분산 환경에서의 효율적인 E-Book 변환을 위한 작업 배분

안재호, 황동엽, 강민지, 최광훈, 김재훈
아주대학교 정보컴퓨터공학과

e-mail : ajh0121, bc8c, sbje, saarc, jaikim@ajou.ac.kr

Job Scheduling for Efficient E-Book Conversion in Load Balancing Systems

Jae-Ho An, Dong-Yeop Hwang, Min-Ji Kang, Kwang-Hoon Choi, Jai-Hoon Kim[1]
Dept. of Information Computer Science, Ajou University
[1]Prof. of Information Computer Science, Ajou University

요 약

전자책(E-Book)에 대한 수요가 커짐에 따라서 전자책 시장이 점점 커지고 있다. 이에 PDF 와 같은 다른 형태의 문서들을 전자책으로 변환하는 프로그램 및 서비스들에 대한 요구가 늘어가고 있다. 전자책의 공급 규모가 커지고 형식이 발전함에 따라서 대규모의 전자책들을 빠르고 효율적으로 변환 가능하게 하는 환경의 조성이 필요하게 되었다. 기존 시장에 배포되거나 출판된 PDF 형식의 문서를 오픈소스 변환 라이브러리를 이용하여 변환할 수 있는 변환기를 작성하고, 이를 이용해 대규모 PDF 를 저장하고 있는 분산 저장 시스템에서 백그라운드 배치 작업으로 변환할 수 있는 구조를 설계 및 제안한다. 본 논문에서는 전자책의 효율적인 변환을 위한 분산 환경에서의 작업 배분 방법을 다룬다.

Keywords : 전자책(E-Book), 작업 배분(Job Scheduling), Hadoop, Load Balancing

1. 서론

최근 해외에서는 전자책(E-Book)에 대한 수요가 커짐에 따라서 전자책 시장이 점점 활발해지면서 Apple, Amazon 과 같은 대형 기업들이 IBooks, Kindle 과 같은 전자책 뷰어(E-Book Viewer)를 제공하여 대량의 전자책을 공급하고 있다. 이러한 전자책들은 저자가 PDF, IMAGE 혹은 TXT 파일로 제작하여 출판사에 전달해주면 EPUB 형식으로 변환하여 소비자들에게 제공된다. 그러나 국내에서는 전자책에 대한 수요가 적어 대형 출판사에서 조차 전자책 출판에 대하여 소극적이다. 이렇다 보니 국내에서는 전자책 표준 형식인 EPUB 형식으로 변환하는 자체 솔루션이 많지 않으며, 한국어(Unicode)를 제대로 지원하는 솔루션은 거의 없다. 또한 Viewer 자체에 포함되어있지 않은 폰트에 대한 mapping 을 해결해주는 솔루션 또한 없다. 따라서, 다른 형식보다 활용도가 높은 PDF 를 EPUB 으로 변환하고 한국어를 완벽하게 지원하며 원래의 폰트를 최대한 살리는 솔루션 구현이 필요하다.

출판사는 전자책을 공급하기 위해 PDF, IMAGE, TXT 등의 파일을 EPUB 으로 변환하는 작업이 필요하다. 그러나 공급량이 커짐에 따라서 출판사에서 변환해야 하는 용량이 매우 커져 단일 컴퓨팅 시스템으로는 시스템 부하가 커질 뿐만 아니라 작업 속도도 매우 느려지게 되는 문제가 생긴다. 이를 분산 환경을 구축하여 효율적으로 변환작업을 해야 할 필요성이 생기게 되었다.

따라서, 본 논문에서는 한국어를 완벽하게 지원하고 원래의 폰트를 최대한 살리는 EPUB 변환 솔루션과 이 작업을 효율적으로 처리할 수 있는 분산 환경의 구조와 분산 처리 방법을 제안한다.

2. 관련 연구.

오늘날에는 정보를 전달하기 위해 문서를 출력하여 직접 만나서 전달하기 보다는 온라인을 통하여 문서를 교환하는 일이 주를 이루게 되었다. 그러나 각 회사마다(혹은 개인) 다른 포맷의 문서를 사용하여 전달받은 문서를 열람하기 위해 다른 문서 뷰어를 구매하거나 설치해야 하는 불편함이 생겼다. 이를 해결하기 위하여 많은 회사나 연구소에서 문서 표준화 연구가 진행되었다[1]. 또한 다른 기종의 문서 형식의 한계를 극복하기 위해 문서의 포맷을 변환해주는 온라인 서비스인 Cometdocs[2]같은 웹사이트가 생겨났고, Microsoft 에서는 hwp 포맷을 Microsoft Word 에서 열람 및 편집이 가능하게 하는 아래아한글 문서 변환 도구[3]를 제공한다. 혹은 각 문서들을 인터넷 상에서 쉽게 전달하고 볼 수 있도록 XML 로 변환하는 연구[4]가 진행되었다. 이 외에 공통된 포맷으로 변환은 불

가능하고 열람만 가능하도록 PDF 포맷으로 문서를 변환하여 전달하기도 한다. 각 문서 편집기마다 PDF 포맷 변환 기능을 지원하는 경우가 많으며, 그렇지 않더라도 Cometdocs 같은 서비스를 이용하여 PDF 로 포맷을 변환하는 것이 용이해졌다. 본 논문에서는 이렇게 PDF 로 작성된 문서를 스마트폰이나 웹에서 보기 용이하도록 EPUB 형태의 전자책으로 변환하는 작업에 중점을 두어 연구를 진행하였다..

PDF 내부의 컴포넌트들을 추출해내기 위해서는 구조 분석을 지원하는 라이브러리를 사용할 수 있다. 이를 지원하는 라이브러리에는 Adobe 사의 공식 유료 라이브러리인 Adobe Acrobat Pro Extended[6], Java 기반의 무료 오픈 소스인 Apache PDFBox[7], PHP 기반의 무료 오픈 소스인 FPDF[8], ActionScript3 기반의 무료 오픈 소스인 AlivePDF[9], .NET 언어를 지원하는 .NET 기반의 무료 오픈 소스인 PDFsharp[10] 등이 존재한다. 본 논문에서는 플랫폼에 구애 받지 않도록 JVM 기반의 Java 언어를 사용하여 구현하기 위하여 Apache PDFBox 라이브러리를 사용하였다.

PDF 나 IMAGE 기반으로 작성된 전자책은 어플리케이션에서 글자 검색기능을 제공할 수 없다. 그러나 일부 사용자들은 전자책을 읽으면서 특정 키워드가 들어가는 문장 혹은 문단을 찾기를 원한다. 이를 위해 기존에는 IMAGE-based 전자책을 OCR(Optical Character Recognition) 방법을 이용하여 IMAGE 에서 WORD 를 추출하여 XML-based EPUB 형태의 전자책으로 변환하는 연구가 진행되었다. 이러한 변환을 Hadoop Distributed File System(HDFS)에서 여러 개의 cluster 를 이용하여 처리할 때의 가장 최적화 된 cluster 수를 비교, 분석하는 연구도 동시에 진행되었다[5]. 그러나 기존의 전자책 변환 연구에는 한국어(Unicode) 지원이나 원본 그대로의 폰트를 mapping 하는 부분에 대해서 연구한 부분은 미약하다.

분산 시스템 환경에서 동적으로 작업을 분배하는 방법에는 일반적으로 두 가지로 나누어진다. 작업을 실행하기 전에 전체 시스템에서 작업에 가장 적절한 노드를 찾아 할당하는 초기 작업 배치방식과 부하의 불균형이 발생했을 때, 실행중인 작업을 부하가 작은 노드로 이동하는 프로세스 이동 방식으로 나누어진다. 일반적으로는 초기 작업 배치방식의 성능이 전체 시스템 성능에 큰 영향을 주는 것으로 알려져 있다. 초기 작업 배치 시, 이전에는 작업의 자원 요구 형태에 대한 선행 지식을 필요로 하는데 이 선행 지식의 정확도가 떨어지면 성능이 떨어지기 때문에 선행 지식을 필요로 하지 않고 작업이 잘못 배치되는 상황을 방지하는 유효 작업수를 이용한 동적 부하 분산 시스템이 연구되기도 하였다.[11]

3. 전자책 변환 구현

본 논문에서는 PDF 형식의 파일을 EPUB 형식의 전자책으로 변환하는 것을 다룬다. 기존의 Amazon Kindle 과 같은 전자책은 PDF 각 페이지를 IMAGE 로 변환하여 EPUB 포맷에 맞추어 만들어진다[12]. 그러나 이러한 방식으로 만들어진 전자책은 페이지 전체가 이미지로만 이루어져 있다는 특성 때문에 원본을 그대로 살릴 수 있다는 장점은 있지만, 책 내의 특정 단어나 문장 검색 기능을 지원하지 못한다. 문자를 식별 가능한 전자책을 구현하기 위해서는 PDF 자체를 IMAGE 로 만들어서 전자책을 만드는 기존의 방식과 다르게 PDF 내부의 컴포넌트들을 모두 추출하여 원본의 레이아웃에 맞추어 각 page 별로 XHTML(XML-based) 형태로 그대로 재배열하여 이들을 EPUB 전자책으로 변환하는 방법을 이용한다.

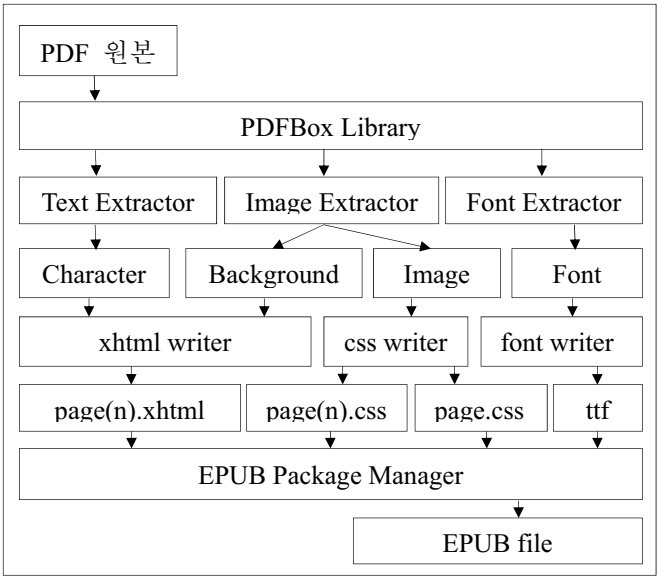

그림 1. PDF 를 EPUB 으로 변환하는 구조

그림 1 은 PDF 를 EPUB 형태로 변환하는 프로그램의 구조 설계도이다. 입력으로 PDF 파일이 받아들여지면 PDFBox 라이브러리에서 Document 로 변환하게 된다. Document 는 변환 과정에서 각 page 로 나눠진다. Text Extractor, Image Extractor, Font Extractor 에 의해서 각 page 의 Text, Background, Image, Font 가 추출되고, 이 각각의 컴포넌트들은 xhtml writer, css writer, font writer 에 의해서 xhtml, css, ttf 파일로 변환된다. 각 page 가 page(n).xhtml 로 생성된다. 전체 Document 에 존재하는 font 들을 page.css 파일에서 전부 맵핑하고 있으며, 각 page 별로 Image, Background Image, Character 들을 page(n).css 파일에서 위치나 속성 등의 정보를 제공한다. 이렇게 만들어진 xhtml, css, ttf 파일들을 EPUB Package Manager 가 EPUB 형식의 전자책으로 변환해준다.

아래의 그림 2 는 변환기의 실행화면이다. 변환기를 실행하면 왼쪽에 탐색기에서 변환할 파일을 찾거나 추가 버튼을 눌러 파일을 선택할 수 있다. 가운데 화면에는 선택한 파일의 목록이 출력되고 잘못 선택한 파일이 있다면 목록에서 삭제할 수 있다. 변환 버튼을 누르면 아래 화면에 진행 상태가 출력되고 오른쪽 칸에 완료된 작업의 목록이 출력된다. 변환한 파일은 Administrator 폴더에 저장된다. 그림 3 은 변환기로 PDF 파일을 EPUB 으로 변환한 결과이다.

다음은 변환기를 batch 작업으로 실행하는 커맨드라인이다. 변환 작업을 GUI 프로그램으로만 실행하지 않고 batch 작업으로도 가능하게 하기 위하여 exe 실행 파일을 만들었다. Parameter(인자)로 변환할 파일의 위치를 입력하면 그 위치의 PDF 파일을 변환한다. 저장 위치는 GUI 프로그램과 동일하다. 변환기와 PDF 파일은 편의를 위해 C 드라이브 root 에 들어있다.

C://Converting PDF to E-BOOK.exe C://Filename.pdf

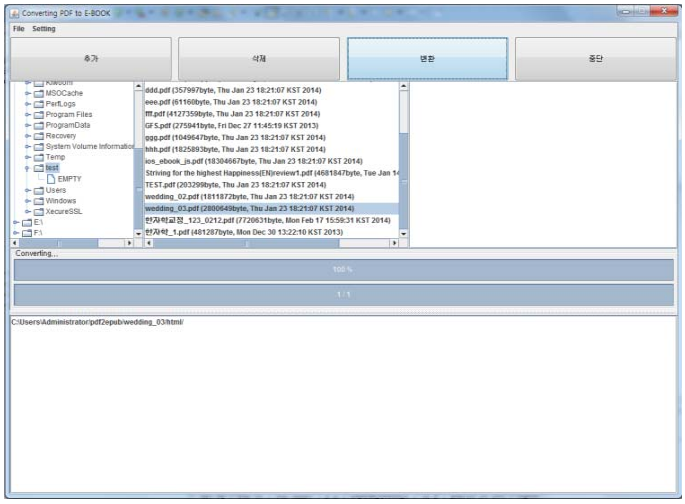

그림 2. 변환기 실행 화면

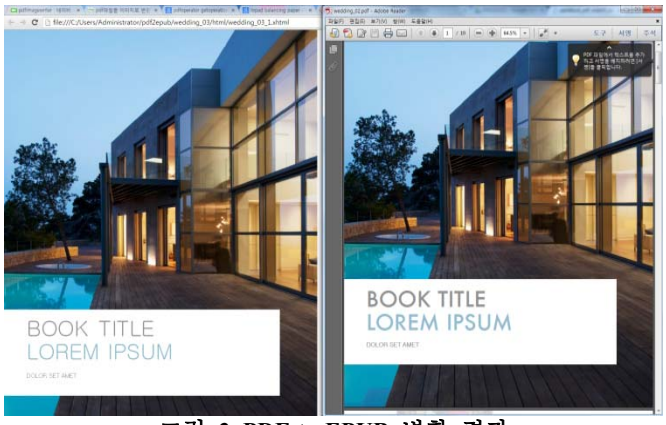

그림 3. PDF-to-EPUB 변환 결과
(왼쪽 : 변환 후, 오른쪽 : 변환 전)

4 - 1. 전자책 변환 분산 시스템

전자책은 내부에 텍스트만 들어있는 것이 아니라 이미지나 효과 등이 들어가있기 때문에 용량이 클 수 있고 원본에 포함된 내용도 용량이 매우 클 수 있다. 이러한 전자책이 출판시장이 커짐에 따라서 변환 요구되는 용량이 TB 급을 넘어감에 따라서 이를 한 컴퓨터에서만 처리하기에는 시간이 매우 오래 걸린다. 아래의 그림은 각각 다른 페이지 수의 PDF 를 변환했을 때 걸리는 시간을 그래프화 한 것이다.

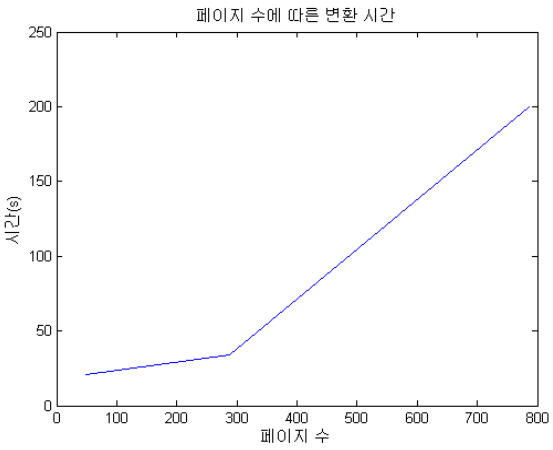

그림 4. 페이지 수에 따른 변환 수행 시간

그림 4 를 보면 페이지 수가 많아질수록 걸리는 시간이 일정하게 증가하는 것이 아니라 급격하게 증가한다. 287 페이지에서 약 30 초 걸리던 작업이 약 3 배 분량인 786 페이지에서는 200 초가 소요되었다. 출판사에서 변환하는 전자책이 한 권당 300 페이지 정도 된다고 가정했을 때, 100 일 경우, 한 시간 이상이 걸린다. 거기다 원본 PDF 에 이미지가 많거나 특이 속성이 많이 포함되어 있다면 걸리는 시간은 더욱 길어진다. 이처럼 많은 양의 변환 작업은 시간도 오래 걸리고, 부하도 많이 걸려서 분산하여 처리하는 시스템의 필요성이 대두되었다. 본 논문에서는 분산 저장 구조 시스템인 Apache HDFS(Hadoop Distributed File System)[13]를 이용하여 구축할 예정인 분산 처리 환경을 제시한다.

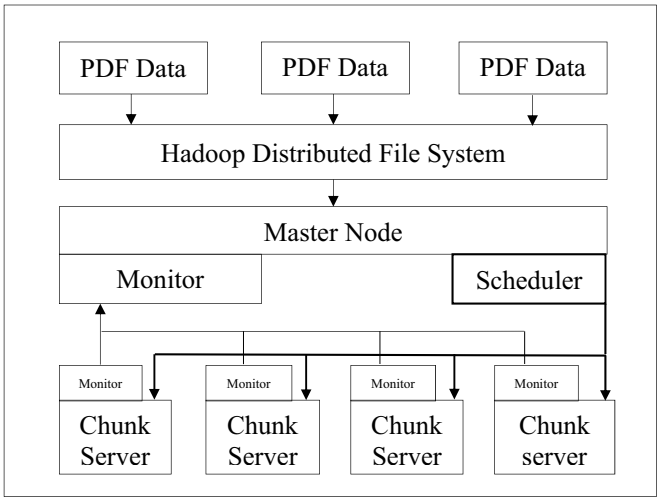

그림 5. 전자책 변환 분산 시스템

4 - 2. Load Balancing

대용량의 PDF 파일 변환을 프로그램에 요청한다. Master Node 에서는 이 요청 받은 파일들을 각 Chunk Server 에 배분하여 작업을 실행하도록 명령한다. Master Node 는 주기적으로 각 Chunk Server 의 자원 사용률을 확인한다. CPU 사용률, Memory 사용률 등 변환 작업에 영향을 미치는 자원의 사용률을 주기적으로 확인하고 일정 이하의 자원이 사용되고 있는

Chunk Server 에 작업을 배분한다. 만약 특정 Chunk Server 의 자원 사용률이 일정 이상을 넘어간다면 Master Node 에서는 더 이상 해당 node 에는 작업을 할당하지 않는다.

위와 같은 변환 분산작업을 통하여 현재 수행중인 다른 작업에 간섭을 최소화하여 백그라운드에서 효율적으로 변환작업을 할 수 있을 것이라 예상된다.

5. 결론 및 향후 연구 과제

국내의 전자책 시장에서 유통되는 전자책은 대부분이 원본의 폰트를 완벽하게 살리지 못하고 한국어를 완벽하게 지원하지 못한다. 또한 전자책을 IMAGE 를 이용하여 만들었을 경우 글자 검색 기능을 제공하지 못한다는 단점이 있다. 이를 해결하기 위해 원본의 폰트를 그대로 살리는 mapping 방법을 적용하고 유니코드를 지원하여 한글과 한자를 정확하게 변환하여 주며 글자 검색 기능을 지원하는 변환기를 구현하였다.

또한 본 논문에서는 유니코드를 지원하여 한글과 한자를 정확하게 변환하는 모듈을 기반으로 시스템 백그라운드에서 효율적인 일괄작업이 가능한 분산시스템을 설계, 제안하였다.

향후 연구과제로는 4-1, 4-2 에서 제안한 대용량의 전자책을 변환하기 위한 분산 시스템 환경을 실제로 구축하고 변환 속도를 개선하는 과정이 필요하다. 또한 변환기 프로그램 자체의 프로시저를 개선하여 변환기 자체의 속도를 향상시키는 방법도 연구할 계획이다.

참고문헌

[1] IDPF, “EPUB 3”, Retrieved 8 Dec 2012, http://idpf.org/epub/30, 14 March 2014
[2] Online Document Management System http://www.cometdocs.com
[3] Microsoft Word 를 위한 아래아한글 문서 변환 도구 http://microsoft.com/ko-kr/download/detail.aspx?id=36772
[4] 주원균, 양명석, 김태현, 이민호, 최기석 “전자문서의 XML 문서로의 변환 및 저장 시스템”, 한국컴퓨터종합학술대회 논문집, vol. 33, no.1(c), (2006), pp. 106-108
[5] Tae Ho Hong, Chang Ho Yun, Jong Won Park, Hak Geon Lee, Hae Sun Jung and Yong Woo Lee “Big Data Processing with MapReduce for E-Book”, International Journal of Multimedia and Ubiquitous Engineering, vol. 8, no. 1, (2013) January, pp. 151-162
[6] Adobe Acrobat Pro Extended 9 of Adobe Systems http://www.adobe.com/kr/products/acrobatpro.html
[7] PDFBox of the Apache Software Foundation http://pdfbox.apache.org/
[8] PDF generator FPDF Library http://www.fpdf.org
[9] PDF generation library for Adobe Flash http://code.google.com/p/alivepdf
[10] .NET library for generating PDF http://sourceforge.net/projects/pdfsharp
[11] 최민, 김남기 “동적 부하 분산 시스템에서 효율적인 작업 크기 계산을 통한 성능 개선”, 정보처리학회논문지 A, 제 14-A 권, 제 6 호(2007.12), pp. 357-362
[12] Amazon Kindle Read, Review, Remember https://kindle.amazon.com
[13] Apache Hadoop Distributed File System http://hadoop.apache.org