



A Relative Study on Task Schedulers in Hadoop MapReduce

Jisha S Manjaly

Department of Computer Science and Engineering
MG university, India.

Chinnu Edwin A

Department of Computer Science and Engineering
MG University, India.

Abstract— Hadoop is a framework for BigData processing in distributed applications. Hadoop cluster is built for running data intensive distributed applications. Hadoop distributed file system is the primary storage area for BigData. MapReduce is a model to aggregate tasks of a job. Task assignment is possible by schedulers. Schedulers guarantee the fair allocation of resources among users. When a user submits a job, it will move to a job queue. From the job queue, job will be divided into tasks and distributed to different nodes. By the proper assignment of tasks, job completion time will reduce. This can ensure better performance of the job. In this paper we study different default and extended task schedulers in Hadoop MapReduce.

Keywords— Hadoop, HDFS, MapReduce, Scheduling, FIFO, Fair, Capacity

I. INTRODUCTION

Apache Hadoop [17] is framework for distributed applications. Doug Cutting is inspired by Google MapReduce and he introduced Hadoop MapReduce for distributed applications. Apache Hadoop mainly deals with BigData processing on distributed environment. Hadoop distributed file system (HDFS) is the storage area in this framework. MapReduce is the technique for BigData processing and analyzing in parallel.

II. LITERATURE SURVEY

Hadoop framework [13][14] is an open source development by Google for distributed application. Many computer nodes are worked in parallel for processing BigData. For cluster setup Hadoop uses more reasonably priced computer machines. Hadoop is a simplified programming model for efficient and automatic data distribution in cluster nodes. HDFS[13] is the file system that supports Hadoop framework for BigData storage. The management of BigData is a challenging issue. The architecture consists of a single Namenode and many slave Datanodes. In the storage area disk has a block size. Normal block size in HDFS is 64MB. By default there are three replica of each block in different Datanodes. When a Datanode fails, the replicate block will be available in other Datanodes and hence hardware failure can be resolved. HDFS provides high throughput and fault tolerance for application that uses large datasets. MapReduce [13] follows key-value pair format for data processing. MapReduce has two phases called Map and Reduce. The output of Mapper will move to Reducer as input. A Combiner combines output from the Mapper. If more than one Reducer is present in the system, then partitioner is used for splitting the output. Mapper maps the input and the Reducer aggregates the result of Mapper. Both structured and unstructured data can be analyzed by MapReduce. Hadoop MapReduce jobs are assigned from the JobTracker to TaskTrackers. Large number of nodes works in parallel in order to produce aggregate result. The intermediate tasks and the Map output will be stored in the disk rather than HDFS.

In Hadoop, schedulers are playing a vital role in job assignment. FIFO [15] [16], Fair [15] [16] and Capacity schedulers [15] [16] are the default schedulers in Hadoop and later, many enhanced schedulers were developed. In a distributed environment, task scheduling is more effective because completion of job execution becomes faster and it results in improved performance. For big data analysis, scheduling is worthy while for small data processing, scheduling is an added hindrance.

III. SCHEDULERS IN HADOOP MAPREDUCE

Task assigning is one of the important processes in Hadoop. Schedulers are responsible for doing task assignment. Two types of nodes called JobTracker and TaskTracker are taking part in the job execution process. JobTracker acts as a Coordinator of all jobs executing in the machine while the TaskTracker executes the tasks and sends a continuous report to the JobTracker.

A. Default FIFO Scheduler

FIFO scheduler is implemented by Google. It is the default scheduler for almost all Hadoop applications. The Job queue is processed in First in First out fashion. With the scheduler, the main drawback is that only after finishing the previous job, next jobs in the job queue will be assigned. The scheduler implementation is simple and efficient.

B. Fair Scheduler

Fair scheduler [15] [16] is introduced by Facebook. Here, fair sharing of resources is possible. Main advantage of the scheduler is that whenever slot becomes free, shorter jobs can be assigned. Unlike FIFO scheduler, the smaller

jobs need not wait in order to complete a big job. Main drawback is that tasks will be allocated to all the slots in the cluster with maximum slot capacity. In fair scheduling, pool of jobs will be there. Fair scheduler follows three concepts. Those are

1. Jobs placed in a pool of jobs
2. Each pool is having a guaranteed capacity
3. Fair sharing of resources

C. Capacity Scheduler

Capacity scheduler [15] [16] is introduced by Yahoo. Capacity scheduler is very effective for large scale applications. Other than job pool, here multiple job queues are allocated. There is a guaranteed capacity for each queue. When the queue capacity exceeds the job will be allocated to other queues. Here, higher priority jobs can access resources faster than lower priority jobs. In the job queue, Capacity scheduler follows FIFO scheduling with priority. But there is a limit on percentage of running tasks per user. So that users can share the cluster setup equally.

D. Longest Approximate time to End (LATE) Scheduler

Longest Approximate Time to end scheduler [10] is mainly focusing on speculative execution of tasks. In speculative execution, when a task execution is performing slowing, on the back end an equivalent task is also running. If that back end task complete fastly performance will be improved. Due to speculative execution response time will be improved. In a heterogeneous environment LATE scheduler is robust. Main features of LATE Scheduler are following :

1. Assigning priorities for the tasks for Speculative Execution.
2. Select the fastest nodes for Speculative execution.
3. Plugging the speculative tasks in order to avoid thrashing.

E. Delay Scheduler

Delay scheduler [1] is an extended version of Fair scheduler. Delay scheduler is introduced to improve data locality and fairness of Fair scheduler. Before a task is assigned the data needed for the task will be checked in the slot. If data is available then the task is assigned to the slot. Otherwise the task assignment is delayed in order to get a free slot for the data needed for the particular task. This method is better in terms of performance than the allocation of tasks in the TaskTracker.

F. Dynamic Priority Scheduler

Thomas Sandholm proposed dynamic priority scheduler[3]. The scheduler considers priority of users. Capacity distribution is dynamically performed. This is an extension of FIFO and Fair schedulers. The performance of the scheduler is better in large clusters. Fairness and improved priority of jobs in the queue are its advantages when compared to the default schedulers in Hadoop. This scheduler has been improving the JobTracker service. There are two phases in this scheduler.

1. Dynamic Priority Allocator- It permits slot allocation dynamically.
2. Priority Enforcer-Enforce Resource sharing.

G. Resource Aware Scheduler

Resource aware scheduler [2] [4] focuses on resource utilization such as CPU utilization, IO utilization, Disk Utilization and Network utilization. Suppose a particular node in the cluster is having less CPU utilization and another node in the same cluster is having IO utilization. At this time a job comes in the job queue that needs less CPU utilization .The better way is assigning that job to the node that focuses on less CPU utilization when compared to other nodes. The resource utilization is not bothered in default schedulers on Hadoop. Main aim of Resource aware scheduler is Job performance management.

H. Deadline Constraint Scheduler

Deadline Constraint Scheduler [11] [12] for Hadoop ensures the jobs whose deadlines can be met are scheduled for execution. When a job is submitted a deadline test is performed. The test checks if the job is finished within a specified deadline. If the job finishes within the specified deadline, it can schedule. Otherwise it will fail. The scheduler can be used in the applications where threshold value is important. In the scheduler, jobs are allocated on demand. The scheduler is used for larger clusters.

I. Learning Scheduler

In learning Scheduler [5] [6], jobs are classified as good or bad. Pattern classifier classifies the jobs. According to the resource utilization, good jobs will be considering for further processing. Good job does not make any overload to the TaskTrackers. Bad jobs will be rejected. The scheduler considers CPU utilization, Memory utilization, IO utilization and Network utilization. If more than one good job arrive in the job queue, job will be selected by the more expected utility function. Task assignment is similar to default schedulers.

J. COSHH

COSHH [7] is an acronym for Classification and Optimization based Scheduler for Heterogeneous Hadoop Systems. Both cluster and application level heterogeneity is considered here. COSHH addresses fairness and minimum share requirements. By the consideration, better tasks assignment is done in heterogeneous [8] environment. The scheduler takes minimum job completion time when compared to other schedulers. COSHH uses optimization techniques to minimize searching overhead. In COSHH, data will be replicated on the suggested resources and hence, data locality problem can be reduced.

K. Network Aware Scheduler

Data locality is the main problem for task assigning in many schedulers. Network Aware scheduler [9] provides an administrator control script to find the location of the node that has the data needed for the task. Up to a limit, data

locality problem can be removed in this scheduler. Here, FIFO and Fair scheduler is extended with network awareness. When a task is assigned to the TaskTracker, the data needed for executing the task will be checked. If it is available, then the task is assigned to the slot. Otherwise the scheduling of the task is delayed for specific delay.

IV. COMPARISON

The tables 1.A and 1.B below give the comparative study of different schedulers.

TABLE 1.A
COMPARISON OF SCHEDULERS

Name of the Scheduler	Preemption	Priority in job queue	Fairness/ Fair sharing of Resources	Working	Homogeneous/ Heterogeneous
Default FIFO Scheduler	NO	No by Default	NO	Better work with small clusters	Homogeneous
Fair Scheduler	YES	YES	Fair share of the cluster capacity over time	Better work with small clusters	Homogeneous
Capacity Scheduler	No Preemption when job fails	No by Default	YES	Better work with large clusters	Homogeneous
LATE Scheduler	YES	YES	YES	Better work with large clusters	Both
Delay Scheduler	YES	YES	Less Fairness than Fair Scheduler	Better work with small clusters	Homogeneous
Dynamic Priority Scheduler	YES	YES	YES	Better work with large clusters	Homogeneous
Resource Aware Scheduler	YES	YES	YES	Better work with large clusters	Both
Deadline Constraint Scheduler	YES	YES	YES	Better work with large clusters	Both
Learning Scheduler	YES	YES	YES	Better work with small clusters	Both
COSHH	YES	YES	YES	Better work with large clusters	Heterogeneous
Network Aware Scheduler	YES	YES	YES	Better work with large clusters	Homogeneous

TABLE 1.B
COMPARISON OF SCHEDULERS

Name of the Scheduler	Speculative Execution	Head of line problem	Sticky Slots	Locality Problem	Job Allocation	Implemented
Default FIFO Scheduler	NA	NA	NA	YES	Statically	YES
Fair Scheduler	NA	YES	YES	YES for small jobs	Statically	YES
Capacity Scheduler	NA	NA	NA	YES	Statically	YES
LATE Scheduler	YES	NA	NA	YES	Statically	YES
Delay Scheduler	NA	NA	NA	Improved compared to Fair scheduler	Statically	YES
Dynamic Priority Scheduler	NA	NA	NA	NO	Dynamically	YES
Resource Aware Scheduler	NA	NA	NA	YES	Both	YES
Deadline Constraint Scheduler	NA	NA	NA	YES for small jobs	Dynamically	Not in real time
Learning Scheduler	Enabled	NA	NA	YES	Both	YES
COSHH	NA	YES	YES	NO	Dynamically	Not in real time
Network Aware Scheduler	NA	NA	NA	NO	Dynamically	YES

V. CONCLUSIONS

The paper gives an overall idea about different task schedulers in Hadoop MapReduce. It compares the properties of various task schedulers.

REFERENCES

- [1] Matei Zaharia , Dhruva Borthakur, and Joydeep Sen Sarma, “Delay Scheduling: A Simple Technique for Achieving Locality and Fairness in Cluster Scheduling” , *Yahoo! Research , University of California*, Berkeley , 2009.
- [2] Mark Yong, Nitin Garegrat, Shiwali Mohan, “Towards a Resource Aware Scheduler in Hadoop”, December 21, 2009.
- [3] Thomas Sandholm, Kevin Lai, “Dynamic Proportional Share Scheduling in Hadoop ”, in *Social Computing Lab, Hewlett-Packard Labs, Palo Alto*, CA 94304, USA, 2010.
- [4] Jord Polo, Claris Castillo, David Carrera , Yolanda Becerra , Ian Whalley , Malgorzata Steinder , Jordi Torres , and Eduard Ayguad , “Resource-aware Adaptive Scheduling for MapReduce Clusters ”, in *Barcelona Supercomputing Center (BSC) and Technical University of Catalonia(UPC)*, December 2011.
- [5] Jaideep Datta Dhok , “Learning Based Admission Control and Task Assignment in MapReduce ”, *Search and Information Extraction Lab , International Institute of Information Technology Hyderabad - 500 032, INDIA*, June 2010 .
- [6] Jaideep Dhok , Vasudeva Varma , “Using Pattern Classification for Task Assignment in MapReduce ” , in *Search and Information Extraction Lab International Institute of Information Technology, Hyderabad, India* .
- [7] Aysan Rasoolia , Douglas G. Downa , “COSHH: A Classification and Optimization based Scheduler for Heterogeneous Hadoop Systems ” , in *Department of Computing and Software, McMaster University, L8S 4K1, Canada*.
- [8] Aysan Rasooli , Douglas G. Down , “A Hybrid Scheduling approach for Scalable Heterogeneous Hadoop Systems ”, in *Department of Computing and Software McMaster University, 2012*.
- [9] Praveen kumar Kondikoppa , Chui-Hui Chiu, Cheng Cui and Lin Xue and Seung-Jong Park , “Network-Aware Task Scheduling of MapReduce Framework over Multi-Clouds and High Speed Networks ”, in *Department of Computer Science, Center for Computation & Technology, Louisiana State University, LA, USA, 2012*.
- [10] Matei Zaharia, Andy Konwinski, Anthony D. Joseph, Randy Katz, Ion Stoica, “Improving MapReduce Performance in Heterogeneous Environments”, in *University of California, Berkeley, 2012*.
- [11] Ashay Raut, Dhaval Hansaliya, Rasesh Mori, ASVS Teja , “Deadline aware scheduler for Hadoop Yarn” in *Cloud Computing (cse 565), Nov’12*
- [12] Kamal Kc, Kemafor Anyanwu , “Scheduling Hadoop Jobs to Meet Deadlines” in *Cloud Computing Technology and Science (CloudCom), IEEE Second International Conference on Nov. 30 2010-Dec. 3 2010*
- [13] Tom white, Hadoop Definitive Guide, Third Edition, 2012
- [14] Hadoop Tutorial: <http://developer.yahoo.com/hadoop/tutorial/module1.html>
- [15] Hadoop job scheduling: <http://blog.cloudera.com/blog/2008/11/job-scheduling-in-hadoop/>
- [16] Hadoop Scheduling: <http://www.ibm.com/developerworks/library/os-hadoop-scheduling/>
- [17] Apache Hadoop: <http://Hadoop.apache.org>