글로버스를 이용한 분산 웹 크롤러의 설계 및 구현

이지선° 김양우° 이필우°°
동국대학교 정보통신공학과°, 한국과학기술정보연구원°°
sun79°@dongguk.edu, ywkim@dongguk.edu, pwlee@kisti.re.kr

Design and Implementation of Distributed Web Crawler Using Globus Environment

Jisun Lee°, Yangwoo Kim°, Philwoo Lee°°
Dept. of Information & Telecommunication Engineering, Dongguk Univ.°, KISTI°°

요 약

대부분의 웹 검색 엔진들과 많은 특화된 검색 도구들은 웹 페이지의 색인화와 분석을 위한 전처리 단계로 대규모 웹 페이지들을 수집하기 위해 웹 크롤러에 의존한다. 일반적인 웹 크롤러는 몇 주 또는 몇 달의 주기에 걸쳐 수백만 개의 호스트들과 상호작용을 통해 웹 페이지 정보를 수집한다.

본 논문에서는 이러한 크롤러의 성능향상과 효율적인 실행을 위해 그리드 미들웨어인 글로버스 툴킷을 이용하여 분산된 크롤러를 제안한다. 본 웹 크롤러의 실행은 그 기능의 분산처리를 위한 각 호스트 서버들을 글로버스로 연결하고, 인증하여, 작업을 할당하는 단계와, 크롤러 프로그램이 실행되어 자료를 수집하는 단계, 마지막으로 이렇게 수집된 웹 페이지 정보들을 처음 명령한 시스템으로 반환하는 단계로 나누어진다. 결과 수집 작업을 보다 분산화 할 수 있게 하였으며, 여러 대의 저 비용의 시스템에서 고 비용, 고 사양의 서버의 성능을 얻을 수 있었으며, 확장이 용이하고, 견고한 크롤러 프로그램 및 시스템 환경을 구축할 수 있었다.

1. 서 론

현재 인터넷의 보급과 기술은 기하급수적으로 발전하여 소위 정보의 바다라는 말을 실감할 수 있게 되었다. 이와 더불어 정보의 바다인 인터넷 속에서 사용자가 원하는 정보를 보다 신속하고 정확하게 찾는 기술이 이슈화되었고, 이에 따라 정보를 수집하고 검색할 수 있는 다양한 검색 엔진들이 생겨났다. [1]

현재 사용되는 대부분의 검색 엔진의 경우, 하나의 거대한 서버가 전 세계에 존재하는 수많은 정보원으로부터 정보를 받아, 일일이 인덱스를 만들고 이를 검색하는 방법을 사용한다. 이 방법은 하나의 서버에 지나친 오버헤드를 가중시키며, 정보량의 증가에 따른 검색 시간의 증가, 네트워크의 부하 등의 다양한 문제들을 발생시킨다. 이러한 문제들은 검색서버를 여러 대로 분산시키는 방법을 통해 어느 정도 해결될 수 있으나, 절차적으로 진행되는 수집과 색인화 작업은 선 작업인 수집의 처리량의 증가에 따라 후작업의 지연을 발생시키게 되었다. [3],[7] 이러한 지연을 미연에 방지하기 위한 한 방법으로 본 논문에서는 검색 엔진에서 문서 수집을 하는 웹 크롤러를 글로버스 상에서 분산되어 수행되도록 구현하여 보았다.

글로버스는 IBM에서 제공하는 그리드 분산 컴퓨팅을 위한 미들웨어로, 분산되어 있는 시스템들을 연결시켜 상단의 응용 프로그램이 하나의 시스템에서 수행되는 것과 같은 서비스를 제공해주는 인터페이스 역할을 한다.

2. 본 론

2.1 관련 연구

2.1.1 웹 크롤러

정보의 바다 인터넷에서 사람을 대신해서 문서를 수집하는 것이 크롤러이며, 문서 수집 로봇, 스파이더, 문서 수집 에이전트 등으로 불리기도 한다.

크롤러는 검색엔진에서 자료의 수집을 담당한다. HTML의 참조 링크를 따라 웹 페이지의 정보를 가져온다. 가져오는 형태는 HTML문서가 될 수도 있고, 텍스트가 될 수도 있다.

정보검색의 결과는 검색 대상의 규모에 따라 좌우되므로 대부분의 검색 엔진들은 대규모의 컬렉션을 얻기 위해 크롤러에 의존하고 있다.[2]

2.1.2 글로버스 툴킷

글로버스 툴킷은 그리드 분산 컴퓨팅을 위한 대표적인 미들웨어로 꼽히는 인터페이스로 시스템들을 통합시키기 위해 필요로 하는 다양한 서비스들을 각각의 독립적인 요소로써 제공하고 있다. 그러면서 기존의 각 시스템 및 네트워크의 관리 정책이나 운영 도구들을 무시하지 않고 각 요소들과 협력해 시스템 통합 환경을 구축할 수 있게 한다.[6]

제공하는 핵심 서비스는 크게 4가지로 나눌 수 있다. 첫째로 자원정보 서비스는 그리드 내에 존재하는 자원들의 상태를 공유하고 사용자들에게 제공하는 요소인 디렉토리 서비스이다.

둘째로 자원관리 서비스는 원격지의 분산 자원들을 동시에 사용할 수 있게 하고, 자원들 간에 존재하는 관리상의 상이함을 처리하여 준다.

셋째로 데이터관리 서비스가 있는데, 이것은 원격지에

있는 파일을 사용해 작업을 처리하길 원하거나 원격지에서 처리한 작업을 또 다른 저장 장치에 저장하는 일을 처리한다.

마지막으로 보안 서비스는 분산된 자원을 공유함에 따라 자연히 발생하는 문제들에 대해, 사용자의 입장에서는 안전성과 편리성을 만족시키고, 관리자의 입장에서는 안전한 보안을 제공하여 준다. [4]

본 논문에서 제안하는 분산 크롤러 시스템은 글로버스 버전 3.0에서 제공하는 위의 기본적인 서비스들을 기반으로 하여 구축되며 동작한다.

2.2 제안 크롤러

2.2.1 기존 크롤러의 문제점

크롤러는 몇 주 동안, 혹은 몇 달 동안 계속 수행되면서, 인터넷으로부터 문서 혹은 정보들을 계속적으로 수집하는 프로그램이다. 한 서버 내에서 문서 수집 작업과 색인작업을 동시에 수행하려면 서버에 가중되는 부하는 점점 증가하게 된다. [5]

자료 수집을 위한 크롤러를 다른 서버로 분리해 내는 것은 그리 복잡한 작업이 아니다. 그러나 크롤링 작업은 관점에 따라 색인 작업보다 더 많은 정보를 다루어야 할 경우가 많아 단일 서버에서 처리하는 것보다 많은 호스트들을 연결, 분산하여 병렬로 작업을 수행하는 것이 보다 효율적이라는 사실을 알 수 있다. [3]

2.2.2 설계

본 논문에서 제안하는 크롤러가 실행되는 환경은 물리적으로는 LAN으로 연결되고 논리적으로는 글로버스로 연결된 4개의 호스트로 이루어진다. 이들 모두는 동일한 버전인 리눅스 Redhat9.0을 운영체제로 가지며, 그 위에 글로버스3.0을 설치하였다. 여기서 운영체제의 종류는 달리하여도 영향을 받으며, 이것은 글로버스의 특성이라고 할 수 있다. 호스트 1~3은 크롤링 작업을 의뢰받아 호스트 4를 대신하여 수행한 후, 크롤링 결과를 호스트 4로 넘겨준다. 호스트 4는 크롤러 매니저로써 크롤러를 시작하고 중지하며, 결과를 넘겨받아 관리한다. 이러한 과정은 다음의 코드로 표현될 수 있다.

```
request running permission (by obtaining a token)
load pageToFetch
extract hypertext links from page
for all found links do
if new link
  record link in database (so that we avoid it next time)
  recurse for new page (launch new thread running same algorithm)
 release token
die (thread terminates)
```

그림 1 크롤러의 pseudo code

크롤러의 기본 기능은 한 페이지를 가져와 URL을 추출하고, 다시 URL 링크를 따라 또 다른 페이지를 가져오는 것이다. 따라서 크롤러의 내부 구조는 페이지에서 URL을 추출하는 부분, 쓰레드의 개수를 제어하는 부분, HTTP 연결을 제어하는 부분, 가져온 페이지를 파일로 변환하는 부분, 마지막으로 메인 함수가 들어있는 부분을 포함하는 5개의 클래스로 구성된다.

실행은 3단계로 진행된다.

첫 번째는 작업을 할당하는 단계로 원격으로 연결된 호스트에게 작업을 할당한다. 연결된 호스트의 보안을 위해 프록시 인증 단계를 거치는데, 글로버스의 인증 단계를 사용한다. 두 번째로 자료 수집의 단계로 웹 문서의 URL을 추출하여 다음 문서로 링크를 따라 반복적으로 웹 문서를 수집한다. 여기서 웹 크롤러가 실행된다. 각 호스트에서 크롤러는 처음 주어지는 URL 목록 파일을 가지고 웹상의 문서들을 긁어오며, 한 페이지씩 읽어오면서, 앵커(anchor) 태그를 이용하여, 참조하는 URL을 추출한다. 추출한 URL은 목록 파일로 저장되고, 각 페이지는 텍스트 파일로 저장된다. 크롤러는 위의 과정들을 반복적이고 독립적으로 실행하게 된다. 마지막으로 세 번째 단계에서는 크롤러의 작업 결과를 작업을 할당했던 호스트에게 반환하는 단계이다.

2.2.3 구현

본 논문에서는 크롤러의 작업 할당을 위해 글로버스로 연동되는 여러 대의 호스트를 물리적으로 LAN에 의해 연결되도록 구성하였다.

제안 크롤러는 검색 엔진에서 일반적으로 사용하는 HTTP(Hyper Text Transfer Protocol)를 사용한다. 검색 엔진에서는 HTTP의 특성을 이용하여, 크롤러가 웹상에서 문서를 읽어올 수 있으며, 앵커 태그를 이용하여, URL을 추출하도록 하였다. HTTP의 기본적인 특성인 Request/Response 형식을 이용하기 위해 강력한 네트워크 프로그래밍용 API를 제공하는 자바로 구현하였다. 자바는 HTTP의 연결과, Socket 연결을 간단한 API로 구현하여 준다. 또한 자바는 다중 쓰레딩 방식을 지원하여, 크롤러의 효율을 극대화 할 수 있다.

제안된 크롤러는 우선 호스트들의 논리적으로 글로버스 상에서 연결이 성립된 후 시작된다. 글로버스를 통해 크롤링 작업을 하기 위해 컴퓨팅 파워의 대여를 약속한 호스트들끼리 연결을 맺게 되고, 약속을 맺은 호스트들은 크롤링 프로그램을 글로버스의 명령어를 통해 복사하여 가지고 있으며, 실행 명령은 프로그램으로 짜여져 한 호스트에서 글로버스 명령어의 실행으로 모든 호스트에 적용하게 하였다. 각각의 호스트는 실행 명령을 받고 크롤러를 시작하게 된다. 크롤러는 5개의 부분으로 나누어져 각 부분은 하나의 클래스로 생성되었다.

크롤링 작업이 시작되면, 작업을 던진 호스트에는 크롤링 작업으로 인해 추출된 URL들을 볼 수 있게 되며, 각각의 저장된 파일들은 로컬의 저장소에 남아있게 된다. 남겨진 파일들은 크롤링 후에 색인 작업을 하게 될 때 로컬에서 바로 색인작업에 이용된다.

이렇게 함으로써 작업을 던진 호스트로의 네트워크 사용량을 최소화 할 수 있고, 따라서 분산 처리의 단점을

보완한다.

2.3 결 과

각각의 호스트로 작업을 할당한 결과, <그림2>와 같이 각각의 크롤러가 수집해 오는 URL의 목록을 원격의 호스트에서 볼 수 있었다.

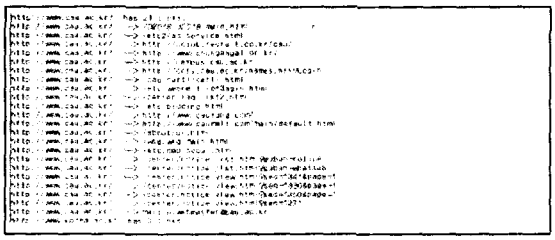

그림 2 호스트 1에서 가져온 URL 목록

또한 <그림 3>은 로컬에 저장된 웹 페이지들의 파일화된 것을 목록 화하여 나타내는 것으로 향후 색인 작업 시 로컬에서 가져다 사용할 수 있다.

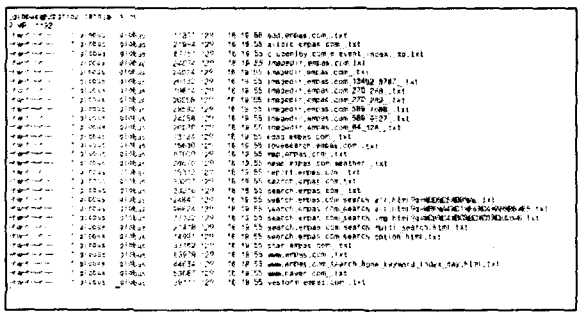

그림 3 호스트 2에 저장된 웹 페이지

3. 결 론

최근 인터넷의 엄청난 발전은 정보의 양으로 대변되었다. 정보의 양이 증가하면서, 인터넷에서 사용자가 원하는 정보를 신속하고 정확하게 찾는 기술의 수요가 점차 높아지게 되었다. 결과 검색엔진이 만들어졌고, 초기 검색 엔진은 하나의 거대한 서버로 전 세계에 존재하는 정보원에서 정보를 받아 하나하나 색인을 만들어 찾는 구조를 가졌다.

그래서 서버의 분산화가 전개되었고, 작업은 모듈화 되어 수집 단계와 색인 단계로 나누어지기도 하였다. 그러나 이 두 단계는 서로 연계적으로 설계되고 동작되기 때문에 발생하는 작업 의존도로 인하여 분산처리 임에도 불구하고 단일 서버와 비슷한 성능을 내기도 하였다.

이러한 문제의 해결을 위해 본 논문에서는 원격의 호스트들을 통합하여 주는 글로버스라는 미들웨어를 사용하여, 수집 작업을 보다 분산화 할 수 있게 하였다. 결과 여러 대의 저 비용의 시스템에서 고 비용, 고 사양의 서버의 성능을 얻을 수 있었으며, 확장이 용이하고, 견고한 크롤러 프로그램 및 시스템 환경을 구축할 수 있었다.

본 논문에서는 주로 검색 엔진에서 사용하는 웹 크롤러의 분산 수행에 초점을 맞추었지만, 검색 엔진의 다른 모듈, 즉 색인 작업이나 질의 처리 등으로 적용분야를 확장할 수 있을 것이다. 또한 로컬에 저장된 데이터들을 관리하는 메커니즘의 개발도 필요한 사항일 것이다.

또한 글로버스로 연동되어 실행되므로 향후 그리드 정보 검색 연구를 위한 크롤러로 응용할 수도 있을 것이다.

4. 참고 문헌

[1] 김형근, "정보탐정의 설계 및 구현", 웹 코리아, 제4회 WWW workshop, 1996.

[2] 박춘, "지능 이동 대행자를 이용한 인터넷 검색 엔진 시스템 설계와 분석", 연세대학교 대학원, 1998

[3] 전준식, "로봇과 RDBMS를 이용한 분산 환경 기반 지리정보 웹 검색 에이전트의 설계 및 구현", 한국교원대학교 대학원, 1998

[4] 이종숙, 홍정우, "마이크로소프트웨어: 그리드 미들웨어 해부", 서울:웅진 마이크로소프트웨어, 2002.

[5] Arvind Arasu, Junghoo Cho and Hector Garcia-Molina, "Searching the Web", ACM Transactions on Internet Technologies, 1(1), 2001.

[6] Globus Project Home Page, " http://www.globus.org"

[7] Vladislav Shkapenyuk and Torsten Suel, "design and Implementation of a High-Performance Distributed Web Crawler", Preceedings of the 18th International Conference on Data Engineering, 2002.