

D-004

P2P 情報検索における DHT の階層化の提案

澤木 宏徳 †

† 立命館大学大学院理工学研究科

小柳 滋 ‡

‡ 立命館大学

1 はじめに

P2P ネットワークにおける検索手法として、従来までは Flooding 検索が用いられてきた。Flooding 検索とは Gunutella[1] などを用いられている手法で、各ノードは隣接するノードすべてにクエリを転送していく。Flooding 検索は実装が容易であるという一方で、ホップ数に対してクエリが極端に増大する。そこでクエリに生存時間 (TTL: Time-To-Live) を設けるのが一般的であるが、それでは確実な検索を行うことができない。

そこで近年、分散ハッシュテーブル (DHT: Distributed Hash Table) を用いた検索手法が提案されている。DHT 検索は高速かつ確実性の高い検索手法であるが、ノードの参加・離脱によって性能が低下する。

本論文では上記問題点を解決するため、RING 型の DHT アルゴリズムである Chord[2] の階層化を提案する。

2 既存アルゴリズム

2.1 DHT

DHT 検索においては各ノードがハッシュテーブルを持つ。ノードの IP アドレス・所有データが同一のハッシュ関数により、ネットワーク内の識別子 (ID) を与えられる。各データが保有されるノードは ID により一意に決定される。

2.2 Chord

Chord は、Ring のネットワークトポロジをもつ DHT である。各ノードは Successor List, Finger Table という 2 つの経路表を保有する。Successor List により Ring 型のネットワークが構築され、Finger Table により各ノードは検索を行う。各データはそのデータより大きく、最も近い ID をもつノードに所有される。

検索には図 1 のような Finger Table を用いる。 2^n のハッシュ空間において、Finger Table には $(\sum_{i=0}^{n-1} NodeID + 2^i)$ から最も近いノードへの経路が格納される。Finger Table をもとにクエリを転送し続けることで、検索効率を $O(\log N)$ にまで向上させる。

このように参加ノードが増加しても確実な検索を可能にする DHT だが、一方で参加離脱耐性の低さが指摘されている。新規ノードの参加・離脱により、各ノードの経路表を再構成する必要があるためである。DHT 検索においては、経路表を更新するために多くのメッ

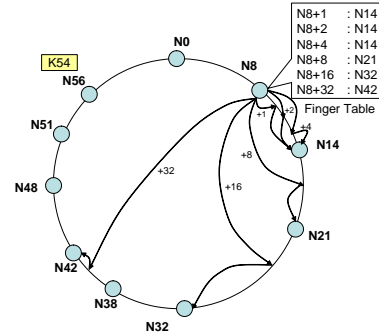


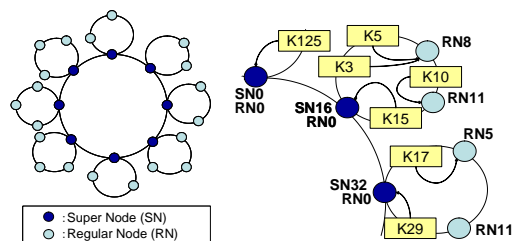
図 1: Finger Table

セージがノード間で交換される。再構成中の経路表の信頼性は非常に低下するため、検索要求が発生しても目的のデータを発見できないことがある。

3 提案機構

ノードの参加離脱が頻繁に起こりうる状況では、Chord では Finger Table の書き換えをしなければ検索できない。そこで本研究では参加ノードを Super Node (SN) と Regular Node (RN) という 2 つのノードに分けることで Finger Table の書き換え回数を減少させる、階層型の Chord を提案する。

3.1 基本アーキテクチャ



(a) ネットワーク

(b) キー管理

図 2: 階層型 Chord におけるネットワークとキー管理

SN と RN は図 2(a) のようなネットワークを形成する。SN は ID が近い RN 群の中から選出され、RN は SN の下で Ring 型のネットワークを形成する。それぞれのネットワークを SN ネットワーク、RN ネットワークと呼ぶ。両 Ring 型ネットワークは Chord のアーキテクチャを構築する。

The proposal of the hierarchical DHT method on P2P IR

†Hironori SAWAKI ‡Shigeru OYANAGI

†Graduated School of Science and Engineering, Ritsumeikan University

‡Ritsumeikan University

図 2(b) に、ノードと管理するキーの対応を示す。これは、SN ネットワークにおいて 2^7 のハッシュ空間を、RN ネットワークには 2^4 の空間を想定した際の図である。全ノードには SN 用の ID と RN 用の ID が割り当てられ、データはすべて RN の ID によって管理される。

各ネットワークは Chord のアルゴリズムで構築されているため、各 RN は RN ネットワーク用の Finger Table を持ち、SN は SN ネットワーク用の Finger Table を持つ。SN は SN ネットワークによって大域的な検索を行い、その後 RN ネットワーク内で局所的な検索を行う。SN は他の SN がもつ RN ネットワーク上の RN へのパスを持たないため、RN の参加離脱によって SN の Finger Table は書き換えられることがなく、DHT の問題点であった参加離脱耐性を向上させる。なお、全 RN は自身が所属する RN ネットワークの SN への経路を保有する。

3.2 データ取得までの流れ

データ検索要求の発生から取得までの基本的な流れを図 3 に示す。まず、ハッシュ化された検索単語をクエリとして検索を開始する。ノードは自身の所属する RN ネットワークの SN にクエリを送信する。クエリを受け取った SN は、SN 用 Finger Table によって SN ネットワーク内を検索する。最終的にクエリを受け取った SN は、RN 用 Finger Table によって自身の RN ネットワークにクエリを転送する。自身がクエリを管理するならば、クエリに対する検索結果を検索要求者に返信し、検索を終了する。

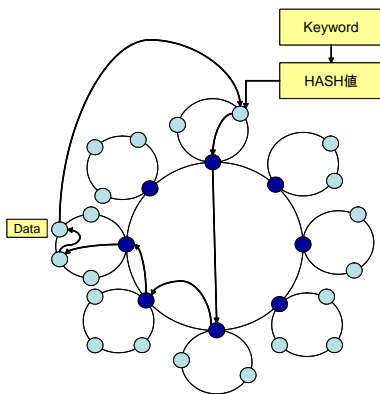


図 3: 階層型 Chord における検索の流れ

3.3 ノードの参加・離脱

ノードの参加・離脱におけるキーの管理について述べる。RN 用の ID によって保有するキーが決定され、RN 用 ID = 0 を管理する RN が SN となる。同様に、SN が参加・離脱する際には RN 用 ID = 0 を管理する RN とキーを授受し、RN/SN ネットワーク内のノードの経路表を書き換える。

4 評価と考察

2^{10} のハッシュ空間において 2^8 ノードがネットワークに参加している状況を想定する。実験として、ノードの参加離脱を 100 回ずつ繰り返し行った。

4.1 実験 1: Finger Table の更新回数

ノードの参加・離脱を繰り返し行い、その際に Finger Table を更新したノード数を計測する。結果、Finger Table が更新されたノードは、100 回の参加・離脱につき Chord では平均 1156.3 ノード、階層型 Chord では平均 514.61 ノードであり、約半分に減少した。以上より本研究の目的である経路表の更新回数の減少に成功、つまり参加離脱耐性を向上することができているといえる。

また階層型 Chord においては SN ネットワーク用と RN ネットワーク用という 2 つの Finger Table が存在するが、SN 参加・離脱時と RN 参加・離脱時の更新回数を計測すると、約半数が SN 更新時の SN/RN 用 Finger Table の更新であることがわかる。実験は全体の 25% のノードを SN として行ったため、RN の比率が増えればこの割合が減ると考えられる。

4.2 実験 2: ホップ数の比較

1 ノードから全キーに対して検索を行い、その際のホップ数を測定する。結果、Chord の平均・分散がそれぞれ 3.65, 1.81 であるのに対して、階層型 Chord の平均・分散は 3.87, 2.04 となった。

以上より階層型 Chord は既存手法の Chord より平均・分散ともに劣っていることがわかるが、その低下率は少ないといえる。

5 おわりに

ノード数を増加させて RN と SN の比率を変化させ、最適な比率を見つけることが必要である。また、現在は SN はハッシュ関数によってのみ決定されるため、帯域や接続時間の点で問題のあるノードが SN に選ばれる可能性がある。実際のネットワークに適用する上で、SN の選出手法を考える必要がある。

参考文献

- [1] Gnutella, <http://www.gnutella.com/>
- [2] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, Hari Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications", pp.149-160. ACM Press, 2001.
- [3] アンドリュウ, S. タネンバウム, 水野忠則・相田仁・東野輝夫・太田賢・西垣正勝訳, "コンピュータネットワーク 第 4 版", 日経 BP 社, 2003