DESIGN OF FILE SYSTEM ARCHITECTURE WITH CLUSTER FORMATION ALONG WITH MOUNT TABLE

Sheetu Sharma[1], Vikas Gupta[2]

[1]Computer Science & Engg., AIET, Faridkot, Punjab, India
[2]Assistant Professor, Department of ECE, AIET, Faridkot, Punjab, India

Abstract

Architecture for comprehensive dataset is defined as a File System. In a comprehensive cluster set, there are a large number of servers where data are directly stored. Cluster is used to store tuples from one or more relations physically closed to other in the database. Clustering is a way of storing data on a disc. In this proposed research work the file system architecture is maintained with cluster formation and mount table specification. The cluster formation is an intelligent formation which works on keyword based feature analysis on files. The related files are kept in one cluster. Along with this, the mount table is attached which is a table, that stores the keyword information as well as other metadata related to each file contained in the system. The proposed research work is extended in two main phases. In the first phase, the distributed architecture is defined in clusters. Once the architecture is defined, in the second phase the user query is filtered and the keyword is extracted from it. An extracted keyword of query is used in the hash table, to find corresponding cluster having all related files. Then all files related to that keyword are displayed. Then upon selection of one particular file, the whole content of that file is displayed. This research work is focused on solutions to get or retrieve data from large data source in less time and at less computation cost.

Keywords: *Hadoop Distributed File System, Query Optimization, Indexing Technique, Distributed System, Cluster.*

----------***----------

1. INTRODUCTION

Distributed system provides distributed file system architecture and framework for the analysis and transformation of large data sets using the Map Reduce, Middleware paradigm. An important characteristic of a distributed system is the partitioning of data and computation across many of the hosts and executing application computations in parallel close to their data [4]. A distributed system cluster scale's computation capacity, storage capacity, and IO bandwidth by simply adding commodity servers. One of the best features of distributed environment is a query processing for various clients and users. Many processing queries usually need optimization for searching purposes. Generally, the query optimizer cannot be accessed directly by users: once queries are submitted to the database server, and parsed by the parser, they are then passed to the query optimizer where optimization occurs [6]. However, some database engines allow guiding the query optimizer with hints. Most query optimizers represent query plans as a tree of "plan nodes". A plan node encapsulates a single operation that is required to execute the query. The nodes are arranged as a tree, in which intermediate results from the bottom of the tree. Each node has zero or more child nodes-those are nodes whose outputs are fed as input to the parent node. For example, a join node will have two child nodes, which represent the two join operands, where sort node would have a single child node. The leaves of the tree are nodes which produce results by scanning the disk, For example by performing an index or sequential scan [8]. Huge databases need to optimize fetching of data so to provide fast and efficient query reply to requested query by various users. In Our research work, a good query optimization process which will be used in a distributed environment [2].

Query optimization is a function of many relational database management systems. The query optimizer attempts to determine the most efficient way to execute a given query by considering the possible query plans [6]. A query is requested for information from a database. It can be as "finding the address of a person with AA123-890," or more complex like "finding the average salary of all the employed married ladies in London between ages 25 to 35, that earn less than their husband", Query result is generated by accessing relevant database data and manipulating it in a way that yields the requested information .The purpose of query optimization, which is an automated process, is to find the way to process a given query in minimum time [2]. Thus query optimization typically tries to approximate the optimum by comparing several commonsense alternatives, to provide in a reasonable time a "good enough" plan which typically does not deviate much from the best possible result [2, 8]. The descriptive data have been always in the form of document and these documents exist in different file formats. When the work is performed for particular enterprise, it contains a vast collection of files over the system. In such case the management of these files and handling the file system query is itself a challenging task [19]. Query optimization based architecture is proposed in this thesis research work to define file system architecture. This is quite beneficial as:-

1. As the work is based on a cluster based, it reduces the size of a database query. Instead of maintaining the file

system individually, the management of the specific clusters is easy to represent.
2. The cluster definition enables the easy migration of the sub-file system on different location physically.
3. While working on distributed systems, such kind of architecture is more beneficial to maintain the cluster location wise.
4. As the cluster formation is keyword based, the query analysis easily identify the required cluster.
5. As the mount table is maintained for each cluster to maintain the cluster data, the query processing will be more effective.

2. OBJECTIVES OF RESEARCH WORK

The main work presented in the system is to define file system architecture with query optimization. The research work is divided in terms of some research objectives given as under.

1. Design of File System architecture along with cluster formation and Mount table's specification.
2. Implementation of keyword based clustering.
3. Generation of separate mount table for each cluster.
4. Implementation of the client side, respective to query fetching, analysis and getting results based on analysis.
5. Fetching data from the query optimization process.

3. PROBLEM FORMULATION

The Query optimizer is very important part of data mining to get desired data from a large source of data. Query optimizer which makes the user's query in a format which helps in the further process of fetching data in lesser time. The file system is one of the most traditional and widely used mechanisms to maintain the user data in the form of distributed system [4]. In this research work file system architecture has been proposed based on query optimization process. This is basically designed for large file dataset where the user has terabytes or petabytes of data in the form of files and there is a need to avail the information to the user effectively on request. The presented work is defined in two stages. In the first stage the file system architecture is defined and on second stage the effective user query is defined. To maintain the data effectively a clustered file system is defined. In this presented approach complete file system will be divided in the form of clusters and the cluster definitions are based on keyword analysis over the system. Each cluster separately maintains a mount table to keep track of the files presented in the system.

As the user passes a query to the system, at first the keyword extract from the query is to be performed [3]. Based on the keyword based match at first the relative cluster is to be identified. Now to retrieve the relative file content, a search is to be implemented on mount table that contains the descriptive information along with location specification for each file of the cluster. From this mount table, search the actual path of the related contents is to be displayed to the user. The query processing is performed in two steps, first to identify the cluster and second to identify the file location and other information within the cluster.

4. PROPOSED METHODOLOGY

In this proposed work a keyword based analysis is available to generate the cluster of distributed file system. The proposed work is an intelligent system which searches items with optimized query and in which the similar files are maintained in one cluster. To keep the file system information, a mount table is maintained that stores the file keywords as well as the metadata related to each file contained in the system. In the second phase the user query is processed and the keyword extraction is performed. Based on keyword analysis the cluster will be selected and the query is performed on that specific cluster.

4.1 Source of Data

In this proposed work, we need some dummy dataset or the file set on which the work will be presented. This kind of data can be online or offline. Along with the files user also need the metadata related to the file system. This kind of dataset can be driven either by using the global or the private web or we can take the file system used by the earlier researchers. The file system must have the following properties

1. A Large file system with large number of files.
2. Metadata of files should be available.
3. Files should be in a query based format such as text files.
4. Files must be capable of performing different operations such as read, write operations.

5. IMPLEMENTATION

The presented work is about the generation of file system architecture. The work is based on a distributed cluster based architecture in which the keyword analysis over the file is used for cluster generation. To provide the effective processing, each cluster maintains a mount table. As the query is passed by the user, at first the cluster identification is performed and just after that mount table is processed to get the file path and the related Meta data.

To solve the problem of distributed file system processing a novel clustered approach is suggested [19], as shown in Fig.-1, the user sends a request to the global query interface of distributed file system; the query is received by the query analyzer which analyzes the query with the help of global schema. Query analyzer breaks the query into sub-queries and sends to cost optimizer for cost estimation of sub queries. Cost optimizer analyzes the cost with the help of data dictionary. Query distributor has the main responsibility to receive sub-queries from the cost optimizer and sends to the appropriate local optimizer of local site of a cluster. A mount table is placed between local optimizer and cluster. Mount tables are like the index of the book which contains all the information of the files like location and extension. A local and global query optimizer concept comes in multidatabase systems (MDBS) and local database

system (LDBS). Here when a large number of databases integrated it becomes MDBS. Now when a user puts a query then that query is optimized by Global Query Optimizer to know on which database to go among various heterogeneous databases. After going to one database, then local query optimizer works to find now from which table retrieves data. The query will be performed on the mount table and relatively the file found from the system along with file information.

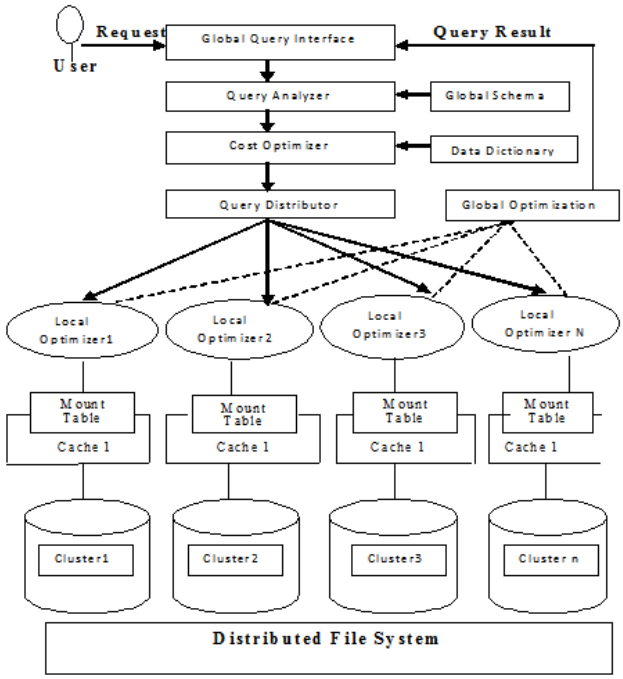

Fig -1: Modified Hadoop Framework for Data Management.

6. RESULTS AND DISCUSSION

Steps to be followed to get the desired data from server as shown in Fig-2:

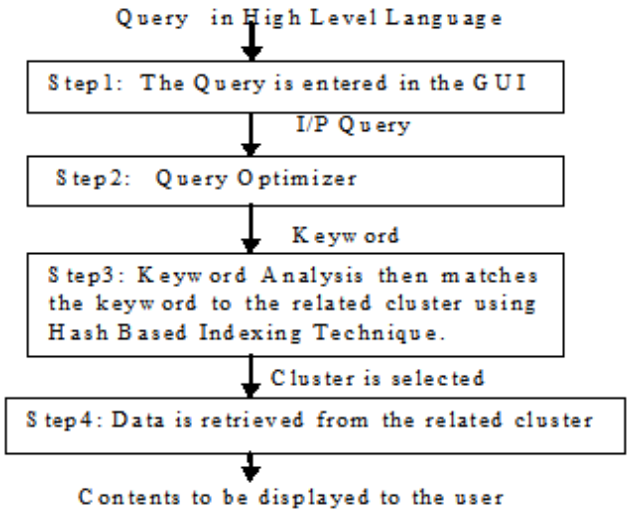

Fig -2: Steps to Get Desired Data

Step (1), Fig-3 shows the GUI where the user can enter any type of query. In this interface a search box is visible and the text box is visible, where user can type the query to be answered. Just adjacent to it, there is a submit button. After entering the query in the text box then submit button is clicked which in turn performs the action.

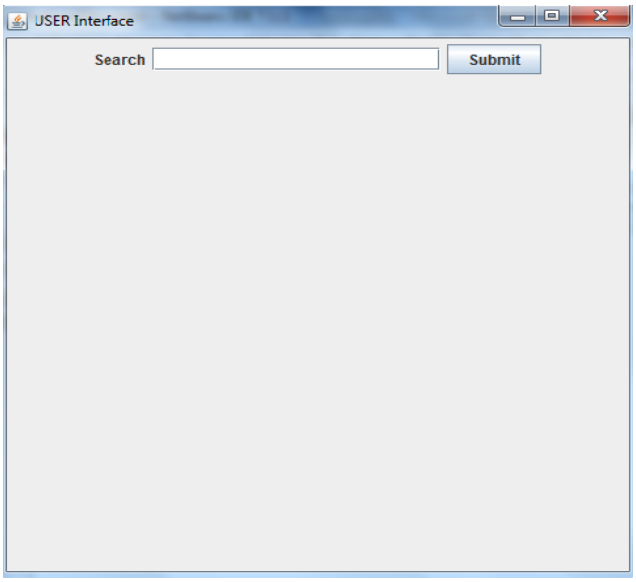

Fig-3: Graphical User Interface for User

Step (2), Fig-4 shows the data that has been entered For Example the question is "what is webmining?" Or anything else which the user wants. The first step is that the user's entered query is optimized by the query optimizer. It breaks the whole query into small units. All the extra words, special character and the punctuation in the query are rejected. For Example words like "what", "is" and the punctuation i.e. the "?". The remaining keyword i.e. "webmining" is left in the search box. Then the submit button is clicked to perform its actions.

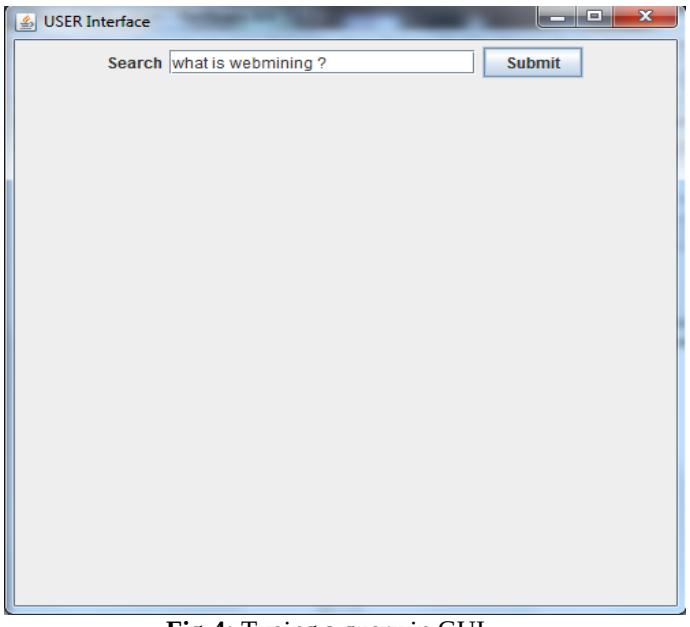

Fig-4: Typing a query in GUI

Step (3), Fig-5 illustrates that keyword analysis then matches the related keyword i.e. is "webmining" with the related cluster and retrieves all the files from the cluster and provides to the user. So that user may select the most appropriate file according to their need.

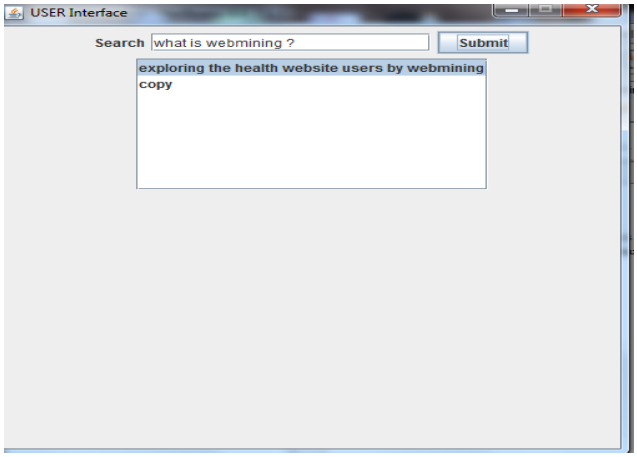

Fig-5: Displaying the most suitable files related to the query keyword

Then all the files are displayed to the user and user selects the file which is requested by user. Now double click on it, to see the contents of that particular file.

Step (4), Fig-6 shows the whole contents of the selected file with scroll option facility. Any type of data can be retrieved and shown to users as per its requirements.

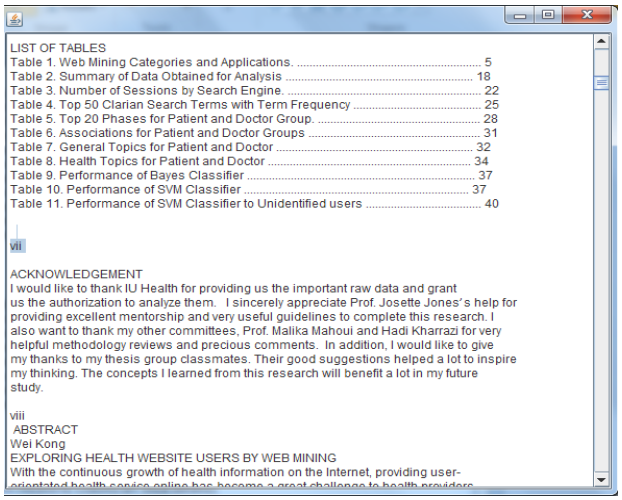

Fig -6: Content of the selected file is displayed.

7. CONCLUSIONS

The Proposed research work focus on, how data is retrieved from large servers in a less span of time and at less cost. Data management is a tough task when data is present in very large amount [19].It is quite difficult to get the desired data in a short time and at less cost.

To achieve the objective a Modified Hadoop framework is made where data is distributed on various servers. In every server there is group of clusters and each cluster have the same type of data in different formats. Map table is maintained with every cluster which stores path of every file present in a cluster, and so that whenever a user selects the file, that file is read from the server and is displayed to the user. To retrieve the file from the server following steps are followed:

1. Query optimizer which breaks the query into words.
2. The irrelevant words are removed from the query.
3. The remaining word in the query is searched by the keyword analysis of the related clusters and keyword cluster is used in the hash table, to find corresponding cluster having all related files.
4. All files related to that keyword are displayed.
5. Then upon selection of one particular file, the whole content of that file is displayed.

In order to compare the time consumed in answering the query through Modified Hadoop and Hadoop system Interface. The example of "Webmining" has been taken up. As shown in table-1, there are five files present in a "webmining" cluster namely: Rs, srs, advantages of webmining, disadvantages of webmining, webmining is datamining.

Table-1: Comparison of Time Consumed by Modified Hadoop Vs Hadoop

Webmining Cluster (Files)	Modified Hadoop (Time Consumed)	Hadoop (Time Consumed)
Rs	858millesecs	2730millesecs
Srs	905millesecs	2762millesecs
Advantages of webmining	796millesecs	1170millesecs
Disadvantages of webmining	795millesecs	2995millesecs
Webmining is datamining	390millesecs	1404millisecs

So it is concluded that the time consumed in answering a query by Proposed Hadoop quite less then the time consumed in answering a query by Hadoop system.

The utilization of resources to execute the query is the Computation cost- the resources are like Processor, RAM as shown in table-2. The data very well present that the computation cost is quite less using Modified Hadoop as compared to Hadoop system. In Modified Hadoop the processor is used as 2.3GHz as compared to 4.30GHz in Hadoop. In Modified Hadoop, RAM consumes 2GB as compared to 3GB in Hadoop as shown in a Table 2.The memory consumed is lesser and the processing is done at a faster rate as compared to Hadoop.

Table -2: Comparison of Resources that are used in Modified Hadoop Vs Hadoop

Framework	Processor(GHz)	RAM(GB)
Proposed Hadoop	2.3	2
Hadoop	4.2	3

The resources used in Modified Hadoop and Hadoop system are graphically represented as shown in a Chart -1:

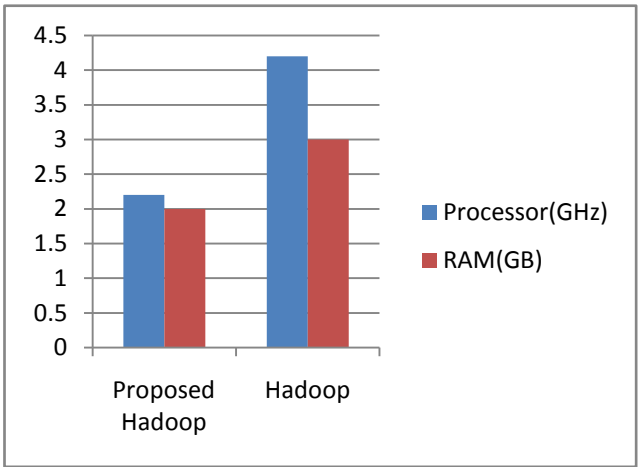

Chart -1: Graphical Representation of Resources that are used in Both System

So at the end the results are provided in shorter period and at a lesser cost which is the objective of the proposed research work.

FUTURE SCOPE

In future, this research can be enhanced by implementing the distributed database operations on fetching process of query optimization by introduction of fuzzy neural selection of dataset while selection of content for various operations

REFERENCES

[1]. Haroun Rababaah "Distributed Databases Fundamentals and Research" ,Advanced Database – B561. Spring 2005. Dr. H. Hakimzadeh Department of Computer and Information SciencesIndiana University South Bend in 2005

[2]. AlaaAljanaby, EmadAbuelrub, and Mohammed Odeh, "A Survey of Distributed Query Optimization", The International Arab Journal of Information Technology, Vol. 2, No. 1, January 2005.

[3]. Navneet kaur ,Rajdeep Kaur ,Navjot kaur "EFFICIENT KEYWORD SEARCH IN RELATIONAL DATABASES "**,,**International Journal of Advanced Research in Computer Engineering & Technology ,Volume 1, Issue 3, May2012.

[4]. Konstantin Shvachko, Hairong Kuang, Sanjay Radia, Robert Chansler Sunnyvale,"The Hadoop Distributed File System" in 2010.

[5]. Xinhua Xu,"A Study on Query Optimization for Federated Database Systems ", Computer and Information Sceience, Vol.2, No.1, April 18-21, 2009, pp. 225-232.

[6]. Query optimization, http.//query optimization.org.

[7]. Ms. M.C. Nikose,"Query Optimization in Object Oriented Databases through Detecting Independent Subqueries", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 2, February 2012.

[8]. Dr. G. R. Bamnote, S. S. Aggrawal "Introduction to Query Processing and Optimization", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 7,July 2013.

[9]. Hive. http://hadoop.apache.org/hive/.

[10]. AbdelkaderHameurlain and Franck Morvan, "Evolution of Query Optimization Methods", Trans. on Large-Scale Data- &Knowl.-Cent. Syst, Springer, LNCS 5740, pp. 211–242, 2009.

[11]. Abhijit Banubakode et al. / International Journal of Computer Science & Engineering Technology (IJCSET**)** Query Optimization in Object-Oriented Database Management Systems:ISSN : 2229-3345 Vol. 1 No. 1,2009.

[12]. M.A. Kashem, Abu Sayed Chowdhury, Rupam Deb, and Moslema Jahan, "Query Optimization on Relational Databases for Supporting Top-k Query Processing Techniques", ISSN 2078-5828, ISSN,JCIT 2010.

[13]. Saurabh Kumar, Gaurav Khandelwal, Arjun Varshney, Mukul Arora," Cost-Based Query Optimization with Heuristics, International Journal of Scientific & Engineering Research Volume 2, Issue 9, September-2011 1 ISSN 2229-5518

[14]. Robert Fourer, Jun Ma Kipp "Optimization Services: A Framework for Distributed Optimization" Robert Fourer, Jun MaMartin Northwestern University, Evanston, Operations Research manuscript OPRE-2008-09-495 in 2010.

[15]. Hadoop. http://hadoop.apache.org/.

[16]. Grid Chen He, Derek Weitzel, David Swanson, Ying Lu "Distributed Hadoop Map Reduce On the Grid" Computer Science and Engineering, University of Nebraska – Lincoln, 2009.

[17]. Vivek Shrivastava, "An Idea of Extraction of Information Using Query Optimization and Rank Query", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 4, February 2012.

[18]. Surajit Chaudhuri, "An Overview of Query Optimization in Relational Systems" in 2012.

[19]. Sheetu Sharma, Vikas Gupta, "Design of File System Architecture with Cluster Formation along with Mount table: A Review." The International Journal of Engineering and Science (IJES) || Volume || 3 || Issue || 6 || 2014 || ISSN (e): 2319 – 1813 ISSN (p): 2319 – 1805.