**Imperial Journal of Interdisciplinary Research (IJIR)**
Vol.2, Issue-1 , 2016
ISSN : 2454-1362 , www.onlinejournal.in

# Task Scheduling in Hadoop

**Sagar Mamdapure**
*SAE,Kondhwa*

**Munira Ginwala**
*SAE,Kondhwa*

**Neha Papat**
*SAE,Kondhwa*

## Abstract

*Hadoop is widely used for storing large datasets and processing them efficiently under distributed computing environment. It consists of Hadoop Distributed File System (HDFS) for storing large data and MapReduce to process it. It is efficient when low response time is required. It is mainly used to handle data intensive applications. It is gaining lot of popularity now-a-days. Aim of Hadoop is to provide parallelize job execution across multiple nodes. Hence many scheduling algorithms have been proposed in the past. Locality, synchronization and fairness are the three important scheduling issues in MapReduce. The common aim of scheduling algorithms is to reduce the execution time of a parallel applications and also to solve these issues*

***Keywords***- *Hadoop, Task scheduling, Fair scheduler, Capacity scheduler*

.

## 1. Introduction

MapReduce is a distributed computing model. Map and Reduce are two important functions of Hadoop. Map accepts set of input and converts it into intermediate key-value pair. Each mapper loads the set of files local to that machine and processes them. MapReduce library will combine all the values associated with the same key and transfers these key-value pairs to Reduce function. This is the only communication step in MapReduce. Reduce function will convert these values to smaller set of values. Usually just zero or one output value is produced per Reduce operation. The intermediate values are provided to the user's reduce function via an iterator which allows us to handle lists of values that are too large to fit in memory. Individual map tasks do not share information with each other. They are unaware about existence of other mappers. The user never perform any operation for moving data from one machine to other. It is handled by the MapReduce platform itself under guidance of different keys and values.

Hadoop has the capacity of configuring the jobs, submitting them and controlling their execution. MapReduce takes care of failures. If particular node fails, then required data can be taken from other node.
Task scheduling in Hadoop involves allocating 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.

It consists of following three phases:

1. **Map**: JobTracker will divide the map tasks and Reduce tasks to idle TaskTrackers. Map function will scan the input data and performs map operation on that data. A set of intermediate key- value pairs will be generated during phase. These results will be written on disk.

2. **Partition & Shuffle**: After completion of first stage of mapping, generated key-value pairs need to be transferred to Reduce function. This process of moving intermediate key-value pairs to the reducers is known as shuffling. Each map task can send the results to any partition. But all values for the same key are always reduced together regardless of from which they are generated.

3. **Reduce**: Reduce function will accept intermediate key-value pairs and perform reduce operation. It will write the final results to output file. These output files written by the Reducers are then kept in HDFS for user or performing other MapReduce tasks.

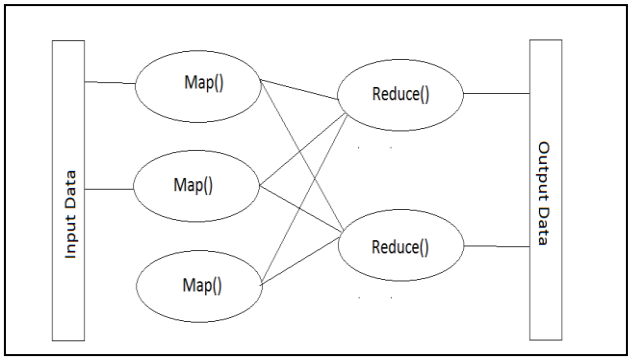


Figure 1. Scheduling

There are three important scheduling issues in MapReduce:

**Locality:**
It is the vital issue that affects on performance in shared environment. During this time, scheduler assigns MapReduce tasks to available slots. These are the slots where the underlying storage layer holds the input intended for processing.

**Synchronization**
It is the process of transferring the intermediate output of the map processes to the reduce processes as input. Generally, reduce process will start at a time when Map Process is completely finished. Due to this dependency, a single node will affect on the entire process. It will cause the other nodes to wait until it is finished. Hence, Synchronization of these two phases is essential to enhance overall performance of MapReduce Model.

**Fairness**
Utilization of the shared clusters may get hampered due to a map reduce job with a heavy workload. Hence the shorter jobs get starved and do not get their desired share to resources. Assigning appropriate amount of load to each job sharing cluster should be considered since the demands of workload can vary.

## 2. Schedulers in Hadoop

### 2.1. Fair scheduler:

Fair scheduling is a scheduling algorithm that assigns jobs such that all jobs fairly gets an equal share of resources over time. When only one job is submitted, that job utilizes the entire cluster but when other jobs are submitted, available tasks slots are assigned to the new jobs therefore each job gets the access to the resources and utilizes the same amount of CPU time. Fair scheduler allows short jobs finish in optimal time while not starving long jobs. It is also an easy way to share a cluster between multiple of users. Apart from assigning equal resources to the jobs,fair scheduling also work with job priorities – according to the priorities weights are assigned to determine the fraction of total compute time that each job gets. The fair scheduler applies a technique to organize jobs into pools, and distributes resources optimally between these pools. Automatically, each user gets a separate tool and an equal share of that pool. Another alternative is to set a job's pool based on the user's Unix group or any jobconf property. Among all the pools, jobs can be scheduled by FIFO scheduling as well as fair scheduling. The Fair Scheduler allows fair sharing as well as it also assigns guaranteed minimum shares to pools, which ensures that certain groups or users always get minimum required resources. A pool containing job gets its minimum share of resources but when the pool does not need its assigned share, the excess of the resources is divided between other pools.
If any pool is not able to get its minimum share for some period of time, fair scheduler can also support preemption of jobs in other pools. The pool can terminate tasks from other pools to get free room to run. Preemption is used to assure that "production" jobs are not starved while also letting the Hadoop cluster deployment for experimental and research jobs. Also, a pool can be allowed to preempt tasks if half of its fair share for a configurable timeout (generally set larger than the minimum share preemption timeout) is not met. When choosing tasks to terminate, the fair scheduler chooses the most-currently-launched tasks from over-allocated jobs, to minimize wastage in computation. Preemption should not be confused as it does not let the preempted jobs to fail, because Hadoop jobs is capable of tolerating lost tasks; but only the tasks will take longer to finish. The Fair Scheduler can put a certain extent to the number of concurrently running jobs per user and per pool. This comes handy when a user have to submit hundreds of jobs at once, or for ensuring that intermediate data does not fill up disk space on a cluster when too many concurrent jobs are running.

### 2.2. Capacity scheduler:

The Capacity Scheduler is a scheduling algorithm designed to run Hadoop Map-Reduce as a shared cluster in an operator-friendly manner and thus maximizing the throughput and the utilization of the cluster while running Map Reduce applications. Usually all organization have their own set of resources whose capacity is sufficient to meet the organization's SLA under peak or near peak conditions. This generally gives way to poor average utilization and the overhead of taking care of and controlling multiple independent clusters, one for every organization. Sharing clusters between organizations is an efficient yet cheap manner of running large Hadoop installations since this allows them to take benefits of economies of scale without creating private clusters. However, organizations are concerned about the way to share a cluster because they are sensitive about other organizations using the resources that are important for their SLAs. The Capacity Scheduler is designed in such a way that it allows sharing a large cluster while imparting each organization a minimum share of the total capacity. The main idea behind capacity scheduling is that the available resources in the Hadoop Map-Reduce cluster is divided in such a way that multiple organizations who collectively raise the cluster based on their needs. They are also provided with the option that an organization can take charge of any excess capacity not being used by others. This creates flexibility for the organizations in an efficient manner.

Sharing clusters across organizations creates the need of solid support for multi-tenancy as every organization must have sufficient capacity and safe-guards to make sure the shared cluster facilitates to single rouge job or user. The Capacity Scheduler provides a well-defined set of limits to make sure that a single job or user or queue must not take more than necessary amount of resources in the cluster. Also, the JobTracker of the cluster, especially, is a vital resource and the Capacity Scheduler puts extent on tasks and jobs that are initialized or are in queue from a single user to make sure that fairness and stability of the cluster can be achieved. The Capacity Scheduler has the following advantages: Capacity, Guarantee, Security, Elasticity, Multi-tenancy, Operability, Job Priorities etc.

The disadvantage of capacity scheduler is that it is very complex to implement.

## 3. Literature Survey

To get better assignment of tasks and load balancing, the MapReduce work mode as well as task scheduling algorithm of Hadoop platform is analyzed. This paper introduces the idea of weighted round-robin scheduling algorithm into the task scheduling of Hadoop and ballyhoo the weight update rules through analyzing all the situations of weight update. By experiment it is clear that it is effective in making task allocation and achieving good balance when it is applied into the Hadoop platform which uses JobTracker scheduling strategy.[1]

This paper discusses Cloud Computing is rising as a reinstatement machine paradigm shift. Hadoop MapReduce has become a robust Computation Model for massive process apprehension on distributed commodity hardware clusters like Clouds. During this paper they have studied numerous scheduling improvement in a scheduling techniques, a brand new scheduling algorithm with Hadoop like Fair4s scheduling algorithm with its extended purpose allows processing large as well small jobs with effectual fairness without starvation of small jobs.[2]

This paper throws light on job scheduling algorithms for Hadoop and proposed an algorithm by using Bayes Classification which optimizes previous job scheduling algorithms. The proposed algorithms is summarized in the following sections. Those scheduling algorithms which are based on Bayes Classification, in that algorithms the jobs in job queue will be classified into two types i.e. bad job and good job by using Bayes Classification. The JobTracker select a good job from job queue when it gets task request and select tasks from good job queue to allocate JobTracker, then the execution will feedback to the JobTracker. Hence the algorithms based on Bayes Classification influence the job classification by learning the result of feedback. The JobTracker will select the most appropriate job to execute on TaskTracker. [3]

The most popular implementations for private clouds are the Hadoop based clusters. Because workload traces are not available publicly, different cloud solutions with easily available benchmarks compares and evaluates previous work. In this paper, we have used a recently released Cloud benchmarks suite CloudRank-D to quantitatively evaluate five different Hadoop task schedulers including FIFO (First In First Out), capacity, naïve fair sharing, fair sharing with delay and Hadoop On Demand (HOD) scheduling. The experiments show

that with a good scheduler, the performance of a private cloud can be improved by 20%. [4]

Scalability is the feature of Cloud which has improved its use in business and other applications. As Hadoop is an efficient solution for handling big data, it is adopted by most of the cloud providers. It has the capability of achieving desired performance level. Scheduling large number of tasks is really a great challenge and heterogeneous nature deployed Hadoop systems makes it more difficult. This paper analyzes the performance of different Hadoop schedulers like FIFO and Fair sharing and compares them with the COSHH scheduler (Classification and Optimization based Scheduler for Heterogeneous Hadoop), which has been developed by the authors Aysan Rasooli and Douglas Down. [5]

Hadoop provides default FIFO scheduler which schedules job in first in first out order. But this technique may not be useful for certain tasks. Hence in such situation alternate scheduler needs to be chosen. In this paper they have conducted a study on various schedulers that will help to understand the details of particular schedulers. It would be useful to make best choice of scheduler according to our need. [6]

MapReduce is emerging as an essential programming model for various fields such as data mining, simulation and indexing. Hadoop is an open source implementation of Hadoop which is mainly used when low response time is required .Hadoop's performance is related to its task scheduler which assumes that the cluster nodes are homogeneous and tasks make improvement linearly. It uses these assumptions to decide when to re-execute tasks that appear to be stragglers. In reality, the homogeneity assumptions do not always hold. An especially compelling setting where this occurs is a virtualized data center for example Amazon's Elastic Compute Cloud (EC2). This paper shows that Hadoop's scheduler can cause performance reduction in heterogeneous environments. They design a new scheduling algorithm called as Longest Approximate Time to End (LATE). This algorithm is highly robust to heterogeneity. LATE can improve Hadoop's response time by factor of 2 in clusters of 200 virtual machines in Elastic Compute Cloud. [7]

This paper proposes an algorithm which 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. It brings up machine learning based solutions to their approach and try to maintain a resource balance on the cluster by not overwhelming any of the nodes, thereby reducing the overall run time of the jobs. When compared to Yahoo's Capacity scheduler, it saves run time of around 21% in the case of heuristic based approach and around 27% in case of machine learning based approach. [8]

## 4. Conclusion

In this paper we have done a survey on task scheduling in Hadoop platform. We have also studied various schedulers like Fair and Capacity scheduler. These are some of the scheduling algorithms which are used for task scheduling. We observed that by combining these algorithms, scheduling process could be more easy and efficient.

## 5. References

[1] Jilan Chen, Dan Wang and Wenbing Zhao, "A Task Scheduling Algorithm for Hadoop Platform", JOURNAL OF COMPUTERS, VOL. 8, NO. 4, APRIL 2013

[2] Mr.A.U.Patil , Mr T.I Bagban, Mr.A.P.Pande , "Recent Job Scheduling Algorithms in Hadoop Cluster Environments: A Survey ", International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 2, February 2015

[3] Yingjie Guo, Linzhi Wu, Wei Yu, Bin Wu, Xiaotian Wang, "The Improved Job Scheduling Algorithm of Hadoop Platform"

[4] Shengyuan Liu, Jungang Xu, Zongzhen Liu, Xu Liu, "Evaluating Task Scheduling in Hadoop-based Cloud Systems", 2013 IEEE International Conference on Big Data

[5] Aysan Rasooli and Douglas G. Down, "A Hybrid Scheduling Approach for Scalable Heterogeneous Hadoop Systems"

[6] B.Thirumala Rao,Dr. L.S.S.Reddy, "Survey on Improved Scheduling in Hadoop:MapReduce in Cloud Environments", International Journal of Computer Applications (0975 – 8887)Volume 34– No.9, November 2011

[8] Radheshyam Nanduri, Nitesh Maheshwari, Reddyraja. A and Vasudeva Varma, "Job Aware Scheduling Algorithm for MapReduce Framework", 2011 Third IEEE International Conference on Coud Computing Technology and Science

.