A brief review of scheduling algorithms of Map Reduce model using Hadoop

Adhishtha Tyagi[#1], Sonia Sharma[*2]

[#]*Computer Science Department, JMIT Radaur*
[1]*Mtech Scholar,* [2] *Assistant Professor*

Abstract — *Scheduling has been an active area of research in computing systems since their inception. Hadoop framework has become very much popular and most widely used in distributed data processing. Hadoop has become a central platform to store big data through its Hadoop Distributed File System (HDFS) as well as to run analytics on this stored big data using its MapReduce component.*
The main objective is to study MapReduce framework, MapReduce model, scheduling in hadoop, various scheduling algorithms and various optimization techniques in job scheduling. Scheduling algorithms of MapReduce model using hadoop vary with design and behaviour, and are used for handling many issues like data locality, awareness with resource, energy and time.

Keywords — *Big Data, MapReduce and Hadoop framework, MapReduce model.*

I. Introduction

Big Data according to McKinsey refers to "datasets whose size are beyond the ability of typical database software tools to capture, store, manage and analyze". Big Data has become very popular in Information Technology, where data is becoming complex and growing day by day.
According to IBM Big Data consists of three attributes:
1. Variety: It means that the generated data are different in type. It generally refers to structured, semi-structured or unstructured data in large amount.
2. Volume: It concerns the large quantities of data that is generated continuously; they might reach hundreds or thousands of terabytes.
3. Velocity: Where processing and analyzing data must be done in a fast manner to extract value of data in appropriate time[1].
The advent of Big Data brings about many challenges to the existing data processing architecture. Hence, there arises a need for big data frameworks to complement the companies' existing infrastructure to support the analysis of the new data. There are many Big Data processing frameworks currently available in the market such as Dryad, Apache Hadoop, MapReduce , Apache Hadoop Yarn, and Apache Spark to name a few[2].

A. MAPREDUCE AND HADOOP

MapReduce is a framework for processing huge amounts of data in parallel by distributing the job into various independent tasks. Hadoop is Apache's open source implementation of the MapReduce framework[3].

Hadoop Distributed File System (HDFS) and Hadoop MapReduce are the key services of Hadoop platform. A Hadoop system can be described based on three factors: cluster, workload and user. Each of these factors is either homogeneous or heterogeneous which reflects the heterogeneity level of the Hadoop system[4].

B. MAPREDUCE MODEL

The MapReduce programming model consists of two functions: Map and Reduce. Input data is divided into fixed sized blocks and fed into parallel Map tasks. Then Map task process the data chunks and produces the intermediate output in the form of key-value pair tuples. These tuples are shuffled across different reduce nodes based on key values. Each Reduce task performs three steps:

- copy - the output of map task is copied to reducer nodes,
- sort - the collection of outputs is sorted based on their key values and
- reduce - aggregation is applied to the data.

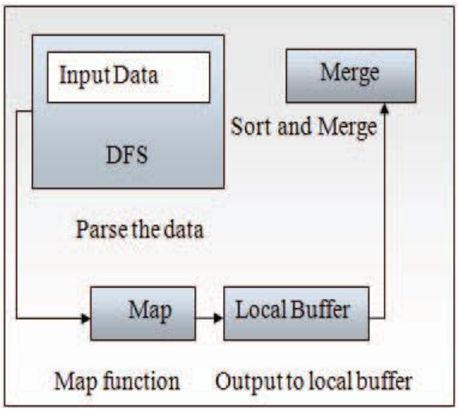

Fig. 1: Phase of execution in Map Task

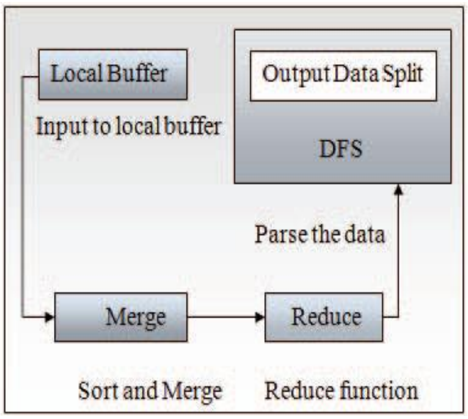

Fig. 2: Phase of execution in Reduce Task

The MapReduce architecture consists of:

- one master (Jobtracker) and
- many workers or slaves (Tasktrackers)

The JobTracker receives job submitted from user, divides it into map and reduce tasks. Then it assigns the tasks to TaskTrackers, monitors the progress and finally after the completion of the entire job, JobTracker informs the status of the job to the user. Each TaskTracker has a fixed number of map and reduce task slots which helps to determine how many map and reduce tasks it can run at a time[5].

C. Scheduling in hadoop

Hadoop is a multi-user data warehouse which is used to support a variety of different types of processing jobs, with a pluggable scheduler framework providing greater control. The main objective of scheduling is to maximize throughput, minimize the completion time; overhead and available resources must be balanced by allocating jobs to processors[6].

- Hadoop uses a default scheduler i.e. FIFO. The jobs submitted earlier gets preference over the jobs submitted later. The JobTracker pulls oldest jobs first from the job queue. This scheduler is mostly used when execution order of job is not important.
- Capacity Scheduler is developed by Yahoo that allows multiple tenants to securely share a large cluster. It supports hierarchical queues to ensure that resources are shared among the sub-queues of an organisation before other queues are allowed to use those resources.
- Fair Scheduler developed by Facebook, the core idea behind fair scheduler is to assign resources to each job such that an average each job gets equal share of available resources.
- Delay scheduling has an objective to address the conflict between locality and fairness. Delay scheduling temporarily relaxes fairness to improve locality by allowing jobs to wait for scheduling on a node with local data. In this, if data for the task is not present, then a task tracker waits for some time. If there is any request from local node for task assigning, then the scheduler will check the size of the job; if the job is too short, then he scheduler will skip that job and look for any subsequent jobs available to run.
- Dynamic Proportional Scheduling provides more job sharing and prioritization that results in increasing shares of cluster resources and more differentiation in service levels of different jobs. This improves response time for multi-user Hadoop environments.

II. LITERATURE SURVEY

Xiangming Dai and Brahim Bensaou in 2016, proposed a novel task scheduling algorithm for Hadoop MapReduce called dynamic priority multiqueue scheduler (DPMQS).
DPMQS i) increase the data locality of jobs, and, ii) dynamically increase the priority of jobs that are close to complete their Map phase, to bridge the time gap between the start of the reduce tasks and the execution of the reduce function for these jobs. Then discussion regarding the details of DPMQS and the practical implementation, thereafter assess its performance in a small physical cluster and large-scale simulated clusters so as to compare it to the other schedulers in Hadoop. Both real experiments and simulation results show that DPMQS decreases the response time, and demonstrate that DPMQS is not flexible with the changes in the cluster geometry[7].

Divya M. and Annappa B. in 2015, focused on optimizing job scheduling in Hadoop. Workload Characteristic and Resource Aware (WCRA) Hadoop scheduler classifies the jobs into CPU bound and Disk I/O bound. As concerned to the performance, nodes in the cluster are classified as CPU busy and Disk I/O busy. The amount of primary memory in the node should be more than 25% before scheduling the job. The performance parameters of Map tasks i.e. the time required for parsing the data, map, sort and merge the result, and of Reduce tasksi.e. the time to merge, parse and reduce is used to categorize the job as CPU bound or Disk I/O bound. Tasks are assigned the priorities based upon their minimum Estimated Completion Time. The jobs are scheduled on a compute node in a way so that the jobs already running will not be affected. Experimental results have given 30% improvement in performance as compared to Hadoop's FIFO, Fair and Capacity scheduler[4].

Qinghua Lu and Shanshan Li in 2015, proposed a genetic algorithm based job scheduling model to improve the efficiency of BDA. To implement the scheduling model, leverage the estimation module to predict the performance of clusters when processing jobs and also evaluated the proposed job scheduling model in terms of feasibility and performance[8].

YongLiang Xu and Wentong Cai in 2015, The Dynamic Task Splitting Scheduler (DTSS) is proposed to mitigate the tradeoffs between fairness and data locality during job scheduling. DTSS dynamically splits a task and executes the splitted task immediately, on a non-data-local node, to improve the fairness. Analysis shows that it is possible to improve both the fairness and the performance by adjusting the proportion of the splitted task. DTSS is shown to improve the makespan of different users in a cluster by 2% to 11% as compared to delay scheduling in the cases where it is difficult to obtain data-local nodes on a cluster. The experiments show that DTSS is not a suitable scheduler under conditions where jobs are able to obtain data-local nodes easily[2].

Ke Wang and Iman Sadooghi in 2015, aimed to address the YARN scaling issues through a distributed task execution framework, MATRIX, originally designed to schedule the executions of data-intensive scientific applications of many-tasks on supercomputers. They also run and simulate MATRIX with fine-grained sub-second workloads, giving the efficiency of 86.8% at 64 thousand cores for the 150ms workload[9].

Qutaibah Althebyan and Omar ALQudah in 2014, proposed a new scheduling algorithm that was based on a multi-threading principle. The proposed algorithm, divided the cluster into multi blocks where each one of them is scheduled by a special thread. Two major factors used to test the algorithm are: the simulation time and the energy consumption. The scheduler is then compared with existing schedulers and the results showed the superiority and the preference of proposed scheduler over the existing schedulers[1].

Chen He and David Swanson in 2013, created a novel real-time scheduler for MapReduce, which is used to overcome the deficiencies of existing scheduler. It avoids accepting jobs that will lead to deadline misses and improves the cluster utilization. They implement scheduler in Hadoop system and experimental results show that scheduler provides deadline guarantees for accepted jobs and achieves good cluster utilization[3].

Li Liu and Yuan Zhou in 2012, formulated the preemptive scheduling problem under deadline constraint, and then proposed its algorithms. The experimental results showed that the preemptive scheduling approach is more efficient than the non-preemptive one for executing jobs under a certain deadline[6].

Hong Mao and Shengqiu Hu in 2011, focused on optimizing the scheduler of MapReduce framework in task level. They cared about the hardware configuration and real-time workload of the nodes in a hadoop cluster and aied at shortening time cost of MapReduce jobs and improving hardware resource utilization rate. They put forward a load driven task scheduler that is used to assign tasks to TaskTrackers according to the workload of slave nodes. It is based on a Dynamic Slot Controller (DSC) which is used to adjust Map task Slot (MS) and Reduce task Slot (RS) of TaskTrackers running on slave nodes adaptively. Load-driven task scheduler reduces time consumption of MapReduce job by 14% and improves the rate of CPU utilization of hadoop cluster by 34% when processing 10GB data[10].

Kamal Kc and Kemafor Anyanwu in 2010, extended real time cluster scheduling approach in an account for the two-phase computation style of MapReduce and developed a criteria for scheduling jobs based on user specified deadline constraints and discussed implementation and pre-evaluation of a Deadline Constraint Scheduler for Hadoop which ensures that only jobs whose deadlines can be met are scheduled for execution[5].

III.TABLE

Overview Of Various Techniques And Findings In Job Scheduling:

AUTHOR	**YEAR**	**TECHNIQUE /METHOD**	**FINDINGS**
Kamal Kc, Kemafor Anyanwu	2010	Scheduling hadoop jobs to meet deadlines	Extended real time cluster scheduling approach to derive minimum map and reduce task count criteria for performing task scheduled with deadline constraints.
Hong Mao, Shengqiu Hu, Zhenzhong Zhang, Limin Xiao, Li Ruan	2011	A load driven task scheduler with adaptive Dynamic Slot Controller for	Optimize the scheduler of MapReduce framework in task level and care about the hardware configuration

		MapReduce	and dynamic workload of the nodes in a hadoop cluster.
Li Liu, Yuan Zhou, Ming Liu, Guandong Xu, Xiwei Chen, Dangping Fan, Qianru Wang	2012	Developing deadline-based MapReduce schedulers by using the non-preemptive scheduling approach	Employed the preemption to the deadline constraint scheduling model in order to overcome the issues, such as the starvation caused by the non-preemptive scheduling.
Chen He, Ying Lu, David Swanson	2013	Real-time scheduler for MapReduce, to overcome the deficiencies of an existing scheduler	RTMR(Real-Time MapReduce) scheduler overcomes the deficiencies of an existing algorithm and achieves good cluster utilization and 100% job success ratio, ensuring the realtime property for all admitted MapReduce jobs
Qutaibah Althebyan , Omar ALQudah, Yaser Jararweh, Qussai Yaseen	2014	A new scheduling algorithm based on a multi-threading principle	The algorithm is based on multi – threading principles, where the infrastructure is divided into blocks, every block is scheduled by a special thread, and this scheduling is achieved in a synchronous time.
YongLiang Xu, Wentong Cai	2015	Dynamic Task Splitting Scheduler (DTSS) to mitigate the tradeoffs between fairness and data locality during job scheduling	DTSS dynamically splits the task and schedule one part of original task for processing at a non local data node immediately. It is possible to improve the performance and the fairness of the users through dynamically splitting of tasks.
Divya M., Annappa B.	2015	Workload Characteristic and Resource Aware (WCRA) Hadoop scheduler that classifies the jobs into CPU bound and Disk I/O bound	Workload Characteristics and Resource Aware Hadoop job scheduler is designed that considers the job characteristics and node status before making the scheduling decisions. The experiments have shown that consideration of the performance of maptasks and reducetasks in the classification of workload improves the Hadoop performance.
Qinghua Lu, Shanshan Li, Weishan Zhang	2015	A genetic algorithm based job scheduling model to improve the efficiency of Big Data Analytics(BDA)	Genetic algorithm based approach, uses a performance estimation module, for obtaining optimized jobs scheduling

			scheme that have the optimized time and cost consumption. The optimized solutions can be used to enable effective scheduling strategies, and then in the actual running, system can make use of the chosen scheduling scheme to execute data processing jobs.
Ke Wang, Ning Liu, Iman Sadooghi, Xi Yang, Xiaobing Zhou, Tonglin Li, Michael Lang, Xian-He Sun, Ioan Raicu	2015	Distributed Task Execution to overcome hadoop scaling limitations	Proposed to leverage the distributed design wisdoms of the MATRIX task execution framework to overcome the scaling limitations of Hadoop towards extreme scales. MATRIX addressed the scaling issues of YARN by employing distributed resource management, distributed data-aware task scheduling, and distributed metadata management using key-value stores.
Xiangming Dai, Brahim Bensaou	2016	A novel task scheduling algorithm for Hadoop MapReduce called dynamic priority multiqueue scheduler (DPMQS)	DPMQS firstly increases the data locality rate of map tasks by launching them as near as possible to their input data and secondly increases the priority of jobs that are near to completing their map phase to bridge the time that reduce tasks spend waiting for the completion of the map phase.

IV. Conclusions

Various research papers have been reviewed for the survey mentioned in the literature. Each scheduler considers the resources like CPU, Memory, Job Deadlines etc. To overcome the issue of big data storage and processing the open source framework Hadoop can be used. Starvation caused by non pre-emption scheduling is reduced by pre-emption to the deadline constraint scheduling model. Real Time MapReduce scheduler achieves good cluster utilization. Dynamic Task Splitting Scheduler improves the performance and fairness of users through dynamically splitting the tasks. Future work will consider enhancement in scheduling using particular scheduler.

References

[1] Qutaibah Althebyan , Omar ALQudah, Yaser Jararweh, Qussai Yaseen, "*Multi-Threading Based Map Reduce Tasks Scheduling*", 5th International Conference on Information and Communication Systems (ICICS), 2014.

[2] YongLiang Xu, Wentong Cai, "*Hadoop Job Scheduling with Dynamic Task Splitting*", International Conference on Cloud Computing Research and Innovation, 2015.

[3] Chen He, Ying Lu, David Swanson, "*Real-Time Scheduling in Map Reduce Clusters*", IEEE International Conference on High Performance Computing and Communications & IEEE International Conference on Embedded and Ubiquitous Computing, 2013.

[4] Divya M., Annappa B., "*Workload Characteristics and Resource Aware Hadoop Scheduler*", IEEE 2nd

International Conference on Recent Trends in Information Systems (ReTIS), 2015.

[5] Kamal Kc, Kemafor Anyanwu, "*Scheduling Hadoop Jobs to Meet Deadlines*", 2nd IEEE International Conference on Cloud Computing Technology and Science, 2010.

[6] Li Liu, Yuan Zhou, Ming Liu, Guandong Xu, Xiwei Chen, Dangping Fan, Qianru Wang, "*Preemptive Hadoop Jobs Scheduling under a Deadline*", Eighth International Conference on Semantics, Knowledge and Grids, 2012.

[7] Xiangming Dai, Brahim Bensaou, "*Scheduling for response time in Hadoop MapReduce*", IEEE ICC 2016 SAC Cloud Communications and Networking, 2016.

[8] Qinghua Lu, Shanshan Li, Weishan Zhang, "*Genetic Algorithms based Job Scheduling for Big Data Analytics*", International Conference on Identification, Information, and Knowledge in the Internet of Things, 2015.

[9] Ke Wang, Ning Liu, Iman Sadooghi, Xi Yang, Xiaobing Zhou, Tonglin Li, Michael Lang, Xian-He Sun, Ioan Raicu, "*Overcoming Hadoop Scaling Limitations through Distributed Task Execution*", IEEE International Conference on Cluster Computing, 2015

[10]Hong Mao, Shengqiu Hu, Zhenzhong Zhang, Limin Xiao, Li Ruan, "*A Load-Driven Task Scheduler with Adaptive DSC for MapReduce*" , IEEE/ACM International Conference on Green Computing and Communications, 2011.