Project Report

COMP4047 Internet and World Wide Web

HONG KONG BAPTIST UNIVERSITY
SEMESTER 1, 2016 – 2017

Group 9

12210102	MAK KIT TIN
15208249	LEUNG KIN HIN
15211886	WU WAI LOK
15207323	TANG SZE MAN

Table of Content

Introduction

The project's aim is to build a search engine which can be divided into two parts, i.e. client side and server side.

On the server side, it handles the web crawling, data storage and sorting by URL's ranking. On the client side, there will be a website and CGI server which allows users to search a keyword through the browser.

In the project demo, "http://www.hkbu.edu.hk/eng/main/index.jsp" will be used as an example to show the functions of the search engine.

The design and implementation will be explained in the following parts.

Project Design

Crawling Algorithm

In this crawling algorithm, there would be an initial web address for launching it. The crawler would scan the whole web file and remove all the coding parts (invisible words) on the web file.

During the process of tags remove, there would be two steps which are on going concurrently.

First, once the algorithm has found the line contains any hyper links, the algorithm would store it into a URL pool if it fulfils three conditions.

1. URL pool has not reached the maximum size X yet.
2. Processed URL pool has not reached the maximum size Y yet.
3. the URL has not been put into the processed URL pool yet.

Second, after removing all the invisible words (tags) in each line, the algorithm will go through the scanning process of visible words for page viewers. All the words would be captured and store it into the database. The data storage structure and algorithm would be discussed in the section below.

The crawling process will stop when processed URL pool has reached Y (the maximum size of processed URL pool) or when URL pool is empty with the completion of current webpage scanning.

Data Storage Structure

Each keyword would be stored in to a text file which has the same name of the keyword itself. Each file will contains all the links which are related to that certain keyword of the text file.

The file would be in the structured each line in “number of word occurrences in at that URL, webpage title, URL, sponsor”. During the scanning process, if the link is not recorded in that keyword’s text file yet, it would be written as a new line at the end of the file. If the link has been recorded before, the number of word occurrences will be increased by 1.

All the text file would be stored into a subfolder named as its first character. Hence, there would be subfolders in the database from “a” to “z”. It can greatly enhanced the efficiency of search query because the query would filter out all the not-relate subfolder in the first place. It will minimise searching from all files to approximately 1/26 files assuming that the distribution of every first character of words is in average. It means that it will reduce more than 96% of the files that are not related to the keyword under this assumption.

Ranking & Sorting Algorithm

The ranking algorithm in this search engine is quite simple. URL who is sponsored would be at the top of that keyword’s search. If there are more than one sponsors, then the number of word occurrences will be the key to determine its ranking. This process will be done right after the crawling process by server side rather than client side. It would speed up the queries of users because there will be no more sorting when the users search the keyword but it will return the whole list.

Project Implementation

Server Side

Crawler.java

It is responsible for the crawling algorithm. There are a few important variables here. Integer X and Y represents the URL pool size and the processed URL pool size. String ArrayLists to store the URL pool and processed URL pool.

There are two main methods with some methods to support its process. One is getKeyword(String url) and another is getUrl(String url).

> getKeyword(String url) is responsible for getting keywords from the page and store it into the specific text file.
> getUrl(String url) is responsible for getting URLs in the processing page and put into the URL process pool.

However, there are two methods which are important in this java file which are createFolder() and mkDirs(File root, List<String> dirs, int depth). These two methods are responsible for database folder creation and file creation.

In the demo program, it uses "http://www.hkbu.edu.hk/eng/main/index.php" as a sample to launch the crawling algorithm. X equals to 10 and Y equals to 100. The database storage location would be "C:\db" .

When the program is initialised, the program will call the deleteDB.java for database clearance and createFolder() in order to create an empty database structure. Then the crawling methods will start and gather the keywords and URLs. At the end, it will call the sorting.java for implementing the ranking and sorting algorithm.

deleteDB.java

It is responsible for database clearance which will be called by Crawler.java before crawling algorithm starts.

Item.java
It's the structure of class Item which enables methods of get and set for each records in database.

sorting.java
It's the ranking and sorting algorithm. It applies bubble sort here in order to sort the word occurrences. It will process all the files in the database and sort the lines order in terms of if it is sponsor and the by word occurrence count.

Client Side

SearchServer.java

It is responsible for web display for clients. It contains all the HTML codes and styles require in the HTML file. In this program, it will call Query.java in order to get the results from the database.

Query.java

This Java program is for the keyword query operation. There will be the input parameter. In this sense, this program will find the text file in the database by finding the folder named as the first character of the input keyword.

server.py

It is the CGI server provided by tutor without changes.

search.py

It is the python program which is used to call the SearchServer.java to complete the search process. It is provided by the tutor and remain unchanged.

index.html

It is the webpage that is shown when the CGI server is started and accessed by client which allows users to input a keyword to search the result. It is provided by the tutor without modification.