ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

A Framework on MapReduce Programming Model

Jami Bhagyalakshmi
M.Tech Student
Department of CSE
GMR Institute of Technology,
Rajam.

Dr.R.Priya vaijayanthi, M.Tech, (Ph.D)
Associate Professor,
Department of CSE
GMR Institute of Technology,
Rajam.

ABSTRACT

Big Data is a collection of data that is large or complex to process using on-hand database management tools or data processing applications. Big Data has recently become one of the issues important in the networking world. Hadoop is a distributed paradigm used to manipulate the large amount of data. This manipulation contains not only storage as well as processing on the data. Hadoop is normally used for data intensive applications. It actually holds the huge amount of data and upon requirement perform the operations like data analysis, result analysis, data analytics etc. Now a day's almost every online users search for products, services, topics of interest etc. to compute PageRank using the MapReduce approach to parallelization. This gives us a way of computing PageRank that can in principle be automatically parallelized, and so potentially scaled up to very large link graphs, i.e., to very large collections of webpages. In this post I describe a single-machine implementation which easily handles a million or so pages. In future posts we'll use a cluster to scale out much further – it'll be interesting to see how far we can get. I've discussed PageRank and MapReduce at length review the basic facts. Let's start with PageRank. The idea is that we number webpages 0,\ldots,n-1. For webpage number j there is an associated PageRank q_j which measures the importance of page j. The vector q = (q_0,\ldots,q_{n-1}) of PageRanks is a probability distribution, i.e., the PageRanks are numbers between 0 and 1, and sum up to one, in total. The PageRank q_j measures the importance of page j; the bigger the PageRank, the more important the page.The upshot is that the PageRank vector q can be defined by the equation (explanation below).

INTRODUCTION

BIG DATA:

People upload videos, take pictures on their cell phones, text friends, update their Facebook status, leave comments around the web, click on ads, and so forth. Machines, too, are generating and keeping more and more data.The exponential growth of data first presented challenges to cutting-edge businesses such as Google, Yahoo, Amazon, and Microsoft. They needed to go through terabytes and petabytes of data to figure out which websites were popular, what books were in demand, and what kinds of ads appealed to people. Existing tools were becoming inadequate to process such large data sets. Google was the first to publicize Map-Reduce—a system they had used to scale their data processing needs.

Reasons for the growth of Big Data are

- Increase of storage capacities
- Increase of processing power
- Availability of data

90% of the data in the world today has been created in last two years alone. Structured format has some limitationswith respect to handling large quantities of data. Thus, there is a need for perfect mechanism, like Big Data, to handle these increasing quantities.

DEFINING BIG DATA:

Big Data is the term applied to data sets whose size is beyond the ability of thecommonly used software tools to capture, manage, and process within a tolerableelapsed time.

Big data can be analyzed with the software tools commonly used as part of advancedanalytics disciplines such as predictive analytics, data mining, text analytics and statistical analysis. The semi-structured and unstructured data may not fit well in

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

traditional data warehouses based on relational databases. Furthermore, data warehouses may not be able to handle the processing demands posed by sets of big data that need to be updated frequently or even continually -- for example, real-time data on the performance of mobile applications or of oil and gas pipelines. As a result, many organizations looking to collect, process and analyze big data have turned to a newer class of technologies that includes Hadoop and related tools such as YARN, Map-Reduce, Spark, Hive and Pig as well as NoSQL databases. Those technologies form the core of an open source software framework that supports the processing of large and diverse data sets across clustered systems.

Big data analytics enables organizations to analyze a mix of structured, semi-structured and unstructured data in search of valuable business information and insights. Big data analytics is the process of examining large data sets containing a variety of data types i.e., big data to uncover hidden patterns, unknown correlations, market trends, customer preferences and other useful business information. The analytical findings can lead to more effective marketing, new revenue opportunities, better customer service, improved operational efficiency, competitive advantages over rival organizations and other business benefits.The primary goal of big data analytics is to help companies make more informed business decisions by enabling data scientists, predictive modelers and other analytics professionals to analyze large volumes of transaction data, as well as other forms of data that may be untapped by conventional business intelligence (BI) programs. That could include Web server logs and Internet clickstream data, social media content and social network activity reports, text from customer emails and survey responses, mobile-phone call detail records and machine data captured by sensors connected to the Internet of Things.

CHARACTERISTICS OF BIG DATA:

- Large Volume
- Heterogeneous
- Autonomous sources with decentralized and distributed control
- Explore complex relationships among data

Big data relies on 3 important aspects of data complexity:
The 3 V's of big data:

- Extreme Volume of data
- Wide variety of types of data
- Velocity at which the data must be processed

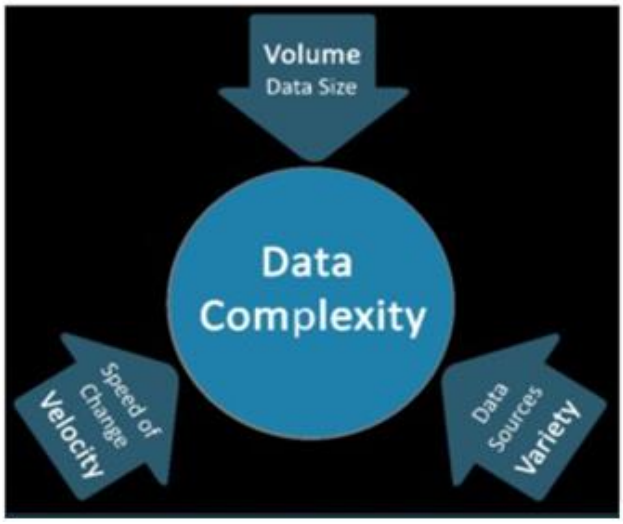

Fig 2.1:Big Data

There are 2 more V's introduced recently as per the researches done:

- Veracity-uncertainity of data/accuracy
- Value of data

SOURCES OF BIG DATA:

- Web Logs
- Sensor Network
- Social Media
- Internet text and documents
- Internet pages
- Search index data
- Atmospheric science, astronomy, biochemical, medical records
- Scientific Research
- Military surveillance
- Photography archives

HADOOP:

Hadoop is a platform that provides both distributed storage and computational capabilities. It is an open source software project that enables the distributed

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

processing of large data sets across clusters of commodity servers. It is designed to scale up from a single server to thousands of machines, with a very high degree of fault tolerance.Hadoop is a distributed master-slave architecture that consists of Hadoop distributed file system (HDFS) for storage and Map-Reduce for computational capabilities. Rather than relying on high-end hardware, the resiliency of these clusters comes from the software's ability to detect and handle failures at the application layer. In a "normal" relational database, data is found and analyzed using queries, based on the industry-standard Structured Query Language (SQL). Non-relational databases use queries, too; they're just not constrained to use only SQL, but can use other query languages to pull information out of data stores. Hadoop is more of a data warehousing system - so it needs a system like Map-Reduce to actually process the data. Hadoop can handle all types of data from disparate systems: structured, unstructured, log files, pictures, audio files, communications records, email. Even when different types of data have been stored in unrelated systems, you can dump it all into your Hadoop cluster with no prior need for a schema. In other words, you don't need to know how you intend to query your data before you store it. By making all of your data useable, not just what's in your databases, Hadoop lets you see relationships that were hidden before and reveal answers that have always been just out of reach.

EXISTING SYSTEM

Need an algorithm to rank web pages based on importance efficiently.PageRank is a link analysis algorithm that assigns a numerical weightingto each element of a hyperlinked set of documents, with the purpose ofmeasuring its relative importance within the set. Votes cast by pages that are themselves "important" weigh more heavily and help to make otherpages "important".PageRank is a probability distribution used to represent the likelihoodthat a person who is just randomly clicking on links will arrive at any particularpage.

Consider:

- B(u) denotes the set of all the pages linking to 'u'.
- L(v) denotes the size of set of all the pages from 'v'.

Page Rank of a page 'u' is: $PR(U)=\sum PR(v)/L(v)$

Damping factor:

The PageRank theory holds that even an imaginary surfer who is randomly clicking on links will eventually stop clicking. The probability, at any step, that the person will continue is a damping factor d. Various research studies show that damping factor is 0.85.

New page rank of the page 'u' is

$$PR(A) = 1 - d + d\left(\frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)} + \cdots\right)$$

DRAWBACKS:

- Graph processing using Pagerank algorithms results in inefficient results due to more number of iterations in calculation.
- Time taken to calculate the pagerank is more.

PROPOSED SYSTEM

Basic Implementation:

Although it can apply to other graph algorithms too, wedescribe the earlier work on graph analysis based on MapReducein terms of the page-rank algorithm. A graph in aMapReduce framework is typically represented as a set ofdirected edges, where each edge is represented as a key-valuepair with the source vertex as the key and the destinationvertex as the value. Each vertex p contains the identifier of thevertex p:id and its meta-data, which includes its current pagerankvalue p:pageRank and the number.

Mapper Algorithm:

Step 1: function MAP(Vertex from, Vertex to)
Step 2: Emit (from.id, (from,to))
Step 3: p ←from.pageRank/from.numOfOutLinks
Step 4: Emit (to.id, p)
Step 5: end function

We first describe the basic approach of applying MapReduce to the graph algorithms described.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

The mapper function given in Algorithm applies to each key-value pair, with the source vertix serving as a key. It computes the pagerank contributions from the source vertex to the destination vertex and emits the destination vertex id as the key and its corresponding fraction of page-rank as the value. In addition to the page-rank contributions, the mapper regenerates the graph structure by emitting the source vertex id as the key and the whole edge (a pair) as the value.

The reducer, described in Algorithm, below gets the pagerankcontributions from each of the incoming edges to avertex, along with the graph topology associated with thevertex. These page-rank contributions are aggregated to getthe updated page-rank value of the vertex. The reducer alsoupdates the page-rank value of the source vertex and therevised edge is written back to the disk. This completes an iteration of the page-rank computation and the output is thenfed again to the mapper to begin the next iteration.

Reducer for the Basic Implementation of page-rank:

Step 1 : function REDUCE(ID m, List [p1,….., pn])
Step 2 : s ← O
Step 3 : M ← null
Step 4 : N ← [ ]
Step 5 : for all p E [p1,….., pn] do
Step 6 : if IsPair(p) then
Step 7 : M ← p.from
Step 8 : insert p.to into N
Step 9 : else
Step 10: s ←s + p
Step 11: end if
Step 12: end for
Step 13: M.pageRank ← s
Step 14: for all n E N do
Step 15: Emit (M, n)
Step 16: end for
Step 17: end function

Implementation:

The Proposed system is implemented to analysis the large amount of data using the Map Reduce Paradigm for PageRank algorithm. It has the following codes that are to be implemented in order to get the output PageRank.

OUTPUT SCREENS

Fig 5.1: Creating directory in hdfs

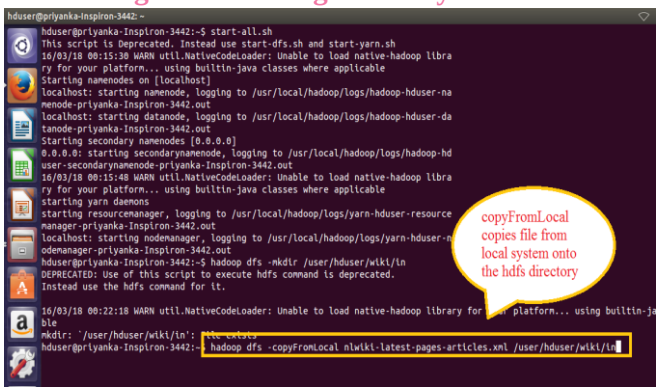

Fig 5.2: Loading input file into hdfs

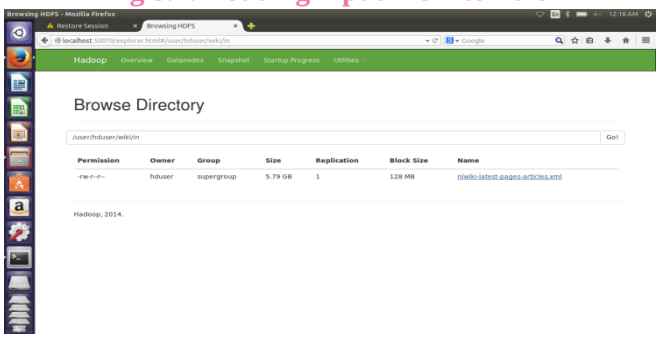

Fig 5.3: Input file

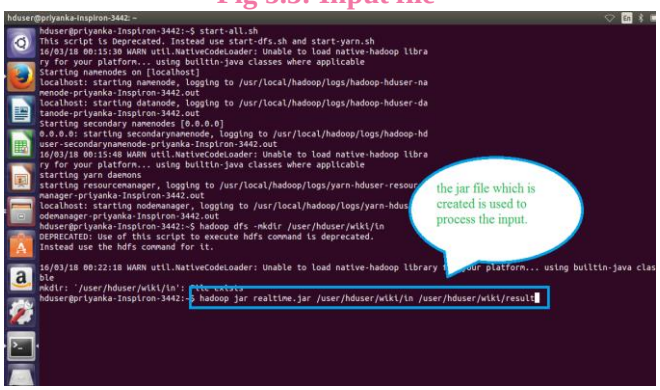

Fig 5.4: Running jar file to process input

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

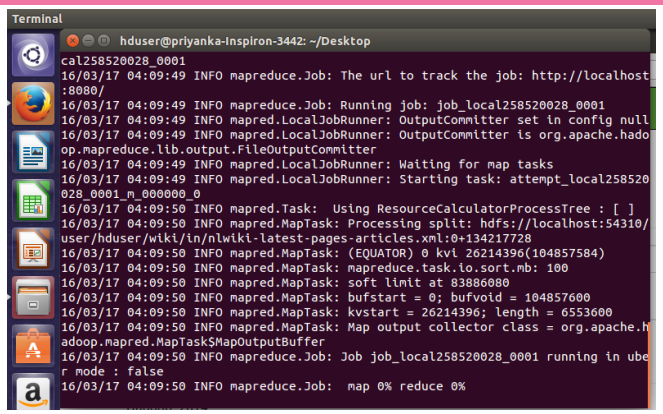

Fig 5.5.1: Output preprocessing

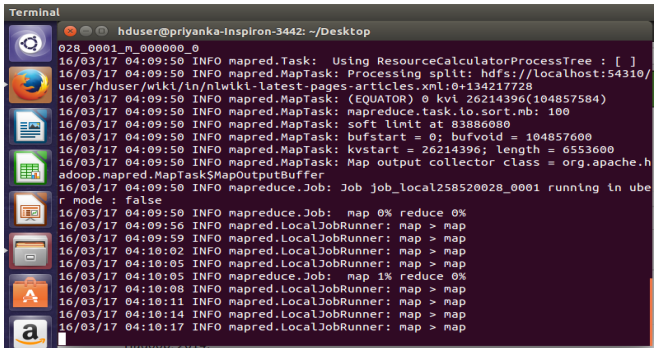

Fig 5.5.2: Output preprocessing

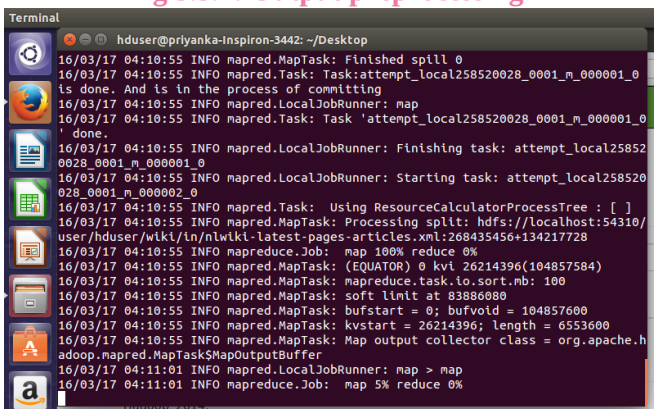

Fig 5.5.3: Out preprocessing

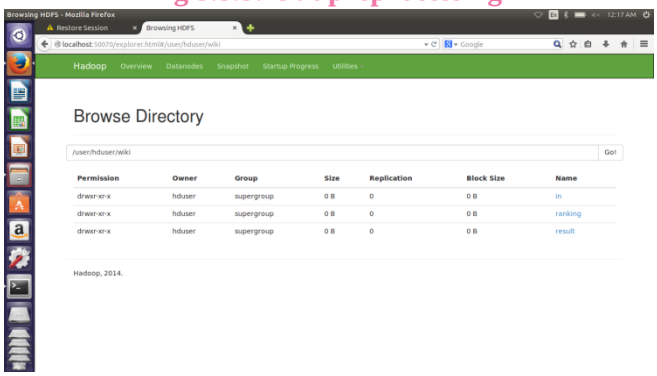

Fig 5.6: Output file generation

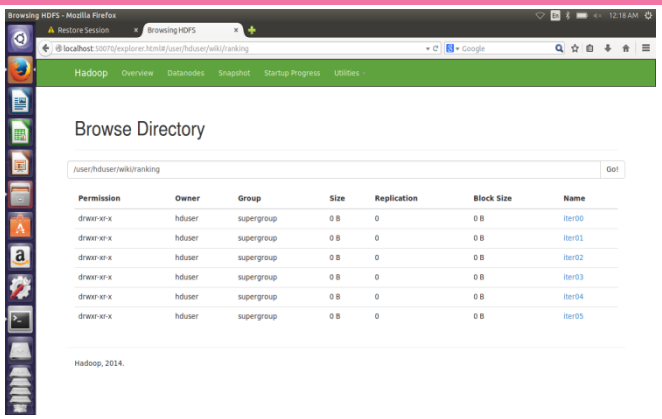

Fig 5.7: Intermediate results

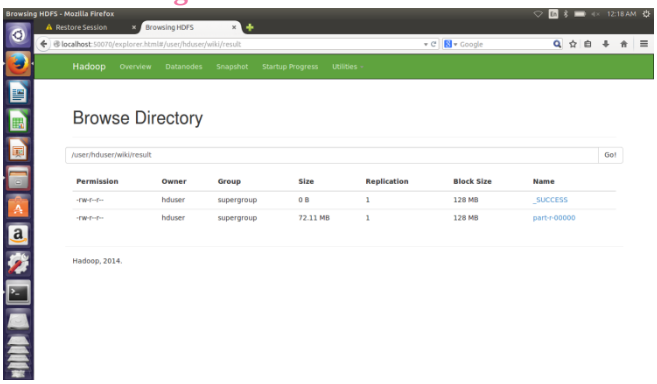

Fig 5.8: Final result

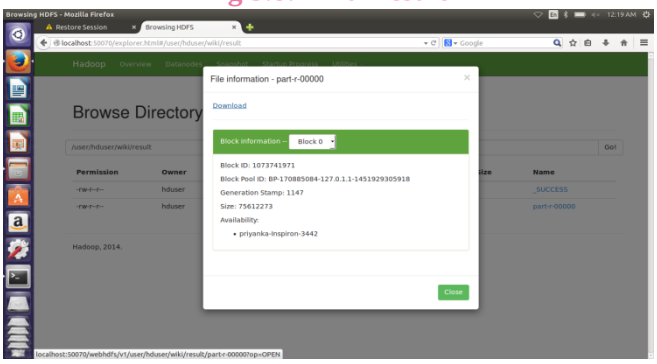

Fig 5.9: Download of final result

INPUT FILE SAMPLE:

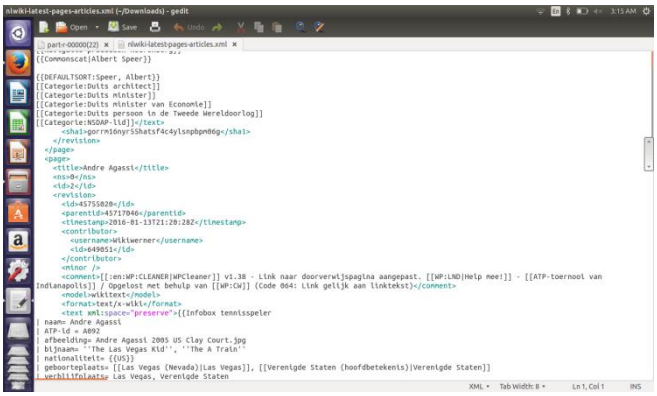

Fig 5.10: INPUT file

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

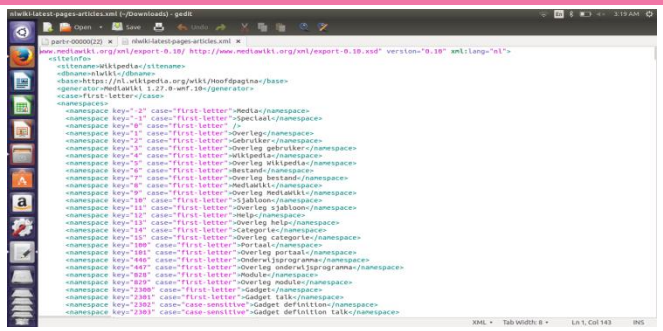

Fig 5.10.2: Input file

OUTPUT FILE SAMPLE:

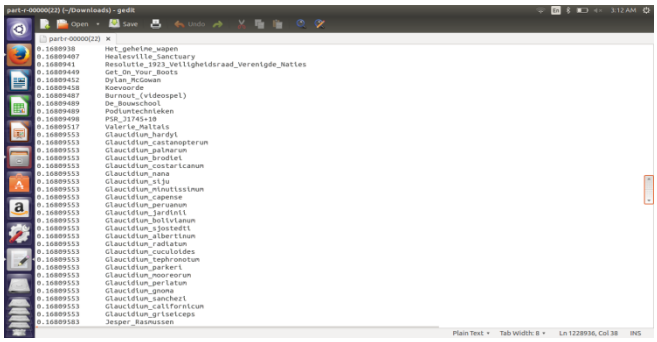

Fig 5.11: Output file

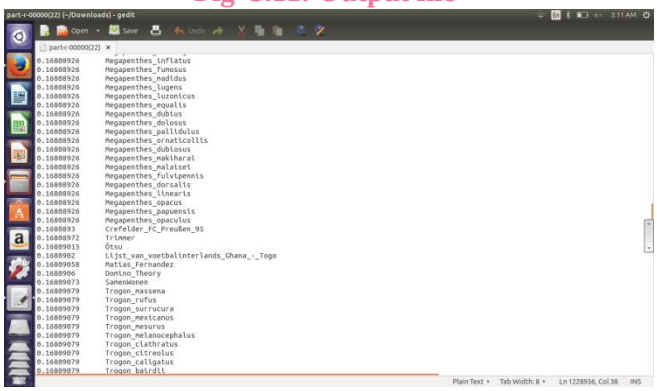

Fig 5.11.1: Output file

CONCLUSION

To improve the efficiency of graph analysis, some earlier work has been done on reducing the size of the input so that graph partitions are small enough to fit in the memory of a single cluster node.MapReduce is a distributed processing framework that enables data intensive computations. The framework, inspired by the functional programming paradigm, has two main components, a mapper and a reducer. A mapper works on each individual input record to generate intermediate results, which are grouped together based on some key and passed on to the reducers. A reducer works on the group of intermediate results associated with the same key and generates the final result using a result aggregation function. The processing units of the MapReduce framework are key-value pairs. An instance of the MapReduce framework with 3 mappers and 2 reducers.

Graph analysis in a distributed frameworks, such as Map- Reduce, is a challenge. There has been approaches for theanalysis of graph algorithms, but most of these approaches has a high communication cost because of the shuffling and sorting phases of the map-reduce. Our approach detaches the immutable graph topology from the analysis and as a result we get an improved performance as there is less communication cost.

FUTURE ENHANCEMENT

This project is done using Basic Implementation Algorithm to calculate the pagerank using Mapreduce in Single node cluster. In future we wish to run this project on Distributed node and even using Apache Spark.

Here it is more advantageous as works with Inmemory Computations which decreases the time taken for intermediate results storage in HDFS and its preprocessing for Final Output.

REFERENCES

[1] G. Salton, A. Wong, C. S. Yang. A vector space model for automatic indexing.Communications of the ACM, version.18 n.11, pages.613-620, Nov. 1975.

[2] Anna Huang," Similarity measures of Text document clustering", NZCSRSC 2008, Christchurch, New Zealand, April 2008.

[3] GeorgeTsatsaronis and Vicky Panagiotopoulou. A generalized vector space model for text retrieval based on semantic relatedness. Proceedings of the EACL 2009 student research workshop, pages 70-78, April 2009.

[4] J Dittrich, JA Quiané-Ruiz .Efficient big data processing in HadoopMapReduce. Proceedings of the VLDB Endowment, 2012 - dl.acm.org, Volume 5

Issue 12, August 2012,Volume 5 Issue 12, August 2012.

[5] WeizhongZhao,Huifang Ma, Qing He. Parallel KMeans Clustering Based on MapReduce. Processdings of First international conference on cloudcom 2009, pages 674-679, Beijing, China, December 1-4, 2009.

[6] Sanjay, G., G. Howard, and L. Shun-Tak, The Google file system, in Proceedings of the nineteenth ACM symposium on Operating systems principles. ACM: Bolton Lan, ding, NY, USA 2003.

[7] Jeffrey, D. and G. Sanjay,MapReduce: simplified data processing on large clusters. Commun. ACM, 51(1): pages 107113 2008. [8] Apache LuceneHadoop[EB/OL]. http://hadoop.apache.org/.