6ZK-4

情報爆発時代における P2P 情報検索向きデータ配置手法

Data Allocation Scheme for P2P IR in Information Explosion Era

倉沢 央† 高須 淳宏‡ 安達 淳‡

Hisashi Kurasawa Atsuhiro Takasu Jun Adachi

東京大学 大学院 / The University of Tokyo†

国立情報学研究所 / National Institute of Informatics‡

1. はじめに

情報爆発時代と呼ばれる昨今，検索システムは日常生活において重要なツールの 1 つとなった．従来，検索システムは集中型のアーキテクチャで構築されることが主流であった．この集中型のアーキテクチャは容易に構築でき，さらに情報を集中管理できる．しかし，スケーラビリティ，検索対象の網羅性，管理・維持コストといった問題点を内在している．既存の大規模検索サービスではこれらの問題点に対して，高度な専門知識や膨大なコンピューティング資源，豊富な資本力で解決を図っている．一方で，収益性の少ないサービスや一般ユーザが主体となっているサービスではこれらの解決法は難しい．

このような問題に対して，大規模なシステムに向いている分散型アーキテクチャで検索システムを構築することで解決を試みた研究が近年盛んになりつつある．Peer-to-Peer(P2P)ネットワークを用いることで，参加するユーザ(ノード)が多ければ多いほど大規模な検索システムの運営が可能となり，膨大な情報を P2P 情報検索システム(P2P IR)で整理することが出来る．P2P IR は大規模システムへの対応だけでなく，余剰コンピューティング資源の有効活用の点でも期待されている．これまでの研究で，分散型検索の検索精度を集中型とほぼ同等にする手法が提案されている[2]．しかしながら，分散型検索システムにおける索引構築コスト[3] や検索実行時のコストが未だボトルネックとなっており，普及の障壁となっている．

我々は P2P IR における検索実行時のコストを削減するためのデータ配置法，Concordia-1 を提案した[1]．Concordia-1 は文書データの配置場所を，検索時に索引参照のために接続するノードと関連づけ，文書における重みの大きな単語の索引を管理するノードに文書データを配置する．実験により，問い合わせに適合する文書ほど収集が容易となり，検索の実行時間を削減できることを確認した．本稿では Concordia-1 の評価実験の結果とその考察について報告する．

2. Concordia-1 のアーキテクチャ

Concordia-1 の特徴は以下のとおりである．

- Concordia-1 では DHT を用いて文書ごとに単語を索引に登録することで，検索問い合わせに対する各文書との適合度を算出可能である．つまり，集中型検索システムとほぼ同様の精度の検索結果を実現している．
- Concordia-1 では文書中の単語の重みに基づいて文書データを配置するノードを決定することで，検索問い合わせに適合する文書ほど効率良く収集可能である．

2.1. 索引構造と文書情報の登録

Concordia-1 では，ノードは DHT 上に索引を作成する．この DHT 上の索引は各単語の索引の集合体である．各単語の索引は，単語のハッシュ値に最も近いノードが管理する．索引には 1 つの文書につき，文書名，文書における単語の重み，文書における重みの大きな上位 n 単語のハッシュ値，そして文書のデータへのポインタの合計 4 つの項目が登録される．文書における単語の重みの計算には，Okapi で採用された BM25 と呼ばれる確率モデルを基にした式を用いた．文書 *d* における単語 *t* の重みは，

$$w(t,d) = \log\frac{N}{df}\cdot\frac{(k+1)\cdot tf}{k\cdot\{(1-\alpha)+\alpha\cdot dl/avdl\}+tf}$$

のように定義する．ここで，*w(t,d)*は文書 *d* における単語 *t* の重み，*k* と *α* は定数，*N* は文書コレクションに含まれる文書の総数，*tf* は *d* における t の出現頻度，*df* は全文書における *t* を含む文書数，*dl* は *d* の長さ，*avdl* は文書の長さの平均値を表す．

式で用いる変数のうち，ノードのみが保持する情報だけでは *df* と *N*， *avdl* の取得が難しい．*df* に関しては DHT 上に構築した索引をあらかじめ参照することで解決する．他の値については予備実験にて検索精度に与える影響が小さいことがわかったので，本研究では簡略化して定数を設定することにした．

2.2. 単語の重みに基づいたデータ配置

P2P IR では，検索問い合わせに適合する文書を探すだけでなく，効率良く適合する文書を収集できることも重要である．既存手法では，文書を収集するたびに文書のデータを管理するノードのアドレスを ID を元に探す必要がある．そのため，収集する文書数が多ければ多いほど多くのノードのアドレスを探す手間がかかり，検索応答に時間がかかってしまう．

そこで Concordia-1 では，文書のデータを検索索引と関連づけることで，適合文書の収集時間の短縮を図る．ノードのアドレスを探すコストを削減するため，Concordia-1 では検索時に必ず接続されるノード，つまり，検索質問に含まれる単語の索引を管理するノードで，適合度の算出と文書の収集を完結できるようなデータの配置を行う．具体的には，文書を索引へ登録するのと同時に，ノードは文書における重みの大きな上位 *n* 単語のハッシュ値に該当するノードに文書のデータの複製を配置する．文書の検索問い合わせとの適合度は，問い合わせに含まれる単語の文書における重みの和を用いる．それゆえ，文書における単語の重みに基づいて，データを単語の索引を管理するノードに配置することで，問い合わせとの適合度の高い文書ほど収集が容易になる．Concordia-1 における索引付けとデータ配置を示したものが図1である．

2.3. 検索とデータ収集

ユーザが問い合わせに適合する文書を発見・収集するには，まず検索問い合わせから単語を抽出する．次に，その単語の索引を保持するノードに接続し，索引を参照する．そして,索引に登録されている文書それぞれの検索問い合わせとの適合度を求め，適合度の高い文書を収集する．その際，文書のデータは主に索引を保持するノードから収集する．もし索引を保持するノードに文書のデータが存在しない場合は，索引に登録されたその文書における重要単語を参照し，その単語の索引を担当する

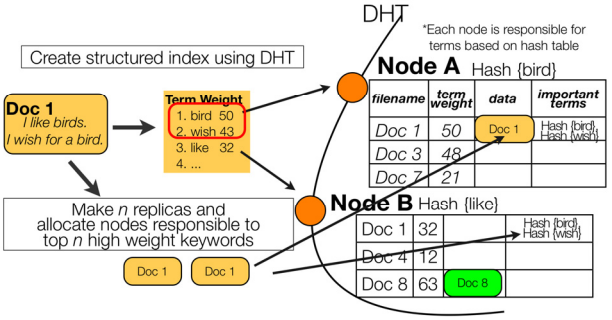

図 1 Concordia-1 における索引付けとデータ配置

3. 評価

3.1. 適合文書の収集時に接続するノード数

Concordia-1 では各文書について単語の重みを算出し，重みの大きな単語を管理するノードに文書のデータを配置する．この手法に類似したデータ配置法として，各索引ノードにおいて単語の索引に登録された文書を重みでソートし，その上位 *n* 文書を予め収集する手法が考えられる．各文書における単語の重みを用いる Concordia-1，各索引中の重みを用いる手法，そして文書と無関係な場所にデータを配置する手法とで検索時の収集効率を比較する実験を行った．実験には文書コレクションとして TREC の Tipster 3 を，英単語のステミングに Porter stemmer を，検索タスクに TREC-2 ad hoc & TREC-3 routing topics のトピック 50 件を用いた．表 1 は Tipster 3 を各手法において文書の複製を配置したときの複製数を比較したものである．Concordia-1 では各文書における重みの上位 *n* 単語の索引ノードに文書の複製を，比較手法では各索引中の重みの上位 *n* 文書の複製を配置した．単語の索引によってはあまり文書が登録されないものもあるため，比較手法におけるストレージコストの変化は比例になっていない．表 1 から，各単語の索引における重みの上位 5 文書を配置するストレージコストは，Concordia-1 において各文書における上位 4 単語もしくは 5 単語の索引ノードに配置するストレージコストとほぼ同じだとわかる．

図 2 は各単語の索引ノードに配置されるデータ数を比較したものである．比較手法では索引ノードに均一に複製が配置されるのに対して，Concordia-1 では一部の索引ノードへのストレージコストが大きくなってしまうことがわかる．

図 3 は各手法で複製を配置したとき，トピックとの適合度の高い上位文書を取得するのに接続するノード数をシミュレーションした結果である．実験ではネットワーク内に存在するノード数は簡単のため単語数と同一の値に設定した．Concordia-1 のほうが冗長化の度合いが等しいとき効率の良い収集を実現できることがわかった．

以上のことから，各文書における単語の重みを用いた Concordia-1 は，ノードに対してのストレージコストを分散する工夫が必要ではあるが，検索問い合わせとの適合度の高い文書を効率よく収集するには適した配置法であることが確認できた．

3.2. 検索応答時間

Concordia-1 では索引の参照と適合文書の収集を同一ノードで実行することで，検索応答時間を短縮している．筆者らは検索応答時間の実測値を測り，評価を行った． 4 つの PC クラスタの合計 248 ノードで，それぞれ 5 プロセスの Concordia-1 を実行し，1,240 の peer で構成されるネットワークを構築した．マシン間は gigabit Ethernet で接続されていた．実験には3.1と同様のデータを用いた．検索タスクのトピック 50 件を検索し，適合度の上位 *n* 文書を取得するのに必要な時間を測定した．Concordia-1 で配置する複製は 5 つに設定した．単語の重みに無関係なノード 5 つに文書の複製を配置する手法を Baseline とした．

図 4 は Concordia-1 と Baseline の検索応答時間の比較を示したものである．実験結果より，Concordia-1 は検索問い合わせに適合する文書を短時間で収集可能なことがわかった．つまり，文書中の単語の重みに基づいて文書データを配置することで，P2P ネットワークから効率良く適合文書を収集できることがわかった．実験では文書の収集に最低 180 秒必要なことがわかる．これは，検索タスクは平均 62.72 単語で構成される文章であったため,索引の参照時間が大きな影響を及ぼしていると思われる．

表 1 配置法ごとの文書の複製数の比較

	Concordia-1	索引中の重みに基づいた複製配置
Top 1	336,310	817,897
Top 2	672,620	1,075,697
Top 3	1,008,930	1,259,765
Top 4	1,345,240	1,410,761
Top 5	1,681,550	1,542,057

4. おわりに

本稿では Concordia-1 の評価について述べた．P2P IR において単語の重みに基づいてデータを配置することで，検索を効率よく短時間で実行可能なことを確認した．

現在はストレージコストや索引構築コストをノード間で分散できるように DHT 構造から見直し，検討を行っている．

参考文献

[1] H. Kurasawa, H. Wakaki, A. Takasu, and J. Adachi. Data Allocation Scheme Based on Term Weight for P2P Information Retrieval. In *Proceedings of ACM WIDM 2007*, 2007.

[2] M. Bender, S. Michel, P. Triantafillou, G. Weikum, and C. Zimmer. MINERVA: Collaborative P2P Search. In *Proceedings of VLDB '05*, 2005.

[3] G. Skobeltsyn, T. Luu, I. P. Zarko, M. Rajman, and K. Aberer. Web Text Retrieval with a P2P Query-Driven Index. In *Proceedings of SIGIR '07*, 2007.

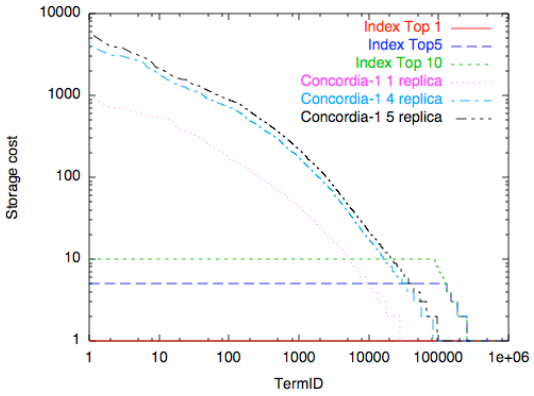

図 2 各単語についてのストレージコストの比較

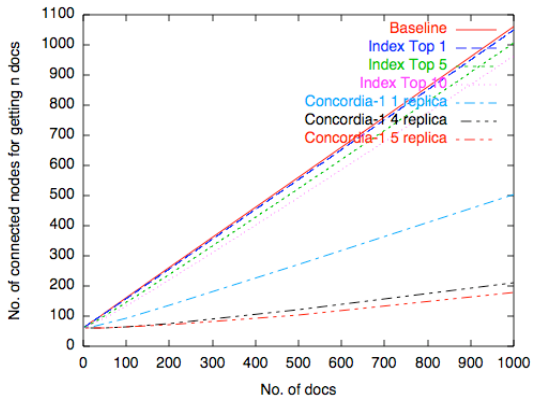

図 3 適合する上位文書の収集時に接続するノード数の比較

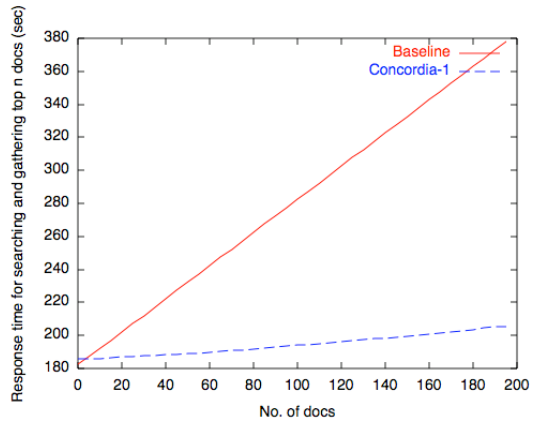

図 4 適合する上位文書の収集時間