ISSN: 2456-9348
Vol (02) _Issue (04) Impact Factor: 4.520

iJETRM
International Journal of Engineering Technology Research & Management

SURVEY ON SCHEDULING IN HADOOP SYSTEM

V.Thanga Sharmila[1],
Dr. S. Raja Ratna [2]
[1,2]Department of Computer Science and Engineering, VV College of Engineering, India
[1]sharmi1695@gmail.com
[2]gracelinrr@yahoo.com

ABSTRACT
Hadoop is an open source software framework used for distributed storage and processing of dataset of big data using MapReduce programming model. Scheduling is the process by which job specified by some means is assigned to nodes that complete the work. A scheduling policy is used in Hadoop when the job requests the number of map and reduce task slots requested exceeds the limit. As each worker node has a fixed number of the map and reduce task slots that determine how many map and reduce tasks it can run at the same time. The objective of this paper is to provide a general overview of scheduling in Hadoop system. It covers relevant works, different scheduling techniques, and various types of schedulers. Challenges associated with comparing several scheduling techniques are also highlighted.

Keywords:
Hadoop, MapReduce, Scheduling, Data locality, JobTracker, TaskTracker, HDFS

INTRODUCTION

Hadoop is a popular open-source implementation which supports for distributed shuffles. It splits files into large blocks and distributes them across nodes in a cluster. It then transfers packaged code into nodes to process the data in parallel [19]. This approach takes advantage of data locality [23], [31], [30], nodes manipulating the data they have access to allow the dataset to be processed faster and more efficiently than it would be in a more conventional supercomputer architecture that relies on a parallel file system where computation and data are distributed via high-speed networking. Scheduling is a method that gives requesters access to time-based resources and a major paradigm for computer technology, transportation and manufacturing. In computer technology, the goal of scheduling is to minimize the tasks' durations [11].

A Hadoop job consists of an input data set, a list of Java classes and settings files. Input file is divided into smaller chunks called splits which are distributed among the nodes that carry out processing. Splits have a fixed-size of 64 MB by default and are maintained by Hadoop Distributed File System (HDFS). When a job is submitted to the cluster, Hadoop divides this job into Map and Reduce tasks. The scheduling of each Map task occurs dynamically and placing input data at the same node where will be processed decreases the volume of data transferred between nodes in a cluster. So, Hadoop tries to assign each Map task to the node where the split is stored. First, Map output is kept in memory. When allocated memory fills, the data is spilled to hard disk and stored on local file system without being handled by HDFS. After Map task finishes, Hadoop starts the Shuffle phase. In this phase, intermediate data is merged, sorted and copied over the network to the nodes where they will be processed by the Reduce tasks [2].

Based on the decision of the scheduler, a task belonging to a job will be selected for execution whenever a node is available. Hadoop considers the location of the input data for a task and tries its best to run on the node in which the data resides [52].

The paper proceeds as follows. Section II describes scheduling analysis and different types of jammers. Section III explains the comparison of various job scheduling techniques. Section IV explains the comparison of various task scheduling techniques. Section V explains the comparison of various hadoop default scheduling techniques. Section VI explains the comparison of various mapreduce scheduling techniques. Finally, Section VIII concludes the paper.

iJETRM
International Journal of Engineering Technology Research & Management

SCHEDULING ANALYSIS

Scheduling is the process by which job specified by some means is assigned to nodes that complete the work. A scheduler carries out the scheduling activity. The goal of scheduling is to minimize the tasks execution time, maximizing throughput, minimizing response_time or minimizing latency. In order to manage load of the cluster, a scheduling approach is needed to improve the overall cluster performance [42]. A scheduling policy is used in Hadoop when the job requests the number of map and reduce task slots requested exceeds the limit. As each worker node has a fixed number of the map and reduce task slots that determine how many map and reduce tasks it can run at the same time. Scheduling is classified as Job Scheduling, Task Scheduling, Hadoop Default Scheduling and MapReduce Scheduling.

Job Scheduling

Users submit jobs to a queue, and the cluster runs them in order. However, as organizations place more data in their Hadoop clusters and develop more computations they want to run, another use case becomes attractive to share a MapReduce cluster between multiple users. However, sharing of resources among jobs requires support from the Hadoop job scheduler to provide guaranteed capacity to production jobs and good response time to interactive jobs while allocating resources fairly between users. Job Scheduling in Hadoop falls into three categories: Job based scheduling, Aware job scheduling and Adaptive job scheduling

Task scheduling

Task scheduling in Hadoop allocates appropriate tasks of the jobs to appropriate server. The Hadoop scheduling model is a Master/Slave cluster model. The JobTracker coordinates all worker nodes. JobTracker is responsible for management of all task servers while TaskTracker executes tasks on the corresponding nodes. The scheduler is present in the JobTracker. Task of scheduler is to allocate resources to TaskTrackers for executing the tasks. Task scheduling in Hadoop falls into three categories: Task based scheduling, Aware task scheduling and Adaptive and Improved task scheduling.

Hadoop default scheduling

Default scheduling in hadoop is JobQueueTaskScheduler, which is a FIFO scheduler. If you want, you can change the default scheduler to either Capacity Scheduler or Fair Scheduler based on your requirement. Three scheduling are available in Hadoop: the FIFO scheduling, Capacity scheduling, and Fair scheduling.

MapReduce scheduling

Mapreduce is the data processing framework that automatically handles failures. It deals with the implementation for processing and generating large datasets with a parallel distributed algorithm on a cluster. Input data is splitted and fed to each node in the map phase. The results generated in this phase are shuffled and sorted then fed to the nodes in the reduce phase. MapReduce scheduling can be classified as: MapReduce based scheduling, Enhanced MapReduce scheduling and Aware MapReduce scheduling.

JOB SCHEDULING IN HADOOP

Job Scheduling in Hadoop is classified as Job based scheduling [47], [42], [2], [52], [21], [23], [3], [46], Aware job scheduling [9], [17], [36], [51], [8], [43], [12], [10], [4], [13], [20], [33] and Adaptive job scheduling [19], [11].

Job based scheduling

Hadoop job based scheduling provides guaranteed capacity to production jobs and good response time to interactive jobs while allocating resources fairly between users. Different job based scheduling algorithms are listed in Table 1.

iJETRM
International Journal of Engineering Technology Research & Management

Table 1. Job based Scheduling

Technique	Description	Advantages	Disadvantages
Job Scheduling Based on Hybrid Ant-Genetic Algorithm [47]	Schedules the tasks according to system status. Accelerates global convergence to speed up the local search and quickly find the suitable node.	Improves system efficiency, convergence speed and accuracy.	Not suitable to cloud computing environments.
Job Aware Scheduling for Heterogeneous Cluster [42]	Schedules non-local map tasks based on three criteria: job execution time, workload of the job and earliest deadline first.	Increases the resource utilization and reduces the average waiting time.	Lower the number of non-local map tasks, the average waiting time is high.
Job Scheduling with Dynamic Task Splitting [52]	Splits a task dynamically and execute the split task immediately on remote node to improve the fairness.	Improves fairness, performance and makespan.	Not a suitable scheduler to assign job to data-local nodes.
Preemptive Hadoop Jobs Scheduling under a Deadline [21]	Schedules the job based on number of available slots, input data sizes, current system status and deadline.	Overcomes the starvation caused by non-preemptive scheduling.	Low performance in completion time and slot utilization.
Job Scheduling for Multi-User MapReduce Clusters [23]	Allocates task slots among pools and each pool allocates its slots among multiple jobs in the pool.	Provides data locality and interdependence between map and reduce tasks.	Lowers the throughput.
Job Scheduling for Optimizing Data Locality [3]	Allow the tasks from different jobs that shares the same data blocks are allowed to run on the same node sequentially.	Makes adequate resource reservations.	Does not assign the task dynamically.

Aware job scheduling

The main purpose of Aware job scheduling is to increase the resource utilization of both I/O bound and CPU bound jobs [43] and allows the scheduler to assign the jobs according to the current status of the TaskTracker's, context i.e. the job characteristics (CPU or I/O bound) and the resource characteristics like Computational or I/O strength of the nodes in the cluster [4]. Aware job scheduling is further classified into resource aware job scheduling [9], [17], [8], [43], [12], [10] and context aware job scheduling [4], [13].

Resource aware job scheduling

The main idea of resource aware job scheduling is balancing different kinds of workload in TaskTracker to increase the resource utilization of both I/O bound and CPU bound jobs [43]. Status of compute node and characteristics of the jobs submitted by the client is found and the tasks are scheduled based on its resource utilization. It considers different parameters of map task and reduce task to accurately calculate the amount of time each steps of map task and reduce task uses different resources to find the task execution time [12]. Different resource aware job scheduling algorithms are listed in Table 2.

iJETRM
International Journal of Engineering Technology Research & Management

Table 2. Resource aware job scheduling

Technique	Description	Advantages	Disadvantages
Resource and Deadline-aware Job Scheduling [9]	Allocates resources to individual jobs based on job completion time and future resource availability. Temporarily delays low priority jobs or jobs with distant deadlines.	Takes future resource availability into account, minimize job deadline misses.	Lower response time.
Coupling Task Progress for MapReduce Resource-Aware Scheduling [17]	Coordinates the progresses of Map tasks and Reduce tasks, to optimize the task allocation jointly.	Relieves the starvation problem and improves the overall data locality.	Less improvement in job response times.
Fine-Grained Resource-Aware Scheduling [36]	Scheduling is performed at phase level in which it divides tasks into phases, where each phase has a constant resource usage profile.	Improves execution parallelism and resource utilization.	Problem in meeting job deadlines.
Scheduling Based on Task-Dependency and Resource-Demand [51]	Schedules the task for execution when the needed resources are available based on resource demand, resource capacity and urgency.	Reduces the makespan and improves resource utilization.	Does not support for other cloud computing paradigms.
Load Feedback-Based Resource Scheduling [8]	Assigns appropriate resources for execution to different task. Once the load in host exceeds, resources are cancelled otherwise resources will be allocated.	Balances the workload efficiently and improve system performance.	Data locality optimization is less
Resource Aware Scheduling for Heterogeneous Workloads based on Load Estimation [43]	Taking job into account, it organizes the jobs into several groups and balances the different kind of workloads in TaskTracker.	Increases the resource utilization of both I/O bound and CPU bound jobs.	Increase overhead on JobTracker.

Context aware job scheduling

Context aware is the capacity of an application or software to detect and respond to environment changes. A context-aware system is able to adapt its operations to current state without human intervention, therefore improving the system's usability and efficiency. In pervasive grids, the scheduling is a task that may be benefited in context-aware systems, collecting data about the grid resources and making decisions based on the data collected [13]. There is need for a Context Aware Scheduler for Hadoop which knows the context i.e. the job characteristics (CPU or I/O bound) and the resource characteristics like Computational or I/O strength of the nodes in the cluster [4]. Different context aware job scheduling algorithms are listed in Table 3.

iJETRM
International Journal of Engineering Technology Research & Management

Table 3. Context aware job scheduling

Technique	Description	Advantages	Disadvantages
Context Aware Scheduling [4]	Uses node classifier to classify the nodes according to computation or disk capability. Scheduler searches for the jobs with requirement match and data locality.	Improves performance and execution time.	Increase network traffic before assigning a task to a node.
Context-Aware Scheduling over Pervasive Environments [13]	Collects context information i.e., available resources on nodes in order to detect the dynamic changes in the resources occurring in pervasive environments.	Dynamically adapt its scheduling to the execution environment and provides performance gains.	Does not measure the impact of context on high dynamic environments.

TASK SCHEDULING IN HADOOP

Task Scheduling in Hadoop is classified as Task based scheduling [27], [7], [37], [40], [44], [38], Aware task scheduling [22], [32], [29], [39], [24], [6], [30], [25] and Adaptive and Improved task scheduling [48], [31], [55].

Task based scheduling

Task scheduling tries to maintain co-operation among the jobs running on cluster. If the incoming task does not disturb the tasks already running on that node, it will allocate a task on a node. From the list of available pending tasks, this algorithm selects the one which is most compatible with the tasks already running on that node. Different Task based scheduling algorithms are listed in Table 4.

Table 4. Task based scheduling

Technique	Description	Advantages	Disadvantages
Round robin algorithm [27]	It scans each and every server in a round robin fashion and assigns the task to the server to exploit data locality.	Assignments depend only on the number of servers and the remote cost function.	Impossible to find the optimal assignment.
Load balancing task scheduling algorithm [7]	In order to balance the load among the cluster, it uses load balancing method of histogram sampling, thus the bottleneck problem of network gets reduced.	Performance of the design is better in higher load.	Too much external command calls reduces the performance and reliability.
A virtual machine based task scheduling [37]	It helps to improve the data locality in which the physical server running VM acts as a storage node that holds the data replica needed by computing node.	Improves performance of MapReduce and improved data locality.	Does not improve the performance for machine learning MapReduce applications.

Aware task scheduling

Aware task scheduling tries its best to allocate enough resources (containers) to meet the (soft) deadline specified in the application's SLA [32], to minimize the total power consumption in the air conditioning (A/C) system that provides the cooling for maintaining the temperature [6], based on predicted failures information [25], basis of data locality that will minimize data-local traffic [29]. The aware task scheduling is further classified into locality aware task scheduling [29], [39], [24] and failure aware task scheduling [30], [25].

iJETRM
International Journal of Engineering Technology Research & Management

Locality aware task scheduling

Locality Aware Task Scheduling assigns global information as the number of block replicas, their distribution and the divergence of nodes performance in the cluster [22] on the basis of data locality that will minimize data-local traffic [21] and fair sharing [26]. Different locality aware task scheduling algorithms are listed in Table 5.

Table 5. Locality aware task scheduling

Technique	Description	Advantages	Disadvantages
Locality Premised Reducer Scheduling in Hadoop [29]	JobTracker schedules the task to run on TaskTracker by accepting the heart beat messages. TaskTracker runs map and reduce tasks and sent the result to the JobTracker	Minimizes data-local traffic, shuffling, network congestion.	Does not enable to work in shared (heterogeneous) environment.
Scalable Lightweight Locality-aware Scheduling [39]	According to the number of unhandled local blocks, the nodes are sorted. The candidate node is selected based on node importance and assigns the node to it.	Completely eliminates off-switch scheduling and reduce the total job execution time	Needs optimization for the shuffle and reduce phases.
Delay Scheduling [24]	It assigns free slots to the job that has less number of tasks and also it searches for local task in the job to achieve data locality.	Achieves nearly optimal data locality and increase throughput	Number of node increases, efficiency gets reduced.

Failure aware task scheduling

Failure Aware Task Scheduling predicts the potential outcome of new tasks and adjusts its scheduling decisions accordingly to prevent them from failing [25] that enables an early yet smart action for fast failure recovery while still operating within a specific scheduler objective [30]. Upon failure detection, rather than waiting for an uncertain amount of time to get resources for recovery tasks, it leverages a lightweight preemption technique to carefully allocate these resources. Different locality aware task scheduling algorithms are listed in Table 6.

Table 6. Failure aware job scheduling

Technique	Description	Advantages	Disadvantages
Failure- Aware Scheduling [30]	It monitors the running tasks by heartbeat messages. When the failure is detected it list the failed task and preempt them. Upon preemption it releases all the allocated slots.	Considers data locality and improves the performance.	Waits or kills primitive to ensure the QoS requirements.
Adaptive Failure-Aware Scheduling [25]	It records previously executed tasks and jobs. Then the task and job attributes are extracted from the log file to determine the failure.	Reduces CPU and memory usages and reduces the failure rates.	When the prediction model is retrained at fixed time intervals, performance gets degraded.

Adaptive and improved task scheduling

In Adaptive and Improved Task Scheduling task trackers can adapt to the change of load at runtime, obtain tasks in accordance with the computing ability of their own and realize the self-regulation while avoiding the complexity of algorithm [48] based on cache locality and data locality [31]. Self Adaptive scheduling can decide the start time points of each reduce tasks dynamically according to each job context, includes the task completion time and the size of map output [55]. Different adaptive and improved task scheduling algorithms are listed in Table 7.

iJETRM
International Journal of Engineering Technology Research & Management

Table 7. Adaptive and improved task scheduling

Technique	Description	Advantages	Disadvantages
Adaptive Task Scheduling Strategy based on Dynamic Workload Adjustment [48]	JobTracker assigns the task to the TaskTracker when it receives heart beat messages from the TaskTracker. TaskTracker adapts to the change in load dynamically.	Efficient and reliable algorithm, which can make cluster stable, scalable, efficient, and load balancing.	Less execution efficiency. Load in the cluster increases.
Improved task scheduling algorithm based on cache locality and data locality [31]	Map tasks are sorted by the priority and according to the location of the data the selection matrix is found. It schedules the tasks based on the weighted bipartite graph maximum matching.	Optimizes the local cache and data locality. Reduces the data transmission amount.	Only seeks to reduce data transmission and so load in the cluster increases.
Self-Adaptive Scheduling Algorithm for Reduce Start Time [55]	Reduce task reads the map output data in a copy phase. Copy operations completed at a concentrated duration, that can decrease the waiting time of the reduce tasks.	Reduces completion time and system average response time.	Does not enable to work in shared (heterogeneous) environment.

HADOOP DEFAULT SCHEDULING

Default scheduling in Hadoop is classified as FIFO Scheduling, Fair Scheduling [40], [41], [44], Capacity and Hybrid Scheduling [42], [43], [45].

FIFO scheduling

The original scheduling algorithm that is integrated within the JobTracker is called FIFO. In FIFO scheduling, a JobTracker forces out the jobs from a work queue, oldest job first. This schedule has no concept of the priority or size of the job, but the approach is simple to implement and efficient.

Fair scheduling

The Fair Scheduler emerges out of Facebook's need to share its data warehouse between numerous users. The activity of the scheduler is then to guarantee that the task with the highest priority is scheduled next. It effectively tradeoffs the performance and the fairness, and reduces the makespan of MapReduce jobs by utilizing multi-level queue, time factor, job urgency factor, and domain resource ratio [35]. The scheduling algorithm is adaptive, because it can dynamically tune some working parameters such as the job priority and the waiting time for resource allocation [50]. Different fair scheduling algorithms are listed in Table 8.

iJETRM
International Journal of Engineering Technology Research & Management

Table 8. Fair scheduling

Technique	Description	Advantages	Disadvantages
Performance-Fairness Scheduling [35]	It uses multi-level queue for resource allocation. It allocates resources to the job with lower resource ratio and the job is delayed if it has higher resource ratio.	Reduces the makespan and improve the CPU and memory utilization.	Does not dynamically compute proper load factor.
Improving Fair Scheduling [50]	It separates small and large sized jobs and determines the job with high priority. It then dynamically adjusts the delay time based on locality.	Provides good performance and reduce the average turnaround time of a job.	Needs more enhancements for reduce tasks.
Fair Scheduling for Distributed Computing Clusters [26]	It is based on graph data structure that encodes cluster structure and sets of waiting tasks. The cost and capacities allows to map min-cost flow to fair Scheduling assignment.	Gets better fairness and improves data locality.	No explicit attempt is made to share the network or other resources.

Capacity and hybrid scheduling

The Capacity Scheduler from Yahoo offers comparable usefulness to the Fair Scheduler but takes a somewhat different technique. In the Capacity Scheduler, we use a number of named queues. Each queue has a configurable number of map and reduce slots. The scheduler gives each queue its capacity when it contains jobs, and shares any unused capacity between the queues. Improved capacity scheduler improves the existing scheduler issues that help the scheduler to execute the task in less time [41]. Hybrid scheduler is combination of three Hadoop schedulers: FIFO, Fair Sharing, and COSHH (Classification and Optimization based Scheduler for Heterogeneous Hadoop) [5]. Different capacity and hybrid scheduling algorithms are listed in Table 9.

Table 9. Capacity and hybrid scheduling

Technique	Description	Advantages	Disadvantages
Virtualization Using Capacity Scheduling [1]	A virtualized Hadoop cluster is set up with the master node on a physical machine and slave nodes on VMs.It schedules tasks based on the RAM availability and virtual memory in slave nodes before allocating any job.	Provides greater computing capacity with lesser resources and requires fewer physical machines.	Not suitable for heterogeneous clusters
Hybrid Scheduling [5]	It is the combination of FIFO, Fair Sharing, and COSHH. It uses a set of jobs and considers the job priority and fair share of users to make better scheduling decision.	Suitable for scalable and heterogeneous Hadoop systems.	Not suitable for homogeneous environments.
Dynamic Capacity Scheduling [41]	It puts the job in varied queues and allocates system capability to each	Improves the performance and reduces the execution	Does not support in a heterogeneous environment

iJETRM
International Journal of Engineering Technology Research & Management

queue. When the queue is loaded, it assigns resources equally to each job.	time.

MAPREDUCE SCHEDULING IN HADOOP

MapReduce Scheduling in Hadoop can be classified into MapReduce based Scheduling [18], [53], [45], [54], [16], Aware MapReduce Scheduling [14], [34], [15] and Enhanced MapReduce Scheduling [28], [49].

MapReduce based scheduling

MapReduce Scheduling maintains statistics specific to the opportunistic environment, e.g., node availability rates and pair wise availability correlations, and utilizes this information in scheduling decisions to improve fairness [53]. The multi-level privacy scheduler with data clearance levels and cloud authorization levels provides the mapping of map and reduce tasks to cloud resources can satisfy the organization's privacy constraints [45] and allows user specify a job's deadline and tries to make the job be finished before the deadline[54]. Different MapReduce based scheduling algorithms are listed in Table 10.

Table 10. MapReduce based scheduling

Technique	Description	Advantages	Disadvantages
Delay Tails in MapReduce Scheduling [18]	When the new job is submitted, the mappers join map task and reducers join reduce task. It launches reducers depending on the map task progress and delay tail is found	Reduces starvation times and expedites the processing of all jobs.	Order gain is not always attainable under general conditions
Multi-Job MapReduce Scheduling [53]	When new map/reduce task comes, it computes fractional share of map/reduce slots and map/reduce correlation. Based on this task, it is sorted and scheduled.	Reduces the variability in job completion times and improves the fairness of scheduling.	Not suitable for heterogeneous clusters.
Privacy Aware Hybrid Cloud Scheduling [45]	It guides where the data is to be replicated and to which cloud based on multilevel privacy constraint. It provides which node has authorization to process a map or reduce task.	Flexibly schedules MapReduce processing on data of different sensitivity levels meets company's privacy requirements	Needs to improve performance.

Aware MapReduce scheduling

In Load Aware MapReduce Scheduling to facilitate task scheduling on heterogeneous environments with dynamic loading, a task response time estimation method based on the collected system-level information is used [14]. Scheduling provides a method to add network awareness to global MapReduce so that a scheduler has the information about data locations to launch map tasks at nodes with data [34]. Different aware MapReduce scheduling algorithms are listed in Table 11.

iJETRM
International Journal of Engineering Technology Research & Management

Table 11. Aware MapReduce scheduling

Technique	Description	Advantages	Disadvantages
Load-Aware Scheduling [14]	It checks if there is any task that needs to be scheduled and then estimates the task response time on the TaskTracker. If the response time is shortened it assigns the task to TaskTracker.	Reduces average response time and improves the utilization.	Does not estimate novel response time for Hadoop clusters.
Network-Aware Scheduling [34]	It provides a method to add network awareness to global MapRedue. It has information about virtual machine and physical location of data to launch map tasks at nodes with data.	Reduction in execution time for varying workloads.	Reduce phase adds significant reduction to the overall execution time.

Enhanced MapReduce scheduling

Enhanced MapReduce Scheduling formulates the causes for the waste of system slot resources, through analysis for the current MapReduce scheduling mechanism, which results in the reduce tasks waiting around and the proposes SARS (Self Adaptive Reduce Scheduling), it can determine the start time point of each reduce task dynamically according to each job context, including the completion time of the task and the size of map output [28]. It uses historical information to adjust the stage weights of map and reduce tasks when estimating task execution times [49]. Different Enhanced MapReduce scheduling algorithms are listed in Table 12.

Table 12. Enhanced MapReduce scheduling

Technique	Description	Advantages	Disadvantages
Enhanced Scheduling Algorithm for Avoiding Delay [28]	It schedules the reduce tasks when some of the map tasks are finished but not all. The completed output of map task is given as the input to the reduce task.	Reduces completion time and system average time.	Not suitable for heterogeneous clusters.
Enhanced Self-Adaptive MapReduce Scheduling Algorithm [49]	Uses K-means clustering algorithm to classify historical information into K clusters and estimates task's stage weights to identify slow tasks and re-execute them.	Finds the correct slow reduce task and estimation of TimeToEnd becomes more accurate.	Data locality optimization is less

CONCLUSION

iJETRM
International Journal of Engineering Technology Research & Management

This paper has surveyed the main usage of scheduling in hadoop system, classification of scheduling techniques and its effective scheduling. Four different types of scheduling involved in hadoop have also been discussed. Among the four, job scheduling is found to be the smarter and efficient one. Various scheduling techniques are surveyed and its methodology, advantages, and disadvantages are also compared.

ACKNOWLEDGMENTS

This work is supported in part by Anna University recognized research center lab at VV College of Engineering, Tisaiyanvilai, Tamil Nadu, India.

REFERENCES

[1] Aparna Raj, Kamaldeep Kaur, Uddipan Dutta, V Venkat Sandeep and Shrisha Rao, "*Enhancement of Hadoop Clusters with Virtualization Using the Capacity Scheduler*", IEEE Conference on Services in Emerging Markets, pp: 50 – 57, 2012.

[2] Aprigio Bezerra, Porfidio Hern´andez, Antonio Espinosa and Juan Carlos Moure, "*Job Scheduling in Hadoop with Shared Input Policy and RAMDISK*", IEEE conference, Pages: 355 – 363, Spain,2014.

[3] Aprigio Bezerra, Porfidio Hernández, Antonio Espinosa and Juan Carlos Moure, "*Job Scheduling for Optimizing Data Locality in Hadoop Clusters*", Association for Computing Machinery, Pages: 271 – 276, September 15 - 18 2013, Madrid, Spain

[4] Arun Kumar K, Vamshi Krishna Konishetty, Kaladhar Voruganti and G V Prabhakara Rao, "*CASH: Context Aware Scheduler for Hadoop*", Association for Computing Machinery, Pages: 52 – 61, USA, 2012.

[5] Aysan Rasooli and Douglas G. Down," *A Hybrid Scheduling Approach for Scalable Heterogeneous Hadoop Systems*", IEEE conference, pp:1284 – 1291, Hamilton, Canada, 2012.

[6] Bing Shi and Ankur Srivastava, "*Thermal and Power-Aware Task Scheduling for Hadoop Based Storage Centric Datacenters*", IEEE conference , USA, 2010.

[7] Cai Yandong, Liu Yan and Zhang Qinglei, "*Load balancing task scheduling algorithm in Hadoop platform*", IEEE International Conference on Measuring Technology and Mechatronics Automation, Pages :605 – 608 ,China, 2015.

[8] Dan Tao, Zhaowen Lin and Bingxu Wang, "*Load Feedback-Based Resource Scheduling and Dynamic Migration-Based Data Locality for Virtual Hadoop Clusters in OpenStack-Based Clouds*", Tsinghua science and technology, pp: 149 – 159, Volume 22, Number 2, April 2017.

[9] Dazhao Cheng, Jia Rao, Changjun Jiang and Xiaobo Zhou," *Resource and Deadline-aware Job Scheduling in Dynamic Hadoop Clusters*", IEEE International Parallel and Distributed Processing Symposium, Pages: 956 – 965, Shanghai, China, 2015.

[10] Dazhao Cheng, Xiaobo Zhou, Palden Lama, Jun Wu and Changjun Jiang, "*Cross-platform Resource Scheduling for Spark and MapReduce on YARN*", IEEE transactions on computers, pp: 1 – 14, November 2016.

[11] Deveeshree Nayak, Venkata Swamy Martha, David Threm, Srini Ramaswamy, Summer Prince and G¨unter Fahrnberger, "*Adaptive Scheduling in the Cloud - SLA for Hadoop Job Scheduling*", IEEE conference 2015, Pages:833 – 837, London, UK, July 28-30, 2015.

[12] Divya M. and Annappa B.," *Workload Characteristics and Resource Aware Hadoop Scheduler*" IEEE Conference on Recent Trends in Information Systems, Karnataka, 2015.

[13] Guilherme W. Cassales, Andrea S. Charao, Manuele Kirsch Pinheiro, Carine Souveyet and Luiz A. Steffenel, "*Context-Aware Scheduling for Apache Hadoop over Pervasive Environments*", International Conference on Ambient Systems, Networks and Technologies, Pages: 202 – 209, 2015.

[14] Hsin-Han You, Chun-Chung Yang and Jiun-Long Huang, "*A Load-Aware Scheduler for MapReduce Framework in Heterogeneous Cloud Environments*", Association for Computing Machinery, Pages: 127 – 132, March 21-25, 2011, TaiChung, Taiwan.

[15] Jeffrey Dean and Sanjay Ghemawat, "*MapReduce: Simplified Data Processing on Large Clusters*", Google, Inc. 2004.

[16] Jian Tan, Xiaoqiao Meng and Li Zhang, "*Coupling Task Progress for MapReduce Resource-Aware Scheduling*", IEEE conference, New York, 2015.

[17] Jian Tan, Xiaoqiao Meng and Li Zhang, "*Delay Tails in MapReduce Scheduling*", Association for Computing Machinery, Pages: 5 – 16, London, England, UK, June 11–15, 2012.

[18] Jiazhen Han, Zhengheng Yuan, Yiheng Han, Cheng Peng, Jing Liu and Guangli Li, "*An Adaptive Scheduling Algorithm for Heterogeneous Hadoop Systems*", IEEE ICIS 2017,Pages: 845 - 850, Wuhan, China,may 2017.

[19] Jisha S Manjaly and Varghese S Chooralil,"*TaskTracker Aware Scheduling for Hadoop MapReduce*", IEEE International Conference on Advances in Computing and Communications, Pages: 278 - 281, Kochi, India, 2013.

[20] Li Liu, Yuan Zhou, Ming Liu, Guandong Xu, Xiwei Chen, Dangping Fan and Qianru Wang, "*Preemptive Hadoop Jobs Scheduling under a Deadline*", IEEE International Conference on Semantics, Knowledge and Grids, Pages: 72 – 79, Australia,2012.

[21] Lizhe Wang, Samee U. Khanc, Dan Chena, Joanna Kołodziej, Rajiv Ranjan, Cheng-zhong Xu and Albert Zomaya, "*Energy-aware parallel task scheduling in a cluster*", Elsevier Future Generation Computer Systems, Pages: 1661–1670 18 March 2013.
[22] Matei Zaharia, Dhruba Borthakur, Joydeep Sen Sarma, Khaled Elmeleegy, Scott Shenker and Ion Stoica, "*Job Scheduling for Multi-User MapReduce Clusters*", Electrical Engineering and Computer Sciences, University of California at Berkeley, Pages: 1 – 16, April 30, 2009.
[23] 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*", Association for Computing Machinery, Paris, France, April 13–16, 2010.
[24] Mbarka Soualhia, Foutse Khomh and Sofiene Tahar, "*ATLAS: An Adaptive Failure-Aware Scheduler for Hadoop*", IEEE Conference, 2015, Canada.
[25] Michael Isard, Vijayan Prabhakaran, Jon Currey, Udi Wieder, Kunal Talwar and Andrew Goldberg, "*Quincy: Fair Scheduling for Distributed Computing Clusters*", Microsoft Research, Silicon Valley, Pages: 1 – 20, USA.
[26] Michael J. Fischer, Xueyuan Su and Yitong Yin, "*Assigning Tasks for Efficiency in Hadoop*", SPAA'10, Thira, Santorini, Greece, June 13–15, 2010.
[27] Nadhiya.H, Nilavunesan.D, Raju.S, Anitha.R and Rajalakshmi.S, "*Enhanced Scheduling Algorithm for Avoiding Delay in MapReduce*", International Journal for Research in Applied Science & Engineering Technology, Volume 4, Issue VIII, India, August 2016.
[28] Nusrat Fatma , Remant Kr. Singh, Shafeeq Ahmad and Prachi Srivastava, "*Locality Premised Reducer Scheduling in Hadoop*", IEEE International Conference on System Modeling & Advancement in Research Trends, Pages:222 – 224, Lucknow, India, November, 2016.
[29] Orcun Yildiz, Shadi Ibrahim, Tran Anh Phuong and Gabriel Antoniu, "*Chronos: Failure-Aware Scheduling in Shared Hadoop Clusters*", IEEE International Conference on Big Data, Pages: 313 – 318, France, 2015.
[30] Peng zhang, Chunlin Li and Yahui Zhao, "*An improved task scheduling algorithm based on cache locality and data locality in Hadoop*",17th IEEE Conference on Parallel and Distributed Computing, Applications and Technologies,China, pp. 244 – 249,Dec 2016
[31] Ping Li,Lei Ju, Zhiping Jia and Zhiwen Sun ,"*SLA-Aware Energy-Efficient Scheduling Scheme for Hadoop YARN*", IEEE International Conference on High Performance Computing and Communications, Pages: 623 – 628, China, 2015.
[32] Poornima K S and Akshatha G, "*Scheduling With Tasktracker For Mapreduce In Hadoop*", International Journal of Contemporary Research in Computer Science and Technology (IJCRCST), Volume3, Issue 7, July 2017, Karnataka,India.
[33] Praveenkumar Kondikoppa , Chui-Hui Chiu, Cheng Cui, Lin Xue and Seung-Jong Park, "*Network-Aware Scheduling of MapReduce Framework on Distributed Clusters over High Speed Networks*", Association for Computing Machinery , Pages: 38 - 43, September 21, 2012, San Jose, California, USA.
[34] Qi Wang and Xiaojun Huang, "*PFT: A Performance-Fairness Scheduler on Hadoop YARN*", IEEE conference, pp: 76 – 80, China, 2016.
[35] Qi Zhang, Mohamed Faten Zhani, Yuke Yang and Raouf Boutaba, "*PRISM: Fine-Grained Resource-Aware Scheduling for MapReduce*", IEEE transactions on cloud computing, vol. 3, no. 2, April/June 2015.
[36] Ruiqi Sun, Jie Yang, Zhan Gao and Zhiqiang He, "*A virtual machine based task scheduling approach to improving data locality for virtualized Hadoop*", IEEE ICIS, Taiyuan, China, June 4-6, 2014.
[37] S.Gunasekaran, L.SaiRamesh, S.Sabena, K.Selvakumar, S.Ganapathy and A.Kannan, "*Dynamic Scheduling Algorithm For Reducing Start Time In Hadoop*", Association for Computing Machinery, August 25-26, 2016, Pondicherry, India.
[38] Shadi Ibrahim, Hai Jin, Lu Lu, Bingsheng He, Gabriel Antoniu and Song Wu, "*Maestro: Replica-Aware Map Scheduling for MapReduce*", IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, Pages: 435 – 442, 2012.
[39] Shengli Gao and Ruini Xue ," *BOLAS+: Scalable Lightweight Locality-aware Scheduling for Hadoop*", IEEE TrustCom- BigDataSE-ISPA, Pages: 1077 – 1084, Chengdu, China, 2016.
[40] Shiori KURAZUMI, Tomoaki TSUMURA, Shoichi SAITO and Hiroshi MATSUO, "*Dynamic processing slots scheduling for I/O intensive jobs of Hadoop MapReduce*", IEEE Conference on Networking and Computing, Pages: 288 – 292, Japan, 2012.
[41] Shivani Thakur, Rupinder Singh and Sugandha Sharma, "*Dynamic Capacity Scheduling in Hadoop*", International Journal of Computer Applications, Volume 125 – No.15,Pages: 25 – 28, September 2015.
[42] Supriya Pati and Mayuri A. Mehta, "*Job Aware Scheduling in Hadoop for Heterogeneous Cluster*", IEEE International Advance Computing Conference (IACC), Banglore, India ,pp.: 778 – 783,june 2015.
[43] Sutariya Kapil B. and Sowmya Kamath S., "*Resource Aware Scheduling in Hadoop for Heterogeneous Workloads based on Load Estimation*", IEEE conference, Tiruchengode, India, July 4-6, 2013.
[44] Thomas Sandholm and Kevin Lai, "*Dynamic Proportional Share Scheduling in Hadoop*", Hewlett-Packard Laboratories, USA.

[45] Toon Degryse and Sucha Smanchat, "*MapReduce Scheduling in Hybrid Cloud with Multi-level Privacy*", Association for Computing Machinery, December 11-13, 2015, Brussels, Belgium.
[46] Xiao Qin and Hong Jiang, "*A Dynamic and Reliability-Driven Scheduling Algorithm for Parallel Real-time Jobs on Heterogeneous Clusters*", New Mexico Institute of Mining and Technology, Socorro, New Mexico.
[47] Xiaofei Huang, Hui Zhou and Wei Wu, "*Hadoop Job Scheduling Based on Hybrid Ant-Genetic Algorithm*", IEEE conference 2015, Pages: 226 – 229, china, 2015.
[48] Xiaolong Xu, Lingling Cao, and Xinheng Wang, "*Adaptive Task Scheduling Strategy Based on Dynamic Workload Adjustment for Heterogeneous Hadoop Clusters*", IEEE SYSTEMS JOURNAL, VOL. 10, NO. 2, Pages: 471 - 482, JUNE 2016.
[49] Xiaoyu Sun, Chen He and Ying Lu, "*ESAMR: An Enhanced Self-Adaptive MapReduce Scheduling Algorithm*", IEEE International Conference on Parallel and Distributed Systems, Pages 148 – 155, 2012, U.S.A.
[50] Ya-Wen Cheng and Shou-Chih Lo," *Improving Fair Scheduling Performance on Hadoop*", IEEE conference, Hualien, Taiwan, 2017.
[51] Yi Yao, Jiayin Wang, Bo Sheng, Jason Lin and Ningfang Mi," *HaSTE: Hadoop YARN Scheduling Based on Task-Dependency and Resource-Demand*", IEEE International Conference on Cloud Computing, pages: 184 – 191, Boston, 2014.
[52] YongLiang Xu and Wentong Cai, "*Hadoop Job Scheduling with Dynamic Task Splitting*", IEEE International Conference on Cloud Computing Research and Innovation, Pages:120 – 129, Singapore, 2015.
[53] Yuting Ji, Lang Tong, Ting He, Jian Tan, Kang-won Lee, and Li Zhang, "*Improving Multi-Job MapReduce Scheduling in an Opportunistic Environment*", IEEE International Conference on Cloud Computing, Pages: 9 – 16, 2013, USA.
[54] Zhuo Tang, Junqing Zhou, Kenli Li and Ruixuan Li, "*A MapReduce task scheduling algorithm for deadline constraints*", Springer Science and Business Media, New York, 2012.
[55] Zhuo Tang, Lingang Jiang, Junqing Zhou, Kenli Li and Keqin Li, "*A Self-Adaptive Scheduling Algorithm for Reduce Start Time*", Future Generation Computer Systems, March 26, 2014.