

修士論文概要書

CD

2008年 2月提出

学籍番号 3606U053-6

| | | | | | |
|-----------|------------------------------|----|-------|----|------------|
| 専攻名(専門分野) | 情報ネットワーク専攻 | 氏名 | 高木 邦孝 | 指導 | 甲藤 二郎 教授 印 |
| 研究指導名 | 画像情報処理研究 | | | 教員 | |
| 研究題目 | DHTにおける負荷分散を目的とした複製配置手法の特性改善 | | | | |

1. はじめに

Peer-to-Peer(P2P) ストリーミングは,サーバの負荷を増加させることなくコンテンツを多人数に送信することが可能であり,近年注目されている技術である. また, P2P は非構造化 P2P と構造化 P2P の 2 種類に分類することができ, 本稿では構造化 P2P の一つである分散ハッシュテーブル(DHT)に着目する.

DHT では, システム共通の 1 つのハッシュ関数を用意し, ノードとコンテンツに一意的 ID が割り当てられネットワークを構築する. また DHT を用いることにより検索対象のコンテンツを保有しているノードを確実に発見することが可能である. しかし, 個々のノードの処理能力に関係なく, アクセス数が多いコンテンツを保持するノードに大きな負荷がかかる問題点がある. そこで, 以上の問題点を解決する複製配置手法について検討する.

2. 従来手法

文献[2]では非構造化 P2P における複製配置技術について述べられている. そして, Square-root Replication を提案し, 非構造化 P2P での理想的な複製数を式(1), (2)から算出できることを解析から示している.

$$R = \sum_{i=1}^D r_i \quad (1)$$

$$r_i = R \frac{\sqrt{f_i}}{\sum_{i=1}^D \sqrt{f_i}} \quad (2)$$

- D : 全データ数
- R : 全複製数
- r_i : データ i の複製数
- f_i : データ i への検索要求回数

また, 各ノードが保持するコンテンツ数の違いによる, 各ノードの処理の不公平性に着目し, Chord[1]において successor list のノードと保持するコンテンツを均等にして負荷分散を実現している手法がある. しかし, アクセス数の違いによる不公平性については解決できていない.

3. 提案手法

3.1 複製配置

本提案ではアルゴリズムに Chord を使用し, finger table を用いて複製を配置していく. 配置場所を図 1 に示す. 最初の複製を自分の $ID+2^{n-1}$ を管

理するノード B に配置し, その次の複製を自分の $ID+2^{n-2}$ のノード C に, その次を先ほど複製を配置したノード B の $ID+2^{n-2}$ を管理するノード D に配置する. このように配置していくことで, 複製を ID 空間全体に分布させることができる. また successor list を使用した場合に比べ, アクセスが ID 空間の一部に集中することを防ぐこともできる.

3.2 複製配置実行手段

オリジナルを管理するノード(図 1 ノード A)においてアクセス数が閾値を超えたら複製を配置する. アクセスの流れを表 1 に示す(閾値 = 10 としている). このようにすることで, アクセスの公平性の実現と, flash crowd のようなアクセスの集中に対応することができる.

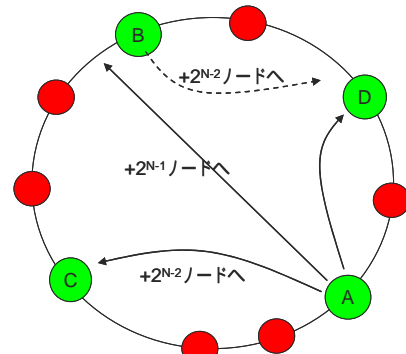


図 1 複製配置場所

表 1 アクセス推移

| ノード A | ノード B | ノード C |
|-----------------------|-------|-------|
| 1 | | |
| 2 | | |
| ... | | |
| 10 | | |
| ノード A が ノード B に 複製 配置 | | |
| 10 | 1 | |
| 10 | 2 | |
| ... | ... | |
| 10 | 9 | |
| 10 | 10 | |
| 11 | 10 | |
| 11 | 11 | |
| 12 | 11 | |
| ... | ... | |
| 20 | 19 | |
| ノード A が ノード C に 複製 配置 | | |
| 20 | 19 | 1 |
| 20 | 19 | 2 |

3.3 検索

従来通りの DHT アルゴリズムで行うが, 検索の際に最終目的地までの検索パス上に目的コンテンツがあれば, 検索を終える. そのようにすることで, 検索の際のホップ数を減らすことができる.

4. 評価

4.1 実験環境

PlanetSim を使い、ノード数を 1000、コンテンツ数 10000 とし、アルゴリズムに Chord を使用したとして Zipf の法則 (Zipf 係数 = 1.2) に基づくようにアクセスを 10000 回行った。また複製配置の際の閾値を 10 とした。

比較手法として、複製配置手法として代表的なリクエストしたノードに複製を配置する Owner Replication と、Square-root Replication を使用した。また、Square-root Replication での各コンテンツの複製数は提案の総複製数を元に算出した。そして、Square-root Replication、Owner Replication 共に最終アクセス先をオリジナル、複製含め公平とした。

4.2 結果

図 2 に複製数、図 3 アクセス数、図 4 にホップ数の結果を示す。また、Owner Replication の総複製数が約 10000 個、Square-root Replication と提案の複製数が共に約 150 個となり、Owner Replication の複製数が突出している。そして Owner Replication の場合、複製数が多いために、アクセスが最も分散できている。ただし、ディスク資源を浪費していることは自明であり、特に高精細映像のような大容量コンテンツの蓄積・配信に課題が残る。一方、Square-root Replication は、複製の総数は提案と同じであるが、人気の高いコンテンツの複製数が少ないために、Chord と同様

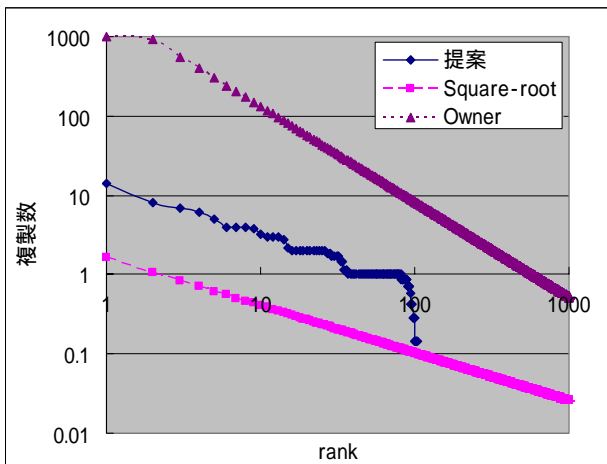


図 2 複製数の比較

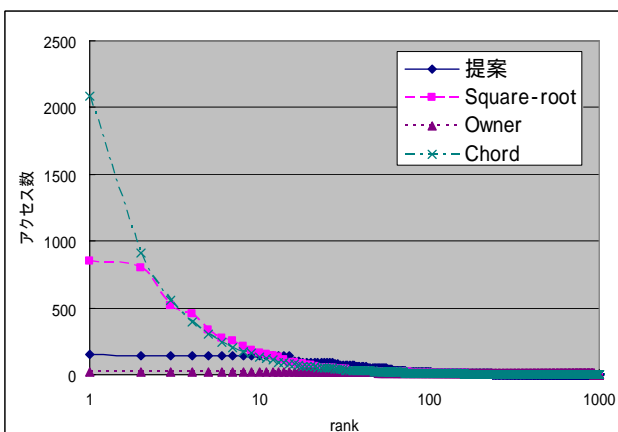


図 3 アクセス数の比較

に、特定ノードへのアクセスの偏りが観測される。これらに対して提案方式では、ディスク資源の節約、アクセスの負荷分散共に良好な特性を実現できていることがわかる。またホップ数は検索途中で目的コンテンツを発見する場合があるために、少なくなっている。

最後に、図 5 に、複製は提案と同様に行うが、複製場所を successor list にある順番通りとした場合のノード毎のアクセス頻度の比較を示す。successor list を用いた場合、アクセスが ID 空間の一部に集中しているが、finger table を用いた場合、アクセスが ID 空間全体に分散していることがわかる。

5. まとめ

本稿では DHT のネットワークにおけるアクセスの不公平さに着目し、アクセス数を元に複製を配置することで、アクセスの分散化を行った。今後は数学的な解析を行い、さらに理想的な複製配置手法や検索方法の改善などが考えられる。

参考文献・発表文献

- [1]Stoica, I., et al. "Chord: A scalable peer-to-peer lookup service for Internet applications" ACM SIGCOMM 2001.
- [2]E. Cohen and S. Shenker, "Replication Strategies in Unstructured Peer-to-Peer Networks", Proc. ACM SIGCOMM 2002.
- [3]高木邦孝, 蘇洲, 甲藤二郎 "DHT における負荷分散を目的とした複製配置手法" 電子通信情報学会 総合大会 March 2007

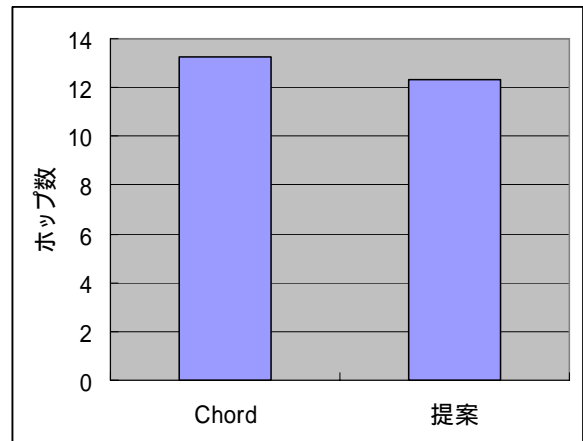


図 4 ホップ数の比較

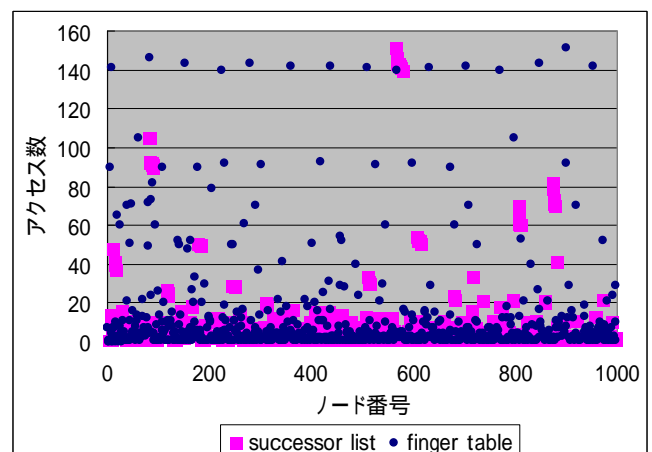


図 5 successor list 方式とのアクセス頻度の違い