1A-6

ルーティングテーブルを利用した構造化 P2P ネットワーク推測手法

武田 敦志 †

† 東北学院大学教養学部情報科学科

1 はじめに

Chord#などの範囲検索を可能とする構造化 P2P ネットワークでは，動的に負荷を分散させる仕組みが必要不可欠である [1]．動的な負荷分散を行うためには P2P ネットワーク全体を推測する手法が必要であり，情報を集計するための P2P ネットワークを構築する手法 [2] や，ランダムに選んだサンプルノードから全体を推測する手法 [3] が提案されてきた．しかし，これらの手法には，集計対象となる情報ごとにネットワークを構築しなければならないという問題や，推測の結果が不正確であるという問題があった．

そこで，本稿では，ルーティングテーブルを利用して構造化 P2P ネットワーク全体の状態を推測する手法を提案する．提案手法では，各ノードのルーティングテーブルに近隣ノードの情報を関連付け，ルーティングテーブル更新時にこれらのノードの情報を集約することにより，P2P ネットワーク全体の状態を推測する．提案手法を利用することにより，構造化 P2P ネットワークの参加ノードの総数・登録オブジェクトの総数・最も高い負荷がかかっているノードなどを推測することが可能となる．本稿では，シミュレーションによる評価を行い，提案手法により P2P ネットワークの参加ノードの総数を正確に推測できることを示す．また，提案手法を利用することにより，従来手法よりも効果的な動的負荷分散が可能であることを示す.

2 構造化 P2P ネットワーク推測手法の提案

2.1 構造化 P2P ネットワーク推測手法の概要

本稿では，ルーティングテーブルを利用して構造化 P2P ネットワーク全体の状態を推測する手法を提案する．提案手法では，各ノードのルーティングテーブルに近隣ノードの情報を関連付け，ルーティングテーブル更新時にこれらのノードの情報を集約することにより，P2P ネットワーク全体の状態を推測する．提案手法を利用することにより，参加ノードの総数・登録オブジェクトの総数・最も高い負荷がかかっているノードなどを推測することができる．この章では，構造化 P2P ネットワーク Chord#に対して提案手法を適用し，ルーティングテーブルの構築法と推測のためのの情報集約法について述べる．

Estimation of Structured P2P Network with Routing-Table Aggregation
†Atsushi TAKEDA
†Department of Information Science, Tohoku Gakuin University

```
// update routing table
n.update ( ) {
  estimation = ( 1, count(n.objects), load(n) );
  node = n.successor;
  c_distance = distance(node);
  p_distance = 0;
  for (i = 0; p_distance < c_distance; i = i + 1) {
    n.routes[i].estimation = estimation;
    n.routes[i].node = node;
    estimation =
      aggregate(estimation, node.routes[i].estimation);
    node = node.routes[i].node;
    p_distance = c_distance;
    c_distance = distance(node);
  }
}
```

図 1: ルーティングテーブルの構築手順

2.2 ルーティングテーブルの構築

提案手法では，構造化 P2P ネットワークに参加している各ノードは以下の属性情報を有する．

$$
\begin{aligned}
\mathrm{node} &:= \ < \mathrm{id}, \mathrm{successor}, \mathrm{objects}, \mathrm{routes} > \\
\mathrm{id} &:= \ \mathrm{INTEGER} \\
\mathrm{successor} &:= \ \mathrm{node} \\
\mathrm{objects} &:= \ \{\mathrm{object}_0, \mathrm{object}_1, \mathrm{object}_2, \cdots\} \\
\mathrm{routes} &:= \ \{\mathrm{route}_0, \mathrm{route}_1, \mathrm{route}_2, \cdots\} \\
\mathrm{object}_i &:= \ < \mathrm{key}, \mathrm{value} > \\
\mathrm{route}_i &:= \ < \mathrm{node}, \mathrm{estimation} > \\
\mathrm{estimation} &:= \ < \mathrm{node\_count}, \mathrm{object\_count}, \\
&\qquad \mathrm{max\_load} >
\end{aligned}
$$

ここで，*id* は ID 空間上の位置・*successor* は ID 空間上の前方ノード・*routes* は直接通信するノード一覧・*objects* は管理しているオブジェクト一覧・*estimation* はノードの状態に関する情報である．図 1 にルーティングテーブル *routes* を構築するための手順を示す．構造化 P2P ネットワークでは，直接通信を行うノードを介して新たなノードの情報を取得する．この時，提案手法ではノードの状態に関する情報 *estimation* の集約を行い，この情報をルーティングテーブルに関連付ける．

2.3 構造化 P2P ネットワークの推測

図 2 に，指定された ID を管理するノードを検索する手順を示す．提案手法では，ノードを検索する時，検索を実施するノードから検索対象のノードまでの間に存在するノードの状態情報を集約する．この検索を，ID 空間上の後方に位置するノードまで行うことにより，構造化 P2P ネットワーク全体のノードの状態情報を集約

```
// search a node at specified id
n.search ( id ) {
  if ( n.isResponsible(id) ) {
    ret.node = n;
    ret.estimation = (0, 0, 0);
    return ret;
  }
  forward = n.routes[0];
  for (i = 1; i < count(n.routes); i = i + 1) {
    if ( distance(n.routes[i].id) < distance(id) ) {
      route = n.routes[i];
    }
  }
  result = route.node.search(id);
  ret.node = result.node;
  ret.estimation =
      aggregate(route.estiamtion, ret.estimation);
  return ret;
}
```

図 2: 推測情報を集約するための検索手順

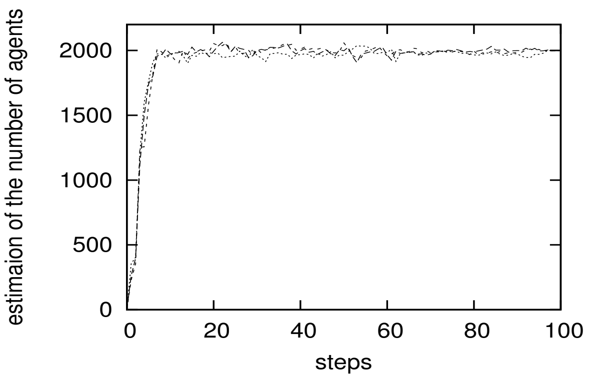

図 3: 各ノードが推測した参加ノード数の総計

できる．これにより，ネットワーク全体の状態を推測することが可能となる．

3 シミュレーション評価

提案手法の有効性を検証するため，P2P ネットワークシミュレータを Java を用いて実装した．このシミュレーションでは，提案手法を適用した構造型 P2P ネットワーク Chord#に 2000 個のノードが参加し，これらのノードが 20000 個の UNIX のファイルを分散管理している状況を想定している．また，各ノードは 1 ステップごとに図 1 に示した *update* を実行する．

図 3 に，提案手法を用いたときに各ノードが推測した参加ノードの総数を示す．この図では，ランダムに選んだ 4 個のノードによって推測された参加ノードの総数を示している．シミュレーション開始時，ノードの状態情報の集約が十分ではないため，参加ノードの総数の推測は不正確である．しかし，実行ステップの進行にしたがって情報の集約がなされるため，推測される参加ノードの総数は正確なものとなる．

図 4 に，提案手法を用いたときの動的負荷分散の効果と，従来手法であるサンプリングされた情報に基づ

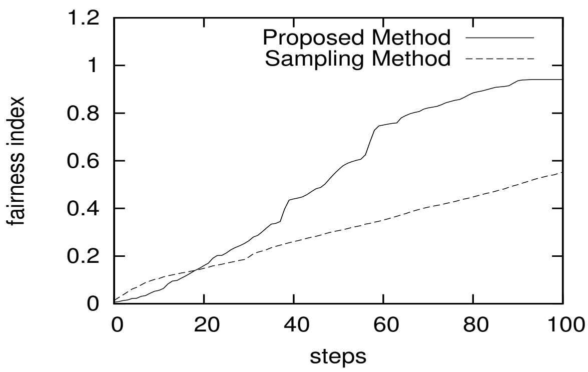

図 4: 提案手法を利用した動的負荷分散

いた動的負荷分散の効果を示す [3]．ここでは，評価指標として Faireness Index を用いる [4]．平均的に負荷が分散されている場合，Fairness Index は 1 に近づく．図 4 より，従来手法に比べて，提案手法の方が早く負荷分散できていることがわかる．これは，従来手法がランダムに選ばれたノードとのみ負荷分散していたのに対し，提案手法は集約された情報をもとに効果的に負荷分散していることを示している．

4 まとめ

本稿では，構造化 P2P ネットワークの全体の状態を推測する手法を提案した．提案手法では，ルーティングテーブルにノードの状態に関する情報を関連付け，この情報を集約することにより P2P ネットワーク全体の状態を推測する．提案手法により，参加ノードの総数・登録オブジェクトの総数・最も負荷の高いノードなどが推測可能となる．本稿では，シミュレーション結果より，提案手法を用いて参加ノードの総数を推測できることを検証した．また，提案手法を用いることにより効果的な動的負荷分散が可能となることを示した．

謝辞

本研究は文部科学省科学研究費補助金挑戦的萌芽研究 (90424001) の助成を受けて実施したものである．

参考文献

[1] Thorsten Schütt, Florian Schintke, and Alexander Reinefeld. Range queries on structured overlay networks. *Computer Communications*, Vol. 31, No. 2, pp. 280–291, 2008.

[2] Brighten Godfrey Karthik, Karthik Lakshminarayanan, Sonesh Surana, Richard Karp, and Ion Stoica. Load balancing in dynamic structured p2p systems. *Proceedings of INFOCOM 2004*, Vol. 4, pp. 2253–2262, 2004.

[3] Ashwin R. Bharambe, Mukesh Agrawal, and Srinivasan Seshan. Mercury: Supporting scalable multi-attribute range queries. *Proc. ACM SIGCOMM*, Vol. 34, pp. 353–366, 2004.

[4] Rajendra K. Jain, Dah-Ming W. Chiu, and William R. Have. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. *DEC Research Report TR-301*, 1984.