Postprint: Building a Full-Text Indexing System for WARC Documents

Authors: Hu Jiying, Wu Zhenxin, Xie Jing, Zhang Zhixiong

Date: 2017-10-11T00:00:00+00:00

Abstract

【目的】Objective: To develop a parsing and indexing system for Web ARChive (WARC) files, fully exploiting the value of archived resources from scientific websites.

【应用背景】Application Background: In the field of web resource collection and archiving, the WARC file format has been widely adopted. With the diversification of web information, existing WARC file indexing tools are increasingly unable to meet users' diverse query requirements.

【方法】Method: A modular approach is adopted to parse WARC files. Commonly used indexing tools are analyzed and compared, and the Solr platform is selected to develop a full-text indexing system.

【结果】Results: Content-based retrieval and access services for WARC files are implemented, and faceted search content such as discipline classification, resource type, and archiving time is added to the WARC index, revealing WARC file content from multiple dimensions.

【结论】Conclusion: Provides users with rich archived data and information from scientific websites, improving retrieval and access efficiency.

Full Text

Preamble

ChinaXiv Partner Journal, Issue 270, 2016, No. 5

Building a Full-Text Indexing System for WARC Documents

Hu Jiying, Wu Zhenxin, Xie Jing, Zhang Zhixiong
(National Science Library, Chinese Academy of Sciences, Beijing 100190, China)

Abstract

[**Objective**] This paper develops a parsing and indexing system for WARC (Web ARChive) files to fully exploit the value of archived scientific website resources. [**Context**] The WARC file format has been widely adopted in web resource harvesting and archiving. However, as online information becomes increasingly diverse, existing WARC indexing tools struggle to meet users' varied search requirements. [**Methods**] We employed a modular approach to parse WARC files and, after analyzing and comparing common indexing tools, selected the Solr platform to develop a full-text indexing system. [**Results**] The system enables content-based retrieval and access services for WARC files and enhances the index with faceted search capabilities including subject classification, resource type, and archival time, revealing WARC file content from multiple dimensions. [**Conclusions**] The system provides users with rich archived data from scientific websites and significantly improves retrieval efficiency.

Keywords: Web archive; WARC file; Modular parsing; Solr indexing

Introduction

As the Internet deeply permeates social life, the pace of online information updates continues to accelerate, making the lifecycle of web resources increasingly ephemeral. Harvesting and preserving valuable web resources to meet current and future access needs has become a critical mission for preservation institutions worldwide. Archiving important online resources, particularly scientific and commercial information, has risen to the level of national strategic importance.

The International Internet Preservation Consortium (IIPC) [1] has been dedicated to promoting global web archiving activities since its establishment. With members including over 40 national libraries and archives from around the world, IIPC plays a crucial role in the field of web information preservation. To better support harvesting, access, and exchange among archiving institutions, IIPC proposed the standardized WARC file format, which became an ISO international standard in 2009 [2]. The WARC format offers large information capacity, supports extensibility and compression, and is easy to manage, making it widely adopted in numerous Web Archive projects [3].

As more research institutions participate in preservation efforts and the volume of archived resources grows rapidly, the utilization and sharing of archived resources have become increasingly important aspects of Web Archive initiatives. Key challenges facing archiving projects include enhancing user experience to meet diverse query needs and leveraging advanced technologies to effectively reuse archived resources to maximize their value. Improvements in WARC file parsing and indexing technologies have become critical to addressing these challenges. This paper details how the Web Archive team at the National Science Library, Chinese Academy of Sciences, has effectively implemented WARC parsing and built a full-text indexing system in its practice of archiving network

resources from major international scientific institutions.

2. Current Status of WARC Indexing Tools and Requirements Analysis

Web archive content involves not only massive data volumes but also nonlinear dynamic growth over time. To enable retrieval and access of WARC archived files, parsing and indexing WARC files is essential. The most commonly used open-source tools include Wayback [4], NutchWAX [5], and WERA [6].

Wayback, provided by the Internet Archive (IA), is a URL-based indexing and access tool that can replay archived pages from different points on a timeline, but it does not support full-text content retrieval, making it difficult to satisfy users' diverse search needs. NutchWAX [7] enables full-text indexing and retrieval based on content and has been applied in many national web archiving projects. However, the Nutch development team has shifted its strategic focus [8] and rarely provides subsequent maintenance and updates for NutchWAX, concentrating instead on web crawler development. Additionally, NutchWAX relies on the Hadoop file system to build indexes and cannot create indexes locally; functional expansion is also difficult, requiring extensive code modifications. Consequently, NutchWAX is no longer a suitable platform for providing full-text retrieval.

WERA is another commonly used web archive content query and browsing tool, jointly developed by IA and the Norwegian National Library. It provides both URL-based retrieval similar to Wayback and full-text search capabilities. However, since its developers adopted NutchWAX as the search engine kernel, WERA has not fundamentally resolved the limitations faced by NutchWAX. Solr [9] is a widely used open-source enterprise search platform with full-text indexing and faceted search capabilities. After comparing the performance of NutchWAX and Solr, IA found that NutchWAX' s retrieval performance was significantly lower than Solr' s when the number of preserved documents increased dramatically, and thus recommended adopting Solr as the future platform for full-text retrieval. Currently, the British Library, the Netherlands Institute for Sound and Vision, and IA have begun exploring the use of Solr to address full-text retrieval in WARC archiving projects.

In implementing its project to archive network information from major international scientific research institutions [10], the National Science Library, Chinese Academy of Sciences, needed to build a Web Archive access service platform to provide long-term and effective support for researchers, intelligence analysts, and science and technology managers to utilize archived resources. This required not only basic access services for archived resources but also support for users to conduct research and analysis of web-based scientific information along the temporal dimension, as well as provision of historical data support for data mining and reuse. Implementing these functions requires robust support from the underlying indexing system, yet existing WARC indexing tools struggle to

meet these requirements. Therefore, during project development, we explored building a full-text indexing system for WARC files using the Solr platform to satisfy user needs.

In the field of long-term preservation and retrieval of network resources, research related to Solr is still in its early stages, with many issues requiring resolution, particularly the fact that Solr cannot directly index WARC files. To address these problems and extract more information from WARC files, this paper leverages the structured characteristics of WARC files, adopts a modular scheme to implement WARC file parsing, extracts index field content from parsing results, and utilizes the Solr platform to build a full-text indexing system for WARC documents.

3.1. WARC File Structure Example and Storage Scheme

WARC can safely carry large volumes of data objects in a single file, providing a mechanism to concatenate multiple resource records (WARC Records) into a single long file (WARC file), storing "web-crawled content" in the order that content blocks are captured from the Internet. To facilitate data preservation and sharing, WARC format files consist of one or more WARC records simply concatenated together, with the first record typically describing subsequent records. Using the harvesting of a scientific website (http://www.iahe.org/) as an example, Figure 1 [Figure 1: see original paper] shows the information contained in the generated WARC file.

In general, WARC record content is either the direct result of a retrieval (web pages, embedded images, URL redirection information, DNS hostname query results, standalone files, etc.) or comprehensive resources that provide additional information for archived content (such as metadata, transformed content). Each WARC Record consists of a simple text header and an arbitrary data content block. The first line of the WARC record header declares that the record uses a given version of the WARC format, followed by a variable number of named fields terminated by a blank line. This structure exhibits hierarchical and structured characteristics.

Due to the large volume of website content, harvesting website information generates numerous WARC files. The storage scheme and delivery method depend on software and application implementation. This paper uses Heritrix for site harvesting, and the harvesting results can be stored across multiple WARC files. The maximum size of each WARC file can be configured and managed through Heritrix' s order.xml configuration file by setting the max-size-bytes parameter in the component. Heritrix automatically numbers the WARC files generated in a single harvesting task. For example, if the maximum WARC file size is set to 1GB during harvesting, only one WARC file is generated when the data collected from a site is less than 1GB, with the filename containing sequence number 0000. When the size exceeds 1GB, multiple WARC files are stored sequentially using 0001, 0002, 0003, etc., as shown in Figure 2 [Figure 2:

see original paper].

3.2. Modular Parsing of WARC Files

Based on the structured characteristics of WARC files, this paper adopts a hierarchical parsing approach with modular parsing functions when designing the WARC file parsing scheme. The WARC file parsing process is shown in Figure 3 [Figure 3: see original paper] and can be divided into four functional modules: WARC Record acquisition module, header acquisition module, WARC Record content block acquisition module, and content block parsing module.

(1) WARC Record Acquisition Module
A WARC file consists of multiple resource records (WARC Records) concatenated in sequence. The WARC Record acquisition module splits the WARC file into multiple WARC Records, transforming the task from parsing WARC file content to parsing WARC Record content. In addition to core web harvesting toolkits, Heritrix source code also provides an IO operation package for the WARC format: org.archive.io.warc. This paper utilizes core classes from this package to read WARC files, including WARCReader.java, WARCRecord.java, and WARCReaderFactory.java. During WARC document parsing, the get(String warcFilePath) function in the WARCReaderFactory class is called iteratively to obtain WARC Reader objects, and traversing the entire WARC document yields each WARC Record.

(2) WARC Record Header Acquisition Module
Each WARC Record has a consistent structure, comprising a set of WARC record headers and a content block. After obtaining a WARC Record, the getHeader() function in the WARC Record class completes parsing of the header information, extracting the named field information from the WARC Record header, which is stored in a Map data structure. Table 1 shows the parsing results for a WARC Record header.

(3) WARC Record Content Block Acquisition Module
Similarly, after obtaining a WARC Record, the Warcrecord.read() function reads the WARC Record content and stores it as a byte array.

(4) WARC Record Content Block Parsing Module
While the previous modules use core classes from Heritrix's WARC format IO operation package to obtain WARC Record header fields and content blocks, parsing of the WARC Record content block is completed through a self-developed parseContent(byte[] content) module. The WARC Record content block consists of an HTTP protocol header, two blank lines, and resource content. First, the HTTP protocol header portion is extracted from the WARC Record by blank line separation and parsed to obtain the data information it contains. The primary purpose of this step is to determine the Content-Type field content through HTTP protocol header parsing, which identifies the resource type of the archived page. Different parsing tools are then invoked based on resource type: PDFBox for PDF resources, POI for Word, PPT, and Excel resources,

HTMLParser for HTML resources, etc. After obtaining the title and content of the archived page, the information is saved in a Map data structure. Ultimately, the Map stores the WARC Record' s header fields, title, and content, completing the full parsing of a single WARC Record.

Due to the diversity and complexity of web resource content, certain special resources may be unparsable or require excessive parsing time. To ensure WARC file parsing efficiency, a daemon thread is defined to execute WARC file parsing, with execution time limits set to guarantee smooth parsing progress. This modular parsing approach simplifies WARC file parsing implementation, enabling WARC file content to be divided into independent components based on WARC structure. This facilitates content indexing construction and provides convenient, easy-to-use interfaces for future data mining, metadata extraction, and data analysis services.

4. Building a Full-Text Index for WARC Files Based on the Solr Platform

A Web Archive system must not only achieve long-term preservation of WARC files but also establish effective indexes for parsed WARC content to provide users with efficient retrieval and access functions. Considering Solr' s powerful advantages in full-text indexing and faceted search, this paper utilizes Solr tools to develop a WARC file content indexing module to optimize user retrieval and access experience.

4.1. Automated Indexing System Workflow Design

To achieve automated real-time indexing of WARC files, a monitoring module was developed to detect new WARC file generation in real time. When new WARC files are generated, they are automatically added to the indexing queue to await indexing by the indexing module, ensuring real-time updates to WARC file indexes and searchable content. Figure 4 [Figure 4: see original paper] illustrates the automated indexing system design. The first step involves integrating the WARC generation system, indexing module, and WARC file parsing module to achieve automated continuous indexing of WARC files. The harvesting system built by the National Science Library, Chinese Academy of Sciences, performs periodic site harvesting. After each harvesting task completes, the generated WARC files are automatically uploaded to a storage array for unified storage. To enable real-time indexing of newly generated WARC files, the monitoring module periodically polls the storage array to check for new WARC files. When new files are detected, they are added to the indexing queue to await indexing. The indexing module retrieves indexing tasks from the queue and calls the WARC acquisition and parsing modules to perform hierarchical reading and parsing of WARC files, with parsing results stored in a Map data structure. The indexing module then retrieves values for index fields from the Map, including the archived page' s URL, archival time, resource type, title,

page content, etc.

4.2. Index Field Design

In addition to fields obtained from WARC parsing, the index field design includes auxiliary extension fields from the database, such as subject classification, source site name, and site type. Each index corresponds to one WARC Record in a WARC file, representing information for one archived page. The detailed index field design is shown in Table 2 .

4.3. Indexing Strategy Design

To improve user retrieval efficiency and better reveal archived WARC file content, two different indexing strategies were designed: comprehensive WARC record indexing and archived page indexing. Although each record in both strategies represents an archived URL, the first strategy uses WARC_Record_ID as the unique identifier, with each index representing a complete WARC Record. When building indexes, WARC parsing fields and database-read fields are sequentially written to corresponding index fields.

The second strategy uses the archived page' s URL as the unique identifier and adds a new datelist index field. This multi-value field stores different archival times for the archived page, representing the number of times the page has been archived and each archival timestamp. When building an index for each WARC Record, the system first queries whether the URL record exists in the index. If not, WARC parsing fields and database-read fields are sequentially written to corresponding index fields. If the URL already exists in the index, indicating the URL has been archived before, the current archival time is merged with previous archival times and rewritten to the datelist field, while other field contents are updated to the latest archived page content.

The advantage of these two indexing strategies is that they satisfy users' diverse query needs. The index using WARC_Record_ID as the unique identifier provides complete information for each harvesting of the archived URL, while the index using archived page URL as the unique identifier merges multiple harvesting information for the URL, clearly presenting the complete harvesting history of each URL. This significantly reduces index volume, improves retrieval speed, and the two indexes can support and complement each other.

5. Application Effect Analysis of the Web Archive Access Platform

The National Science Library, Chinese Academy of Sciences, has essentially completed development of a Web Archive access platform for important research institutions, implementing modular parsing of WARC files and building a full-text indexing system based on content. Compared with commonly used open-source indexing tools, this platform offers more comprehensive access and retrieval

functions, displaying archived resources to users from multiple perspectives, as shown in Table 3 .

As of January 2016, the Web Archive platform had collected and archived a total data volume of 26TB (compressed package size), comprising over 20,000 WARC files with an index volume exceeding 500GB. The platform provides URL and full-text retrieval while adding faceted content such as subject classification, resource type, and archival time, revealing WARC file content from multiple dimensions. Users can perform content retrieval by entering keywords through the homepage search portal, as shown in Figure 5 [Figure 5: see original paper]. Search results display harvested URLs, extract titles and content for document types such as HTML, PDF, DOC, PPT, and TXT, and show each URL' s archival count, subject domain, site name, and latest archival time. Users can also sort results by relevance or date, as shown in Figure 6 [Figure 6: see original paper].

In building the Web Archive system for international important scientific institution websites, the open-source software Heritrix was used to achieve large-scale distributed harvesting of web content, with storage in WARC file format. This not only enables long-term preservation of scientific website resources but also facilitates sustainable development of archived resources. Based on Heritrix' s WARC file I/O operation package, hierarchical reading of WARC files was implemented. Modular parsing of different types of network resources was conducted on the read WARC content, laying the foundation for building Solr indexes and providing corresponding interfaces for further data mining, metadata extraction, and data analysis services.

The WARC file indexing and access system developed based on the Solr platform adds faceted content including subject classification, resource type, and archival time, revealing WARC file content from multiple dimensions. The system provides URL and full-text retrieval for archived resources while significantly improving retrieval efficiency for WARC file content. Additionally, the platform can parse and index generated WARC files in real time, tracking dynamic changes in web resources, which is crucial for online real-time updates of Web Archives.

Web Archive resources represent a tremendous treasure trove. How to deeply mine and leverage the value of archived resources represents one of the main future challenges and a direction for future research. We hope our exploration can provide useful reference for colleagues in the web preservation field.

References

[1] IIPC Members [EB/OL]. [2015-12-25]. http://netpreserve.org/about-us/members.

[2] ISO 28500: 2009 WARC File Format [EB/OL]. [2009-05-15]. http://www.iso.org/iso/home/store/catalogue_

[3] Qu Yunpeng. Research on Standardized WARC File Format [J]. Researches on Library Science, 2014(24): 20-25, 28.

[4] Sun Zhiru, Wu Zhenxin, Qu Yunpeng. Analysis of Index Strategies in Web Archive[J]. New Technology of Library and Information Service, 2009(4): 14-18.

[5] Wu Zhenxin, Qu Yunpeng, Li Chengwen, et al. Constructing a System for Harvesting and Preserving Chinese Web Information Resources Based on Open Source Software [J]. New Technology of Library and Information Service, 2009(7-8): 6-10.

[6] WERA 0.4.2RC1 [EB/OL]. [2006-01-17]. http://archive-access.sourceforge.net/projects/wera/.

[7] NutchWAX 0.11.0-SNAPSHOT API [EB/OL]. [2007-02-20]. http://archive-access.sourceforge.net/projects/nutchwax/apidocs/overview-summary.html.

[8] SOLR-Nutch Report [EB/OL]. [2011-01-31]. http://archive.org/~aaron/iipc/solr-nutch-report.html.

[9] Solr Features [EB/OL]. [2016-01-25]. http://lucene.apache.org/solr/features.html.

[10] Wu Zhenxin, Zhang Zhixiong, Xie Jing, et al. Developing Web Archive System of International Institutions Based on IIPC Open Source Software [J]. New Technology of Library and Information Service, 2015(4): 1-9.

Author Contributions

Hu Jiying: Designed parsing and indexing schemes, system development, manuscript writing;
Wu Zhenxin: Designed parsing and indexing schemes, manuscript writing and final revision;
Xie Jing: Designed indexing schemes, system development;
Zhang Zhixiong: Refined indexing scheme design.

Conflict of Interest Statement

All authors declare no conflict of interest.

Manuscript Received: 2016-02-25

Revised Manuscript Received: 2016-03-22

Note: Figure translations are in progress. See original paper for figures.