Big Data Analytics

Dr Sirintra Vaiwsri | Email: sirintra.v@itm.kmutnb.ac.th

Apache Spark

Apache Spark (Aven, 2018; Antolínez García, 2023)

- Apache Spark was introduced in 2009 by Matei Zaharia.
- Apache Spark provided an open source in 2010.
- Apache Spark became a part of the Apache Software Foundation in 2013.
- Apache Spark is an alternative to MapReduce in Apache Hadoop.
- Apache Spark overcomes the disadvantage of MapReduce which has high computation cost and disk between the Map and Reduce functions.

Apache Spark (Aven, 2018; Antolínez García, 2023)

- Characteristics of Apache Spark are distributed, fault-tolerant, and in-memory structure (called Resilient Distributed Dataset: RDD).
- In-memory is the main advantage of Apache Spark because intermediate data for computations can be stored in Random Access Memory (RAM), thus, it is faster than Apache Hadoop.
- Apache Spark is written in Scala (on top of Java Virtual Machine: JVM and Java).
- Apache Spark can also implemented using Python based (called PySpark).

Key Components of Apache Spark

Unified (Chambers and Zaharia, 2018)

- Apache Spark supports many tasks required in data analytics such as data loading, data streaming process, and machine learning process.
- These supports provide easier and more efficient for writing codes for data analytics.
- Apache Spark also allows different libraries to work together to provide high performance data analytics.
- Furthermore, Apache Spark has continued expanding the built-in APIs to improve performance and cover more workloads.

Computing Engine (Chambers and Zaharia, 2018)

- Apache Spark loads data from storage and computes them in-memory.
- Apache Spark overcomes Apache Hadoop where Hadoop requires HDFS and MapReduce to compute the program.
- Apache Spark computes faster than Apache Hadoop.

Libraries (Chambers and Zaharia, 2018)

- Apache Spark supports standard libraries, external libraries, and personally created libraries.
- There are many open-source Apache Spark libraries.
- This makes it provides more functions to support data analytics.
- There are also many open sources for external libraries which we can use for various specific tasks.

Key Benefits of Apache Spark

Simpler to use and operate (Antolínez García, 2023)

- Apache Spark supports different programming languages.
- Apache Spark is inspired by MapReduce:
  - Apache Spark divides a large problem into smaller problems.
  - Apache Spark distributes parts of small problems to different solvers.
  - Each solver creates a solution.
  - Apache Spark collects solutions that solve those small problems.
  - Apache Spark assembles solutions to provide the final result that solves a large problem.

Fast (Antolínez García, 2023)

- Databricks won the Daytona GraySort contest by using Spark to sort 100TB in 23 minutes.
- It was faster than the previous world record that used 72 minutes using Hadoop.
- However, the sorting using Spark took place on disk, not in-memory.
- Therefore, Spark has high performance both on disk and in-memory where the disk will be used when data is not suitable to be used in-memory.

Scalable (Antolínez García, 2023)

- Apache Spark provides parallelised data processing.
- It allows the processing of different tasks on datasets.
- It has distributed workloads from several nodes in a cluster.
- More nodes can be added to the cluster.
- Apache Spark can run locally, called local mode.

Ease of Use (Antolínez García, 2023)

- Apache Spark provides a unified engine.
- Apache Spark supports different use cases such as batch and streaming.
- Apache Spark also supports machine learning and graph computation.

Fault Tolerance (Antolínez García, 2023)

- Apache Spark is fault tolerance where the system can continue working when a failure occurs and can recover after a failure occurs.
- Apache Spark has a Resilient Distributed Dataset (RDD) which is an immutable collection of data and a building block of Apache Spark data structure.
  - Resilient - recompute the lost partitions when a failure occurs (fault-tolerance).
  - Distributed - process in parallel
  - Dataset - set of processed data.

Apache Spark Architecture

Spark Application (Aven, 2018; Antolínez García, 2023)

- Spark Application can run on a single machine or on a cluster.
- Spark Application consists of a Driver, Cluster Manager, and Executor.
- Spark Application in cluster mode has master/slave architecture.

How Apache Spark Works?

(Aven, 2018; Antolínez García, 2023; Chambers and Zaharia, 2018)

- Client submits an application in Spark on the Driver (master node).
- The Driver creates a SparkSession (Spark 2.0) or SparkContext (Spark 1.0) to start the application.
- The Driver communicates with the Executor (slave node) to provide resources required for the application and execute the application code.
- The Driver also communicates with the Cluster Manager to conduct scheduling to keep track of resources.

Apache Spark Ecosystem

Spark Core (Antolínez García, 2023)

- Spark Core is the heart of Apache Spark which supports in-memory computing, fault-tolerance, and parallel computing.
- At low-level, it works on RDDs and cluster manager.
- For high-level, it supports libraries such as Spark SQL, Streaming, MLlib, GraphX, etc.

Spark API (Antolínez García, 2023)

- Spark supports several application programming interfaces (APIs).
- The supported programming languages such as SQL, Scala, Java, Python, and R.
- Therefore, it is easy to work with different developments.

Spark SQL, DataFrames, Dataset

(Antolínez García, 2023)

- Spark SQL allows users to query structured data.
- Spark provides data abstraction called DataFrames.
- Spark DataFrames has Spark Schema which contains column name, data type, and nullable properties.
- Datasets in Spark are immutable.
- Datasets allow users to compile an error test before executing the application.
- Datasets are only available to Java and Scala APIs.

Streaming, MLlib, GraphX (Antolínez García, 2023)

- Streaming - conducts stream computation on stream data.
- MLlib - machine learning libraries for analysing data.
- GraphX - supports graph computations and analysis.

References

- Aven, J. (2018). *Data Analytics with Spark Using Python*. Addison-Wesley Professional.
- Antolínez García, A. (2023). *Hands-on Guide to Apache Spark 3: Build Scalable Computing Engines for Batch and Stream Data Processing*. Berkeley, CA: Apress.
- Chambers, B., & Zaharia, M. (2018). *Spark: The definitive guide: Big data processing made simple*. " O'Reilly Media, Inc.".