Vol.2015-IOT-31 No.3
Vol.2015-SPT-15 No.3
2015/9/25

Annoteaサーバ“Wasabi”でのウェブアノテーション全文検索手法の試験実装

安藤 大志[†1,a)]　山岸 一貴[†1]　萩原 威志[†1]

概要：アノテーション (注釈) 技術は，インターネット上の様々なツールとして使用されている．例として，YouTube の動画上に表示させることができるクリックが可能なテキストやエリアとしてのアノテーション，ニコニコ動画のコメントデータを動画アノテーションとして利用する技術などが挙げられる．最近では，Microsoft が発表した Windows 10 の新ブラウザ「Microsoft Edge」で表示した Web ページに直接キーボードまたは手書きでメモを残したり，落書きしたり，ハイライトすることができ，それらを保存してユーザ同士で共有できるアノテーション機能が紹介されている．W3C のプロジェクトである Annotea は，ウェブアノテーションを扱うためのデータ型や通信プロトコルを定めている．Annotea では，URL に紐付けされたウェブアノテーションをアノテーションサーバから検索して取得することができる．しかし，URL 以外のウェブアノテーションコンテンツに対して検索を行いデータの取得を行うことができない．
本研究では，研究室で開発中の Annotea サーバ“Wasabi”を使用して，コンテンツの全文検索手法の試験実装を行った．サーバ側を拡張せずにクライアント側で実装を行い，アノテーションを作成したときは検索インデックス自体もアノテーションとして作成しサーバ側へ登録した．この際に，大量のポストを行うのでサーバテストも兼ねて実験した．

Test implementation of web annotation full-text search techniques on Annotea server “Wasabi”

TAISHI ANDOU[†1,a)]　KAZUKI YAMAGISHI[†1]　TAKESHI HAGIWARA[†1]

Abstract: Annotations technique is used as a variety of tools on the Internet. As an example, the annotation of the text or the area capable clicks that can be displayed on YouTube videos, such as techniques utilizing a comment data NicoNico the video annotations like. In recent years, Microsoft annotation feature of the new browser ”Microsoft Edge” of Windows 10, which was announced have been introduced, that can leave a note in the indicated direct keyboard or handwriting on the Web page, graffiti, highlighted, and shared among users to save them. Annotea a W3C project, defines the data type and the communication protocol for handling web annotation. In Annotea, it can be obtained by searching the web annotations that are tied to the URL from the annotation server. However, it is not possible to carry out the acquisition of data is performed to search the web annotation content other than URL. In this study, using the Annotea server ”Wasabi” being developed in the laboratory, and tested implementation of full-text search technique of content. It implemented on the client side without extending the server side, when you create an annotation was registered to create as a search index itself annotation server side. In this case, the experiments were performed also serves as server test so do a lot of post.

[†1] 現在，新潟大学
Presently with Niigata Uniersity
[a)] t-ando@cs.ie.niigata-u.ac.jp

1. はじめに

我々の研究室ではかねてよりウェブアノテーションを扱う Annotea[1] を実装したサーバとして“Wasabi”の開発を進めている [2][3]．Wasabi は個人アノテーションサーバとしての利用を想定しており，他の Wasabi サーバと P2P ネットワークを介してデータの相互参照が可能となる．しかし，現段階では Wasabi に対して大量のクエリが発生した場合どのような問題が発生するかがまだ判明していない．

本研究では，そのような大量クエリが発生する場合における Wasabi の試験を行った．新たに提案するアノテーション全文検索手法は多くのクエリを生成するので，これを Wasabi に対して使用した．また，本研究で Wasabi に P2P ライブラリである PIAX[4] による再実装を行った．

本論文の構成は以下の通りである．第 2 章でウェブアノテーションの根幹技術である Annotea を説明する．試験実装で使用する Annotea サーバ Wasabi については，第 3 章で使用した P2P ライブラリ PIAX と一緒に紹介する．第 4 章でウェブアノテーション全文検索の仕組みを説明し，第 5 章で実際の試験実装を行う．

2. Annotea

Annotea は 2001 年頃に W3C で進められたプロジェクトである．Annotea は HTTP，RDF，XML に基づいて構築されており，任意のウェブリソースに対してコメント，メモ，説明といったアノテーションを付加することが可能になる．Annotea が実装されたアプリケーションではドキュメントに関連するアノテーションをサーバから取得することで，グループ内での情報共有ができるようになる．

Annotea におけるアノテーションは，作者や注釈日時いった注釈情報を持つプロパティ部とそのコンテンツを持つボディ部によって構成される．これらプロパティにはそれら情報が一意のものと表現するためのアノテーション ID が割り当てられる．

表 1 注釈プロパティ
Table 1 Annotation property

プロパティ名	プロパティ値	内容
annotates	URI	ウェブリソース URI
body	URI	コンテンツ
Description	URI	アノテーション ID
creator	テキスト	作成者
created	yyyy-mm-ddThh:mm:ssZ	作成日時

Annotea プロトコル [5] は，**表 1** のようなプロパティを持つアノテーションをクライアント／サーバ間で通信する手順を定めたものである．通信の際，アノテーションは RDF/XML 形式にデータ化される．Annotea クライアントは HTTP リクエストを用いて注釈ウェブアノテーションの Post，Query，Download，Update，Delete を要求し，Annotea サーバは要求に従いデータベースを操作する．

3. Wasabi

Wasabi では，複雑な設定なしで複数の Wasabi と協調動作することができるように，Annotea サーバの P2P 化を行った．Wasabi は，笹川・佐藤によって P2P フレームワーク JXTA[6] を用いて設計・開発されてきたが，現在 JXTA の開発がストップしており，古いフレームワークとなっていた．そこで使用する P2P フレームワークを PIAX へと変更し，開発を進めている．

3.1 システム構成

Annotea プロトコルにおいてアノテーション操作は，Post/Query/Download/Update/Delete の 5 つあり，それぞれアノテーションの新規投稿/検索/取得/更新/削除を行う．P2PAnnotea サーバとしての Wasabi の構成を，図 1 に示す．

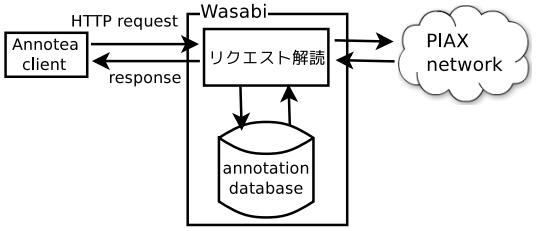

図 1 構成図

3.2 Wasabi への一斉リクエスト送信

Wasabi は，PIAX ネットワーク参加時に「自分が Wasabi のノードであることを表すための固有キー」を SkipGraph に挿入する．他の全 Wasabi ノードに対するリモート操作は，このキーを指定し，discoveryCall という機能を用いることで行った．

3.3 アノテーション相互参照

3.3.1 アノテーション Post

アノテーション Post(投稿) は，リクエストを受信した Wasabi ノードのデータベースに保存する (図 2)．

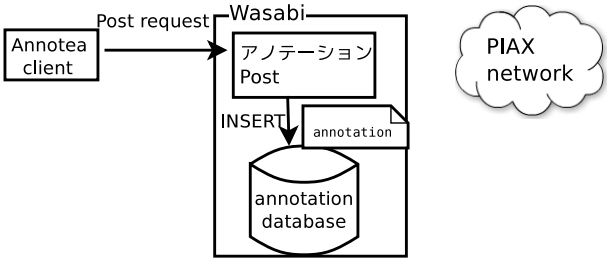

図 2 アノテーション Post の動作

3.3.2 アノテーション Query

特定の URI についてのアノテーションを検索する Query リクエストについては，P2P ネットワークを構成している全 Wasabi ノードのデータベースを検索する．ただし，毎回ネットワーク全体へリクエストを行うのでは，いつ Annotea クライアントにアノテーションデータを返すことができるのかわからない．そこで，Query リクエストに対するレスポンス時間をできるだけ短縮するため，Wasabi では「URI についてのアノテーションデータ収集と DHT へのデータ格納 (図 3)」と，「URI をキーに DHT からアノテーションデータを取得し，アノテーション検索リクエストに返答 (図 4)」を，並行して行っている．

全 Wasabi ノードへ，3.2 節で説明した discoveryCall を用い，アノテーションを収集する．後の Query リクエストに対応するため，URI をキー，収集したアノテーションデータをバリューとして DHT に格納した (図 3)．

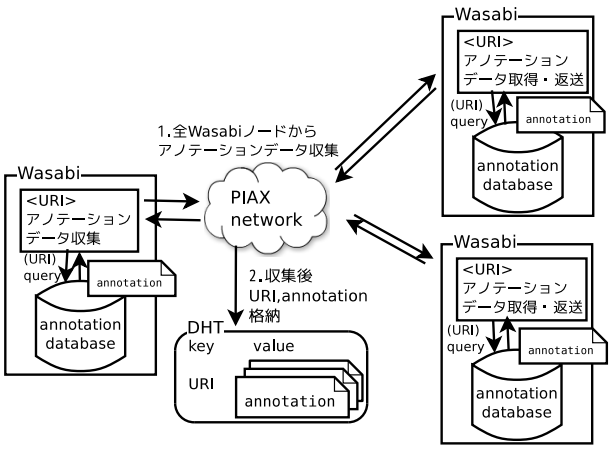

図 3 アノテーションデータ収集と DHT 格納

アノテーション Query リクエストを受信した Wasabi ノードは，DHT に対して URI をキーに指定することでアノテーションデータを取得する (図 4)．

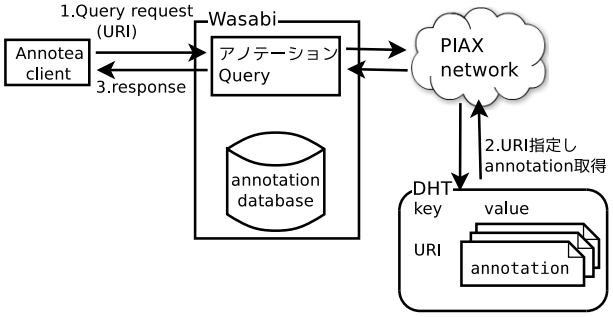

図 4 アノテーション Query の動作

3.3.3 アノテーション Update

アノテーション Update(更新) は，アノテーションを所有している Wasabi ノードを特定しなければならない．

Wasabi で生成するアノテーション ID には，初回にデータベース構築時に生成するデータベース ID が含まれる．Wasabi はネットワーク参加時にデータベース ID をキー，ピア ID をバリューとして，DHT に格納することとしている．

Update リクエストを受信した後，以下の動作を行う (図 5)．

(1) DHT に対してアノテーション ID に含まれるデータベース ID をキーに指定し，ピア ID を取得する．
(2) ピア ID の Wasabi ノードへ転送する．
(3) アノテーションの Update を行う．

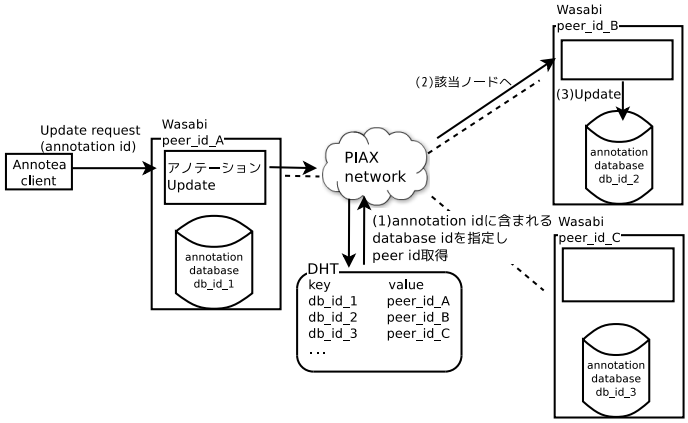

図 5 アノテーション Update の動作

4. アノテーション全文検索手法

Annotea では Annotea クライアントがアノテーションを取得するとき，URI をキーとして，それが annotates プロパティと一致したアノテーションにアクセスすることが可能である．しかし，annotates 以外のプロパティを用いた検索を行うことはできない．例えば，アノテーションが body プロパティに「天気予報」というコンテンツを持ち，この「天気」に対して検索をかけたくても annotates プロパティの値ではないので，データを取得することができない．そこで，そのような検索を実現するために索引情報自体をアノテーションとして登録しておくことでコンテンツ検索を実現するクライアントを試作した．

まず，annotates プロパティ値として，下記 URI のように既存の URI スキームと被らないような独自の URI スキームに，アノテーションの body プロパティのコンテンツを分解しそれを検索対象文字列（search string）として付け加えて一つの URI としてエンコードする手法を行った．そのようにして次のように作成した URI を索引 URI と呼ぶ．

$$wasabi://search/(search\ string)/$$

コンテンツの分割には n-gram 法を用いる．その中でも検索漏れを回避し，分かち書き手法としても使用される bigram を用いることにした．図 6 のようにコンテンツの文字数が多いほど一度に作られる索引の数は多くなる．

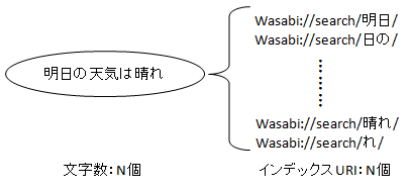

図 6 索引 URI の作成
Fig. 6 Creating a Index URI

索引は body プロパティに，アノテーション ID をリストとして保管する．これらアノテーション ID をプロパティに持つアノテーションは body プロパティのコンテンツの一部に検索対象の文字列を含んでおり，検索結果として取得可能となる．

次に，索引 URI を Annotea サーバへクエリしそれら URI と一致する索引をダウンロードする．ダウンロードしたデータに既存の索引が存在しなければ，新しく生成した索引にアノテーション ID を載せて投稿する．既存の索引が存在している場合，索引の body プロパティのアノテーション ID のリストを取得する．このリスト内容に先ほど割り振られたアノテーション ID を新たに追加し，Annotea サーバへ索引の更新を行う．

検索時は，Annotea クライアントが入力した検索文字列を対象に，索引を作成した手順と同様図 6 のようにして URI を生成し Annotea サーバへ索引のクエリをかける．ダウンロードが成功した索引それぞれに記載されてあるアノテーション ID をさらに Annotea サーバへクエリを行い，アノテーション本体のデータを取得する．

この手法では，図 6 のようにコンテンツ文字数によりクエリの量が増減する．したがって，アノテーション投稿により多くのクエリが発生する状況を実現できる．

5. アノテーション全文検索の試験実装

アノテーション全文検索クライアントの試験実装を行なった．このクライアントでは，アノテーション投稿の度にアノテーションが持つコンテンツに対する索引を生成する．これにより，サーバに対するクエリが数多く発生する．また，Wasabi を複数動作させる場合，同じ端末上で動かしネットワーク遅延は考えないものとする．

初めに，Wasabi への投稿用データとしてアノテーションデータをランダムで 500 個作成する．次に，以下（1）から（4）の条件で Wasabi に対して作成したデータを投稿し，その結果を記録した．

(1) データが空の Wasabi を 1 台起動し，試験データを投稿
(2) （1）の結果に再度同じデータを投稿
(3) 新たに 1 台 Wasabi を起動し，2 台の Wasabi でネットワークを構築，試験データを投稿
(4) 3 台以上の Wasab でをネットワークを構築し，試験データを投稿

5.1 データ準備

試験実装で扱う投稿用データとして，全文検索の対象となるアノテーションデータを用意する．アノテーション body プロパティのコンテンツとなるデータは，アノテーション投稿が実際に行われることを想定し，注釈文や説明文のようなものが望ましい．そのようなデータを取得するために，Google が提供する API である Goodle Custom Search Engine（CSE）[7] を使用した．これにより最初に入力した検索文字列から Google 検索結果を JSON 形式で得られる．API の都合上，一度に 10 件しか検索結果を得られないが，それらから新たな検索ワードをランダムに抽出し，再度検索して新たな 10 件の検索結果を得る手法をとった．検索結果を自動的に注釈プロパティの各種プロパティへ入れるプログラムにより，試験用データを合計 500 個作成した．

5.2 動作試験

試験前に前節において作成したデータ 500 個を調べたところ，コンテンツを bigram することにより作成される索引の数は合計で 12502 個である．これらを考慮したうえで（1）から（4）のそれぞれの条件下において Wasabi へ投稿した結果をそれぞれ示す．

初めに，データが空の Wasabi を 1 台起動し試験データを投稿する場合について試験する．試験データをすべて投稿し終えるまでに掛かる時間，投稿した総アノテーション数を以下**表 2** に結果を示した．

表 2 試験 1 結果
Table 2 Test 1 Result

終了時間	アノテーション総数
1355753 ms	13002 個

次に，条件（1）の結果に再度同じ試験データ 500 個を投稿した結果を**表 3** に示す．

表 3 試験 2 結果
Table 3 Test 2 Result

終了時間	アノテーション総数
1655654 ms	13515 件

試験 1 の時点で試験データ 500 個における索引はすべ

て作成されている．同じデータを投稿した場合，アノテーション 500 個のデータのみ増えるはずがそれ以上増えていた．原因については現在調査中．

次に，データが空の Wasabi 新たに起動し，条件 (1) で使用した Wasabi とネットワークを構築した状態で試験データを投稿した．投稿結果を見たところ，単体の Wasabi に投稿したときの投稿終了時間より 2 倍ほど速くなっていることがわかった．次に，この原因を探るため PIAX が作成したキャッシュを除外し，再度試験データを投稿した．その結果はキャッシュがあるときよりも投稿時間が速くなっていた．最後に，キャッシュもデータベースも除いたデータがすべて空の Wasabi2 台でネットワークを構築して投稿試験を行った．その結果，今までの状態の中で一番速い投稿時間となった．それぞれの状態と結果の表を**表 4** に示す．

表 4 試験 3 結果
Table 4 Test 3 Result

状態	終了時間
データベース有り，PIAX キャシュあり	580026 ms
データベース有り，PIAX キャシュなし	560685 ms
データベースなし，PIAX キャシュなし	422433 ms

Wasabi を 2 台繋げて投稿した方が，単体で動作させているものに投稿するより投稿時間が速いという結果がわかったが原因に関しては現在調査中．

最後に，3 台以上の Wasab でをネットワークを構築し，試験データを投稿する．試験の結果，ネットワークを構築した Wasabi のどれもデータを持たなかった場合，1 台の Wasabi までならデータをすべて投稿し終えることが確認できた．しかし，Wasabi 1 台にデータが既に存在しており，データが空である別の Wasabi へ同じデータを投稿した場合，投稿途中で急激に投稿時間が遅くなる現象が起きた．

3 台以上でネットワークを構築した場合，この現状は必ず発生した．現在，原因を調査中している途中である．

5.3 動作試験まとめ

Wasabi が単体動作または複数で連携動作している場合において，個人サーバに数万のクエリが集中すような状況における試験を行った．実装試験の結果課から Wasabi に登録される際に何かしらの問題が起こっていると考えられる．また，索引の更新処理が必要な場合に DHT へアクセスできなくなることが頻繁に起こった．これら実装試験の結果より，Wasabi は排他制御において機能も不十分であるといえる．したがって，現時点の Wasabi は Annotea サーバとして十分に機能するとはいえない．

6. まとめ

本論文では，Annotea クライアントに対しアノテーション全文検索手法を実装し，現在研究室で開発中の Annotea サーバ Wasabi に対して試験した．元々個人ユーザ利用を想定している Wasabi に，この手法によって発生する大量のクエリを送り発生するであろう問題を試験し結果を記録した．まだ全文検索手法や Wasabi は改良できる余地が残っている．例えば，全文検索手法の索引作成は現段階において同期処理で行っているので，非同期にした場合さらにサーバ側のエラーが顕著に表れると予想できる．その場合においても Wasabi が動作するよう改善し試験を行う必要がある．今後も改良を重ねていく予定である．

参考文献

[1] Annotea Project, http://www.w3.org/2001/Annotea/.
[2] 笹川透，瀬野瑛，萩原威志, Annotea の P2P 型実装を用いた掲示板の構築（2010）
[3] 佐藤慶太，瀬野瑛，萩原威志, JXTA を用いた自律分散型 Annotea サーバ Wasabi の実装（2014）
[4] PIAX, http://www.piax.org/.
[5] Annotea Protocols, http://www.w3.org/2001/Annotea/User/Protocol.html.
[6] JXTA, https://jxta.kenai.com/.
[7] Google Custom Search, https://developers.google.com/custom-search/