ISSN No:-2456-2165

Anatomization of Hadoop Map Reduce Job Run Workflow in Big Data

[1]Vinitha .V
Department of Master of Computer Application
AMC Engineering College Bangalore
University VTU

[2]Velantina .V
Department of Computer Science Engineering (M.Tech)
C.B.I.T Kolar
University VTU

Abstract:- Big Data has grown rapidly with wide spread of infrastructures in less than a decade. Big data aims to develop strategies to better leverage data in today's data-driven economy. Hadoop Map Reduce was designed to run Map Reduce jobs only, it can run on the same cluster and share resources efficiently. The generation next of Hadoop's compute platform is YARN which is a global resource manager runs as a master daemon on particular machine and The Classic Map Reduce had some issues like resource utilization, workload problems, to address the scalability issues we had to move towards the YARN Map Reduce Job Run which worked efficiently than the Classic Map reduce. we discuss about the valuable most important analytics tool that is Hadoop, its architecture, Hadoop Map reduce and the Working of Map Reduce Job Run workflow, these concepts can be used to develop large scale distributed applications to exploit computational power of nodes and compute the applications intensively.

Keywords:- *Big data, Hadoop, Map Reduce, YARN.*

I. INTRODUCTION

As we know that from various sources huge amount of data is generated. it may be in the form of unstructured ,structured and semi-structured data. Usually ninety percent of data generated is unstructured data and is generated in large volume in size like zeta bytes and peta bytes and terabytes of data are generated every day. Hadoop is a big data analytic tool which can handle operations of handling large amount of data. As data comes from everywhere like online transactions, cell phone GPS signal data etc. The big data is usually defined by five v's that is Volume, Variety, Value, Veracity, and Velocity.

The very known technology used for Big data is Hadoop which is huge data processing system for batch processing system. It supports distributed cluster system, Platform to handle scalable applications, enables parallel data processing. The famous Hadoop users are IBM, Google, Face book, Hp, Twitter.

➢ *Architecture of hadoop*

The Hadoop supports distributed files system mechanism, Traditional hierarchical file organization, Single namespace for the entire cluster, Write once read many model. A small Hadoop cluster consists of single master and multiple worker nodes.

The data node is called the slave node consist of Data node and Task tracker. The Hadoop distributed file system consists of two main categories they are Hadoop distributed file system and Map Reduce. Figure 1 depicts the architecture of HDFS.

- **HDFS Name node** – It contains all the meta data information in main memory, Lists of all the files, List of all the Blocks for each file, List of all the data nodes for each block, File attributes and records every change in metadata.

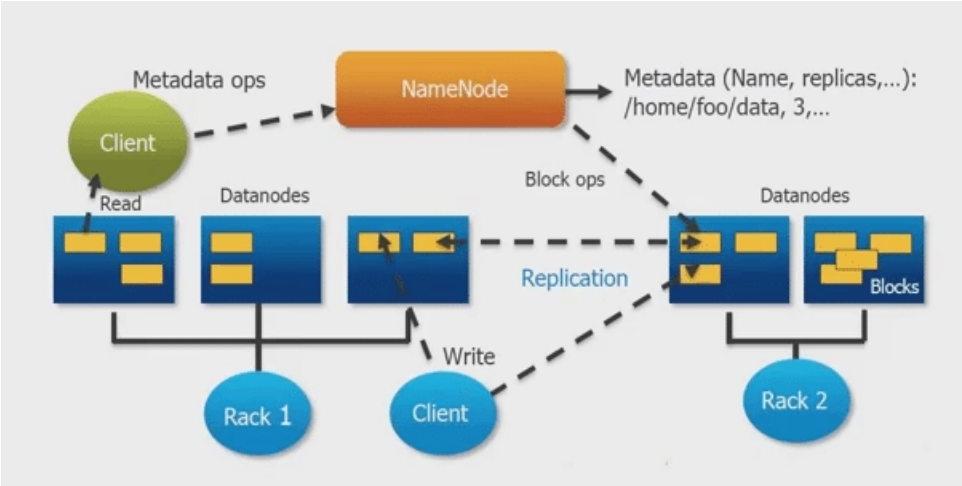

Fig 1: - Architecture of Hadoop Distributed File System

- **HDFS Data node** –Periodic validation checksums is done and the Name node as all the report of the existing blocks.

- **Secondary Name node** – It is a Backup node which consists of data of the Name node, a copy of information which is there at Name node is completely copied to secondary Name node so that when any failure occurs at any time the data can be retrieved from secondary name node.

➢ *Hadoop Map Reduce*

The Map Reduce is powerful paradigm for the parallel computation.

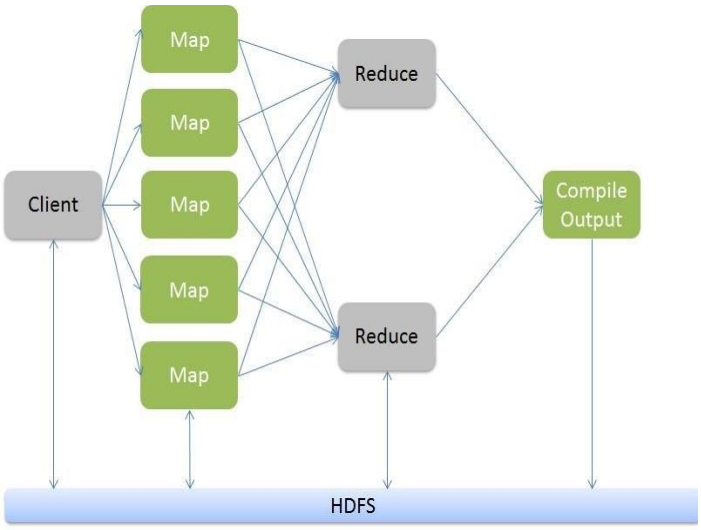

Fig 2:- Map Reduce Flow

Hadoop uses Map Reduce to execute jobs on files in HDFS and will intelligently distribute computation over cluster and takes computation to data. It consists of Map and Reduce programs.

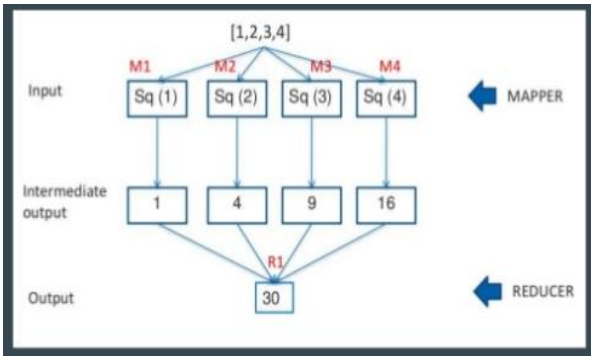

Fig 3:- Example of Map Reduce

The Reduce program returns list constructed by applying s function on the list passed as the second argument can be identified.

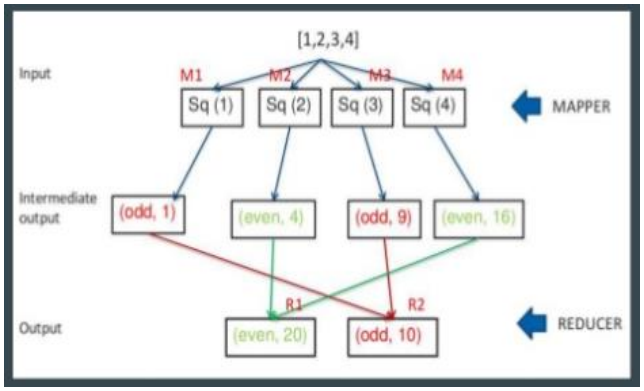

Fig 4:- Map Reduce Example

The Map Reduce program uses Key – value pair to map the data. The shuffle and sorting technique is used during the map reduce process. The shuffle technique is used to reduce the redundancy of information in data it eliminates a duplicate copy if its repeated multiple times. The sorting method sorts the data that is given by the input phase and maps it for providing corresponding output. The working of the Map reduces Job run workflow is described below in detail.

II. PROPOSED SYSTEM

➢ **Map Reduce Job Run: -** It focuses on a job object to run a Map Reduce JOB with a single method call submit().The map reduce function used is JobClient.runJob (conf).The classic work is also called as Map Reduce 1, that is job tracker ,task trackers, yarn for the new framework.

❖ *Classic Map Reduce (Mapreduce1)*

As shown in Figure 5, the classic Map Reduce has four main properties:

➢ The submission of the map reduce job is done by Client.
➢ Job tracker used to coordinate job run. This is a java application that contains main class as Job Tracker.
➢ Task trackers used to run the tasks when a job has been split into Task Trackers, which is java application with task tracker as main class.

- **Job Submission: -** The job submission process implemented by the Job Submitter contains the following:

➢ The job tracker calls the job is ready for execution by calling the method submit Job () on Job Tracker.

- **Task assignment: -** Task trackers executes on a simple loop which sends heart beat signals to job tracker. Heartbeats tell to the job tracker that the task tracker is alive but they also double as a channel for messages.

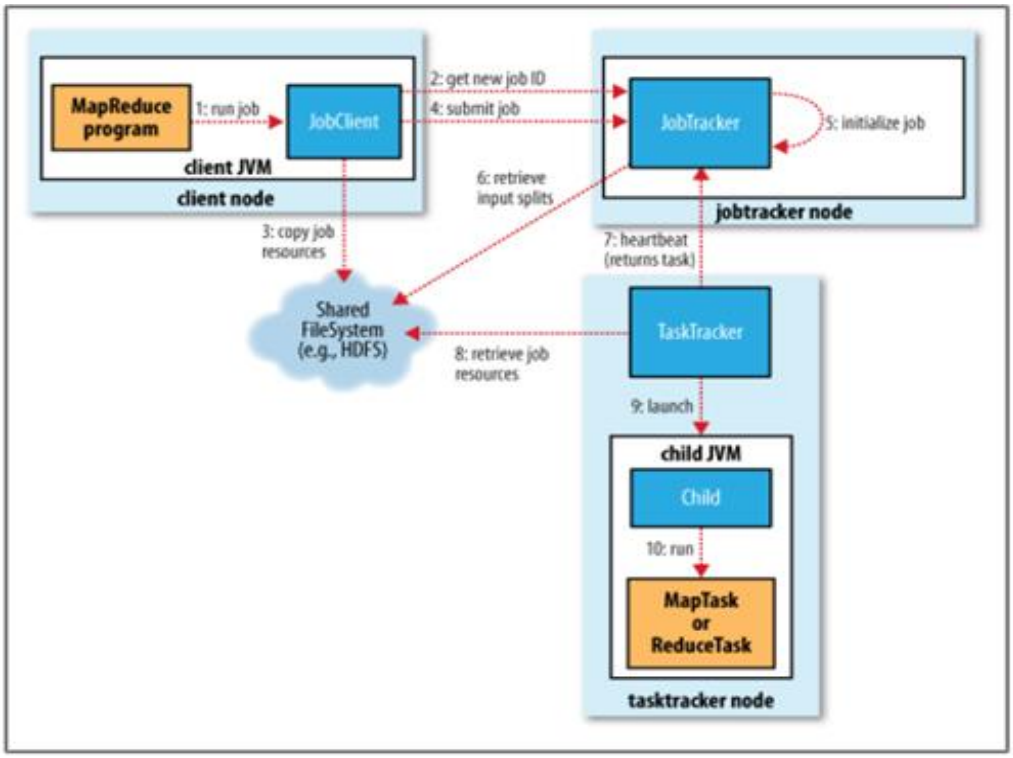

Fig 5:- Classic Map Reduce Job Run Work Flow

- **Task Execution:-** First it identifies the job JAR file by copying it from shared file system to the task trackers file system. Task runner used to run the task.

- **Progress and status updates:-** The important thing for the user to get feedback on how the user gets feedback for job progressing is shown in Fig 7.

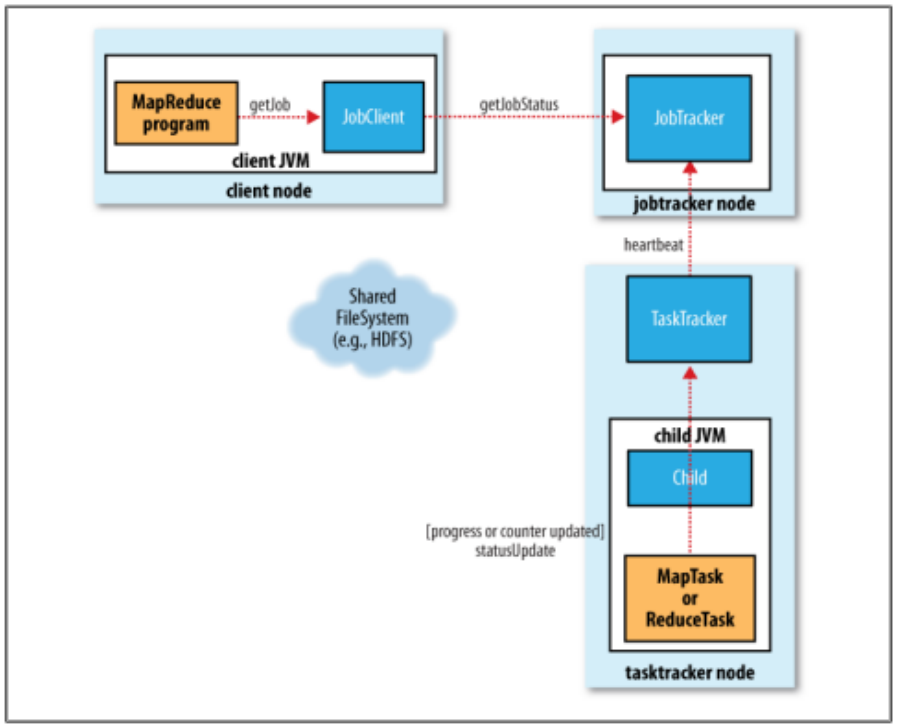

Fig 6:- Progress and status update workflow

- **Job Completion: -** When the job tracker receives a notification that the task for a job is complete it changes the status that the job is successful.

➢ *Yarn (Map Reduce 2)*

Yet another Resource Negotiator or YARN Application Resource Negotiator, job tracker used for job scheduling and task progress monitoring. YARN used to separate two main roles that is independent daemons, a resource manager to manage the use of resource. Entities involved in YARN are the client which can be used to submit the map reduce job.

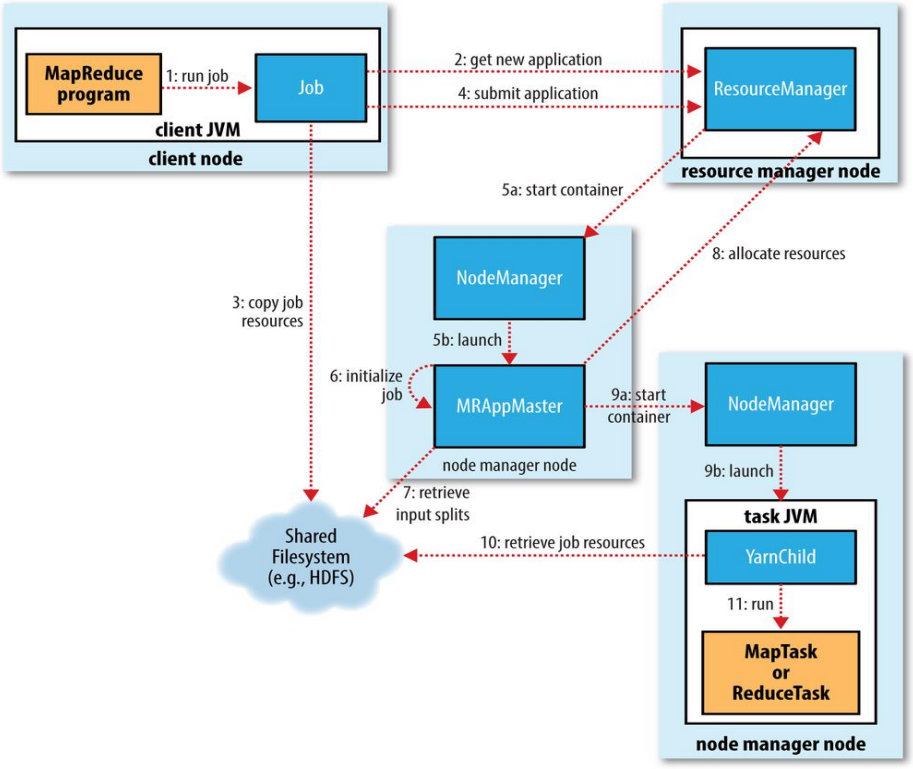

Fig 7:- Yet another Resource Negotiator Map Reduce Job Run Workflow

- **Job Submission**: - The submission of jobs is done using the same user API as Map Reduce. Framework name is set to yarn finally the jobs is submitted by submit Application() method.

- **Task Assignment: -** When the job does not qualify an user task then the application master requests containers for all the map and reduce tasks in the job from the resource manager.. Slots have a maximum memory allowance, utilization when tasks use less memory and YARN resources are more fine grained.

- **Task Execution: -** Once a resource manager assigns a task. The scheduler of the application master starts the container by contacting the node manager, the task is executed by a java application whose main class is Yarn Child. It localizes the resources and runs the map reduce task.

- **Job completion: -** For progress of every five seconds the client checks whether the job has completed for polling the application master.

III. CONCLUSION

As technology are blooming with emerging trends the availability of big data and analytic software have provided a unique data analysis. In this paper as we discussed about most valuable Big data tool like Hadoop, its architecture and operations and also the Map Reduce Job workflow that is functioning efficiently. The era of big data is here and these are truly revolutionary to achieve greater insights in business and technology.

REFERENCES

[1]. Enhanced DTLS with CoAP-based authentication scheme for the internet of things in healthcare application Priyan Malarvizhi Kumar , Usha Devi Gandhi , DOI 10.1007/s11227-017-2169
[2]. Fan Z, Kulkarni P, Gormus S, et al. Smart grid communications: overview of research
[3]. Challenges, solutions, and standardization activities. IEEE Commun Surv Tutor. 2013;15(1):21-38.
[4]. Khurana H, Hadley M, Lu N, Frincke DA. Smart-grid security issues. IEEE Security and Privacy. 2010;8(1):81-85.

[5]. Ericsson GN. Cyber security and power system communication 2014: essential parts of a smart grid infrastructure. IEEE Trans Power Deliv. 2010;25(3):1501-1507.
[6]. Chen PY, Cheng SM, Chen KC. Smart attacks in smart grid communication networks. IEEE Commun Mag. 2012;50(8):24-29.
[7]. Fouda MM, Fadlullah ZM, Kato N, Lu R, Shen X. Towards a light-weight message authentication mechanism tailored for Smart Grid communications. Computer Communications Workshops (INFOCOM WKSHPS). New York: IEEE; 2011:1018-1023.
[8]. Wu D, Zhou C. Fault-tolerant and scalable key management for smart grid. IEEE Trans Smart Grid. 2011;2(2):375-381.
[9]. Xia J, Wang Y. Secure key distribution for the smart grid. IEEE Trans Smart Grid. 2012;3(3):1437-1443.
[10]. Yan Y, Qian Y, Sharif H. A secure data aggregation and dispatch scheme for home area networks in smart grid. Global Telecommunications Conference (GLOBECOM 2011). New York: IEEE; 2011:1-6.
[11]. Lu R, Liang X, Li X, Lin X, Shen X. EPPA: an efficient and privacy-preserving aggregation scheme for secure smart grid communications. IEEE Trans Parallel Distrib Syst. 2012;23(9):1621-1631.
[12]. Wong K-S, Kim MH. Preserving differential privacy for similarity measurement in smart environments. Sci World J. 2014;2014:1-9.
[13]. Hur JB, Koo DY, Shin YJ. Privacy-preserving smart metering with authentication in a smart grid. Appl Sci. 2015;5(4):1503-15