Journal of
Information Technology & Software Engineering

Review Article Open Access

Job Scheduling in Big Data – A Survey

Seethalakshmi V*, Govindasamy V and Akila V

Department of Computer Science and Engineering, Pondicherry Engineering College, Puducherry, India

Abstract

A huge amount of information (Information in the unit of Exabyte or Zettabyte) is called Big Data. To quantify such a large amount of data and store electronically is not easy. Hadoop system is used to handle these large data sets. To collect Big Data according to the request, Map Reduce program is used. In order to achieve greater performance, Big Data requires proper scheduling. To reduce starvation and increase the use of resource and also to assign the jobs for available resources, the scheduling technique is used. The Performance can be increased by implementing deadline constraints on jobs. The goal of the paper is to study and analyze various scheduling algorithms for better performance.

Keywords: Job scheduling; Hadoop; MapReduce operation; Big data

Introduction

Currently, the phrase Big Data has become most fashionable in IT region. It refers to a broad area of dataset which are tough to be maintained by traditional functions [1]. Big Data can be used in Economics and Commerce, Finance, Electronic shopping, Medicare, Astrophysics, Oceanology, Manufacturing and numerous different areas. These datasets are most difficult. The data size is increasing exponentially day by day in extremely huge quantity. As information is rising in capacity, in variety and with huge speed, it also increases the complexities in handling it. Big Data is a developing area. It has a lot of investigation problems and objections to address. The major problems in Big Data are: i) Managing data quantity, ii) Analysis of Big Data, iii) Privacy of information, iv) Holding of massive quantity of information, v) Information visualization, vi) Job scheduling in Big Data, vii) Fault tolerance.

Research Methodology

Managing information quantity

The huge quantity of information/data imminent from various areas of education such as genetics, astrophysics, weather forecasting, etc makes it extremely hard for the biologists to handle [1,2].

Analysis of big data

It is hard to diagnose Big Data due to inhomogeneous and incompleteness of information. Composed information can be in various methods, diversity and structure [3].

Privacy of information in the context of big data

There is a general fear regarding to the improper utilization of individual information, mainly through connecting the information from numerous resources. Handling secrecy is both a scientific and a Sociological issue [3].

Storage of massive quantity of information

It constitutes the issue of how to identify and cache main data which are separated from unorganized information, proficiently [1,3].

Information visualization

Information handling methods should be proficient enough to allow real time visualization [1].

Job scheduling in big data

These issues focus on adequate scheduling of tasks in a distributed environment [4].

Fault tolerance

It is one more problem in Hadoop frame in Big Data. In Hadoop, Name Node is an only point of crash. Duplication of chunk is one of the fault tolerance method used by Hadoop. Fault tolerance methods must be powerful enough to handle failure in distributed environment. MapReduce provides a perfect frame for handling of such huge datasets by using analogous and distributed programming methods [5,6].

MapReduce

MapReduce operation depends on double functions such as Map and Reduce operation. Both, the Map and Reduce operations are written based on the needs of the customer. The Map operations obtain an input pair and produce a set of middle key. Then, the MapReduce files will gather the entire middle value that is combined with the similar middle key and transfer them into the Reduce operation for additional operations. The Reduce function obtains a middle key with an integrated set of values. It associates those values to make it as a lesser set of values. Figure 1 shows the entire process of MapReduce.

Hadoop Architecture

Scheduling techniques which are captured by the master node are called as JobTracker scheduling technique captured by the slaves nodes are called as Task Tracker which will execute the tasks.

A Hadoop cluster consists of a one master node and several slave nodes. Figure 2 shows Hadoop Architecture. A single master node comprises of a JobTracker, TaskTracker, Data node and Name node [7].

Job tracker

The main role of the Jobtracker is to manage the task trackers and tracking source availability. The JobTracker is a node which controls the job implementation method. Jobtracker executes MapReduce tasks to a

***Corresponding author:** V Seethalakshmi, Research Scholar, Department of Computer Science and Engineering, Pondicherry Engineering College, Puducherry, India, Tel: + 02-01155516545; E-mail: seethalakshmi.v@pec.edu

Received February 07, 2018; **Accepted** February 24, 2018; **Published** March 05, 2018

Citation: Seethalakshmi V, Govindasamy V, Akila V (2018) Job Scheduling in Big Data – A Survey. J Inform Tech Softw Eng 8: 226. doi: 10.4172/2175-7866.1000226

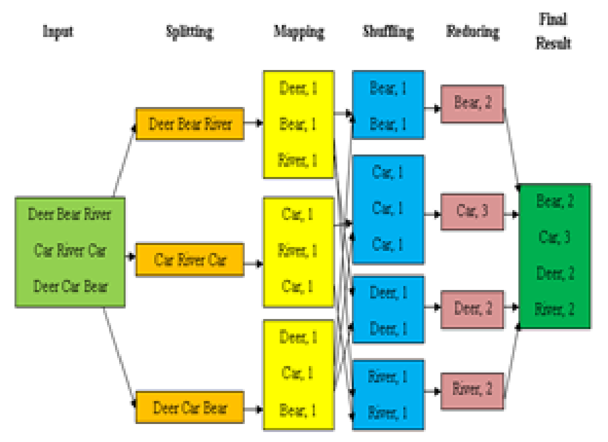

Figure 1: The overall MapReduce word count process.

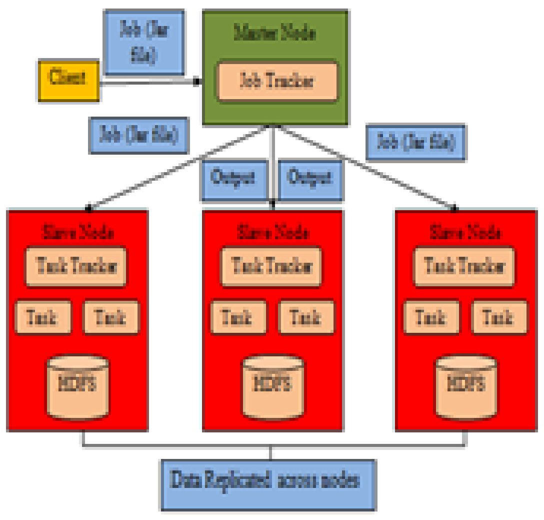

Figure 2: Hadoop architecture [11].

particular node in the group. User submits task to the Jobtracker. When the task is finished, the Jobtracker informs its status. User applications can request the Jobtracker for data.

Task tracker

It follows the sort of the Jobtracker and informs the Jobtracker with its position periodically. Task trackers run the tasks and delivery the reports to Jobtracker, which maintain a whole report of each and every job. Every Task tracker is designed with a set of slots which specify the number of tasks that it can allow.

Name node

The name node refers to block areas. Whenever a data node undergoes a floppy corruption of a precise chunk, At the Initial table and modified table gets updated, but whenever a data node is identified to be dead due to system crash or a node failure, both the tables get restored. Updating the tables is based on only loss of the nodes. It does not depend on any neighbor chunks or any block areas to find its destination. Each chunk is divided with its job nodes and various allocated method (Table 1).

Data node

The Data node stores the data in Hadoop system. All data nodes send a heartbeat message to the name node for every three seconds to say that they are alive. If the name node does not get any heartbeat message from a particular data node for ten minutes, then the name node considers that the data node is dead or out of service. It starts some new data node for the process. The data node restores the name node with the chunk data periodically.

Job Scheduling in Bigdata

For scheduling Hadoop presents default FIFO scheduler in which the tasks are scheduled in FIFO sequence. Since this scheduler is not fine for a few tasks, substitute scheduling techniques are used. The design for referring priority for jobs was planned earlier period. Fair Scheduler and Capacity Scheduler were result discharged into the Hadoop shared respectively. This section explained various Job Scheduling techniques [8,9].

Default FIFO scheduling

First In First Out (FIFO) Scheduling is a default scheduler of Hadoop. In this technique, JobTracker pulls the initial task first from the job queue. The FIFO Scheduler is not considering a size or priority of the task. In Hadoop, the built-in scheduler is FIFO. Once the job is divided into specific tasks they are feed into the queue and owed to idle slots as they are available on TaskTracker nodes. Generally every task will utilize the whole group, so the jobs require waiting for their turn. The map task in the job with information closest to the slave is picked up by the Task Tracker. The main disadvantage of FIFO technique is poor response time [10,11].

Fair scheduling

Fair scheduling is an approach of assigning assets to jobs where all jobs have an equally distributed. When there is only one task working, the task will utilize the whole group. When new tasks are surrendered, free task slots are owed to the new tasks, so that each one gets an equivalent bulk of CPU time. It permits small jobs complete within a reasonable time while not starving lengthy tasks. The scheduler organizes tasks by assets pool, and distributes sources evenly between the pools. By default, for every user there is an independent pool. There are certain restricts for concurrently working Map tasks and Reduce tasks on “TaskTracker” of node.

The major plan behind the fair scheduler is to distribute assets to tasks such that every task gets an equivalent distribute of the available assets. Thus tasks that need fewer times are able to use the CPU and complete simultaneously with the implementation of jobs that need extra time to implement. Sufficient assets are owed for shorter jobs to complete quickly. At the same time, longer jobs are assured to not be starved of assets. Due to this act interactivity among Hadoop tasks appears and offers higher respectively of the Hadoop group to the variety of job types submitted t. Fair scheduler constructed the Facebook.

Capacity scheduling

Both the capacity scheduling and fair scheduling are very similar in their algorithm design. The main difference is that fair scheduler utilizes pools whereas capacity scheduler utilizes queues. Capacity scheduler is assigned to an organization in each queue and assets are dividing among these queues. In the capacity scheduling tasks are allocated to several queues and each queue is assigned a certain capacity. Assume if the queue is weighty then it examines for yet unallocated assets. In order to raise the use of assets and it permits the re-allocation of assets to queues utilize the fullest capacity. As soon as the tasks come in the

Scheduling Algorithms	Technology	Advantages	Disadvantages
Default FIFO Scheduling	The algorithm schedules the tasks based on their priorities in first-in first-out	1. Cost of entire cluster scheduling process is less. 2. It is easy to implement and is well-organized.	1. It is designed only for its' own type of job. 2. The performance is low when the algorithm is run numerous types of jobs. 3. The techniques results in reduced response time for little jobs compared to the bulky jobs.
Fair Scheduling [8]	Here the various Users/jobs are equal distribution for compute resources in the system.	1. It is Less complex 2. They have Both the little and big clusters are worked well. 3. It can provide quick response times for little jobs mixed with bigger jobs.	This Mechanism Does not consider the job load of each node.
Capacity Scheduling [10]	It enhances Maximization the resource use and throughput in multi-tenant group environment.	Ensure guaranteed access with the potential to recycle unused capacity and prioritize tasks within queues over huge cluster.	These schedulers are more complex compare to other schedulers
Dynamic Proportional Scheduling [12]	They are Designed for data intensive workload and try to retain data locality during job implementation.	1. It is a flexible and fast scheduler compare to other scheduler. 2. This scheduler improves response time.	If the method eventually crashes, then all unfinished low priority processes gets lost.
Resource-Aware Adaptive Scheduling (RAS) [13]	It is used for Dynamic Free Slot Advertisement and Free Slot Priorities/ Filtering	It improves the Job performance.	This allows Only takes action on appropriate slow tasks.
MapReduce task scheduling with deadline constraints (MTSD) [15]	The scheduling technique achieves nearly complete overlap via the new idea of including decrease in the overlap.	1. Here the Computation time is reduced 2. This system Increase performance for the main class of shuffle-heavy Map Reductions.	It works better with small clusters only.
Delay Scheduling [18]	This is used to concentrate on the conflict between locality and fairness.	It is very Simple to implement.	No details specified.
Multi Objective Scheduling	The implementation type consider as all the MapReduce tasks are autonomous, there is no nodes crash before or during the scheduling calculation and the scheduling result is taken only based on current knowledge.	It keeps performance is high.	In this scheduling algorithm Execution Time is too large.
Hybrid Multistage Heuristic Scheduling (HMHS)	In this paper Johnson's algorithm and Min-Min and Dynamic-MinMin algorithm used	This system achieves not only high data locality speed but also high cluster use.	It does not ensure reliability.

Table 1: Comparison of various job scheduling algorithms in Bigdata.

queue, the task that is presently working is finished and the assets are passed on the innovative queue. Dissimilar capacity scheduling, First In First Out scheduler, permits priority based scheduling. The major disadvantage of this scheduler is that, the capacity scheduler is the most difficult of all three schedulers.

Dynamic proportional scheduling

As claimed by Sandholm and Lai, Dynamic Proportional scheduling gives a lot of job allocation and prioritization that ends in rising segment of cluster assets and a lot of separation in service stages of numerous jobs [12]. Multi-user Hadoop environments reduce response time for this algorithm.

Resource-aware adaptive scheduling (RAS)

To increase utilization of resource among machines even as monitoring the completion time of process. RAS proposed by Polo et al. for the Map Reduce with multi-job workloads [13].

Zhao et al. provides job scheduling algorithm based on the Resource Attribute Selection to work out its resource allocated by sending a cluster of test jobs to an implementation node previously a job is scheduled and so select optimal node to implement a job reliable with resource requirements and appropriateness between the resource node and consequently the job, which uses history job information [14].

MapReduce job scheduling with deadline constraints (MTSD) algorithm

According to Tang et al. [15] algorithm locates double deadline: Map-deadline and Reduce-deadline. Reduce-deadline is simply the clients' job destination. Pop et al. [6] present a classic method for a periodic job scheduling by since a scheduling method with totally various queues for periodic and a periodic role and deadline, since the major constraint progresses a technique to guess the quantity of assets required to schedule a cluster of an interrupted jobs or purpose, by since along implementation and information transmission costs [16]. Based on a numerical classical, and by using dissimilar imitation circumstances, MTSD established the subsequent reports: (1) varied sources of autonomous an episodic jobs will be calculated approximating to a only one; (2) when the quantity of assessed resources transcend a information centres competence, the jobs exodus between completely diverse regional centres' is that the appropriate resolution with significance the global deadline; and (3) during heterogeneous information center, we want advanced diversity of resources for an equivalent demand with significance the deadline constraints. In MapReduce, Wang and Li [2] detailed the job scheduling, for dispersed information centers on heterogeneous systems through adaptive heartbeats, task deadlines and data locality [17]. Task deadlines are dividing alongside the foremost information quantity of jobs. With the thought of constraint, the job scheduling is wrenched as a task downside in each heartbeat, during which adaptive heartbeats are hypothetical by the process times of jobs and task are sequencing in terms of the separated deadlines and jobs are planned by the Hungarian algorithmic schedule. On the knowledge of information transfer and process times, the furthermost suitable data center for all mapped jobs are determined within the reduce part.

Delay scheduling

The purpose is to agree with the dispute among locality and fairness. Once a node desires for a job or function, if the head-of-line job cannot design a local job, then the scheduler neglects that job and appears to future jobs. If a job has been mislaid then, it tends to license it to design non-native jobs, to circumvent starvation.

This scheduling provisionally relaxes fairness to persuade higher locality through permitting jobs to appear for scheduling on a node amongst native information. Song et al. it provides a game statement based system to solve scheduling difficulties by unravelling a Hadoop scheduling issue into two stages like task level and job level [18]. For the job level scheduling, prototypical bid is used to have assurance to the fairness and decrease the common waiting time. For jobs level, alteration scheduling disadvantage into assignment problematic and utilize Hungarian methodology to improve the trouble Wan et al. [12] provides multi-job scheduling algorithm in MapReduce reinforced game assumption that obligate with the competition for resources among numerous jobs.

Multi objective scheduling

Nita et al. [1] explain about scheduling algorithm entitled Multi Objective Scheduling Algorithm of Many Task in Hadoop (MOMTH) by considering unbiased functions associated to resources and clients within the alike time with constraints alike to deadline and budget.

The enact model takes into version as all MapReduce jobs are autonomous. As there is no nodes breakdown prior to during scheduling calculation, scheduling resolution is taken exclusively found on the present data. Bian et al. present scheduling approach. Consistent with this scheduling approach, the cluster discovers the velocity of the present nodes and creates approximate backups of the in-between MapReduce data which consequences to a great performance cache server. The information produced by that node could get incorrect shortly. Hence the cluster could resume the implementation to the preceding level rapidly if there are many nodes going incorrect, then the cut back nodes will scan the Map output from the cache server or from together the cache and also from the node, and keep its performance great.

Hybrid multistage heuristic scheduling (HMHS)

Chen et al. [13] explain heuristic scheduling algorithm entitled Hybrid Multistage Heuristic Scheduling (HMHS) that makes an effort to clarify the scheduling trouble by rending it into 2 sub problems: sequencing and dispatching. For sequencing, they utilize heuristic maintained Pri (the modified Johnson's algorithm). For dispatching, they indorse double heuristics techniques such as Min-Min and Dynamic Min-Min [19].

Discussion

This paper provides the taxonomy of Hadoop schedulers based on various parameters such as priority, time, resource etc. It discusses about how different task scheduling algorithms help in gaining better outcome in Hadoop cluster. Furthermore, this paper also discusses about the advantages and the disadvantages of various task scheduling algorithms. This comparison results show the advantage and the disadvantage of all scheduling algorithm. Thus, all algorithms are important in job scheduling.

Conclusion

This paper provides an overall idea about the various jobs scheduling algorithms in the Big Data. Further it compares most of the properties of various task scheduling algorithms. Individual scheduling techniques improves the efficiency, data locality, makespans, fairness and performance of this job scheduling are elaborated and discussed. However, the scheduling technique is an open area for researchers to explain.

References

1. Nita MC, Pop F, Voicu C, Dobre C, Xhafa P (2016) F- MOMTH: Multi-object scheduling algorithm of many task in hadoop. Cluster Computing 19: 1011-1024.
2. Wang J, Li X (2016) Task scheduling for MapReduce in heterogeneous networks. Cluster Computing 19: 197-210.
3. Assuncao M, Calheiros R, Bianchi S, Netto M, Buyya R (2014) Big data computing and clouds: Trends and future directions. J Parallel Distrib 79-80: 3-15.
4. Kaur M, Shilpa (2014) Big data visualization tool with advancement of challenges. International Journal of Advanced Research in Computer Science and Software Engineering 4.
5. Suthaharan S (2014) Big data classification: Problems and challenges in network intrusion prediction with machine learning. ACM SGMETRICS Performance Evaluation Review 41: 70-73.
6. Pop F, Dobre C, Cristea V, Bessis N, Xhafa F, et al. (2014) Deadline scheduling for a periodic tasks in inter-cloud environments: A new approach to resource management. Journal of Super Computing 71: 1754-1765.
7. Chen J, Wang D, Zhao W (2013) A task scheduling algorithm for Hadoop platform. Journal of Computer 8: 929-936.
8. Bardhan S, Menasce DA (2013) The anatomy of MapReduce jobs, scheduling, and performance challenges. Conference of the Computer Measurement Group, San Diego, CA, USA.
9. Zhao Y, Chen L, Li Y, Liu P, Li X, et al. (2013) RAS: A task scheduling algorithm based on resource attribute selection in a task scheduling framework-IDCS. LNCS 8223: 106-119.
10. Song G, Yu L, Meng Z, Lin X (2013) A game theory based MapReduce scheduling algorithm emerging technologies for information systems, Computing and Management 236: 287-296.
11. Apache Hadoop 2.9.0 – Hadoop: Capacity Scheduler.
12. Wan C, Wang C, Yuan Y, Wang H (2013) Game-based scheduling algorithm to achieve optimize profit in mapreduce environment. ICIC, LNCS 7995: 234-240.
13. Chen H, Shen Y, Chen Q, Guo M (2013) HMHS: Hybrid multistage heuristics scheduling algorithm for heterogeneous MapReduce system ICA3PP, Part I, LNCS 8285: 196-205.
14. Chen Y, Alspaugh S, Katz RH (2012) Interactive analytical processing in big data systems: A cross-industry study of MapReduce workloads proceeding of the VLDB endowment 5: 12.
15. Tang Z, Zhou J, Li K, Li R (2012) A MapReduce task scheduling algorithm for deadline constraints cluster computing 16: 651-662.
16. Jorda P, Castillo C, Carrera D, Becerra Y (2011) Resource-aware adaptive scheduling for MapReduce cluster in ACM/IFIP/USENIX. International Conference on Distributed Systems Platforms and Open Distributed Processing 16: 187-207.
17. Sandholm T, Lai K (2010) Dynamic proportional share scheduling in Hadoop. Proceeding of the 15th Workshop on Job Scheduling Strategies for Parallel Processing 6253: 110-131.
18. Matei Z, Borthakur D, Sarma SJ, Elmeleegy K (2010) Delay scheduling a simple technique for achieving locality and fairness in cluster scheduling. In: Proceedings of the 15th European Conference on Computer Systems 2: 265-278.
19. Borthakur D (2007) The Hadoop distributed file system: Architecture and design. Hadoop Project Website 11: 21.