International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Result Grabbing and its Analysis using Data Analytics

Puru Agrawal[1], Rajesh Deshmukh[2], Monika Gehi[3]

[1]CSE-8TH Semester,
Shri Shankaracharya Institute of Professional Management and Technology, Raipur
puru.agrawal@ssipmt.com

[2]Assistant Professor, Computer Science & Engineering Department,
Shri Shankaracharya Institute of Professional Management and Technology, Raipur
my.mail.raj@gmail.com

[3]CSE-8TH Semester,
Shri Shankaracharya Institute of Professional Management and Technology, Raipur
monika.gehi@ssipmt.com

Abstract: ***Data analytics is the science of examining raw data with the purpose of drawing conclusions about that information. In this paper we describe the use of data analytics and custom-made tool for grabbing the results of engineering students from the university website. This data is then used for calculating various required parameters by the college management.***

Keywords: web crawler, result analysis, result grabber, data analytics.

1. Introduction

Data analytics is used in many industries to allow companies and organization to make better business decisions and in the sciences to verify or disprove existing models or theories [1]. Data analytics is a powerful tool for getting useful information from a large set of raw data and has seen applications in many scientific and engineering domains. During the simulation process, the experiments generate very huge amounts of data.. To date, there is no customized software for grabbing the data published online mainly because of the missing of a platform to support applications that involve intensive data access and analytical process. This is due to the fact that every university publishes the results in a different format from other university. In this paper, we present the analysis of the system our team developed in the past few months. The main idea behind this result grabbing tool is to store the data in a relational database management system (DBMS) to take advantage of the declarative query interface (i.e., SQL), data access methods, query processing, and optimization mechanisms of modern DBMSs [2] [3] [4]. The main practical use of this software is in various Engineering colleges that need the result of the students for improving the efficiency and overall result of the colleges. For this, we developed a crawler that indexes through the entire page in which the results are published and captures only that information which are helpful in the result analysis. This data is then saved in the RDBMS and also generated is and Excel Sheet. Other calculations and analysis are performed by the system and the final results are shown to the end user. As a result the colleges are in a much better position to capture and analyze the data. We have developed a prototype of Result Grabbing system based on the Java system and experiments using real results published online by the University for Various Exams. The user only needs to enter the link of the page where the data is published; the starting and the ending roll number for which the results need to be analyzed.

2. Why Use Quantitative Approaches?

Quantitative methods of data analysis can be of great value to the person who is attempting to draw meaningful results from a large body of qualitative data. The main beneficial aspect is that it provides the means to separate out the large number of confounding factors that often obscure the main qualitative findings. Take for example, a study whose main objective is to look at the role of non-wood tree products in livelihood strategies of smallholders. Participatory discussions with a number of focus groups could give rise to a wealth of qualitative information. But the complex nature of inter-relationships between factors such as the marketability of the products, distance from the road, access to markets, percent of income derived from sales, level of women participation, etc., requires some degree of quantification of the data and a subsequent analysis by quantitative methods. Once such quantifiable components of the data are separated, attention can be focused on characteristics that are of a more individualistic qualitative nature. Quantitative analytical approaches also allow the reporting of summary results in numerical terms to be given with a specified degree of confidence. So for example, a statement such says that 45% of households use an unprotected water source for drinking may be enhanced by providing 95% confidence limits for the true proportion using unprotected water as ranging from 42% to 48%. Here it is possible to say with more than 95% confidence that about half the households had no access to a protected water supply, since the confidence interval lies entirely below 50% [5][6][7]. This means that quantitative methods of data analysis helps in analyzing even those data sources which have no well defined structure. In our case of result grabbing, we are not having data in a well-defined format. We are only getting plane text. We thus need to first

use quantitative approach for finding the necessary information from it and then saving the data in a well-defined format for the presentation purposes.

3. Web Crawler

Web crawlers – also known as robots, spiders, worms, walkers, and wanderers – are almost as old as the Web itself. The first crawler, Matthew Gray's Wanderer, was written in the spring of 1993, roughly coinciding with the first release of NCSA Mosaic [Gray]. Several papers about Web crawling were presented at the first two World Wide Web conferences [Eichmann 1994; McBryan 1994; Pinkerton 1994]. However, at the time, the Web was two to three orders of magnitude smaller than it is today, so those systems did not address the scaling problems inherent in a crawl of today's Web. Obviously, all of the popular search engines use crawlers that must scale up to substantial portions of the Web. However, due to the competitive nature of the search engine business, the designs of these crawlers have not been publicly described. There are two notable exceptions: the Google crawler and the Internet Archive crawler. Unfortunately, the descriptions of these crawlers in the literature are too terse to enable reproducibility. The Google search engine is a distributed system that uses multiple machines for crawling [Brin and Page 1998; Google]. The crawler consists of five functional components running in different processes. A URL server process, a scalable extensible Web crawler reads URLs out of a file and forwards them to multiple crawler processes. Each crawler process runs on a different machine, is single-threaded, and uses asynchronous I/O to fetch data from up to 300 Web servers in parallel. The crawlers transmit downloaded pages to a single StoreServer process, which compresses the pages and stores them to disk. The pages are then read back from disk by an indexer process, which extracts links from HTML pages and saves them to a different disk file. A URL resolver process reads the link file; de relativizes the URLs contained therein, and saves the absolute URLs to the disk file that is read by the URL server. Typically, three to four crawler machines are used, so the entire system requires between four and eight machines. The Internet Archive also uses multiple machines to crawl the Web [Burner 1977; Internet. Each crawler process is assigned up to 64 sites to crawl, and no site is assigned to more than one crawler. Each single threaded crawler process reads a list of seed URLs for its assigned sites from disk into per-site queues, and then uses asynchronous I/O to fetch pages from these queues in parallel. Once a page is downloaded, the crawler extracts the links contained in it. If a link refers to the site of the page it was contained in, it is added to the appropriate site queue; otherwise it is logged to disk.

There are five main components of any web crawler [8][9][10]. They are:

- A component (called the URL frontier) for storing the list of URLs to download.
- A component for resolving host names into IP addresses.
- A component for downloading documents using the HTTP protocol.
- A component for extracting links from HTML documents, and
- A component for determining whether a URL has been encountered before.

From the above components of the web crawler, our prototype is developed which uses only the first three components as we are mainly concerned with the contents of the page rather than the number of links in that page.

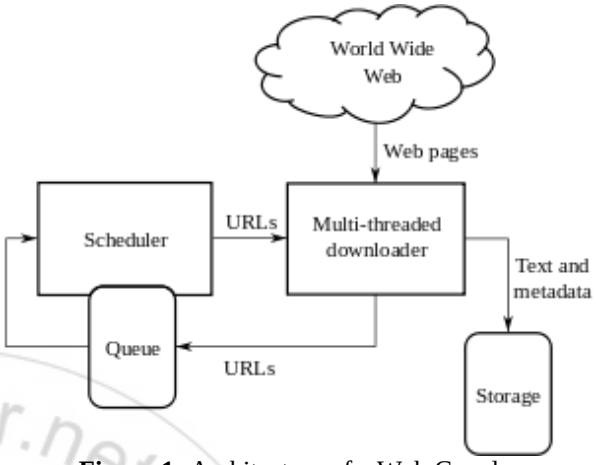

Figure 1: Architecture of a Web Crawler

4. About The Software Prototype

Result Grabbing Tool is a desktop application based on windows platform for downloading result from the website and browse it when offline. This program can quickly download the result from website and saves all the files in the hard drive in their native format. After downloading, all links within the website are reconstructed and creating a complete hard drive copy of the site that you can view at your own place without being connected to the Internet [11]. This project is implemented as a software utility, which can be installed in the system. The utility can be installed in any system with windows, Linux, UNIX or MacOS platform. The main purpose for preparing this document is to give a general insight into the analysis and requirements of the existing system or situation and for determining the operating characteristics of the system.

The main objective of the System is that down load the page or entire site easily with the images and other files as pages for offline viewing. In these files will be saved directly on hard disk for further usage. Downloading result for each student manually is inconvenient, time consuming and frustrating. Result grabbing tool addresses these problems by allowing you to download a quantity of information from web site to your hard drive. Downloading can be done in off peak hours when access lines are available and less expensive. The data may then be edited, browsed, studied, or used in any way offline at your convenience. This makes the use and modification of the information gathered more economical and efficient. It is simple to use and cost effective for anyone working with web content [11] [12].

5. What Is The Need?

Result Grabbing Tool is utility software developed particularly for use in universities and colleges where the result is published online and then the faculties have to invest

a lot of their useful time for manually entering it into large Excel sheets. These steps necessary and storing the results is required because:

- After a fixed duration the results are removed from the website.
- The results are needed to compare the performance of students and calculate certain other valued from it.
- The results are required during various other activities such as Campus drives, Scholarship awards etc.

As we see that keeping the results is an important process for any college, and a lot of time is wasted in manually copying the records, we came up with this idea of grabbing the results using custom made software that grabs the result in the format as given by the CSVTU.

6. Key Features

- This software is built on the Java Programming language and provides platform independent functionality.
- This software is very lightweight software and does not require any special extra hardware for its implementation.
- The results are grabbed, processed and the output is given in the specific format required by the college.
- Automatically skipping the invalid roll numbers.
- Automatically getting the subject names of the group.
- Calculating the number of pass and fail students.
- Analyzing subject-wise result.
- Deleting the unwanted values.
- Grabbing all possible details of the results.

7. Output of the Project

The current prototype generates output in text format which is later converted to Excel sheets and SQL queries. Here is the screenshot of the raw data output from this prototype.

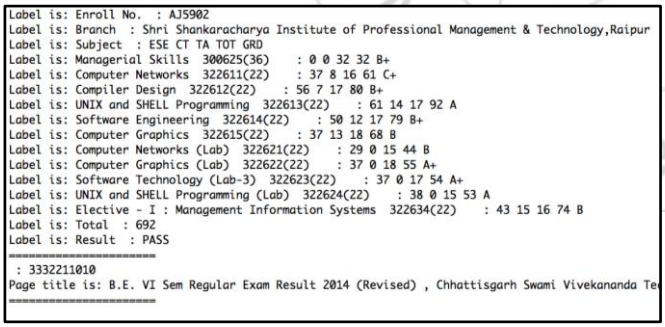

Figure 2: Output in form of Raw Data

8. Future Scope

- Almost every project is subjected to change depending on the client's requirements. The system and the architecture of our software is a compatible one, so addition of new modules can be done without much difficulty.
- Downloading is limited to files of one department; we propose to extend this by downloading for all departments and ultimately for each and every student of university.
- Grabbing the result to your email, could it be attached?
- Web grabbing results could be combined with more software and used more conveniently.
- The grabbing can be modified such that the grabbing process is so speed or very rapid.
- By the increasing interest of the grabbing the complete website at a time, the relevant implementation will be more prosperous in the future [13].

References

[1] http://searchdatamanagement.techtarget.com/definition/data-analytics
[2] http://www.journalofbigdata.com/content/2/1/9/abstract
[3] http://www.authormapper.com/search.aspx?val=journal:Journal+of+Big+Data
[4] http://link.springer.com/article/10.1186/s40537-014-0009-5
[5] http://www.studymode.com/essays/Finance-62076085.html
[6] http://docslide.us/documents/quantitative-analysis-approaches-to-qualitative-data.html
[7] http://www.personal.rdg.ac.uk/~snsabeya/ConfPaperBySavitri.doc
[8] https://archive.org/stream/bitsavers_dectechrep_97036/SRC-RR-173_djvu.txt
[9] http://www.wendangwu.com/doc/content/201101/28/27584659788.html
[10] http://www.ijcsi.org/papers/IJCSI-9-1-1-389-395.pdf
[11] http://www.codewithc.com/web-grabber-asp-net-project
[12] https://www.bluesquirrel.com/products/grabasite
[13] http://freeprojectscode.com/java-projects/web-grabber-project/1062

Author Profiles

Puru Agrawal is a Computer Science & Engineering student currently pursuing his Engineering degree from Shri Shankaracharya Institute of Professional Management & Technology, Raipur. His areas of interests and research include cloud computing, Data Base Management and Android Software development.

Rajesh Deshmukh received Master of Technology in C.S.E. (Honors) from Chhattisgarh Swami Vivekanand Technical University, Bhilai. Currently he is perusing Doctor of Philosophy in CSE from Dr. C.V. Raman University, Bilaspur and working as Assistant Professor in Shri Shankaracharya Institute of Professional Management & Technology (Department of Computer Science Engineering), Raipur, India and has more than 5 years teaching experience. He has published and presented more than 18 research papers in various National / International Journals and Conferences. He is CSI Student Chapter - Raipur Division IV Coordinator and also an active Member of various Professional Societies like CSI, IEEE, ACM, IACSIT, IAENG, SDIWC, IWA and CSTA. He is Member of Reviewer Panel in IJCSI, IJSER, IJERT, JETR and IJTEL. He has Authored One Book based on his research work with German Publisher. His area of research includes Mobile Ad hoc Network Routing Protocols, Wireless Sensor Networks, Distributed System, Cloud Computing and Operating System.

Monika Gehi is a Computer Science & Engineering student currently pursuing his Engineering degree from Shri Shankaracharya Institute of Professional Management & Technology, Raipur. Her areas of interests and research include cloud computing, Big Data and Data Analytics.