

L-14 高スループット更新のパイプライン化 Web ロボット

The Pipelined Web Robot for High Throughput Update

宇田川 稔 佐藤 永欣 上原 稔 酒井 義文 森 秀樹

Minoru Udagawa Nobuyoshi Sato Minoru Uehara Yoshifumi Sakai Hideki Mori

1. はじめに

インターネットの普及に伴い、Web ページの検索が重要になってきた。Web ページの検索はサーチエンジンによって行われる。既存のサーチエンジンは集中型アーキテクチャに基づく。集中型アーキテクチャでは、ロボットが文書を収集し、インデクサがそれらの文書からインデックス(転置ファイル)を生成する。このような方法では、更新時間が長くなり、新鮮な情報を検索することが困難である。ここで、更新時間とは、文書が公開されてからその文書を検索できるようになるまでの期間である。

分散型アーキテクチャに基づくサーチエンジンを分散型サーチエンジンという。分散型サーチエンジンでは、各 Web サーバ上で局所的にインデックスを作成するため、更新時間が短い。そこで、我々は新鮮な情報を検索するために分散型アーキテクチャに基づく「協調サーチエンジン(Cooperative Search Engine, CSE)」[1]を開発した。

協調サーチエンジンでのインデックス更新処理では、様々な文書収集アクセス法により高速収集を実現してきた。しかし、マシン負荷が大きかったり、タイムロスが生じたりと必ずしも最適化されているとは言えなかった。

そこで本研究では、今まで、別プロセスとされていた文書収集とインデックス作成を併せたインデックス更新処理を、パイプライン化を図る手法について提案する。

各プロセスをスレッドにより細分化することにより、ボトルネックの把握が容易になり、プロセス単位での負分散も可能になる。

2. 関連研究

文書の高速収集に関する研究はいくつか行われている。

早稲田大学の山名は、インターネットに対応した分散型 Web ロボットを開発している。これは、対象 Web サーバに対し、回線距離などを考慮し、各 Web ロボットに割り振り、協調して高速収集を実現するものである。[2]

NTT サイバースペース研究所の能登は、複数のマシン、プロセス、スレッドによる多重化により、利用できる回線帯域を上限として収集速度を必要に応じて向上できるスケラビリティを持った Web ロボットの設計を提案している。[3]

北海道大学の大島は、学内イントラネットに対応し、ローカルのサーバ群の情報は、情報の鮮度が重要になる点を考慮し、高速な文書収集ロボットの開発を行っている。学内の文書収集のスループットで 26.5[file/s]という研究結

果を発表している。[4]

しかし、これらの研究では文書収集のみのスループットである。インデックス更新には、文書収集に加え、インデックスの生成が必要である。次に実際のサーチエンジンのインデックス更新を紹介する。

Google[6]は、今日存在するサーチエンジンの中で、世界最大の情報を持ち、ヒット率も高いサーチエンジンとして知られている。Google は約 8000 台を用い、クラスタコンピューティングにより 7,000 万件/日の検索を 1 件あたりほぼ 0.5 秒で処理する。しかし、Google は、問題点も存在する。それは、更新時間の遅さである。Google では、ある Web ページが更新された場合、実際の検索に反映されるのに 60~90 日を要していると言われている。

CSE では組織内の Web サーバに高速にアクセスする方法として、現在まで以下の方法を取り入れ考えてきた。

直接アクセス方法は、NFS を含むファイルシステムによりアクセスする方法である。最も高速な方法であると言える。

アーカイブアクセス方法は、リモートの Web サーバに CGI を用いて複数のファイルをアーカイブかつ圧縮して転送したものを受信側で解凍する。アーカイブ、解凍に時間を要すると考えられるが、複数のファイルを転送するのでファイル当たりの通信遅延は比較的小さくとも言える。

最後に、一般的なサーチエンジンのアクセス方法として用いられる HTTP アクセスがある。

前 2 つは、実現すると高速な文書収集を実現できる。しかし、対象の Web サーバにアカウントのない場合、やはり HTTP アクセスで収集を行う必要がある。

3. クラスタマシンによる Web サーバ単位の更新実験

本学内の 54 Web サーバ、5570 ファイルを対象とし、多くのサーチエンジンで用いられている Wget や Htrack に代表されるロボット、Namazu、Freya、SGSE などのインデクサのアプリケーションを用いて、割り振り時に余メモリ容量が空いているノードに Web サーバ単位で過去に収集された文書容量から割り振り、インデックス更新処理、ボトルネックを測定した。実験の環境は、5 台のマシンを OS に Redhat6.2、クラスタ OS である SCore[6]を用いて、サーバマシン 1 台 (CPU:Pentium III 886MHz、Mem:128MB、NIC:Intel/pro 100+)、ノードマシン 4 台 (CPU:Celeron300MHz、Mem:128MB) とし構築した。各ノードは、クラスタサーバ内の文書保存ディレクトリ、インデックス作成ディレクトリ、文書キャッシュディレクトリを NFS により共有している。また、各ノードには一時的保存ディレクトリが存在し、収集文書や中間インデックスを一時的に貯めておくことができる。

この実験結果から、以下のことを理解した。

- 分散された組織内では、ネットワークの遅延が大きいに影響を及ぼす。
- 文書収集に対し、インデックス作成はマシンに依存し処理の上でボトルネックになってしまう。
- Web サーバ単位であると、処理コストをどう組み合わせても、ノードごとの均一なコストにするのは困難であり、完全な負荷分散の実現は難しい。
- 分散させて処理を行うと、最後にマージ処理が必要であり、そこで時間ロスをしてしまう。

4. インデックス更新パイプライン処理

インデックス更新をパイプライン化した構成を図 4 に示す。プログラムは、オブジェクト指向言語である RUBY を用いて構築し、各プロセスをスレッドにより細分化することによって、ボトルネックの把握が容易になる。

- 楕円は、タスクの始末端を表す。
- 四角は、プロセスを表す。
- ひし形は、分岐を表す。例えば、文書型の一致、対象ではないホスト、収集された文書の確認をとる。
- ○は、スレッドによる並列処理を表す。例えばインデックス作成時、文書の更新日時、パス、タイトル、要約、著者などは、逐次にする必要がなく、並列処理を行っている。
- 矢印は、処理の流れを表す。

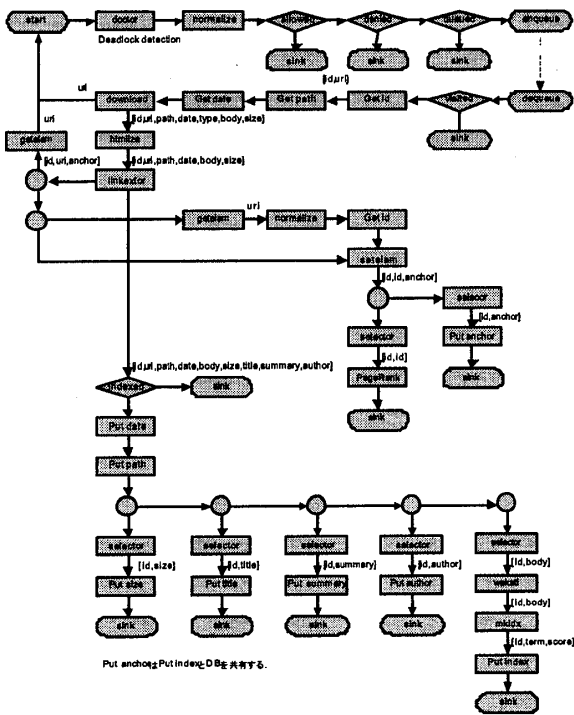


図 1. インデックス更新のパイプライン

5. 評価

ボトルネックを特定するため、実験を行った。実験には、PC/AT 互換機(CPU: Dual-Pentium III 933MHz, Mem: 1GB, NIC: Intel/pro 100+)のマシンを用いた。インデックス更新の各プロセスの 1 回の呼び出し当たり

の平均処理時間を表 1 に示す。

処理	処理時間[ms]
文書収集	1.56
インデックス作成	72.20

表 1. インデックス更新過程の処理時間

表 1 より、文書収集よりインデックス作成のプロセスのほうが、インデックス更新でボトルネックになっているのが分かる。そこで、インデックス作成内のプロセスを解析した。表 2 は、インデックス作成の各プロセスの 1 回の呼び出し当たりの平均処理時間である。

処理	処理時間[ms]
分かち書き	24.83
Score 取得	46.85
属性保存 (著者など)	0.40
インデックス保存	0.12

表 2. インデックス作成の各処理時間

表 2 より、分かち書きとスコア取得がボトルネックになっていることが分かった。分かち書きのほうは、1 文書ずつ分かち書きのアプリケーションを呼び出しているため、他のプロセス起動の時間がロスを生んでいる。また、スコア取得のほうは、行単位でタグ、語の解析処理を行っているため、処理時間が増えていると考えられる。

6. まとめ

今回は、インデックス更新のパイプライン化を図ることにより、処理上でのボトルネックを特定することができた。今後、このボトルネックの解消を中心に、前研究結果である各 Web サーバに対するネットワーク遅延の考慮、またクラスタマシンでの並列処理も視野に入れて、プロセス間の通信コストについて考えていきたい。

参考文献

- [1] 佐藤 永欣、上原 稔、酒井 義文、森 秀樹、"Fresh Information Retrieval in Cooperative Search Engine," In Proceedings of the ACIS 2nd International Conference on Software Engineering, Artificial Intelligence, Networking & Parallel/Distributed Computing(SNPDP'01), pp.104-111, (2001.8.20)
- [2] 山名早人等「分散型 WWW ロボットによる WWW 情報収集」 DEWS'98、<http://www.etl.go.jp/~yamana/Publications/ABST/D EWS98/24.htm>
- [3] 能登信晴等「スケーラブルな WWW 情報収集ロボットの設計と実装」 DPSWS (2000)
- [4] 大島利充、高井昌彰「マルチスレッドを用いた学内限定 Web 検索ロボット」情報処理学会第 62 回全国大会講演論文集, 3T-07, (2000)
- [5] 山名早人、近藤秀和「サーチエンジン Google」 IPSJ Magazine Vol.42 NO.8 Aug. 2001
- [6] SCore クラスタソフトウェア、<http://pdswww.rwcp.or.jp/home-j.html>