International Journal of Engineering, Science and Humanities
An international peer reviewed, refereed, open-access journal
Impact Factor 7.9 www.ijesh.com ISSN: 2250-3552

Apache Hadoop's Second Generation: Performance and Resource Management through YARN

Vanisha Mavi
Research Scholar, Shobhit Institute of Engineering &, Technology (Deemed to be University), Meerut (UP)
Vanisha.mavi@gmail.com
Nidhi Tyagi
Professor, CSE, Shobhit Institute of Engineering & Technology, (Deemed to be University), Meerut (UP)

ABSTRACT
The rapid expansion of digital technologies has resulted in an enormous growth of data generated by individuals, organizations, and technological systems. This large and complex collection of data is commonly referred to as Big Data. Big Data typically involves datasets that are too large, fast-growing, or complex to be processed efficiently using traditional data management systems and conventional analytical tools. As a result, organizations increasingly rely on advanced technologies capable of storing, managing, and analysing such massive volumes of information.
One of the most widely used technologies designed to address Big Data challenges is Hadoop. Hadoop is an open-source framework that enables the distributed storage and processing of large datasets across clusters of computers. Its architecture allows data to be stored and processed in parallel, thereby improving scalability, reliability, and efficiency. A key component of Hadoop is the MapReduce programming model, which divides large data processing tasks into smaller subtasks that can be executed simultaneously across multiple computing nodes. This distributed approach significantly enhances the speed and efficiency of data processing.
Furthermore, the Hadoop ecosystem has evolved with the introduction of YARN (Yet Another Resource Negotiator), which serves as the next generation resource management platform for Hadoop. YARN separates resource management from data processing, enabling multiple data processing frameworks to run on the same cluster. This paper provides an overview of Big Data, Hadoop architecture, the MapReduce model, and the role of YARN in modern data processing systems.

Keywords- Big Data, Hadoop, MapReduce, YARN (Yet another Resource Negotiator).

1. INTRODUCTION

Big Data is the large and complex data that is difficult to use the traditional tools to store, manage, and analyze in an acceptable duration. Big Data needs a new processing model which

International Journal of Engineering, Science and Humanities
An international peer reviewed, refereed, open-access journal
Impact Factor 7.9 www.ijesh.com ISSN: 2250-3552

has the better storage, decision-making and analyzing abilities. This is the reason why Big Data technology born. The term big data described as a collection of structured, unstructured and semi structured data [1]. The Big Data technology provides a new way to extract, interact, integrate, and analyze of Big Data the Big Data strategy is aiming at mining the significant valuable data information behind the Big Data by specialized processing. Big Data consists a large dataset that cannot be managed effectively by the common Database Management System (DBMS). These data sets range from terabyte to Exabyte.

In recent years, we have been drowning in the ocean of data that was produced by the development of the Internet, the mobile Internet, the Internet of the things and the social networks. A photo that uploaded on Instagram is about 1mb; a video that uploaded to YouTube is about dozens of mega sizes. Chatting online, browsing websites, playing online games, and shopping online will also turn into data that may be stored in any corner in the world. According to a report of IBM, there are 2.5 quintillion bytes of data that we create every day. Ninety percent of the data was created in the last two years [2].

2. HADOOP

Hadoop is a distributed system infrastructure researched and developed by the Apache Foundation. The users can develop the distributed applications although they do not know the lower distributed layer so that the users can make full use of the power of the cluster to perform the high-speed computing and storage. The core technology of Hadoop is the Hadoop Distributed File System (HDFS) and the MapReduce [3]. HDFS provides the huge storage ability while MapReduce provides the computing ability of the Big Data. Since HDFS and MapReduce have become open source, their low cost but high processing performance helped them to be adopted by many enterprises and organizations. With the popularity of the Hadoop technologies, there are more tools and technologies which are developed on the basis of the Hadoop framework.

2.1 Relevant Hadoop Tools

The core technologies are HDFS and MapReduce, Chukwa, Hive, Hbase, Pig, Zookeeper and so on are also indispensable. They provide the complementary services and higher-level service on the core layer [4].

- MapReduce is a programming model for parallel computing on the large-scale data sets.
- HDFS is a distributed file system. Because is high fault-tolerant.
- Chukwa is an open-source data collection system which is used for data monitoring and analysis. Chukwa is built on the HDFS and MapReduce framework. Chukwa stores the data by HDFS and relies on the MapReduce to manage the data. Chukwa is a flexible and powerful tool to display, monitoring, and analyze the data.
- Hive was originally designed by Facebook. It is a data warehouse based on the Hadoop which

International Journal of Engineering, Science and Humanities
An international peer reviewed, refereed, open-access journal
Impact Factor 7.9 www.ijesh.com ISSN: 2250-3552

provides data sets searching, special query, and analysis. Hive is a structured data mechanism which supports the SQL query languages like RDBMS to help those users who are familiar with SQL queries. This kind of query language is called Hive QL.

- Hbase is a distributed and open-source database.
- Pig is a platform which was designed for the analysis and evaluation of the Big Data. The significant advantage of its structure is that it can afford the highly parallel test which is based on the parallel computing.
- Zookeeper is an open-source coordination service for the distributed applications. It is used mainly to provide users' synchronization, configuration management, grouping, and naming services.

3. MAPREDUCE

MapReduce is a standard functional programming model. This kind of model has been used in the early programming languages, such as Lisp [5]. The core of the calculation model is that can pass the function as the parameter to another function. Through multiple concatenations of functions, the data processing can turn into a series of function execution. MapReduce has two stages of processing. The first one is Map and the other one is Reduce. The reason why the MapReduce is popular is that it is very simple, easy to implement, and offers strong expansibility. MapReduce is suitable for processing the Big Data because it can be processed by the multiple hosts at the same time to gain a faster speed.

3.1 MapReduce Architecture

The MapReduce operation architecture includes the following three basic components [6]:-

- Client: Every job in the Client will be packaged into a JAR file which is stored in HDFS and the client submits the path to the job Tracker.
- Job Tracker: Job Tracker is a master service which is responsible for coordinating all the jobs that are executed on the MapReduce. When the software is on, the Job Tracker is starting to receive the jobs and monitor them. The functions of MapReduce include designing the job execution plan, assigning the jobs to the task Tracker, monitoring the tasks, and redistributing the failed tasks.
- Task Tracker: The task Tracker is a slave service which runs on the multiple nodes. It is in charge of executing the jobs which are assigned by the job Tracker. The task Tracker receives the tasks through actively communicating with the job Tracker.

3.2 Limitations of MapReduce

Although MapReduce is popular all over the world, most people still have realized the limits of the MapReduce. There are following the four main limitations of the MapReduce [7]:

- **The bottleneck of Job Tracker**: -the Job Tracker should be responsible for jobs allocation,

International Journal of Engineering, Science and Humanities
An international peer reviewed, refereed, open-access journal
Impact Factor 7.9 www.ijesh.com ISSN: 2250-3552

management, and scheduling. In addition, it should also communicate with all the nodes to know the processing status. It is obvious that the Job Tracker which is unique in the Map Reduce, tasks too many tasks. If the number of clusters and the submission jobs increase rapidly, it will cause network bandwidth consumption. As a result, the Job Tracker will reach bottleneck and this is the core risk of MapReduce.

- **The Task Tracker**: -Because the jobs allocation information is too simple, the Task Tracker might assign a few tasks that need more sources or need a long execution time to the same node. In this situation, it will cause node failure or slow down the processing speed [8].
- Jobs Delay: - Before the MapReduce starts to work, the Task Tracker will report its own resources and operation situation. According to the report, the Job Tracker will assign the jobs and then the Task Tracker starts to run. As a consequence, the communication delay may make the job Tracker to wait too long so that the jobs cannot be completed in time.
- Inflexible Framework: - Although the MapReduce currently allows the users to define its own functions for different processing stages, the MapReduce framework still limits the programming model and the resources allocation.

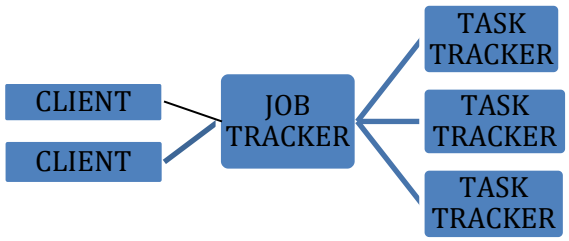


Figure 1: Architecture of MapReduce.

4. YARN

YARN is a key element of the Hadoop data processing architecture that provides different data handling mechanism, including interactive SQL (Structured query language) and batch processing. In YARN, the Resource Manager will be the resources distributor while the Application Master is responsible for the communication with the Resource Manager and cooperate with the Node-manager to complete the tasks [9]. Yarn can be considered as operating system of Hadoop ecosystem. It improves the performance of data processing in Hadoop by separating the resource management and scheduling capabilities of map reduce from its data processing components.

YARN became a sub-project of the larger Apache Hadoop project. Sometimes called Map Reduce 2.0, YARN is a software rewrite that decouples MapReduce's resource management and

International Journal of Engineering, Science and Humanities
An international peer reviewed, refereed, open-access journal
Impact Factor 7.9 www.ijesh.com ISSN: 2250-3552

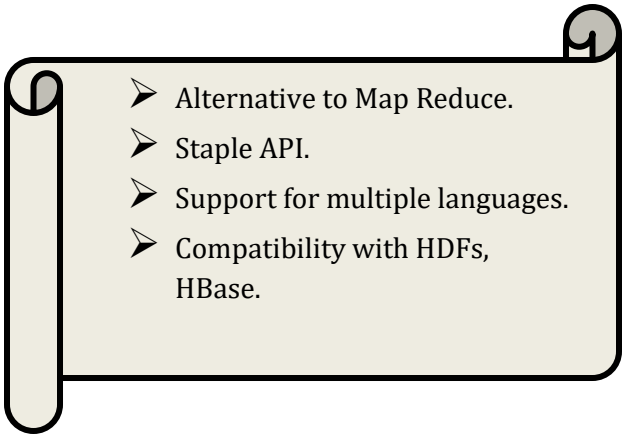
scheduling capabilities from the data processing component, enabling Hadoop to support more varied processing approaches and a broader array of applications [10]. For example, Hadoop clusters can now run interactive querying and streaming data applications simultaneously with MapReduce batch jobs. The original incarnation of Hadoop closely paired the Hadoop Distributed File System (HDFS) with the batch-oriented MapReduce programming framework, which handles resource management and job scheduling on Hadoop systems and supports the parsing and condensing of data sets in parallel [11].

YARN combines a central resource manager that reconciles the way applications use Hadoop system resources with node manager agents that monitor the processing operations of individual cluster nodes. Running on commodity hardware clusters, Hadoop has attracted particular interest as a staging area and data store for large volumes of structured and unstructured data intended for use in analytics applications. Separating HDFS from MapReduce with YARN makes the Hadoop environment more suitable for operational applications that can't wait for batch jobs to finish.

The designers have put forward the next generation of MapReduce: YARN. Given the limitations of MapReduce, the main purpose of YARN is to divide the tasks for the Job Tracker. In YARN, the Resources are managed by the Resource Manager and the jobs are traced by the Application Master. The Task Tracker has become the Node Manager. Hence, the global Resource Manager and the local Node Manager compose the data computing framework. In YARN, the Resource Manager will be the resources distributor while the Application Master is responsible for the communication with the Resource Manager and cooperate with the Node Manager to complete the tasks [12].

4.1 YARN architecture

Compared with the old MapReduce Architecture, it is easy to find out that YARN is more structured and simpler. Then, the following section will introduce the YARN architecture.



International Journal of Engineering, Science and Humanities
An international peer reviewed, refereed, open-access journal
Impact Factor 7.9 www.ijesh.com ISSN: 2250-3552

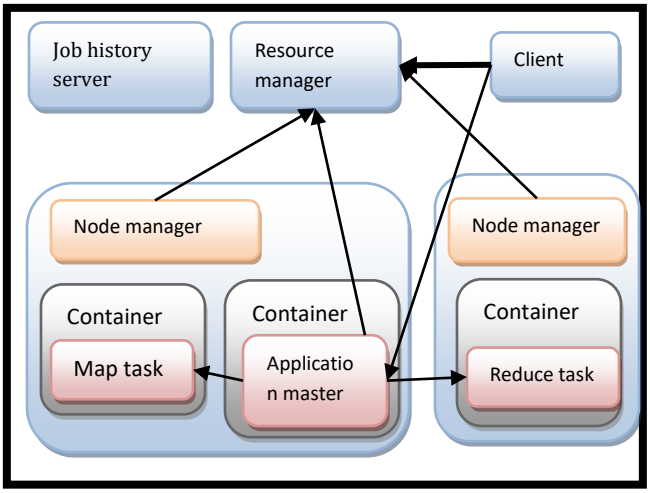


Figure 2: YARN architecture

There are following four core components of the YARN Architecture [13]**:**

- **Resource Manager**
  According to the different functions of the Resource Manager, is a designer has divided it into two lower-level components: The Scheduler and the Application Manager. On the one hand, the Scheduler assigns the resource to the different running applications based on the cluster size, queues, and resource constraints. The Scheduler is only responsible for the resources allocation but is not responsible for the monitoring the application implementation and task failure. On the other hand, the Application Manager is in charge of receiving jobs and redistributing the containers for the failure objects.
- **Node Manager**
  The Node Manager is the frame proxy for each node. It is responsible for launching the application container, monitoring the usage of the resource, and reporting all the information to the Scheduler.
- **Application Master**
  The Application Master is cooperating with the node Manager to put tasks in the suitable containers to run the tasks and monitor the tasks. When the container has errors, the Application Master will apply for another resource from the Scheduler to continue the process.
- **Container**
  In YARN, the Container is the source unit which is the available node splitting the organization resources. Instead of the Map and Reduce source pools in MapReduce, the Application Master can apply for any numbers of the Container.

International Journal of Engineering, Science and Humanities
An international peer reviewed, refereed, open-access journal
Impact Factor 7.9 www.ijesh.com ISSN: 2250-3552

Due to the same property Containers, all the Containers can be exchanged in the task execution to improve efficiency.

4.2 Advantages of YARN Compared to the MapReduce [14]

- YARN greatly enhances the scalability and availability of the cluster by distributing the tasks to the job Tracker. The Resource Manager and the Application Master greatly relieves the bottleneck of the job Tracker and the safety problems in the MapReduce.
- In YARN, the Application Master is a customized component. That means that the users can write their own program based on the programming model. This makes the YARN more flexible and suitable for wide use.
- YARN, on the one hand, supports the program to have a specific checkpoint. It can ensure that the Application Master can reboot immediately based on the status which was stored on HDFS. On the other hand, it uses the Zookeeper on the Resource Manager to implement the failover. When the Resource Manager receives errors, the backup Resource Manager will reboot quickly. These two measures improve the availability of YARN.

The cluster has the same Containers are the Reduce and Map pools in MapReduce. Once there is a request for resources, the Scheduler will assign the available resources in the cluster to the tasks and regard the resource type [15]. It will increase the utilization of the cluster resources.

4.3 YARN's Requirements

- **Scalability:** In Hadoop 1.0, job Tracker is becoming bottleneck due to too much responsibility like resource management, application management. You want your architecture to be scalable.
- **Multitenancy:** Multitenancy means serving multiple tenants on the same platform. This is very desirable requirements for any cloud environment.
- **Serviceability:** Hadoop released the new feature nearly every three months. This requirement talks about serviceability which allows users to run job on older version and allow developer to test new feature easily.
- **Locality awareness:** Application wants task to schedule on container which is closed to its input data in HDFS.
- **High cluster utilization:** Generally, application will request for resources larger than their need and hold on the resources which will decrease the utilization.
- **Reliability/Availability:** This equipment talks about reliably allocate resources and tracks their availability.
- **Secure and auditable operation:** Multi tenant feature of Hadoop will require secure and auditable resource allocation for better isolation between different tenants.

International Journal of Engineering, Science and Humanities
An international peer reviewed, refereed, open-access journal
Impact Factor 7.9 www.ijesh.com ISSN: 2250-3552

- **Support for Programming Model Diversity:** MapReduce is widely used but it is not idle for all the large-scale computation. So YARN needs Support for Programming Model Diversity.
- **Flexible Resource Model:** The need for the resource can be dynamic depending on the tasks progress, so you need flexible resource model.
- **Backward compatibility:** Ecosystem of Hadoop 1.0 has been huge and you want them to run on YARN, so we need backward compatibility.

5. CONCLUSIONS

In this paper YARN provides:

- Greater scalability
- Higher efficiency
- And enables a large number of different frameworks to efficiently share a cluster.

These claims are substantiated both experimentally (via benchmarks), and by presenting a massive-scale production experience of Yahoo!—which is now 100% running on YARN. Finally, we tried to capture the great deal of excitement that surrounds this platform, by providing a snapshot of community activity and by briefly reporting on the many frameworks that have been ported to YARN. This paper says, YARN can serve as both a solid production framework and also as an invaluable playground for the research community.

REFERENCES

1. Vinod Kumar, Vavilapall, Arun C Murthy, Chris Douglas, Sharad Agarwali, Mahadev Konar, Robert Evans, Thomas Graves, Jason Lowe, Hitesh Shah, Siddharth Seth, Bikas Saha, Carlo Curino, Owen O'Malley, Benjamin Reed,Eric Baldeschwieler, "Apache Hadoop YARN: Yet Another Resource Negotiator", SoCC'13, 1–3 Oct. 2013, Santa Clara, California, USA, ACM978-1-4503-2428-1.
2. Jenifer Jothi Mary1, Dr. L. Arockiam2","A Study on Basic Concepts of Big Data".
3. Apache tez. http://incubator.apache.org/projects/tez.html.
4. Amogh Pramod Kulkarni1, Mahesh Khandewal2", "Hadoop and Introduction to YARN", International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 4, Issue 5, May 2014).
5. J. Dean and S. Ghemawat", "MapReduce: simplified data processing on large clusters", "Commun. ACM,51(1):107–113, Jan. 2008."
6. "Hortonworks Hadoop YARN. http://hortonworks.com/hadoop/yarn/.

7. "Ibrahim Abaker Targio Hashem,Nor Badrul Anuar, Abdullah Gani, Ibrar Yaqoob, Feng Xia, Samee Ullah Khan", "MapReduce: Review and open challenges", Scientometrics ,DOI 10.1007/s11192-016-1945-y, 1 February 2016.
8. "Kiejin Park and Limei Peng", "A Design of High-speed Big Data Query Processing System for Social Data Analysis", International Journal of Applied Engineering Research ISSN 0973-4562 Volume 11, Number 14 (2016) pp 8221-8225.
9. "Arinto Murdopo, Jim Dowling", "Next Generation Hadoop: High Availability for YARN".
10. "Kala Karun. A, Chitharanjan.K", "A Review on Hadoop–HDFS", Infrastructure Extensions, Proceedings of 2013 IEEE Conference on Information and Communication Technologies (ICT 2013).
11. " Smita Konda, Rohini More", "Big Data in HDFS with Zookeeper and Flume", International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 02 Issue: 09 | Dec-2015 www.irjet.net p-ISSN: 2395-0072.
12. "Ashish Sharma, Snehlata Vyas", "Hadoop2 Yarn", IPASJ International Journal of Computer Science (IIJCS), Volume 3, Issue 9, September 2015.
13. "Harshawardhan S. Bhosale, Prof. Devendra P. Gadekar", "A Review Paper on Big Data and Hadoop", International Journal of Scientific and Research Publications, Volume 4, Issue 10, October 2014.
14. "Jenifer Jothi Mary1, Dr. L. Arockiam2", "A Study on Basic Concepts of Big Data", International Journal of Emerging Trends in Computing and Communication Technology, Volume 1, No 3, August 2015 ISSN: 2348 4454.
15. "Avita Katal, Mohammad Wazid, and R H Goudar", "Big Data: Issues. Challenges, Tools and Good Practices", 2013 IEEE.
16. "Firat Tekiner and John A. Keane", "Big Data Framework", International Conference on System, 2013 IEEE.
17. "S. Loughran, D. Das, and E. Baldeschwieler", "Introducing Hoya–HBase on YARN", http://hortonworks.com/blog/introducing-hoya-hbase-on-yarn/2013.