

Meta Search Engine for NWRC

A project report by
Swathi Chitteddi

Submitted to
Dr.Vijay Raghavan

Fall 2000

The center for Advanced computer studies
The University of Louisiana at Lafayette

Metasearch Engine for NWRC final Report

Project Member

Swathi Chitteddi (svc0638@louisiana.edu)

Source Code directory:

In lincsun03 -> /local/linc/apache/cgi-bin/raghavan/vvp7188

The project URL:

<http://lincsun03.cacs.usl.edu/~vvp7188>

1. Introduction:

This project is continuation to the work done by Varaprasad Penaganti. The Metasearch Engine with one database was developed and the results were seen at the command line previously. The work done by the previous student was continued to develop a web interface for the project and increase the number of databases.

The main aim of the project is to develop a Metasearch Engine for various databases used by the National wetlands research center (NWRC). NWRC uses different databases for searching information and the results are obtained at different user interfaces. To see all results of all the search engines at one user interface, Metasearch engine project is developed. This will send query to all the databases from one interface and displays all the results at one place. With this users can see all the results at singer user interface and helps to find the information he is looking for soon.

An effort is made in this project to design and implement a simple Metasearch Engine that uses the following bibliographic databases as the underlying search engines.

1. EPA Publications - www.epa.gov/epahome/pubsearch.html
2. Pub Science Database - <http://pubsci.osti.gov/srchfrm.htm>
3. USGS Database - <http://water.usgs.gov/swra/>

These databases are chosen after discussing with NWRC staff.

2. Architecture of the Metasearch Engine

General Architecture of Metasearch engine:

The Meta search engine takes keyword and No of documents that can be displayed for each database as the input. User can select databases from which he wants to seach for the information. The user can select one or more databases from the three databases available. These three databases are 1. USGS Database, 2. EPA Database, 3.

PUB SCIENCE Database. The user can also specify the number of documents that should be displayed for each database on the result page. Each database has its own rules for the keyword. For example some databases don't support AND and OR. If the user enters the keyword which is not supported by that particular database, the results will not displayed. Clicking the help link on the first page user can see the information about the keywords supported by each database.

User clicks the search button after entering the keyword, No of documents and selects the databases. Then the program reads the file Metafile.txt. The contents of the file are (for each database or Search engine)

1. The name of the database or the search engine.
2. The URL of the particular database.
3. The starting point where the hit section cutting begins
4. The ending point where the hot section cutting ends.

All the information is stored in the file for the three databases.

The server then spawns threads for the different databases using the URL read from the file. Each thread takes care of reading the results given by that particular database to the query. For a particular query there could be many pages of the hits. The thread takes care of going through all the pages of the results and it then cuts out only the hit section. Once the thread has finished getting all the results, it stops. The server then prints out the results of each of the different threads. So in essence, the results for the particular keyword given to different databases or search engines are displayed.

3.1) User Interface

The first page of the Metasearch engine has user interface constructed using HTML forms. The user has to enter keyword, No of documents and has to select the databases from which he wants to search for the information. The user can enter one or more keywords in the keyword text box. The words can be separated by the AND or the OR conditions depending on the databases. More information regarding how many words can be entered and in what combination can be found by the user by clicking help link.

The user can ask for the number of documents that can be obtained from searching each database and maximum number can be 50. The user has to select one or more databases from the three databases that are present on the page.

3.2) Query Formulation, Submission and Results

Once the user submits the keyword for search, the data is transferred from one page to another with the help of the shell script. This script is stored in particular file and it runs the java program taking the data from the previous file. This java program reads the data and parses it to get the required information. The data consists information about the databases selected, keyword and no of documents. Once all the values are known, the file (metafile.txt) consisting the databases information is read. Only the information about the databases selected is read from the file. The URL of each of the database that is present in the file has some word in the place of the actual keyword. This word has to be replaced by the actual keyword before this URL can be used as the query. Keyword is replaced by calling a program and each database has separate program for this as each database has different format for the URL and replacement of the keyword has to be done in different way. Some databases accept the only certain format of the keyword combinations. Depending on the format the programs were written.

If a new database has to be added to already present databases, then the details of the database has to be added to the metafile.txt first. Then later a separate program should be written for replacing the words in the URL with the actual word for searching.

Once the keyword is replaced in the URL as required, thread is created. Another program called MetasearchThread.java is written and database name(dbName), changed database URL(changedDBUrl), the first cutting point of the hit section(firstIndex), second cutting point of the hit section(secondIndex), Number of documents(NoOfdocs), search word(keyword), starting number of the documents(stofdocs) are sent as the arguments to the constructor of the program while creating thread. For each database threads are created and the results are displayed for each of the selected databases. For each database, the no of documents will be as specified by the user. On the result page the user is provided with the facility to search for more documents for each of the

databases. He has to click search button for it and on the next page user can see next documents whose count will be the number of documents specified by the user. Like this user can continue till he has seen all the documents obtained by the database or the search engine.

For example user can enter the word mercury in the keyword text box, enter the number of documents and select any or all the databases and click search. On the next page user can see the results.

4. Design

Index.html: Starting page of the Metasearch Engine.

The user can enter search word, Number of documents for each database and select the databases from which he wants to do the search. Once the form is submitted, a shell script program `inter.cgi` is called.

Inter.cgi: Shell script program connecting html and java program.

This program reads the values submitted as a single word and runs the java program with the single word as the argument. The java program that is run by the `inter.cgi` is `Metasearch.java`.

Metasearch.java: Java program for parsing the values and creating threads.

This program reads the argument and parses it accordingly. All the values are stored in separate arguments. The information regarding the selected databases is obtained by reading the values from the file `metafile.txt`. After reading information for each of the selected database, function written to replace the keyword in the database URL is called. There is separate function for each of the databases as there are different formats of the URL for each of them. For each of the database their respective programs are called. In the functions, the keyword is replaced in the URL as required. Here it is checked whether the user is asking for the AND or OR combination search and the action is performed accordingly. After the URL is changed, thread is created for that database. New instance of the Program `MetasearchThread.java` is created with some arguments that

are specified above. After the thread is complete another thread for another database is created. Thus the process is continued till all search is done for all the selected databases.

MetasearchThread.java :

The names of the functions used in this program are run, startconnection, getPage and getHits.

In the function run, function startconnection is called.

In function startconnection, first getPage function is called with the database URL as argument passed to it. In this function, URL connection is first established using the database URL obtained from the previous program. After obtaining the connection, the page obtained is read into a buffer and later stored as a string in a variable. This result obtained from the function is stored in some other variable. After this, function getHits is called with the following arguments: result obtained from getPage(hitData), database name(db_name), Number of documents(doc_count), starting cutting point of the hit section(doc_fIndex), last cutting point of the hit section(doc_lIndex,), starting number of the count of the documents(st_docs), Database URL(doc_url). In this function, first the database name is checked and code is written separately for the display of the results. Each of the databases has their own format for the display on their websites. Thus to have proper display at our Metasearch Engine web site, separate programs are written for parsing the web page to get the results in the required format. During the parsing the web page, the values of starting cutting point of the hit section and the last cutting point of the hit section are used. Parsing is done for the second time for formatting the documents. For parsing the java commands such as indexOf and substring are used a lot. After the documents are parsed they are displayed as the html pages and the results are shown to the user. User can ask for more documents for each database or search engine by clicking 'more'. When user asks for more documents again, inter.cgi is called by passing keyword, number of documents, starting number of the documents and that particular database as the hidden variables. Again all the process is repeated.

5. Future Directions:

This project can work with three databases and can be expanded to any number of databases by adding its information to the file and adding some more code to the program. I

Some directions, which need to be further explored are as follows:

1. The results that are obtained from various databases are displayed. Some postprocessing needs to be done so as to remove duplicates and rank them.
2. There is need for the generic method for the cutting of the results. The logic behind this is that the user would be able to simply add a new database in the metafile.txt specifying the information of the new database and the Metasearch Engine would be able to search the newly added database.
3. A mechanism for handling the threads should be developed so that speed and the robustness of the system can improve.
4. The system can show more documents for each database at a time. It can be improved such that user can see the more documents for all the databases or the selected databases at a time.