Volume 6, Issue 8, August 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Web Documents Indexing Techniques- A Review

Huma Siddiqui
Student of M.Tech, Department of CSE, MIET, Meerut, Uttar Pradesh, India

Dr. Nidhi Tyagi
Associate Professor, Department of CSE, MIET, Meerut, Uttar Pradesh, India

Abstract- ***Search engines has a particular agents called web crawler which crawls web documents online, and they are analyzed and indexed and made available to user. Since, users have scarcity of time and thus want to access the relevant data accurately and timely. System in terms of time and accuracy more efficient indexing should be used. This can be achieved by adopting MapReduce approach for indexing the documents.***

Keywords- Search engine, Web Crawler, Indexing techniques, MapReduce, Indexing through MapReduce.

I. INTRODUCTION

The largest repository of hyperlinked web documents is WWW which is accessible through the internet. The tools for extracting and exploring specific information on the web are search engines. As technology change and rapidly increases, internet has become the main source for information exchange and also one of the most important media of information research. The exponential increase in the data on web requires efficient searching technique. Search engines simplify the task of searching for the huge WWW. Fig.1 shows the general architecture of search engine.

Main components of search engine are: web crawler, indexer, query indexer.

Web Crawler

The crawler is also called spider that parses the web collecting information and stores them into a huge repository [1]. It assembles a corpus of web pages, index them, and allow users to make queries against the index and also find the web pages that match the queries.

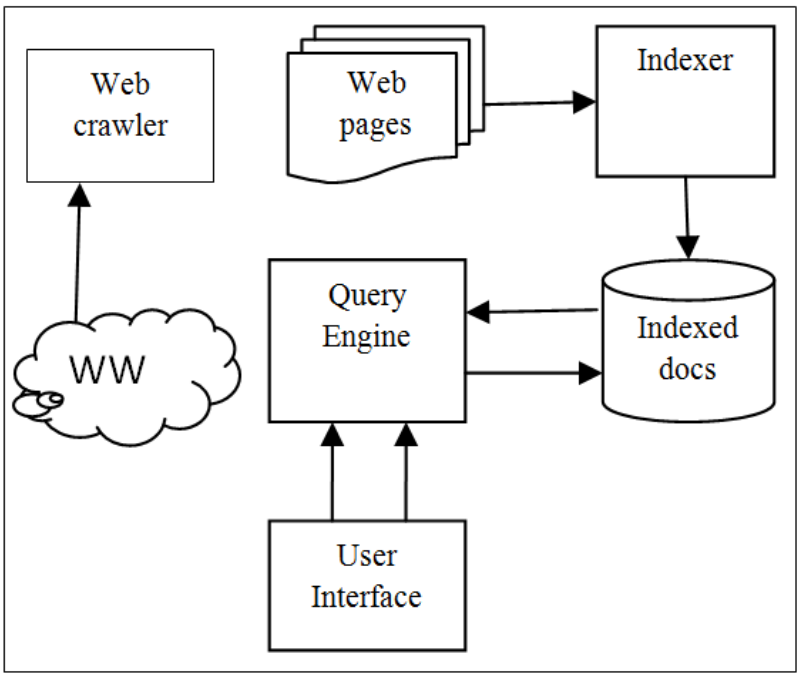

Figure1: General Architecture of Search Engine

Indexer

The process of systematic arrangement of records is called indexing. User can find information more quickly by indexing. This module prepares the index of the local database. Every Web document has an associated ID number called document identifier, and it is the local database. And it is assigned whenever a new URL is parsed out of a web page [1]. The web pages are collected by the crawlers and indexer takes and stores them into a well-organized index. Fig. 2 shows the indexing process [2].

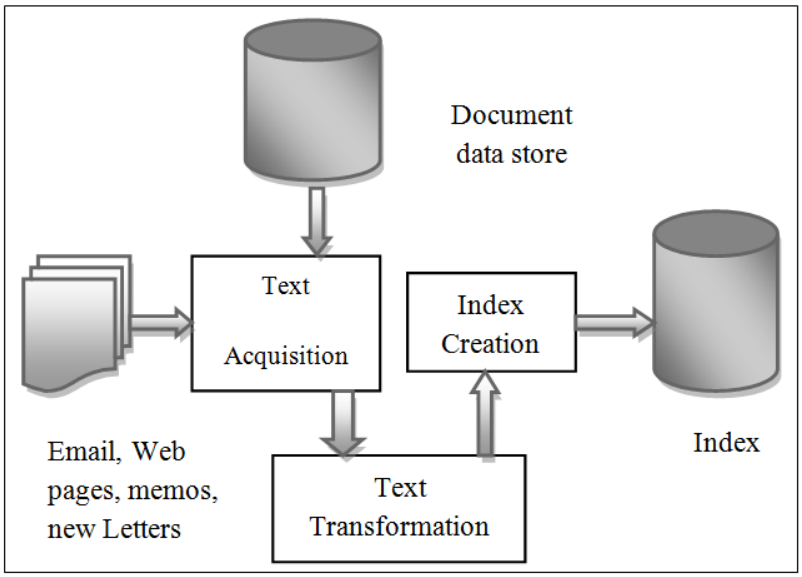

Fig 2: Indexing Process

- **Text acquisition**: It identifies and making available the documents which will be searched by crawler or scanning the web.
- **Text transformation:** It transforms text into index terms and features.
- **Document data stores**: It manages the large numbers of documents and structured data. Document components are typically stored in a compressed form for efficiency. Structured data consists of document metadata and other information extracted from the documents such as links and anchor text.
- **Index creation:** It gathers or record information about words, features, documents.

Query engine

This module handles the user queries by searching the index. The queries entered by user are in the form of keywords and search engine matches their keywords in the indexer. The heart of search engine is an index module which increases the speed of searching.

II. INDEXING TECHNIQUES

Different indexing approaches have been discussed below:

- **Suffix tree**: Suffix tree has figuratively structured like a tree which produce the index of all keywords of a document. Built by storing the suffixes of words. The suffix tree is a type of trie. Tries support extendable hashing, which is important for search engine indexing [3].
- **Inverted index:** It is created by trie generator. Each entry of inverted index is a data structure which stores the keyword, meanings and corresponding document ids. By a tree searching algorithm, this inverted index can be explored to acquire a match for a particular query keyword in search-insert fashion.
- **Citation index:** It Stores citations or hyperlinks between documents to support citation analysis. It symbolizes the conceptual organization of scientific ideas as recognized by publishing research authors [5].
- **N-gram index**: It Stores sequences of length of data to support other types of retrieval or text mining. e.g. words in document or words in phrase. These n-grams are used within text operation or text comparison [4].

From the critical look at the above indexing techniques following problem have been identified.

- The existing system depends only on keywords matching.
- Relevant result does not found.
- A major drawback of citation index is that storing a word in the tree may require more space for its storage [6].

To overcome these problems, an efficient indexing technique is required, which can optimize the searching task in terms of time and accuracy. The technique of MapReduce can be used for indexing the web documents, for better results.

III. MAPREDUCE

MapReduce is a programming paradigm for the processing of huge amounts of data by distributing work tasks over multiple processing machines [7]. A MapReduce job generally splits the input data-set into independent chunks which are processed by the map tasks in a completely parallel manner. The framework sorts the outputs of the maps, which are then input to the reduce tasks. Typically both the input and the output of the job are stored in a file-system. The major process of MapReduce is shown in figure 3.

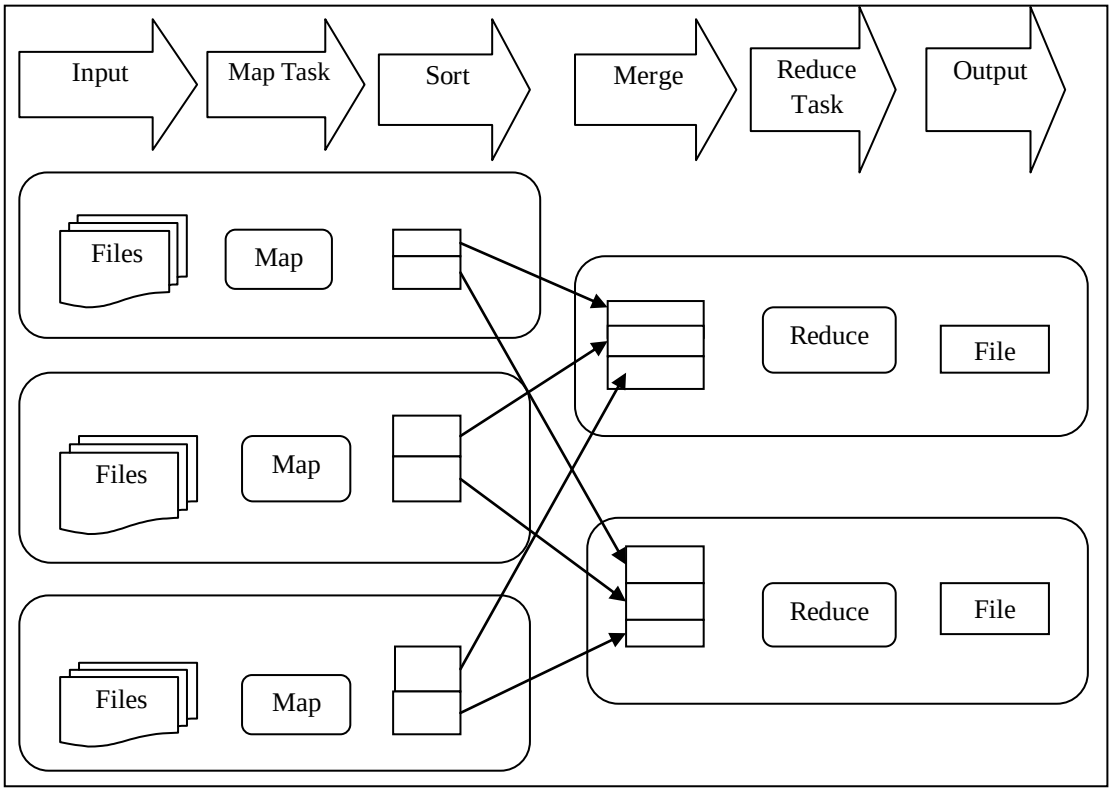

Figure 3: MapReduce Technique

MapReduce has designed from a functional program, where functions provide definitions of operations over input data. A single MapReduce job is defined by the user as two functions. The map function takes in a key/value pair (of type <key1, value1>) and produces a set of intermediate key/value pairs (<key2, value2>). The outputs from the map function are then automatically sorted by their key (key2). The reduce function collects the map output and then merges the values with the same key to form a smaller final result.

IV. INDEXING THROUGH MAPREDUCE

The process of indexing through MapReduce is explained here.

A. MapReduce indexing by Dean and Ghemawat

"The map function parses each document, and emits a sequence of <word, document ID> pairs. The reduce function accepts all pairs for a given word, sorts the corresponding document-IDs and emits a <word list(document ID)> pair. The set of all output pairs forms a simple inverted index. It is easy to expand this computation to keep track of word positions [7].''

B. MapReduce indexing by Nutch

In this strategy instead of emitting terms, it only tokenises the document during the map phase, hence emitting <doc-ID, Document> tuples from the map function. Each document contains the textual forms of each term and their corresponding frequencies. The reduce phase is then responsible for writing all index structures. As one emit is made per-document so, it is called per document indexing. This indexing strategy will emit less, but the value of each emit will contain substantially more data (i.e. the textual form and frequency of each unique term in the document) [7].

C. Per-posting list indexing (Terrier)

The indexing process is split into multiple map tasks. Each map task operates on its own subset of the data and documents are processed by each map task, compressed posting lists are built in memory for each term. However, when memory runs low or all documents for that map have been processed, the partial index is flushed from the map task, by emitting a set of <term, posting list> pairs for all terms currently in memory. These flushed partial indices are then sorted by term, map and flush numbers before being passed to a reduce task. As the flushes are collected at an appropriate reduce task, the posting lists for each term are merged by map number and flush number, to ensure that the posting lists for each term are in a globally correct ordering. The reduce function takes each term in turn and merges the posting lists for that term into the full posting list, as a standard index. Elias-Gamma compression is used as in non-distributed indexing to store only the distance between doc-IDs [7].

V. CONCLUSION

This paper describes various techniques available for indexing the web documents. It also discusses the method of MapReduce that can give more optimal result if applied on a search engine. MapReduce have the ability to perform code

optimization for easy implementation through its program, means that an execution in MapReduce will be cheaper to produce and maintain.

ACKNOWLEDGEMENT

I take this opportunity to thank all in individuals for their guidance, help and timely support .It gives me great pleasure and immense satisfaction to present this paper. Which result of unwavering support, expert guidance and focused direction of my guide Dr. Nidhi Tyagi to whom I express my deep sense of gratitude and humble thanks, for valuable guidance throughout the work.

REFERENCES

[1] Priyanka S. Zaware, Satish R. Todmal "Inverted Indexing Mechanism for Search Engine", International journal of Computer Application (0975-8887) Volume 123 – No.17, August 2015.

[2] http://www.cs.sfu.ca/CourseCentral/456/jpei: Information Retrieval and Web Search-Search Engine Architecture.

[3] http://en.wikipedia.org/wiki/Search_engine_indexing.

[4] Daniel Robenek, Jan Platos, and Vaclav Snasel "E_cient in-memory data structures for n-grams indexing", Dateso, pp. 48{58, ISBN 978-80-248-2968-5, 2013

[5] https://faculty.ist.psu.edu/giles/IST497/presentations/Mathew.ppt

[6] Gusfield, Dan" Algorithms on Strings, Trees and Sequences": Computer Science and Computational Biology. USA: Cambridge University Press 1997. *ISBN 0-521-58519-8* ,1997

[7] Richard McCreadie , Craig Macdonald, Iadh Ounis "MapReduce indexing strategies: Studying scalability and efficiency" Information Processing and Management, Elsevier doi:10.1016/j.ipm.2010.12.003, 2011

