Table of Contents

[1. What is Hadoop? 1](#_Toc122141243)

[2. What is Apache Spark? 1](#_Toc122141244)

[3. Similarities and differences 1](#_Toc122141245)

[3.1 Fault Tolerence 1](#_Toc122141246)

[3.2 Processing Capabilities 2](#_Toc122141247)

[3.3 Performance 3](#_Toc122141248)

[References 3](#_Toc122141249)

# 1. What is Hadoop?

Hadoop solves the problems of having extensive amounts of data and computing (geeksforgeeks, 2022) via parallel processing and using distributed storage (Zhasa, 2022). It is intended to make the storage, management, and analysis of big data easier (Inouye, 2022) and was built in Java for writing MapReduce code and can be accessed via many programming languages (geeksforgeeks, 2022).

# 2. What is Apache Spark?

Apache Spark is a data processing engine which was intended for efficient large-scale data analysis. It is frequently used by data scientists to support complex data analytics and machine learning algorithms (Inouye, 2022). It also focuses on processing data via parallel processing across a cluster. However, unlike Hadoop, it uses random access memory to process and cache data.

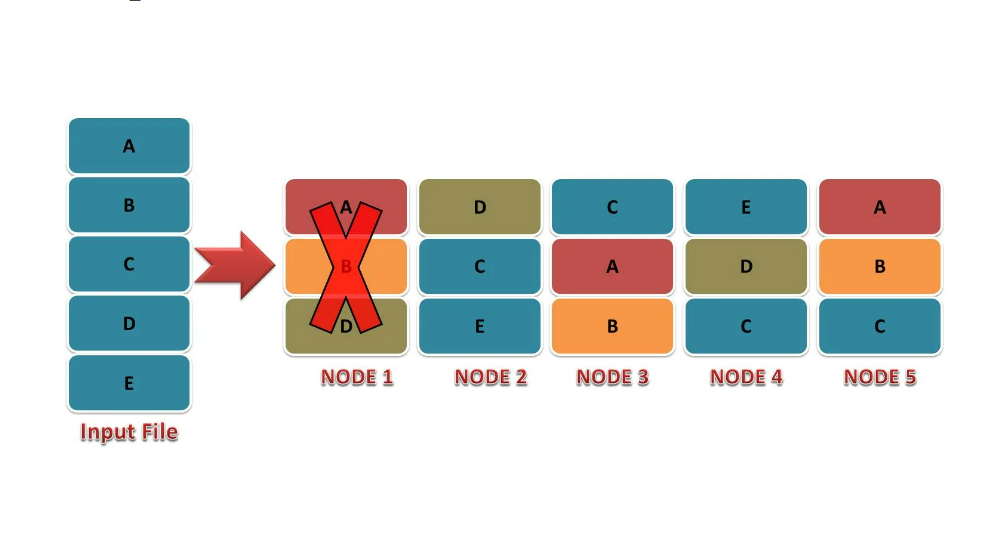
# 3. Similarities and differences

Hadoop and Apache spark share some similarities and have some differences. They will be listed and explained below with some illustrations.

## 3.1 Fault Tolerence

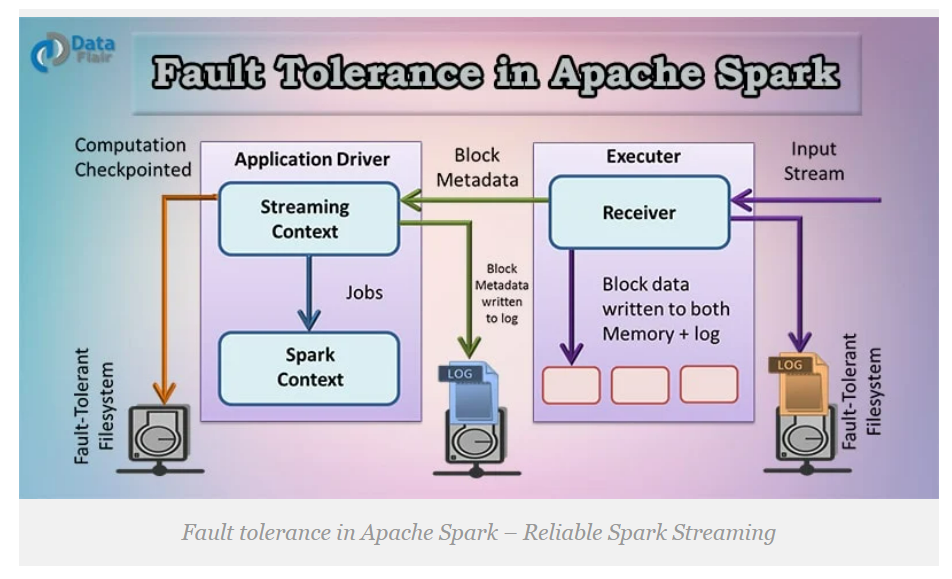
One similarity is that they both do have fault tolerance. Therefore, in the event of one component of Spark or Hadoop failing, they can still operate. This makes it such that in the event of an error, the chances of losing data are almost zero.

Their fault tolerance systems and procedures are different, however.



1. Figure showing example of Hadoop replication system for fault tolerence (DataFlair , n.d.)

It can be seen that for Hadoop, they use a replication system. This is when the data is copied and stored onto a few different nodes, like the picture above., Then in the event of the failure of one node, data can still be obtained from the other nodes. Like how in the image above, Node 1 has failed. However, data from file A can still be obtained from nodes 3 and 5. If nodes 1, 3, and 5 fail at the same time however, data of file A probably cannot be accessed until after the nodes are up and running again. The chances of Nodes 1, 3, and 5 failing at the same time is slim however, which makes this a pretty good system to be used for fault tolerance.



2. Figure showing example of Apache Spark Self Recovery Process for fault tolerance (DataFlair, n.d.)

On the other hand, Apache Spark has a self-recovery process. The way that is works is that the data from the input stream to the receiver will be written to both the memory and log. The block metadata is also written to a log to keep track of everything. Then, in the event that a component fails like the receiver, and a part of the data is missing, Spark can use the logged data to recover the missing data. This is the same with the log with the Block Metadata. If the receiver fails while there is data being streamed in however, the data being streamed in after it has failed will most likely be lost. Then in that case, the only way to obtain that data is via that data’s source.

## 3.2 Processing Capabilities

Graphical user interface

Description automatically generated

3. Figure showing the core components of Hadoop and Apache Spark (Lawton, 2022)

Both Hadoop and Spark can be used to process huge amounts of data. And to do so, Hadoop mainly uses MapReduce even though YARN enables Hadoop to have an expanded workload processing. This is because MapReduce can be used to do batch streaming for long times (Lawton, 2022).

As for Spark, it can substitute MapReduce to run batch workloads. Furthermore, Spark can provide more advanced APIs which allow for more processing use cases (Lawton, 2022). Moreover, unlike Hadoop, Spark does provide graph processing as well as real-time stream processing. Using Spark Streaming, Spark allows for receiving near real-time results. In order for Hadoop to achieve these functions, however,, it will need to be combined with other components (Macrometa, 2021).

## 3.3 Performance

As for performance, when Hadoop processes using MapReduce, it is usually slow and may be a little difficult in terms of management for this. In comparison, Spark is faster than Hadoop for processing batch jobs using memory. It is claimed that Spark is generally 100 times faster than Hadoop in this case (Lawton, 2022).

This is due to Spark using random access memory (RAM) instead of having the intermediate data be read and written to disks, which is what Hadoop does with MapReduce. It stores the data on various sources. Then, they are processed by MapReduce in batches. Therefore, slowing down the processing speed (IBM Cloud Education, 2021).

However, when many workloads which tend to run longer are being run on the same cluster, Hadoop may be better than Spark as running the same thing on Spark may cause issues with memory and may cause all applications to run slower (Lawton, 2022).

Therefore, while Spark has a faster processing speed, when there is a heavy workload, such as running and processing multiple data pipelines, which tend to run slower, at the same time, it may cause Spark to run more slowly in general. Spark may use a lot of RAM to process since Spark processes the data in RAM using the Resilient Distributed Dataset (RDD) concept. In this case, Hadoop may perform better than Spark as Hadoop is disk-bound and reads and writes files to Hadoop File System (HDFS) (geeksforgeeks, 2022)

# References

DataFlair . (n.d.). *How HDFS achieves Fault Tolerance? (with practical example)*. Retrieved from DataFlair : https://data-flair.training/blogs/learn-hadoop-hdfs-fault-tolerance/

DataFlair. (n.d.). *Fault tolerance in Apache Spark – Reliable Spark Streaming*. Retrieved from DataFlair: https://data-flair.training/blogs/fault-tolerance-in-apache-spark

geeksforgeeks. (30 September, 2022). *Difference Between Hadoop and Apache Spark*. Retrieved from geeksforgeeks: https://www.geeksforgeeks.org/difference-between-hadoop-and-apache-spark/?ref=rp

IBM Cloud Education. (27 May, 2021). *Hadoop vs. Spark: What's the Difference?* Retrieved from IBM Cloud Education: https://www.ibm.com/cloud/blog/hadoop-vs-spark

Inouye, J. (26 May, 2022). *Apache Spark vs Apache Hadoop: Compare data science tools*. Retrieved from TechRepublic: https://www.techrepublic.com/article/apache-spark-vs-hadoop/

Lawton, G. (17 February, 2022). *Hadoop vs. Spark: An in-depth big data framework comparison*. Retrieved from TechTarget: https://www.techtarget.com/searchdatamanagement/feature/Hadoop-vs-Spark-Comparing-the-two-big-data-frameworks

Macrometa. (23 October, 2021). *Apache Spark Vs Hadoop*. Retrieved from Macrometa: https://www.macrometa.com/event-stream-processing/apache-spark-vs-hadoop

Zhasa, M. (12 December, 2022). *What Is Hadoop? Components of Hadoop and How Does It Work*. Retrieved from simplilearn: https://www.simplilearn.com/tutorials/hadoop-tutorial/what-is-hadoop