Survey on MapReduce and Scheduling Algorithms in Hadoop

Rakesh Varma

Information Technology Department, B & B Institute of Technology, Vallabh Vidyanagar, India

Abstract: ***We are living in the data world. It is not easy to measure the total volume of data stored electronically. They are in the unit of zettabytes or exabytes referred as Big Data. It can be unstructured, structured or semi structured, they are not convenient to store as well as process with normal data management methods and with machine having limited computational power. Hadoop system is used to process large datasets with efficient and inexpensive manner. MapReduce program is used to collect data as per the request. To process large volume of data proper scheduling is required to achieve greater performance. The objective of the research is to study MapReduce and analyze different scheduling algorithms that can be used to achieve better performance in scheduling.***

Keywords: Big Data, Hadoop, HDFS, MapReduce, Scheduling Algorithms

1. Introduction

To store and process the data in the unit of terabytes or petabytes means large datasets having different scale and complexity which requires new architecture, innovative processing techniques and intelligent algorithms for proper scheduling [8] in Hadoop. To process the data of modern era and future various big data processing methods are introduced. Hadoop is the open source software founded by Apache [2]. It is Linux based software now days used by famous websites like Google, Yahoo, Facebook, Amazon, AOL and many more. It is a framework for processing voluminous datasets. Hadoop provides better storage capacity for large datasets and performs parallel processing of big data that gives better computational power to all the job tasks. It works in batch processing mode. It has two major components HDFS [3] [4] (Hadoop Distributed File System) for huge data storage and MapReduce for processing huge amount of datasets [5].

2. Big Data

Data beyond to the storage capacity and beyond to the processing power is referred as Big Data. The flood of data is coming from various sources in the world. Facebook hosts near about 10 billion of photos which taken one petabytes of storage. The NYSE (NewYork Stock Exchange) generates one terabytes of new trade data per day. And the latest the large Harden Collider in Switzerland will produce about 15 petabytes of data per year [5]. On-line retail transactions, various sensor networks, Air lines route recording, GPS based vehicle tracking system, RFID readers, machine logs, CC cameras all these are responsible for growing of big mountain of data.

The amounts of data generated by machines are greater than that generated by people. Whenever the data size increases they are creating problems for existing algorithm for processing. So it’s a biggest challenge to store and process the large amount of data sets. Hadoop is introduced to store and process the huge amount of data in less time.

3. Hadoop

Hadoop was created by Doug Cutting the creator of open source search technology. It is an open source parallel processing framework. In the year 2003, Google Labs published the white paper that describes the architecture of Google using Google distributed file system (GFS) [6]. They have also published MapReduce in the December 2004. The team members of Yahoo have launched new technique of Hadoop distributed file system (HDFS) and MapReduce in the year of 2007. Doug Cutting is the key member of that team and he had given the name Hadoop for his new invention.

The main objective of Hadoop system is to hide the details of parallel processing that includes data distribution to processing nodes, restarting failed subtasks and consolidation of results after the computation. This open source framework allows programmers to implement parallel processing programs that focus on their computational problem. There are two major components of Hadoop (1) Hadoop distributed file system (HDFS) [7] [8]; which is used for storing large amount of datasets with high degree of throughput. These huge datasets are stored on number of clusters. (2) MapReduce: It is a software framework for processing huge datasets with the clusters of commodity servers through parallelization [2] [4].

4. Hadoop Distributed File System

HDFS is inspired from GFS (Google file system). It is a high scalable, reliable and manageable file system. It provides a framework to store various big data in to the clusters of commodity servers. It has great features like high availability, scale-out architecture, load balancing, security, replication for data reliability, flexible access, fault tolerance, easy management, low latency and high data throughputs. It provides parallel reading and processing of data. It does not support file or record locking. Files are normally writes once and writing operations are limited to file appends [9]. In HDFS bandwidth scales linearly with the number of nodes and disks. This file system is designed to improve performance and make replication of data easier.

HDFS has master/slave architecture. As shown in the Fig 1, it consists of one NameNode and multiple DataNodes. There are three master services as NameNode, Secondary NameNode and JobTracker. Two slave services are TaskTracker and DataNodes. NameNode is responsible for mapping of data blocks to different DataNodes and manage various operations like open a new file, closing and renaming files and directories [7] [8]. To create new blocks, deletion and replication of data blocks are the key tasks of DataNodes. The contents of the DataNode will be written to the local disks attached with it. For high availability of data HDFS takes proper replication of every file. In HDFS block size is either 64MB or 128MB and the default replication factor is 3. NameNode manages the log files (metadata) referred as NameSpace, which stores all the details of directories and files with its replication copies. It keeps entire NameSpace in memory that makes NameNode much simpler in design. Secondary NameNode works as a backup of main NameNode. Every TaskTracker sends a heartbeat signal at the regular interval of 3 seconds to NameNode to indicate that the DataNode is active. JobTracker waits for 30 seconds; if no heartbeat signal comes within that time amount it considered that DataNode may be dead or working at very low speed.

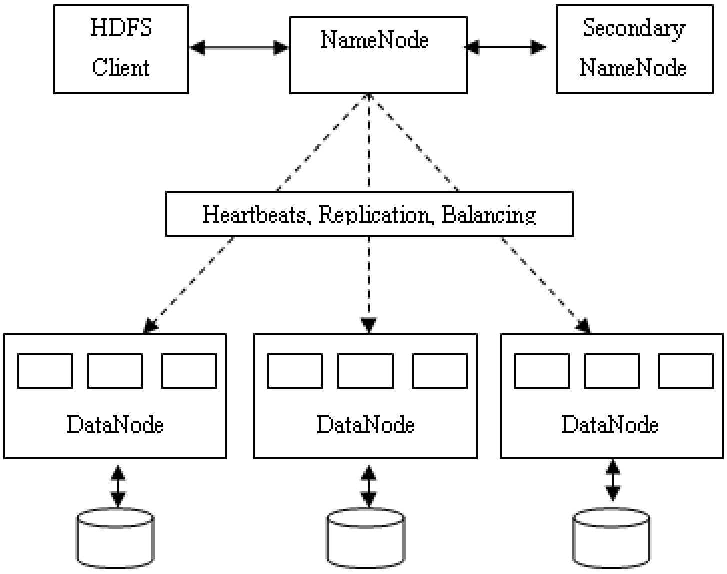

Figure 1: HDFS architecture

5. Hadoop MapReduce

MapReduce was originally proposed by Google to handle large scale web search applications. It is a programming model and associated implementation for processing and generating large sets. Hadoop MapReduce is a software framework for distributed processing of large datasets on compute clusters of commodity hardware [2] [5]. The framework takes care of scheduling tasks, monitoring and re-executing any failure tasks. The main benefit of MapReduce is that it allows programmers to abstract from the issues of scheduling [10], parallelization, portioning so programmers can easily focus on developing their applications. Hadoop MapReduce program is composed of Map and Reduce [11] [12] functions. Map function performs filtering and sorting of data and Reduce performs a summary operation. Both the functions are implemented by developer. Map function takes an input pair and produces a set of intermediate {key, value} pairs as shown fig 2. These pairs are shuffled across different reduce tasks based on {key, value} pairs. Each Reduce task accepts only one key at a time and process data for the key and outputs the results as {key, value} pairs.

The JobTracker receives a job submitted by user, breaks it into number of maps and reduce tasks, assigns the tasks to TaskTrackers, monitors the current progress and at the last when all the tasks are completed informed back to the user. In Hadoop system, all the jobs have to share the clusters of commodity servers for data processing, so proper scheduling policy and algorithm are required.

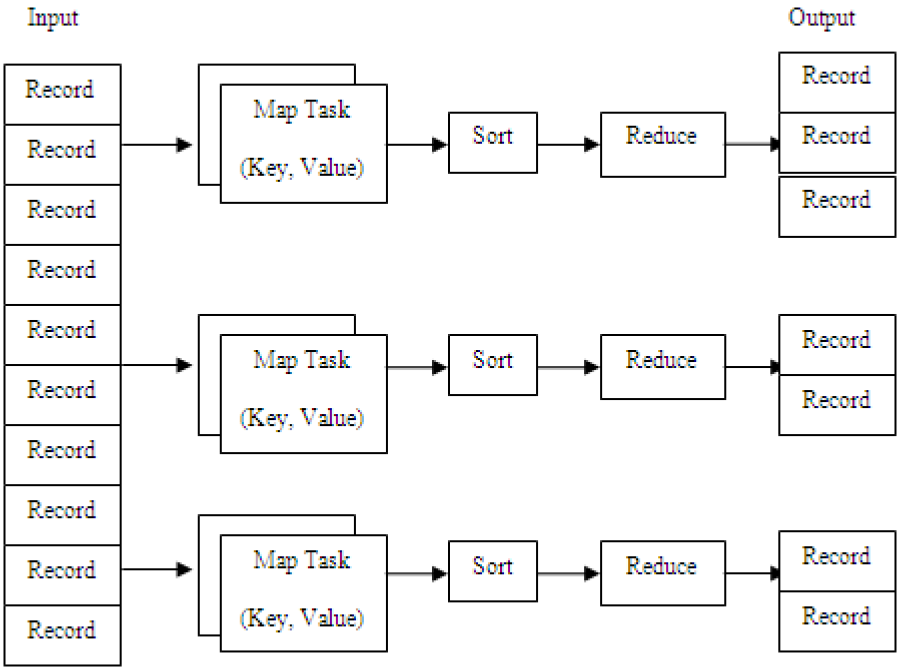

Figure 2: Hadoop MapReduce Structure

6. Scheduling Algorithms

The default scheduling algorithm is based on First in First out (FIFO) order where users' jobs are executed. Then after priority criteria to decide the priority of task was added. That would be helpful to perform critical task first compare to other task. Facebook and Yahoo have done significant work in developing various schedulers. Various schedules are discussed in this section.

A) FIFO Scheduler

It's based on queue mechanism; all the individual tasks are loaded into the job queue and submitted to free slots as they become available on TaskTracker nodes [8]. Problem of sharing resources in a fairly manner between users requires a better scheduling policy. All jobs need to complete in a timely manner and provide better response time to every job.

B) Fair Scheduler

This mechanism was developed at Facebook [7] to manage access the Hadoop cluster. The objective of this scheduler is to assign each user a fair share of the cluster capacity over a time [13] [16]. Users may assign jobs to pools, with each pool allocated a guaranteed minimum number of Map and Reduce slots [12] [19]. Its preemptive technique that means the scheduler will kill tasks in pools running over capacity in order to give the slots to the pool running under capacity. Priority criteria are also assigned to various pools. Here tasks are scheduled in interleaved manner based on priority.

C) Capacity Scheduler

The Capacity Scheduler [20] [18] was designed to allow organizations to share Hadoop clusters in a predictable and simple manner. It provides capacity guarantees for queues while providing elasticity for queues' cluster utilization in the sense that unused capacity of a queue can be harnessed by overloaded queues that have a lot of temporal demand [14]. Here, the number of users is large and the goal of this policy is to ensure a fair allocation of computation resources amongst users. The capacity scheduler allocates jobs based on the submitting user to queues with configurable numbers of Map and Reduce slots [11]. Queues that contain various jobs are given their configured capacity, while the remaining capacity in a queue is shared with other queues.

D) LATE-Speculative Executions

Sometime in Hadoop, task will be completed slowly, due to heavy load on the node, some failure may be there or slow background processes. The scheduler will find out the slow running task to launch another task as a backup task that is referred as speculative execution of tasks. To select speculative tasks, scheduler monitors task progress using a progress score between 0 and 1. If the background work completes faster, the job performance is improved. Researcher have proposed a new method for speculative execution called Longest Approximate Time to End (LATE) [1] algorithm that uses a different metric to schedule tasks for speculative execution. This method would be optimal if nodes ran at consistent speeds and if there was no cost to launching a speculative task [13].

E) Delay Scheduling

This method performs well in Hadoop workloads because Hadoop tasks are short relative to jobs, and because there are multiple locations where a task can run to access each data block [15]. When a node requests a particular task, if the head of line job can't assign local task, scheduler skip that task and looking for next jobs. Here, proper precaution steps are compulsory required to avoid starvation effect in delay scheduling mechanism [7] [8]. This method improves problem of locality by asking jobs to wait for scheduling opportunity on a node with local data. There are other methods like dynamic priority scheduling, deadline constraint scheduler and resource aware scheduling available for better scheduling in Hadoop.

7. Conclusions

To store Big Data with efficient manner and process those data with best computational power with minimum time Hadoop mechanism is used. In this paper we have studied Hadoop distributed file system, MapReduce and various scheduling algorithms to improve the speed of Hadoop system for data retrieval and job processing. Each of the scheduling method tries to utilize resources like Memory, CPU and job deadlines. Various researches are going on to improve the computation power of Hadoop system with large number of clusters.

References

[1] Matei Zaharia, Hruba Borthakur, Joydeep Sen Sarma, Khaled Elmeleegy, Scott Shenker, Ion Stoica, "Job Scheduling for Multi-User MapReduce Clusters", Electrical Engineering and Computer Sciences, University of California at Berkeley

[2] Apache Hadoop: http://Hadoop.apache.org

[3] Hadoop Distributed File System: http://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

[4] Hadoop Tutorial: https://developer.yahoo.com/hadoop/tutorial/module1.html

[5] Tom white, "Hadoop Definitive Guide", Third Edition,2012

[6] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung.The Google file system. In 19th Symposium on Operating Systems Principles, pages 29–43, Lake George, New York, 2003

[7] B. Thirumala Rao, Dr. L. S. Reddy, "Survey on Improved Scheduling in Hadoop MapReduce in Cloud Environments". IJCA, Volume 34-No.9, 2011

[8] Harshawardhan S. Bhosle, Devendra P. Gadekar : "Big Data Processing Using Hadoop: Survey on Scheduling" IJRS, Volume 3 Issue 10, 2014

[9] http://hortonworks.com/hadoop/hdfs/

[10] Michael Stonebraker, Daniel Abadi, David J. DeWitt, Sam Madden, Erik Paulson, Andrew Pavlo, Alexander Rasin "MapReduce and parallel DBMS:friends or foes?", ACM, Vol. 53 No. 1, 2010.

[11] Dean, J. and Ghemawat, S., "MapReduce: a flexible data processing tool", ACM 2010.

[12] DeWitt & Stonebraker, "MapReduce: A major step backwards", 2008.

[13] Hadoop Fair Scheduler. http://hadoop.apache.org/docs/r1.2.1/fair_scheduler.html

[14] V. Krishna Reddy, B. Thirumala Rao, LSS Reddy, "Research issues in Cloud Computing", Global Journal Computer Science & Technology Vol. 11, no. 11, June 2011,pp.70-76

[15] Matei Zaharia, Dhruba Borthakur,Joydeep Sen Sarma, Khaled Elmeleegy, Scott Shenker, and Ion Stoica. "Delay scheduling: a simple technique for achieving locality and fairness in cluster scheduling in EuroSys 10", Proceedings of the 5th European conference on Computer systems, pages 265–278, New York, NY, USA, 2010. ACM.

[16] Radheshyam Nanduri, Niteshaheshwari, Reddy Raja,Vasudeva Varma, "Job Aware Scheduling Algorithm for MapReduce Framework", 3rd IEEE International Conference on Cloud Computing Technology and Science Athens, Greece.

[17] Mark Yong, Nitin Garegrat, Shiwali Mohan: "Towards a Resource Aware Scheduler in Hadoop" in Proc. ICWS, 2009, pp:102-109

[18] Jagmohan Chauhan, Dwight Makaroff and Winfried Grassmann, "The Impact of Capacity Scheduler Configuration Settings on MapReduce Jobs"

[19] DongjinYoo, Kwang Mong Sim, "A comparative review of job scheduling for mapreduce", Multi-Agent and Cloud Computing Systems Laboratory, Proceedings of IEEE CCIS2011.

[20] Hadoop's capacity scheduler: http://hadoop.apache.org/core/docs/current/capacity/scheduler