Load Rebalancing for Distributed File System using Hadoop

S. Dhrisya[1]R. Karthika[2]P. Soundharya[3]
[1,2,3]Student
[1,2,3]Department of Computer Science &Engineering
[1,2,3]P.A. College of Engineering & Technology, Coimbatore, Tamilnadu, India

Abstract—*Distributed file systems square measure key building for cloud computing applications supported the Map Reduce paradigm. The type of file systems, nodes are parallels their serve computing and storage functions, a file is separated into variety of chunks allotted in distinct nodes and MapReduce tasks is performed in parallel across the nodes. Files also be dynamically created, deleted and appended, due to imbalance in load of a distributed filing system, the file chunks are not distributed consistently as attainable among the nodes. In an exceedingly cloud computing setting, failure is that the norm, and nodes is upgraded, replaced, and supplemental within the system. Appearing distributed file systems in production system powerfully depend upon a central node for chunk reallocation. The dependence is clearly short in a very large-scale, failure–prone setting as a result of the shopper load balancer is anesthetize significant employment that is linearly scaled with size of the system, and the performance bottleneck and failure at single purpose. In this paper, a completely distributed load rebalancing rule are going to be given to wear down the load imbalance drawback. The performance of our proposal are going to be enforced within the Hadoop distributed filing system to scale back the movement value and recursive overhead.*

Keywords—*Apache maven, Clouds, DHT, HDFS, Load Balance, MapReduce*

I. INTRODUCTION

Cloud Computing could be a compelling technology. In clouds, purchasers assign the resources dynamically, with none readying and management of resources. Distributed classification system is ancient model of classification system that is employed within the variety of chunks for cloud computing. Cloud computing application relies on the practicality of MapReduce programming employed in distributed classification system. MapReduce is that the master-slave design in Hadoop. Master and slave act like Namenode and Datanode severally. Master takes massive drawback, divides it into sub drawback and assigns it to employee node to multiple slaves to resolve drawback one by one. In distributed classification system, an outsized file is split into variety of chunks and allocates every chunk to separate node to perform MapReduce perform parallel over every node.

Load reconciliation is that the main issue within the giant scale distributed systems attributable to increase in storage and networkLoad should be balance over multiple nodes to improve system performance, resource utilization, response time and stability. Load leveling is split into 2 categories: static and dynamic. In static load leveling algorithmic rule, it does not think about the previous behavior of a node whereas distribute the load. The dynamic load leveling rule, it checks the previous behavior of node whereas distribute the load.

The load rebalancing task is employed to eliminate the load on central node. In load reconciliation formula, storage nodes square measure structured over network supported the distributed hash table (DHT); DHTs change nodes to self-recognize and repair whereas it uniformly provides the search practicality in node here aim to cut back the movement price that is caused by load rebalancing of nodes to maximize the network information measure. Each Chunk server initial notice the light-weightnode or significant node. The large numbers of file chunks are incorporated from significant node to light weight node to balance their load. This method repeats till all significant nodes becomes the sunshine nodes. To beat this load equalization drawback every node perform load rebalancing algorithmic rule severally.

II. RELATED WORKS

Hadoop distributed filing system by KonstantinShvachko, Hairong Kuang, Sanjay Radia, Henry M. Robert Chansler in 2012[2] provides a distributed filing system and a framework for the analysis and transferring the terribly great deal of knowledge sets victimization the Map Reduce paradigm.. The algorithms of Load reconciliation is predicated on DHTs are extensively studied. Even so, most existing system solutions are designed while not each the both movement price and node non uniformity and take into account important maintenance network traffic to DHTs. G.Naveen, J.PraveenChander in 2015[9] the planned a reallocation of file chunks to decreases the movement value however conjointly exploits capable nodes toincrease the entire system performance. The algorithmic program reduces recursive overhead introduced to the DHTs the maximum amount as potential the results ar indicated to every node performs load rebalancing algorithmic program severally while not effort the worldwide information and exhibits a quick convergence rate. The storage nodes area unit build as a network depends on distributed hash tables (DHTs).

Secured load rebalancing in cloud rising distributed file systems in production systems heavily rely on a central node for chunk reallocation. This dependence is clearly inadequate in a very large-scale, failure-prone atmosphere so the central load balancer is drug sizable work that is linearly scaled with size of the system, and will therefore become the performance bottleneck and also the single purpose of failure.Bhavya.G, Chaitra.B.N, Chaitra.G, Shruthi in 2015[8] planned a completely distributed load rebalancing algorithmic program is given to influence the load imbalance drawback. Aim to scale back network traffic or movement value caused by rebalancing the masses of nodes the maximum amount as doable to maximize the network information measure out there to traditional applications. As failure is that the norm, nodes square measure new supplemental to sustain the general system

performance leading to the heterogeneousness of nodes. In the system conjointly give security for the info hold on on cloud through secret writing and decipherment ideas. The data to be saved in the cloud is encrypted earlier than storage for more protection. When the server manage performs operations on facts like deletion or updating load imbalance trouble takes place. This hassle can be solved by using the rebalancing algorithm which balances the load in the cloud after the above operations carried out Encryption of data, Splitting the records and Sending Data to Cloud [1],[2],[6] . The encryption system is carried out over the statistics through AES algorithm. The encrypted file is made in chunks and saved in a range of nodes. The encrypted file is partitioned into a wide variety of chunks and is allotted in wonderful nodes. The splitter documents are stored in the cloud and can be accessed from anywhere every time wished. Thus storing a single file in a range of nodes has greater protection when in contrast to the file that is stored in a single node.

The drawback of existing approach is too designed besides considering each motion price and node heterogeneity and additionally introduce significance of keeping the network visitors in DHTs. For getting the higher result, proposed algorithm the nodes are function in an allotted sequence in which nodes handles their load-balancing tasks independently except thinking about the international understanding related to the system

III. DOMAIN:PARALLEL &DISTRIBUTED SYSTEMS

The evolution of Internet, the big amount of statistics is growing widely, and the storage and processing capability of large set of facts grow to be a trouble. Cloudcomputing is the biggest solution for the problem. Cloud two computing is the largest solution for the problem. Cloud computing gives low cost of hardware useful resource and improves the utilization of machine equipment. In distributed file structures the chunk file of every node is at once proportional to the load of a node. The substitute of node and addition of nodes in the file system are happen by means of the advent and deletion of file in clouds, the chunks of every archives are not separated as commonly as viable among the nodes.

A. Cloud Computing

Cloud computing is largest outsourcing of computer services. It has a potential to set up on wanted resources in community. Essential series of assets like network, storage, central processing unit and memory are supplied by means of cloud computing.

As cloud computing is one of the essential science to grant an easily gaining access to the community and helps open source cloud Provider.

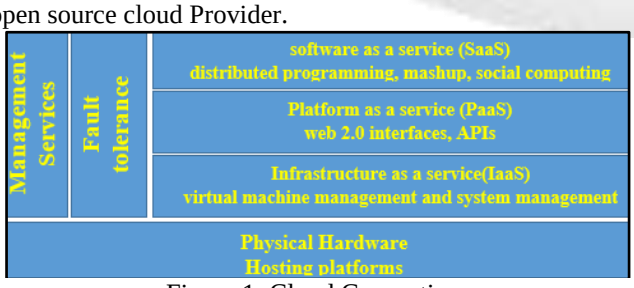


Figure 1: Cloud Computing

In fig 1 shows three cloud computing services:

1) Infrastructure as a service (IaaS)
2) Software as a service (SaaS)
3) Platform as a service (PaaS)

To be aware of about famous variations of cloud choices are Software-as-a-Service (SaaS) and Infrastructure-as-a-Service (IaaS). The SaaS with cloud Carrier Company hosts your organization functions and related information on its servers and storage structures. Users reap get admission to SaaS applications the usage of a Web browser, and your organization usually pay a rate per person per month, with IaaS, the provider provides digital machines, bodily servers, storage, switching, and connectivity sources to run your agency features on a pay-as-you-go basis. Platform as a provider (PaaS) offers a platform permitting customers to develop, run, and manipulate purposes barring the complexity of building and protecting the infrastructure typically associated with developing and launching an app. PaaS encompass services for utility design, application development, trying out and deployment as nicely as offerings such as team collaboration, web service integration, security, storage, state administration.

B. Mapreduce

MapReduce has been facilitated Google as a programming framework to analyze large amounts of data. It makes use of for allotted records processing on giant datasets across a cluster of machines. The input data is too large, the computation desires to be distributed across heaps of machines within a cluster in order to end each phase of computation in a realistic amount of time. The disbursed idea implies to parallelize computations without difficulty and the use of are execution as the foremost approach for fault tolerance. Google in no way launched their execution of MapReduce[6]. Finally, the Apache Company made handy a concrete implementation of MapReduce named HadoopDevelopers can processing the difficult computations in a convenient way through the usage of mapreduce and also hides the functions of records distribution, parallelization, and fault tolerance.

The special characteristic of MapReduce is that it can each interpret and analyze each structured and unstructured information throughout many nodes through using of an allotted share nothing structure. Share nothing architecture is a distributed computing structure consisting of more than one nodes. Each node is independent, and its own disks, memory, and I/O devices in the community. The type of architecture, every node is self-sufficient and shares nothing over the network, there are no points of rivalry throughout the system. In Figure 2. MapReduce programming model drives from three critical phases:

1) Map phase

Partition into M Map characteristic (Mapper); every Mapper runs in parallel. The outputs of Map segment are intermediate key and value pairs.

2) Shuffle & Sort

The output of each Mapper is partitioned with the aid of hashing the output key. In this phase, the wide variety of partitions is equal to the number of reducers; all key and value pairs in shuffle section share the same key that belongs to the same partition. After partitioning the Map

output, each partition is saved with the aid of a key to merge all values for that key.

After partitioning the Map output, each partition is stored by a key to merge all values for that key. MapReduce libraries written in numerous programming languages, include, LISP, Java, C++, Python, Ruby and C. A presentation of the MapReduce workflow includes; a dataset is divided into a number of devices of records and then each unit of statistics is processed in a Map phase. Finally, the blended in Reduce segment to produce the final output.

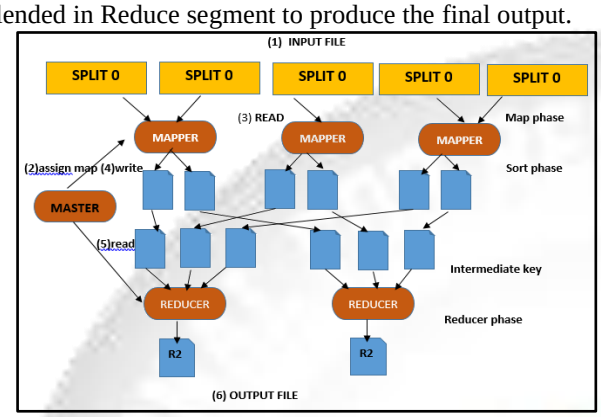


Fig. 2: Mapreduce Workflow

1) Execution starts via strolling the MapReduce program.
2) One replica of the application placed on a laptop is particular and that computing device is known as the grasp node. The relaxation of the program is assigned to employee (slave) nodes through grasp node.
3) The Mapper characteristic receives a Map task, read the content material of a chunk of information and invoke the user defined Map function.
4) The output information in the temporary buffers are saved on the neighborhood disk and the bodily reminiscence addresses of these buffered facts are dispatched to the master node.
5) A Reducer feature informs by way of grasp node about the bodily reminiscence addresses; it uses a remote method to get admission to the buffered data from the Mappers on the neighborhood disks.
6) The Reducer sends every special key and its consequent set of intermediate values to the Reduce function. The final output is on hand in the Reducer; then it is saved to the disbursed file system (HDFS).

C. Hadooop

Apache Hadoop is a group of two open-source software that supports the processing and managing the massive datasets in allotted manner. It is one of the most fascinating third-party implementations of Map Reduce for distributed computing. Hadoop affords a framework to assist massive dataset dispensed computing on a cluster of servers. Hadoop was stimulated in 2006 by Doug Cutting(named via his son's stuffed elephant) now being used with the aid of predominant companies, which includes Amazon, two IMB, Yahoo, Facebook and a developing range of different agencies. Hadoop gives its own disbursed file system referred to as Hadoop Distributed File System (HDFS), HDFS is a allotted file gadget designed to shop several copies of data block in a cluster nodes, to allow legitimate and expeditious computations. A easy Hadoop cluster consists of n≥1 machines walking the Hadoop software.

Hadoop makes use of HDFS to keep a dataset and applies the MapReduce's electricity to distribute and parallelize the processing of this dataset. Stream information is first fragmented into common HDFS-sized block (64MB) and then smaller packets (64KB) through the user. It capability a file into HDFS is divided into sixty four MB chunk and every chunk is resided on a specific working machine

D. Introduction to Hadoop

Hadoop has allows for distributed storage and processing the large quantity of datasets the usage of map decrease function. The essential function of Hadoop is the segregating of statistics and computation throughout lots of host nodes and executing the information in parallelly as shown in figure 3

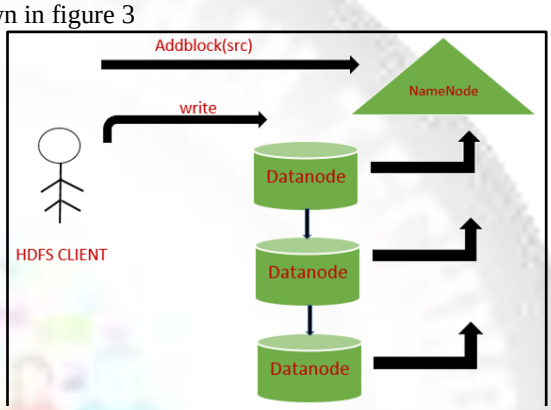


Fig.3: Overview of Hadoop

IV. HADOOP DISTRIBUTED FILE SYSTEM

The Hadoop Distributed File System (HDFS) [4], [7] is a allotted file device designed to run on commodity hardware. The typically distinction between HDFS and other Distributed File System are:

- The hardware are made in low fee components.
- HDFS produce excessive throughput and fault-tolerance for appropriate utility datasets.
- HDFS once firstly constructed as infrastructure for the Apache Nutch web search engine project.

HDFS, separate file machine for storing software program information and metadata [8], [9]. HDFS structure consists of NameNode, DataNode, and HDFS Client. A HDFS Cluster consist of DataNode and NameNode, every DataNode excute the huge quantity of software concurrently. The fundamental factors of HDFS are that, it is exceedingly fault tolerance, appropriate for functions with large data units. Hadoop Distributed File System Architecture shown in figure: 4

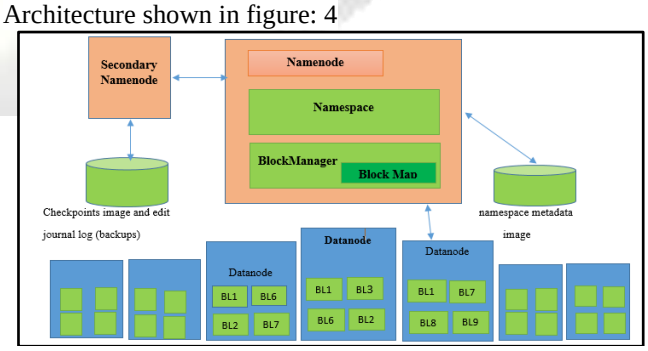


Fig.4:HDFS Working Scenario

In master/slave structure of HDFS, consists of single NameNode. The main work of hold close server that manages the complete file system of namespace and manipulate to gaining get right of entry tothe file [5]. HDFS namespace is a hierarchy of documents and directories. The files and directories, attribute like permissions, modification, get entry to time namespace and disk house quotas. A file is separate into one or extra blocks and set of blocks are saved in DataNodes. A DataNodes stores facts in the archives in its nearby gadget and check knowledge about HDFS file system. In HDFS every block of records are disbursed as separate file.

V. DISTRIBUTED HASH TABLE

A distributed hash table (DHT) is a type of a decentralized allotted device that offers a lookup service comparable to a hash table: (key, value) pairs are stored in a DHT, and any collaborating node can effectively retrieve the cost related with a given key. Responsibility for preserving the mapping from keys to values is distributed amongst the nodes, in a way that a exchange in the set of participants motives a minimal quantity of disruption [17]. The permits a DHT to scale to extremely giant numbers of nodes and to handle continual node arrivals, departures, and failures.

The storage nodes are structured as a network based totally on dispensed hash tables (DHTs), discovering a file chunk can genuinely refer to rapid key look up in DHTs, given that a unique handle (or identifier) is assigned to every file chunks. DHTs allow nodes to self-organize and Repair continuously presenting seem up performance in node dynamism. DHTs assurance that if a node leaves, then its regionally hosted chunks are reliably migrated to its successor; if a node joins, it allocates the chunks whose IDs without delay precede the becoming a member of node from its successor to manage. Our inspiration closely relies upon on the node arrival and departure operations to migrate file chunks amongst nodes, its simplifying the machine provision and management. The chunk servers in our concept model are prepared as a DHT community. The ordinary DHTs assurance that a node leaves, then it is locally hosted file chunks are reliably migrated to its successor; if a node joins, then it allocates the chunks whose IDs are at once precede the becoming a member of node from its successor to manage.

VI. APACHE MAVEN

Apache Maven is a software program assignment administration and comprehension tool. Based on the idea of a project object model (POM), Maven manage a project's build, reporting and documentation from a central piece of facts.

Maven is a construct automation device used specifically for Java tasks. The word maven capacity "accumulator of knowledge" in Yiddish. Maven addresses two factors of building software: First, it describes the building structure of software, and second, it describes its dependencies. Contrary to previous tools like Apache First, it describes how software program application is built, and second, it describes its dependencies Ant, it uses conventions for the construct procedure, and only exceptions want to be written down. An XML file describes the software program project being built, its dependencies on different exterior modules and components, the construct order, directories, and required plug-ins. It comes with pre-defined goals for performing certain well-defined duties such as compilation of code and its packaging. Maven dynamically downloads Java libraries and Maven plug-ins from one or greater repositories such as the Maven 2 Central Repository, and shops them in a nearby cache.[4] This nearby cache of downloaded artifacts can additionally be up to date with artifacts created by way of nearby projects. Public repositories can additionally be updated

A. Project Object Model

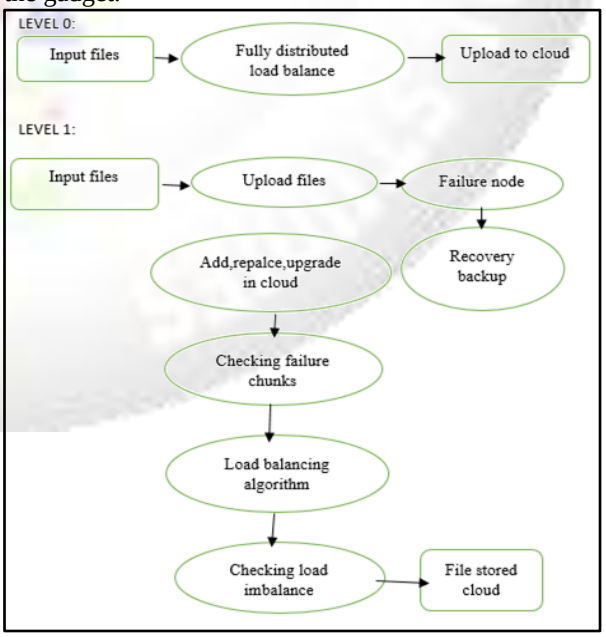
A Project Object Model (POM) affords all the configuration for a single project. General configuration covers the project's name, its proprietor and its dependencies on different initiatives. One can additionally configure person phases of the construct process, which are carried out as plugins. One can configure the compiler-plugin to use Java model 1.5 for compilation, or specify packaging the venture even if some unit assessments fail.

Larger initiatives ought to be divided into numerous modules, or sub-projects, each with its own POM.Write a root POM through, one can bring together all the modules in a single command. POMs additionally inherit configuration from different POMs. All POMs inherit from the Super POM by default. The Super POM presents default configuration, such as default supply directories, default plugins, and so on.

VII. SYSTEM DESIGN

A. Data Flow Diagram (DFD)

Data go with the flow diagrams symbolize one of the most ingenious equipment used for structured evaluation. A Data Flow design or (DFD) as it is rapidly as is additionally acknowledged as a bubble chart. The bubble represents statistics transformations and lines symbolize statistics waft in the gadget.



VIII. EXPERIMENTAL RESULT

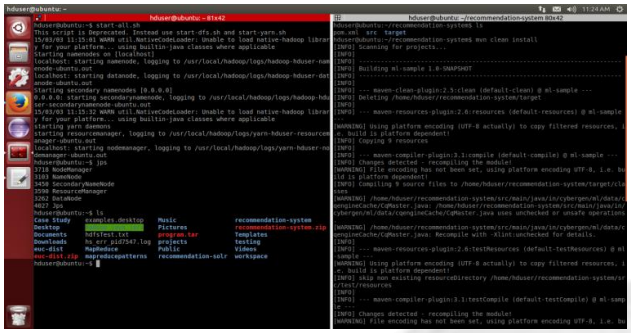


Fig. 5: Loading Hadoop(left), loading libraries using Apache Maven(right)

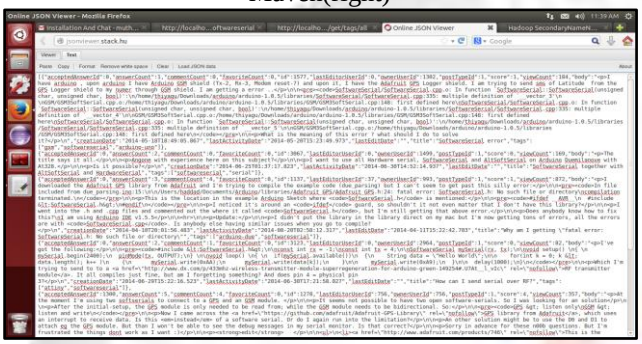


Fig. 6:File viewed using JSON viewer

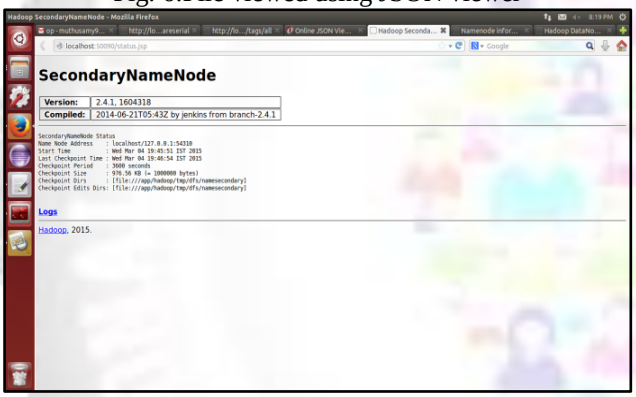


Fig. 7: Secondary Namenode Details

Fig. 8: Datanode detail

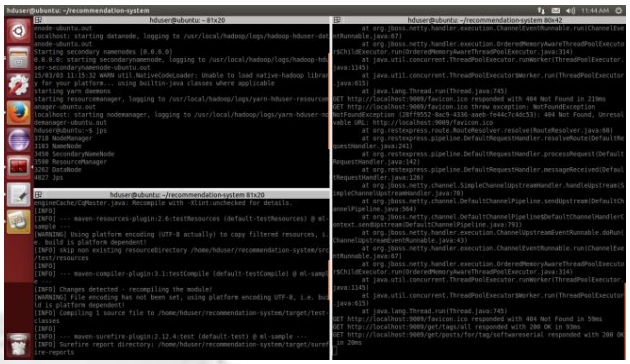


Fig. 9:Showing Time Taken for Distributing a File

IX. CONCLUSION

A novel load-balancing algorithm to deal with the load-rebalancing hassle in large-scale, dynamic, and dispensed file structures in clouds introduced in this undertaking. Our notion strives to stability the masses of nodes and minimize the demanded motion cost as a great deal as possible, whilst taking gain of physical community locality and node heterogeneity. Investigated the overall performance of our idea and in contrast it in opposition to competing algorithms are synthesized probabilistic distributions of file chunks. Load-balancing algorithm famous a speedy convergence rate. The effectivity and effectiveness of our sketch are similarly validated by way of analytical fashions and a actual implementation with a small-scale cluster surroundings

REFERENCES

[1] Load Rebalancing for Distributed File Systems in Clouds Hung-Chang Hsiao, Member, IEEE Computer Society, Hsueh-Yi Chung,Haiying Shen, Member, IEEE, and Yu-Chang Chao 2013

[2] The Hadoop Distributed File System KonstantinShvachko, Haring Kuang, Sanjay Radia, Robert Chansler,2012

[3] Study of Hadoop Distributed File system in Cloud Computing Radhika M. Kharode, Anuradha R. Deshmukh, IJARCSSE,2015

[4] Survey on Load Rebalancing for Distributed File System in Cloud Prof. Pranalini S. KetkarAnkita BhimraoPatkureIJIRAE ,2014

[5] Review of Load Balancing for Distributed Systems in Cloud Radha G. Dobale*, Prof. R. P. SonarIJARCSSE, 2015

[6] Distributed Meta data Management sssScheme in HDFS ,MrudulaVarade, VimlaJethani, ijsrp,2013

[7] Enhance Load Rebalance Algorithm for Distributed File Systems in Clouds Kokilavani .K, IJEIT, 2013

[8] Secured Load Rebalancing In CloudBhavya.G,Chaitra.B.N,Chaitra.G, Shruthi.C,IJARCSSE,2015

[9] Implementation of DHT on Load Rebalancing In Cloud ComputingG.Naveen, J.PraveenChander PG,ICMEET 2015

[10] large-scale data processing using mapreduce in cloud computing environment Samira Daneshyar and Majid Razmjoo, IJWSC,2012

[11] An Improved Hadoop Data Load Balancing Algorithm Kun Liu 1, 2, Gaochao Xu , and Jun'e Yuan , JOURNAL OF NETWORKS
[12] G, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, "Dynamo: Amazon's Highly Available Key-Value Store," Proc. 21st ACM Symp. Operating Systems Principles (SOSP '07), pp. 205-220, Oct. 2007.
[13] Rao, K. Lakshminarayanan, S. Surana, R. Karp, and I. Stoica, "Load Balancing in Structured P2P Systems," Proc. Second Int'l Workshop Peer-to-Peer Systems (IPTPS '02), pp. 68-79, Feb. 2003.
[14] D. Karger and M. Ruhl, "Simple Efficient Load Balancing Algorithms for Peer-to-Peer Systems," Proc. 16th ACM Symp. Parallel Algorithms and Architectures (SPAA '04), pp. 36-43, June 2004.
[15] P. Ganesan, M. Bawa, and H. Garcia-Molina, "Online Balancing of Range-Partitioned Data with Applications to Peer-to-Peer Systems," Proc. 13th Int'l Conf. Very Large Data Bases (VLDB '04), pp. 444-455, Sept. 2004.
[16] J.W. Byers, J. Considine, and M. Mitzenmacher, "Simple Load Balancing for Distributed Hash Tables," Proc. First Int'l Workshop Peer-to-Peer Systems (IPTPS '03), pp. 80-87, Feb. 2003.
[17] G.S. Manku, "Balanced Binary Trees for ID Management and Load Balance in Distributed Hash Tables," Proc. 23rd ACM Symp. Principles Distributed Computing (PODC '04), pp. 197-205, July 2004.
[18] Bharambe, M. Agrawal, and S. Seshan, "Mercury: Supporting Scalable Multi-Attribute Range Queries," Proc. ACM SIGCOMM '04, pp. 353-366, Aug. 2004.
[19] Y. Zhu and Y. Hu, "Efficient, Proximity-Aware Load Balancing for DHT-Based P2P Systems," IEEE Trans. Parallel and Distributed Systems, vol. 16, no. 4, pp. 349-361, Apr. 2005.
[20] H. Shen and C.-Z. Xu, "Locality-Aware and Churn-Resilient Load Balancing Algorithms in Structured P2P Networks," IEEE Trans. Parallel and Distributed Systems, vol. 18, no. 6, pp. 849-862, June 2007.
[21] Q.H. Vu, B.C. Ooi, M. Rinard, and K.-L. Tan, "Histogram-Based Global Load Balancing in Structured Peer-to-Peer Systems," IEEE Trans. Knowledge Data Eng., vol. 21, no. 4, pp. 595-608, Apr. 2009.
[22] H.-C. Hsiao, H. Liao, S.-S. Chen, and K.-C. Huang, "Load Balance with Imperfect Information in Structured Peer-to-Peer Systems," IEEE Trans. Parallel Distributed Systems, vol. 22, no. 4, pp. 634-649, Apr. 2011.
[23] Karger, D., Lehman, E., Leighton, T., Panigahy, R., Levine, M. and Lewin, D, "Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the World Wide Web", In Proceedings of the TwentyNinth Annual ACM Symposium on theory of Computing, ACM Press, New York, 1997, pp. 654-663.
[24] Bing Li, Yutao He, Ke Xu, "Distributed Metadata Management Scheme in Cloud Computing ", In Proceedings of IEEE in PCN&CAD CENTER, Beijing University of Post and Telecommunication, China, 2011.
[25] Sage A. Weil, Kristal T. Pollack, Scott A. Brandt, Ethan L.Miller ,"Dynamic Metadata Management for Petabyte-scale File Systems ", In Proceedings of IEEE University of California, 2004.