ISSN:0975-9646

GooDoop: Local Search Engine using Hadoop

Ganesh Phadatare, Ketankumar Rathod, Vinayak Shukla, Mahesh S. Salunkhe

Dept. of Computer Science and Engineering
K.I.T's College Of Engineering, Kolhapur

Abstract— **The advent of web and search engines have made online se arching a common method for obtaining information. With the highly dispersed amount of information internet has become a primary need. Regardless of the same, the frequency of interruption in internet connection is one of the largest on the planet and for an organization in private or for an institution the internet access rates are at formidable ranges. The increasing dependency on such information has created an utter need for an alternative to the existing online search system. The notion of our project "Local search Engine based on Hadoop" is collection of static informative pages and accessing those information without any internet connection. These static contents are stored on different nodes of the hadoop cluster. For intense processing of large no of static contents we bring in the use of Hadoop platform. Hence conscience behind our project is to collect such web contents at hadoop nodes and to make it available locally without any need of internet to any node present in the network.**

I. INTRODUCTION

Hadoop is the platform providing manipulation of "big data" enabling development of software for reliable, scalable distributed computing. Instead of relying on underlying hardware for delivering high-availability, library itself provides detection and handling of failure at application layer itself.

The internet service in India is prone to frequent interruptions making it inaccessible for users. An organization or institution generally requires certain information which falls under its area of working. Web pages containing this information are frequently accessed by many nodes present in the organization's network. Some users tend to store important pages on their machines. Our project aims to make all this valuable web pages available when the internet is down. This will enable a node in the network to have an offline search for the desired content and access it if it's present on any other node in the network.

The chances of getting such content are generally high as set of queries fired in the particular organization is the same. Static content such as Wikipedia pages seldom change. So, the content doesn't need update and can be accessed at any time once it's saved. The Hadoop platform is used so as to process these numerous web pages available in large number.The project concentrates on developing an offline search engine with ranking algorithm implemented.

II. PROPOSED WORK

The **Problem statement**: Design and Deploy a Local Search engine, which provides the static web contents stored at different nodes in the Hadoop Cluster according to users' search query.

Modules:

- **HTML files collection**: Program for collection of html files from different terminals to the hadoop nodes. Conversion of these html files to a single sequence file.
- **Hadoop Implementation**: A hadoop program to access the html sequence file and rank them according to the occurrence of the keyword searched.
- **User interface**: A user-friendly interface where the query can be fired and the corresponding result can be seen through web browser.

A. Functional requirements:

- A query firing system through which query will be passed.
- Collection of html static pages from various PCs to hadoop nodes.
- Conversion of multiple html input files to create single sequence output file.
- A ranking system that will provide each page a precise rank.

B. Non Functional requirements:

- Hadoop cluster up and running.
- Tomcat server always running on background.
- Timely updated html pages at every node.

III. SYSTEM ARCHITECTURE AND DESIGN

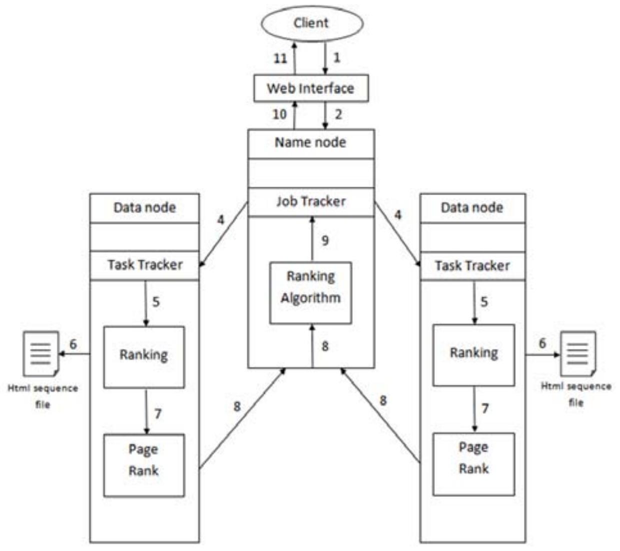
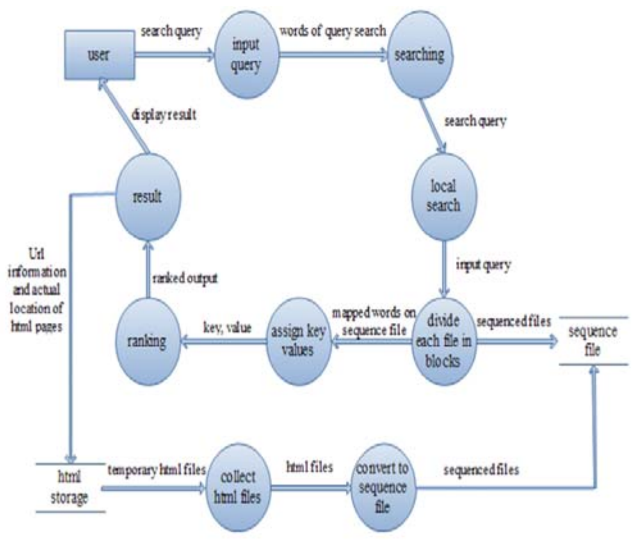


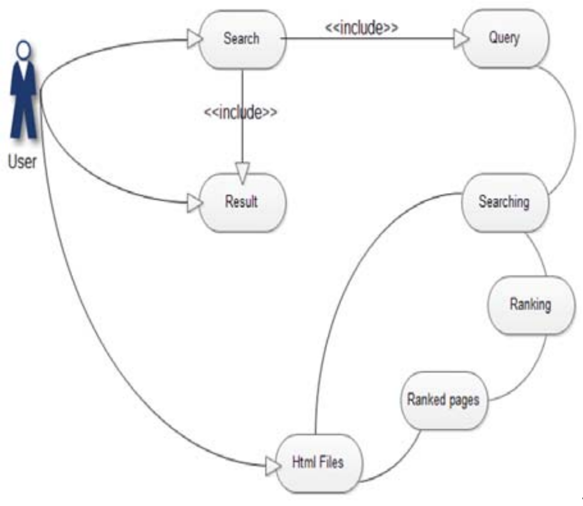
Fig 1. System architecture

Notations:
1. User fires queries using web application.
2. Servlet sends request to jobtracker.
3. Jobtracker distributes task among Tasktrackers.
4. Tasktracker calls ranking algorithm.
5. Ranking algorithm reads html sequence file for queries.
6. Results are provided rank page wise.
7. Pages are sorted according to rank by ranking algorithm.
8. Results are provided to jobtracker.
9. Jobtracker forwards the results to web interface.
10. Web interface displays the result to the client.

1. *Data flow Diagram*



2. *Use case diagram*



IV. Module Specification

A. Algorithm for collection of HTML files at each node:

This module comprises of developing an efficient algorithm which will periodically search for HTML on the hard disk. It collects all the static web content and stores it in dedicated partition.

- We have implemented this module by using client and server applications. The HTML files are transferring from client to server. The client scans the source directory for individual files. It then sorted only '.html' or '.htm' files from source directory.
- Then it sends each file as an object to server. The object also contains the details of file like file name , file size etc. The server receives each object and it is creating and writing those files in the specified destination directory.
- We have implemented FileEvent.java program whose object contains the file along with the necessary metadata related with that file. The client sends the objects of FileEvent to server. The FileEvent.java program implements the serializable interface.
- Then server.java program is written which is accepting socket connection from client. And after getting connected, it is receiving FileEvent objects continuously till the last file is received.(The FileEvent object of last file has reminder attribute value as zero).After receiving all files the server exits itself.
- In the client.java program ,the client is connecting to server and creating FileEvent objects for each file in the source directory and sending to server application. It will display the source path destination path etc.
- To collect the Html files from multiple clients simultaneously we have implemented multithreading application. Through this more than one client can connect to the server at the same time and can transfer the collected HTML files to server's destination directory.
- Both the client -server programs are run by system at startup without any manual intervention. To run the program at system reboot we have use the crontab file using command – crontab –e.
- Then to compile and run the programs at system startup we have use the following commands in crontab file-
  - @reboot service iptables stop
  - @reboot java Client //run client program
  - @reboot java Server //run server program

B. Hadoop program for offline search

- This module comprises of two parts:
  a. First part is a searching algorithm that searches for fired query on every node present in the network and comes up with the results that it procures from all the nodes.
  b. Second part consist of ranking algorithm such as those found in advanced search engines to come up with most relevant results as far as possible.
- Initially all the html files at the different hadoop nodes are converted into a sequence file.
- The hadoop module get the input form the servlet page as an argument and these arguments are stored in the .txt file (args.txt). Multiple word and length argument can be passed at a time for which the search is to be called.
- The hadoop program calls the *job-tracker* which in turn calls the *task-trackers* and performs the *map* and *reduce* functions.
- The program accesses the sequence file and reads each

file as a string. All possible combinations of given query are formed and ranking is done for each combination.

- A count is maintained for each page and reduced to get the total count. The sorting function is called to sort the html pages in order of their decreasing counts. These counts are termed as the ranks for the pages.
- This formatted output is then passed to the servlet page to preview it to the user.

C. User interface:

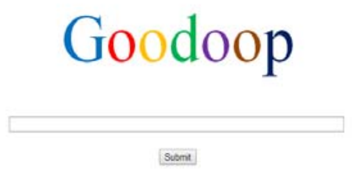
It is an interface that is easy to use and helps the users for getting most appropriate results possible.

- A servlet page is designed which is simple and easy to use. The page consists of and text area where the query to search for the matter is written to search in the data repositories.
- Then A submit button is provided with which the hadoop program is being called to process the query. To the hadoop program the query is passed as an argument.
- The result in then fetched as the processing is done by the hadoop program. This formatted result is again shown to the user through the similar servlet based interface from where the user can access those pages and get the desired output.
- At a single time multiple user can fire the query from different terminals (not necessary hadoop is installed on such nodes) and can get the output in the desired time irrespective of the query length or the number of queries.

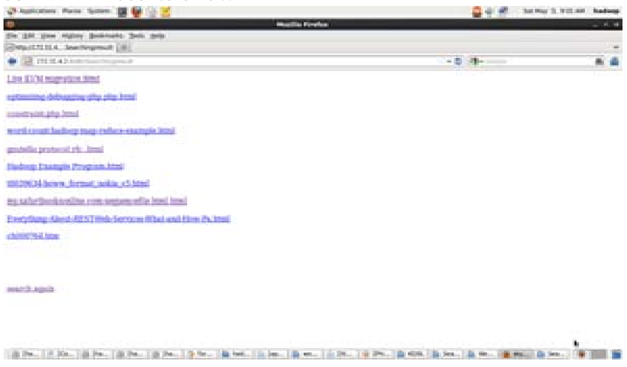
V. OBSERVATIONS

Snapshots

a. User Interface to fire the query



b. Results shown


Applications Places System
Mozilla Firefox
File Edit View History Bookmarks Tools Help

Live KVM migration.html

optimizing debugging php.php.html

construct.php.html

word count hadoop map reduce example.html

gnutella protocol rfc..html

Hadoop Example Program.html

00029634 howto_format_nokia_c5.html

[illegible].html.html

Everything About REST Web Services What and How Pt.html

ch00794.htm

search again

VI. PROCESS MODEL

- The process model implemented by our group is Spiral Model. Several different modules where implemented simultaneously and consequently the improvements and changes were made.
- The model helped us in the types of application, where the design is highly subjective and changes have to be made at a very regular basis as per the change in software used or the requirements to be fulfilled.

VII. CONCLUSION

The project hence implements a web application which results the html documents or html contents from different nodes in a network according to users request at local level. Hence reduces the unnecessary use of internet making the search more efficient.

VIII. FUTURE WORKS

- As the technique works on big data similar search engine can be developed for working on images, large audio and video files.
- Also this search engine can be used in development of intelligence system such as data mining etc.

REFERENCES

[1] Hadoop-The Definitive Guide Edition-by Tom White
[2] Administration Guide – Redhat5.RHEL5
[3] www.docs.apache.hadoop.org
[4] We Do Hadoop www.hortonworks.com
[5] Hadoop Beginner's Guide by Garry Turkington.
[6] Professional Hadoop Solutions by Boris Lublinsky, Kevin T. Smith, Alexey Yakubovich (WROX).
[7] Servlet – www.w3school.com & courses.courseservlets.com
[8] SSH – www.linuxproblem.org