빅데이터 분석 서비스 지원을 위한 지능형 웹 크롤러

Intelligent Web Crawler for Supporting Big Data Analysis Services

서동민, 정한민

한국과학기술정보연구원 소프트웨어연구센터 컴퓨터지능연구실

Dongmin Seo(dmseo@kisti.re.kr), Hanmin Jung(jhm@kisti.re.kr)

요약

빅데이터 분석을 위해 활용되는 데이터로는 뉴스, 블로그, SNS, 논문, 특허 그리고 센서로부터 수집된 데이터 등 매우 다양한 유형의 데이터가 있다. 특히, 신뢰성 있는 데이터를 실시간 제공하는 웹 문서의 활용이 점차 확산되고 있다. 그리고 빅데이터의 활용이 다양한 분야로 점차 확산되고 웹 데이터가 매년 기하급수적으로 증가하면서 웹 문서를 자동으로 수집하는 웹 크롤러의 중요성이 더욱 커지고 있다. 하지만, 기존 크롤러들은 일부 사이트에서 수집된 웹 문서에 포함된 URL만을 기반으로 웹 문서를 수집하기 때문에 사이트 전체 웹 문서를 수집할 수 없는 문제를 가진다. 또한, 수집된 웹 문서에 대한 정보를 효율적으로 관리하지 못하기 때문에 중복된 웹 문서를 수집하는 문제를 가진다. 그래서 본 논문에서는 웹 사이트의 RSS와 Google Search API를 통해 기존 웹 크롤러의 문제를 해결하고 RMI와 NIO를 활용해 서버와 클라이언트 간 네트워크 연결을 최소화해 빠른 크롤링 기능을 제공하는 분산형 웹 크롤러를 제안한다. 또한, 제안하는 웹 크롤러는 웹 문서를 구성하는 태그들에 대한 키워드 유사도 비교를 통해, 분석에 활용되는 중요 콘텐츠만을 자동 추출하는 기능을 제공한다. 마지막으로, 기존 웹 크롤러와 제안하는 크롤러의 성능 평가 결과를 통해 제안하는 웹 크롤러의 우수성을 입증한다.

■ 중심어 : | 웹 크롤러 | 콘텐츠 추출 | 빅데이터 | RMI | NIO |

Abstract

Data types used for big-data analysis are very widely, such as news, blog, SNS, papers, patents, sensed data, and etc. Particularly, the utilization of web documents offering reliable data in real time is increasing gradually. And web crawlers that collect web documents automatically have grown in importance because big-data is being used in many different fields and web data are growing exponentially every year. However, existing web crawlers can't collect whole web documents in a web site because existing web crawlers collect web documents with only URLs included in web documents collected in some web sites. Also, existing web crawlers can collect web documents collected by other web crawlers already because information about web documents collected in each web crawler isn't efficiently managed between web crawlers. Therefore, this paper proposed a distributed web crawler. To resolve the problems of existing web crawler, the proposed web crawler collects web documents by RSS of each web site and Google search API. And the web crawler provides fast crawling performance by a client-server model based on RMI and NIO that minimize network traffic. Furthermore, the web crawler extracts core content from a web document by a keyword similarity comparison on tags included in a web documents. Finally, to verify the superiority of our web crawler, we compare our web crawler with existing web crawlers in various experiments.

■ keyword : | Web Crawler | Content Extraction | Big-data | RMI | NIO |

1. 서론

최근 빅데이터(Big-data)가 IT 뉴스의 핵심 키워드로 부상했다. 빅데이터란 대용량 데이터 활용·분석을 통해 가치 있는 정보를 추출하고, 이를 바탕으로 대응 방안 도출 또는 변화를 예측하기 위한 정보화 기술을 말한다. 세계 각국의 정부와 주요 민간 기업들은 빅데이터가 새로운 경제적 가치의 원천이 될 것으로 예상하고, 빅데이터를 활용한 시장 동향, 신산업 발굴 등 경제적 가치 사례 및 효과를 제시하고 있다[1][2]. 일례로, [그림 1(a)]와 같이 미국 국립보건원은 1,000 Genomes Project를 통해 획득한 200TB의 유전자 정보를 아마존 웹 서비스를 통해 연구자들에게 제공함으로써, 새로운 질병에 대한 빠른 진단 서비스와 난치병 및 불치병에 대한 치료제 개발의 가능성을 제시했다[3]. 그리고 [그림 1(b)]와 같이 SK 텔레콤은 뉴스, 블로그, 카페, SNS 등에서 발생하는 다양한 온라인 여론 정보를 수집·분석해 기업의 마케팅 전략수립, 홍보, 고객관리 서비스를 지원하는 스마트 인사이트를 서비스하고 있다[4].

(a) NIH의 1,000 Genomes Project

(b) SKT의 스마트 인사이트

그림 1. 빅데이터를 활용한 서비스 예

빅데이터 분석을 위해 활용되는 데이터로는 뉴스, 블로그, 카페, SNS, 논문, 특허 그리고 센서로부터 수집된 데이터 등 매우 다양한 유형의 정형 또는 비정형 데이터가 있다. 그리고 블로그, 카페, SNS와 같이 개인의 의

견이 반영된 주관적인 데이터보다는 뉴스, 논문, 특허와 같이 사실을 바탕으로 한 객관적인 데이터가 신뢰성 있는 정보를 기반으로 한 의사결정 및 트렌드 분석에 있어 중요한 비중을 차지한다. 특히, 논문과 특허 외에도 신뢰성 있는 데이터를 실시간 제공하는 웹 문서의 활용이 점차 확산되고 있다. 일례로, [그림 2(a)]와 같이 미국 에너지부는 웹 기사, 과학문헌 그리고 특허 데이터 분석을 통해 태양광발전 시장에 관한 데이터 분석을 통해 경제적, 사회적 이슈가 태양광발전 설비 설치에 미치는 영향 분석 모델을 개발해 태양광발전 설치비용을 절감했다[5]. 그리고 한국과학기술정보연구원은 과학기술분야 웹 기사, 논문 그리고 특허로부터 기술, 연구자, 기관 정보를 추출·분석해 최신 기술동향 정보를 제공하는 InSciTe Advanced를 개발했다[6][7].

(a) DOE의 태양광발전 프로젝트

(b) KISTI의 InSciTe Advanced

그림 2. 웹문서 기반의 빅데이터 활용 서비스 예

웹 크롤러(Web Crawler)는 방대한 웹 문서를 제공하는 웹에서 특정 사이트의 웹 문서를 자동으로 수집하는 기술을 말한다[8][9]. 특히, 빅데이터의 활용이 다양한 분야로 점차 확산되고 웹 데이터가 매년 기하급수적으로 증가하면서 웹 크롤러의 중요성은 더욱 커지고 있다. 기존에 개발된 웹 크롤러는 일반 크롤러(General Crawler)와 분산 크롤러(Distributed Crawler)로 분류할 수 있다[10]. 하지만, 기존에 개발된 웹 크롤러들은 수집할 웹 문서 URL 목록인 시드(Seed)를 기준으로 웹

문서를 수집하고 수집된 웹 문서에 포함된 URL을 다음 시드로 활용해 다음 웹 문서를 수집하기 때문에, 첫 시드가 특정 웹 사이트의 전체 데이터를 수집할 수 있는 대표 시드가 아니면 해당 웹 사이트의 전체 데이터를 수집할 수 없는 문제를 가진다. 또한, 웹 문서 수집 속도 향상을 위해 활용하는 분산 크롤러의 경우, 스케줄링은 서버 측에서 오직 수행되고 클라이언트 측에서는 서버로부터 전달받은 시드를 기준으로 웹 문서를 수집하기 때문에 서로 다른 클라이언트가 중복된 웹 문서를 수집(Overlapped Crawl)하는 문제를 가진다.

본 논문에서는 RMI(Remote Method Invocation)와 NIO(New Input/Output) 기반의 분산형 웹 크롤러를 제안한다. 제안하는 웹 크롤러는 웹 사이트의 RSS(Rich Site Summary)[11]와 Google Search API[12]를 통해 시드를 수집함으로써 특정 웹 사이트의 전체 데이터를 수집할 수 있는 기능 제공과 Overlapped Crawl 문제를 해결했다. 그리고 RMI와 NIO를 활용해 서버와 클라이언트 간 네트워크 연결을 최소화해 빠른 크롤링 기능을 제공한다. 또한, 빅데이터 분석에 활용되는 웹 데이터만을 추출하기 위해, 웹 문서를 구성하는 태그들에 대한 키워드 유사도 비교를 통해 분석에 활용되는 중요 콘텐츠만을 자동 추출하는 기능을 제공한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존 웹 크롤러들의 기능과 문제점들을 설명하고 3장에서는 제안하는 분산형 웹 크롤러에 대해 기술한다. 4장에서는 기존 크롤러와 제안하는 크롤러의 성능 평가 결과를 통해 제안하는 크롤러의 우수성을 입증한다. 마지막으로 5장에서는 결론을 맺는다.

II. 관련연구

1. 기존 웹 크롤러들의 웹 문서 수집 기술

웹 크롤러는 웹 로봇, 웹 스파이더 등의 이름으로 불리기도 하며, 수많은 웹 문서를 자동으로 돌아다니며 각종 정보를 수집하는 프로그램이다. ‘크롤러’는 ‘기어 다니는 것’이라는 의미로, 수집된 웹 문서 내 포함된 URL(Uniform Resource Locator)을 기반으로 다른 웹

문서들을 돌아다니며 정보를 수집하는 기능 때문에 이런 이름이 붙었다[8]. 기존 웹 크롤러들은 일반형 웹 크롤러와 분산형 웹 크롤러로 구분할 수 있다.

[13],[14]와 같은 일반형 웹 크롤러는 단일 시스템에서 시드(Seed) 수집, 필터 그리고 수집 방법을 정하는 시드 스케줄링과 시드에 대한 웹 문서를 수집하는 크롤링 기능이 모두 수행된다. 일반형 웹 크롤러의 경우, 단일 시스템에서 시드를 관리하기 때문에 Overlapped Crawl 문제는 발생하지 않는다. 하지만, 방대한 웹 문서를 단일 시스템에서 크롤링 하기 때문에 많은 시간이 요구되는 문제를 갖는다.

[10]과 같은 분산형 크롤러는 크롤링 시간을 줄이기 위해 서버-클라이언트 환경으로 크롤링 한다. 서버는 초기 시드를 클라이언트에 분배하고 클라이언트가 수집한 웹 문서를 전달받는다. 클라이언트는 서버로부터 전달받은 시드를 기준으로 크롤링하고, 다음 크롤링을 위해 수집된 웹 문서에서 시드 추출 및 크롤링을 반복적으로 수행한다. 하지만, 분산형 웹 크롤러는 빠른 웹 문서 수집에 목적을 두고 있기 때문에 분산 환경에서의 네트워크 트래픽을 최소화하기 위해 각 클라이언트 간 수집한 웹 문서에 대한 정보를 실시간 공유하지 않는다. 그래서 [그림 3]과 같이 각 클라이언트는 다른 클라이언트가 수집한 웹 문서에 대한 정보 없이 자신이 수집한 웹 문서 내에 포함된 모든 URL을 시드로 사용하기 때문에 다른 클라이언트에서 수집된 문서를 재수집하는 Overlapped Crawl 문제가 발생한다.

그림 3. 기존 분산형 웹 크롤러의 Overlapped Crawl

기존 웹 크롤러는 주어진 시간 내 사용자가 요구하는

대부분의 웹 문서들은 정보 분석에 활용되는 중요 콘텐츠인 제목, 게시일, 작성자, 본문 외에도 메뉴, 광고, 댓글 등의 콘텐츠를 함께 포함하고 있다. 웹 문서를 기반으로 한 정보 분석 서비스의 품질을 향상시키기 위해서는 웹 문서로부터 메뉴, 광고와 같은 비 본문 영역 등을 제거하고 본문에 적합한 정보만을 정확하고 빠르게

추출하는 것이 중요하다.

웹 문서로부터 본문을 추출하기 위해 가장 많이 사용되는 방법은 DOM 트리를 기반으로 웹 문서의 구조적 특징을 이용하는 기법이 있다. 대표적인 기법으로는 VIPS[16]가 있다. [그림 6]과 같이 VIPS는 웹 문서를 DOM 트리로 표현하고, 트리 내 각 노드별로 가로, 세로 크기 및 배경 색상과 문서 내에서의 절대적 출현 위치 정보 등을 노드의 특징으로 구성 후, 노드들의 문서 내 출현 밀집도 등을 이용하여 그룹별로 본문 여부를 결정한다. 이 기법은 단순히 화면에 보이는 영역 정보만을 이용하거나 트리 구조로 표현되는 문서의 구조적 특징만을 이용해 본문을 추출하는 방법으로, 광고, 댓글 등과 같이 본문과 관계없는 내용들이 웹 문서에서 더 큰 영역을 차지하는 경우 본문을 제대로 추출하지 못하는 문제를 갖고 있다[16][17].

그림 6. VIPS의 본문 추출 과정

최근 VIPS의 문제를 해결하기 위해 HTML 문서로부터 텍스트 블록을 추출하고, 블록 내의 단어 및 하이퍼링크된 단어 수와 전후 블록의 정보 등을 웹 문서의 특징으로 추출하여 블록별로 본문 여부를 구분하는 Boilerplate detection 기법[18]이 제안되었다. 하지만, 뉴스와 같이 웹 문서 구조가 복잡하지 않은 웹 환경에서만 높은 추출 정확도를 가질 뿐, 문서 구조가 복잡한 웹 환경에서는 일부 본문이 추출되지 않거나 광고 및 댓글 등이 함께 추출되는 문제를 갖고 있어, 다양한 구조의 웹 문서를 대상으로 하는 빅데이터 분석 서비스에는 적합하지 않다.

III. 제안하는 지능형 웹 크롤러

1. RMI와 NIO 기반 분산형 웹 크롤러

제안하는 웹 크롤러는 대용량 웹 문서를 Overlapped Crawl 문제없이 빠르게 수집하는 방법을 제공한다. [그림 7]은 제안하는 분산형 웹 크롤러의 시스템 구성도를 보여준다. 제안하는 크롤러는 RMI를 기반으로 한 대의 서버와 다수의 클라이언트로 구성된 분산형 크롤링 환경을 제공한다.

그림 7. 제안하는 분산형 웹 크롤러의 시스템 구성도

프로토콜 스펙이 복잡해질수록 소켓(socket) 프로그래밍의 구현 난이도가 높아지고 이에 따른 통신상의 에러가 발생할 확률도 높아진다. 프로토콜 스펙을 지켜 구현하는 것은 난해하고 통신상의 에러가 빈번하게 발생하는 문제를 해결하기 위해, 로컬함수를 호출하는 것과 같은 방식으로 원격함수를 호출하는 방식인 RPC(Remote Procedure Call)의 개념이 도입되었다. RPC란 네트워크 프로토콜 스펙에 영향을 받지 않고 원격함수를 호출하는 방식으로 RPC 기법을 사용하면 통신상의 에러율이 감소하고, 네트워크 프로그램 구현이 용이해진다. RMI는 Java에서 제공되는 RPC 기법 중의 하나로 Java5부터 그 기능이 매우 안정화 및 경량화 되었고, 기존 소켓 방식처럼 서버와 클라이언트가 항상 네트워크로 연결되어 있을 필요가 없어 네트워크 활용률이 좋다. [19]의 결과에 따르면, 소켓 방식에 비해 RMI 네트

워크 성능이 평균 160% 개선된 것을 확인할 수 있다. 제안하는 분산형 웹 크롤러는 RMI를 기반으로 클라이언트-서버 환경을 구축했다.

제안하는 분산형 웹 크롤러 환경에서 서버는 활용 가능한 클라이언트에게 수집할 사이트의 RSS URL 또는 수집할 사이트의 메인 페이지 URL과 수집 기간(시작일과 종료일) 정보를 전달한다. 그리고 클라이언트가 수집한 웹 문서의 URL과 웹 문서를 저장·관리한다. RSS URL을 전달받은 클라이언트는 최근 게시된 웹 문서를 수집하는 크롤러로써, RSS 파일을 수집 후 해당 파일에서 수집할 URL을 추출해 시드로 사용한다. 사이트의 메인 페이지 URL과 수집 기간을 전달받은 클라이언트는 과거 게시된 웹 문서를 수집하는 크롤러로서, 전달받은 정보를 Google Search API에 입력 값으로 주고 획득한 결과 웹 문서에서 수집할 URL을 추출해 시드로 사용한다. Google Search API는 사이트 메인 페이지 URL과 수집 기간을 입력으로 주면, 수집 기간에 게시된 해당 사이트의 웹 문서 URL 목록을 결과로 제공하는 Google의 대표 서비스이다.

그림 8. 제안하는 서버 UI

[그림 8]은 제안하는 서버 UI를 보여준다. [그림 8]에서 ①은 제안하는 크롤러를 통해 웹 문서를 수집할 수 있는 웹 사이트 목록을 보여준다. 현재 과학기술 정보를 제공하는 110개의 사이트가 등록되어 있고, 사이트의 메인 페이지 URL과 RSS URL 정보 등록만을 통해 다른 사이트에 대한 등록이 용이하다. ②는 활용 가능

한 클라이언트의 정보를 보여준다. ③은 서버가 각 클라이언트에게 전달한 크롤링 정보를 보여주고, 클라이언트의 작업 수행을 직접 관리할 수 있는 기능을 제공한다. ④는 각 클라이언트가 수행한 크롤링 결과 정보를 보여주고, 해당 정보는 모두 로그파일로 기록된다.

클라이언트는 웹 문서를 수집할 때마다 해당 웹 문서의 URL을 서버의 데이터베이스에 직접 저장하고, 웹 문서를 수집하기 전에 수집할 웹 문서의 URL이 서버의 데이터베이스에 저장되어 있는지 확인한다. 이는 클라이언트 간 수집한 웹 문서의 정보 공유를 위한 통신비용 없이 Overlapped Crawl 문제를 해결하는 기능으로, 각각의 클라이언트가 직접 서버의 데이터베이스를 활용하기 때문에 서버가 클라이언트를 관리하는데 있어 부하는 증가하지 않는다. 또한, 클라이언트는 서버로부터 요청받은 모든 크롤링을 완료 후 수집된 모든 웹 문서를 Java의 NIO 기술을 활용해 서버로 일괄적으로 전송한다. [그림 9]와 같이 NIO는 커널 버퍼를 직접 사용하는 non-blocking 입출력 방식으로, 기존 소켓을 기반으로 한 blocking 입출력 방식에 비해 빠른 입출력 속도를 제공하고 blocking으로 인한 서버 부하도 줄일 수 있다.

그림 9. 기존 IO 방식과 NIO 방식의 차이점

표 1. 제안하는 NIO 기반 파일 전송 모듈

Algorithm 1: migration_File

```

Input : byte[] _srcFile
Output :: HTML file migrated from a client
01: FileOutputStream fos =
    new FileOutputStream(trgFile)
02: FileChannel fcs = fos.getChannel();
03: ByteBuffer buf = ByteBuffer.wrap(_srcFile)
04: fcs.write(buf);
05: fcs.close();
06: fos.close();
07: End
  
```

[표 1]은 클라이언트에서 서버로 수집된 웹 문서 전송을 위해 제안한 NIO 기반 파일 전송 모듈의 알고리즘

음을 보여준다. 제안하는 모듈은 RMI 환경에서 원격으로 호출되기 때문에 Java 원형 클래스 기반의 개체만을 매개변수로 전달해야 한다. 그래서 클라이언트에서는 수집된 웹 문서를 *byte* 클래스를 통해 전달하고, 서버에서는 *FileChannel* 클래스를 기반으로 커널 버퍼를 직접 사용해 NIO 기반의 고속 파일 입출력을 지원한다.

[그림 10]은 제안하는 클라이언트 UI를 보여준다. 클라이언트는 'Start' 버튼을 통해 서버와 초기 연결을 수행하고 'Crawling' 버튼을 통해 크롤링 작업을 시작한다. 그리고 'Migration' 버튼을 통해 수집된 웹 문서를 서버로 일괄적으로 전송한다. 마지막으로 텍스트 상자를 통해 클라이언트의 크롤링 상황을 실시간 제공한다.

그림 10. 제안하는 클라이언트 UI

2. 키워드 유사도 비교 기반 웹 콘텐츠 추출 기술

제안하는 분산형 웹 크롤러에서 클라이언트는 웹 문서로부터 정보 분석에 활용될 중요 콘텐츠인 제목, 작성자, 게시일, 본문을 자동으로 추출하는 기능을 내부적으로 제공한다. [그림 11]은 제안하는 웹 콘텐츠 추출기의 시스템 구성도를 보여준다. [그림 11]과 같이 제안하는 웹 콘텐츠 추출기는 중요 콘텐츠를 추출하는 규칙을 자동 생산해 중요 콘텐츠만을 추출하는 방법 및 장치로, 중요 콘텐츠 추출 규칙을 자동 생성하는 규칙 생성기(Rule Generator), 주어진 웹 문서에서 내비게이션 콘텐츠를 제거하는 내비게이션 콘텐츠 제거기(Navigation Content Eliminator) 그리고 중요 콘텐츠 추출 규칙 및 키워드 유사도 비교를 통해 중요 콘텐츠를 추출하는 중요 콘텐츠 추출기(Core Context Extractor)로 구성되어 있다.

그림 11. 제안하는 웹 콘텐츠 추출기의 시스템 구성도

규칙 생성기는 기존 웹 콘텐츠 추출에 사용된 웹 문서들에서 중요 콘텐츠 표현을 위해 사용되는 태그와 중요 속성을 기반으로, 중요 콘텐츠 표현을 위해 활용 가능한 태그와 중요 속성에 대한 집합을 자동 생성한다. 예를 들어, 사이트 A의 웹 문서에서 본문을 나타내는 태그가 `<div class="content-main"><P>`이고 사이트 B의 웹 문서에서 본문을 나타내는 태그가 `<div id="article-block"><P>`이면, 규칙 생성기는 태그명과 태그 속성의 조합을 통해 `<div class="content-main"><P>`, `<div id="article-block"><P>`, `<div class="article-block"><P>`, `<div id="content-main"><P>`와 같은 본문 추출을 위한 태그 집합을 자동 생성한다. 그리고 추출 규칙이 정해지지 않은 사이트의 웹 문서가 수집되면 규칙 생성기를 통해 만들어진 태그 집합을 활용해 중요 콘텐츠를 추출한다.

내비게이션 콘텐츠 제거기는 웹 문서에서 중요 콘텐츠 이외의 메뉴, 광고 등의 콘텐츠를 제거하는 모듈로 아래 규칙에 포함되는 태그와 콘텐츠를 제거한다.

- 규칙 1) 스크립트(script) 태그 제거 - 댓글 표현을 위해 사용
- 규칙 2) 내비게이션(nav, header, footer, ul) 태그 제거 - 메뉴 표현을 위해 사용
- 규칙 3) 80 단어 미만의 텍스트를 포함하고 있는 태그 제거 - 광고 표현을 위해 사용
- 규칙 4) 링크 태그 내 포함된 단어 수가 링크를 포함한 부모 태그 내 포함된 단어 수(링크 태그 내 포함된 단어 제외)보다 많은 경우 제거 - 광고 표현을 위해 사용

중요 콘텐츠 추출기는 내비게이션 콘텐츠 제거기를 통해 내비게이션 콘텐츠가 제거된 웹 문서에 대해 규칙 생성기를 통해 생성된 태그명과 태그 속성 집합을 기반으로 가장 유사도가 높은 태그를 찾아 콘텐츠를 추출한다. 예를 들어, 규칙 생성기를 통해 본문 추출을 위해 자동 생성된 태그 집합으로 <div class="content-main"><P>, <div id="article-block"><P>, <div class="article-block"><P>, <div id="content-main"><P>가 있고, 새로 수집된 웹 문서에서 본문을 추출하기 위한 정답 태그가 <div class="article-body"><P>라고 가정하면, 중요 콘텐츠 추출기는 <div id="article-block"><P>와 키워드 유사도가 80%인 <div class="article-body"><P>를 본문 태그로 구분하고 해당 태그의 콘텐츠를 추출한다. 물론, 키워드 유사도가 낮아 중요 콘텐츠를 추출하기 위한 태그가 잘못 선정될 수 있지만, 지속적인 사용자 보정과 추출 규칙 보강을 통해 제안하는 시스템의 정확도를 광범위한 웹 사이트를 대상으로 계속 향상시킬 수 있다.

IV. 성능 평가

[표 2]는 CNET[20], Infoworld[21], ComputerWorld[22] 웹 사이트에서 각각 2012년 게시된 임의의 웹 문서 30개와 2013년 1월부터 8월 사이에 게시된 임의의 웹 문서 30개를 선정 후, Google Search API를 통해 해당 웹 문서들의 URL이 검색 결과로 획득되는지를 실험한 결과를 보여준다. 이 실험 결과는 Google Search API를 통해 각 사이트의 과거 데이터를 효율적으로 수집할 수 있음을 보여준다. 하지만, Google Search API의 경우 각 웹 사이트에 최근 게시된 웹 문서의 URL을 제공하지 못하는 경우도 있는데, 이것을 보완하기 위해 제안하는 분산형 웹 크롤러는 각 웹 사이트의 RSS를 활용하여 최근 문서를 수집할 수 있다.

표 2. Google Search API의 URL 수집률

	CNET	Infoworld	ComputerWorld
2012년	100%	100%	100%
2013년	100%	100%	100%

제안하는 분산형 웹 크롤러의 우수성을 입증하기 위해, [10]에서 제안된 Parallel Crawlers와 성능 평가를 수행했다. 성능 평가는 4GB RAM과 Intel Core 1.7GHz로 구성된 Window 7 운영체제를 갖는 서버와 클라이언트에서 수행되었다. [그림 12]는 제안하는 크롤러의 클라이언트 수에 따라 CNET, Infoworld, ComputerWorld 사이트의 2012년도 전체 웹 문서를 수집하는데 소요된 시간을 보여준다. [그림 12]를 통해 클라이언트 수가 증가할수록 전체 웹 문서를 수집하는데 소요되는 시간이 감소하는 것을 확인 할 수 있었고, 소켓 방식의 분산 웹 크롤링 환경을 제공하는 [10]에 비해 RMI 방식의 분산 웹 크롤링 환경을 제공하는 제안하는 분산형 웹 크롤러의 성능이 우수한 것을 확인할 수 있었다. 또한, 제안하는 크롤러는 3개 사이트에 대해 총 2440건의 웹 문서를 수집했고, 수집된 모든 웹 문서가 해당 사이트의 웹 문서였다. 하지만, [10]의 경우 2850건의 웹 문서가 수집되었고 그중 1140건이 광고 또는 타 사이트의 웹 문서였다. 즉, 제안하는 분산형 웹 크롤러는 웹 문서 내 포함된 URL이 아닌 수집할 사이트의 RSS와 Google Search API를 통해 시드를 획득하기 때문에 정확한 웹 문서를 수집할 수 있었다.

그림 12. 클라이언트 수에 따른 크롤러 성능

V. 결론 및 향후 연구

본 논문에서는 특정 사이트의 전체 웹 문서를 효율적으로 수집할 수 있는 RMI와 NIO 기반 분산형 웹 크롤

러를 제안했다. 또한, 웹 문서의 중요 콘텐츠를 나타내는 태그에 대한 키워드 유사도 비교를 통해 중요 콘텐츠를 자동 추출하는 콘텐츠 추출기를 지원한다. 제안하는 분산형 웹 크롤러는 신뢰성 있는 웹 문서를 수집하는 기능을 제공함으로써, 다양한 빅데이터 분석 서비스의 확산을 가속화할 것으로 기대한다. 향후에는 제안하는 분산형 웹 크롤러에서 지원하는 110개 웹 사이트들에 대한 성능 평가를 수행함으로써, 제안하는 분산형 웹 크롤러의 신뢰성과 우수성을 다각적인 차원에서 입증할 계획이다.

참 고 문 헌

- [1] 성원경, 이상환, 정한민, 박경석, 이승우, 김선태, 황미녕, 조민희, *과학기술 빅데이터 추진과제 발굴 및 활용 극대화를 위한 추진전략 마련 기획연구*, 교육과학기술부, 2013.
- [2] 김정숙, “빅 데이터 활용과 관련기술 고찰”, 한국콘텐츠학회논문지, 제10권, 제1호, pp.34-40, 2012.
- [3] <http://www.1000genomes.org>
- [4] <http://www.smartinsight.co.kr>
- [5] <http://www.energy.gov>
- [6] M. K. Lee, S. W. Lee, J. Y. Kim, D. M. Seo, P. K. Ki, H. M. Jung, J. H. Lee, T. H. Kim, H. K. Koo, and W. K. Sung, “InSciTe Advanced: Strategic Decision-Making Support Service based on Technology Opportunity Discovery Model,” J. of Information An International Interdisciplinary, Vol.16, No.1(B), pp.639-698, 2013.
- [7] 이진희, 김태홍, 이승우, 김평, 이미경, 서동민, 김진형, 정한민, “정보 가치 향상을 위한 정보 분석 기술의 테크놀로지 인텔리전스 적용”, 정보과학회논문지 : 컴퓨팅의 실제 및 레터, 제18권, 제2호, pp.153-157, 2012.
- [8] http://en.wikipedia.org/wiki/Web_crawler
- [9] 김영광, 이원구, 윤화목, 신성호, 이민호, “웹 자원이 카이빙을 위한 웹 크롤러 연구 개발”, 한국콘텐츠학회논문지, 제11권, 제9호, pp.9-16, 2011.
- [10] J. H. Cho and H. G. Molina, “Parallel Crawlers,” In Proc. of International World Wide Web Conference, pp.124-135, 2002.
- [11] <http://en.wikipedia.org/wiki/RSS>
- [12] <https://developers.google.com/custom-search>
- [13] J. H. Cho, H. G. Molina, and L. Page, “Efficient Crawling through URL Ordering,” J. of Computer Networks and ISDN Systems, Vol.30, pp.161-171, 1998.
- [14] A. Heydon and M. Najork, “Mercator: A Scalable Extensible Web Crawler,” In Proc. of International World Wide Web Conference, pp.219-229, 1999.
- [15] 김진일, 권유진, 김성렬, 박근수, “그래프 탐색 기법을 이용한 효율적인 웹 크롤링 알고리즘”, 한국정보과학회 가을학술발표논문집, 제35권, 제2호, pp.299-300, 2008.
- [16] D. Cai, S. Yu, J. R. Wen, and W. Y. Ma, “VIPS: a Vision-based Page Segmentation Algorithm,” Microsoft Technical Report, 2003.
- [17] 송문원, 김우승, 김명원, “텍스트 블록 주변의 맥락을 이용한 HTML 문서 본문 추출”, 정보과학회논문지 : 소프트웨어 및 응용, 제40권, 제3호, pp.155-163, 2013.
- [18] C. Kohlschutter, P. Fankhauser, and W. Nejdl, “Boilerplate Detection using Shallow Text Features,” In Proc. of ACM International Conference on Web Search and Data Mining, pp.441-450, 2010.
- [19] 방승준, 안진호, “소켓 및 RMI 기반 자바 메시지 전달 시스템의 구현 및 성능평가”, 한국인터넷정보학회지, 제8권, 제5호, pp.11-20, 2007.
- [20] <http://www.cnet.com>
- [21] <http://www.infoworld.com>
- [22] <http://www.computerworld.com>

저 자 소 개

서 동 민(Dongmin Seo)

정회원

- 2002년 : 충북대학교 정보통신공학과(공학사)
- 2004년 : 충북대학교 정보통신공학과(공학석사)
- 2008년 : 충북대학교 정보통신공학과 공학박사)

▪ 2008년 ~ 2010년 : 한국과학기술원 전산학과 연수연구원

▪ 2010년 ~ 현재 : 한국과학기술정보연구원 컴퓨터지능연구실 선임연구원

<관심분야> : XML, 시맨틱웹, 이동객체 데이터베이스, 센서 네트워크

정 한 민(Hanmin Jung)

정회원

- 1992년 : 포항공과대학교 전자계산학과(공학사)
- 1994년 : 포항공과대학교 전자계산학과(공학석사)
- 2003년 : 포항공과대학교 컴퓨터공학과(공학박사)

▪ 1994년 ~ 2000년 : 한국전자통신연구원 선임연구원

▪ 2000년 ~ 2004년 : (주)다이렉스트 연구소장/기술이사

▪ 2004년 ~ 현재 : 한국과학기술정보연구원 컴퓨터지능연구실 실장

<관심분야> : 시맨틱 웹, 정보검색, 자연어처리, HCI