

P2P ネットワーク上での効率的な類似度検索

鈴木伸和[†] 菅谷至寛[†] 阿曾弘具[†]

膨大な数のコンテンツを有する分散型データベースにおいて、効率的な検索手法が重要となる。その中で、データの類似性を計算し内容の近いものを自動的に得ることができる検索システムは必要な情報を取得する上で有効である。そこで、本稿では P2P ネットワークを用いた類似度検索システムを提案する。まず、非構造化オーバーレイネットワークを自己組織化していくことにより、検索効率および検索精度が向上することを示す。さらに、構造化オーバーレイネットワークを利用した類似度検索システムについて検討を行う。

Efficient Similarity Retrieval on Peer-to-Peer Network

NOBUKAZU SUZUKI,[†] YOSHIHIRO SUGAYA[†] and HIROTOMO ASO[†]

Efficient retrieval is strongly desired in a huge distributed database environment. A retrieval system in which we can acquire similar data with query data are effective and efficient to obtain beneficial information. In this paper, we propose two similarity retrieval systems on peer-to-peer systems. First, we describe the efficiency and accuracy are improved by using self-organization on unstructured overlay network. Second, we consider a data distribution method on structured overlay network.

1. はじめに

インターネットの普及により、様々な利用者が情報を発信するようになり、Web 上に存在する情報量は増加の一途を辿っている。膨大な情報の中から有益な情報を見つけ出すために、インターネットを利用した情報検索システムや情報共有システムが注目されている。本研究では、データセット全体からあるデータに類似したデータを探すという類似度検索に着目し、P2P ネットワーク技術を利用したデータベース上での類似度検索システムについて検討を行った。

P2P システムが構築するオーバーレイネットワークは、非構造化オーバーレイネットワークと構造化オーバーレイネットワークに大別される。非構造化オーバーレイネットワークは、コンテンツ配置や検索を柔軟に行うことが可能であり、ブロードキャスト型の P2P クエリルーティングによる探索を行う。その探索では、近傍にある探索対象コンテンツを効率よく発見できる一方で、クエリのホップ数を制限しているため、目的のコンテンツがネットワークに存在しても発見できない場合がある。そこで、検索効率を向上させるために、

ネットワーク近傍の情報を各々のノードが収集し、近傍で行われた検索の結果をキャッシュする手法が提案されている¹⁾。

一方、構造化オーバーレイネットワークはノード間の隣接関係に数学的規則があるネットワークである。代表的な構造化オーバーレイネットワークとして分散ハッシュテーブル (DHT) が挙げられる。DHT には CAN²⁾, Chord³⁾, Pastry⁴⁾, Tapestry⁵⁾, Kademlia⁶⁾ などがある。DHT では検索対象となるものにキー (例えばファイル名) を定義し、ある特定のハッシュ関数を用いてキーのハッシュ値から配置するノードを決定する。あるキーに対する検索を行う際は同一のハッシュ関数を用いてキーのハッシュ値から、通信するノードを決定する。DHT での検索効率はノード数 N に対して $O(\log N)$ となる。DHT を用いない構造化オーバーレイネットワークには、Skip Graph⁷⁾ がある。Skip Graph では、ノードの参加・離脱、検索などにかかるコストが $O(\log N)$ と DHT と同等の性質を持つ。Skip Graph はキーとノードが 1 対 1 に対応しているが、単一ノードが複数のキーを保持可能であるような拡張された Skip Graph が複数提案されている¹¹⁾。

非構造化オーバーレイネットワークと構造化オーバーレイネットワークではそれぞれの特徴が異なるため、各々のオーバーレイネットワークを用いた手法を考案し、比較検討を行う。本稿では、個々のノードの

[†] 東北大学 大学院 工学研究科 電気・通信工学専攻
Department of Electrical and Communication Engineering, Graduate School of Engineering, Tohoku University

判断でネットワークの自己組織化を行う手法を提案する。また、拡張された Skip Graph を用いた類似度検索システムの提案を行う。

2. 分散データベース

類似度検索はデータベースシステムにおける検索方式の一種である。クエリデータを指定し、データベースから類似したデータを自動的に検索する支援システムは、データベースの中から適切な情報を探し出す上で有効な手法である。本研究ではある情報に対してその内容の特徴付けるキーワードをひとつあるいは複数抽出した集合をインデックスデータとよび、インデックスデータを P2P ネットワークによって分散管理するデータベースを対象とする。以降、インデックスデータを簡単にインデックス、あるいはデータと呼ぶ。データ間の類似度を計算し、クエリデータ q に対する類似度が大きいデータから順に k 個列挙する top-k 検索システムを考える。

3. 非構造化 P2P システム

3.1 非構造化オーバーレイネットワーク上での類似度検索

非構造化 P2P ではクエリの伝搬はホップ数の上限 (TTL, Time-To-Live) を設定したフラッディングにより行われる。あるデータ q に対する簡単な top-k 検索は、以下ようになる。

- (1) ローカルデータ内で q に対する類似度の高い順に並べ、 k 番目のデータの類似度を s_q とする。
- (2) クエリ発行ノードは、 q との類似度が s_q 以上のデータを要求するクエリを隣接ノードに送信する。
- (3) クエリを受け取ったノードはローカルデータ内で q との類似度を算出し、上位のデータをクエリ発行ノードに向けて送信する。TTL が 0 に達しない場合は隣接ノードに TTL を 1 減じてクエリを転送する。
- (4) クエリ発行ノードは送られてきた検索結果をマージして、ランキングを再構成する。

クエリはあるデータとの類似度が s_q 以上であるデータを探索するという範囲検索となっている。

3.2 ネットワークの自己組織化

フラッディングを利用した検索方式では、メッセージが届かない限り検索が行われることはないが、ネットワーク負荷の観点から単純に TTL を大きくすることはできない。そこで、検索率向上のためには検索がヒットしそうなノードを近傍に置くという自己組織化手法を提案する。提案手法ではクエリを受け取ったノードが以下のルールにしたがってクエリに対する処理を行う自己組織化アルゴリズムを提案する。

ルール 1 ヒットしたデータが m 個未満のとき、そ

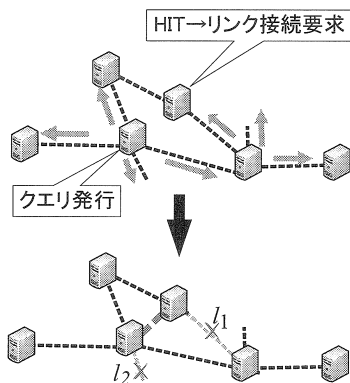


図 1 リンクの張替え

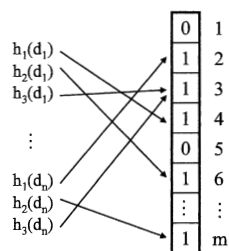


図 2 Bloom filter の作成

これらのデータをクエリ発行ノードに送信する。

ルール 2 m 個以上のとき、データの送信は行わず、検索結果のサマリを作成しクエリが経由してきたノードに向かって送信する。その後、クエリが経由してきたリンクを切断し (図 1 中の l_1)、クエリ発行ノードとリンクを接続する。クエリ発行ノードは、リンク数が最大接続数に達している場合既に確立しているリンクをひとつ切断する (図 1 中の l_2)。

ルール 2 の条件を満たすノードが発見されたときのリンクの張替えの様子を図 1 に示す。リンクを切断するときには、ローカルデータとサマリとの距離が最も大きいリンクを切断する。これによって、類似したデータを持つノードが近くに再配置される。

3.3 Bloom filter を用いた検索結果のサマリ

クエリの転送をより効率よく行うために、検索結果のサマリをクエリが辿った経路上に残し、クエリ転送を制御する機能を持たせる。

まずは、経路上でヒットしたデータに含まれている全キーワードを Bloom filter⁸⁾ に格納する。Bloom filter は固定長 m のビット列であり、初期状態は全てのビットが 0 である。 $[1, m]$ の値を返す k_h 個のハッシュ関数によって格納したいキー d_i のハッシュ値 $h_1(d_i), \dots, h_{k_h}(d_i)$ を求め、指定されたビットに 1

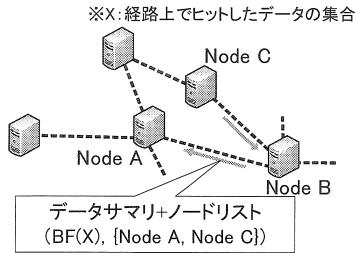


図3 検索履歴の作成と共有

を立てる。図2は $k_h = 3$ の場合のキーワードの集合 $d = \{d_1, \dots, d_n\}$ に対する Bloom filter を表している。例えば、図2において $h_1(d_1) = 4$, $h_2(d_1) = 6$, $h_3(d_1) = 3$ であるので、4ビット目、6ビット目、3ビット目をそれぞれ1にする。既に1となっているビットはそのままにする。キーワードの集合 d から作られた Bloom filter を $BF(d)$ とすると、あるキーワード key_q が d に格納されているかどうかをチェックしたいときは、格納に用いた k_h 個のハッシュ関数にキーを通して指定されるビットが全て1であるかをチェックすればよい。 key_q のみが格納された Bloom filter ($BF(key_q)$) と $BF(d)$ との AND をとったものが $BF(key_q)$ と同一のものかをチェックすることでも同様のチェックを行える。そのチェックが通った場合、そのキーを保持している可能性があるとなり、通らない場合はキーは確実にないことが分かる。

検索結果のサマリはフラッディング終了時、経路上でルール2の条件を満たすノードが存在したときに作成する。検索結果のサマリは経路上でヒットしたデータ全ての情報を格納した Bloom filter と、クエリ発行ノードおよびルール2の条件を満たしたノードのアドレスのリストで構成される。図3において、 $k=3$ の場合を例に説明する。クエリの辿る経路がノードA, B, Cとし、クエリにヒットしたデータがそれぞれのノードで3個、1個、4個だったとすると、これらの合計8個のデータを格納する Bloom filter を作り、これにヒットしたデータが k 以上であったノードAとCのアドレスを付加したものをこの経路の検索結果サマリとする。検索結果サマリの複製は経路上の各ピアが保存し、過去の履歴を共有することになる。

3.4 検索履歴を用いたフラッディングの制御

検索結果のサマリが蓄積したノードでは、フラッディングによるクエリメッセージの伝搬を行う前に、サマリの中から類似したデータに関する検索結果が既にあるかを探す。発見した場合は図4の上の図のように、隣接ノードにクエリを転送せずサマリにあるアドレスにクエリメッセージの転送を試みる。こうして発見したノードは別の検索を行うときにもヒットするデータを持っている可能性が高いと考え、リンクの接続を要求する(図4の l_3)。類似した検索を発見できなかつ

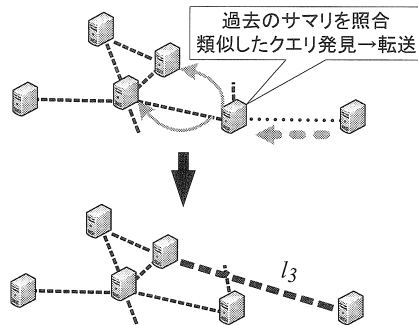


図4 データサマリを用いたクエリの転送

た場合には通常どおりフラッディングにより転送する。

3.5 実験

シミュレーションにより提案した自己組織化アルゴリズムの有効性を確認する。データがランダムに配置された初期状態から検索を繰り返して自己組織化を行ったときの検索精度の変化を評価する。データセットには4741個の所有書籍リストを用い、データ間の類似度は、Dice係数を用いた。データ I, I' に現れるキーワード数をそれぞれ $|I|, |I'|$ とし、 I と I' のどちらにも現れるキーワード数を $|I \cap I'|$ としたとき、Dice係数は以下の式で表される。

$$\text{Dice coefficient} = \frac{2|I \cap I'|}{|I| + |I'|}$$

データ全体をクラスタリングにより10のカテゴリに分類し、各ノードにはデータカテゴリをランダムにひとつ割り当て、そのカテゴリから10のデータを選んでクエリ候補とする。ノードはこのクエリ候補データに対する検索のみを行う。Ravasz, Barabasiらの手法⁹⁾¹⁰⁾によるスケールフリーネットワーク性を有するトポロジを構築し、自己組織化をする前のネットワーク初期状態とした。ノード数 N を256, 512のそれぞれの場合でシミュレーションを行った。シミュレーション実験におけるパラメータは表1の通りである。提案手法によって構築されたP2Pネットワーク上での一回の検索における再現率 (accuracy) とクエリを受け取ったノード数 (number of visit node) を図5に示す。提案手法により、一回の検索でクエリ処理を行うノード数が減少しながらも、時間が経過し自己組織化が進むにつれて検索精度が向上していることが確認された。

4. 構造化P2Pシステム

非構造化オーバーレイネットワークは柔軟な検索が可能である一方で、検索効率やスケラビリティ上の問題がある。そこで、top-k検索を行うP2P検索システムに、構造化オーバーレイネットワークを用いた場合の検討も行った。構造化オーバーレイネットワーク

表 1 実験パラメータ

ネットワーク	
ノード数 N	256,512
各ノードの最大隣接ノード数	2-10
TTL	5
インデックス数	4741
Bloom filter	
ビット数	1000
ハッシュ関数の数	3
検索で求めたいデータ数	k=10
検索間隔	1sec
検索回数	1000
検索履歴の有効期限	100sec

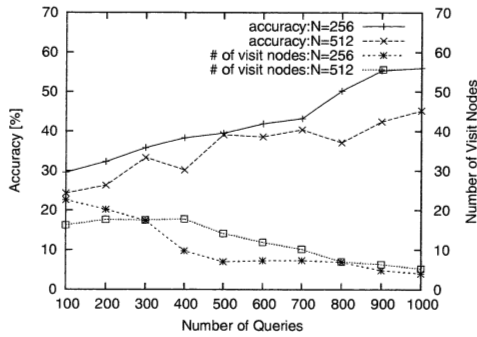


図 5 検索精度と訪問ノード数

では、ノード間のトポロジに一定の構造を規定することにより高いスケラビリティが得られ、効率的な資源の探索やトポロジの構築およびルーティングが可能である。

4.1 キーワードに基づくインデックス分散

本研究のようにインデックスを分散させる場合、インデックス内のキーワードをキーとして DHT に分散させる手法が考えられる。インデックス内の各キーワードからハッシュ値を計算し、それらのハッシュ値に対応する全てのノードにインデックスの複製を管理させる。あるインデックスに対する類似度検索を行う際には、そのインデックスに含まれる全てのキーワードについて、対応するノードにクエリを転送する必要がある。そのため、重要なキーワードを求めてそれらに対応するノードにのみクエリを転送する、またはキーワードの相関に応じてインデックスの複製の保存先を調整する、といった方法を用いることによって、キーワード数の多いインデックスほど増大するルーティングコストを削減する必要がある。

4.2 データの多値化を用いたインデックス分散

そこで、インデックスに含まれるキーワード数にルーティングコストが依存しないよう、インデックスデータを多次元 2 値ベクトルとして表現し、多次元データの局所性を扱える構造化オーバーレイネットワークを

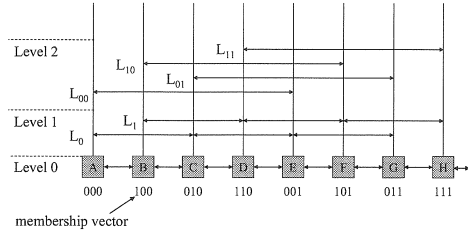


図 6 Skip Graph の構造

用いる手法を提案する。各々のキーワードの有無を 1, 0 に対応させて並べることで、インデックスデータは多次元 2 値ベクトルとすることができる。

多次元データをデータの局所性を保持したままデータをマッピングする構造化オーバーレイネットワークには、ZNet⁽¹¹⁾がある。ZNet では多次元データを空間充填曲線である Z-ordering によって 1 次元データ (Z 値) に変換する。各次元が 0 から $2^r - 1$ の整数値をとる多次元データ (x_1, x_2, \dots, x_d) が与えられ、次元 i の値 x_i を $x_{i,r}x_{i,r-1}\dots x_{i,1}$ と 2 進数で表現すると、Z 値は以下のビットインターリーブ処理によって求まる。

$$\begin{aligned} & \text{interleave}^{d,r}(x_{1,r}\dots x_{1,1}, x_{2,r}\dots x_{2,1}, \dots, x_{d,r}\dots x_{d,1}) \\ & = x_{1,r}x_{2,r}\dots x_{d,r}x_{1,r-1}\dots x_{d,r-1}\dots x_{1,1}\dots x_{d,1} \end{aligned}$$

Z-ordering を用いた変換によって、多次元データは局所性を保ったまま 1 次元数直線上に配置される。ZNet では、通常の Skip Graph と同様にネットワークの参加時に割り当てられた membership vector に基づいたオーバーレイネットワークを構築する。Skip Graph の構造を図 6 に示す。ZNet は Skip-Graph を拡張したもので、複数のキーを保持できるように各ノードに Z-ordering 上で連続した領域が割り当てられる。これを Z Range と呼び、各ノードの Z Range は割り当てられた領域の Z 値の最小値 zMin 及び最大値 zMax を用いて $[zMin, zMax]$ と表す。レベル 0 で隣接するノードは、管理する Z Range も連続している。各ノードは表 2 のようなルーティングテーブルを管理する。ルーティングテーブルには各レベルの隣接ノードと隣接ノードが管理する Z Range を格納する。多次元データに関する範囲検索は、多次元空間上で矩形領域を指定し、この領域に対応する Z 値を求める。このとき、複数の Z Range に分割されるが、それぞれの Z Range を管理するノードをルーティングによって探索する。

ZNet を用いる目的は、できるだけ近いホップ数で高い検索精度を得ることにある。先に述べたように、インデックスは全ての存在するキーワードに対して含まれる・含まれないを示した 2 値の高次元データと考えることができる。しかし、ビット列で表されるインデックスをそのまま ZNet の 1 次元数直線上にマッピングしても類似したデータが近い距離にあるといった関係性は得られない。ZNet を用いて、データの局

表 2 ルーティングテーブル (ノード D の例)

Level	Neighbor Node	Z Range
2	left	-
	right	H [192,255]
1	left	B [128,159]
	right	F [172,191]
0	left	C [160,163]
	right	E [168,171]

所性がオーバーレイネットワーク上に反映されたインデックスデータベースを構築するには、含まれているキーワードが重複しているほど、またインデックスのサイズが近いほど、データ空間で近くに配置されるようなデータ変換が必要である。多値データに変換する手法として、Web ページの検索に用いられる tf-idf 法があるが、データベースの全文書の情報を用いてキーワードに重み付けを行う手法であるため、各ノードがネットワーク全体の情報を、取得する必要あるいは近隣ノードの情報を用いて推定する必要がある。

本稿では、全体情報を必要としない l 次元の多値データに変換することを考える。3.3 で述べた Bloom filter を各インデックスに対して作成し、Bloom filter のビット列を l 個のグループに分割する。 i 番目のグループで 1 が立っているビットをカウントし、その数を i 次元目の値とする。図 7 では、 $l=4$ とし、Bloom filter の j 桁目のビットを $j \equiv i(\text{mod } l)$ を満たすグループ i にカウントしている。

この変換方法によって、各インデックスがどのように ZNet 上に分散されるかを調べた。 $l=16$ とし、3.5 で作成したインデックスデータの bloom filter から多値データに変換する。これらの多値データを Z-ordering によって 1 次元化する。ZNet は負荷分散が保たれている状態を想定し、インデックスを分配する際にノードが保持するインデックス数が均等となるよう Z-ordering を分割した。この状態で、データ間の類似度とそれらのデータを管理しているノードのレベル 0 上でのホップ数の関係を求めた。類似度をいくつかの区間に区切り、平均ホップ数とその分散を図 8 に示す。類似度検索において取得したい情報は類似度の大きいデータであるが、類似度の大きいデータほどネットワーク上で互いに近いノードに配置されていることが分かる。

5. まとめ

本稿では構造化・非構造化 P2P ネットワークを用いた類似度検索システムについて提案した。自己組織化を用いた非構造化類似度検索システムのシミュレーションを行い、その性能を確認した。

今後、構造化型類似度検索システムの構築を考えている。キーワードに基づくシステムに対してはルーティングコスト削減手法の検討を行う。また、多次元

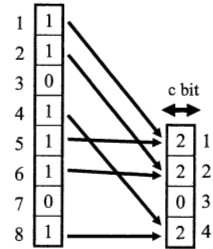


図 7 Bloom filter の多値データ化

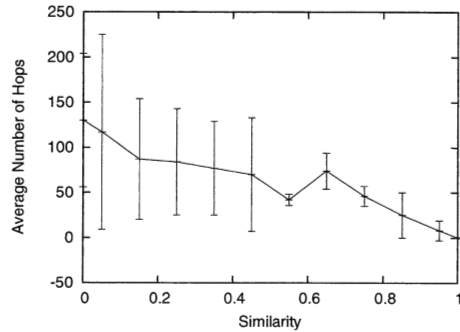


図 8 類似度とレベル 0 での平均ホップ数

データを用いるシステムに対しては、近隣情報を用いた全体情報の推定し、インデックスの多次元データを動的に最適化するための手法に対する検討を行う。さらに、様々な評価実験を行い、効果的かつ効率的なシステムの構築を目指したい。

参考文献

- 1) Bo Yang, A. R. Hurson, Yu Jiao, T. E. Potok, "Multimedia Correlation Analysis in Unstructured Peer-to-Peer Networks," Proceedings of WoWMoM'06.
- 2) S. Ratsanamy, P. Francis, M. Handley, and R. Karp, "A Scalable Content-Addressable Netwrk," Proceedings of ACM SIGCOMM Conference, pp. 161-172, 2001.
- 3) I. Stoica, R. Morris, D. Karger, M. F. Kaashoek and H. Balakrishnan : "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications," Proceedings of ACM SIGCOMM 2001, pp.149-160, Aug. 2001
- 4) A. Rowstron and P. Druschel : "Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems," Proceedings of IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001), pp.329-350, Nov. 2001
- 5) B. Y. Zhao, J. Kubiatowicz and A. D. Joseph :

- “Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing,” Technical Report UCB//CSD-01-1141, University of California, Berkeley, Apr. 2001
- 6) P. Maymounkov and D. Mazieres, “Kademlia: A peer-to-peer information system based on the XOR metric,” Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02), vol. 258, p. 263, Mar. 2002.
 - 7) J. Aspnes and G. Shah, “Skip Graphs,” Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '03), pp. 384-393, Society for Industrial and Applied Mathematics, 2003.
 - 8) B. Bloom, “Space/time trade-offs in hash coding with allowable errors,” *Comm. of ACM*, vol. 13, no. 7, pp. 422-426, July 1970.
 - 9) E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A.-L. Barabasi, “Hierarchical Organization of Modularity in Metabolic Networks,” *Science*, vol. 297, no. 5586, pp. 1551-1555, August 2002.
 - 10) E. Ravasz and A.-L. Brabasi, “Hierarchical Organization in Complex Networks,” *Physical Review E*, vol. 67, p. 026112, 2003.
 - 11) Y. Shu, B. C. Ooi, K.-L. Tan, and a. Zhou, “Supporting multi-dimensional range queries in peerto-peer systems,” Proceedings of the Fifth IEEE International Conference on Peer-to-Peer Computing (P2P'05), vol. 00, pp. 173-180, IEEE Computer Society, 31 Aug.-2 Sept. 2005.