

P2P 環境における構造概要を利用した XML データの検索

呉 俊輝[†] 天笠 俊之^{†,††} 北川 博之^{†,††}

[†] 筑波大学システム情報工学研究科 〒 305-8573 茨城県つくば市天王台 1-1-1

^{††} 筑波大学計算科学研究センター 〒 305-8577 茨城県つくば市天王台 1-1-1

E-mail: †chunhuiwu@kde.cs.tsukuba.ac.jp, ††{amagasa,kitagawa}@cs.tsukuba.ac.jp

あらまし P2P ネットワークは、新たな分散コンピューティングパラダイムとして、近年注目されている。中でも、分散ハッシュ表 (DHT) は、中央サーバを持たないピア型 P2P 環境において、効率的な検索手段を提供する手法として広く用いられている。一方、XML は、標準的なデータ交換フォーマットとして普及しており、多くのデータが XML で記述されるようになってきている。本研究では、P2P 環境において、分散して格納された XML データの検索手法を提案する。具体的には、XML データを葉ノード単位に分解し、そのパス情報、ノードラベル、およびテキスト値の組として分散ハッシュ表に格納する。さらに、XML データの構造概要 (Strong DataGuide) を DHT に格納することにより、XPath による問合せを可能にする。

キーワード P2P, XML, XPath, DHT, DataGuide

XML Retrieval Using Structural Summary in Peer-to-Peer Environment

Chunhui WU[†], Toshiyuki AMAGASA^{†,††}, and Hiroyuki KITAGAWA^{†,††}

[†] Graduate School of Systems and Information Engineering, University of Tsukuba
Tennohdai 1-1-1, Tsukuba, Ibaraki, 305-8573 Japan

^{††} Center for Computational Sciences, University of Tsukuba
Tennohdai 1-1-1, Tsukuba, Ibaraki, 305-8577 Japan

E-mail: †chunhuiwu@kde.cs.tsukuba.ac.jp, ††{amagasa,kitagawa}@cs.tsukuba.ac.jp

Abstract Peer-to-Peer (P2P), a new paradigm in the field of distributed computing, has attracted much attention recently. Above all, DHT (Distributed Hash Table) offers an effective and powerful search mechanism without a central server, which could be a potential bottleneck. On the other hand, XML (Extensible Markup Language) has become the de-facto standard for electronic data exchange, and a huge amount of data is now described in XML. In this paper, we propose a novel method to storage and retrieval of distributed XML in P2P networks. In this method, an XML data is decomposed according to text-values in leaf nodes, and they are stored in DHT along with their path information and node labels. Furthermore, in order to realize retrieval of arbitrary XML subtrees, the Strong DataGuide of the XML being stored is also stored in another DHT. We can thus enable XPath queries using the DHTs.

Key words P2P, XML, XPath, DHT, DataGuide

1. はじめに

近年、計算機性能の向上とネットワークインフラの発展により、大規模なデータに対して高速な処理が可能になる分散コンピューティング技術が注目を集めている。中でも、Peer-to-Peer(P2P) 技術は、インターネットを通じて大規模な分散コンピューティング環境が容易に構築できる技術として、近年活発に研究されている。

P2P システムは、従来のサーバ・クライアント型システムと

の対比で、完全な分散システムであるピア型と、両者の中間的な性質を持つハイブリッド型とに分類することができる。ハイブリッド型 P2P システムでは、データの所在を管理する中央サーバ (索引サーバ) の存在を仮定しており、全ての検索処理は索引サーバを通じて行われ、その結果を利用してピア同士が通信を行う。このため、システムはシンプルであるが、索引サーバの故障やネットワークにおける局所的な故障がサービス全体の停止の要因となることがある。一方、ピア型 P2P システムでは、ピア同士がサーバを介さずに動作するため、局所

的な障害があっても全体サービスは持続する。

代表的なピア型 P2P システムとしては、ファイル共有システム Gnutella [1] が挙げられる。ただし、Gnutella ではネットワークに隣接のピアに対して、リクエストを送信する処理が発生し(フラディング)、不要なトラフィックが大量に生じるといった問題が知られている。この問題を解決するため、分散ハッシュ表(Distributed Hash Table; DHT) と呼ばれる、P2P ネットワークでの効率的なオブジェクト配置と通信の枠組みが提案されている。

一方、XML [2] は、1998 年に W3C 勧告となってから現在までの間に急速に普及し、計算機間のデータ交換の標準フォーマットとなっている。XML は、タグを用いて文書やデータの意味・構造を記述するマークアップ言語を記述するためのメタ言語であり、木構造やグラフ構造を記述可能なことから、様々な分野で広く応用されている。最近では、P2P 技術の発展と XML の普及に伴い、こういった XML データを P2P 環境で利用することも一般的に行われるようになってきている。例えば、物理学や医療分野では、それぞれ物理シミュレーションデータのメタデータや電子カルテを XML で記述して、P2P ネットワークを通じて交換するという応用が出始めている。このことから、今後は P2P 環境で大量の XML データを効率良く格納・検索すること重要になってくるものと思われる。

これまで、P2P 環境での XML データの格納、検索の手法はいくつか提案されている。例えば、XP2P [3] では、XML データを予めフラグメント化しておき、P2P 上に分散格納する方式を提案している。その際、フラグメント内に含まれるパス式をパスフィンガープリンティングと呼ばれる特殊な方法でハッシュ値に変換し、DHT に格納する。検索処理では、問合せパス式を同様の手法でハッシュ値に変換し、対応するデータを保持するピアを特定する。ただし、この手法は、XPath における descendant 軸 (//) のような、XML の階層構造を再帰的にたどる必要のある問合せに対して、効率的でないという欠点がある。

そこで本研究では、ピア型 P2P 環境における新たな XML データの格納、検索方式を提案する。具体的には、XML データをテキスト情報を保持するノードと葉ノードに分解し、その要素名とテキスト値の組合せを、DHT に格納する。これを C-DHT(Content-DHT) と呼ぶ。さらに、任意の部分 XML データを検索、再構築するための手がかりとして、Strong DataGuide [4] を構築し、別途 DHT に格納する。これを、S-DHT(Structural Summary-DHT) と呼ぶ。これらを併せて利用することで、問合せ処理に必要なネットワーク参照回数を減らすことができる。

論文の構成は次の通りである。まず 2. で DHT や XML の関連技術について基本的事項を説明する。次に 3. で本研究における提案手法と問合せ処理機構を述べる。4. で提案手法による評価実験について述べる。5. で関連研究を紹介し、最後に 6. でまとめと今後の課題について述べる。

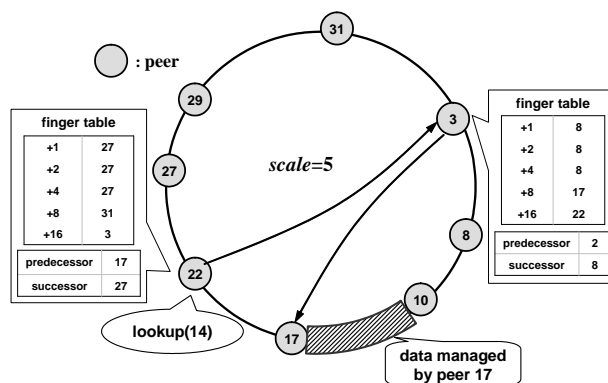


図 1 Chord の ID サークル
Fig. 1 ID Circle of Chord

2. 基本的事項

2.1 分散ハッシュ表 (DHT)

分散ハッシュ表 (DHT) は、ピア型 P2P 環境におけるオブジェクト検索を可能にする手法である。DHT では、P2P に参加するホスト(ピア)と検索対象となるオブジェクトを、仮想的なネットワーク(オーバーレイネットワーク)上に配置する。各ピアやオブジェクトは、ハッシュ関数(例えば、MD5 や SHA-1 など)を用いて、それぞれの識別子(IP アドレス、ファイル名など)からハッシュ値を求め、そのピアやオブジェクトが仮想的なネットワーク上の位置、およびどのピアがどのオブジェクトを管理することを決める。各ピアは、ルーティング表を利用することで、検索したいオブジェクトを管理するピアへメッセージを転送することで、検索が可能になる。DHT には、オーバーレイネットワークやルーティング表の構成法にバリエーションがあり、これまでいくつかの方法が提案されている。代表的なものとして、Chord [5]、CAN [6]、Tapestry [7] などがある。以下では、CAN と Tapestry を簡単に説明し、本研究で用いる Chord についてはより詳しく紹介する。

CAN では、ネットワーク空間を多次元空間とみなし、空間の分割や結合を行うことによってピアやオブジェクトの管理を行う。各ピアはシステムで割り当てられた重複しない部分空間に対応し、その空間内に存在するオブジェクトを管理している。ルーティング情報として、ある次元で接している隣接空間に割り当てられたピア情報を保持している。 N をネットワーク上のオブジェクト数、次元数を d とした場合に、検索に必要なメッセージ数は $O(d \times N^{\frac{1}{d}})$ である。

Tapestry では、ピアの識別 ID に対するプレフィックススペースのルーティング処理を行うことで効率化を図っており、任意の整数パラメータ b (システム設計時に決める値)とした場合に、検索に必要なメッセージ数は $O(\log_b N)$ である。

Chord では、オーバーレイネットワークとして、 N 個のオブジェクトからなる仮想的な円構造のハッシュ空間(ID サークル)を用いる。オブジェクトにはピアとデータが含まれており、すべてのオブジェクトは $scale$ ビットの識別 ID を与えられる。最大で格納できるオブジェクト数は 2^{scale} である。識別 ID が

i のピアは、担当するデータの識別 ID、フィンガーテーブル、前後に位置するピアの ID 情報 (*predecessor*, *successor*) を保持している。 i が担当するデータとは、前ピア (*predecessor*) から i までのハッシュ値の区間が該当する。フィンガーテーブルには、 i から *scale* 個の他ピアへのリンクがあり、リンク先は $i + 2^k$ ($k = 0, 1, 2, \dots, scale - 1$) 位置の担当ピアである。このため、オブジェクトの検索に要するメッセージ数は $O(\log_2 N)$ である。

図 1 にオブジェクト検索の流れを示す。ピア 22 からキー 14 が保持する値を得たいとすると、ピア 22 のフィンガーテーブル内で最もキー 14 に近いのはピア 3 であるため、まずピア 3 にアクセスする。これを繰り返すことで、キー 14 の情報を保持するピア、すなわちキー 14 の *successor* へアクセスする。この例では、キー 14 の *successor* はピア 17 であるため、最終的に、ピア 22 はピア 17 にアクセスし、キー 14 に対応する値を取得する。

2.2 XML データと構造概要

本研究では、XML のデータモデルを $T = (V, E, r)$ とする。ここで V はノード集合、 E は枝集合、 r は XML 木の根ノードであり、 r は V の要素である。本研究では、以下の議論を簡単にするため、属性ノードと要素ノードは区別せずに扱う。図 2 に XML データの例を示す。根ノードの子要素には *book*, *music* ノードがあり、カッコ内にはテキストノードの値を表している。

XML データの検索の効率化のために、XML データための索引構造がよく研究されている。特に、データの概要を把握するためのいわゆる構造概要 (Structural Summary) は、これまでに多くの手法が提案されている。本研究では、その中で最も単純なものの一つである **Strong DataGuide** を利用する。これは、概念的には XML 木の根ノードから葉ノードへ向かって、共通のタグを集約することで得られた木構造である。まず DataGuide の形式的な定義を示す。

定義 DataGuide

情報源 s の DataGuide d は、1) s の全てのラベルパスが d において一つのデータパス・インスタンスを持ち、2) d の全てのラベルパスは s のラベルパスになっているものを言う。ここで、ノード o のラベルパスとは、 o から辿ることのできるエッジ (e_1, e_2, \dots, e_n) のラベルの系列 l_1, l_2, \dots, l_n である。また、ノード o のデータパスとは、 o から辿ることのできるノードとラベルの系列 $l_1, o_1, l_2, o_2, \dots, l_n, o_n$ である。□

d では s の全てのパスを含んでいるものの、 s の複数のノードが d の一つのノードに集約されることから、 d は s を要約していると言うことができる。Strong DataGuide は、任意のパス式のターゲット集合という視点から、 s と d が区別できないものと言う。図 2 に XML データから DataGuide, Strong DataGuide の例を示す。以降 Strong DataGuide を *DG* と呼ぶ。

3. 提案手法

本研究では、DHT に基づく P2P ネットワークに新たな XML データの格納、検索方式を提案する。基本的には、XML デー

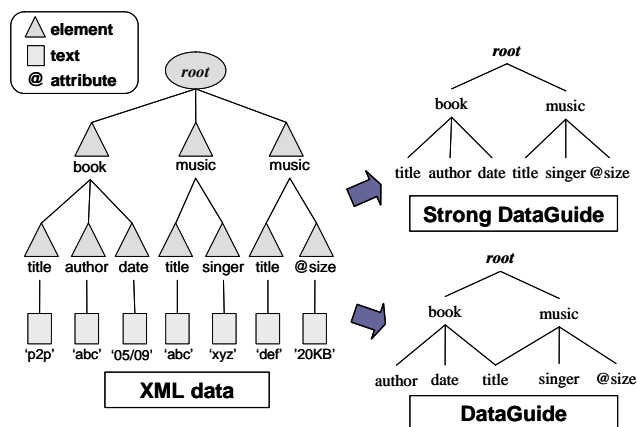


図 2 XML データとその DataGuide, Strong DataGuide
Fig. 2 XML data, DataGuide, and Strong DataGuide

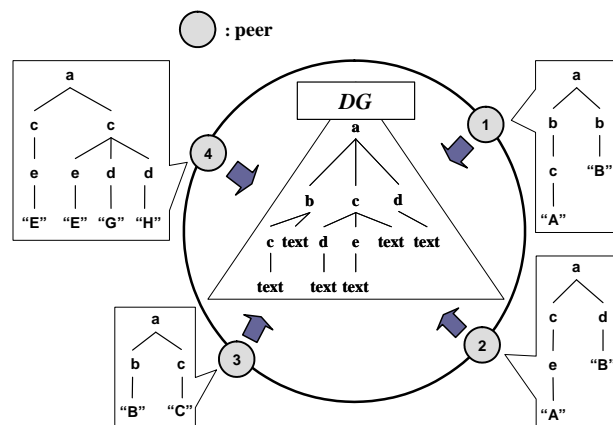


図 3 各ピアが保持しているデータから DG の生成
Fig. 3 Generation of DG from XML

データをノード単位に分割した上で、要素名をキー、テキストを含む関連情報を値として DHT に格納する。しかしながら、これだけでは、中間ノードをルートとする部分 XML データの再構成ができない。このため、本研究では、P2P ネットワーク上のピアが保持している全ての XML データ上に構築された *DG* を手がかりに検索処理を行う。*DG* は、別途専用の DHT に格納され、データの追加や削除があったときに、随時更新される。

本研究の前提として、ネットワーク上の格納モデルには Gilbert 等によって提唱されたソフト状態モデル [8] を採用する。このモデルでは、各ピアがネットワーク上に格納するデータに有効期限を設ける。有効期限内にリフレッシュ操作が行われれば、データはそのままネットワーク内に保持されるが、ネットワーク障害や離脱などによってリフレッシュされない場合、期限切れとともに破棄される。また、P2P ネットワーク上の各ピアが保持する XML データには、グローバルなスキーマを前提とせず、任意の整形 XML データを扱うことができるものとする。

なお、検索には XPath のうち *child* 軸 ($/$)、*descendant* 軸 ($//$) と述語 ($[]$) を含む $XP^{U, //, []}$ [9] を用いる。

3.1 C-DHT による XML データの格納

提案手法では、基本的に XML データをノード単位に分割

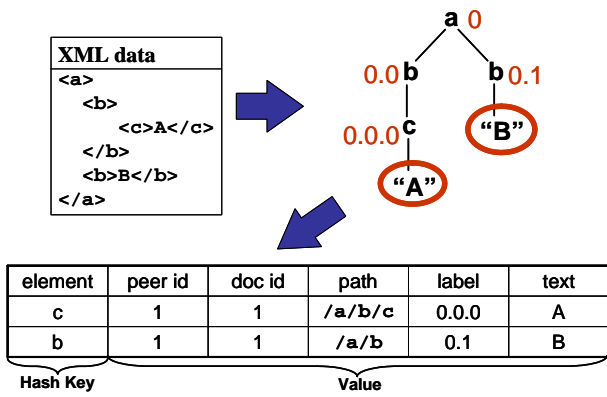


図 4 XML データの C-DHT への格納
Fig. 4 Storing XML in C-DHT

した上で、DHT に格納する。このための DHT を C-DHT (Content-DHT) と呼ぶ。具体的には、XML データのノードのうち、1) テキストノードだけを含む要素ノード、2) 属性ノード、3) テキストノードを含まない葉要素ノードの 3 種類のノードを DHT に格納する。このとき、ハッシュキーは要素名、格納される値は (ピア ID, 文書 ID, パス, 順序ラベル, テキスト) の組とする。要素名をハッシュキーとすることで、要素名を元に、その内容を保持しているピア ID とそのピア内での当該文書 ID, その文書内でのパス式とノードの位置情報 (順序ラベル), テキスト値などを得ることができる。特に、パスと順序ラベルは重要で、これによって後述する述語を含むような XPath 問合せを処理することが可能となる。なお、テキストノードや属性ノードを持たない中間ノードについては検索対象とはせず、DHT には写像しない。これによって、データ件数を低減することができる。

ところで、データの追加が発生した場合、順序ラベルの更新が問題となる。本研究では議論を単純化するために Dewey Order [10] を用いることにしているが、その代わりに ORD-Path [11] を用いれば、順序ラベルの付け直しなしに無制限なノードの挿入を行うことができることを指摘しておく。図 4 に XML データを C-DHT に格納する例を示す。この例は図 3 において、ピア 1 が保持している XML データの中で、葉ノード c, b を格納する例である。ここでは要素名 c, b をキーとして、 $(1, 1, /a/b/c, 0.0.0, A), (1, 1, /a/b, 0.1, B)$ の組を C-DHT に格納する。

3.2 S-DHT による構造概要の格納

S-DHT (Structural Summary-DHT) は、P2P ネットワーク上のピアが保持している全 XML データから構築される構造概要を格納するための DHT である。結果として、現在ネットワーク内に格納されている全ての XML データの要素名、属性名、パス式が格納されているので、検索処理時の索引として用いることができる。具体的には、構造概要の各要素名をキーとし (順序ラベル, 親ノード, 子ノード, 子ノードの順序, 統計値) の組を値とした分散ハッシュ表である。順序ラベルは、C-DHT でも用いた Dewey Order を用い、構造上、複数出現する同名の要素を区別する。子ノードの順序は、親ノードから見

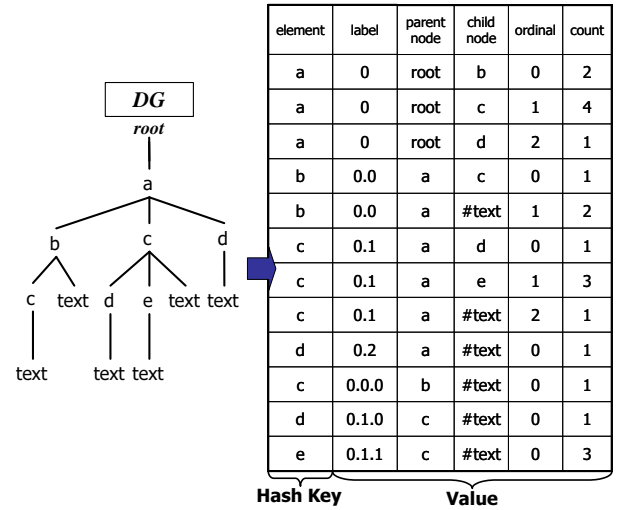


図 5 Strong DataGuide の S-DHT への格納
Fig. 5 Storing Strong DataGuide in S-DHT

た子ノード間の順序を表す。統計値は当該パスがネットワーク上の XML データ中で出現した回数を表しており、問合せ処理の効率化に利用される。図 5 に実際の構造概要を S-DHT に格納する例を示す。 b をカレントノードをすると、 b をキー、 $(0.0, a, c, 0, 1)$ の組を値として S-DHT に格納する。

3.3 問合せ処理

以下では、提案手法での問合せ処理方法について述べる。まず $/, //$ だけを含む単純パスと、述語を含む問合せの場合に分けて説明する。

3.3.1 単純パスの問合せ処理

単純パスの問合せ処理は、与えられた検索パスに該当する XML ノードを C-DHT を使って探索する探索フェーズと、探索フェーズで得られた XML ノードを頂点とする部分木を S-DHT と C-DHT の両方を使って再構築する再構築フェーズからなる。与えられた検索パスを $q=|e_1|e_2|\dots|e_n$ とすると処理手順は以下の通りである。ここで、 $|$ はロケーションステップの分離記号であり、 $/$ または $//$ である。

探索フェーズ

- (1) 検索パスを構文解析し、末端の要素名 e_n を取得する。
- (2) e_n をキーとして C-DHT を参照し、対応するデータを保持しているピアを特定する。
- (3) (ピア ID, 文書 ID, パス, 順序ラベル, テキスト) の組から、パスが q を満たすものを選択し、結果集合 R に加える。ここで、ロケーションステップとして $e_1//e_2$ のように $//$ が含まれる場合は、文字列マッチの正規表現を用いれば、 $e_1/*e_2$ とすることでマッチング処理が可能である。

再構築フェーズ

- (4) e_n をカレント要素名として S-DHT を検索し、 e_n の全ての子要素名 e_m を獲得する。
- (5) e_m を元に C-DHT を検索し、パス名に e_m を含む情報を獲得して R に加える。
- (6) 検索パスとカレント要素名を置き換え、(5) からの処理を続ける。 R のサイズが増えなくなるまで繰り返す。

C-DHT						S-DHT			
element	peer id	doc id	path	label	text	element	parent node	child node	count
...	Discovery : Result R_1					a	root	b	2
b	1	1	/a/b	1.2	"B"	Construct:		...	
b	2	1	/a/b	1.1	"B"	b	a	c	1
Construct:Result R_2					...	b	a	text	2
c	1	1	/a/b/c	1.1.1	"A"	
c	3	1	/a/c	1.2	"C"	...	b	text	1
...	Path and query: not match					...	Construct end		

図 6 問合せ $//b$ の DHT 参照過程
Fig. 6 Querying Processing of $//b$

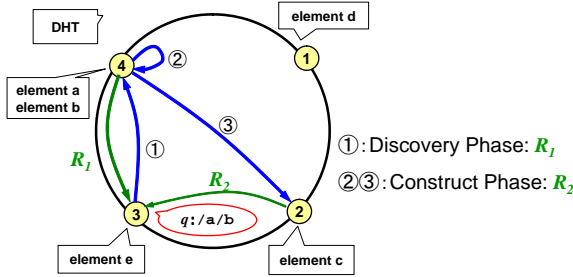


図 7 結果の生成と問合せメッセージの転送
Fig. 7 Result Generation and Messaging

おおまかに言うと、探索フェーズでは、検索パスの末端の要素名 e_n を手がかりに、テキストノードを直接含む情報を探索する。続く再構築フェーズでは、 e_n をルートとする部分 XML データを、S-DHT の情報を手がかりに再構築する。具体的には、S-DHT から、 e_n の子要素である可能性のある要素名を取得し、それをキーとして C-DHT を検索することで、テキスト値を得る。これを部分 XML データの再構築が完了するまで繰り返す。単純な例として、 $//b$ を処理する過程を図 6 に示す。また、結果の生成と処理に必要なピア間の問合せメッセージの転送は図 7 に示す。

容易に推測できるように、このアルゴリズムは検索対象が単純なテキスト値しか含まない場合は極めて効率が良い。反面、結果がある程度の大きさを持つ部分 XML データの場合は効率が悪化する。これに対処する方策として、部分 XML データのサイズがある程度の大きさを持つと見積られる場合、C-DHT の内容から直接にピア ID を得て、データを保持しているピアから直接に部分 XML データを取得方法が考えられる。どちらのアルゴリズムを選択するかは、S-DHT に格納されている統計情報などを利用して推測することが可能であると考えられる。

3.3.2 述語を含むパスの問合せ処理

問合せ q が述語を含む場合、より複雑な処理が必要である。ここでは簡単な例を用いて処理の概要を示す。例として、 $q=/a/b[c]$ なる問合せを考える。まず q から全てのパス式を抽出する（この例の場合は $/a/b$ と $/a/b/c$ になる）。得られたパス式集合のそれぞれの要素に対し、3.3.1 で示したアルゴリズムを適用し、解の候補集合を得る（例では、それぞれ R_1 , R_2 とする）。

次に、 R_1 と R_2 が XML データ上で関連をもっているかど

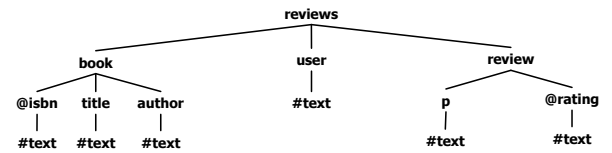


図 8 浅い XML データの構造
Fig. 8 Structure of shallow XML data

うか（先祖子孫関係にあるかどうか）を構造結合 (Structural Join) [12] によって調べる。具体的には、順序ラベル (Dewey Order) の性質から（ピア ID, 文書 ID, 順序ラベル）について、ピア ID, 文書 ID が等しく、順序ラベルがプレフィックスを共有しているかどうかを調べる。構造結合の結果残ったデータが検索結果となる。

4. 評価

提案手法の有効性を検証した。その結果を報告する。

データセットとして、IBM の XML データ生成プログラム ToXgene [13] を用いて生成した浅い XML データと、XMark [14] の xmlgen を用いて生成した深い XML データの二種類の人工データを用いた。図 8, 9 にそれぞれのデータの概要を示す。

Chord のスケールは 10、ネットワーク上のピア数は 1,000 とし、XPath 問合せの処理に要する DHT の参照回数と、ネットワーク上のピア間でのメッセージ転送量をホップ数で算出した。浅い XML データ

浅いデータに対して、表 1 に示す問合せでの比較を行った。個別の問合せ処理に要する DHT の参照回数と、ネットワーク上のピア間のメッセージ転送量を表 1 に示す。 Q_1 の処理では、要素名 $user$ を検索キーとして C-DHT, S-DHT を参照するだけで処理が完結するので、DHT の参照回数、メッセージ転送量とも少ない。 Q_2 の処理についても同様である。 Q_3 は述語を含むため、まず $//review$ と $//review/@rating$ について処理を行い、それぞれの結果を構造結合によって結合する処理を行う。このため、上の二つに比べ、DHT 参照回数、メッセージ転送量とも増えている。 Q_4 , Q_5 についても同様である。

深い XML データ

深い XML データでの評価結果を表 2 に示す。こちらに関しても、テキストしか含まない単一要素や、リーフに近いごく小さい部分 XML データを検索するような問合せでは、DHT 参照回数、メッセージ転送量とも少ない。

一方、 Q_5 のように、ある程度の大きさを持つ部分データを検索する場合、S-DHT を手がかりに繰り返し C-DHT を参照する必要があるため、多数の DHT 参照とメッセージ転送が必要になってしまうことが分かる。このような場合は、S-DHT の統計量を用いて結果部分データのノード数を見積り、ある閾値を超える場合は、そのデータを保持しているピアと直接データをやりとりすることによって、処理を効率化することが考えられる。

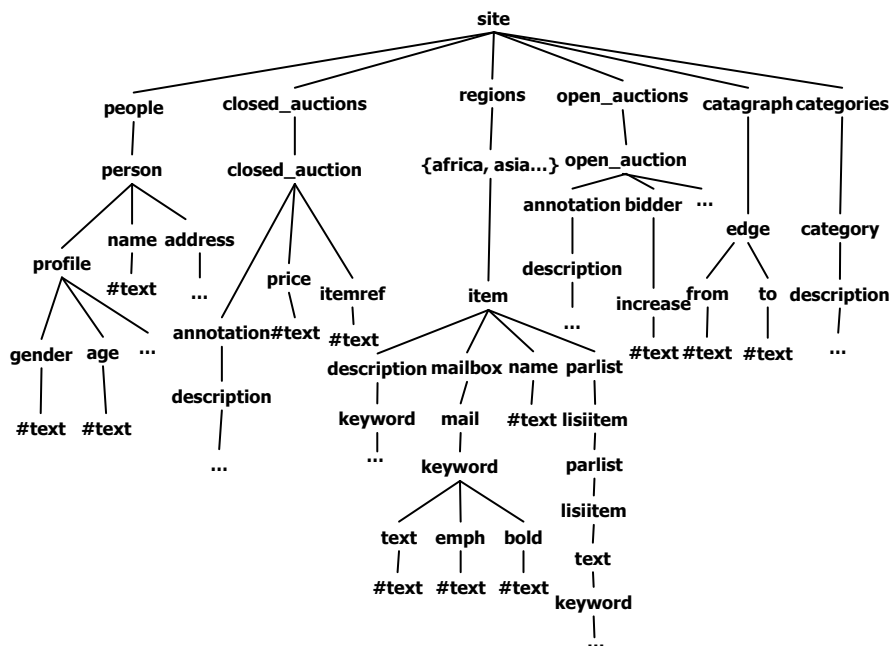


図 9 深い XML データの構造
Fig.9 Structure of deep XML data

表 1 浅い XML データの結果

Table 1 Result of evaluation: shallow XML(# of peers:1000)

Query Benchmark	# of DHT Lookups	Avg.# of Hops
Q1://user	2	19
Q2://book/title	2	19
Q3://review[@rating="3"]	4	39
Q4://review[../book/@isbn="6075782316"]	6	58
Q5://book[author="Zvi Mansouri"]	8	78

表 2 深い XML データの結果

Table 2 Result of evaluation: deep XML(# of peers:1000)

Query Benchmark	# of DHT Lookups	Avg.# of Hops
Q1://item/name	2	19
Q2://mail[date="24/02/2000"]//keyword	8	78
Q3://bidder	12	119
Q4://person[name="Hayato"]	24	234
Q5://asia	112	1116

5. 関連研究

Bonifati 等は XML データをフラグメント化して P2P 環境で検索する XP2P という手法を提案している [3]。XP2P では、格納したい XML データをあらかじめ検索単位にフラグメント化しておく。フラグメントを DHT へ格納する際のハッシュキーには、対象フラグメントの親フラグメントからの相対パスを用いる。この際、単なるハッシュ関数ではなく、Rabin によって提案されたフィンガープリンティング [15] を用いる点に特徴がある。

問合せ処理では、DHT への格納キーと検索キーが正確に一致

する場合は極めて効率の良い検索が可能であるが、そうでない場合は、最悪の場合ほぼ全数探索を行わなければならない。また、パスをそのまま DHT のキーに用いているため、// のような XML に特徴的な問合せを効率的に扱えないという問題もある。

Galanis 等は XML の要素ノード名をハッシュキー、XPath をハッシュ値としてキー サマリーを DHT に格納することにより、各要素ノードの検索を柔軟に行うことができる手法を提案している [16]。

P2P を用いた多様なデータの格納と検索という観点から、松下等はシグネチャを用い、オブジェクトが持つ多様な特徴量

を DHT で検索する手法を提案している [17] . また , Cai 等の RDFPeers は , RDF トリプルの主語 , 述語 , 目的語を , それぞれの DHT に格納することにより , RDF トリプルのための検索機構を提案している [18] . 的野等の RDFCube は , オーバレイネットワーク上に , RDF トリプルに対応した 3 次元構造のハッシュ空間を構築することにより , 効率的に RDF の検索を実現している [19] .

6. ま と め

本論文では , P2P ネットワークを用いて , XML データを効率的に格納や検索するための手法を提案した . XML データの内容を格納する C-DHT と , 構造概要を格納する S-DHT を組み合わせることで , 任意の部分 XML データを P2P ネットワークから検索することが可能となる .

今後の課題としては , 大規模な実験環境とデータによる性能評価を行う予定である . また , ピア間の負荷分散を考慮する必要がある . C-DHT や S-DHT では要素名をハッシュキーとしている . このため , 同一の要素名を持つノードはすべて同じピアに格納される . このため , 特定の要素名を多く持つような XML データの場合 , 特定のピアに偏ってデータが配置される可能性がある . このような状況に対応するための負荷分散機構の検討を行う予定である .

謝 辞

本研究の一部は , 科学技術振興機構戦略的創造研究推進事業 CREST , および科学研究実費補助金 , 若手研究 (B)(#17700110) , 基盤研究 (B)(#15300027) により行われた .

文 献

- [1] Gnutella website, <http://www.gnutella.com>
- [2] <http://www.w3.org/TR/REC-xml>. World-Wide Web Consortium: Extensible Markup Language 1.0.
- [3] A. Bonifati, U. Matrangolo, A. Cuzzocrea, and M. Jain, XPath Lookup Queries in P2P Networks, In *WIDM*, 2004.
- [4] R. Goldman and J. Widom, DataGuides:Enabling Query Formulation and Optimization in Semistructured Databases. *Proc. VLDB*, pp.436-445(1997).
- [5] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, Chord: A scalable peer-to-peer lookup service for Internet applications, In *Proc. ACM SIGCOMM*, 2001.
- [6] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, A Scalable Content-Addressable Network, In *SIGCOMM*, 2001.
- [7] B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph, Tapestry:An infrastructure for fault-resilient wide-area location and routing. Technical Report, UCB/CSD-01-1141, U.C.Berkley, 2001.
- [8] S.Gilbert and N.Lynch: Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services, *SIGACT News*, 33(2), 2002.
- [9] T. Schwentick, XPath Query Containment. *SIGMOD Record*, Vol.33, No.1,2004.
- [10] I. Tatarinov, S. Viglas, K. Beyer, J. Shanmugasundaram, E. Shekita and C. Zhang, Storing and Querying Ordered XML Using a Relational Database System, *SIGMOD*, pp.204-215(2002).
- [11] P. O'Neil, E. O'Neil, S. Pal, I. and G. Schaller, ORDPATHS: Insert-Friendly XML Node Labels, *SIGMOD*, 2004.
- [12] S. Al-Khalifa, H. Jagadish, N. Koudas, J. Patel, D. Srivastava and Y. Wu, Structural Joins: A Primitive for Efficient XML Query Pattern Matching. *Proc. ICDE*, 2002.
- [13] ToXgene website , <http://www.alphaworks.ibm.com/tech/toxgene>
- [14] XMark website , <http://monetdb.cwi.nl/xml/>
- [15] M. Rabin, Fingerprinting by Random Polynomials, Technical report, CRCT TR-15-81, Harvard University, 1981.
- [16] L. Galanis, Y. Wang, S. Jeffery and D. DeWitt, Locating Data Sources in Large Distributed Systems, *VLDB*, 2003.
- [17] 松下亮, 北川博之, 石川佳治, P2P 環境におけるシグネチャを用いたオブジェクト検索方式, 情報処理学会論文誌: データベース, Vol.44, No.SIG 12(TOD 19), pp139-149, 2003 年 9 月
- [18] M. Cai and M. Frank, RDFPeers: A Scalable Distributed RDF Repository based on a Structured Peer-to-Peer Network, In *WWW2004*, 2004.
- [19] 的野晃整, サイドミルザパレビ, 小島功, P2P 環境における RDF データのための三次元索引に基づいた検索, *DBWeb*, pp.9-16(2005).
- [20] 天笠俊之, 植村俊亮, リージョンディレクトリを用いた関係データベースによる大規模 XML データ処理, 日本データベース学会 Letters, 3, 2, pp.33-36(2004).