"CLOUD BASED SEARCH ENGINE"

A

Project Report

submitted in partial fulfillment of the

requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

by

Name	Roll No.
Mayank Sharma	R970213029
Sushant Pal	R970213037
Deepank Dixit	R970213019

under the guidance of

Dr. Kingshuk Srivastava

Department of Computer Science & Engineering

Centre for Information Technology

University of Petroleum & Energy Studies

Bidholi, Via Prem Nagar, Dehradun, UK

October 2016

CANDIDATE'S DECLARATION

I/We hereby certify that the project work entitled **"Analysis of Inference algorithm**" in partial fulfilment of the requirements for the award of the Degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING with specialization in Oil and Gas Informatics and submitted to the Department of Computer Science & Engineering at Center for Information Technology, University of Petroleum & Energy Studies, Dehradun, is an authentic record of our work carried out during a period from **August**, **2016** to **November**, **2016** under the supervision of **Dr. Kingshuk Srivastava, Assistant Professor**.

The matter presented in this project has not been submitted by us for the award of any other degree of this or any other University.

Mayank Sharma R970213029
Sushant Pal R970213037
Deepank Dixit R970213019

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: 25/11/2016

Dr. Kingshuk Shrivasta
Project Guide

Dr. M.Venkatadri
Program Head – B-tech Cse(Oil and Gas Informatics)
Center for Information Technology
University of Petroleum & Energy Studies
Dehradun – 248 001 (Uttarakhand)

ACKNOWLEDGEMENT

We wish to express our deep gratitude to our guide **Dr. Kingshuk Srivastava**, for all advice, encouragement and constant support he has given us throughout our project work. This work would not have been possible without his support and valuable suggestions.

We sincerely thank our respected Program Head of the Department, **Dr. M.Venkatadri**, for his great support in doing our project in **Area** at **CIT**.

We are also grateful to **Dr. Maneesh Prateek, Associate Dean** and **Dr. Kamal Bansal Dean CoES**, UPES for giving us the necessary facilities to carry out our project work successfully.

We would like to thank all our **friends** for their help and constructive criticism during our project work. Finally we have no words to express our sincere gratitude to our **parents** who have shown us this world and for every support they have given us.

Name	**Mayank Sharma**	**Sushant Pal**	**Deepank Dixit**
Roll No.	**R970213029**	**R970213037**	**R970213019**

ABSTRACT

A Search engine is a software system that is designed to search for information on the entire Web. The search results are generally presented in a line of results often referred to as search engine results pages.

The information may be a mix of web pages, images, and other types of files. Some search engines also mine data available in databases or open directories. Search engines also maintain real-time Information by running an algorithm on a web crawler.

Search engines get their information by web crawling from site to site. The web crawler sends certain information back, to be indexed depending on many factors, such as the titles, page content, JavaScript, Cascading Style Sheets (CSS), headings, as evidenced by the standard HTML markup of the informational content, or its metadata in HTML meta tags.

Web crawling is a straightforward process of visiting all sites on a systematic basis.

Also, this project proposes to develop a search algorithm for efficient, quick and accurate search results with systematic indexing.

Indexing means associating words and other definable tokens found on web pages to their domain names and HTML-based fields.

The associations are made in a public database, made available for web search queries. A query from a user can be a single word. The index helps find information relating to the query as quickly as possible.

Keywords: search engine, web crawler, indexing

TABLE OF CONTENTS

1. Introduction

The web creates new challenges for information retrieval. The amount of information on the web is growing rapidly, as well as the number of new users inexperienced in the art of web research. People are likely to surf the web using its link graph, often starting with high quality human maintained indices such as Yahoo! or with search engines. Human maintained lists cover popular topics effectively but are subjective, expensive to build and maintain, slow to improve, and cannot cover all esoteric topics. Automated search engines that rely on keyword matching usually return too many low quality matches. To make matters worse, some advertisers attempt to gain people's attention by taking measures meant to mislead automated search engines. We have built a large-scale search engine which addresses many of the problems of existing systems. It makes especially heavy use of the additional structure present in hypertext to provide much higher quality search results. We chose our system name, Google, because it is a common spelling of googol, or 10100 and fits well with our goal of building very large-scale search engines.

Web Search Engines -- Scaling Up: 1994 - 2000

Search engine technology has had to scale dramatically to keep up with the growth of the web. In 1994, one of the first web search engines, the World Wide Web Worm (WWWW) [McBryan 94] had an index of 110,000 web pages and web accessible documents. As of November, 1997, the top search engines claim to index from 2 million (WebCrawler) to 100 million web documents (from Search Engine Watch). It is foreseeable that by the year 2000, a comprehensive index of the Web will contain over a billion documents. At the same time, the number of queries search engines handle has grown incredibly too. In March and April 1994, the World Wide Web Worm received an average of about 1500 queries per day. In November 1997, Altavista claimed it handled roughly 20 million queries per day. With the increasing number of users on the web, and automated systems which query search engines, it is likely that top search engines will handle hundreds of millions of queries per day by the year 2000. The goal of our system is to address many of the problems, both in quality and scalability, introduced by scaling search engine technology to such extraordinary numbers.

1.1Requirement Analysis

Software Requirements:

OS: Windows, Linux

Browser: Google Chrome, Mozilla, Internet Explorer, Opera

Hardware Requirements:

Minimum 1GHz Processor

Minimum 4 GB RAM

Minimum 512 MB Hard-disk space

2. Working scenario

2.1 Literature Review

Aside from tremendous growth, the Web has also become increasingly commercial over time. In 1993, 1.5% of web servers were on .com domains. This number grew to over 60% in 1997. At the same time, search engines have migrated from the academic domain to the commercial. Up until now most search engine development has gone on at companies with little publication of technical details. This causes search engine technology to remain largely a black art and to be advertising oriented. Google, has a strong goal to push more development and understanding into the academic realm [1].

Another important design goal was to build systems that reasonable numbers of people can actually use. Usage was important to us because we think some of the most interesting research will involve leveraging the vast amount of usage data that is available from modern web systems. For example, there are many tens of millions of searches performed every day. However, it is very difficult to get this data, mainly because it is considered commercially valuable [3].

To build an architecture that can support novel research activities on large-scale web data. To support novel research uses, Google stores all of the actual documents it crawls in compressed form. One the main goals in designing Google was to set up an environment where other researchers can come in quickly, process large chunks of the web, and produce interesting results that would have been very difficult to produce otherwise [3].

In the short time the system has been up, there have already been several papers using databases generated by Google, and many others are underway. Another goal is to set up a Spacelab-like environment where researchers or even students can propose and do interesting experiments on our large-scale web data [5].

In 1994, some people believed that a complete search index would make it possible to find anything easily. According to Best of the Web 1994 Navigators, "The best navigation service should make it easy to find almost anything on the Web (once all the data is entered)." However, the Web of 1997 is quite different. Anyone who has used a search engine recently, can readily testify that the completeness of the index is not the only factor in the quality of search results [2].

"Junk results" often wash out any results that a user is interested in. In fact, as on November 1997, only one of the top four commercial search engines finds itself (returns its own search page in response to its name in the top ten results). One of the main causes of this problem is that the number of documents in the indices has been increasing by many orders of magnitude, but the user's ability to look at documents has not. People are still only willing to look at the first few tens of results. Due to this, as the collection size grows, we need tools that have very high precision (number of relevant documents returned, say in the top tens of results). Indeed, we want our notion of "relevant" to only include the very best documents since there may be tens of thousands of slightly relevant documents. This very high precision is important even at the expense of recall (the total number of relevant documents the system is able to return). There is quite a bit of recent optimism that the use of more hypertextual information can help improve search and other applications. In particular, link structure and link text provide a lot of information for making relevance judgments and quality filtering [4].

2.2 Problem Statement

Non relevant search results which reduce the accuracy and efficiency of the search engine and increases the user’s time to obtain the desired result.

As the collection size grows, tools are needed that have very high precision (number of relevant documents returned, say in the top tens of results). Indeed, the notion of "relevant" to only include the very best documents since there may be tens of thousands of slightly relevant documents. This very high precision is important even at the expense of recall (the total number of relevant documents the system is able to return). There is quite a bit of recent optimism that the use of more hypertextual information can help improve search and other applications

2.3 Objective

To develop a search engine and its algorithm for efficient, quick and accurate search results with systematic indexing.

2.4 Methodology

Step 1:

Gather relevant data by crawling using a web crawler

Step 2:

Store the acquired data into a database

Step 3:

Develop and implement an algorithm for searching

Step 4:

Migration over cloud platform

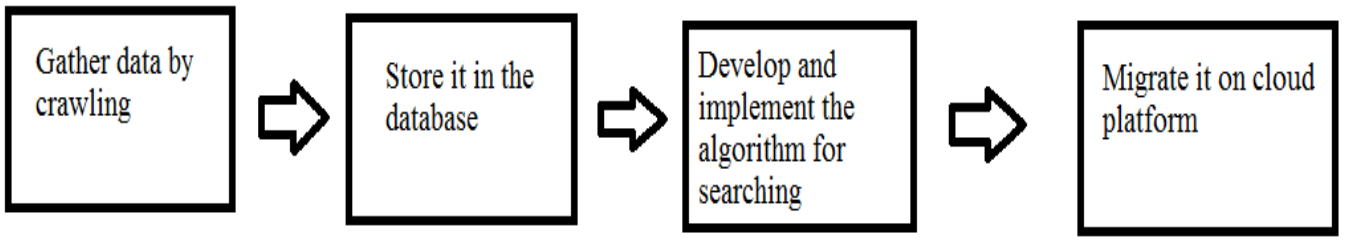

Fig 2.1

2.5 Architecture

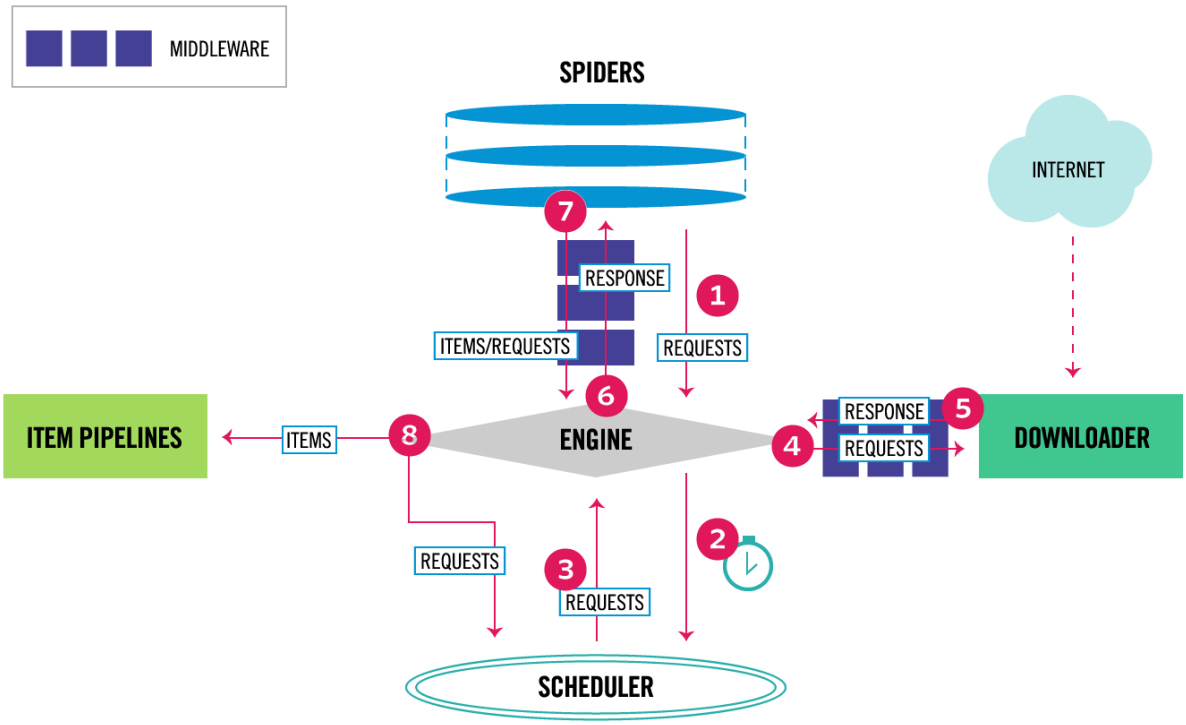

Fig 2.2

```
askUbuntu/
    crawler.cfg           # deploy configuration file
    askUbuntu/            # project's Python module, you'll import your code from here
        __init__.py
         Model.py
        items.py          # project items definition file
        pipelines.py      # project pipelines file
        settings.py       # project settings file
        spiders/          # a directory where you'll later put your spiders
            __init__.py
            ubuntu.py
```

References

- [Abiteboul 97] Serge Abiteboul and Victor Vianu, *Queries and Computation on the Web*. Proceedings of the International Conference on Database Theory. Delphi, Greece 1997.
- [Bagdikian 97] Ben H. Bagdikian. *The Media Monopoly*. 5th Edition. Publisher: Beacon, ISBN: 0807061557
- [Chakrabarti 98] S.Chakrabarti, B.Dom, D.Gibson, J.Kleinberg, P. Raghavan and S. Rajagopalan. *Automatic Resource Compilation by Analyzing Hyperlink Structure and Associated Text.* Seventh International Web Conference (WWW 98). Brisbane, Australia, April 14-18, 1998.
- [Cho 98] Junghoo Cho, Hector Garcia-Molina, Lawrence Page. *Efficient Crawling Through URL Ordering.* Seventh International Web Conference (WWW 98). Brisbane, Australia, April 14-18, 1998.
- [Gravano 94] Luis Gravano, Hector Garcia-Molina, and A. Tomasic. *The Effectiveness of GlOSS for the Text-Database Discovery Problem.* Proc. of the 1994 ACM SIGMOD International Conference On Management Of Data, 1994.
- [Kleinberg 98] Jon Kleinberg, *Authoritative Sources in a Hyperlinked Environment*, Proc. ACM-SIAM Symposium on Discrete Algorithms, 1998.
- [Marchiori 97] Massimo Marchiori. *The Quest for Correct Information on the Web: Hyper Search Engines.* The Sixth International WWW Conference (WWW 97). Santa Clara, USA, April 7-11, 1997.
- [McBryan 94] Oliver A. McBryan. GENVL and *WWWW: Tools for Taming the Web. First International Conference on the World Wide Web.* CERN, Geneva (Switzerland), May 25-26-27 1994.
- [Page 98] Lawrence Page, Sergey Brin, Rajeev Motwani, Terry Winograd. *The PageRank Citation Ranking: Bringing Order to the Web.* Manuscript in progress.
- [Pinkerton 94] Brian Pinkerton, *Finding What People Want: Experiences with the WebCrawler.* The Second International WWW Conference Chicago, USA, October 17-20, 1994.
- [Spertus 97] Ellen Spertus. *ParaSite: Mining Structural Information on the Web.* The Sixth International WWW Conference (WWW 97). Santa Clara, USA, April 7-11, 1997.