자바스크립트 함수 처리가 가능한 분산처리 방식의 웹 수집 로봇의 설계

김대유, 남기효*, 김정태
목원대학교, (주)위너다임*

Design of Web Searching Robot Engine Using Distributed Processing Method Application to Javascript Function Processing

Daeyu Kim, Ki-Hyo-Nanm*, Jung-Tae Kim
Mokwon University, Winnerdigm*
E-mail : jtkim3050@mokwon.ac.kr

요 약

기존의 웹 수집 로봇에서 처리 하지 못하는 자바스크립트 함수 링크를 처리하기 위하여 인터넷 익스플로러의 "Active Script Engine"을 사용 하였다. 또한 자바스크립트 함수 링크를 처리 하였을 경우 웹 수집 로봇의 수집량을 측정하기 위하여 웹 수집 로봇을 개발하였다. 웹 수집 로봇을 개발하기 위해서 구글봇과 네이봇 등 웹 수집 로봇의 구조를 파악하여, 수집 로봇에 활용되는 구성요소를 구현하고 분산처리형태의 웹 수집 로봇을 설계 하여 개발했다. 또한 개발된 웹 로봇에 제안된 자바스크립트 처리 모델을 추가하여 성능평가를 하였다. 성능평가 방법은 자바스크립트를 사용하는 웹 사이트의 게시판을 대상으로 하여 웹 수집량을 비교 분석하는 것 이다. 웹 사이트 게시물 1000개인 경우, 일반 웹 로봇의 경우에는 1페이지밖에 수집 하지 못하였고, 제안된 웹 로봇의 경우 1000개 이상의 웹 페이지를 수집하는 결과를 얻었다.

1. 서 론

인터넷 이용이 활발해짐에 따라 수많은 정보들이 웹 문서의 형태로 공개되고 있으며, 이러한 웹 문서들을 효과적으로 검색하기 위하여 웹 검색 서비스들이 이용되고 있다. 웹 로봇은 지정된 URL 리스트에서 시작하여 웹 문서를 수집하고, 수집된 웹 문서에 포함된 URL들을 추출과정과 새롭게 발견된 URL에 대한 웹 문서 수집과정을 반복하는 소프트웨어로서 웹 검색 서비스의 구축을 위해서는 웹 로봇을 이용한 웹 문서 수집이 선행되어야 한다. 1990년대 중반의 웹 문서 수는 현재에 비하여 매우 적었기 때문에, 최초로 개발된 웹 로봇 Wanderer를 포함하여 이 당시 개발된 다수의 웹 로봇들은 대용량의 웹문서들을 수집하도록 설계되지 않았다. 현재는 전 세계적으로 30억 개 이상의 웹 문서들이 존재하며, 국내에도 5천만 개 이상의 웹 문서들이 존재하고 있다. 따라서 이처럼 많은 수의 웹 문서들은 효율적으로 수집 할 수 있는, 즉 초당 수백 또는 수천 개의 웹 문서들을 수집 할 수 있는 웹 로봇의 필요성이 증가되고 있다.[1][2] 인터넷이 발달됨에 따라서 동적인 웹사이트가 증가하고 있다. 사용자들이 원하는 게시물을 등록하고, 삭제 하고 수정할 수 있는 웹사이트가 대부분 차지하고 있으며, 웹 로봇은 이러한 특성들을 고려하여 설계되어야 한다. 웹 수집 로봇은 URL을 통해서 웹 문서를 수집하게 되는데, 웹 사이트 개발자에 의하여 만들어진 자바스크립트 함수로 링크가 연결되어 있는 경우에 해당 페이지를 수집 할 수 없으며, 해당 페이지에 연결된 웹 링크를 찾아갈 수 없기 때문에 많은 웹 페이지를 놓치게 된다. 또한 웹 수집 로봇은 대부분의 기업/업체(검색엔진)에서 사용되고 있지만 웹 수집 로봇의 소스는 공개되어 있지 않다.

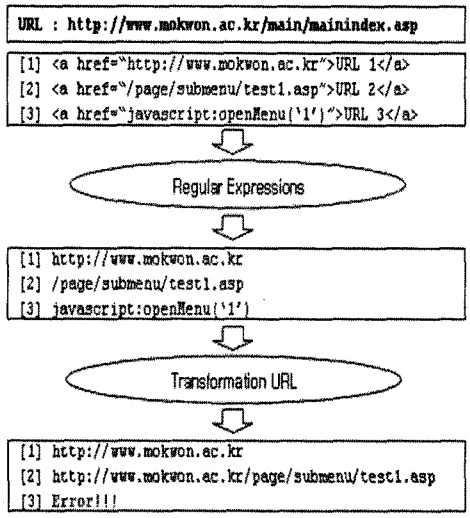

그림 1. 스크립트 처리가 불가능

본 논문에서는 그림 1과 같이 기존의 웹 로봇에서는 처리 하지 못하는 자바스크립트 함수 링크를 처리 하는 자바스크립트 모델을 제안하고, 제안된 스크립트 모델을 사용하는 웹 수집 로봇을 설계 및 구현 하였다. 또한 제안된 스크립트 모델을 사용하여 웹 수집 로봇의 수집 페이지의 양을 대상으로 웹 로봇의 수집량을 대상으로 성능평가 하였다. 웹 수집 로봇의 웹 페이지 수집량은 중요하기 때문이다. 마지막으로 결론 및 향후 연구에 대하여 알아본다.

2. 관련 연구

구글봇(Googlebot)은 웹 검색 서비스를 제공하는 구글에서 사용하고 있는 웹 로봇으로, 스탠포드 대학의 학생이였던 Page & Brin에 의해 개발되었다.[3] 구글은 이러한 구글봇을 이용하여 전 세계를 대상으로 30억개 이상의 웹 문서를 수집하고 있다. 또한, 구글은 상업화된 이후에도 스탠포드 대학과 웹 문서 수집에 관련된 연구를 지속적으로 수행하고 있으며, 그 결과로는 웹 문서들이 병렬 수집, 중복된 문서들의 검출, 동적인 웹 문서들의 수집, 웹 문서들의 수정 주기 분석 등이 있다.

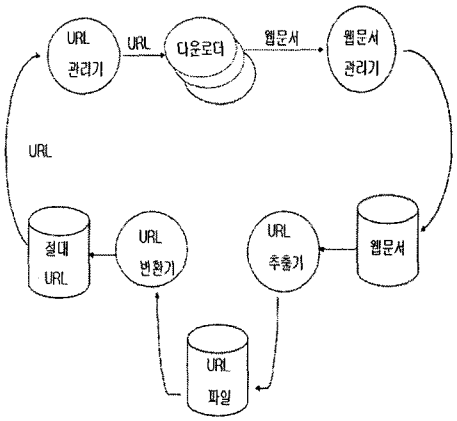

그림 2. 구글봇 시스템의 구조

그림 2는 구글봇 시스템의 구조를 보여 준다. 구글봇은 URL 관리기, 다운로더, 웹 문서 관리기, URL 추출기, URL 변환기로 구성되어 있으며, 각각의 구성 요소는 독립적인 프로세스로서 존재한다. URL 관리기는 수집할 웹 문서들의 URL들을 다수의 다운로더들에게 분배한다. 각각의 다운로더는 서로 다른 컴퓨터에서 실행되고, 웹 문서 관리기는 다운로드된 웹 문서들을 압축하여 디스크에 저장한다. URL 추출기는 디스크에 저장된 웹 문서들로부터 URL들을 추출하고, URL 변환기는 이 URL들을 절대 URL로 변환하여 저장한다. 네이봇(nabot)은 웹 검색 포탈 네이버에서 사용하는 웹 로봇으로서 국내 및 일본의 웹 문서들을 수집한다. 네이봇은 데이터베이스 관리 시스템이 MySQL을 사용하여 수집된 웹 문서들을 관리하며, 또한 과거에 수집된 웹 문서들을 지속적으로 수집하기 위하여 지금까지 수집된 전체 웹 문서들의 URL을 관리한다. 네이봇은 관리중인 URL들이 지시하는 웹 문서만을 수집하고, 수집된 웹 문서들로부터 발견된 새로운 URL들을 URL 데이터베이스에 추가한다. 따라서 새롭게 발견된 URL들이 지시하는 웹 문서들은 다음번 네이봇 수행 시에 수집된다. 그림 3은 네이봇 시스템의 구조를 보여준다. URL 분배기는 관리중인 URL들을 다수의 컴퓨터에 분산되어 있는 웹 문서 수집기들에게 분배하며, 웹 문서 수집기는 다음과 같은 작업들을 수행한다. 첫째, URL의 IP를 검사하여 국내 또는 일본의 웹 문서인지를 확인한다. 둘째, 로봇 배제 기준을 준수하기 위해서 웹 서버의 robots.txt 파일의 내용을 확인한다. 셋째, 웹 문서를 다운로드

한다. 넷째, 다운로드된 문서로부터 URL들을 추출하여 URL 검사기로 전달한다. 다섯째, 웹 문서의 내용을 분석하여 유해 또는 스팸 문서인가를 검사한다. 마지막으로, 웹 문서를 압축하여 데이터베이스에 저장한다. 한편, URL 검사기는 전달된 URL들 중에서 블랙리스트에 포함된 URL들과 기존의 URL 제거한 후, 나머지 URL들을 URL 데이터베이스에 추가한다.

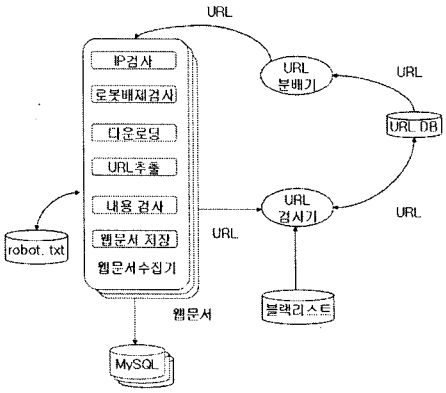

그림 3. 네이봇 시스템의 구조

로봇 배제 표준은 "robots.txt" 파일 또는 로봇 메타 태그를 이용하여 웹 로봇들의 우베 문서 수집을 제한한다. 즉, 웹 서버 관리자들은 "robots.txt"파일에 웹 로봇들의 이름을 명시함으로써 사이트내의 전체 웹 문서들에 대한 수집을 방지하거나, 또는 디렉토리 이름들을 명시함으로써 일부 웹 문서들에 대한 수집만을 방지할 수 있다. 또한, 웹 문서 작성자들은 로봇 메타 태그를 이용하여 작성된 웹 문서에 대한 수집을 제한할 수 있다. 따라서, 웹 로봇은 웹 문서를 수집하기 전에 "robots.txt" 파일과 로봇 메타 태그들을 확인하여, 지정된 내용을 준수해야 한다. 그림 4에서 사용되는 Meta 태그를 이용하여 로봇의 접근을 거부하는 방법은 일반적인 방법이 아니며, 아직까지는 일부의 로봇만 지원하고 있다.

```
User-agent: *
Disallow:
```

그림 4. 로봇 배제 표준의 예제

```
<meta name="Robots" content="Noindex,Nofollow" />
```

그림 5. HTML 메타태그 로봇 배제의 예제

3. 제안된 웹 수집 로봇의 설계 및 구현

제안된 웹 수집 로봇은 로봇 관리자로부터 시작 되지 않으며, 큐를 중심으로 시작된다. 로봇 관리자는 현재 쌓여 있는 메시지 큐를 처리하기 위해서 각 관리자(수집 관리자, 다운로드 관리자, 에러 처리 관리자, 스크립트 관리자)를 생성하게 되며 처리된 큐는 삭제된다. 수집 처리 관리자에서는 페이지의 웹 주소를 가져오는데, 가져온 URL을 메시지 큐에 입력하고, 자바스크립트로 연결된 함수 링크의 경우에도 메시지 큐를 작성하여 등록한다. 이렇게 등록된 큐를 로봇 관리자에서 처리 하면, 웹 페이지의 모든 문서를 수집 할 수 있게 되는 것이다. 여기서 스크립트 관리자는 본 논문에서 제안된 관리자 이다. 이 스크립트 관리자는 수집 처리 관리자에서 처리 할 수 없는 링크 즉 자바스크립트 함수로 연결된 링크를 처리 하도록 도와주는
역할을 한다.

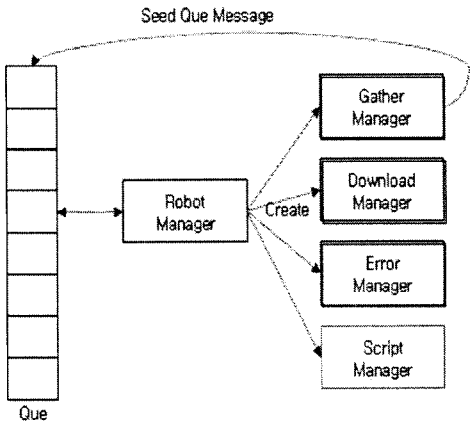

그림 6. 웹 수집 로봇의 기본 구조

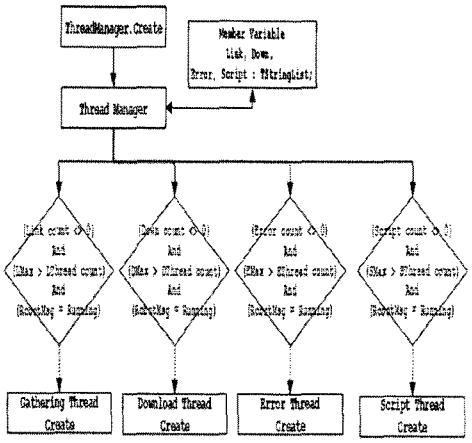

그림 7. 로봇 관리자의 구성

웹 수집 로봇			
기존 로봇		제안된 로봇	
수집량	소요시간	수집량	소요시간
133	59.95	1464	395.67
1	1.61	76	17.36
22	3.2	70	18.16
155	100.94	1650	1587.42
20	8.83	30	9.95

자바스크립트가 사용되는 경우 수집량의 차이가 확연하게 나타나고 있는 것을 확인 할 수 있다.

기존의 로봇에서는 *****.gangwon.kr 도메인을 처리 했을 경우에는 자바스크립트 함수 링크를 처리 하지 못하여 1페이지 밖에 수집하지 못하는 문제점이 있었지만, 제안한 웹 로봇의 경우에는 76페이지를 수집하였다.

4. 결론

본 논문에서는 웹 수집 로봇의 수집처리 관리자에서 페이지의 URL파싱처리 중 처리 하지 못하는 자바스크립트 함수 링크를 처리하기 위하여 인터넷 익스플로러객체에서 제공하는 "Active Script Engine"을 활용하여 처리하는 방법을 제안하였다. 본 논문에서 제한하는 사용자 정의 자바스크립트 함수 링크를 처리하는 모델을 사용하는 경우 지금까지 처리 하지 못하였던 자바스크립트 함수 링크의 문제를 해결하여 더욱 많은 양의 웹 페이지 수집이 가능 할 것이다. 현재 웹2.0의 시대를 도래 하면서 사용되는 언어가 증가되고 있다. 그중에 특히 관심을 가져야 할 부분이 바로 AJax이다. 현재 몇 개의 특정 웹사이트의 경우 AJax로 개발되어 있는 경우 웹 로봇의 수집이 불가능한 것을 확인 할 수 있었다. 이 문제를 해결하기 위해서는 해당 언어의 링크 처리 기법을 위한 연구가 수행되어야 한다.

참고문헌

[1] M. Gray, "Internet Growth and Statistics: Credits and Background" http://www.mit.edu/people/mkgray/net/background.htm

[2] Kwang Hyun Kim, "A Methodology for Performance Evaluation of Web Robot, Korea Information Processing Society Vol. 11, No. 3, June 2004, pp. 563-565