Developing a Highly Automated Web Archiving System Based on IIPC Open Source Software

Zhenxin Wu, Jing Xie, Jiying Hu, Zhixiong Zhang
National Science Library, Chinese Academy of Sciences
33 Beisihuan Xilu, Zhongguancun
Beijing P.R.China ,10019
+86-(10)-82628382
wuzx,xiej,hujy,zhangzhx{@mail.las.ac.cn}

ABSTRACT
In this paper, we describe our development of a highly automated web archiving system based on IIPC open source software at the National Science Library (NSL). We designed a web archiving platform which integrates with popular IIPC tools, as well as developing several modules to meet special requirements of the NSL. We have applied a cooperative mode of central management server and collecting client, which can complete the unified management of seeds and support the collaborative work of multiple crawlers. Some modules were developed to improve the automation of web archiving workflows and provide more services.

General Terms
Infrastructure challenges; Frameworks for digital preservation; Preservation workflows; Innovative practice.

Keywords
Open source software, Web archive, Platform development Process automation.

1. INTRODUCTION
Web information, which is considered to have cultural heritage value, is protected under laws in many countries. Web archiving refers to the activities of capturing, preserving and delivering web information over time. It provides a reliable way to preserve the web information permanently and effectively. Far more than one hundred projects are ongoing all over the world.

In science and technology (S&T) fields, a large amount of information is published on the Web. The emphasis of international web archiving activities has steadily been shifted to S&T information on the Internet. The National Digital Information Infrastructure Preservation Program (NDIIPP) published a report called "Science @ Risk: Toward a National Strategy for Preserving Online Science" [1], which shows that preserving online science has explicitly become a national strategy.

The important web information of S&T has become an indispensable part of open resources. With keen awareness of the significance of web archiving, the National Science Library (NSL), Chinese Academy of Sciences has paid close attention to web archiving practices since 2006, and carried out research with funding support from Chinese National Social Sciences. In 2013, NSL began to develop a platform for archiving the important web information of S&T. In this paper, we describe our practice of developing a highly automated web archiving system (NSL-WebArchive) based on IIPC open source software. A highly automated platform, which greatly reduces manual work, offers an important advantage for web archiving in the long term.

2. EXTENSION OF WEB ARCHIVING FRAMEWORK BASED ON IIPC OPEN SOURCE SOFTWARE

2.1 Basic Web Archiving Framework of IIPC
The International Internet Preservation Consortium (IIPC),[1] which was founded in 2003, has more than 40 members from over 25 countries, including national, regional and university libraries and archives and non-profit organizations and commercial service providers. It promotes international cooperation and resource sharing.

IIPC has funded a variety of web archiving tools that can be used to select, harvest and archive Web information, like Heritrix[2], Web Curator Tool (WCT)[3] Wayback[4], NutchWAX[5]. And these tools have been widely applied around the world. The most popular four tools cover basic web archiving, as well as WARC [2], which has been international standard web archive format (ISO 28500).

Only a few web archiving projects have been launched in China, and there is a lack of cases of utilizing the above-mentioned open source tools to design a large-scale web archiving system. So far, the National Library of China is the only institute in China to have deployed the experimental system based on the IIPC framework and has carried out archiving activities for several years.

2.2 Specific Needs of the NSL
According to practices as reported in the literature, the web archiving framework of IIPC often needs to be enhanced or adapted to meet local needs. On the one hand, NSL-WebArchive will harvest large-scale web information periodically, and on the other hand the harvest frequency and the harvesting speed should be low enough so that it will not affect daily access. This causes a

[1] http://netpreserve.org/

[2] https://Webarchive.jira.com/wiki/display/Heritrix/Heritrix

[3] http://Webcurator.sourceforge.net/

[4] http://sourceforge.net/projects/archive-access/files/wayback/

[5] http://archive-access.sourceforge.net/projects/nutch/

tension between harvest cycle and harvesting speed. Meanwhile, the more crawling tasks, the more manual labor will be involved, so automation of large-scale, distributed Web information harvesting and in-depth analysis of archived information, became the key issues to be resolved when developing NSL-WebArchive. At the same time, there is a need to support in-depth analysis services of archived information.

(1) Develop NSL Local Web Archive Management Tools

IIPC has funded a variety of web archiving tools for managing the web harvesting process such as Netarchive Suite and the WCT. But they do not meet our requirements for several reasons.

First, NSL-WebArchive provides access and analysis services based on subjects. We add more descriptive information for the target sites, including institution type, subject area, important research fields, etc. We can provide content-based faceted search, site browse and personalized recommendations. Second, in order to achieve crawling efficiently, we need to get more information about the process of crawling to adjust collection strategies. Third, to develop a highly automated web archiving system, we need to monitor and manage the process of crawling, including the running status of multiple crawlers and the sites that are being crawled. If we use the open source software, we must spend a lot of time analyzing source code and developing additional functions.

Considering the pros and cons, we decided to reuse a product of another project undertaken by our team to develop web archive management platform. Moreover, the National library of France and the British Library have both developed a scheduling management platform to achieve better management results. The British Library has visited our institution for in-depth communication. During the development process, we have given serious consideration to their experiences and lessons.

(2) Enhance Distributed Heritrix Framework

The project is currently in its initial stage. In order to save funds, the computers are not powerful and the configuration is at a low level. The number of sites crawled by Heritrix in parallel on a single server is limited. To improve collection efficiency, we develop a distributed Heritrix Framework, so a number of sites can be crawled at the same time. This framework has two advantages:

A) A number of sites can be crawled in parallel at the same time. So one site can be crawled slowly enough to reduce the pressure for both the crawler computer and remote web site server.

B) One crawling task can be dispatched to different crawler computers randomly, so crawling behavior can be marked as different IPs, and will be likely to be regarded as attack behavior.

(3) Enrich Full-Text Retrieval Function

We use solr cloud as a full-text search engine, so the platform can provide not only full-text retrieval but also faceted retrieval and facet navigation. These functions can support the data analysis module in our future work.

2.3 Extension of Web Archive Framework

Based on the IIPC framework, the NSL has designed an extended solution. See Figure 1 below (particularly the parts with blue lines).

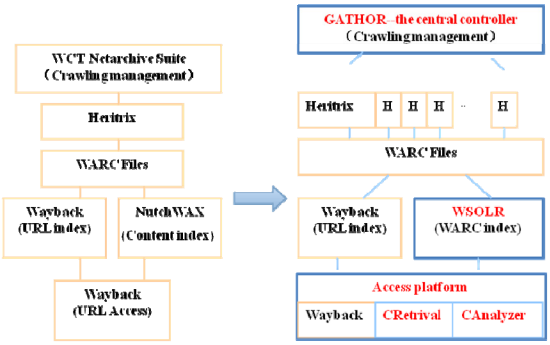

Figure 1. The extended web archive framework

2.3.1 Implementing efficient distributed web archiving management

NSL-WebArchive intends to crawl web information of a relatively fixed and clear website group and does an entire domain crawl for each seed. As the most popular crawler, Heritrix is the best choice for NSL-WebArchive.

Because of so many seeds and internet etiquette, NSL-WebArchive has to deploy many crawlers to execute distributed harvesting tasks at low frequency and speed. The number of crawlers can be increased or decreased according to the tasks.

An efficient distributed web archiving management platform is certainly necessary for NSL-WebArchive, which can manage harvesting tasks and control the distributed crawlers to implement crawling.

2.3.2 Developing an easily recognizable naming convention for WARC files

Each instance of Heritrix uses the default naming rules if it is not changed purposely. But if there are multiple Heritrix systems deployed at the same time, the default naming rules of the crawling configuration files and harvesting files of each Heritrix need to be modified, to allow managers to identify and manage WARC files easily and effectively.

So, an easily recognizable WARC file naming convention becomes necessary. When designing the naming rules, we have had to take many things into account, such as distinguishing these WARC files from different crawlers which are deployed in different servers, and the same seed needing to be collected many times.

2.3.3 Implementing highly-automated processes

Due to a larger number of crawling tasks that need to be configured, managed and periodically scheduled as well as quality control of crawling, we need to realize the automation of crawling task management to reduce manual work.

Multiple distributed crawlers have been deployed in NSL-WebArchive. Unfortunately, Heritrix cannot store WARC files in a remote server, but only in a specified directory of a local server. Each Heritrix has its own result directory even if they are in the same server. Additionally, Wayback can only provide automatic indexing and browse or access service for a specified local directory, so one Wayback cannot work for different Heritrix systems at the same time. NSL-WebArchive will provide a solution for collecting the WARC files from different crawlers in order to facilitate the subsequent management or service.

Without a Hadoop system to use NutchWAX, NSL-WebArchive intends to develop an alternative WARC full-text indexing tool-WSolr (WARC Solr)

2.3.4 Enrich the ways to use archived information

Users need more ways to use archived information. Based on the above-mentioned Solr index, NSL-WebArchive adds a retrieval module named CRetrival，which can provide full-text retrieval and faceted browsing according to subject, timestamp and site, etc.

Finally, NSL-WebArchive intends to supoort content mining and analysis by developing the CAnalyzer module in the future.

3. BUILD UP NSL-WebArchive Platform

Based on the above requirements, we have designed the platform framework with the following three basic principles:

1) The platform framework will integrate with open source software and the customized modules which are developed by the NSL, so that the platform can make full use of the advantages of open source software as well as meet local requirements. And this platform can be built in a short time with better compatibility and seamless upgrade.

2) A cooperative model of central management server and collecting client is applied, which can complete the unified management of seeds and support multiple crawlers' collaborative work.

3) Some modules are developed to improve the automation of web archiving workflows and provide more services.

3.1 NSL-WebArchive Function Framework

NSL-WebArchive applies a cooperative model of central management server and collecting client so that it can implement a distributed crawling and archiving system. As shown in Figure 2, there are three levels, collection level, storage level and access level.

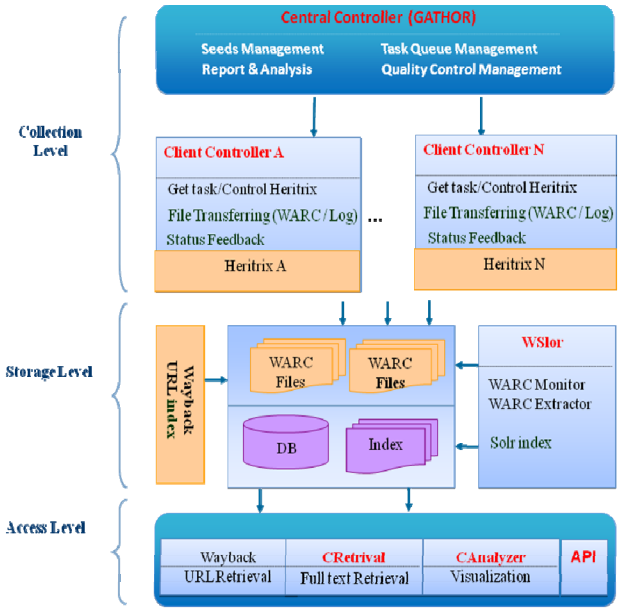

Figure 2. NSL-WebArchive function framework.

3.1.1 Collection Level

The central management server is responsible for the configuarion and management of crawling seeds, and generating and managing the crawling task queue. Meanwhile, the central management server can monitor the status of each crawling task by receiving a report from each client in time.

Each collecting client contains a client controller and an instance of Heritrix. The client controller gets a new crawling task from the task queue of the central management server, and controls Heritrix to crawl web information from the Internet until the crawling task is finished. Then, the WARC files which are stored on local disk of the collecting client will be transmitted to the specified directory in remote server through an FTP pipe, and the current crawling task report will be recorded in database of the central management server.

3.1.2 Storage Level

The storage level stores all WARC files from each collecting client. In addition, we use Wayback and WSolr to create index files in order to provide retrieval and access services.

3.1.3 Access Level

The access level integrates Wayback, CRetrival and CAnalyzer. It provides a series of services, including URL retrieval, content-based retrieval, content analysis and visualization services. APIs will be provided for other system calls, which will be convenient for researchers who are interested in analysis and use of the archived data.

3.2 Workflow of NSL-WebArchive Platform

The workflow supported by the NSL-WebArchive Platform is shown in Figure 3.

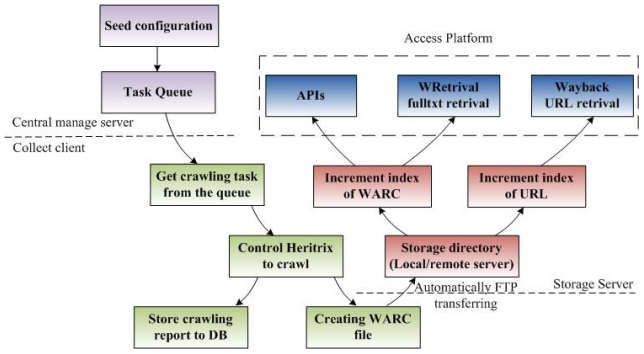

Figure 3. Workflow of NSL-WebArchive platform.

(1) The Manager configures and manages the seeds on the central management server. According to the configuration of each seed, the server will automatically generate the crawling task and put it into the queue on schedule.

(2) The collecting client gets a task from the task queue of the central management server and controls Heritrix to crawl web information from the Internet as well as monitoring the status of Heritrix. When each crawling task is completed, the client will automatically transmit WARC files to the specified directory in the remote server, and then delete the WARC files on its local disk. Finally, crawling logs which are generated by Heritrix for each task will be abstracted and stored in management database of the central management server, and be ready for supporting further analysis and quality control.

(3) Wayback will automatically monitor the specified directory, create an index of the new uploaded WARC files, so users can directly access the new archived data through Wayback.

(4) Similarly, WSolr will automatically monitor the specified directory, extract related information and create incremental Solr index for the new uploaded WARC files, so users can do full text retrieval and facet navigation with CRetrival.

3.3 Advantages of the Collecting Client Active Mode

The NSL-WebArchive Platform is a distributed system, including one central management server and multiple collecting clients. In this system, a definite advantage is the active mode initiated by the collecting client. This platform established an RMI [6] communication pipeline between the central management server and the collecting client. The collecting client actively obtains new crawling task and reports its status to the central management server, so the central management server needs not query each collecting client, and reducing the pressure on both sides, the server and the client. If one collecting client is down, crawling tasks will be assigned to other collecting clients automatically. Unexpected events will not affect the whole platform, and the crawling task will not fail out.

The task token -- which contains the whole description of the crawling task -- is a key element of this distributed system. In one communication between the central management server and the collecting client, the client receives a new task token, decrypts the token, gets crawling task information, and controls Heritrix to carry on the crawling task.

The task token contains: task ID, seed URL, crawling domain, crawling speed and pressure,crawling frequency, seed configuration parameters, etc.

3.4 Developing Multiple Modules to Enhance Process Automation

3.4.1 Task Scheduling Module of the Central Management Server

NSL-WebArchive needs to do a lot of management work, such as seeds management, crawling task configuration, periodically scheduling tasks and quality control. The central management platform implements automated cyclic operation of tasks through a task scheduling mechanism.

This task scheduling mechanism requires the administrator to specify settings for each site collection, including the collection depth, collection frequency, maximum collection time, maximum download, maximum number of jumps, maximum path depth, and the collecting period.

Then the central management server periodically generates collection tasks by setting the timer. The management server periodically checks the collecting period of all sites, and determines whether a new task should be created for a site. If it is overdue, this new task will be put into the job queue.

The collecting client actively obtains tasks from the task queue which is generated by the management server, and generates the necessary configuration file for Heritrix and then calls Heritrix to start collecting. When the task is finished, it obtains the task from the management server again.

In short, once crawling tasks have been configured correctly, the task scheduling module (task scheduler) can dispatch a large number of tasks periodically with the task scheduling mechanism.

3.4.2 WARC File Collecting Module of the Collecting Client

The collecting client periodically executes crawling tasks through the workflow mechanism. The entire process includes some functional modules, from actively obtaining collection command to sending reports of collection results.

Because WARC files are created in different directories by different Heritrix systems, we have developed a collecting module (WARC Gather) for the automatic collection of WARC files. After the collecting client monitors the end of the collection task, WARC Gather transmits WARC files to the specific directory in the remote server by using FTP. After uploading successfully, these local WARC files are deleted. Meanwhile, it transmits log files to the central management server by using the same method.

This module not only solves the remote storage problem of Heritrix, but it also automatically collects WARC files from multiple distributed Heritrix systems.

3.4.3 Status Report Modules

The log files of Heritrix can be uploaded to the management server by WARC Gather. Then the log analysis module of the management server deals with these log files and parses out all sorts of the collection status parameters of each URL and stores them in the database.

Status reports include:

1) The basic report include consumption of time, the number of successful URL, the number of failure URL, the amount of data downloaded, etc.
2) The senior report include proportion of document type, proportion of HTTP status code, seed collection information, URL list and error analysis. All the information is stored in the management database. By adding the task ID to Heritrix source code, statistical data of each crawling task can be viewed.

There is another report status module in the collecting client. By automatically analyzing Heritrix logs, this module monitors the crawling status of Heritrix and presents the crawling status to the central management server whenever necessary, such as the ending of a crawling task or any interruption of a crawling task.

3.5 A Standard Naming Convention

There are four kinds of files that need an effective standard naming convention in NSL-WebArchive.

3.5.1 Seed File

Each crawling task will need a seed file which is created by the client controller after it gets a task from the task queue of the central management server. This seed file is used to store the URL of the target site for Heritrix.

The naming format for seed file is “site domain-seeds.txt”.

3.5.2 Configuration File

Each crawling task will need a configuration file to store crawling parameters for Heritrix.

The naming format of configuration file is "site domain.xml".

[6] http://download.oracle.com/javase/tutorial/rmi/index.html

3.5.3 Task Folder and Task File

Heritrix will generate a task folder for each task in which the crawling log and report are stored. In order to manage the task more easily, we put all the tasks of each month into one sub-folder named with "year-month" in the task folder, e.g., 201403, 201404, 201405 and so on. The task folder "201403" means that it is generated in March 2014 and stores all the tasks that are carried out in that month.

The naming format of task file is "site domain- timestamp"

The UTC time zone is employed for the creation of the time of task folder. Its timestamp format is "yyyyMMddHHmmss".

3.5.4 WARC Storage Directory and WARC File

The WARC files are stored in the remote storage server. The collecting client automatically generates a (new) folder in when it uploads WARC files. As mentioned above, the naming format of each folder is "year-month".The naming format of WARC files is "site domain-WARC file creation time -serial number-Hostname".

1) The site domain is used as prefix to the file name.
2) The WARC file creation time employs UTC time zone. Its format is "yyyyMMddHHmmss".
3) Serial number is the sequence number of WARC files generated in each crawling task. The WARC file size is predefined.

Take www.las.ac.cn for example:

Task folder is www. las.ac.cn -20140323084011.

WARC file is www.las.ac.cn-20140323084024-00000-Hadoop-master-180.warc.gz

3.6 Developing WSolr and CRetrival

WSolr includes three functions: automatic monitoring of WARC files, content extraction of WARC files, and incremental indexing of Solr.

WSolr uses the same mechanism as Wayback to realize automatic monitoring. Meanwhile, it uses three underlying classes of Wayback, WARCReaderFactory, WARCReader, and WARCRecord to parse the content of WARC files. These modules are used to extract and analysis WARC files.

CRetrival can provide content-based search. It can also provide faceted search of archived sites according to time, subject and resource types. By analyzing crawling logs of Heritrix, it can also provide status summary of each crawling task for each seed.

By extracting data from WARC files, NSL-WebArchive not only enriches the search and access services, but it also lays a good foundation for further services of data mining and data analysis.

The goal of WAnalyer is to do an in-depth analysis of archived content by using visualization techniques. At this moment, it is still in the planning stages. A detailed description of this module is not within the scope of this paper.

4. ANALYSIS OF RUNNING NSL-WebArchive

The NSL-WebArchive platform was complete and put online as a beta version in May 2014. 228 seed sites have been periodically crawled and archived. Until Sept. 2015 a total of 20 TB data (compressed) had been archived. The total number of WARC files is more than **1,200** and the total number of URL is **11,392,701**.

Overall, the NSL-WebArchive platform has achieved good results, which are described as follows.

1) The central manage server provides more effective management functions, which reduced the manual work greatly.
2) By developing multiple modules, NSL-WebArchive significantly improves the degree of automation.
3) WARC file extraction module and Solr faceted indexing not only enriches data retrieval, but also lays a good foundation for the further services of data mining and data analysis.

5. EPILOGUE

The NSL-WebArchive platform not only archive the cultural (science) heritage, but also use data mining to support effective assessment of S&T policy, strategic decisions, trends analysis of domain analysis, and predict future trends, etc.

By developing the NSL-WebArchive platform, NSL has accumulated experiences on large-scale web archiving, especially on system management, scalability, automation, and information reuse. In future work, we need to optimize the crawling strategy by analyzing crawling logs, to enhance data preservation and management of WARC files registration and data backup.

6. REFERENCES

[1] National Digital Information Infrastructure and Preservation Program. 2012. Science @ Risk: Toward a National Strategy for Preserving Online Science. Library of Congress, Washington, DC.

[2] ISO 28500:2009 Information and Documentation -- WARC File Format.