



# Big Data Architectures and Concepts

Audrey Tembo Welo<sup>1</sup>, Hervé Lubaki Kinzonzi<sup>2</sup>, Bila Khonde Noel<sup>3</sup>, Mbuyi Mukendi Eugène<sup>4</sup>

<sup>1,3,4</sup> Faculty of Science and Technology, University of Kinshasa, Kinshasa, D.R.Congo

<sup>2</sup> Inspection, Central Bank of Congo, Kinshasa, D.R.Congo

Email: <sup>1</sup>[audreytembo72@gmail.com](mailto:audreytembo72@gmail.com), <sup>2</sup>[hervelubaki@gmail.com](mailto:hervelubaki@gmail.com), <sup>3</sup>[noel.bila@unikin.ac.cd](mailto:noel.bila@unikin.ac.cd), <sup>4</sup>[eugenembuyi@gmail.com](mailto:eugenembuyi@gmail.com)

## ARTICLE INFO

### Article history:

Received 26 April 2023

Revised 24 November 2023

Accepted 21 December 2023

Available online 30 December 2023

### Keywords:

Big Data,  
Big Data architecture,  
Hadoop,  
Hadoop cluster,  
distributed architectures

### IEEE style in citing this article:

A. T. Welo, H. L. Kinzonzi, B. K. Noel, and M. M. Eugène, "Big Data Architectures and Concepts," *Journal of Innovation Information Technology and Application (JINITA)*, vol. 5, no. 2, pp. 97–104, Dec. 2023.

## ABSTRACT

Nowadays, the processing of big data has become a major preoccupation for businesses, not only for storage and processing but also for operational requirements such as speed, maintaining performance with scalability, reliability, availability, security, and cost control; ultimately enabling them to maximize their profits by using the new possibilities offered by Big Data. In this article, we will explore and exploit the concepts and architectures of Big Data, in particular through the Hadoop open-source framework, and see how it meets the needs set out above, in its cluster structure, its components, its Lambda and Kappa architectures, and so on. We are also going to deploy Hadoop in a virtualized Linux environment, with several nodes, under the Oracle Virtual Box virtualization software, and use the experimental method to compare the processing time of the MapReduce algorithm on two DataSets with successively one, two, and three and four Datanodes, and thus observe the gains in processing time with the increase in the number of nodes in the cluster.

## 1. INTRODUCTION

The use of new devices and tools (hardware and software) by companies or individuals generates data that today, according to statisticians, will amount to 47 zettabytes in 2020 and could reach 150 zettabytes by 2025, where every day we generate several bytes of data through messages we send or receive, videos we post, posts and comments on social networks, GPS signals, climate information, and many others. This makes the way of dealing with data in business take a different direction as the data to be processed has become voluminous [19]. Every company aims to maximize its profits and to achieve its objectives better, it must manage the data at its disposal well. That's why they need to put in place and maintain a solid architecture that allows them to do so. This explosion of data is increasing year on year, so to process it more effectively we need to use appropriate processing and analysis tools. This data comes from devices connected to fixed and mobile computer networks [16], such as tablets, smartphones, computers, etc., and can help the company obtain information about users' locations, movements, interests, consumption habits, etc. Given that in the context of this work we are aiming to process big data quickly, we cannot ignore the importance of data in the life of a company or an individual, given that it is increasing all the time. As Big Data is a new field, understanding and manipulating these concepts required a lot of reading time, and in the face of existing software, we had to spend a lot of time manipulating and exploiting big data processing software such as the Hadoop platform.

---

BigData is a revolution in the field of digital information processing. The term Big Data is used to designate a significant volume of structured or unstructured data [10]. Big Data is based on internal company data, analysis of customers' data, information from online services, and consumer opinions posted on social networks. It is also linked to the development of technology, which has led to an explosion in the amount of data, making it necessary to develop the means to store and manage this huge amount of data. It is therefore defined in terms of how large masses of data can be processed and exploited optimally. The concept of Big Data is characterized by several aspects, including the management of large amounts of data, the variety that this data can have, i.e. the data can be structured, semi-structured, or even unstructured, the time taken to process this data, etc. Many IT managers and authorities in the sector tend to define Big Data in terms of three main characteristics: Volume, Speed, and Variety: Volume, Speed, and Variety [12]. BigData offers an opportunity to exploit immense data, while storing and using datasets using distributed systems in which the different parts of the data are stored in different places but brought together using software, in the case of our work, we used Hadoop as the software. BigData refers to the speed at which data is generated, captured, shared, and updated. Evolving technologies mean that businesses and consumers alike are generating data in a short space of time. Data and results are often available in real-time. For this reason, we used six virtual machines to develop our work, to respond to the concept linked to the speed of data processing.

Regarding the volume of data to be stored, we used HDFS (Hadoop Distributed File System), one of the main components of Hadoop, which operates on the master/slave principle, in a cluster where the data and services are stored on several different machines [14]. The Hadoop distributed file system is made up of [17], [18] :

- A single NameNode that plays the role of the master, managing the various client file accesses and performing operations such as opening, closing, and renaming files. The NameNode contains information about the data stored in the various nodes (the metadata). The application interacts only with the NameNode, and the latter interrogates the corresponding nodes to obtain the information requested by the application and then provides it.
- One or more DataNodes, which act as slaves, storing data and performing file system operations if requested by the client, as well as creating, replicating, and blocking files when requested by the NameNode.
- A secondary Namenode, which in the event of a NameNode failure, will continue the work done by the NameNode.

Datasets stored in the Hadoop Distributed File System (HDFS) are processed by MapReduce. It automatically slices a dataset into data fragments of the same size [14] and then applies an algorithm to these fragments to process them at the same time on available nodes in the cluster. It provides fault tolerance in that the faulty node can be restarted or the task can be assigned to another node.

## 2. RELATIVE WORDS

Many research projects deal with Big Data, in particular those [7], [20], [21] by Boumraou Kahina and Kedjar Hakim [7], HADJARI Imane, Benbachir Meriem, Boukhatem Fatima [20] and Shravya Nethula [21]. Boumraou Kahina and Kedjar Hakim [7] have implemented an interface managing the communication and connectivity of the cluster nodes, they have not used a backup master server (secondary NameNode) to remedy the failure that this master server may have and their study of the execution time involved 3 and 9 nodes. Shravya Nethula [21] compared the performance between the MapReduce algorithm on Compuverde shared storage (Compuverde File System -CVFS) and the MapReduce algorithm on HDFS and she used four nodes. By way of comparison, the purpose of our article is to implement Hadoop in a virtual environment, with an additional backup master server (secondary NameNode), and to analyze the improvement in performance using the MapReduce algorithm fed by two datasets of different sizes successively with one, two, three and then four data nodes.

## 3. BIG DATA ARCHITECTURES

Since traditional database systems do not meet the requirements of Big Data processing, they are not capable of handling massive data of various kinds in real-time. To take advantage of the benefits of Big

---

Data in business, it is necessary to push back the limits of the systems, particularly in terms of the volume of data to be analyzed, the processing speed, and the variety of data to be managed [2]. The implementation of a big data architecture within an enterprise allows for batch processing of data sources in real-time, giving the possibility of exploiting voluminous data, while transforming unstructured data into structured data to facilitate its exploitation, and also to centralize existing data and those from different sources in different formats to promote predictive analysis and which allows for tasks based on machine learning and artificial intelligence technologies [1]. Previously, the use of such an architecture was reserved for the major web players such as Google, Facebook, LinkedIn, and Yahoo, as it was very expensive and required the company to have a large number of data scientists, analysts, and architects [3]. It is thanks to the work carried out by Doug Cutting and his colleagues that Big data technologies were made open to all with the support of Yahoo. They worked on a project called Hadoop which was eventually adopted by a large number of operators who made it the reference platform for Big data.

### **3.1. Type of Big Data Architecture**

#### **3.1.1. Lambda architecture**

It is an architecture invented by Nathan Marz, to designate a generic, scalable, and fault-tolerant data processing architecture, based on his experience working on the BackType and Twitter distributed data processing systems [3], [5]. This architecture is the most commonly used for processing and managing large data in real-time and batch mode simultaneously. It allows functional separation of storage, consumption, and complex real-time processing with the ability to store and process large volumes of data (batch) while integrating the most recent data into the results [2], [6]. However, it is the most widely used because of the parameters it offers for successful processing, such as resistance to failures, fast response time during processing, scaling, and the possibility of merging block data processing (batch) and new data input (real-time). The idea of this architecture is to build a model of a real-time data processing system as a series of three layers: a batch layer, a velocity layer, and a service layer to get a perfect view of the data [6]. The purpose of a lambda architecture is not only to store data but also to make it available to other applications to exploit and extract value from it. It provides complete views of the data set.

##### **a. Batch layer**

This layer takes care of storing all the data, as the information keeps coming into the data system, this incoming data is stored as it is without any derivation or transformation i.e. in its raw form in the Batch layer. Any new data stream that arrives at the Batch layer is calculated and processed using MapReduce or machine learning. The result of this processing is stored as a batch view [1], [2].

##### **b. Speed layer (Real Time)**

This layer processes only recent data and provides more recent results incrementally using view computation to complement batch views, and also has the role of removing obsolete real-time views (post batch processing) [6], [15]. It supports the service layer to reduce latency in responding to requests. As its name suggests, the speed layer has low latency because it only processes real-time data and has a lower computational load.

##### **c. Serving Layer**

This layer is used to store and present to clients the views created by the batch and real-time layers [1]. In this layer, the following tools can be used Apache Cassandra, MongoDB, Elasticsearch, CouchBase.

#### **3.1.2. Kappa architecture**

It is an architecture based on the principle of merging the real-time and batch layers, with all data passing through a single path using a stream processing system. It is based on the streaming architecture in which a series of incoming data is first stored in a messaging engine such as Apache Kafka. From there, a streaming engine reads and transforms the data into an analyzable format and then stores it in an analytical database that end users can query [1], [2].

Kappa is a simplified, dedicated data processing architecture used in streaming layer deployment models where data sources are both batch and real-time and where end-to-end latency requirements are very strict.

### **3.2. Big Data Architecture Implementation Technologies**

The figure below illustrates some of the implementation technologies of the Big Data architecture. For the elaboration of our article, we will base ourselves on open-source technology and in particular on Hadoop [14], [17], [18].
















Technologies	Couche Batch	Couche temps réel	Couche service
Open source	  	  	  
Google			
Amazon			

Figure 1: Big data implementation technologies

### 3.3. Hadoop and its basic components

Hadoop relies on three basic components, which are essential elements of the framework and which will be the subject of our first article, namely [14], [17] :

- HDFS (Hadoop Distributed File System) is a distributed file system that manages distributed data storage and provides the fault tolerance required when operating a cluster. It consists of a NameNode (master node) which plays the role of managing the slave nodes by assigning tasks to them, it contains metadata (information on the data stored in the various nodes) a DataNode (slave node) allows the data to be stored and a secondary NameNode.
- Map reduce is a programming model for manipulating and processing cluster data sets.
- Yarn (Yet Another Resource Negotiator) is a component responsible for resource management between applications in the cluster and for task scheduling.

### 4. DEPLOYING HADOOP IN A VIRTUALISED ENVIRONMENT

We have chosen the open source technology that offers us the processing of big data, it is the Hadoop framework that will be installed in an open source virtualization software "Oracle VM VirtualBox", this will host six virtual machines of the Linux environment where we have three Ubuntu machines, one Kubuntu machine, and two Lubuntu machines. So in our Hadoop cluster, we have one name node (under Ubuntu), one secondaryNameNode (under Lubuntu), and four data nodes (two under Ubuntu, one under Kubuntu, and one under Lubuntu). To find out the Hadoop version used, in the Linux terminal, we type the command "Hadoop version" (see Figure 2).

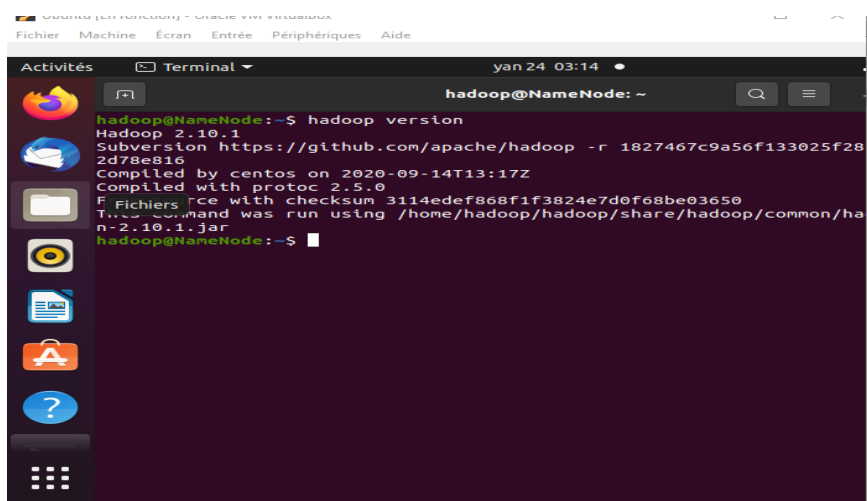


Figure 2: Hadoop version

To implement the basic components of Hadoop while respecting the concept of the master and slave nodes evoked by HDFS, we installed Hadoop in an Ubuntu virtual machine that we configured as the master. We then administered it remotely with ssh (Secure Shell) to copy the same version of Hadoop and all the configurations to the five other machines using a classic tool for copying files in an encrypted manner between remote computers, which is "scp".

```

hadoop@NameNode: ~/hadoop/etc/hadoop
-->
limitations under the License. See accompanying LICENSE file.
-->
<!-- Put site-specific property overrides in this file. -->
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>3</value>
  </property>
  <property>
    <name>dfs.block.size</name>
    <value>134217728</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:///home/hadoop/hadoopdata/hdfs/namenode</value>
  </property>
  <property>
    <name>dfs.namenode.checkpoint.dir</name>
    <value>file:///home/hadoop/hadoopdata/hdfs/namesecond</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:///home/hadoop/hadoopdata/hdfs/datanode</value>
  </property>
</configuration>
    
```

Figure 3: Configuration of the hdfs-site.xml file

NameNode communicates with the 4 data nodes and a Secondary NameNode to execute and analyze various processes that make use of mapreduce, Hdfs (Hadoop Distributed File System), and Yarn (Yet Another Resource Negotiator) which are core components of Hadoop.

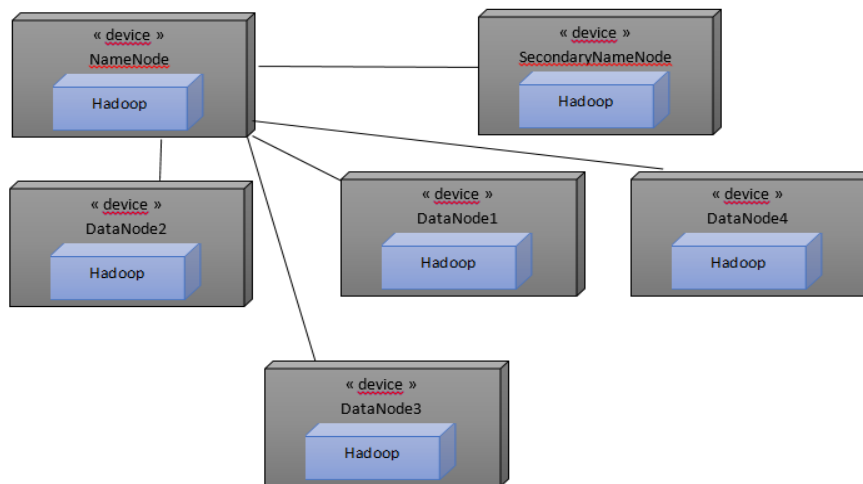


Figure 4: Deployed components

We have stored two datasets [8], [9] in HDFS to which we are going to apply parallel processing with MapReduce, one of the main components of Hadoop; its main role is to retrieve large data from the HDFS and then carry out parallel processing on it. It has two main functions, Map and Reduce; Map is used to decompose and map the data, while reduce is used to mix and calculate [22]. We used WordCount to process our datasets [8], [9], The WordCount MapReduce algorithm is used to count the number of occurrences of each observation and also to group similar observations from each dataset. Next, using the experimental method, we are going to carry out a comparative study, with graphs, on the processing times of MapReduce successively with one, two, three, and four data nodes in our cluster for each dataset.

**5. RESULT**

The figures below show the launch of the Hadoop cluster when executing a MapReduce job and using hdfs (Hadoop distributed file system) in a distributed node.

**Processing time graphs**

❖ For the first dataset, we noted the following:

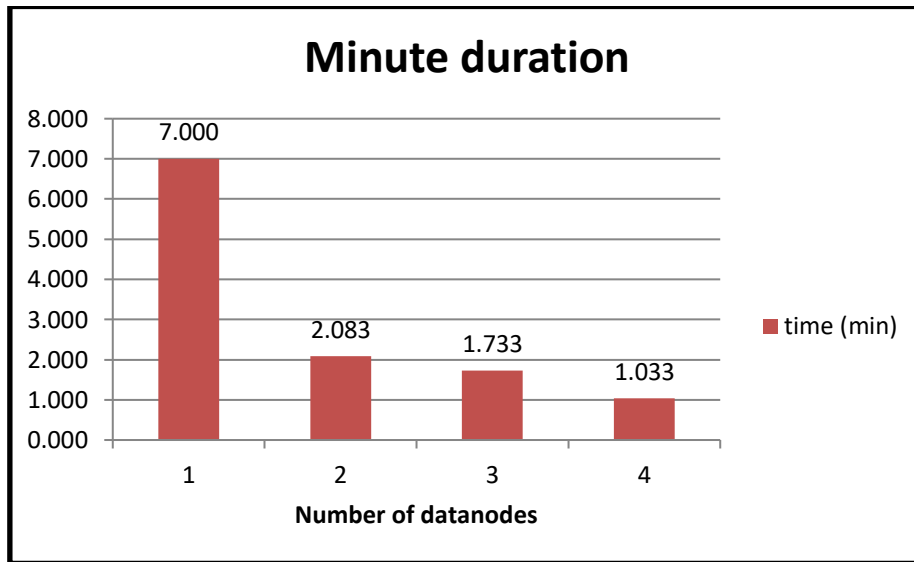


Figure 5: Processing time in minutes for 1,2,3 and 4 data nodes in the cluster using the movies.csv dataset

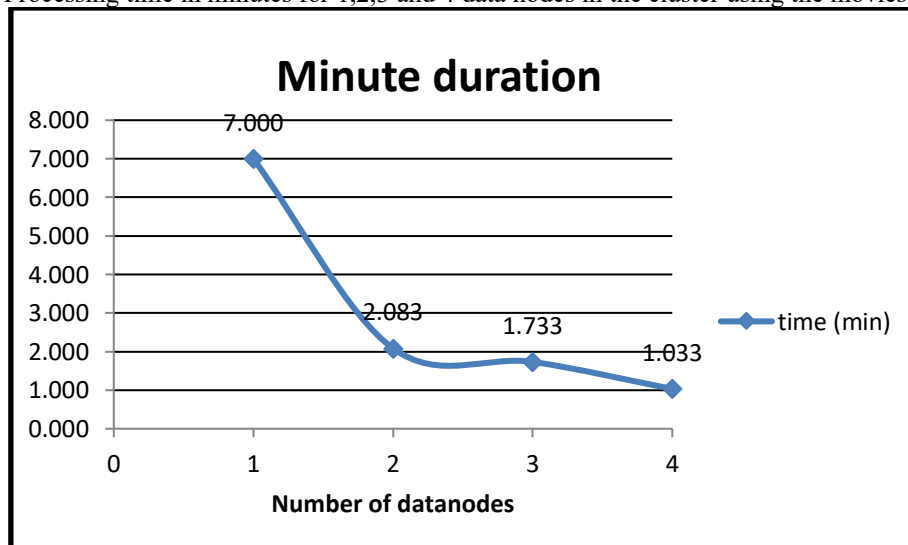


Figure 6: variation in processing time as a function of the number of data nodes in the cluster using the movies.csv dataset

The results were obtained by using the dataset [8] with 1, 2, 3, and 4 data nodes in the cluster. We can see that as the number of data nodes increases, execution time decreases. For the second dataset, we noted the following figure 7 and 8:

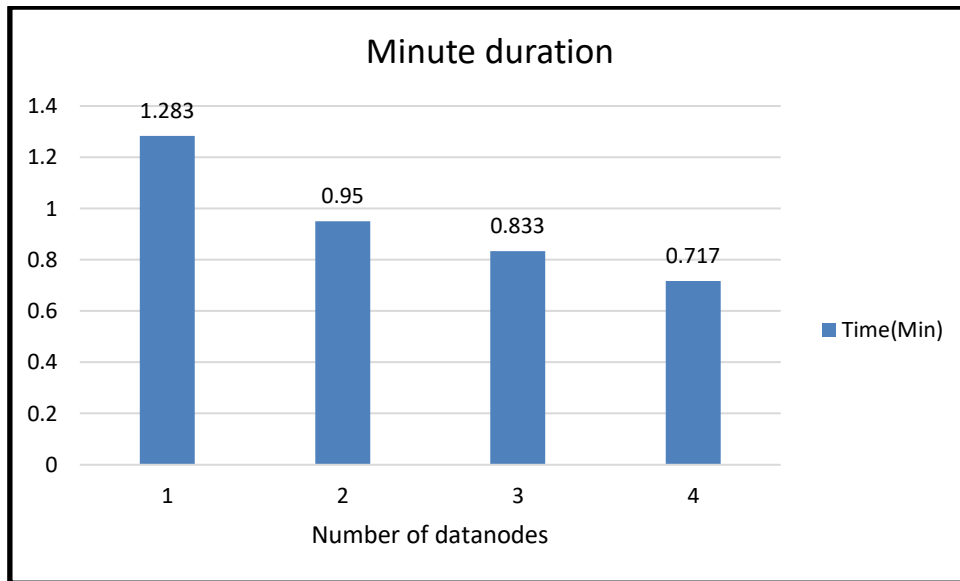


Figure 7: Processing time in minutes for 1,2,3 and 4 data nodes in the cluster using the taxi.csv dataset

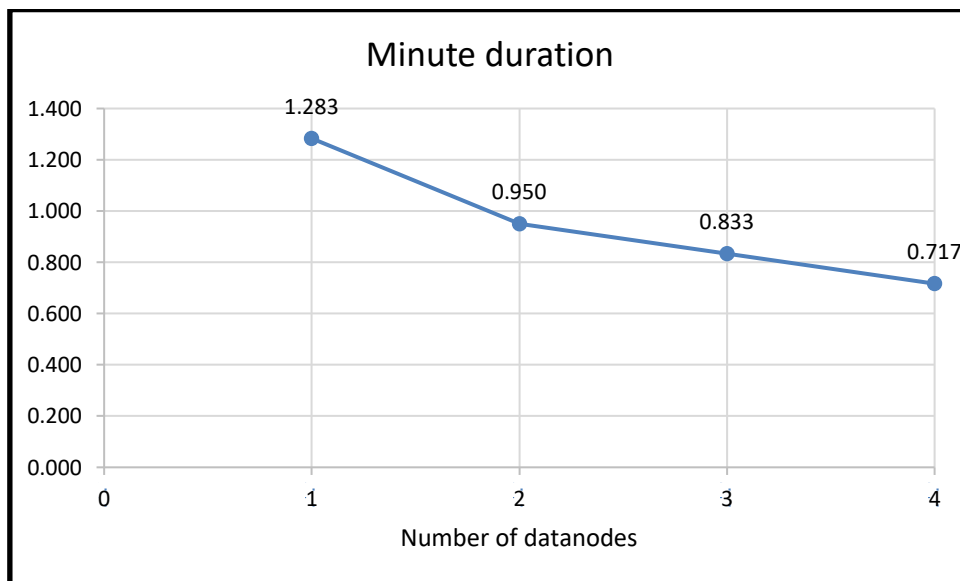


Figure 8: Variation in processing time calculated in minutes according to the number of data nodes in the cluster using the taxi.csv dataset

The results obtained using dataset [9], show us that the execution time is smaller than the execution time of dataset [8] given the difference in size of the two datasets.

**6. CONCLUSION**

Big data architectures are based on distributed architectures, which make it possible to divide the storage and processing load of a single machine between several machines to improve speed, responsiveness, and performance. In this article, we deployed Hadoop in a kappa architecture where we focused on the batch layer. We worked with 6 virtual machines, one of which played the role of a name node, four others as data nodes, and one as a secondary name node in a virtualized environment. For the two datasets [8], [9] we applied MapReduce processing by comparing the processing time in a cluster made

up successively of one, two, three, and four data nodes. This clearly showed a gain in processing time with an increase in the number of data nodes. This observed gain depends on the slowness of our virtualized environment. The time-saving in a real environment will be more consistent. Data processing time in the cluster depends on the number of nodes operating and the size of the data set to be processed. The more nodes in the cluster, the shorter the data processing time, and the larger the dataset size, the longer the processing time.

## References

- [1] J. B. N. Penka, S. Mahmoudi, and O. Debauche, "A new Kappa Architecture for IoT Data Management in Smart Farming," in *The 18th International Conference on Mobile Systems and Pervasive Computing (MobiSPC)*, Leuven, Belgium, Aug. 9-12, 2021, *Procedia Computer Science*, Sep. 2021.
- [2] G. K. Kalipe and R. K. Behera, "Big Data Architectures: A Detailed and Application Oriented Analysis," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 8, no. 9, Jul. 2019, ISSN: 2278-3075.
- [3] J. Lin, "The Lambda and the Kappa," University of Waterloo, Sep./Oct. 2017, *IEEE Internet Computing*.
- [4] H. Hashem, "Modélisation intégratrice du traitement BigData," Thèse de doctorat, Télécom SudParis, Ecole doctorale STIC, Université Paris-Saclay, Evry, France, Sep. 19, 2016.
- [5] A. Gillet, É. Leclercq, and N. Cullot, "Évolution et formalisation de la Lambda Architecture pour des analyses à hautes performances - Application aux données de Twitter," 2021 *ISTE OpenScience*, Published by ISTE Ltd., London, UK, openscience.fr.
- [6] J. Kreps, "Questioning the Lambda Architecture," Jul. 2, 2014, [Online]. Available: <https://www.oreilly.com/radar/questioning-the-lambda-architecture/>.
- [7] B. Kahina and K. Hakim, "Mise en place d'un cluster Hadoop de dix (10) postes avec interface d'exécution de jobs MapReduce à l'Ecole Nationale Supérieure en Science et Technologie de l'Informatique (ENSTI), 2019-2020," Université A/Mira de Bejaia Faculté des Sciences exactes, Département Informatique.
- [8] GroupLens, "MovieLens Datasets," [Online]. Available: <https://grouplens.org/datasets/movielens/>. [accessed 18/06/2023]
- [9] City of New York, "NYC TLC Trip Record Data," [Online]. Available: <https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page>. [accessed 25/06/2023]
- [10] P. Ducange, M. Fazzolari, and F. Marcelloni, "An overview of recent distributed algorithms for learning fuzzy models in Big Data classification," *Journal of Big Data*, vol. 7, article 19, 2020, <https://doi.org/10.1186/s40537-020-00298-6>.
- [11] AA. De Mauro, M. Greco, and M. Grimaldi, "What is big data? A consensual definition and a review of key research topics," *Computer Science*, Published 17 February 2015.
- [12] C. Avci, B. Tekinerdogan, and I. N. Athanasiadis, "Software architectures for big data: a systematic literature review," *Big Data Analytics*, vol. 5, no. 5, 2020, <https://doi.org/10.1186/s41044-020-00045-1>.
- [13] R. F. Babiceanu and R. Seker, "Big Data and Virtualization for manufacturing cyber-physical systems: A survey of the current status and future outlook," *Computers in Industry*, vol. 81, pp. 128-137, Sep. 2016.
- [14] P. Nerzic, "Outils pour le BigData," IUT de Lannion - Dept Informatique - February-March 2019
- [15] M. Feick, N. Kleer, and M. Kohn (Eds.), "Fundamentals of Real-Time Data Processing Architectures Lambda and Kappa," in *SKILL 2018, Lecture Notes in Informatics (LNI)*, Gesellschaft für Informatik, Bonn, 2018.
- [16] "Big data et objets connectés," Institut Montaigne, April 2015.
- [17] JJ. Lejeune, "Hadoop une plate-forme d'exécution de programmes Map-reduce," October 8, 2013
- [18] RR. Moussa, "Apache Hadoop Ecosystem," ZENITH Team Inria Sophia Antipolis DataScale project, February 26, 2015.
- [19] R. Herschel and V. M. Miori, "Ethics & Big Data," *Technology in Society*, vol. 49, pp. 31-36, May 2017.
- [20] I. Hadjari, M. Benbachir, and F. Boukhatem, "Big DATA: Conceptions, architectures, fonctionnements et applications," End-of-study project Master in Industrial Engineering, University of Abou Bakr Belkaid-Tlemcen, 2017.
- [21] S. Nethula, "Implementation of the Hadoop MapReduce algorithm on virtualized shared storage systems," MScS-2016-05, Faculty of Computing, Blekinge Institute of Technology, SE-371 79 Karlskrona, Sweden.
- [22] Apache Hadoop, "MapReduce Tutorial," [Online]. Available: [https://hadoop.apache.org/docs/r1.2.1/mapred\\_tutorial.html](https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html). Last Accessed: 05/18/2022 13:56:23.