情報処理学会論文誌　Vol. 51　No. 9　1905–1915 (Sep. 2010)

推薦論文

P2P ネットワークにおける Bloom Filter を用いた移動履歴に基づくユーザ探索手法の提案

白木　徹[†1]　寺西裕一[†1,†2]　竹内　亨[†2]
春本　要[†3]　西尾章治郎[†1]

本稿では，ユーザの移動履歴が P2P ネットワークにおいて分散管理されている状況のもと，指定した移動履歴を持つユーザを効率的に探索するための手法を提案する．提案手法では，移動履歴を，訪問場所を示す spot ID と訪問時刻を示す time の組を 1 つの要素とした系列とし，各要素に Bloom Filter を適用したビット論理和をユーザ（ピア）ごとに算出する．この算出された Bloom Filter を Bloom Filter の P2P ネットワークでの探索手法である Bloom Finger Table（BFT）に適用し，移動履歴の AND/OR 検索を実現する．提案手法により，時間指定を含め指定した経路の一部またはすべてをたどったユーザの探索が効率的に実現できる．また，ユーザ探索におけるメッセージ数を削減し，検索適合率を上昇させるため，行動範囲が類似したユーザのピア ID が近傍となるピア ID の割当て法（地理的ピア ID）を提案する．提案手法の有効性をシミュレーションにより評価し，従来手法である Multi-key Skip Graph（MKSG）との比較により 1,000 ピアがそれぞれ 100 の移動履歴を持つ状況での経路検索で AND 検索において，メッセージ数は 80%，OR 検索においては，88% 削減できることを確認した．

Proposal of a User Search Method Based on Movement Records Utilizing Bloom Filter for P2P Network

Toru Shiraki,[†1] Yuuichi Teranishi,[†1,†2]
Susumu Takeuchi,[†2] Kaname Harumoto[†3]
and Shojiro Nishio[†1]

In this paper, we propose a P2P user search method based on movement records. In our proposal, a Bloom Filter is applied to each spotID and time to combine all movement records for one user (peer) as a fixed length bit array. To search a user who followed specified course, we propose a AND/OR search method based on Bloom Finger Table (BFT), which extends finger table, a routing table of Chord DHT to achieve multiple attribute retrieval. By using proposed method, we can efficiently retrieve a user who followed a part of or all of the specified course including time. Additionally, in order to reduce the number of messages and to raise precision of search for a user search, we propose a peer-ID assignment method based on user's geographical foothold. We evaluated our proposal by simulations comparing with Multi-key Skip Graph that is an existing search method on P2P networks. In AND search, we confirmed the number of messages can be reduced by 80% and in OR search, 88% when there are 1,000 peers and each peer has 100 movement records.

1. はじめに

近年，GPS などの位置情報を検出できる機器の普及やユビキタスネットワーク環境の実現にともない，ユーザの現在位置を利用し，その場所に応じた情報やサービスを提供する位置情報サービスの研究開発がさかんに行われている．また，ユーザの現在位置だけでなく，ユーザの過去の移動履歴を応用したサービスに関する研究開発も行われつつある．過去の移動履歴を応用したサービスとしては，たとえば，共通の訪問地を持つユーザ間で相手が未訪問の訪問地を相互に推薦するサービス[8]，特定の会議などのイベントに参加した人に対して通知を行うメッセージ送信サービス，インフルエンザなどの感染者が訪れた場所と同じ場所を訪れた人に注意喚起をするパンデミック対策などが考えられる．こうしたサービスの実現には，複数の地点を含む移動履歴を条件として指定したユーザの探索を行える必要がある．一方，ユーザの移動履歴は，ユーザ自身が管理するコンピュータや信頼できるプロバイダ上のコンピュータなど，ユーザごとに異なる主体が管理するコンピュータ上に保存されることになると考えられる．この場合，移動履歴の共有の形態はユーザの履歴を保持

†1 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University
†2 独立行政法人情報通信研究機構
National Institute of Information and Communications Technology
†3 大阪大学大学院工学研究科
Graduate School of Engineering, Osaka University
本稿の内容は 2009 年 7 月のマルチメディア，分散，協調とモバイル（DICOMO2009）シンポジウムにて報告され，同プログラム委員長により情報処理学会論文誌ジャーナルへの掲載が推薦された論文である．

© 2010 Information Processing Society of Japan

するコンピュータが分散する P2P 型となる．本研究では，コンピュータ（ピア）間が P2P 型で接続される P2P ネットワークにおいて移動履歴が共有される状況のもと，ユーザ探索を効率的に行える機構の実現を目的とする．

移動履歴は，一般に座標と時刻の組の系列となるが，ここではユーザが訪れた場所が地名などにより識別できる想定とし，訪問場所と時刻のペアを要素とする複数の要素を，移動履歴として各ピアが保持する状況を想定する．複数の要素を各ピアが保持する状況下での探索が実行可能な P2P ネットワーク技術として，Multi-key Skip Graph（MKSG）[3] などがすでに提案されている．しかし，MKSG を含め従来の P2P ネットワーク技術では，移動履歴のように検索に指定するキーが複数ある場合，キーごとに独立して検索を実行しなければならないなど，メッセージ数が増大してしまうという問題が生じる．一方，複数の要素および検索キーをそれぞれ固定長のビット列に集約できる Bloom Filter [2] を用い，指定した複数のキーを持つピアを 1 回の検索で実行可能とする P2P ネットワーク技術として Bloom Finger Table（BFT）[7] がある．しかし，BFT はキーワードなどの単純な検索キーに基づく検索方法であり，そのままでは経路などの指定が必要な移動履歴の検索に適用できない．

そこで本研究では，BFT を拡張し，移動履歴を要素として扱ったうえで，複数の訪問場所，複数の訪問時刻を指定した経路検索を行えるようにする方法を提案する．また，検索に該当するピアにクエリを送信する際にメッセージが分散し，メッセージ数が増大するという問題を回避するため，ユーザの活動拠点をもとにしたピア ID の設定方法（地理的ピア ID）もあわせて提案する．

2. 構造型 P2P ネットワークによる移動履歴の管理

2.1 想 定 環 境

本研究では，ユーザが訪れた場所が，既知の地名などにより識別できる想定とし，識別子（spot ID と呼ぶ）が割り振られていることを前提とする．ユーザがある場所を訪れた際にその場所の spot ID と時刻 time のペアを要素とし，これをユーザごとに蓄積したものを移動履歴とする．ここでは，ある 1 人のユーザのすべての移動履歴は 1 つのピアに保存されるものとする．また，1 つのピアには 1 人のユーザの移動履歴のみが保持されるものとする．すなわち，1 人のユーザが 1 ピアに対応する P2P ネットワークを構築するという想定である．

従来研究で扱われてきた P2P ネットワークは非構造型と構造型に大別できる．非構造型は，フラッディングと呼ばれる無作為検索が行われるため，ネットワークにかかる負荷や検索にかかる遅延が大きく，本研究で想定される過去の移動履歴を利用したサービスには適していないと考えられる．よって，本研究では指定した移動履歴を保持するピアの探索を行える構造化 P2P ネットワークを構築する方法を提案する．

移動履歴の検索は，複数の spot ID と時刻のペアの系列からなる経路を指定した検索となる．本研究では特に，経路による検索として,「AND 検索」と「OR 検索」を扱うこととする．AND 検索は，指定した複数の spot ID と時刻のペアの系列すべてを移動履歴として保持するピアの探索である．OR 検索は，指定した複数の spot ID と時刻のペアの系列のうちいずれかを移動履歴として保持するピアの探索である．すなわち，AND 検索は，ある経路をたどった人を探し出す検索であり，OR 検索は，ある経路を一部でもたどった人を探し出す検索である．パンデミック対策などでは，ある時刻に指定した経路を一部でもたどった人を検索する必要があるが，たとえば，訪問経験に応じて提供する内容を変えるサービスなどでは，時刻は重要ではなくなる．よって，時刻を指定せず，spot ID を複数指定した経路のみによる検索も扱える必要がある．

2.2 構造型 P2P ネットワークによる経路検索

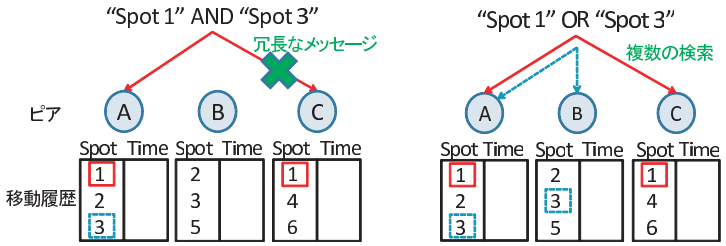
各ピアが複数の要素を保持する状況において，指定した要素を含むピアを探索できる構造型 P2P ネットワークとして，DHT と Multi-Key Skip Graph があげられる．

DHT には Chord [9]，Pastory [6]，Tapestry [10]，CAN [5]，Kademlia [4] などの実装がある．方式は異なるが，いずれもピア間で何らかのショートカットリンクを構成することにより，$O(\log n)$ で指定したキーに対応する値を検索できる（n はピア数）．DHT では登録されたキーに対応する値を各ピアが分担して検索可能とするが，キーの分担は数学的に決定されるため，どのピアがどのキーを分担するかがあらかじめ分からない．各ピアが持つ移動履歴の検索を可能とするには，各ピアが持つ移動履歴の spot ID，時刻をキーとし，当該移動履歴を保持するピアのアドレスリストなどを値として登録しておけば，spot ID，時刻をキーとしたピアを検索できる．

Multi-key Skip Graph（MKSG）は，ピアが持つキーを検索可能とする構造型 P2P ネットワーク技術の 1 つである Skip Graph [1] を拡張した手法であり，ピアが複数のキーを保持可能とするものである．Skip Graph（および MKSG）は ‘membership vector’ と呼ばれる乱数に基づく階層化ショートカットリンクを構成しており，$O(\log n)$ でピアを検索できる（n はピア数）．ユーザに対応する各ピアが，複数の spot ID，時刻をキーとして保持して MKSG を構成すれば，移動履歴に基づく検索を実現できる．

DHT，MKSG いずれも，spot ID のみでも経路の検索を行えるようにするには，spot ID，時刻のペアに加え，spot ID もキーとすればよい．DHT の場合，1 つの箇所を多くの

© 2010 Information Processing Society of Japan



(a) AND 検索による問題点　　(b) OR 検索による問題点

図 1　既存手法における移動履歴検索の問題点
Fig. 1　Problems in the existing search methods to apply movement records.

ユーザが訪問した場合に担当ピアがボトルネックになる可能性があり，どちらかといえばそのようなボトルネックが生じない MKSG のほうが経路検索には適していると考えられる．

2.3　AND/OR 検索における問題点

上記技術は 1 度に 1 つの要素しか検索に指定できないため，まず，AND 検索を実現するには，検索対象となる経路のうち 1 つの要素をキーとして指定した検索を実行し，得られるピアの中で，経路の他の要素をすべて保持するピアを探すことになる．この場合，最初に指定する検索キーを多数のピアが保持していたとき，冗長なメッセージ送信が生じる．たとえば，**図 1** (a) のように spot 1 と spot 3 の AND 検索を実行したい場合，まず spot 1 を検索キーに指定して検索を行うとピア A，C が該当する．これらのうち，spot 3 を保持するピア A のみが検索に合致するため，結果としてピア C には冗長なメッセージが送信されることになる．

一方 OR 検索は，経路の要素それぞれをキーとして検索を行う必要が生じる．したがって経路の要素数に対応する回数だけ検索を行う必要があり，メッセージ数が増大してしまう．たとえば，図 1 (b) のように spot 1 と spot 3 の OR 検索を実行したい場合，spot 1 を保持するピアの検索と spot 3 を保持するピアの検索を独立して行う必要がある．

3.　Bloom Filter を用いた移動履歴に基づくユーザ探索

3.1　Bloom Finger Table

Chord を拡張し，1 回の検索で複数要素を検索可能とする手法に Bloom Finger Table（BFT）がある．BFT は Chord におけるショートカットリンク構造を構成する Finger Table に Bloom Filter を適用したものである．

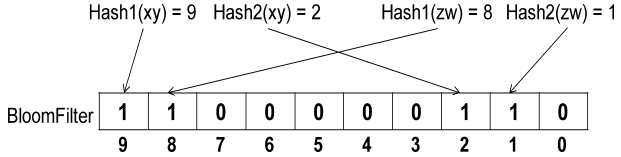


図 2　Bloom Filter
Fig. 2　Bloom Filter.

Bloom Filter とは，ある要素が存在しているかどうかを一定のビット列で表現できるデータ構造である．Bloom Filter を構成するには，m ビットのビット列をすべて '0' に初期化し，0 から $m-1$ の値を返す k 個のハッシュ関数により要素のハッシュ値を求める．求めたハッシュ値をビット列のインデックスとして扱い，該当する箇所のビット値を '1' に変更することで要素が存在することを表す．たとえば，**図 2** では要素 xy にハッシュ関数 1 と 2 を用いてハッシュ値 9 と 2 を求める．この値に該当するビット列の箇所を '0' から '1' に変更する．要素 zw でも同様の操作を行うことで複数の要素を 1 つのビット列で表現することができる．また，ある要素が存在するかどうかを判断するときには，その要素から求めたハッシュ値に該当するビットの箇所が '1' であればその要素が存在すると判断することができる．Bloom Filter は要素が集合のメンバであるかどうかのテストに使うことができるデータ構造であり，偽陽性（False Positive）による誤検出の可能性があるが，偽陰性（False Negative）はない．

Chord の Finger Table は，2 のべき乗距離離れた ID を持つピアへのショートカットリンクをテーブルとして保持している．BFT では，各ピアが保持するすべての要素の Bloom Filter を生成し，論理和をとり集約したものをピアの Bloom Filter としている．さらに Finger Table 上のショートカットでたどりつくピア群の Bloom Filter の論理和をとり集約したものを Finger Table と対応付けて Finger Bloom Filter（FBF）として保持する．検索時は，検索要求の複数要素に対応する Bloom Filter の論理和をとった Bloom Filter を生成する．この検索要求から生成した Bloom Filter を Request Bloom Filter（RBF），1 つのピアが保持するすべての要素から生成した Bloom Filter を Node Bloom Filter（NBF）と呼ぶ．検索要求を受け取ったピアは，検索要求を受け取ったピアの FBF が RBF を包含するピアへ検索要求を転送する．ピアは自身が持つ全要素について検索要求を満たすかどうかを調べ，合致すれば検索要求元ピアへ応答する．たとえば，**図 3** では，ピア 0 の Finger

© 2010 Information Processing Society of Japan

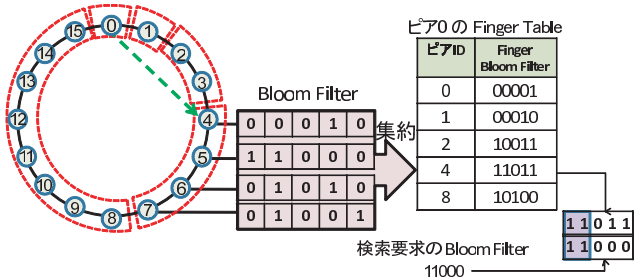


図 3　Bloom Finger Table
Fig. 3　Bloom Finger Table.

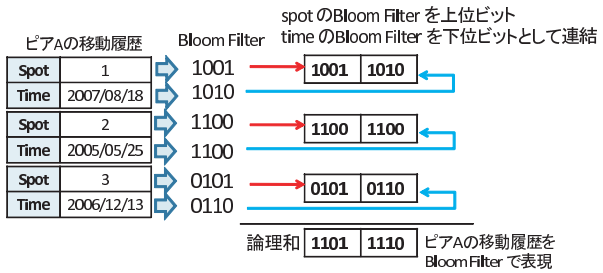


図 4　移動履歴への適用例
Fig. 4　An example of applying movement records.

Table はピア 0 の Bloom Filter とピア 1 の Bloom Filter，ピア 2，ピア 3 の論理和から生成した Bloom Filter，ピア 4 からピア 7 までの論理和から生成した Bloom Filter，ピア 8 からピア 15 までの論理和から生成した Bloom Filter を含んでいる．図 3 ではピア 4 に検索要求との包含関係が成り立つためピア 4 へ検索要求が転送されている．

3.2　提 案 手 法

3.2.1　移動履歴への Bloom Filter の適用

本研究では，BFT を拡張し，ユーザの訪問場所と時刻のペアを要素とする Bloom Filter を扱い，ユーザの移動履歴を検索可能とする方法を提案する．提案手法では，検索キーとなる spot ID，時刻（time）それぞれに Bloom Filter を適用し，spot ID により生成される Bloom Filter を上位ビット列，time により生成される Bloom Filter を下位ビット列として連結する．**図 4** は，移動履歴への適用例である．この例は，spot 1，2，3 の経路をたどって移動したことを Bloom Filter により表現している．まず，spot 1，時刻 2007/08/18 を移動履歴の要素として Bloom Filter を生成し，同様にして spot 2，時刻 2005/05/25 と spot 3，時刻 2006/12/13 それぞれを移動履歴の要素として Bloom Filter を生成する．ピア A のすべての要素から生成した Bloom Filter の論理和をとり集約したものがピア A の移動履歴を表す NBF となる．この Bloom Filter により，場所と時刻の両方を指定した検索，場所のみを指定した検索，時間のみを指定した検索のいずれにも対応できる．

移動履歴に基づく FBF の更新とそれに基づく検索は基本的に BFT と同様である．FBF を生成するため，各ノードは Finger Table 内のノードと通信し，探索経路の範囲内の全ノードの NBF の論理和をとる．FBF の生成には，各ノードが Finger Table 内のノードと通信を行う必要があり，全ノードの NBF を生成するには $O(n \log n)$ の処理コストがかかる．FBF 生成方法の詳細は文献 7) を参照されたい．

また，本手法では検索要求を転送する際は要素の状態で送信し，受け取ったピアが Bloom Filter に変換し検索を行う．そのため，次項以降で説明する AND 検索および OR 検索どちらも RBF のメッセージサイズやメッセージ量は変わらない．本手法は Chord を基盤とした手法でありルーティングはどちらも変わらず，Chord と本手法における検索コストは同程度であるといえる．

3.2.2　AND 検索

要素として移動履歴を用いているものの基本的には AND 検索は BFT による検索と同様である．指定される複数の spot ID と時刻のペアの系列すべてを移動履歴として保持するピアの探索を行うためには，移動履歴から Bloom Filter を生成したのと同様の方法で検索要求の移動履歴から Bloom Filter を生成し，それらの論理和をとって生成した RBF を検索のキーとすればよい．BFT と同様に，検索転送の際，自ピアの NBF との間にビット列の包含関係が成り立ち，かつ要求の移動履歴をすべて保持している場合，検索要求元ピアに検索要求を満たすピアとして，そのピア ID を応答として送信する．

図 5 (a) は，AND 検索の動作を実現する関数 *search_AND* の擬似コードである．検索の始点となるピアの Finger Table それぞれについて順に RBF と包含関係を調べ，包含関係が成り立てば該当するピアに検索要求を転送する．検索要求を受け取ったピアにおいても同様に再帰的に処理を実行する．もし，自ピア以外の FBF がすべてについて包含関係が成り立たなければそこで検索を終了する．RBF が自ピアの NBF（FBF）を含む場合，移動履歴を保持する可能性のあるピアであり，そうでない場合は，当該移動履歴を含むピアは見つからなかったことになる．*notify_forward*，*notify_match*，*notify_not_match* は，それぞれ，

© 2010 Information Processing Society of Japan

```
requester_id：　発行ピアのピア ID
request_list：　検索に指定する spot_id, time の組のリスト
BFT：　(id, fbf) の組のリスト．id はピア ID，fbf は Finger Bloom Filter
RBF_AND：　AND 検索時の検索要求を Bloom Filter に変換したビット列
RBF_OR：　OR 検索時の検索要求を Bloom Filter に変換したビット列
make_bloom_filter(id, id)：　spot_id, time から Bloom Filter を生成する処理
find_node(id)：　id に該当するピアを見つける処理
n.notify_forward(id)：　ピア ID が id であるピアに検索要求が転送されたことをピア n に通知する処理
n.notify_match(id)：　ピア ID が id であるピアが検索要求にマッチしたことをピア n に通知する処理
n.notify_not_match(id)：　ピア ID が id であるピアが検索要求にマッチしなかったことをピア n に通知する処理

n.search_AND(requester_id, request_list) {
  for i ← 0 to request_list.length − 1
    RBF_AND ← RBF_AND ∩ make_bloom_filter(request_list[i].spot_id, request_list[i].time);
  r ← find_node(requester_id);
  for i ← BFT.length − 1 downto 1
    if (BFT[i].bf ∩ RBF_AND = RBF_AND)
      n′ ← find_node(BFT[i].id);
      n′.search_AND(requester_id, request_list);
      r.notify_forward(n′.id);
    if (BFT[0].bf ∩ RBF_AND = RBF_AND)
      r.notify_match(n.id);
    else
      r.notify_not_match(n.id);
}
```

(a) AND 検索を実現する関数 search_AND

```
n.search_OR(requester_id, request_list) {
  for i ← 0 to request_list.length − 1
    RBF_OR[i] ← make_bloom_filter(request_list[i].spot_id, request_list[i].time);
  r ← find_node(requester_id);
  for i ← BFT.length − 1 downto 1
    for j ← 0 to RBF.length − 1
      if (BFT[i].bf ∩ RBF_OR[j] = RBF_OR[j])
        n′ ← find_node(BFT[i].id);
        n′.search_OR(requester_id, request_list);
        r.notify_forward(n′.id);
      break;
  if (BFT[0].bf ∩ RBF_OR = RBF_OR)
    r.notify_match(n.id);
  else
    r.notify_not_match(n.id);
}
```

(b) OR 検索を実現する関数 search_OR

図 5　擬似コード
Fig. 5　Pseudocode.

転送した，検索要求を満たした，検索要求を満たさなかったことを要求元ピアに通知する関数である．要求元ではこれらの通知をもとに，転送先において検索を満たすピアが存在したかどうかを知ることができる．

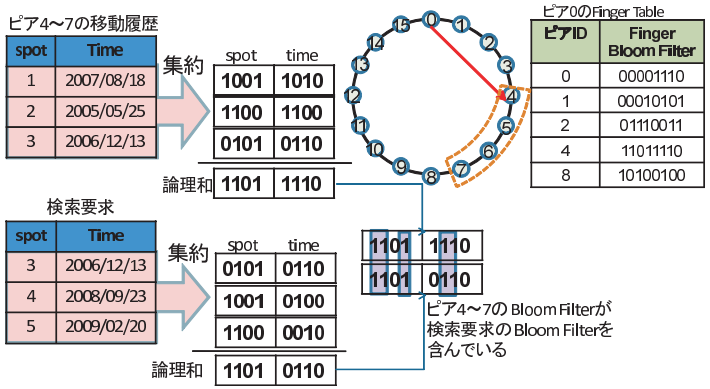


図 6　AND 検索
Fig. 6　AND Search.

たとえば，**図 6** ではピア ID 4〜7 が保持する要素から生成された Bloom Filter を，ピア 0 におけるピア 4 向けの Finger Table に保持している．検索要求は spot 3，4，5 を経由する経路の検索であり，これら 3 つの要素による Bloom Filter からも同様に論理和をとった RBF を生成する．検索時には，ピア 0 の FBF と RBF とでビット列の比較を行う．この検索要求の転送を再帰的に，ビット列の包含関係が成り立たなくなるまで繰り返す．図 6 の例ではピア 4 が担当する範囲であるピア 4〜7 の移動履歴に対応する Bloom Filter が RBF を包含するため，ピア 4 に検索要求が転送される．ピア 4 のみに送信されるためピア 1〜3，7〜15 へ検索要求の経路が存在しないと判断でき，検索要求の転送は行わない．この例では，FBF が RBF を包含しているが，実際には，各ピアが保持する移動履歴は検索要求を満たしておらず，偽陽性による誤検出が発生している．

提案手法は，多数のピアが同じ移動履歴を保持している場合があったとしても，検索要求にあるすべての要素を含むピアへのみ検索要求が転送されるため，メッセージ数を削減することができる．

3.2.3　OR 検索

OR 検索は，指定される複数の spot ID と時刻のペアの系列のうちいずれかを移動履歴として保持するピアの探索を行うものである．

図 5 (b) は，OR 検索の動作を実現する関数 *search_OR* の擬似コードである．OR 検索では検索要求の要素ごとに RBF を作成し，それらを連結したものを検索要求とする．検索

© 2010 Information Processing Society of Japan

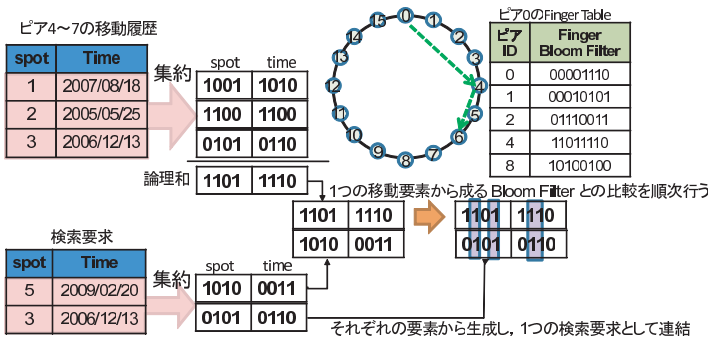


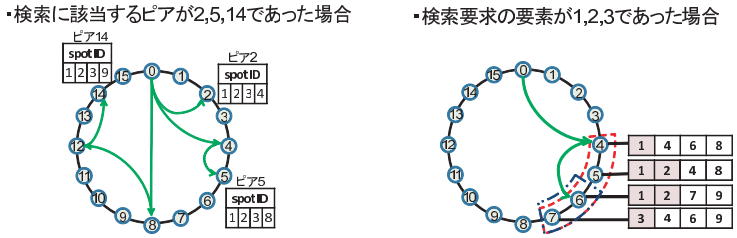
図 7　OR 検索
Fig. 7　OR Search.

時，検索要求の連結された各 RBF のうち，いずれかが FBF に包含される場合に，対応するピアに検索要求を転送する．検索要求の転送は，AND 検索と同様に，再帰的に包含関係が成り立たなくなるまで繰り返される．また，AND 検索と同様，転送の際，NBF との間に包含関係が成り立ち，かつ要求の移動履歴をすべて保持している場合，検索要求元ピアに検索要求を満たすピアとして，そのピア ID を応答として送信する．

たとえば，**図 7** では，ピア 5 の要素の 1 つに spot 2 への移動履歴，ピア 6 の要素の 1 つに spot 3 への移動履歴を含んでいるとする．検索要求では spot 5 と spot 3 への移動履歴を含んでいる．ピア ID 4〜7 が保持する要素から生成された Bloom Filter と 1 つ目の検索要求である spot 5 への移動履歴から生成した RBF との包含関係を調べる．この例では包含が成り立たないため 1 つ目の検索を終了し，2 つ目の検索要求である spot 3 への移動履歴から生成した RBF との包含関係を調べる．2 つ目の検索要求で包含が成り立ったため，まず，検索要求はピア 4 へ転送される．ピア 4 でも同様の包含関係を再帰的に調べ，最終的には spot 3 を保持しているピア 6 へ検索要求を転送する．このとき，検索要求は，すべての検索要素の Bloom Filter の連結を含むため，サイズは大きくなるが，転送される検索要求メッセージとしては 1 つのみで済む．Bloom Filter のデータサイズは，大きくても数キロバイトであり，一般のデータと比較すると小さいため，検索要求のサイズが問題とはならないと考える．

3.3　地理的ピア ID の設定

BFT では，Chord と同様にハッシュ値によってピア ID が割り当てられるため，ピアは



(a) 検索要求の転送が分散する例　　(b) 検索適合率が悪化する例

図 8　BFT のみの使用で問題が生じる例
Fig. 8　An example of problem in using only BFT.

保持する移動履歴にかかわらず ID 空間に一様に割り当てられる．そのため，ある検索要求に合致するピアが複数存在した場合，それらのピアの ID 空間上での配置も一般に一様となる．その結果，検索メッセージの転送も**図 8** (a) のように分散してしまい，検索メッセージが増大する原因となる．

また，FBF を求める際に NBF の論理和をとるため，保持する移動履歴にかかわらずピアが ID 空間上に配置されることにより，実際には検索要求に合致するピアが探索経路上に存在しないピアにも検索メッセージが転送される状況が発生する可能性が高くなる．

たとえば，図 8 (b) に示すように検索要求の要素 1，2，3 であった場合には，それらの要素を保持しているピアは存在しないが，ピア 4〜7 の論理和をとった結果，要素 1，2，3 が存在すると見なされ，ピア 0 → 4 へ検索要求が転送され，さらにピア 6〜7 の論理和からピア 4 → 6 へ検索要求が転送され，無駄なメッセージを送信してしまう．

このような問題を緩和するためには，与えられた任意の RBF に対して，それに適合する FBF の数ができるだけ少なくなるよう，各ピアの NBF に基づいてピア ID を決定できればよい．そのためには，NBF が類似したピアに対して，ID 空間上で近くなるようにピア ID を割り当てることができればよい．移動履歴を扱う場合，移動履歴が類似したユーザに対応するピアを ID 空間上で近くに配置すればよいことになる．これを「地理的ピア ID」と呼ぶ．移動履歴が類似するとは，ユーザ同士の訪れる場所が同じである，つまりユーザが保持する spot ID が同じであることを意味する．

地理的ピア ID の設定方法にはいくつかの方法が考えられるが，ここではその設定方法は限定しない．たとえば活動拠点が近いユーザ同士は移動履歴も類似するという想定のもとで活動拠点に基づいてピア ID を設定する方法や，移動履歴に基づき定期的にピア ID の割当

© 2010 Information Processing Society of Japan

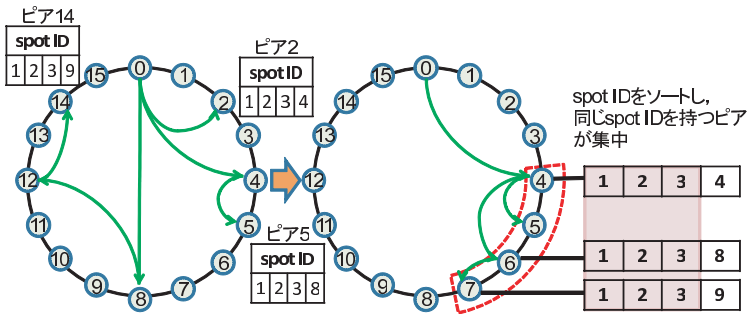


図 9　地理的ピア ID の設定の一例
Fig. 9　An example of setting geographical Peer-ID.

てを変更するような設定方法などが考えられる．前者の実現方法としては，たとえば，発行するピア ID の管理者を設け，各ピアが自身が保持する過去の移動履歴の重心点，履歴が多い地域，自宅の住所などを申請時に送付し，管理者がそれらをもとに ID を決定するといった方法が考えられる．**図 9** は，地理的ピア ID による効果を示している．図 8 (a) では検索要求の転送が分散した結果，検索メッセージ数は増大していたが，地理的ピア ID の設定を行うことで検索メッセージ数の増大を抑制できている．

4. 評　　価

提案手法の効果を確認するため，シミュレーションによる評価を行った．本稿では特に，提案手法によるメッセージ数の削減効果について確認している．

4.1 シミュレーション環境

ユーザの移動は，Zipf の法則[11] に従うものとする．Zipf の法則は経験的に得られた法則であり，様々な現象（Web ページへのアクセス頻度や都市の人口など）において確認されている．ここでは，ユーザによる訪問地が Zipf の法則に従うと仮定する．すなわち，シミュレーション環境では，多くの人が訪問する場所は集中し，少ない人が訪問する場所が多く存在するものとしている．

Zipf の法則では，要素は下記の式から要素のサイズによって順位付けされる．式中の k は順位，N は要素数，s は頻度の偏りのパラメータである．s の値が大きいほど距離が小さい値の頻度が高くなる．シミュレーションではこのパラメータ s を $s = 1$ とした．たとえば，rank 1 の要素のサイズが 30 ならば，rank 2 の要素のサイズは約 15，rank 3 の要素のサイズは約 10 となる．

$$f(k; s, N) = \frac{1/k^s}{\sum_{n=1}^{N} 1/n^s}$$

（N：全要素の数，k：rank）

シミュレーションにおいては，Zipf の法則に沿った乱数が返されるため多少のばらつきはあるが，全要素の数が 100 のときに rank 1 の要素のサイズはおよそ 13 とした．

また，地理的ピア ID による効果を評価するため，地理的ピア ID が設定されている場合は，隣接するピアどうしが 60%が同じ spot ID を持つものとした．ただし，訪問時刻は乱数により決定している．検索要求として，5 カ所の場所を訪問したユーザの検索を行う．検索要求として，5 カ所が訪問者数が多い場所，少ない場所に偏らず，均一に選ばれている場合（偏りなし）と，訪問頻度が上位の場所のみから選ばれている場合（偏りあり）の 2 種類を想定した．

上記検索について，Multi-key Skip Graph（MKSG）との比較を行った．MKSG では各ピアが保持する要素をキーとした．MKSG は範囲検索を行えるが，検索対象が連続ではないため範囲検索は行わない実装としている．

検索に合致するピアの数は，AND 検索では全ピアの 5%，OR 検索では全ピアの 10%となるよう検索条件を設定した．シミュレーション環境のパラメータ設定を**表 1** に示す．

本シミュレーションでは，全ユーザが訪問しうる訪問地（spot）の数を 1,000 とした．これは，たとえば，市町村単位での spot 定義や，1 つの都市内の観光スポットの spot 定義とすれば十分な数であると考える．一方，time は，長期にわたる記録を想定して，約 15,000 とした（時間単位であれば 625 日，日付単位であれば約 40 年）．Bloom Filter ではそのサイズが大きいほど偽陽性が減るが，ネットワークトラヒックの増大を抑えるうえでは 1 つのパケットで送信できる程度の大きさに抑えられることが望ましい．よって，ここでは 1 パケットに収まる大きさとして 4,096 ビット（512 バイト）で Bloom Filter を表現する想定とした．Bloom Filter においては，要素数と Bloom Filter のサイズに応じて，偽陽性の発生確率が小さくなる最適なハッシュ関数の数を求めることができる[2]．ここでは Bloom Filter のサイズと比して要素数が多い想定で偽陽性が発生しやすい環境を扱っており，ハッシュ関数の数は最小値である 1 とした．シミュレーションではピア数を最大 1,000 まで変化

© 2010 Information Processing Society of Japan

表 1　シミュレーション環境
Table 1　The simulation parameters.

ピア ID のサイズ	2^{160}
Bloom Filter のビット長	4096 bit
ハッシュ関数の数	1
1 ユーザが管理する要素数	100
spot の数	1,000
time の数	14,925
ピアの最大数	1,000

させ，それぞれの検索におけるメッセージ数，ホップ数，検索適合率を評価した．

4.2　評 価 結 果

4.2.1　検索要求が訪問者順の偏りがない訪問場所を含む場合

図 10 (a)，(b) は，それぞれ，検索要求が訪問者順の偏りがない訪問場所を含む場合の AND/OR 検索のメッセージ数を示している．いずれの検索においても，提案手法が MKSG よりもメッセージを大幅に削減できていることが分かる．

AND 検索では，提案手法は検索条件のすべての要素を持つピアに絞ったメッセージ転送が行われるのに対し，MKSG では，検索に指定された 1 つの要素で検索を行っているため，AND 条件での条件を絞った検索が行えず，メッセージ数が増えていた．

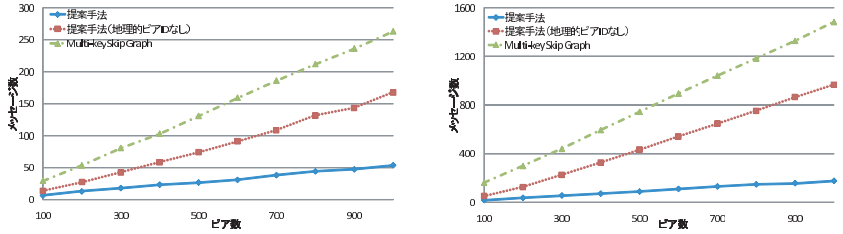
OR 検索では，検索にマッチするピア数が増えるため，メッセージ数も AND よりも増えているが，MKSG では，すべての検索条件の要素それぞれの検索が必要となるため，本シミュレーションでは 5 回検索が実行されるのに対し，提案手法は 1 回で済むため，メッセージ数が少なく抑えられている．

地理的ピア ID が設定されている場合は特にメッセージ数を低減させることができており，この環境では，AND 検索で 80%，OR 検索で 88%削減されている．

4.2.2　検索要求が訪問者数の上位に偏った訪問場所を含む場合

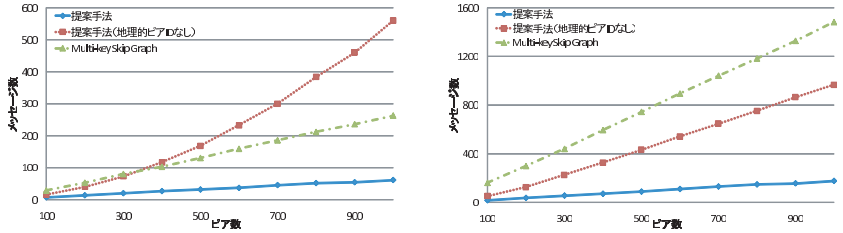
図 11 (a)，(b) は，それぞれ，検索要求が訪問者数の上位に偏った訪問場所を含む場合の AND/OR 検索のメッセージ数を示している．訪問者数が多い箇所のみに対する検索が行われると，その場所に行ったことのあるユーザが多数存在することとなるため，メッセージ数が必然的に増大する．

AND 検索では，地理的ピア ID の設定を行わない場合は，MKSG よりもメッセージ数が多くなってしまっている．これは，訪問者数が上位の箇所に相当する Bloom Filter のビットが，多くのピアで 1 となることにより，False Positive が増加するためである．しかし，



(a) AND 検索のメッセージ数　　(b) OR 検索のメッセージ数

図 10　検索メッセージ数（偏りなし）
Fig. 10　The number of messages (uniform selection).



(a) AND 検索のメッセージ数　　(b) OR 検索のメッセージ数

図 11　検索メッセージ数（偏りあり）
Fig. 11　The number of messages (skewed selection).

地理的ピア ID の設定を行った場合はその影響を受けることなく，メッセージ数を比較手法から 76%程度削減できている．

検索に適合するピアの数が，偏りなしの場合と同等であるため，OR 検索は，図 10 (b) とほぼ同様の傾向となっている．地理的ピア ID の設定を行った場合，MKSG よりもメッセージ数を 88%削減できている．

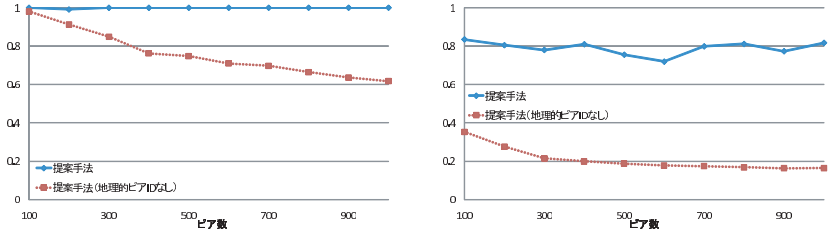
4.2.3　検索適合率

提案手法では，Bloom Filter の論理和がとられるため，ピアが検索に指定した要素を保持していないピアへもメッセージが送信されてしまう場合がある．この冗長ピアの数が Bloom Filter のパラメータ設定に影響すると考えられるため，

検索適合率＝正解を保持するピア数/検索が到達したピア数

をそれぞれの検索について調べた．

© 2010 Information Processing Society of Japan



(a) AND 検索の適合率　　(b) OR 検索の適合率

図 12　検索適合率
Fig. 12　The precision of the search.

AND/OR 検索（偏りなし）における検索適合率を**図 12** (a)，(b) に示す[⋆1]．

AND 検索では，地理的ピア ID の設定を行わないと次第に検索適合率が下がっている．これは，ピア数が増えるに従って集約する範囲内のピア数が増えることで Bloom Filter において 1 となるビットが増えるためである．OR 検索も同様に地理的ピア ID の設定を行わないと次第に検索適合率が下がっている．AND 検索と比べると，検索に該当するピア数が多くなるため，適合率が下がっている．シミュレーション環境のパラメータ設定で，AND 検索では 60%程度，OR 検索では 20%程度の適合率となっており，メッセージ数が増加してしまう要因になっている．

いずれも，地理的ピア ID の設定を行うことで，適合率の低下が抑えられている．これは集約する範囲内のピアが保持する移動履歴が類似するため，それらで論理和をとっても Bloom Filter の 1 になるビットが増えないためである．

5. ま と め

本稿では，Chord をベースとして複数要素検索を可能とする Bloom Finger Table を拡張し，ユーザの移動履歴検索を実現する方法，および，同手法において検索要求の転送を抑えるための地理的ピア ID を提案した．また，提案手法と Multi-Key Skip Graph との比較評価を行い，メッセージ数を削減できる効果を確認した．

評価におけるシミュレーションでは，要素数が多く，Bloom Filter が比較的小さい環境を想定したが，より要素数が少ない，あるいは，Bloom Filter のサイズが大きい環境では，ハッシュ関数の数を増やしたほうが偽陽性が減る可能性がある．

本評価は，Bloom Finger Table があらかじめ構築できている環境を想定していたが，実際に運用する際には Bloom Filter を正しい値に保つための処理が必要である．提案手法では，利用者が新たに移動するたびに Bloom Filter を更新しなければならず，メンテナンスコストが問題となる．ユーザの移動範囲が広範にわたらず，日常繰り返される移動では Bloom Filter および Bloom Finger Table を更新しなくても済む Bloom Filter 構造の検討などが必要であろう．また，本評価における地理的ピア ID は，あらかじめ決定されている想定としていたが，実際のユーザの属性や移動履歴からどのようにピア ID を決めるかについてはさらなる検討が必要である．

今後上記についての検討を進めたうえで，実際の移動履歴を管理可能な実装を進め，P2P 型分散環境におけるユーザ移動履歴の管理基盤の確立を目指したい．

謝辞　本研究の一部は，総務省委託研究「ユビキタス・プラットフォーム技術の研究開発」の一環として実施したものである．ここに記して謝意を示す．

参 考 文 献

1) Aspnes, J. and Shah, G.: Skip Graph, *Association for Computing Machinery Symp. on Discrete Algorithms*, pp.384–393 (Jan. 2003).
2) Bloom, B.: Space/time trade-offs in hash coding with allowable errors, *Communications of the Association for Computing Machinery*, Vol.13, No.7, pp.422–426 (July 1970).
3) 小西佑治，吉田　幹，竹内　亨，寺西裕一，春本　要，下條真司：単一ノードに複数キーを保持可能とする Skip Graph 拡張，情報処理学会論文誌，Vol.49, No.9, pp.3223–3233 (2008).
4) Maymounkov, P. and Mazieres, D.: Kademlia: A peer-to-peer information system based on the XOR metric, *Proc. International Workshop on Peer-to-Peer Systems*, Vol.258, pp.53–65 (Mar. 2002).
5) Ratnasamy, S., Francis, P., Handley, M. and Karp, R.: CAN: A scalable content-addressable network, *Association for Computing Machinery SIGCOMM Conf.* pp.161–172 (Aug. 2001).
6) Rowstron, A. and Druschel, P.: Pastory: Scalable, decentralized object location and routing for large-scale peer-to-peer systems, Lecture Note in Computer Science, Vol.2218, pp.329–350 (Nov. 2001).
7) 佐藤一帆，松本倫子，吉田紀彦：複数キーワード検索に対応した分散ハッシュ型 P2P ネットワーク，第 6 回情報科学技術フォーラム，CD-ROM (Sep. 2007).
8) Shinoda, H., Takeuchi, S., Teranishi, Y., Harumoto, K. and Shimojo, S.: A ubiq-

⋆1 偏りありの場合も検索適合率の傾向は同様であったため，ここでは割愛している．

© 2010 Information Processing Society of Japan

uitous activity navigation method by collaborative filtering based on behavioral characteristics, *Proc. International Conf. on Computers in Education*, pp.297–302 (Nov. 2007).
9) Stoica, I., Morris, R., Kargery, D., Kaashoek, M.F. and Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications, *Proc. Association for Computing Machinery SIGCOMM Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pp.149–160 (Aug. 2001).
10) Zhao, B., Kubiatowicz, J. and Joseph, A.: Tapestry: An infrastruture for fault-tolerant widearea location and routing, *Computer*, Vol.74, pp.11–20 (June 2001).
11) Zipf, G.K.: *Human Behavior and the Principle of Least Effort*, Addison-Wesley, Cambridge, MA (Feb. 1949).

(平成 22 年 1 月 10 日受付)
(平成 22 年 6 月 3 日採録)

推　薦　文

本稿は，P2P 環境を対象に，指定した移動履歴を持つユーザを効率的に探索する方法を提案している．提案手法は，ブルームフィルタを効果的に使った新規性の高いものであり，シミュレーションにより有効性も示されている．今後，より実際的なケースを想定した評価を追加することで，有用性も示されると期待でき，推薦に値すると考えられる．

（マルチメディア，分散，協調とモバイル（DICOMO2009）シンポジウム
プログラム委員長　土井美和子）

白木　　徹
平成 21 年大阪大学工学部電子情報エネルギー工学科卒業．現在，同大学院情報科学研究科博士前期課程在学中．P2P ネットワークに関する研究に従事．

寺西　裕一（正会員）
平成 5 年大阪大学基礎工学部情報工学科卒業．平成 7 年同大学院基礎工学研究科博士前期課程修了．同年日本電信電話株式会社入社．平成 17 年大阪大学サイバーメディアセンター講師，平成 19 年同大学院情報科学研究科准教授，現在に至る．博士（工学）．マルチメディア情報システム，ユビキタス応用システム等の研究開発に従事．

竹内　　亨（正会員）
平成 13 年大阪大学基礎工学部情報科学科卒業．平成 15 年同大学院基礎工学研究科博士前期課程修了．平成 18 年同大学院情報科学研究科博士後期課程修了．博士（情報科学）．同年同研究科マルチメディア工学専攻助手，平成 19 年同助教．平成 21 年情報通信研究機構専攻研究員，現在に至る．ソーシャルネットワークやオーバレイネットワークを活用した応用情報システムの研究開発に従事．

春本　　要（正会員）
平成 4 年大阪大学基礎工学部情報工学科卒業．平成 6 年同大学院基礎工学研究科博士前期課程修了．同年大阪大学大学院工学研究科情報システム工学専攻助手．平成 11 年大阪大学大型計算機センター講師，平成 12 年同大学サイバーメディアセンター講師を経て，平成 16 年同大学大学院工学研究科助教授（平成 19 年より准教授）となり，現在に至る．博士（工学）．データベースシステム，マルチメディア情報システム等の研究に従事．電子情報通信学会，IEEE 各会員．

© 2010 Information Processing Society of Japan

西尾章治郎（フェロー）

昭和 50 年京都大学工学部数理工学科卒業．昭和 55 年同大学院工学研究科博士後期課程修了．工学博士．京都大学工学部助手，大阪大学基礎工学部および情報処理教育センター助教授，大阪大学大学院工学研究科情報システム工学専攻教授を経て，平成 14 年より大阪大学大学院情報科学研究科マルチメディア工学専攻教授となり，現在に至る．平成 12 年より大阪大学サイバーメディアセンター長，平成 15 年より大阪大学大学院情報科学研究科長，その後平成 19 年より大阪大学理事・副学長に就任．この間，カナダ・ウォータールー大学，ビクトリア大学客員．データベース，マルチメディアシステムの研究に従事．現在，Data & Knowledge Engineering 等の論文誌編集委員．本会理事を歴任．本会論文賞を受賞．電子情報通信学会フェローを含め，ACM，IEEE 等 8 学会の各会員．

© 2010 Information Processing Society of Japan