位置情報を用いた構造化P2PのChurn環境におけるコスト評価

M2007MM006 久田 章太
指導教員：河野浩之

1 はじめに

構造化 P2P オーバレイネットワークはスケーラブルなデータ場所を提供し，大規模分散アプリケーションなどにおいて重要視されているが，Churn 問題や物理トポロジとオーバレイトポロジが一致しないなどの課題がある．

本研究では物理トポロジとオーバレイトポロジの不一致を軽減させるために，ヒルベルト曲線とランドマーククラスタリングにより，物理トポロジの位置情報をノード ID に関連付けるアルゴリズムを提案する．そして，高い Churn 環境下においてもよいパフォーマンスを提供するためにリング構造オーバレイネットワークを用い，率先的なショートレンジリンクと確率的なロングレンジリンクを導入する．これらにより上記の問題の解決を図る．

2 構造化 P2P ネットワークとその関連研究

2.1 構造化 P2P ジオメトリに関する研究

K. Gummadi らは [1] で構造化 P2P システムで使われているジオメトリをルーティングジオメトリとルーティングアルゴリズムにより分類した．ルーティングジオメトリとはノードと経路の配置であり，ルーティングアルゴリズムはルーティングジオメトリ上での次のホップや経路が選ばれる方法である．彼らはさまざまなルーティングジオメトリの比較を行うために隣人選択柔軟性と経路選択柔軟性を使用した．隣人選択柔軟性はノードがジオメトリに基づく隣人を選ぶ際に，隣人として選ぶことができるノードの量と定義した．経路選択柔軟性は検索を行う際に，ノードが次のホップ先として選ぶことができる経路の量と定義した．その結果，Ring ジオメトリが隣人選択柔軟性，経路選択柔軟性，ともによい結果を示すとした．各ジオメトリの隣人選択柔軟性と経路選択柔軟性を表 1 に示す (n はジオメトリに参加しているノード数)．

2.2 Churn に関する研究

Churn とはノードの頻繁な参加，離脱の事であり，各ノードが自律的で対等な P2P オーバレイネットワークにおいてよく起こる重要な問題である．Churn が高いと，ほんの少し前に DHT においたデータが次の瞬間に得ることができない状況が頻繁に起こってしまう．

P. Kersch らは [3] で Churn 環境下での構造化 P2P システムのルーティングオーバレイのメンテナンスを分析した．彼らはリングトポロジにおいて率先的な自己組織化ショートレンジリンクと確率的なロングレンジリンクの二重戦略を提案した．ノードの参加や離脱はポアソン過程に由来させ，ロングレンジリンク接続の確率過程はマルコフ連鎖モデルに由来させた．そして Churn 環境下での分析のために Liven-Nowell らが [4] で定義したシステムの半減期を使用した．システムの半減期とは新しいノードの参加により，システム内の全ノードのうち半分が入れ替わるまでの時間である．シミュレーションにおいて，ロングレンジ接続密度が $[1.4, 3.0]$，ショートレンジ接続数が $[3, 5]$ の時が最適であるとした．またショートレンジ接続メンテナンス頻度は $[10, 50]$ が最適であるとした．一方，相対的検索率はアプリケーションに依存するため最適値を決めることができないとした．しかしシステムの半減期が 1 分という非常に高い Churn 環境下においてもネットワークを維持することができ，ネットワークサイズと Churn 環境から独立した非常に高い有効性があると結論した．

3 位置情報を考慮した構造化 P2P の提案

3.1 位置情報を考慮したノード ID

本研究ではランドマーククラスタリングとヒルベルト曲線により，物理トポロジ上で距離が近いノード同士をオーバレイネットワーク上でも近くなるような ID を生成する．図 1 にその流れを示し，説明する．

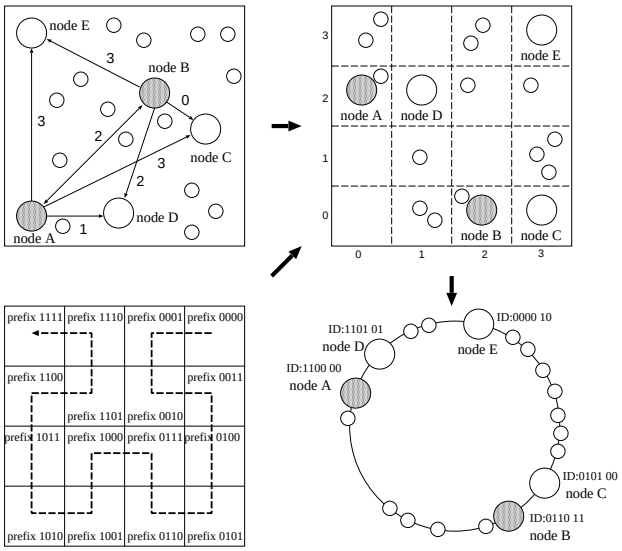

図 1 1 次元へのマッピング

図 1 左上のようにノードが配置されているとする．小さな白丸はその他大勢のノードである．node A，node B をランドマークとする．各ノードはランドマークからの距離を測定し，ランドマークを軸としたユークリッド空間へ再マッピングする (図 1 右上)．図 1 左下のようにヒルベルト曲線が通った順にプレフィックス ID を割り当てる．図 1 右上と図 1 左下を重ね合わせ，ノードのプレフィックス ID を決定する．サフィックスにランダム ID を充てることでノード ID を決定し，図 1 右下ようなオーバレイネットワークを構築する．

表 1 構造化 P2P のジオメトリに関する柔軟性比較

ジオメトリ	Tree	Hyper Cube	Ring	Butterfly	XOR	Hybrid
隣人選択柔軟性	$n^{\log \frac{n}{2}}$	1	$n^{\log \frac{n}{2}}$	1	$n^{\log \frac{n}{2}}$	$n^{\log \frac{n}{2}}$
経路選択柔軟性	1	$(\log n)!$	$(\log n)!$	1	1	1

3.2 ロングレンジリンクの生成モデル

本研究では P. Kersch ら [3] が提案したポアソン過程に基づく確率的ロングレンジリンクを用いる．各リンクは双方向接続，双方向ルーティングとする．リング構造オーバレイネットワーク上の反対側のノードを最も遠いノードとし，時計回り・反時計回りにおいてロングレンジリンク・ショートレンジリンクを持つ．ノードは最も近い N_S 個のノードへのショートレンジリンクを持つ．d_{min}，d'_{min} を最も遠いショートレンジリンクノードへの距離として定義する．ロングレンジリンクは最も遠いノードから距離を減少させる方向で生成する $(L_0, L_1, ..., L_N,\ L'_0, L'_1, ..., L'_N)$.

図 2 に灰色の丸のノードが持つ確率的リング構造オーバレイネットワークのノード接続を示す．図の黒丸がロングレジリンクを結んだノードで，斜線の丸がショートレンジリンクを結んだノードである．また，点線の先のノードが反対側のノードである．

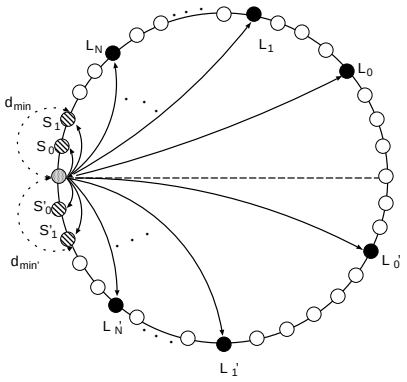

図 2 確率的接続を用いたリング構造オーバレイ

ロングレンジリンクの生成は到着時間間隔 λ の定常ポアソン過程に依存する．この到着時間間隔 λ を接続密度として定義する．この λ は検索パフォーマンスとメンテナンスアルゴリズムに重要な働きを示す.

図 3 にロングレンジリンクを生成する範囲を示す．d_i を i 番目のロングレンジリンクとした時，$i+1$ 番目のロングレンジリンク d_{i+1} が接続する範囲を $[\frac{d_{i+1}}{c}, d_{i+1}c]$ と定義する．もし $d_i < d_{i+1}c$ なら，次に接続されるロングレンジリンクが距離空間で常に近くなるように，d_i より大きい範囲をカットする．そしてこの範囲内にあるノードの 1 つとロングレンジリンクを結ぶ (図 3 の白丸).

4 位置情報を考慮した確率的接続の適用

本研究ではシミュレータを用いた実験により，提案手法と従来のランダム ID との比較評価を行う．シミュレータは事前に作成したシナリオに基いて動作する．

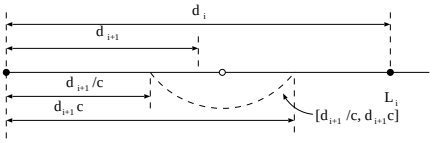

図 3 ロングレンジリンクの接続範囲

4.1 シナリオの作成

シナリオには総ノード数 N_{max}，nodeID，time，action が書かれている．action には join(参加)，leave(離脱)，put(データ格納)，get(データ取出し) がある．ノードの action のタイミングは Kato らが [2] で定義した $lifemean$(参加から離脱までの時間)，$deathmean$(離脱から参加までの時間)，$putinterval$(put を行う間隔)，$getinterval$(get を行う間隔)，$putmax$(ノードの参加後，put を行う回数) を用いる．$\frac{lifemean \times deathmean}{lifemean + deathmean}$ により，およそのシステムの半減期が分かる．ノードの参加から離脱までの挙動を以下に示す.

1. join によりオーバレイネットワークへ参加
2. $putinterval$ 毎にデータを DHT へ $putmax$ 回 put
3. $getinterval$ 毎にデータを DHT から get
4. $lifemean$ 時間経過したら leave によりオーバレイネットワークから離脱

4.2 シミュレータの実装

本研究のシミュレータは Java1.6 UPDATE 10 により作成した．シミュレータのパラメータにはショートレンジリンク数 N_S，接続密度の最適値 λ_{opt}，接続密度の幅 $\frac{\triangle\lambda}{\lambda}$，システムの半減期でのショートレンジメンテナンス頻度 M がある.

4.2.1 リンクの生成

時計回り・反時計回りのリンクとも生成方法は同じであるので，ここでは反時計回りの場合のみを述べる．リンクは双方向接続なので，リンクを生成されたノードはリンクを生成してきたノードをリンクに加える.

図 4 に $N_S = 2$ の場合のロングレンジリンクとショートレンジリンクの生成を示す．黒丸はロングレンジリンクを生成したノードで，斜線の丸はショートレンジリンクを生成したノードである．まず範囲 L_0range を決定し，範囲内にある参加ノードの 1 つとロングレンジリンク L_0 を生成する．次に範囲 L_1range を決定し，範囲内にある参加ノードの 1 つとロングレンジリンク L_1 を生成するということを繰り返して行き，ロングレンジリンク L_N まで生成する．範囲 $L_{N+1}range$ を決定し，ロングレンジリンクを生成しようとしたノードとの距離が距離 d_{min} 以下

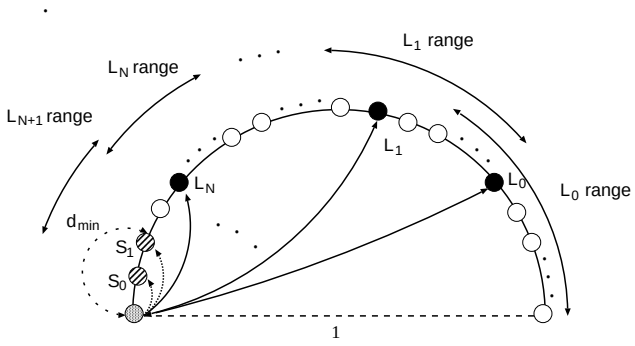

図 4 ノードのリンク生成

になった時にロングレンジリンクの生成を終了する．ロングレンジリンクの生成が終わるとショートレンジリンクの生成が始まる．ショートレンジリンクは自身に最も近いノードから N_S 個のノードとリンクを生成する．

ノードがオーバレイネットワークから離脱するときには，他のノードへの通知を行わない．これはノードが必ずしも正規の手続きを取って終了する場合ばかりではないからである．全てのノードが何の通知もなしに離脱するので，最悪のケースといえる．

4.2.2 メンテナンス

ロングレンジリンク，ショートレンジリンクは他ノードが接続した時に増加する．また get や put を行った際に，ルーティング先として選択したノードが無い場合，そのノードをリンクから削除するので減少する．

ロングレンジリンクメンテナンスは $-\frac{N_L}{\ln d_{min}}$ で定義される接続密度 λ が範囲 $[\lambda_{min}, \lambda_{max}]$ を超えた時に行われる (N_L は時計回り・反時計回りそれぞれのロングレンジリンク数)．λ_{min} と λ_{max} はそれぞれ $\lambda_{opt} - \Delta\lambda_{opt}$, $\lambda_{opt} + \Delta\lambda_{opt}$ で計算される．λ が λ_{max} を超えた場合，$\lambda < \lambda_{max}$ となるまでロングレンジリンクをランダムに削除する．λ が λ_{min} 未満になった場合，$\lambda_{opt} \leq \lambda$ となるまで $-\frac{1}{x \ln d_{min}}$ の確率でリンクを生成する (x はノードとの距離)．

ショートレンジリンクメンテナンスはショートレンジメンテナンス頻度 M の間隔で時計回り・反時計回りとも，同時に行われる．このメンテナンスは最も近いノードから N_S 個のノードをショートレンジリンクに登録する．

各メンテナンスとも，ノードを削除した場合は削除したことを削除先のノードに通知し，削除されたノードは通知してきたノードをロングレンジリンクから削除する．

5 シミュレータを用いた位置情報の有無による性能評価

ここでは提案手法で ID をつけた場合と従来のランダムに ID をつけた場合での性能を比較し，評価する．本研究で使用するシナリオのパラメータのデフォルト値を表 2 に，シミュレータで使用するパラメータのデフォルト値を表 3 に示す．ノードは 512×512 のマップに分散させた．

表 2 シナリオのデフォルト値

$lifemean$	3600
$deathmean$	1200
$putmax$	10
L	10

表 3 シミュレータのデフォルト値

N_S	5
λ_{opt}	3.0
$\frac{\Delta\lambda_{opt}}{\lambda_{opt}}$	0.3
M	20

5.1 ホップ数とノード間の転送距離

提案手法とランダム ID でのホップ数と転送距離を初期ノード数 1,000，5,000，10,000，50,000 の場合において測定した．

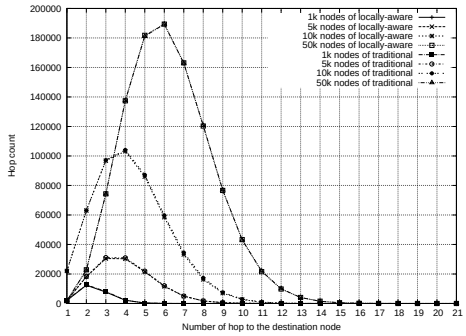

図 5 位置情報の有無によるホップ数の変化

図 5 に初期ノード数におけるホップ数の量を示した．横軸は目的ノードまでに必要なホップ数で，縦軸はシステムの半減期での各ホップの総量である．各初期ノード数において，提案手法とランダム ID のグラフはほとんど一致している．また初期ノード数 10,000 の場合において，ランダム ID とのホップ数の平均誤差は −1.8% となり，位置情報を付加してもホップ数に影響が無いことが分かった．

初期ノード数が 10,000 ノードの場合において最もホップ数が多かった 4 ホップの時の転送距離の平均と標準偏差を図 6 に示す．横軸は何番目のホップかを表し，縦軸はノード間の距離を表している．

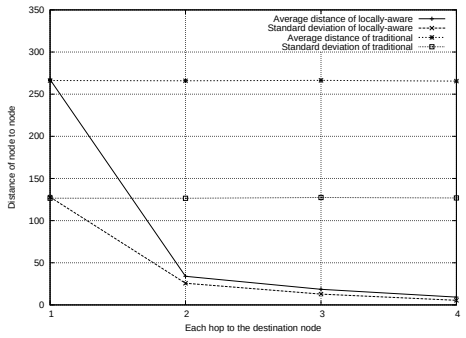

図 6 4 ホップ時における転送距離の平均と標準偏差

ランダム ID ではクエリがホップ毎にマップを縦横断している．標準偏差も高く，ホップごとにばらつきがある．一方提案手法では 2 ホップ目には大きく減少し，目的ノードへ到着するまで転送距離は減少していく．標準偏差も

2 ホップ目以降減少している．これらのことからノード間での転送距離はホップごとに確実に減少していることが分かる．ホップ毎の減少率は 87.3%，45.4%，50.2%となった．またランダム ID に比べ，総転送距離は 69.0%減少した．これらのことから本研究で提案した手法が大きな効果を挙げていることが分かる．

5.2 Churn 環境下での get 成功率

初期ノード数 10,000 におけるシステムの半減期を 1 分，5 分，10 分，15 分，20 分と変化させ，get 成功率を調べた．図 7 にその結果を示す．横軸はシステムの半減期を分で表し，縦軸は get 成功率を示す．

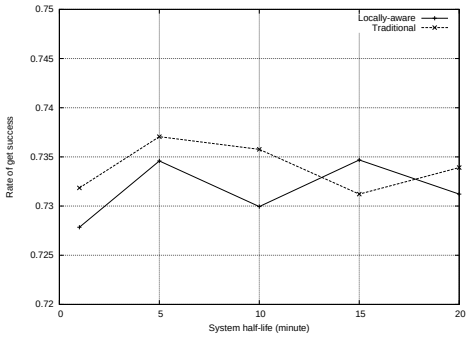

図 7 Churn 環境の変化による get 成功率

システムの半減期が 1 分の時のランダム ID と提案手法の get 成功率はそれぞれ 0.731，0.727 となり，Churn が高い環境下でもデータを DHT から入手できていることが分かる．システムの半減期が 20 分の時の get 成功率はそれぞれ 0.733，0.731 となり，システムの半減期が 1 分と 20 分の差がそれぞれ 0.002，0.004 となった．ランダム ID の場合との差が 0.002 であることを考えると提案手法が先行研究の特徴を十分に生かしているといえる．また，多くの場合においてランダム ID のほうが get 成功率が高くなった．これは物理トポロジ上へランダムにノードが分散されていたとしても，ランドマーククラスタリングにより再マッピングした際に偏りが出てしまうため，あるノードが離脱した時点で多くのデータが DHT 上からなくなってしまうためだと考えられる．

5.3 ノード位置に偏りがある場合のデータ保持量

ノードの物理トポロジに偏りがある場合のデータ保持量の比較を行った．ノードの分布は Zipf 分布に従い，中心点を 0 から 5 まで増やした．図 8 に各ノードのデータ保持量の標準偏差を示す．横軸は中心点の個数を表し，縦軸は標準偏差を表している．

ランダム ID では中心点の個数にかかわらず，標準偏差は $[8.398, 8.485]$ となった．これはノード ID が物理トポロジに左右されず，データが均一に分散するためである．一方，提案手法では中心点が 1 つの時に標準偏差が 24.482 となった．中心点が 0 の場合はノードをランダムに分散させているのでランダム ID とほぼ同じとなった．また，中心点が増えると標準偏差が下がっていくのはノードが分散されていくためであると思われる．

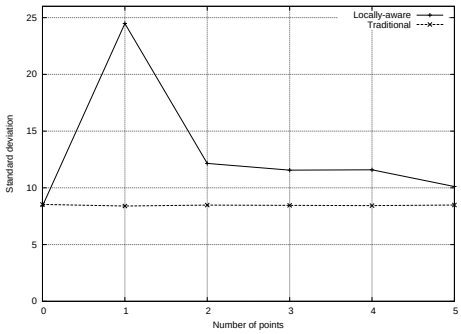

図 8 データ保持量の標準偏差

ノードの物理トポロジに最も偏りがある中心点が 1 つの場合を見ると，提案手法とランダム ID の最大データ保持量はそれぞれ 219，1,571 であった．また 1,000 を越えるデータを保持していたノードが 3 ノード現れるなど，少数のノードが大量のデータを保持していた．提案手法では物理トポロジ上の疎の部分が ID 上でも疎になってしまうため，大量のデータを保持するノードが現れてしまう結果となった．

6 まとめ

本研究ではランドマーククラスタリングとヒルベルト曲線を用いることで位置情報を考慮したノード ID を生成するアルゴリズムを提案し，確率的リング構造オーバレイネットワークに適用した．

提案手法により，各クエリの転送距離をホップ毎に大きく減少させることができたが，一部ノードに多くのデータが集中する問題が生じた．この問題は，DHT のデータサイズや各ノードの性能と，ネットワークトラフィックとのトレードオフにより判断する必要がある．

参考文献

[1] K. Gummadi, R. Gummadi, S. Gribble, S. Ratnasamy, I. Stoica, “The Impact of DHT Routing Geometry on Resilience and Proximity,” Proceedings of the ACM SIGCOMM Symposium on Network Architectures and Protocols, pp.381-394, 2003.

[2] D. Kato, T. Kamiya, “Evaluating DHT Implementations in Complex Environments by Network Emulator,” Proceedings of the 6th International Workshop on Peer-to-Peer Systems, pp.97-103, 2007.

[3] P. Kersch, R.Szabo, L. Cheng, K. Jean, A. Galis, “Stochastic Maintenance of Overlays in Structured P2P Systems,” Computer Communications, Vol. 31, No. 3, pp. 603-619, 2008.

[4] D. Liven-Nowell, H. Balakrishnan, D. Karger, “Analysis of the Evolution of Peer-to-Peer Systems,” Proceedings of the Twenty-First Annual Symposium on Principles of Distributed Computing, pp.233-242, 2002.