4U-03

キャッシュサーバによる協調サーチエンジンの高速化

西田 喜裕 † 山本 崇 † 佐藤永欣 † 上原 稔 † 森 秀樹 †

† 東洋大学工学部情報工学科

1 はじめに

我々は、従来の集中型のサーチエンジンの問題点である、検索対象となるWebサーバの数が増加した時の文書の収集、インデックス情報の作成と管理、検索時の負荷等を解決するために、分散した多数のサーチエンジンを協調して動作する協調サーチエンジン (Cooperative Search Engine,CSE) を開発した。しかし、従来のCSEでは、検索結果の操作が非効率的で、結果の1ページ目が表示されるまでの遅さや、その次の10件の表示 (Next10) といった問題があった。そこで、これらを解決するためにCSEにキャッシュサーバ (Cache Server, CS) を開発、導入する。

2 従来のCSE

2.1 従来のCSEの構成と動作

キャッシュサーバについて述べる前にキャッシュサーバ導入以前のCSEの構成と動作、問題点について述べる。まず、従来のCSEは大きくわけて二つの部分から構成される。ひとつはユーザからの検索要求の受け付け、検索動作を行うLocal Meta Search Engine(LMSE)、もう一つは検索式に対して適切なLMSEを選択するためのLocation Server(LS)である。CSEはこれらの構成要素が協調動作することにより一つの検索エンジンとして機能する。その様子を図.1に示す。

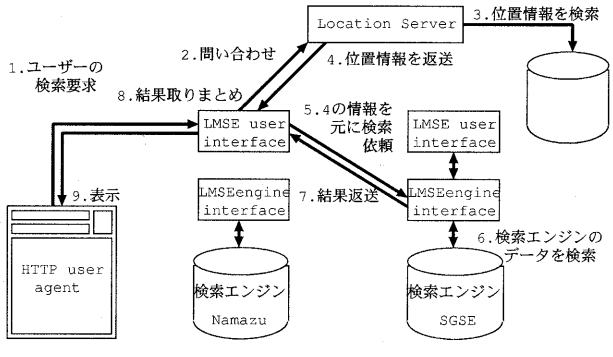

図 1: 検索時のCSEの動作図

Performance Improvement of Cooperative Search Engine by using Cache

Yoshihiro Nishida

Department of Information and Computer Sciences, Faculty of Engineering,Toyo University

表 1: 従来のCSEで検索にかかる時間

	時間	平均
検索した場合	13分59秒07	8秒39
検索しなかった場合	4分6秒05	2秒46
実行時間の差	9分53秒02	5秒93

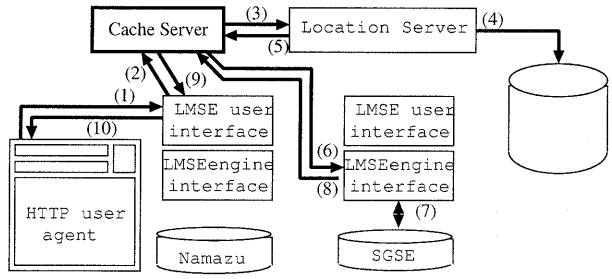

図 2: キャッシュサーバがある場合の検索動作

2.2 従来のCSEの評価

従来の構成のCSEで、実際に100回の検索を行なわせた場合と、検索のかわりに何もしないプロセスを実行させた場合にかかった時間を計測した。結果は表.1である。ここで、「実行時間の差」は検索とデータの送信にかかった時間とみてよい。これにより、ユーザの要求を受け取ってから結果を返すまでにかかる時間のうちの大部分が検索とデータの送信にかかっていると言える。

3 キャッシュサーバ

3.1 キャッシュサーバ導入後のCSEの検索時の動作

次に、本研究の目的であるキャッシュサーバをCSEに含めた時の検索時の動作を説明する。まず、キャッシュにヒットしなかった場合は図.2のようになっている。

1. (1)(2) あるLMSEのユーザインターフェース部がユーザから与えられた検索式を受け取るところまでは同じであるが、ここからロケーションサーバではなく、キャッシュサーバに検索式を送信する。
2. (3)(4) 与えられた検索式がキャッシュに無かった場合、つまり、その検索式が最近問合わされたものでなければ、キャッシュサーバは、ロケーションサーバに適切なLMSEのURLを問い合わせる。

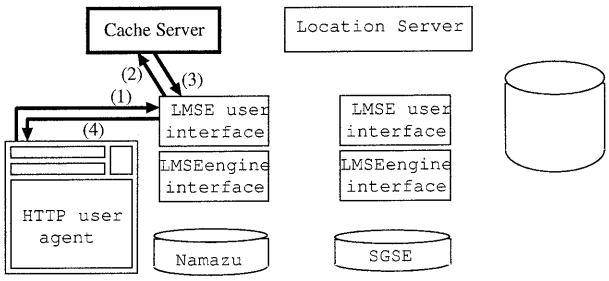

図 3: キャッシュサーバがあり、キャッシュにヒットした場合の検索動作

3. (5) ここでロケーションサーバより受け取った URL は、その検索式のためのキャッシュファイルに書き出しておく。
4. (6)(7)(8) 次に、その URL で表される LMSE に検索を行わせ、結果を受け取る。また同じファイルに検索結果を書き出しておく。
5. (9)(10) そして、元の LMSE に検索結果を送信し、その LMSE は結果を整形してユーザに送信する。

次に、与えられた検索式がキャッシュにあった場合は、図.3 のようになっている

1. (1)(2) 同様に、ユーザから検索式を与えられた LMSE のユーザインターフェース部は、キャッシュサーバに問い合わせる。
2. (3)(4) その検索式に対応するキャッシュファイルから検索結果を読み出し、指示された範囲の検索結果を返す。

このように、単一のロケーションサーバの負荷を軽減するとともに、LMSE の CGI プロセスを起動するオーバーヘッドを回避することでき、さらに、CSE で多く発生する通信量も減らすことができる。また、初めて問合わされた検索式の場合でも、検索結果を複数のページに分割した場合の 2 ページ目以降の検索結果の送信も、キャッシュを用いることにより容易に扱うことができる。

3.2 プロトコルの変更

LMSE のユーザインターフェース部に対して、「適切な LMSE の URL」ではなく、「検索結果」を送信する場合、LMSE のユーザインターフェース部を作りなおす必要があり、従来の CSE にそのまま組み込むことはできない。そこで新たに「SEARCH」メソッドを追加し、LMSE のユーザインターフェース部もそれに対応させた。

表 2: 従来の CSE と CS 導入後の CSE との比較

検索式	CGI 方式	CS 方式
mail(1 ページ目)	2.53 秒	1.45 秒
link(1 ページ目)	1.60 秒	1.17 秒
mail(2 ページ目)	0.04 秒	0.37 秒
link(2 ページ目)	0.03 秒	0.28 秒

SEARCH 開始位置 件数 検索式

ここで、開始位置は検索結果の表示開始位置、件数はその位置からの件数、検索式は、ユーザから与えられた検索式である。

3.3 キャッシュサーバを導入した CSE の検索の評価

ある検索式 (語)、ここでは「mail」と「link」という語を選んで、CGI を用いている従来の CSE とキャッシュサーバを導入した CSE に検索を行なわせ、1 ページ目と 2 ページ目の到着にかかった時間を計測した。また、その時のヒットした URL の数もあわせて表.2 に示した。従来の CSE、つまり CGI 方式の方よりも、CS を導入した CSE の方が、結果の 1 ページ目の到着が速くなっている。これは、1 ページ目に必要な量の結果を得られた時点でユーザに送っているからである。検索結果において、上位の結果は重要であるため、良い結果であるといえる。2 ページ目の到着時間に着目すると、従来の CGI 方式の方が短い時間となっているが、ユーザの感覚的にはこの差はほとんど変わらないものと言える。

4 まとめ

本研究で提案したキャッシュサーバを CSE に導入することにより、1 ページ目の到着までの時間を短縮するとともに、Next10 の問題も解決することができた。また、最近検索された式については、キャッシュから結果を返すことによって、高速に検索結果を返すことができるようになった。今後は LS と CS を統合し、それを分散させることによって、より有効にキャッシュを活用すること、また、CS による検索結果の公開、非公開の選択といった機能を追加することを予定している。

References

[1] 山本崇、佐藤永欣、西田喜裕、上原稔、森秀樹『協調検索エンジンの研究』,DICOMO'99,p169-174(1999.6)

[2] 西田喜裕、山本崇、佐藤永欣、上原稔、森秀樹『分散サーチエンジンにおける協調型検索』,SWoPP'99,p87-92(1999.8)