Survey on MapReduce with Scheduling and Partitioning In Big Data

C.Nandhini[1], P.Premadevi[2]

[1]*PG Scholar*, [2]*Assistant Professor*

[1,2]*Angel College of Engineering and Technology, Tamil Nadu, India*

[1]Email Id: nandhini28cse@gmail.com

[2]Email Id: prema.acetcse@gmail.com

***Abstract*—Big data is used to describe a massive volume of both structured and unstructured data that is so large that it's difficult to process using traditional database and software techniques. MapReduce is programming model is implemented that helps developer to process massive amount of structured and unstructured data in parallel manner across heterogeneous environment. It was developed by Google in 2004 Hadoop is used to implement the MapReduce. The MapReduce programming model mainly used for load balancing in Cloud environment. This paper provides an overview of MapReduce Programming model with scheduling and partitioning in Big data**

***Keywords*— Big data, MapReduce, Hadoop, scheduling, load balancing.**

I. INTRODUCTION

Big data [18] is a collection of large data sets. Due to their giant size it is not efficient to process those using traditional methods. Various problems that have to be faced in processing big data are capturing the data, storage, search, sharing, transferring, analysis etc. The trend to giant data sets is because of extra information derived from analysis of a single large set of correlated data, as compared to separate smaller sets with the equivalent total amount of data. This big data helps in finding the relations between various fields which may help in various ways, like decision making, understanding the business trends, long term planning, fighting crime, and getting real-time roadway traffic conditions[28]. But due to their correlated behavior it becomes difficult to query them [35]. Professionals are trying to make results from this huge amount of data.

A. Hadoop Overview

Hadoop [33] is the Apache Software Foundation open source and Java-based implementation of the Map/Reduce framework. Hadoop was created by Doug Cutting. Hadoop originated from Apache Nutch3 which is an open source web search engine was a part of the Lucene project. Nutch was an ambitious project started in 2002, and it soon ran into problems with the creators realizing that the architecture they had developed would not scale to the billions of web pages on the Internet. But in 2003, a paper was published that described Google's distributed file system - the Google File System (GFS) [31]. Hadoop was born in February 2006, when they decided to move NDFS and Nutch under a separate subproject under Lucene. In January 2008, Hadoop made its own top level project under Apache and HDFS or Hadoop Distributed File System was name kept instead of NDFS. Hadoop provides various tools which help in processing of vast amounts of data using the Map/Reduce framework and, additionally, implements the Hadoop Distributed File System (HDFS).

B. Hadoop Distributed File System Or HDFS

HDFS is a filesystem which is designed for storing very giant files with streaming data access patterns. HDFS runs on clusters on commodity hardware. HDFS was designed keeping in mind the ideas behind Map/Reduce and Hadoop. This implies that it is capable of handling datasets of much bigger size than conventional file systems (even petabytes). These datasets are divided into blocks and stored across a cluster of machines which run the Map/Reduce or Hadoop jobs. This helps the Hadoop framework to partition the work in such a way that data access is local as much as possible.

A very important feature of the HDFS is its "streaming access". Once the data is generated and loaded on to the HDFS, it assumes that each analysis will have a large proportion of the dataset. So the time taken to read the whole of dataset is more important than the latency occurred in reading the first record. This has its advantages and disadvantages. On one hand, it can read bigger chunks of contiguous data locations very fast, but on the other hand, random seek turns out to be a so slow that it is highly advisable to avoid it. Hence, applications for which low-latency access to data is critical will not perform well with HDFS.

C. MapReduce

MapReduce [15] is a parallel programming technique which is used for processing very large amounts of data. Processing of this much large amount of data can be done efficiently only if it is done in a parallel way. Each machine handles a small part of the data. MapReduce is a programming model that allows the user to concentrate more on code writing [22]. He needs not be worried about the concepts of parallel programming like how he will

distribute the data to different machines or what will happen on failure of any machine etc. This all is inbuilt in MapReduce framework.

In MapReduce, the input work is divided into various independent chunks [34]. These chunks are processed by the Mappers in the totally parallel way. The Mappers process the chunks and produce the output, and then this output is sorted and fed as input to the reducers. The framework itself will handle scheduling of tasks, monitoring of tasks and re-executing the tasks which have been failed [23]. MapReduce allows a very high aggregate bandwidth across the cluster because typically the nodes that store and the nodes that compute are the same. Means the MapReduce framework and distributed file system runs on the same set nodes due which it provides high aggregate bandwidth.

The input to this framework is provided in the form of key/value pairs and it produces the output also in the form of key/value pairs. All the coding is done in the form of two functions that is Map and Reduce. So the developers must have knowledge that how he will represent his coding in the form of Map and Reduce functions [12]. The output of the Map function is sorted and fed to the Reduce function. MapReduce is very suitable when there is a large amount of data involved. For small amount of data it might not be that useful. MapReduce can be run on the commodity hardware with reliability and in a fault tolerant way. Therefore it can be said that MapReduce is a software framework which allows writing the programs with a ease which involves large amount of data and the developer need not to worry about anything else when his data is being processed parallelly.

Before MapReduce large scale data processing was difficult because before that one had to look after the following factors himself but now they all are handled by the framework.

- Managing thousands or hundreds of processors
- I/O Scheduling
- Status and monitoring
- Managing parallelization and distribution
- Fault/crash tolerance

MapReduce provides all of these easily because these all are inbuilt in MapReduce framework [10]. Before MapReduce one needed to have a complete knowledge about all these factors, had to do separate coding for them like how to distribute the data if any node fails, keeping a check on processors etc. But now with MapReduce all that is inbuilt, user only can concentrate on his own implementation.

II. SCHEDULING IN HADOOP

The important process in Hadoop is assigning Task among the Map and Reduce nodes. The responsible for task assignment is scheduler. There are two types of nodes called JobTracker and TaskTracker are playing important role in the process of job execution. The Coordinator is JobTracker for executing all jobs in machine while the TaskTracker processing the tasks and sends heartbeat signal continuously to the JobTracker to indicate it's alive.

A. Default FIFO Scheduler

The Google was implemented the FIFO scheduler. All the Hadoop applications use the FIDO as default Scheduler. The jobs are present in the queue which can be processed as First in First out fashion. The main drawback with this FIFO scheduling is the next job or task in the job queue will be allocated only after finishing the previous job. The implementation of the Scheduler is efficient and very simple.

B. Fair Scheduler

The Fair Scheduler [7], [17], [19] was developed at Facebook. The possible solution with this scheduler is available resources can be shared effectively. The advantage of this scheduler is shorter job can be assigned when the slot becomes free. So the shortest job or smaller job no needs to wait for the completion of larger task. The main drawback is tasks are allocated to all the slots with higher slot capacity. Pools of jobs are present in fair scheduling. The Scheduler follows the following concept

1. Jobs are placed in the pool
2. Each pool is having maximum capacity
3. Resources are shared fairly.

C. Capacity Scheduler

Capacity Scheduler [17], [19] which was implemented by Yahoo and it is very effective for large scale applications. Instead of job pool, multiple queues are used and each queue has guaranteed capacity. The capacity of the queue is exceed means the task is allocated to another queue. Compare to lower priority jobs, the higher priority jobs can access more resources efficiently. The capacity Scheduler follows the priority concept with FIFO in the job queue. But the user having limitations on percentage of running tasks. So the cluster setup can be equally shared by the user.

D. Longest Approximation Time to End (LATE)

In LATE [28], the task can be executed speculatively. In this type of speculative execution the performance of the task execution is slow means another equivalent task can be running on back end. If the performance of the back end task is fast means performance of the scheduler is improved. Because the scheduler's response time will get improved. The LATE is robust in heterogeneous environment.

Features

- During speculative execution, the priority is assigned to the tasks.
- Faster node can be selected for speculative execution.
- Thrashing can be avoided because of speculative.

E. Delay Scheduler

The extended version of fair Scheduler is delay scheduler [24]. It is introduced to improve the data locality and performance of Fair Scheduler. During the allocation of task to the slot, it first checks the slot whether it having the available data for the completion of task. Else the task is delayed to get free slot which having the data for the particular task. This method is very efficient for the Task tracker to assign the task.

F. Dynamic priority scheduler

The dynamic priority scheduler [5] is proposed by Thomas sandholm. It takes user's priority .The distribution of capacity is performed dynamically. The extension of both FIFO and Fair scheduler is dynamic priority scheduler.The performance is better in large cluster. The advantage compared to default FIFO is fairness and jobs priority. It includes two phases

- Dynamic priority allocator- slot allocations dynamically
- Priority enforcer- It enforce the resource sharing

G. Resource aware scheduler

The resource utilization is focused such as I/O, CPU, Disk and Network. The job is assigned to the node which having less utilization of CPU. The performance management is main goal in Resource aware scheduler [14], [25].

H. Scheduler Based On Deadline Constraint

Many researchers are working on the Scheduling for improving the policies of scheduling in Hadoop. Recent works such as Dynamic proportional Scheduler, Delay Scheduler offer variety of service for Hadoop jobs allowing user to adjust the priority level assigned to their jobs. But it does not guarantee that the job will be processed by a specific deadline. So the Deadline Constraint Scheduler [3], [20] solves the issue of deadlines and focuses more on increasing system utilization [12].

Hadoop based data processing is done by a job execution cost model and takes user deadline as part of its input. The model determines the available slot based on set of assumptions:

- All nodes are homogeneous and processing cost for map and reduce node is equal.
- Input data is distributed in uniform manner.
- Reduce tasks starts after all map tasks have completed.
- The input data is already present in HDFS.

The job is scheduled based on the deadline and independent of the number of jobs running in cluster. After job is submitted it first computes the availability of free slots at given time running in the system. Then schedulability test is performed to decide whether the job can be completed within the deadline or not. The job is enlisted after the schedulability for assigning the job to slots.

I. Learning Scheduler

Learning methodology [13], [14] is applied on job to classify good or bad. For the purpose of classification pattern classifier is used. The classification is based on the resource utilization. If the job consumes less resources means it is good otherwise bad. Good jobs won't make any trouble to Task Tracker and bad jobs are terminated. It considers utilization such as CPU, I/O, Disk and network. Task assignment is very closer to default scheduler when more than one good job occurred.

J. Classification and Optimization based scheduler for Heterogeneous Hadoop system

Here both cluster and application level heterogeneity is considered [4] . By using this scheduler we can achieve better task assignment in heterogeneous environment. If it takes minimum completion time compare to other scheduler. The data locality can be reduced by replicating the data

K. Network Aware Scheduler

It is used to achieve the data locality by helping the administrator to find the data location for the task. The network awareness [26] is implemented with FIFO and Fair scheduler. The slot is checked to know the availability of the data before assigning the task. If there is no data for the task means it will be delayed for specific duration

III. PARTITIONING IN MAPREDUCE

In a MapReduce, partition takes place between the Mapper and reducer. The output of the Mapper is processed by the reducer with the help of partitioning mechanism. The role of partitioner is allocating the output of the Mapper to reducer. The performance of the MapReduce is based on the distribution of job among the nodes. The distribution process is based on the partitioning mechanism. By using better partitioning algorithm we can achieve the load balancing and memory consumption.

A. FIFO

The output of the Mapper is processed as FIFO fashion. Many Mappers are processing the task, For example consider there are 3 Mappers and 2 reducers. The first result is produced by second Mapper means it will be allocated to first reducer and vice versa. The implementation is simple and it is not effective.

B. Round Robin

Round robin model is better than FIFO. In round robin the output of the Mapper is treated as Round Robin manner. The Mapper produces the 3 results and number of reducer is 2. Here it fetches the first output to first reducer and second output to second reducer and third output to first reducer.

C. Hash Code

Most advanced and effective model is hash code. ASCII value is used. The hash function is computed on the output of Mapper. The count value of reducer

is added with the output of hash code. The final result indicates the proper reducer node.

IV. CONCLUSION

Big data is a collection of structured and unstructured data, which can be processed by using MapReduce programming model. This can be implemented on Hadoop platform. This paper contains the information about Big data and its challenges, Hadoop, HDFS and MapReduce Programming model. Some of the Scheduling and partitioning mechanisms are described in this paper.

REFERENCES

[1] Apache. Hadoop documentation, http://Hadoop.apache.org/core. 2008.

[2] Alex Nazaruk,Michael Rauchman "Big Data in Capital Markets" SIGMOD'13, June 22–27, 2013, New York

[3] AshayRaut, DhavalHansaliya, Rasesh Mori, ASVS Teja , "Deadline aware scheduler for Hadoop Yarn" in *Cloud Computing(*cse 565),Nov'12

[4] Aysan Rasoolia , Douglas G. Downa ,"COSHH: A Classification and Optimization based Scheduler for Heterogeneous Hadoop Systems " , in *Department of Computing and Software, McMaster University*, L8S 4K1, Canada.

[5] B. He, W. Fang, Q. Luo, N. K. Govindaraju, and T. Wang. Mars: a MapReduce framework on graphics processors. In PACT '08: Proceedings of the 17th international conference on Parallel architectures and compilation techniques, pages 260–269, New York, NY, USA, 2008. ACM.

[6] Hadoop's Capacity Scheduler: http://Hadoop.apache.org/core/docs/current/capacity_scheduler.html

[7] Hadoop's Fair Scheduler http://Hadoop.apache.org/common/docs/r0.20.2/fair_scheduler.html

[8] "Hadoop fundamentals" [online] http://bigdatauniversity.com/courses/

[9] Hieu Hanh Le, Satoshi Hikida, Haruo Yokota "NameNode and DataNode Coupling for a Power-Proportional Hadoop Distributed File System" DASFAA (2) Pg. 99-107, 2013

[10] Hui Jin,Kan Qiao,Xian-He Sun,Ying Li"Performance under Failures of MapReduce Applications" CCGRIDPg. 608-609,2011

[11] J. Dean and S. Ghemawat. MapReduce: simplified data processing on large clusters. Commun. ACM, 51(1):107–113, 2008

[12] Jeffrey Dean, Sanjay Ghemawat "MapReduce: simplified data processing on large clusters." Commun. ACM Vol. 51,2008

[13] Jaideep DattaDhok ,"Learning Based Admission Control and Task Assignment in MapReduce ", *Search and Information Extraction Lab ,International Institute of Information Technology Hyderabad* - 500 032, INDIA, June 2010 .

[14] Jaideep Dhok , Vasudeva Varma , "Using Pattern Classification for Task Assignment in MapReduce " , in *Search and Information Extraction Lab International Institute of Information Technology, Hyderabad*, India .

[15] Jens Dittrich, JorgeArnulfo Quian´eRuiz "Efficient Big Data Processing in Hadoop MapReduce" Proceedings of the VLDB Endowment, Vol. 5, No. 12, August 27th 31st 2012, Istanbul, Turkey.

[16] Jord Polo, Claris Castillo, David Carrera , Yolanda Becerra , Ian Whalley , Malgorzata Steinder , Jordi Torres , and Eduard Ayguad , "Resource-aware Adaptive Scheduling for MapReduce Clusters ", in *Barcelona Supercomputing Center (BSC) and Technical University of Catalonia*(UPC),December 2011.

[18] Hadoop job scheduling: http://blog.cloudera.com/blog/2008/11/job-scheduling-in-hadoop/

[19] Hadoop Scheduling: http://www.ibm.com/developerworks//library/os-hadoop-scheduling/

[20] Kamal Kc, KemaforAnyanwu , "Scheduling Hadoop Jobs to Meet Deadlines" *in Cloud Computing Technology and Science (CloudCom), IEEE Second International Conference* on Nov. 30 2010-Dec. 3 2010

[21] K. Kc and K. Anyanwu, "Scheduling Hadoop Jobs to Meet Deadlines", in Proc. CloudCom, 2010, pp.388-392.

[22] Kyuseok Shim "MapReduce Algorithms for Big Data Analysis" Proceedings of the VLDB Endowment, Vol. 5, No. 12, 2012

[23] Lizhe Wang,Jie Tao,Holger Marten,Achim Streit,Samee U. Khan,Joanna Kolodziej,Dan Chen "MapReduce across Distributed Clusters for Data-intensive Applications", IEEE 2012

[24] Matei Zaharia , Dhruba Borthakur, and Joydeep Sen Sarma, "Delay Scheduling: A Simple Technique for Achieving Locality and Fairness in Cluster Scheduling" , *Yahoo! Research ,University of California*, Berkeley , 2009.

[25] Mark Yong, Nitin Garegrat, Shiwali Mohan, "Towards a Resource Aware Scheduler in Hadoop", December 21, 2009.

[26] Matei Zaharia, Andy Konwinski, Anthony D.Joseph, Randy Katz, IonStoica, "Improving MapReduce Performance in Heterogeneous Environments", in *University of California*, Berkeley,2012.

[27] Marissa Mayer "The Coming Data Explosion" Richard MacManuscitedas http://www.readwriteweb.com/archives/the_coming_data_explosion.ph,May 30, 2010

[28] Nathan Eagle"Big data, global development, and complex social systems",ACMNew York, NY, USA2010

[29] Praveen kumar Kondikoppa , Chui-Hui Chiu, Cheng Cui and Lin Xue and Seung-Jong Park ,"Network-Aware Task Scheduling of MapReduce Framework over Multi-Clouds and High Speed Networks ", in *Department of Computer Science, Center for Computation & Technology, Louisiana State University*, LA,USA,2012.

[30] Pascal Hitzler and Krzysztof Janowicz "Linked Data, Big Data, and the 4th Paradigm" IOS Press and the authors SemanticWeb 4,2013

[31] Sanjay Ghemawat,Howard Gobioff,Shun-Tak Leung"The Google file system" SOSPPg. 29-43,2003

[32] Thomas Sandholm,Kevin Lai ,"Dynamic Proportional Share Scheduling in Hadoop ", in *Social Computing Lab, Hewlett-Packard Labs, Palo Alto*, CA 94304, USA,2010.

[33] "What is Hadoop"[online] http://Hadoop.apache.org/

[34] Wei Pan, Zhanhuai Li,Qun Chen,Shanglian Peng,Bo Suo,Jian Xu "MSMapper: An Adaptive Split Assignment Scheme for MapReduce" Springer Berlin Heidelberg,2012

[35] Wenfei Fan"Querying Big Social Data" Springer Berlin Heidelberg BNCOD2013

AUTHORS

C.Nandhini pursuing M.E. Computer Science and Engineering in Anna University, Chennai. She did her B.E., Computer Science and Engineering in Anna University of Technology, Coimbatore. She is a member of MISTE, IAENG. Her area of interest is Database, Data Mining and Big data. She has presented 3 papers in National Conferences and attended various

seminars and workshops to improve her knowledge in various domains.

P.Premadevi working as a Assistant Professor in Angel College of Engineering and Technology, Tirupur. She did her M.Tech., Computer Science and Engineering in Anna University of Technology, Coimbatore. . She did her B.E., Computer Science and Engineering in Anna University, Chennai. She is a member ISTE, IAENG. Her area of interest is Database, Data Mining and Big data. She has published 3 papers in various journals and presented 5 papers in National and International conferences and attended many workshops, seminars and FDP.