

Scheduling Job Queue on Hadoop using Hybrid Hadoop Fair Sojourn Protocol

M. Gowthami*, J. Briskilal and R. Jayashree

Department of Computer Science and Engineering, Vel Tech High Tech Dr. Rangarajan Dr. Sakunthala Engineering College, Avadi, Chennai - 600062, Tamil Nadu, India; gowthamimailme@gmail.com, briskilal.j@gmail.com, shreej808@gmail.com

Abstract

Background/Objectives: This paper gives an idea how to avoid starvation and minimize response time using hybrid hadoop fair sojourn protocol. **Methods/Statistical analysis:** The scheduler assigns the resources to the cluster so that job dimension is conditional whereas the job makes improvement towards its conclusion. Scheduling selection help the approach of computer-generated time and cluster funds are focused on jobs according to their arrangement. This ensures that not any of jobs (small or large jobs) undergo starvation. **Findings:** The Shortest Remaining Processing Time (SRPT)¹, ranks the least amount of work to complete, is the one to minimize response time (sojourn time). The Hybrid Hadoop Fair Sojourn Protocol (hybrid HFSP)² to pause jobs with higher SRPT and allow other waiting jobs in queue based on First Come First Serve (FCFS). **Application/Improvements:** Relating to existing method the accuracy is developed in suggested method.

Keywords: First Come First Serve (FCFS), Hadoop, Shortest Remaining Processing Time (SRPT)

1. Introduction

Hadoop is an open Java-based software design skeleton that maintains the handling of large data sets in a parallel and distributed working environment. It makes use of the hardware hadoop is highly scalable and Fault tolerant. Hadoop runs in cluster² and eliminates the use of a super computer. Hadoop is the widely used big data processing engine³ with a simple master slave setup. Big data in most companies are processed by hadoop by submitting jobs to master. The master distributes the job to its cluster and process map reduce takes sequentially. But now a days growing data need and competition between service provider leads to the increased submission of jobs to the master. This concurrent jobs submission on Hadoop focuses us to do scheduling on hadoop cluster so that response time will be acceptable for each job.

2. Hybrid Hadoop Fair Sojourn Protocol

The hybrid HFSP is used to know prior job size information before the process gets started so that starvation can be eliminated and response time of the process is also reduced. These schedulers merge FCFS and Processor Scheduling (PS)^{4,5}. Once a job is submitted to master, the master will allocate the cluster resources correspondingly between all the active jobs. For each phase of the job, cross scheduler will calculate SRPT value.

Based upon the SRPT value, the job is planned in a queue. The job which has higher SRPT is passed in a queue for a while and the rest is processed by a master. While calculating the SRPT^{6,7} value the scheduler knows the prior job size information so that starvation can be avoided for each phase of the job.

*Author for correspondence

3. Related Work

Size based scheduling address a problem of scheduling a job^{8,9}, which means that how resources are allocated to running jobs. In this work, they used a scheduling concept, such as processor sharing and FCFS.

The disadvantages of first come first serve are to serve one job at one time so that small job will suffer from starvation. The demerits of processor sharing are that the arriving new jobs will take more time to complete and overall process completion time will get delayed¹⁰.

4. Architecture Diagram

Figure 1 depicts that the admin will collect all the datasets of information. The batch job is execution of a series of program, each on a set or batch of inputs, rather than a single input. The series of steps in a batch process are often called as job. The jobs are submitted to the hybrid scheduler and also stored in Highly Distributed File System (HDFS). A hybrid scheduler is a scheduling discipline that clubs different scheduling criteria in one algorithm. It clubs FCFS and PS.

The scheduler FCFS is don't know about the prior job size information so that job that is reached first is to be worked first. So, that worked first SRPT value is calculated. The job that has high SRPT will wait in queue and other jobs are submitted to the master.

The master will allocate the cluster resources to all the slaves' equally¹¹. The highly distributed file system has data about tab separated value and comma separated value. The slave will perform the task sequentially and the results are thrown to the master. The master will club all the output of the slave and gives final output of the job submitted.

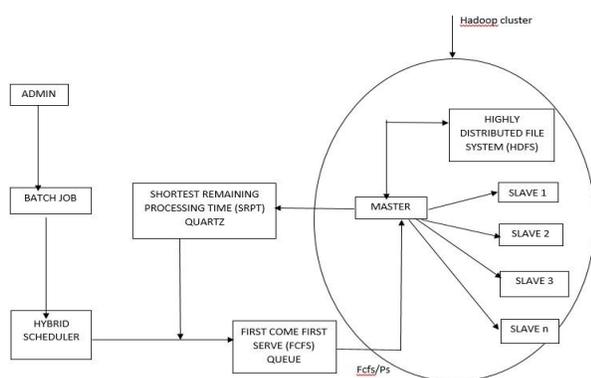


Figure 1. Architecture diagram.

5. Implementation

- I. Big data and environment
- II. Running a Batch Job through FCFS
- III. Size based scheduling on concurrent jobs
- IV. Extending HFSP for mistreatment (i.e.) Starvation Map Reduce jobs

5.1 Big Data and Environment

In Figure 2, the enormous Collection of data is recovered from open basis datasets so as to be publicly accessible from major Application Providers like Amazon. Big Data Representations were examined and also an operational law of the plan is established. The TSV⁵ (Tab Separated Values) and CSV⁵ (Comma separated values) files are Stored in HDFS (Highly Distributed File System) and were read through Master and operated using Java API to itself established by us which is simply modifiable, developer approachable and light weighted.

5.2 Running a Batch Job through FCFS

In Figure 3, the batch job is a backend job running in hadoop clusters and also called as long running jobs as it is scheduled to process bulk data so that the application would makes use of the results produced for updation. Sample jobs are submitted to hadoop master and hadoop master will run the jobs based on a well known technique called First Come First Serve Manner (FCFS). Parallel execution of job is done by hadoop cluster and the results are shown through a well-known Framework called Map Reduce. The Mapper task is done first in slave nodes and reduce task will be done in Master to throw the output.

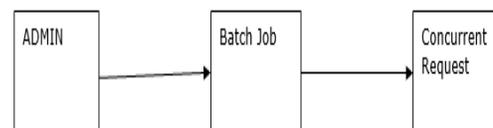


Figure 2. Big data and environment.

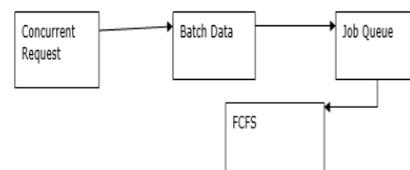


Figure 3. Running a batch job through FCFS.

5.3 Size based Scheduling on Concurrent Jobs

In Figure 4, the n number of jobs are submitted to the hadoop Master and Master will schedule the jobs based on FCFS and PS in a hybrid way. The Capacity of cluster will be analyzed so as to share resources between concurrent jobs arriving to Master. A threshold will be maintained to balance load in slaves and Resource scheduling will not be done further if limit is reached. The Arriving jobs will put in queue until resource gets free in cluster.

5.4 Extending HFSP for Job Mistreatment

In Figure 5, in which jobs may find long waiting time in queue, we extend our hybrid Approach which clubs FCFS and PS to put running jobs on hold for some time, if the particular job has high SRPT. Based on aging of the waiting jobs and SRPT the lengthy running jobs may be locate on hold and the waiting jobs which have high priority will be executed for a while and constant evaluated for SRPT for new jobs to arrive for execution. This Proposed methodology shows high throughput in job completion.

6. Algorithm

```
function ASSIGNPHASETASKS( resources )
  for totally resource s ∈resources do
    if ∃(Job in preparation stage) and
      Tcurr < T then
      job ← choose job to train with minimum first virtual size
```

```
    ASSIGN(s, job) Tcurr ← Tcurr + 1
  else
    job ← choose job with minimum virtual time
    ASSIGN
  N(s, job) end if
end for end function
function ASSIGN(resource, job) task ← choose job by
  lower ID on or after job allocate task to resource
end function
function RELEASERESOURCE( task ) if task is a
  preparation task then
    Tcurr ← Tcurr - 1 end if
end function
```

In Figure 6, the procedure we used a scheduler called hadoop fair sojourn protocol, which don't know about the prior job size information so, this paper propose a new concept called hybrid hadoop fair sojourn protocol which is used to know prior job size information before the process gets started so that starvation can be eliminated and response time of the process is also reduced. These schedulers club PS and FCFS. Once a job is submitted to master, the master will allocate the cluster resources equally among all the running jobs. For each phase of the job, cross scheduler will calculate SRPT value. Depending

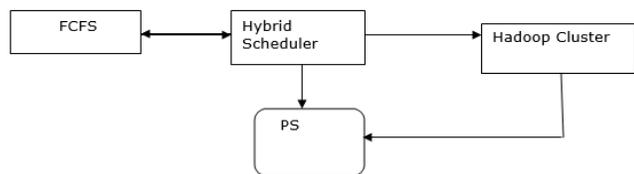


Figure 4. Size based scheduling on concurrent jobs.

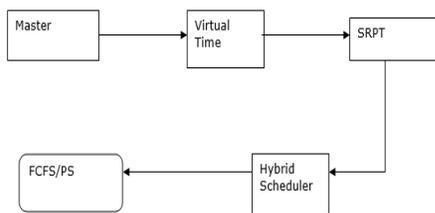


Figure 5. Extending HFSP for job mistreatment.

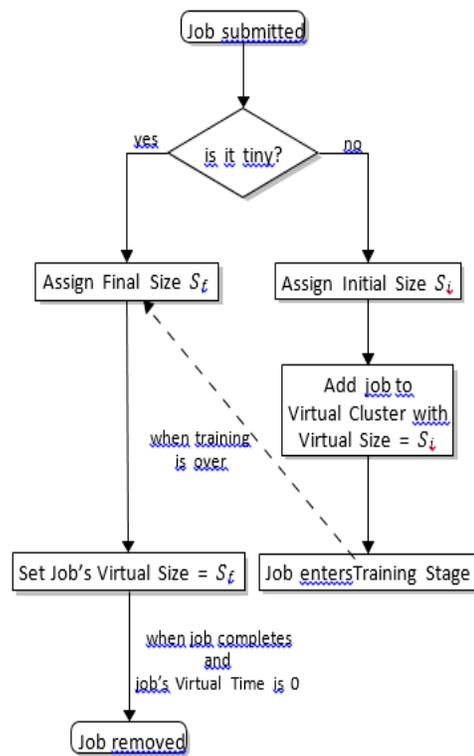


Figure 6. Job allocation algorithm.

upon the SRPT value, the job is scheduled in a queue. The job which has higher SRPT is passed in a queue for a while and the rest is processed by a master¹². While calculating the SRPT value the scheduler knows the prior job size information so that starvation can be avoided for each phase of the job.

7. Conclusion

Resource allocation plays a vital role in hadoop cluster. We mainly focus on resource allocation and starvation. We proposed a new concept called hybrid hadoop fair sojourn protocol which is mainly responsible for minimizing the response time and know the prior job size information. SsHybrid¹⁰ HFSP is simple design using hadoop is used the concept of aging and starvation is eliminated and overall process time is minimized.

8. Future Work

This paper presents the new scheduling protocol called hybrid hadoop fair sojourn protocol which is used to reduce response time and to avoid job starvation. The scheduler distributes the resources to the cluster such that job scope is inferred while the job makes development on the way to its conclusion.

Scheduling selection help the approach of virtual time and cluster properties are absorbed on jobs allowing to their prearrangement. This ensures that not any of jobs (small or large jobs) undergo starvation. The SRPT, ranks the minimum amount of job to complete, is the single to minimize response time (sojourn time). The Hybrid Hadoop Fair Sojourn Protocol (hybrid HFSP)⁹ to pause jobs with higher SRPT and allow other waiting jobs in queue based on FCFS. We extend HFSP i.e. hybrid HFSP to pause jobs with higher SRPT and allow other waiting jobs in queue based on FCFS.

9. References

1. Bodkhe B, Ahire A, Chaudhari M. Dynamic MR: A dynamic slot allocation optimization framework for mapreduce clusters. *International Journal of Science & Technology*. 2016; (6):1–7.
2. Malik R, Garg N, Sheikh MG. Using friendship recommender algorithm on hadoop for node maintenance. *International Journal of Science & Technology*. 2014; 6.
3. Ren K, Kwon YC, Balazinska M, Howe B. Hadoop's adolescence: An analysis of Hadoop usage in scientific workloads. *Proceedings of VLDB*; 2013.
4. Ananthanarayanan G, Ghodsi A, Shenker S, Stoica I. Effective straggler mitigation: Attack of the clones. *National Spatial Data Infrastructure*. 2013; 13.
5. Zaharia, Chowdhury M, Das T, Dave A, Ma J, McCauley M, Franklin MJ, Shenker S, Stoica I. Resilient distributed datasets: a fault- tolerant abstraction for in-memory cluster computing. *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*; 2012. p. 2–2.
6. Zaharia M. Delay scheduling: a simple technique for achieving locality and fairness in cluster scheduling. *Proceedings of ACM EuroSys*; 2010.
7. Dean J, Ghemawat S. Map reduce: simplified data processing on large clusters. *Proceedings of USENIX OSDI*; 2004.
8. Friedman E, Henderson S. Fairness and efficiency in web server protocols. *Proceedings of ACM SIGMETRICS*; 2003.
9. Bender MA, Chakrabarti S, Muthukrishnan S. Flow and stretch metrics for scheduling continuous job streams. *Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics; 1998. p. 270–79.
10. Tanenbaum AS, Tannenbaum A. *Modern operating systems*. Prentice Hall Englewood Cliffs. 1992; 2.
11. Nagle J. On packet switches with infinite storage, *IEEE Transactions on Communications*. 1987; 9.
12. Schrage LE, Miller LW. The queue m/g/1 with the shortest remaining processing time discipline. *Operations Research*. 1966; 4.