

## P2P 環境下における BSP 探索手法

由利 誠† 能登谷 淳一†† 草薙 良至†† 笠井 雅夫††

† 秋田県立大学システム科学技術研究所 †† 秋田県立大学システム科学技術学部

## 1 はじめに

本研究では、P2P 環境下において BSP 探索問題を高速に解決する手法を提案する。従来、P2P 上での探索問題については主に DHT(Distributed Hash Table) による完全一致検索に関する研究 [3][5] が多く行われているが、近年、多様な探索問題の P2P 上での解決に対する要求が増大している。[2] 特に BSP 探索問題は計算幾何学分野で広く研究されている応用分野の広い問題である。[4] 本研究で提案する手法は、DHT 探索のために広く利用されている Kademia フレームワーク [5] に類似の構造を BSP 探索に利用する。Kademlia は 1 次元空間の再帰的二分割による索引構造を完全一致検索に利用するが、本研究では同様の索引構造を多次元空間に適用することで、分散環境下における BSP 探索を支援する手法を実現する。

## 2 関連研究

P2P 環境下における、高速な検索手法として、Chord[3]、Kademlia[5] などの DHT を用いた完全一致検索手法が提案されている。DHT はキーと値のペアを管理する分散索引機構であり、キーをクエリとし、対応する値を返す完全一致検索を高速に解決する。完全一致検索以外のクエリを処理できる P2P 環境下での探索手法として、BATON[1] や VBI-Tree[2] などの手法が提案されている。BATON は、P2P ネットワークの構造に平衡二分木を利用することにより、範囲探索に対応する手法である。VBI-Tree は、P2P ネットワーク上で R 木などの階層ツリーを利用するためのフレームワークであり、多次元の完全一致探索や範囲探索を可能としている。

## 3 分散 BSP データ構造

本研究では、P2P 環境下における BSP 探索を実現するため、Kademlia による DHT 探索と類似の構造を用いる。Kademlia における検索プロセスは、再帰的に二分された 1 次元 ID 空間を二分探索していく手法ととらえることができる。一方、BSP 探索は、再帰的に二分された (多次元) 空間を探索する操作であり、Kademlia の探索手法を多次元に拡張することが本研究の基本的な考え方となる。

本研究では、各ノードに対し、超平面の列  $H$  と、方向列  $S$  を考える。方向列は要素  $l, r$  からなり、 $H$  と  $S$  により、各ノードが管理すべき空間を決定する。 $S$  は Kademlia における ID に対応する意味をもつ。

$S$  の  $i$  番目の要素  $s_i$  は、該当するノードが  $H$  の  $i$  番目の超平面  $h_i$  で区切られた二つの開半平面  $h_i^l, h_i^r$  のどちらに属する

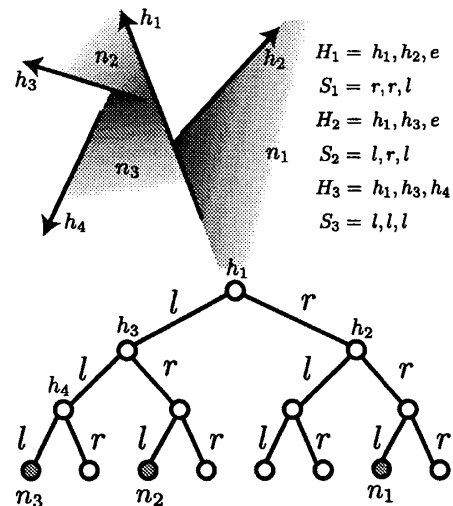


図 1: 提案手法における二分空間と探索例

かを表している。各ノードが存在する領域  $R$  は、

$$R = \bigcap_{i=0}^m h_i^{s_i}$$

で決定される。図 1 に、超平面を有向線分として表し、二次元空間を分割した例を示す。ここで、ノード  $n_i$  における超平面の列と方向列は、それぞれ  $H_i, S_i$  である。図 1 に示すような仮想的な二分木を考えると、根からノードに至るまでのパスに存在する方向と有向線分が、各ノードが所有すべき超平面列と方向列となる。

また、 $H$  は、空の要素  $e$  を許す。方向列の長さは  $m$  で固定し、実際には超平面列の要素  $h_i$  に対応する方向列の要素  $s_i$  が空要素  $e$  であったとしても、いずれかの方向を割り当てるものとする。

Kademlia では、ハッシュ空間上での「距離」を排他的論理和 (XOR) により定義している [5] が、提案手法においてもこれと対応させ、方向列  $X, Y$  の距離  $distance(X, Y)$  を次のように定義する。

$$distance(X, Y) = \max \{ i | x_i \neq y_i \}$$

図 1 において、例えば、 $distance(n_1, n_2) = 3$  である。

## 4 提案手法における検索の手続き

提案手法における検索の手続きは、各ノードが保持する経路表を利用して行われる。経路表は長さ  $j$  のリスト  $m$  個からな

り,  $i$  番め ( $1 \leq i \leq m$ ) のリストには, 距離が  $i$  となるノードに関する  $j$  個のノード情報 (IP アドレス, ポート番号, 方向列) が格納される。

本手法において探索とは, 空間の 1 点を指定し, その領域を管理すべき  $j$  個<sup>1</sup> のノードに関するノード情報を取得することを目標としている。この探索の操作を, node lookup と呼ぶ。

node lookup では, 他ノードに対する単純な「問い合わせ」を繰り返すことで行われる。問い合わせは, 図 2 に示すような, 仮想的な二分木を利用する。問い合わせを受け取ったノードは, 目標点とそのノードが含まれる最小の部分木を考え, その反対側の部分木に存在する  $j$  個のノード情報を送信する。これは, 経路表から  $k$  に最も近い  $j$  個のノード情報を, 問い合わせ送信元ノードに返信することと等しい。

node lookup を開始するノードは, 最大の要素数が  $j$  個に制限された候補リストを用い, これを更新していくことで探索を行う。候補リストの更新は, 問い合わせ点を管理すべきノードにより近いノードが得られたら, それを追加することで行われる。

図 2 に, 提案手法における探索例を示す。今,  $a$  が点  $p$  を管理するノードを検索する。 $a$  は,  $a$  と  $p$  が含まれる,  $h_1$  を根とする部分木の, 左部分木に位置する  $b$  へ問い合わせを送信する。 $b$  は, 同様に  $h_3$  を根とする右部分木に位置する  $c$  の情報を  $a$  に送信する。 $a$  は,  $c$  を候補リストに追加し,  $c$  に対して問い合わせの送信を行う。 $c$  は,  $h_4$  を根とする左部分木に位置する  $d$  の情報を  $a$  に送信し,  $a$  はそれを候補リストに追加する。よって,  $a$  は,  $p$  が存在する領域に割り当てられた  $d$  と,  $d$  に近い複数のノードを取得できる。

ノード  $n$  が, 点  $p$  を管理するノードを検索する場合の, 具体的な探索手順を次に示す。

1. ノード  $n$  が所持する  $H, S$  に対し,

$$\max \left\{ i \mid \left( p \in h_i^l \wedge s_i \neq l \right) \vee \left( p \in h_i^r \wedge s_i \neq r \right) \right\}$$

となる  $i$  を求め, 経路表の  $i$  番目のリストより,  $\alpha$  個のノード情報を取り出し, 問い合わせを行う。

2. 問い合わせを受け取ったノードは, 同様の手順で,  $j$  個のノード情報を取り出し, 問い合わせ送信元ノード  $n$  に返信する。
3. 問い合わせの返信を受け取った  $n$  は,  $j$  個のノード情報を候補リストに追加, 更新する。
4. 候補リストに問い合わせを送信していないノードが存在すれば, 問い合わせを行う。候補リスト中のすべてのノードが問い合わせ済みになったとき, 終了する。

ここで,  $\alpha$  は問い合わせの並行度であり, 同時に  $\alpha$  個のノードへ問い合わせを行うことを意味する。

## 5 実装

前節で述べた探索問題を解決できるよう, 実装を行った。対象とする空間は二次元平面とし, 各ノードに割り当てる超平面列は, 各ノード間で整合性が保たれるよう割り当てた。また, 各ノードに割り当てる方向列はランダムに決定した。

検証は, 一台の計算機で動作するシミュレータを用いて行い, 探索処理が正常に動作することを確認した。

<sup>1</sup>探索結果が  $j$  個である理由は, キーと値を  $j$  個のノードに重複して保持させることで, 冗長性を確保するためである。[5]

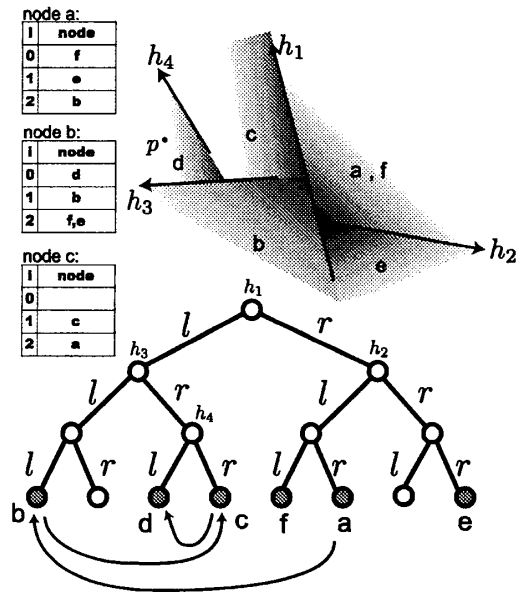


図 2: 提案手法における探索例

## 6 まとめ・今後の課題

本研究では, Kademia の検索手法を拡張することで, BSP 探索問題を解決可能であることを示した。

今後の課題として, ノードやデータの追加・削除時における経路表や保持データの更新方法が挙げられる。本稿では点を指定することによる BSP 領域の特定を探索問題として扱った。今後は, 実際の応用も視野に含め, 他の探索問題や応用も考えていく必要がある。BSP の応用例としては, コンピュータグラフィックスの分野における, 隠面除去 [4] などが挙げられる。

また, 解析的, 実験的な性能評価もあわせて行う必要がある。

## 参考文献

- [1] H.V.Jagadish, B.C.Ooi, and Q.H.Vu. Baton: A balanced tree structure for peer-to-peer networks. In VLDB, pages 661-672, 2005.
- [2] H.V.Jagadish, B.C.Ooi, Q.H.Vu, R.Zhang, and A.Zhou. Vbi-tree: A peer-to-peer framework for supporting multi-dimensional indexing schemes. In ICDE, pages 34-43, 2006.
- [3] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In Proceedings of the ACM SIGCOMM '01 Conference, San Diego, California, August 2001.
- [4] M. ドバーグ, M. ファン, クリベルト, M. オーバマーズ, O. シュワルツコップ共著, 浅野哲夫訳, “コンピュータジオメトリ” 近代科学社 (2000)
- [5] P. Maymounkov and D. Mazieres. Kademia: A Peer-to-peer Information System Based on the XOR Metric, In Proc. of IPTPS 2002, pp. 53-65 (Mar. 2002).