様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成　２９　年　　６　月　　１　日現在

機関番号：　１４７０１
研究種目：　基盤研究(C)（一般）
研究期間：　2014～2016
課題番号：　２６３３０３６５

研究課題名（和文）全文検索サービスの自動生成システム

研究課題名（英文）An Automatic Generation System for Full-Text Retrieval Services

研究代表者
村川　猛彦（田中猛彦）（Murakawa, Takehiko）

和歌山大学・システム工学部・准教授

研究者番号：９０３０４１５４

交付決定額（研究期間全体）：（直接経費）　　3,200,000円

研究成果の概要（和文）：検索サービス構築の支援にあたり，全文検索エンジンの検索性能比較，および効率の良い全文検索サービスの自動生成を行うシステムを開発した．
検索性能比較に関しては，無償の全文検索エンジンのいくつかのバージョンについて，1台の計算機上に実行環境を構築して共通の文書を登録した．Apache Solrではバージョンにより該当件数が異なる事例を発見した．
自動生成に関しては，Dockerコンテナとして動作する，全文検索サービス生成プログラムを開発した．これにより保有コンテンツの公開やクラウドサービスへの送付をすることなく，インデックスが作られ，Webブラウザを介して全文検索が可能となった．

研究成果の概要（英文）：Aiming at supporting the information retrieval service development, we constructed the systems for comparing the search performance and for automatic generation of full-text services.
Execution environments were set in a server computer where several versions of the same free search engine were introduced and the common documents were registered respectively. We found search terms that brought inconsistent numbers of relevant documents.
A program that works as a Docker container and generates information retrieval services was also developed. When one sends documents, the program makes a set of files for an information retrieval service. By using this program, the registered documents are indexed and we can afford a full-text search without releasing the possessed contents to the public or uploading them on cloud services.

研究分野：　データ工学

キーワード：　情報検索　全文検索　Webサービス　性能比較　スタンドアロン　検索エンジン

１．研究開始当初の背景

インターネット上の情報検索には，Google などが提供する検索サービス（サーチエンジン）が広く使われている．それとは別に，企業・組織・個人が有する，必ずしも全体の公開が望まれないコンテンツを対象とした，全文検索サービスの開発や提供も多く見られる．独自サービスの構築により，瞬時かつ漏れのない検索結果の提供や，アクセスログを通した詳細な分析も可能となる．構築にあたっては，Web サーバと既存の全文検索エンジンを導入し，Web アプリケーションとして実現するのが一般的である．本研究課題において全文検索エンジンとは，Google などのサーチエンジンではなく，全文検索を行うソフトウェアを指す．

Groonga や Elasticsearch など，無償の全文検索エンジンも公開されており，手軽に利用できるようになっている．その検索性能は，採用する全文検索エンジンに大きく依存する．全文検索サービスの構築に不慣れな開発者にとっては，どのような全文検索エンジンを使用すればよいか分からないという問題点がある．たまたま知って使用した全文検索エンジンが，想定する検索サービスに対して機能不足または機能超過という事態も考えられる．

本研究課題では，検索サービス構築の支援にあたり，(1)全文検索エンジンについて，同一で異なるバージョン間の検索性能を比較すること，および(2)検索対象文書群を入力に与えれば，効率良く全文検索サービスを自動生成すること，がこれまで十分に整備されてこなかった点を考慮し，それぞれの環境構築ならびにシステム開発に取り組むこととした．

２．研究の目的

(1) 全文検索エンジンの検索性能比較

検索サービスの開発においては，全文検索エンジンの選定が重要となる．選定支援に関する先行研究として，スケーラビリティ評価や検索性能比較が試みられている[1][2].

しかしながら，全文検索エンジンの検索性能比較は，同一ソフトウェアの異なるバージョン間でも行う必要がある．例えば Groonga は，バージョン更新を頻繁に行っている全文検索エンジンであるが，開発者にとっては，どのバージョンを採用すれば良いか，運用開始後のアップグレードに追随すべきか否かといった課題が発生する．不用意な追随によって，検索サービスが動かなくなったり，検索性能が劣化したりすることは，運用上，避けなければならない．

そこで上記の課題を踏まえ，検索性能の比較を行った．Apache Solr および Groonga の主要バージョンについて，共通の文書群に対する検索環境を構築し，同一の検索語で該当文書数を求めた．

(2) 全文検索サービス自動生成

全文検索サービスの開発・運用・保守において，わずかな操作で，サービスとして動作するものが生成できれば，動作確認およびデバッグ・改善を通じたサービス品質の向上が期待できる．そこで全文検索サービスの自動生成を実施できるシステムの開発を行うこととした．

その実現にあたり，Docker を採用することとした．Docker は，コンテナ型仮想化技術を使ったアプリケーションの実行環境を構築・運用するためのプラットフォームである．コンテナ型仮想化では，ホスト OS の上に論理的な区画（コンテナ）を作り，アプリケーションを動作させるのに必要なライブラリやファイルなどをコンテナ内に閉じ込め，あたかも個別のサーバのように使うことができる．これにより，ホスト型仮想化，ハイパバイザ型仮想化といった他の仮想化に比べ，軽量・高速で動作するという特長がある．

Docker は Linux 上で動作するものであるが，Windows や macOS 向けには Docker Toolbox が提供されており，VirtualBox による仮想環境で専用の Linux ディストリビューションを稼働させることで，計算機内部で Linux およびコンテナを動かす．また Amazon EC2 や Google Cloud Platform などのクラウド環境でも動作する．

以上を踏まえ，本研究では，Docker コンテナとして動作する，全文検索サービス生成プログラムを開発することとした．検索対象文書を送ると，全文検索サービスを稼働させるのに必要なファイルを生成する．これにより保有コンテンツの公開やクラウドサービスへの送付をすることなく，インデックスが作られ，Web ブラウザ上で全文検索が可能となる．

３．研究の方法

(1) 全文検索エンジンの検索性能比較

比較に使用した Apache Solr のバージョンは，1.4.1，3.1.0，3.6.2，4.6.0，4.9.0 の 5 種類である．Linux が稼働する 1 台の計算機上にすべてのファイルを配置した．検索対象文書群として，古典籍の書誌情報に関する約 12,000 個のファイルを使用した．バージョンごとにポート番号を変えて起動し，全文検索が行えるよう文書登録を行った．スクリプトファイルを作成し，検索語を与えて実行すると，各バージョンおよび grep コマンドで検索を行い，それぞれの該当文書数を出力するようにした．

Groonga は毎月 29 日（肉の日）に新たなバージョンがリリースされている．比較環境の構築にあたっては Groonga の機能を Ruby から利用するためのライブラリである Rroonga のバージョン 4.0.8, 5.0.0, 5.0.1, 5.0.2, 5.0.3, 5.0.4，5.0.5，5.0.8，5.0.9 を用いた．コンテナ仮想化プラットフォームである Docker を用いて，各バージョンの Rroonga および全文検索用データベースのコンテナ化を試みた．

(2) 全文検索サービス自動生成

全文検索サービスの生成を行う Web アプリケーションの Docker コンテナを開発した．

コンテナ作成のための指示ファイル（Dockerfile），全文検索サービスで使用されるいくつかのファイルを zip 形式でアーカイブしたファイル，および自動生成を行う Ruby スクリプトなどで構成されている．

この Web アプリケーションの画面例を図 1 に示す．

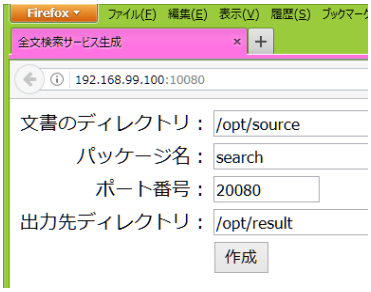

図 1 全文検索生成の入力画面例

検索対象文書群のディレクトリ名，パッケージ名，ポート番号，出力ディレクトリ名を入力できる．「作成」ボタンを押すと，出力ディレクトリに，全文検索サービスを Docker で稼働させるためのファイルが生成される．ファイルは 1 つの docker-compose.yml，全文検索エンジン（Elasticsearch）と Web サーバ（nginx）に関する 2 つの Dockerfile，およびイメージ生成に必要なファイルで構成され，連携させて全文検索サービスとして稼働する．なお，検索対象文書群のディレクトリ名および出力ディレクトリ名は，docker コマンドに特定のオプションを付けることにより，ホスト OS 上のディレクトリに対応付けることができる．これにより，ブラウザと Docker コンテナとの間でファイルの受渡しが不要となる．

4．研究成果

(1) 全文検索エンジンの検索性能比較

Apache Solr の各バージョンに対し，与えた検索語に対する該当文書数を表 1 に示す．バージョン 3.6.2，4.6.0，4.9.0 の値はどれも同じだったため「3.6.2 以降」とした．検索語が 2 文字の場合は，バージョン 3.1.0 を除き，値が同じであること，バージョン 3.6.2 以降では字数を増やすことで該当文書数が増える事例があること，バージョン 1.4.1 は字数によらず grep と同じであることが分かった．

表 1 各検索語の該当文書数

	1.4.1	3.1.0	3.6.2 以降	grep
山寺	2,116	0	2,116	2,116
石山寺	1,282	560	2,116	1,282
高山寺	828	581	2,116	828
室町	534	17	534	534
室町時代	472	113	4,284	472

バージョン 3.6.2 以降では，bigram に区切った上でいわゆる OR 検索（石山寺であれば「石山 OR 山寺」）を行っていると考えられる．引用符を付けて"石山寺"を検索語とすると，バージョン 1.4.1 および grep と該当文書数が一致した．

Groonga の各バージョンに対し，全文検索用データベースのファイルは，各バージョン共通で 7 つ，バージョン 5.0.9 のみさらに 1 つで構成されていた．次に，各バージョンのコンテナ内にあるデータベースのファイルを，ホスト OS にコピーし，他のバージョンのコンテナでマウントを試みた．各バージョンのデータベースを読み出して検索を行ったところ，簡単な検索語に対して該当文書数は同一となり，調査したバージョンでは，検索性能に差が見られないことが判明した．

(2) 全文検索サービス自動生成

生成した検索サービスについて説明する．ブラウザで所定のページを開くと，検索フォーム，登録件数等が表示される（図 2）．

図 2 全文検索サービス画面例（初期状態）

検索フォームは，検索対象として「url」「title」「body」のいずれかを選ぶドロップダウンリストと，テキスト入力欄，および検索ボタンで構成される．ユーザが選択および記述を行い，ボタンを押すと，該当件数および上位 10 件の記事が表示される（図 3）．

図 3 全文検索サービス画面例（検索結果）

61,612 件の防災ブログ記事を検索対象文書群に用いて，全文検索サービス実現の可否を検証するとともにデータサイズや処理時間の比較を行った． Ubuntu

16.04 の Linux サーバ（CPU：Intel Xeon E5-2650 v2（2.60GHz，16 スレッド），主記憶：64GB）上で Docker ホストを稼働し処理させた．

全件のファイルより，docker-compose.yml 他を作成した場合では，ビルド時に途中終了した．文書登録時にエラーが発生しており，登録用のファイルを 10,000 件ごとに分割してから，順に登録するよう，Dockerfile を修正してから再度ビルドを行うと，無事に Docker イメージが作られ，起動した全文検索サービスは，当該サーバより利用できた．

文書群の先頭より 100 件，1,000 件，2,000 件，4,000 件，8,000 件，20,000 件を取り出し，4 節に述べた方法でファイルを生成したところ，いずれもビルドができ，全文検索サービスも正常に動作した．件数ごとの，文書と Docker イメージのサイズを，図 4 に示す．文書サイズは件数に比例しており，Docker イメージサイズはほぼ線形関数となっている．後者の傾きがより大きいのは，インデックスの増大を示している．

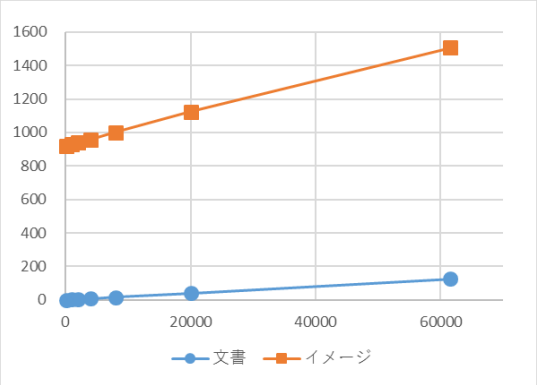

図 4 件数ごとのデータ量(MB)

ビルドに要した時間は，100 件から 8,000 件までは 31～40 秒で差がなかったものの，20,000 件では 49 秒となった．分割して 61,612 件を登録した場合では 76 秒を要した．

以上より，開発したシステムでは妥当な時間的コスト等で全文検索サービスを生成できるものと考えられる．

〈引用文献〉

[1] 早坂良太；林貴宏；尾内理紀夫：「規模の拡張に対応した検索エンジンの開発」，コンピュータ ソフトウェア，Vol.26，No.4，pp.138-156，2009.

[2] 河中健馬，渡上将治，村川猛彦，中川優：全文検索エンジン選定支援システムの構築，情報知識学会誌，Vol.21， No.2，pp.191-196 (2011).

５．主な発表論文等

〔雑誌論文〕（計 8 件）

① 村川猛彦：全文検索サービス自動生成の試み，情報知識学会誌，Vol.27，No.2，pp.219-224，2017．査読無

② 村川猛彦：災害記事データベースの構築および応用―記事収集，全文検索，およびテキスト分析―，和歌山大学災害科学教育研究センター研究報告，Vol.1，No.1，pp.11-16，2017．査読無

③ Takehiko Murakawa：Constructing Performance Comparison Environment of Search Engines，Proceedings of Fourth International Conference on Advances in Computing，Communication and Information Technology (CCIT 2016)，Birmingham，United Kingdom，pp.90-93，2016．査読有

〔学会発表〕（計 5 件）

① 永井謙也，村川猛彦：個人利用に適した情報管理システムにおけるユーザインタフェースの検討，2016 年度情報処理学会関西支部支部大会，大阪大学中之島センター（大阪市），2016.

② 村川猛彦：全文検索エンジン Groonga のバージョン間比較について，2016 年電子情報通信学会総合大会，九州大学伊都キャンパス（福岡市），2016.

③ 村川猛彦，藤井浩平：全文検索エンジンの検索性能比較について，2015 年電子情報通信学会総合大会，立命館大学びわこ・くさつキャンパス（滋賀県草津市），2015.

④ 野田長寛，村川猛彦：個人利用向けの情報管理システムの構築，2015 年電子情報通信学会総合大会，立命館大学びわこ・くさつキャンパス（滋賀県草津市），2015.

〔その他〕

① 検索の達人を目指せ！，2016 年度和歌山大学公開体験学習会
http://www.crea.wakayama-u.ac.jp/event/taiken2016/docs/p12.pdf

② 検索の達人を目指せ！，2015 年度和歌山大学公開体験学習会
http://www.crea.wakayama-u.ac.jp/event/taiken2015/docs/p10.pdf

③ 検索の達人を目指せ！，2014 年度和歌山大学公開体験学習会
http://www.crea.wakayama-u.ac.jp/event/taiken2014/docs/p11.pdf

６．研究組織

(1)研究代表者

村川 猛彦（田中猛彦）
(MURAKAWA, Takehiko)
和歌山大学・システム工学部・准教授
研究者番号：90304154