공학 연구논문집 제29권 제1호 pp. 1~13, 울산대학교.
Journal of Engineering Research Vol. 29, No. 1, pp. 1~13, 1998
University of Ulsan.

메타 검색 엔진과 Oracle ConText Option을 이용한 정보 검색 엔진에 관한 연구*

양영걸 · 옥철영
전자계산학과

<요 약>

본 논문에서는 서버의 부하를 줄이고 네트웍의 트래픽을 줄일 수 있도록 지역 네트웍에 분산된 다중 에이전트 기반의 메타 검색 엔진:"하나로"를 구현했다. "하나로"는 작업 에이전트, 로봇 에이전트, 결과 필터링 에이전트로 구성되어 있으며, 지역 네트웍의 서버들에 분산 설치된다. 그리고, 각 검색 엔진에서는 검색 결과를 데이터베이스에 저장하고, 웹 로봇을 이용하여 웹 페이지에 대한 자료를 수집한다. 수집된 자료는 Oracle ConText Option으로 색인된 후에 검색 서비스로 제공된다.

A Study on the information search engine by using a meta search engine and Oracle ConText Option

Young-Keol Yang · Cheol-Young Ock
Dept. of Computer Science

<Abstract>

This paper proposes a meta search engine, "HANARO" which is based on the multiple agents distributed in LAN, in order to reduce the load of the server and the network traffic. "HANARO" consists of work agent, robot agent, and result filtering agent and is installed at each server in LAN and assigned a search area. The "HANARO" in each server is to search the information of assigned area and to store the result in database. The stored result is used to gather the specific information about web pages using a web

* 본 연구는 1997년도 울산대학교 학술연구비 및 대학원활성화를 위한 특별연구비에 의하여 수행된 결과임.

robot. The gathered information is indexed by Oracle ConText Option and provided for the user's query.

1. 서 론

인터넷의 등장으로 정보화 시대가 도래하고 있다. 중요한 정보 제공자의 역할을 하는 인터넷을 효과적으로 이용하기 위해서는 자신이 원하는 정보를 어떻게 빠르고 정확하게 찾아낼 수 있는가 하는 것이 중요하다. 이 문제를 효과적으로 해결하기 위해 생겨난 것이 웹 검색 엔진이다. WWW에서 웹 검색 엔진의 도움없이 신속하고 정확한 정보를 찾기란 사실상 불가능하다.

웹 검색 엔진은 크게 세 부분으로 구성되어 있다. 첫 번째로 자료 수집의 역할을 하는 웹 로봇이 있다. 웹 로봇은 인터넷에 연결되어 있는 각종 서버에 접근하여 데이터를 가져온다. 두 번째로 웹 로봇이 수집한 자료를 가공하는 색인 시스템이 있다. 색인 시스템 내에는 언어에 따라 처리하는 형태소 분석기가 포함되어 있다. 세 번째로 사용자 인터페이스가 존재한다. 사용자 인터페이스를 통해서 질의를 분석하고, 분석한 질의를 색인 시스템에 전달하며 검색된 결과를 사용자에게 보여주는 검색 시스템으로 구성되어 있다. 이러한 웹 검색 엔진을 구성하기 위해서는 강력한 연산을 지원하는 시스템과 방대한 데이터를 보관할 수 있는 부가 저장 공간의 확보가 절대적으로 필요하다.[1, 2, 4, 5, 6, 7, 9, 10]

정보 검색 시스템을 설계하고, 구현하는 데 있어서 한글을 정확하게 분석할 수 있는 기술 뿐만 아니라 필요한 자원을 갖춘 시스템이 필요하다. 이러한 이유로, 기 개발된 한글 검색 엔진과 영어 검색 엔진을 이용해서 사용자에게 종합적인 결과를 제공하는 실용적인 검색 엔진이 개발되고 있으며, 이런 종류의 검색 엔진을 메타 검색 엔진이라 부른다. 메타 검색 엔진은 검색 엔진이 가지는 부가 저장 공간이나 주기적으로 동작하면서 웹 공간에서 자료를 수집해야 하는 웹 로봇, 그리고 복잡한 색인 시스템도 필요치 않다.[8]

본 논문에서 제시하는 메타 검색 엔진은 지역 네트웍을 기반으로 하며, 네트웍에 접속되어 있는 각종 서버에 메타 검색 엔진을 설치한다. 메타 검색 엔진이 설치된 서버마다 특정 검색 분야를 할당하며, 사용자가 질의 입력 서버에 접속해서 해당 분야를 선택한 후, 질의를 입력한다. 그러면, 질의 입력 서버는 사용자의 질의에 맞는 해당 분야 서버로 질의를 보내고 메타 검색 엔진이 동작하도록 설계한다. 이렇게 여러 서버로 작업을 분산시키고 여러 서버에 검색 엔진을 설치하면 한 서버에 대한 부하를 줄일 수 있다. 그리고, 각 메타 검색 엔진이 위치하는 서버마다 각 분야로 분류되어 있는 서버에 사용자의 질의와 검색 결과를 지속적으로 저장할 수 있는 데이터베이스를 둔다. 사용자가 해당 분야에 같은 질의를 했을 경우, 각 서버에서는 자체 데이터베이스를 검색한 후 결과를 사용자에게 보여준다. 물론, 질의가 데이터베이스에 존재하지 않을 경우, 메타 검색 엔진을 이용함으로써 네트웍 부하를 줄일 수 있다. 마지막으로 각 서버에 저장된 데이터베이스를 바탕으로 정보 검색 서비스를 제공하도록 만든다. 현재, 각 RDBMS회사들은 자사의 데이터베이스에 텍스트 처리기술을 합해서 정보 검색이 가능하도록 제품을 개발해서 판매하고 있다. 본 논문에서는 Oracle사에서 개발한 색인, 검색 시스템인 ConText Option을 이용해서 자체 정보 검색 엔진으로 이용한다. 사용자가 질의 입력을 위해 서버에 접속해서 해당 분야에 대한 질의를 입력하면, ConText Option으로 구성한 정보 검색 엔진에 질의를 보낸 후에 결과를 살펴본다. 만약, 자료가 존재하지 않으면, 그 서버에 설치된 메타 검색 엔진을

이용해서 외부 정보 검색 엔진을 이용하게 한다. 이 과정을 지속적으로 반복하면, 특정 분야에 맞는 검색 엔진이 구성된다.[1, 3, 11]

본 논문에서는 기존의 메타 검색 엔진이 가지는 단점인 서버 집중적인 문제를 해결하고, 네트웍 부하를 줄일 수 있는 메타 검색 엔진을 설계한다. 그리고, 해당 분야에 맞는 자료를 보유한 검색 엔진으로 서비스 될 수 있는 시스템을 구성한다.

2. 메타 검색 엔진과 Oracle ConText Option

2.1 메타 검색 엔진 구조 및 분류

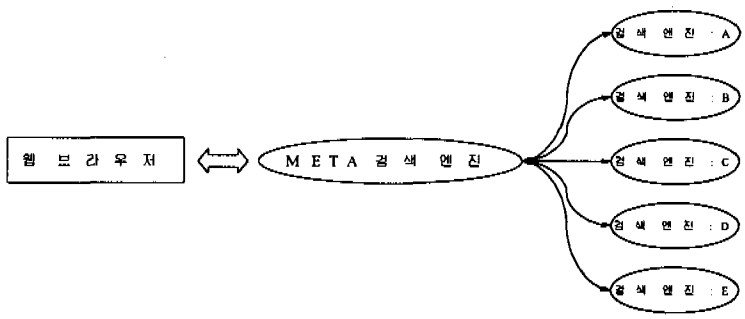

그림 1 메타 검색 엔진의 구조

<그림 1>은 메타 검색 엔진의 구조를 보여주고 있다. 메타 검색 엔진은 사용자의 질의를 분석하고, 이들 질의를 병렬적으로 여러 검색 엔진에 보내고, 반환된 결과를 조합한 후에 사용자에게 필요한 정보만을 제공한다.

메타 검색 엔진을 분류해 보면, 서버를 통해서 결과를 제공받는 서버 기반의 메타 검색 엔진이 있고, 클라이언트에서 자료를 조합해서 실 사용자에게 제공하는 클라이언트 기반의 메타 검색 엔진 두 가지로 구분이 된다.

2.1.1 서버 기반의 메타 검색 엔진

Savvy Search는 병렬적으로 여러 검색 엔진에 질의를 보내고, 다중 사용자에게 서비스를 제공하는 통합된 인터페이스를 제공한다. 이 시스템은 세 가지 형태의 에이전트-발신 에이전트(dispatch agent), 인터페이스 에이전트(interface agent), 표시 에이전트(display agent) - 들로 구성되어 있다. [12]

MetaCrawler는 1995년 6월에 서비스를 시작한 멀티쓰레드형 검색 에이전트이다. Savvy Search와 유사한 방식으로, 동시에 여러 검색 엔진에 질의를 전송하고, 결과를 조합해서 사용자에게 보여주는 구조로 되어 있다. 특징은 사용자와 문서의 지역적인 위치, 형태 등을 고려해서

문서를 정렬하며, 문서의 검증을 자동으로 한다. Savvy Search와 다른 점은 항상 9개의 검색 엔진에 질의를 보내는 것이며, 이는 반응시간이 길어지는 문제가 있지만, 사용자에게 종합적으로 다양한 정보를 제공할 수 있다는 장점이 되기도 한다.[13]

2.1.2 클라이언트 기반의 메타 검색 엔진

Fish Search는 질의어에 적합한 문서를 자동적으로 검색하는 클라이언트 기반의 검색 에이전트이다. Mosaic의 웹 주소 검색 에이전트를 수정한 버전으로 사용자가 검색하는 것과 동일하게 수행한다.

Ms. Dachanni는 국내에서 서비스중인 대표적인 메타 검색 엔진으로 다중 에이전트를 기반으로 하고 있다.[8]

2.2 Oracle ConText Option

Oracle ConText Option은 Oracle7과 SQL을 결합하여 동작하는 검색 엔진 프로그램으로 다음과 같은 네가지 장점을 제공한다.

■ 데이터베이스 수준의 아키텍처

문서관리를 할 때, RDBMS에서 얻을 수 있는 확장성, 보안성, 통합성, 내고장성, 관리용이성을 이용할 수 있다.

■ 표준에 근거한 개발 환경

ConText Option은 문서 데이터를 관계형 데이터처럼 취급하므로 표준 SQL을 이용하여 검색하는 것이 가능하다. 다양한 툴들을 이용하여 데이터 탐색과 문서 탐색, 문서 변형을 할 수 있다. 따라서, 신속한 개발 및 텍스트 기능을 쉽게 업그레이드할 수 있다.

■ 진보된 문서 검색 기술

ConText Option은 문서 검색, 변형 및 분류 기능을 지원하며, 대용량 데이터베이스에서 원하는 정보를 빠르게 찾을 수 있다. ConText Option이 지원하는 주요 기능은 다음과 같다.

- 자동 문서 변형
- 자동 분류
- Soundex/Fuzzy Matches
- 근접 탐색
- 부울 로직(Boolean Logic)
- Relevance Ranking
- Term Weighting
- 동의어 사전 지원
- Stop Lists

■ 확장가능한 구조

ConText Option의 확장 가능한 구조에서는 새로운 언어를 지원하기에 편리하고 특수한 검색 엔진 및 서비스를 개발할 수 있게 한다.

사용법은 Oracle ConText Option Administrator Guide 또는 Oracle ConText Option Developer's Guide를 참조하기 바란다.[11]

3. 검색 엔진의 설계

3.1 검색 엔진의 설계

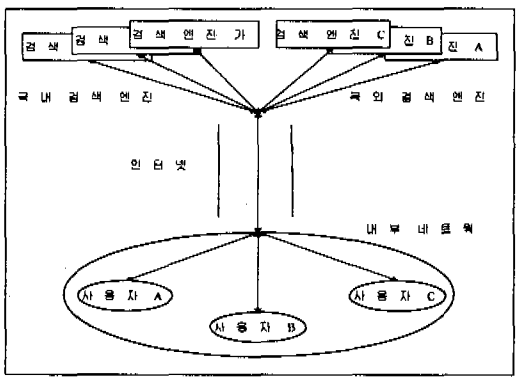

그림 2 외부 검색 엔진의 이용

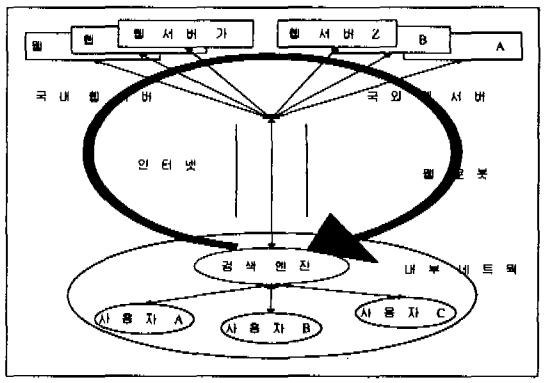

그림 3 지역 네트웍에 설치된 웹 검색 엔진

국내외에 여러 정보 검색 엔진이 존재한다. <그림 2>와 같이 지역 사용자들이 원격지에 존재하는 검색 엔진에 접속할 경우, 네트웍 트래픽이 발생하며 시간도 많이 소모된다. 네트웍 부하를 해결하는 방법은 <그림 3>과 같이 지역 네트웍에 웹 검색 엔진을 만드는 것이다. 하지만, 웹 로봇과 색인 시스템, 한글 처리 시스템을 개발하는 것이 쉽지 않다

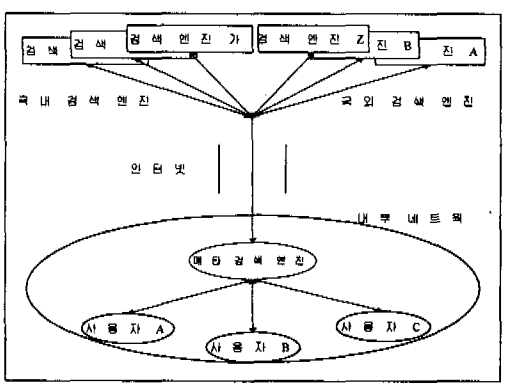

그림 4 LAN에 설치된 메타 검색 엔진

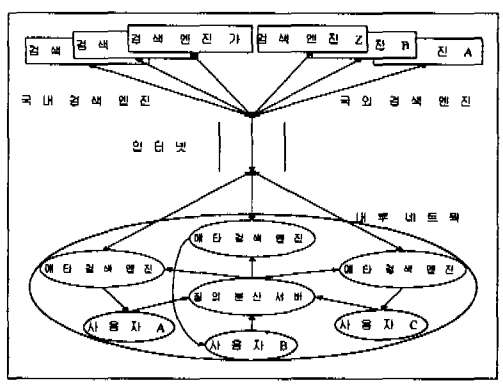

그림 5 LAN에 분산된 메타 검색 엔진

<그림 4>와 같이 기존의 정보 검색 시스템을 이용하는 메타 검색 엔진을 개발해서 지역 네트웍에서 사용하는 것을 생각할 수 있다. 그러나, 메타 검색 엔진은 병렬로 각 검색 엔진에 요청을 보내므로, 더 많은 네트웍 트래픽을 발생시킨다. 또한 한 서버에 여러 사용자들이 접속하고, 여러 검색 엔진에 자료를 요청을 하기 때문에 서버에 대한 부하도 상당히 발생한다.

서버에 대한 부하를 줄이는 방법으로 LAN에 분산된 메타 검색 엔진을 설계한다. <그림 5>에서 보는 바와 같이 사용자가 접속하는 질의를 입력 서버와 메타 검색 엔진을 분리시킨다. 그리고, 메타 검색 엔진을 LAN에 설치된 여러 시스템에 분산시킨다. 이렇게 구성하면, 한 서버에 대한 부하가 감소하게 된다. 서버에는 각각의 분야를 할당하고, 사용자는 자신이 검색하고자 하는 분야를 선택하면, 분야에 맞는 서버에 연결함으로써 분야별로 메타 검색 엔진이 실행된다.

다음으로 네트웍 트래픽을 감소시킬 수 있는 방법을 제시한다. 메타 검색 엔진이 동작하는 내부에 사용자의 질의와 검색 결과를 지속적으로 저장할 수 있는 데이터베이스를 둔다. 메타 검색 엔진의 특징 중 한가지가 다른 정보 검색 엔진을 이용하므로 대규모의 부가 저장 장치가 필요치 않다는 점이다. 하지만, 메타 검색 엔진들은 네트웍 부하를 줄이고 반응 속도를 높이기 위해 자체 캐쉬를 포함하고 있다.[8] 이 캐쉬를 본 논문에서는 사용자의 질의와 검색 결과를 저장하고 관리할 수 있는 데이터베이스로 확장한다. <그림 6>과 같이 사용자가 같은 질의를 하거나 기존의 질의 정보를 조회할 수 있게 함으로써 네트웍 트래픽을 줄일 수 있게 한다.

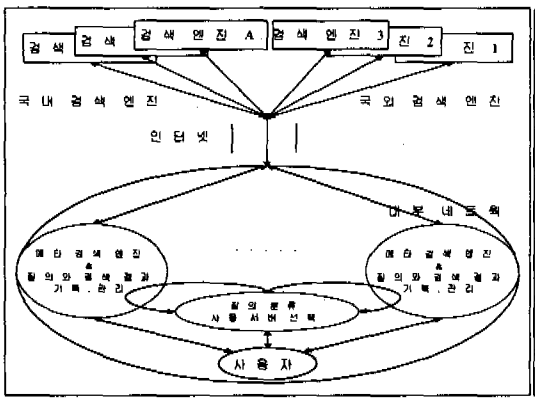

그림 6 분야별로 동작하는 메타 검색 엔진

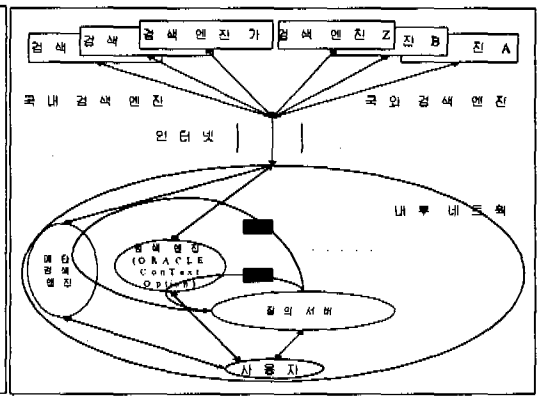

그림 7 ConText Option을 이용한 검색 엔진

각 서버마다 해당 분야를 설정함으로써, 해당 분야에 대한 정보가 데이터베이스에 축적이 될 것이다. 이 자료는 한 분야에 대한 자료이므로 기존의 검색 엔진과 차별화시킬 수 있다. 기존 검색 엔진은 모든 분야에 대한 정보를 포함하므로 사용자가 조회를 하지 않는 정보도 가지고 있다. 그러나, 위의 데이터베이스는 한 분야에 대해서만 자료를 가지고 있으므로 해당 분야에 관련이 있는 사용자에게 더 큰 가치가 있다. 이 자료를 웹 로봇과 Oracle RDBMS와 Oracle ConText Option을 이용해서 <그림 7>과 같은 정보 검색 시스템을 구현해서 서비스할 수 있다.

본 논문에서는 "정보 검색" 분야에 한정해서 검색하도록 메타 검색 엔진의 분야를 설정했으며, 500여 개의 검색 결과를 Oracle ConText Option으로 색인한 후에, 검색 서비스를 제공하도록 구성하였다.

3.2 검색 엔진 구조

3.2.1 다중 에이전트 구조

메타 검색 엔진은 기존의 웹 검색 엔진의 결과를 바탕으로 사용자에게 정보를 제공하므로, 웹 검색 엔진에서 검색 결과의 형식을 수정할 경우 이를 이용할 수 없게 된다. 이 문제를 효과적으로 해결하기 위해서 필터링 에이전트 부분을 작업 에이전트에서 분리함으로써 손쉽게 수정할 수 있게 한다. 또한, 메타 검색 엔진의 반응 시간을 향상시키기 위해서, 검색 엔진별로 로봇 에이전트를 동작시킨다. 웹 검색 엔진에 대한 반응 시간이 길어질 경우, 로봇 에이전트 상태를 제어함으로써 효과적으로 처리할 수 있다.

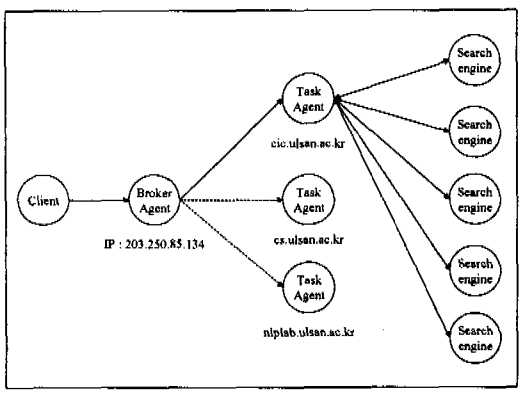

그림 8 LAN에 분산된 메타 검색 엔진

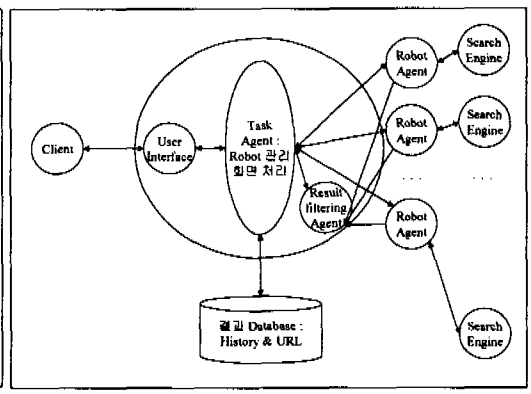

그림 9 에이전트들간의 관계

이처럼 다중 에이전트의 구조를 채택함으로써 검색 엔진의 유지보수를 쉽게 하며, 검색 엔진의 반응 시간을 줄인다.

3.2.2 분산 에이전트 구조

분산 에이전트 구조는 웹 상에 여러 복제 에이전트를 산재시켜서 사용자에 의해 전송되는 검색 요청을 여러 에이전트중의 하나를 선택해서 수행하도록 구성된다. 이 구조는 하나의 집중화된 검색 에이전트 시스템보다 네트웍의 부하와 서버의 작업량을 줄일 수 있는 이점이 있다. 이 구조는 사용자와 에이전트를 연결시키는 역할을 담당하는 브로커 에이전트와 사용자의 질의를 직접 검색하는 작업 에이전트로 구성된다.

분산 에이전트 구조 시스템의 이점은 다음과 같다.

- 한 서버에 에이전트들이 집중되지 않기에 작업 에이전트로부터의 반응 시간이 짧아진다.
- 분산된 에이전트로 인해서 시스템의 효율성과 안정성이 증가한다.

3.3 에이전트 설계

3.3.1 브로커 에이전트

브로커 에이전트는 사용자의 접속을 관리하고 작업 에이전트에 사용자의 질의를 배당하는 역할을 담당한다.

3.3.2 작업 에이전트

사용자 인터페이스에서 전달된 질의어를 분석한 후에, 데이터베이스를 검색하고, 로봇 에이전트를 생성시켜서 각 검색 엔진으로 정보를 요구하도록 한다. 검색 결과를 결과 필터링 에이전트가 조합한 후에, 작업 에이전트가 HTML형식으로 변환해서 사용자에게 전송한다.

3.3.3 로봇 에이전트

로봇 에이전트는 사용자의 질의를 가지고 각 검색 엔진에 접속해서 결과를 가지고 오는 역할을 수행한다.

3.3.4 필터링 에이전트

로봇 에이전트에 의해서 생성된 파일을 분석해서, 사용자에게 정보를 제공할 수 있는 형태로 재구성하는 역할을 담당한다.

3.4 웹 로봇 프로그램

ConText Option과의 연동을 위해서 데이터베이스에 저장된 URL을 가지고 각 URL에 대한 정보를 파악하는 역할을 한다. 웹 로봇은 주기적으로 해당 URL을 점검하면서 변경여부를 체크해야 한다.

3.5 순위 알고리즘

각 검색 엔진에서 전송된 결과를 사용자에게 보여주기 전에 순위를 부여한 후에 정렬해서 보여준다. 각 검색 엔진의 결과에는 점수가 부여되어 있는데, 이 점수를 상대점수로 계산하고 순위에 대해서도 점수를 부여한다. 각 검색 결과는 검색 결과에 대한 상대 점수와 출현 순서에 대한 점수를 합쳐서 계산하고, 중복되는 결과는 각 점수를 더한다.

4. 검색 엔진에 대한 평가

4.1 메타 검색 엔진의 구현

메타 검색 엔진은 Windows-NT 4.0과 Solaris 2.5.1 운영체제 두 가지 버전을 개발하였다. 그러나, Windows-NT에서 동작하는 메타 검색 엔진의 성능이 Solaris 성능의 1/4 수준으로 평가되었다.

본 논문에서는 한정된 영역 중에서 "정보 검색"을 주제로 자료를 수집하는 검색 엔진을 제작하였다. <그림 11>과 같은 구조로 구현하였으며, Solaris 2.5.1 시스템에 설치된 메타 검색 엔진에서 "정보 검색" 영역을 담당한다.

4.2 ConText Option의 색인

로봇 에이전트에 의해서 작성된 SQL파일은 sqlplus를 이용해서 색인을 수행한다. 일련의 SQL 파일 스크립트를 거쳐서 작업이 진행된다. <그림 12>는 질의어 '검색 | 엔진'을 입력해서 찾은 결과를 보여주는 화면이다. Oracle ConText Option은 Oracle 데이터베이스가 제공하는 다양한 검색 연산자를 지원하며, 검색 수행 시간도 외부 검색 엔진을 이용할 때보다 상당히 줄어든다.

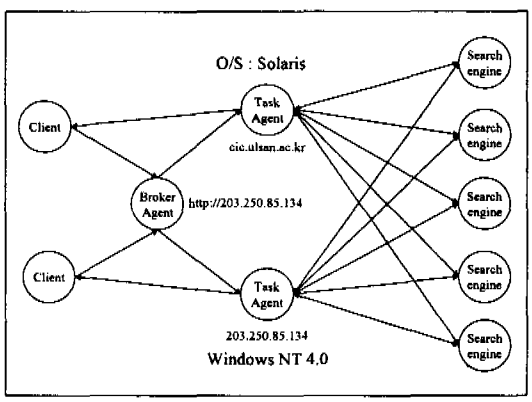

그림 10 LAN에 구현된 메타 검색 엔진

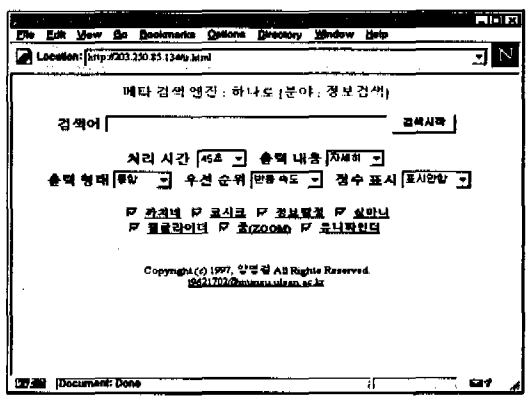

그림 11 특정 분야에 대한 메타 검색 엔진

따라서, 외부 검색 엔진에 접속하는 비율을 줄일 수 있으며, 네트워 트래픽도 감소시킬 수 있다. 현재 500여 개의 "정보검색"에 관련한 자료를 수집해 놓았으며, 사용자의 질의를 처리하는 시간은 약 2초 전후가 소요된다.

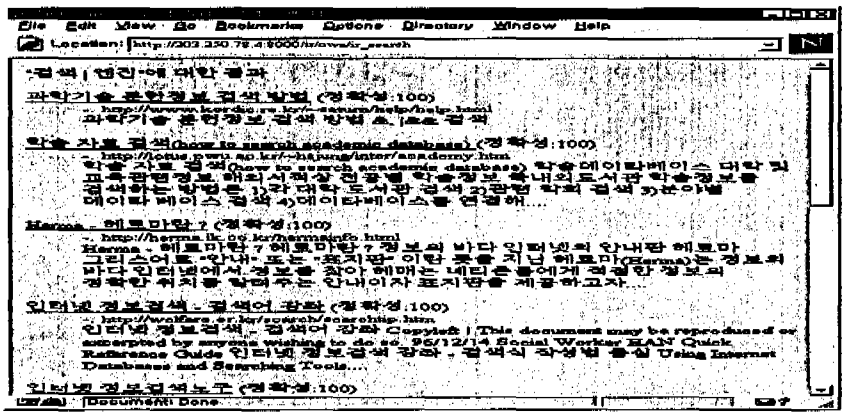

그림 12 ConText Option을 이용한 검색 모습

4.3 성능 평가

4.3.1 메타 검색 엔진 평가

메타 검색 엔진:"하나로"의 성능 평가에 사용된 시스템은 다음과 같다

사용 시스템 1
- -. 한글 Windows NT 4.0 Workstation Service Pack 3
- -. CPU : Pentium 120MHz
- -. RAM : 48M
- -. H.D.D. Virtual Memory : 100M

사용 시스템 2
- -. SUN ULTRA Enterprise 3000
- -. OS : Solaris 2.5.1
- -. CPU : 167MHZ UltraSPARC 2EA
- -. Memory : 128MB

<그림 13>와 <그림 14>를 보면, Windows NT의 PC와 Solaris 2.5.1의 SUN 시스템의 성능은 비교하기 힘들 정도로 차이가 심하게 나타나는 것을 알 수 있다. 두 시스템의 자원을 비교해도 차이가 많이 발생함을 알 수 있다. 두 시스템의 반응 시간을 따져 보면, 한글 검색 엔진은 Windows NT의 경우 40초 이후에 반응을 하지만, Solaris 2.5.1에서는 10초 이내에 반응을 하고 있다. 영문 검색 엔진의 반응 시간도 한글 검색 엔진과 마찬가지로 Solaris 2.5.1시스템에서 빠르게 반응한다. 이 정보를 바탕으로 반응 시간과 우선 순위를 옵션으로 설정해서 더욱 빠른 정보를 사용자에게 제공하고 있다.

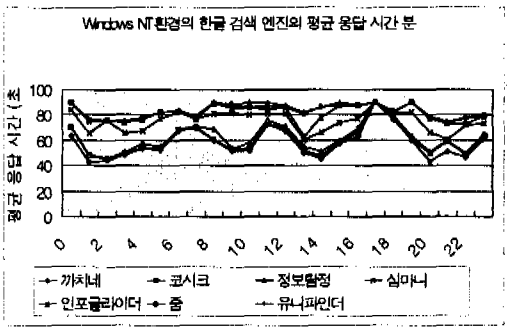

그림 13 Windows NT에서 한글 검색 엔진 평균응답 시간 분석

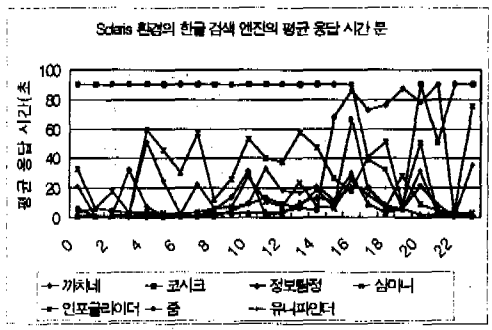

그림 14 Solaris에서 한글 검색 엔진의 평균 반응 시간분석

분산 메타 검색 엔진을 구축할 경우에 서버의 부하를 줄일 수 있는데, <그림 15><그림 16>를 통해서 알 수 있다. NT 운영체제에 설치된 메타 검색 엔진을 대상으로 평가를 했다. <그림 15>는 단일 사용자가 질의를 했을 경우, 각 검색 엔진의 반응 시간을 보인 것으로 90초 이내에 반응을 하고 있다. 그러나, <그림 16>의 경우, 다중 사용자(2명 이상)가 질의를 했을 경우, 대체적으로 80초 이후에 반응을 하고 있으며, 최대 190초에 이르기도 한다. 이처럼 각 검색 엔진에 대한 반응 시간을 고려해 보면, 다중 사용자가 동시에 메타 검색 엔진을 사용할 경우 서버와 네트웍에 대한 부하로 인해서 반응 시간이 길어진다는 것을 알 수 있다.

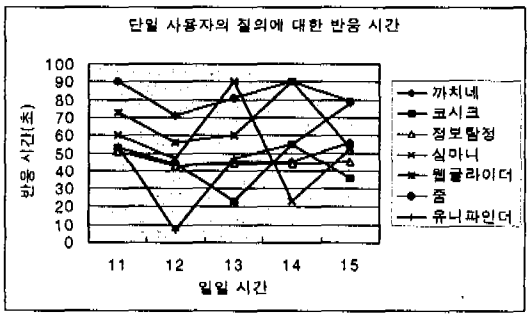

그림 15 단일 사용자 질의에 대한 반응 시간 : NT환경

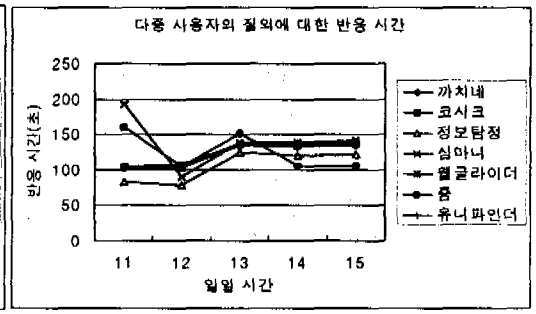

그림 16 다중 사용자 질의에 대한 반응 시간 : NT환경

4.3.2 ConText Option을 이용한 검색 엔진

ConText Option을 사용해서 구축해 놓은 검색 엔진은 Solaris 2.5.1 환경하에 구축해 놓았으며, 검색 소요 시간은 지역 네트웍과 서버의 부하에 영향을 받는다. 시스템 부하가 높은 시간일 경우 약 5초 내외에 검색을 완료하며, 사용자가 줄어든 시간은 약 2초 내외에 반응을 완료한다. <그림 17>을 보면, 지역 네트웍을 사용하는 사용자의 경우는 외부 검색 엔진을 이용하는 것보다 훨씬 빠른 결과를 얻을 수 있다.

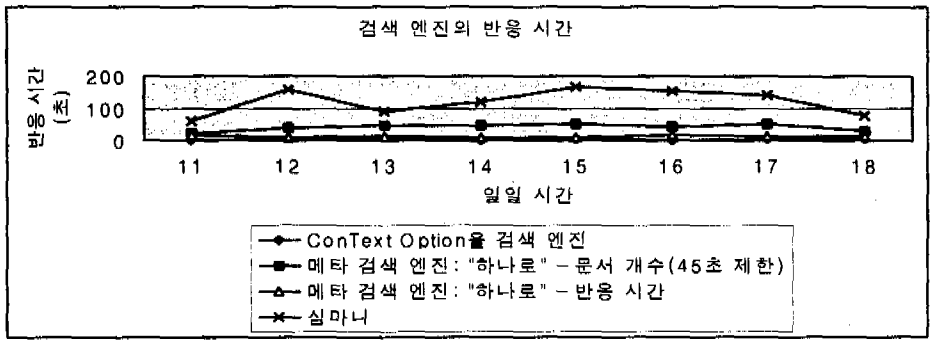

그림 17 검색 엔진에 따른 반응 시간 분석

5. 결론 및 향후 연구

본 논문에서는 다중 에이전트에 기반하고 분산 환경에서 동작하는 메타 검색 엔진 : "하나로"를 설계하고 검색 엔진의 결과와 Oracle ConText Option을 이용해서 제한된 분야에 대해 정보를 제공하는 검색 엔진을 구현하였다.

LAN에 존재하는 여러 서버들에 각 분야를 담당하도록 작업 에이전트를 분산시키고, 해당 분야에 맞는 서버가 사용자의 질의를 처리하며, 검색 결과를 데이터베이스에 저장하는 구조를 갖도록 구성해서 한 서버에 대한 작업량을 줄일 수 있다.

또한, 검색 엔진 결과에 대해 웹 로봇을 이용해서 데이터를 수집한 뒤, ConText Option을 통해 색인 작업을 거쳐서 검색 서비스를 제공하도록 구성했다. 그 결과 사용자에게 빠른 정보를 제공하고, 외부 검색 엔진에 접속하기 위한 시간과 네트웍 트래픽을 줄일 수 있다.

그러나, 본 시스템이 갖는 문제점도 있다. 분야에 맞지 않는 질의를 했을 경우, 이를 처리할 수 있는 해결책이 존재하지 않는다. 그리고, 지역 네트웍에 존재하는 서버에 Oracle과 ConText Option을 설치하고, 이를 관리, 교육하는데 어려움 등이 있다.

앞으로 본 논문에서 제시한 다중 에이전트를 쓰레드(Thread)형태로 다시 개발한 후에, 메타 검색 엔진을 실제적으로 교내 전산망에 적용시켜서 분야별로 웹 검색을 가능하게 하고, Oracle ConText Option을 이용해서 학과 또는 연구실에 검색 엔진을 구축하는 것이다.

참 고 문 헌

[1] 양영걸, 안미정, 옥철영, "META 검색 엔진을 이용한 웹 로봇에 관한 연구", 한국정보과학회 1997년 가을 학술발표논문집 제24권 2호, pp. 461~464

[2] 김태헌, "웹 상의 정보를 수집하기 위한 로봇 에이전트의 개발", 1996년 울산대학교 전자계산학과 석사 논문

[3] 최중민, "에이전트의 개요와 연구방향", 1997. 3 정보과학회지 제15권 3호, pp. 7~16

[4] 권혜진 외 8명, "국내 웹 정보 검색 기술의 동향", 1997. 10 정보과학회지 제15권 10호,

pp. 16~23

[5] 강병주, 이지동, 최기선, "데이터베이스 관리 시스템에 기반한 웹검색엔진의 구현", 1997년도 제9회 한글 및 한국어 정보처리 학술대회, pp. 211~218

[6] 박민우, 박철제, "대화형 인덱싱을 위한 로봇 에이전트의 설계 및 구현", 1997년도 제9회 한글 및 한국어 정보처리 학술대회, pp. 219~226

[7] 대구대학교 정보통신 공학부 컴퓨터 응용연구실, "까치네 로봇의 원리와 서비스 형태", http://www.kachi.com/~hunkim/w4

[8] Hyunsuk Seung, "A Client-Oriented Distribution Architecture for Web Search Agent", 1996. 6. Department of Information and Communication Engineering Korea Advanced Institute of Science and Technology

[9] Joseph Williams, "Bots and other Internet Beasities", Sames.net Publishing, 1996.

[10] William B. Frakes, Ricardo Baeza-Yates, "Information Retrieval data Structures & Algorithms", Prentice Hall, 1992

[11] Oracle7 ConText Option Developer's Guide, Release 1.1

[12] Savvy Search, http://www.cs.colostate.edu/~dreiling/smartform.html

[13] MetaCrawler, http://www.metacrawler.com