

# 多様な視点の共有を可能にする自律分散型コンテンツ参照方式

宮下 山斗<sup>†</sup> 石川 尋代<sup>†,††</sup> 寺岡 文男<sup>†</sup> 金子 晋文<sup>†,††</sup>

<sup>†</sup> 慶應義塾大学理工学部 〒 223-8522 横浜市港北区日吉 3-14-1

<sup>††</sup> 慶應義塾大学デジタルメディア・コンテンツ統合研究センター 〒 223-8523 横浜市港北区日吉本町 2-1-1

E-mail: mine@inl.ics.keio.ac.jp, hiroyo@dmc.keio.ac.jp, {tera,kaneko}@ics.keio.ac.jp

あらまし インターネット上のウェブコンテンツに対するユーザの視点は様々であり、それらを共有できればコンテンツに対する様々な切り口を得られると同時に、周囲の関連コンテンツを発見するための大きな手がかりとなる。本論文では他者と視点を共有するための分散型コンテンツ参照方式「カタログシステム」を提案する。カタログシステムでは各ユーザがファイル間の関係性を「カタログ」に記述し、ファイルそのものから分離して作成・管理することにより、ユーザが所有していないファイルであってもそれらの関係性を記述可能にする。またファイル間の関係性の記述には自然言語等を使用せず、有向グラフのみを用いることによって関係性に多様な解釈の可能性を残す。カタログはファイルと同様に個人の資産であるとみなし、所有者が自律分散的に管理する。さらにファイルからそのファイルを掲載する全てのカタログを参照可能にし、有向グラフを辿ることで多様な視点を共有しながらファイルを閲覧していくことを可能にする。評価では本システムを用いたアプリケーションで有向グラフをインタラクティブに取得し、それらをリアルタイムに可視化できることを確認した。

キーワード 分散型コンテンツ参照, カタログ, 共有

Yamato MIYASHITA<sup>†</sup>, Hiroyo ISHIKAWA<sup>†,††</sup>, Fumio TERAOKA<sup>†</sup>, and Kunitake KANEKO<sup>†,††</sup>

<sup>†</sup> Faculty of Science and Technology, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, 223-8522

<sup>††</sup> Research Institute for Digital Media and Content, Keio University, 2-1-1 Hiyoshihoncho, Kohoku-ku, Yokohama, 223-8523

E-mail: mine@inl.ics.keio.ac.jp, hiroyo@dmc.keio.ac.jp, {tera,kaneko}@ics.keio.ac.jp

## 1. はじめに

インターネット上のテキストや画像、音声といったウェブコンテンツに対する利用者の視点は様々であり、関連するコンテンツの認識もまた様々である。例えばニュートンの画像ファイルに対して「物理学者」という視点を持つ利用者は、関連するコンテンツとしてアリストテレスやケプラー、アインシュタインなどに関するコンテンツを挙げると予想される。また同一のニュートンの画像ファイルに対して「愛猫家」という視点を持つ利用者も存在する。この利用者は村上春樹やリンカーン、チャーチルなどの愛猫家に関するコンテンツを、ニュートンの画像ファイルに関連するコンテンツとして挙げると推測できる。

このようなウェブコンテンツに対する多様な視点を他者と共有できれば、閲覧中のコンテンツに対する様々な切り口を得られると同時に、周囲の関連するコンテンツを発見するための大きな手がかりとなる。現在のインターネットではニュートンの画像に対して「愛猫家」という視点をもたない利用者は、村上春樹などのコンテンツとの関連性を得ることは困難である。そ

のような利用者は、ニュートンの画像ファイルに紐付けられた「愛猫家」というキーワードを偶然発見して、それをもとに再検索しなければ「愛猫家」に関するコンテンツを参照することはできない。このように現在のインターネットではコンテンツ間の関係性が明示されておらず、関連するコンテンツの発見は閲覧者の偶発性に委ねられている。

そこで本研究では図1に示すような分散型コンテンツ参照方式「カタログシステム」を提案する。本研究ではウェブコンテンツをテキストや画像、音声といったファイルそのものと、それらの関係性を表すものに分けて考える。ウェブページを記述するHTMLにはページ間の関係性を表すハイパーリンクが埋め込まれているが、そのほかにもテキストや画像、音声などの情報が混在している。本システムでは利用者がそれぞれの視点からファイル間の関係性を「カタログ」に記述し、ファイルそのものとは別に管理する。

カタログにファイル間の関係性を記述する際には自然言語等を用いず有向グラフのみを使用し、ファイル間の関係性に多様な解釈の可能性を残す。このようなカタログをユーザ間で共有

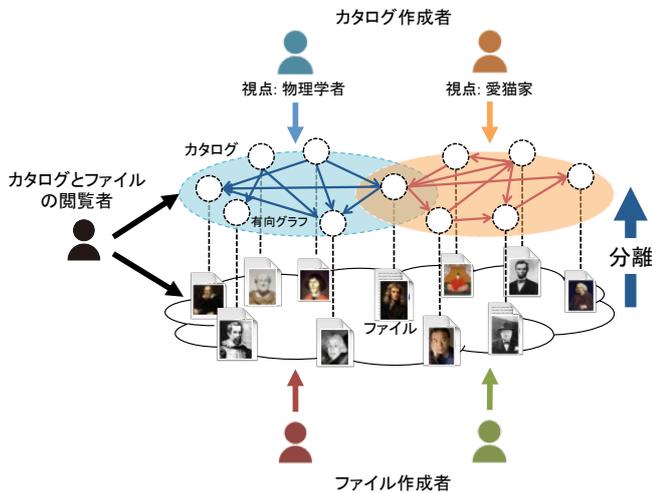


図 1 分散型コンテンツ参照方式「カタログシステム」

することによって、ユーザの解釈に広がりをもたせ、発展性のある多様な視点の共有を実現する。

本システムにおけるユーザは以下の 3 種類である。

- ファイルの作成者

ファイルの作成者はインターネット上のウェブコンテンツと同様に自由にファイルを作成し共有する。本研究ではファイルにはグローバルユニークな識別子が与えられるものとし、ファイルは所有者によって自律分散的に管理されていることを前提とする。

- カタログの作成者

カタログの作成者はファイルを選択して自由にカタログを作成し共有する。またカタログにはファイル同士の関係性だけでなく、カタログ同士の関係性やファイルとカタログの関係性も記述することができる。

- ファイルとカタログの閲覧者

ファイルの閲覧者はそのファイルを掲載している全てのカタログを取得できる。さらにそのカタログを構成する有向グラフを辿って関連するファイルを取得できる。ただしファイルとカタログの閲覧には認証と認可が必要である。

上記のユーザがカタログシステムを利用することを前提とし、本システムに対する要求事項とアプローチを以下に挙げる。

(1) 自由なファイル間の関係性の定義

一般的なファイルシステムではディレクトリによってファイル間の階層化された関係性を表現できるが、そのためにはシステムにユーザのアカウントが存在しなければならない。カタログシステムではファイルシステムのユーザ以外であっても自由にカタログを作成できるように、ファイルとカタログの管理基盤を分離する。

(2) カタログの資産性の確保

本研究ではファイルと同様にカタログも所有者の重要な資産であるとみなし、これらとともに「コンテンツ」と呼ぶ。本システムではカタログの資産性を確保するために、カタログは作成者自身が管理し所有する。

(3) 閲覧中のファイルに対する全ての視点の参照

閲覧中のファイルに対するあらゆる視点を網羅的に参照できることは重要である。本システムでは、ファイルとそれを掲載している全てのカタログの対応関係を分散して管理する。具体的にはそれらの対応関係は各ファイルを提供するドメインごとに分散して管理されている。これによってカタログ数の増加に対するスケーラビリティを確保しつつ、閲覧中のファイルを掲載する全てのカタログを参照可能にする。

(4) 異なる視点を区別した共有

カタログ数が増加するとそれらの作成者を選択して参照できることが重要になってくる。カタログは学者やアーティスト、知人といった様々なユーザによって作成されることが想定される。本システムではカタログの閲覧者がそれらを区別して参照できるように、作成者の情報を含むグローバルユニークな識別子をカタログに与えている。

本論文の構成は以下のとおりである。2 章では有向グラフを用いたファイル間の関係性の定義とカタログシステムの設計概要について述べる。3 章ではカタログシステムの実装について述べる。4 章ではカタログシステムを用いたアプリケーションの動作確認と、カタログの取得時間に関する評価について述べる。5 章ではカタログシステムの関連研究としてフォークソノミーについて述べる。6 章では結論と今後の課題について述べる。

## 2. 設 計

### 2.1 ファイル間の関係性の定義

本システムでは、ユーザがそれぞれの視点にもとづいてファイル間の関係性を「カタログ」に記述し他者と共有する。以下では 2.1.1 節で自然言語を用いたファイルに対する意味付けの問題点について述べ、2.1.2 節で有向グラフのみを用いたファイル間の関係性の定義について述べる。

#### 2.1.1 既存の意味付けの問題点

画像投稿サイトの Flickr [1] や、ソーシャルブックマークサイトの Del.icio.us [2] では、画像やウェブページに対してタグと呼ばれるメタデータが利用者の視点にもとづいて付与される。タグとはファイルの意味を自然言語で表現したメタデータであり、利用者が付与したタグの類似性をもってファイルが分類される。

このようなファイルへのタグの付与は利用者に対して先入観を与え、関連ファイルを発見する手がかりを失う可能性を有する。例えばニュートンの画像ファイルに対して「物理学者」というタグが付与された場合、利用者は「愛猫家」や「錬金術師」といったニュートンのその他の側面を見落としがちになる。

またタグには自然言語が用いられていることからファイルに対して曖昧な意味付けがなされる。具体的には、自然言語における多義語や類義語、短縮語などの曖昧性が指摘されている [5]。多義語の例では“mine”という単語は“私のもの”や“鉱山”といった複数の意味を有するため、タグのキーワードが一致したとしても異なる意味で付けられたタグが検索結果に反映される可能性がある。

さらにインターネット上のファイル間には様々な関係性が存

在し、それらをタグのみで記述することは困難である。たとえばニュートンの肖像画は複数描かれており、それらの画像ファイルを描かれた年代順に並べるといった表現はタグのみでは困難である。また様々な関係性ひとつひとつに対して専用の記述構造を用意し、それらの記述方式を一般ユーザが全て習得することは容易ではないと考えられる。

以上のような問題点から、本システムではタグのように自然言語等を用いた分類をせず、有向グラフのみを使用してファイル間の関係性を記述している。

### 2.1.2 有向グラフの利用

ファイル間の関係性を定義するにあたり、筆者らはグラフ理論に着想を得た。グラフ理論にはノード間の関係性に方向性をもたない無向グラフと、方向性をもつ有向グラフが存在する。無向グラフはファイル間の対等性の表現には適しているが、順序関係などの表現には無向グラフ以外の情報が必要になるため適していない。一方、有向グラフでは双方向の有向グラフを用いることでファイル間の対等性を表現することができ、順序関係等も有向グラフのみで表現することが可能である。

そこで本システムでは有向グラフを使用してファイル間の関係性を簡潔に記述する。有向グラフは始点となるコンテンツおよび終点となるコンテンツから構成されるものとする。また自然言語等を使用せず有向グラフのみを用いることによって、ファイル間の関係性に多様な解釈の可能性を残す。これによってファイルに対する画一的な意味付けを排除し、発展性のある多様な視点の共有を実現する。

カタログは有向グラフの組み合わせ法に応じて様々な関係性の表現が可能である。図2は筆者らが考える有向グラフを用いたカタログの例であり、以下のように“chain”、“view”、“flat”と名付けた。

#### (a) chain

ファイルの閲覧順序等を指定する表現を指す。たとえば複数のニュートンの肖像画の画像ファイルを描かれた年代順に並べるといった表現ができる。

#### (b) view

あるファイルを起点とし、それに関連するファイルを列挙する表現のことを指す。viewの関係性はHTMLのハイパーリンクによる関連文書の結びつけに類似している。

#### (c) flat

複数のファイルが双方向グラフによってメッシュ状に参照されておりそれぞれが対等な関係であることを表現する。flatは複数のファイルに同一のタグを付加することに類似している。

また本システムでは、ファイル間だけでなくカタログ間の関係性を定義する際にも有向グラフを使用し「カタログのカタログ」を作成することが可能である。カタログ間の関係性の定義によって階層化された関係性の表現などが可能となる。たとえば図3のように「哺乳類」や「爬虫類」という視点から作成されたカタログが存在したとする。これらのカタログを双方向の有向グラフで結ぶことによって「生物」という視点のカタログを作成することができる。

上記の「生物」という視点はあくまでカタログの閲覧者が解

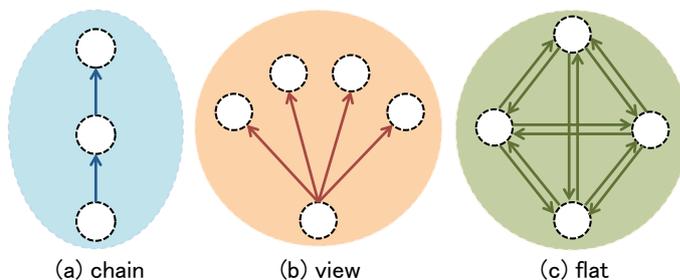


図2 有向グラフを用いたカタログの例

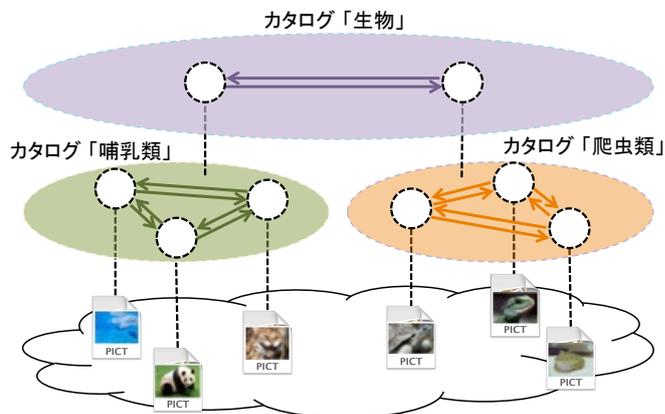


図3 カatalogを構成要素にもつCatalog

釈するものであり、自然言語による説明がなされているわけではない。たとえば、図3では「生物」の代わりに「脊椎動物」と解釈するカタログの閲覧者も存在する。

## 2.2 設計指針

### (1) 自由なファイル間の関係性の定義

本システムではカタログ作成者が所有していないファイルであっても、自由にそれらの関係性を記述できるようにファイルとカタログの管理基盤を分離して設計している。一般的なファイルシステムではディレクトリによって、階層化されたファイル間の関係性を表現することが可能である。しかしながら、ファイルシステムではディレクトリやファイルに対してユーザが書き込み権限を持っていなければそれらを移動することはできない。

### (2) カatalogの資産性の確保

現在のウェブでは個人の資産であるファイルは所有者によって自律分散的に管理されている。本研究ではCatalogも同様に個人の資産であるとみなし所有者が自律分散的に管理する。ただしユーザがメールサーバを運用することが技術的に困難な場合にメールのホスティングサービスを利用するように、Catalogの管理をサービス提供者に委託することは認められるものとする。

### (3) 閲覧中のファイルを掲載する全てのCatalogの参照

閲覧中のファイルに対するあらゆる視点を網羅的に参照することは重要である。本システムでは、ファイルとそれを掲載している全てのCatalogの対応関係を分散して管理する。具体的にはそれらの対応関係は各ファイルを提供するドメインごとに分散して管理されている。これによってCatalog数の増加に

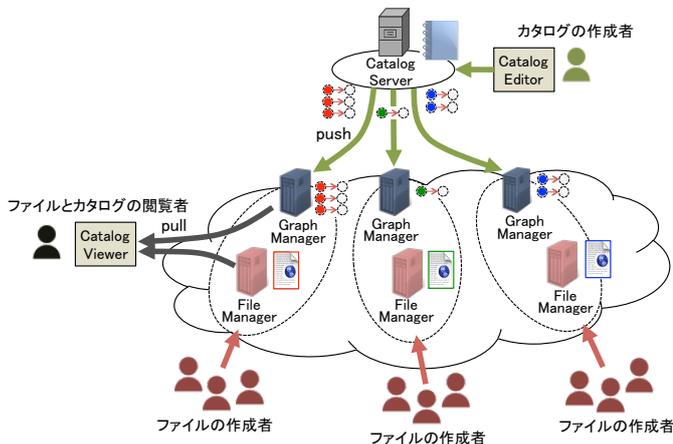


図 4 システムの構成要素

対するスケーラビリティを確保しつつ、閲覧中のファイルを掲載する全てのカタログを参照可能にする。

#### (4) 異なるカタログを区別して共有

カタログ数が増加するとそれらの作成者を選択して参照できることが重要になってくる。カタログは学者やアーティスト、知人といった様々なユーザによって作成されることが想定される。本システムではカタログの閲覧者がそれらを区別して参照できるように、作成者の情報を含むグローバルユニークな識別子をカタログに与えている。

#### 2.3 システムの概要と構成要素

カタログシステムは図 4 のように Catalog Editor, Catalog Server, File Manager, Graph Manager, Catalog Viewer の 5 つのモジュールから構成される。

本研究では、ファイルは 128 bit の File ID でグローバルに識別されることを前提とする。またそれらのファイルは所有者が契約する File Manager が自律分散的に管理するものとする。

カタログシステムでは、(i) カatalogの所有者が Catalog Editor と呼ばれるアプリケーションを用いてカタログを編集する。(ii) 編集されたカタログは Catalog Server と呼ばれるカタログ所有者が管理するサーバへ保存される。その後、(iii) Catalog Server はカタログを構成する有向グラフを Graph Manager へ分散して保存する。(iv) カatalogとファイルの閲覧者は、Catalog Viewer と呼ばれるアプリケーションを用いてファイルとカタログを取得する。

以下では各モジュールの詳細について述べる。

##### • Catalog Editor

Catalog Editor はエンドホストのマシン上で動作するカタログ編集用のアプリケーションである。筆者らは図 5 のような GUI アプリケーションを作成した。今回作成した Catalog Editor は博物館が所有するデジタル収蔵品の画像ファイルを対象として、それらを File Manager から取得し有向グラフで結んでカタログを編集することができる。

##### • Catalog Server

Catalog Server には Catalog Editor で編集されたカタログが保存される。カタログの資産性を確保するために、カ

タログの作成者は自身の管理下にある Catalog Server でカタログを管理し所有するものとする。ただしサーバ維持費や技術的な理由からサーバ管理が困難であると判断し、かつ自身のカタログに資産性を見出さない場合はウェブのホスティングサービスのように Catalog Server の管理を他者に委託することも想定する。

##### • File Manager

本システムではユーザのファイルへのアクセス制御を File Manager で行うことを前提としている。File Manager はストレージサービスのプロバイダが管理しており、ファイル所有者は自身の契約している File Manager を介してファイルにアクセスし編集するものとする。

##### • Graph Manager

Graph Manager には Catalog Server で公開を許可されたカタログの有向グラフが分散して保存されており、それらの有向グラフは一般ユーザが自由に参照できる。

またユーザが閲覧しているファイルからそのファイルを掲載している全てのカタログを参照可能にするため、ファイルとそれを掲載しているカタログの対応関係を Graph Manager で管理する。具体的には有向グラフに対して Global Catalog ID と呼ばれる 128 bit のグローバルユニークな識別子を与え、有向グラフの始点および終点となるファイルとその有向グラフを含むカタログの対応を Graph Manager で管理する。

本システムではストレージサービスのプロバイダは File Manager と同様に Graph Manager も運用する。カタログシステムでは閲覧数の多いファイルほどカタログに掲載されやすく、また多くのカタログに掲載されているファイルほど閲覧されやすくなると考えられるため、カタログシステムを運用していくことは自身のファイルを広く公開したいユーザにとって有益になるといえる。そこで Graph Manager の管理負担を考慮して、カタログシステムではグラフの始点および終点のファイルを管理しているプロバイダが運用する Graph Manager 到有向グラフを分散して保存する。

##### • Catalog Viewer

Catalog Viewer は有向グラフで表されたファイル間の関係性を可視化するためのアプリケーションである。筆者らは図 6 に示すようなデジタル収蔵品の関係性を、取得したカタログにもとづいてリアルタイムに可視化し、ファイルを辿っていくための GUI アプリケーションを作成した。Catalog Viewer では有向グラフの色でカタログの違いを区別しており、球体はひとつひとつのファイルを表す。

## 3. 実装

前述した設計にもとづきカタログシステムを実装した。3.1 節ではファイル、カタログ、有向グラフに付与したグローバルユニークな識別子について述べる。また、3.2 節では各種 API と、カタログの生成時と、読み込み時の動作シーケンスについて述べる。

### 3.1 各種識別子

カタログシステムを実装するにあたりファイル、カタログ、

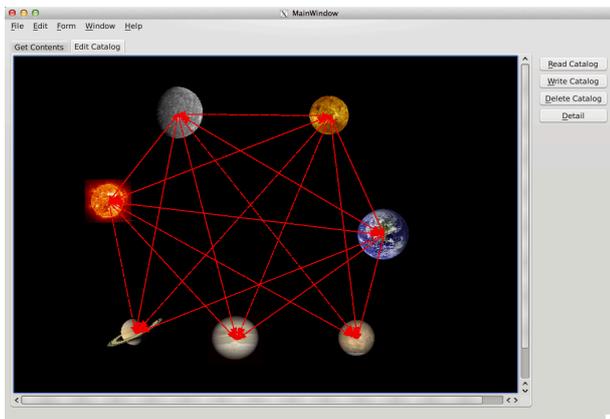


図 5 Catalog Editor

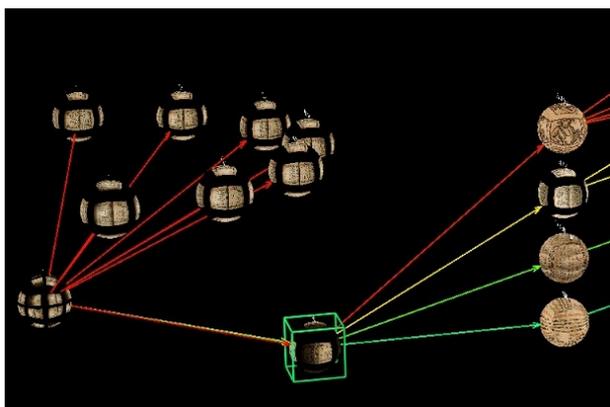


図 6 Catalog Viewer

有向グラフに対して、それぞれ File ID, Global Catalog ID, Graph ID と呼ばれるグローバルユニークな識別子を与えた。以下でその詳細について述べる。

#### ( a ) File ID

本システムでは簡単化のため、ファイルをグローバルに識別するための File ID を 128 bit とし、図 7 にそのフォーマットを示す。File ID は Global Catalog ID と区別するために最上位ビットを 0 とする。また上位 32 bit は File Manager をグローバルに識別するための File Manager ID フィールドであり、続く 64 bit は File Manager 内でファイルを識別するための Content ID フィールドである。末尾の 32 bit はファイルのバージョンを管理するための Version ID フィールドである。

#### ( b ) Global Catalog ID

Global Catalog ID はカタログをグローバルに識別することのできる 128 bit の識別子である。Global Catalog ID のフォーマットを図 7 に示す。先頭の Catalog Server ID は 48 bit の識別子であり世界中の Catalog Server をグローバルに識別する。2 目目の Owner ID は 16 bit の識別子であり、Catalog Server にアカウントをもつカタログ作成者を約 6 万人まで識別することができる。3 目目の Local Catalog ID は 32 bit の識別子であり、特定のカタログ作成者が所有するカタログを識別することができる。Local Catalog ID によって各 Owner ID につき約 42 億のカタログを識別することが可能となる。4 目目の Catalog Version ID は 32 bit の識別子であり、Catalog

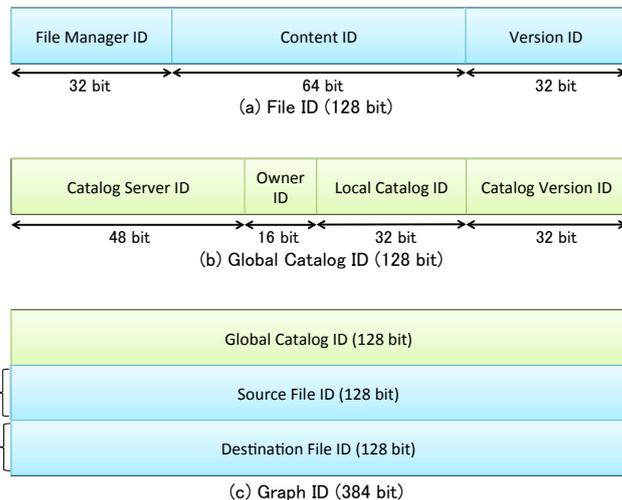


図 7 File ID, Global Catalog ID, Graph ID のフォーマット

Version ID はカタログが更新される度にインクリメントされカタログの編集履歴を管理するために用いられる。

#### ( c ) Graph ID

Graph ID は有向グラフをグローバルに識別することのできる 384 bit の識別子である。Graph ID のフォーマットを図 7 に示す。Graph ID は、その有向グラフが属しているカタログの Global Catalog ID と、有向グラフの始点および終点にあたるファイルの File ID によって構成される。

### 3.2 各種 API と動作シーケンス

今回のカタログシステムの試作では実装上の都合により、Catalog Editor から有向グラフおよびファイルを収集する機能を分離し、Collector と呼ぶモジュールとして実装した。

またカタログシステムを構築するにあたって Catalog Server API, Catalog Client API, Catalog Editor API, Catalog Viewer API, Collector API を作成した。Catalog Server API および Catalog Client API はカタログの編集者がカタログの書き込み、読み込み、削除をする際に用いられる。Catalog Editor API, Catalog Viewer API は上述したものより高レベルな API であり、アプリケーション作成者が Catalog Editor や Catalog Viewer を効率的に開発可能にするために作成した。Collector API は有向グラフを Graph Manager から収集するための API であり、設計上は Catalog Viewer が動作するマシン上に実装されるべき低レベルな API である。

以下ではこれらの API を用いたカタログの生成と読み込みの動作について述べる。

#### 3.2.1 カatalog生成時の動作

カタログ生成時の動作シーケンスを図 8 に示す。カタログを生成するには、まず Catalog Editor が編集されたカタログを Catalog Server へ送信する。Catalog Server はカタログを構成する有向グラフの Graph ID を Catalog Database へ保存する。次に各 Graph ID をそれぞれ適切な Graph Manager へ送信する。Graph ID を受信した各 Graph Manager はそれらを Graphs Database へ保存する。

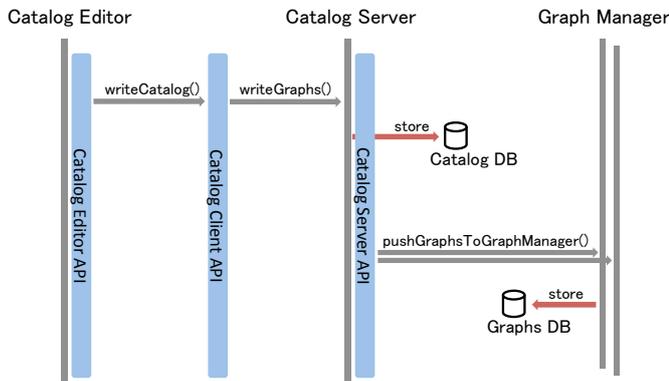


図 8 カタログ生成時の動作

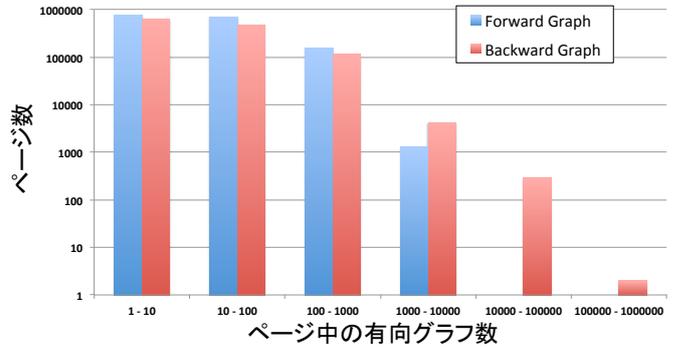


図 10 有向グラフの分布

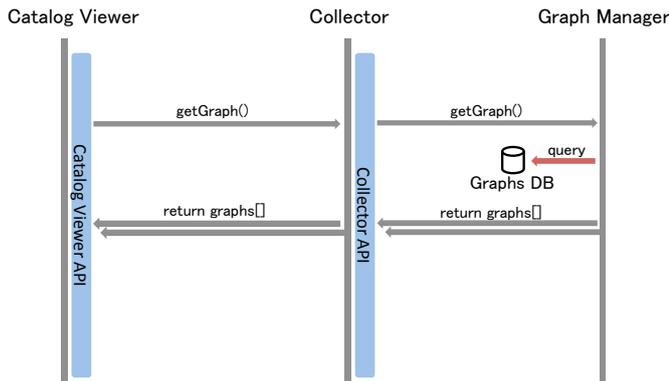


図 9 カタログ読み込み時の動作

### 3.2.2 カタログ読み込み時の動作

Catalog Viewer アプリケーションがカタログを読み込む際の動作シーケンスを図 9 に示す。カタログの読み込み際には、まず Catalog Viewer が閲覧中のファイルを開始点とする有向グラフを取得するために、File ID を Collector へ送信する。File ID を受信した Collector は適切な Graph Manager へその File ID を送信する。File ID を受信した Graph Manager は Graphs Database からそのファイルを開始点とする有向グラフを検索し、有向グラフの集合を Collector へ送信する。Collector は有向グラフの集合を受信すると Catalog Viewer へそれらを送信する。

## 4. 評価

本システムでは Catalog Viewer のような、ユーザがインタラクティブにカタログを取得し、カタログの有向グラフをリアルタイムに可視化するというアプリケーションの要求を想定している。そうしたアプリケーションの動作確認を行い、Wikipedia のリンクデータから作成したカタログの取得時間の性能を評価した。

Wikipedia の記事中のリンクの集合は編集者たちの視点が統合された一つのカタログとみなすことができる。Wikipedia では、記事の編集者たちが記事中のキーワードからそのキーワードに関して述べたページリンクを張ることができる。記事中にキーワードが存在しない場合であっても関連項目の節が設けられ、その記事に関連するページリンクを張ることができる。

このように Wikipedia のリンクは記事間の関係性を表しておりカタログの有向グラフと類似した働きをする。

本評価では Wikipedia のリンクを有向グラフに変換した後、Graph Manager へ保存し、Catalog Viewer が有向グラフを取得する際の取得時間を測定した。

### 4.1 評価環境

#### 4.1.1 評価用カタログの生成

評価用カタログを生成するにあたり、日本語 Wikipedia のリンクデータを <http://dumps.wikimedia.org/jawiki/> からダウンロードし、61,119,925 本の有向グラフを作成した。図 2 に示すような“view”のように、同一ページを起点とする有向グラフの集合をそれぞれ一つのカタログとみなし Global Catalog ID を割り当てた。

作成した有向グラフは図 10 に示すような、べき乗法則の分布に従っていた。べき乗法則の分布は図書館の本の貸出数などにみられる分布であり、利用者の人気によって貸出数が一部の本に偏り様に分布しないことが特徴である。カタログの場合も同様に、ユーザからの知名度が高いファイルは多くのユーザに閲覧されるためカタログに掲載されやすくなり、逆に知名度の低いファイルはカタログに掲載されにくくなると考えられる。

また有向グラフはひとつのファイルに着目すると、Forward Graph と Backward Graph の 2 つに分類できる。Forward Graph は対象のファイルから別のファイルへ結合している有向グラフであり、Backward Graph は別のファイルから対象のファイルへ結合している有向グラフである。各総数はそれぞれ 1,635,642 本、1,248,182 本となり、それらを 8 台の Graph Manager へ均等に分散した。

#### 4.1.2 ネットワーク環境

本評価は図 11 のようなネットワークトポロジで行った。8 台の Graph Manager を 4 台、2 台、2 台の 3 つのグループに分け、異なるサブネットに分散して配置した。また Collector と Catalog Viewer をサブネット 1 に配置した。サブネット 1 の Collector からサブネット 2 の Graph Manager までの往復遅延は約 42 msec であり、サブネット 3 の Graph Manager までの往復遅延は約 132 msec であった。

Graph Manager はそれぞれ仮想マシン上で動作させた。また Collector と Catalog Viewer はそれぞれ物理マシン上で動作させた。

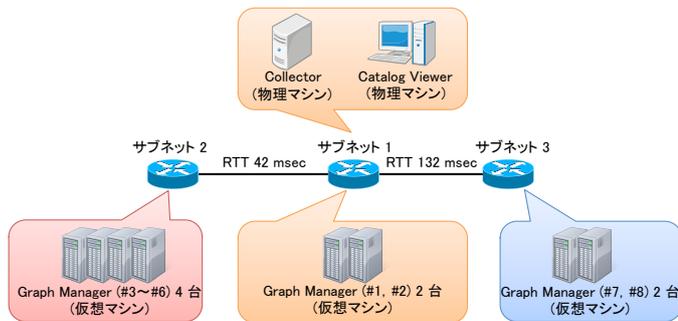


図 11 ネットワークトポロジ

#### 4.2 取得した有向グラフ数による取得時間の性能の評価

Catalog Viewer が 15,000 個の File ID を指定し、それらの File ID を始点とするような有向グラフを Collector に要求した。また Collector から取得した有向グラフの数と、Collector に要求してから全ての有向グラフを取得するまでの時間を計測した。上記の測定を 5 回行い平均を計算した。

評価結果は図 12 のようになった。グラフの横軸は取得した有向グラフの数、縦軸は全有向グラフの取得時間を表し、各系列は 8 台の Graph Manager を表す。

ユーザがインタラクティブにカタログを取得し、その有向グラフをリアルタイムに可視化するアプリケーションでは、画面に描画可能な有向グラフは数十までと想定される。図 12 から、取得した有向グラフ数が 20 以下の範囲では取得時間が 0.5 秒以下であることが分かる。このことから可視化アプリケーションにおいて有向グラフを取得する際の遅延はユーザが許容可能な範囲であり、アプリケーションがインタラクティブに動作することが示された。

取得する有向グラフがアプリケーションの画面に描画可能な範囲を超えるような場合はその数を制限する必要がある。そこで Graph Manager から有向グラフを検索した際の結果に優先順位をもたせてユーザへ提供する機能が重要となる。たとえば始点と終点のファイルが一致する有向グラフの数の統計情報を用いて、ユーザに関連度の高いファイルを優先的に提示する機能が有用である。またユーザがどの有向グラフを選択して次のファイルを参照したかといった履歴情報を用いた機能も有用となる。

### 5. 関連研究

カタログシステムのような、ファイルに対する様々な視点を共有するシステムの例としてフォークソノミーが挙げられる。フォークソノミー (Folksonomy) は folks (人々) と taxonomy (分類学) を組み合わせた造語であり、ユーザが画像や動画などのファイルに対してタグと呼ばれるメタデータを自由に付加する分類体系である。従来の専門家による分類体系の構築とは異なり、ユーザが誰でも分類作業できることが特徴である。

フォークソノミーにおける情報共有の方法は 2 つに分けられる [3]。ファイルに付加されたタグ情報のみを公開し共有する Simple tagging と、どのユーザがタグを付加したのかなどの情報を集約して共有する Collaborative tagging である。カタロ

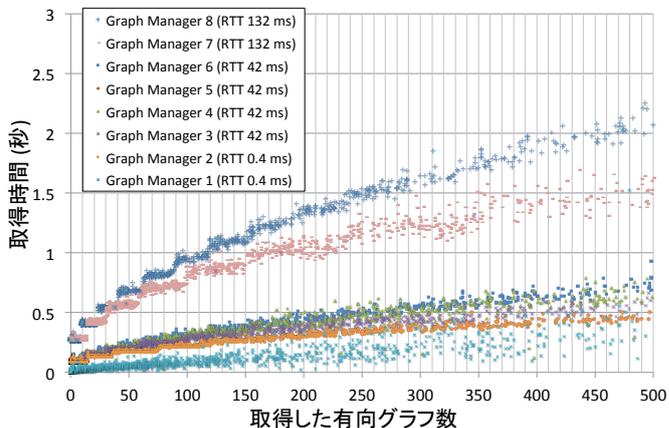


図 12 取得有向グラフ数と取得時間の分布

グシステムではカタログにオーナー情報が付与されているため Collaborative tagging に類似しているといえる。

フォークソノミーにおける最も基本的な検索方法は「タグ検索」である。タグ検索はユーザがキーワードを入力し、システムが入力されたキーワードに一致するタグが付加されたファイルを提示する検索手法である。このタグ検索では少なくともユーザが自然言語化されたクエリを作成する必要がある。自然言語は多義語などの曖昧性を有するため、タグのキーワードが一致したとしても異なる意味で付けられたタグが検索結果に反映される可能性がある。カタログシステムではタグ検索のような自然言語を用いた関連ファイルの検索を行わずに、ファイルを掲載している全てのカタログを参照可能にすることで関連ファイルを利用者に提示する。

また「タグクラウド」は頻繁に使用されるタグや、時事関連の関心が高まっているようなタグを提示することで、検索に用いるタグの選択を支援するものである。たとえば del.icio.us ではファイルに付加されたタグを幾つか表示して、文字の大きさと重要性を区別する。

タグクラウドでは同様の意味を持つタグがまったく別のものとして扱われてしまうためタグ検索が冗長なものになってしまう。そこで類似しているタグをまとめてグループ化し階層的に提示するタグクラスタリングという検索手法が存在する。Begelman ら [4] はスペクトラルクラスタリング (spectral clustering) という手法でタグのクラスタリングを行い、自動的にタグをグループ化し検索の効率を高める方法を提案している。

### 6. おわりに

本論文ではユーザのファイルに対する様々な視点を共有するための「カタログシステム」の提案とその動作確認について述べた。様々な視点にもとづくファイル間の関係性をファイル閲覧者に対して明示的に提示するためには、ファイルの本体から切り離してそれらの関係性を記述することが重要である。本研究ではファイル間の関係性をカタログとして表現し、カタログの作成者自身が自律分散的に管理してその資産性を確保した。ファイル間の関係性の表現には自然言語等を使用せず、有向グ

ラフのみを用いることによってファイル間の関係性に多様な解釈の可能性をもたせた。ファイルの所有者でなくてもそれらの関係性を記述できるように、ファイルとカタログの管理基盤を分離した。またカタログにグローバルな識別子を与えて個々のカタログをユーザが識別可能にした。さらにファイルとそれを掲載しているカタログの対応関係を管理し、ファイルからカタログを逆引き可能にした。

以上のようなカタログシステムを試作し、カタログの取得時間を評価した。取得した有向グラフ数とその取得時間の測定結果から、本システムを用いて有向グラフをリアルタイムに可視化するアプリケーションが動作可能であることを確認した。

今後の課題として、本システムでは特定ファイルをノードにもつすべての有向グラフを取得することが可能であるが、それらに優先順位をもたせてユーザに提供する仕組みは実装されていない。今後、カタログシステムをより活用していくためには、前述したタグクラウドやタグクラスタリングのように、ユーザが作成した膨大なカタログに統計データを用いて有用なカタログを提供する機能について検討する必要がある。

## 文 献

- [1] Flickr, Available at: <http://www.flickr.com/> (accessed 4 February 2013)
- [2] Del.icio.us, Available at: <http://del.icio.us/> (accessed 4 February 2013)
- [3] Gene Smith, Tagging: people-powered metadata for the social web, New Riders Publishing, 2007.
- [4] Grigory Begelman, Philipp Kelller and Frank Smadja, Automated tag clustering: improving search and exploration in the tag space, Proceedings of the Collaborative Web Tagging Workshop at WWW 2006.
- [5] Scott Golder and Bernardo Huberman, Usage patterns of collaborative tagging systems, Journal of Information Science, 2006, vol. 32 no. 2 198-208