフィックが発生してしまう。そこで、この手法では、管理している k 個分の先祖ノードの情報を用いて、更新伝播木の分断が発生したときに、分断した箇所の周辺のピアのみで更新伝播木を修復する。そのため、少ないトラフィックで、最大 $k-1$ 個までの連続した先祖ノードの不当な退出まで修復が可能である。

3. データアクセス頻度を考慮した更新伝播法

文献 [9] で提案した木構造に基づく複製更新伝播法では、更新伝播時の負荷分散と遅延減少を実現し、高いネットワーク耐性を備えている。この手法では、データに更新が発生した場合に、複製を所持する全てのピアに更新データを伝播することを前提としている。しかし、複製を所持するピアの中には、その複製へのアクセス頻度が小さく、更新データを必要としないピア

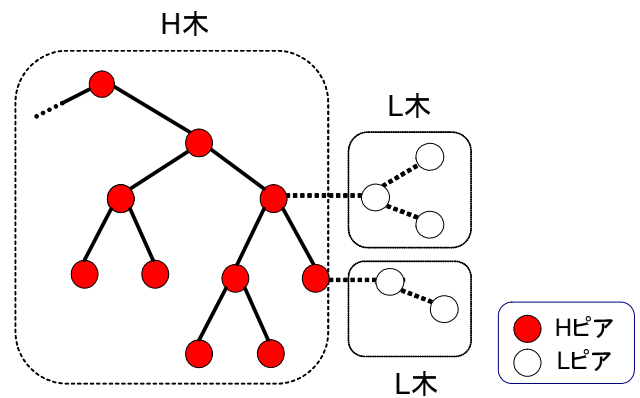


図 2 更新伝播木の構成

が存在することが考えられる。このようなピアに更新データを伝播しても、そのデータが利用されない可能性が高いため、更新データの伝播にかかる負荷が余分なものになってしまう。

そこで提案手法では、この木構造に基づく複製更新伝播法を拡張し、更新データを伝播するピアを制限することにより、更新伝播時の負荷や遅延をさらに抑制する。

3.1 更新伝播木の構成

提案手法における更新伝播木の構成例を図 2 に示す。提案手法では、複製を所持するピアを、データへのアクセス頻度が高く、常に最新のデータを必要とするピア (H ピア) と、データへのアクセス頻度が低く、あまり最新のデータを必要としないピア (L ピア) に分類し、H ピアで構成される更新伝播木 (H 木) と、L ピアで構成される更新伝播木 (L 木) に分けて論理ネットワークを構成する。また、1 つのデータに対して 1 つの H 木を構成し、L ピアは各 H ピアに付属する形で複数の L 木を構成する。さらに、複製を所持する各ピアは、更新発生時に最新のデータの必要性を調査し、適宜 H ピア、L ピアへの移動を行う。

3.2 ピアの管理情報

H ピアは、H 木における自身の上位 k 個の先祖ノードと子ノード、自身に付属する L 木の根ノード、および、その L 木に参加する L ピアの数に関する情報を管理する。一方、L ピアは、L 木における自身の親ノードと子ノード、自身が参加する L 木が付属している H ピア (責任 H ノード) の情報を管理する。

3.3 更新情報の伝播

提案手法における更新情報伝播の様子を図 3 に示す。データを更新したオリジナルノードは、H 木に沿って更新データを伝播させる。更新データを受け取った H ピアは、H 木における自身の子ノードに更新データを送信する。また、自身に付属する L 木の根ノードに、無効化情報を送信し、データが古くなったことを通知する。無効化情報を受け取った L ピアは、自身の所持する複製を無効化し、L 木における自身の子ノードに無効化情報を送信する。データを無効化した L ピアが最新のデータを必要とする場合、自身の責任 H ノードにデータを要求し、最新のデータを取得する。

このように、提案手法は、更新データを伝播させるピアを制限することにより、更新伝播時の負荷や遅延を抑えることができる。

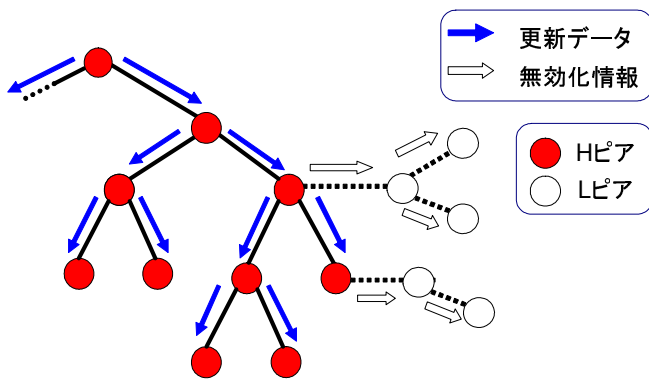


図 3 更新情報の伝播

3.4 新規ピアの参加

新たにデータを取得したピアは、自身の記憶領域に複製を作成し、そのデータの更新伝播木に参加する。このとき、新たに更新伝播木に参加するピア（新規参加ピア）は、まず H ピアとして H 木に参加するものとする。新規参加ピアの H 木への参加手順を以下に示す。

(1) データ取得元のピアが H ピアであれば、その H ピアがクエリに応答する。データ取得元のピアが最新のデータを所持する L ピアであれば、その L ピアがクエリに応答し、データと同時に L ピアの責任 H ノードの情報を返す。データ取得元のピアが無効化情報を受信した L ピアであれば、その L ピアのは自身の責任 H ノードの情報を返す。ここで、新規参加ピアの参加位置の決定を行うピアを責任ノードと呼ぶ。最初は、クエリに応答した H ピア、もしくは L ピアから通知された責任 H ノードが責任ノードとなる。

(2) 責任ノードは、自身がオリジナルノードでなければ、更新伝播木における自身から k 個上位の先祖ノード（先祖ノードが k 個未満の場合は根）にあたるピアに、子ノードの数 x を問い合わせる。 $x < n$ の場合は、新規参加ピアをその先祖ノードの子として参加させる。まず、新規参加ピアの親ノードとなるピアは、自身の子に関する情報として、新規参加ピアを追加する。一方、新規参加ピアは親ノードから $k-1$ 個上位まで（先祖ノードの数が $k-1$ 個未満の場合は根まで）の先祖ノードの情報を受け取り、親ノードと $k-1$ 個の先祖ノードの情報を、自身の先祖ノードの情報として記録する。

(3) 手順(2)によって新規参加ピアの参加位置を決定できなかった場合、責任ノードは、自身の子の数 y を確認する。 $y < n$ の場合は、新規参加ピアを自身の子とし、 $k-1$ 個上位まで（先祖ノードの数が $k-1$ 個未満の場合は根まで）の先祖ノードの情報を新規参加ピアに送る。また、責任ノードは、自身の子に関する情報に、新規参加ピアを追加する。

(4) 手順(3)において $y = n$ の場合、責任ノードは、自身の子ノードの中からランダムに 1 つを選択し、その子ノードを責任ノードとする。新たに責任ノードとなったピアは、(2)以降の手順に従って、新規参加ピアの更新伝播木における参加位置を決定する。

3.5 ピアの脱退

複製の置き換えなどにより、複製を削除するピアは、その

データの更新伝播木から脱退する。このとき、ピアの脱退により更新伝播木が分断されるため、H ピア、L ピアともに、以下の手順に従って、更新伝播木の修復を行う。ここで、更新伝播木から脱退するピアを脱退希望ピアと呼ぶ。

(1) 脱退希望ピアが更新伝播木の葉ノードの場合、自身の親ノードに木からの脱退を通知する。脱退通知を受けた親ノードは、自身のもつ子ノードに関する情報から脱退希望ピアを削除する。脱退希望ピアは、自身の先祖ノード（L ピアの場合は親ノード）に関する情報を全て削除し、脱退を完了する。

(2) 脱退希望ピアが子ノードをもつ場合、自身の子ノードの中からランダムに 1 つを選択し、脱退メッセージを伝える。脱退メッセージを受け取ったピアは、自身が葉ノードでない場合、同様の手順で子ノードを選択し、脱退メッセージを伝播させる。脱退メッセージを受け取った葉ノードは、自身と脱退希望ピアの位置を入れ替える。葉ノードの親ノードは、自身のもつ子ノードに関する情報からその葉ノードを削除する。

(3) 脱退希望ピアの位置に入れ替えられたピアは、新たな位置における親ノードと、 k 個下位の子孫ノードもしくは葉ノード（L ピアの場合は子ノード）まで、ピアが入れ替わったことを通知する。通知を受け取った各ピアは、自身のもつ子ノードおよび先祖ノード（L ピアの場合は親ノード）に関する情報に含まれる脱退希望ピアを、入れ替えられたピアの情報に変更する。その後、脱退希望ピアは、先祖ノード（L ピアの場合は親ノード）や子ノードに関する情報を全て削除し、脱退を完了する。

3.6 ピアの移動

4.1 節で述べたとおり、提案手法では、複製を所持する各ピアが、そのデータの必要性を調査し、H ピアと L ピアを適宜移動する。

3.6.1 移動基準の決定

複製を所持する各ピアは、自身および他のピアが自身の所持する複製にアクセスした回数の合計をデータアクセス回数として保持する。また、複製を所持している間に更新情報（更新データもしくは無効化情報）を受け取った回数を更新発生回数として保持する。各ピアは、更新情報を受け取ったときに、自身のデータアクセス回数と更新発生回数を比較し、H ピア-L ピア間を移動するかを決定する。具体的には、データアクセス回数が更新発生回数以上で、自身が L ピアである場合、H ピアへ移動する。一方、データアクセス回数が更新発生回数より小さく、自身が H ピアである場合、L ピアへ移動する。

3.6.2 H ピアから L ピアへの移動

L ピアへ移動することを決定した H ピア（移動ピア）は、まず自身に L 木が付属しているかどうかを調べる。L 木が付属している場合は、その L 木を他の H ピアに付け替えた後、移動ピア自身も L ピアとしてその L 木に参加する。以下に、移動手順の詳細を、図 4 を用いて説明する。

(1) 移動ピアは、図 4(a) のように、H 木における自身の子ノードに脱退メッセージを伝播させ、葉ノードと位置を入れ替える（移動ピアが葉ノードであった場合は親ノードに脱退を通知する）ことにより 3.5 節と同様の手順に従って H 木から脱

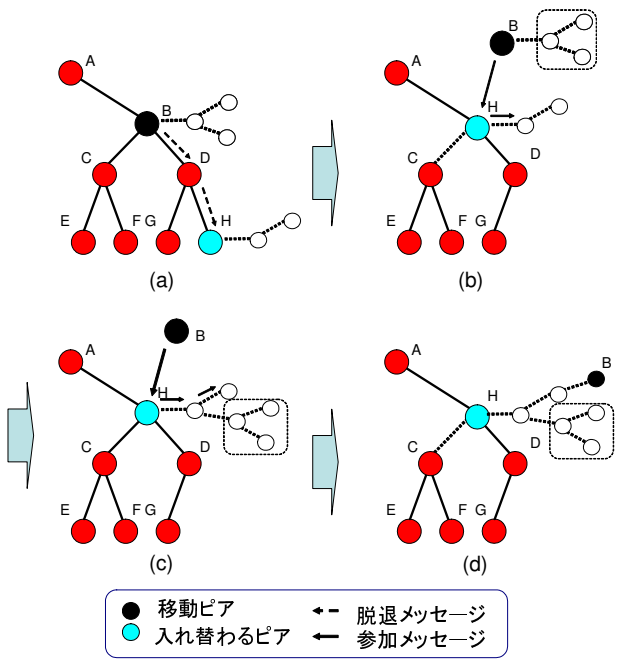


図 4 H ピアから L ピアへの移動

退する。

(2) H 木から脱退した移動ピアは、自身に L 木が付属している場合、図 4(b) のように、自身と位置を入れ替えたピア (移動ピアが葉ノードの場合はその親ノード) にその L 木の参加メッセージを送信する。参加メッセージを受け取った H ピアは、自身に L 木が付属していない場合は、受け取った L 木を自身に付属する L 木とする。自身に L 木が付属している場合、その L 木の根ノードに参加メッセージを送信する。その後、その L 木の根ノードを責任ノードとし、3.4 節と同様の手順に従って L 木に参加する。

(3) 自身に付属する L 木が存在しない、もしくは、自身に付属する L 木の移動を完了した移動ピアは、図 4(c) のように、自身と位置を入れ替えたピア (移動ピアが葉ノードの場合はその親ノード) に、L ピアとしての参加メッセージを送信する。参加メッセージを受け取った H ピアは、自身に L 木が付属していない場合は、移動ピアを自身に付属する L ピアとして参加させる。自身に L 木が付属している場合、その L 木の根ノードに参加メッセージを送信する。その後、その L 木の根ノードを責任ノードとし、3.4 節と同様の手順に従って L 木に参加する (図 4(d))。

3.6.3 L ピアから H ピアへの移動

H ピアへ移動することを決定した L ピアは、図 5(a) のように、L 木における自身の子ノードに脱退メッセージを伝播させ、3.5 節と同様の手順に従って L 木から脱退する。その後、図 5(b) に示すように、自身の責任 H ノードに参加メッセージを送信し、3.4 節と同様の手順に従って H 木に参加する (図 5(c))。

4. 性能評価

本章では、提案手法の性能評価のために行ったシミュレーション実験の結果を示す。実験では、非構造 P2P ネットワーク上でデータ共有を行う環境において、文献 [9] で提案した H

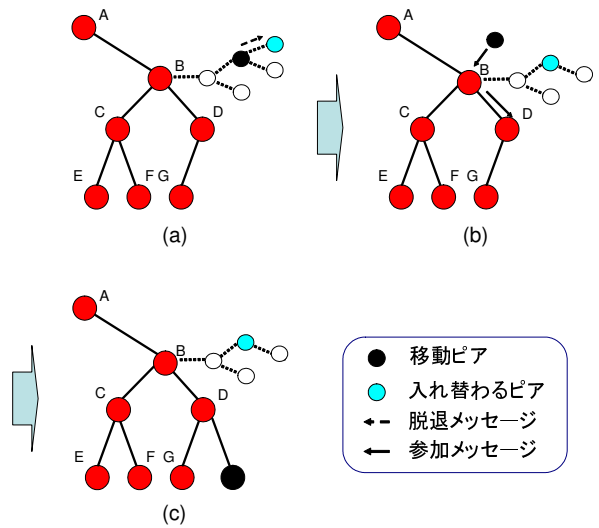


図 5 L ピアから H ピアへの移動

ピア、L ピアを分類せずに常に全てのピアに更新データを伝播する手法 (従来手法)、データアクセス回数よりも更新発生回数が大きい場合に L ピアに移動せず複製を削除する手法 (OnlyH 法)、2.2 節で説明したチェーン構造を用いた更新伝播法 (チェーン伝播法) ($m = 5$)、および提案手法の性能を比較した。

4.1 シミュレーション環境

シミュレーション実験では、P2P ネットワークに参加するピアの数を 1,000 とし、それらがべき法則 (Power-Law Random Graph: PLRG) [1] に従ってネットワークを構成するものとした。ここで、 i 番目のピアの隣接ピア数 d_i を以下の式で与え、一部のピアにリンクが集中する環境を実現した。

$$d_i = \lfloor 20 \cdot i^{-0.4} \rfloor \quad (1)$$

データの種類を 100 とし、全ピアのうち、ピア番号が 1 から 100 までのピアがそれぞれ、データ番号 1 から 100 のデータのオリジナルを所持するものとした。各ピアはそれぞれ、1 タイムスロット毎に 0.1 の確率であるデータを要求する。要求するデータ番号の分布は Zipf 分布に従うものとし、データ番号が小さいデータに対する要求ほど頻繁に発生するものとした。具体的には、 j 番目のデータの要求確率 q_j を、以下の式で与えた。

$$q_j = \frac{j^{-\alpha}}{\sum_{k=1}^{100} k^{-\alpha}} \quad (2)$$

上式において、 α はデータの要求頻度の偏りを決定するパラメータであり、Zipf 係数と呼ばれる。シミュレーション実験では、Zipf 係数は 0.5 とした。クエリの伝播には初期 TTL=1 のエキスパンディングリングを用い、データを発見できなかった場合は、TTL の値を 1 ずつ増加させて再度検索を行うものとした。各ピアは、クエリに回答したピアのうち、検索ネットワーク上のホップ数が最も小さいピアが所持するデータにアクセスするものとした。

複製の配置方式には、オーナー複製法 [5] を用いた。オーナー複製法では、データ要求が成功した際、クエリを発行したピアにのみ複製を配置する。更新伝播木は H 木、L 木ともに 2 分木 ($n=2$) とした。

各データのサイズは全て等しく、複製を保有可能な数は全てのピアで10(データのオリジナルを所持するピアは11)とした。各ピアが複製を作成する際にデータ記憶領域に空きがない場合は、所持していた複製の中で最も古い複製を削除し、新たな複製を作成するものとした。また、オリジナルデータは削除しないものとした。オリジナルノードは、1タイムスロットごとに一定の確率(更新発生率)でデータを更新するものとした。

以上の環境において、更新発生率を0.01~0.09まで変化させ、10,000タイムスロットが経過したときの、以下の値を評価した。なお、実験では、アクセス頻度が最も大きいデータ番号1のデータに注目して、このデータに関する評価値のみを採取した。

- 更新情報伝播負荷

更新発生時に、Hピアに更新データを伝播することによる負荷、および、Lピアに無効化情報を伝播することによる負荷のシミュレーション時間全体の合計。ただし、更新データのサイズを100、無効化情報のサイズを1とし、伝播されるデータの総量を負荷とした。

- 論理ネットワーク維持負荷

ピアの参加・脱退時に更新伝播用の論理ネットワーク(木構造、チェーン構造)を修復するために交換されるメッセージ、および、Hピア、Lピアへの移動時に交換されるメッセージの負荷の合計(シミュレーション時間全体)。このとき交換されるメッセージのサイズは、無効化情報と同じく1とした。

- 平均更新伝播遅延

更新伝播時の、H木の根からH木上の各節点までのホップ数の平均値。

- 検索成功率

対象データの検索成功率。

- 検索負荷

対象データ検索時における、エキスパンディングリングにより発生する検索メッセージの負荷(シミュレーション時間全体)。このときのメッセージのサイズも1とした。

- Hピアの負荷

各Hピアに付属するL木上のLピア(無効化情報を受信済)から、最新のデータを要求される回数(シミュレーション時間全体)。

ここで、チェーン伝播法は、複製を所持する全てのピアに更新データを伝播させるという点でUPT法と同じであり、更新情報伝播負荷、検索成功率および検索負荷の結果はUPT法と等しくなる。そのため、チェーン伝播法に関しては、論理ネットワーク維持負荷と平均遅延の結果のみを記載する。

4.2 評価結果

4.2.1 更新情報伝播負荷

更新情報伝播負荷を図6に示す。図6において、横軸は更新発生率を表し、縦軸は更新情報伝播負荷を表す。結果より、全ての手法において、更新発生率が増加するにつれて、更新情報伝播負荷が増大している。ここで、従来手法において、更新発生率とともに更新情報伝播負荷が線形的に増加しているのに対し、提案手法とOnlyH法では、負荷の増加量が小さく抑えら

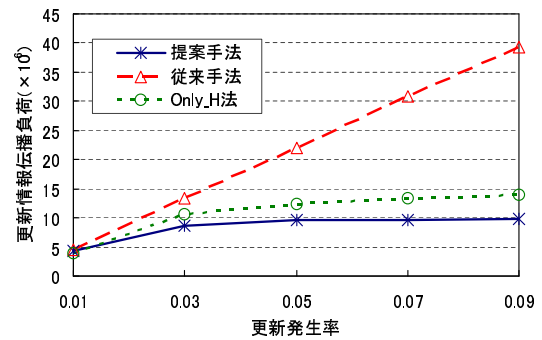


図6 更新情報伝播負荷

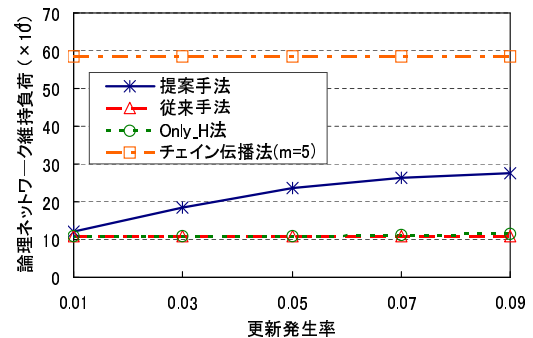


図7 論理ネットワーク維持負荷

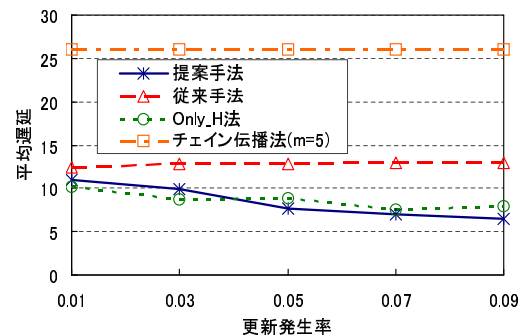


図8 平均遅延

れている。従来手法では複製を所持する全てのピアに更新データを伝播するため、更新発生率に比例して負荷が大きくなる。一方、更新発生率が大きくなるにつれて、提案手法では、HピアがLピアに移動する可能性が大きくなり、OnlyH法では、複製が削除される可能性が高くなる。そのため、更新データを伝播すべきピアの数が減少し、負荷が抑えられる。また、提案手法とOnlyH法では、提案手法における負荷の方がより小さく抑えられている。これは、提案手法ではLピアでありながら最新のデータを要求することが可能である一方で、OnlyH法では、最新のデータを要求したピアは必ずH木に参加しなければならないため、その分更新データを受け取る回数が増えると考えられるためである。

4.2.2 論理ネットワーク維持負荷

論理ネットワーク維持負荷を図7に示す。図7において、横軸は更新発生率を表し、縦軸は論理ネットワーク維持負荷を表す。結果より、提案手法では、更新発生率が増加するにつれて

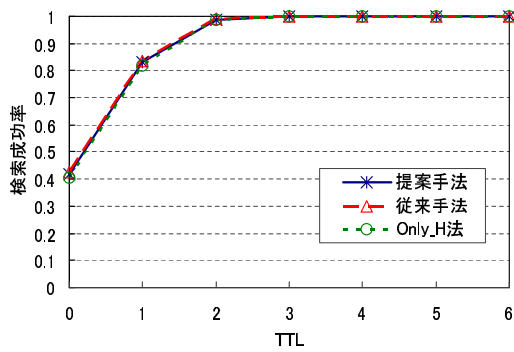


図9 検索成功率 (更新発生率: 0.01)

木構造維持負荷が大きくなるが、従来手法および OnlyH 法では、更新発生率とは関係なく、木構造維持負荷が小さい値で一定に保たれている。これは、提案手法ではピアの参加、脱退に加えて、更新発生時に H ピアと L ピアの入れ替えを行うため、他の手法に比べて各ピア間で交換されるメッセージ数が多くなるためである。一方、従来手法と OnlyH 法では、ピアの入れ替えは行わないため、各ピアで交換されるメッセージ数が少なくなる。また、更新発生率は各ピアの参加、脱退には影響を与えないため、木構造維持負荷は一定となる。チェーン伝播法では、参加、脱退の度に左右 m 個ずつのピアにメッセージを送信しなければならないため、木構造を用いた他の 3 つの手法に比べてチェーン構造維持負荷が大きくなる。

提案手法では、更新発生率が高くなるにつれて木構造維持に必要なメッセージ数が増加してしまう。しかし、図 6 の結果から分かるように、木構造維持負荷の増加量に比べ、更新情報伝播による負荷の減少量の方が大きく、全体の負荷は減少する。

4.2.3 更新伝播時の遅延

平均更新伝播遅延のシミュレーション結果を図 8 に示す。図 8 において、横軸は更新発生率を表し、縦軸は平均遅延を表す。結果より、従来手法に比べ、提案手法と OnlyH 法における遅延が小さくなっている。これは、従来手法では複製を所持するピア全てに更新を伝播しなければならないのに対し、提案手法および OnlyH 法では、更新データを送るべきピアが H ピアのみに限定されるためである。また、これら 2 つの手法は、更新発生率が高くなるにつれて、データアクセス回数よりも更新発生回数の方が大きくなるピアが増加し、更新データを送るべきピアの数がより減少する。そのため、更新発生率が大きくなるにつれて、遅延が小さくなる。チェーン伝播法では、左右 m 個ずつのピアに直線的に更新データを伝播させるため、遅延が大きくなる。 m の値を大きくすることで遅延は減少するが、その場合は前節で述べた論理ネットワーク維持負荷が増加してしまう。

4.2.4 検索成功率

更新発生率が低い環境 (更新発生率: 0.01) と高い環境 (更新発生率: 0.09) における、検索成功率を図 9, 図 10 に示す。図 9, 図 10 において、横軸は TTL を表し、縦軸は検索成功率を表す。なお、TTL=0 のとき、データを要求したピア自身が、そのデータのオリジナルもしくは複製を所持していた場合に、検

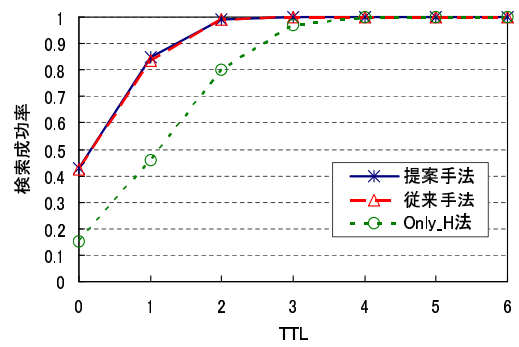


図10 検索成功率 (更新発生率: 0.09)

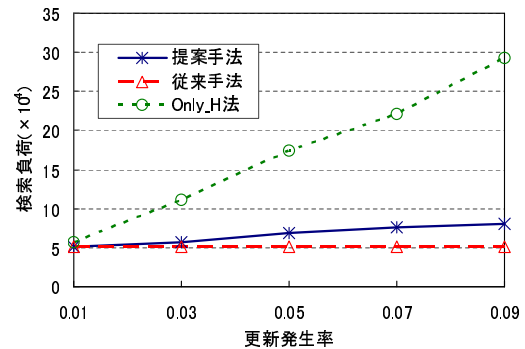


図11 検索負荷

索成功とする。また、提案手法において、L ピアは最新のデータを所持する H ピアへのリンクを所持しているため、L ピアを発見した場合でも検索は成功であるとした。まず、図 9 の結果より、更新発生率が小さい環境において、3 つの手法で検索成功率に大きな差が見られないことが分かる。これは、更新があまり発生しないため、更新発生回数がアクセス回数を上回るピアが少なくなるためである。これにより、提案手法における L ピアに移動するピアや、OnlyH 法における複製を削除するピアがほとんど存在せず、3 つの手法における更新伝播木の形状がほぼ等しくなるため、検索成功率に大きな差が生じない。

一方、図 10 の結果より、提案手法と従来手法では、複製をもつピア数が同じであるため検索成功率に差が見られないのに対し、更新発生率が大きい環境において、複製を所持するピアの数が少なくなる OnlyH 法における検索成功率が、他の 2 つの手法と比較して小さくなっている。

4.2.5 検索負荷

検索負荷を比較した結果を図 11 に示す。図 11 において、横軸は更新発生率を表し、縦軸は検索負荷を表す。結果より、従来手法では検索負荷が一定に保たれている一方で、提案手法と OnlyH 法では、更新発生率が大きくなるにつれて、検索負荷が増加している。これは、提案手法においては、L ピアを発見した場合、その L ピアの責任 H ノードにデータ要求を送る必要があるためである。また、OnlyH 法においては、図 10 の結果からも分かるように、更新発生率が大きくなると、TTL が小さいときの検索成功率が低下し、検索メッセージの再送回数が増加するためである。

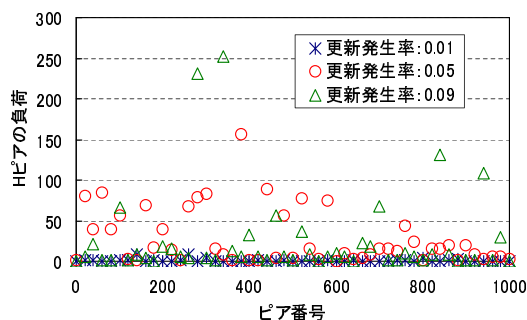


図 12 H ピアの負荷

4.2.6 H ピアの負荷

提案手法において、更新発生率を変化させ、H ピアの負荷を調べた結果を図 12 に示す。ここで、負荷を調べるピアは、1000 個の中から無作為に 50 個のピアを選択した。

図 12 において、横軸はピア番号を表し、縦軸はそのピアの負荷を表す。結果より、更新発生率が高い場合、1 つの H ピアが受け取るデータの要求回数が増える一方で、更新発生率が低い場合、受け取るデータ要求回数が少なくなっている。これは、更新発生率が高い場合、L ピアの割合が多くなり、1 つの H ピアに多くの L ピアが付属するためである。各 H ピアは、自身に付属する L 木上の L ピアからの最新データ要求に回答するため、付属する L ピアの数が多いほど H ピアの負荷が大きくなる。更新発生率が 0.09 の場合、極めて負荷の高い H ピアが存在するほか、更新発生率が 0.05 の場合でも、H ピアの負荷は全体的に大きくなっている。一方、更新発生率が 0.01 の場合、1 つの H ピアに負荷が集中することがなくなっていることが分かる。

ただし、本稿の実験環境を含め、一般的な環境では、H ピアの負荷 (L ピアからのデータ要求の負荷) よりも、更新伝播の負荷の方が支配的となる。L ピアの数 H ピアの数に比べて非常に多い場合など、特殊な場合を除いて、H ピアの負荷は現実的には大きな問題とならない。

5. あとがき

本稿では、P2P モデルを用いたデータ共有サービスにおいて、ピアが共有するデータ (複製) に更新が発生する環境を想定し、複製の更新にかかる遅延や負荷を抑制する複製更新伝播法を提案した。提案手法では、筆者らがこれまでに提案した複製更新伝播法を拡張し、データのアクセス頻度に応じて異なる更新伝播木を形成する。更新発生時には、データのアクセス頻度の高いピアには更新データを伝播する一方で、データのアクセス頻度の低いピアには無効化情報のみを伝播することで、更新伝播時の負荷や遅延の減少を実現する。

提案手法の性能を評価するために、シミュレーション実験により、データアクセス頻度を考慮せず複製を所持する全てのピアに更新データを伝播する手法、および、L ピアへの移動を行わず複製を削除する手法との比較を行った。その結果から提案手法は、更新伝播時の負荷や遅延を抑える一方で、検索時に必

要なメッセージ数が抑えられることを確認した。なお、本研究では、式 (2) の Zipf 係数の値を変化させることにより、異なるアクセス特性の環境においても同様の実験を行った。紙面の都合上、詳細については記載しないが、その結果、本稿で説明した $\alpha=0.5$ の場合と同様の結果が得られ、異なるアクセス特性の環境での提案手法の有効性が確かめられた。

本研究では、各ピアのデータへのアクセス特性、および、データの更新発生間隔が大きく変化しない環境を想定している。今後、各ピアのアクセス特性やデータの更新発生間隔などを動的に変化させた場合の適切な H ピアと L ピアの移動基準の検討を行う予定である。また、更新伝播用の論理ネットワークの維持負荷を軽減するために、L ピアを木構造以外の方法で維持する方法を検討する予定である。さらに、本稿の提案手法と筆者らがこれまでに提案した複製更新伝播法 [9] を組み合わせることにより、不当な退出が発生する環境における性能の評価についても行う予定である。

6. 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」、科学技術振興調整費「先端融合領域イノベーション創出拠点の形成：ゆらぎプロジェクト」、特定領域研究 (18049050) 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] L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman: "Search in Power-Law Networks," *Physical Review E*, Vol. 64, No. 4, pp. 46135–46143 (Sept. 2001).
- [2] E. Cohen and S. Shenker: "Replication Strategies in Unstructured Peer-to-Peer Networks," *Proc. SIGCOMM'02*, pp. 177–190 (Aug. 2002).
- [3] A. Datta, M. Hauswirth, and K. Aberer: "Updates in Highly Unreliable, Replicated Peer-to-Peer Systems," *Proc. ICDCS'03*, pp. 76–85 (May 2003).
- [4] J. Jannotti, D.K. Gifford, K.L. Johnson, M.F. Kaashoek, and J.W. O'Toole: "Overcast: Reliable Multicasting with an Overlay Network," *Proc. OSDI'00*, pp. 197–212 (Oct. 2000).
- [5] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker: "Search and Replication in Unstructured Peer-to-Peer Networks," *Proc. ICS'02*, pp. 84–95 (June 2002).
- [6] X. Tan and S. Datta: "Building Multicast Trees for Multimedia Streaming in Heterogeneous P2P Networks," *Proc. ICMCS'05*, pp. 141–146 (Aug. 2005).
- [7] D. A. Tran, K. A. Hua, and T. T. Do: "Zigzag: An Efficient Peer-to-Peer Scheme for Media Streaming," *Proc. INFOCOM'03*, pp.1283–1292 (Apr. 2003).
- [8] Z. Wang, S. K. Das, M. Kumar, and H. Shen: "Update Propagation Through Replica Chain in Decentralized and Unstructured P2P Systems," *Proc. P2P2004*, pp. 64–71 (Aug. 2004).
- [9] 渡辺 俊貴, 木戸 裕樹, 原 隆浩, 西尾 章治郎: "P2P ネットワークにおける障害耐性向上と遅延減少のための木構造に基づく複製更新伝播について," 情報処理学会マルチメディア, 分散, 協調とモバイルシンポジウム (DICOMO 2006) 論文集, Vol. 2006, No. 6, pp. 313–316 (July 2006).