

階層的クラスタを用いた DHT におけるノード数の偏りによる影響の排除

尾崎永径, 上田達也, 安倍広多, 石橋勇人, 松浦敏雄

大阪市立大学大学院創造都市研究科

1 はじめに

著者らは、ノード間のネットワーク距離を基にした階層的なクラスタリング^[1]を用いた DHT(以下林田 DHT)を提案している^[2]。しかし、この DHT では、ノード数の分布に偏りがある場合にホップ数が多くなるという問題点があった。本研究では、この問題に対処した DHT を提案する。

2 先行研究

2.1 階層的クラスタリング

文献 [1] では、インターネット上のノード集合をノード間のネットワーク距離を基に分割する手法を提案している。この手法では、図 1 のように親クラスタである C の下に子クラスタ A, B が所属しており、さらに A, B の下に子クラスタが所属しているような構造のクラスタを生成する(階層的クラスタ)。A, B のように同一の親クラスタに所属するクラスタ同士を兄弟クラスタと呼ぶ。

なお本稿では、親クラスタに含まれる子クラスタの数を最大 2 とする。

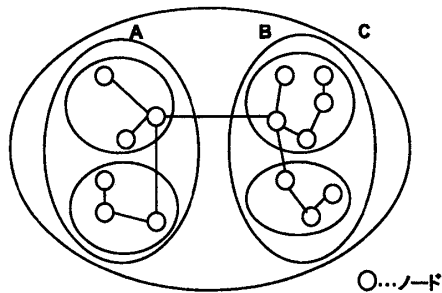


図 1: 階層的クラスタ

2.2 林田 DHT

2.2.1 データ構造

林田 DHT は Chord^[3] を基礎としている。各クラスタごとに 1 つの Chord リングを生成し、クラスタに所属するノードをリング上に配置する。各ノードは、一意なノード ID を保持し、所属するすべてのクラスタのリングに同時に参加している。各階層ではポインタとして successor, predecessor を保持し、

さらに最下層では finger table を保持している。

2.2.2 lookup

lookup は以下のように行う。まず検索するノードが所属する最下層のリングで Chord と同様のルーティングを行い、最下層でのルートノードを探す。ルートノード上に目的のデータが見つからない場合、階層を 1 段上がって predecessor を順に辿り、その階層でのルートノードを探す(図 2)。データが見つかるか、最上層のルートノードまで到達するまでこれを繰り返す。

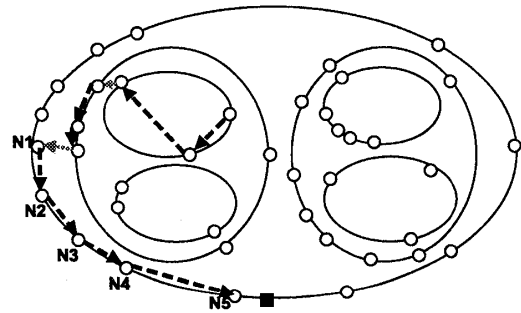


図 2: 林田 DHT での lookup

2.3 Crescendo

Crescendo^[4] は Chord を階層化したものである。データ構造は林田 DHT と似ているが、ノード N は以下の条件を満たす兄弟クラスタのノード N' への(複数の)ポインタを持つ。(1) ID 空間での N からの距離が 2^n 以上で最も近く ($n = 1, 2, \dots$), (2) N' の ID が N と N の successor との間にある。

3 提案手法

3.1 データ構造

本提案手法は林田 DHT に基づいているが、最下層より上の階層では predecessor, successor を廃し、代わりに以下で述べる Sibling Predecessor Set と Sibling Successor を追加する。

3.1.1 Sibling Predecessor Set

Sibling Predecessor Set (SPS) は兄弟クラスタに所属するノードへのポインタ集合である。図 3 のように、Level i のノード (N1) は、Level $(i-1)$ での predecessor (N1.pred) との間にある兄弟クラスタ上のすべてのノードへのポインタを保持する。これにより兄弟クラスタ間でノード数に偏りがあっても 1 ホップでそのレベルにおけるルートノードへ到

Elimination of Node Unbalance Effect in a Hierarchical Cluster based DHT

Hisamichi Ozaki, Tatsuya Ueda, Kota Abe, Hayato Ishibashi, Toshio Matsuura
Graduate School for Creative Cities, Osaka City Univ.

達できる。例えば、図2の場合、林田DHTではN1とN5の間を4ホップかかるところSPSを用いることで1ホップに削減できる。

なお、SPSの大きさは有限であり、その適切な大きさについては4.1で述べる。

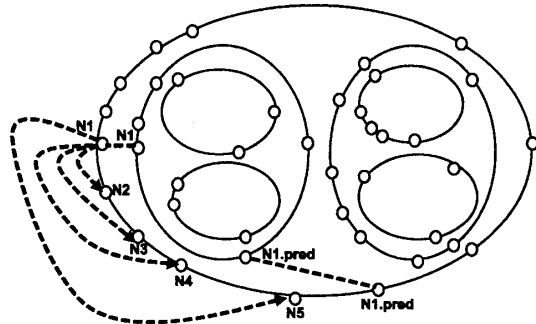


図3: Sibling Predecessor Set

3.1.2 Sibling Successor

Sibling Successor (SS) は、SPSとは逆向きのポインタである。SSは、SPSの維持のために使用する。

3.2 アルゴリズム

ここではlookupとjoinについて述べる。leaveとstabilizeは紙面の都合で省略する。

3.2.1 lookup

lookupは基本的には2.2.2節で述べた方法と同じであるが、階層を1段上がったあとSPSを用いて1ホップでルートノードへ到達する点異なる。

3.2.2 join

ノード(N)がDHTに参加する場合、最下層はChordのアルゴリズムで参加する。次に、successor(N.suc)は、自身のSPSからNの両隣りに位置するノード(図4のA、B)にNが参加したことを伝える。これにより、NとN.sucの間にあるノード(A)は、SPSにNを追加し(①)、間がないノード(B)はSSをN.sucからNに付け替える(②)。同時にNは、N.sucのSPSから自身のSPS、SSを作成する(③)。最後に、N.sucのSPSからN以降のノードを削除する。これを最上層まで繰り返す。

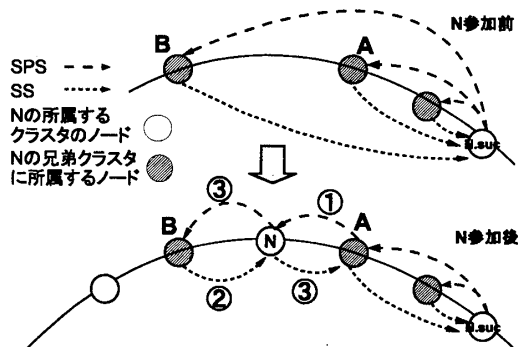


図4: SPS, SSの更新

4 考察

ネームサーバ間でのフルメッシュのネットワーク距離を測定したKing Data Set^[5]を用いて文献[1]のクラスタリングを乱数の種を変えながら10回行い、その結果に基づいて考察する。クラスタリングの条件はノード数1740、深さ10、子クラスタの最大数2である。

4.1 SPSの適切な大きさ

兄弟クラスタ間のノード数の比は最大278であった。ノードIDがChordリング内で必ずしも一様に分布していないことを考慮すると、SPSの大きさはこの比の数倍程度でよいと考えられる。

4.2 林田DHT・Crescendoとの比較

上の方法で生成したクラスタ上で林田DHT、Crescendoと提案手法を用いて最上位クラスタのルートノードまでのlookupのホップ数をシミュレーションにより調査した。林田DHTでは最大ホップ数が431.0、平均ホップ数が118.2、Crescendoでは最大ホップ数が26.1、平均ホップ数が13.8、提案手法では最大ホップ数が7.9、平均ホップ数が7.0となり、提案手法の有効性が確認された。

5 おわりに

本研究はインターネット上のネットワーク距離に基づいた階層的クラスタリングを利用し、ノード数の偏りを考慮したDHTを提案した。実験では兄弟クラスタ間のノード数の比が大きくなっているが、この点に関してはクラスタリングアルゴリズムの改良を含めて対応を検討中である。

謝辞 本研究の一部は科研費(19500058)の助成を受けたものである。

参考文献

- [1] 上田達也 ほか. P2P手法によるインターネットノードの階層的クラスタリング. 情報処理学会論文誌, Vol.47, No.4, pp. 1063-1076, 2006.
- [2] 林田紳也 ほか. インターネット上の距離の基づいた階層的クラスタを活用した分散ハッシュテーブル. 情報処全大 第3分冊, pp. 569-570, 2007.
- [3] I. Stoica, et al. Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications. In *Proc. of the 2001 ACM SIGCOMM*, pp. 149-160, 2001.
- [4] P. Ganesan, et al. Canon in G major: designing DHTs with hierarchical structure. *24th Int'l Conf. on Distributed Comput. Sys.*, pp. 263-272, 2004.
- [5] K. Gummadi, et al. King: Estimating Latency between Arbitrary Internet End Hosts. *SIGCOMM Internet Measurement Workshop*, 2002.