International Journal of Computing and Business Research (IJCBR)

ISSN (Online) : 2229-6166

Volume 3 Issue 1 2012

# AN EFFICIENT CRAWLING ALGORITHM FOR OPTIMIZATION OF WEB PAGE FOR MAJOR SEARCH ENGINES

Sanjeev Dhawan[1], Pooja Choudhary[2]

Dept.of Computer Science & Engg., UIET, Kurukshetra University Kurukshetra, India

rsdhawan@rediffmail.com[1], pdhakla@gmail.com[2]

**Abstract :** Search engine optimization (SEO) is the process of improving the visibility and scope of a website or a web page in search engines' search results. In general, the earlier (or higher ranked on the search results page), and more frequently a site appears in the search results list, the more visitors it will receive from the search engine's users. SEO may target different kinds of search, including image search, local search, video search, academic search news search and industry-specific vertical search engines. As an Internet marketing strategy, SEO considers how search engines work, what people search for, the actual search terms or keywords typed into search engines and which search engines are preferred by their targeted audience. Optimizing a website may involve editing its content and HTML and associated coding to both increase its relevance to specific keywords and to remove barriers to the indexing activities of search engines. Promoting a site to increase the number of backlinks, or inbound links, is another SEO tactic. This paper proposes an efficient crawling algorithm for indexing and searching from the database.

Keywords: crawler, optimization, search engine, search engine optimization

## I Introduction

Search Engine Optimizers a term adopted by an industry of consultants who carry out optimization projects on behalf of clients, and by employees who perform SEO services in-house. Search engine optimizers may offer SEO as a stand-alone service or as a part of a broader marketing campaign. Because effective SEO may require changes to the HTML source code of a site and site content, SEO tactics may be incorporated into website development and design. The term "search engine friendly" may be used to describe website designs, menus, content management systems, images, videos, shopping carts, and other elements that have been optimized for the purpose of search engine exposure.

**II Review of Literature**

Presented here is a rather eclectic collection papers written by some of the leadering individuals in the search engine scene.

Sergey Brin and Lawrence Page "The Anatomy of a Large-Scale Hypertextual Web Search Engine"[7] In this paper, we present Google, a prototype of a large-scale search engine which makes heavy use of the structure present in hypertext. Google is designed to crawl and index the Web efficiently and produce much more satisfying search results than existing systems. The prototype with a full text and hyperlink database of at least 24 million pages is available at http://google.stanford.edu/. To engineer a search engine is a challenging task. Search engines index tens to hundreds of millions of web pages involving a comparable number of distinct terms. They answer tens of millions of queries every day. Despite the importance of large-scale search engines on the web, very little academic research has been done on them. Furthermore, due to rapid advance in technology and web proliferation, creating a web search engine today is very different from three years ago.

Lawerence & Brin *et.al.*"The PageRank Citation Ranking: Bringing Order to the Web" [40] the importance of a Web page is an inherently subjective matter, which depends on

the readers interests, knowledge and attitudes. But there is still much that can be said objectively about the relative importance of Web pages. This paper describes Page Rank, a method for rating Web pages objectively and mechanically, effectively measuring the human interest and attention devoted to them. We compare Page Rank to an idealized random Web surfer. We show how to efficiently compute Page Rank for large numbers of pages. And, we show how to apply Page Rank to search and to user navigation.

Jeffrey Dean and Sanjay Ghemawat [41] map Reduce is a programming model and an associated implementation for processing and generating large data sets. Users specify a _map_ function that processes a key/value pair to generate a set of intermediate key/value pairs, and a _reduce_ function that merges all intermediate values associated with the same intermediate key. Many real world tasks are expressible in this model, as shown in the paper. (Note: this is a program that Google uses to recompile its index in addition to other tasks).

Sanjay Ghemawat *et.al.* [42] we have designed and implemented the Google File System, a scalable distributed file system for large distributed data-intensive applications. It provides fault tolerance while running on inexpensive commodity hardware, and it delivers high aggregate performance to a large number of clients. While sharing many of the same goals as previous distributed file systems, our design has been driven by observations of our application workloads and technological environment, both current and anticipated, that reflect a marked departure from some earlier file system assumptions. This has led us to reexamine traditional choices and explore radically different design points. The file system has successfully met our storage needs. It is widely deployed within Google as the storage platform for the generation and processing of data used by our service as well as research and development efforts that require large data sets. The largest cluster to date provides hundreds of terabytes of storage across thousands of disks on over a thousand machines, and it is concurrently accessed by hundreds of clients.

Chris Burges *et.al.* "*Learning to Rank Using Gradient Descent*"[43] we investigate using gradient descent methods for learning ranking functions; we propose a simple probabilistic cost function and we introduce RankNet, an implementation of these ideas using a neural network to model the underlying ranking function. We present test results on toy data and on data from a commercial internet search engine.

Krishna Bharat and George A. Mihaila "When Experts Agree: Using Non-Affiliated Experts to Rank Popular Topics"[] in response to a query a search engine returns a ranked list of documents. If the query is on a popular topic (i.e., it matches many documents) then the returned list is usually too long to view fully. Studies show that users usually look at only the top 10 to 20 results. However, the best targets for popular topics are usually linked to by enthusiasts in the same domain which can be exploited. In this paper, we propose a novel ranking scheme for popular topics that places the most authoritative pages on the query topic at the top of the ranking. Our algorithm operates on a special index of "expert documents." These are a subset of the pages on the WWW identified as directories of links to non-affiliated sources on specific topics. Results are ranked based on the match between the query and relevant descriptive text for hyperlinks on expert pages pointing to a given result page. We present a prototype search engine that implements our ranking scheme and discuss its performance. With a relatively small (2.5 million page) expert index, our algorithm was able to perform comparably on popular queries with the best of the mainstream search engines.

Clara Yu *et.al.* "Patterns in Unstructured Data" [36] the need for smarter search engines is a presentation suggesting several methods of improving search engine relevancy including latent semantic indexing and multi-dimensional scalining.

Jon M. Kleinberg [37] the network structure of a hyperlinked environment can be a rich source of in- formation about the content of the environment. We develop a set of algorithmic tools for extracting information from the link structures of such environments, and

effectiveness in a variety of contexts on the World Wide Web. The central issue we address within our framework is the distillation of broad search topics, through the discovery of "authoritative" information sources on such topics. We propose and test an algorithmic formulation of the notion of authority, based on the relationship between a set of relevant authoritative pages and the set of" hub pages" that join them together in the link structure. Our formulation has connections to the eigenvectors of certain matrices associated with the link graph; these connections in turn motivate additional heuristics for link-based analysis.

Krishna Bharat, Monika R. Henzinger “Improved Algorithms for Topic Distillation in a Hyperlinked Environment” [38] this paper addresses the problem of topic distillation on the World Wide Web, namely, given a typical user query to find quality documents related to the query topic. Connectivity analysis has been shown to be useful in identifying high quality pages within a topic specific graph of hyperlinked documents. The essence of our approach is to augment a previous connectivity analysis based algorithm with content analysis. We identify three problems with the existing approach and devise algorithms to tackle them. The results of a user evaluation are reported that show an improvement of precision at 10 documents by at least 45% over pure connectivity analysis. In 2000 “SearchPad: Explicit Capture of Search Context to Support Web Search” [39] Experienced users who query search engines have a complex behavior. They explore many topics in parallel, experiment with query variations, consult multiple search engines, and gather information over many sessions. In the process they need to keep track of search context -- namely useful queries and promising result links, which can be hard. We present an extension to search engines called Search Pad that makes it possible to keep track of "search context" explicitly. We describe an efficient implementation of this idea deployed on four search engines: AltaVista, Excite, Google and Hotbot. Our design of Search Pad has several desirable properties: (i) portability across all major platforms and browsers, (ii) instant start requiring no code download or special actions on the part of the user, (iii) no server side storage, and (iv) no added client-server communication overhead. An added benefit is that it

allows search services to collect valuable relevance information about the results shown to the user. In the context of each query Search Pad can log the actions taken by the user, and in particular record the links that were considered relevant by the user in the context of the query. The service was tested in a multi-platform environment with over 150 users for 4 months and found to be usable and helpful. We discovered that the ability to maintain search context explicitly seems to affect the way people search. Repeat Search Pad users looked at more search results than is typical on the web, suggesting that availability of search context may partially compensate for non relevant pages in the ranking.

Sepandar Kamvar "Adaptive Methods for the Computation of PageRank" [44] we observe that the convergence patterns of pages in the PageRank algorithm have a non uniform distribution. Specifically, many pages converge to their true PageRank quickly, while relatively few pages take a much longer time to converge. Furthermore, we observe that these slow-converging pages are generally those pages with high PageRank. We use this observation to devise a simple algorithm to speed up the computation of PageRank, in which the PageRank of pages that have converged are not recomputed at each iteration after convergence. This algorithm, which we call Adaptive PageRank, speeds up the computation of PageRank by nearly 30%.

**III Proposed Algorithm and Data flow diagram**

In this section an efficient crawling algorithm for indexing and searching from the database is proposed. This algorithm indexes the web pages and creates a Database Table (Master Table) which considers the various relevant fields for optimization. A child (Sub-Relation) Table is created according to search queries and fetched results which is stored in memory cache and used for further search. Figure 1 (a) and 1(b) describes the data flow diagram and proposed algorithm.

International Journal of Computing and Business Research (IJCBR)

ISSN (Online) : 2229-6166

Volume 3 Issue 1 2012

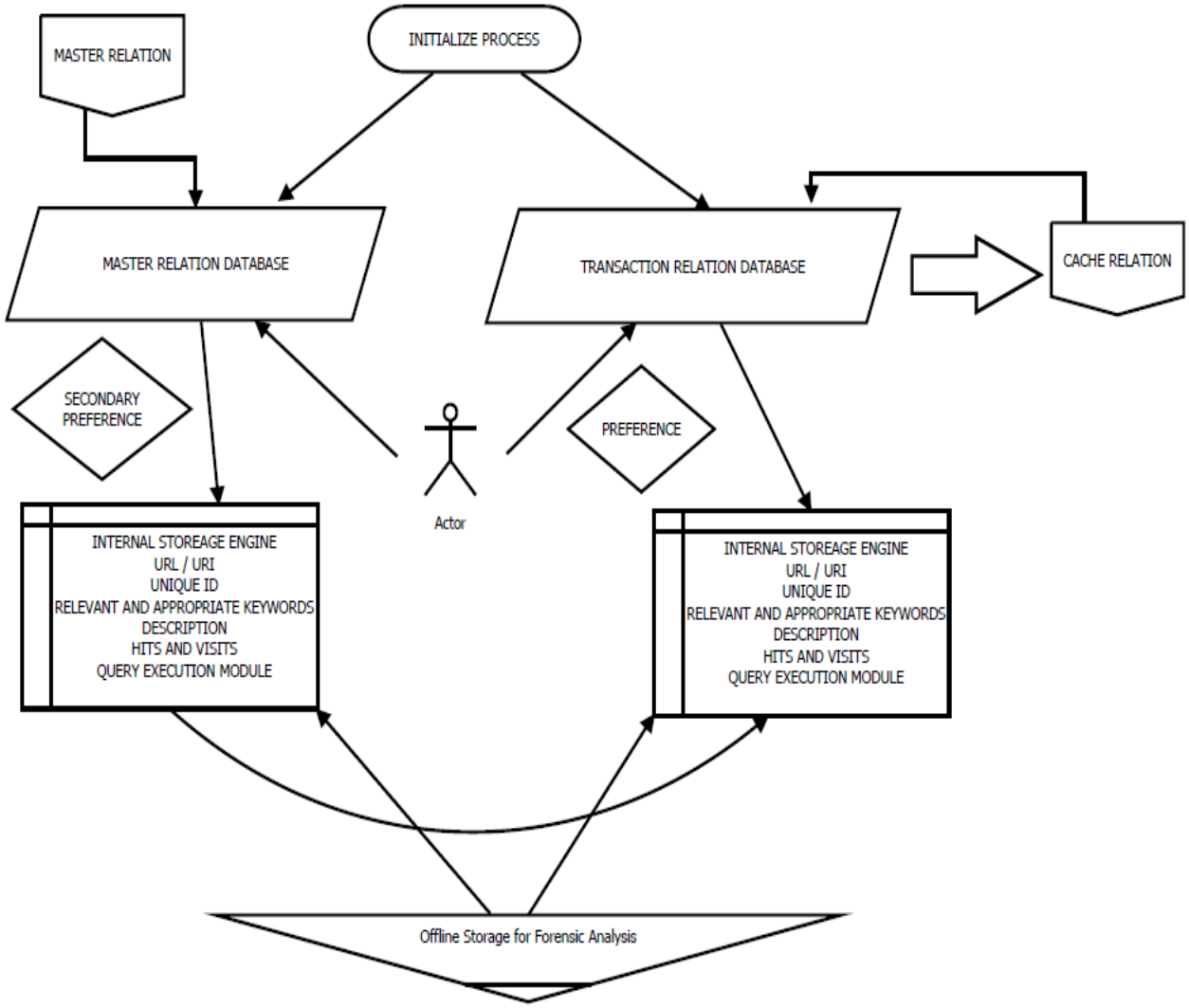


**Figure 1(a) DATA FLOW DIAGRAM FOR PROPOSED CRAWLING ALGORITHM FOR INDEXING OF WEB PAGES ALGORITHMIC APPROACH**

International Journal of Computing and Business Research (IJCBR)

ISSN (Online) : 2229-6166

Volume 3 Issue 1 2012

**ALGORITHM FOR PROPOSED CRAWLING ALGORITHM FOR INDEXING OF WEB PAGES ALGORITHMIC APPROACH**

MODULE - 1 : Database Processing Engine

Create a Database Super Relation (Master Table) RM consisting of relevant fields for optimization

a. ID

- Unique ID for each Record in the RM

b. Page Title

- Title of the Web Page/URL/URI

c. URL / URI

- Universal Resource Locator/Identifier

d. Description

- Brief Description of the URL/Web Application Address with the unique products, Services and area of expertise.

e. IP Address (Allow / Disallow)

f. List of Relevant Keywords for Crawlers

- List of most relevant keywords or phrases for identification of area, services and

  Products of the URI/URL

g. Classification and Categories of Keywords

- Details of the Categories of the Keywords so that search engine can categorize the URL/URI uniquely

MODULE - 2 : Query Processing Engine and Keywords Analysis Panel

- Save Type of Search Queries and Results Fetched
- Creation of Child (Sub-Relation) $R_T$ Table
- A Separate Relation => Memory Cache
- Memory Cache =>
- Relation Structure Same as Master Table $R_M$
- Child Table $R_T$ => Memory Cache $M_C$

MODULE - 3 : Analysis of Transaction Relation

- Search from Child Relation RT
- If Record (Ri) Found in Child Relation RT

  Success

  Else

  Search from Master Relation RM
- Insert record in Child Relation RT

Next Attempt => Child Relation (RT)=> Master Relation (RM)

MODULE - 5 : Save relevant Search Results in a separate database table

MODULE - 6 : Show relevant database table fields to the users while search

MODULE - 7 : Save unique id and keywords of the search queries and results

MODULE - 8 : Display Report based on

- Query Execution Time $Q_{ET}$
- Hits => Specific Record (Ri) => Cache MC
- Keywords $K_i$

**FIGURE 1(b) ALGORITHM FOR PROPOSED CRAWLING ALGORITHM FOR INDEXING OF WEB PAGES ALGORITHMIC APPROACH**

## Brief Description of algorithm:

**Search for query:**

In proposed algorithm the database of search engine has two tables (i.e. master table and child table) for indexing web pages and their optimization. When a user search for a query, the query parser passes it first to the child table which has same relational field as master table. If record founds into the child table then results are shown to user in the form of relevant database table field (URL/URI, UNIQUE ID, RELEVANT AND APPROPRIATE KEYWORDS, DESCRIPTION) otherwise the query is passed to master table which shows the result to user and also store it into child relation. Both tables also store the results into forensic analyzer which analyzed and described the results into the form of Query Execution Time, number of hits and visit and keyword searching time.

**Creation of master table:**

Every time when a user searched for any query, the query parser passes it to the indexer. The indexer will always search the records and index them according to the relevant and appropriate keywords and then store it as master table into the database that has following fields for every record or search query:

a. ID

- Unique ID for each Record in the RM

b. Page Title

- Title of the Web Page/URL/URI

c. URL / URI

- Universal Resource Locator/Identifier

d. Description

- Brief Description of the URL/Web Application Address with the unique products, Services and area of expertise.

e. IP Address (Allow / Disallow)

f. List of Relevant Keywords for Crawlers

- List of most relevant keywords or phrases for identification of area, services and Products of the URI/URL

g. Classification and Categories of Keywords

- Details of the Categories of the Keywords so that search engine can categorize the URL/URI uniquely

## IV Conclusion

In order for Web search engines to continue to improve, they must leverage an increased knowledge of user behavior, especially efforts to understand the underlying intent of the searchers. This paper proposes an efficient crawling algorithm for indexing and searching from the database. This algorithm indexes the web pages and creates a Database Table (Master Table) which considers the various relevant fields for optimization. A child (Sub-Relation) Table is created according to search queries and fetched results which is stored in memory cache and used for further search. My future work will consider the results of this algorithm in the form of Query Execution Time, Hits and Keywords.

## Acknowledgement

I am very grateful to my supervisor Dr. Sanjeev Dhawan (Assistant Professor), University Institute of Engineering and technology,Kurukshetra University, Kurukshetra, for his scholarly guidance and for giving me time and suggestion during comprehensive discussions. I really admire his analytic capabilities due to which I got too able to direct my efforts towards a young and emerging research area.

## References

[1] Beel, Jöran and Gipp, Bela and Wilde, Erik (2010). "*Academic Search Engine Optimization (ASEO): Optimizing Scholarly Literature for Google Scholar and Co*." Journal of Scholarly Publishing. pp. 176–190. Retrieved 2010-04-18.

[2]Brian Pinkerton "*Finding What People Want: Experiences with the WebCrawler*" (PDF). The Second International WWW Conference Chicago, USA, October 17–20, 1994. Retrieved 2007-05-07.

[3]Cory Doctorow (August 26, 2001). *"Metacrap: Putting the torch to seven straw-men of the meta-utopia".* e-LearningGuru. Archived from the original on 2007-04-09. Retrieved 2007-05-08.

[4]Pringle, G., Allison, L., and Dowe, D. (April 1998) "*What is a tall poppy among web pages?"* Proc. 7th Int. World Wide Web Conference. Retrieved 2007-05-08.

[5]Brin, Sergey and Page, Larry (1998). *"The Anatomy of a Large-Scale Hypertextual Web Search Engine*". Proceedings of the seventh international conference on World Wide Web. pp. 107–117. Retrieved 2007-05-08.

[6]Zoltan Gyongyi and Hector Garcia-Molina (2005). "*Link Spam Alliances*" (PDF). Proceedings of the 31st VLDB Conference, Trondheim, Norway. Retrieved 2007-05-09.

[7]Danny Sullivan (September 29, 2005). "*Rundown On Search Ranking Factors*". Search Engine Watch. Retrieved 2007-05-08.

[8]Christine Churchill (November 23, 2005). "*Understanding Search Engine Patents*". Search Engine Watch. Retrieved 2007-05-08.

[9*]"Google Personalized Search Leaves Google Labs - Search Engine Watch (SEW)"* searchenginewatch.com. Retrieved 2009-09-05.

[10] "*Will Personal Search Turn SEO On Its Ear?".* www.webpronews.com. Retrieved 2009-09-05.

[11]"*8 Things We Learned About Google PageRank*" www.searchenginejournal.com. Retrieved 2009-08-17.

[12]"*PageRank sculpting*". Matt Cutts. Retrieved 2010-01-12.

[13]"*Google Loses "Backwards Compatibility" On Paid Link Blocking & PageRank Sculpting*". searchengineland.com. Retrieved 2009-08-17.

[14]"*Personalized Search for everyone*". Google. Retrieved 2009-12-14.

[15]Laurie J. Flynn (November 11, 1996). "*Desperately Seeking Surfers*". New York Times. Retrieved 2007-05-09.

[16]David Kesmodel (September 22, 2005) *"Sites Get Dropped by Search Engines After Trying to 'Optimize' Rankings*" Wall Street Journal. Retrieved 2008-07-30.

[17]Adam L. Penenberg (September 8, 2005). "*Legal Showdown in Search Fracas*". Wired Magazine. Retrieved 2007-05-09.

[18] "*Google's Guidelines on Site Design*". google.com. Retrieved 2007-04-18.

[19]*"Submitting To Search Crawlers: Google, Yahoo, Ask & Microsoft's Live Search*". Search Engine Watch. 2007-03-12. Retrieved 2007-05-15.

[20]"*Submitting To Directories: Yahoo & The Open Directory*" Search Engine Watch. 2007-03-12. Retrieved 2007-05-15.

[21]Cho, J., Garcia-Molina, H. (1998). "*Efficient crawling through URL ordering*". Proceedings of the seventh conference on World Wide Web, Brisbane, Australia. Retrieved 2007-05-09.

[22]Jill Whalen (November 16, 2004). "*Black Hat/White Hat Search Engine Optimization*". searchengineguide.com. Retrieved 2007-05-09.

[23]"*What's an SEO? Does Google recommend working with companies that offer to make my site Google-friendly*?". google.com. Retrieved 2007-04-18.

[24]Andy Hagans (November 8, 2005) *"High Accessibility Is Effective Search Engine Optimization"* A List Apart. Retrieved 2007-05-09.

[25]Matt Cutts (February 4, 2006). "*Ramping up on international webspam*" mattcutts.com/blog Retrieved 2007-05-09.

[26]"*What SEO Isn't*" blog.v7n.com. June 24, 2006 Retrieved 2007-05-16.

[27] Melissa Burdon (March 13, 2007) "*The Battle Between Search Engine Optimization and Conversion: Who Wins*?" Grok.com. Retrieved 2007-05-09.

[28] Andy Greenberg (April 30, 2007) "*Condemned To Google Hell*" Forbes Archived from the original on 2007-05-02 Retrieved 2007-05-09.

[29] Jakob Nielsen (January 9, 2006) "*Search Engines as Leeches on the Web*" useit.com. Retrieved 2007-05-14.

[30] Graham, Jefferson (2003-08-26). "*The search engine that could*". USA Today. Retrieved 2007-05-15.

[31] Jack Schofield (2008-06-10) "*Google UK closes in on 90% market share*" London: Guardian. Retrieved 2008-06-10.

[32] "*Search King, Inc. v. Google Technology, Inc., CIV-02-1457-M*" (PDF) docstoc.com. May 27, 2003. Retrieved 2008-05-23.

[33] Stefanie Olsen (May 30, 2003). "*Judge dismisses suit against Google*" CNET. Retrieved 2007-05-10.

[34] Chris Burges, Tal Shaked, Erin RenshawC, Ari Lazier, Matt Deeds Nicole Hamilton, Greg Hullender "*Learning to rank using gradient descent*" Published ICML'05 Proceedings of the 22nd international conference on Machine Learning Pages 89- 96.

[35] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, Greg Hullender " learning to rank using gradient descent" Published ICML'05 Proceedings of the 22nd international conference on machine learning Pages 89-96

[36] Clara Yu, John Luadardo, Maciej Ceglowski, J. Scott Payne "*Patterns in unstructured data*" National Institute For Technology & Liberal Education (NTILE).

[37] John M. Kleinberg "*Authoritative sources in a hyperlinked environment*" Journal of the ACM (JACM) Volume 46 Issue 5, Sept. 1999.

[38] Krishna Bharat and Monika R. Henzinger "Improved algos for topic distillation in a hyperlinked environment" SIGIR:98 Proceedings of the 21st annual international ACM SIGIR Conference on Research & development in information retrieval Pages 104-111.

[39] Krishna Bharat "*Search Pad: explicit capture of search content to support web search*" The International Journal Of Computer Telecommunication Networking Volume 33 Issue 1-6, June 2000 Pages 493- 501.

[40] Lawerence& Brin, Sergery & Motwani, Rajiv &Winograd,Terry (1999) "*The Page Rank Citation Ranking:Bringing order to the web"* Technical Report Stanford Infolab.

[41] Jeffery Deans & Sanjay Ghemawat "*Map Reduce: Simplified data processing on large clusters"* Communication of the ACM – 50th anniversary issue: 1958-2008 CACM Homepage Archieve Volume 51 issue 1, January 2008 Page 107- 113.

[42]Sanjay Ghemawat, Howard Gobioff & Shun- TakLeung "*The Google file system*" 19th ACM Symposium on operating systems principles, lake George, NY October 2003.

[43] Jaroslaw Balinski and Lzeslaw Danilowicz "*Re- ranking method based on inter-document distances*"Information Processing & Management: an international Journal Volume 41, Issue 4, July 2005 Pages 759-775

[44] Kamvar, Sepandar & Haveliwala, Taher & Golub, "*Adaptive methods for the computations of page rank*" Gene(2003) Technical Report Standford.

[45] Krishna Bharat and George A. Mihaila " *When experts agree: using non affiliated experts to rank popular topics*" ACM Trans. Inf. Syst. Vol. 20(2002), pp. 47-58