1A1-OS-17a-2

ハイパフォーマンス・エラスティック・クローリング

「億」を超える超高性能クローラの実現に向けて

High performance elastic crawling

藤井 秀明[*1] Hideaki Fujii　岩瀬 高博[*2] Takahiro Iwase　原口 弘志[*1] Hiroshi Haraguchi　泥谷 誠[*1] Makoto Hijiya　岩爪 道昭[*1] Michiaki Iwazume

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

[*2] 株式会社神戸デジタル・ラボ Kobe Digital Labo., Inc.

We developed and have operated a web crawler for an information analytical system named WISDOM to construct a web archive of billion pages. And now, we are planning to build up a high performance crawler dealing with more large-scale and various media data. In this paper, we show a result of verification of very large URL data management, which is one of challenges to realize a crawler in “Big Data” era, as well as a brief explanation about our past effort.

1. はじめに

独立行政法人情報通信研究機構（以下，「NICT」とする）では，アジアを中心として諸外国の生活，経済，産業等に関する情報を Web，ニュース，新聞等の情報メディアより収集し，翻訳，要約，情報分析技術等により入手した情報を整理・構造化し，質の高い基盤的情報資源やアプリケーションをサービス化して利活用可能とするアジア情報ハブを構想している[岩爪 2012]．同構想の実現に向けて，ネットワーク上の大量の情報を集積化するためのインフラとして 40 億ページ以上の Web アーカイブを含む大規模情報基盤の研究開発に取り組んでいる．

一方，NICT では，情報分析システム WISDOM[WISDOM]の研究開発において，インターネット上の情報を収集するための Web クローラを開発・運用し，ユニーク URL で約 7 億ページ規模の Web アーカイブを構築してきた．

アジア情報 HUB および大規模情報基盤の構築に向け，現在の Web クローラを大幅に見直し，ビッグデータ時代に対応したよりスケーラブルかつハイパフォーマンスな次世代クローラが不可欠になっている．

本稿では，これまでの取り組みとして NICT クローラを概説し，次にビッグデータ時代に対応した次期クローラの構想を提案し，実現に向けた諸課題について検討する．最後に，課題解決の端緒として，クローラの制御において最も基本的な情報資源となる URL データベースに焦点をあて，数十億～数百億オーダーの URL 情報を実行的に管理・検索可能とするための手法について検証した結果を紹介する．

2. NICT クローラ

2.1 システム構成および機能の概要

NICT クローラは，インターネット上に公開されている Web ページを自動で収集するシステムである．言語情報をベースとした情報分析で用いるデータの収集が目的であるため，収集対象は原則 HTML ファイルのみとなっており，画像やその他メディアファイルは対象外としている．

システムは，クローラ本体と URL 情報を管理する URL DB により構成される．クローリングの基本的な処理フローを図1に示す．(1)URL DB より収集対象の URL リストを取得し，(2)DNS サーバに IP アドレスを問い合わせ，(3)並列してページを収集し，(4)ページを解析してリンク抽出などを行い，(5)抽出したデータにより URL DB を更新する．

上記の処理を基本としつつ，NICT クローラでは目的に応じて，新規・更新クローラ，深度クローラ，RSS クローラの 3 種別のクローラが運用されている．新規・更新クローラは，新たに発見された URL を対象とする新規クロールと，一度収集された Web ページについて更新の有無を確認するために一定間隔を空けて再度収集する更新クロールを行う．深度クローラは，ニュースサイトなど有用なページが高頻度で生成されるサイトを定点観測し，未収集ページを収集する．「深度」の名称は，設定された回数だけ起点となるページからリンクを辿り収集するところに由来する．RSS クローラは，ブログなどの RSS フィードを継続的に取得し，新規 URL が含まれていれば，それを収集する．

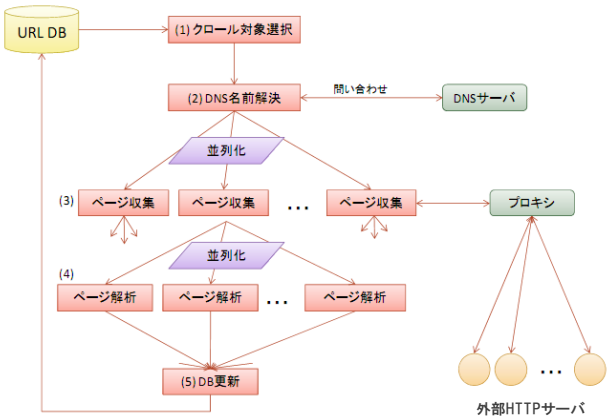

図 1 NICT クローラの処理概要

2.2 運用状況と実績

NICT クローラの運用構成を表 1 に示す．新規・更新クロールは 4 並列により実施する構成となっており，収集後に遅滞なく処理を進めるためページ解析は 24 並列の構成としている．深度クローラと RSS クローラについては，各 1 プロセスにて実行している．

ネットワーク環境としては，JGN-X[JGN-X]に接続されており，そこから民間プロバイダ経由でインターネットにアクセスしている．

連絡先：藤井秀明，独立行政法人情報通信研究機構，〒619-0289 京都府相楽郡精華町光台 3-5，h-fujii@nict.go.jp

これまでの収集実績は，ユニークで約 7 億ページを収集，1 日あたり最大 1,000 万ページ程度を収集している．

表 1 NICT クローラの運用構成

種別	処理	ノード数	プロセス数
新規・更新	制御，DB登録	1	1
	ページ収集	4	4
	ページ解析	6	24
深度	全ての処理	1	1
RSS	全ての処理	1	1

2.3 課題

NICT クローラの平均的な 1 日のトラフィック状態を図 2 に示す．収集開始直後は，プロバイダとの契約帯域を十分に活用して収集しているが，徐々に利用帯域が減少し，10 時間後にはクロール処理が停止している．これは，収集したページから抽出したリンクを URL DB に登録・更新する処理がクローリングの処理速度に追いつかず，意図的に収集を抑制，停止しているためである．

URL DB には MySQL を利用しており，当初はストレージとして HDD を用いていた．しかしながら，登録処理のボトルネックが顕著となってきたため，高速化を意図して SSD を採用した経緯がある．これにより登録処理のパフォーマンスは向上したが，それでも図 2 に見られるように，さらなる処理速度の向上が課題である．

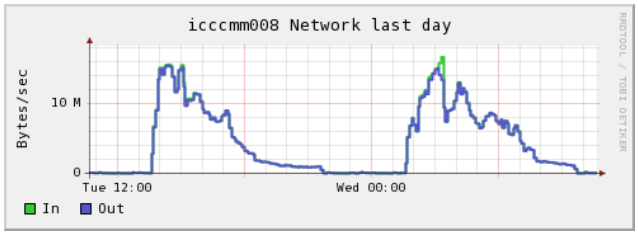

図 2 NICT クローラのトラフィック状態

3. 次期クローラ構想

3.1 次期クローラの目標

ライフログ，ソーシャルネットワークサービスや IOT（Internet of things）と呼ばれる電子センサなどから，大量かつ非定型なデータがリアルタイムに発信されている．このようなデータは「ビッグデータ」と呼ばれ，その分析や解析を通して有益な情報がもたらされることが期待されている．

NICT においてもユニークで 40 億ページ規模のアーカイブ構築を計画しており，研究開発における活用が目指されている．このアーカイブ構築に必要なデータを収集することが，次期クローラの目標である．

この目標を達成するために解決する必要があると思われる課題を次節にて検討する．

3.2 ビッグデータ時代の課題

ビッグデータの定義として Volume，Velocity，Variety のいわゆる「3 つの V（the three Vs）」[O'Reilly 2012]が知られているが，クローリングの観点からは Volume と Variety が重要なポイントになると考えられる．

Volume とは大量のデータを意味する．次期クローラで 40 億の新規ページを 4 年間で収集すると想定した場合，1 日あたり約 270 万ページの収集が必要となる．運用に際しては新規と更新の両クロールを行うため，取得ページ数に占める新規ページの割合を 15%とすると，約 1,800 万ページ/日（210 ページ/秒）もの高い収集能力が求められることになる．

Variety とは非定型で多様なデータを意味する．これまで通常の Web ページがクロール対象であったが，今後はそれに加え SNS サイトやミニブログ，音声・動画データなどと多様なデータのクローリングが求められることになる．

このようなビッグデータを対象とするクローリングの課題と関連する先行研究をまとめると以下のようになる．

(1) データ入出力の高速化

クローラは収集した HTML などのコンテンツを保存する以外に，収集日時，URL，レスポンスコードやコンテンツのハッシュ値などのメタデータをデータベースに登録している．これらのデータは，収集済み URL かどうかの判定や更新クロール実施可否の判定などに利用される．2.3 節において説明した NICT クローラのボトルネックのように，データ件数が増加するとデータへのアクセス速度が低下し，クロール処理全体のパフォーマンスに影響を及ぼす．

クローラの入出力高速化に関連する先行研究としては，サーバ1台を用いて 41 日間で 63 億ページを収集した IRLbot[Lee 2008]や，Mercator[Najork 2001]などがある．

(2) スケーラビリティの確保

クローリングの目的にもよるが，ビッグデータ対応の観点からすると，無尽蔵に生成される大規模なデータが収集対象となる．ネットワーク帯域が許容する範囲内で最大限の収集効率を求めるためには，複数のノードで並列して収集する必要がある．理想的には，状況に応じてノードを追加（あるいは除去）することで，線形にパフォーマンスが向上（あるいは抑制）可能であることが望ましい．

複数マシンに分散して収集を行うクローラとしては，既出の Mercator や，集中制御や単一障害点を排除し完全分散処理を実現した UbiCrawler[Boldi 2004]などがある．

(3) クロール戦略の最適化

上記(2)と関連するが，どのページをどの順番で収集するかというスケジューリングが，限られたリソース制約の下における収集数の向上には不可欠である．例えば新規クロールと更新クロールを実施する場合，データの網羅性の観点から新規クロールの稼働率を上げれば，それに反比例して更新クロールの稼働率は低下しデータの鮮度が落ちる．一方，鮮度を保つためには更新クロールの比率を上げればよいが，収集ページ数が増加するほど更新対象も増えるため，収集効率を向上するためには最適な更新間隔を推定する必要がある．

クロール戦略は目的に応じて設定されるものであるため，上記に取り上げたクローラもそれぞれの戦略を持っている．興味深いものとしては，効率的な更新クロールのため更新間隔を推定してスケジューリングを行う手法[田村 2008, 2009]や，計算量を制御することで高速なスケジューリングを図る手法[森本 2011][山田 2004]などがある．

(4) 収集対象の多様化

HTML や動画像などのファイルに限らず，音声や動画のストリーミングデータの収集も今後求められてくると推察される．その

他，掲示板やブログなどでは，ページ単位ではなく投稿や記事単位で収集・保存することも考えられる．

管見の限りでは，ストリーミングデータのクローリングに関する研究は発表されていないと思われる．掲示板やブログ関連では，掲示板を対象とするクロールデータの管理方法[Yunhua 2011]やブログクローラの研究開発[Hurst 2009]などがある．

(5) 重複コンテンツの排除

ビッグデータ規模のクローリングにおいては，収集段階だけではなく，保存先のストレージ容量の確保も問題となることが多々ある．このため重複するコンテンツを排除する仕組みが重要である．重複するコンテンツは，DNS のエイリアスを利用した同一ページを指し示す異なる URL や，ミラーサーバなどが原因である．その他，ページが更新される前に更新クロールを行った結果として，重複するページが収集される．これら重複コンテンツの排除には，ページのハッシュ値を比較する方法が一般的である．しかしながら，本質的ではない箇所（バナー広告など）の違いにより，異なると判断される近重複コンテンツの除去は課題である．

このような近重複コンテンツの検出においては，shingling[Broder 1997]が標準的に利用されている．また，URL パターンにより重複コンテンツを推測することで，そもそもコンテンツをダウンロードすることなく排除する DUST ルール[Bar-Yossef 2007]や，学習アルゴリズムを導入してそれを汎用化させた手法[Dasgupta 2008]などが提案されており，次期クローラへの導入を検討したい．

次期クローラは，本節にて取り上げた課題の対策を組み込んだシステムとする構想である．そこで，まずは「(1)データ入出力の高速化」の解決策としとて，NoSQL を活用した URL データ管理の是非を検討したい．

4. NoSQL による URL データ管理の性能検証

4.1 検証環境

URL DB の高速化対策として，NICT クローラで利用していた RDBMS（MySQL）に代わり，NoSQL の採用を検討した．NoSQL は，RDBMS のような厳格なデータ一貫性や柔軟なクエリ操作がない代わりに，スケーラビリティと可用性の向上に特化したデータ管理システムである[Cattel 2011]．この NoSQL のなかで，Key Value Store と呼ばれるタイプの okuyama[okuyama]を採用し検証を実施した．その内容と結果を以下に示す．

なお本稿で紹介する検証は，okuyama の使用方法や設定などの習得及び確認を主眼に実施したもので，事前テスト的な位置付けである点を断っておきたい．また，使用した okuyama のバージョンも古く，検証データ数やノード数も想定する 40 億ページと比較して小規模であり，あくまでも傾向をつかむ程度のレベルである．

4.2 検証環境

以下に検証環境として，サーバ構成を表 2，サーバのマシンスペックを表 3，okuyama のバージョンと設定を表 4，検証データの内容を表 5 に示す．

表 2 サーバ構成

項目	構成
マスタノード	2ノード，2プロセス/ノード
データノード	4ノード（メイン）＋ 4ノード（サブ），2プロセス/ノード

表 3 マシンスペック

項目	内容
CPU	Intel Xeon 5650 2.66GHz x 2（12core/node）
Memory	72Gbyte
Storage	2Tbyte x 22（RAID-6）

表 4 okuyama のバージョンと設定

項目	内容	説明
バージョン	0.8.6	—
memoryMode	False	データを永続化する。
dataMemory	False	データをファイルに保存する。
keyMemory	True	キーをメモリに保存する。

表 5 検証データの内容

項目	内容
キー	URL
値	URL、プロトコル、文字コード、ページ言語、MD5SUM値、Last-modified値などのメタデータをタブ区切りで結合した文字列

4.3 検証内容と結果

前節で示した検証環境を用いて，URL データの Insert テストと Search/Replace テストを実施した．以下にテスト内容と結果を示す．

(1) Insert テスト

約 12 億件の URL データを用いて，クライアントの 4 プロセスから，各々異なるマスタノードプロセスに対して URL データを Insert するテストを実施した．なおテスト開始時点において，okuyama に登録されているデータは 0 件である．

結果の一部を図 3 に示す．縦軸は 1 万件あたりの登録時間，横軸は累積の登録件数を表す．7 億件あたりまでは徐々に処理時間が増加しているが，その後は多少の増減があるものの基本的には 1 万件 40～50 秒の間で安定したパフォーマンスが計測された．時折，処理に 90 秒弱の時間がかかっている箇所が見受けられる．これは Java のガベージコレクションが働いているためと推察される．

(2) Search/Replace テスト

Insert テストにて登録した約 12 億件の URL データを用いて，クライアント上の 4 プロセスから，各々異なるマスタノードのプロセスに対して URL データを Search し，値を Replace するテストを実施した．

結果の一部を図 4 に示す．処理時間は 1 万件 150 秒から 300 秒までと，比較的幅が大きく開いている．それぞれ異なるマスタノードにアクセスしている 3 プロセスがほぼ同一のタイミング

で変動していることより，データノード側の原因と推察されるが，正確にはより詳しく調査する必要があると思われる．なおプロセス 1 については，他の 3 プロセスとパフォーマンスの傾向が異なり，またテスト途中で異常終了したことを考慮すると，テストスクリプトに何らかの不具合があったと推察される．

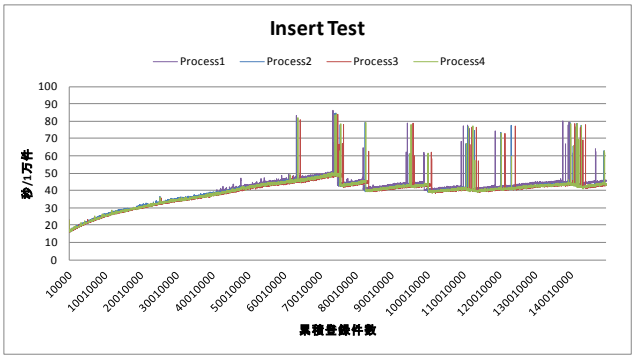

図 3 Insert テスト結果

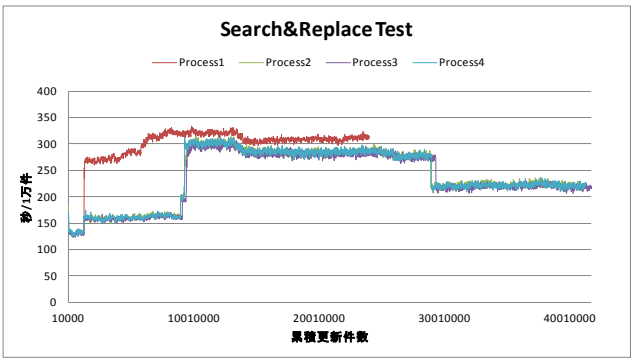

図 4 Search/Replace テスト結果

4.4 考察

データ入出力の高速化という点においては，Insert 能力が 50 秒/1 万ページとすると，約 1,700 万ページ/日となり，目標値に近い値が求められた．しかしながら，Search/Replace においては，300 秒/1 万ページならば約 290 万/日となり目標値を大きく下回る結果となった．

本検証では旧バージョンの okuyama を利用しており，ノード数も 10 ノードと小規模であるため，NoSQL のスケーラビリティが十二分に発揮されているとは言い難い．最新の okuyama を利用し，80 台以上の計算機を利用した別のテスト結果[岩瀬 2012]では高パフォーマンスが計測されているため，これを参考により本番環境に近い状態での検証を実施する必要があると言える．

5. おわりに

本稿においては，我々がこれまでに行ってきたクローリングの取り組みを紹介するとともに，ビッグデータ時代を見据えた次期クローラの構想と簡易な検証の結果を示した．今後は，より実際的な規模と構成による URL データ管理検証の実施と，その他の課題に対する検討を進めていきたい．

参考文献

[Bar-Yossef 2007] Ziv Bar-Yossef, Idit Keidar and Uri Schonfeld : Do Not Crawl in the DUST: Different URLs with Similar Text, in Proceedings of the 16th International World Wide Web Conference, 2007.

[Boldi 2004] Paolo Boldi, Bruno Codenotti, Massimo Santini and Sebastiano Vigna : UbiCrawler: A Scalable Fully Distributed Web Crawler, Software – Practice & Experience, 2004.

[Broder 1997] Andrei Z. Broder, Steven C. Glassman, Mark S. Manasse and Geoffrey Zweig : Syntactic Clustering of the Web, SRC Techical Note, Systems Research Center, 1997.

[Cattel 2011] Rick Cattel : Scalable SQL and NoSQL Data Stores, ACM SIGMOD Record, ACM, 2011.

[Dasgupta 2008] Anirban Dasgupta, Ravi Kumar and Amit Sasturkar : De-duping URLs via Rewrite Rules, KDD'08, ACM, 2008.

[Hurst 2009] Matthew Hurst, Alexey Maykov : Social Streams Blog Crawler, IEEE International Conference on Data Engineering, IEEE, 2009.

[JGN-X] JGN-X ホームページ, http://www.jgn.nict.go.jp/

[Lee 2008] Hsin-Tsang, Derek Leonard, Xiaoming Wang and Dmitri Loguinov : IRLbot: Scaling to 6 Billion Pages and Beyond, WWW2008, ACM, 2008.

[Najork 2001] Marc Najork, Allan Heydon, High-Performance Web Crawling, SRC Research Report, Systems Research Center, 2001.

[okuyama] okuyama 公式サイト, http://okuyama-project.com/ja/index.html

[O'Reilly 2012] O'Reilly Radar Team : Planning for Big Data –A CIO's Handbook to the Changing Data Landscape, O'Reilly Media, Inc, 2012.

[WISDOM] NICT 情報分析システム WISDOM, http://wisdom-nict.jp/

[Yunhua 2011] GU Yunhua, SHEN Shu, ZHENG Guansheng : Appjlication of NoSQL Database in Web Crawling, International Journal of Digital Content Technology and its Applications, AICIT, 2011.

[岩瀬 2012] 岩瀬他: ビッグデータ・イン・メモリ, 2012 年度人工知能学会全国大会, 1A1-OS-17a-3, 2012.

[岩爪 2012] 岩爪他: アジア情報 HUB プロジェクト(第一報), 2012 年度人工知能学会全国大会, 1A1-OS-17a-1, 2012.

[田村 2008] 田村 孝之, 喜連川 優 : 大規模 Web アーカイブ更新クローラにおける Web サーバアクセススケジューリング手法, 第 6 回日本データベース学会年次大会(DEWS2008), 2008.

[田村 2009] 田村 孝之, 喜連川 優 : 大規模 Web アーカイブ更新のための階層的スケジューリング手法, 情報処理学会誌データベース, 一般社団法人情報処理学会,2009.

[森本 2011] 森本 浩介, 上田 高徳, 打田 研二, 山名 早人 : ウェブサーバへの最短訪問間隔を保障する時間計算量が O(1) のウェブクローリングスケジューラ, 第 3 回データ工学と情報マネジメントに関するフォーラム (DEIM), 2011.

[山田 2004] 山田 雅信, 高橋 俊行, 田浦 健次郎, 近山 隆 : インクリメンタル PageRank による重要 Web ページの効率的な収集戦略, 情報処理学会論文誌 コンピューティングシステム, 一般社団法人情報処理学会, 2004.