Paper from Proceeding of the National Conference "Science in Media 2012" Organized by YMCA University of Science and Technology, Faridabad, Haryana (India) December 3rd -4th 2012 Published by IJMRS's International Journal of Engineering Sciences, ISSN (Online): 2277-9698
www.ijmrs.com

# Issues & challenges in effective design of search engines

Ashlesha Gupta and Ashutosh Dixit, YMCA Univ. of Sc. & Tech., Faridabad

***Abstract— With vast expansion of Web as Information resource, extracting knowledge from the Web is becoming difficult. To find Web pages, one typically uses search engines that are based on the Web crawling framework. This paper describes the basic architecture, and components of search engine. Major issues and challenges in implementing effective Web crawlers are also identified. An insight into the next generation search engines is being provided.***

## I. INTRODUCTION

WWW is a vast resource of hyperlinked and heterogeneous information including text, audio, video, image etc. that continues to grow rapidly at million pages per day. With rapid increase in information resources available via WWW and users of the Internet, it is becoming difficult to manage and access the desired information on the web. Therefore, majority of users use information retrieval tools like search engines to find the desired information from the WWW. A Search Engine is an information retrieval system which helps users finds information on WWW by making the web pages related to their query available. With a search engine, users have to type in "keywords" relating to the information that they need. The search engine would then return a set of results that match best with the keywords entered.

Many people believe that by using these information retrieval tools they can easily find the information on the topic they are looking for on the Web. However, many Web information services deliver inconsistent, inaccurate, incomplete, and often irrelevant results. For many reasons, existing Web search techniques have significant deficiencies with respect to robustness, flexibility, and precision. Since the first Web information services were based on traditional information retrieval (IR) algorithms and techniques. However, most IR algorithms were developed for smaller, more coherent collections than what the Web has become: today's Web searching requires new techniques.

This article offers an overview of search-engine architectures, Crawler and its types and discusses problems search engines face in indexing the web in maintaining or enhancing search-engine performance quality.

## II.SEARCH ENGINE ARCHITECTURE

A Search Engine is an information retrieval system which helps users finds information on WWW by making the web pages related to their query available. With a search engine, users have to type in "keywords" relating to the information that they need. The search engine would then return a set of results that match best with the keywords entered. A Web Search Engine can therefore be defined as a software program that takes input from the user, searches its database and returns a set of results. It is important to note that the search engine does not search the internet; rather it searches its database, which is populated with data from the internet by its crawler(s). Web search engines work by storing information about many web pages, which they retrieve from the WWW itself. These pages are retrieved by a Web crawler which follows every link it sees. Exclusions can be made by the use of robots.txt. The contents of each page are then analyzed to determine how it should be indexed. Data about web pages are stored in an index database for use in later queries. The typical architecture of a search engine is [7]:

The major components of search engine are Crawler, Indexer and Query processor. A crawler traverses the web by following hyperlinks and storing

downloaded pages in a large database. It starts with seed URL and collects documents by recursively fetching links and storing the extracted URL's into a local repository. The Indexer processes and indexes the pages collected by the crawler. It extracts keywords from each page and records the URL where each word has occurred. The query processor is responsible for receiving and filling search requests from user. The query processor processes user queries and returns matching answers in an order determined by a ranking algorithm.

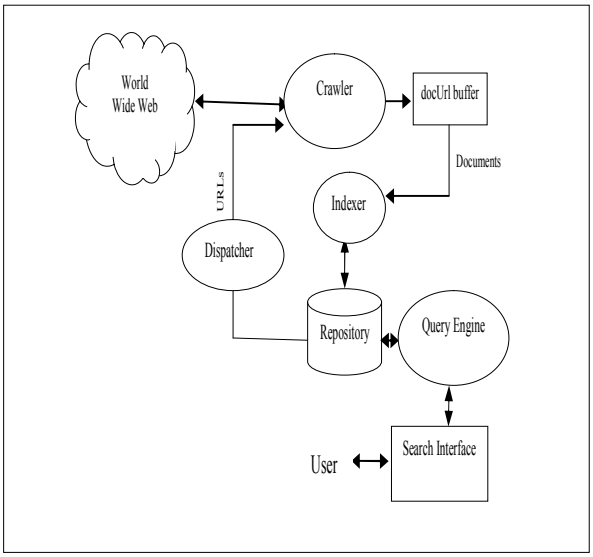


Figure 1 : Architecture of Search Engine

## III. CRAWLER

A crawler is a program that downloads pages and stores them in repository maintained at search engine side. The objective of crawling is to quickly and efficiently gather as many useful web pages as possible, together with the link structure that interconnects them.

The basic operation of any hypertext crawler is as follows. The crawler begins with one or more URLs that institute a seed set. It picks a URL from this seed set, and then fetches the web page at that URL. The fetched page is then parsed, to extract both the text and the links from the page (each of which points to another URL). The extracted text is fed to a text indexer. The extracted links (URLs) are then added to a URL queue, which at all times consists of URLs whose corresponding pages have yet to be fetched by the crawler. The entire process may be viewed as traversing the web graph. Figure2 shows the flowchart for the working of a typical crawler.

As shown in Figure 2 Robot.txt files carries downloading permissions and also specifies the files to be excluded by the crawler.

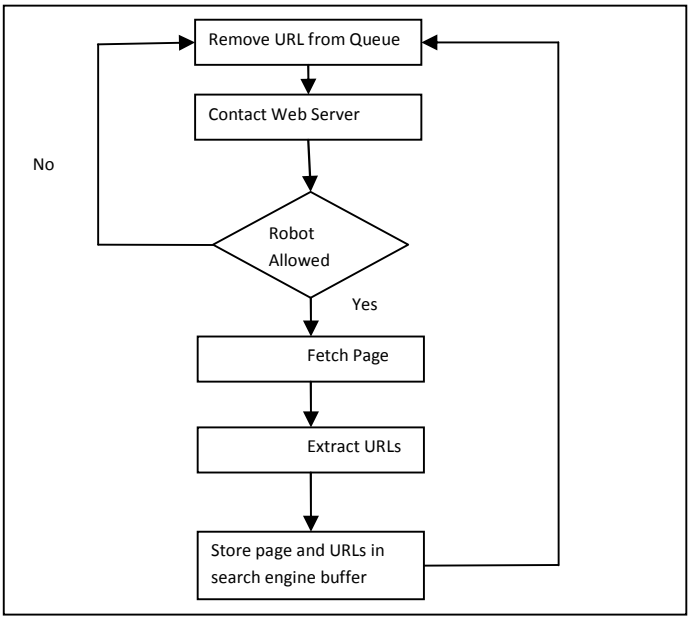


Figure 2 : Flowchart for working of a typical crawler

Based on their page retrieval and documents refreshing techniques crawlers can be divided into several categories

- Focused Crawler: A focused crawler or topical crawler is a web crawler that attempts to download only web pages that are relevant to a pre-defined topic. The goal is to select links that lead to documents of

interest, while avoiding links that crawler uses an additional classifier to select most promising links on a relevant page and the crawling cycle starts with a seed list which contains URLs that are relevant to the topic of interest. The main components of a focused crawler are Classifier, Distiller and download workers. Classifier makes relevance judgments on pages crawled and takes decision whether to expand links or not. Distiller determines a measure of centrality of crawled pages to determine visit priorities.

- Parallel Crawlers: In a parallel crawler multiple processes run in parallel to perform the downloading task, so that download rate is maximized. A parallel crawler consists of multiple crawling processes referred as C-Proc. Each C-Proc performs the basic tasks that a single process crawler conducts. A parallel crawler may be categorized as Intra-site crawler, where all processes run on the same local network and communicate through high speed interconnect such as LAN and Distributed Crawler where processes run at geographically distant locations connected by the Internet. Coordination among processes may be achieved in three ways: Independent (no coordination), Dynamic assignment (central coordinator divide the Web into small partitions and dynamically assigns each partition to a C-Proc for download), Static assignment (Web is partioned and assign-each c-Proc before the start of a crawl.
- Migrating Crawlers: In migrating crawlers migrants move to the data sources i.e. servers before the actual crawling process is started. After performing all crawling tasks, migrant may move on the next server or may return to the originating node. The main advantage of this approach is that it allows us to distribute crawling functionality with in a distributed system and hence reduce the network load.

## IV. ISSUES AND CHALLENGES IN WEB SEARCH ENGINES

In order to download the large number of web pages from the web, a highly efficient crawling system is needed. The enormous size of the web coupled with the dynamic nature of documents poses following issues towards design of an efficient crawling system.

1. How to crawl best pages: Since a crawler can download only some part of the web at any time, it must be biased towards downloading important pages first. search engines *should not* index the entire Web. An ideal search engine should know all the pages of the Web, but there are contents such as duplicates or spam pages that should not be indexed.

2. Overlapping of web documents: Overlap problem occurs when multiple crawlers running in parallel download the same web document multiple times due to the reason that one web crawler may not be aware of another having already downloaded the same page. Also many organizations mirror their documents on multiple servers to avoid arbitrary server corruption. In such a situation, crawlers may also unnecessarily download many copies of the same document. Moreover to improve the quality of downloaded web documents, multiple crawlers running in parallel must have global image of collectively downloaded web pages so that redundancy may be avoided

3. Network bandwidth/traffic problem: In order to maintain the quality, the crawling process is carried out using either of the following approaches: Crawlers can be generously allowed to communicate among themselves or they cannot be allowed to communicate among themselves at all.

In the first approach network traffic will increase because crawlers communicate among themselves more frequently to reduce the overlap problem whereas in second approach, if they are not allowed at all to communicate then as a result same web documents may be downloaded multiple times thereby consuming the network bandwidth. Thus both approaches put extra burden on network traffic.

4. Change of web documents: Changing of web documents is a continuous process. Of course, the frequency of change varies from document to document. Search engines should not only focus on the sizes of their indices, but also on their up-to-dateness. This change must be reflected at the search engine repository failing which a user may get an obsolete image of the web documents. Search engines face problems in keeping up to date with the entire Web, and because of its enormous size and the different update cycles of individual websites, adequate crawling strategies are needed.
5. Web Content: Web documents differ significantly from documents in traditional information systems On the Web, documents are written in many different languages, whilst other information systems usually cover only one or a few selected languages. Documents are indexed using a controlled vocabulary, which allows it to search for documents written in different languages with just one query.
   Another difference is the use of many different file types on the Web. Search engines today not only index documents written in HTML, but also PDF, Word, or other Office files. Each file format provides certain difficulties for the search engines. In the overall ranking, all file formats have to be considered. There are some characteristics, which often coincide with certain file formats, such as the length of PDF files, which are often longer than documents written in HTML. The length of documents on the Web varies from just a few words to very long documents. This has to be considered in the rankings.

   Another problem is the documents structure. HTML and other typical Web-documents are just vaguely structured. There is no field structure similar to traditional information systems, which makes it a lot more difficult to allow for exact search queries.

6. Spam: Everyone knows that spam is a problem from his or her own e-mail account. Like with e-mail accounts, spammers try to flood search engine indices with their contents. It is very important for search engines to filter these pages to keep their indices clean and keep a good quality of their results.
7. Temporal Quality of downloaded web documents: The quality of downloaded documents can be ensured only when web pages of high relevance are downloaded by the crawlers but the crawlers today are not capable of automatically tracking the user trends or the topics of current interest. Therefore, a crawler should be capable to automatically track the current trend topics and download web pages that meet user's current need.
8. Almost all available crawlers provide the parallelism through running the parallel crawling instances on the single machine. This imposes overload not only on the host machine but also leads to network congestion problem. Therefore there is a need for a crawler to avoid network load and congestion by distributing the load among the servers.

## V. NEXT GENERATION WEB

Due to the Web's continued growth, today's Web searches require new techniques- exploiting or extending linkages among web pages. A lot of research is going on in developing new web page retrieval, indexing and ranking techniques. Intelligent crawling is the need of the future which can be achieved using semantic search.

Semantic search aims to extend and improve traditional search processes based on IR technology. These intelligent search engines incorporates Web semantics and uses more advanced search techniques based on concepts such as machine learning. These approaches enable intelligent Web information services, personalized Web sites, and semantically empowered search engines.

## VI. CONCLUSION

This paper describes general architecture of a search engine along with its major components is given. A detailed discussion
of web crawlers and their types is also discussed.
Although the web is a huge repository of information but it lacks quality control. Step toward a Semantic Web are steps toward intelligent searching and support the vision of the next generation search engines.

## VII. REFERNCES

1. Monica Peshave and Kamyar Dezhgosha, "How Search Engines Work and a Web Crawler Application". Department of Computer Science, University of Illinois, Springfield USA.
2. Marios D. Dikaiakos, Athena Stassopoulou, and Loizos Papageorgiou, An investigation of web crawler behavior: Characterization and metrics, by Computer Communications 28 (2005), 880–897.
3. Stefan Büttcher, Charles L. A. Clarke, and Gordon V. Cormack. Information Retrieval: Implementing and Evaluating Search Engines. MIT Press, Cambridge, Mass., 2010.
4. L. Page, S. Brin, R. Motwani and T. Winograd, The pageRank citation ranking: bringing order to the web, Technical Report, Stanford InfoLab, 1998.
5. Dirk Lewandowski "Web searching, search engines and Information Retrieval, Information Services & Use 25(2005)3
6. Neelam Duhan, A. K. Sharma, Komal Kumar Bhatia " Page Ranking Algorithms: A Survey " In IEEE International Advance Computing Conference (IACC 2009) Patiala, India, 6-7 March 2009
7. Ashutosh Dixit, A.K. Sharma, Ashlesha Gupat " Prospective Term based Mathematical model for Page Ranking",ACM International Conference and workshop on Emerging Trends in Technology, ICWET 2012, Feb 24-25, 2012 TCET Kandawali (E) ,Mumbai
8. Allan Heydon and Marc Najork. Mercator: A scalable, extensible web crawler. World Wide Web, 2(4):219–229, 1999.
9. http://www.google.com/technology/index.html, Our Search; Google Technology
10. Douglas E. Comer, "The Internet Book", Prentice Hall of India, New Delhi,2009