Imperial Journal of Interdisciplinary Research (IJIR)
Vol-3, Issue-2, 2017
ISSN: 2454-1362, http://www.onlinejournal.in

Mini Search Engine

Amit M Havinal*, [1]Bharat Nilajkar, [2]Akshay R Salunke, [3]Chetan Chougala
*Department of Computer Science & Engg.,
Don Bosco Institute of Technology, Bangalore, Karnataka, India
[1,2,3] Department of Computer Science & Engg.,
Angadi Institute of Technology & Management, Belgaum, Karnataka, India

Abstract: *This Mini Search Engine is mainly based on a file search application which searches the user desired file from the distributed systems in a network and download the same. It provides an interactive user interface and optimized search results. This also provides the user with the privilege to publish the files which he/she intends to share. The main objective of our tool is to perform the above features across heterogeneous platforms i.e. Windows or Linux. Thus the scope of this search engine is to share document files, audio and video files and also provides the feature of Live Chat among the users that are online at one instance of time.*

Keywords: *File search, File share, Indexing, Crawler, Live chat, File Segregation.*

1. INTRODUCTION

In the past decade, there has been a proliferation of computing networks and platforms in the organization — Windows, LINUX. Windows and Linux interoperability is prerequisite in today's increasingly common heterogeneous computing environments. As collaboration has become increasingly important to effective business operations, here has been dramatic growth in the number of client and server applications. These heterogeneous environments in which each client stores file-level data in its own native file system, increase the complexity of the storage infrastructure, create data accessibility issues, make it difficult to share files, and drive up the overall cost of operations. The diversity of client and server applications creates a range of file-sharing challenges. Clients continue to store file-level data in their native file systems, allowing data to form in isolated pockets that make file sharing difficult. When applications and file services are on the same system, application processing resources become tied up during heavy file sharing activity, which creates difficulty in balancing workloads.

Heterogeneous file sharing allows file sharing among multiple client platforms. The protocols that accomplish this reside above the native file system of the Server and control the file shares accessible on the client network. Thus our tool enables heterogeneous file sharing between clients with different native operating systems. Apart from providing cross-protocol file sharing, Servers also adhere to standard security access methods for the respective native client environments.

1.1 Problem Definition

File Sharing in LAN requires the users to know the specific location of the file i.e. IP address. Different file sharing software's are available for sharing files among Windows platforms or Linux platforms. Thus our tool provides a feature of sharing files between Windows or Linux and also the user can search the file in a network without the need not know the exact location of the desired file.

1.2 Objective and Scope

Mini Search Engine allows you to share files between the clients connected by LAN in an organization. In a Local Area Network, the users need to know the location of the file along with the IP address or name of the system on which it is present. This makes the task of sharing files cumbersome. Also, special security measures are needed to stop users from using programs and data that they should not have access. Thus mini search engine aims at solving these shortcoming ,by providing a tool to the user to enable him to search files based on keywords and also download them to his/her system . The user need not know the exact location of the file, thus making his/her task easy. This also provides the user with the privilege to publish the files which he/she intends to share.

Objective: The main objective of this tool is to perform the above features across heterogeneous platforms i.e. Windows or Linux.

Scope: The scope of this tool is to share document files, audio and video files and also provides the

feature of Live Chat among the users that are online at one instance of time.

1.3 Existing System:

You Consider any company or any organization, if they want to access any document, file or some developed code (software development company), which exists in some remote computer or on different flour or in different branch of same company, firstly they have share that file in LAN (Local Area Network) or they have to travel till there and get it through some storage media (like Pen drive) or lastly they have to use e-mail facility. This is the most considerable problem all the companies.

For Example:

* principal of an Institute asks for a faculty details, the subordinate has to go to department ask the details and get an hard copy or a soft copy in pen drive.
* Software developer who in need of some code which is developed some other person has to approach that person and get the code through e-mail or through pen drive.

1.4 Proposed System:

In this we build an application that will enable the user to retrieve the desired information or file from a remote distributed system. We have optimized the efficiency and time required to achieve this task.

When a user searches the required data, text or key word is matched by controller through the dumps. This matching of keyword is achieved using pattern matching algorithms. Once the required information is located in any one of three dumps I.e. first-Last Modified Files, second-recently used files and finally-rarely used files, only the system contained that information will be responded back, where other systems are idle. From this we achieve better utilization of resources and increased performance. The responded information will be displayed on user's screen in secured manner.

The bellow figure illustrates the concept:

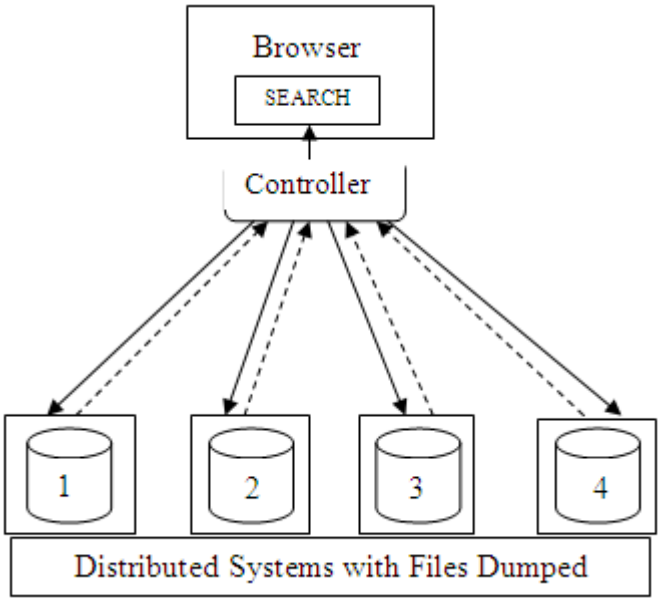


Fig: Working of Mini Search Engine

1. Last Modified Files: Contains the files which are modified or used very frequently.
2. Recently Used Files: Contains the files which are used recently.
3. Rarely Used Files: Contains the files which are used very rarely.
4. Other files: Contains the files which are not been used at least once.
Note: These files can be transferred from one server to other depending upon the access time n usage.

2. METHODOLOGY

2.1 MSE Working Methodology

Each system or a node can act as server or client. Initially a publisher will send a list of filenames it offers to share to the discovery station i.e. the main server, which maintains a database of all the files offered by different publishers on the network with their respective locations. In a particular session a client first makes a request for a desired file to the Discovery Station.

Fig-1: File Sharing Architecture

Discover Station (DS) then maps the requested file name in the database based on the keyword and returns to the client a list of filenames which match the request along the IP address. The client then selects appropriate file from the list and requests for a connection to the publisher which will now act as a server. After a connection is established a direct download from one machine to another is then possible. By using this method the workload on the server is greatly reduced as the communication is between both clients from this point forward.

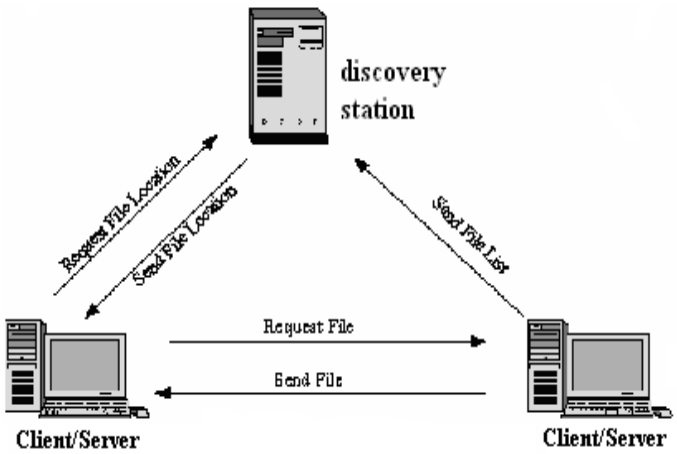
2.2 Client Process

The client is a process (program) that sends a message to a server process (program), requesting that the server perform a task (service). Client programs usually manage the user-interface portion of the application, validate data entered by the user, dispatch requests to server programs, and sometimes execute business logic. The business logic includes sending the current shared folder files list and the current system IP address(Index Implementation). It also responsible for sending the updated files list (if any) to the server (Crawler implementation). The client-based process is the front- end of the application that the user sees and interacts with. The client process contains solution-specific logic and provides the interface between the user and the rest of the application system. One of the key elements of a client workstation is the graphical user interface (GUI).

2.3 Server Process

Firstly when the user logs in, the self-index table will be updated with client's files list(Indexing). Then server process (program) fulfills the client request by performing the task requested. Server programs generally receive requests from client programs, execute database retrieval and updates, manage data integrity and dispatch responses to client requests. The process is also responsible for periodic updation of index table as and when client requests(Crawler). The server-based process "may" run on another machine on the network. This server could be the host operating system or network file server; the server is then provided both file system services and application services. The server process performs the back-end tasks that are common to similar applications.

2.4 Indexing



Search engines are an essential tool for modern life. We use them to discover new information on diverse topics and to locate a wide range of resources. The search process in all practical search engines is supported by an inverted index structure that stores all search terms and their locations within the search able document collection. Inverted indexes are highly optimized, and significant work has been undertaken over the past fifteen years to store, retrieve, compress and understand heuristics for these structures. In this paper, we propose a new self-organizing inverted index based on past queries. We show that this access-ordered index improves query evaluation speed by 25%–40% over a conventional, optimized approach with almost indistinguishable accuracy. We conclude that access-ordered indexes are a valuable new tool to support fast and accurate web search.

2.5 Crawler

The World Wide Web is an interlinked collection of billions of documents formatted using HTML. Ironically the very size of this collection

has become an obstacle for information retrieval. The user has to shift through scores of pages to come upon the information he/she desires. Web crawlers are the heart of search engines. Web crawlers continuously keep on crawling the web and find any new web pages that have been added to the web, pages that have been removed from the web. Due to growing and dynamic nature of the web; it has become a

In this paper, we are concentrating on focus crawler which search for the relevant web pages based on the keyword we give. Actually it forms a hierarchy of links. The crawler on the particular web page for a particular keyword, which we give as input. It will search for the link on that seed URL and after that switch to that link and find another link on that web page but it should match with the keyword, it will do like that until it reach the limit that we set. But it may be possible that it will not found the number of links that we set before. Then it shows that the web page is not having any further link for that particular keyword. While fetching the links the crawler also make sure that it should fetch only the unique links, means that it should not revisit the same link again and again.

Finally, when we finished with the links, we will give one txt file as input and run the three pattern matching algorithm. Pattern here is the text only. The algorithms that we used are KMP (Knutt-Morris-Pratt), BMM(Boyer-Moore) and finite automata.

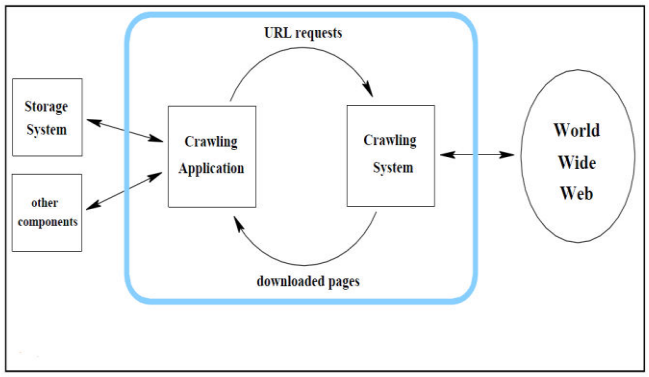


Fig-2:Basic two components of the Crawler

The typical design of search engines is a "cascade", in which a Web crawler creates a collection which is indexed and searched. Most of the designs of search engines consider the Web crawler as just a first stage in Web search, with little feedback from the ranking algorithms to the crawling process. This is a cascade model, in which operations are executed in strict order: first crawling, then indexing, and then searching.

3. RESULT

This implementation of file sharing tool is an effort towards providing the basic functions present in existing tools. With its simple interface it provides an easy to use application to even first time users who have no previous experience of sharing files. Our tool provides unique feature of sharing files across heterogeneous platforms and also acts as a search engine in the network.

One can not only share text files but also audio and video files. It also has a additional feature to view the downloaded audio, video and text files. In a network one client can communicate to the other using the feature of live chat. An easy, well thought-out and useful interface together with combined features of searching, downloading and live chat makes it a user-friendly tool for file sharing across heterogeneous platforms.

REFERENCES

[1].M. Sunil Kumar, P.Neelima, "Design and Implementation of Scalable, Fully Distributed Web Crawler for a Web Search Engine", International Journal of Computer Applications (0975 – 8887)Volume 15– No.7, February 2011

[2].Pooja gupta, Mrs. Kalpana Johari, "IMPLEMENTATION OF WEB CRAWLER", Second International Conference on Emerging Trends in Engineering and Technology, ICETET-09

[3].Steven Garcia, Hugh E. Williams, Adam Cannane," Access-Ordered Indexes", Australian Computer Society, Inc. 27th Australasian Computer Science Conference,
The University of Otago, Dunedin, New Zealand. Conferences in Research and Practice in Information Technology, Vol. 26. V. Estivill-Castro, Ed.

[4]. Vladislav Shkapenyuk, Torsten Suel," Design and Implementation of a High-Performance Distributed Web Crawler", Work supported by NSF CAREER Award NSF CCR-0093400, Intel Corporation, and the New York State Center for Advanced Technology in Telecommunications (CATT) at Polytechnic University, and by equipment grants from Intel Corporation and Sun Microsystems.