YouSeer Tutorial

What is this YouSeer?

YouSeer is an open source search engine framework, which was built on top of other open source components. It's part of the general SeerSuite framework. YouSeer utilizes Hereitrix as a crawler and solr as an indexing system. The framework provides software to ingest the documents harvested by Heritrix into solr. The ingesting software is very flexible and allows for user-specific data extraction implementations. Further, YouSeer provides a simple interface to query the index and another interface to retrieve cached versions of the documents.

What is Heritrix?

Heritrix is the Internet Archive's web crawler, which was specially designed for web archiving. It is open-source and written in Java. The main interface is accessible using a web browser, and there is a command-line tool that can optionally be used to initiate crawls. Internet Archive and the Nordic national libraries on specifications developed Heritrix jointly. The first official release was in January 2004, and members of the Internet Archive and other interested third parties have continually improved it.

Heritrix runs the following steps in a loop: 1) Choose a URI from among all those scheduled. 2) Fetch that URI. 3) Analyze or archive the results. 4) Select discovered URIs of interest, and add to those scheduled. 5) Note that the URI is done and repeat. These tasks are parallelized across threads for the time being, with plans to expand the parallelization to be machine-wide

Heritrix by default stores the web resources it crawls in an Arc file. The Internet Archive has used the Arc file format since 1996 to store their web archives. An Arc file stores multiple archived resources in a single file in order to avoid managing a large number of small files. The file consists of a sequence of URL records, each with a header containing metadata about how the resource was requested followed by the HTTP header and the response. Arc files range between 100 to 600 MB.

What is Solr?

Solr is an open source enterprise search server based on the Lucene Java search library, with XML/HTTP and JSON APIs, hit highlighting, faceted search, caching, replication, and a web administration interface.

Requirements:

1) Java 1.6 or greater
2) Solr release (the test was conducted on version 1.3)
3) Internet browser
4) Heritrix release (the test was conducted on version 1.14.3)

5) Apache tomcat, to host the web application

The application was tested on Linux. In theory it should work on all platforms as the entire code is written in java (including heritrix and solr)

Downloading YouSeer:

You can download the source code along with the executables from sourceforge: https://sourceforge.net/projects/youseer/

You can also get a VMware image of fedora 11 that has the system installed and running. The VM has:

1) Heritrix 1.14.3
2) Solr 1.3
3) MySQL 5
4) YouSeer 0.1
5) YouSeerUI 0.1
6) Apache tomcat 6.0

The user youseer is identified by the password: ***heritrixsolr***

The root password for the VM is ***thoughtpolice***

Getting Started:

This tutorial doesn't provide instructions for running heritrix or solr as they have their own comprehensive tutorial pages.

You can get Heritrix from Source Forge:

https://sourceforge.net/projects/archive-crawler/files/archive-crawler%20(heritrix%201.x)/

This tutorial will guide you through the heritrix setup and your first crawl job:

http://crawler.archive.org/articles/user_manual/index.html

Solr can be downloaded from apache:

http://www.apache.org/dyn/closer.cgi/lucene/solr/

Although the solr index structure is easier to customize than lucene's or nutch's, more reading is necessary to familiarize yourself with it. The link to the official tutorial page is below.

http://lucene.apache.org/solr/tutorial.html

The following tutorials for solr, written by Grant Ingersoll from IBM, are more informative than the official tutorial.

http://www.ibm.com/developerworks/java/library/j-solr1/

http://www.ibm.com/developerworks/java/library/j-solr2/

http://www.ibm.com/developerworks/java/library/j-solr-update/?S_TACT=105AGX01&S_CMP=HP

Ready to go?

(add something about ARC files and solr)

If you want to test the framework right away without crawling or modifying the structure of the index, ARC files are provided. Test crawling created these ARC files.

Test ARC files: www.personal.psu.edu/mxk479/youseer/ARCfiles

A working Solr schema is provided. Other solr schemas are possible with modification of the configuration file.

Solr schema: www.personal.psu.edu/mxk479/youseer/schama.xml

In order to replace Solr schema with the one you have downloaded put it in the folder: $solr_home/example/solr/conf

Then restart solr. Download the ARC files to your local machine. Now you are ready to go.

Basically the schema contains the following fields:

```
<field name="URL" type="string" indexed="true" stored="true" required="true" />
<field name="path" type="string" indexed="true" stored="true" required="true" />
<field name="title" type="text" indexed="true" stored="true"/>
<field name="type" type="string" indexed="true" stored="true"/>
<field name="recordOffset"  type="slong" indexed="true" stored="true"/>
<field name="text" type="text" indexed="true" stored="false" multiValued="true"/>
<field name="file_content" type="text" indexed="true" stored="true"/>
<uniqueKey>URL</uniqueKey>
<copyField source="URL" dest="text"/>
<copyField source="path" dest="text"/>
<copyField source="file_content" dest="text"/>
<copyField source="title" dest="text"/>
```

Note: the configuration file supplies the mapping between these fields and the ingesting system fields, so if you have your own field names don't worry. You can still have them, but you need to modify the configuration file. We'll get to that later.

Let's test drive!

Ok, now you are ready to test the system. Navigate to
$youseer-home/release
and then you should be able to execute commands similar to the one below:

java -jar YouSeer.jar [IndexUrl] [Path_of_ARCfiles] [Cached_virtual_folder][Number of threads] [Waiting_time]

IndexUrl is the URL of solr that is used to post the documents. If you are using solr's JAR file, then it should be: http://localhost:8983/solr/update

Path_of_ARCfiles is the absolute path of the folder that contains the ARC files to be processed. All subfolders will be check as well.

Cached_virtual_folder: We will retrieve the cached copies of the document from the ARC files rather than the database or index. To do so we will map the **Path_of_ARCfiles** folder to a virtual path in the web server, in our case tomcat. This will allow the users to move the ARC files wherever they want as long as they keep the same hierarchy. The user will have to add the virtual folder entry in the web server configuration file. We'll mention how to do it with Tomcat.

So, let's assume that the ARC files are in the folder
/home/youseer/arcdata
and we choose the virtual path to be
/cached

Now each ARC file in the folder will be accessible by the URL:
http://localhost/cached/ARCfileName.arc.gz
Now let's assume that you decided to move the arcdata folder to another hard disk (you got more money and decided to buy new HDD for storage). In this case you need to go into configuration file of tomcat (or your web server) and map the /cached virtual path to the new location of the folder. Note that you need to keep the same hierarchy of subfolders.

Number of threads: number of working threads that are processing documents. It's recommended to have the same number as the cores on your machine. The number of threads you specify only determines the processing threads. Beyond that, there is always one extra thread for reading the ARC files.

Waiting_time: the ingestion part can be configured to recheck for newly crawled ARC files every few minutes. This is provided using a waiting time parameter. If you omit this option the default value will be 10 minutes. If you make the time parameter equal to zero the application will terminate immediately after processing all the available ARC files.

So, a typical program call will be:

```
java -jar YouSeer.jar http://localhost:8983/solr/update /home/userX/ARCfiles /cached 3 0
```

The above call will process all the ARC files in the folder: /home/userX/ARCfiles and submit them to the index reached on localhost:8983. It'll create three processing threads and terminate immediately after processing the contents of this folder.

Now you should see the number of processed documents on your terminal. You can check the log file, which is created in the same directory, ARCSubmitterLog_YYYY_MM_DD_HH_MM_SS.txt

The database file, ***submitter.db*** should contain an entire log of the process. You can query the database to get the details; more on this later.

Ok, let's try out this search engine. You can either navigate to solr admin page or use youseer user interface.

Solr admin page is: http://localhost:8983/solr/admin/ (replace localhost:8983 with the URL of your index)

Or you can deploy the companion *youseerui* war to your web server and navigate to the index.html page. To use youseerui as a web interface:

1) Copy the file youseerui.war to the folder $tomcat-home/webapps .
2) Start tomcat, if it's not running. The server should extract the war file and it will create a new folder named youseerui.
3) Copy the folder $youseer-home/release/lib to $tomcat-home/webapps/youseerui/WEB-INF
4) If you are using the schema that we provided, and the default solr jar executable, then you can skip this step. Otherwise, you need to modify the web.xml file in $tomcat-home/webapps/youseerui/WEB-INF. The comments in the XML file will describe each field; modify them according to your own settings.

Setting the configuration file

Ok, you are probably wondering now where that database file came from, or where the Submitter found the schema fields? It's all mentioned in the SubmitterConfig.xml file which contains all these settings.

The config file has 3 main parts: schemaConfiguration, indexedDataTypes, and databaseConfiguration.

The schemaConfiguration section provides the mapping between the important indexed fields of any document on the web (HTML and others) and the fields defined in solr schema. So, you need to provide the name of the URL fields in solr's schema file, in addition to the title, content, type, cache, offset. We'll describe each of them:

schemaConfiguration

URL: the easiest thing to understand, and you probably will have the same field to describe the URL of the document.

Title: the name of the field that will hold the title of the document in your index.

Content: the content of every document will be stripped from HTML tags and then indexed in solr. This field will hold the raw text content of the document.

Type: the field in the index, which will save the type of the indexed document.

Cache: the field in the index that will have the relative path of ARC file that contains this document.

Offset: since the document is stored in an ARC file which contains thousands of other documents, it's useful to store the offset within that ARC file for each document so that we can retrieve the document quickly. This will hold the name of the offset field in the index.

indexedDataTypes

This will list the file types that you wish to process and index. By default, the system can handle all the text/* files using standard java I/O libraries. Other file types like PDF or Microsoft word are supported using Apache TIKA 0.3. You can choose what files you want to index by listing their type in this field. The available formats are listed here.

databaseConfiguration

This section will provide the database configuration for the application to use. YouSeer ships with SQLite as the default database, but you are welcomed to use another database if you have it installed on your system. However, using SQLite was optimized so that it doesn't affect performance of the crawling, ingesting, or the retrieving system. It's only used to log the data during the ingesting section; the insert commands are cached and then executed in batches to avoid writing to the disk for every ARC record.

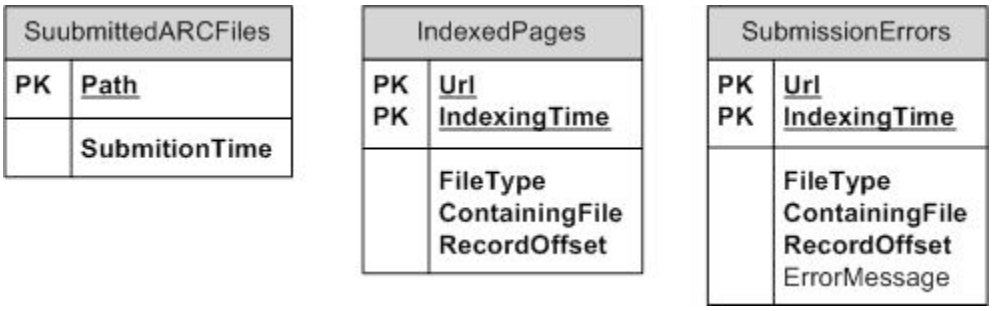
To use MySQL or any other DBMS, replace the configuration file with the appropriate values. For example to connect to CrawlDB in your MySQL server with username user1 and password pass, the values will be:

```
<databaseConfiguration>

<provider>com.mysql.jdbc.Driver</provider>
<connectionString>jdbc:mysql://localhost/CrawlDB?user=user1&amp;password=pass</connectionString>

</databaseConfiguration>
```

Please note that you have to create the database, as the application will only create the tables in case they don't exist. The user should have create privilege and insert on the database. It's recommended to have a dedicated user for this job that have all rights on the CrawlDB.

YouSeer's JAR file adds *mysql-connector-java-5.1.8-bin.jar* to its classpath variable. However, mysql connector **is NOT shipped** with the distribution. You need to download it and put it in the folder: $YouSeer-home/release/lib. If you are using a different DBMS, you should make sure that you add the connector JAR to the lib folder, and modify the classpath of the JAR file (in the manifest) to include the new JAR file. Please note that providing the classpath variable while running a JAR file will be discarded by the JVM, as all the required classes have to be mentioned in the JAR itself.

The schema of the database is:



FAQ:

How to extract my own data fields?

Extracting your own fields is very easy and straightforward. All you need to do is to implement the *GenerateCustomeData* method in the *Extractor* class. This method will be passed an object of type *SubmitterDocument*which that contains all the information you need to process documents. You'll find a commented out example method in the class .

The idea of the system is that for each SubmitterDocument, the system will process it and generate a solr document on the fly. The Extractor class can augment the contents of each solr document. So, for every document the ARCSubmitter will pass the SubmitterDocument to the Extractor, and expect it to return XML tag. This tag is then added to the being generated solr document. Eventually, the entire document is submitted to solr.

Let's look at this example, in which we want to extract the publication date of each article in The New York Times. The date in extracted from the URL of the page, as each URL will have the date. Note that the process is similar if the data will be extracted from the content of the page rather than the URL.

```
public static String GenerateCustomeData(SubmitterDocument doc)
 {
            StringBuilder command = new StringBuilder();
             String url = doc.getUrl();
String date = url.substring(23, 27) + "-" + url.substring(28, 30) + "-" + url.substring(31, 33) +
"T00:00:00Z";
            command.append("<field name=\"publication_date\">").append(date).append("</field>").
append(ARCSubmitter.LINE_SEP);
return command.toString();

}
```

A typical URL in The New York Times will be like this:

http://www.nytimes.com/2009/07/26/science/26robot.html

How to use my own converters?

If you are not satisfied with Apache TIKA conversion tools and you wish to use your own convertor to extract the data, this is easy too. Every *SubmitterDocument* object has the raw binary data of the document stored in the ByteContent byte array. The getter can access this, and you can do whatever kind of processing you prefer on the raw data.

How to configure the virtual directory in Tomcat to map to my cached folder?

Add the following directive to your [$Tomcat-Home]/conf/server.xml file, where the path is the virtual name you want to use, and docBase is the absolute path of the folder which you want to map.

<Context path="/cached" docBase=”/heritrix-1.14.3/jobs/CollegianArcAll/arcs" crossContext="false" debug="0" reloadable="true" />

Now if you decide to move your ARC files from this folder to another one, you only need to modify this directive in your web server and all the cached links will be working. Note that the database is not modified and you need to update it yourself, though it’s an easy task to do.

How to query the SQLite database?

Download the command line accessing program for your platform here. And follow this tutorial.