

地理位置情報を扱うスキップ構造を用いた P2P ネットワーク

中村 陽一[†] 斉藤 裕樹[‡] 戸辺 義人[‡]

[†] 東京電機大学大学院 工学研究科

[‡] 東京電機大学 未来科学部

近年、センサネットワーク技術の発展、GNSS (Global Navigation Satellite System: 全世界的航法衛星システム) 機能を持つ小型デバイスの普及により位置情報サービス (LBS: Location-based Service) の利用分野が急速に拡大し、注目が集まっている。LBS で用いる情報は実世界の様々な情報であり、常に増加していくものである。そのため分散環境にて情報を管理する必要がある。また実世界の情報は位置依存情報であるため、位置情報を加味した管理方法が問われる。そこで本研究では、地理位置情報を扱うスキップ構造を用いた P2P ネットワーク GeoSkip を提案する。GeoSkip は、各ノード毎にそのノードを中心に一定の角度で空間を分割し、その分割エリアごとにリンクを構築することで、1次元の情報のみ扱う従来の SkipGraph の概念を2次元に拡張する。これにより位置依存データの効率的な分散配置、検索を行う。

Peer-to-Peer Network with Skip Structure for Location-based Information

Yoichi Nakamura[†] Hiroki Saito[‡] Yoshito Tobe[‡]

[†] Graduate School of Engineering, Tokyo Denki University

[‡] School of Science and Technology for Future Life, Tokyo Denki University

GNSS (Global Navigation Satellite System) equipped mobile devices and the improvement of the technology for sensor networks have enabled Location-based Services. The location-based services deal with real world information which is collected from mobile devices and sensors. Due to large amount of collected data, we should manage such data in distributed fashion. This paper proposes a scalable peer to peer network architecture, called GeoSkip. Geoskip extends 1-dimensional SkipGraphs to 2-dimensional content space in order to achieve efficient data processing for location-based contents.

1 はじめに

位置情報サービス (LBS: Location-based Service) は、位置に関する情報を提供するサービスである。例としてはナビゲーション、天気予報、飲食店情報、カーナビがある。日本において LBS は 1990 年代から始まったとされるが、近年、センサネットワーク技術の発展、GNSS (Global Navigation Satellite System: 全世界的航法衛星システム) 機能を持つ小型デバイスの普及により位置情報サービス) の利用分野が急速に拡大し、注目が集まっている。Google による Google ストリートビューのサービスが日本においても開始され、より LBS に対する人々の関心は高まっている。LBS で用いる情報は実世界の事象を扱うため常に情報は増加していき膨大なものになる。

しかし、現状の多くのアプリケーションやサービスのシステムアーキテクチャは集中管理型が多い。集中管理型はデータを一箇所のサーバにデータを一括して管理しているため負荷の集中、障害などが起きた際、サービスの質の低下、停止などの問題が生じやすい。また LBS には災害時での利用も想定されるため、事故によるサービスの完全停止は起きてはならない。そのため近年、Peer-to-Peer ネットワーク (P2P) による分散管理型に注目が集まっている。また近年、エネルギー分野において電力自由化にともなう、電力市場の形成、自然エネルギーや分散電源の活用、Smart Grid[1] といった大型発電所を中心とした集中型から分散型へ移行するという研究が行われつつある。そのためエネルギーの分散管理という

観点からも P2P の分散管理技術が応用できるため注目され始めている。

P2P を構築する方式には大きく、Unstructured P2P と Structured P2P の二つに分類される。Unstructured P2P は、ノード間のトポロジを規定していない P2P である。目的とする資源を検索する際には、全ての隣接ノードに対してメッセージの転送を行う。そのため、遠くにある資源を見つけることが困難となるため、ノード数が増えるに従い規模拡張性が問題となる。Unstructured P2P を用いた代表的なアプリケーションには、Gnutella や Winny などがある。一方、Structured P2P は、ノード間のトポロジに、一定の構造を構成することで資源の検索や経路制御の効率を高めたオーバレイネットワークである。目的とする資源を検索する際はトポロジにしたがって、隣接ノードにメッセージの転送をすればよい。そのためメッセージの転送が効率化され、規模拡張性が高い。このような方式には主に Chord[2]、CAN[3]、Kademlia[4] などの分散ハッシュテーブル (DHT) が提案されている。DHT はハッシュテーブルの技術を分散環境で実現している手法である。データをハッシュ関数を用いて一意なハッシュ値に投影して、ハッシュテーブルにて管理し、高速なデータ検索と効率的な保管を可能としている。一方、SkipGraphs[5] は SkipList 構造を分散環境にて実現している手法である。SkipList は確率的アルゴリズムを用いたデータ構造である。これら Structured P2P は「高いスケラビリティ」、「高速な検索、ルーティング可能」、「頻繁なノード参加・離脱に対応」、「すべてのノードと通信可能」という利点がある。そのため、従来はサーバで行っていたサービスを実現させるためには、Structured P2P が有用である。しかし、DHT はその特性上、データの持つ順序性を壊してしまうため位置情報を扱うのには向かない。SkipGraphs はハッシュ関数を使わないため、データの順序性を失わず扱えるが、1 次元情報しか扱うことができない。そのため 2 次元情報である位置情報は扱うことができない。

そこで我々は、2 次元情報をそのまま扱うことができ規模拡張性を満たすスキップ構造を持つ P2P ネットワーク GeoSkip を提案する。

本論文では、以下、第 2 章で GeoSkip の構造を説明し、第 3 章にて位置情報にも続く検索アルゴリズムを提案する。第 4 章で GeoSkip の構築方法を述べる。第 5 章ではシミュレーションにより評価を行う。第 6 章で関連研究を紹介し、第 7 章で本稿のまとめと今後の課題を述べる。

2 GeoSkip の構造

GeoSkip では、位置依存情報を扱うため、2 次元座標とその座標に存在する情報を 2 次元の ID 平面上に分散配置されたノードで管理を行う。また、従来の DHT とは異なり、ハッシュ変換を行わずに SkipGraphs の経路制御を 2 次元上に拡張することで、情報の連続性を保ったネットワーク構造を保持する。以下に、GeoSkip のネットワーク構造と経路制御について述べる。具体的には、任意の座標が与えられたとき、その座標の情報を管理するノードへ到達するための経路制御手法を説明する。その後、ノードの参加離脱アルゴリズムを述べる。

2.1 ノードの定義

各ノードは、ノード ID と $Key(x, y)$ の 2 つの値を持つ。ノード ID は GeoSkip ネットワーク全体で一意であり、偏り無く分散されていることが望ましい。これは SkipGraphs の VectorID に相当するものである。ノード ID の桁数の定義は特には無いが、DHT などオーバレイネットワーク構造に用いられる桁数と同程度が望ましい。次に Key はそのノードの持つ位置情報 (2 次元座標) である。また位置情報は全ノードが同じ座標軸を用いていなければならない。位置情報の座標軸としては GPS で利用している世界測地系や日本国内限定での日本測地系などがある。このノード ID と Key の 2 つの値を用いて P2P の構造を構築する。Key は座標値をそのまま用いるため元の 2 次元情報の持つ順序性を崩すことなくノードの持つ位置情報を扱うことが可能となる。

2.2 経路表、ノード間リンクの構成

次にノードの持つ経路表について述べる。各ノードは、自ノードを基準とした周囲のノードの方角と距離を保持する。方角は、ユークリッド平面上で自ノードを中心に一定の角度 θ ごとに平面を分割したとき、周囲のノードがどの分割平面上にあるかを示す。ここで、 θ は式 (1) (2) の条件を満たすように設定する。

$$0 < \theta \leq \frac{\pi}{4} \quad (1)$$

$$\theta \mid \frac{\pi}{2} \quad (2)$$

これにより分割した各面を反時計回りに順に S_i ($i = 0, 1, 2, \dots, \frac{2\pi}{\theta} - 1$) とする。まずノードは各分割平面 S_i 内で Key の値が最も近い、すなわち 2 点間で

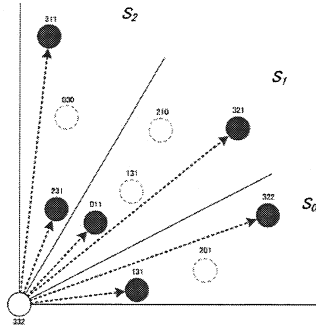


図 1: 平面分割角 θ を $\pi/6$ としたときの経路表の例

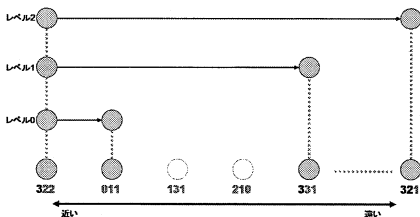


図 2: レベル別の経路

の距離が最も近いノードへの到達方法（物理アドレス）と位置情報、距離を経路表に保持する。例として図 1 は、 $ID = 332$ （本論では説明のため ID は 4 進数表記とする）のノードの第一象限の経路表の例を表しており、各分割平面 (S_0, S_1, S_2) において最も近いノード ID を持つノードはそれぞれ $ID = 131, ID = 011, ID = 231$ である。この経路をレベル 0 とする。次に各分割平面上において、4 進数表記で接頭 1 桁が等しいノード ID を持つノード（図 1 では $ID = 322, ID = 321, ID = 311$ ）へ到達方法と位置情報、距離を保持し、その経路をレベル 1 とする。次に接頭 2 桁まで等しい（例： $ID = 331, ID = 333$ など）ノードへの経路はレベル 2 となり、以下同様にして最大でノード ID の桁数分のレベルまで経路を保持する。ノード ID が偏り無く分散されている場合、図 2 に示すように経路のレベルが高いほど、ノード間の距離が長くなる。図 2 は図 1 の S_1 の分割平面における経路を距離に応じて 1 次元に表したものである。図 3 はあるノードの経路表の視覚的に表したものである。

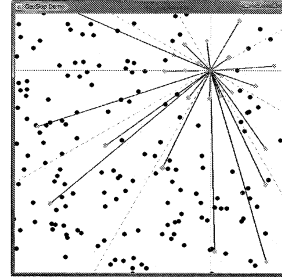


図 3: 経路表の全体像

2.3 ノードの新規参加, 離脱

新規ノードが GeoSIP のネットワークに参加する際、次の手順に従う。

1. 入り口となる既にネットワークに参加しているノードに対して、検索要求を送る。
2. 検索要求を受け取ったノードは、新規ノードの情報（位置情報）を用いて検索をし、新規ノードに最も近いノードを見つける。そしてその結果を新規ノードに伝える。
3. 新規ノードはこのノードに対して再度参加要求 (Join) を送り、このノードの持つ経路表の情報を受信し、経路表からレベル 0 経路を参照し、登録されているノード群に対して Join を送る。
4. Join を送った各ノードから経路表を受信し、分割領域判定と距離判定を行い参加ノードのレベル 0 経路の経路表を構築する。
5. その経路表からレベル 1 の経路となるノードを探す。そしてレベル 1 の経路となるノードに以下同様に参加要求を送る。

これらの動作をにレベル 2, レベル 3... レベル N (最大でノード ID の桁数 - 1) と続けていく。ノードがネットワークから離脱する際は、離脱ノードの経路表に登録されている全てのノードに対して、自分の経路表を送る。離脱ノードから経路表を受け取ったノード達は、その経路表を参考に、自らの経路表を更新する。

3 位置情報に基づく検索アルゴリズム

検索目的の座標が与えられたとき、その座標の情報を管理するノードへ到達するための検索手法について

Algorithm :

```
1 sd_angle = Math.toDegrees(  
2   atan2(dst.y-src.y,dst.x-src.x));  
3 if(sd_angle < 0 )  
4   sd_angle = sd_angle + 360;  
5 int i = sd_angle / theta  
6 return i;
```

図 4: 分割平面判定アルゴリズム

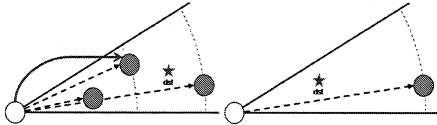


図 5: (i) より近いノードが存在した場合 (ii) 経路上にはより近いノードが存在しない場合

て説明する. GeoSkip では2つの手順 (skip,narrow) を用いる.

3.1 近接ノードへの到達 (skip)

まず, 検索目的となる座標 (dst) が, 検索クエリ発行ノード (src) の経路表のどの分割平面に入るかを判定する. dst と src との位置から, src を中心としての dst との2点による角度 $\angle sd$ を出す. $\angle sd$ を θ で除算した商 (整数値) を i とし, S_i が dst の存在する分割平面となる. 分割平面判定アルゴリズムを図 4 に示す. 検索要求の転送手順は以下のとおりである.

1. S_i の一番高いレベルの経路から順に dst の座標を越えずかつ最も近いノードを検索し, このノードに検索要求を転送する. (図 5 (i))
2. 検索要求を受信したノードも同様の経路探索を行いさらに近いノードに転送を行う.
3. 自ノードよりも近いノードがないと判定された場合 (図 5 (ii)), このノードが近傍ノードとなる.

このノードが近傍ノードとなる. このようにスキップ構造を利用した効率の良い転送手段をとることで検索目的の周辺までのホップ数を削減することが期待できる.

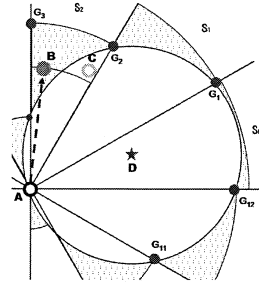


図 6: 絞り込みの対象となる領域とより dst に近いノードが存在する場合

3.2 目的座標への絞り込み (narrow)

3.1 節のステップにより近接ノード (ここではノード A とする) となったノード A の経路表に次のような条件を満たすノード (ここではノード B_i とする) が存在する場合, ノード A の経路表に存在しない, より検索目的に近いノードが存在する可能性がある. 以下図を用いて説明する. 図 6 で, 星印 D を dst とした場合, S_2 においてノード C のほうがノード A より D に近いが, ノード B のほうがノード A に近いので経路表にはノード B が登録されている. このようなことが起きる条件とは, ノード A の分割平面 S_i において, ノード A からみて目的の座標が存在する領域を S_{ad} , ノード A の S_i の経路表に入っているノードを B_i , 目的地を D, D を中心とした $d(A, D)$ (A と D との距離) を半径とした円 (絞込円) と A を通る傾き $\tan(i \times \theta)$ の直線との交点を G_i としたとき, 以下3つのうちいずれかが成り立つノード B_i が存在する場合 (図 6 の斜線の範囲にノードが存在する) である.

- $ad + 1 \leq i \leq ad + \frac{\pi}{2\theta}$ のとき $d(A, B_i) < d(A, G_i)$ が成り立つ
- $ad - \frac{\pi}{2\theta} \leq i \leq ad - 1$ のとき $d(A, B_i) < d(A, G_{i+1})$ が成り立つ
- $i = ad$ のとき $d(A, B_i) < 2d(A, D)$ が成り立つ

そこで近接するノードが存在するか否かを判定するために絞込みを行う. 上記の条件を満たしたノード B に対して反復的に検索要求を順次転送する. 検索要求を受け取ったノード B の動作アルゴリズムを図 7 に示す. ノード B が絞込み検索は全ての分割平面ではなく, ノード B にとって D が存在する分割平面を S_{bd} と, そこから時計回りか反時計回りのどちらかの $\frac{\pi}{2\theta}$ の分の分割平面だけを対象とする. ($\theta = \frac{\pi}{6}$ とした場合, $S_{11+1} = S_0, S_0-1 = S_{11}$ となる). そして時計回りか反時計回りかの判定は, ノード B が

Algorithm :

```
1 Node A //絞込みクエリを送信したノード
2 Location D //目的の位置
3 Node tmp := null
4 Sbd := B.AreaJudgment(dest)
5 for i:= 0 to  $\pi/(2*\theta)$ 
6   Area Stmp
7   if isClockwise(A.location,D) do
8     Stmp := Sbd+i
9   else do
10    Stmp := Sbd-i
11  end if
12  if A.AreaJudgment(
13    B.getRTable(Stmp,0)) = Sab
14    tmp := B.getRTable(Stmp,0)
15    break
16  end if
17  if tmp != null do
18    send tmp to A
19  else do
20    send narrowend to A
21  end if
```

図 7: 絞込みアルゴリズム

ノード A と目的地 D との位置関係を自分を中心として比較し、D がノード A に対して時計回りの方向か反時計回りの方向かにより行う。これは図 7 のアルゴリズムの 8 行目に対応する。次に S_{bd} に近い分割平面から順にレベル 0 の経路表を参照し、ノードがノード A の経路表のノード B が存在する分割平面 S_{ab} に入るかを判定する。し入るのであればノード B はそのノードの情報をノード A からの問い合わせに対する返答として送信し、動作を終了する。これは図 7 のアルゴリズムの 12~15 行目にあたる。もし全ての分割平面に条件を満たすノードが存在しない場合、ノード B はノード A に絞込み終了を伝えるメッセージを送信する。図 8 のように、ノード B 上では S_1 に条件を満たすノード C が存在するので、その情報を A に送信する。ただしノード C がノード A より D に近い場合、絞込み要求でなく skip メッセージを送信する。ノード C が上述した条件を満たさなかった場合、絞込み要求を送信する。そして絞り込み要求を受け取ったノード C は図 7 のアルゴリズムに従い、経路表を参照するが、条件にあるノードが存在しないため (図 8 の斜線)、ここで絞込み

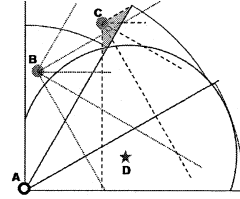


図 8: 絞込みの対象領域

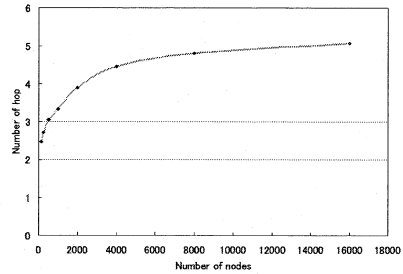


図 9: ノード数に対するホップ数の平均

が終わることになる。そして A は次の分割平面において上述した絞込み条件に適応したノードに対して絞込み要求を送る。この動作を繰り返して、全ての分割平面にて絞込み終了となった場合、A が D に最も近いノードであると確定し、検索は終了する。

4 評価

本研究では、提案するアルゴリズムをシミュレーションによって評価を行った。シミュレーションの条件は次の通りである。ノード数は 125, 250, 500, 1000, 2000, 4000, 8000, 16000 の 9 通りを領域サイズ 300×300 に一様分布させた。平面分割角度 θ は $\frac{\pi}{6}$ である。図 9 はノード数ごとのランダムな 2 地点間での検索にかかるホップ数の平均を示す。その結果、8000 ノードで約 4.8 ホップに対して、倍の 16000 ノードであってもホップ数は約 5 とノード数の増加に対してホップ数は増加は抑えられていることがわかる。よってより多くノードが存在するネットワークであるほどノード数増加に対するホップ数の増加が抑えられる。したがって本手法の規模拡張性が確認できる。

5 関連研究

Globase.KOM[6] は LBS のための P2P 技術を提案している。Globase.KOM は super peer 方式を用いた tree 構造の P2P を用いて分散して構築している。super peer 方式を用いた tree 構造を用いた場合、tree のルートや特別な処理を行う super peer の負荷が多い等の問題が生じる。ドロネー P2P[7, 8] は領域分割を効率的に行うため離散幾何学のドロネー図を P2P を用いて分散で実現している。

- [8] 大西真晶, 源元佑太, 江口隆之, 加藤宏章, 西出亮, 上島紳一: ノード位置を用いた P2P モデルのためのドロネー図の自律分散生成アルゴリズム, 情報処理学会論文誌: データベース, Vol. 47 No. SIG4(TOD29), pp.51-64 (2006).

6 おわりに

我々は位置依存情報に適した高いスケーラビリティを実現可能とするために、一定角度による空間分割を用いてスキップ構造を持つ経路作成を行う P2P ネットワークを提案した。シミュレーションでは一様分布配置によるノード数の増加とによる検索にかかるホップ数の変化を調べ、規模拡張性の有無を確認した。今後の課題として、ノード ID による検索手法、範囲検索の実現方法や接検索といった技術応用、悪意のあるノード対策などセキュリティ等実用面での問題などを検討していく予定である。

参考文献

- [1] SmartGrids: <http://www.smartgrids.eu/>
- [2] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan: Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications. ACM SIGCOMM 2001.
- [3] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp: A Scalable Content Addressable Network. ACM SIGCOMM 2001.
- [4] Petar Maymounkov, David Mazieres: Kademlia: A Peer-to-peer Information System Based on the XOR Metric. The International workshop on Peer-To-Peer Systems 2002 (IPTPS2002).
- [5] James Aspnes, Gauri Shah: Skip Graphs. ACM SIAM Symposium on Discrete Algorithms 2003.
- [6] Aleksandra Kovacevic, Nicolas Liebau, and Ralf Steinmetz: Globase.KOM - A P2P Overlay for Fully Retrievable Location-based Search. 7th IEEE International Conference on Peer-to-Peer Computing (IEEE P2P2007).
- [7] Filipe Araujo, Luis Rodrigues: GeoPeer: A Location-Aware Peer-to-Peer System. 3rd IEEE International Symposium on Network Computing and Applications (IEEE NCA04).