IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 15, Issue 1 (Sep. - Oct. 2013), PP 46-50
www.iosrjournals.org

Survey on Job Schedulers in Hadoop Cluster

Bincy P Andrews[1], Binu A[2]
[1](*Rajagiri School of Engineering and Technology*, Kochi)
[2] *(Rajagiri School of Engineering and Technology*, Kochi*)*

Abstract : *Hadoop-MapReduce is a powerful parallel processing technique large for big data analysis on distributed commodity hardware clusters such as Clouds. For job scheduling Hadoop provides default FIFO scheduler where jobs are scheduled in FIFO order. But this scheduler might not be a good choice for some jobs. So in such situations one should choose an alternate scheduler. In this paper we conduct a study on various schedulers. Hopefully this study will stimulate other designers and experienced end users understand the details of particular schedulers, enabling them to make the best choices for their particular research interests.*
Keywords: *Cloud Computing, Hadoop, HDFS, MapReduce, schedulers*

I. INTRODUCTION

Hadoop-MapReduce is a powerful parallel processing technique large for big data analysis on distributed commodity hardware clusters such as Clouds. For job scheduling Hadoop provides default FIFO scheduler where jobs are scheduled in FIFO order. But this scheduler might not be a good choice for some jobs. So in such situations one should choose an alternate scheduler. In this paper we conduct a study on various schedulers. Later on the ability to set the priority of a Job was added. Facebook and Yahoo contributed significant work in developing schedulers i.e. Fair Scheduler and Capacity Scheduler respectively which subsequently released to Hadoop Community. Following chapter is organized as follows chapter2 provides details on various schedulers followed by concluding remarks.

II. JOB SCHEDULERS IN HADOOP

Different schedulers are as follows:

2.1 FIFO scheduler

FIFO stands for first in first out. In FIFO scheduling, Job Tracker pulls oldest job first from job queue. It doesn't consider about priority or size of the job. Hadoop's built-in scheduler runs jobs in FIFO order. Scheduler scans through jobs in order of priority when a task slot becomes free. Then it picks the map task in the job with data closest to the slave. The disadvantage of FIFO scheduling is poor response times for short jobs compared to large jobs. Possible solution is Hadoop on Demand (HOD). It uses the Torque resource manager for node allocation based on the needs of the virtual cluster. Main advantage of HOD is that it automatically prepares configuration files, and then initializes the system based on the nodes within the virtual cluster. Following are some of its advantages

- The HOD virtual cluster can be used in a relatively independent way.
- It is also adaptive in that it can shrink when the workload changes.
- automatically de-allocates nodes from the virtual cluster when it has no running jobs
- with less sharing of the nodes, there is greater security
- Improved performance because of a lack of contention within the nodes for multiple users' jobs.
- Poor Locality
- Poor Utilization

2.2 Fair scheduler

Fair scheduling is a method of assigning resources to jobs such that all jobs get, on average, an equal share of resources over time. If there is a single job running, the job uses the entire cluster. When other jobs are submitted, free task slots are assigned to the new jobs, so that each job gets roughly the same amount of CPU time. It lets short jobs complete within a reasonable time while not starving long jobs. The scheduler actually organizes jobs by resource pool, and shares resources fairly between these pools. By default, there is a separate pool for each user. There are the limits of concurrently running Map and Reduce tasks on "Task Tracker" of node. The fair scheduler can limit the number of concurrently running jobs from each user and from each pool. Running job

Table 2.1 summarizes details of FIFO scheduler However, HOD has two disadvantages

Definition	▪ First in first out
Algorithm	▪ Job Tracker pulls jobs from a work queue, oldest job first.
Advantages	▪ Simple to implement and efficient
Disadvantages	▪ has no concept of the priority or size of the job

Table 2.1 FIFO scheduler

limits are implemented by marking jobs as not runnable if there are too many jobs submitted by the same user or pool. When there are idle task slots, the scheduler processes in two steps, at first idle task slots are allocated between the job pools and secondly are allocated between jobs in the job pools. The main idea behind the fair share scheduler is to assign resources to jobs such that on average over time, each job gets an equal share of the available resources. Thus jobs that require less time are able to access the CPU and finish simultaneously with the execution of jobs that require more time to execute. Owing to this behavior interactivity among Hadoop jobs occurs and provides greater responsiveness of the Hadoop cluster to the variety of job types submitted. The fair scheduler was developed by Facebook.

Definition	▪ Fair scheduling is a method of assigning resources to jobs such that all jobs get, on average, an equal share of resources over time
Algorithm	▪ When there is an available slot open, the scheduler will arrange this slot to the job which has the largest job Deficit.
Advantages	▪ less complex ▪ works well when both small and large clusters ▪ it can provide fast response times for small jobs mixed with larger jobs
Disadvantages	▪ does not consider the job weight of each node

Table 2.2 Fair scheduler

Following are its features:

- The fair scheduler organizes jobs into *pools*, and divides resources fairly between these pools. Each pool can have a "guaranteed capacity" that is specified through a config file
- Excess capacity that is not going toward a pool's minimum is allocated between jobs using fair sharing. Introduction
- Within each pool, jobs can be scheduled using either fair sharing or first-in-first-out (FIFO) scheduling.
- the Fair Scheduler allows assigning guaranteed minimum shares to pools
- The Fair Scheduler can limit the number of concurrent running jobs per user and per pool. Fair Scheduler can limit the number of concurrent running tasks per pool.

2.3 Capacity scheduler

In Capacity scheduling, there are some queues. Each queue has its own assigned resources and uses FIFO strategy in itself. In order to prevent some user take too much of resources in one queue, the scheduler can limit the resources for the jobs from each user. When scheduling, all queues are monitored, if a queue does not use its allocated capacity, the spare capacity will be assigned to other queues. Jobs with a higher priority can access to resources sooner than lower priority jobs. We can configure the capacity scheduler within multiple Hadoop configuration files. This Capacity scheduler was developed by Yahoo! The Capacity Scheduler supports the following features:

- Hierarchical Queues
- Capacity Guarantees
- Security
- Elasticity
- Multi-tenancy
- Operability
- Resource-based Scheduling
- Support for multiple queues, where a job is submitted to a queue.
- Queues are allocated a fraction of the capacity of the grid in the sense that a certain capacity of resources will be at their disposal.
- Free resources can be allocated to any queue beyond its capacity.
- Queues optionally support job priorities (disabled by default).

- Within a queue, jobs with higher priority will have access to the queue's resources before jobs with lower priority
- In order to prevent one or more users from monopolizing its resources, each queue enforces a limit on the percentage of resources allocated to a user at any given time
- Support for memory-intensive jobs

In short The Capacity Scheduler works according to the following principles:

- The existing configuration is read from capacity
- An initialization poller thread is started and worker threads are also initialized.
- When a job is submitted to the cluster, the scheduler checks for job submission limits to determine if the job can be accepted or not.
- If the job can be accepted, the initialization poller checks the job against initialization limits

Whenever the job-tracker gets the heartbeat from a task tracker, a queue is selected from all the queues.
Both capacity scheduler & fair scheduler have following features in common:

- Both of them support multiple queues or pools, which mean both of them, are good for multiple users who share a Hadoop cluster.
- Each queue or pool supports the mode of FIFO or the mode of different priorities.
- Both of them can share spare slots to other queues or pools.

Following are its differences:

- Different strategies
- Different memory constraints

2.4 Longest approximate time to end

It tries to detect a slow running task to launch another equivalent task as a backup in other words it does speculative execution of tasks. But it does not ensure reliability of jobs. If bugs cause a task to hang or slow down then speculative execution is not a solution. Speculative execution relies on certain assumptions:

- Uniform Task progress on nodes
- Uniform computation at all nodes.

2.5 Delay scheduling

In delay scheduling when a node requests a task, if the head-of-line job cannot launch a local task, we skip it and look at subsequent jobs. However, if a job has been skipped long enough, we start allowing it to launch non- local tasks, to avoid starvation. The key insight behind delay scheduling is that although the first slot we consider giving to a job is unlikely to have data for it, tasks finish so quickly that some slot with data for it will

definition	▪ The Capacity Scheduler is designed to allow sharing a large cluster while giving each organization capacity guarantees.
Algorithm	▪ When there are open slots in some task Tracker, the scheduler will choose a queue, then choose a job in that queue, then choose a task in the job and last give this slot to the task
Advantages	▪ ensure guaranteed access with the potential to reuse unused capacity and prioritize jobs within queues over large cluster
disadvantages	▪ The most complex among three schedulers

Table 2.3 Capacity scheduler

free up in the next few seconds. The key insight behind delay scheduling is that although the first slot we consider giving to a job is unlikely to have data for it, tasks finish so quickly that some slot with data for it will free up in the next few seconds. Delay scheduling is a solution that temporarily relaxes fairness to improve locality by asking jobs to wait for a scheduling opportunity on a node with local data. When a node requests a task, if the head-of-line job cannot launch a local task, it is skipped and looked at subsequent jobs.

definition	▪ Queue based scheduling relaxing fairness for locality enhancement
Algorithm	▪ although the first slot we consider giving to a job is unlikely to have data for it, tasks finish so quickly that some slot with data for it will

	free up in the next few seconds
Advantages	▪ Simplicity of scheduling
disadvantages	▪ No particular

Table 2.4 Delay scheduler

2.6 Dynamic priority scheduling

The purpose of this scheduler is to allow users to increase and decrease their queue priorities continuously to meet the requirements of their current workloads. The scheduler is aware of the current demand and makes it more expensive to boost the priority under peak usage times. Thus users who move their workload to low usage times are rewarded with discounts. Priorities can only be boosted within a limited quota. All users are given a quota or a budget which is deducted periodically in configurable accounting intervals. How much of the budget is deducted is determined by a per-user spending rate, which may be modified at any time directly by the user. The cluster slots share allocated to a particular user is computed as that users spending rate over the sum of all spending rates in the same accounting period. It supports capacity distribution dynamically among concurrent users based on priorities of the users. It allows users to get Map or Reduce slot on a proportional share basis per time unit. These time slots can be configured and called as allocation interval. It is typically set to somewhere between 10 seconds and 1 minute. This model appears to favour users with small jobs than users with bigger jobs. To avoid starvation, queue blocking and to respond to user demand fluctuations more quickly pre-emption is also supported. Following are some of its features:

- This scheme discourages the free-riding and gaming by users.
- Possible starvation of low-priority tasks can be mitigated by using the standard approach in Hadoop of limiting the time each task is allowed to run on a node.
- It allows administrators to set budgets for different users and let them individually decide whether the current price of preempting running tasks is within their budget or if they should wait until the current users run out of their budget.
- It can be easily be configured to mimic the behavior of the other schedulers.

definition	▪ Allow users to increase and decrease their queue priorities continuously to meet the requirements of their current workloads.
Algorithm	▪ Supports capacity distribution dynamically among concurrent users based on priorities of the users.
Advantages	▪ can be easily be configured
disadvantages	▪ If the system eventually crashes then all unfinished low priority processes gets lost.

Table 2.5 Dynamic priority scheduler

2.7 Deadline constraint scheduler

Deadline Constraint Scheduler addresses the issue of deadlines but focuses more on increasing system utilization. It is based on following assumptions:

- All nodes are homogeneous nodes and unit cost of processing for each map or reduce node is equal
- Input data is distributed uniform manner such that each reduce node gets equal amount of reduce data to process
- Reduce tasks starts after all map tasks have completed;
- The input data is already available in HDFS
- Schedulability of a job is determined based on the proposed job execution cost model independent of the number of jobs running in the cluster.
- After a job is submitted, schedulability test is performed to determine whether the job can be finished within the specified deadline or not.
- A job is schedulable if the minimum number of tasks for both map and reduce is less than or equal to the available slots.
- When a deadline for job is different scheduler assigns different number of tasks to Task Tracker and makes sure that the specified deadline is met.

Its features may be summarized as follows:

- Focuses on deadline constraint of job.
- Predicts total time required for job completion.
- Schedules job in order of their deadlines
- Helps in Optimization of Hadoop implementation.

Capacity Planning based on resource usage

definition	▪ Focuses on deadline constraint of job
Algorithm	▪ Deadline Constraint Scheduler addresses the issue of deadlines but focuses more on increasing system utilization. It is based on following assumptions
Advantages	▪ Helps in Optimization of Hadoop implementation
disadvantages	▪ Nodes should be uniform in nature which incurs cost

Table 2.6 Deadline constraint scheduler

2.8 Resource aware scheduler

Resource Aware Scheduling in Hadoop has become one of the Research Challenges in Cloud Computing. Scheduling in Hadoop is centralized, and worker initiated. Scheduling decisions are taken by a master node, called the Job Tracker. The Job Tracker maintains a queue of currently running jobs, states of Task Trackers in a cluster, and list of tasks allocated to each Task Tracker. Each Task Tracker node is currently configured with a maximum number of available computation slots. Each Task Tracker node monitors resources such as CPU utilization, disk channel IO in bytes/s, and the number of page faults per unit time for the memory subsystem. Two of its approaches are:

- Dynamic Free Slot Advertisement
- Free Slot Priorities/Filtering

III. CONCLUSION

This paper summarizes several existing schedulers used in Hadoop along with their advantages and disadvantages. Each of these specified Schedulers relies on the resources like CPU, Memory, Job deadlines and IO etc. we can also see that most of these schedulers discussed in this paper addresses one or more problem. And choice of this scheduler for a particular job is up to the user. Also schedulers discussed here requires cluster network with similar systems. Future work will consider scheduling in Hadoop in environments with dissimilar systems. Ability to make Hadoop scheduler resource aware is one the emerging research problem that grabs the attention of most of the researchers as the current implementation is based on statically configured slots.

Acknowledgements

We are greatly indebted to the college management and the faculty members for providing necessary facilities and hardware along with timely guidance and suggestions for implementing this work.

REFERENCES

[1] http://www.ibm.com/developerworks//library/os- hadoop-scheduling/
[2] http://blog.cloudera.com/blog/2008/11/job-scheduling-in-hadoop/
[3] http://cps516project2donghe.weebly.com/fifo.html
[4] http://hadoop.apache.org/docs/current/hadoop yarn/hadoop-yarn-site/CapacityScheduler.html
[5] http://hadoop.apache.org/docs/stable/fair_scheduler.html
[6] "Survey on Improved Scheduling in Hadoop MapReduce in Cloud Environments " by b thirumala rao