3E3-OS20-8

Web サイエンス基盤のための大規模意味資源のスケーラブルな集積化の試み

An Approach to Scalable Gathering and Storing Very Large Semantic Resources for Web Science Infrastructure

岩爪 道昭[*1] 河原 大輔[*2] 兼岩 憲[*3] 赤峯享[*1†]
Michiaki Iwazume Daisuke Kawarahara Ken Kaneiwa Susumu Akamine

加藤 義清[*1‡] 大西 加奈子[*4] 小林 一郎[*4]
Yoshikiyo Kato Kanako Onishi Ichiro Kobayashi

[*1]独立行政法人 情報通信研究機構
National Institute of Information and Communications Technology

[*2]京都大学大学院 情報学研究科
Graduate School of Informatics, Kyoto University

[*3]岩手大学 工学部
Faculty of Engineering, Iwate University

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

In this paper, we propose a scalable and elastic approach to gathering and storing semantic resources on the Internet. Web archives are important to research and development in Web science. Recently, a large amount of structured data has been published on the network. We implemented a prototype system of a distributed virtualized crawler and RDF data store environment to develop a large-scale linked data archives.

1. はじめに

Web は，実世界の様々な活動や事象に関する情報が蓄積され，学術研究，文化，社会生活，経済活動等において不可欠なインフラとなっており，これらの諸活動に資するより安全で利便性の高い利活用のためのサービスが求められている．このような要請に応えるためには，Web をアーカイブとして集積化した上で，それに基づく高度な分析技術および利活用技術の開発，検証することが不可欠である．

一方，Web 上に流通する情報は，ますます多様化，大規模化しつつある．セマンティック Web の研究分野では，生物，医学，地球環境計測および電子政府等の分野において，個別に蓄積・管理されていた各種データが，セマンティック Web の仕様に準拠した RDF と呼ばれる主語-述語-目的語の 3 つ組み形式（トリプル）で Web 上に公開され，他の公開 RDF データ，オントロジと相互リンクされることで，大規模な意味空間が形成しつつある．このような RDF データのオープン化の取り組みは，Linking Open Data と呼ばれ[1]，DBPedia[2]を中心にして，2010 年 8 月時点の推計で，約 25 億の RDF トリプル，約 3 億 9500 万個の相互リンクを持つ規模に達している．

この超大規模知識ベースを背景として，新しい Web 基盤技術の研究開発が加速されるとともに，多様なサービスの創出が期待されている．

本研究では，以上の経緯を踏まえウェブサイエンス基盤のためのウェブアーカイブ構築の一環として Linked Data アーカイブ構築のための RDF データを対象にした分散クローリング，分散ストア環境の試作を試みた．

以下，第 2 章では，ウェブクローリングに関する関連研究について概観し，本研究の背景とする分散クローリング，ストア環境の必要性について述べる．第 3 章では，現在我々が取り組んでいる大規模 Web アーカイブのための分散仮想クローラおよびデータストア環境のアーキテクチャについて概説する．第 4 章では，試作した実験システムの予備テスト結果について説明する．第 5 章では，本研究の課題と展望について述べる．

連絡先：岩爪 道昭，独立行政法人情報通信研究機構 ユニバーサルコミュニケーション研究所，京都府相楽郡精華町 3-5，TEL: 0774-98-6873，E-mail: iwazume@nict.go.jp

†現在，日本電気株式会社
‡現在，Google株式会社

2. 関連研究

2.1 関連研究

Web が出現して以来一般的な Web 文書のクローラについては数多くの研究開発がなされている[3]．セマンマンティック Web 文書検索エンジン Swoogle[4]は，Web クローラにより収集した 10,000 個以上のオントロジを検索することが可能であるが基本的にキーワードによる文書検索のため例えば RDF 文書の内容に関する問い合わせはできない．Slug[6]は，Jena API[7]による RDF ストア機能を有するセマンティック Web 文書クローラであるが数十億～数百億トリプル規模の大規模な RDF データを取り扱うことが困難である．また，Linked Data に特化したクローラとしては，LDSpider[8]，LSCrawler[9]も提案されているが，日々増大しつつある Linked Data 空間に対してスケーラブルにクロ

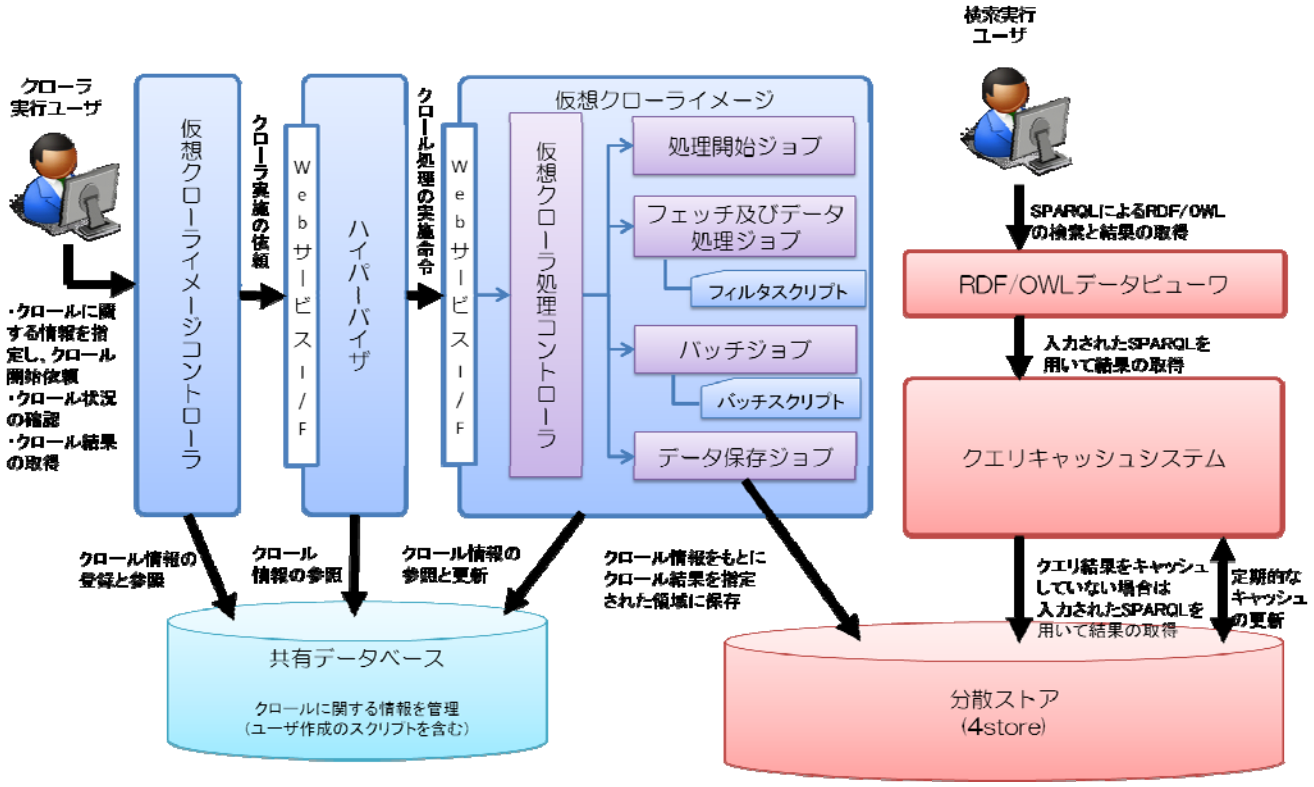

図 1: 分散仮想クローラのアーキテクチャ概観

ーラを並列化し，大量の RDF データをストアするアーキテクチャにはなっていない．

2.2 より柔軟かつスケーラブルなクローラの必要性

日々変化が激しく，多様かつ大量の情報がパブリッシュされる Web においては，予め全ての収集対象を想定してクローラを設計・実装することは不可能である．先行研究を踏まえ，大規模な意味資源の集積化のためには以下のような要件に応える必要があると考えている．

スケーラビリティ

Web は巨大であり絶えず増加しており，また収集対象規模の増大に応じてクローリングおよびストア環境がスケールアウトしていくクラウド対応型アーキテクチャが不可欠である．

大規模かつ変動する計算機・ネットワーク環境への対応

大規模クローリングの運用では，全てを人手で制御することが事実上不可能な為，クロールするために必要な資源(計算資源，ネットワーク資源，ストレージ資源など)を自動的に制御する機能が必要である．

ユーザニーズへの迅速な対応

新しい種類の情報資源やユーザが指定した特定のタイプの情報資源だけをクロールする機能も必要である．

3. 分散仮想クローラおよびデータストア環境

3.1 アーキテクチャ概観

本研究で提案する分散・仮想化クローラの構成を図に示す．

仮想クローライメージコントローラ

ユーザからクロール処理依頼を受け付ける．ユーザにより指定されたクロール情報(シード URL，各種スクリプトなど)を共有データベースに登録し，ハイパーバイザにクロール実施の依頼を行う．

ハイパーバイザ

仮想クローライメージコントローラからクロール実施依頼を受け付ける．インターフェースとして Web サービスインターフェースを保持している．共有データベースからクロール情報を取得し，その情報をもとに自身が管理する仮想クローライメージから，クロール情報に最適な仮想クローライメージを選択または生成して，それらに対してクロール処理の実施命令を行う．実施命令を行う際，仮想クローライメージが行うべき詳細処理を指定する．

仮想クローライメージ

ハイパーバイザからクロール実施命令により実際のクロール処理を行う．インターフェースとして Web サービスインターフェースを保持している．ハイパーバイザから指定を受けた実行すべき詳細処理を，仮想クローライメージ内に含まれる仮想クローラ処理コントローラが実行する．詳細処理としては以下の四つを保持する．

(ⅰ)処理開始ジョブモジュール

登録されているクロール情報をもとに，クロール実施に必要な情報(設定ファイルなど)の準備を行う．

(ⅱ)フェッチ，データ処理ジョブモジュール

処理開始ジョブで生成された情報をもとに収集対象となるデータを取得し，データに対してユーザにより登録されたフィルタースクリプトを実行して保存対象データかどうかの判断を行う．フィルタリング処理の例としては，特定の文言を含む場合のみ保存対象とするフィルタリングなどがある．後述の実験システムでは，フィルタリング処理として，取得したデータが RDF/OWL データかどうかの判断を行っている.また，取得した収集対象候補に対しては，ユーザにより登録された VisitURL スクリプトを実行して，収集除外対象でない場合は新たな収集対象として追加する．

(ⅲ)バッチジョブモジュール

フェッチ，データ取得ジョブにより保存対象とみなされたデータに対して，ユーザにより登録されたバッチスクリプトを実行して，保存対象データへ特定の処理を行う．この処理が実行された時点で，取得したデータは保存される形式に変換されていることになる．処理の例として，特定の文言でフィルタリングを実行した場合はその文言の出現回数を数える処理や，保存対象のデータのメタ情報を作成する処理がある．後述の実験システムでは，圧縮形式のデータがあった場合はデータを解凍する処理を行っている．

（iv）データ保存ジョブ

バッチジョブにより保存形式に変換されたデータは，データのタイプに応じて分散ストアへ保存・格納する．

共有データベース

ユーザから実行依頼を受けたクロール情報を管理する.仮想クローライメージコントローラ，ハイパーバイザ，仮想クローライメージの全てから操作できる領域となる．管理する情報として，以下がある．

・クロールを実行するための情報(シード URL，　実行時間，結果保存先情報など)
・依頼を受けたクロールの実行状況
・ユーザ作成の各種スクリプト

分散ストア

クロール結果を保存する．クローラの構成要素の一部としてあらかじめ準備しておくこともできるが，外部のデータストレージサービスを使用するようなケースも考えられる．後述の実験システムでは，RDF/OWL を登録できる分散ストアとして 4store[11]を採用している．

データビューワ

SPARQL クエリを受け付けて分散ストア上のデータの検索を行う．

クエリキャッシュシステム

高速に検索を実行するためにクエリの結果をキャッシュする．

3.2 仮想クローライメージ処理の流れ

仮想クローライメージにおけるクローリング処理の概要を以下に示す．

事前条件

・クロール条件は Web サーバを経由して取得．
・ユーザがクロール開始の "「URL」,「階層」" を設定．

メインフロー

Step1：クロール対象 URL 生成

クローラが動作するための設定ファイルを生成．クロール開始の URL はユーザが入力した URL を使用し，設定ファイルを生成．

Step2：クローラファイル設定

設定した User Defined のクロール条件を元に設定ファイルを生成．

Step3：クローラ実行

（ｉ）クローラがクロールを実行し，データを取得．RDF/OWL データの場合もアウトリンクを取得．

（ii）クローラが動作中に取得した結果に対して設定ファイルに記述された条件を元に新規作成したフィルタを実行．フィルタリング内容：保存対象ファイルを選別
条件：ファイルの内容をパースし，xm l のルート要素が RDF となっているファイル内から保存対象となりうる URL を取得

（iii）取得したデータを保存．

（iv）クローラが終了条件を満たした場合，クロールを終了．

Step4：バッチ処理

クロールにより新たに一時保存されたデータを監視し，発見し次第，発見したファイルが圧縮されている場合は解凍．

Step5.：後処理

分散ストアに保存されていないデータを監視し，発見し次第，分散ストアに保存．

4. 実験システムによる予備検証

本提案手法に基づく実験システムを試作し，4ノードの小規模 PC クラスタ環境を構築（図を参照），仮想化クローライメージ，分散ストア環境の動作検証を行った．各ノードの仕様は以下に示す．

・OS：CentOS5.5
・CPU：Intel(R) Core i5-650 3.20GHz(コア数:2)
・メモリ：4GB
・ディスク：1TB

4.1 クローリング予備テスト

試作した分散・仮想化クローラの動作検証のため，DBpedia のデータセットのページをシードとし，深度 3 でクローリングを行った．その結果，収集した URL 総数は 13,499 個，そのうち内取得された RDF の総数は 105 個であった．そのうち DBPedia 内で収集した URL 総数は 11,131 個，取得した RDF ファイル数は 22 個であった．収集したトリプルの総数は未集計であるが DBpedia 内から取得した RDF ファイルのうち任意 10 個 RDF ファイルに含まれる総トリプル数は 17,000 個であった．Linked Data のハブの一つである DBpedia のデータセットページをシードとしたにもかかわらず，総 URL 数に対して RDF 文書の総数が 1%に満たないの RDF 以外の情報資源へのリンクが多数含まれることによるものである.今後 RDF データのみを選択的に収集する場合には，

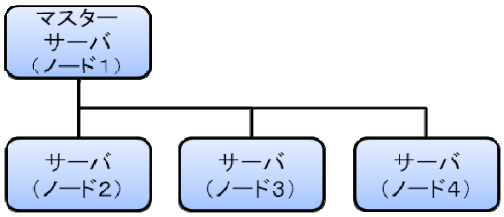

図 2: 検証用小規模 PC クラスタ構成

表 1: 分散ストア検証用サンプルデータ概要

ファイル名	トリプル数	ファイルサイズ (KB)
Angela_Merkel.rdf	190	51
Cambridge.rdf	615	142
George_W._Bush.rdf	623	152
Led_Zeppelin_III.rdf	142	39
Manchester.rdf	1,546	339
Nicolas_Sarkozy.rdf	272	60
Oliver_Stone.rdf	177	39
Spain.rdf	13,362	2,815
Takashi_Miike.rdf	138	31
The_Lord_of_the_Rings.rdf	98	22
合　計	17,163	3,690

生成元データ

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema# ">
    <rdf:Description rdf:about="http://dbpedia.org/resource/Angela_Merkel">
        <dbpprop:governmentType xmlns:dbpprop="http://dbpedia.org/property/" xml:lang="en">
        Shire district, City
        </dbpprop :governmentType>
    </rdf:Description>
</rdf:RDF>
```

ダミーデータ

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema# ">
    <rdf:Description rdf:about="http://dbpedia.org/resource/Angela_Merkel">
        <dbpprop:governmentType xmlns:dbpprop="http://dbpedia.org/property/" xml:lang="en">
        1298594931004Shire district, City
        </dbpprop :governmentType>
    </rdf:Description>
</rdf:RDF>
```

図 3: 分散ストア性能検証用ダミーデータ例

4.2 分散ストア環境の検証

大規模な Web クローリングでは，クローリングのスケジューリング加え，収集したデータのファイル I/O，インデックス生成がスケーラビリティにおける律速条件となる．本研究では，本来，クローラと分散ストアの一体的な性能検証が必要であるが，その予備検証として分散ストア単体での検証を実施した．分散ストアとしては，RDF に限定した検証を行うため，4store[11]を試験的に採用し，その動作検証を行った．4store は，Garlik[12]においても採用されているスケーラブルな SPARQL 検索エンジン，RDF ストアデータベースを備えた，8GB メモリ搭載 9 ノードの PC クラスタで 150 億トリプルを取り扱った実績があり，ハッシュ値衝突の問題があるものの 500 億～700 億トリプルが上限値と推定されている．

検証にあたっては，DBpedia データセットから抽出した 17,163 トリプルを元データ（表 1 参照）としてダミーデータを生成し（図 3 参照），5 億トリプルを上限として，1,000 万トリプル毎の分散ストアへの登録処理時間を計測した（表 2 参照）．また，5 億トリプル格納時における新規データ登録所要時間についても計測を行った（表 3 参照）．図 4 は 1,000 万トリプル毎のデータ登録処理時間の変化をグラフ化したものである．以上の結果から，比較的非力なマシンによる小規模 PC クラスタにおいてもノード数に対して実行的にレベルでスケールすることが明らかになった．

5. まとめ

本研究では RDF に代表される意味資源の大規模集積化のための分散仮想クローラ，ストア環境のアーキテクチャを提案，試作し，小規模な実験を行った．今後は，クローリングアルゴリズムの効率化を進めるとともに，100 ノード（1200 コア）程度のより大規模な PC クラスタシステム上に分散クローリング，ストア環境を構築，試験運用し，アーカイブ構築を進める予定である．

参考文献

[1] http://linkeddata.org/

[2] http://wiki.dbpedia.org/Datasets

[3] C. Olston and M. Najork: Web Crawling, Foundation and Trends in Information Retrieval, Vol.4, No.3, pp175-246, 2010.

[4] http://swoogle.umbc.edu/

[5] Hsin-Tsang Lee, Derek Leonard, Xiaoming Wang, Dmitri Loguinov: IRLbot: scaling to 6 billion pages and beyond. 427-436, www2008, 2008

[6] Leigh Dodds, Slug: A Semantic Web Crawler, http://www.ldodds.com/blog/2004/12/slug-a-simple-semantic-web-crawler/

[7] http://jena.sourceforge.net/

[8] Isele, R.; Umbrich, J.; Bizer, C.; and Harth, A.:LDSpider: An open-source crawling framework for the Web of Linked Data, In 9th International Semantic Web Conference (ISWC2010), (2010)

[9] M. Yuvarani, N.Ch.S.N. Iyengar, A. Kannan; LSCrawler: A Framework for an Enhanced Focused Web Crawler Based on Link Semantic, 2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI'06), pp794 – 800, 2006

[10] http://4store.org/

[11] http://www.garlik.com/

表 2: 5 億トリプルの新規登録所要時間

目的トリプル数		クラスノード数（コア数）			
		1 (2)	2 (4)	3 (6)	4 (8)
5億	処理時間	42:11:04	25:02:28	17:07:52	14:27:00

表 3: 5 億トリプル格納時の新規データ登録所要時間

目的トリプル数		クラスタノード数（コア数）			
		1（2）	2（4）	3（6）	4（8）
5億	平均登録数（トリプル/秒）	13	10	10	10
	インデックス更新時間（秒）	0.045796	0.045424	0.04722	0.047255

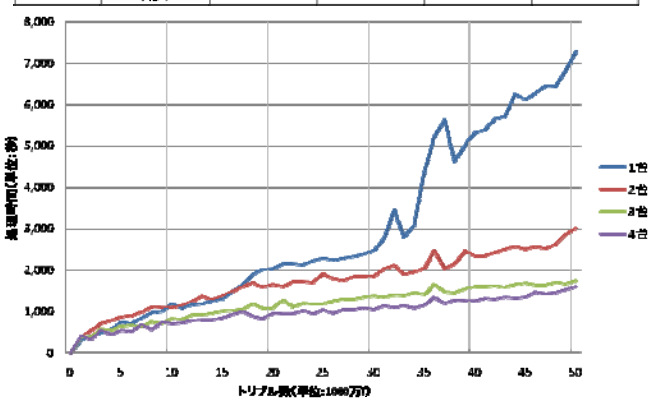

図 4: 分散ストア検証用サンプルデータ