小規模な分散 Web ロボットの最適化に関する一考察*

幸谷智紀†
静岡理工科大学‡

1. 初めに

教育としてサーチエンジンを構築する試みはテネシー大学で既に行われコースウェアが出版されているが，これは線型計算と情報技術との橋渡しを目指したものであるのに対し，我々の目指すのは 3 層 Web プログラミング技術の応用としての，教育及び応用研究用のサーチエンジンシステムである。最終的にはこのシステムを様々な用途に応用していくことを予定しており，従って，用途に合わせた実践的なサーチエンジンシステムとその技術を学ぶための教材開発は今後も継続していかねばならない。そのためには，プロトタイプ版でも現状の商用サーチエンジンの基本機能を持つものを作り上げ，その性能評価を行ってシステムの限界を知っておく必要がある。

本稿ではこのサーチエンジンシステムの概要を説明し，その性能評価を行った結果を示す。

2. サーチエンジンシステムの概要

最終的に作成したサーチエンジンシステムは Fig.1 のような構成になっている。最小限の 3 つのコンポーネント構成で実現できるようにしてある。各コンポーネントは次のような機能を持つスクリプトもしくはソフトウェアを使用している。

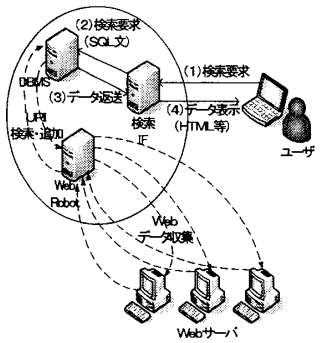

Fig. 1: サーチエンジンのシステム

DBMS フリーで使用できる MySQL[5] を採用する。収集した全てのデータはここに格納される。

検索用 IF 現状ではまだ仕様が固まっていないが，最終的には PHP スクリプトによる実装を予定している。検索結果は後述するランキングの高い順に表示されるようになる。

Web ロボット Perl による Web データ収集プログラムで，教育用途向けのものは現在のところ 400 行程度の規模である。robots.txt による収集制御に対応している。

図では DBMS，検索用 IF，Web ロボットを別マシンで動作させているが，これらを 1 台のマシン上で動作させることも可能である。これらのうち，特に重要な機能である Web ロボットの基本動作は

1. 既知の URI にアクセスし，テキストデータのみを取得
2. 取得したテキストデータ (HTML など) から新たな URI を取得 ⇒ 1 へ

というきわめて単純なものであるが，一歩間違えると悪質なアタックツールにもなりかねない危険な側面を持つ。そのため，次の機能を持つことが，安全な Web ロボットの運用には不可欠である。

- 同一 Web サーバにアクセスする際には，必ず十数秒以上の間隔をあける。
- robots.txt[4] によるアクセス制御に従う。

現在のところ，これら全ての機能を満足する機能を備えた言語環境はそう多くない。そのため，パフォーマンスの向上に関しては難はあるものの，今のところは Perl とそのモジュール群を利用して Web ロボットを実装してある。この結果，スクリプトサイズも比較的少ない行数に抑えられている。

*Trial Implementation of Small-sized Web Robot for Distributed Enviromnent

†Tomonori KOUYA
‡Shizuoka Institute of Science and Technology

3. ベンチマーク結果

以上のような機能を備えたサーチエンジンシステムを動作させた結果をここでは示す。

3.1 本学内 LAN からの URI 収集と WebRank 計算結果

まず静岡理工科大学研究実験棟 LAN から URI 収集を行った結果を示す。条件は以下の通りである。

- 静岡理工科大学 LAN からアクセスしたので，Firewall と研究実習棟 Proxy の 2 段の Proxy を介して外部 7Mbps 回線を用いたことになる。
- 使用ハードウェア・ソフトウェア環境は次の通り

  CPU Intel Pentium 4 (1.8GHz)
  RAM 1GB
  HDD 40GB
  OS CentOS 5 x86_32 版
- WebRobot(r3.pl) は前述したように Perl スクリプトとして実装し，1 プロセスのみ起動。
- Yahoo! Japan のトップページ (`http://www.yahoo.co.jp/`) からスタート。
- 1URI にアクセスするごとに 10 秒の wait を置く。
- robots.txt の指示に従ってアクセス制限をしているので，意図的に WebRobot を拒否しているサイトの情報は収集できない。
- テーブルに記憶する URI は接続確認されたもののみである。アクセスできなかったものは記憶しない。

この結果，一週間で約 3 万 URI が収集できた。更に収集を行い，最終的には約 8 万 URI 集めるに至った。

3.2 URI 収集性能

更なる高速化を目指すため，URI 収集においてボトルネックになりそうな部分の改善を行って，実験を行った。改良したのは以下の 3 点である。

- ネットワークをより高速な外部回線 (ベストエフォート 100Mbps) を持つ所に変更
- WebRobot の処理をマルチプロセス化し，並列動作を行うようにした
- Cache メモリを搭載した RAID カードを用いて SATA HDD を RAID0 構成にし，ローカルストレージの性能向上を図った

マルチプロセス化した WebRobot の動作概念図を Fig.2 に示す。

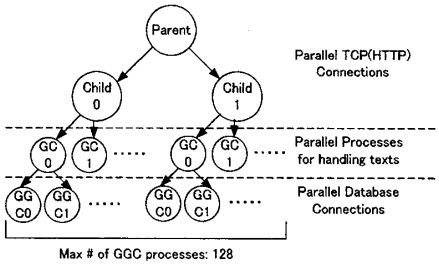

Fig. 2: Web Robot 処理の並列化

親 (Parent) プロセスは子 (Child) プロセスを fork を使って二つ起動した後はプログラム終了まで何もしない。子 (Child) プロセスは Web からのデータ収集を行うための TCP 接続 (HTTP) を担当し，接続が完了してから孫 (GC, Grandchild) プロセスを生成する。孫プロセスはひ孫 (GGC,Grand Grandchild) プロセスを生成し，テキスト処理を行った後，すぐに終了する。ひ孫プロセスはデータベースとの入出力を担当し，処理を終えた後はすぐに消えるようになっている。

また，並列動作の性能向上を確実化するために，Quad-core CPU を搭載した次のようなマシン上で動作実験を行った。

CPU Intel Core2Quad 6600
RAM 4GB
HDD 250GB
OS CentOS 5 x86_64 版

これに加えて，I/O 性能の向上を図るため，大容量の Cache メモリを搭載した RAID ボードを上記のマシンに乗せ，RAID0 の機能を用いた。この結果，I/O 性能のボトルネックが解消され，より大量の URI が収集できる可能性が増える。

以上のような環境で URI 収集数がどのように変化したかについては講演時に示す。

参考文献

1) 竹口友大・幸谷智紀, ランク機能付きサーチエンジンの開発および I/O ボトルネック対策, 第 70 回情報処理学会全国大会講演集, 2008.
2) 幸谷智紀・竹口友大, “サーチエンジンを作ろう”, 未公開テキスト.
3) M.W.Berry, M.Browne, Understanding Search Engine, SIAM, 1999.
4) `http://www.robotstxt.org/`
5) MySQL, `http://www.mysql.com/`