VT Web Archiving

Build a web archiving system to archive vt.edu webpages. Integrate Heritrix based crawls with Wayback Machine and index archived files using Solr.

Dev Mehta | Anthony Rinaldi | mehtadev@vt.edu | rinaldi1@vt.edu |
CS 4604 | Virginia Tech | For Mohamed Magdy and Tarek Kanan |
5/6/2014

Table of Contents

Executive Summary

The project was executed in four stages: background research, local setup, machine setup and finding effective configurations. Before the first client meeting, background research and careful study of the documentation was required to understand archiving technologies like Heritrix and Wayback. During the first client meeting, a timeline of deliverables was agreed upon with the clients to establish deadlines and ensure consistent deliverable work. It was recommended by the clients to have a working setup of Heritrix running on local machines for research purposes before moving to the CentOS machine. After establishing local working setups and learning from them, we decided to install Heritrix and start setting up our system on the CentOS machine. During this stage, we also tried to tweak configurations to customize crawls. The hardest part of the project was finding and studying documentation for effective configurations of required technologies.

Researching about Heritrix configurations to setup custom crawls consumed a lot our research time due to lack of proper documentation. Most of the information was accessed and research from https://webarchive.jira.com/wiki/display/Heritrix/Heritrix. The information for setting up and running Heritrix was very detailed and useful. Moreover, we also found outside sources for more ways to run Heritrix with login information on custom ports and servers from the command line. Then, we started researching ways to setup Tomcat plugin for Wayback Machine. Researching allowed for a deeper knowledge of these technologies and assisted during the local setup of the archiving system.

The setup of Heritrix locally was the first step in setting up the system. Heritrix 1.14 was installed and run first locally with login information using a custom port: 1222 and server: webarchive.cc.vt.edu. Soon after, Heritrix 1.14 was installed and run on the CentOS machine. Then, a default crawl job was run on cs.vt.edu to study how Heritrix behaved. It took almost 14 days for the crawl job to crawl cs.vt.edu and finish running. The crawl log and warc files from this job were studied carefully to determine further steps. Since the job took 14 days to run, it was necessary to research custom configurations that allowed for faster and specific crawls. At this time, another meeting with the clients was held to get help with custom Heritrix job configuration and Tomcat setup. At this point, Heritrix was upgrade from version 1.14 to 3.2 to speed up crawl jobs and Tomcat was setup to run Wayback Machine. Then, the focus was shifted to determining the best job configuration to crawl all and only vt.edu web pages.

Finding the right configuration to run Heritrix was a process of rigorous trial and error due to lack of proper documentation. More rules and properties need to be set in order to find the right configuration for vt.edu specific crawls. A lot of time was spent on this stage of the executive process due to a lack of documentation on Heritrix rules. The final crawl on vt.edu domain took 24 hours and was 547 GB in size. This job had to be terminated due to non-domain crawls at the end of the run.

In order to continue building upon this project, more resources are needed to store crawl files. In the beginning, there should be a minimum of 10TB of storage space available for saving

warc files. The amount of storage space should be increased as needed to store new crawls. Also, a Debian-based distribution of Linux would be more ideal due to its ease of usage and widespread use. It would also be optimal to have separate machines for Heritrix and Wayback Machine. This will allow a separation of the back-end and the front-end systems and allow crawls to not hinder the performance of Web Archive visitors.

User's Manual

Figure 1: VT Web Archive front page.

End-users of VT Web Archive can access the system via http://webarchive.cc.vt.edu/. From there, a user can enter a web address and search our archives to see if they have been crawled.

The system is very straightforward and simple to use. A normal internet user would be able to navigate the site with very little frustration or confusion.

Developer's Manual

Prepare the System

Install the latest version of the Oracle Java Runtime Environment. This varies system-by-system, however the most reliable way is using the Oracle recommended method for your operating system. Additionally, you must install Heritrix to a location which has a gargantuan amount of space, or configure it to save jobs there.

Heritrix

1. Download the latest version of Heritrix. Note: we are running version 3.2.0.
2. Extra the downloaded .tar file.
3. Run Heritrix using the following command:

```
heritrix-3.2.0/bin/Heritrix -a [username]:[password] -p [portNumber] -b [server address]
```

-a [username]:[password] allows you to set a custom username and password for administering Heritrix through the browser.

-p allows you to set a custom port to run Heritrix on. Note: you need to open up this port to make Heritrix publicly available.
-b allows you to set a custom IP address to host Heritrix on. This should be the user-viewable IP address.

Note: You **must** use https protocol to access the Heritrix configuration page.

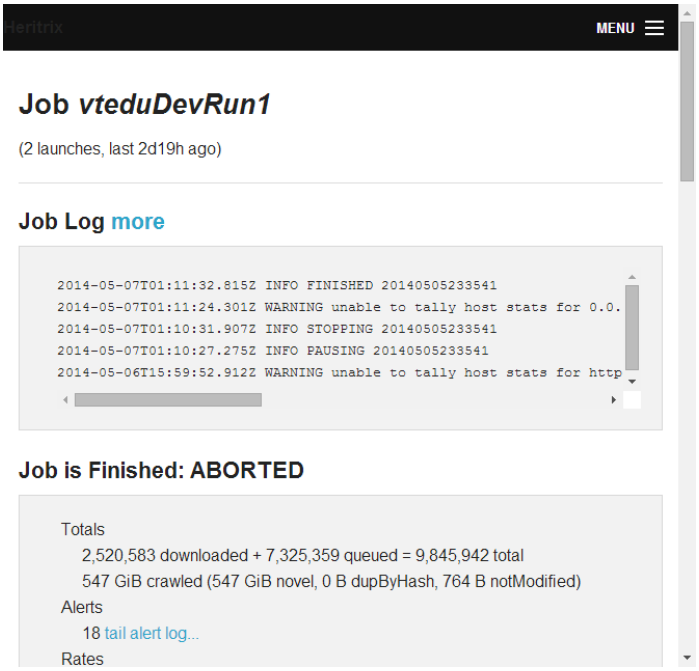

Figure 2: Job Configuration

The default setup will work, however it will enforce “polite” aka slow crawling. This needs to be adjusted. Ensure you allow multiple connections to the same server, as it appears all vt.edu sites are on the same server. The one-thread-per-server rule is hard coded into the frontier, so this should be modified also. The way around this is running multiple instances of Heritrix on the same VM, which has been possible since version 1.4. The key is creating your own scheduling system which when a scheduled crawl should run, it instantiates a new Heritrix system automatically, runs the crawl, then finishes.
In order to run a custom crawl jobs to increase crawl depth, the following properties were modified in the configuration file:

```
<bean class="org.archive.modules
    .deciderules.TooManyHopsDecideRule">
    <property name="maxHops" value="100" />
</bean>
<!-- ...but ACCEPT those more than a configured
link-hop-count from start... -->
<bean class="org.archive.modules
    .deciderules.TransclusionDecideRule">
    <property name="maxTransHops" value="100" />
    <property name="maxSpeculativeHops" value="100"
/>
</bean>
```

Figure 3: Editing the job configuration file

Also, a rule was added to ensure only domains with vt.edu were crawled. Note: at the beginning of the crawl job, this rule seemed to be effective, however towards the end it started crawling non-domain URLs.

```
 <!-- ...but REJECT those from a configurable (initially empty) set of REJECT
SURTs... -->
 <bean class="org.archive.modules.deciderules.surt.OnDomainsDecideRule">
      <property name="decision" value="ACCEPT"/>
 </bean>
```

The crawl job with this configuration took a day to crawl 547 GB of data. This job was terminated because of non-domain based crawls towards the end of the crawl.

Tomcat

The Wayback Machine, as described in the next section, runs as a plugin to Apache Tomcat. After testing, the best version to use with the current version of Wayback is Tomcat 7. Ensure you download the latest version of that major revision from http://tomcat.apache.org/.

Figure 4: Tomcat configuration file

Installation of Tomcat is simple enough; extract the package to a given folder with enough free space to host a website. Once Tomcat is installed, it needs to be configured. In conf/tomcat-users.conf, you can configure users for Tomcat. This is not required, as we have access to the command-line and configuration files. Next, open `conf/server.xml`. Locate the Connector for HTTP/1.1, which defaults to port 8080. If you want to leave Wayback on a non-standard port, then don't change it and make note of the port. Otherwise, switch it to port 80 and restart Tomcat.

Wayback Machine

Wayback is the main frontend for navigating Heritrix crawls. Do NOT download Wayback from Sourceforge. That repository has not been updated since 2011. As of the writing of this document, the primary repo for Wayback is available on GitHub: https://github.com/internetarchive/wayback. Download the latest snapshot and prepare to install the software into Tomcat.

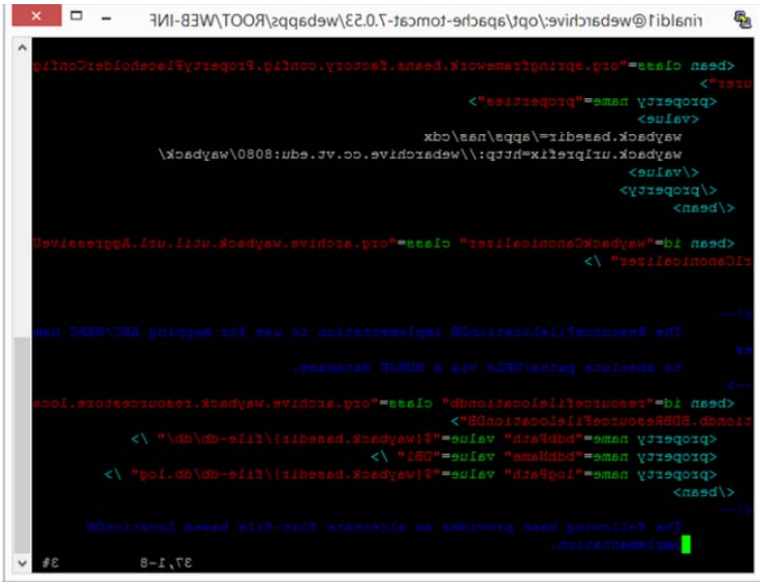

Figure 5: Wayback configuration

Installation of Wayback can be tricky if done improperly. Firstly, it must be run as the ROOT plugin for Tomcat. So, go to the folder `[tomcat]/webapps/` and remove the `ROOT` folder. Then rename the Wayback .war file to `ROOT.war` and place in the Tomcat webapps folder. When Tomcat is running, it will automatically unpack the plugin and begin execution. Check to ensure the program is running by going to `http://[server]:[tomcat port]/`.

The configuration of Wayback is simple, yet easily made tedious. First, you must configure the base directory and user-accessible URL for Wayback. Here is an example of the first Wayback configuration file: `[tomcat]/webapps/ROOT/WEB-INF/wayback.xml`

```
<bean class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">
    <property name="properties">
      <value>
        wayback.basedir=/apps/nas/
        wayback.urlprefix=http://[tomcat server]:[tomcat port]/wayback/
      </value>
    </property>
</bean>
```

The base directory should be within the path for the Heritrix jobs folder, as well as a location where temporary parsing files can be stored and edited. The urlprefix is prepended to all served relative http calls. Note that there is a `/wayback/` at the end of it. By default, the server serves to the `/wayback/` virtual directory. This can be changed by a careful editing of the Wayback configuration files.

Next, configure `[tomcat]/webapps/ROOT/WEB-INF/BDBCollection.xml`. This will configure the path to the Heritrix jobs directory. See an example of the important segment of the configuration, with the two important properties bolded:

```
<bean id="datadirs" class="org.springframework.beans.factory.config.ListFactoryBean">
    <property name="sourceList">
      <list>
        <bean
class="org.archive.wayback.resourcestore.resourcefile.DirectoryResourceFileSource">
          <property name="name" value="files" />
          <property name="prefix" value="${wayback.basedir}/../heritrix-3.2.0/jobs/" />
```

```
        <property name="recurse" value="true" />
      </bean>
    </list>
  </property>
</bean>
```

The prefix value should be the Heritrix jobs directory, or wherever your Heritrix jobs are saved. To ensure compatibility, keep the relativity to the Wayback basedir. Also, change recurse to true to ensure all subfolders are checked automatically for WARC files.

Once Wayback is configured, restart Tomcat. If configured correctly, it will automatically run Wayback and it will start loading your Heritrix crawls. After the software is verified to work as expected, then customize the webpage template located in `WEB-INF/template`. Most of the static data is located in `UI-header.jsp`. For example, change the search year drop-down to only go back as far as you customize, and automatically increment as time moves on:

```
<select id="searchyear" name="<%= WaybackRequest.REQUEST_DATE %>" size="1">
  <option value="" selected><%= fmt.format("UIGlobal.selectYearAll") %></option>
</select>
<script>
var myselect = document.getElementById("searchyear"), year = new
Date().getFullYear();
var gen = function(max) {
  do {
    myselect.add(new Option(year--,max--),null);
  } while(year>2013);
} (5);
</script>
```

Opening Ports

Opening ports to the network is an important step which changes depending on the type of system the servers are running on. For example CentOS (RHEL) and Debian/Ubuntu have very different methods for allowing incoming connections.

Here are example IPTables configuration statements necessary to open a port:

```
iptables -A INPUT -p tcp -m tcp --dport [port to open] -j ACCEPT
iptables-save
service iptables restart
```

It is possible, though unlikely, that this will not work. If that is the case, you will need to disable IPTables and replace it with a firewalling system compatible with your machine configuration.

Lessons Learned

Described below are the issues we faced and the solutions developed to solve those problems.

Timeline/Schedule

This project should have been started earlier to allow time for more trial and errors. The timeline set during the first client meeting should have been thoroughly followed throughout the semester.

Problems

There was a lack of documentation regarding building custom configured Heritrix jobs. Finding the right rules and properties to customize crawls for all and only vt.edu web pages was a process of trial and error which consumed a lot of time.

Issues with CentOS firewalling were also encountered during Tomcat setup for Wayback Machine to allow public access to the Tomcat server.

Solutions

A solution to building the perfect crawl job was not attained due to the lack of documentation. More trial-and-error runs are required to build a perfect configuration for crawl jobs on vt.edu pages.

CentOS firewalling issue was solved by configuring IPTables to open and accept necessary ports as described above in the Developer's Manual Ports section.

Future Work

The next task to be completed is creating the best possible configuration for Heritrix. Future research is needed to find the best configuration to crawl all and only vt.edu webpages. A solution should entail the crawling frontier in a customized way to allow only the download of Virginia Tech domain hosted files or rejecting all but those files. This will be a rigorous process of trial and error.

Solr should be the second item which needs to be set-up for indexing and searching crawl files. After setting up Solr, a scheduling system must be developed in order to schedule crawl jobs with the proper configuration.

Additionally, to reach a production rather than a pre-release status, there must be scheduler.

Acknowledgements

The tremendous help provided by Mohamed Magdy, mmagdy@vt.edu, and Tarek Kanan, tarekk@vt.edu, in helping with configuring these technologies and guiding towards our goal is appreciated. The patience and time dedicated by them during this semester is regarded with the utmost respect and regard.

The guidance and support provided by Dr. Fox, fox@vt.edu, in helping acquire resources for system setup is also greatly appreciated.

References

Apache Tomcat. (n.d.). - *Welcome!*. Retrieved May 8, 2014, from http://tomcat.apache.org/

Heritrix. (n.d.). *Heritrix*. Retrieved May 8, 2014, from

https://webarchive.jira.com/wiki/display/Heritrix/Heritrix

internetarchive/wayback. (n.d.). *IA's public Wayback Machine*. Retrieved May 8, 2014, from

https://github.com/internetarchive/wayback