DHTを用いたWeb情報共有手法

伊藤 嘉昭　志田 晃一郎　横山 孝典　兪 明連
東京都市大学大学院

1. はじめに

近年Webと連動するサービスが登場している．Webに情報をつけるGoogle Sidewiki[1]、ユーザのアクセスログからランキング作成をするPathtraq[2]などがある．そのようなサービスを実現するためにはユーザの数に応じてサーバを設置するなどの大きなコストがかかる．しかし，P2P技術を用いることでサービスの提供に必要なコストを減らすことができる．

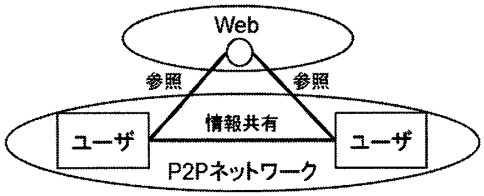

図1　Web情報共有

本研究では図1のように，Webページに対して情報を付加し，P2Pネットワーク上で共有する環境の実現法を提案する．

2. P2Pネットワーク

P2Pネットワークには非構造型P2Pと構造型P2Pがある．構造型P2Pでは分散ハッシュテーブル(DHT)を用い、キーをURLとすることで対応したノードを一意に決めることができる．よってDHTとWebは相性が良いといえるので，今回は構造型P2Pを用いる．

本研究ではDHTアルゴリズムのうちChord[3]を用いる．Chordは一次元で円状という単純なネットワーク構造をしている．ノードはノードIDを整数とし大小比較によって時計回りに昇順で点在する．そしてルーティング維持用にID空間上で自分より左回りに最初に存在するノードであるpredecessor，右回りに最初に存在するノードであるsuccessorを保持する．各ノードは(predecessor, self]の区間の(Key, Value)ペアを保持する．こうしてネットワークに参加している全てのノードでハッシュテーブルを分散保持する．Chordは効率的なルーティングの実現のためにルーティング専用のスキップリストであるfinger tableを持つ．finger tableは式(1)で表されるノードを格納する．

$$(ID_n + 2^k) \bmod 2^m (0 =< k < m). \qquad (1)$$

式中のID_nはネットワークにおけるノードのID，mは使用するハッシュ関数のbit数である．finger tableによって1回のホップで探索範囲を半分にすることができるため，効率的なルーティングを提供している．

Chordでは各ノードがハッシュ空間の一部分を担当しコンテンツを管理しているが，コンテンツのアクセス頻度は考慮されていない．よって，Webのような特定のコンテンツにクエリが集まるような状況では，負荷が偏ってしまう．

3. 負荷分散手法

DHTにおいてのアクセス頻度に応じた負荷分散手法を示す．複製を用いることでクエリの届く頻度の分散を目指す．この手法は自身が管理するコンテンツのクエリを閾値以上受け取った場合，複製を配置することで負荷分散を図る．このとき複製はキーの値の反時計回りに2^k進んだノードに配置する．ノードがクエリを受け取ったとき，そのクエリに対する複製を持っている場合，その場で返答処理を行う．

図2ではkey4のコンテンツのアクセス頻度が高いときの複製配置方法とノード7，12がkey4を持つノードを探索する場合を示す．

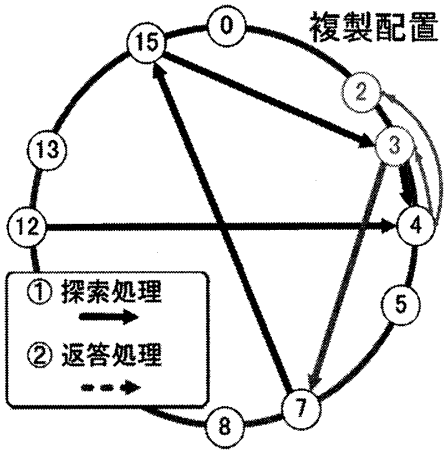

図2　複製配置方法

ノード4は管理しているコンテンツの一定時間内のアクセス頻度が閾値以上になった場合反時計回りに2^0進んだノードであるノード3へ複製を配置する．複製を配置したIDを覚えておき，アクセス頻度がまだ高ければ複製を配置したIDよりも戻った位置にいる2^k前のノードに複製を置く．ノード7，12が探索する場合はChord

A load balancing method for Chord
Yoshiaki Ito, Koichiro Shida, Takanori Yokoyama and Myungryun Yoo
Tokyo City University, Graduate School

のアルゴリズムにしたがって探索をおこなっているが，途中で複製があるため複製を持っているノードが返答をすることで探索を終了する．これによって途中の複製によってクエリを止めることができ，短時間に多くのクエリが流れるような環境でも負荷分散が図れる．

また複製元ノードは複製を持つノードに定期的に通信をおこなう．Valueに変更がある場合複製を更新する．複製元ノードが複製を持ってるノードの離脱を検知したときは新たな複製を配置することで負荷を分散する．

4. システム構成

Web情報共有にDHTを用いるシステム構成を図3に示す．提案するシステムではローカルなプロキシサーバを使うことでWeb情報共有にDHTを用いる．

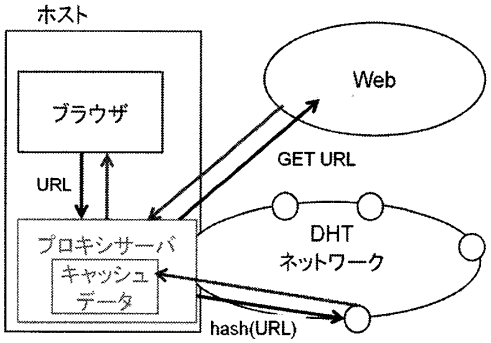

図3 Web情報共有にDHTを用いるシステム構成

プロキシサーバはDHTネットワークでノードの役割を果たす．ホストがWebを参照するとき，プロキシサーバに参照するURLなどの情報が届く．プロキシサーバはWebにそのまま中継するとともに，DHTネットワークからそのURLをキーとした情報を取得する．

また，キャッシュデータとして過去に自分が取得したデータを保存しておく．キャッシュデータはファイル名をhash(URL)，つまりそのURLをハッシュ関数にかけたものにする．ハッシュの値からURLを導き出すことはほぼ不可能であり，なおかつデータ自体もURLをキーとして暗号化することで，自身が担当したWebの情報もわからなくなるのでノードの平等性を確保することができる．参加するときはキャッシュデータをファイル名のハッシュ値に従って，DHTネットワークに書き込む．これによって，DHTネットワーク上の情報がなくなってしまったとしても，ある程度は復旧することができる．

5. 負荷分散評価

シミュレーションによって負荷分散の評価を行った．ノード数1000，コンテンツ数10000，クエリ発生回数1ノードあたり10回，1回のクエリの発生で時間を1進める．コンテンツのアクセス頻度は「出現頻度を1位，2位，3位，と順位をつけていくとき，順位をn倍すると頻度は$1/n$倍になる」というZipfの法則に従う．また、提案手法では過去1000時間でのアクセス回数が30，50を超えているときに複製を配置した．これらの条件で過去1000時間での負荷のばらつきを計測する．また，負荷分散の指標としてBalance Index[4]を用いた．Balance Indexは式(2)で表わされるδの値であり，1に近づくほど負荷が分散していることを示す．

$$\delta = \frac{\left(\sum_{i=1}^{N} X_i\right)^2}{\left(N\sum_{i=1}^{N} X_i^2\right)}. \quad (2)$$

ただしXiはノードiのアクセス数，Nは全ノード数を表す．

Balance Indexの時間による変化を表わしたグラフが図4である． Chordに比べ，提案手法では時間がたつ

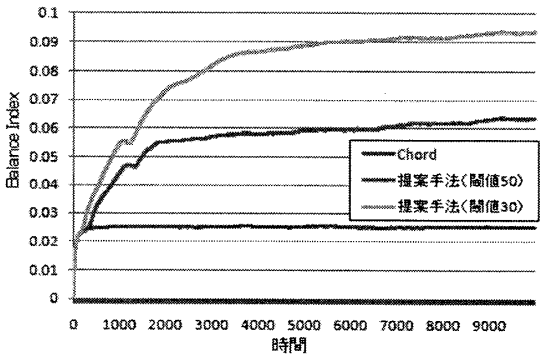

図4 Balance Indexの時間による変化

につれて負荷が分散していることがわかる．また，最終的に閾値30の時は複製が70個，閾値50の時は複製が29個となった．閾値が低い方が負荷分散効果は高くなっているが，複製の数が増えてしまうため複製の更新に時間がかかってしまう．よって，古い情報を得てしまう可能性が高くなる．

6. おわりに

Webページに対して情報を付加し，P2Pネットワーク上で共有する環境の実現法を提案した．Webのような特定のコンテンツにクエリが集まるような状況での負荷分散手法の効果をシミュレーションによって確認した．

参考文献

1) Google Sidewiki, http://google.com/sidewiki
2) Pathtraq, http://pathtraq.com/
3) I. Stoica, R. Morris, D. R. Karger, M. F. Kaashoek, and H. Balakrishnan: “Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications”, In Proc. of ACM SIGCOMM 2001, pp. 149-160 (Aug. 2001).
4) Chiu, D. and Jain, R, “Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks,” Computer Networks and ISDN Systems, vol.17, pp.1-14, 1989.