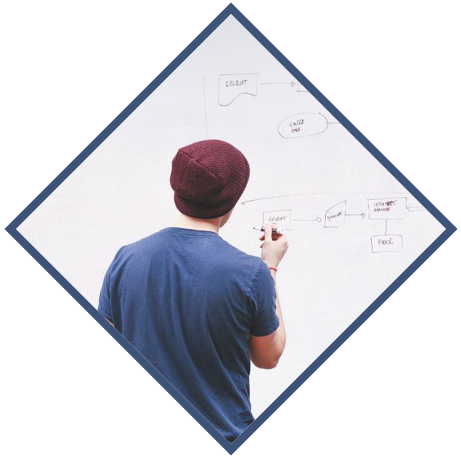
MapReduce & Apache Spark

239306V-Banujan C



HELLO!

I am Banujan Chandrakanthan

Postgraduate Student (MSc in CS - 2023)
Department of Computer Science & Engineering
Faculty of Engineering
University of Moratuwa

Introduction to MapReduce

- MapReduce is a programming model and framework used for processing large data sets in a distributed environment.
- It was first introduced by Google in 2004 as a way to parallelize and distribute computation across a large number of commodity servers.
- The idea behind MapReduce is to break down a large data set into smaller chunks, which can be processed independently in parallel by a cluster of computers.
- MapReduce consists of two phases: the map phase and the reduce phase. In the map phase, data is processed and transformed into key-value pairs. In the reduce phase, the data is aggregated and summarized based on the key-value pairs generated by the map phase.
- MapReduce is typically implemented in distributed systems such as Apache Hadoop, which provides a reliable and scalable infrastructure for processing large data sets.

Introduction to Apache Spark

- Apache Spark is an open-source, distributed computing system used for processing large data sets in a parallel and efficient manner.
- It was first introduced in 2014 by the Apache Software Foundation as a faster and more flexible alternative to MapReduce.
- Spark provides a unified platform for batch processing, interactive querying, streaming, and machine learning workloads, making it a versatile tool for big data processing.
- Spark is designed to work with various data sources, including Hadoop Distributed File System (HDFS), Cassandra, HBase, and Amazon S3.
- The core abstraction in Spark is Resilient Distributed Datasets (RDDs), which are fault-tolerant, distributed collections of data that can be processed in parallel across a cluster of machines.
- Spark provides a rich set of APIs in multiple programming languages, including Scala, Java, Python, and R, making it accessible to a wide range of developers and data scientists

Comparison of MapReduce & Spark

Ease of Use

- MapReduce: It requires developers to write a lot of code to implement even simple data processing tasks. It has a steep learning curve, and developers need to have a deep understanding of distributed computing concepts.

- Spark: provides a more user-friendly interface with a simpler programming model. It provides a higher-level API and more abstraction layers, which makes it easier for developers to write complex data processing workflows without having to worry about low-level details.

Fast Processing

- MapReduce: MapReduce is known for its batch processing capabilities, which makes it a good fit for processing large data sets offline. However, it can be slow to iterate on data and doesn't handle real-time data processing very well.

- Spark: It is designed for both batch and real-time processing. It provides in-memory processing capabilities and can handle iterative algorithms much faster than MapReduce. Additionally, Spark provides a streaming API for real-time data processing, making it a more versatile tool for big data processing

Conclusion

- MapReduce has a steep learning curve and is better suited for batch processing of large data sets.
- Apache Spark provides a simpler programming model and is faster at iterative processing, making it more versatile for both batch and real-time data processing.