

Novel Job Scheduling Algorithm using Dynamic Queue Management Scheme in Hadoop

Anuradha Sharma¹, Er. Sheenam Malhotra²

M.Tech Student, Dept. of CSE, Sri Guru Granth Sahib World University, Punjab, India¹

Assistant Professor, Dept. of CSE, Sri Guru Granth Sahib World University, Punjab, India²

ABSTRACT: In today's digital world data produced by individuals and organizations ranges from tera-bytes to zeta-bytes which is named as Big Data. But the most common difficulty for the companies is to store and analyze such amount of data with few resources. The most prominent solution for this problem is given by Apache Hadoop that works in distributed environment. With increase in use of Hadoop, job scheduling has become an important factor to achieve high performance. Various scheduling algorithms are proposed in past few years for optimal utilization of cluster resources. In this paper, a novel job scheduling algorithm is proposed which focuses on reduction of execution time with maximum resource utilization. The results show significant decrease in completion time of a job as compared to existing job aware scheduling algorithm.

KEYWORDS: Big Data, Hadoop, HDFS, MapReduce, Job Scheduling.

I. INTRODUCTION

Due to rise in storage capacities, processing power and availability of data, the size of data around the world is expanding day by day [1]. Traditional data warehousing and management systems fall short of tools to evaluate this data. It is a difficult task to analyze data as it involves large distributed file systems which should be fault tolerant, expandable and scalable [2]. Almost every sector and business function in the enterprise has a big data application. This increase in data generation is being driven by organization individuals; moving from analog to digital technologies, social media and Internet of Things. The amount and heterogeneity of data with the rate it is generated, makes it complex for the present computing infrastructures to deal with big data [3].

The most extensively used tool for big data landscape is Hadoop, an open source implementation of MapReduce Framework currently maintained by Apache Foundation. Hadoop offers variety of utilities such as distributed file system, job scheduling, resource managing potential and java API for writing applications [4]. The two vital components of Hadoop- HDFS (Hadoop Distributed File System) and MapReduce are used for storing, managing and analyzing the data. MapReduce is a minimization method which makes use of file indexing with mapping, categorization, shuffling and finally reducing [2].

Due to increase in data intensive jobs, scheduling in Hadoop has become a significant factor to attain high performance by efficiently managing the resources in Hadoop cluster. The purpose of job scheduling is to process and finish as many jobs as possible in less time. Though several job scheduling algorithms are offered focusing on problems such as data locality, fairness, synchronization and resource awareness; but do not concern much about the execution time of a job. In this paper, the proposed novel job scheduling algorithm uses dynamic queue management scheme which focuses on the criteria of reducing the job execution time with maximum utilization of resources.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

In the remaining part of the paper, section II explains related work done. In section III the proposed algorithm is discussed. Section IV provides performance analysis of proposed work. Section V specifies conclusion and future scope.

II. RELATED WORK

Hadoop schedulers are pluggable components which allocate resources to jobs. In a variety of schedulers, well-known are the default FIFO, Fair and Capacity schedulers. The default Hadoop scheduler works using FIFO queue i.e. execute jobs on first come first out basis. However, it does not provide fair share of resources among users. The Fair scheduler intends to give each user a fair share of cluster capacity and system resources. It is much extensive towards better resource utilization and supports preemption also. Yahoo's Capacity scheduler is developed to run Hadoop applications as a communal and multi-tenant cluster while exploiting the throughput and utilization of cluster. Its design is mostly similar to fair scheduler but it also provides features like hierarchical queues, capacity guarantees and operability [1] [3] [5].

Data locality, synchronization and fair share of resources are the major factors affecting scheduling of jobs. In [6] [7], focus is on improving data locality rate. The Matchmaking technique is developed to enhance map task's data locality. A locality marker is used to mark nodes and to ensure that each node has a fair chance to grab its local tasks [6]. In [8], authors proposed a technique named delay scheduling that addresses the problem of data locality and fairness. For resolving synchronization overhead, two techniques are discussed- asynchronous processing and speculative execution in [9-12]. User constraints such as priority, deadlines are also considered important in Hadoop [13] [14]. In [14] a MapReduce Task Scheduling algorithm is proposed which allows user to specify job's deadline. In shared environments, considering the workload of jobs various techniques were discussed in [15-17] to manage & improve the performance of Hadoop. In [18], authors proposed job aware scheduling algorithm which reduces the average waiting time in heterogeneous cluster, but can be decreased to some more extent.

Thus, various job scheduling algorithms have been proposed in recent years as discussed in literature [6-18]. However reduction in execution time of job has not been focused. Reducing the execution time of a job on single node will reduce the load on Hadoop cluster as well and will maximize the resource utilization. Hence, a novel job scheduling algorithm is proposed which will schedule the tasks in pipelining manner so as to decrease the time for their execution.

III. THE PROPOSED NOVEL JOB SCHEDULING ALGORITHM

The proposed scheduling algorithm is designed for Hadoop cluster having a NameNode and DataNode. The NameNode will act as Master Node and DataNode as the Slave Node. NameNode is responsible for monitoring the DataNode and maintaining all the metadata while DataNode holds the actual data for the jobs to be executed. A job is composed of various sub-tasks which are categorized as map and reduce tasks. It involves input data, split phase, Map phase, intermediate data, Reduce phase and output data as shown in Fig. 1.

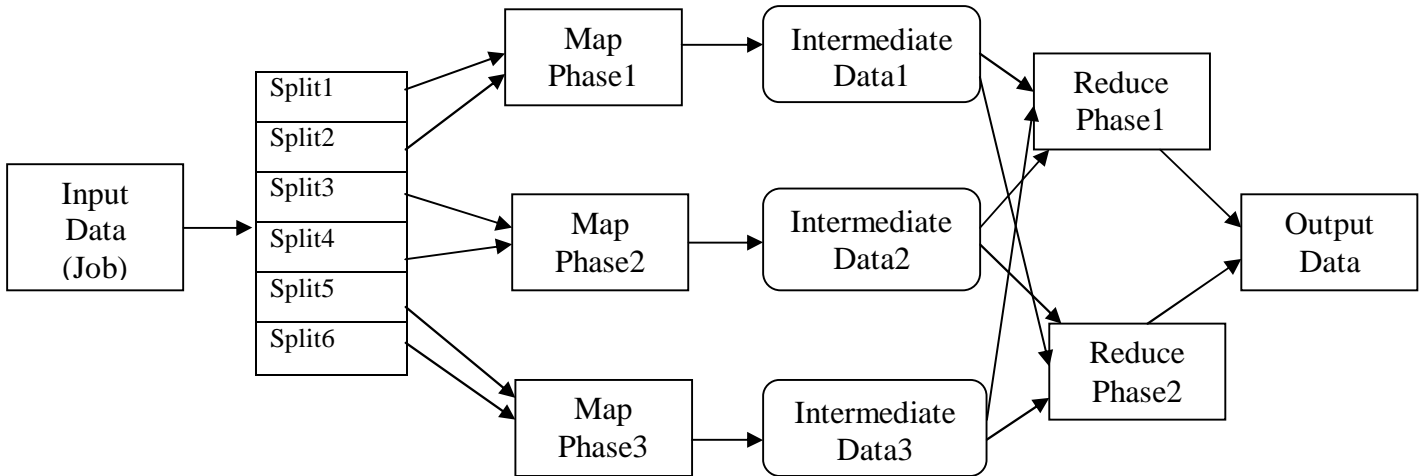


Fig. 1 Hadoop MapReduce Workflow

The steps for proposed novel job scheduling algorithm are described below and are shown in Fig. 2. The Master Node maintains the queue for jobs. The jobs are initially submitted to CsLeafQ managed by QueueManager. Initialize all nodes of Hadoop distributed file system (HDFS) and resources will be allocated dynamically by ResourceManager according to need of the job. Capacity scheduler will be initialized after resource allocation which will start Map and Reduce functions to execute tasks.

```

Input: Job Queue, NameNode, DataNode
Output: Execution time of completed job
1. FOR (CsLeaf Q: QueueManager)
2. Initialize (CapacityScheduler)
3. ResourceManager.add (ResourceToPreempt)
4. IF (Resources greater than Resource Calculator) then
5.     Initialize Pipelining()
6.     Preempt + Resource (QueueManager. Get LeafQueue)
   // Collect Running Container
7.     FOR (capacitysched: sched)
8.         IF (Resource. greater than Resource Calculator) then
9.             Pipelining( linearly.processed)
10.            FOR (appsched: sched.getCapacityAppSchedulable)
11.                FOR (RMContainer: getLiveContainer)
12.                    Running Container.add(c)
13.                    Apps.put (c, appsched.getApp ())
14.                    Queues.put (c, sched)
   //kill all process
15.     IF (time~= null) then
16.     IF (time + waitTimeBeforeKill<clock.getTime ()) then
17.     Create preempted Container Status
18.     (Container .getContainerId ())
19.     CompletedContainer (Containers, Status, RMContainer.KILL)

```

Fig. 2 Algorithm for proposed work

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

Once the MapReduce phase is started, it will split the job into sub parts according to the size of job and these will be placed in a queue. ResourceCalculator will analyze the resources i.e. to check whether a job is allocated sufficient resources or not. If resources allocated are greater than ResourceCalculator’s limit, job execution will be finished. Otherwise pipelining will be activated so as to utilize resources in precise manner.

The Map and Reduce functions will execute tasks of a job through pipelining. RunningContainer of scheduler will be collected which will tell the availability of new resources if any. If a node reports in with a finished container, the Capacity Scheduler selects a suitable queue to exploit the newly available resources based on capacity. Tasks in the Container will be killed and execution time for the job is estimated.

IV. PERFORMANCE ANALYSIS

The proposed algorithm is evaluated using two parameters: average waiting time and resource utilization of the system. The experimental analysis is carried out using MapReduce WordCount job as benchmark in Hadoop 2.2.0. Five MapReduce jobs are assumed each having different size and are submitted to a queue and their execution time is shown in Table I. The performance of the proposed algorithm for best case, worst case and average case is evaluated. The execution time for jobs with larger sizes is considerably less. The pipelining concept has made efficient use of resources available while executing map and reduce phases.

Table I- Execution time for various jobs

| Job Id | Execution Time (min:sec) | | | | | |
|--------|--------------------------|----------|--------------|----------|------------|----------|
| | Best Case | | Average Case | | Worst Case | |
| | Existing | Proposed | Existing | Proposed | Existing | Proposed |
| J1 | 1.12 | 0.58 | 1.22 | 1.08 | 1.31 | 1.25 |
| J2 | 1.29 | 1.05 | 1.32 | 1.21 | 1.38 | 1.32 |
| J3 | 1.41 | 1.18 | 1.51 | 1.38 | 1.58 | 1.51 |
| J4 | 2.01 | 1.42 | 2.09 | 1.55 | 2.12 | 2.07 |
| J5 | 2.25 | 2.04 | 2.31 | 2.11 | 2.42 | 2.37 |

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

Graphical representation:

Graphs are made between time calculated using existing and proposed methods for best, average and worst case. Fig. 3, 4, 5 shows the graphical representation in form of bar charts for best, average and worst case respectively. The best case is when system make highest utilization of resources only for map reduce task and average case occurs when it executes local map tasks and making them available for other resources also. However the worst case is likely to occur seldom.

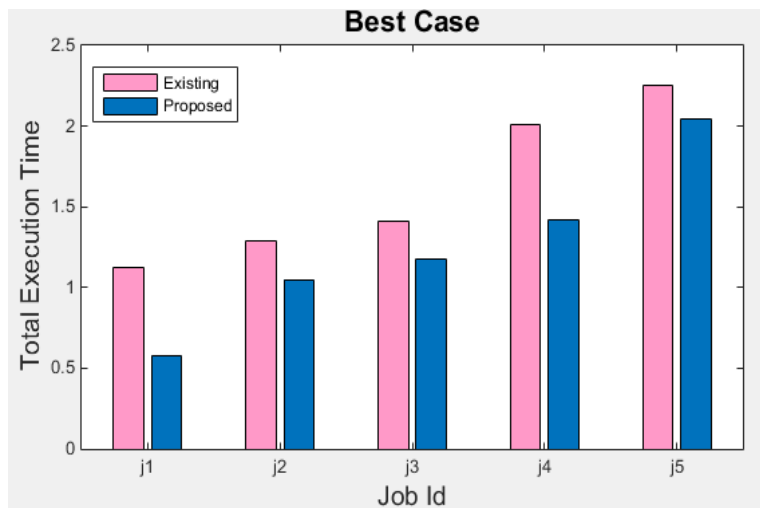


Fig. 3 Best Case

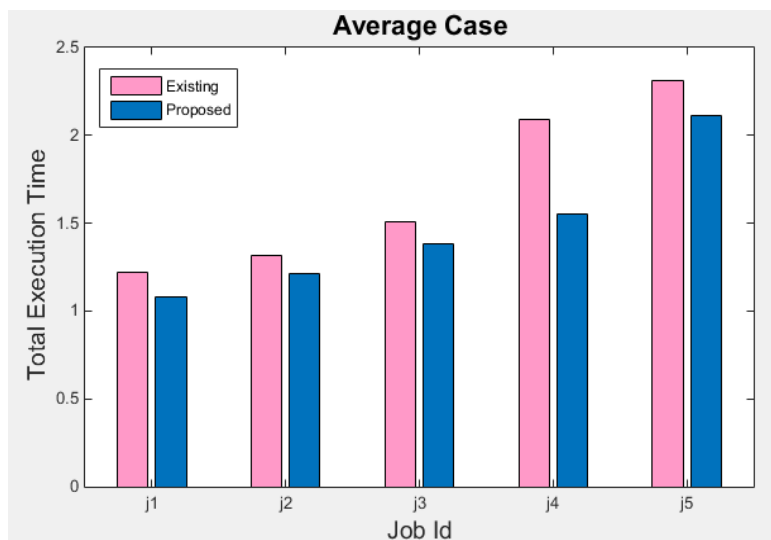


Fig. 4 Average Case

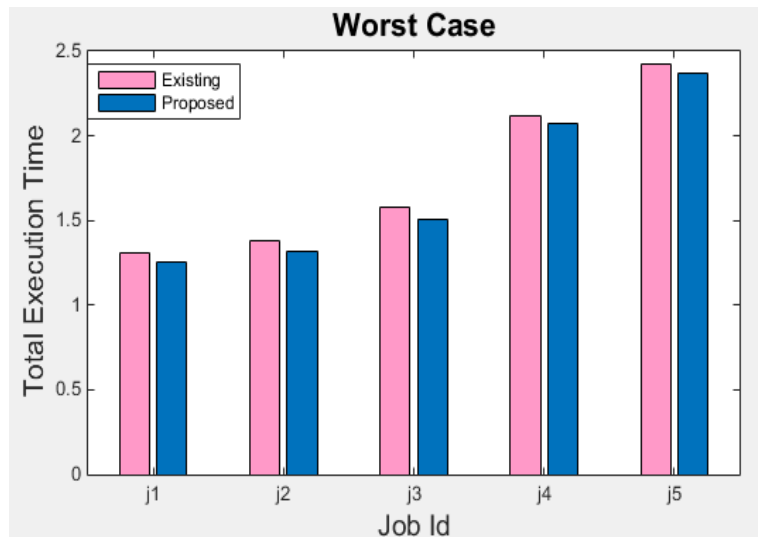


Fig. 5 Worst Case

The results shows execution time of jobs for proposed technique is less than existing approach. The execution time is reduced up to 30% to 60% from existing technique.

v. CONCLUSION AND FUTURE SCOPE

From the above discussion it is clear that huge amount of data are being stored everyday by various IT companies, which is a challenging task for them to make efficient use of system resources. The goal of above work is to reduce execution time of jobs and utilizing system resources in best possible manner. The concept of queue management and pipelining is introduced which help the jobs to run in queue manner and sharing resources among sub queues. The performance of proposed scheduler is compared with the exiting one which shows considerable decrease in average response time. The execution time is reduced up to 30% to 60% when proposed novel job scheduler is applied.

The proposed algorithm has been focused on criteria of job execution time and maximum utilization of system resources with complete fairness criteria. The jobs can be executed in less time thereby reducing the average waiting time also. In future, the performance can be evaluated on heterogeneous cluster setting and different benchmarks. The performance can be analyzed based on criteria of earliest deadline first.

REFERENCES

- [1] Namrata Singh and Sanjay Agrawal, "A review of research on MapReduce scheduling algorithms in Hadoop," in *Computing, Communication & Automation (ICCCA), 2015 International Conference on*, Noida, India, pp. 637-642, 2015.
- [2] Shankar Ganesh Manikandan and Siddharth Ravi, "Big Data Analysis Using Apache Hadoop," in *IT Convergence and Security (ICITCS), 2014 International Conference on*, Beijing, China, pp. 1-4, 2014.
- [3] Sreedhar C., N. Kasiviswanath, and P. Chenna Reddy, "A Survey on Big Data Management and Job Scheduling ," *International Journal of Computer Applications (0975 – 8887)* , vol. 130, no. 13, pp. 41-49, 2015.
- [4] Vasiliki Kalavri and Vladimir Vlassov, "MapReduce: Limitations, Optimizations and Open Issues," in *2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, Melbourne, Australia, pp. 1031-1038, 2013.
- [5] B. Thirumula Rao and Dr. L.S.S.Reddy, "Survey on Improved Scheduling in Hadoop MapReduce in Cloud Environments," *International Journal of Computer Applications (0975-8887)*, vol. 34, no. 9, pp. 29-33, 2011.
- [6] Chen He, Ying Lu, and David Swanson, "Matchmaking: A New MapReduce Scheduling Technique," in *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*, Athens, Greece, pp. 40-47, 2011.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 6, June 2016

- [7] Jiong Xie et al., "Improving MapReduce performance through data placement in heterogeneous Hadoop clusters," in *Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW)*, 2010 IEEE International Symposium on, Atlanta, USA, pp. 1-9, 2010.
- [8] M. Zaharia et al., "Delay Scheduling: A Simple Technique for Achieving Locality and Fairness in Cluster Scheduling," in *Proceedings of the 5th European conference on Computer systems*, ACM, New York, USA, pp. 265-278, 2010.
- [9] Dongjin Yoo and K.M. Sim, "A Comparative Review of Job Scheduling for MapReduce," in *2011 IEEE International Conference on Cloud Computing and Intelligence Systems*, Beijing, China, pp. 353-358, 2011.
- [10] Quan Chen, Daqiang Zhang, Minyi Guo, Qianni Deng, and Song Guo, "SAMR: A Self-adaptive MapReduce Scheduling Algorithm in Heterogeneous Environment," in *Computer and Information Technology (CIT)*, 2010 IEEE 10th International Conference on, Bradford, England, pp. 2736-2743, 2010.
- [11] Marwa Elteir, Heshan Lin, and Wu-chun Feng, "Enhancing MapReduce via Asynchronous Data Processing," in *Parallel and Distributed Systems (ICPADS)*, 2010 IEEE 16th International Conference on, Shanghai, China, pp. 397-405, 2010.
- [12] K.Kambatla, N. Rapolu, S. Jagannathan, and A. Grama, "Asynchronous Algorithms in MapReduce," in *2010 IEEE International Conference on Cluster Computing*, Heraklion, Greece, pp. 245-254, 2010.
- [13] Kamal Kc and Kemafor Anyanwu, "Scheduling Hadoop Jobs to Meet Deadlines," in *Cloud Computing Technology and Science (CloudCom)*, 2010 IEEE Second International Conference on, Indianapolis, USA, pp. 388-392, 2010.
- [14] Zhuo Tang, Junqing Zhou, Kenli Li, and Ruixuan Li, "MTSD: A Task Scheduling Algorithm for MapReduce Base on Deadline Constraints," in *Parallel and Distributed Processing Symposium Workshops & Phd Forum (IPDPSW)*, 2012 IEEE 26th International, Shanghai, China, pp. 2012-2018, 2012.
- [15] Phuong Nguyen, Tyler Simon, Milton Halem, David Chapman, and Quang Le, "A Hybrid Scheduling Algorithm for Data Intensive Workloads in a MapReduce Environment," in *Utility and Cloud Computing (UCC)*, 2012 IEEE Fifth International Conference on, Chicago, USA, pp. 161-167, 2012.
- [16] J. Polo, D. Carrera, Y. Becerra, M. Steinder, and I. Whalley, "Performance-driven task co-scheduling for MapReduce environments," in *2010 IEEE Network Operations and Management Symposium - NOMS 2010*, Osaka, Japan, pp. 373-380, 2010.
- [17] K. Arun Kumar, V. K. Konishetty, K.Voruganti, and G. V. Prabhakara Rao, "CASH: context aware scheduler for Hadoop," in *Proceedings of the International Conference on Advances in Computing, Communications and Informatics*, New York, USA, pp. 52-61, 2012.
- [18] Supriya Pati and Mayuri A. Mehta, "Job Aware Scheduling in Hadoop for Heterogeneous Cluster," in *Advance Computing Conference (IACC)*, 2015 IEEE International, Bengaluru, India, pp. 778-783, 2015.