ISSN: 2321-7782 (Online)
Volume 3, Issue 3, March 2015
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Simultaneous MapReduce Jobs are arranged for depreciating their formulation period

N.L. Anto Arokia Mary[1]
PG Scholar
Department of CSE
Francis Xavier Engineering College
Tirunelveli, Tamil Nadu, India

Dr. D.C. Joy Winnie Wise[2]
Professor/Head
Department of CSE
Francis Xavier Engineering College
Tirunelveli, Tamil Nadu, India.

S. Malai Arasan[3]
Assistant Professor
Department of CSE
Francis Xavier Engineering College
Tirunelveli, Tamil Nadu, India.

Abstract: ***Big Data deals with the larger datasets which focus on storing, sharing and processing the data. Since, larger data sets are available the usage of map reduce cluster has also been increased in larger amount. The reality is, enormous numbers of jobs are dumped into the cloud which increases the workload. Moreover jobs get executed unorderly and there is a huge influence in the general formulation period of the application. The truth is that a group of tasks which processed affects the processing time and also decreases usage of cluster. Our aim is to minimize the total formulation period and utilization of resources in best manner .In this paper we introduce the weight based scheduling to eliminate the additional work of job which comes from similar input data sets. Weight based scheduling merges the partition equalizer from multiple users for the initial job set execution and the result is stored for the future reference. Eventually, by using MapReduce simulator performance for various workloads are examine and compared.***

Key Words: Bigdata; Hadoop; MapReduce; MapReduce Simulator; Workloads

I. INTRODUCTION

Big data is the collection of bulk data sets which come in different volume, different speed and different forms that makes it complex to process using the traditional database management tools. Bigdata as a new technology it is mainly designed to extract value from large volumes of a wide variety of data by enabling high velocity. Normally this data comes from all over the world in various forms from numbers and words to images and videos. Big data applications include business intelligence, data storage, data processing and data analysis. To bring up the exact value, you need optimal computing power, business analytics technologies and intelligent skill processing.

The main challenges of data processing in big data are parallelization, distribution, scheduling, monitoring, storage capability and fault tolerance. Since, scheduling is one of the major challenges and it forms the basis for resource management. We would like to focus and provide an efficient method for scheduling the jobs using map reduce framework in Hadoop. Bigdata is not processed as like in the traditional database, instead the data are massively parallel processed (MPP). Using databases employ columnar architectures the column containing the solution to the query is processed and unstructured data storage is enabled.

Hadoop is one of the efficient tools used to analyse big data. Map reduces and Hadoop distributor file systems (HDFS) are the main components of Hadoop. Two functions to express the computation are Map and Reduce. Each map task initial work is to split the data logically which is present inside the HDFS. The map operates by reading the data and allocating map functions

to the data that is read. This data is then categorised and separated for different reduce task. The result is then written back to the Hadoop distributor file system. The reduce stage consists of three phases: shuffle, sort and reduce.

i) In the shuffle phase: Intermediate files from the local disk of the machine that already executes the map task.

ii) In sort: Sorting is applied on the intermediate data set which comes after the compilation of shuffle.

iii) In the reduce phase**:** Once the sorting is performed the resultant data is send to the user-defined reduce function. The output is stored back to the HDFS.

In Hadoop master node performs the job scheduling and manages the slaves available under them. Each slave has a map slot and reduce slot to perform the task which is preconfigured. These slaves send the cue containing number of free slot and the information of current task. Scheduling is performed based on the free slot information obtain from the cue.

II. Problem Statement

In the Existing methods the dynamic load balancing the main drawback is the mean waiting time and the mean slow down time are reduced at the server farm system. MapReduce workloads of optimize scheduling of slot allocation does not mention how to use this slot allocation function for various application and for various sizes of input data sets. FLEX does not provide a support for optimizing the completion time of multiple jobs.

Most recent works have overcome the drawbacks of the previous problems. But the existing work provides technique for only a current job set. Problems can arise when there are subsequent job set and moreover some task may execute very slowly than the other tasks. Finding this slower task is very important. Since, these tasks may extend the execution time of the whole job. Most works provide technique for scheduling but do not consider resources allocation as one of the vital parts of scheduling. Adding resource allocation to the end of scheduling is a critical part and much needed.

III. Proposed System Methodologies In Subsequent Job Set

The main goal is to process the given workload in a shortest time and to determine the jobs execution order that minimizes the overall processing time. In partition equalizer we partition the jobs into partitions and then identify the adequate resource allocation for each partition, such that the formulate duration of these partitions are equalized. We introduce a new method called weight based algorithm when there are subsequent job set. The separate job queue will be allotted for each partition. First job set is already scheduled and it is stored. When the Subsequent job set arrives it is processed to be scheduled. Once the map and reduce task is completed it verifies with data logs. If the task count is equal to or below the stored information count, then it will be directly allocated to the respective racks in the cluster without performing the remaining task. When there is no task count is equal to or nearby the subsequent job set it will perform the entire above task and it will be stored for future reference.

IV. Adaptive Technique

A. Splitting the job:

When the sequent list of the job enters the partition equalizer will split into two lists, where the first list contains the jobs which have small execution time. The second list consists of the jobs having larger execution time from the main list. The first list is named as jobs alpha, which contains job one to job split and the second list is named as jobs beta which contains job split+1 to job n

B. Job Ordering and Sorting:

Jobs from the original set are taken and they are sorted and placed in the list and this list is then ordered. This job sorting is performed in descending order from head and tail towards the middle. If the phase type in duration property is map phase means then the job is placed from head else from the tail.

C. Job Configuration and Cluster Allocation:

Based on the sorting taken place in the precious module, the sorted jobs are allocated to the corresponding cluster based upon its size. The job is processed with the allocated cluster and makes span time of the partitioner and compared with the previous existing best time. By using these results, we can find the order of job and the cluster utilization along with the minimized formulate duration

D. Weight based algorithm:

The best optimized result will be checked only once when all the possible combination are completed. Due to this a heavy amount will be wasted if there is any delay in the final outcome. In this paper we specify a fixed time with in which the partition must get over. When each partition completes it will check with the best optimized result. So that, scheduling is kept within the provided time this will keep the schedule in time and also we can find the slower task which increase the execution time.

The final outcome with all the details such as the map count, reduce count, combination of jobs and their corresponding clusters are stored in the log file. The opportunity to access the log file will not affect the running task sets. The second job set enters the scheduling and completes the entire map and reduce process. Before allocating to the corresponding clusters, the job set checks with the log file and set up the stage weights to see whether there are any similar maps and reduces counts done, and then the same cluster will be allocated to the new job set. This eliminates the new search for the allocation of clusters i.e. the equalizer portion.

If there is no related map and reduce counts stored in the log file, then this job set follow the same process as the first job set and calculate stages weights after tasks have been finished and finally it updates the information in the log file which will be useful for the upcoming job sets. Once the partition has been allocated with the corresponding cluster, It is checked with fixed time which is pre-defined then it moves to further partition.

V. SYSTEM DESIGN

The overall process of subsequent jobs to be scheduled for minimizing the formulation period and the corresponding cluster allocation are all described in the fig.1.System Architecture.

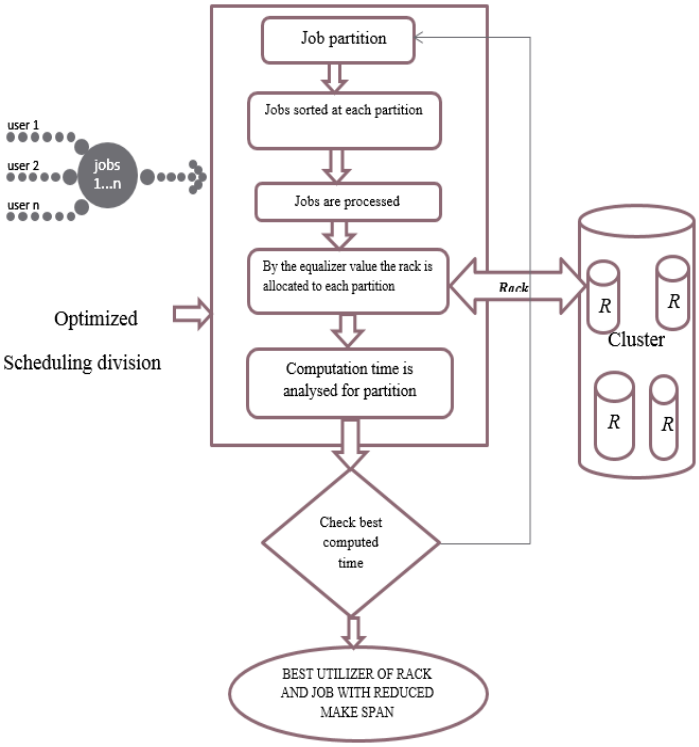


Fig. 1 System Architecture

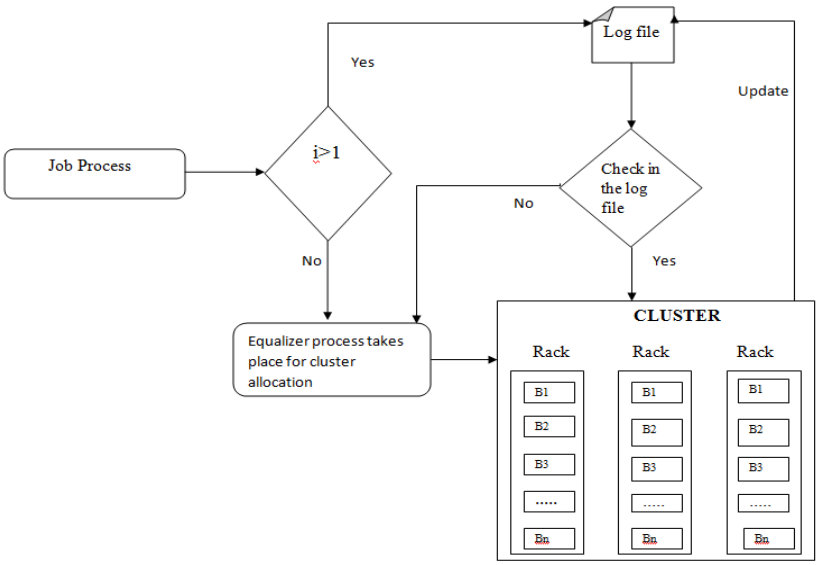


Fig. 2 Job Process

VI. Implementation And Results

I have implemented my work in a simulation environment called MapReduce simulator, where all the daemons of Hadoop like (Name node, Data node, Secondary namenode, Job tracker, Task tracker) are configured already inside. These daemons have a specific responsibility; some resides only one server, some across multiple server. In MR environment different workload management strategy can be evaluated and analyzed using SimMR Based on the statistical property of workloads SimMR uses execution traces of real workloads which is in Hadoop cluster. There are three component present in the SimMR.(A Trace Generator, A Simulator Engine, A pluggable scheduling policy).

- A Trace Generator: This generator generates reusable map reduce workload.
- A Simulator Engine: Follows the functionality of job master and task level execution in Hadoop.
- A pluggable scheduling policy: It helps the scheduler to take decision on job ordering and allocation of resources to different job based on time.

We generate set of jobs which define the interest of production workload. The main objective is to process the provided workload to minimize the time and to determine the order of executing jobs. We determine the execution order by processing the workload with short span of time.

In the fig 2 processing of job is described when i<1 then it will move to cluster allocation part, When it is greater than one then it will move to the log file. In the log file checking process will take place, if it is already present it will directly allocate to the corresponding cluster, if not it will go for the cluster allocation process. Once the partition is allocated to the cluster the details will be updated to the log file, as specified earlier

VII. Conclusion And Future Enhancement

The overall execution time of the MapReduce jobs is been reduced using the concept of weight based algorithm along with partition equalizer by the scheduling criteria and we also observe in which order the job executed and how the execution time is reduced and the usage of the cluster resources increased. This efficiently utilizes the characteristics and properties of optimized job schedule. We are evaluating this with a variety of different MapReduce workloads to measure the performance gains.

The future work involves in improving the mean job delay with the advanced scheduling concepts that still makes the formulate duration to be minimized.

ACKNOWLEDGMENT

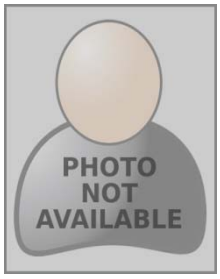
I thank the Lord Almighty who has been with me through every walk in my life, for guiding me and for the blessings showered on me to complete the project in successful manner. I convey my sincere thanks to Dr. D.C. Joy Winnie Wise Professor and Head, Department of Computer Science and Engineering, Francis Xavier Engineering College who inspired me and supported me throughout. I also thank my guide Dr. D.C. Joy Winnie Wise HOD and S. Malai Arasan Assistant Professor, Computer Science and Engineering for their valuable guidance throughout and it is great privilege to express my gratitude to them.

References

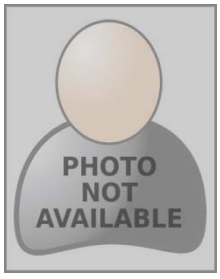
1. Liya Thomas and Syama.R, "Survey on MapReduce Scheduling Algorithms", International Journal of Computer Applications, June 2014.
2. Abhishek Verma., Ludmila Cherkasova. and Roy Campbell H. "Orchestrating an Ensemble of MapReduce Jobs for Minimizing Their Formulate duration', IEEE Transactions on Dependable and Secure Computing, sep 2013.
3. A.Verma, L.Cherkasova, and R.H. Campbell, " Play it Again SimMR!", IEEE int'l Conf. Autonomic Computing(ICAC), 2011.
4. L. Tan and Z. Tari, 'Dynamic Task Assignment in Server Farms: Better Performance by Task Grouping', Proc. Int'l Symp. Computers and comm..(ISCC), 2002.
5. S. Johnson, 'Optimal Two- and Three-Stage Production Schedules with Setup Times Included', Naval Research Logistics Quarterly, 1954

AUTHOR(S) PROFILE

N.L. Anto Arokia Mary, born in 1991 in Tirunelveli is a PG Scholar in Francis Xavier Engineering College. She pursued her B.E from the same college in Tirunelveli. Her interests include networking, cloud computing and big data.



Dr. D.C. Joy Winnie Wise completed PhD from Anna University Chennai. She received M.E from National Engineering College and B.E from PMC tech Thanjavur. She has published numerous papers in various international journals. Her area of interest includes computer networks and communication. She is presently working as a professor and head for the department of computer science and engineering in FX engineering college.



S. Malai Arasan received M.E from University College of Engineering and B.E Infant Jesus College of Engineering. His area of interest includes computer networks, Cloud computing. He is presently working as a assistant professor in the department of computer science and engineering in FX engineering college

