6X-7

協調サーチエンジンのためのインデクサの開発

山本 崇† 佐藤 永欣† 西田 喜裕† 上原 稔† 森 秀樹† 酒井 義文†

†東洋大学工学部情報工学科

1 はじめに

我々は我々が開発している協調サーチエンジン (Cooperative Search Engine、以下 CSE) においてインデックス更新時間の短縮化により、更新間隔を短縮した運用が可能なサーチエンジンを目指している。CSE とは複数の局所サーチエンジンを構築し、これらを協調動作させ検索作業や、インデックスファイルの更新作業を行なうものである。

本稿では、インデックス更新時における CSE の利点、インデックス更新手順、及び、CSE における運用に特化したインデクサの機能について述べる。

2 関連研究

分散型検索エンジンに関する研究は、各所で行なわれている。検索対象となる文書の収集部分の分散化は JEIDA[1] 等で行われている。収集部分の分散化は、多数の文書を取得するためにかかる時間の短縮を主な目的にしている。google[2][3] では、URL のリストを送信する URL Server、取得した情報を保存する Store Server、HTML 文書を取得する crawler を用いて文書の収集を行なっている。この crawler を分散させることで、分散収集を行なっている。また、Indexer、URL Resolver、sorter の各構成要素が協調動作し、検索用のデータベースの作成を行っている。

3 CSE のインデックス作成時の動作

3.1 CSE の構成

CSE には、検索動作や、インデックスファイルの更新作業を行う Local Meta Search Engine(LMSE)、CSE 内の各 LMSE に関する情報を管理する Location Server(LS)、検索時に LMSE からの要求を受け、LS や LMSE との通信を行ない検索結果を取得、キャッシュする、Cache Server(CS) の三つの構成要素がある。LMSE は内部的に検索エンジンを用いる構成をしている。今回、CSE での運用に特化した検索エンジン (Local HTML Document Search Engine,LHSE) の作成を行なった。また、インデクサを操作する機能を持つ Update Manager(UM) を作成した。UM は LMSE の構成要素の一部とみなす。本稿では、UM と LHSE のインデクサの持つ機能を、CSE におけるインデクサの持つ機能とする。

3.2 インデックス更新の過程

CSE におけるインデックス更新作業の概略図を図 1 に示す。

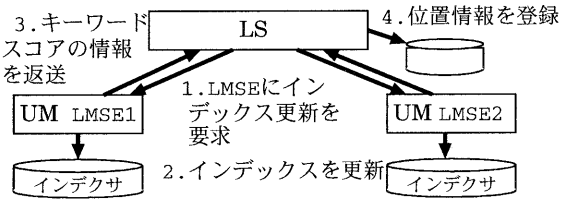

図 1: CSE の検索時の動作

更新作業は LS が主体となり処理を行なう。これは CSE 全体を制御するのに都合がよいからである。更新作業は LMSE と LS が処理を行なう。更新作業は以下のようにして行なわれる。

1. LS は LMSE にインデックスの更新を要求する。
2. 要求を受けた LMSE はインデックス更新作業を行ない、キーワードとスコアの情報を抽出する。
3. LMSE はインデックスの更新が終ったら、抽出した情報を LS へ送信する。
4. LS は送信されてきた情報を元に、検索動作時に使用する、位置情報を更新する。

4 CSE の利点

インデックス更新時の動作は、(1) 文書の取得 (2) インデックスファイルの作成、の二つに分けて考える事ができる。文書の取得は、一般的なサーチエンジンでは、HTTP を用い、ネットワークを介して収集する。この時、ネットワークや収集対象の Web サーバに過負荷をかけないようにする必要があり、短期間に多量の文書に関する情報を収集する事は困難である。そのため、文書の内容が変更されても、その内容がサーチエンジンに反映されるまで時間がかかってしまう。CSE において、この問題を解決する方法について述べる。

4.1 文書収集の高速化

CSE では、検索対象となる文書を保持するサーバ毎に LMSE を構築するという構成をとり、文書収集時には、各文書にはローカルファイルとしてアクセスす

Update process and indexer in Cooperative Search Engine
Takashi Yamamoto
Department of Information and Computer Sciences, Faculty of Engineering,Toyo University

ることにより短期間に多量の文書に関する情報の収集を行なっている。LHSE は前回更新時より変更のあった文書を探索する機能を有する。この事により、文書に変更が加えられてから短期間で、更新された内容が反映された検索結果を得たり、削除された文書に関する情報を検索結果から除外することができる。

4.2 インデックスファイル作成の高速化

4.2.1 NFS 共有ファイルを用いた高速化

インデックスファイルの作成に時間がかかってしまうと、収集した情報を短期間で反映させることができない。この問題の解決するため、LHSE は Web サーバ以下に、いくつかのマシンが存在し、NFS によってファイルを共有する環境において動作する並列にインデックスを作成する機能を有する。このシステムの概要を以下に示す。

- 一つの master プロセスと複数の slave プロセスによって動作する。
- master プロセスは slave プロセスに仕事を割り振る役割を持つ。slave プロセスは指示に従い、インデックスの作成を行なう役割を持つ。
- NFS 共有ファイルにより同期、通信を行なう。

4.2.2 処理の外部依頼による高速化

前述した並列インデックス作成システムとは別に、自分以外の LMSE に処理の一部を依頼する機能を有する。このシステムの概要を以下に示す。

- 各 LMSE は、自分自身の作業の推定所要時間を把握し、LS への通知を行なっている。LS は、すべての LMSE の推定所要時間を把握している。
- LS は、処理に時間がかかると思われる LMSE へ他の LMSE を紹介する。
- LS から紹介された LMSE は、その LMSE へ処理の依頼を行なう。

5 評価

今回作成したシステムを用いて実験を行なった結果を示す。実験には本学の工学部計算機センター (cc.eng.toyo.ac.jp、以下 cc.eng) の HTML 文書を対象に測定を行なった。cc.eng は 17 台の ss20 で構成されており、16070 個の HTML 文書が保持されている。

今回実験を行なう前段階として、一般的な HTML 文書を対象に Web サーバの全ユーザが頻繁に更新を行なうユーザであるとした場合、全体の文書のうち比較的最近更新された文書の占める割合、つまりインデックス更新時に、処理の対象となるファイルの割合が、どの程度になるのか調査を行なった。比較的頻繁に更新を行なっていると考えられるユーザの HTML 文書に注目し、測定した結果を表.1 に示す。

表 1: 頻繁に更新するユーザの最終更新日

ユーザ数	全文書数	7 日以内	1 日以内
989	61640	7377(12.0%)	1458 (2.4%)

表 2: cc.eng における更新時間

	1 プロセス	並列 (17 プロセス)
所要時間	34:39	11:15

この結果をもとに、今回の cc.eng における実験では 1 日に 1 回インデックスの更新を行なった場合を想定し cc.eng に保持されている HTML 文書の 2.4%に相当する 386 個のファイルを対象にインデックス更新を行なった。ファイルのサイズの合計は 9776435byte であった。まず、cc.eng 上に LMSE を構築し、並列インデクシングを行なわなかった場合と 17 台のマシンでそれぞれ一つの slave プロセスを動作させ並列インデクシングを行なった場合の所要時間を表.2 に示す。

次に、cc.eng で並列インデクシングを用いることができない環境と仮定し lute,mutsuki の二つの計算機上に LMSE を構築し、処理の外部依頼を行なった結果を測定した。lute,mutsuki は cc.eng の更新作業時には、更新作業を行なっておらず、要求があれば即、依頼を受けられる状態で測定を行なった。その結果、更新所要時間は 11 分 44 秒となった。

6 まとめと今後の課題

本稿では、インデックス更新時における CSE の利点を述べ、短期間でインデックスの更新を完了させることを目的とした CSE の更新手順、及びインデクサの機能について述べた。

参考文献

[1] 『次世代分散型情報検索システムに関する調査研究報告書』 http://www.jeida.or.jp/committee/jisedai/top.html (1997)

[2] *Google* http://www.google.com

[3] Sergey Brin,Lawrence Page 『*The Anatomy of a Large-Scale Hypertextual Web Search Engine*』, Seventh International World Wide Web Conference http://www7.scu.edu.au/programme/fullpapers/1921/com1921.htm

[4] 高林哲 『全文検索システム *Namazu*』 http://openlab.ring.gr.jp/Nnamazu/