웹 로봇 구현 및 한국 웹 통계보고

김 성 진† · 이 상 호††

요 약

웹 로봇은 웹 문서를 다운로드하고 저장하는 프로그램이다. 현재 웹 로봇 구현에 대한 여러 연구들이 진행되고, 웹에 대한 다양한 통계들이 보고되고 있다. 첫째, 본 논문에서는 새로운 웹 로봇을 개발하고, 개발된 웹 로봇의 전체적인 구조와 구현 결정들을 기술한다. 둘째, 약 7천 4백만 한국 웹 문서들에 대한 여러 통계치를 보고한다. 셋째, 1,424개의 한국 웹 사이트를 지속적으로 관찰하여 웹 문서들의 변경 경향을 조사한다. 본 논문에서는 웹 문서의 변경에 영향을 미치는 요소들이 식별된다. 식별된 요소는 갱신할 웹 문서를 선택하기 위한 정보로서 유용하게 활용될 수 있다.

Implementation of a Web Robot and Statistics on the Korean Web

Sung Jin Kim† · Sang Ho Lee††

ABSTRACT

A web robot is a program that downloads and stores web pages. Implementation issues for developing web robots have been studied widely and various web statistics are reported in the literature. First, this paper describes the overall architecture of our robot and implementation decisions on several important issues. Second, we show empirical statistics on approximately 74 million Korean web pages. Third, we monitored 1,424 Korean web sites to observe the changes of web pages. We identify what factors of web pages could affect the changes. The factors may be used for the selection of web pages to be updated incrementally.

키워드 : 정보검색(Information Retrieval), 웹 검색 시스템(Web Retrieval System), 웹 로봇(Web Robot), 웹 크롤러(Web Crawler), 한국 웹 통계(Korean Web Statistics)

1. 서 론

인터넷 사용자들은 원하는 문서를 찾기 위하여 웹 검색 시스템에 질의한다. 웹 검색 시스템은 보유한 웹 문서에서 사용자의 질의에 적합한 문서를 추천한다. 웹 로봇(robot)은 - 경우에 따라 크롤러(crawler), 스파이더(spiders), 웜(worm), 워커 walker)로 불린다 - 웹 사이트들을 방문하여 웹 문서들을 자동적으로 수집하는 프로그램이다. 웹 검색 시스템들은 웹 로봇을 사용하여 웹 문서들을 수집한다. 웹 로봇은 웹 검색 시스템에서 매우 중요한 역할을 담당하고 있다.

웹의 출현과 더불어 웹 로봇에 대한 연구들[1-6]이 진행되어 왔다. 이들 연구에서는 웹 로봇의 구조, 작업 흐름, 구현상의 쟁점들이 논의되었다. 웹 로봇의 구현에서 고려되는 사항은 DNS 서버에 대한 병목 현상 최소화, 웹 사이트에 대한 부하 감소, URL의 중복처리, 문서 내용의 중복처리, 넓이-우선 문서 탐색 등이 있다. 웹 로봇 구현에 대한 연구는 웹 사이트에 대한 부하를 최소화하면서 문서 수집 속도를 최대화하는 것을 중요한 목적으로 한다. 인터넷 아카이브(Internet Archive)[1]은 사이트 기반의 문서수집을 수행하는 웹 로봇으로서, 빠른 중복 URL 처리를 위하여 블룸 필터(Bloom filter)를 사용하였다. Heydon과 Najork는 컴팩 SRC 크롤러(Compaq SRC crawler)[5, 7]의 구조를 보이고, 자바(Java) 웹 로봇 구현시 결정한 사항들을 기술하였다.

본 논문에서는 웹 로봇 구현에 고려되어야 하는 다양한 선택 사항들에 대한 지침을 제공한다. 이를 위하여, 시드 기반의 웹 로봇이 소개되고 구현과 운영에 대한 통계 자료들이 제공된다. 시드 기반의 웹 로봇은 사이트 기반의 웹 로봇에 비하여 보다 많은 웹 문서 수집이 가능하다. 또한, 시드 문맥 교환(seed context switching) 기법을 통하여 다수의 사이트에 대한 동시 문서수집을 효율적으로 수행한다. 본 논문에서는 기존 로봇 개발자들이 보고한 통계 결과에 비해 최근 통계 자료를 제공할 뿐 아니라, 기존 연구에서 제공되지 않은 통계 정보를 다룬다.

웹에 대한 각종 통계치는 웹 로봇과 같은 웹 어플리케이

※ 본 연구는 한국과학재단 목적기초연구(R 01 2000 00403) 지원으로 수행되었음.
† 준 회 원 : 숭실대학교 대학원 컴퓨터학과
†† 정 회 원 : 숭실대학교 컴퓨터학부 교수
논문접수 : 2002년 11월 23일, 심사완료 : 2003년 6월 2일

션 설계와 자원 할당에 대한 지침이 된다. 예를 들어, URL 문자열의 평균, 최대 길이는 시스템 개발자들이 URL 처리에 할당할 메인 메모리의 양을 결정하는데 사용될 수 있다. Heydon과 Najork는 1999년에 컴팩 SRC 크롤러[5]를 이용하여 6천 5백만 문서에 대해 HTTP 응답 코드(response code), 문서 크기 분포, MIME 타입 종류에 대한 통계 자료를 제공하였다. Shkapenyuk과 Suel은 2001년에 그들이 개발한 고성능 웹 로봇[6]을 이용하여 1억 2천만 웹 문서에 대한 HTTP 응답 코드, 평균 웹 문서 크기에 대한 통계 자료를 제공하였다. 본 로봇은 2002년에 한국의 웹 문서들을 대상으로 하여 약 7천 4백 만개의 웹 문서를 요청하고 6천 7백 만개의 웹 문서를 성공적으로 다운로드하였다.

수집된 웹 문서들을 최신 상태로 유지하기 위한 웹 로봇 운영은 크게 두 가지로 나뉜다. 주기적 로봇(periodic robot)은 처음 웹 문서를 수집한 것과 동일한 방법으로 웹 문서를 다시 수집하여 이전 문서 집합을 대체한다. 증분 로봇(incremental robot)은 구축한 웹 문서들을 유지하면서 변경되었을 확률이 높은 웹 문서만을 갱신한다[2]. 수집된 웹 문서의 최신 상태 유지 관점에서, 주기적 로봇과 증분 로봇의 효과성은 웹 문서의 변경 경향에 좌우된다[2]. 증분 로봇은 문서들의 변경주기를 정확하게 예측할수록 네트웍 대역폭의 사용을 줄이면서 웹 문서들을 최신상태로 유지할 수 있다. 웹 문서의 변경 경향에 대한 연구는 매우 중요함에도 불구하고, 현재까지 소수의 연구 결과가 발표되었다. Cho와 Garcia-Molina는 네 개의 도메인(".com", ".edu", ".net/.org", ".gov")에 속한 웹 문서들의 평균 변경 주기(interval)를 계산하여 ".com" 도메인에 속한 웹 문서들이 빈번하게 변경됨을 보였다[2, 8]. Brewington과 Cybenko는 미국 웹 문서들의 대부분이 업무 시간(오전 5시부터 오후 5시) 중에 변경된다는 연구 결과를 보였다[9]. 본 논문에서는 웹 문서의 변경에 영향 주는 다양한 요소를 고려하여 각 요소별 변경 주기 결과를 제시하였다. 예를 들어, 문서 내에 존재하는 URL의 종류에 따라 웹 문서의 변경 주기가 영향이 있음이 나타났다. 즉, '<FRAME>' 태그를 포함한 웹 문서나 URL을 포함하지 않는 웹 문서는 다른 문서들에 비해 변경 주기가 길게 나타났다.

본 논문은 다음과 같이 구성되었다. 2장에서는 본 로봇의 시스템 구조를 보이고 구현 고려 사항을 기술한다. 3장에서는 7천 4백 만개의 한국 웹 문서들에 대한 통계치를 나타낸다. 4장에서는 웹 문서들의 변경 경향을 관찰하고 웹 문서의 변경에 영향을 미치는 요소를 나타낸다. 마지막으로 5장에서 결론을 맺고 향후 연구 계획을 기술한다.

2. 웹 로봇

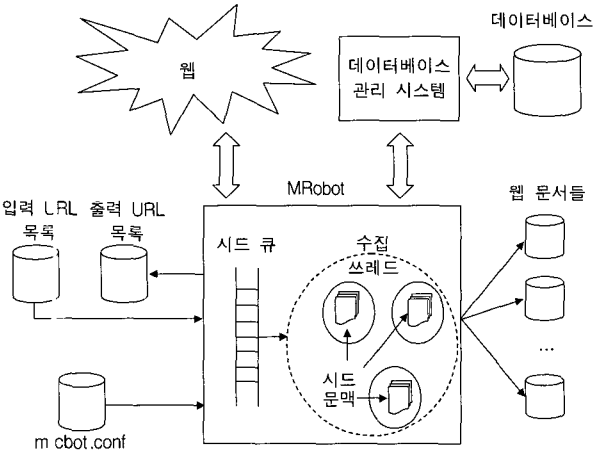
본 장에는 개발된 웹 로봇의 구조와 작업흐름이 나타나 있다. 일반적으로 하나의 웹 사이트는 다수의 웹 문서들로 구성된다. 웹 로봇은 사이트내의 한 문서를 선택하고, 선택한 문서로부터 같은 사이트에 속하는 문서들을 수집하기 시작한다.

사이트 기반 웹 문서 수집[1]은 유용한 문서 수집 기법이다. 웹 로봇은 각 사이트에 대한 IP 주소와 로봇 배제 규칙 정보를 메모리에 저장하여 같은 사이트 내의 웹 문서 수집에 반복적으로 사용한다. 또한, 한 사이트내의 문서들을 쉽게 한 장소에 저장할 수 있다. 다중 기계/쓰레드들은 다른 기계/쓰레드들과 중복을 고려하지 않고 각 사이트의 문서 수집 작업을 수행할 수 있다. 사이트 기반 웹 문서 수집은 홈페이지로부터 사이트내의 웹 문서들을 다운로드하기 시작한다. 따라서, 홈페이지로부터 연결되지 않는 문서들은 다운로드되지 못한다.

사이트 기반 웹 문서 수집은 시드 기반의 문서 수집으로 확장되었다. 시드 기반 웹 문서 수집은 하나의 사이트에 대해 여러 개의 수집 시작 문서를 가질 수 있다. 같은 사이트에 존재하는 문서 p와 q에 대하여 p가 q로부터 동일 사이트내의 문서(N, N이 $\emptyset$인 경우 포함)들을 통해서 연결될 수 있으면, 문서 p가 문서 q로부터 지역적으로 연결(locally reachable)되었다고 한다. 주어진 문서(우리는 이것을 '시드'라 한다)로부터 지역적으로 연결될 수 있는 웹 문서의 집합을 *파티*(party)라 한다. 일반적으로 웹 로봇은 사이트의 홈페이지(또는, 메인 페이지)를 시드로서 선택한다. 홈페이지로부터 연결되지 않은 문서가 존재하는 경우에, 웹 로봇은 홈페이지이외의 시드를 가질 수 있다. 즉, 시드 기반 웹 로봇은 한 사이트에 대해 하나 이상의 시드와 *파티*를 보유함으로써 웹 사이트(또는 전체 웹)의 커버리지(coverage)를 확장할 수 있다.

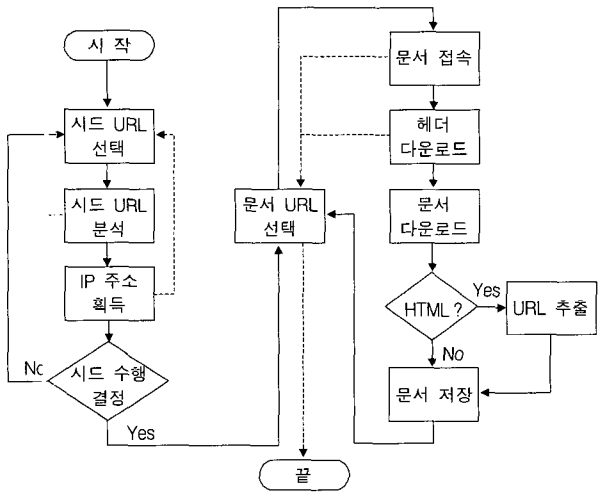
웹 로봇의 전체적인 구조는 (그림 1)에 나타나 있다. '입력 URL 목록'과 '출력 URL 목록'은 시드 URL을 포함한 텍스트 파일이다. 웹 로봇은 문서 수집을 시작하기 전에 '입력 URL 목록'의 시드 URL을 데이타베이스에 저장한다. 웹 로봇은 데이타베이스에 저장된 시드 URL로부터 문서 수집을 수행한다. 문서 수집 중에 새롭게 발견된 시드 URL은 데이타베이스에 저장된다. 웹 로봇이 문서 수집을 마치고, 데이타베이스에 저장된 시드 URL들은 '출력 URL 목록'에 저장된다. '출력 URL 목록'은 '입력 URL 목록'의 시드 URL을 모두 포함한다. '출력 URL 목록'은 이후의 문서 수집에서 '입력 URL 목록'으로 사용될 수 있다. 웹 로봇의 환경 설정 파일은 'mrobot.conf'로 나타나 있다. 'MRobot'은 실제 문서 수집을 담당하는 하위 시스템이다. 'MRobot'은 데이타베이스에서 시드 URL을 읽고 이를 시드 큐(seed queue)에 적재한다. 'MRobot' 내에는 다수의 수집 쓰레드들이 존재한다. 각 수집 쓰레드는 시드 큐에서 시드 URL을 얻고 해당 *파티*의 문서를 수집한다. 각 *파티* 당 하나의 시드 문맥(seed

context)이 생성된다. 수집된 문서는 디스크에 하나의 파일로 저장된다.



(그림 1) 웹 로봇 구조

수집 쓰레드의 작업 흐름은 (그림 2)에 묘사되어 있다. 일반선, 단속선, 점선은 각각 일반, 에러, 끝 흐름을 나타낸다. "시드 URL 선택" 단계에서 쓰레드는 처리할 시드 URL을 시드 큐에서 가져온다. "시드 URL 분석" 단계에서 시드 URL 문자열은 구문 분석되고, 구문 오류가 검사된다. 수집 쓰레드는 호스트(host) 이름, 포트 번호 등 수집할 *파티*의 필수 정보를 얻는다. 호스트 이름은 "Git IP" 단계에서 IP 주소로 바뀐다. "시드 수행 결정" 단계에서 쓰레드는 해당 *파티*의 웹 문서 수집 여부를 판단한다. 예를 들어, 본 로봇은 IP 주소에 근거하여 해당 *파티*의 한국 상주 여부를 판단하고, 한국에 상주하는 *파티*만을 수집한다.

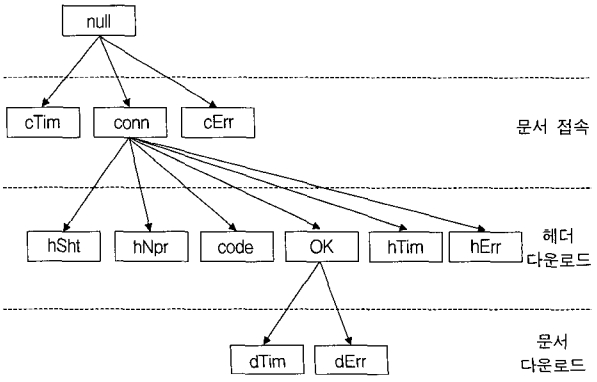


(그림 2) 수집 쓰레드의 작업 흐름

"문서 URL 선택" 단계에서, 수집 쓰레드는 *파티*에서 다운로드할 웹 문서의 URL을 선택한다. 이 단계는 (그림 2)에서 보여지는 바와 같이 반복적으로 수행된다. 이 단계가 최초로 수행될 때, 수집 쓰레드는 *파티* 내에 어떤 문서 URL이 있는지 알 수 없으므로 시드 URL을 선택한다. 쓰레드는 시드 문서를 다운로드하고 *파티*에 속하는 문서의 URL을 찾는다. 쓰레드는 발견한 문서 URL의 문서를 다운로드하여 *파티*에 속하는 새로운 문서 URL을 찾는다. 쓰레드는 *파티*에 속하는 새로운 문서 URL을 찾지 못할 때까지 문서 다운로드와 문서 URL 찾기를 반복한다. *파티* 내의 웹 문서들은 시드 문서를 루트(root)로 하는 트리(tree)로 표현될 수 있다. 시드에서 발견된 문서 URL의 문서는 깊이 1인 문서가 된다. 마찬가지로 깊이가 1인 문서로부터 발견된 문서는 깊이 2인 문서가 된다. 다운로드할 웹 문서 선택은 넓이-우선 문서 탐색을 따른다. 넓이-우선 문서 탐색은 효과적인 문서 수집 순서로 알려져 있다[10]. 웹 문서는 "문서 접속", "헤더 다운로드", "문서 다운로드" 단계를 거쳐서 다운로드된다.

다운로드한 문서가 HTML 문서이면, 본 로봇은 해당 문서에서 URL을 추출한다. 추출한 URL 중에서 다운로드한 문서와 같은 *파티*에 속하는 URL이 내부 URL이 된다. 내부 URL이 아닌 URL은 외부 URL로 간주된다. 수집 쓰레드는 추출된 URL을 내부 URL과 외부 URL로 나눈다. 이중 내부 URL은 수집 쓰레드 내에 저장된다. 외부 URL은 후보 시드 URL이 된다. 후보 시드 URL 중에서 웹 로봇이 이미 보유한 시드 URL과 중복되지 않는 URL이 실제 시드 URL이 된다. 시드 URL은 데이타베이스에 저장된다. 마지막으로, 수집 쓰레드는 다운로드한 웹 문서를 디스크에 저장한다.

웹 문서들의 네트워 상태는 가변적이다. 웹 로봇이 어떤 웹 문서를 다운로드하는데 실패했더라도, 그 문서는 다음 시도에서 성공적으로 다운로드될 수 있다. 경우에 따라 현재 웹 문서의 네트워 상태 정보는 과거의 네트워 상태 정보와 관련이 있다. 본 로봇은 각 문서의 네트워 상태 정보를 다음 문서 수집시 상태 예측에 사용한다. (그림 3)은 본 로봇에서 기록 가능한 네트워 상태 정보와 상태 정보의 변경 경로를 묘사한다. 다운로드 요청이 이루어지지 않은 문서의 네트워 상태는 널(NULL)이다. "문서 접속" 단계에는 세 가



(그림 3) 네트워 상태 정보

지 네트웍 상태가 존재한다. 'cErr', 'cTim'는 네트웍 에러와 타임아웃을 의미한다. 'Conn' 상태는 웹 서버와의 네트웍 연결이 성공적으로 이루어졌음을 의미한다. "헤더 다운로드" 단계에서 'hSht'는 다운받은 헤더가 너무 짧음을 의미한다. 'hNpr'은 해당 웹 페이지가 HTTP 프로토콜에 의해서 접근되지 않음을 의미한다. 'code'는 웹 서버의 응답 코드를 나타낸다. 'code'가 200이고 MIME 타입이 "text/html"이면, 네트웍 상태는 'OK'가 된다. 'hTim', 'hTim', 'hErr', 'dTim', 'dErr'은 "헤더 다운로드"와 "문서 다운로드" 단계에서 네트웍 시간초과와 에러를 나타낸다.

많은 웹 로봇들은 방대한 분량의 데이터 처리를 위하여 DBMS(데이타베이스 관리 시스템)를 사용하고 있다. 예를 들어, Heydon과 Narjork는 URL 관리를 위하여 커넥티비티 서버 2(Connectivity Server 2)를 구현하여 사용한다[5]. 본 로봇은 다중 쓰레드 환경에서 데이터의 동시성 제어와 문서 수집 동안에 얻어진 방대한 통계정보의 관리를 위하여 DBMS를 사용한다. 본 로봇은 데이타베이스 내에 두 종류의 테이블을 유지 관리한다. 한 종류의 테이블은 시드 URL에 관련된 정보를 저장하는 테이블로서 한 개가 존재한다. 나머지 한 종류의 테이블은 문서 URL에 관련된 정보를 저장하는 테이블로서 여러 개가 존재한다. 시드 URL 정보를 저장한 테이블은 시드 식별자, 시드 URL 문자열, 시드 URL 문자열에 대한 해시 값, 시드 문서의 내용에 대한 해시 값, *파티* 내의 모든 문서를 다운로드한 소요 시간, 시드 문서의 네트웍 상태 등을 저장한다. 본 로봇은 *파티* 내의 웹 문서들을 수집하면서 새로 발견된 시드 URL을 테이블에 삽입한다. 또한, 본 로봇은 *파티*의 문서 수집을 마칠 때마다 테이블에서 해당 시드의 컬럼 정보를 갱신한다. 문서 URL 정보를 저장한 테이블은 문서 식별자, 문서 URL 문자열, 문서 URL 문자열에 대한 해시 값, 문서의 내용에 대한 해시 값, 문서의 발견 깊이, 문서를 다운로드한 소요 시간, MIME 타입, 문서 내용이 저장된 운영체제 파일 이름, 문서의 네트웍 상태 등을 저장한다. 본 로봇은 문서 수집 동안에 발견된 새로운 문서 URL을 테이블들에 저장한다. 또한, 본 로봇은 한 문서를 다운로드하고 디스크에 저장할 때마다 해당 문서 URL의 컬럼 정보를 갱신한다.

웹 로봇은 같은 문서의 반복적인 다운로드를 피하여야 한다. 웹 로봇은 URL을 발견할 때마다 이미 발견했던 URL인가를 판단한다. 이러한 URL 중복을 처리하기 위하여, 적절한 중복 URL 인식 구조가 필요하다. 웹 로봇이 발견했던 URL 문자열 자체를 모두 저장하는 것은 방대한 저장 공간을 필요로 한다. Suel과 Yuan은 URL 문자열의 크기를 10 바이트 이하로 줄일 수 있는 압축 기법[11]을 소개하였다. Burner는 발견한 URL을 블룸-필터[1]에 저장하였다. 블룸-필터는 작은 메인 메모리 공간으로써 빠른 URL 중복 처리를 가능하도록 한다. 하지만, 발견되지 않은 URL을 발견된 URL로 인식하는 "잘못된 긍정(false positive)" 현상을 유발한다. 웹 로봇은 블룸-필터의 크기를 크게 하여 "잘못된 긍정"의 발생 가능성을 줄일 수 있다. 시드 기반의 웹 문서 수집에서 한 *파티* 내의 문서 URL은 다른 *파티*의 문서 URL과 중복되지 않는다. 따라서, 본 로봇은 내부 URL들의 중복 처리에 블룸-필터를 사용한다. 본 로봇은 각 *파티*마다 하나의 블룸-필터를 생성하고 각 *파티*내의 웹 문서만을 고려하여 중복 URL을 검사한다. 본 로봇은 전체 웹 문서에 대한 블룸-필터를 생성하지 않는다.

본 로봇은 다수의 쓰레드를 이용하여 웹 문서를 수집한다. 이러한 설계는 웹 로봇이 사용하는 네트웍 대역폭을 늘리는 반면 쓰레드들 간의 동기화를 필요로 한다. 본 로봇 시스템에서는 두 종류의 동기화가 필요하다. 첫째, 수집 쓰레드가 시드 큐에서 시드 URL을 얻을 때, 시드 큐는 동기화되어야 한다. 동기화 없이 다수의 쓰레드가 시드 큐를 접근하면, 같은 시드 URL이 둘 이상의 쓰레드에서 중복 처리될 수 있다. 또한, 처리되지 않은 시드 URL이 발생될 수 있다. 본 로봇의 설계에는 널리 알려진 "생산자/소비자" 쓰레드 모델이 채택되었다. 본 로봇에서 수집 쓰레드는 '생산자/소비자" 모델에서 소비자 쓰레드에 해당한다. 둘째, 각 수집 쓰레드가 발견한 외부 URL이 하나의 시드 URL 목록에 추가될 때, 시드 URL 목록은 동기화되어야 한다. 외부 URL은 시드 URL 목록에서 중복된 항목이 없을 때 추가될 수 있다. 이러한 중복 검사 시간 동안에 나머지 쓰레드들은 기다려야 한다. 추가하려는 외부 URL의 중복 검사 시간을 줄이는 것은 매우 중요하다. 본 로봇은 시드 URL 목록을 테이블로 구성하고 DBMS를 통하여 시드 URL 목록에 대한 동기화 문제를 효율적으로 해결한다.

웹 로봇은 웹 문서를 빠르게 수집하면서 웹 서버에 과부하를 주지 않아야 한다. 이를 위하여 웹 로봇은 한 웹 서버에서 연속적으로 문서를 다운로드하지 않는다. 웹 로봇은 한 웹 서버에서 문서를 다운로드하고 다음 시도에서는 다른 웹 서버에서 문서를 다운로드한다. 예를 들어, Shakapenyuk과 Suel는 로봇이 한 웹 사이트의 문서를 30초에 최대 한번만 다운로드하도록 하였다[6]. 본 로봇에는 다수의 수집 쓰레드가 동작하고, 각 쓰레드는 동시에 다수의 *파티*에서 문서를 다운로드한다. 본 논문에서는 시드와 *파티*의 정보 집합을 시드 문맥(seed context)라고 한다. 본 로봇은 IP 주소, 시드 URL 문자열, 포트 번호, 수집해야 할 문서 URL 목록, *파티*의 블룸-필터, *파티*에서 마지막으로 다운로드한 문서의 다운로드 소요 시간 등을 시드 문맥으로 구성하였다. 특히, *파티*에서 마지막으로 다운로드한 문서의 소요시간을 *'l-시간'* 이라고 한다. 시드 문맥은 메인 메모리에 저장된다. 수집 쓰레드가 동시에 여러 *파티*의 웹 문서를 다운로드할 때, 수집 쓰레드는 각 *파티*의 시드 문맥을 보유한다. 수집 쓰레드가 동시에 처리하는 *파티*의 양이 많아질수록 보유하는

시드 문맥의 개수도 많아진다.

시드 문맥은 하나의 활성 시드 문맥과 다수의 비 활성 시드 문맥들로 나뉜다. 수집 쓰레드는 활성 시드 문맥에서 하나의 웹 문서를 선택하고 다운로드한다. 다운로드를 마치면 비 활성 쓰레드 중의 하나가 활성 쓰레드로 전환되고, 기존의 활성 시드 문맥은 비 활성 시드 문맥으로 전환된다. 수집 쓰레드는 활성 시드 문맥을 포인터로 가리키고 있다. 시드 문맥 전환이 일어나면, 수집 쓰레드는 새로운 활성 시드 문맥으로 포인터를 이동시킨다. 이러한 전환을 시드 문맥 전환(seed context switching)이라고 한다. 비 활성 시드 문맥이 활성 시드 문맥으로 전환되기 위한 조건은 비 활성 상태로 존재한 시간이 30초 보다 크고 ('*l* - *시간*' * 10) 보다 커야 한다. 시드 문맥에서 모든 문서 URL의 문서가 다운로드되면, 시드 문맥은 해제되고, 새로운 *파티*의 시드 문맥이 생성된다.

웹 로봇은 URL을 IP로 변환하기 위하여 많은 DNS(Domain Name Server) 참조를 필요로 한다. DNS 참조에 생기는 병목 현상은 웹 로봇의 전체 성능을 현저하게 감소시킬 수 있다. 이러한 병목 현상은 DNS로의 질의가 블락킹(blocking)되고 동기화되는 특성으로부터 기인한다. 즉, DNS에 질의하는 쓰레드는 결과를 얻을 때까지 기다린다. 또한, 같은 기계에서 하나의 쓰레드가 DNS에 질의 중이면, 나머지 쓰레드들은 먼저 질의중인 쓰레드가 결과를 얻을 때까지 기다려야 한다. DNS 병목현상을 해결하기 위해, 기존의 웹 로봇[1, 5, 6]은 비 동기적인 DNS 클라이언트를 사용하였다. Shakapenyuk과 Suel는 GNU *adns*라는 비 동기적 DNS 클라이언트를 사용하였고[6], Heydon과 Narjork는 고유한 DNS 클라이언트를 구현하여 사용하였다[5]. 본 로봇은 병목현상을 줄이기 위해 DNS에 대한 질의 횟수를 줄이도록 고안되었다. 같은 *파티*내의 웹 문서들은 시드 문서와 동일한 IP 주소를 가진다. 한 *파티*에 속한 웹 문서들의 IP 주소는 DNS에 대한 질의 없이 시드 문맥에 저장된 시드 문서의 IP주소를 참조하여 얻어질 수 있다. 따라서, 본 로봇은 *파티* 당 한번의 DNS 질의를 필요로 한다.

본 로봇은 웹 문서를 하나의 파일로 저장한다. 같은 *파티*의 웹 문서들은 같은 디렉토리에 저장된다. 이러한 디렉토리를 '시드 디렉토리'라고 한다. 일반적인 파일 시스템에서 한 디렉토리는 수십/수백 만개의 파일들을 포함할 수 없다. 따라서, 본 로봇은 시드 디렉토리 안에 몇 개의 하위 디렉토리들을 생성하고, 각 하위 디렉토리가 10,000개의 파일들을 포함하도록 하였다. 또한, 본 로봇은 다수의 시드 디렉토리를 포함하는 상위 디렉토리를 생성하였다. 또한, 다수의 상위 디렉토리를 포함하는 '디스크 디렉토리'를 생성하였다. 본 로봇은 실제 디스크 장치를 '디스크 디렉토리'에 마운트하여 디스크 입/출력 작업을 분산하였다. 마지막으로, 파일 이름이나 디렉토리의 이름을 생성할 때 URL 문자열을 사용하여 생성하는 것은 좋지 않다. 왜냐하면 URL 문자열로 허용되는 문자들이 파일시스템에서 허락되지 않는 경우가 있기 때문이다. 본 로봇은 숫자의 조합으로 파일과 디렉토리의 이름을 구성하였다.

3. 한국 웹 통계

본 장은 한국의 수집 가능한 모든 웹 문서들에 대해 통계 정보를 제공한다. 제시된 통계 자료는 웹 로봇과 같은 웹 어플리케이션의 설계와 자원 할당에 대한 지침이 될 수 있다. 본 시험에는 128,446개의 국내 시드가 사용되었다. 문서 수집은 30개의 수집 쓰레드로 이루어 졌다. 각 수집 쓰레드는 동시에 15개의 시드 문맥을 가지고 작업하였다. 본 로봇은 각 *파티* 당 640,000 비트 크기의 블룸-필터를 할당하였다. 즉, 웹 로봇은 하나의 시드로부터 최대 640,000개의 웹 문서를 수집할 수 있다. URL의 최대 길이는 255문자로 설정되었다. 본 로봇은 한국의 웹 문서들만을 수집하였다. 한국 웹 문서의 판단은 IP 주소에 근거하였다.

본 웹 로봇은 총 128,446개의 시드 문서를 요청하여 118,852개의 시드 문서를 성공적으로 다운로드하였다. 또한, 총 73,954,114개의 웹 문서를 요청하여 67,488,127개의 웹 문서를 다운로드하였다. 한 웹 문서를 다운로드하는 동안 5초 이상 자료의 전송이 없으면 해당 웹 문서의 다운로드를 포기하였다. 웹 로봇은 하나의 시드 문서로부터 평균 600개 문서를 다운로드할 수 있었다. <표 1>과 <표 2>는 웹 문서의 요청에 따른 웹 서버 응답 결과를 나타낸다. 응답 결과의 분포는 1999년 [5]에서 발표된 결과와 비슷하게 나타났다. 즉, 로봇은 요청한 문서의 약 10퍼센트(percent)에 해당하는 문서를 다운로드할 수 없었다. 이러한 결과는 웹 문

〈표 1〉 시드 문서 응답 결과

응답 결과	개 수	비 율
code	5,259	4.09%
noIP	4,726	3.68%
dTim, hTim	609	0.48%
OK	111,852	87.08%
나머지 결과	6,000	4.67%
합 계	128,446	100.00%

〈표 2〉 웹 문서 응답 결과

응답 결과	개 수	비 율
cErr	78,316	0.11%
나머지 결과	227,310	0.30%
dErr, dTim	26,448	0.03%
code	6,133,913	8.29%
OK	67,488,127	91.26%
합 계	73,954,114	100.00%

서에서 발견된 URL의 약 10 퍼센트가 문서로 연결되지 않았음을 의미한다.

<표 3>은 성공적으로 다운로드된 웹 문서들을 대상으로 하여 URL 문자열의 길이 통계치를 나타낸다. 프로토콜 문자열(예를 들어, "http://")은 시드 URL 문자열에 포함되지 않았다. 시드를 다운로드하기 위한 특정 포트 번호(예를 들어, " : 8080")는 시드 URL 문자열에 포함되었다. 시드 URL의 최대 길이는 100문자로 제한되었다. 문서 URL 문자열은 시드 URL 문자열을 포함하지 않는다. 문서 URL은 최대 255문자까지 허용되었다. 본 시험에서 시드 URL 문자열의 길이가 49문자보다 크면 성공적으로 문서를 다운로드하지 못하였다. 즉, 데이터베이스 내에 시드 문자열을 저장하기 위한 공간은 50 바이트면 충분한 것으로 나타났다. <표 4>는 시드 URL의 형태를 나타낸다. "xxx"는 점('.')문자를 포함하지 않는 임의의 문자열을 의미한다.

〈표 3〉 URL 문자열 길이

	최소 길이	최대 길이	평균 길이
시드 URL 문자열의 길이	5	49	17.38
문서 URL 문자열의 길이	0	255	71.78

〈표 4〉 시드 URL 형태

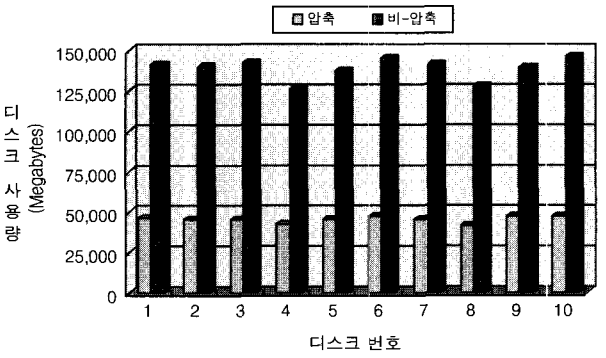
시드의 형태	개 수	비 율
"xxx"	0	0%
"xxx.xxx"	4,405	3.43%
"xxx.xxx.xxx"	55,918	43.53%
"xxx.xxx.xxx.xxx"	67,301	52.40%
"xxx.xxx.xxx.xxx.xxx"	810	0.63%
나 머 지	12	0.01%
합 계	128,446	100.00%

웹 문서의 평균 크기는 1997년 [1]에서 5킬로바이트, 1999년 [5]에서 15킬로바이트, 2001년 [6]에서 13킬로바이트로 보고되었다. 본 시험에서 한 문서의 최대 크기를 2메가바이트로 제한하였을 때, 평균 웹 문서 크기는 18,451바이트로 측정되었다. <표 5>는 웹 문서의 크기 분포를 나타낸다. 웹 문서들의 대부분은 크기가 256킬로바이트보다 작았고, 2메가바이트를 초과하는 크기의 웹 문서는 소수가 존재하였다. 즉, 웹 로봇이 약 1메가바이트의 메모리를 문서 다운로드를 위하여 할당할 경우, 99.9% 이상의 문서를 성공적으로 다운로드할 수 있다. 웹 로봇 개발자는 문서 저장을 위한 고정길이 자료 구조를 설계할 때, <표 5>에 근거하여 크기가 설정될 수 있다. 웹 로봇이 문서를 수집하는 동안에 크기가 무한대로 측정되는 문서가 종종 발견되었다. 이러한 웹 문서는 대부분 의미 없는 문서로 판명되었다. 예를 들어, 어떤 ASP(Active Server Page) 문서는 같은 에러 메시지를 반복적으로 출력하였다. 웹 문서는 평균적으로 약 14킬로바이트 크기에 해당하는 보이지 않는 내용(예를 들어, 태그, 스크립트 태그로 둘러싸인 내용들)을 포함하고 있었다.

〈표 5〉 웹 문서의 크기

문서 길이	개 수	문서길이	개 수
~256K	67,437,167	~1280K	3,474
~512K	31,512	~1536K	2,813
~768K	8,358	~1792K	561
~1024K	3,567	~2048K	675
평균 웹 문서 길이 : 18,451바이트			

웹 문서는 GZIP 형식으로 압축되어 저장되었다. 압축된 파일은 해당 웹 문서의 시드 번호에 기준하여 각 디스크로 나뉘어 저장되었다. (그림 4)는 각 디스크의 사용량을 나타낸다. 본 로봇은 67,488,127개의 웹 문서를 저장하는데 약 400기가바이트의 공간을 사용하였다. 또한, 웹 문서들은 효과적으로 각 디스크에 분산되었다. (그림 4)는 웹 문서 압축을 수행하지 않을 경우의 디스크 사용량을 부가적으로 나타내고 있다. 웹 문서를 압축하지 않고 저장할 경우에 본 로봇은 약 1.2테라바이트의 공간을 필요로 하였다. 웹 문서 압축은 실제 필요한 공간의 1/3 크기만으로 웹 문서 저장을 가능하도록 하였다.



(그림 4) 디스크 사용량

<표 6>은 문서 깊이에 따른 문서 개수를 나타낸다. 본 로봇은 128,446개의 깊이 0인 문서를 요청하였고, 111,852개의 문서를 다운로드하였다. 다운로드한 111,852개의 문서에서 623,250개의 내부 URL이 발견되었다. 본 로봇은 발견된 내부 URL(623,250개)의 문서를 요청한 결과 556,768개의 웹 문서를 다운로드하였다. 발견된 웹 문서의 약 90퍼센트는 성공적으로 다운로드되었으나, 10퍼센트는 해당 웹 문서로 연결되지 않았다. 깊이 5까지의 웹 문서를 고려할 때, 문서 수집의 깊이가 깊어질수록 발견된 내부 URL의 절대 개수가 많았다. 하지만, 깊이 0, 1, 2, 3, 4의 웹 문서에서 발견된 내부 URL의 개수는 각각 5.6, 6.0, 3.9, 2.4, 1.6로 나타났다. 즉, 문서 수집의 깊이가 깊어질수록 한 페이지에서 새롭게 발견되는 내부 URL의 개수는 적어졌다.

〈표 6〉 문서 깊이별 문서 개수

깊 이	요청한 문서 개수	다운로드한 문서 개수
0	128,446	111,852
1	623,250	556,768
2	3,348,544	2,885,489
3	11,319,688	10,005,216
4	24,054,489	21,628,864
5	34,479,697	32,299,938
합 계	73,954,114	67,488,127

4. 한국 웹 사이트 관찰

본 장에서는 한국 웹 사이트들을 관찰하여 웹 문서의 변경 경향을 조사한다. 관찰 대상은 웹 사이트의 랭킹 정보를 제공하는 사이트(랭크서브, http://www.rankserv.com/)에서 높은 순위를 갖는 사이트들로 선정되었다. '랭크서브'는 웹 사이트들을 15개의 카테고리(category)로 분류하고 각 카테고리별로 한국 웹 사이트의 순위를 부여하고 있다. 본 시험에서는 각 카테고리별로 상위 100위까지의 사이트들을 관찰 대상으로 선정하였고, 이 중에서 시험에 참여하는 것을 원하지 않는 사이트들을 제외하였다. 최종적으로 1,424개의 웹 사이트들이 관찰 대상으로 선정되었고, 모든 사이트 URL이 시드 URL로 사용되었다.

웹 문서들의 변경을 관찰하는 방법은 두 가지가 있다[2]. 첫 번째로 웹 로봇은 관찰할 문서를 정해놓고 매번 정해진 문서들을 재 수집하여 이전에 수집한 문서와 내용을 비교할 수 있다. 이러한 방법은 새로 발견된 페이지를 수집하지 못하는 단점이 있다. 두 번째로 웹 로봇은 관찰할 웹 사이트를 정해놓고 각 사이트별로 정해진 개수만큼의 문서를 수집하여 이전에 수집된 문서들과 비교할 수 있다. 본 시험에서는 두 번째 방법이 사용되었다. 본 로봇은 사이트 당 최대 3,000개의 문서를 요청하도록 설정되었다. 최대 문서 수집 깊이는 9로 제한되었다. 각 문서에는 5초의 타임아웃이 설정되었다. 본 논문에서는 로봇이 1,424개의 사이트로부터 웹 문서를 수집하는 작업을 일괄 수집(batch crawling)이라고 한다. 본 시험에서는 총 9차례의 일괄 수집이 이틀 간격으로 수행되었다.

<표 7>은 웹 로봇이 한번의 일괄 수집에서 요청하고 다운로드한 문서의 통계치를 나타낸다. 본 로봇은 첫 번째 일괄 수집에서 1,424개의 시드 문서를 요청하여 1,061개의 시드를 성공적으로 다운로드하였다. 또한, 다운로드한 시드로부터 1,406,908개의 문서 URL을 발견하였다. 본 로봇은 발견한 문서 URL에 해당하는 웹 문서를 요청하였고 최종적으로 1,211,677개의 문서를 다운로드하였다. 매 일괄 수집에서 본 로봇이 다운로드한 시드와 문서의 개수는 비슷하게 나타났다. 예를 들어, 첫 번째 일괄 수집에서 다운로드한 문서 개수와 세 번째 일괄 수집에서 다운로드한 문서 개수는 비슷하다. 이러한 현상은 첫 번째와 세 번째 일괄 수집에서 다운로드한 문서들이 비슷함을 의미하지 않는다. 두 일괄 수집에서 서로 다른 30만개의 문서가 발견되었다. 즉, 세 번째 일괄 수집에서 발견된 30만개의 문서는 첫 번째 일괄 수집에서 발견되지 않은 문서였다. 아홉 번의 일괄 수집에서 953개의 시드는 매번 성공적으로 다운로드되었다. 또한, 328개의 시드는 한번도 성공적으로 다운로드되지 못하였다. <표 8>은 다운로드 성공 횟수별 문서 개수를 나타낸다. 다운로드 실패는 (그림 3)에서 네트웍 상태가 'OK'가 아님을 의미한다.

〈표 7〉 웹 문서 수집 현황

일괄 수집 번호	요청한 시드 개수	다운로드한 시드 개수	요청한 문서 개수	다운로드한 문서 개수
1	1,424	1,061	1,406,908	1,211,677
2	1,424	1,068	1,455,310	1,220,577
3	1,424	1,061	1,418,636	1,221,553
4	1,424	1,056	1,407,297	1,212,688
5	1,424	1,057	1,390,072	1,196,345
6	1,424	1,058	1,406,211	1,207,717
7	1,424	1,061	1,415,663	1,222,457
8	1,424	1,066	1,422,022	1,224,266
9	1,424	1,059	1,400,288	1,204,671

〈표 8〉 다운로드 횟수

다운로드 횟수	시드 문서 개수
9	953
8	93
7	12
6	9
5	7
4	6
3	3
2	7
1	6
0	328
합 계	1,424

웹 로봇이 임의의 시드 문서를 다운로드하였을 때, 다음 일괄 수집에서 같은 시드를 성공적으로 다운로드할 확률을 SS라고 하자. 웹 로봇이 임의의 시드 문서에 대한 다운로드를 실패하였을 때, 다음 일괄 수집에서 해당 문서를 다운로드하지 못할 확률을 FF라고 하자. S는 시험에서 관찰된 시드의 개수이고, N은 일괄 수집 횟수라고 하자. 본 시험에서 S는 1,424가 되고, N은 9가 된다. 또한, N번의 일괄 수집에서 다운로드 성공횟수가 n인 시드들의 집합을 $S(n)$라고 하고, $S(n)$에 속한 시드의 개수를 $|S(n)|$이라고 하자. 본 시험에서 $|S(8)|$은 93이다(<표 8> 참조).

SS와 FF는 <표 8>에 기반하여 예측될 수 있다. SS의 산출 예를 들면 다음과 같다. 성공적으로 다운로드한 시드 문서 s가 있다고 하자. 첫째, s는 $S(0)$에 속한 시드가 아니다. s는 $S(1)$, $S(2)$, …, $S(9)$ 중의 하나에 속한 시드이다. 즉, s는 $(S - |S(0)|) = 1428 - 328 = 1096$개 시드 중 하나이다. s가 $S(1)$에 속할 확률은 $(|S(1)|/(S - |S(0)|)) = (6/(1424 - 328))$이다. 어떤 i가 1과 N사이의 정수라고 할 때, 해당 시드가 $S(i)$에 속할 확률은 $(|S(i)|/(S - |S(0)|))$이다. 둘째, s가 $S(1)$에 속한 시드라면, 로봇은 다음 일괄 수집에서 s를 다운로드하지 못할 것이다. 만약 s가 $S(2)$에 속한 시드라면, 로봇이 다음 일괄 수집에서 성공적으로 다운로드할 확률은 $((2-1)/(N-1)) = (1/8)$이다. 이와 같이, s가 $S(i)$중의 하나라고 가정할 때, 다음 일괄 수집에서 성공적으로 문서를 다운로드할 확률은 $((i-1)/(N-1))$이다. 셋째, s가 $S(i)$ 중의 하나이고 다음 일괄 수집에서 성공적으로 문서를 다운로드할 확률은 $(|S(i)|/(S - |S(0)|)) \times ((i-1)/(N-1))$이다. 마지막으로, SS는 모든 $(|S(i)|/(S - |S(0)|)) \times ((i-1)/(N-1))$의 합이다. SS는 식 (1)과 같이 표현될 수 있다. SS와 유사하게 FF는 식 (2)와 같이 표현된다. 본 시험에서 SS와 FF는 0.9638과 0.7426로 측정되었다. 즉, 성공적으로 다운로드된 시드는 다음 문서 수집 시도에서도 성공적으로 다운로드될 확률이 약 96%로 나타났다.

$$SS = \frac{1}{(S - |S(0)|)} \cdot \frac{1}{(N-1)} \cdot \sum_{i=1}^{N} |S(i)| \cdot (i-1) \quad (1)$$

$$FF = \frac{1}{(S - |S(N)|)} \cdot \frac{1}{(N-1)} \cdot \sum_{i=0}^{N-1} |S(i)| \cdot (N-1-i) \quad (2)$$

URL의 도메인에 따른 시드 문서 변경은 <표 9>에 나타나 있다. <표 9>는 도메인별 시드 개수, 9번의 일괄 수집에서 모두 성공적으로 다운로드된 시드 개수, 문서의 내용이 변경되지 않은 시드 개수, 문서 변경이 한번 이상 있었던 시드 개수와 변경주기를 나타낸다. 변경 주기는 관찰 기간을 변경횟수로 나눈 값이다. 예를 들어, 어떤 시드가 10일 동안 관찰되었고 그 기간에 5번 변경되었으면, 해당 시드의 변경주기는 10일/5 = 2일이다. 관찰한 사이트들 중에서 약 50%에 해당하는 761개 시드가 "kr" 도메인에 속하였다. "kr", "com", "net", "org"이외에 10개 이상의 시드를 포함한 도메인은 존재하지 않았다. "net" 도메인에 속한 시드가 116개가 존재하였고, 이 중에서 70개의 시드들이 모든 일괄 수집에서 성공적으로 다운로드되었다. 70개의 시드들 중에서 45개의 시드(64.3%)는 내용이 변경되지 않았다. 또한 25개의 시드(35.7%)는 한번 이상 변경되었다. 변경된 문서의 평균 변경 주기는 9일이었다. 문서 변경 관찰 기간이 19일임을 고려할 때, 본 시험에서 나타날 수 있는 최대 변경 주기는 19일이다. <표 9>에서 "net" 도메인에 속한 시드들이 다른 도메인에 속한 시드에 비해서 자주 변경되지 않음을 알 수 있다.

〈표 9〉 도메인별 시드 문서 변경 경향

도 메 인	시드	성공 시드		변경 안된 시드		변경된 시드		변경 주기
		개 수	비 율	개 수	비 율	개 수	비 율	
kr(한국)	761	523	68.7%	344	65.8%	179	34.2%	6.86
com(기업)	485	320	66.0%	187	58.4%	133	41.6%	6.32
net(네트웍)	116	70	60.3%	45	64.3%	25	35.7%	9.03
org(공공기관)	21	14	66.7%	6	42.9%	8	57.1%	6.24

<표 10>은 "kr" 도메인을 "ac.kr", "co.kr", "go.kr", "or.kr", "pe.kr"로 세분화하고, 각 도메인별 변경 경향을 나타낸다. "기업" 도메인에 속한 558개의 시드들 중에서 모든 일괄 수집에서 성공적으로 다운로드된 시드는 379개였다. 이 중 234개의 시드(61.7%)가 한번도 변경되지 않았고, 145개의 시드(38.3%)가 한번 이상 변경되었다. "기업" 도메인은 다른 도메인에 비해서 "변경된 시드" 비율이 높고 변경주기를 낮게 나타났다. "개인" 도메인은 다른 도메인에비해 "변경 안된 시드" 비율이 높고 변경주기가 높게 나타났다. <표 9>, <표 10>에서 "기업" 도메인에 속한 시드들이 다른 도메인에 속한 시드에 비해 자주 변경되고 있음을 알 수 있다. 또한, "개인" 도메인에 속한 시드들은 자주 변경되지 않음을 알 수 있다.

〈표 10〉 한국 도메인별 시드 문서 변경 경향

한국 도메인	시드	성공 시드		변경 안된 시드		변경된 시드		변경 주기
		개수	비율	개수	비율	개수	비율	
ac.kr(학교)	55	37	67.3%	26	70.3%	11	29.7%	10.65
co.kr(기업)	558	379	67.9%	234	61.7%	145	38.3%	6.16
go.kr(정부)	24	18	75.0%	13	72.2%	5	27.8%	9.98
or.kr(공공기관)	85	63	74.1%	50	79.4%	13	20.6%	8.88
pe.kr(개인)	16	15	93.8%	12	80.0%	3	20.0%	15.83

본 시험에서는 142개의 시드(관찰한 1,424개의 시드에서 약 10%를 임의로 추출한 시드)에서 다운로드한 웹 문서들을 관찰하였다. 웹 로봇은 9번의 일괄 수집에서 총 320,787개의 웹 문서를 다운로드하였다. 이 중 32,114개의 웹 문서는 매 일괄 수집에서 성공적으로 다운로드되었다. 159,210개의 웹 문서는 단 한번의 일괄 수집에서만 다운로드되고 그 이외에는 다운로드되지 못하였다. <표 11>은 9번의 일괄 수집에서 얻어진 문서들을 다운로드 성공 횟수별로 구분하였다. <표 12>는 문서 깊이별 문서의 수와 변경 주기를 나타낸다. 대부분의 웹 문서들은 깊이 3, 4, 5에서 다운로드되었고, 두 번 이상 다운로드되는 확률이 0.5 미만이었다. 특히, 깊이 4에서 다운로드된 문서가 가장 많았고, 깊이 4인 문서의 38.54 퍼센트에 해당하는 문서만이 두 번 이상

다운로드되었다. 변경 주기를 보면, 깊이 0인 문서(즉, 시드 문서)가 자주 변경되지 않음을 나타낸다.

〈표 11〉 관찰 대상 문서

다운로드 횟수	문서 개수	비 율
1	159,210	49.63%
2	32,114	10.01%
3	16,319	5.09%
4	14,179	4.42%
5	9,408	2.93%
6	6,923	2.16%
7	5,899	1.84%
8	15,700	4.89%
9	61,035	19.03%
합 계	320,787	100.00%

〈표 12〉 문서 깊이별 변경 경향

문서 깊이	문서 개수	두 번 이상 다운로드된 문서		
		개 수	비 율	변경 주기
0	128	111	86.72%	7.66
1	2,282	1,493	65.43%	5.35
2	19,972	11,229	56.22%	5.81
3	78,579	33,490	42.62%	6.36
4	88,943	34,283	38.54%	4.85
5	64,472	31,391	48.69%	5.36
6	47,176	20,773	44.03%	5.21
7	12,719	6,359	50.00%	5.42
8	4,391	2,258	51.42%	4.95
9	2,125	1,494	70.31%	3.77

본 논문에서 URL은 'f-URL', 'a-URL', 'h-URL'의 세 가지 종류로 구분된다. "<frame>" 태그에서 발견된 URL은 'f-URL'이 된다. "<a href>" 태그에서 발견된 URL은 'a-URL'이 된다. 그 외의 URL은 'h-URL'이 된다. 예를 들어, 자바 스크립트로부터 발견된 URL은 'h-URL'이 된다. 웹 문서의 "연결 종류"는 웹 문서가 포함하는 URL의 종류에 따라서 'h', 'a', 'f' 문자의 조합으로 표현된다. 예를 들어, "연결 종류"가 "0a0"인 웹 문서는 하나 이상의 'a-URL'을 포함한다. "연결종류"가 "haf"인 웹 문서는 'h-URL', 'a-URL', 'f-URL'을 각각 하나 이상 포함한다. URL을 포함하고 있지 않는 웹 문서의 "연결종류"는 "000"이 된다. 본 시험에서 "0af"인 웹 문서는 발견되지 않았다. 웹 문서의 "연결종류"와 변경 주기의 관계는 <표 13>에 나타나있다. URL을 포함하지 않은 문서("연결종류"가 "000"인 문서)들의 대부분(98.3 %)은 관찰 기간동안 변경되지 않았다. 또한, "<frame>" 태그를 포함한 문서("연결종류"가 "00f", "h0f", "haf"인 문서)들도 자주 변경되지 않는 것으로 관찰되었다. <표 12>에서 시드 문서의 변경주기가 길게 나타나는 현상은 많은 시드 문서들이 "<frame>" 태그를 포함하기 때문에 발생한다.

〈표 13〉 문서 연결 종류별 변경 경향

연결 종류	두 번 이상 다운로드된 문서	변경되지 않은 문서		변경된 문서		
		개 수	비 율	개 수	비 율	변경 주기
000	12,016	11,813	98.31%	203	1.69%	15.33
00f	2,104	2,056	97.72%	48	2.28%	17.65
0a0	35,478	16,906	47.65%	18,579	52.35%	4.36
h00	4,969	2,564	51.60%	2,405	48.40%	10.13
h0f	134	105	78.36%	29	21.64%	16.77
ha0	88,882	24,354	27.40%	64,528	72.60%	6.31
haf	15	12	80.00%	3	20.00%	19

5. 결론 및 향후 계획

본 논문에서는 다음의 세 가지 내용이 논의되었다. 첫째, 개발된 웹 로봇의 전반적인 구조와 구현상의 몇 가지 중요한 사항들이 기술되었다. 둘째, 약 7천 4백 만개 한국 웹 문서들의 통계 정보가 보고되었다. 웹 문서들에 대한 통계 정보는 웹 로봇뿐 아니라 웹 어플리케이션 개발 및 설계에 유용하게 사용될 수 있다. 셋째, 1,424개의 웹 사이트에서 발견된 웹 문서들의 변경 경향이 관찰되었다.

본 로봇은 다중 기계(multiple machine)를 사용하는 웹 문서 수집 시스템에 쉽게 적용될 수 있다. 관리자는 본 로봇을 각 기계에 설치하고 수집할 시드 URL을 분배한다. 각 기계에서 발견된 내부 URL들은 다른 기계에서 발견된 내부 URL과 중복되지 않는다. 따라서, 한 기계의 웹 로봇은 다른 기계에서 동작 중인 웹 로봇과 상관없이 내부 URL에 해당하는 웹 문서를 수집할 수 있다. 각 기계에서 발견된 외부 URL들은 모든 웹 로봇이 문서 수집을 마친 이후에 하나의 시드 URL 목록으로 통합된다.

본 논문에 나타난 통계정보를 적용하여 소개된 웹 로봇의 성능을 향상시키는 것이 가능하다. 첫째, 본 논문에 소개된 통계자료에 따르면 문서 깊이가 깊어질수록 더 많은 문서를 수집할 수 있다. 따라서, 웹 로봇의 문서 수집 깊이를 6 이상으로 하여, 웹 문서 커버리지를 확장한다. 둘째, 현재 웹 로봇은 웹 문서와 URL 문자열 처리를 위하여, 가능한 많은 공간을 메모리, 디스크, 데이터베이스에 할당한다. 보고된 통계 정보에 따라 불필요하게 할당된 메모리/디스크 공간을 조정한다. 셋째, 소개된 로봇은 웹 문서 하나당 파일 하나로 압축하여 저장한다. 이와 같은 방식에서 저장되는 파일의 크기는 줄어들지만, 파일의 개수는 그대로 유지된다. 웹 로봇이 수집 대상 문서를 한국에서 모든 국가들의 문서로 확장할 경우 운영체제가 허락하는 파일 개수에 제한을 받을 수 있다. 따라서, 향후 웹 로봇은 여러 개의 파일을 하나로 묶어 압축하는 ZIP 형식 압축 방법이 고려되

어야 한다.

웹 문서의 변경 주기를 알기 위하여 웹 문서들의 과거 변경 기록을 모두 유지 관리하는 것은 쉽지 않다. 향후 과제는 웹 문서의 과거 변경 기록 없이 현재 추출한 문서 정보만을 바탕으로 하여 향후 변경 주기를 예측하는 것이다. 이를 위하여 웹 문서의 변경 주기 예측 모델을 만들고, 웹 문서 변경과 밀접하게 관련 있는 요소를 발견하여야 한다. 웹 문서 수집 시 얻을 수 있는 정보 중 웹 문서 변경과 관련이 있는 요소들이 본 논문에서 일부 소개되었다. 예를 들어, URL을 포함하지 않는 문서는 잘 변경되지 않으며, URL의 접미사가 "co.kr" 또는 "com"인 웹 문서는 자주 변경되었다.

참 고 문 헌

[1] M. Burner, "Crawling Towards Eternity : Building an Archive of the World Wide Web," Web Techniques Magazine, Vol.2, No.5, pp.37-40, 1997.

[2] J. Cho and H. Garcia-Molina, "The Evolution of the Web and Implications for an Incremental Crawler," Proc. 26th VLDB Conf., pp.200-209, 2000.

[3] J. Cho and H. Garcia-Molina, Parallel Crawlers, Proc. 11th WWW Conf., pp.124-135, 2002.

[4] M. Diligenti, F. M. Coetzee, S. Lawrence, C. L. Giles and M. Gori, "Focused Crawling using Context Graphs," Proc. 26th VLDB Conf., pp.527-534, 2000.

[5] A. Heydon and M. Najork, "Mercator : A Scalable, Extensible Web Crawler," International Journal of WWW, Vol.2, No.4, pp.219-229, 1999.

[6] V. Shkapenyuk and T. Suel, "Design and Implementation of a High-performance Distributed Web Crawler," Proc. 18th Data Engineering Conf., pp.357-368, 2002.

[7] A. Heydon and M. Najork, "Performance Limitations of the Java Core Libraries," Proc. 1st Java Grande Conf., pp.35-41, 1999.

[8] J. Cho and H. Garcia-Molina, "Synchronizing a Database to Improve Freshness," Proc. 26th SIGMOD Conf., pp. 117-128, 2000.

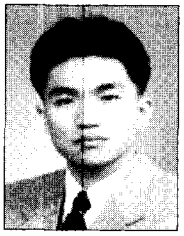
[9] B. Brewington and G. Cybenko, "How Dynamic is the Web?," Proc. 9th WWW Conf., pp.257-276, 2000.

[10] M. Najork and J. L. Wiener, "Breadth-first Crawling Yields High-quality Pages," Proc. 10th WWW Conf., pp. 114-118, 2001.

[11] T. Suel and J. Yuan, "Compressing the Graph Structure of the Web," Proc. 11th Data Compression Conf., pp. 213-222, 2001.

[12] J. Cho, H. Garcia-Molina, and L. Page, "Efficient Crawling through URL Ordering," Proc. 7th WWW Conf., pp. 161-172, 1998.

[13] S. Raghavan and H. Garcia-Molina, "Crawling the Hidden Web," Proc. 27th VDLB Conf., pp.129-138, 2001.



김 성 진

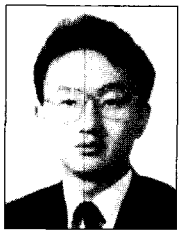
e-mail : lace@nownuri.net

1998년 숭실대학교 소프트웨어 공학과 (학사)

2000년 숭실대학교 대학원 컴퓨터학과 (석사)

2002년~현재 숭실대학교 컴퓨터학과 대학원 박사과정 수료

관심분야 : 인터넷 데이타베이스, 데이타베이스 시스템 성능평가



이 상 호

e-mail : shlee@computing.soongsil.ac.kr

1984년 서울대학교 전산공학과(학사)

1986년 미국 노스웨스턴대 전산학과(석사)

1989년 미국 노스웨스턴대 전산학과(박사)

1990년~1992년 한국전자통신 연구원, 선임 연구원

1999년~2000년 미국 조지 메이슨대 소프트웨어 정보 공학과 교환 교수

1992년~현재 숭실대학교 컴퓨터학부 부교수

관심분야 : 인터넷 데이타베이스, 데이타베이스 시스템 성능 평가 및 튜닝