

P2P 環境での Faceted Search における 効率的な問合せの実現に向けて

諫本有加[†] 渡辺知恵美[†]

P2P ネットワークの発達によって XML データやリレーショナルテーブルなど多様なデータを共有することが可能となっており、様々な分野から注目を浴びている。それらの多くはインタフェースにキーワード検索を利用されており、所望のデータに辿り着かない場合がある。そこで、ユーザにスムーズな検索を促すような検索インタフェースとして Faceted Search が着目されている。本稿では P2P 上で Faceted Search を実現するために提案された手法に対して、効率的な問合せを行うための検証実験を行った。さらにユーザテストを行い、どの条件が選択されやすいかを調べることでノードへのアクセス集中の可能性を検証した。

Efficient a cross-search mechanism using Faceted Search in P2P Environment

YUKA ISAMOTO[†] CHIEMI WATANABE[†]

Because of the development of technologies for a P2P network, P2P network environment has become popular and we can share various types of data such as relational tables and xml data. However, most of all applications have only keyword search interface, then, users can not easily reach any information they want if they do not have any appropriate keywords to search. We focus on Faceted Search, which is one of search techniques for users who have only ambiguous image about what they want to find them by narrowing down the objects interactively. In this paper, we evaluated mechanism using Faceted Search in a P2P environment. Moreover, we examined which query users select and verified the possibility of the access concentration on the node.

1. はじめに

今日、P2P ネットワークを利用したデータ検索が様々な分野から注目を浴びている。P2P アプリケーションの多くは、データの検索を実現するためのインタフェースとしてキーワード検索を採用している。このキーワード検索を実現するため、キーワードを基に作成されたハッシュ値によってデータを分散する分散ハッシュテーブル (DHT) が利用されている。その他、リレーショナルデータベースを P2P で扱うためのスキーム PIER[1]や range query[2]や full text search[3]など多くの検索手法がある。

しかしながら既存の P2P アプリケーションでは、どのようなデータがあるのか見たいなどユーザの目的があいまいな場合、適切なキーワードを見つけるのは難しい。さらに、ユーザが考えたキーワードで検索しても所望のデータに辿り着かない場合もある。そこで先行研究[4][5][6]では、データの種類やそのデータがもつ情報を知らなくても提示された要素を選択するだけで求めるデータに辿り着く検索インタフェースである Faceted Search に着目し、それを P2P 上で実現するための手法を提案した。

本稿では、その手法を用いて効率的な Faceted Search を実現するために検証実験を行い、その結果を述べる。Faceted Search では条件を対話的に絞り込んだり、追加したりしながらデータを検索するため、問合せが複数回行われる。さらに、ユーザの興味やその時のニュースなどにより、特定の条件が頻繁に選択される。先行研究では検索条件を基に P2P 上にデータを分散しているため、ある条件が頻繁に選択されると、特定のノードに問合せが集中する可能性がある。そこで本稿では、ユーザが Faceted Search のアプリケーションでどのような条件から絞り込むかを検証した。さらにノードへの負荷についても検証するため、条件をどの程度繰り返し利用するかを調べた。その際、Faceted Search が目的のあいまいなユーザにも利用されるという点から、目的のあいまいな場合と明確な場合の 2 通りの実験を行った。

また Faceted Search では、ユーザが絞り込みを行うと次の絞り込み条件とその該当件数を求める必要があるが、P2P 上で実現する際にはその集約処理をノード側で行う方法とクライアント側で行う方法の 2 つの手法が考えられる。そこで本稿では、効率的な Faceted Search を実現するためにどちらの手法で行うべきかを検証した。その際、クライアントとしてネットワークが不安定な携帯電話も考えることができるため、有線 LAN と無線 LAN の 2 種類の環境による問合せの処理時間を計測した。

2 節で前提知識として Faceted Search について、3 節では P2P 上で Faceted Search を実現するための手法、4 節ではその手法に対する考察を述べる。5 節ではユーザテストの概要とその結果、6 節では P2P 上での Faceted Search に関する検証について述べ、7 節でまとめる。

[†] お茶の水女子大学大学院人間文化創成科学研究科
Ochanomizu University Graduate School of Humanities and Science

2. 前提

2.1 Faceted Search

近年、さまざまなアプリケーションで Faceted Search を利用したインタフェースが使われている。Faceted Search 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目的のデータに辿り着けるインタフェースである。図 1 に Faceted Search の代表例である Flamenco Search[7]を示す。Faceted Search ではデータを絞り込むための条件をファセット (図中①) と呼び、それぞれのファセットを分類したクラスの名前をファセット名 (図中②)、その中でリストアップされた条件をファセット値 (図中③) と呼ぶ。データを検索する際は、ユーザはファセット値を選択するだけで良い。結果には条件を満たすデータとそのデータがもつファセットが新たにリストアップされるため、ユーザはその結果を見ながら繰り返し検索を行うことができる。

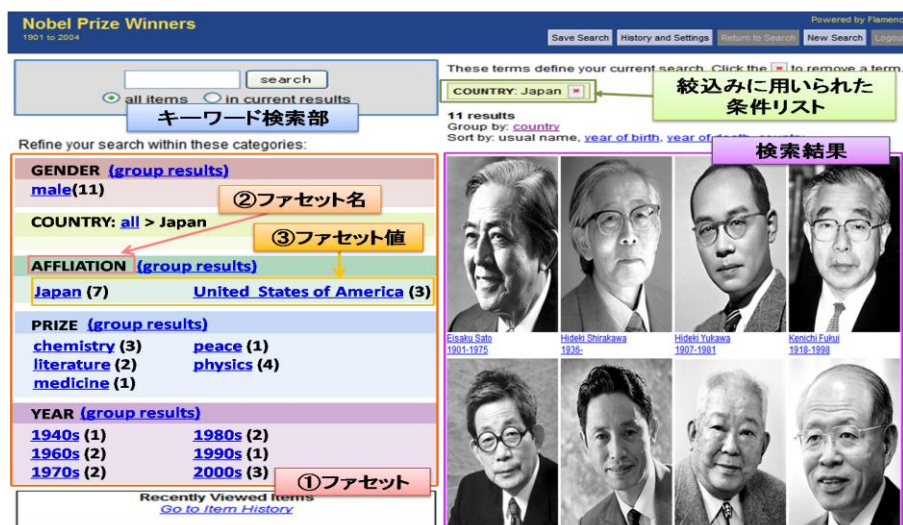


図 1 Flamenco Search[7]

Flamenco Search では、受賞年や受賞分野というファセットの中からユーザが興味のある値 (例えば物理学賞など) を選択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。図 1 の右側は日本人受賞者を検索した際の結果データを示している。図 1 の左側にはデータを絞り込むための条件がリスト表示されてい

る。図 1 では GENDER, COUNTRY, AFFILIATION, PRIZE, YEAR がファセット名であり、その列にリストアップされたそれぞれの条件がファセット値である。したがって図 1 の結果を求める際は、ファセット名=COUNTRY, ファセット値=Japan で絞り込みを行う。図 1 のファセット値の隣に表示された数字は、その条件に該当するデータ数を表す。例えば、ファセット名「YEAR」の「1960s (2)」は 1960 年代に受賞した人が 2 人いることを示す。したがってユーザがこのファセット値を選択すると、日本人の中で 1960 年代に受賞した人物のデータが絞り込まれる。

3. P2P 環境における Faceted Search の実現

先行研究では、P2P 上で Faceted Search を実現するためのデータの格納について、P2P の構造に構造型の Overlay である分散ハッシュテーブル (DHT) を想定した際のデータの分散管理について提案している。

3.1 データの格納方法

図 2 は Flamenco Search のデータがもつ属性を説明のために分かりやすく図式化したものである。ここでは、図 1 の「Eisaku Sato」と「Hideki Shirakawa」の 2 名のデータ構造のみ表している。

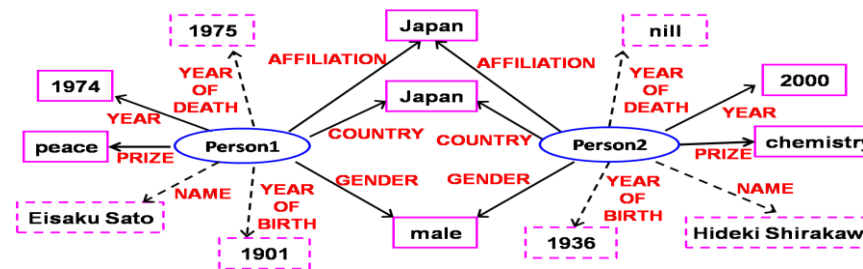


図 2 Flamenco Search のデータの構造

図 2 より、対象データの属性には GENDER, COUNTRY, AFFILIATION, PRIZE, YEAR, NAME, YEAR OF BIRTH, YEAR OF DEATH があるが、これらの中からファセットに利用できるものを選択する必要がある。その際、全ての属性の中で多くのデータがもっている属性とともっていない属性に分類し、ファセットに適す属性を選択する。例えば図 2 では、Person1 は対象データを一意に表す属性、GENDER は複数の対象データに共通している属性である。このように分類すると、GENDER, COUNTRY, AFFILIATION, PRIZE, YEAR の 5 つが複数のデータで共通しており、ファセットに

適切だと考えられる。図 2 ではファセットとして適切な属性を実線の矢印で、それ以外の属性を点線の矢印で示している。なお、リレーショナルテーブルや XML データの中からファセットとして適切な属性を選択する手法が[8][9]で提案されている。

ファセットの情報をリレーションで表現する方法が文献[10]で提案されており、先行研究[4]ではその中から P2P データの配置及び検索に適する表現を比較検討し、以下の表現を選択した。

FacetInfo= (target_id, facet_name, facet_entity)

target_id	facet_name	facet_element
person1	country	Japan
person1	gender	male
person1	affiliation	Japan
person1	prize	peace
person1	year	1974
person2	gender	male
person2	country	Japan
...

表 1 図 2 のファセットデータのリレーション表現

表 1 は図 2 のデータの構成を表しており、対象データの id と対応するファセットを 1 タプルとして表現している。表 1 のテーブルに対して Faceted Search による問合せをする場合、タプルの結合を行う必要がある。例えば、図 2 で prize:peace で絞込みを行うと person1 が該当する。このデータは prize:peace 以外に gender:male, affiliation:Japan などをファセットにもつ。これらを次の絞込み条件として求めるため、表 1 のテーブルを用いて以下の SQL が発行される。なおこの時点では、1 つのコンピュータ上の DB を想定されている。

```
SELECT FA.facet_name, FA.facet_element, count(FA.target_id)
FROM FacetInfo FA, FacetInfo FQ
WHERE FQ.facet_name='prize'
      AND FQ.facet_element='peace'
      AND FQ.target_id=FA.target_id
GROUP BY FA.facet_name, FA.facet_element;
```

この SQL では prize:peace を満たすタプルを求め、それらの target_id を用いて FacetInfo テーブルの自己結合を行っている。さらに、facet_name と facet_element をグループ化し、そのファセットを持つデータ件数を求めている。このように Faceted Search を実現するためにはテーブルの自己結合とファセットによる集約を行う必要がある。

次にこの処理を P2P 上で行う場合について述べる。

3.2 P2P 環境におけるデータの分散管理

DHT ではハッシュ関数によって算出された key とその value のペアを各ノードが分散管理し、検索時には key を指定することでそれに対応する value を取得する。したがって、DHT 上で図 2 のデータに対して Faceted Search を実現するには、表 1 の何を key、何を value にするか考えなければならない。図 2 の場合、prize:peace を選択すると person1 に属す gender:male などが次の絞込み条件として必要となることから、ユーザが選択した条件を key、それに対応するタプルを value として各ノードに分散することが考えられる[10]。しかしながら、prize:peace と gender:male のハッシュ値が異なる場合、別々のノード上に管理されてしまう。3.1 項の SQL で示したように、ファセットの取得には自己結合が必要であるため、gender:male を求める際には条件を満たすデータのファセットを管理する全てのノードのテーブルと自己結合を行わなければならない。この処理には prize:peace を満たすデータ分だけ他のファセットを取得するための問合せが必要となり、ネットワークトラフィックが増大してしまう。そこでネットワークトラフィックを抑え、かつファセットを求めることができるように表 1 のテーブルを予め target_id で自己結合したものを分散することにした[4]。自己結合したものを表 2 に示す。

FacetInfo= (facet_name1, facet_entity1, target_id, facet_name2, facet_element2)

facet_name1	facet_element1	target_id	facet_name2	facet_element2
country	Japan	person1	gender	male
country	Japan	person1	affiliation	Japan
country	Japan	person1	year	1974
country	Japan	person1	prize	peace
gender	male	person1	country	Japan
...
country	Japan	person2	gender	male
...

表 2 表 1 を target_id で自己結合した結果テーブル

表 2 の facet_name2, facet_element2 は facet_name1 と facet_element1 を満たすデータをもつその他のファセットである。facet_name1 と facet_element1 を key, target_id, facet_name2, facet_element2 を value として各ノードに分散すると、表 2 の country:Japan とその条件を満たすデータの全てのファセットが同じノード上に存在することになる。したがって、図 1 のように country:Japan で絞り込んだ場合、その条件を key にもつノード上で以下の SQL を実行して集約結果を求めることができる。

```
SELECT FA.facet_name2, FA.facet_element2,count(FA.target_id)
FROM FacetInfo FA
GROUP BY FA.facet_name2,FA.facet_element2;
```

求めたデータに対してさらに prize:peace で絞り込んだ場合、以下の SQL を実行する。

```
SELECT FA.facet_name2, FA.facet_element2,count(FA.target_id)
FROM FacetInfo FA, FacetInfo FQ
WHERE FQ.facet_name2='prize'
      AND FQ.facet_element2='male'
      AND FQ.target_id=FA.target_id
GROUP BY FA.facet_name1,FA.facet_element1
```

2 回目の問合せの際も 1 つのノード上で処理を行うため、自己結合によるネットワークトラフィックが増えることはない。

また、各ノードへのデータの分散、各ノードからの結果取得の際は DHT で定義されている以下の関数を利用する。

ファセットデータの登録 : put(key,value)

ファセットデータの取得 : get(key)

表 2 の 1 番上のタプルをノードに分散する場合、country:Japan を key として put("country:Japan","(person1, gender,male)")を実行する。そしてこのタプルの管理ノードを決めるために、ハッシュ関数 h(country:Japan)を実行し、そのハッシュ値を割り当てられたノードにデータを登録する。また country:Japan を満たすデータのその他のファセットを取得する際は get(country:Japan)を実行する。そしてその条件からハッシュ値を算出し、そのハッシュ値をもつノードに問合せを行う。そのノードは、保持するタプルに対して次の絞り込み条件となるファセットとその該当件数を求める。

3.3 問合せ処理

P2P における Faceted Search の処理の流れを以下に示す。

Step1. ユーザが条件を指定すると、その条件を key にして求められたハッシュ値と一致するノードに問合せ、該当タプルを絞り込む。

Step2. 集約結果としてファセット名とファセット値、さらに該当件数を取得する。

Step3. 取得した結果を表示し、その中からユーザが次の絞り込み条件を選択する。

Step2 において、図 3 に示した 2 通りの手法が考えられる。

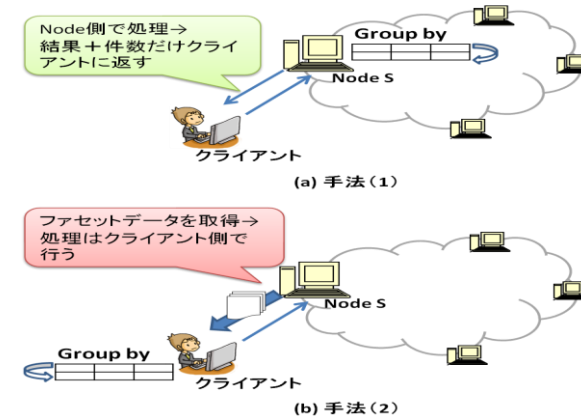


図 3 問合せの 2 つの手法

- ◆ 手法 (1) : ノード側で集約処理を行い、結果のみ取得する方法 (図 3 の(a))
ユーザが条件を選択した場合、ハッシュ関数の値に一致するノードに接続し、そのノード上でクエリを発行する。クライアント側は集約結果のみ取得するため、トラフィックは抑えられる。しかしながら、絞り込みの数だけ P2P ネットワークに接続するため、合計処理時間は大きくなるかもしれない。特にクライアントが携帯電話など、ネットワークが不安定なものであった場合、ノードへの接続時間が増えると考えられる。また 1 つ目の条件を key に固定すると、1 つ目に選択されやすい条件を key に割り当てられたノードに問合せが集中してしまう。
- ◆ 手法 (2) : ノードから該当タプルをそのまま取得し、クライアント側で集約処理を行う方法 (図 3 の(b))

1 回目の絞り込み条件に該当する全ファセットデータをクライアント側に取得し、その後の絞り込み検索は P2P ネットワークに接続せずにクライアント側で実行する。そのためネットワークが不安定な場合、2 回目以降の処理時間は手法 (1) より速くなると考えられる。しかしながらクライアント側に大量のファセットデータをダウンロードするため、トラフィックが増大し、合計処理時間は大きくなるかもしれない。

手法 (1) で述べられているように 1 つ目の条件を key に固定した場合、複数のユーザが同じ条件を 1 つ目を選択すると、ある特定のノードに処理が集中することが懸念される。また、効率的に処理を行うためにどちらの手法で行うべきか検証する必要がある。これらについて次節で詳しく述べる。

4. 効率的な実行のために

3 節で述べたように、ユーザの絞込み方と集約の方法によって Faceted Search の問合せの効率が左右される。効率的な問合せを実現するために検証しなければならないのは以下の 2 点である。

1. 複数のユーザで選択する条件が集中するか

複数のユーザが同じ条件から検索を行った場合、その全てのユーザが 1 つのノードに問合せを行う。さらにその条件を指定したまま検索を繰り返すと、毎回同じノードに問合せを行うことになる。これでは特定のノードに負荷がかかってしまう。そこで、Faceted Search による問合せを効率よく行うためにはユーザの選択する条件が集中しないかを考える必要がある。また Faceted Search は予め検索条件がリストアップされており、目的があいまいなユーザも利用できることから、ユーザの目的の有無も考慮する必要がある。この 2 つの場合では条件の選び方や検索を繰り返す回数が異なると考えられる。したがって、特定の目的がない場合と明確な場合にユーザがどのような条件を選択するか実験を行った。この実験については 5 節で述べる。

2. 3.3 項の手法 (1), (2) のどちらでファセットデータを集約するか

手法 (1) に関してはノードへの接続時間、手法 (2) に関しては取得するタプルの量によって処理が遅くなることが考えられる。さらにネットワークが不安定な場合、手法 (1) ではノードへの毎回の接続に時間がかかるため、手法 (2) の方が処理が速いかもしれない。しかしながら手法 (2) では、タプルの取得に多くの時間を費やす可能性がある。そこで、ネットワークが安定な場合と不安定な場合に対し、手法 (1), (2) の処理速度を計測した。この実験については 6 節で述べる。

5. ユーザが選択する条件の検証

Faceted Search ではユーザは結果を見ながら複数回条件を指定し、絞込みを繰り返すことができる。したがって 4 節で示したように、1 つ目の条件を key として固定した際に多くのユーザが同じ条件を 1 つ目を選択し検索を繰り返すと、特定のノードに問合せが集中してしまうと考えられる。そこで我々は、実際にそのようなことがあるか、それはどのような場合におこるかを調べたいと考え、ユーザテストを行った。

5.1 実験概要

この実験では Flamenco Search のアプリケーション (図 1) を利用し、5 人のユーザに対して以下の 2 種類の実験を行った。

◆ 実験 (1): 興味のある人物を検索

実験 (1) ではユーザにファセットを自由にクリックしてもらい、その検索の中で興味をもった人物が現れた時点で終了してもらった。この実験では、目的がないユーザが Faceted Search でどのような絞込みを行うか、つまり複数のユーザが同じような検索の順序をたどるのか、全く違う流れで検索を行うのかを検証した。

◆ 実験 (2): ある条件に合う人物を検索

実験 (2) ではユーザに事前に伝えた条件を基に検索を行ってもらった。条件を提示することで、ユーザが明確な目的を持っている状態を作り出した。この実験で提示した条件は以下の 6 つである。

- 1) 20, 30 年代前に理科に関する賞を取った男性
- 2) 受賞当時の年齢は 80 歳前後で、既に亡くなっている
- 3) アメリカ人またはフランス人
- 4) その年に同じ賞を取った人が 2 人いる
- 5) 名前には C と J が入っている
- 6) 眼鏡をかけている

これらの条件を満たす人物は 1987 年にノーベル化学賞を取った C.J ペターゼンである。この実験では、求めたいデータがあり、そのデータの情報を知っているユーザが Faceted Search でどのように絞込みを行うかを調べた。

実験 (1), (2) のどちらにおいても、複数のユーザで 1 つ目を選択する条件がどの程度共通するかを調べた。さらに、1 つ目の条件を指定したまま検索を繰り返す回数を調べ、1 つ目の条件を key にもつノードに複数回問い合わせるかを検証した。

5.2 実験結果と考察

実験 (1) の結果を表 3 に示す。

条件	ユーザ数 (人)	(繰り返し利用した回数, ユーザ数)
country:Japan	2	(5,1), (2,1)
country:Costarica	1	(1,1)
country:France	1	(2,1)
gender:female	3	(13,1), (3,1), (5,1)
gender:male	1	(1,1)

year:2000s	2	(4,1), (2,1)
year:2004	1	(2,1)
prize:peace	2	(2,1), (2,1)
prize:literature	1	(11,1)

表 3 興味のある人物を検索した際に選択されたファセット

表 3 は 1 つ目に選択された条件とそのユーザ数、さらにその条件を繰り返し利用した回数とそのユーザ数を示している。例えば 1 行目の場合、`country:Japan` を 1 つ目に選択したユーザが 2 人おり、その内その条件を用いて絞込みを 5 回行ったユーザが 1 人、2 回絞込みを行ったユーザが 1 人いたことを示す。表 3 より条件 `country:Japan` や `year:2000s` の選択回数が多いことから、目的が明確でない場合、ユーザは自分自身に関連する条件を選択しやすと言える。また条件を繰り返し利用した回数結果から、1 つ目の条件を選択したままさらに絞込みを行うユーザが多いことが分かる。

次に実験 (2) の結果を表 4 に示す。

条件	ユーザ数 (人)	(繰り返し利用した回数, ユーザ数 (人))
gender:male	5	(3,2), (6,1), (13,1), (17,1)
country:France	1	(3,1)
year:1990s	1	(1,1)
year:1980s	1 (同じユーザが 3 回)	(1,1), (1,1), (2,1)
year:1970s	1 (同じユーザが 3 回)	(1,1), (1,1), (5,1)
year:1970	1 (同じユーザが 2 回)	(1,1), (1,1)
year:1971	1 (同じユーザが 2 回)	(1,1), (1,1)
year:1972	1 (同じユーザが 2 回)	(1,1), (1,1)
year:1973~1979	1	各 (1,1)
prize:chemistry	1	(5,1)
prize:physics	1	(2,1)

表 4 条件を満たすデータを検索した際に選択されたファセット

表 4 は 1 つ目に選択された条件とそのユーザ数、さらにその条件を繰り返し利用した回数とそのユーザ数を示している。表 4 より、条件 `gender:male` の選択回数が多いことが分かる。これは 6 つの条件のうち「男性」という確実な条件から絞込みを行った

ためである。また、`country:France` が `country:United States of America` より該当件数が少なかったため、`country:France` で検索したユーザもいた。さらに繰り返した回数から、実験 (1) と同じように 1 つ目に選択した条件を複数回利用するユーザが多いことが分かる。

実験 (1), (2) の結果より、1 つ目に選択されやすい条件があること、及びその条件が複数回繰り返し利用されることが分かる。実験 (1) のように目的がない場合には、自分に関係のあるデータを探すことが多い。そして 1 回目で絞り込まれたデータに対して、さらに条件を選択してどのようなデータがあるか調べる。また実験 (2) のように明確な目的がある場合 (例えばあるニュースで話題になった人物を検索したいなど)、特定の条件が集中して選択される。例えば `Flamenco Search` では、2010 年のノーベル賞化学賞を日本人が受賞したため、多くの日本人ユーザがその人を検索するかもしれない。その際、`country:Japan` や `prize:chemistry` の条件が 1 つ目の条件として選択されることが多くなり、その条件を `key` にもつノードに問合せが集中してしまう。さらにそれらの条件に加え、`year:2010` などの条件を追加し検索を繰り返すと、同じノード上で何度も処理が行われるため負荷がかかる。そこで、条件が複数になった場合にはその中から `key` とする条件をランダムに抽出し、問合せごとに `key` を変えることで、検索を進めても特定のノードに処理が集中しないようにすることが必要となる。

6. P2P における Faceted Search に関する検証

4 節で述べたように、効率的な `Faceted Search` を実現するためにはファセットの集約をノード側とクライアント側のどちらかで行うかべきか検証する必要がある。本節では、3.3 項で提示された 2 つの手法による処理時間を求め、どちらが有効であるか比較を行った。また、ネットワークの状態によって有効な手法が異なるかを調べるため、ネットワークが安定な場合と不安定な場合の 2 つの場合を想定した。

6.1 実験環境

本稿の実験では対象データ 10,000 件、ファセット名 10 種類、ファセット値 20 種類で生成された 900,000 タプルを対象に手法 (1), (2) の処理時間を比較した。これらの手法に対して、先行研究では 2 回の絞込みを行い、その合計処理時間で効率性を比較している。しかしながら 5 節の実験結果より、`Faceted Search` による問合せでは 1 つ目の条件を固定したまま、複数回検索を繰り返すユーザが多いことが分かっている。したがって、1 つ目の条件で複数回絞込みを行った場合の処理時間を比較する必要があるため、本稿では 2 つ目の絞込み条件を 3 種類用意し、ユーザが 2 つ目の条件を 3 回指定し直すことを想定した。さらに合計処理時間だけでなく、1 つ 1 つの絞込み時間が重要であると考え、それぞれの絞込みの平均処理時間を求め比較を行った。その際、選択を繰り返すにつれデータの選択率が減少するようにした。これは条件の選択

を行うごとに、対象データ数を減らすと仮定したためである。また3節で述べたように条件を key にして分散するため、その条件によって1台のノードで管理するタプル数が異なる。そこで、1回目の絞込みに用いる条件は対象データの選択率ごとに用意し、処理時間の比較を行った。そして、これらの4回の絞込みを1セットとし、1セットを10回繰り返した際のそれぞれの検索の平均時間を求めた。

さらに我々は、どちらの手法が有効であるかだけでなく、どの処理に時間がかかるかを検証した。そこで手法(1)では、ノードへの1回目の接続、1回目の検索、結果表示、2回目の接続、2回目の検索、結果表示、3回目の接続、3回目の検索、結果表示、4回目の接続、4回目の検索、結果表示のそれぞれの時間を計測した。また手法(2)では、ノードへの1回目の接続、クライアント側のDBへのデータの挿入、1回目の検索、結果表示、2回目の検索、結果表示、3回目の検索、結果表示、4回目の検索、結果表示のそれぞれの時間を計測した。ネットワーク環境には有線LANと無線LAN(b-mobileを使用)の2種類を用意し、それぞれ手法(1)、(2)の結果を比較した。

6.2 実験結果

クライアントが有線LANで接続されている際の各手法における対象データの選択率別の処理時間を示した結果を図4、図6の(a)に示す。

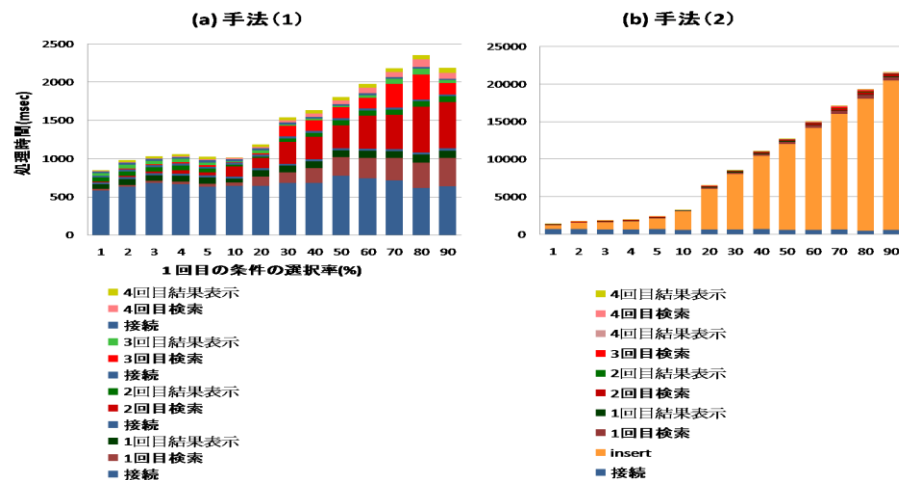


図4 有線LANの場合の処理時間の比較

図4の(a)は手法(1)の4回の絞込みにおけるそれぞれの接続、検索、結果表示の平均処理時間を、図4の(b)は手法(2)の接続時間、データの挿入、4回の絞込みにお

けるそれぞれの検索、結果表示の平均処理時間を示している。図6の(a)では、4回の絞込みを1セットとして検索した際にかかる合計処理時間の結果である。図4の(a)より、手法(1)は1回目のノードへの接続に多くの時間がかかっている。これは、1回目の接続ではノードへのルーティングに時間がかかるが、2回目以降はキャッシュされたノードに接続することで接続が速くなるためである。一方、手法(2)はタプル取得に多くの時間がかかり、さらにデータの選択率が高くなるにつれてその時間は増大している。選択率1%では約900タプルであるが、90%では約81000タプルまで増加するため、タプルの挿入時間が大幅に増える。手法(1)の2回目以降の接続時間が速いため手法(2)の2回目以降の処理時間と大きな差がない、さらに手法(2)のタプル取得時間がタプル数に比例して増大するという点から、2つの手法の合計処理時間の差は選択率が高くなるにつれて大きくなる(図6の(a))。図6の(a)より有線LANの場合、ノード側で全ての処理を行う手法(1)の方が効率的であると分かる。

次にクライアントが無線LANで接続されている際の各手法における対象データの選択率別の処理時間を示した結果を図5、図6の(b)に示す。

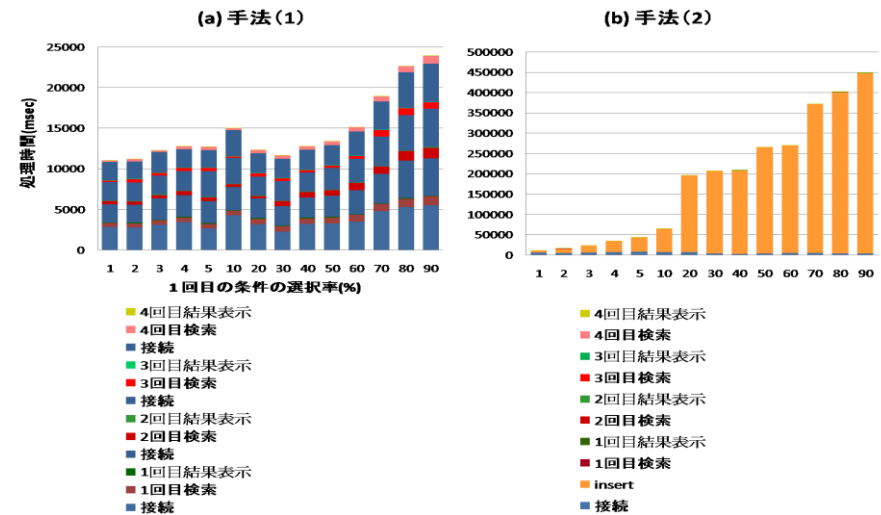


図5 無線LANの場合の処理時間の比較

図5の(a)は手法(1)の4回の絞込みにおけるそれぞれの接続、検索、結果表示の平均処理時間を、図5の(b)は手法(2)の接続時間、データの挿入、4回の絞込みにおけるそれぞれの検索、結果表示の平均処理時間を示している。図6の(b)では、4回の

絞込みを1セットとして検索した際にかかる合計処理時間の比較を行っている。図5の(a)より、手法(1)は毎回のノードへの接続が3秒ほどかかっており、図4の(a)の比べると処理時間全体が大きくなっている。このように無線LANの場合、ノードへの接続時間の増大から手法(2)の方が処理時間が速くなると思われた。しかしながら図5の(b)より、手法(2)はタプルの取得に大幅な時間を費やしている。これは、手法(1)のノードへの接続時間よりはるかに大きく、選択率が高くなるにつれ増大している。したがって手法(1)は毎回のノードへの接続に時間がかかるが、手法(2)のタプル取得時間がさらに大きな時間を費やすことから、2つの手法の合計処理時間の差は選択率が高くなるにつれて大きくなる(図6の(b))。図6の(b)より無線LANの場合、ノード側で全ての処理を行う手法(1)の方が効率的であると分かる。

以上より、ネットワーク環境に関わらず手法(1)の方が有効であることから、どちらの場合もノード側で集約を行い、クライアント側は結果のみ取得することにする。

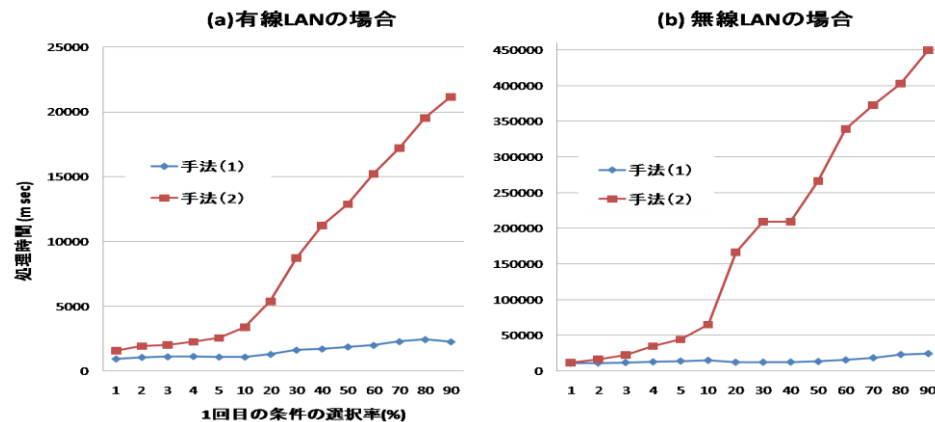


図6 有線LANと無線LANの場合の各手法の処理時間の比較

7. まとめと今後の課題

本稿では、P2P環境におけるFaceted Searchの実現のための手法を提案した。また、Faceted Searchインタフェースに関するユーザテストを行い、条件のアクセス集中について述べ、さらに提案した手法に対してP2P環境におけるFaceted Searchの処理時間を評価した。ユーザテストでは、Faceted Searchを利用した際にユーザがどのような条件から絞込みを行うか、さらにその条件を用いて絞込みを何回繰り返すかを検証した。

その際Faceted Searchを利用する際のユーザの状態を考え、目的があいまいな場合と明確な場合の2種類の状態を想定した。その結果、どちらの場合も複数のユーザが同じ条件を選択すること、さらにその条件を選択したまま複数回絞込みを繰り返すことが分かった。したがって特定のノードへのアクセス集中が懸念されることから、今後は絞込み時のノードの負荷分散について考察を進める必要がある。またP2P環境におけるFaceted Searchの処理時間の評価に関しては本稿で提案している2つの手法での処理時間を求め、どちらの手法がより効率的なFaceted Searchを実現できるか比較を行った。その際、ネットワーク環境による結果の違いを調べるため、ネットワークが安定と不安定の2つの場合を想定し実験を行った。今後は、その結果を基にP2P上での実装を行い、さらに、分散管理されるデータ量についても考察を進めていきたい。

参考文献

- 1) R. Huebsch, J. Hellerstein, N. Lanham, B. T. Loo, S. Shenker, and I. Stoica : Querying the Internet with PIER, In Proc. of the 29th International Conference on Very Large Data Bases, pp.321-332(2003).
- 2) C. Zheng, G. Shen, S. Li, and S. Shenker: Distributed Segment Tree: Support Range Query and Cover Query over DHT, In Proc. of the Fifth International Workshop on Peer-to-Peer Systems(IPTPS)(2006).
- 3) W. Rao, A. W. Fu, L. Chen, and H. Chen: STAIRS: Towards Efficient Full-Text Filtering and Dissemination in a DHT Environment, In Proc. of the 25rd IEEE International Conference on Data Engineering(ICDE'09), pp.198-209(2009).
- 4) 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).
- 5) 齋藤真衣, 渡辺知恵美: P2P環境におけるRDFデータを対象にしたFaceted Searchの実現, 第2回データ工学と情報マネジメントに関するフォーラム(DEIM2010), C9-4(2010).
- 6) C. Watanabe, M. Saito: Data Assignment for Faceted Search in P2P Environment, The 2010 International Conference on Parallel and Distributed Processing Techniques and Applications, (PDPTA'10), (2010).
- 7) Flamenco Search: UC Berkeley School of Information The Flamenco Search Interface Project <http://flamenco.berkeley.edu/index.html>.
- 8) S. B. Roy, H. Wang, G. Das, U. Nambiar, and M. Mohania : Minimum Effort Driven Dynamic Faceted Search in Structured Databases, In Proc. of the 17th ACM Conference on Informational and Knowledge Management, pp.13-22(2008).
- 9) XMLデータに対するファセットナビゲーションのためのフレームワークFoXの提案, 第1回データ工学と情報マネジメントに関するフォーラム(DEIM2009), B7-6(2009).
- 10) D. Abadi, A. Marcus, S. Madden, and K. Hollernbach.: Scalable Semantic Web Data Management Using Vertical Partitioning, In Proc. of the 33rd International Conference on Very Large Data Bases, pp.411-422(2007).