情報処理学会論文誌 Vol. 51 No. 4 1142–1151 (Apr. 2010)

時間に基づく階層化とValueの集約配置手法による耐Churnオーバレイネットワーク

洞井晋一[†1,†2] 松浦知史[†1,†3]
藤川和利[†1] 砂原秀樹[†1,†4]

オーバレイネットワークの研究分野では分散ハッシュテーブルに関する研究がさかんであり，特にトポロジに制約を加えた構造化オーバレイネットワークが注目されている．しかし，構造化オーバレイネットワークはノードの頻繁な参加と離脱（Churn）により，検索クエリや検索対象（Value）の喪失によって検索成功率の低下が問題視されている．また，その単純な構造から複雑な検索機能を提供することが困難であったり，ノードの特性を考慮せずに負荷を均等に分散するなどの問題もあげられる．本論文ではこれらの問題に対しValueの配置方法を変えることで成功率の低下・複雑な検索機能の提供・ノード特性を考慮した負荷分散の問題を解決する．提案手法ではオーバレイネットワークをノードの参加時間に応じて階層化し，Valueを時間が経過するに従って安定した階層へと集約しつつ配置する．この集約の効果によって複雑な検索を少ないクエリで解決することが期待できるほか，参加時間の長い計算機にはValueの集合体を割り当てることで，計算機の特性に応じた負荷の割当てを実現する．ノードの生存時間分布にWeibull分布を用いた実験の結果，提案手法はDHTと比べ検索の成功率が約35%向上した．また，従来のものよりネットワーク上を流れるメッセージ数が約10%増加したが，性能には問題のない範囲であり，Valueの内容や検索の頻度によっては，ネットワーク上を流れるトラフィックの削減を期待できる．

Churn Tolerant Overlay Network Using Time Layered and Time Aggregation Methods

Shinichi Doi,[†1,†2] Satoshi Matsuura,[†1,†3]
Kazutoshi Fujikawa[†1] and Hideki Sunahara[†1,†4]

Lots of overlay network based on DHT (Distributed Hash Table) have been developed in recent years. There are some overlay networks called “Structured Overlay Network” which have a topology restriction. However, we think that DHTs have the following two drawbacks. DHTs do not think about the heterogeneity of the characteristics, which is contains Churn behavior, of nodes, and do not support complex queries. The purpose of this paper is to describe the design and evaluation of our overlay netowork, an id assignment method for a hierarchical overlay network. Our solution forms a multi-ring topology using a session time of a node. The reason why we consider a session time of a node is that heavy users tend to use a high-performance PCs. Key-values are gradually gathered in a high layer ring as time passes, and our solution supports multi-attribute queries with these key-values. The experiment result, which is used Weibull distribution as node session time, shows that our hierarchical overlay network achieve approximately 35% success rate increase. And the number of messages is approximately 10% increase, however it have no practical impact on real networking.

1. まえがき

従来のサーバ・クライアント型のサービス提供方式に対して，サーバを介さないP2P型のサービス提供が注目を集めている．P2P型のサービスはサーバにおける負荷軽減や耐故障性・可用性の向上などのほか，近年ではトラフィックの低減にもつながると期待されている．また，P2P型のサービス提供は物理的なネットワーク構成の上に新たにネットワークを構築することから「オーバレイネットワーク」と呼ばれている．オーバレイネットワークの研究分野では，分散ハッシュテーブル（DHT: Distributed Hash Table）に関する研究がさかんであり，トポロジに制約を加えた構造化オーバレイネットワークが注目されている．構造化オーバレイネットワークでは従来の構造化しないものと比べて以下の点で優れているといわれる．

- 検索結果に対する応答が保証されている．
- 負荷を均等に分散する．

構造化オーバレイネットワークでは検索対象となるデータ，もしくはデータへのポイン

†1 奈良先端科学技術大学院大学情報科学研究科
Graduate School of Information Science, Nara Institute of Science and Technology
†2 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University
†3 情報通信研究機構
National Institute of Infromation and Communications Technology
†4 慶應義塾大学大学院メディアデザイン研究科
Graduate School of Media Design, Keio University

タ（Value）をどの計算機（ノード）が管理するかが明確であるため，検索クエリに対して Value の有無にかかわらず，応答を得ることができる．また，暗号学的ハッシュ関数を用いることで割当てを均一に行うため，負荷が参加しているノードに均一に割り当てられる．

これらの性質から，主にサーバの冗長化技術として構造化オーバレイネットワークは注目されているが，一般のユーザも参加するような P2P ネットワークを構成するためには以下のような問題点が残されている．

- ノードの Churn 動作（参加と離脱）による検索成功率の低下
- 計算機の性能を考慮しない Value の割当て
- 範囲検索などの複雑な検索が困難

サーバの冗長化と違い，ユーザのノードによるオーバレイネットワークでは頻繁な Churn 動作を考慮しなければならない．構造化オーバレイネットワークはこの Churn 動作が頻繁に発生した場合，ノードの離脱にともなう Value の喪失によって，検索の成功率が低下する．これは，Value を配置するノードがハッシュ関数によって明確に定まるため，ノードの特性を考慮した Value の配置を行えないためである．また，計算機の特性を考慮しない負荷の分散は，サーバなどが参加するハイブリッドな環境や，性能の高い計算機が参加している環境でそれらの計算機の性能を活用することもできない．これらに加え，ハッシュ関数は Value 間の関連性を考慮しないため，複雑な検索を行うためにはクエリをフラッディングさせる必要がある．クエリのフラッディングはトラフィックの増大を招くばかりか，クエリの到達先ノードや中継ノードが増えることで Churn 動作の影響を受けやすくなり，これも検索成功率の低下を招くことになる．また，Key の範囲検索を行うためには Key の始点と終点をあらかじめ定義する必要があり，その定義によっては Key の偏りやクエリの増大を招いてしまう．特に時間情報のような始点と終点を定義することが困難な属性は範囲検索そのものも非常に困難である．

本論文ではこれらの問題を，ノードの参加時間に基づく階層化と時間経過にともないデータを集約する手法により解決する．前提条件として，提案するオーバレイネットワークは以下の環境で構築されるものとする．

- サービス提供者などによる高性能計算機
- サービスのヘビーユーザの提供する高性能計算機
- 一般ユーザによる計算機

オーバレイネットワークではサービスレベルの維持のために，サービス提供者が計算機を提供することが現実的である．それらの計算機は高性能なものを期待でき，また参加時間も非常に長い．また，サービスを利用するユーザには高性能な計算機を保有し，かつ参加時間の長いものも想定できる[14]．これらに一般ユーザの計算機を加え，オーバレイネットワークを構築し，特に高性能計算機を有効に利用することで検索の成功率向上を主な目的とする．提案するオーバレイネットワークはリング状のオーバレイネットワークをノードの参加時間を用いて階層化し，参加時間が長いほど上位の階層とする．このため，下位層に配置された Value は Churn の影響を受け不安定となり低い検索成功率のままであるが，上位層に配置された Value は高い検索成功率を期待することができる．また，長く参加しているノードほど高性能な計算機であるため上位層により多くの Value を割り当てることで計算機性能を考慮したオーバレイネットワークを構築することができる．そこで，従来用いられている暗号学的ハッシュ関数に代わり，Value の Key を定めるための新しい関数を導入し，上位層のノードへ Value が集約されるように *ID* 空間を提供する．新しい関数ではデータの時間属性に着目し，データが時間経過にともなって次第に集約していく手法を用いる．時間を引数にとり，階層と *ID* を返り値とする新しいハッシュ関数を提案し，データの配置・検索を行う．時間属性は多くのデータにとって有意な情報であるため，他のオーバレイネットワークでは難しい時間情報によるデータの検索機能を提供することが可能となる．提案手法はデータが集約されて配置されているため，この検索機能も少ないクエリによって提供することができる．また，複数の属性を指定した複雑な検索であっても，初期クエリとして時間情報を利用することでデータ数を絞り込み，より高速に検索結果を得ることが期待できる．

以下に本論文の構成を述べる．2 章に関連研究について従来の構造化オーバレイネットワークについて問題点を説明する．3 章に提案手法としてオーバレイネットワークの階層化手法と提案する関数について述べる．4 章に実験による提案手法の評価を述べ，5 章に実験結果から得られた考察を述べる．最後に 6 章で本論文をまとめる．

2. 関 連 研 究

本章ではオーバレイネットワークの関連研究について説明する．構造化オーバレイネットワークの著名なものとしては Chord[13] があり，以降，多数の方式[8),10),11),15)] が提案されてきている．これらの方式はトポロジの制約方法やネットワークの構築手法に違いはあるものの，暗号学的ハッシュ関数を用いた DHT の実現を目指したものである．DHT ではコンテンツの検索に用いる Key と参加ノードの IP アドレスを，ハッシュ関数 H を用いて *ID* に変換し，同じ ID 空間に写像することでコンテンツ情報（Value）を管理するノードを定める．H には SHA-1 や MD5 などが用いられるため，ノードが多ければ Value は *ID* 空間

に対し均一に分散し，コンテンツの管理を均等に分散できる．しかし，Value を *ID* 空間に均一に分散させることはノードの特性を考慮した負荷分散にはならない．特に Churn が頻繁に起きる環境においては検索の成功率が大きく低下する．また，暗号学的ハッシュ関数は Key 間の関連性を考慮しないため，たとえば Key の範囲を指定した検索などは DHT では非常に難しい．範囲検索を DHT 上で実現するためにはコンテンツの検索に用いる Key を量子化し，その Key すべてについて検索を行う必要があるが，これはクエリのフラッディングをする必要があるため問題点が多い．

Churn への対策としては実装時の工夫[4] が主に行われており，複製の配置やキャッシュ効果を狙ったものなどが多い．しかし，根本的にデータの配置を変えるような方式は提案されていない．データの配置を改善すれば，複製の配置やキャッシュ効果と合わせて実装することで，より再現率の高いオーバレイネットワークの構築が期待できる．

Key の範囲検索を可能にするために Key の配置に連続性を持たせる手法[5),7)] が提案されている．しかし，暗号学的ハッシュ関数を用いない方式は Key の偏りによる負荷の不均衡が発生する．この不均衡はノードの特性を考慮していないため，Key の多くが不安定なノードへと割り当てられた場合，システム全体の性能を下げる原因になりうる．オーバレイネットワーク上に B-Tree を構築する BATON[6] では負荷の偏りに対して木構造の再構築によって負荷の偏りを減らしている．また，複数属性に対する範囲検索を実現した Mercury[2] では負荷の低いノードを負荷の高い *ID* に再参加させることで負荷の偏りを減らしている．しかし，これらの方式のように Key の偏りを減らすために広く分散して配置することは，範囲検索時のクエリを増やすことになり，Churn の影響によって検索の成功率低下につながる．

一方で，オーバレイネットワークを階層化する方式が提案されている．代表的なものに Cyclone[1]，Canon[3] がある．これらは文献 9) で言及されているように，Homogeneous 型と Super-peer 型に分けることができる．Homogeneous 型の Cyclone では検索のクエリ数の削減や，オーバレイネットワークの安定化に重きを置いており，ノードの特性を考慮せずに階層化する．また，Super-peer 型の Canon ではノードの特性を考慮しているものの 2 階層の階層化であるため，ノードの特性が複雑な場合に対処することが難しい．たとえば，サーバ・ヘビーユーザ・一般ユーザ・ライトユーザという構成によるオーバレイネットワークの場合，2 階層での差別化だけではヘビーユーザの計算機能力を活かすことや，逆に一般ユーザへ負荷が高くなる可能性など適用が難しい．また，Super-peer の選出によっては規模拡張性に欠けるという欠点もある．加えて，これらの方式も Key の範囲を指定した検索は困難である．

以上の関連研究をまとめると，暗号学的ハッシュ関数を用いた方式ではノードの特性を考慮しないデータ配置により，検索成功率の低下，計算機の性能を考慮しない負荷分散，複雑な検索機能を提供できないといった問題点がある．また，既存の範囲検索可能なオーバレイネットワークはノードの特性を考慮した負荷分散を行ってはいない．ノードの特性を考慮した階層化手法も 2 階層のものが主であり，複雑な検索機能の提供は行われていない．本論文ではこれらの 3 つの問題を解決する．

3. 提 案 手 法

本章では提案手法について述べる．本論文ではノードを区別する ID とコンテンツの Key である ID として 0 から 1 の ID 空間を用いる．実際には Chord などの手法と同様に 160 bit の ID 空間を用い，その場合は 0 から 1 の ID 空間を 0 から $2^{160}-1$ の ID 空間に相互変換する．ノードの ID は Chord などの手法と同様，IP アドレスなどを適切なハッシュ関数を用いて ID へ変換を行う．以下にコンテンツの Key を算出する関数と階層化オーバレイネットワークの構築法について説明する．

3.1 概　　要

提案手法では Chord と同様のリング状オーバレイネットワークを構築するが，ノードは参加時間の長さに応じてリング状の隣接ノードを変化させ，複数階層のオーバレイネットワークを構築する．参加時間の短いノードは下位層のリングに参加するが，参加時間が長くなるに従い上位層のリングにも参加する．サービス提供者の投入した計算機やヘビーユーザの利用している計算機は，参加時間が長くなるため上位層のリングへ配置される．このため，上位層では Churn による Value の喪失が起きにくいため，検索の成功率を向上させることができる．これは参加時間の長いノードの方が今後も参加している確率が高い[12)] ため，Churn の影響を受けにくいためである．

ノードは各リングに対して *ID* の管理範囲が定められ，Value はその時間属性から *ID* と階層を表す L を算出し，管理するノードを決める．このとき，古いデータほど上位層へ集約されるような *ID* と L への変換を行うため，Value が集合体として上位層で管理される．上位層と下位層では Value の割当て範囲が異なるが，上位層ほど性能の高い計算機を想定しているため問題にはならない．Value の検索クエリには検索対象の時間属性から Value と同様に *ID* と L を算出し，管理しているノードへ送信される．複雑な検索を行う場合は，まず検索する時間範囲を定め，その時間範囲を担当しているノード群に検索クエリを送信する．クエリを受け取ったノードは，指定された条件の Value を管理している Value 内から

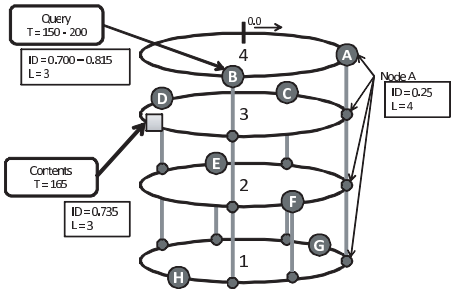

図 1　階層化オーバレイネットワーク概要図
Fig. 1　Hierarchical overlay network.

検索し，結果を送信元のノードへと返信する．Value は時間が古くなるほど集合体として管理されているため，古い時間範囲であればノード群は多くならない．

図 1 に概要図を示す．図は A から H までの 8 ノードによるオーバレイネットワークを表している．ここで T は時刻を表す．図のノード A は 4 階層目（$L = 4$）に参加していることを表しているが上位の階層に属しているノードはそれより下位の階層にも属しているため，$L = 3$，2，1 にも参加している．ID は 0 から 1 の値をそれぞれの階層で時計回りに定義しており，ノード A は $ID = 0.25$ である．図中のコンテンツは $T = 165$ から ID と L を算出し，オーバレイネットワークに参加しているノードに割り当てられる．例では $ID = 0.735$，$L = 3$ であるため，ノード B がコンテンツを保持する．検索クエリは検索対象となる時間から ID と L を算出し，適切なノードへとクエリを送信する．検索クエリには時間の範囲を指定でき，例では $T = 150$ から $T = 200$ の間のコンテンツを検索している．それぞれをハッシュ関数を用いて ID と L に変換し，$L = 3$ の $ID = 0.700$ から $ID = 0.815$ の範囲となり，管理しているのはノード B であるため，クエリはノード B へ転送される．ノード B はクエリを受け取ると検索結果のコンテンツを検索元のノードへと返信し，検索を終える．

3.2　階層化オーバレイネットワークの構築

提案するオーバレイネットワークに参加するノードは以下の情報を識別子とする．

- リング上の識別子（ID）
- 参加した時刻（T）

ノードはそれぞれルーティングテーブルを持ち，識別子とそのノードと通信するために必要な IP アドレスなどの情報を保持する．ID は Chord と同様に IP アドレスなどをハッシュ関数を用いて変換する．参加時刻は現在時刻 Now を用いて，以下のように階層 L へ変換される．

$$L = \lfloor \log_2(Now - T) \rfloor \tag{1}$$

各ノードは他のノード情報として ID と T を受け取り，式 (1) を用いて現在の階層を各自で判断するため，時間経過にともなってノードの階層が変わったとしても他のノードへ知らせる必要はない．

ノードは L 以下の階層すべてに所属する．各階層は Chord と同様のリングトポロジとなるため，successor，predicessor はそれぞれ L 個のノードとなる．successor は ID の管理範囲に密接に関係があるため，定期的にメッセージを送受信することで生存の確認を行い，管理範囲が変化していないかをつねに監視する．式 (1) のように L は参加時間の対数と定義しているため，時間経過に対して L が正比例して増えることはない．たとえば経過時間が 1 カ月の L は 21 であるが，1 年が経過しても L は 24 である．実際には $Now - T$ に対して下限の閾値を用いるため，生存の確認を行う相手ノードはもっと少なくなる．また，階層 l の successor が $l + 1$ の successor と同じ場合，l より下の階層については生存確認を行わないため，L の値が大きくなったとしてもシステム全体を複雑化することはない．ノードの参加や離脱はほぼ Chord と同様であるが，リングの多重化を防ぐために最も参加時間の長いノードをルーティングテーブルに必ず保持する必要がある．このノードはサービス提供者の用意する計算機になることを想定しているため，参加する際にサービス提供者などが用意する初期ノード（ブートストラップサーバ）から受け取ることにより対処する．

3.3　データの配置

階層化オーバレイネットワークにデータを配置するための関数を以下に定義する．ID は 0 から 1 の値をとり，Now は現在時刻，T はコンテンツの時間情報を示す．また，L の算出には式 (1) を用いるが，ルーティングテーブル内の最も古いノードの階層 L_{max} により制限され，$L \leq L_{max}$ となる．ID の算出には次の式 (2) を用いる．

$$ID = \frac{T \ \mathbf{mod}\ 2^L}{2^L} \tag{2}$$

時間経過にともなって 2^L の値は大きくなるため，T の変化に対して ID の変化は小さくなり，ID 間の距離は小さくなる．つまり，L の値が大きい上位の階層では T の変化に対する ID 間の距離が短く，T の差が大きい場合でも同じノードに配置されることが多くなる．そのため，Value は同じノードに集合体のようにして管理されることになり，検索クエリを削減できるほか，時間情報を初期クエリとした他の複雑な検索機能を提供することが可能となる．また，式 (2) は基本的に T の連続性を壊さないため，クエリをフラッディングする

ことなく T による範囲検索機能を提供することができる．

4. 評　　価

本章では提案手法の評価としてオーバレイネットワークのエミュレーションによる評価を述べる．

以下の 3 つのオーバレイネットワークを比較することで提案手法を評価する．

- 階層化したもの（階層化と式 (2) による ID）
- 階層化しないもの（式 (2) による ID）
- DHT（リングトポロジと SHA-1 ハッシュ関数による ID）

また，ノードの生存時間分布は以下のものを用いた．

- 対数を利用した分布
- Weibull 分布
- 正規分布

生存時間が t であるノードの分布（$f(t)$）を以下のように定義したものが対数を用いた分布である．

$$f(t) = -\log t \tag{3}$$

この分布では提案手法を適用した場合に各階層のノード数が等倍となり，ノードに割り当てられる Value の範囲が生存時間に正比例する．ノードへの負荷を生存時間に正比例して割り当てるのであれば，この対数を利用した分布が理想的な分布となる．また，Weibull 分布は論文 14) で指摘されている分布であり，既存のオーバレイネットワークを計測した結果から得られる分布である．ユーザの参加するサービスとしてオーバレイネットワークを構築すれば，短い間隔で参加離脱を繰り返すノードよりも，少し長めの間隔で繰り返すノードが多くなることが想定できるため，Weibull 分布に近い分布になると考えられる．最後に一般的な正規分布を用いる．これはユーザの挙動としては考えにくいが，想定とは外れた分布の場合の動作として評価する．ノードは作成されたときに生存時間が定められ，以降は生存時間ごとに参加と離脱を繰り返す．実験時間を 0 から 1 としたときのノード数の分布を**図 2** に示す．離脱したノードは同じ生存時間を用いて参加し，ノードの生存時間分布や参加ノード数は変化しない．

以上のようなオーバレイネットワークを擬似的に構築し，検索の成功率と検索クエリ数，ノードの保持している Value の数，オーバレイネットワーク上で交換されたメッセージの総数を評価した．Java を用いて提案手法の動作を行うプログラムを作成し，各ノードをスレッドによる並列動作することで提案手法のエミュレーションした．実装では Chord にある finger table は作成せず，1 度でも送受信を行ったノード情報をすべてルーティングテーブルへ追加することで検索の高速化を行った．これは，提案手法を比較する対象は Chord ではなく DHT 全般であるため，Chord のようにルーティングテーブルを制限する方式よりも，制限せずにより多くのエントリを追加する方式の方が Churn の影響を受けにくいためである．比較対象となる DHT も同様の実装を用いたため，エミュレーション条件の違いは Key の配置手法と階層化の有無だけである．

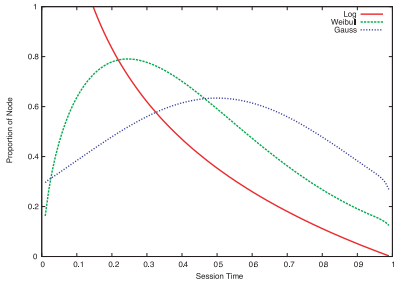

図 2　実験に用いたノードの生存時間分布
Fig. 2　Hierarchical overlay network.

ランダムな時刻を起点とした 60 秒の範囲検索を 10 秒間隔で行い，実際に共有されている Value のうち検索に該当する Value すべてを取得できた場合に成功，残りの場合を検索失敗とした．また，DHT を用いた場合は範囲検索を行うことができないが，Value を 1 秒間隔で量子化し，クエリも同様に量子化することによって代替した．つまり，60 秒の範囲検索の場合は 60 クエリを送信することになる．

4.1　実 験 結 果

評価実験では 500 ノードによるオーバレイネットワークを実験環境上で 2 時間動作させた．図 2 における最大の生存時間が 2 時間に相当する．また，計算機 10 台を用いて同じ実験を行い，それらの平均値を実験結果として評価した．以下に各分布での実験結果を示す．また，**表 1** に実験結果の平均値を示す．

4.1.1　対数を用いた分布の実験結果

図 3 に検索成功率の結果を，**図 4** にクエリ数の実験結果を示す．これらのグラフは横軸にクエリの検索対象となる時刻を用いているため，横軸が大きいほど古いデータに対する検索である．図 3 に示すとおり，提案手法を用いることで特に古い Value に対する成功率

表 1　検索結果の平均値
Table 1　Results of experiments.

	対数を用いた分布			Weibull 分布			正規分布		
実験方式	階層化	非階層化	DHT	階層化	非階層化	DHT	階層化	非階層化	DHT
成功率 (%)	45.3	10.0	4.1	44.9	28.2	9.6	31.4	18.4	15.7
クエリ数	52.3	74.9	94.7	58.3	75.5	94.3	66.0	75.4	93.3
メッセージ数	1142.9	1025.7	1038.7	1147.6	1041.2	1051.8	1109.3	997.0	1008.0

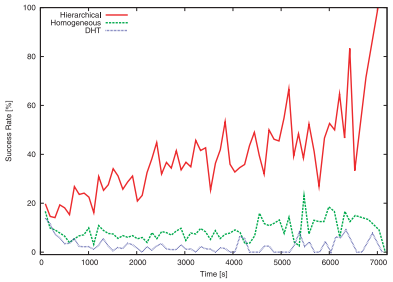

図 3　対数を用いた分布の場合の検索成功率
Fig. 3　Rate of query success: log function.

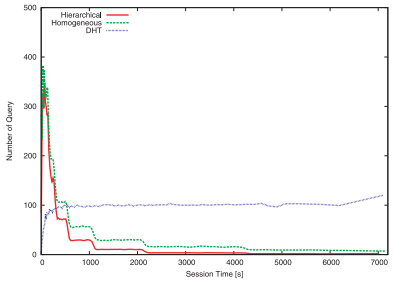

図 4　対数を用いた分布の場合のクエリ数
Fig. 4　Number of query: log function.

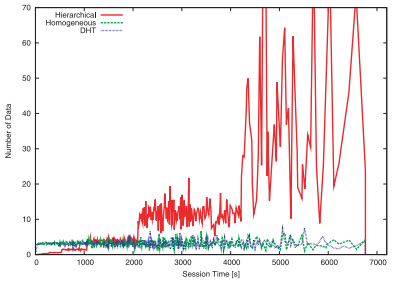

図 5　ノードの生存時間と Value 数の関係
Fig. 5　Relationship between session time of node and number of value.

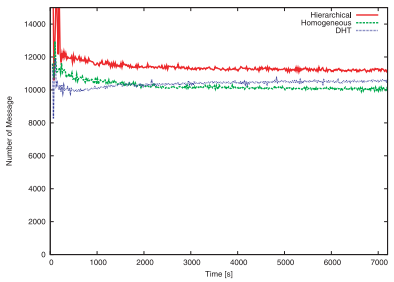

図 6　メッセージ数の時間経過による変化
Fig. 6　Number of message.

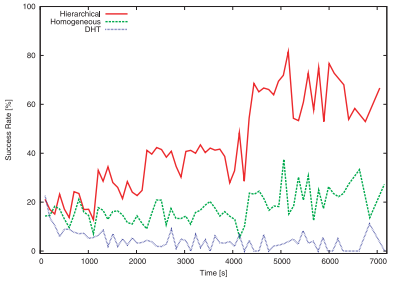

図 7　Weibull 分布の場合の検索成功率
Fig. 7　Rate of query success: Weibull distribution.

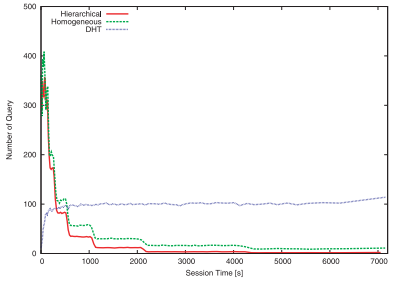

図 8　Weibull 分布の場合のクエリ数
Fig. 8　Number of query: Weibull function.

に顕著な差が出ているといえる．表 1 から，階層化したことによって約 35%向上し，DHT と比較すれば約 41%向上したことが分かる．また，図 4 と照らし合わせると，提案手法はクエリ数を減らしつつ成功率が向上したことが分かる．表 1 から，階層化することで約 22 のクエリを削減できたことが分かる．なお，DHT は量子化を行うことで範囲検索を行ったため，クエリの数は一定であるが，検索範囲を広げればクエリ数は線形に大きくなる．実験の結果では 60 秒の範囲検索によって約 94 のクエリを確認した．これは，クエリの転送などによってクエリ数が増加したためである．

図 5 にノードの生存時間と保持している Value の数を，**図 6** にメッセージ数の時間変化を示す．図 5 は横軸にノードの生存時間を用いているため，ノードは生存時間が長くなるほど Value が多くなることが確認できる．しかし，生存時間の長いノードほど高性能な計算機であることを仮定しているため，想定どおりに Value が割り当てられているといえる．また，図 5 のように Value が集約されたため，クエリ数を減らしつつ成功率を向上させることができたといえる．一方で，図 6 に示すようにメッセージ数が階層化したものだけ増加している．これは階層化の数だけ生存確認を行う必要があるためである．表 1 から増えたメッセージ数は約 100 であることが確認でき，階層化によって 11.4%，DHT と比較して 10.0%の増加となった．これに関しては 5 章で詳しく述べる．

4.1.2　Weibull 分布による実験結果

4.1.1 項と同様に，**図 7**，**図 8**，**図 9**，**図 10** にそれぞれ実験結果を示す．Weibull 分布を

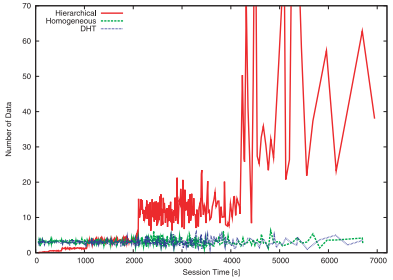

図 9　ノードの生存時間と Value 数の関係
Fig. 9　Relationship between session time of node and number of value.

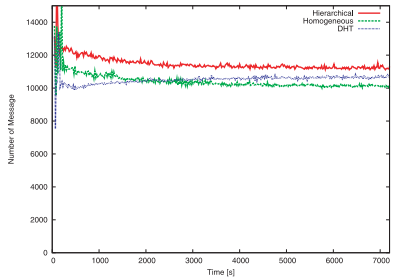

図 10　メッセージ数の時間経過による変化
Fig. 10　Number of message.

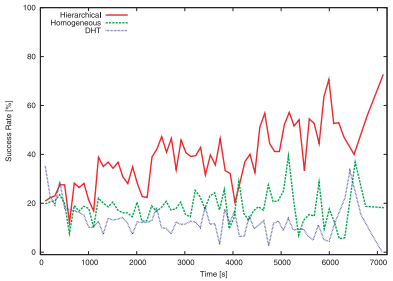

図 11　正規分布の場合の検索成功率
Fig. 11　Rate of query success: normal distribution.

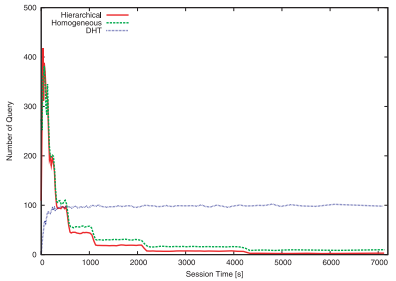

図 12　正規分布の場合のクエリ数
Fig. 12　Number of query: normal distribution.

用いた場合でも，4.1.1 項の実験とそれほど差はない．提案手法が性能を向上させていることが分かる．

4.1.3　正規分布による実験結果

最後に正規分布を用いた場合について**図 11**，**図 12**，**図 13**，**図 14** にそれぞれ実験結果を示す．正規分布を用いた場合，他の分布に比べて Churn の頻度が低いため，全体的に検索の成功率が高い．また，ノードの保持する Value の数も生存時間に対してそれほど増えてはいない．これは，上位層にノードが多く割り当てられたためと考えられる．しかし，正規分布であっても従来の手法と比較して提案手法の検索は成功率が高いことが分かる．

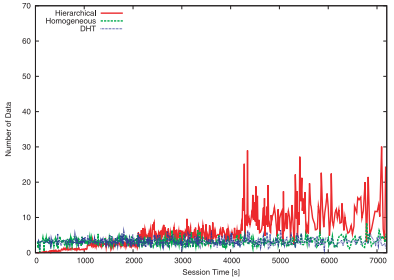

図 13　ノードの生存時間と Value 数の関係
Fig. 13　Relationship between session time of node and number of value.

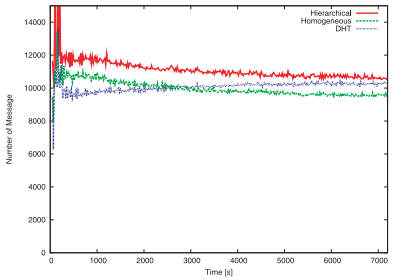

図 14　メッセージ数の時間経過による変化
Fig. 14　Number of message.

4.2　実験結果まとめ

実験により，すべての分布において検索成功率が向上していることを確認した．検索に必要なクエリも削減することができたが，オーバレイネットワーク上を流れるメッセージ数は増加した．また，生存時間の長いノードへ Value を多く割り当てることができているため，負荷を計算機の性能に応じて分散できているといえる．さらに，Value は時間の連続性を破壊することなく集合体としてノードに管理されているため，時間を初期クエリとした複雑な検索機能を提供することが可能である．

5.　考　　察

本章では実験結果に基づいて提案手法の考察を行う．まず，提案手法の利点と欠点について述べ，その後に今後の課題について述べる．

5.1　利点と欠点

実験の結果から，提案手法は従来手法に比べて少ないクエリ数で成功率の高い検索機能を提供できたといえる．また，ノードの生存時間の分布にほとんど影響されず階層化と集約による効果を実験結果から得ることができた．そのため，実際にサービスとして利用した場合に，想定とは異なる生存時間分布であったり，分布が変化するような状況であったりしても階層化と集約による検索成功率の向上を期待することができる．

一方で，メッセージ数が従来の手法に比べて約 10%増加した．これは階層化したことによって，各階層での安定化動作を行う必要が出たため，ノードの生存確認を行うメッセージ（Stabilize メッセージ）が増加したことが原因である．しかし，Stabilize メッセージは

単純なメッセージであるため，ネットワーク上を流れるトラフィックは小さく，ノードの動作もメッセージの送信元に返信するだけであり負荷も非常に小さい．ICMP を用いることを考え 1 メッセージあたり 32 バイトと想定すれば，100 メッセージの増加は 3.2 キロバイト程度となり，現状のインターネット環境ではほとんど無視することができる．このことから，安定化動作による Stabilize メッセージの増加はオーバレイネットワークおよび実ネットワークの性能にほとんど影響せず，大きな欠点ではないと考えられる．また，どの分布においても時間経過にともなってメッセージ数の差異が小さくなっていることが分かる．これは，Value を配置する際に同一宛先への Value はまとめて送信されるため，時間経過によって階層化が進めばまとめられる Value の数が多くなり，メッセージ数が削減されるためである．複数の Value を圧縮するなどすれば，実際にネットワーク上を流れるトラフィックは逆転することも考えられる．

また，提案手法は上位層へ負荷を集中させるため，上位層のノードが限界に達すればオーバレイネットワーク全体の性能に影響する．本論文ではサービス提供者が高性能計算機を投入することを前提としているため，上位層に配置されたノードの限界は従来のサーバの限界として扱うことができ，1 サーバあたり 1 万台程度のクライアント接続によって限界に達することが想定できる．こうした場合はサービス提供者が新たな計算機を投入することで対処できる．従来の DHT であれば負荷の集中する Key があったとしても，その Key を担当するノードの ID を得ることや，事前にそれを知ることも困難であるため，計算機の投入によって負荷を軽減することが難しい．このことから，上位層へ負荷を集中させることは提案手法の利点と考えることもできる．

5.2　今後の課題

提案手法によって検索結果の成功率を向上させることはできたが，インターネットサービスの主流であるサーバ・クライアントな環境ではほぼ 100%に近い成功率が実現していることを考えれば，まだ安定したサービスを提供できる基盤に至っているとはいえない．しかし，提案手法は Value の配置に改良を加えた手法であるため，他の Churn 対策手法をそのまま応用することが可能である．例として Value の複製を別のノードへ置く手法がある．これは Value を担当しているノードの離脱時に次の担当となるノードへあらかじめ Value を配置しておく手法である．多くの複製を配置することはトラフィックの増大と Value 管理の難しさが問題となるが，提案手法は上位の階層ほどノードが少なく，また Churn の起きる確率も少ないため，より少ない数の複製を置くだけで再現率の大幅な向上を期待できる．こうした手法を確立し，その効果を評価することは現実的な運用をするうえでの今後の課題となる．

また，想定と異なり一般ユーザの生存時間と計算機の性能が一致しない場合が考えられる．これはノード情報に階層の上限値と下限値を設けることによって対処できる．たとえば性能の低い計算機が長くオーバレイネットワークに参加する場合，上限値を設定しておくことによって Value の割当てを制限することができる．また，逆にサーバのような高性能の計算機は，下限値を設けることで初期の段階から上位層へ参加し，Value の割当てを増やすことができる．なお，この上限値と下限値は参加者の意図を反映させないために，自動的に設定されるような仕組みやサービス提供者のみが設定できるような枠組みが必要である．

一方で提案手法は実際のネットワークの特性を考慮せずに構築するため，遅延や帯域の大きいノードが上位層に配置される可能性がある．現実的なネットワーク特性を考慮した場合，遅延や帯域の制限が提案手法の利点である検索成功率に影響することはほとんどない．しかし，100%に近い成功率を実現するためにメッセージ数の間隔を短くするなどの改善を行うのであれば，それらのネットワーク特性を考慮する必要がある．こうした改善を行ううえで，提案手法は他の手法に比べて負荷の集中しているノードを把握しやすいため，より簡潔な変更で大きな改善を行えることが期待できる．

6. ま と め

オーバレイネットワークの分野では構造化オーバレイネットワークに関する研究がさかんであるが，ユーザが自由に参加できる Churn の多い環境では検索の成功率が低下するという問題があった．また，構造が単純であるため複雑な検索が難しく，範囲検索などを実現するためには負荷の偏りが問題であった．本論文では参加時間の長いノードほど性能の高い計算機であると仮定し，オーバレイネットワークを階層化することでそれらのノードへ Value を偏らせ，検索成功率が高く複雑な検索機能を提供できるオーバレイネットワークを提案した．実験の結果，理想的な対数を用いた分布では DHT と比べて検索成功率が約 41%の向上，現実的な Weibull 分布を用いた場合でも DHT と比べて約 35%の向上が見られた．また，階層化によるオーバヘッドも約 100 メッセージであり，実ネットワークでの性能にはほとんど影響しないことも確認した．今後は他の手法と併用することで検索の成功率が 100%に近づくよう改良を加えるほか，ノードの他の属性も考慮した階層化に取り組むことで，ユーザの参加可能なオーバレイネットワークの構築を目指す．

謝辞　本研究の一部は，総務省委託研究「ユビキタスサービスプラットフォーム技術の研究開発」による成果である．ここに記して謝意を表す．

参 考 文 献

1) Artigas, M.S., Lopez, P.G., Ahullo, J.P. and Skarmeta, A.F.G.: Cyclone: A novel design schema for hierarchical dhts, *5th IEEE International Conference on Peer-to-Peer Computing, 2005, P2P 2005*, pp.49–56 (2005).
2) Bharambe, A.R., Agrawal, M. and Seshan, S.: Mercury: Supporting scalable multi-attribute range queries, *SIGCOMM '04: Proc. 2004 Conference on Applications, Technologies, Architectures and Protocols for Computer Communications*, pp.353–366, ACM Press, New York, NY, USA (2004).
3) Ganesan, P., Gummadi, K. and Garcia-Molina, H.: Canon in g major: Designing dhts with hierarchical structure, *Proc. 24th International Conference on Distributed Computing Systems, 2004*, pp.263–272 (2004).
4) Godfrey, P.B., Shenker, S. and Stoica, I.: Minimizing churn in distributed systems, *Proc. 2006 Conference on Applications, Technologies, Architectures and Protocols for Computer Communications*, pp.147–158, ACM New York, NY, USA (2006).
5) 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*, pp.113–126 (2003).
6) Jagadish, H.V., Ooi, B.C. and Vu, Q.H.: BATON: A balanced tree structure for peer-to-peer networks, *Proc. 31st International Conference on Very Large Data Bases*, pp.661–672, VLDB Endowment (2005).
7) Matsuura, S., Fujikawa, K. and Sunahara, H.: Mill: An information management and retrieval method considering geographical location on ubiquitous environment, *Applications and the Internet Workshops, 2006. SAINT Workshops 2006 International Symposium on Applications and the Internet*, p.4 (2006).
8) Maymounkov, P. and Mazieres, D.: Kademlia: A peer-to-peer information system based on the xor metric, *Peer-to-Peer Systems: 1st InternationalWorkshop, IPTPS*, pp.53–65, Springer Berlin/Heidelberg (2002).
9) Pappas, V., Massey, D., Terzis, A. and Zhang, L.: A comparative study of the dns design with dht-based alternatives, *Proc. INFOCOM 2006* (2006).
10) Ratnasamy, S., Francis, P., Handley, M., Karp, R. and Schenker, S.: A scalable content-addressable network, *SIGCOMM '01: Proc. 2001 Conference on Applications, Technologies, Architectures and Protocols for Computer Communications*, pp.161–172, ACM Press, New York, NY, USA (2001).
11) Rowstron, A. and Druschel, P.: Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems, *Middleware 2001*, p.329, Springer Berlin/Heidelberg (2001).
12) Saroiu, S., Gummadi, P.K., Gribble, S.D., et al.: A measurement study of peer-to-peer file sharing systems, *Proc. Multimedia Computing and Networking*, Vol.2002, p.152, Citeseer (2002).
13) Stoica, I., Morris, R., Liben-Nowell, D., Karger, D.R., Kaashoek, M.F., Dabek, F. and Balakrishnan, H.: Chord: A scalable peer-to-peer lookup protocol for internet applications, *IEEE/ACM Trans. Netw.*, Vol.11, No.1, pp.17–32 (2003).
14) Stutzbach, D. and Rejaie, R.: Understanding churn in peer-to-peer networks, *Proc. 6th ACM SIGCOMM Conference on Internet Measurement*, pp.189–202, ACM New York, NY, USA (2006).
15) Zhao, B.Y., Kubiatowicz, J.D. and Joseph, A.D.: Tapestry: An infrastructure for fault-tolerant wide-area location and routing, Technical Report, Berkeley, CA, USA (2001).

(平成 21 年 7 月 7 日受付)
(平成 21 年 12 月 17 日採録)

洞井　晋一（学生会員）

平成 17 年神戸大学海事科学科卒業．平成 19 奈良先端科学技術大学院大学情報科学研究科修士課程修了．同年より同大学情報科学研究科博士課程．オーバレイネットワーク，分散処理システムの研究に従事．

松浦　知史

平成 15 年立命館大学理工学部物理学科卒業．平成 17 年奈良先端科学技術大学院大学情報科学研究科修士課程修了．同年より同大学情報科学研究科博士課程．オーバレイネットワーク，センサネットワークの研究に従事．

藤川　和利（正会員）

昭和 63 年大阪大学基礎工学部情報工学科卒業．平成 3 年同大学大学院基礎工学研究科博士後期課程退学後，同年大阪大学基礎工学部助手等を経て，平成 14 奈良先端科学技術大学院大学情報科学センター助教授，平成 17 年同大学情報科学研究科助教授，現在に至る．博士（工学）．分散処理システム，マルチメディアシステムの研究開発に従事．電子情報通信学会，IEEE，ACM 各会員．

砂原　秀樹（正会員）

昭和 58 年慶應義塾大学工学部電気工学科卒業．昭和 63 年同大学大学院博士課程修了．同年電気通信大学情報学部助手．平成 6 年奈良先端科学技術大学院大学情報科学センター助教授．平成 13 年同大学情報科学センター教授．平成 17 年同大学情報科学研究科教授，現在に至る．工学博士．インターネット，大規模広域分散環境，ネットワーク，並列処理，オペレーティングシステム，電子図書館に関する研究に従事．電子情報通信学会，ACM，IEEE 各会員．