Advances in Computational Sciences and Technology
ISSN 0973-6107 Volume 10, Number 8 (2017) pp. 2309-2316
© Research India Publications
http://www.ripublication.com

A Brief Preview of Efficient Hadoop Job Schedulers

Mukesh Singla
Research Scholar, OPJS University
Rajgarh (Sadulpur) Churu, Rajasthan, India.

Abstract

BigData is a collection of large and complex data sets which is growing rapidly in every day to day life. Many challenges had been faced like storing, managing, and effective analyzing which was not overcome by the traditional approach. Therefore advanced approach have been introduced which results in the beneficial of many companies for analyzing and handling the bigdata. That one of the advanced approach is Hadoop which is scalable and inexpensive. Hadoop approach helps in analyzing, storing, processing and managing large amount of data. It provides the decision making capability by loading and storing large amount of data. This paper helps in brief examination of various schedulers for efficient data.

INTRODUCTION

1.1 What is BigData?

The creation of large and growing files is mainly defined as bigdata. the rapidly increase of data is generated from the electronic devices which we are using now a days like computers, smart phones , sensor networks and social websites using for communication in our daily lives like twitter, facebook etc. These data cannot be efficiently managed and accessed by any traditional approach.

Challenges of bigdata, mainly we have three volume, variety and velocity

1. Variety (type) - The type and nature of the data that is produced like text, audio and video files in different format, images etc.
2. Volume (size)-the quantity of generated and stored data that is size of data. large files in terabytes or zetabytes.

3. Velocity (time)-speed of data processing and analyzing [1].

1.2 Approaches for bigdata?

i. Traditional Approach: To deal with bigdata, there is a need of approach that is traditional analytical tool .These tools are made to visualize, ingest, analyze and reproduce which gives result by processing the information. This increasing in quantity of data was collected by domains like business, banking and finance, engineering research, health care, scientific resources, IOT, health care and many more. These traditional Analytical tools were more in use in earlier era because now quantity of data is growing rapidly with passing time which leads to the assignment of the platform. Some challenges were also faced during the increase of data and they are storing, managing, and effective analyzing. To make the smarter decisions in aided domains data must be smashed out and had to face the above challenges before analytical tool to get correct, quick and beneficial information [2].

ii. Advanced Approach: Many challenges were faced in old architecture of data processing due to Bigdata. Therefore advanced approach have been introduced which results in the beneficial of many companies for analyzing and processing the bigdata. There is many framework or tools available in advanced approach and they are dryad, apache Hadoop MapReduce, yarn and many more. This overwhelms the disadvantage of traditional approach which is not scalable and costly. But this hadoop approach is scalable and inexpensive. They breakdown the data into small chunks and then computation is performed on them and then result is combined [3].

1.3 What is Hadoop?

Hadoop is a set of tools that support running of applications on Big Data. These set of tools are called projects like hive, hbase, pig and many more. Hadoop is an open source set of tools and it is distributed under Apache license. Hadoop has the feature to filter the data and gets some relevant information to the company. Hadoop is technology or tool that is required to process and analyzed bigdata. Many organizations and companies such as Amazon, facebook, twitter, yahoo has adopted Hadoop [3].

1.4 Need for Hadoop?

In term BigData, data is the major factor in today’s life that makes the BigData. The limitation of data to be stored or accessed in database does not mean that data is bigdata. The requirement of new tools and techniques to analyzed and processed the data is defined as bigdata [4]. Google introduced BigData when distributed computation had been started in last ten years of 19th century. Therefore Google give a new discovery of distributed file system named as GFS (Google File System) in starting of 2000.this file system handles and manages huge amount of data i.e. storing, analyzing, retrieving and processing the data. Due to its functions demand had been increased by the users which were not available because it was not open source. Hence new technique or framework had been introduced named as Hadoop [5].

1.5 Architecture of Hadoop

At very high level hadoop has simple architecture, it has two main components and they are MapReduce and HDFS(Hadoop Distributed File System).It follows the master-slave concept that is master node runs the JobTracker process and each slave node runs a TaskTracker process. JobTracker goal is to break bigger task into smaller pieces and send each piece of computation to the task tracker. TaskTracker job is to process the smaller piece of task that is given to this particular node. Job Tracker and the Task Tracker fall under the main component that is MapReduce. MapReduce processes the data which is stored in file system [6]. HDFS: from the name only it signifies that it stores all the data that needs to be processed. Data node and name node comes under the umbrella of HDFS component [7]. DataNode job is to manage the piece of data that has been given to this particular node. NameNode is responsible to keep an index; index tells which data is residing on which DataNode [4].

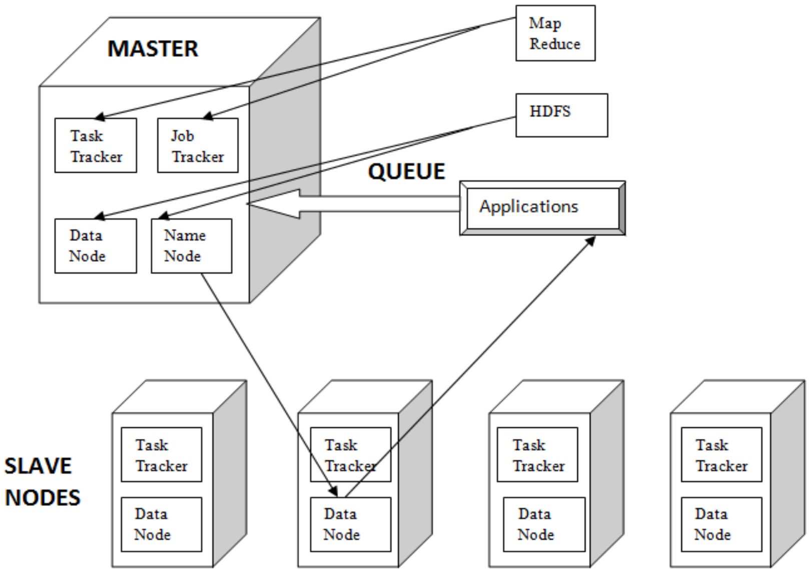

Figure 1.1 Hadoop Architecture

1.6 Hardware Failure in Hadoop:

1. Built in fault tolerance (for slave): by default it maintains three copies of each file and these files are scattered along different computers. So when computer fails, the system keeps on running data is available from different nodes and once that failed node get fixed Hadoop will take care of that and copied some other data files to that node. It is not limited to disk failing but also applicable to task tracker services, if any of the computer fails than job tracker will detect the failure and it will ask some other task tracker to perform the same job.
2. Master Backup (for master): it is a single point of failure. The tables that are maintained by the name node are backed up. That backup tables are copied to various computers. The enterprise version of Hadoop also keeps two master one is the main master and second is the backup master(in case master dies) so it is not a single point of failure[5].

1.7 Optimization

The performance can be made efficient by optimizing HDFS and MapReduce which are the main components of Hadoop.
HDFS optimization: we can achieve optimization in HDFS in two ways and they are:

i. Optimization of data storage structure
ii. Optimization of data storage strategy

It has limitation that they are very costly to optimize it.
MapReduce optimization: optimization in MapReduce is done in three aspects and they are:

i. Optimizing application program
ii. Modifying configuration parameters
iii. Improving the scheduling strategy of the scheduler

Improving the scheduler is the best method to achieve optimization as it is applicable with every Hadoop system [8].

LITERATURE REVIEW

2.1 Hadoop in Cloud

Due to the expanding growth of data, cloud computing and MapReduce programming model have been developed [9]. The applications which are cloud based must have the potential to process or access the large scale data because the rate of growing data to be processed is much faster than the availability of computing power [10]. Therefore to access BigData we have framework such as cloud computing and platform such as Hadoop to process the workloads. Private clouds are implemented on Hadoop based clusters which are vastly used for resource sharing [7].

2.2 Contention in Hadoop Scheduling

In Hadoop job scheduling there are two main frictions that is data locality and fairness. It is impossible to achieve data locality and fairness for all the jobs of users. Data locality has been negotiated if fairness has been achieved and results in low performance and vice versa. Therefore one has to be sacrificed for better performance [4].

2.3 Schedulers to achieve the contention

Various schedulers have been proposed to increase the efficiency of scheduler in Hadoop and they are describing in below table:

Table 2.1 Features and merits of various schedulers

TECHNIQUE	AUTHOR & YEAR	BASED ON	FEATURES	MERITS
Dynamic Hadoop Fair Schedulers (DHFS)	Shanjiang Tang, Bu-Sung Lee, Bingsheng He in 2013	Dynamically allocating map (or reduce) slots to map and reduce tasks.	fairness for cluster and pools	Improves the performance and utilization of the Hadoop cluster significantly.
Octopus, a multi-job fair scheduler	Srikant Padala, Dinesh Kumar, Arun Raj in 2015	Non-preemptive time sharing and spatial resource sharing.	Consider the node capabilities while allocating a job to a set of nodes.	Developing a multi-job fair scheduler for Graphlab.
Real-Time MapReduce (RTMR) scheduler	Chen He, Ying Lu, David Swanson in 2013	cluster-based scheduling	Real- time property for all admitted MapReduce jobs.	better cluster utilization and ratio of job success
JobTracker Initiative Task Scheduler	Kazuki YAMAZAKI∗, Ryota KAWASHIMA∗, Shoichi SAITO∗ and Hiroshi MATSUO∗ in 2013	Load of the cluster	Execution time of each job was reduced, when two or more applications were executed concurrently.	Efficient CPU utilization

Constraint Scheduler	Kamal Kc, Kemafor Anyanwu in 2010	real time cluster scheduling approach	minimum map and reduce task count criteria	deadline achieved in performing task scheduling
FSPY (Fair Sojourn Protocol in YARN) scheduler	Yang Liu∗, Yukun Zeng∗, Xuefeng Piao† in 2016	Size-based scheduling.	calculate job virtual sizes and achieves a satisfying precision	improve responsiveness with guaranteeing fairness

CONCLUSION:

In this paper we have studied many techniques for making the efficient scheduler so that we can speed up our system or data retrieval. Different schedulers like Dynamic Hadoop Fair Schedulers improves the improves the performance and utilization of the Hadoop cluster, Octopus, develops a multi-job fair scheduler by considering the node capabilities, Real-Time MapReduce (RTMR) scheduler provides better cluster utilization and ratio of job success, FSPY (Fair Sojourn Protocol in YARN) scheduler improve responsiveness with guaranteeing fairness by calculating job virtual sizes etc.

REFERENCES

[1] Althebyan, Qutaibah, et al. "Multi-threading based map reduce tasks scheduling." *Information and Communication Systems (ICICS), 2014 5th International Conference on*. IEEE, 2014.

[2] Pradhananga, Yanish, Shridevi Karande, and Chandraprakash Karande. "High performance analytics of bigdata with dynamic and optimized hadoop cluster." *Advanced Communication Control and Computing Technologies (ICACCCT), 2016 International Conference on*. IEEE, 2016.

[3] Xu, Yongliang, and Wentong Cai. "Hadoop Job Scheduling with Dynamic Task Splitting." *Cloud Computing Research and Innovation (ICCCRI), 2015 International Conference on*. IEEE, 2015.

[4] Hannan, Shaikh Abdul. "An overview on Big Data and Hadoop." *International Journal of Computer Applications* 154.10 (2016).

[5] Alam, Anam, and Jamil Ahmed. "Hadoop architecture and its issues." *Computational Science and Computational Intelligence (CSCI), 2014 International Conference on*. Vol. 2. IEEE, 2014.

[6] Elkholy, Amr M., and Elsayed AH Sallam. "Self adaptive Hadoop scheduler for heterogeneous resources." *Computer Engineering & Systems (ICCES), 2014 9th International Conference on*. IEEE, 2014.

[7] Liu, Shengyuan, et al. "Evaluating task scheduling in hadoop-based cloud systems." *Big Data, 2013 IEEE International Conference on*. IEEE, 2013.

[8] Shu-Jun, Pei, et al. "Optimization and Research of Hadoop Platform Based on FIFO Scheduler." *Measuring Technology and Mechatronics Automation (ICMTMA), 2015 Seventh International Conference on*. IEEE, 2015.

[9] Feller, Eugen, Lavanya Ramakrishnan, and Christine Morin. "On the performance and energy efficiency of Hadoop deployment models." *Big Data, 2013 IEEE International Conference on*. IEEE, 2013.

[10] Kapil, B. Sutariya, and S. Sowmya Kamath. "Resource aware scheduling in Hadoop for heterogeneous workloads based on load estimation." *Computing, Communications and Networking Technologies (ICCCNT), 2013 Fourth International Conference on*. IEEE, 2013.

[11] Tang, Shanjiang, Bu-Sung Lee, and Bingsheng He. "Dynamic slot allocation technique for MapReduce clusters." *Cluster Computing (CLUSTER), 2013 IEEE International Conference on*. IEEE, 2013.

[12] Padala, Srikant, et al. "Octopus: A multi-job scheduler for Graphlab." *Big Data (Big Data), 2015 IEEE International Conference on*. IEEE, 2015.

[13] He, Chen, Ying Lu, and David Swanson. "Real-time scheduling in mapreduce clusters." *High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC), 2013 IEEE 10th International Conference on*. IEEE, 2013.

[14] Yamazaki, Kazuki, et al. "Implementation and evaluation of the JobTracker initiative task scheduling on Hadoop." *Computing and Networking (CANDAR), 2013 First International Symposium on*. IEEE, 2013.

[15] Kc, Kamal, and Kemafor Anyanwu. "Scheduling hadoop jobs to meet deadlines." *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*. IEEE, 2010.

[16] Liu, Yang, Yukun Zeng, and Xuefeng Piao. "High-Responsive Scheduling with MapReduce Performance Prediction on Hadoop YARN." *Embedded and Real-Time Computing Systems and Applications (RTCSA), 2016 IEEE 22nd International Conference on*. IEEE, 2016.