iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** IV **Month of publication:** April 2015

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Web Crawler Used in Search Engine

Harshali Kshirsagar[1], Pratibha Rewaskar[2], Komal Ramteke[3]
Computer Science and Engineering Department, SGBAU University

Abstract-- The World Wide Web (WWW) is a collection of billions of documents formatted using HTML. Web Search engines are used to find the desired information on the World Wide Web. Whenever a user query is inputted, searching is performed through that database. The size of repository of search engine is not enough to accommodate every page available on the web. So it is desired that only the most relevant pages must be stored in the database. So, to store those most relevant pages from the World Wide Web, a better approach has to be followed. The software that traverses web for getting the relevant pages is called "Crawlers" or "Spiders". A specialized crawler called focussed crawler traverses the web and selects the relevant pages to a defined topic rather than to explore all the regions of the web page. The crawler does not collect all the web pages, but retrieves only the relevant pages out of all. So the major problem is how to retrieve the relevant and quality web pages.
Keywords— web crawler, focussed web crawler

I. INTRODUCTION

Web search is currently generating more than 13% of the traffic to Web sites .The main problem search engines have to deal with is the size of the Web, which currently is in the order of thousands of millions of pages that is too enormous and is increasing exponentially. This large size induces a low coverage, with no search engine indexing more than one third of the publically available Web

Typing "Java" as keywords into Google search engine would lead to around 25 million results with quotation marks and 237 million results without quotation marks. With the same keywords, Yahoo search engine leads to around 8 million results with quotation marks and 139 million results without quotation marks, while MSN search engine leads to around 8 million results with quotation marks and around 137 million results without quotation marks. These gigantic numbers of results are brought to the user, of which only few are relevant and rest is uninteresting to the users. This complete set of circumstances fetches attention to a prime issue which is the relevance of a webpage to a specific topic. The Process used by search engines to index their databases is very clandestine, they use varied number of web crawlers for collecting and arranging information.

II. HISTORICAL BACKGROUND

Web crawlers are almost as old as the web itself. In the spring of 1993, just months after their lease of NCSA Mosaic, Matthew Gray wrote the first web crawler, the World Wide Web Wanderer, which was used from 1993 to 1996 to compile statistics about the growth of the web. A year later, David Eichmann wrote the first research paper containing a short description of a web crawler, the RBSE spider. Burner provided the first detailed description of the architecture of a web crawler, namely the original Internet Archive crawler. Bring and Page's seminal paper on the (early) architecture of the Google search engine contained a brief description of the Google crawler, which used a distributed system of page-fetching processes and a central database for coordinating the crawl. Heydon and Najork described Mercator a distributed and extensible web crawler that was to become the blueprint for a number of other crawlers. Other distributed crawling systems described in the literature include PolyBot, UbiCrawler C-proc and Dominos.

III.FEATURES OF WEB CRAWLER

Robustness: The Web contains servers that create spider traps, which are generators of web pages that mislead crawlers into etting stuck fetching an infinite number of pages in a particular domain. Crawlers must be designed to be resilient to such traps. Not all such traps are malicious; some are the inadvertent side-effect of faulty website development

Distributed: The crawler should have the ability to execute in a distributed fashion across multiple machines.

Scalable: The crawler architecture should permit scaling up the crawl rate by adding extra machines and bandwidth.

Performance and efficiency: The crawl system should make efficient use of various system resources including processor, storage and network bandwidth.

Quality: Given that a significant fraction of all web pages are of poor utility for serving user query needs, the crawler should be biased towards fetching "useful" pages first.

Freshness: In many applications, the crawler should operate in continuous mode: it should obtain fresh copies of previously fetched pages. A search engine crawler, for instance, can thus ensure that the search engine's index contains a fairly current

representation of each indexed web page. For such continuous crawling, a crawler should be able to crawl a page with a frequency that approximates the rate of change of that page.

Extensible: Crawlers should be designed to be extensible in many ways to cope with new data formats, new fetch protocols, and so on. This demands that the crawler architecture be modular.

IV. ARCHITECTURE OF WEB CRAWLING

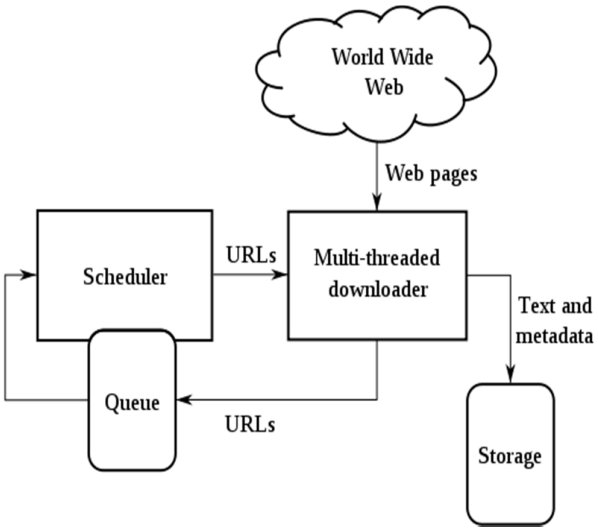


Fig 1: Web Crawler Architecture

The typical architecture involves a scheduler, a module that maintains a queue of URLs to visit, also known as “frontier”, and that sends those URLs in a certain order to one or more downloader’s, that must actually do the network operations. Both communicate through a storage system that may be completely or partially shared. This picture can be refined a bit more if we consider that the scheduling can be further divided into two parts: a long-term scheduling, that must decide which pages to visit next according to quality and/or freshness estimations and a short-term scheduling, that must re-arrange pages to comply with the politeness policy. The time scale for the long-term scheduler is either hours or days, while the time scale for short-term scheduling is in the order of a few minutes or seconds, depending on the waiting time configured for the crawler. The storage can also be further subdivided into three parts: text (or formatted, rich text in case some or all of the HTML tags are kept or other types of documents are indexed), metadata, and links. This is depicted in Figure 1.5.In the case of focused crawlers, the text is important for the classification and prioritization of pages. For most crawlers, the metadata and links are enough for deciding which pages to download next.

V. A SURVEY OF WEB CRAWLERS

The original Google crawler was developed at Stanford. Topical crawling was first introduced by Menczer. Focussed crawling was first introduced by Chakrabarti. A focussed crawler has the following components: (a) How to know whether a particular web page is relevant to given topic, and (b) way to determine how to follow the single page to retrieve multiple set of pages. A search engine which used the focussed crawling strategy was proposed in based on the assumption that relevant pages must contain only the relevant links. So it searches deeper where it founds relevant pages, and stops searching at pages not as relevant to the topic. But, the above crawlers are having a drawback that when the pages about a topic are not directly connected the crawling can stop at early stage. They keep the overall number of downloaded Web pages for processing to a minimum while maximizing the percentage of relevant pages. For high performance, the seed page must be highly relevant. Seed pages can also be selected among the best results retrieved by the Web search engine

a) A standard crawler followed a breadth first strategy. If the crawler starts from a webpage which is n steps from a target document, we have to download before all the documents that are up to n-1 steps from the starting document.

b) A focussed crawler identifies the most relevant links, and ignores the unwanted documents. If the crawler has to start from document that is n steps from target document, it downloads a subset of the documents that are maximum n-1 steps from the Starting document. If the search strategy is optimal, then the crawler takes only n steps to discover the target. A focussed

crawler efficiently seeks out documents about a specific topic and guides the search based on both the content and link structure of the web.

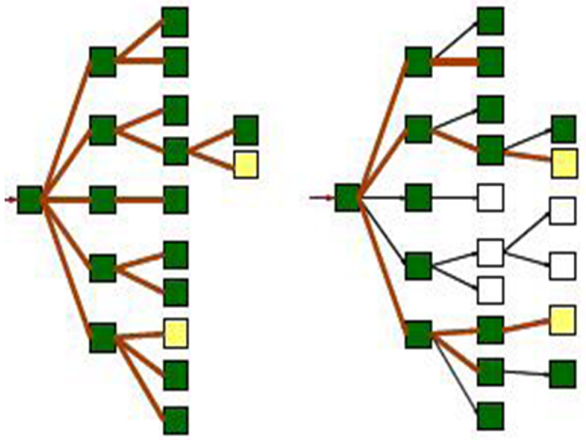


Fig 2: (a) Standard Crawling (b) Focussed Crawling

The above Figure graphically illustrates the difference between a breadth first crawler and a typical focussed crawler. A focussed crawler implements a strategy that
associates a score with each link in the pages it has downloaded. A topical crawler ideally downloads only web pages that are relevant to a particular topic and avoid downloading the irrelevant pages. So a topical crawler can predict the probability that a link to that page is relevant before actually downloading the page.

VI. FOCUSSED CRAWLER

The information can be used to collect more on related data by intelligently and efficiently choosing what links to follow and what pages to discard. This process is Called Focused Crawling. Focused crawling is a promising approach for improving the precision and recall of search on of web page increases and it negatively affects the the Web. It is a crawler that will seek, acquire, index, and maintain pages on a specific topic Best-first search is the most popular search algorithm used in focussed crawlers. In best-first search, URLs are not just visited in the order they are present in the queue; instead, some rules are applied to rank these URLs .But we see there are multiple URLs and topics on a single web page. So the Complexity performance of focussed crawling and the overall relevancy of web page decreases.

VII. COMPARISON OF FOCUSSED AND NON FOCUSSED ALGORITHMS

<table>
<tr><th>Non focussed algorithm</th><th colspan="2">focussed algorithm</th></tr>
<tr><td rowspan="4">Breadth first search:

It uses the frontier as a FIFO queue, crawling links in the order in which they are encountered. The problem with this algorithm is that when the frontier is full, the crawler can add only one link from a crawled page since it does not use any knowledge about the topic, it acts blindly. that is why, also called, Blind Search Algorithm</td><td colspan="2">Approaches</td></tr>
<tr><td>Best First search</td><td>From a given Frontier of links, next link for crawling is selected on the basis of some priority or score. Thus every time the best available link is opened and traversed</td></tr>
<tr><td>Fish search</td><td>For every node we judge whether it is relevant, 1 for relevant, 0 for irrelevant.therfore all relevant pages are assigned the same priority value.</td></tr>
<tr><td>Shark search</td><td>Rather than using binary evaluation of document relevance, it returns a “fuzzy” score, i.e., a score between 0 and 1(0 for no similarity and 1 for perfect “conceptual “match.)</td></tr>
</table>

VIII. SYSTEM ARCHITECTURE OF FOCUSSED CRAWLER

The focussed crawler is made up of four subsystems:

A. Seed pages fetching subsystem.
B. Topic keywords generating subsystems.
C. Similarity computing engine.
D. A spider

The whole working process of the focussed crawler is showed in figure

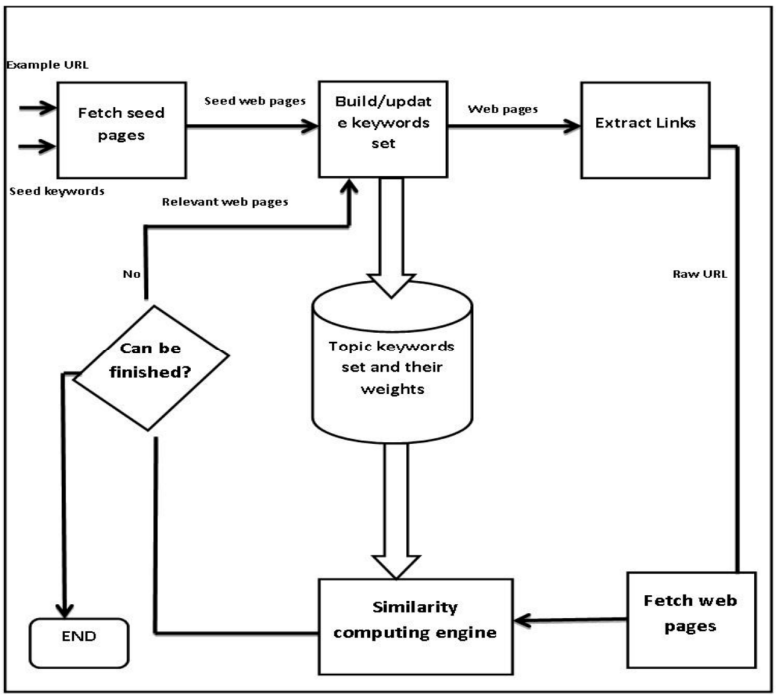


Fig 3: focussed Crawler Architecture

IX. ALGORITHM OF FOCUSSED CRAWLER

A focussed crawler algorithm which efficiently combines link based and content-based analysis to evaluate the priority of an uncrawled URL in the frontier.
Input: topic T, threshold of relevant of page content T1, threshold of relevant of text of
Linkage T2, threshold of count of crawling pages T3;
Output: Web pages relevant to topic
1. While (queue of linkage is not null) ^ (amount of crawling pages < T3) do
2. Get the linkage at the head of queue and downloading web page P the linkage linked
And calculate the relevant topic T
Relevance (P) =similarity (P, T)
If relevance (P) <T1 then
3. Dismiss page P and all of linkages in this page;
4. go to 15:
5. End
6. For each linkages a in the page P do
7. Score a as follows:

Relevance (a) =similarity (a, T)
8. If relevance (a) <T2 then
Dismiss a;
9. go to 6;
10. End
11. If the linkage a has not been crawled then
12. add linkage a into queue of linkage
13. End
14. End

X. FUTURE SCOPE

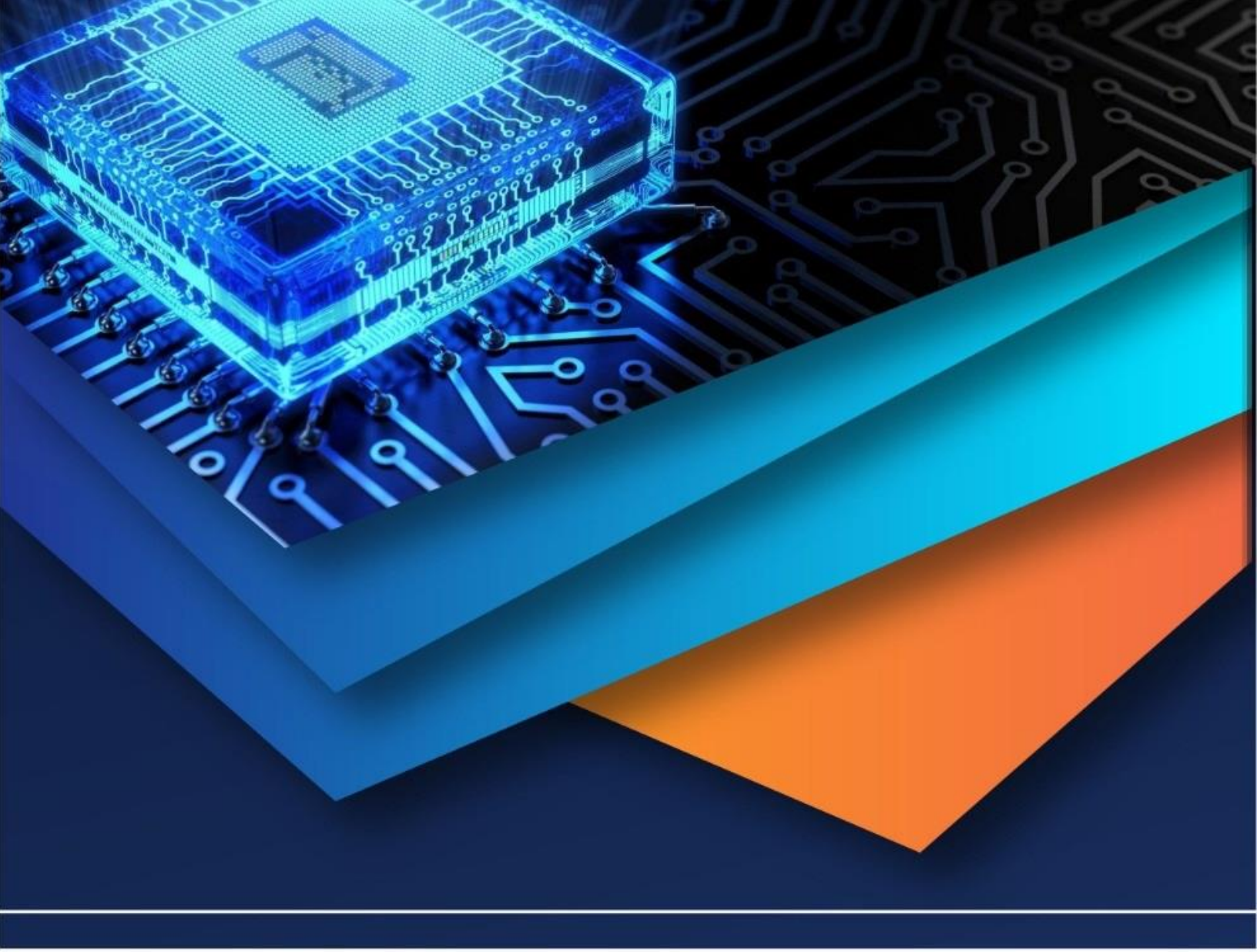
A. Code optimization
B. Blocks can be divided on some other way in which the overall complexity of web page decreases and also it will be easier to apply focussed crawling on these pages.
C. the time complexity must be reduced because crawler efficiency not only depends on to retrieve maximum number of relevant pages but also to finish the operation as soon as possible.

XI.CONCLUSIONS

Web crawlers are the program that uses the graphical structure of the Web to move from page to page. A focussed crawler is a crawler that targets a desired topic and gathers only a relevant Web page which is based upon predefined set of topics and do not waste resources on irrelevant web pages. Best-first search is the most popular search algorithm used in focussed crawlers. The Complexity of web page increases and it negatively affects the performance of focussed crawling and the overall relevancy of web page decreases. Such a focused crawler entails a very small investment in hardware and network resources and achieves desired results. A highly relevant region in a web page may be obscured because of low overall relevance of that page. We will present an algorithm how to efficiently and accurately divide the web page into content blocks and then we will Apply focussed crawling on the content blocks.

REFERENCES

[1] Astha, “Web page content block partitioning for Focussed Crawling” Master thesis, CSE Department, THAPAR UNIVERSITY PATIALA –June 2012
[2] Mr.Ravinder Kumar, Ravikiran Routhu: “Enrichment in Performance of Focussed Crawlers” CSE Department, Thapar University, 2010.
[3] Yang Yongsheng, Wang Hui, " Implementation of Focussed Crawler” COMP 630D course Project Report,2000



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY