Apache Hadoop overview*

Radovan Semjon

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

xsemjon@stuba.sk

16. december 2023

Abstract

In our fast-paced, business-oriented world, information is power. With the right piece of information, you can change everything. Many businesses all over the globe understand that and use data to improve their products, find out more about their customers, adapt to the current market, or simply find the answers to complex business questions. To be able to do it, companies need to collect enormous amounts of data, so-called "big data". However, using classic centralized architecture for processing and analyzing big data is problematic due to its velocity, volume, and heterogeneity. Thankfully, we can take advantage of distributed architecture. One of the best tools that we can use to harness the power of distributed architecture is Apache Hadoop. It is the foundation of modern data science, and it forever changed the way we handle big data. Understanding this piece of software is one of the first steps that one should take in order to dive into the fascinating world of data science. That's why in our article, we explained what Apache Hadoop actually is, how does it work, what its structure is, what it can do, what are its advantages and disadvantages are, and how did it change not just computer science, but society as a whole.

1 Introduction

The amount of information that governments and companies need is growing exponentially. Big data is a term that is used to describe massive datasets that contain data in terabytes, petabytes and even zettabytes. New unstructured and semi-structured data is being generated every single second, especially on big platforms like Instagram, Facebook, dating apps, or even in banks and insurance companies. Using centralized architecture for analyzing heterogeneous type of data quickly, efficiently, and cost effectively is causing a lot of problems. Distributed architecture is providing a solution for storing, processing, and analyzing this type of data.

In 2002 Doug Cutting and Mike Cafarella started working on a project called Apache Nutch. *"Hadoop was inspired by papers published by Google regarding its*

[*] Semestrálny projekt v predmete Metódy inžinierskej práce, ak. rok 2023/2024, vedenie: Ing. Vladimír Mlynarovič, PhD.

approach to handling a huge amount of data for storing, processing, and analysis on locally as well as in distributed on commodity hardware" [8]. After four years, in 2006 the first version of Apache Hadoop - version 0.1.0 was released. After a few more years, in 2013 version 2.2 was released, and in 2017, Apache Hadoop 3.0 was introduced to the world.

Apache Hadoop is an open-source, fault-tolerant, Java-based programming framework that is used for querying large sets of diversly structured data while getting results fast, using reliable and scalable architectures. It utilizes master-slave architecture and it is focusing on carrying the computation to the data rather than the data to the computation. It provides greater flexibility regarding collecting, processing and analyzing data than a conventional database framework. This piece of software is made of multiple modules that are structured in a way that Hadoop can handle different types of failures. In our article we explained these three main modules: HDFS (Hadoop distributed file system), MapReduce, Yarn (yet another resource negotiator) [4]

This paper is organized as follows: **HDFS** is a file system designed to store extremely large datasets across multiple machines in a large cluster. *"HDFS is purely a distributed file system provides the high throughput and access the data in efficient manner"* [1] We covered this topic in Section 2.1. **MapReduce** is another important module of Hadoop's structure. It is used to process data that are kept in HDFS. Map Reduce is explained in Section 2.2. **Yarn** is known as MapReduce 2.0. We dived into Yarn in Section 2.3. We also looked at various Hadoop's limitations in Section 3 and what overall impact has it made on data science and the world in Section 4. [10]

2 Hadoop and it's modules

2.1 HDFS

HDFS (Hadoop distributed file system) is a vital component of Apache Hadoop's structure, and it is the key to unlocking the possibility of using distributed architecture in this framework. It is designed to store large pieces of data across numerous machines in one huge cluster. HDFS stores each file as a **sequence of blocks**. All blocks are the same size except for the last block. To ensure fault-tolerance and efficiency, HDFS has multiple replicas of each block, and you can specify the number of replicas of a file. HDFS consists of two nodes: NameNode and DataNode. NameNode works as so-called "master node" manages file system namespace information like permissions of files and directories or locations of files". DataNode works as a "slave node". [1] [9]

2.2 Map reduce

Processing large scale data is a complicated task, especially if it involves managing hundreds or thousands of processors, parallelization, and distributed environments. The solution for the mentioned issues is Map Reduce. It is part of Apache Hadoop Yarn. Map Reduce is a distributed and parallel programming model that runs on commodity hardware and is used for **processing large datasets** in computer clusters. It supports parallel I/O scheduling, it is fault tolerant while supporting scalability. It also has inbuilt processes for status and monitoring of heterogeneous and large datasets, as in Big Data. Map reduce usually consists of two functions:**map** and **reduce**. *"The map() function takes an input key/value pair and produces a list of intermediate key/value pairs. The MapReduce runtime system groups together all intermediate pairs based on the intermediate keys and passes them to reduce() function for producing the final results."* [8]. Visual representation of map and reduce functions is shown in Figure 1. [9]

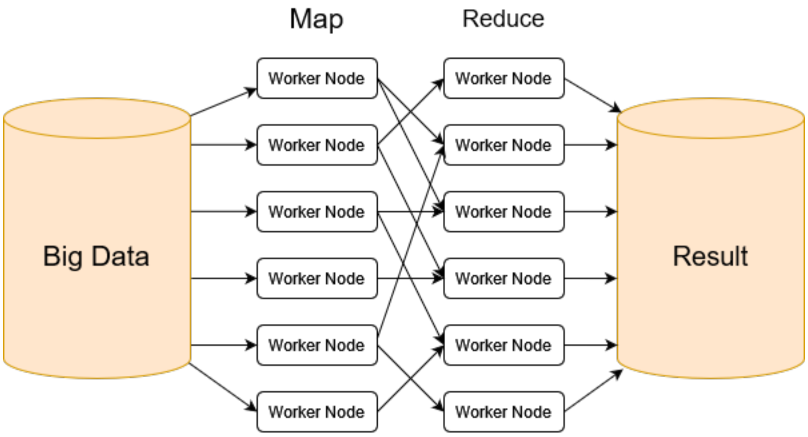

Figure 1: *Visual representation of map and reduce functions. Inspired by [1]*

Map reduce techniques:

- **Prepare the Map() input** - the "MapReduce system" designates Map processors, then it assigns the K1 input key value that each processor will work on, and provides that processor with all the neccessary input data that is associated with that key value.

- **Run the user-provided Map() code** - Map() is run one time for each and every K1 key value, while generating output organized by key values K2.

- **"Shuffle" Map output to the Reduce processors** - the MapReduce system designates Reduce processors and then assigns the K2 key value that each processor should work on, and it provides that processor with all the Map-generated data that is associated with that key value.

- **Run the user-provided Reduce() code** - Reduce() is run one time for eachand every K2 key value produced by the Map step.

- **Produce the final outpu** - the MapReduce system collects all the Reduce output, and sorts them by K2, after that it produces the final outcome. [4] [8]

2.3 Yarn

Yarn is an essential component of Apache Hadoop, which was introduced in 2012 in Hadoop version 2.0. It stands for ”yet another resource negotiator”. This technology is also known as Map Reduce 2.0. The main idea behind Yarn is to **isolate the resource management and processing segments**. The principle is to separate the two significant functionalities, like resource management and job scheduling and monitoring, into multiple discrete daemons. Yarn improves Hadoop's main selling points, for example, scalability or compatibility with MapReduce. It also improved bunch usage, nimbleness, and backing for outstanding burdens other than Map Reduce. [6]

YARN contains multiple components. These components are:

- Resource Manager
- Scheduler
- Node Manager
- Container
- Application Master

”Client submit job to the Resource Manager and Resource Manager Sends jobs to the Node Manager and Node Manager is responsible for execution of jobs and Node Manager send response of job execution back to the Resource Manager and Resource manager send response of job execution to the client” [9]. Application Master is per-application, and its job is to request resources from Resource Manager while communicating with Node Manager with the goal of executing and monitoring jobs. Node manager sends periodic updates about its aliveness to the Resource manager using heartbeat communication. Resource manager is built from two components: Scheduler and Application Master. While Application Master sends Resource Request to the Resource Manage, Scheduler's responsibilities are Scheduling jobs and it is providing resources to the Application Master as a container that contains resources like CPU, Memory etc. [9]

Hadoop is providing a huge amount of parameter configuration settings, that has default value stored in xml configuration files when we deploy it. Configuration files can be site specific, node specific, and application specific. If we want to customize the default configuration, we have three options: customization through coding, through XML files, or by passing value at run time. If we were to change configuration through coding, we could use the advantage of Hadoop providing Configuration Class with methods to access parameter configurations. If we decided to change the values of parameters through XML files, *”Hadoop provides core-site.xml, hdfs-site.xml, and mapred-site.xml with customization facility. Each parameter have set of key-value pair format where Key as a property name and value as a value of property. Final key word is use to prevent from value change. If final is true than user can not override value*” [9]. And finally, if we want to pass a value at Run Time, we can use build-in methods such as hadoop - conf and hadoop-D to. [2]

3 Limitations of Apache Hadoop

Apache Hadoop has a lot of advantages and upsides. There are numerous reasons why this framework is so widely used and why it was a revolution. However, everything has some things that could be improved, and Hadoop is no exception.

3.1 Advantages

1. **Cost effective:**

   Companies need to spend a lot of money on storing and preserving data. Hadoop solved many big data core problems, so the maintenance of these data is cost-effective.

2. **Range of data sources:**

   Data are collected from various sources, so they naturally appear in different forms. Converting all the data into the same single format would be too time-consuming and overall difficult. Thankfully, Hadoop can derive valuable data from any form of data.

3. **Multiple copies:**

   Hadoop automatically duplicates data and creates multiple copies. This ensures that in case of some failure, no data will be lost.

4. **Speed:**

   Every company that works with data in one way or another needs to process, analyze, store, or load data in the fastest way possible. Hadoop does just that with its distributed data storage. [7]

3.2 Disadvantages

1. **Lack of preventive measures:**

   Most of the data that big companies collect is sensitive and must be handled carefully. In Hadoop, the security measures are disabled by default. This is dangerous; if misconfigured, it can lead to big vulnerabilities.

2. **Small Data concerns:**

   Hadoop does not function efficiently in small data environments. Only companies that work with significant amounts of data can fully utilize its functions.

3.3 Security

Like everything else in IT, Hadoop also has vulnerabilities and security issues that are important to address. Distributed architecture (distributed computing and distributed programming) is one of the most vital features in the world of Big Data. However, ensuring the security and safety of those data can be a considerable challenge. Some basic issues that Apache Hadoop as a whole have are:

- an unauthorized user may get an access to Hadoop Distributed File System (HDFS) file via the Remote Procedure Call (RPC) or via Hyper-Text Transfer Protocol (HTTP)
- an unauthorized client may read/write a data block of a file at a DataNode via the pipeline streaming Data-transfer protocol
- an unauthorized client may gain access privileges and may submit a job to a queue or delete or change priority of the job
- an unauthorized user may access intermediate data of Map job via its task trackers HTTP shuffle protocol
- an unauthorized user may access local data which include intermediate Map output or the local storage of the DataNode that runs on the same physical node [13]

"Hadoop lacks a consistent security model. By default, Hadoop assumes a trusted environment. Hadoop has focused on improving its efficiency. Researchers are gradually paying attention to Hadoop security concerns and building security modules for it. However, currently, there is no existing evaluation for these Hadoop security modules." [3] There are more issues with Hadoop; some of them are known and actively handled, and some of them may not be discovered yet. *"Many big enterprises believe that within a few years more than half of the world's data will be stored in Hadoop Framework."* [13] To keep Hadoop's popularity, these vulnerabilities must be solved. If we want our data to be handled properly, we need to think about security, especially in popular and essential tools like Hadoop.

4 Overall impact

Apache Hadoop was created in 2006. No one really new at that time, what revolution, is it going to become. At the beginning of the "Big Data era", storing and working with huge amounts of data efficiently was extremely complicated. Problems included scalability, unstructured data, accessibility, real-time analytics, fault tolerance, and many more. In addition, there was also the problem of data being in various formats. Apache Hadoop solved all of this. This technology allowed the use of distributed computing and storing, which solved the problem mentioned earlier. [11]

It was a massive revolution, and it started a whole new era. Companies could collect, store, and analyze more data than ever before. The number of data that has been stored started increasing even more. In Figure 2 we can see the growth of worldwide data from 2006 to 2020 being exponential. Of course, we cannot say that this data growth started solely because of Apache Hadoop. However, we can say that it was a huge factor for "Big Data," if not the most critical factor. Hadoop also became a basis for amazing data science tools like Apache Spark, framework for big data analytics. *"Apache Spark has emerged as the de facto standard for big data analytics after Hadoop's MapReduce. As a framework, it combines a core engine for distributed computing with an advanced programming model for in-memory processing".* [12]

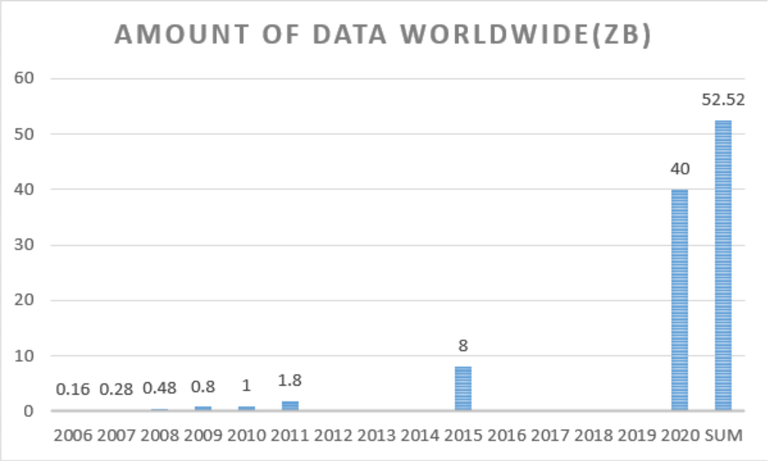

Figure 2: *Visual representation of amount of data worldwide. Data: [5]*

4.1 Discussion

Hadoop made working with data much easier, and we can definitively say that it changed the world. However, was this change positive or negative? Did Hadoop change society for something better or for something rotten? Well, just like everything else, this is two-sided. Big companies like Meta or Google are collecting and using our data for their own profit. This can be in the form of harmless ads. However, the problem arises when they keep track of **everything** we do and create models that can predict our own behavior. This raises many questions about privacy and ethics. What is even worse is when these big companies collect data to create models whose job is to keep our attention for as long as possible so they can show us as many ads as possible(for example, on Instagram). This is bordering with manipulation, and it can destroy our brains. This can be a problem, especially for children using these applications, because their brains are still developing, so we are still determining what such manipulation can do to our future generations. All of this is an example of collecting data for your own profit while destroying society in the process.

However, using big amounts of private data can also be beneficial. Analyze illnesses, create cures and better treatments, or simply learn how to make the products more helpful. Making the world a better place with data should be the goal of our society, and that is the treasure that Hadoop gave us. It's all about the direction and how we use data. To summarize, Hadoop **did not** change the world in any way, it only created new ways of changing it, and it is only up to us to which way we are as a society going to go.

5 Conclusion

Hadoop changed the industry, created many ways to change our society, and started a new era. It has become the foundation of many essential data tools, and understanding this technology is a vital skill for data scientists. In our article, we explained what Apache Hadoop is, looked at it from different angles, and created an overview of fundamental data science tool that every aspiring data scientist should be familiar with. We described basic modules and how they work, explained fundamental concepts, looked at advantages and disadvantages, described security issues, and evaluated what impact this technology has on society. Hadoop is a fascinating and complex technology, so we could not cover everything in our article. However, we believe that we provided a basic understanding of Hadoop and motivated our readers to learn more about the beautiful, complex, and exciting world of data.

6 Reaction on topics from lectures

6.1 Data and people

We already included this topic in Section 4.1

6.2 Hadoop and ethics

When it comes to Hadoop, there is a moral question of storing and working with sensitive personal data. In many cases, technological evolution would not be possible without collecting our data. However, if a company needs that data, they should protect it to the best of their ability and keep it to themselves. However, in the current society, this is a challenge however, we believe that we will reach the "safe data era". We want to try to help this situation and do something about it as a data scientists.

6.3 History of Hadoop

(We already briefly described evolution of Hadoop in Section 1 so we are gonna describe story that led to creation of Hadoop)

Apache Nutch project was the process of building a search engine system that was supposed to index 1 billion pages. After much research on Nutch, they concluded that such a system will cost around half a million dollars in hardware, plus a monthly running cost of 30, 000 dollars. So, they realized that their project architecture would not be capable of working with billions of pages on the web. So, they needed a solution to reduce implementation costs and solve the problem of storing and processing large datasets. This was the motivation behind Hadoop.

References

[1] A. Alam and J. Ahmed. Hadoop architecture and its issues. In *2014 International Conference on Computational Science and Computational Intelligence*, volume 2, pages 288–291, 2014.

[2] T. H. Aung and W. T. Zaw. Improved job scheduling for achieving fairness on apache hadoop yarn. In *2020 International Conference on Advanced Information Technologies (ICAIT)*, pages 188–193, 2020.

[3] G. S. Bhathal and A. Singh. Big data: Hadoop framework vulnerabilities, security issues and attacks. *Array*, 1:100002, 2019.

[4] D. Glushkova, P. Jovanovic, and A. Abell. Mapreduce performance model for hadoop 2. x. *Information systems*, 79:32–43, 2019.

[5] H. Guo, L. Wang, F. Chen, and D. Liang. Scientific big data and digital earth. page 1047, 2014.

[6] S. A. Hannan. An overview on big data and hadoop. *International Journal of Computer Applications*, 154(10), 2016.

[7] R. Jagadale and P. Adkar. A review paper on big data hadoop. *International Journal on Recent and Innovation Trends in Computing and Communication*, 6(5):131–135, 2018.

[8] S. G. Manikandan and S. Ravi. Big data analysis using apache hadoop. In *2014 International Conference on IT Convergence and Security (ICITCS)*, pages 1–4, 2014.

[9] B. J. Mathiya and V. L. Desai. Apache hadoop yarn parameter configuration challenges and optimization. In *2015 International Conference on Soft-Computing and Networks Security (ICSNS)*, pages 1–6, 2015.

[10] J. Nandimath, E. Banerjee, A. Patil, P. Kakade, S. Vaidya, and D. Chaturvedi. Big data analysis using apache hadoop. In *2013 IEEE 14th International Conference on Information Reuse and Integration (IRI)*, pages 700–703, 2013.

[11] A. B. Patel, M. Birla, and U. Nair. Addressing big data problem using hadoop and map reduce. In *2012 Nirma University International Conference on Engineering (NUiCONE)*, pages 1–5, 2012.

[12] S. Salloum, R. Dautov, X. Chen, P. X. Peng, and J. Z. Huang. Big data analytics on apache spark. *International Journal of Data Science and Analytics*, 1:145–164, 2016.

[13] R. Samet, A. Aydın, and F. Toy. Big data security problem based on hadoop framework. In *2019 4th International Conference on Computer Science and Engineering (UBMK)*, pages 1–6, 2019.