

構造化オーバレイにおける 柔軟な経路表を活用したネットワーク近接性の考慮

宮尾 武裕[†] 長尾 洋也[†] 首藤 一幸[†]

東京工業大学[†]

1. はじめに

Chord[1]は経路表の管理をノード ID の距離によって厳密に行っているため、経路表に追加するノードに制限がある。そのため通信遅延を考慮することが難しい。Chord の改良版である FRT-Chord[2]は経路表の管理をノード ID の距離の分布によって行っているため、そういった制限なく経路表にノードを追加することができる。そこで本研究では通信遅延が閾値以下であるノードを経路表に入れることで、ネットワーク近接性を考慮できるような FRT-Chord の拡張アルゴリズムを提案する。

ある閾値を用いたときに、平均遅延が 20%改善され、提案アルゴリズムの有効性が示された。

2. FRT-Chord

FRT-Chord では、自由に経路表へノードを追加することができる。経路表がノードで満たされているとき、削除アルゴリズムによって経路表の管理が行われる。その削除アルゴリズムは経路表内で自ノードと i 番目に近いノード i ($i = 1, 2, \dots$) と自ノードとの ID 距離 d_i ($d_0 = 0, d_i < d_{i+1}$) とすると、 $\log d_i - \log d_{i-1}$ が最小となるものを削除することで、経路表に入っているノードの ID 距離の分布を最適なものに近付ける。

3. 提案アルゴリズム

ノードを経路表に追加する時点で通信遅延が閾値以下のノードのみを追加する。通信遅延が閾値より大きいノードは追加しない。ただし、到達性を保証するために successor と predecessor は遅延に関係なく経路表に追加する。

このアルゴリズムによって経路表内にあるノードとの平均遅延が小さくなるのは明らかである。しかし、経路表に追加するノードを限定するため、経路長が長くなりルーティングにかかる遅延が大きくなることも考えられる。そこで実験により経路長が長くなり遅延が小さく

なることを示す。

4. 実験手法

インターネットのトポロジモデルである TS モデル[3]を用いて通信遅延を設定した。TS モデルはトランジットノードとスタブノードの 2 種類のノードで構成されている。トランジットノード間、トランジットスタブノード間、スタブノード間の遅延をそれぞれ 100ms、20ms、5ms と設定した。各ノード間の通信遅延の分布は図 1 のようになった。最大値は 1000ms、平均は 470ms である。

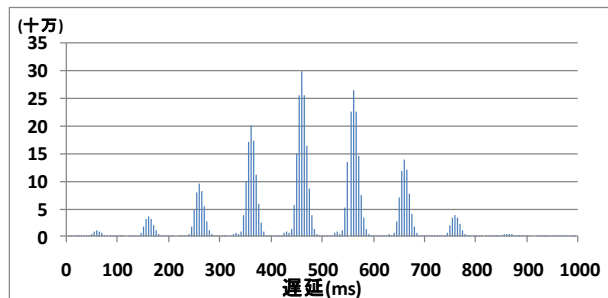


図 1 ノード間の通信遅延の度数分布

オーバーレイ構築ツールキットである Overlay Weaver[4]上に提案アルゴリズムを実装し、一台のマシン上でシミュレーションを行った。閾値を 100ms から 1000ms まで 100ms 毎に変化させた。ノード数を 1 万、経路表の大きさを 16 とし、1 ノードあたり 10 クエリを行うことで経路表を充実させた後 1 ノードあたり 1 クエリつまり 1 万 クエリを行い、その遅延と経路長を計測した。以下の環境で実験を行った。

- Overlay Weaver 0.10.1
- OS: Windows 7 Professional 32bit
- CPU: Intel Core 2 Quad Q9400 2.67GHz
- メモリ: 4.00GB

5. 実験結果

提案手法における閾値と平均遅延、平均経路長の関係は図 2 のようになった。平均遅延は閾値が 400ms のときに 2540ms で最小となった。ノード間の通信遅延の最大値は 1000ms なので閾値を 1000ms にすると、提案アルゴリズムを使わないそのままの FRT-Chord といえる。1000ms のと

Proximity-aware Structured Overlays with Flexible Routing Tables

[†]Takehiro Miyao [†]Hiroya Nagao [†]Kazuyuki Shudo

[†]Tokyo Institute of Technology

きの平均遅延は 3150ms なので遅延が約 20%改善されたといえる。

また平均経路長は閾値 400ms から 1000ms までほぼ一定であるので、経路表内にあるノードの通信遅延が小さい分閾値が 400ms のとき平均遅延が最小になったとわかる。

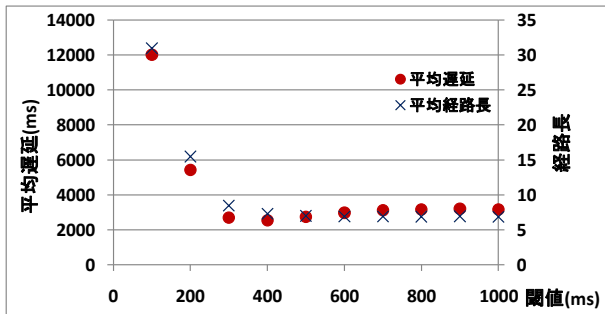


図 2 閾値による平均遅延と平均経路長の変化

図 3 から閾値以下であるノード間の通信遅延の平均が閾値 100ms では 55ms、400ms では 300ms であることがわかる。閾値が 100ms の平均経路長が 30、400ms では 7 であるので、100ms のほうが平均遅延が小さくなると考えられるが、実際は図 1 の結果のとおり閾値が 400ms のほうが 100ms よりルーティングにおいて平均遅延は明らかに小さい。この理由は図 4 より説明できる。

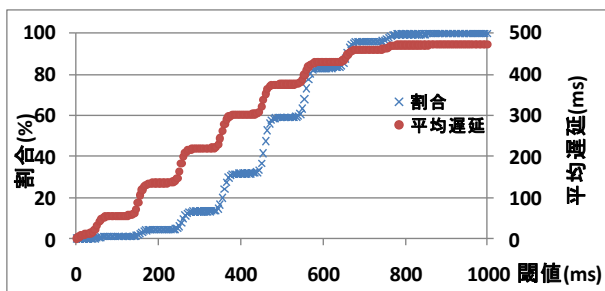


図 3 ノード間の通信遅延が閾値以下である割合とその平均遅延

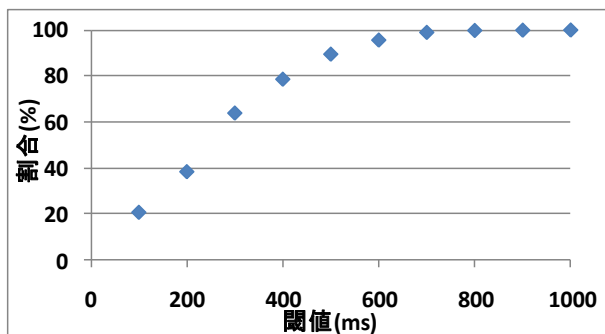


図 4 ルーティングにおいて通信遅延が閾値以下であるノードを使用した割合

図 4 はルーティングにおいて通信遅延が閾値以下であるノードを使用した割合を表したものである。経路表には提案アルゴリズムにより通信遅延が閾値以下のノードしか入らないが、

successor と predecessor だけは例外である。つまり図 4 はルーティングにおいて例外として経路表に維持されている successor を使用した割合である。閾値が 100ms のときルーティングにおいて通信遅延が閾値以下のノードを使用した割合は 20%である。残りの 80%では successor を使用している。successor は図 1 の分布をもつ通信遅延が平均 470ms のノードである。また閾値が 400ms のとき通信遅延が閾値以下のノードを使用した割合は 80%である。よって 1 ホップあたりの平均遅延は閾値 400ms のほうが 100ms より小さくなり、平均経路長の短い 400ms のほうがルーティングにおける平均遅延が小さくなる。

また閾値 100ms ではルーティングにおいて 80%以上で successor が使用されている。なぜなら図 3 より閾値が 100ms のときノード間の通信遅延が閾値以下である割合は全体の 5%程度しかないので、経路表を適した分布に整えられなかったからであると考えられる。

6. まとめ・今後の課題

通信遅延が閾値以下のノードのみを経路表に入れることで追加するノードを減らしても、適切な閾値ならば経路長が長くないことが分かった。また閾値を小さくすることで経路表内のノードとの平均遅延が小さくなる。よって最適な閾値ではルーティングにおいて平均遅延が小さくなるといえる。

クエリ数を増やすとその分経路表に追加するノード数が増えていくので、閾値により追加するノードの割合が小さくなくても経路表が充実する。つまり、最適な閾値はこなししたクエリ数によって変化する。また閾値はノード数や経路表の大きさにも関係があると予想できるので、それらを考慮した閾値の求め方を考えたい。

参考文献

- [1] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer to Peer Lookup Service for Internet Applications," SIGCOMM'01, 2001.
- [2] 長尾 洋也, 首藤 一幸, "柔軟な経路表によるオーバレイネットワークのルーティング方式," DICOM2010, 2010.
- [3] E. Zegura, K. Calvert, and S. Bhattacharjee, "How to model an internet network," INFOCOM'96, 1996.
- [4] 首藤 一幸, "Overlay Weaver," <http://overlayweaver.sourceforge.net/>.