情報処理学会論文誌 Vol. 49 No. 9 3223–3233 (Sep. 2008)

推薦論文

単一ノードに複数キーを保持可能とするSkip Graph拡張

小西佑治[†1] 吉田幹[†2] 竹内亨[†1]
寺西裕一[†1] 春本要[†3] 下條真司[†4]

範囲検索に対応可能な構造化オーバレイネットワークである Skip Graph では，キーとそれを保持するピアが 1 対 1 に対応するため，端末が保持するキーが複数存在する場合に Skip Graph に参加するピアを 1 つ立ち上げるだけでは保持する複数のキーを検索の対象とすることができない．本研究では，単一ピアに複数キーを保持可能とする Multi-key Skip Graph を提案する．Skip Graph を複数キーに対応させるために，単一ノードに保持するキーと 1 対 1 に対応する仮想的なピアを複数配置した．しかし，仮想ピアを複数配置すると範囲検索時にホップ数が増大してしまうため，クエリを受けた単一ノード上に存在するすべての仮想ピアの保持するキーにより検索対象領域を分割し，各部分領域に対応する代表ノードにクエリを転送することでホップ数を減少させるマルチレンジフォワード方式による Multi-key Skip Graph を提案する．提案方式は，オープンソースプラットフォーム PIAX 上に実装し，シミュレーション評価によりピアが保持するキー数が増加しても範囲検索のホップ数増加を抑えられることを確認した．また，クエリ転送処理時間の実測により，マルチレンジフォワード方式を採用することによる処理負荷の増加は実用上問題ないことを確認した．

An Extension of Skip Graph to Store Multiple Keys on Single Node

Yuji Konishi,[†1] Mikio Yoshida,[†2] Susumu Takeuchi,[†1]
Yuuichi Teranishi,[†1] Kaname Harumoto[†3]
and Shinji Shimojo[†4]

A structured overlay network Skip Graph, which can support range retrievals, cannot treat searches of multiple keys on single peer since a search key corresponds to a peer. In this paper, we propose a Multi-key Skip Graph as an extension of Skip Graph. To store multiple keys, we set virtual peers corresponding with stored keys on single node. Query forwarding based on virtual peer causes an increase of query transfer hops between peers in range retrievals. To solve this issue, we propose a Multi-key Skip Graph based on Multi-range forwarding method, which decreases transfer hops by considering the relationship of virtual peers on a single node. Our proposal is implemented on an open-source platform PIAX, which we have been developing. The simulation result shows that our proposal suppress increasing query transfer hops and the result of measuring query transfer processing time shows that the influence of the load increase by adopting our proposal is small.

1. はじめに

ユビキタス環境では莫大な数の端末が存在するため，発生する情報量は膨大なものとなる．膨大な情報を扱ううえで，従来のサーバ集中型のアーキテクチャではサーバに負荷が集中し，サーバの設備投資や運用・保守などの負担が大きい．こうした問題に対処するための技術として，P2P（Peer-to-Peer）ネットワーク技術に注目が集まっている．

P2P ネットワークでは，各端末が各自で情報を管理し他の端末と直接通信を行うため，端末やネットワークの負荷を分散することができ，設備投資コストを抑えることが可能である．P2P ネットワークは多くの端末や情報が遍在するユビキタス環境に適したネットワーク技術の 1 つとして注目を集め，主に情報検索に焦点をあてた研究が進められている．P2P ネットワークの形態として，情報検索時に中央サーバの助けが必要なハイブリッド P2P や，中央サーバが存在しないピュア P2P が存在する．大規模な P2P ネットワークを構築する場合，ハイブリッド P2P は中央サーバがボトルネックとなるため，ピュア P2P が適している．ピュア P2P におけるオーバレイネットワークには大きく分けて非構造化オーバレイネットワークと構造化オーバレイネットワークがあり，効率的に情報を検索するには構造化

†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 株式会社ビービーアール
BBR Inc.
†3 大阪大学大学院工学研究科
Graduate School of Engineering, Osaka University
†4 独立行政法人情報通信研究機構
National Institute of Information and Communications Technology
本稿の内容は 2007 年 6 月のマルチメディア通信と分散処理研究会にて報告され，同研究会主査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．

オーバレイネットワークが適している．

構造化オーバレイネットワークには分散ハッシュテーブル（DHT）ベースのものとそれ以外のものがある．代表的な DHT には CAN [1]，Chord [2]，Pastry [3]，Tapestry [4]，Kademlia [5] などがある．DHT は任意の（key，value）の組の読み書きに対し，ある特定のハッシュ関数によってキーをハッシュ変換し，それにより決定される担当ピアと通信を行う．DHT では，ネットワーク全体の全ピア数を n とするとピアの検索に要するコストが $O(\log n)$ となる．DHT はある特定のキーに対するデータを取得可能であるが，ある範囲をキーとしてデータを取得する範囲検索を行うことはできない．これは，検索範囲内のデータをもれなく取得するには連続した検索範囲をハッシュ変換する必要があるが，キーの少しの違いにより大きく異なるハッシュ値が生成されてしまうことによる．

後者の検索に対応する構造化オーバレイネットワークとして Skip Graph [6] がある．Skip Graph は検索コストが多くの DHT と同様に $O(\log n)$ であり，ピアの参加コストについては DHT の $O(\log^2 n)$ ⋆1 よりも優れた $O(\log n)$ である．さらに，DHT とは異なりキーをハッシュ変換しないため，キーの順番が保たれ範囲検索を実現できる．しかし，Skip Graph は DHT などとは異なりキーとそれを保持するピアが 1 対 1 に対応するため，ピアはただ 1 つのキーしか保持できない．そのため，端末が複数のキーを保持する場合，端末上に Skip Graph のピアを 1 つ立ち上げるのみでは保持する複数のキーを検索の対象とすることができない．たとえば各ピアが保持する情報に付与されたキーワードにより検索を行うアプリケーションでは，ピアが複数の情報を保持する場合や情報に複数のキーワードが付与される場合，ピアが複数のキーワード（キー）を保持する必要があり，対応できない．

そこで本研究では，各端末が複数キーを保持可能とする Multi-key Skip Graph を提案する．Multi-key Skip Graph では複数キーに対応するため，保持するキーと 1 対 1 に対応する仮想的なピアを各端末に配置し，この仮想的なピアにより Skip Graph を構築する．以下，本稿では Multi-key Skip Graph に参加している端末を「物理ノード」，Multi-key Skip Graph を構成するピアを「仮想ピア」と呼ぶ．Multi-key Skip Graph では，従来の Skip Graph のルーティング手法をそのまま適用すると，実際のピア数に対してホップ数が増大してしまうという問題がある．そこで提案手法では，単一物理ノードが保持するキーのうち，検索範囲内のキーに対応する仮想ピアのルーティングテーブルを参照して，複数の物理ノードへクエリを転送することでホップ数を減少させた．提案手法はオープンソースプラットフォーム PIAX [7] に実装し，同シミュレーション機能を用いて評価を行い，その有効性を確認している．また，提案手法の処理時間を実測し，提案手法による処理負荷増加の影響が小さいことを確認した．

以降，2 章で Skip Graph の解説と研究動向について述べ，3 章で Skip Graph の複数キー拡張と拡張にともなって生じる問題について述べる．4 章で提案手法を評価し，5 章で本稿をまとめる．

2. Skip Graph と関連研究

2.1 Skip Graph

Skip Graph はデータ構造である skip list [8] を P2P ネットワークに適用したオーバレイネットワークである．skip list のノードをピアと見なし，skip list の head から tail への単方向検索を任意のピアから双方向に検索できるようにしている．Skip Graph の構成例を**図 1** に示す．図 1 において，正方形がルーティングテーブルのエントリを表し，中の数字がエントリのキーを表す⋆2．ピアはキー順に並んでおり，各ピア間は双方向にリンクしている．エントリの下にある数字は membership vector と呼ばれる各ピアに割り当てられる整数値であり，ここでは membership vector をバイナリ表現している．各エントリは複数のレベルでエントリ間をつなぐリストに属する．エントリが各レベルで属するリストは membership vector によって決まる．具体的には，レベル i（$i = 0, 1, \cdots$）において

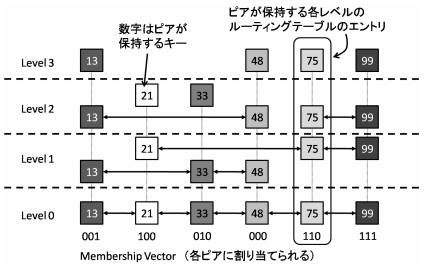


図 1　Skip Graph の構成例
Fig. 1　An example of Skip Graphs.

⋆1 Chord の場合

⋆2 ここではキーとして 10 進整数を用いたが，Skip Graph のキーには全順序関係が定義できる任意のデータ型を用いることができる．

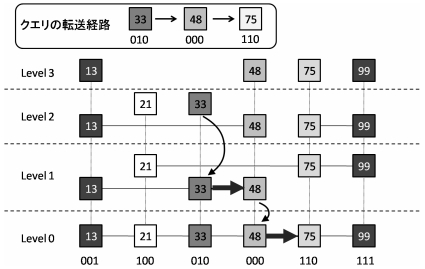


図 2　ピア 010 のキー 33 からキー 75 を検索した際のクエリ転送
Fig. 2　Query forwarding when searching key 75 from key 33 on peer 010.

membership vector の接頭 i 桁までが等しいエントリ群により各リストが形成される．そのため，この membership vector をランダムに割り当てることで，各リストに属するエントリ数のバランスが保たれる．

2.1.1　Skip Graph における検索

Skip Graph では，単一キーの検索は上位レベルから下位レベルへと行われていく．これは，上位レベルでのルーティングの方が下位レベルのそれと比べて検索キーに近づける距離が大きく，より効率的に検索が行えるためである．そこで，各ピアはクエリの届いたレベルから隣接ピアのうち検索キーを超えないピアを探す．該当するピアが見つかった場合はそのピアにクエリを転送する．該当するピアが見つからない場合は，1 つレベルを下げ，再び隣接ピアで該当するピアを探す．

たとえば図 1 において，ピア 010 のキー 33 からキー 75 を検索する場合を考える．検索開始ピアであるキー 33 のピアは，検索キーが自身のキーよりも大きいため，クエリの転送方向を右方向と決定する．そして，最上位レベルの 1 つ下位のレベルから検索キーを超えない最もレベルの高い右の隣接ピアを探す．この場合，レベル 1 においてピア 000 のキー 48 が見つかり，クエリを転送する．クエリ転送先のピア 000 のキー 48 はレベル 1 でクエリを受け，同様に検索を進める．結果としてレベル 0 でクエリを受けたピア 110 のキー 75 は検索キーと自身のキーが一致するため，検索開始ピアであるキー 33 のピアに応答を返す．以上をまとめると，クエリ転送は**図 2** のとおりとなる．もし，目的キーが検索開始ピアのキーよりも小さい場合には，左方向に同様の手順で検索を行う．単一キーの検索における平均ホップ数は $\log n$（n は全ピア数）である．

また，範囲検索の場合，ピアの保持するキーが検索範囲外であれば単一キーの検索と同様にクエリを転送する．すなわち，キーが検索範囲の下限より小さければその下限に向かって，上限より大きければその上限に向かってクエリを転送する．また，キーが検索範囲内であれば，効率良く検索範囲内のピアにクエリが届くように，上位レベルで検索範囲内に存在する隣接ピアにクエリを転送する．

2.1.2　ピアの参加・離脱

ピアが参加する場合，すでに参加しているピアが仲介する必要がある．Skip Graph ではこの仲介ピアが新規参加するピアのキーを検索し，レベル 0 における新規ピアの隣接ピアを調べ，レベル 0 のリストに新規ピアを加える．次に，レベル $i+1$（$i=0,1,\cdots$）における隣接ピア候補を知るため，レベル i において自身の左右それぞれに対して membership vector の接頭 $i+1$ 桁が同じピアを探す．もし左右どちらか一方でも該当ピアが存在するならば，今度はそのピアを基準としてさらに上位レベルの隣接ピア候補を探す．隣接ピア候補が存在しなくなるとこの処理を終了する．ピア参加における平均メッセージ数のオーダは $O(\log n)$ である．

ピアが離脱する際は，まずレベル 1 以上の隣接ピアに離脱を告知しリンクを切る．その後，レベル 0 での隣接ピアに離脱を告知し，ネットワークから離脱する．ピア離脱における平均メッセージ数のオーダは $O(\log n)$ となる．

2.2　関 連 研 究

上記で述べた Skip Graph に対する関心は近年高まりつつあり，いくつかの実現手法や改善手法の提案がなされてきている．

SkipNet [9] は Skip Graph と同様の構造を持つオーバレイネットワークである．SkipNet の特徴は，情報の格納範囲を DHT のようにネットワーク全体ではなく，近接のサブネットワークに制限できることである．このため，一般に DHT が不得意とするローカリティ制御を実現している．

Skip B-Trees [10] は，Skip Graph に B-Tree の考えを導入したものである．各レベルのリストにおいて，ピアをまとめてある大きさのブロックとして扱う．各ブロックではその中に含む最小キーと最大キーを把握しておく．キー検索時に目的キーをブロック内に含んでいればブロック内で，含んでいなければ次のブロックにクエリを転送することで，検索を効率化している．

Rainbow Skip Graph [11] は，Skip Graph のレベル 0 のリストにおいて，ピアをいくつかまとめた core list を作り，各 core list を仮想的に 1 つのピアと見なして上位レベルで Skip Graph を構成する．このとき，各レベルにおける隣接ピアへのリンクを core list に含まれ

るピアで分担して管理することにより，Skip Graph よりも各ピアの保持するリンク数を減らしている．

一方，多次元の範囲検索を行う目的で Skip Graph を活用した研究[12),13)] もある．Skip Graph が扱うキーは 1 次元であるため，対象空間の 1 次元化が鍵となる．SkipIndex [12)] は，多次元のハイパーキューブを K-D tree [14)] を用いて分割し，1 次元に整列させる手法，Znet [13)] は空間充填曲線である Z-ordering によって 1 次元へ写像する手法をとっている．

以上示したとおり，従来の Skip Graph 拡張の研究は Skip Graph の効率化や多次元検索への対応が主であり，Skip Graph を構成するピアがただ 1 つのキーを保持している点は変わりない．しかし，実際の利用を想定すると各ピアに複数のキーを持たせて複数の情報を検索可能としたい場合が多く，単一ピアに複数キーを保持可能とすることに対する要求は高いと考えられる．

3. Skip Graph の複数キー拡張

3.1 Multi-key Skip Graph

Skip Graph のピアに複数キーを保持させる単純な方法として，各端末上においてピアに相当するプロセスを保持するキーの数だけ立ち上げることが考えられる．しかし，これでは多数の情報を発信するために同数のピアを立ち上げる必要があり，メモリ消費量の増大，ピア管理の手間などの問題が生じる．そこで，membership vector が一致する仮想的なピアを端末が保持するキー数だけ配置し，この仮想的なピアによる Skip Graph を構成することを考える．各物理ノードは保持するキーと 1 対 1 に対応する仮想ピアを配置し，物理ノード上の各仮想ピアのルーティングテーブルを管理する．ここでは，この拡張を行った Skip Graph を Multi-key Skip Graph と呼ぶ．Multi-key Skip Graph の構成例を**図 3** に示す．以降では，物理ノードを「P(*membership vector*)」，仮想ピアを「V(*membership vector*，キー)」のように表すこととする．たとえば，membership vector が 01 である物理ノードは「P(01)」，この物理ノード上のキー 8 を保持する仮想ピアは「V(01, 8)」のように表す．

Multi-key Skip Graph に通常の Skip Graph のルーティング手法，すなわち，仮想ピア単位のルーティングを適用した場合，物理ノード上に複数の仮想ピアが存在すると同じ物理ノードが複数回クエリを受け取ってしまうことが考えられる．たとえば，図 3 において P(00) から検索範囲を [1, 6] とした範囲検索を行う場合を考える．なお，ここでは検索元の物理ノード上の仮想ピアが複数検索範囲に含まれる場合，検索範囲に含まれる仮想ピアの

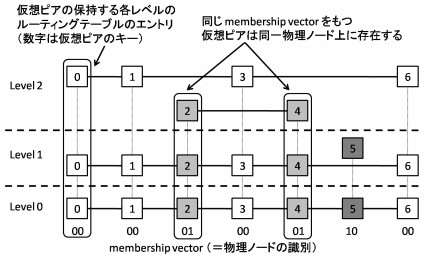


図 3　Multi-key Skip Graph の構成例
Fig. 3　An example of Multi-key Skip Graphs.

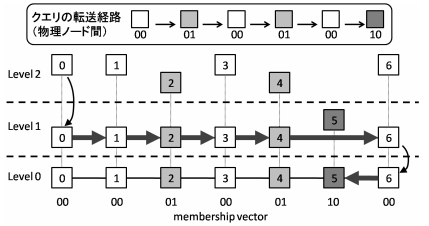


図 4　仮想ピア単位のルーティング手法による範囲検索
Fig. 4　Query forwarding of range retrieval when routing based on virtual peers.

うちキーが最小のものからクエリ転送を行うものとする．そのため，最初のクエリ転送は V(00, 0) から行われる．仮想ピア単位のルーティングでは，最上位レベルの 1 つ下位のレベルであるレベル 1 から検索範囲に含まれる隣接ピアを探し，同一物理ノード上の V(00, 1) にクエリを転送する．その後，同様にしてクエリは V(01, 2)，V(00, 3)，V(01, 4)，V(00, 6)，V(10, 5) と転送されていく．このとき，P(00) と P(01) の間で何度もクエリが往復していることが分かる．クエリ転送の様子を**図 4** に示す．

Skip Graph の検索に要するホップ数が $O(\log n)$（n は全仮想ピア数）になるためには，membership vector の分布がランダムである必要がある．しかし，単一ピアが複数キーを保持することにより，同一の membership vector を持つピア（仮想ピア）が増大し，membership vector の分布に偏りが生じるため，ホップ数が $O(\log n)$ より悪化する．すなわち，単一ピアが複数キーを保持すると，あるリストに属する物理ノード数に対して同じリストに属する仮想ピア数が増大する．このとき，異なる物理ノード上の仮想ピアが互い違いにリス

ト上に配置されていると，クエリが到達するのに要するホップ数が増大し，$O(\log n)$ より悪化する．

この問題は単一キー検索でも生じるが，特に範囲検索の場合に影響が大きくなる．範囲検索では最上位レベルからいくつか下位のレベルになって検索範囲内の仮想ピアにクエリが届き，検索範囲内の最初の仮想ピアが転送したレベルでクエリが多数転送される．Multi-key Skip Graph では下位レベルほど各リストに属する仮想ピア数が増大する．このため，単一キー検索の場合よりも多くの仮想ピアをクエリが経由することになり，ある物理ノードが同じクエリを複数回受け取ってしまう可能性が高くなる．

3.2　マルチレンジフォワード方式

3.1 節で述べたホップ数増大の問題に対処するため，ある物理ノードはクエリを 1 度しか処理しないルーティングを考える．ある検索範囲に関して初めてクエリを処理する場合，その物理ノードが保持するキーのうち検索範囲に含まれるキーにより部分領域を定義する．そして，各部分領域についてクエリ転送処理を行うことで，1 度のクエリ処理により検索範囲に含まれるすべての仮想ピアに関してクエリ処理を行うことができる．

まず，クエリを受け取った物理ノードはクエリの検索範囲をその検索範囲に含まれるキーにより区切る．そして，この区切られた検索範囲それぞれを部分領域とし，各部分領域内で転送先となる代表物理ノードを 1 つ決定する．部分領域は「クエリの検索範囲の下限から検索範囲内の最小キー」,「検索範囲内の隣り合う 2 つのキーの間」,「検索範囲内の最大キーからクエリの検索範囲の上限」の 3 つの場合がある．部分領域が「クエリの検索範囲の下限から検索範囲内の最小キー」か「検索範囲内の最大キーからクエリの検索範囲の上限」のとき，クエリを受けたレベル以下において，自身とは異なる物理ノード上の対象部分領域内のキーを持つ仮想ピアのうち，最も高いレベルで隣接する仮想ピアが存在する物理ノードを代表物理ノードとする．部分領域が「検索範囲内の隣り合う 2 つのキーの間」の場合は，小さいキーを持つ仮想ピアと大きいキーを持つ仮想ピアの両方の隣接ピアの中から，同様に代表物理ノードを決定する．各部分領域の代表物理ノードが同一の場合は，その代表物理ノードへ複数の部分領域情報（部分領域の最小キーと最大キー）をまとめて送ることにより物理ノード間の通信回数を削減する．以上述べた方法によりクエリ転送を行うことで，1 度処理した検索範囲に含まれる範囲に関してはクエリを再度受け取ることがなくなる．1 度のクエリ処理で複数の部分領域にクエリを転送することから，ここではこの方法をマルチレンジフォワード方式と呼ぶ．

図 3 において，P(00) から検索範囲を [0, 6] とした範囲検索をマルチレンジフォワード方式によって行った場合のクエリ転送の様子を**図 5** に示す．図 5 において，検索開始物理ノードである P(00) は検索範囲 [0, 6] を自身の保持する検索範囲内のキーにより部分領域 A，B，C に分割し，各部分領域 A，B，C において代表物理ノードを決める．部分領域 A には隣接する仮想ピアが存在しないため代表物理ノードは存在しない．部分領域 B では隣接する仮想ピア V(01, 2) が部分領域 B 内のキーを持つ自身と異なる物理ノード上に存在するため，代表物理ノードは P(01) となる．部分領域 C では隣接する仮想ピアは V(01, 4) と V(10, 5) の 2 つが存在するが，より高いレベルで隣接しているのは V(01, 4) であるため，代表物理ノードは P(01) となる．ここで，部分領域 B と C の代表物理ノードが同じであるので，P(01) へ 1 度のクエリ転送で P(01) が検索すべき範囲として部分領域 B と C の情報をまとめて送る．クエリを受け取った P(01) は受け取った検索すべき範囲である部分領域 B，C について同様に自身のキーにより部分領域に分割後代表物理ノードを決定して，部分領域 C2 の情報を P(10) に転送する．

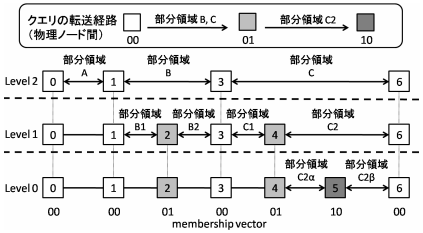


図 5　マルチレンジフォワード方式による範囲検索
Fig. 5　Query forwarding of range retrieval when routing by multi-range forwarding.

図 6 にマルチレンジフォワード方式の擬似コードを示す．擬似コードにおいて，forwardQuery() が各物理ノードにおけるルーティング処理を指す．2 行目から 5 行目にかけて，受け取った検索範囲に対して自身の物理ノード上の全仮想ピアのキーのうち検索範囲内に含まれるものを調べ，得られたキーにより受け取った検索範囲を分割して部分領域を求めている．7 行目から 10 行目にかけて，各部分領域について代表物理ノードを求め，代表物理ノードと送るべき部分領域情報のマップを作成している．12 行目から 16 行目にかけて，各代表物理ノードに先ほど作成したマップを基に部分領域情報をまとめて送信している．このとき，14 行目において転送先の forwardQuery() を呼び出すが，非同期呼び出しのため転送先のクエリ処理結果の応答を待たずに次の処理へ進む．このため，results には一時的に転送先の情報のみを格納し，応答があり次第得られた転送先のクエリ処理結果を格納する．

```
ranges  クエリに含まれる検索範囲（部分領域）の集合
query  クエリ処理を行うための条件
localKeys  検索範囲に含まれる仮想ピアのキーの集合
er, externalRanges  検索範囲を localKeys で分割した部分領域の集合
cr, classifiedRanges  代表物理ノードと転送する部分領域の対応表
results  localKeys のクエリ処理結果と転送先のクエリ処理結果の集合
---
getKeysIn()  localKeys を求める処理
getExternalRanges()  externalRanges を求める処理
getDelegatePeer()  部分領域の代表物理ノードを求める処理
rpcStub(peer).forwardQuery()
   リモートピア peer 上での forwardQuery() の非同期呼び出し
execQuery()  localKeys に対してクエリ処理結果を求める処理
---

01 forwardQuery(ranges, query) {
02   for each range in ranges {
03     localKeys.add(getKeysIn(range));
04     er.add(getExternalRanges(range));
05   }
06
07   for each range in externalRanges {
08     peer = getDelegatePeer(range);
09     cr.get(peer).add(range);
10   }
11
12   for each peer in classifiedRanges {
13     remoteResults =
14       rpcStub(peer).forwardQuery(cr.get(peer), query);
15     results.add(remoteResults);
16   }
17
18   thisResults = execQuery(localKeys, query);
19   results.add(thisResults);
20
21   return results;
22 }
```

図 6　マルチレンジフォワード方式の擬似コード
Fig. 6　Pseudo-code of multi-range forwarding algorithm.

18 行目から 19 行目にかけて，受け取った検索範囲内に含まれるキーに対してクエリを処理し，その結果を results に格納している．最後に 21 行目で forwardQuery() の返り値として results を転送元に送信している．

3.3　マルチレンジフォワード方式のホップ数が $O(\log m)$（m は全物理ノード数）となることの証明

本節ではマルチレンジフォワード方式において検索に要するホップ数の平均が $\log m$（m は全物理ノード数）となることを示す．単一キー検索に要するホップ数の平均は次の命題 1 により $\log m$（m は全物理ノード数）となることから，部分領域に分解されながら伝播される仮想ピアの列が単一キー検索の場合の伝播ピア列に相当することより，同様に平均ホップ数が，$\log m$ となることが分かる．

[補題 1] Multi-key Skip Graph を構成する物理ノードの集合を $\mathbf{P} = \{P_1, P_2, \cdots, P_m\}$ とする．各物理ノード P_i（$1 \leq i \leq m$）において，所有するキーを 1 つのキーに限定した物理ノードを P_i' としたとき，P_i' からなる物理ノードの集合 $\mathbf{P}'$（これを $\mathbf{P}$ の縮退と呼ぶ）により構成される Multi-key Skip Graph において，キーの検索に要する平均ホップ数は $\log m$ となる．

[証明] 物理ノード集合 $\mathbf{P}'$ が，Skip Graph を構成することより，補題が成立する．

[命題 1] Multi-key Skip Graph のキー検索において，クエリが伝播される仮想ピアの列を $X_1(k_1)$，$X_2(k_2)$，$\cdots$，$X_l(k_r)$ とする．ここで，X_i（$1 \leq i \leq r$）は仮想ピア，k_i は仮想ピアが保持するキーを示す．任意の $i < j$（$1 \leq i$，$j \leq r$）において，$X_i \neq X_j$ かつ $k_i \leq k_j$ となるとき，r の平均値は $\log m$ となる．

[証明] Multi-key Skip Graph を構成する物理ノードの集合を $\mathbf{Q}$ とする．すべての i（$1 \leq i \leq r$）について，$Q_i = X_i$ なる物理ノード Q_i の所有するキーを k_i とする $\mathbf{Q}$ の縮退を $\mathbf{Q}'$ とする（$i < j$ において，$Xi \neq Xj$ が成立することより，この縮退は必ず存在する）．X_i 列がキー検索におけるクエリの伝播仮想ピア列であることより，i（$1 \leq i \leq r$）について，$X_{i+1}(k_{i+1})$ は，X_i に隣接し，$k_i \leq k_{i+1} \leq k_r$ が成り立つ X_{i-1} と X_i 間の共通桁数（$i = 0$ のときは，membership vector の桁数）を超えない最大桁の共通接頭辞を持つ．さらに，$i < j$ において $k_i \neq k_j$ となることより，$\mathbf{Q}'$ により構成される Skip Graph において，X_1 を開始点，k_r を検索キーとした場合，クエリの伝播仮想ピア列は，X_i 列と一致することが分かる．補題 1 より，k_r の検索に要する平均ホップ数は，$\log m$ となる．したがって，r の平均値は $\log m$ となる．

4.　評　　価

3 章で示したマルチレンジフォワード方式による Multi-key Skip Graph をオープンソースプラットフォーム PIAX に実装した．PIAX は大阪大学が開発を行っている P2P エージェントプラットフォームであり，センサへの応用[15] や見守り・監視システム[16] などの実用アプリケーションに利用されている．PIAX は Java により開発が行われており，本研究では PIAX に実装されている Skip Graph を改良することで Multi-key Skip Graph を実装し，PIAX のシミュレーション機能を利用して仮想的な通信路により単一 PC 上で評価を行った．

4.1　Multi-key Skip Graph と通常の Skip Graph，DHT との比較

マルチレンジフォワード方式による Multi-key Skip Graph は，通常の Skip Graph の特徴を有しながら，検索に関する性能改善を行ったものである．そのため，DHT と比較した場合，通常の Skip Graph と同様の利点がある．すなわち，仮想ピアの追加・削除に要する平均メッセージ数は DHT よりも少なく，検索に要する平均メッセージ数は物理ノード数のオーダになるため DHT と同等であり，範囲検索が可能である．また，各物理ノードが保持するキー数が多くなるほど，範囲検索における検索経路が多分木になる可能性が高くなり通常の Skip Graph よりも検索に要するホップ数が削減される．提案手法では複数キー保持によりルーティングテーブルが増大するが，保持するキー数だけ物理ノードを立ち上げる場合と比較すると通信などの機能については 1 つのプログラムでよいため，Multi-key Skip Graph の方が全体としてメモリの消費量は少なくなる．

4.2　範囲検索に要するホップ数の評価

範囲検索に要するホップ数を仮想ピア単位のルーティングと比較評価した．ここでのホップ数とは枝分かれして広がっていくクエリの転送経路のうち最も多くの物理ノードを経由した経路のホップ数である．ただし，同一物理ノード上の仮想ピア間でのクエリ転送はホップ数をカウントしていない．本評価では物理ノード数 100 の Multi-key Skip Graph において各ピアにキーとして整数値を一様乱数により与えた．検索範囲は乱数のとりうる値すべてを含む範囲とし，Multi-key Skip Graph の構造を 10 回，また同一構造の Multi-key Skip Graph において検索元となる物理ノードを 10 回変えながら最大ホップ数の平均値を求めた．また，仮想ピア単位のルーティングにおいて検索元の物理ノード上に存在する仮想ピアが検索範囲に複数含まれる場合，どの仮想ピアから最初のクエリ転送を行うかを決める必要がある．検索範囲内の仮想ピアのうちキーが最小の仮想ピアからクエリ転送を行う方法では，検索範囲内に含まれる保持キー数が多いと範囲検索に要するホップ数が非常に悪化する場合がある．そのため，検索範囲内のキーの中央値（ただし検索範囲内の仮想ピア数が偶数の場合は中央 2 値のうち小さい方）であるキーを保持する仮想ピアからクエリ転送を行うものとした．中央値のキーの仮想ピアからのルーティングにより，同一レベルにおけるホップ数を最小キーの仮想ピアからのルーティングよりも少なくできる．同一物理ノード数の Multi-key Skip Graph において各ピアが保持するキー数を変化させたときの範囲検索に要するホップ数の変化を**図 7** に示す．

図 7 において，仮想ピア単位のルーティングでは各ピアが保持するキー数の増加とともに急激にホップ数が増加している．しかし，マルチレンジフォワード方式ではホップ数の増加は見られるものの一定値へ収束に向かっている．物理ノード数が変化しない本評価ではマルチレンジフォワード方式のホップ数 $O(\log m)$（m は物理ノード数）が定数オーダになるため，図 7 のように収束する．また，この 2 つのグラフの差は 3.1 節で述べたように複数物理ノード間でのクエリの往復回数の差によるものである．物理ノード数 10，各物理ノードが保持するキー数 5 と評価環境とは異なるが，Multi-key Skip Graph で範囲検索を行った場合の仮想ピア単位のルーティングとマルチレンジフォワード方式の転送経路を**図 8** に示す（ただし，各物理ノードの最上位レベルのリストは省略している）．各図形は仮想ピアを表しており，同じ形の仮想ピアは同一物理ノード上に存在する．縦方向に線で結ばれた仮想ピアは同一の仮想ピアであり，上に行くほど高レベルのリストに属していることを表す．また，太線がクエリの転送経路である．図 8 (a) と図 8 (b) を比較すると，明らかに仮想ピア単位のルーティングの場合の方が横向きに長く延びた転送経路が多く，マルチレンジフォワード方式ではこの横向きに長い転送経路が減少していることが分かる．このように，マルチレンジフォワード方式では同一レベルでのクエリ転送回数の増加を抑え，少ないホップ数で検索範囲内にクエリを届けることが可能である．

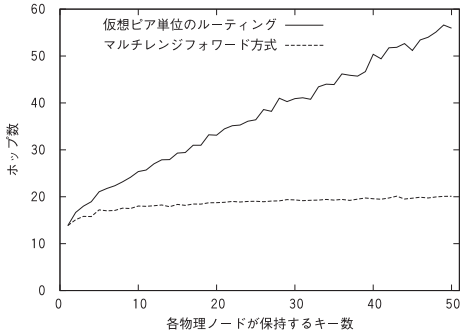
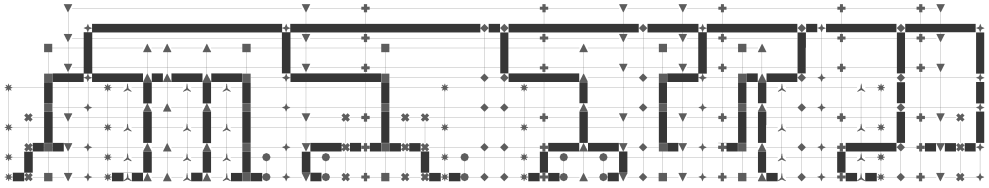


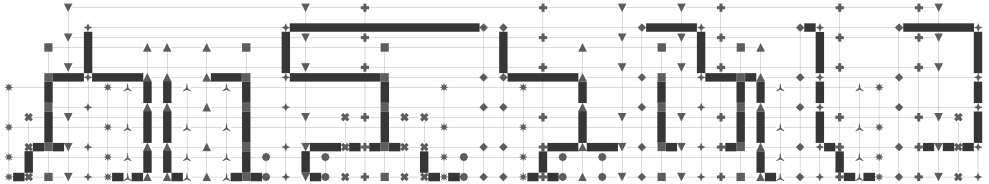
図 7　物理ノード数 100 の Multi-key Skip Graph における範囲検索に要するホップ数の変化
Fig. 7　Tendency of hops of range retrieval on Multi-key Skip Graph with 100 physical nodes.

4.3　処理時間の評価

各ピア内での転送処理時間を Java（J2SE 5.0）により実装した仮想ピア単位のルーティングとマルチレンジフォワード方式とで比較した．この評価は Core2Duo T7200 2 GHz の PC 上で行い，物理ノード数 100 の Multi-key Skip Graph において各物理ノードに範囲検索のクエリを与えたときの処理時間を計測した．与えるクエリは，物理ノードが保持する



(a) 仮想ピア単位のルーティング手法



(b) マルチレンジフォワード方式

図 8　物理ノード数 10，保持キー数 5 の Multi-key Skip Graph における範囲検索の転送経路
Fig. 8　Query forwarding path of range retrieval on Multi-key Skip Graph with 10 physical nodes which have 5 keys each.

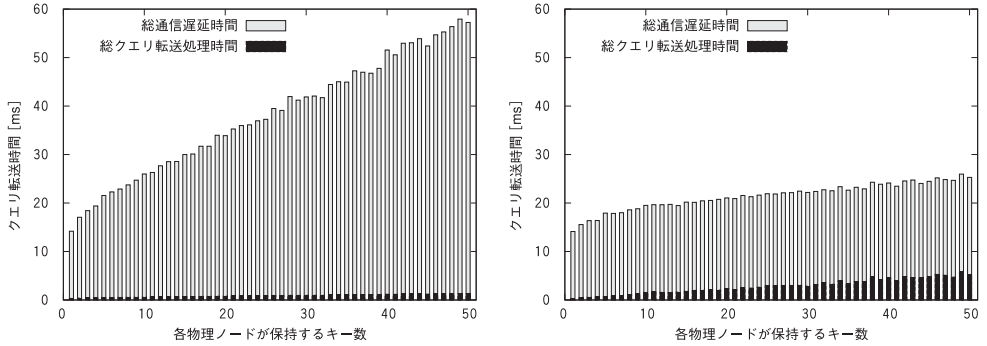
キーのうち検索範囲に含まれるキーの数，物理ノード上の仮想ピアのルーティングテーブルのレベルのうちクエリが転送されるレベルをそれぞれ乱数により変化させた．仮想ピア単位のルーティングでは 1 度のクエリ転送処理で 1 つのキーを処理する．そのため，処理時間は物理ノードの保持キー数に関係なく，仮想ピアの保持するキーが検索範囲外の場合は平均 9 μs，検索範囲内の場合は平均 25 μs 程度であった．一方，マルチレンジフォワード方式では 1 度のクエリ転送処理で検索範囲内のすべてのキーを処理するため，処理時間は保持するキー数が多いほど長くなる．しかし，マルチレンジフォワード方式ではクエリ転送処理時間が増加しているとはいえ，各物理ノードが 50 キーを保持している状況においても 250 μs 程度であった．

ここで実際の利用を想定した通信遅延時間についても考える．1 ホップに要する通信遅延時間は LAN 環境でミリ秒オーダであるが，ここで 1 ms と仮定したとき，物理ノード数 100，保持キー数 50 の Multi-key Skip Graph におけるすべてのキーを含む範囲の範囲検索に要する時間を計算したものを**図 9** に示す．図 9 (a) より，仮想ピア単位のルーティング手法ではキー数に応じてホップ数（転送回数）が増加するため，総クエリ転送処理時間は短くてもキー数の増加に応じて総通信遅延時間の増加が激しく，範囲検索に要する時間が長いことが分かる．一方，図 9 (b) よりマルチレンジフォワード方式では総クエリ転送処理時間は増加するが，キー数の増加に対するホップ数の増加が緩やかであるため総通信遅延時間の増加が抑えられ，範囲検索に要する時間は短くなることが分かる．これはクエリ処理時間がマイクロ秒オーダであるのに対して通信遅延時間がミリ秒オーダであるため，総通信遅延時間が範囲検索に要する時間に大きく影響するためである．実際のネットワーク環境では 1 ホップあたりの通信遅延時間は仮定した 1 ms よりも大きいと考えられ，ホップ数の差が与える総通信遅延時間への影響はさらに大きくなる．よって，マルチレンジフォワード方式によるクエリ転送処理時間増加の影響は，実際のネットワーク環境において十分小さいといえる．

5. ま と め

本稿ではオーバレイネットワークである Skip Graph の複数キー化による検索時の問題点を考慮し，特に範囲検索におけるクエリ転送を最適化した Multi-key Skip Graph を提案した．また提案手法をオープンソースプラットフォーム PIAX により実装し，シミュレーション評価を行った．また，提案手法を採用することによる処理負荷増加の影響についても評価を行った．

提案手法により Skip Graph を単一ピアの保持するキーは 1 つであるという制限を気にすることなく利用可能となった．さらに，単一ピアに複数キーを保持可能としたことで，複数



(a) 仮想ピア単位のルーティング手法　(b) マルチレンジフォワード方式

図 9　物理ノード数 100 の Multi-key Skip Graph における範囲検索に要する時間の変化
Fig. 9　Tendency of consumed time of range retrieval on Multi-key Skip Graph with 100 physical nodes.

種別のキーに対する範囲検索を行うことが可能となる．これは Skip Graph が全順序関係さえ定義されていればキーのデータ型が任意でよいためである．これを利用し，文献 7) では Multi-key Skip Graph を利用して目的別のオーバレイネットワークを複数実装している．

本稿で示した手法は，オープンソースプラットフォーム PIAX に実装されており，PIAX 公式ウェブサイト[⋆1]よりソフトウェアを入手可能である．実装方式などの詳細については，ソースコードを参照されたい．

今後の課題としては，単一キー検索や範囲検索以外のより複雑な検索を可能とするための OR 条件などを含んだ複合検索の扱いなどがあげられる．また，Multi-key Skip Graph では 1 つの物理ノードの障害が複数の仮想ピアの障害につながる．そのため，Skip Graph よりも高い耐性の検討が必要となる．今後，上記課題に取り組むとともに，さらなる検討を進め，実用的なオーバレイ技術基盤の確立を目指したい．

謝辞　本研究の一部は，平成 18，19 年度総務省委託研究「ユビキタスネットワーク認証・エージェント技術の研究開発」の一環として実施したものである．

参 考 文 献

1) Ratsanamy, S., Francis, P., Handley, M. and Karp, R.: A Scalable Content-Addressable Network, *ACM SIGCOMM Conference*, pp.161–172 (2001).
2) Morris, R., Karger, D., Kaashoek, F. and Balakrishnan, H.: Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications, *ACM SIGCOMM 2001*, San Diego, CA (2001).
3) Rowstron, A. and Druschel, P.: Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems, *Lecture Notes in Computer Science*, Vol.2218, pp.329–350 (2001).
4) Zhao, B., Kubiatowicz, J. and Joseph, A.: Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing, *Computer*, Vol.74 (2001).
5) Maymounkov, P. and Mazieres, D.: Kademlia: A peer-to-peer information system based on the XOR metric, *Proc. 1st International Workshop on Peer-to-Peer Systems* (*IPTPS'02*), Vol.258, p.263 (2002).
6) Aspnes, J. and Shah, G.: Skip graphs, *14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp.384–393 (2003).
7) 吉田　幹，奥田　剛，寺西裕一，春本　要，下條真司：マルチオーバレイと分散エージェントの機構を統合した P2P プラットフォーム PIAX，情報処理学会論文誌，Vol.49, No.1 (2008).
8) Pugh, W.: Skip Lists: A Probabilistic Alternative to Balanced Trees, *Workshop on Algorithms and Data Structures*, pp.437–449 (1989).
9) Harvey, N.J.A., Jones, M.B., Saroiu, S., Theimer, M. and Wolman, A.: SkipNet: A Scalable Overlay Network with Practical Locality Properties, *4th USENIX Symposium on Internet Technologies and Systems* (*USITS '03*) (2003).
10) Abraham, I., Aspnes, J. and Yuan, J.: Skip B-trees, Principles of Distributed

⋆1 http://www.piax.org/

Systems, *9th International Conference, OPODIS 2005*, Pisa, Italy, December 2005; Revised Selected Papers, Lecture Notes in Computer Science, Vol.3974, pp.366–380 (2005).
11) Goodrich, M.T., Nelson, M.J. and Sun, J.Z.: The rainbow skip graph: a fault-tolerant constant-degree distributed data structure, *Proc. 17th Symp. Discrete Algorithms*, ACM and SIAM, pp.384–393 (2006).
12) Zhang, C., Krishnamurthy, A. and Wang, R.Y.: SkipIndex: Towards a Scalable Peer-to-Peer Index Service for High Dimensional Data, Technical Report TR-703-04, Technical Report, Department of Computer Science, Princeton University (2004).
13) Shu, Y., Ooi, B.C., Tan, K.-L. and Zhou, A.: Supporting Multi-Dimensional Range Queries in Peer-to-Peer Systems, *P2P'05: Proc. 5th IEEE International Conference on Peer-to-Peer Computing* (*P2P'05*), Washington DC, USA, IEEE Computer Society, pp.173–180 (2005).
14) Bentley, J.L.: Multidimensional binary search trees used for associative searching, *Comm. ACM*, Vol.18, No.9, pp.509–517 (1975).
15) 石芳　正, 竹内　亨, 寺西裕一, 春本　要, 下條真司：P2P エージェントプラットフォーム PIAX のセンサー応用，マルチメディア，分散，協調とモバイル（DICOMO2007）シンポジウム論文集，p.1810 (2007).
16) Teranishi, Y., Tanaka, H., Ishi, Y. and Yoshida, M.: A Geographical Observation System based on P2P Agents, *Proc. 6th Annual IEEE International Conference on Pervasive Computing and Communications* (*PerCom 2008*) *Workshops ATPC 2008*, pp.615–620 (2008).

(平成 19 年 12 月 5 日受付)
(平成 20 年　6 月 3 日採録)

推　薦　文

本稿では，範囲検索に対応可能な構造化オーバレイネットワークである Skip Graph を拡張し，単一ピアに複数キーを保持可能とする Multi-key Skip Graph を実現する方法を提案している．実現方式として，単一ピア上の仮想ピアが持つキーの範囲情報をもとに複数レンジのクエリをまとめて送信することでホップ数を減少させるマルチレンジフォワード方式を提案している．提案方式は実用的な分散範囲検索手法として新規性が高く，また，オーバレイネットワークとしての有用性も高いと考えられ，推薦に値する．

（マルチメディア通信と分散処理研究会主査　櫻井紀彦）

小西　佑治

平成 19 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．

吉田　　幹（正会員）

昭和 61 年京都大学大学院工学研究科情報工学専攻博士後期課程修了．同年日本アイ・ビー・エム株式会社入社．東京基礎研究所勤務．平成 6 年新日鉄ソリューションズ株式会社入社，平成 14 年株式会社ビービーアール設立，現在に至る．電子情報通信学会，人工知能学会各会員．

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，現在に至る．ソーシャルネットワークを活用した情報システムの研究開発に従事．電気学会会員．

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．

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．

下條　真司（正会員）

昭和 56 年大阪大学基礎工学部情報工学科卒業．昭和 61 年同大学院基礎工学研究科博士後期課程修了．博士（工学）．同年大阪大学基礎工学部情報工学科助手．平成 1 年同大学大型計算機センター講師．平成 3 年同助教授．平成 10 年同教授．平成 12 年同大学サイバーメディアセンター教授．平成 20 年独立行政法人情報通信研究機構上席研究員．LAN のアクセス方式の性能評価，分散処理システムの性能評価，分散型オペレーションシステムの研究に従事．電子情報通信学会，ACM，IEEE，ソフトウェア科学会各会員．