一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
SITE2016-83,IA2016-113 (2017-03)

構造化オーバーレイにおける関係性に基づくデータ複製による マルチクエリに対する応答の高速化

小泉　悠介†　渡部　康平††　中川　健治††

† †† 長岡技術科学大学 大学院工学研究科　〒 940–2137 新潟県長岡市上富岡町 1603–1
E-mail: †s133116@stn.nagaokaut.ac.jp, ††{k_watabe,nakagawa}@vos.nagaokaut.ac.jp

あらまし　本研究では，構造化オーバーレイネットワークにおいて，データ間の関係性をデータ配置に反映することにより，探索に要するホップ数を低減して，応答を高速化する方法を提案する．データ間の関係性を反映した最適なデータ組合せを算出する問題を整数計画法により定式化し，ヒューリスティック解法を用いて解く方法を示す．提案手法の評価を応答時間の高速性，計算時間，負荷分散の観点から行った結果，高速性に関しては提案法を適用しない場合に比べて適用した場合の方が最大で約 50%ホップ数の低減を実現することができた．また，計算時間に関しては，応答時間の高速性を維持したまま，定式化した式をヒューリスティック解法を用いずに解いた場合に比べて，最大で約 10 倍高速にデータの組合せを算出できる結果となった．さらに，負荷分散に関しては，提案法を適用しても負荷の偏りはほとんど変化しないことを確認した．

キーワード　構造化オーバーレイネットワーク，マルチクエリ，p2p，ヒューリスティック解法

Speeding Up of Response to a Multi-query Using Data Replication Based on Relationship in Structured Overlay Network

Yusuke KOIZUMI†, Kohei WATABE††, and Kenji NAKAGAWA††

† ††Graduate School of Engineering, Nagaoka University of Technology,
Kamitomioka-cho 1603–1, Nagaoka, Niigata, 940–2137 Japan
E-mail: †s133116@stn.nagaokaut.ac.jp, ††{k_watabe,nakagawa}@vos.nagaokaut.ac.jp

Abstract　In this study, we propose a method speeding up of response to a multi-query in structured overlay networks. The proposed method replicates data based on relationship in the data stored in a structured overlay network. We formulate an optimization problem of data selection to replicate as an integer programming problem, and proposed a heuristic solution of it. Our evaluation shows that the proposed method can reduce the number of hops to about 50%. The heuristic solution can choose replication data about 10 times faster than solving the integer programming problem. Furthermore, we confirmed that the load distribution is almost the same when the proposed method applies.

Key words　Structured Overlay Network, Multi-query, p2p, Heuristics

1 研 究 背 景

近年，ビッグデータの利用が注目を集め，センサーなど様々なデバイスから膨大なデータを収集し，収集したデータを利用する様々なサービスが想定されている．安価なセンサーの開発など，ビッグデータを利用するサービスを支える技術基盤はいくつか挙げられるが，収集した膨大なデータを保存・管理するためのストレージシステムもキーテクノロジーの一つである．

データ量の増加に対してスケーラブルであり，処理コストも小さいため，構造化オーバーレイネットワークを利用したストレージシステムは，ビッグデータ管理におけるスケーラビリティの問題を解決することができると期待されている [1]〜[7]．構造化オーバーレイネットワークを利用したストレージシステムでは，物理ネットワーク上に点在する多数のストレージにより仮想的なネットワークを構成し，データを分散して保存する．各ストレージに保存されているデータをすべて把握している集中管理サーバは存在せず，データは分散管理されているため，集中管理サーバにデータが保存されているストレージを問

い合わせて発見することはできない．ユーザからデータ取得のリクエストがあると，リクエストがあったデータが保存されているストレージにたどり着くまで，各ストレージは隣接するストレージにリクエストを受け渡ししながら検索を行う．

構造化オーバーレイネットワークを利用したストレージシステムにおける最も重要な品質指標の一つは，クエリに対する応答時間である．しかし，多数のデータを同時に要求するマルチクエリでは，クエリに対する応答時間が長くなってしまう可能性がある．構造化オーバーレイネットワークでは，多数のストレージに分散してデータを保存するため，マルチクエリでは多数のストレージを探索し，データを見つける必要がある．

構造化オーバーレイネットワークにおいてデータ間の関係性をデータ配置へ反映することは，クエリに対する応答を高速化する上で重要である．一般的に，保存されるデータには，マルチクエリにより同時に取得されやすい組み合わせや関係性が存在する．関係性の深いデータ同士は，同じストレージに配置されることにより，クエリに対する応答時間が高速化することが期待される．Chord# [7] や Mercury [1] など，一部のプロトコルでは，隣り合う ID を持つデータを同じストレージに配置している．レンジクエリに対応するためだが，隣り合う ID を持つデータは同時に取られやすいという関係性に基づき，データを配置していると捉えることができる．しかし，各データ間の関係性は複雑であり，単純な ID の構造によって表現できるとは限らない．さらに，時間とともに関係性が変化する可能性もあるため，これらのプロトコルでデータ間の関係性のすべてを反映することは難しい．

本研究では，構造化オーバーレイネットワークにおいて，データ間の関係性をデータ配置に反映することにより，マルチクエリ処理に対する応答を高速化する．提案手法では，ID 構造には反映されない各データ間の関係性を考慮してデータを他のストレージに複製する．複製するデータを最適化する問題を整数計画法として定式化し，ヒューリスティック解法により解くことで，限られたストレージ領域で最大限の高速化を実現する．各ストレージはローカルな情報のみで自律分散的に制御を行い，最適なデータ配置を達成する．加えて，提案法ではオリジナルデータの配置を変更せず，複製を行うため，Chord# [7] を始めとする多くのプロトコルと併用が可能であり，探索やデータ配置のプロトコルに依存しない．

本稿の構成は以下のとおりである．第 2 章では，本研究との関連研究の概要を述べ，第 3 章において，本研究における評価で用いる Chord# [7] の説明を行う．第 4 章では，提案法の具体的手順について説明を行い，第 5 章では，提案法の特性を評価する．最後に，第 6 章では，本研究のまとめと今後の展望について述べる．

2 関 連 研 究

構造化オーバーレイネットワークの研究では，多種多様なプロトコルが提案されている．構造化オーバーレイネットワークの研究においては，クエリのホップ数特性と負荷分散特性に着目した研究が数多く報告されてきた．各研究ごとに異なる形の仮想的なネットワークを構成し，各ネットワークの形に基づいた負荷分散が行われている．I.Stoica らによる Chord [4] では，仮想的なネットワークは，各ストレージが円状に並んだ 1 次元的な空間である．また，負荷分散にはハッシュ関数を用いている．Chord は仮想的なネットワークの構造が単純であるため，多くの研究へ参照されている．T.Schütt らによる Chord# [7] は仮想的なネットワークの構造が Chord と同じ，円状に並んだ 1 次元的な空間である．一方で各ストレージの ID には保存されているデータに含まれる情報そのものを用いている．負荷分散には D.Karger らの研究 [8] によるアルゴリズムが適用されている．J.Aspnesa らによる SkipGraphs [5] では，仮想的なネットワークは円状に並んだ 1 次元的な空間を複数有している．各空間に含まれるストレージ数が異なり，探索の際には，ホップ数がより少なくなるように空間を選択する．

構造化オーバーレイネットワーク以外のオーバーレイネットワークに関しては，データの関係性に着目した研究も存在する．Z.Zou らの研究 [9] や A.Crespo らの研究 [10] はセマンティックオーバーレイネットワークの研究である．セマンティックとは，データに含まれる情報の内容を理解させ，データ資源の探索やデータの転送に応用する技術のことである．セマンティックではメタデータというデータに含まれる情報をまとめたデータに基づく．メタデータには作成日時や作成者，データに含まれる情報のキーワードなどが含まれている．メタデータを参照し，データ配置の変更やネットワークの生成を行うことでサービス品質の向上を図っている．メタデータにどのような情報を記述するかはどのようなデータモデルを想定しているかによる．また，一部の構造化オーバーレイネットワークのプロトコルでは，データ間の関係性をネットワークの作成基準に用いている．A.Bharambe らの Mercury [1] では，データに存在する日時や作成者，データ内の数値と行った情報を属性として取り扱っている．ネットワーク作成時には属性値ごとの小規模なネットワークであるハブを作成する．ハブとハブとをリンクでつなげることにより，複数の属性を含むクエリにおいても 1 次元的な探索の繰り返しでデータの取得を実現している．

3 構造化オーバーレイネットワーク

構造化オーバーレイネットワークでは，物理ネットワーク上に配置されたストレージ間を特定の構造を持った仮想的なリンクで結ぶことでネットワークを構築し，データを分散して保存する．全体の情報を集中管理的に扱うことなく，各ノードが自律分散的にリンク情報や保存データを管理することで，大規模データに対してもスケーラブルに動作することができる．

代表的なプロトコルである Chord では，図 1 のようにデータの ID 空間が 1 次元のリング状に構成される．リング状に配置されたストレージが互いにリンクを貼ることによりネットワークを構成する．リング上のストレージの位置，及びデータの保存場所は，ハッシュ関数に基づき均一に分散される．各ストレージは，リング上で隣り合う，あるいは近傍のストレージとのリンク (Successor List) と遠く離れたストレージとのリンク (Finger Table) を管理する．クエリが発生すると，クエリは

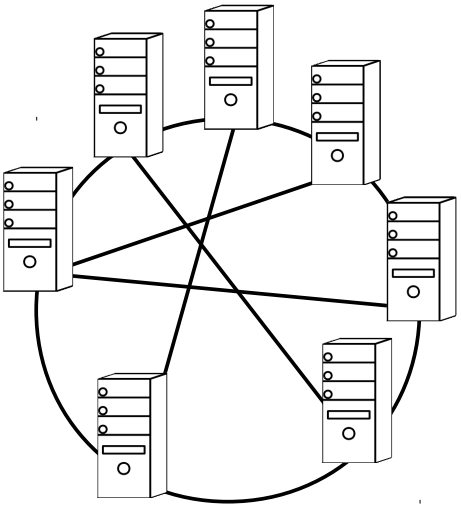

図 1　Chord の ID 空間

リンクを辿ってストレージを探索しながらデータが保存されているストレージまで届けられる．

Chord では，ハッシュ関数によりデータが分散されるため，レンジクエリのような範囲を指定したクエリを処理することはできない．しかし，データの ID 空間をハッシュ関数により崩すことなく構成することでレンジクエリに対応したプロトコルも複数提案されている．Chord をレンジクエリに対応させた Chord#では，ハッシュ関数を用いない負荷分散 [8] により，ID が近い値を持つデータ間の関係性を崩すことなく保管することに成功している．ID が近い値を持つデータ同士は同じストレージ，あるいはリング状で近い配置にあるストレージに保管されるため，関係性が深いデータを同時に取得することができる．ただし，データ間の関係性は ID 空間により表現されている必要があり，リング状でなければならないことに注意する．

4　提　案　法

4.1　データ間の関係性

ネットワーク上で保存されているデータには，同時に取得されやすいデータの組合せや関係性が存在する．データ間の関係性の概念図を図 2 に示す．図 2 では，例として東京，千葉，新潟における 8:00，12:00，15:00 の気象情報がそれぞれ存在する場合を考える．各データは単体で見れば各県各時刻における気象情報のデータとなるが，各長方形で囲まれたデータに着目する．東京のみを囲った長方形におけるデータでは，東京における 1 日の天気に関するデータと捉えることができる．同様に，東京と千葉を囲った長方形におけるデータでは関東における午前中の天気，東京，千葉，新潟すべてを囲った長方形におけるデータでは全県における午前 8 時の天気というデータと捉えることができる．この例では，時刻が近いデータや地理的に近いデータ同士は互いに深い関係にあり，データを要求するユーザはマルチクエリにより，関係性の深いデータを同時に取得しやすい．関係性の深いデータ同士を同じストレージへ配置することができれば，探索するストレージの数を削減することができるため，探索に要する時間が短くなり，応答を高速化することができる．

図 2　データ間の関係性の例

図 2 における例では，地域と時刻による関係性を例として用いたが，データ間の関係性は一般にユーザの利用傾向を反映して複雑であると予想され，既存の技術ではこれらの関係性を適切に反映してデータを配置することは難しい．Chord# [7] などレンジクエリに対応したプロトコルでは，このような関係を ID 空間によって表現し，同時に取得されやすいデータを近いストレージに配置するようにしている．しかし，データ間の関係性は 1 次元など単純な ID 空間にマッピングできるとは限らない．加えて，ユーザの傾向変化を反映して時間とともに変化していく可能性もあるため，事前に予測して ID 空間を設計することは困難である．

4.2　データの複製による関係性の反映

本研究では，データを複製して他のストレージにも保管することにより，Chord# [7] などが提供する ID 空間の設計では考慮することが難しいデータ間の関係性を反映したデータ配置を実現する．複製を用いることにより，構造化オーバーレイネットワークのルーティングプロトコルに影響を与えず，データへの到達性を損なうことなく，データ間の関係性をデータ配置へ反映できる．ネットワークに保存されているデータを複製せず移動させると，本来存在するはずのストレージにデータが存在しなくなる．そのため，プロトコルに基づくデータ探索を行った際に，データへの到達性を維持できなくなる．複製を用いれば，オリジナルのデータの存在するストレージは変わらないため，データへの到達性を維持したまま，データ間の関係性をデータ配置へ反映させることができる．

提案法では，各ストレージで通過したクエリをログとして記録し，ログの件数が一定に達した時点でデータの複製を行う．データの複製を行う際は，データ間の関係性が反映されるデータの組合せを算出して複製する．データ間の関係性が反映されたデータの組合せを複製，保存することによって，データ間の関係性をデータ配置へ反映する．複製するデータの組合せ算出は，クエリのログを用いて各ストレージごとに行うため，他のストレージから情報に頼ることなく，各ストレージが自律分散的に複製するデータの組合せを算出する．

4.3　組合せ算出問題の整数計画問題による定式化

本研究では，最適なデータ組み合わせを算出する問題を，1

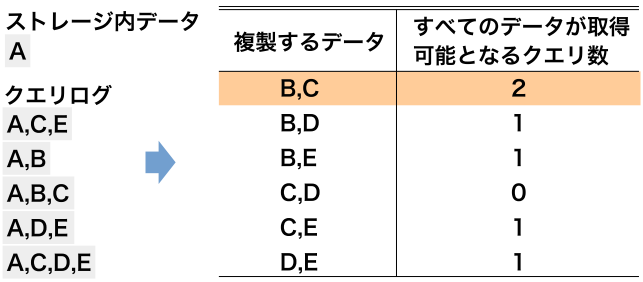

図 3　すべてのデータが取得可能となるクエリ数を最大化するデータ組み合わせ

個のストレージで全てのデータが取得可能となるクエリ数を最大化する問題として捉える．概要図を図 3 に示す．A〜E の 5 つのデータが保管されている構造化オーバーレイネットワーク上で，あるストレージが図中矢印の左側のようなクエリを処理したとする．ストレージにはデータ A が既に保存されているとして，複製したデータを保存する領域のサイズを 2 個とした場合，複製するデータの組合せと，それらのデータを追加保存領域に保存することで取得可能となるクエリの数を，図中矢印の右側に示している．図 3 の場合では，データ B とデータ C を複製するデータとして選択・保存することで，取得可能となるクエリの数が最大化される．クエリログの重複を許して複製するデータを選択すれば，データの関係性だけでなくクエリの発生頻度に偏りがあった場合も偏りを考慮したデータ配置が実現する．

前述の考えに基づき，複製するデータの組合せ算出問題を整数計画問題として定式化した．定式化した問題と各変数を (1) に示す．

$$\begin{cases} \text{maximize} & \displaystyle\sum_{i=1}^{n} x_i & & \\ \text{subject to} & \displaystyle\sum_{j=1}^{m} y_j & \leqq & c, \\ & y_j & \geqq & d_{ij}x_i, \quad \forall i,j \end{cases} \tag{1}$$

$$\begin{pmatrix} c & : & \text{他ストレージから複製したデータの保存領域のサイズ} \\ A & : & \text{利用するクエリの集合} \\ B & : & A \text{ に含まれているデータの集合} \\ n & : & A \text{ の要素数} \\ m & : & B \text{ の要素数} \\ a_i \in A & : & A \text{ の } i \text{ 番目のクエリ } (1 \leqq i \leqq n) \\ b_j \in B & : & B \text{ の } j \text{ 番目のデータ } (1 \leqq j \leqq m) \\ d_{ij} \in \{0,1\} & : & a_i \text{に } b_j \text{が含まれている場合は 1，それ以外は 0 となる指示関数} \\ x_i \in \{0,1\} & : & a_i \text{に含まれるデータを保存する場合は 1，それ以外は 0 となる指示関数} \\ y_j \in \{0,1\} & : & b_j \text{を保存する場合は 1，それ以外は 0 となる指示関数} \end{pmatrix}$$

(1) を整数計画問題として解くことによって，最適なデータの組合せを算出することが可能である．目的関数は，1 個のストレージで取得できるクエリの数を最大化するということを意味している．説明変数は x_i および y_j であり，目的を満たす x_i および y_j の解を求める．制約条件では，ハードウェア上の制約およびデータの選択をする上で矛盾が生じないような制約を定めている．一つ目の制約式は保存するデータ数は複製したデータを保存する追加保存領域を超えないということを意味している．また，二つ目の制約式はデータ b_j が含まれるクエリ a_i を可能とするならば，データ y_j を保存しなければならないということを意味している．

4.4　組合せ算出問題のヒューリスティック解法

整数計画問題は一般に NP 困難であることが知られており，本研究では (1) をヒューリスティック解法により解くことで，高速にデータの組合せを算出する手法を提案する．ヒューリスティック解法では，各クエリに対する評価値を算出し，評価値を最大化するよう貪欲法によりデータの組合せ算出を行う．以下にヒューリスティック解法のアルゴリズムを示す．

（1） 追加保存領域内のデータをクリアする．
（2） i 番目のログのクエリに含まれるデータ集合を Q_i，ストレージ内に保存されているデータ集合を S として，$Q_i - S$ を算出する．
（3） $Q_i - S = Q_j - S$ となる j の個数を数え，N_i とする．
（4） 評価値 $N_i/|Q_i - S|$ が最大となる i を求め，対応するデータをすべて追加保存領域に複製する．
（5） 追加保存領域に空き領域があれば (2) に戻る．なければ終了する．

上記のアルゴリズムを一定量クエリが蓄積される度に実行し，追加保存領域に保存するべきデータを決定する．

5　評　　価

5.1　シミュレーション概要

提案法の特性評価を行うため，Chord#をベースにした構造化オーバーレイネットワークのシミュレーションを行った．複数のストレージからなる構造化オーバーレイネットワークを構成し，10000 個のマルチクエリを生成した．複製データを保存する保存領域のサイズは 1 ストレージあたりデータ 30 個とした．シミュレーションのネットワークについて，提案法を利用した場合とそうでない場合，LRU (Least Recently Used) により組み合わせを算出した場合のそれぞれについて，応答の高速化，計算時間，負荷の分布について各パラメータを変化させた際の特性を測定し，比較した．提案手法では，追加的に保存領域を確保し，データを複製するため，提案手法を用いない場合に比べ，より多くの保存領域を持つことになる．最近到達したクエリに含まれるデータを優先的に保存する LRU 方式で追加保存領域内のデータを決定する方法とも比較し，提案手法の有効性を検証する．追加保存領域内のデータの入れ替えは，1000 クエリ毎に実施し，定常状態に達した状態で性能を比較する．以下に実施するシミュレーションでは，特に言及がない限り表 1 に示すパラメータを利用した．

表 1　シミュレーションパラメータ

ストレージ数	100 個
保存データ数	10000 個
プロトコル	Chord#
クエリ数	10000 回
保存領域のサイズ	30 個
データを更新する間隔	1000 クエリ

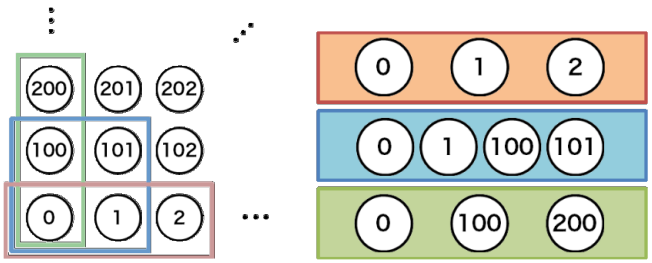

図 4　データ間の関係性を表現するクエリモデル

データ間の関係性を表現するクエリモデルについては，2 次元トーラス上で表現できると仮定した．クエリモデルの概要図を図 4 に示す．構造化オーバーレイネットワーク上で保存されるデータに 1 次元の ID 空間の ID が割り振られているとする．構造化オーバーレイネットワーク上で保存されているデータを図 4 のように全体が四角形となるように 2 次元トーラス上にマッピングする．2 次元トーラス上で近いデータ同士は関係性が深いと仮定し，図 4 における長方形内のデータがマルチクエリとして要求されるとした．

マルチクエリとして要求するデータを決定する長方形は縦および横の長さ，長方形の左下の座標を乱数で決定する．長方形の左下の座標は，ID 0 の点を原点とする Zipf 分布に従うとし，長方形の縦横の長さもそれぞれ Zipf 分布に従うとした．各 Zipf 分布のパラメータは，1.4 とした．

5.2　提案手法による応答の高速化効果

提案手法による応答の高速化効果を検証するため，構造化オーバーレイネットワークを構成するストレージ数を 10 ストレージから 100 ストレージまで変化させた際の平均所要ホップ数の推移を測定した．平均所要ホップ数は，マルチクエリに含まれるデータを全て取得するまでに要するホップ数の平均として定義した．Chord#のネットワークにおいて，提案法を用いない場合と用いた場合とで平均所要ホップ数を比較した．また，提案法において，複製するデータの組合せを算出する方法を LRU，整数計画法による解法，ヒューリスティック解法を用いた場合でそれぞれ平均所要ホップ数を測定，比較した．平均所要ホップ数の推移を図 5 に示す．横軸は，構造化オーバーレイネットワークを構成するストレージ数，縦軸は平均所要ホップ数を表す．図 5 より，どのストレージ数においても提案法を適用しない場合に比べて提案法を適用した場合の方がホップ数が低減できている事がわかる．また，データの組合せ算出に LRU を用いた場合に比べて，最大化問題を解いた場合および評価値法を用いた場合の方がホップ数が低減できていることがわかる．提案法を利用しない場合に比べて，提案法を利用することにより，最大で約 50%のホップ数を低減している．また，ヒューリスティック解法による解が最適化問題を解いた場合と同等の性能を示すことも確認できる．

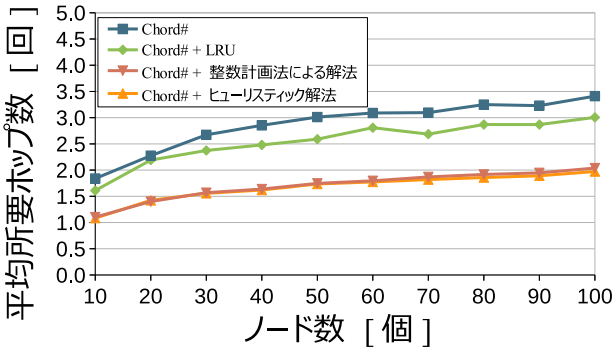

図 5　提案法の利用による平均所要ホップ数の削減効果

5.3　提案手法が負荷の偏りに与える影響

提案法では，データをオリジナルが保存されているストレージとは異なるストレージに複製するため，ストレージ間の負荷が変化する可能性がある．データの複製により，クエリに含まれるデータを転送するのはプロトコルによって定められたデータの担当ストレージとは限らないため，データの担当ストレージを探索している途中の過程で転送する場合もある．構造化オーバーレイネットワークでは，ストレージ間の負荷分散が重要であるが，提案法の導入により各ストレージの負荷が変化し，負荷がより偏ってしまう恐れがある．提案法の導入が各ストレージの負荷の分布に与える影響を検証するために，提案法を利用した場合とそうでない場合の各ストレージの負荷の分布をシミュレーションにより導出し，比較した．

提案法を使用しない場合と，ヒューリスティック解法の提案法を導入した場合の各ストレージの負荷の累積分布を図 6 に示す．図 6 において，累計ストレージ数は，負荷の累計を行ったストレージの総数のことを言う．また累計負荷は，負荷を降順に累計ストレージ数だけ累計した負荷のことを言う．ここで，各ストレージの負荷は，各ストレージがクエリを送信したストレージに対してデータを送信した回数と定義した．今回ベースのプロトコルとして使用している Chord#では，各ストレージが保管するデータ数が均等に分散するよう負荷分散をするため，各ストレージの保存データ数の分布はほぼ均一になっていることに注意する．図 6 より，提案法を適用しない場合に比べて提案法を適用した場合の方がわずかに負荷が集中するが，負荷の分布はほぼ同程度で，提案手法の導入による負荷分散に与える影響は小さいことが確認できる．

5.4　ヒューリスティック解法による計算時間の削減効果

ヒューリスティック解法による計算時間の削減効果を検証するため，保存データ数が 10000 個と 1000000 個の場合について，最適化問題を整数計画問題として解いた場合とヒューリスティック解法を利用して解いた場合の計算時間を比較した．両

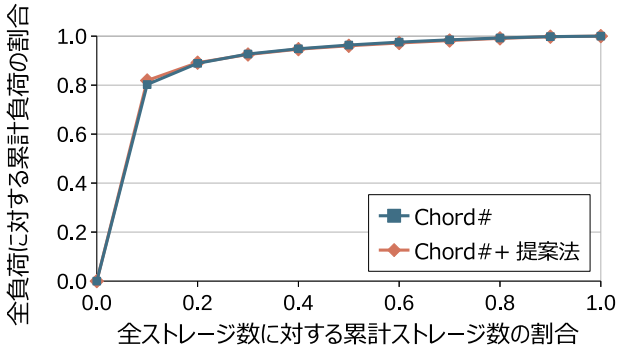

図 6　提案手法の導入が負荷分散に与える影響

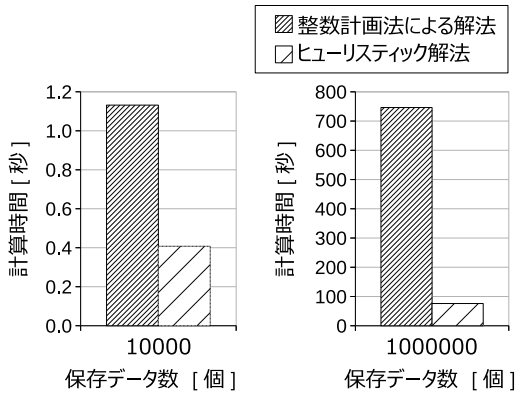

図 7　ヒューリスティック解法による計算時間の削減効果

者の比較を図 7 に示す．図 7 より，どちらのデータ数においても整数計画法を解いた場合に比べてヒューリスティック解法を用いてデータの組合せを算出した場合の方が計算時間が短いことがわかった．測定結果から計算した結果，整数計画法を解いた場合に比べて，ヒューリスティック解法を用いた場合の方が約 3 倍から 10 倍高速であることが確認できた．

6 ま と め

本研究では，構造化オーバーレイネットワークにおいて，データ間の関係性をデータ配置に反映することにより，探索に要するホップ数を低減して，応答を高速化する方法を提案した．データ間の関係性を反映した最適なデータ組合せを算出する問題を整数計画法により定式化し，ヒューリスティック解法を用いて解くことにより，最適な組み合わせと同等の応答時間をより短い計算時間で達成した．

提案手法の評価を高速性，計算時間，負荷分散の観点から行った．その結果，高速性に関しては提案法を適用しない場合に比べて適用した場合の方が最大で約 50%ホップ数の低減を実現することができた．また，計算時間に関しては，定式化した式をヒューリスティック解法を用いずに解いた場合に比べて，最大で約 10 倍高速にデータの組合せを算出できる結果となった．さらに，負荷分散に関しては，提案法を適用しても負荷の偏りはほとんど変化しないことを確認した．

本論文における評価では，クエリに含まれるデータの選択を大きく偏らせた．この偏らせるパラメータは実ネットワークのデータ等に基づくパラメータではない．より現実的なネットワークでの評価を行う上では実ネットワークの結果に基づいたクエリの偏りを実現できるパラメータの選択が必要である．また，高速性の評価におけるストレージ数は最大で 100 までとしているが，構造化オーバーレイネットワークの研究における最大ストレージ数は $10^5 - 10^7$ まで行われることが多い．今後は，ストレージ数を更に増加させた場合に，提案法を適用することでホップ数をどれだけ低減できるか評価を行う予定である．

文　献

[1] A.Bharambe, M.Agrawal, and S.Seshan, “Mercury: Supporting scalable multi-attribute range queries,” in Proceedings of ACM SIGCOMM 2004, 2004.

[2] A.Rowstron and P.Druschel, “Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems,” in Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing (Middleware 2001), 2001.

[3] G.S.Manku, M.Bawa, and P.Raghavan, “Symphony: Distributed hashing in a small w orld,” in Proceedings of the 4th conference on USENIX Symposium on Internet Technologies & Systems (USITS 2003), 2003.

[4] I.Stoica, R.Morris, D.Karger, F.Kaashoek, and H.Balakrishnan, “Chord: A scalable peer-to-peer lookup protocol for internet applications,” IEEE/ACM Transactions on Networking, vol.11, no.1, pp.149–160, 2003.

[5] J.Aspnesa and G.Shah, “Skip graphs,” in Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2003), 2003.

[6] S.Ratnasamy, P.Francis, M.Handley, R.Karp, and S.Shenker, “A scalable content-addressable network,” in Proceedings of ACM SIGCOMM 2001, 2001.

[7] T.Schütt, F.Schintke, and A.Reinefeld, “Range queries on structured overlay networks,” Computer Communications, vol.31, no.2, pp.280–291, 2007.

[8] D.Karger and M.Ruhl, “Simple efficient load balancing algorithms for perr-to-peer systems,” in Proceedings of the 16th Annual ACM symposium on Parallelism in Algorithms and Architectures (SPAA 2004), 2004.

[9] Z.Zou, Y.Wang, K.Cao, T.Qu, and Z.Wang, “Semantic overlay network for large-scale spatial information indexing,” Computers & Geosciences, pp.208–217, 2013.

[10] A.Crespo and H.Garcia-Molina, “Semantic overlay networks for p2p systems,” in Proceedings of 3rd International Workshop on Agents and Peer-to-peer Computing (AP2PC 2004), 2004.