HADOOP: A Comparative Study between Single-Node and Multi-Node Cluster

Elisabeta ZAGAN[1], Mirela DANUBIANU[2]
Stefan cel Mare Unive1rsity of Suceava, 720229, Romania
Integrated Center for Research, Development and Innovation in Advanced Materials
Nanotechnologies, and Distributed Systems for Fabrication and Control (MANSiD)
Stefan cel Mare University, Suceava, Romania

***Abstract*—Data analysis has become a challenge in recent years as the volume of data generated has become difficult to manage, therefore more hardware and software resources are needed to store and process this huge amount of data. Apache Hadoop is a free framework, widely used thanks to the Hadoop Distributed Files System (HDFS) and its ability to relate to other data processing and analysis components such as MapReduce for processing data, Spark - in-memory Data Processing, Apache Drill - SQL on Hadoop, and many other. In this paper, we analyze the Hadoop framework implementation making a comparative study between Single-node and Multi-node cluster on Hadoop. We will explain in detail the two layers at the base of the Hadoop architecture: HDFS Layer with its deamons NameNode, Secondary NameNode, DataNodes and MapReuce Layer with JobTrackers, TaskTrackers daemons. This work is part of a complex one aiming to perform data processing in Data Lake structures.**

Keywords*—Hadoop; HDFS; single-node cluster; multi-node cluster; namenode; secondary namenode; datanodes; jobtracker; tasktrackers

I. INTRODUCTION

Before the term Big Data, appeared about 15 years ago, there were few possibilities to process terabytes of data sets or higher. Once data generation capacity has increased, needs to store and process this data appeared. The large volume of data has brought with it the need for significant changes in the architecture of storage and processing systems. Over the years, several hardware high-end solutions have been found and implemented, which were smart but very expensive.

Starting from 2003, Google developed a new technology having two main components: Google File System (GFS) [1] and MapReduce [2].

Google created a platform on which multiple data management applications could be implemented. At the same time, Doug Cutting had begun working on a new open-source implementation based on the ideas suggested by Google, so Hadoop was born. [3].

Hadoop is a distributed processing software framework that can process both small and large volumes of data across clusters of computers. However, it is recommended for large data sets, because it is able to scale-up from a single server to hundreds.

Over time, Hadoop has consolidated its position through some advantages, such as:

- All data are accessible, therefore is no need to archive/clean data before storage because Hadoop allows increasing storage space without limits.
- Is a scalable architecture that allows the addition of new clusters/servers to the original architecture without limits. There is no need to upgrade the server but you can simply add new clusters to increase storage capacity.
- Is a robust and easy-to-use architecture that can be easily configured by modifying the config file which is usually an excel file.

Hadoop is based on the Hadoop Distributed File System (HDFS) that allows data to be distributed to multiple nodes. Thus, data are read in parallel and the time required for this operation is substantially reduced. Another important specific feature of Hadoop is that it is based on the "write once and read many times" technology. Even if the writing process will take longer, the reading process makes an important contribution to reduce the read/analyze data time.

In the following, we will analyze Hadoop architecture and we will compare its two ways of implementing: Single-node cluster and Multi-node cluster. This work is structured as follows. After the brief introduction discussed in Section I, Section II addresses some related works, and Section III presents the Hadoop architecture. Section IV outlines the two ways of setting up Hadoop and are highlighted the differences between them. Section V is intended for final conclusions.

II. RELATED WORKS

According to the latest research in the field on Hadoop, it can be said that this is a robust and easy-to-use architecture, it can be easily configured by modifying the *config* file. It is also a scalable architecture that allows new clusters/servers to be added to the original architecture without the need for further upgrades.

In [4] the authors present the Hadoop Single-node cluster installation and setup, and also the required software used in their exemplification. They implemented their cluster on Hadoop Version 2.7.2 using *Jdk- 1.7* as java version.

Based on the definitions found in specialty articles, HDFS and MapReduce is a scalable and error-tolerant model that hides all complexities for Big Data analysis. Thus, in Article [5] Hadoop and its components, which include MapReduce and HDFS, are discussed in detail.

The paper [6] presents a methodology based on the reference analysis to guide the implementation of the Hadoop cluster. The authors analyzed local and cloud architectures using centralized and geographically distributed servers. The results of the research done by the authors show that the methodology can be applied dynamically based on different architectures. At the same time, the authors state that the acquired knowledge can be used to improve the data analysis process by understanding Hadoop architecture.

In [7] the authors give a brief description of the Hadoop ecosystem and several components like Pig, Hive, Mahout, Qooop, Hbase, Flume. The authors also implemented a Hadoop-based platform for the analysis of collected tweets. The obtained results are then transferred to graphic charts.

III. Hadoop Architecture

Hadoop is based on master/slaves architecture. Thus, in such architecture, there is a master and more slaves, where the master manages all Hadoop activities by hosting the NameNode and the JobTracker/MapReduce, and the slaves are intended to store the data hosting the DataNodes/HDFS and the TaskTracker/MapReduce.

The following components are parts of Hadoop:

1) HDFS layer: Storage layer based on a master/slave architecture.

a) NameNode (master daemon).

b) Secondary NameNode.

c) DataNodes (slave daemon).

2) MapReduce layer: Data processing layer-based also on a master/slave architecture.

a) JobTrackers (master daemon).

b) TaskTrackers (slave daemon).

HDFS and MapReduce is a scalable and fault-tolerant model that manages all complex processes of BigData analytics. Hadoop distributed file system - HDFS is a Java-based distributed file system that allows the storage of a large volume of data across multiple nodes in a Hadoop cluster. Using HDFS provides multiple benefits such as distributed storage, distributed and parallel computation and horizontal scalability. On the other hand, MapReduce is a programming framework that allows performing distributed and parallel processing on a large volume of data in a distributed environment. MapReduce framework uses JobTracker and TaskTracker to monitor and execute tasks.

Fig. 1 shows the master/slave Hadoop architecture graphically representing the two essential components and their interconnections.

NameNode stores system metadata and manages clients' requests. When data is stored in HDFS, the NameNode is the one that will keep all the storage information. NameNode is actually the master in the master/slaves architecture, which is why it is also called master-node. NameNode is the main node of the architecture, and if this node fails then the entire Hadoop system will fail. Physically the master node will need the best hardware resources because that's where all the metadata will be kept.

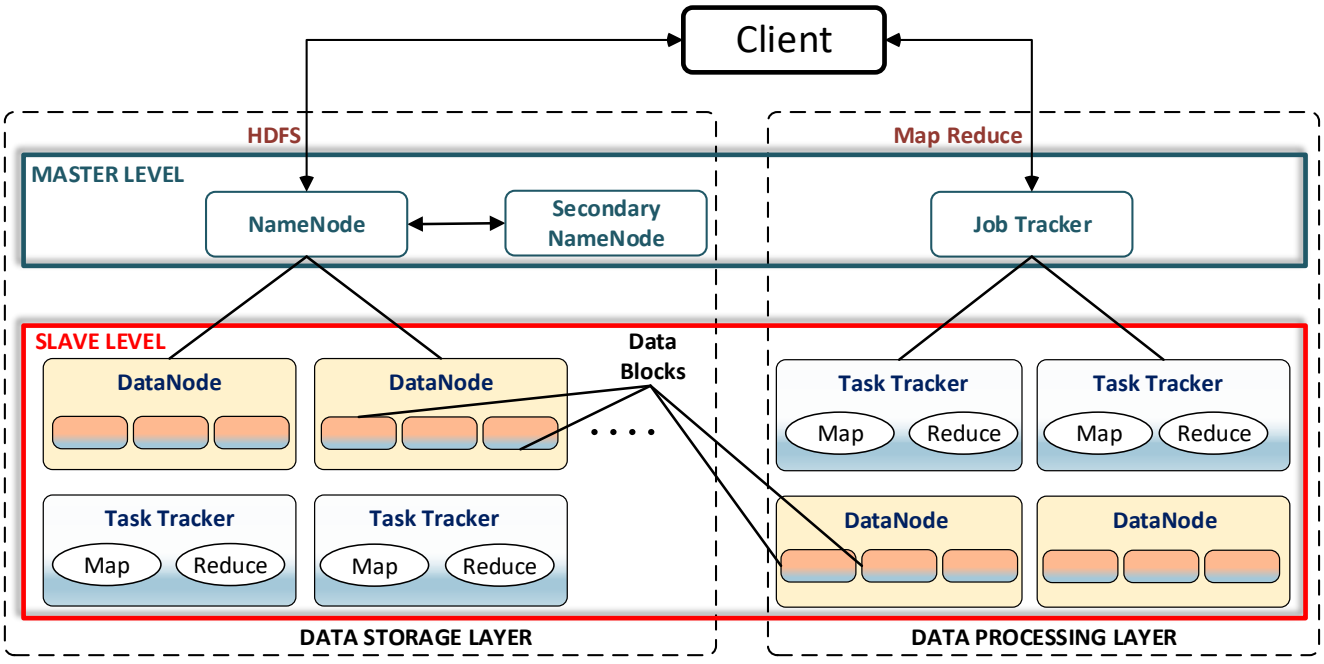

Fig. 1. Hadoop Master/Slave Architecture.

Metadata are data sets that have crucial information about system files: lists of files and folders in HDFS, lists of blocks and where they are stored, information about data permissions (read, write, run), access times, etc.

There are two different types of metadata: FSimages and EditLogs. EditLogs hosts all logs generated by the master node throughout its operation. Any actions and changes that occur in HDFS will be saved in this EditLogs. FSimage are metadata that will be saved in the RAM of the master node and hosts the image of the system files.

Let's assume that the master node has failed, so we need to restart the entire system. When restarting, the master node must reload the image from FSimage and upload all data saved in EditLogs. The larger the log files is, the longer the restart time will be, it can last even tens of minutes, which is not the desired time to wait for a restart. To faster this restart procedure, a second master node named Secondary NameNode has been created, which does nothing but query the logs in EditLogs and update FSimage with the new data. Thus, if the master node fails, it will access the updated FSimage files, reducing the restart time. It is very important not to confuse the function of the Secondary NameNode as a true copy of the NameNode. The Secondary NameNode serves only to update the FSimage in order to reduce the restart time of the main master node in case of a failure. In other words, Secondary NameNode is used to create restart points of the main master node.

In Hadoop, the data are stored in blocks, and these blocks are saved in DataNodes also called SlaveNodes. These DataNodes send information signals once every 3 seconds to the master node, so NameNode can be seen as the heart of the entire system that pulsates and receives information continuously. These signals constantly sent, inform the master node that the data is ready at any time to be accessed. All system operations that take place on a DataNode will be constantly sent to the master node.

As we already saw, within Hadoop the data is stored in HDFS files, and the data is broken into individual blocks. The standard size of a block in Hadoop is 128Mb, but this size can be easily changed from the config file. Therefore, every file that is sent to DataNodes, will be saved in individual blocks of 128Mb. If, for example, there is a 130Mb file that needs to be saved, then it will occupy 128Mb in one block and in another block only 2Mb. This is how data are stored in blocks at the level of HDFS files in Hadoop. The number of blocks is given by the file size divided by the size of the data block. One of the main features of HDFS is that, once data blocks are stored in DataNodes, they will be automatically replicated in different DataNodes to provide fault tolerance. The replication number, in general, should be set up to 3 in the config file.

JobTracker runs on master node level and keeps track of all tasks that are serving DataNodes, known as TaskTrackers. Within this one master and multiple slaves architecture, we will have a single JobTracker and several TaskTrackers. JobTracker is an essential daemon for MapReduce and his job consists in receiving client requests for MapReduce, communicate with the NameNode to determine the location of the data, finds the best TaskTracker nodes to execute tasks based on the data locality and node availability, monitors the TaskTrackers independently, inform the client about the job status. When JobTracker is down, HDFS will continue to work, but MapReduce cannot be started because of the MapReduce jobs that are halted.

TaskTrackers are designed to keep track of all actions that are running on DataNodes, sending the information back to JobTracker. TaskTrackers will be assigned by JobTracker to execute Mapper and Reducer tasks on all DataNodes (Fig. 2). TaskTracker will constantly inform the JobTracker about the progress of the tasks in execution. When a TaskTracker no longer responds, then JobTracker has the ability to assign the task performed by the faulty TaskTracker to another node.

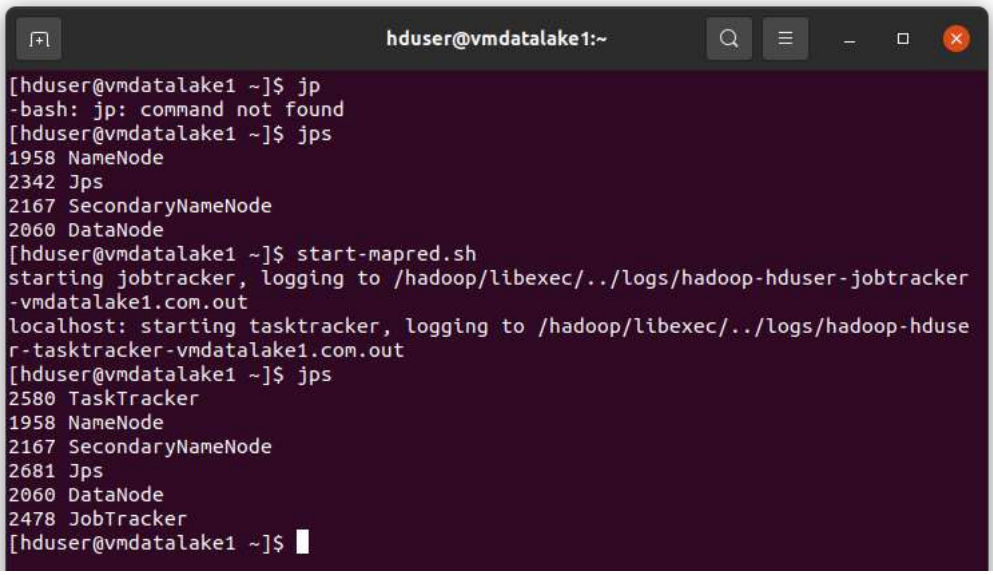

Fig. 2. Single-Node Cluster Daemons.

IV. HADOOP: SINGLE-NODE VERSUS MULTI-NODE CLUSTER

There are two ways to setup a Hadoop architecture: Single-node cluster and Multi-node cluster. In the following, we will perform a comparative study between these two approaches.

For the practical implementation, we used a PC with Ubuntu Desktop 20.04.1 LTS operating system on which we installed Hadoop 3.30. We used VMware Workstation 16 Pro to create virtual machines on which we installed an older version of Centos - Centos-6.3 which is known as a very stable version for this purpose. The purpose of this work is not to explain how to setup and configure, which can be found in [8], [9], but is intended to exemplify the two methods of implementing Hadoop, and to highlight the differences between them.

A. Single-Node Cluster

Single-node cluster is a method of implementing and setting all daemons on a single virtual machine. This setup method is generally used for the study and the test phase, or in environments with fewer data, but in this case, maybe Hadoop is not the most recommended technology to store data. If the volume of data is not large enough then you can hardly notice the main advantages of Hadoop from the first place. After installing, configuring, and starting ssh processes of a Single-node cluster, launching the *jps* command, which is a java virtual machine process status tool, we can see the status of all Hadoop daemons like NameNode, Secondary NameNode, JobTracker, TaskTracker, DataNodes that are currently running on the machine. As we can see in Fig. 2, they all run on the same virtual machine that we name vmdatalake1.

B. Multi-Node Cluster

Setting Hadoop in Multi-node cluster involves the use of more than one virtual machine VM. Each data node runs basically on a different VM. This type of configuration is used in organizations to analyze BigData. In the real environment when you are dealing with petabytes of data, this data must be distributed in hundreds of machines to be processed in real-time. Next, we can see the implementation of a Multi-node cluster using the same system environment as in Single-node cluster. We configured six virtual machines (Fig. 2) by cloning the VM from the Single-Node cluster, each of which we set it up after for specific purposes.

By running the same java jps command on each VM you can see which daemons are running on each dedicated VM, where NN corresponds to the VM dedicated to the NameNode, JT to JobTracker, SNN to Secondary NameNode, DNn to DataNodes: DN1, DN2, DN3. Fig. 3 is a capture launched from another computer connected to the same LAN over the MobaXterm app by using Write commands on all terminals.

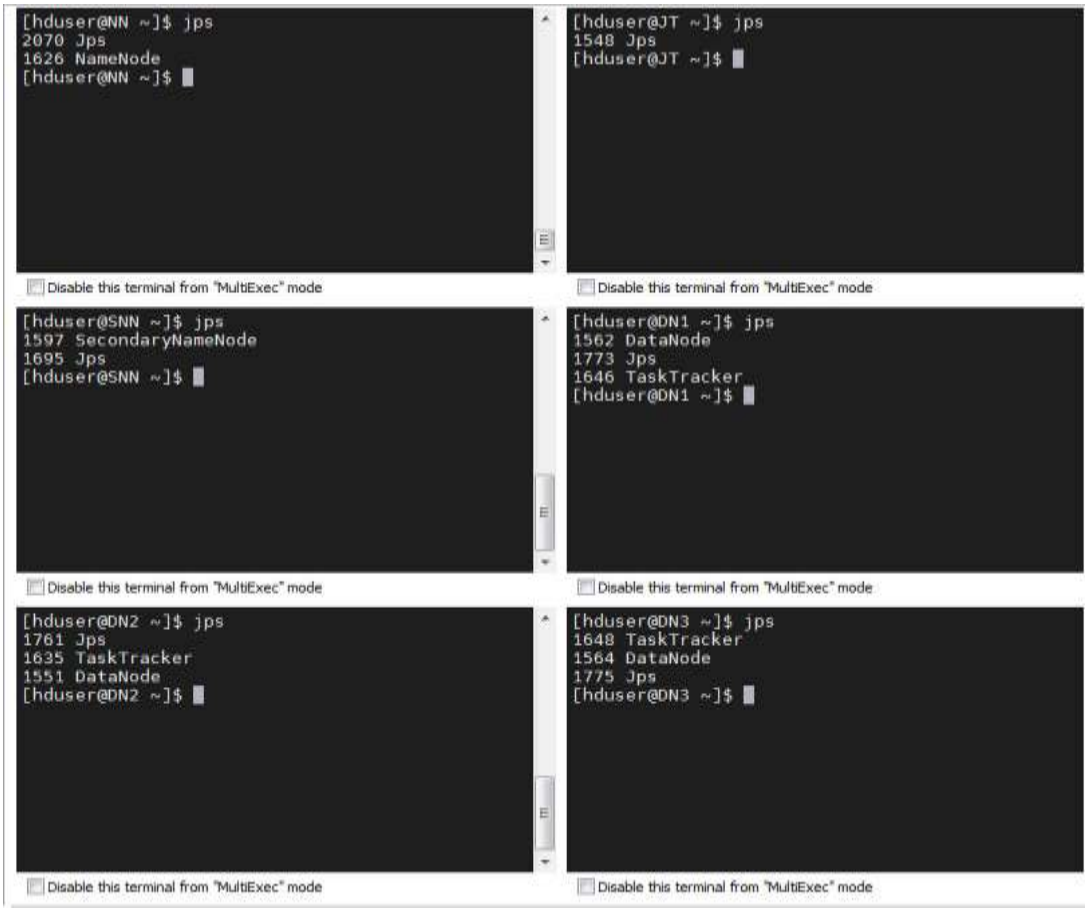

Fig. 3. Multi-Node Cluster Daemons.

By running the command: *hadoop dfsadmin -report* [10] under the NameNode as the HDFS superuser *hduser* in our case, it can be checked if everything is working well. This is one of the first commands that it should be learned as a Hadoop admin, it is one of the commands used frequently to obtain a full report about your data nodes.

Fig. 4 is a report capture over the VMs configured above and we can see: what is the cluster capacity, how much DFS is used and how much is remaining in every single machine, what is the entire cluster capacity, what are the corrupted blocks.

Table I shows a comparison between Hadoop Single-node cluster and Multi-node cluster.

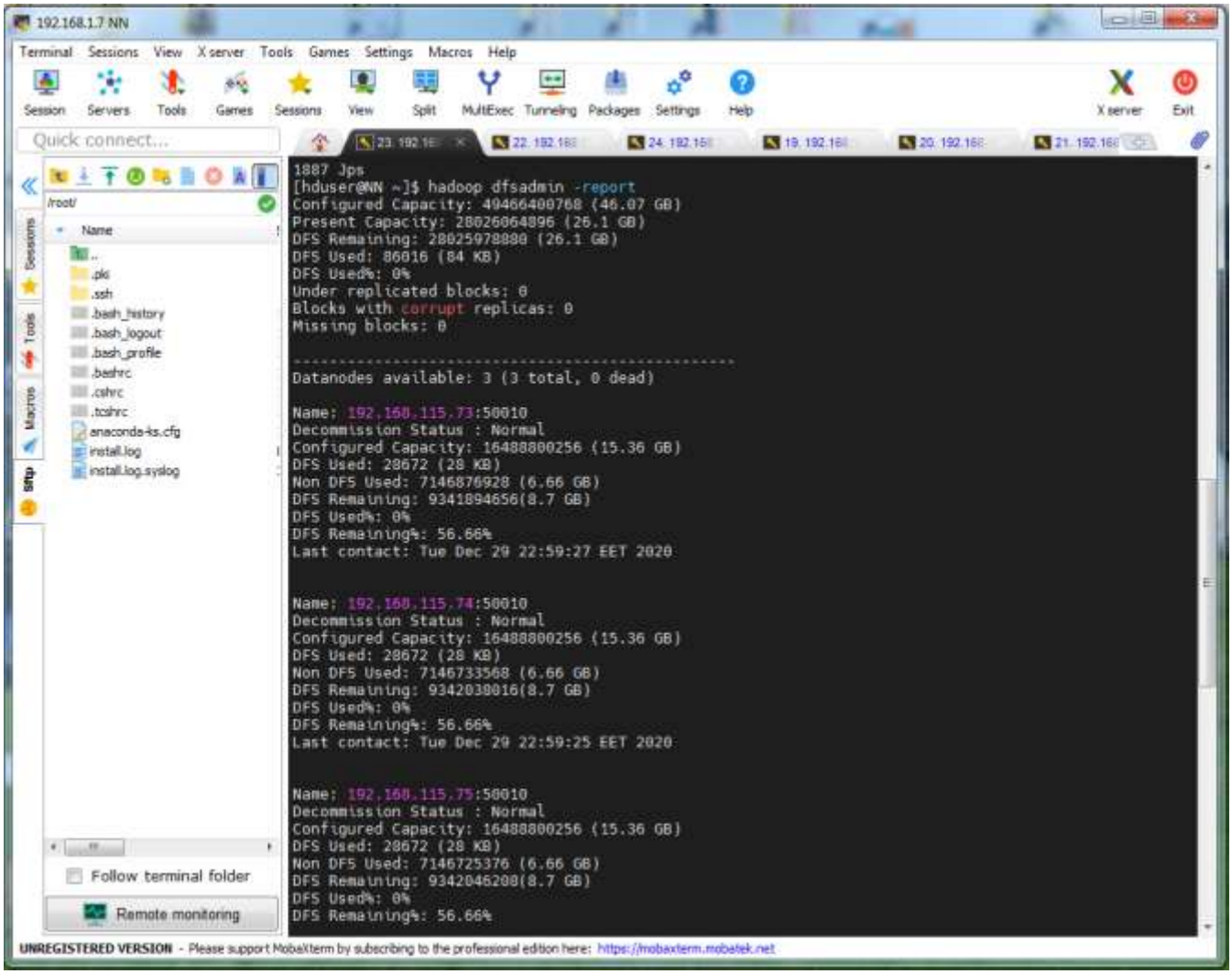

Fig. 4. Hadoop Multi-node Cluster Report.

TABLE I. SINGLE-NODE CLUSTER VERSUS MULTI-NODE CLUSTER

Single-node cluster	**Multi-node cluster**
Hadoop is installed on a single machine or data node.	Hadoop is installed on multiple data nodes runging from a few to hundreds of nodes on a LAN network.
All Hadoop daemons NameNode, DataNode, Secondary NameNode, JobTracker, TaskTracker runs on one single machine.	In distributed mode, NameNode, DataNode, Secondary NameNode, JobTracker, TaskTracker run on different machines.
The replication factor is one in the Single-node cluster.	In Multi-node cluster the replication factor will be greater than one and it should be installed in more than one machine.
Predominantly used in the testing and study phases. It is used also in basic tests.	Used within BigData-type high-volume organizations.
Used to run simple MapReduce processes and HDFS operations.	Used for complex computational requirements such as BigData analytics.

V. CONCLUSIONS

Organizations around the world have found in Hadoop a simple and highly efficient model that works well in the distributed environment, Hadoop becoming more and more used. Applications running on Hadoop clusters are always improved and constantly evolving to meet all market requirements.

In this article we presented a brief description of the Apache Hadoop framework along with its main components, highlighting the major advantages that this BigData storage and analysis system brings. The research is concluded by presenting the two ways of the practical configuration of the Single-node cluster and Multi-node cluster in Hadoop and by a comparison of these implementations carried out in practice using Hadoop 3.0. This work is the initial step of a complex project that aims to contribute to data processing in Data Lake structures.

ACKNOWLEDGMENT

This work is supported by the project ANTREPRENORDOC, in the framework of Human Resources Development Operational Programme 2014-2020, financed from the European Social Fund under the contract number 36355/23.05.2019 HRD OP /380/6/13 – SMIS Code: 123847.

REFERENCES

[1] http://research.google.com/archive/gfs.html (Accessed Nov. 2020).

[2] http://research.google.com/archive/mapreduce.html (Accessed Nov. 2020).

[3] Garry Turkington, Gabriele Modena, “Big data con Hadoop”, May 2015, ISBN: 9788850333431.

[4] Shah, Ankit & Padole, Dr. Mamta. (2019). Apache Hadoop: A Guide For Cluster Configuration & Testing. INTERNATIONAL JOURNAL OF COMPUTER SCIENCES AND ENGINEERING. 7. 792-796. 10.26438/ijcse/v7i4.792796.

[5] Rehan Ghazi, Durgaprasad Gangodkar, “Hadoop, MapReduce and HDFS: A Developers Perspective”, Procedia Computer Science, Volume 48, 2015, Pages 45-50, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2015.04.108.

[6] Correia, R.C.M.; Spadon, G.; De Andrade Gomes, P.H.; Eler, D.M.; Garcia, R.E.; Olivete Junior, C. Hadoop Cluster Deployment: A Methodological Approach. Information 2018, 9, 131.

[7] Can Uzunkayaa, Tolga Ensaria, Yusuf Kavurucub, “Hadoop Ecosystem and Its Analysis on Tweets”, Procedia - Social and Behavioral Sciences, Volume 195, 3 July 2015, Pages 1890-1897.

[8] Oshin Prem, “Installation and Configuration Documentation”, Sep 2017, (Accessed Dec. 2020) https://readthedocs.org/projects/doctuts/downloads/pdf/latest/.

[9] https://www.edureka.co/blog/hadoop-tutorial/#HadoopFeatures (Accessed Dec. 2020).

[10] Eric Sammer, “Hadoop Operations”, September 2012, Published by O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.