Resource Allocation Scheduling of Jobs in Hadoop Clusters based on Priority Levels

J.Tamizh Selvi[1] & T.Cowsalya[2]

[1,2]Assistant Professor, SVS College of Engineering, Coimbatore, Tamil Nadu, India.

ABSTRACT

Big data analytics is the process of examining large data sets containing a variety of data types – i.e., big data -- to uncover hidden patterns, unknown correlations, market trends, customer preferences and other useful business information. Big data analytics (BDA) applications are software applications that process huge amounts of data using large scale parallel processing infrastructure to obtain hidden value. Hadoop is the most mature open source BDA processing framework, which implements the MapReduce programming paradigm**.** Scheduling problem has been an active area of research in computing systems since their inception. Job scheduling has become an important factor to achieve high performance in Hadoop cluster. The scheduling of jobs and resource allocation is done on Hadoop cluster that runs on Hadoop Constraint Programming based Resource Management algorithm. Several scheduling algorithms have been developed for Hadoop-Map Reduce model which vary widely in design and behavior, handling different issues such as locality of data, user share fairness and resource awareness. Resource allocation and scheduling on clouds are required to harness the power of the underlying resource pool such that the service provider can meet the quality of service requirements of users, which are often captured in service level agreements. This paper introduces a survey of the previous work done in the Hadoop-Map Reduce scheduling algorithms for allocating jobs and gives some suggestion for the improvement of it.

Keywords: Big Data, Big Data Analytics, MapReduce, Clustering, Job Scheduling.

I. INTRODUCTION

Big data has become very popular in Information Technology sector. Big data refers to wide range of datasets which are difficult to be managed by existing conventional applications. The system and people uses the web with an exponential generation of large size of data. The size of data on the web is measured in Exabyte (EB) and Petabytes (PB). The firm growth of data is because of advances in digital sensors, computations, communications, and storage that have created large gatherings of data. The name Big Data had been devised, by Roger Magoulas a researcher, to describe this singularity. Hadoop MapReduce is one of the most popular publicly available frameworks for big data processing in the cloud environment. Hadoop provides for the needs of a wide variety of possible users but does not provide a means to optimize the scheduler for individual users. It consists of mainly two functions Map and Reduce. The Map function takes as input the list of unstructured records and emits for each a set of intermediate key-value pairs. For each key, list of values are produced by map libraries which are applied as input to the reduce operation. Then the reduce libraries collate these values and merge into smaller set of values or a single value. The MapReduce examples include wordcount, distributed grep, distributed sort, pattern-based searching etc.

The main research issues in big data are following: 1) Handling data volume, 2) Analysis of big data, 3) Privacy of data, 4) Storage of huge amount of data, 5) Data visualization, 6) Job scheduling in big data, 7) Fault tolerance. Job scheduling plays an important role in improving the overall system performance in big data processing frameworks. To run data analytics jobs that process large volumes of data, one of the important performance metrics that job scheduling policies are designed to optimize is the average *job response time*, defined as the time elapsed from when a job is submitted till when it is complete.

There has been a wide variety of existing job scheduling policies that are proposed to reduce the average job response time by assuming that the complete information on job sizes is known *a priori* ,mostly for recurring jobs. Simple scheduling policies, such as first-in-first-out (FIFO) and Fair scheduler, do not consider job sizes at all and may suffer from long job response times in many cases. With FIFO, small jobs would be delayed if there exist large jobs ahead of them, a common situation in a shared cluster. With Fair scheduling, the scheduler is downgraded to *processor sharing* when multiple long running jobs are submitted together, and its performance becomes much worse than scheduling jobs one by one. The moral of this story is, schedulers that are oblivious to job sizes may not provide the best possible average job response times.

As a result, this survey targets at providing a brief review on the big data analytics and the job scheduling algorithms using Hadoop Map reduce Clusters. This literature survey further organized as given below: Chapter II explains the major concepts of big data analytics and its applications. Chapter III explains the structure of MapReduce Clusters and its working structure. Chapter IV explains the different scheduling algorithms to allocate jobs. Chapter V depicts effective job scheduling algorithm followed by conclusion and future enhancements.

II. BIG DATA-AN ANALYTICS OVERVIEW

A. Big Data

Big Data refers to all the data that is being generated across the globe at an unprecedented rate. This data could be either structured or unstructured. ... Hadoop is the platform of choice for working with extremely large volumes of data. Big Data increasingly benefits both research and industrial areas such as health care, finance service and commercial recommendation.

1) Volume: The size of available data has been growing at an increasing rate. This applies to companies and to individuals. A text file is a few kilo bytes, a sound file is a few mega bytes while a full length movie is a few giga bytes. Organizations collect data in huge volumes to analyze it based on the market needs.

2)Velocity: Velocity is a combined data infrastructure and data management process that addresses different concerns that are visible after the creation and addition of big data objects. Velocity is directly related to the entire data infrastructure and the organizations.

3)Variability: Inconsistency of the data set can hamper processes to handle and manage it.

4)Veracity : Refers to how accurate or truthful a data set may be. The quality of captured data can vary greatly, affecting accurate analysis architecture in managing and delivering data to recipients as quickly as possible.

5)Variety: Variety defines the nature of data that exists within big data. This includes different data formats, data semantics and data structures types.

B. Big Data Analytics and its applications

There are so many big data applications as shown in Fig 1. Few of them are described below:

i) In Clustering: Using clustering (K-means algorithm) through a simple point and click dialog, users can automatically find groups within data based on specific data dimensions. With clustering, it is then simple to identify and address groups by customer type, text documents, products, patient records, click path, behaviour, purchasing patterns, etc.

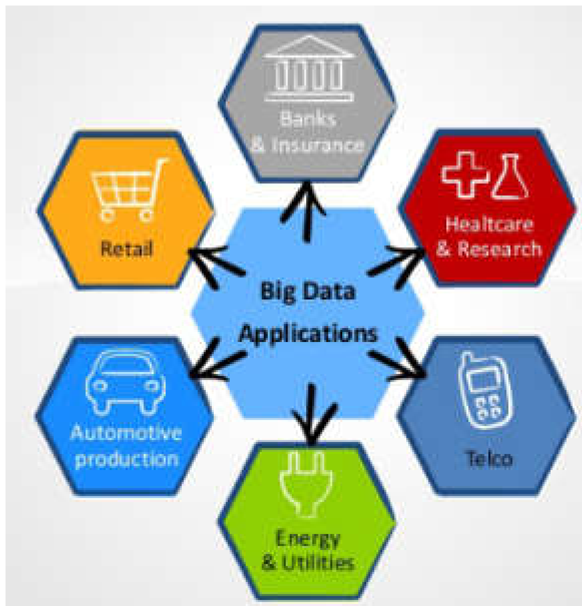

Fig 1: Big Data Applications

ii) *In Data Mining*: Using collective data mining or extraction techniques large sets /volume of data are retrieve from humongous quantity of data. Decision trees illustrate the strengths of relationships and dependencies within data and are often used to determine what common attributes influence outcomes such as disease risk, fraud risk, purchases and online signups.

iii) *Fraud Recognition and Control:* Business operations face many types of fraudulent claims or transaction processing. Hence fraud recognition and control is most resounding big data application . Big data platforms that can verify, analyze, claims and transactions in real time, identifying large scale patterns across so many transactions or detecting inconsistent behavior from an individual user, can change the fraud detection game.

iv) *Social Media Analysis **:*** Of the customer- facing Big Data application examples could discuss, analysis of social media activity is one of the most important. Everyone and their mothers are on social media these days, whether they like company pages on Facebook or tweeting complaints about products on Twitter. A big data solution built to produce and investigates social media activity, like IBM's Cognos Consumer Insights, a fact solution running on IBM's big Insights big data platform, may make the sense of the chatter. Social media data can provide real time insights into how the market is responding to products and campaigns. With those insights, companies can adjust their pricing, promotion, and campaign placement on the fly for optimal results.

III. MAP-REDUCE CLUSTERS

A. Overview

MapReduce breaks the processing into two phases: the map phase and the reduce phase. Programmers can complete a MapReduce application by implementing two functions: the Map function and the Reduce function as given in Fig 2. Further, both of the two functions have pairs as their input and output. Each of the Map tasks processes a pair (i.e., the input of the map function) to generate a set of intermediate pairs (i.e., the output of the map function). Then, the Reduce function is applied to merges the set of intermediate data, which is produced by Map tasks. The output pairs of the reduce function are finally sorted based on key values.

B. MapReduce optimization

A regression-based prediction to help to decide the appropriate number of map and reduce tasks to optimize the scheduling of MapReduce applications. Firstly, the relevant information of the history jobs is collected in preparation for the next regression step. The information needed includes some parameters related to the job scheduling, e.g., the size of the input data set, the number of map and reduce tasks, the real execution time of the job. Hardware heterogeneity occurs because servers are gradually upgraded and replaced in datacenters. The difference in processing capabilities on MapReduce nodes breaks the assumption of homogeneous clusters in MapReduce design and can result in load imbalance, which may cause poor performance and low cluster utilization. Despite these optimizations, most MapReduce implementations such as Hadoop still perform poorly in heterogeneous environments. Thus, running tasks with homogeneous configurations on heterogeneous nodes inevitably leads to sub-optimal performance.

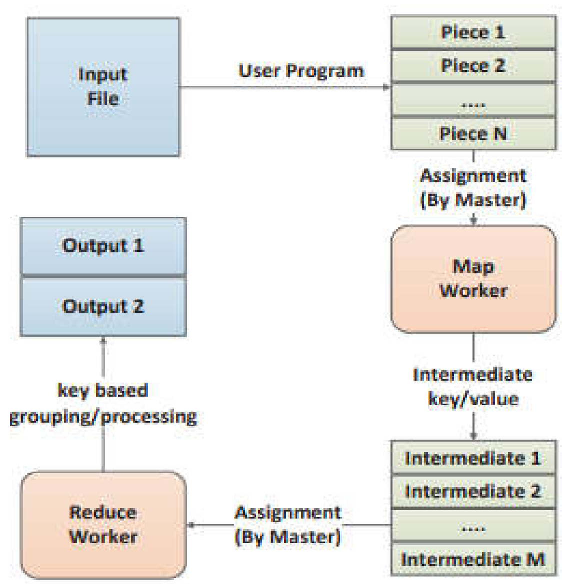

Fig 2: Map-reduce Operation

C. Cluster Techniques

Big data having sparse data as a type is the major challenge for designing and developing efficient techniques for mining partial and inexact data. Clustering techniques are useful methods to distinguish diverse forms which support in countless applications in business. Partition and density-based clustering algorithms are divided into hierarchical order, and these methods is subbifurcated into divisive and agglomerative algorithm' s which works on clustering in hierarchy. The distinct types of clustering algorithms are as follows:

i) Partitioning-based: Partitioning based techniques divide data items into a number of small partitions that are obtained from the divided data items, where separately divider characterizes a cluster, and all the clusters are all quickly determined in such techniques. The requirements that needs to be fulfilled are as follows: 1. One item must be present in each cluster, and 2. One cluster must have one set. A center point is considered as the middling value of all the points and is used obtained from the arithmetic mean, this method is used in K-means algorithm. The items near to the center represent the cluster in K-medoids algorithm.

ii) Hierarchical-based: Hierarchical clustering is a method of cluster analysis which seeks to build a hierarchy of clusters. Strategies for hierarchical clustering generally fall into two types:

a) Agglomerative: This is a "bottom up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.

b) Divisive: This is a "top down" approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.

D. Clustering Benchmark

i)*Dataset Type*: Numeric and categorical qualities must be equally considered in accordance to the physical world.

ii)*Size of Dataset*: The foremost consequence on the clustering superiority depends on the magnitude of the dataset.

iii)*Parameters Taken as Input*: Since a huge number of constraints may disturb cluster superiority because they will be contingent on the values of the constraints, a necessary feature for "practical" clustering is the one that has fewer restrictions.

iv)*Handling Outliers/ Noisy Data*: Noise makes it problematic for a procedure to cluster an object into a applicable cluster as it is challenging to handgrip outlier/noisy data.

v)*Time Complexity*: The time complication of the clustering approaches must be enriched, because, if the process takes extended time, then it can become impossible for applications that manage big data.

vi)*Stability of the Algorithm*: Clustering procedure is the capability to produce the same partition of the data regardless of the command in which the patterns are accessible to the procedure.

vii)*High Dimensionality Handling:* Numerous solicitations necessitate the examination of objects comprising a huge quantity of structures (dimensions) in cluster analysis.

viii)*Cluster Shape*: A clustering procedure needs to provide clusters of subjective shape which should be able to lever actual data and their extensive range of data types.

IV. JOB SCHEDULING AND ITS ALGORITHM

Classification of workload is a major issue to the Big Data community namely job type evolution and job size evolution. Based on job size and disk performance, clusters are been formed with data node, name node and secondary name node. The performance metrics that job scheduling policies are designed to optimize is the average *job response time*, defined as the time elapsed from when a job is submitted till when it is complete. The job scheduling can be done based on the priority levels which is given along with the process. Algorithms in Hadoop MapReduce play a vital role during scheduling of the jobs or tasks. Hadoop has evolved with a variety of pluggable schedulers. These schedulers have their own prominence at the places where they are needed. Schedulers can be classified as Non-Adaptive and Adaptive. Non-Adaptive schedulers are those which are static like the FIFO which do not change with the environmental factors. Adaptive schedulers are the ones which are dynamic in nature and can easily adapt to its changing environment.

FIFO scheduler is the default scheduler of Hadoop based on the concept that the jobs are executed in the order of their submission. Fair scheduler is a scheduler where jobs are allocated for equal amount of resources as a result of which variety of jobs takes less amount of time to complete the process. It was less complex and works well with both small and large cluster. It does not consider the job weight of each node. Job Schedulers are important components of Hadoop MapReduce. Scheduling metrics are based upon improving job response time, throughput, execution time, failure rate, makespan, power, and energy. The optimized scheduler is a flexible task scheduler where the job tracker has a task scheduler which actually schedules tasks. First, it constructs a list known as Task In Progress List, a collection of all running tasks, and caches the list into either on Running Map or non-running Reduce depending on the type of each task. These cache disks are used for the job tracker to manage current map tasks or reduce tasks to be executed. Finally, the scheduler assigns each task to a node randomly via the same heartbeat message protocol. The best case of scheduling a task is when the scheduler locates the corresponding task into the local node.

REFERENCES

[1] J. Chen et aI. , "A Task S cheduling Algorithm for Hadoop Platform," J. of Computers 8 .4, 20 1 3 , pp. 929-9 3 6 .

[2] J.V.Gautam, et al. ,"A survey on job scheduling algorithms in Big data processing",(ICECCT), 2015 IEEE, 978-1-4799-6084-2, 2015.

[3] Thirumala Rao and Dr. L.S.S. Reddy,"Survey on Improved Scheduling in Hadoop MapReduce in Cloud Environments".

[4] M. Collins, “Hadoop and MapReduce: Big Data Analytics,” *Gartner*, 14 Jan. 2011.

[5] N. Lim, S. Majumdar, and P. Ashwood-Smith, “Engineering Re-source Management Middleware for Optimizing the Perfor-mance of Clouds Processing MapReduce Jobs with Deadlines,” *Int’l Conference on Performance Engineering (ICPE)*, 24-26 March 2014, pp.161-172.

[6] Norman Lim, Shikharesh Majumdar, Peter Ashwood-Smith, MRCP-RM: a techniquefor resource allocation and scheduling of MapReduce jobs with deadlines, IEEE Trans. Parallel Distributed Syst. PP (99) (2016),

[7] Yang Liu, Yukun Zeng, Xuefeng Piao, High-responsive scheduling with MapReduceperformance prediction on hadoop YARN, in: IEEE 22nd International Conference on Embedded and Real-time Computing Systems and Applications (RTCSA), 2016,pp. 238–247,

[8] Xiangming Dai, Brahim Bensaou, Scheduling for response time in hadoopMapReduce, IEEE Int. Conf. Commun.

(ICC) (2016) 1–6, http://dx.doi.org/

[9] M. Brahmwar, M. Kumar, G. Sikka, Tolhit – a scheduling algorithm for hadoopcluster, Orig. Res. Article Procedia Comput. Sci. 89 (2016) 203–208.

[10] S. Suresh, N.P. Gopalan, An optimal task selection scheme for hadoop scheduling,IERI Procedia 10 (2014) 70–75.

[11] J. Geetha, N. UdayBhaskar, P. ChennaReddy, Data-local reduce task scheduling,Procedia Comput. Sci. 85 (2016) 598–605.

[12] Mohammad Hammoud and Majd F. Sakr, "Locality-aware reduce task scheduling for MapReduce," 3rd Int. Conf. on Cloud Computing Technology and Science (CloudCom), IEEE, pp. 570-576, 2011.

[13] K. Kc, K. Anyanwu, "Scheduling Hadoop Jobs to Meet Dead-lines,"*Int'l Conf on Cloud Computing Technology and Science*, Nov. 30 2010-Dec. 3 2010, pp.388-392.

[14] Jorda Polo et al. , "Performance-driven task co-scheduling for mapreduce environments," Network Operations and Management Symposium (NOMS), IEEE, pp. 373-380, 2010.

[15] D . Fangpeng and S . G. Akl, " S cheduling algorithms for grid computing:State of the art and open problems ," Vol. 504. Tech. Rep . , 2006.

[16] M . Zaharia et aI . , "Delay scheduling: a simple technique for achievinglocality and fairness in cluster scheduling , " Proc. 5th European conf. on Computer systems. ACM, 20 1 0 , pp. 265-278 .

[17] N. Tiwari, et al.,"Classification Framework of MapReduce Scheduling Algorithms", ACM Computing Surveys, Vol.47, No.3, Article 49, 2015.

[18] Geetha J, et al. ," Hadoop Scheduler with Deadline Constraint ",(IJCCSA) ,Vol. 4, No. 5, October 2014.

[19] C. He, Y. Lu and D. Swanson," Real-Time Scheduling in MapReduce Clusters ", DOI 10.1109/HPCC.and.EUC. 216, 2013.

[20] D. Yoo and K. M. Sim . "A comparative review of job scheduling for MapReduce. " Cloud Computing and Intelligence Systems, IEEE, 2011.