

階層型 DHT における 効率的な探索手法の提案と評価

舟橋知論[†] 中村嘉隆^{††} 白石 陽^{††} 高橋 修^{††}

DHT における探索手法である再帰探索と反復探索は, ノードの離脱といった churn 動作の影響により探索の性能が悪化することがわかっている. churn が少ない場合においては反復探索より再帰探索の方が有効に働き, 多くなると再帰探索が不利となり反復探索が有効となる. よって, 最適な探索を行うために, churn の頻度により再帰探索と反復探索を使い分けることが必要となる. そこで本研究では, churn 率によって再帰探索と反復探索を使い分ける探索手法を提案する. しかし, 一般的な DHT では周囲の churn 率を把握することが困難である. DHT を階層化した階層型 DHT があり, ノードの信頼性を考慮してノードを集約する階層型 DHT では churn 率を把握することができる. そこで, 信頼性を考慮した階層型 DHT に対して再帰探索と反復探索を使い分ける探索手法を適用する. また, 再帰探索, 反復探索のどちらか一方のみを利用した場合と比較して, 両探索手法の利点を活かすことができることをシミュレーション評価により示す.

An Effective Strategy for Recursive and Iterative Lookup on Hierarchical DHT

TOMONORI FUNAHASHI[†]
YOSHITAKA NAKAMURA^{††} YOH SHIRAISHI^{††}
OSAMU TAKAHASHI^{††}

Recursive and iterative lookup on DHT performance deteriorates by churn that nodes leave network. When churn occurrence is low, recursive lookup performs better than iterative, but it turns back when churn turns up frequently. Therefore, it needs to separate between recursive and iterative lookup by frequency of churn for optimal lookup. In this study, we propose lookup strategy that separate between recursive and iterative lookup by churn rate. However, common DHT difficult to know neighbor churn rate. In hierarchical DHT, the DHT considered reliability of nodes can know churn rate. Therefore, we compare our lookup strategy to using either recursive or iterative on the hierarchical DHT.

[†]公立はこだて未来大学大学院

Graduate School of Systems Information Science, Future University Hakodate

^{††}公立はこだて未来大学

Future University Hakodate

1. はじめに

Peer-to-Peer (P2P) はノード同士が対等に通信を行うことで拡張性や負荷分散に優れている. P2P ではデータがネットワーク上のノードに分散配置される. そのため, 効率的な検索技術として Distributed Hash Table (DHT) がある. DHT はノード数が非常に多い場合においても, 比較的小さいホップ数で検索可能な技術である. DHT を利用して Chord[1]や Kademlia[2], Pastry[3]など多くのアルゴリズムが提案されている. DHT で利用するアルゴリズムやノードを中継する経路が同一であっても, その経路上のノードと通信を行う方式は別に定義される. その通信を行う方式として, 再帰探索 (Recursive Lookup) と反復探索 (Iterative Lookup) が知られている[4]. これらは探索における通信の方法に違いがあるため, 探索完了までの待ち時間 (探索遅延時間) とメッセージ数の結果に差が生じる. 一般的に, 探索遅延時間がより短い再帰探索が主に利用される. また, メッセージ数は反復探索よりも多いことが言われているが, 大きな差が出ないため重要視されない. しかし, これらの探索方式は DHT におけるノードの離脱 (churn) 動作の影響により, 探索遅延時間やメッセージ数に大きなオーバーヘッドが生じる可能性がある. このオーバーヘッドは再帰探索の方がより大きくなる可能性があることがわかっている. これにより最適な探索を行うためには, ネットワーク内の churn の頻度によって再帰探索と反復探索を適切に利用する必要があると考えられる. しかし, 通常の均一な DHT ではネットワークの churn の頻度などのノードの特性を考慮して構成されていないため, churn 率の正確な予測が困難である.

DHT を効率的に運用するための構造として DHT を階層化した階層型 DHT がある. 階層型 DHT は階層化されていない均一な DHT と異なり, 目的に応じたノードの集約が可能である. この性質を利用した研究として, 信頼性を考慮した階層型 DHT がある. これはノードの churn 率によってノードをクラスタリングするため, 構成されたクラスタの churn 率を把握することができる.

そこで本研究では, 信頼性を考慮した階層型 DHT において, 分割したクラスタごとに churn 率が異なることを利用して再帰探索と反復探索の効率的な使い分けを提案する.

2. 関連研究

この章では, 本研究で用いる DHT アルゴリズムである Chord, DHT における探索手法である再帰探索と反復探索, その性能と churn との関係を述べ, DHT を階層化した階層型 DHT について述べる.

2.1 Chord

Chord はハッシュ空間をリング状の空間として捉えた DHT アルゴリズムの 1 つで

あり、ハッシュ関数を用いてノードにノード ID と呼ばれる値を設定する。あるノードに対して、ネットワークに存在しているノード ID で直前のノードを predecessor ノード、直後のノードを successor ノードと呼ぶ。これらのノード情報を保持すると同時に、探索時の効率化を図るため successor ノードを複数保持する successor List, finger Table という効率的にルーティングを行うための経路表を保持する。これらのノード情報は churn や故障によって情報が古くなってしまふ。そのため、Chord では経路表を正しく保持するために Stabilization 処理と呼ばれる処理を行う。これは経路表のノードに矛盾が無いかどうか、各ノードが経路表内のノードに問い合わせる処理である。この処理は一定の間隔で行われる。

2.2 再帰探索と反復探索

再帰探索はデータ要求を出す要求元ノードが、他のノードに対して探索を依頼していく探索手法である。一方、反復探索は要求元ノードが探索を行い、他のノードに対して次の探索先を問い合わせっていく探索手法である。Chord を用いた探索におけるホップ数（経路長）が 3 である各探索の動作を図 1 に示す。再帰探索は、まず要求元ノードが適切なノードへ目的のデータを要求するメッセージを転送する（図 1 (1-1)）。この要求メッセージを受信したノードがデータを保持していなければ、その受信ノードの経路表内から目的に近いノードを探して要求メッセージを転送する（図 1 (2-1), (3-1)）。さらに、要求メッセージを受信したことを送信元ノードに知らせるために ACK メッセージを返送する（図 1 (2-2), (3-2), (4-2)）。この操作を目的ノードへ到達するまで繰り返す。一方、反復探索では要求元ノードが要求メッセージを転送した際、そのメッセージに対する返答メッセージを受信する（図 1 (2), (4)）。要求メッセージを受信したノードが目的ノードではない場合、返答メッセージには目的ノードにより近いノードのアドレスを含める。この返答メッセージの内容を利用して要求元ノード自身が目的ノードまでの通信を行う。

再帰探索と反復探索の性能は、通信方法と churn の影響によって定まる。ネットワークの churn 率はネットワーク上のノードが存在している時間によって変化する。ここであるノードがネットワークに存在する残りの時間を R とする。 R はノードの信頼性とも呼ばれる。 R が取り得る値はノードごとに異なるが、値を定めるための関数として指数分布[5]やパレート分布[6]の累積分布関数などが使われる。この R によって churn が頻繁に発生するかどうかを把握することができる。次に、ノードの churn や故障が発生した場合、ノードの経路表の矛盾の発見から修復するまでの時間を S とする。 S は経路表の内容を更新する処理時間に加え、Stabilization 処理による時間も含まれる。これは、Stabilization 処理を行う間隔を広くすると S は大きくなり、頻繁に処理を行うことは S が小さくなることを意味する。

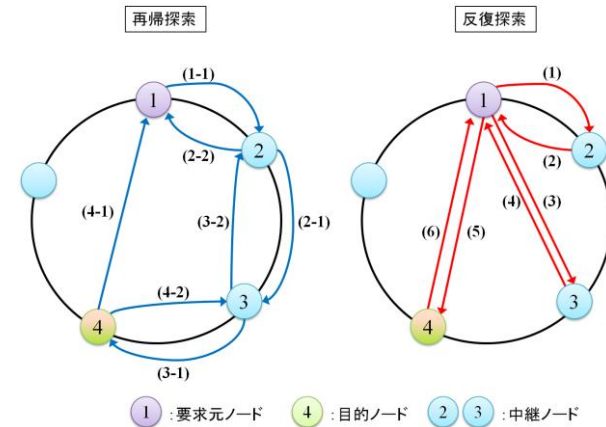


図 1 経路長 3 の場合における再帰探索と反復探索の動作

ここで、あるノードに対する周囲のノードの R についての期待値を $E[R]$, S の期待値を $E[S]$ とする。これらを用いて、探索の際に次ホップ候補のノードがネットワーク上に存在しているノードで、かつ、正確に要求メッセージを送信できる確率 p は次のようになる[7]。

$$p = \frac{E[R]}{E[R] + E[S]} \quad \dots (1)$$

探索開始時に周囲のノードに変化が無く、経路表の更新処理が発生していない場合は $E[S]$ は Stabilization 処理を行う間隔の固定値に近づく。そのため、 p は $E[R]$ に依存する変数となる。 $E[R]$ が大きい場合は周囲のノードが長く存在することを表すため、churn が発生しにくいことを意味する。逆に、 $E[R]$ が小さい場合は周囲のノードがネットワークから離脱する可能性が高く、churn が発生しやすい。これより、 p はネットワークの churn の頻度によって変化するため churn 率を表している。すなわち、 p が大きい場合は churn 率が低く、 p が小さいと churn 率が高いことを意味する。

この churn 率 p と通信遅延によって、再帰探索と反復探索の探索遅延時間が定義される[7]。まず、探索におけるホップ数を表す経路長を l とし、1 ホップにかかる遅延時間を t とする。ここで t はノードの物理的な位置に関係せず、どのノードに対しても一定の値をとると仮定する。さらに、探索中に churn や故障によってノードへのメッセージ送信が失敗してしまう場合におけるタイムアウトの時間を T とする。これより、再帰探索の探索遅延時間の期待値 $E[RL]$ は次のように定義される。

$$E[RL] = (l+1)t + \frac{1-p'}{p'}T \quad \dots (2)$$

また、反復探索の探索遅延時間の期待値 $E[IL]$ は以下ようになる。

$$E[IL] = 2lt + \frac{1-p}{p}lT \quad \dots (3)$$

タイムアウトの時間 T と探索の 1 ホップの遅延時間 t が一定である条件下では、各探索におけるこれらの探索遅延時間の期待値は経路長 l と churn 率 p によって定められる。また式 (2) より、再帰探索では p が小さいと遅延への影響が大きくなることがわかる。一方、反復探索は p が小さくなくてもタイムアウトによる影響を小さく抑えることが出来るが、 p が大きい場合は再帰探索よりも探索遅延時間が大きくなってしまふことがわかる。よって、ネットワークの churn 率が現在どの程度であるかを把握することが、再帰探索と反復探索の性能を十分に活かすために必要である。

2.3 階層型 DHT

階層型 DHT は、均一に論理ネットワークトポロジを構成する DHT を複数のネットワークに細分化して構成する DHT である[8][9]。Chord を用いた 2 階層の階層型 DHT の例を図 2 に示す。階層型 DHT では DHT ネットワークは上位クラスタと下位クラスタに分類される。また、ネットワーク上のノードはスーパーノードと一般ノードに分類される。下位クラスタは一般ノードとスーパーノード、上位クラスタはスーパーノードのみで構成される。スーパーノードはネットワークに参加しているノードから特定の条件を満たすノードが選ばれる。例えば、ストレージ容量や CPU 処理能力が高い高性能なノードやネットワークに長時間存在するノード、帯域幅の広いノードなどが選ばれる。スーパーノードは、所属クラスタ以外のクラスタとの通信を行う場合に他のクラスタへの経路を提供する。階層型 DHT において目的のデータを保持する目的ノードが、同じクラスタに所属しているかどうかはノード ID の上位 m ビットを比較することで判断している。この m は 2^m とすることで階層型 DHT の下位クラスタ数に相当する。探索時には目的データのハッシュ値の上位 m ビットを比較し、同じ場合はクラスタ内の探索を行う。一方、上位 m ビットが異なる場合はスーパーノードを介して上位クラスタで目的のクラスタを探索し、他のクラスタへの探索を行う。

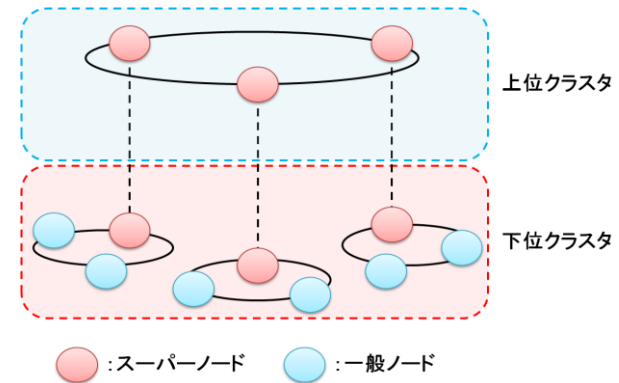


図 2 2 階層の階層型 DHT

階層型 DHT の特徴として、下位クラスタに所属する一般ノードを目的に応じてクラスタリングを行うことや、周囲のノードに対する churn の影響の局所化が可能であることが挙げられる。

この階層型 DHT の研究として信頼性を考慮した階層型 DHT がある[10]。信頼性を考慮した階層型 DHT とは、ノードがネットワークに参加してから離脱するまでの時間によって下位クラスタに所属するノードを決定する階層型 DHT である。この場合における参加してから離脱するまでの時間は残り生存時間の R に相当する。ここでは参加してきたノードの R が近いノード同士を所属クラスタに集めている。また、スーパーノードはクラスタ内で最も R の高いノードが選ばれる。信頼性を考慮した階層型 DHT は R の値によってクラスタリングを行うため、下位クラスタにおける R の期待値 $E[R]$ も同様に、 R の高いノードが集まるクラスタでは $E[R]$ が高く、 R の低いノードが集まるクラスタでは $E[R]$ が低い値となる。ここで Stabilization 処理の間隔を全ノードで一定とすると、 $E[S]$ は全て一定の固定値に近い値を取るようになる。式 (1) より、 p は $E[R]$ の大きさで決定されるためクラスタごとに明確に異なる。これにより、クラスタごとの churn 率の高さを把握することができるため、 p に応じた経路表維持や探索処理をノード動作に取り組むことが可能である。

3. 目的と課題

再帰探索と反復探索は、churn 率 p が低い場合では再帰探索が有効であるが、 p が高い場合は反復探索が有効となる。そこで本研究では、churn の頻度によって探索手法を使い分けて利用し、churn による探索遅延時間の期待値への影響を最小限に抑え

ることを目的とする。また、 p を正確に予測するために、信頼性を考慮した階層型 DHT[10]の構造を利用する。信頼性を考慮した階層型 DHT では、 R によってノードをクラスタリングした結果として、 p が高いクラスタと低いクラスタに分けられる。このクラスタごとに p が異なることを利用して、探索手法を p によって使い分けることができる。しかし、再帰探索と反復探索では探索中にやりとりされるメッセージの内容が異なるため、異なる探索手法をとるクラスタ間の通信を行うことができない。

そこで本研究では、再帰探索と反復探索の相互のメッセージに対して変換処理を行うことで、再帰探索と反復探索の切り替えを行う方式を提案する。また、この変換処理を用いて、各クラスタの探索手法を使い分ける方式が再帰探索、反復探索のどちらか一方のみを用いる場合よりも有効であることを示す。

4. 提案方式

本章では、まず 4.1 節提案方式の概要を述べ、4.2 節以降で提案手法の使い分けについて詳細に述べる。

4.1 提案方式概要

提案方式では信頼性を考慮した階層型 DHT の構成を利用し、再帰探索と反復探索をクラスタごとに使い分けて探索を行う。ここで DHT アルゴリズムは構造が簡易で拡張性があり、また Stabilization 処理を行う Chord を利用する。参加ノードの信頼性 R によってクラスタリングを行って各下位クラスタを構成するが、各クラスタのスーパーノードはクラスタ内ではなくネットワーク内で R の高いノードとする。よって、スーパーノードの R とクラスタに集合するノードの R とは関係が無い。ここではスーパーノードの R を R_s とし、一般ノードの R を R_n として分類する。 R_n の低いノードが集合する下位クラスタ（低信頼クラスタ）では、churn 率を示す p の値が小さいことから反復探索を用いて探索を行う。一方、 R_n の高いノードが集合する下位クラスタ（高信頼クラスタ）では再帰探索を用いて探索を行う。また、上位クラスタはスーパーノードによって構成されるため R_s の集合となる。 R_s はネットワーク内で R が高いことから再帰探索によって探索を行うこととなる。これにより、ネットワーク全体で再帰探索と反復探索が混在する。そこで、再帰探索から反復探索へ、また反復探索から再帰探索への要求メッセージの変換処理をスーパーノードが行うことで上位の高信頼クラスタと下位の低信頼クラスタ間の探索を行う。

探索において利用する経路表の構成は通常 Chord で構成される経路表に加え、階層型 DHT を構成するための経路情報を持つ。例えば、一般ノードは同じクラスタに所属するノードのみを経路表に追加し、所属クラスタのスーパーノードの情報を持つ。また、スーパーノードは下位クラスタの通常経路表を保持し、さらに上位

クラスタの他のスーパーノードの情報を持つ。

4.2 メッセージの変換

再帰探索と反復探索では要求メッセージやそれに対する返答メッセージがやりとりされるが、各探索においてそれぞれのメッセージの扱いは変わる。例えば、返答メッセージは反復探索において次ホップ候補の情報を含むことで探索を進行するが、再帰探索においては ACK メッセージとして扱うため情報は少ない。各探索における要求メッセージと返答メッセージが最低限必要な情報を表 1、表 2 に示す。

表 1 要求メッセージに必要な情報

	メッセージ識別子	目的 ID	要求元ノードアドレス	TTL
再帰探索	○	○	○	○
反復探索	○	○	—	—

表 2 返答メッセージに必要な情報

	メッセージ識別子	次ホップ候補
再帰探索	○	—
反復探索	—	○

再帰探索は要求メッセージの転送を他のノードに任せるため、並列的な探索が可能である。そのため、要求元ノードが最終的に受信するデータがどの要求メッセージに対するものなのかを識別するためのメッセージ識別子や要求元ノードのアドレス、さらに要求メッセージがネットワーク内で無限に転送されることを防ぐための TTL (Time To Live) を設定する。また、返答メッセージは ACK メッセージであるため、要求メッセージに対応した識別子を含めるのみである。一方、反復探索では要求元ノードが探索を制御するため識別子などは必要が無く、目標とする ID 情報のみを要求メッセージに含める。しかし、反復探索は要求メッセージを送信した相手からの返答メッセージによって探索を行うため、返答メッセージには次ホップ候補を複数含める必要がある。このようなメッセージ内容の違いにより、探索手法が異なる場合はメッセージ内容の読み取りと探索を続けることができない。

そこでメッセージ内容の違いを吸収し、探索手法を変換する処理を行う。この処理では再帰探索から反復探索へ、反復探索から再帰探索への変換を行い、階層化されていない均一な DHT の場合では全てのノードで行うことができる必要がある。しかし、提案方式が想定している環境では各探索手法を用いる領域が階層化を行うことでクラスタという単位で扱えることから、異なるクラスタ間の接点で変換処理を行うことで変換処理によるノードの負荷やオーバーヘッドを抑えることができる。提

案方式において、この異なるクラスタ間の接点は低信頼クラスタのスーパーノードに相当する。これは低信頼クラスタが反復探索を使い、上位クラスタが再帰探索を使うためである。低信頼クラスタのスーパーノードは低信頼クラスタ内から上位の高信頼クラスタへ、また上位の高信頼クラスタから低信頼クラスタ内への再帰探索と反復探索の相互変換を提供する。

スーパーノードが反復探索の要求メッセージを受信した場合、宛先が他のクラスタであれば、反復探索の要求メッセージ内容から再帰探索の要求メッセージを作成し探索を行う。しかし、反復探索のメッセージではメッセージの識別子や要求元ノードアドレス、TTL が不足している。そのため、メッセージ識別子を新たに設定し、メッセージ内の中継経路情報から TTL を設定し要求メッセージの作成を行う。また、要求元ノードアドレスはスーパーノードを指定する必要がある。これは低信頼クラスタの一般ノードは再帰探索から反復探索への変換処理を持たないため、再帰探索によるメッセージを受信しても読み取ることができない。そこで、スーパーノードは再帰探索により目的ノードを発見し、結果として受信したデータを要求元ノードへ送信するために再帰探索から反復探索への返答メッセージの変換処理を行う。

一方、再帰探索の要求メッセージを受信した場合はメッセージ内容から目標 ID を読み取ること、反復探索の要求メッセージを作成することができる。同様に、目的ノードからのデータはスーパーノードを介し、変換処理を行うことで送信される。

4.3 探索の動作

提案方式では、下位クラスタを高信頼クラスタ、低信頼クラスタに分類し、それぞれ再帰探索と反復探索を用いて探索を行う。また、上位クラスタが再帰探索を行うことになるため、提案手法の探索の動作は大きく分けて 2 つのパターンが考えられる。(A) 下位の低信頼クラスタから上位の高信頼クラスタの反復探索から再帰探索、(B) 下位の高信頼クラスタから下位の低信頼クラスタへの再帰探索から反復探索のパターンに分類できる。

まず、パターン (A) の探索の動作例を図 3 に示す。

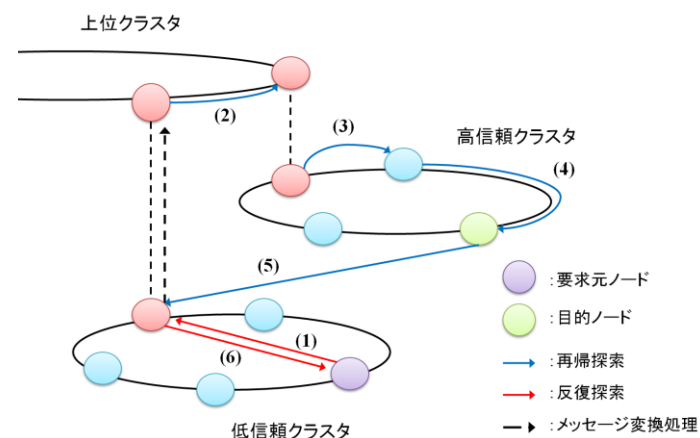


図 3 低信頼クラスタから高信頼クラスタへの探索

探索における要求メッセージと ACK を除く返答メッセージの伝搬の流れは図中の番号で示している。また、上位クラスタと下位クラスタにあるメッセージ変換の矢印は変換処理の実行と探索手法の変換方向を示している。この例では、反復探索の要求メッセージの内容から再帰探索の要求メッセージを作成するメッセージ変換処理を行っている。まず要求元ノードは、反復探索の形式でスーパーノードへ他のクラスタへの探索を依頼する(図 3 (1))。そこでスーパーノードは変換処理を行い、再帰探索を開始する。再帰探索により他のクラスタを探索、目的ノードを発見する(図 3 (2) ~ (4))。目的ノードは要求元ノードへ直接データを送信せず、要求元のクラスタのスーパーノードへ送信する(図 3 (5))。目的ノードから返答メッセージを受信したスーパーノードは、要求元ノードが理解できるように反復探索のメッセージ形式に変換し、データを送信する(図 3 (6))。この探索パターンでは、反復探索のみを用いるより、churn の影響が少ない部分を再帰探索で行うことで全体の探索遅延時間を短くすることができると思われる。

次に、パターン (B) の探索について、動作例を図 4 に示す。

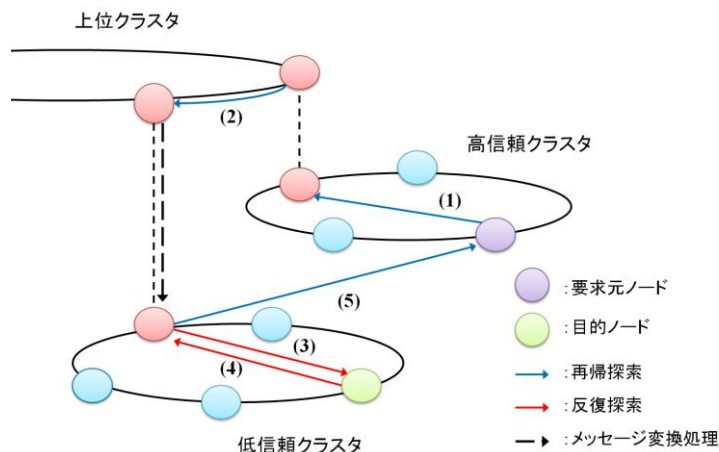


図4 高信頼クラスタから低信頼クラスタへの探索

この探索パターンでは、スーパーノードは高信頼クラスタからの再帰探索による要求メッセージから反復探索の要求メッセージを作成する変換処理を行っている。よって、要求元ノードが再帰探索による要求メッセージを送信した場合、再帰探索により探索を行う範囲は宛先クラスタのスーパーノードまでとなる(図4(1), (2))。スーパーノードはメッセージの変換処理を行って、反復探索による目的ノードの探索を行う(図4(3), (4))。ここで図中(4)の通信により目的データを得ることとなる。スーパーノードに一旦データが渡されてから、要求元ノードのアドレスに再帰探索の返答メッセージに受信したデータを含めて送信する(図4(5))。この探索手法は、再帰探索のみで全ての探索を行うよりも churn の影響が高いことがわかる。低信頼クラスタの部分を実験探索で行うことで、探索遅延時間への影響を抑えることができると考えられる。

5. 実験および評価

提案手法の有効性を評価するため、Overlay Weaver[11]上に提案方式を実装し、既存の再帰探索と反復探索のみを利用した場合との性能を比較する。

5.1 実験環境

本節では、シミュレーションを行った実験環境とシミュレーションで設定したパラメータの詳細について述べる。まず、実験に用いるマシン性能と Overlay Weaver

のバージョンは以下の通りである。

- OS : Winsows 7 Professional 64bit
- CPU : Intel Core i5 3.2GHz
- メモリ : 4.0GB
- Overlay Weaver : 0.10

次に、シミュレーションにおけるパラメータの設定を表3に示す。

表3 シミュレーションにおけるパラメータ一覧

全ノード数 (N)	1000
クラスタ数 (C)	4
遅延時間 (t)	2msec
タイムアウト時間 (T)	5msec

C の値は階層型 DHT におけるスーパーノードの数に相当するため、 N のうち C ノードがスーパーノードとして動作する。残りのノードが一般ノードとして下位クラスタを構成するが、 R の分布に関係無く、クラスタ間のノード数に大きな差は出ないものとする。表3に記載していないパラメータのうち、経路長 l は目的 ID に左右する。目的 ID は再帰探索と反復探索、提案手法において同じ ID を使用することで、 l による探索への影響をできるだけ均一にする。今回のシミュレーションでは同一の目的 ID に対する探索を 100 回試行し、その平均によって l の影響を均一にする。また、再帰探索と反復探索の churn の影響を明確にするため、高信頼クラスタと低信頼クラスタは毎試行において同様のものとする。ここで、高信頼クラスタは p が常に 1 であり、再帰探索を用いるとし、低信頼クラスタとは反復探索を用いるクラスタで p が自由に設定可能なクラスタであるとする。また、式(1)において p が churn 率を示すために S は固定値に近い値である必要がある。そのため、探索中のノードの参加や離脱による経路表の変更処理は発生しないようにする。さらに、Stabilization 処理の実行間隔は 125msec と広く設定し、経路表変更が探索中に発生するのを抑えて $E[S]$ を固定値とみなすこととする。

以下に各ノードが保持する経路情報を示す。

- predecessor ノード
- successor List (最高で 8 ノード分の successor を保持)
- finger Table
- 一般ノードは所属クラスタのスーパーノードの情報を持つ
- スーパーノードは上位クラスタの他のスーパーノードの情報を持つ

ここで、一般ノードは所属クラスタ以外のクラスタを探索する場合、宛先が他のクラスタであることを判断し、スーパーノードへ1ホップで要求メッセージを送信できるものとする。また、上位クラスタにおける探索は上位クラスタのノードを全て経路表に持ち、1ホップで宛先クラスタを探索できるものとする。

このようなパラメータ条件下で、高信頼クラスタから低信頼クラスタへの探索、またその逆への探索における探索遅延時間を各探索手法について計測する。

5.2 性能評価

各探索手法の探索遅延時間について評価を行う。ここで、探索遅延時間は要求メッセージが目的ノードに到達するまでの時間とし、さらに各ノードの内部処理の時間も含む。よって、評価における再帰探索、反復探索の探索遅延時間の期待値 $E[RL]$ 、 $E[IL]$ はそれぞれ以下の式となる。

$$E[RL] = lt + \frac{1-p^l}{p^l} T \quad \dots (4)$$

$$E[IL] = (2l-1)t + \frac{1-p}{p} lT \quad \dots (5)$$

また、性能評価は4.3節で述べた2つの探索パターンに分けて行う。

まず、パターン(B)のような要求元のクラスタが低信頼クラスタであり、目的のクラスタが高信頼クラスタである場合の探索について比較評価を行う。ここでは、高信頼クラスタはchurnの設定を行わないため、 $p=1$ とする。また、低信頼クラスタの要求元ノードが探索終了まで離脱しないと仮定し、スーパーノードに関しても同様に $p=1$ と設定することで、この探索では p の影響が無い場合の探索であると言える。この場合における各探索手法の平均探索遅延時間について、シミュレーション結果を図5に示す。

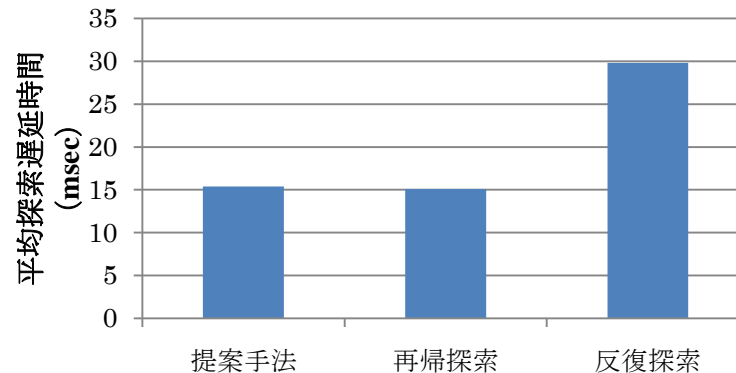


図5 低信頼クラスタから高信頼クラスタへの探索における平均探索遅延時間

図5より、 p の影響が無く、経路長に差がほとんど無い場合、反復探索よりも再帰探索や提案手法がおおよそ2倍の速さで目的ノードを発見できることが分かる。また、再帰探索と提案手法を比較した場合についても、ほぼ同等の性能を持つことが分かる。

次に、パターン(A)のような要求元のクラスタが高信頼クラスタであり、目的のクラスタが低信頼クラスタである探索の場合について比較評価する。ここでは、目的の低信頼クラスタにおいて p を設定する。ここで p は R の分布に関係無く、churnの影響が無い1からchurnの頻度が比較的高い0.5までの範囲を0.1ずつ変化させる。この場合における探索遅延時間の推移を図6に示す。

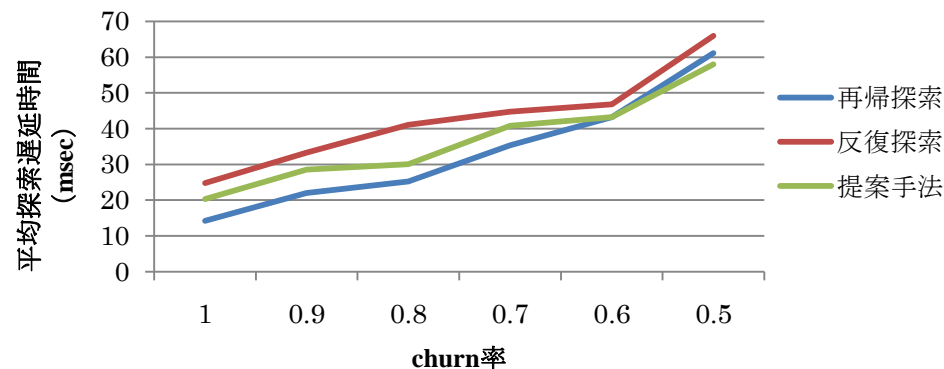


図6 高信頼クラスタから低信頼クラスタへの探索における平均探索遅延時間

図 6 より, $churn$ が多くなると探索遅延時間も増加していることがわかる. 再帰探索と反復探索を比較すると, p が高い時の探索遅延時間は 2 倍ほど差があったにも関わらず, p が低い時ではその差はかなり小さくなっている. このことから, 再帰探索の方が $churn$ による探索遅延時間の増加率が大きいことがわかる. この増加率の原因として考えられる探索中におけるメッセージの送信失敗回数について, シミュレーション結果内から各探索手法における平均送信失敗回数を抽出した結果を図 7 に示す.

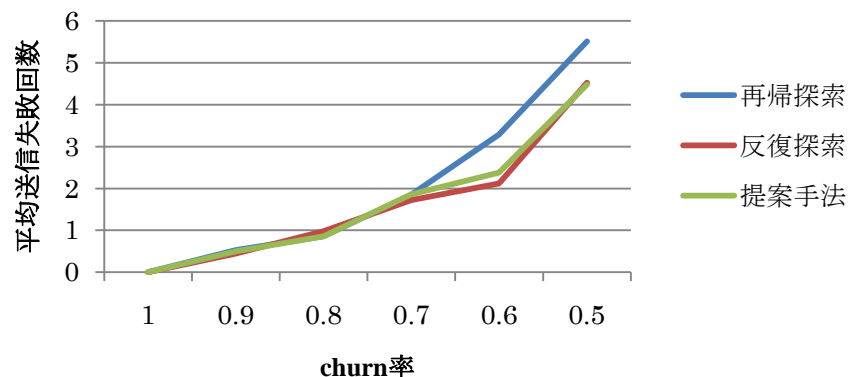


図 7 高信頼クラスタから低信頼クラスタへの探索におけるメッセージの平均送信失敗回数

図 7 から, 再帰探索における探索遅延時間の増加には平均送信失敗回数が原因として含まれることがわかる. しかし, 低信頼クラスタを反復探索で行うことでメッセージの平均送信失敗回数は抑えることができ, 提案手法に対しても同様の効果が得られていることがわかる.

また, これらのシミュレーション結果は目的ノードを発見できた場合の試行に限定されている. $churn$ によって探索中に, 送信失敗増加による探索のタイムアウトや中継経路の情報から探索を中断されることによる目的ノードを発見できないケースがある. 全試行回数とこの目的ノードを発見できなかった回数の割合と $churn$ 率の関係について図 8 に示す.

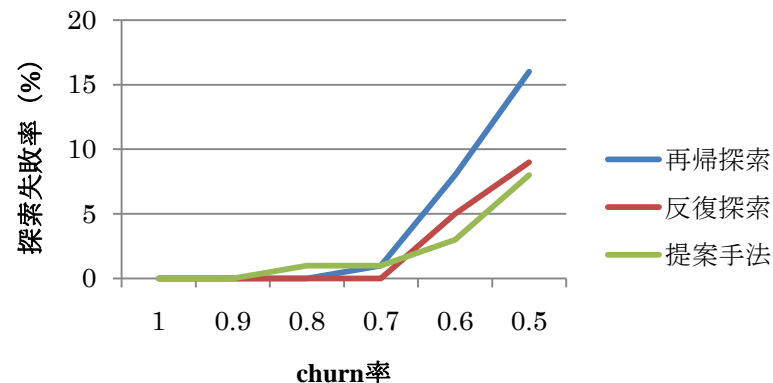


図 8 高信頼クラスタから低信頼クラスタへの探索における探索失敗回数

図 8 から, $churn$ の増加によって再帰探索は探索に失敗するケースが他の探索に比べ多いことがわかる. しかし, 低信頼クラスタで反復探索を用いることで探索失敗に陥るケースも抑えることができている. これも, 提案手法において同様のことが言える.

6. 考察

まず, 低信頼クラスタから高信頼クラスタへの探索のシミュレーション結果について考察する. この場合の探索の様子は図 4 に示されており, 図 4 から提案手法が目的ノードを発見する過程は再帰探索の動作と大きな差が無いことがわかる. よって, 提案手法が再帰探索と同等の平均探索遅延時間を持つことは妥当であると考えられる. またこのことから, スーパーノードにおけるメッセージの変換処理による遅延時間への影響は無視できる程度のものであることも考えられる.

次に, 高信頼クラスタから低信頼クラスタへの探索のいくつかのシミュレーション結果について考察する. 図 6 より, 提案手法は p が高い時は反復探索よりも比較的遅延時間が短く, 再帰探索より遅延時間が長くなっている. しかし, p が低くなっていった場合, $churn$ により生じるオーバーヘッドを抑えることができているため, 再帰探索よりも探索遅延時間が短くなる場合もあることがわかる. この結果について, 本研究ではこの平均遅延時間の関係が式 (4), (5) を満たさないことに注目する. 式 (4), (5) は再帰探索と反復探索における目的ノードを発見するまでの探索

遅延時間の期待値として定義されている。ここで、図 8 が示した探索失敗回数を考慮する。探索遅延時間の期待値は図 6 による探索成功時の平均遅延時間と、図 8 による探索失敗のリスクが含まれると考えられる。よって、探索成功時の平均遅延時間は図 6 のようになったとしても、図 8 のように探索失敗による時間のリスクを含めて考えた場合に期待値としては図 6 のようにはならないと言える。また、例えば探索のタイムアウトは一般的に目的ノードを発見する遅延時間よりも大きいと言えるため、この探索失敗のリスクを適切に設定することで、期待値としての値は式 (4), (5) に近づくと考えられる。これらのことを踏まえ、探索遅延時間の期待値として結果を考察すると、平均探索遅延時間が短い方の探索により近く、探索失敗回数がより少ない方の探索に近い提案手法が最も探索遅延時間の期待値が高いと言える。

7. おわりに

本稿では、信頼性を考慮する階層型 DHT におけるクラスタごとの churn 率の違いと再帰探索と反復探索の churn 率の影響に着目して、クラスタごとに探索手法を使い分けて両探索手法の利点のみを活用する探索手法を提案した。そして、この探索手法の探索遅延時間が再帰探索、反復探索どちらかのみを利用した場合よりも有意であることをシミュレーションにより検証を行った。結果として、再帰探索と反復探索それぞれの有効な場合に探索手法を使い分けることで、提案方式が探索遅延時間の期待値として評価した場合に最も有効であると言える。今後は、ノードの信頼性を具体的な分布によって定義し、churn 率を測定することで動的にその状況に有利な探索手法に切り替える手法の検討が考えられる。また、そのクラスタで用いる探索手法によってノード数が少ない方が有効に働く場合が考えられる。例えば、(2) より、再帰探索はノード数が少ない方が有効に働くと考えられる。今後はこのような経路長やノード数、クラスタ数など様々なパラメータを考慮して再帰探索や反復探索の使い分け、またそのための最適な DHT の構成を検討する。

参考文献

- [1] I. Stoica, R. Morris, D. Karger, M. Frans Kaashoek, and H. Balakrishnan, “Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications”, Proceedings of the SIGCOMM, 2001.
- [2] P. Maymounkov, D. Mazieres, “Kademlia: A Peer-to-peer Information System Based on the XOR Metric”, IPTPS02, pp. 53-65, 2002.
- [3] A. Rowstron and P. Druschel, “Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems”, IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), pp. 329-350, 2001.
- [4] 首藤一幸, 加藤大志, 門林雄基, 土井裕介, “構造化オーバーレイにおける反復探索と再帰探索の比較”, 情報処理学会研究報告「システムソフトウェアとオペレーティング・システム」, Vol. 86, pp. 9-16, 2006.
- [5] 洞井晋一, 松浦知史, 藤川和利, 砂原秀樹, “時間に基づく階層化と Value の集約配置手法による耐 Churn オーバーレイネットワーク”, 情報処理学会論文誌 Vol. 51, No. 4, pp. 1142-1151, 2010.
- [6] S. Saroiu, “Measurement and analysis of Internet content delivery systems. ”, Doctoral Dissertation, 2004.
- [7] D. Wu, Y. Tian, K. W. Ng, “An analytical study on optimizing the lookup performance of distributed hash table systems under churn”, Concurrency and Computation: Practice & Experience, Vol. 19, pp. 543-569, 2007.
- [8] L. Garcés-Erice, E. W. Biersack, P. A. Felber, K. W. Ross, and G. Urvoy-Keller, “Hierarchical Peer-to-peer Systems”, ACM/IFIP International Conference on Parallel and Distributed Computing, 2003.
- [9] S. Zoels, Z. Despotovic, W. Kellerer, “On hierarchical DHT system – An analytical approach for optimal designs”, Computer Communications, Vol. 31, pp. 576-590, 2008.
- [10] 佐藤文明, “参加脱退率を考慮した階層的 DHT システムの構成方式”, 情報処理学会論文誌 Vol. 51, No. 2, pp. 418-428, 2010.
- [11] 首藤一幸, “Overlay Weaver: An Overlay Construction Toolkit“, HTML Available at, “<http://overlayweaver.sourceforge.net/index-j.html>”.