

P2P 環境における RDF データを対象とした Faceted Navigation インタフェースの実現

齋藤 真衣[†] 渡辺知恵美[†]

[†] お茶の水女子大学大学院人間文化創成科学研究科 〒112-8610 東京都文京区大塚 2-1-1

E-mail: [†]maisaito@db.is.ocha.ac.jp, ^{††}chiemi@is.ocha.ac.jp

あらまし 今日個人で所有されている多種多様なデータを互いに公開・検索し合うことで、より活発な知識共有に結びつけることが可能である。また P2P における様々なデータ検索手法が提供されており、P2P アプリケーションはそれらの検索手法を利用したデータ検索を実現している。しかし人によってつけたメタデータが統一されていない場合、簡単なキーワード検索だけで所望のデータに辿り着くことは困難であり、ユーザにスムーズな検索を促す検索インタフェース実現のためにはユーザの検索を支援するためのデザインを考慮する必要がある。本稿では P2P アプリケーションの検索デザインとして、検索条件を予めリストアップしてユーザに提示して検索目的が曖昧なユーザに対して対話的に絞り込み検索を行う手法として近年注目されている Faceted Navigation に着目し、メタデータを記述する枠組みとして RDF を対象とした上で、P2P で Faceted Navigation インタフェース実現のための均等なメタデータの分散方法を検討する。

キーワード P2P, オーバーレイネットワーク, ファセットナビゲーション, RDF, 線形ハッシュ

A cross-search mechanism on the P2P overlay network using Faceted Navigation for RDF data

Mai SAITO[†] and Chiemi WATANABE[†]

[†] Graduate School of Humanities and Sciences, Ochanomizu University

E-mail: [†]maisaito@db.is.ocha.ac.jp, ^{††}chiemi@is.ocha.ac.jp

Abstract By development of the technologies for P2P network, we can share various type of data such as relational table, xml data and so on, and we can query the data not only by simple keyword search but also SQL-like query expression. Applications which are developed by using these technologies should design query interface so that it helps users to reach any information they want. Especially on P2P network, when there is no rule for adding metadata of the objects, it is not easy to find appropriate keywords for searching. In this paper, we focus on 'faceted navigation', which is one of design patterns for helping query behaviours, and we investigate some architectures for applying faceted navigation interfection on P2P network.

Key words P2P, Overlay Network, Faceted Navigation, RDF, Linear Hash

1. はじめに

今日さまざまなデータが個人で所有されている中、それを互いに公開・検索することで、より活発な知識共有へと結びつけることが可能である。ファイル共有システムなど個々人の間で P2P による様々なデータ検索手法が研究されており、分散ハッシュテーブル (DHT) のキーによる検索やそれを応用させたキーワードによる部分一致検索、リレーショナルデータベース (RDB) を P2P で扱うためのスキーム PIER [8] 等が提案されている。P2P アプリケーションはこれらの検索手法を利用し、

ユーザの必要とするデータの検索を実現する。検索時、ユーザは目的のデータを取得するまでに問合せを行ってその結果を得、そしてその結果をもとに問合せを行うという操作を繰り返す。ユーザが所望するデータに辿り着けるかどうかは問合せの条件に左右されることから、ユーザのスムーズな検索を支援するような検索インタフェースのデザインを考慮することが重要となる。特に P2P アプリケーションの場合には P2P にて共有するデータに所有者が銘々にメタデータをつける場合、人によって同じオブジェクトに対しても異なるメタデータをつける場合もあり、簡単なキーワード検索のみで必要なものを絞り込むこ

とが難しい場合も考えられる。そこで先行研究では P2P アプリケーションにおける検索デザインとして Faceted Navigation が有用と思われることから、Faceted Navigation を実現するためのリレーショナルスキームを提案し、P2P における検索コストを比較した。

本稿ではメタデータを記述する枠組みとして RDF(Resource Description Framework) [1] に着目し、P2P で RDF データを対象とした Faceted Navigation を実装するための均等なデータの分散管理方法を提案する。2 節で想定する状況、アプリケーションにおけるユーザの検索パターンとその中で Faceted Navigation について詳しく述べ、3 節で関連研究について紹介し、4 節で P2P におけるデータの分散管理方法について述べる。最後に 5 節でまとめと今後の課題を提示する。

2. 前 提

本節ではまず想定する状況と対象データ構造を述べ、それを踏まえた上で考慮すべき検索時のデザインパターン、さらにその中で着目した Faceted Navigation インタフェースについて詳細を示す。

2.1 想定する状況と対象データ構造

本稿で想定する状況及び対象とするデータ構造について述べる。ユーザが所有する何かのデータに対し、そのデータがどんなデータであるかを定義するためにメタデータが記述される。メタデータの記述方法の一つとして図 1 に示したような RDF が挙げられる。RDF とは、トリプルと呼ばれる主語 (subject)、述語 (property)、目的語 (object) の 3 つの要素から成る。subject はリソース、property は subject の特徴や subject と object の関係、object は subject と関係するリソース、もしくは property の値を示す。RDF に従ってメタデータを

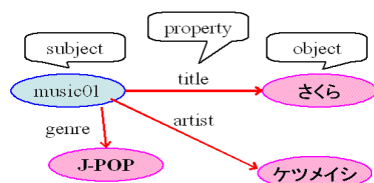


図 1 RDF トリプル

記述することで柔軟な表現が可能となり、その代表例としては人に関する情報を記述する FOAF(The Friend of a Friend) [2] や音楽データに関する情報を記述する MusicBrainz [3] がある。例えば図 1 のような音楽データに対しては“さくらというタイトルの曲を歌っているアーティスト”といったようにその属性を組み合わせて記述することで相互関係を辿ることができる。しかし、たとえ構造が決まっていたとしてもメタデータの付け方は人それぞれであり、例えば<title: さくら>など<属性名: 属性値>といった形でタグのようにユーザが自由に付与することが考えられる。別のユーザにとっては自分が求めたいデータに対してどんなタグが付けられているのか分からず、すぐに目的のデータに辿り着くことが困難となってしまう。特に漠然とした目的を持ったユーザにとっては、興味のあるデータに辿り着くためには何をキーワードに指定すればよいか見当がつか

ない。そこでそのようなユーザに対して検索を支援することが求められる。通常ユーザは何かのアプリケーションを通して検索を行うことが多いため、ユーザが検索に利用するアプリケーションのデザインパターンの観点から、検索をサポートする必要があると考えられる。

2.2 検索におけるデザインパターン

本項では一般的なアプリケーションにおけるデザインパターンについて述べる。Peter [9] による分類を以下に示す。

- Federated Search (複数・横断サーチ)
- Faceted Navigation (複数・横断ナビゲーション)
- Best Bets (最適結果の提示)
- Category Match (カテゴリマッチ)
- Auto Suggest (自動提示)
- Contextual Search (コンテキスト連動サーチ)

Federated Search とは異なる検索方法やインタフェースについての知識を必要とせず、1 つの検索インタフェースで複数のサイトから情報を得られるようにサポートするものであり、Faceted Navigation はデータの属性リストと該当件数を提示することでユーザの次の絞り込みを誘導する。また検索結果として最適とされるコンテンツを推薦、提示する Best Bets の他、ユーザの検索キーワード入力の際、Google ツールバーのように入力された文字からキーワード候補を表示する Auto Suggest などが挙げられている。RDF データの検索方法として Faceted Navigation を用いたものが提案されており [5] [6]、我々は本項で挙げた検索デザインの中で Faceted Navigation に着目して以下議論を行う。

2.3 Faceted Navigation

Faceted Navigation とは、データの検索条件として属性やメタデータ項目を予めリスト表示しておき、それを選択することで検索クエリなしでユーザを目的のデータに導くことができるインタフェースである。検索対象データの属性名をファセット、その属性値をファセット要素としたとき、Faceted Navigation ではまずファセットを選択し、次にそのファセット要素を選択するという操作を繰り返すことで検索対象データを絞り込んでいく。リストからファセットを選択するとそのファセット要素と該当件数がリストアップされ、ファセット要素が選択されると絞り込まれた検索対象データの持つファセットが新たにリストアップされる。すなわち検索を進めるごとに集約演算が行われ、検索対象データを絞り込んでいく。Faceted Navigation の代表例として、Flamenco Search [10] が挙げられる。図 2 に Flamenco Search のインタフェースを示す。Flamenco Search では、受賞年や受賞分野を指定することで歴代のノーベル賞受賞者の中から該当データを絞り込むことができる。絞り込み検索の流れを Flamenco Search を例に詳しく述べる。まず歴代のノーベル賞受賞者全員のデータに対し、例えば属性名 = GENDER の属性値 = male で絞り込む。この 1 回の絞り込みにより属性名 = GENDER の属性値 = male という属性を持つノーベル賞受賞者のデータが検索対象データとなり、検索対象データが持つ属性とその該当件数がインタフェースに表示される。さらにその検索対象データに対し属性名 = COUNTRY の属性値 = Japan という属性で絞り込むと、GENDER = male と

COUNTRY=Japan を共に満たすノーベル賞受賞者のデータが検索対象となり、同様に次の絞り込み条件として検索対象データの属性と該当件数がユーザーに提示される。

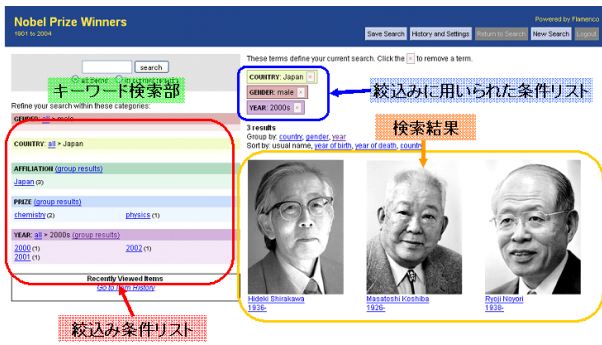


図2 Flamenco Search

3. 関連研究

本稿では対象データを RDF データとすることから、本節では関連研究として RDF データを P2P で共有する方法、さらに P2P 上で処理するために PIER と組み合わせることから、RDF データを RDB に格納する方法について紹介する。

まず RDF データの P2P における共有に関して、RDF-Peers [7] や RDFCube [11] などが提案されている。RDFPeers とは分散ハッシュテーブル (DHT) を用いて RDF トリプルをネットワーク上のノードに格納し、その検索を実現する分散 RDF データベースである。DHT とはキーと値をペアにしたハッシュテーブルを各ノードで分散管理し、キーによって効率的な検索を行うものである。RDF トリプルの subject, property, object それぞれをキー、RDF トリプルそのものをキーに対する値とし、1つのトリプルに対して3つのキーが割り当てられる。subject, property, object のいずれかを条件とすれば、該当するトリプルを得ることができる。それに対し RDFCube とは subject, property, object に基づいた三次元構造のハッシュ空間 (RDFCube) に RDF データを写像し、それが RDFCube の一定領域内に存在するか否かをビットで表現する。導入したビットの演算を行うことで RDF データのハッシュ値の範囲を絞り込んだ後、RDF データそのものの結合演算を行い、分散環境での結合演算の効率を図る。

また、Adabi 氏らによって RDF データの RDB への格納に関する研究が提案されている [4]。この研究では RDF データを RDB に格納する3つのアプローチとして、Triple Store, Property Tables, Vertical Partition が挙げられている。また [4] では検索アプリケーション例として彼らが開発している RDF の Faceted Navigation による検索システム Longwell [5] をあげ、当該システムで Faceted Navigation による検索で発行される問合せ文をあげ、それらの性能を評価している。

RDF データを P2P で扱う技術 [7] [11] 及び RDF データを Faceted Navigation で検索するための技術 [6] はそれぞれ提案されているが、それらを組み合わせたものはこれまで確認されていない。

4. P2P における Faceted Navigation の実現

本節では P2P において Faceted Navigation を実現するため、4.1 項で P2P におけるデータの格納方法を示し、その中から2つの格納法について 4.2 項で比較する。さらに 4.3 項で線形ハッシュを用いた均等なデータの分散管理方法、4.4 項で P2P 上の複数ノードによる件数取得処理について述べる。なお本稿で想定する P2P ネットワークは、関連研究の多くと同様にセンターサーバのないピア P2P であり、分散ハッシュテーブル (DHT) による構造化されたオーバーレイネットワークを想定する。

4.1 P2P におけるデータ格納方法

P2P において RDF データを管理する方法として、本稿では RDF データをリレーショナルデータに変換し、PIER [8] 等リレーショナルデータを P2P で管理する手法を利用して P2P で管理することとする。先行研究 [12] では、[4] で提案された RDF データをリレーショナルデータとして格納する手法として、図3に示した Triple Store, Property Table, Vertical Partition それぞれについて問合せの体系化及びパフォーマンスコストの比較検討を行った。Faceted Navigation では検索対象となる

Triple Store			Property table				
s_id	prop	obj	s_id	propA	propB	propC	propD
1	A	a1	1	a1	b1	c1	d1
1	B	b1	2	a1	b2	c2	d1
1	C	c1	3	a1	b2	c1	d2
1	D	d1	4	a2	b2	c2	d3
2	A	a1	5	a1	b1	c1	d2
2	B	b2					
2	C	c2					
2	D	d1					
...	...	...					
5	A	a1					
5	B	b1					
5	C	c1					
5	D	d2					

Vertical Partition							
propA		propB		propC		propD	
s_id	obj	s_id	obj	s_id	obj	s_id	obj
1	a1	1	b1	1	c1	1	d1
2	a1	2	b2	2	c2	2	d1
3	a1	3	b2	3	c1	3	d2
4	a2	4	b2	4	c2	4	d3
5	a1	5	b1	5	c1	5	d2

図3 Triple Store, Property Table, Vertical Partition

RDF データの property や object の値を検索条件としてユーザーが指定することで対象データを絞り込み、また対象データの持つ property や object の値を提示するといった絞り込み操作を重ねる必要があることから、ユーザーが指定した検索条件を全て満たすデータを求めるために subject の id(s_id) による結合を繰り返す必要がある。上記を考慮した結果、Triple Store(TS) を予め s_id で自己結合させたもの (JTS) を property 及び object の値をキーに分散管理させ、より結合処理を少なくして DHT への問合せ回数を抑えることが最適と実証された。

4.2 TS と JTS の比較

本節では JTS と自己結合する前の TS を改めて比較する。図4に TS と JTS を DHT に分散させた様子を示す。図4に示したように、例えば prop=A かつ obj=a1 という条件で検索し DHT からデータを取得してくる際、TS は prop=A かつ obj=a1 という条件で選択演算を行い、さらに該当タプルの s_id で自己結合して中間テーブルを生成することで prop=B, C, D それぞれに関する集約結果を得ることができる。それに対し JTS では各 property に関するデータをまとめて管理する1つのノードから取得してくることから、ある条件で問合せを行った際、TS と JTS では P2P ネットワーク上のどれくらいの数のノードにデータを分散させるか、すなわち各ノードが負担す

るデータ量に違いがあると言える。2手法を比較した結果を図

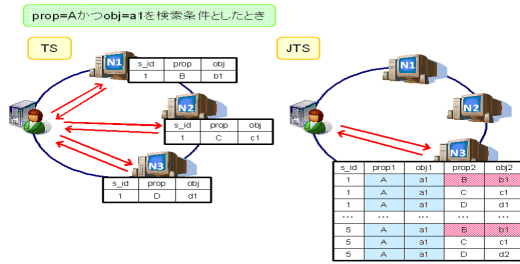


図4 TSとJTSの分散管理の様子

5に示す。図5に示したように、ある条件におけるDHTへの

	TS	JTS
ある条件によるDHTへの問合せ回数	$s+1$	1
あるキーに対する該当タプル数	p	$p \times s$

p:property数 s:該当subject数

図5 TSとJTSのコスト比較

問合せ回数はJTSが1回で済むのに対し、TSでは自己結合のために該当subjectの数だけ他のpropertyの値を取得するための問合せが必要となり、もし該当subjectが10万個あった場合、10万回の問合せが発行される。またあるキーに対して該当するタプル数はTSがproperty数のみで済むのに対し、JTSでは該当するsubjectの数それぞれに対するproperty数の分となる。P2Pでのネットワークコストを抑えるためには、DHTへの問合せ回数及びあるキーに対する該当タプル数をどちらも少なくするような処理が最適となる。

4.3 線形ハッシュを用いたデータの分散管理

本稿では各ノードが負担するデータ量がより均等になるよう、JTSを用い、線形ハッシュによってDHTで管理するデータ量を分割することとする。

4.3.1 線形ハッシュ

線形ハッシュとはハッシュ空間の改善を目指した動的ハッシュの手法の一つであり、キー値keyとハッシュ関数Hの他にパラメタとしてレベルi、最初のバケット数Nを用いる。ここでレベルとはハッシュされたキー値が何ビットで表わされるかを示すものであり、オーバーフローが発生した際に分割するバケットの位置を分割ポインタPで定める。線形ハッシュアルゴリズムは以下の通り。なお、 $H_i(key) = H(key) \bmod (2^i N)$ 、Pの初期値を0（最初のバケットの位置）とする。

- (1) $a = H_i(key)$ のとき、 a の値によって該当バケットにデータを保存する。ただし $a < P$ ならば、 $a = H_{i+1}(key)$ とする。
- (2) (1) でデータを挿入した際、バケット内の要素数を超過してオーバーフローしたらバケットを分割する。ただし分割するのは分割ポインタがあるバケットとするため、オーバーフローが生じているバケットが分割されるとは限らないが、分割ポインタのないバケットでオーバーフローが生じた場合は一時的にそれを許すものとする。
- (3) 分割後はPを次のバケットに動かす。全バケットが分割されると、レベルが上がると共にPは再び最初のバケットを示す。ここでの各パラメタの動きは以下の通り。

$P = P + 1$

if $P \geq 2^i$ then $P = 0, i = i + 1$

(1)~(3) を繰り返すことで、データ全体を均等に分割することができる。

4.3.2 DHTへの登録 (put)

上述の線形ハッシュを用い、P2P上で同様のデータ分割処理を行うことを考える。例として、 $\text{prop} = A$ かつ $\text{obj} = a1$ という絞込み条件で各ノードがDHTへ分散させる際のデータ分割処理の流れの詳細を示す。分散管理するタプルをそれぞれ $t1, t2, \dots, tn$ 、1つのノードでまとめて管理するタプルの最大値を $t_{\max}$ とする。

(0) バケット内の要素数 $= t_{\max}$ とする。

(1) $\text{prop}A:a1$ という条件に対し、線形ハッシュを用いて $t1$ から順に管理ノードを決定し、DHTへ分散させる。

(2) 最初のバケット数 $N=1$ とし、タプル数に応じて分割する。

・ $tn \leq t_{\max}$ の場合：まとめて1つのノードで管理

・ $tn > t_{\max}$ の場合：線形ハッシュを用いて分割したものを複数ノードで管理

(3) ハッシュ関数のキーを“ $\text{prop}A:a1[\text{バケット番号}]$ ”とし、DHTへ分散させる（図6）。

(4) $\text{prop}A:a1$ をキー、分割ポインタP、レベルiを値としてバケット分割時に必要なパラメタもDHTで管理する。

(5) バケットの分割が生じた際には、該当バケット番号を付与した新たなキーでDHTへ分散させる。

$t1 \sim tn$ に対し(1)~(5)を繰り返し、P2P上で線形ハッシュを用いてデータを分割する。

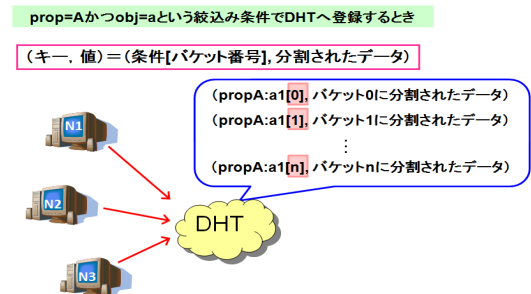


図6 線形ハッシュによる分散管理

4.3.3 DHTへの問合せ (get)

線形ハッシュを用いてDHT上でデータを分散させた上で、それに対する問合せについて図7に示した流れの詳細を述べる。

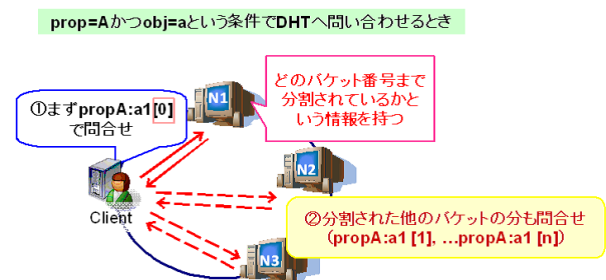


図7 DHTへの問合せ

(0) クライアント側では目的のデータがいくつに分割されているか把握していないため、バケット 0 を管理するノードは該当する全データの分割バケット数も管理する。

(1) クライアントはまず“絞込み条件 [0]”(例えば propA:a1[0]) をキーとし、DHT に対して問合せを行う。

(2) バケット 0 を管理するノードは自身の管理するデータに加え、分割バケット数を返す。

(3) クライアントは分散バケット数に応じて各バケットに対し問合せを行うことで、条件に該当する全データを取得する。

4.4 該当件数取得

上記で述べたように線形ハッシュを用いたデータ分割により各ノードの負担を均等にし、複数ノードによる問合せの並列処理を行う。さらに分散管理しているデータを各ノードからそのまま全て取得してくるのではなく、各 property の集約結果の件数を各ノードで計算し、その計算結果のみをクライアント側に返すことでネットワークコストの削減を図る。このとき集約結果の property のリストは別で用意し、クライアント側ではリストと件数を共に取得できるようにする。

5. まとめと今後の課題

本稿ではより柔軟な表現が可能である RDF によって記述されたメタデータを検索対象とした上で、我々ユーザが用いるさまざまな検索パターンの中でもユーザのよりスムーズな検索をサポートする Faceted Navigation に着目し、P2P における Faceted Navigation 実現のための手法について述べた。今後は提案した手法に基づいた評価実験を行うと共に、データ更新時の処理や RDF データに対するあらゆる問合せに対応できるようさらなる考察を進めていきたい。

文 献

- [1] World Wide Web Consortium: Resource Description Framework(RDF). “ <http://www.w3.org/RDF/> ”
- [2] The Friend of a Friend (FOAF) project “ <http://www.foaf-project.org/> ”
- [3] MusicBrainz “ <http://musicbrainz.org/> ”
- [4] D. Abadi, A. Marcus, S. Madden, and K. Hollenbach.: “ Scalable Semantic WebData Management Using Vertical Partitioning, ” In *Proc. the 33rd International Conference on Very Large Data Bases*, pp. 411–422, September 2007.
- [5] Longwell “ <http://simile.mit.edu/wiki/Longwell> ”
- [6] E. Oren, R. Delbru, and S. Decker.: “ Extending faceted navigation for RDF data, ” In *Proc. the 5th International Semantic Web Conference*, pp. 559–572, November 2006.
- [7] M. Cai and M. Frank.: “ RDFPeers: A Scalable Distributed RDF Repository based on A Structured Peer-to-Peer Network, ” In *Proc. the 13th international conference on World Wide Web*, pp. 650–657, May 2004.
- [8] R. Huebsch, J. Hellerstein, N. Lanham, B. T. Loo, S. Shenker, and I. Stoica.: “ Querying the Internet with PIER, ” In *Proc. the 29th International Conference on Very Large Data Bases*, pp. 321–332, September 2003.
- [9] P. Morville: “ Search Patterns, ” In *Proc. iA SUMMIT*, April, 2008.
- [10] Flamenco Search: UC Berkeley School of Information The Flamenco Search Interface Project “ <http://flamenco.berkeley.edu/index.html> ”
- [11] 的野晃整, M.Said, 小島功: “ RDFCube: 分散 RDF データベースのための三次元ハッシュ索引, ” 日本データベース学会論文誌, Vol.4, No.4, pp5-8(2006).

- [12] 齋藤真衣, 渡辺知恵美: “ P2P 環境における Faceted Navigation インタフェース実現のための諸検討, ” 情報処理学会研究会報告, 2008-DBS-146, pp.283-288 (2008).
- [13] 齋藤真衣, 堀之内武, 渡辺知恵美: “ 対話的な絞込み操作を考慮した P2P による地球流体データアーカイブサーバの横断検索, ” 日本データベース学会論文誌, Vol.7, No.1, pp.91-96(2008).