NM - Big Data Analysis
Homework 4

1. Compare Hadoop and Spark.

Hadoop Distributed File System (HDFS):

- This stores files in a Hadoop-native format and parallelizes them across a cluster. It manages the storage of large sets of data across a Hadoop Cluster. Hadoop can handle both structured and unstructured data.
- YARN: YARN is Yet Another Resource Negotiator. It is a schedule that coordinates application runtimes.
- MapReduce: It is the algorithm that actually processes the data in parallel to combine the pieces into the desired result.
- Hadoop Common: It is also known as Hadoop Core and it provides support to all other components it has a set of common libraries and utilities that all other modules depend on.
- Hadoop is built in Java, and accessible through many programming languages, for writing MapReduce code, including Python, through a Thrift client. It's available either open-source through the Apache distribution, or through vendors such as Cloudera (the largest Hadoop vendor by size and scope), MapR, or HortonWorks.

Spark:

- Apache Spark is an open-source tool. It is a newer project, initially developed in 2012, at the AMPLab at UC Berkeley. It is focused on processing data in parallel across a cluster, but the biggest difference is that it works in memory. It is designed to use RAM for caching and processing the data.
- Spark performs different types of big data workloads like:
    - Batch processing.
    - Real-time stream processing.
    - Machine learning.
    - Graph computation.
    - Interactive queries.

2. What is Apache Spark?

Apache Spark is a data processing framework that can quickly perform processing tasks on very large data sets, and can also distribute data processing tasks across multiple computers, either on its own or in tandem with other distributed computing tools. These two qualities are key to the worlds of big data and machine learning, which require the marshalling of massive computing power to crunch through large data stores. Spark also takes some of the programming burdens of these tasks off the shoulders of developers with an easy-to-use API that abstracts away much of the grunt work of distributed computing and big data processing.

3. Explain the key features of Apache Spark.

Key features of Apache Spark:

a. Swift Processing
b. Dynamic in Nature
c. In-Memory Computation in Spark
d. Reusability
e. Fault Tolerance in Spark
f. Real-Time Stream Processing
g. Lazy Evaluation in Apache Spark
h. Support Multiple Languages
i. Active, Progressive and Expanding Spark Community
j. Support for Sophisticated Analysis
k. Integrated with Hadoop
l. Spark GraphX
m. Cost Efficient

4. What are the languages supported by Apache Spark and which is the most popular one?

Scala, Java, Python and R. Among these languages, Scala and Python have interactive shells for Spark. The Scala shell can be accessed through spark-shell and the Python shell through pyspark. Scala is the most used among them because Spark is written in Scala and it is the most popularly used for Spark.

5. What are benefits of Spark over MapReduce?

The primary difference between Spark and MapReduce is that Spark processes and retains data in memory for subsequent steps, whereas MapReduce processes data on disk. As a result, for smaller workloads, Spark's data processing speeds are up to 100x faster than MapReduce.

6. Explain the concept of Resilient Distributed Dataset (RDD).

RDD was the primary user-facing API in Spark since its inception. At the core, an RDD is an immutable distributed collection of elements of your data, partitioned across nodes in your cluster that can be operated in parallel with a low-level API that offers transformations and actions.

7. How do we create RDDs in Spark?

There are three ways to create an RDD in Spark.

- Parallelizing already existing collection in driver program.
- Referencing a dataset in an external storage system (e.g. HDFS, Hbase, shared file system).
- Creating RDD from already existing RDDs.

8. What is Executor Memory in a Spark application?

Every spark application has same fixed heap size and fixed number of cores for a spark executor. The heap size is what referred to as the Spark executor memory which is controlled with the spark.executor.memory property of the –executor-memory flag. Every spark application will have one executor on each worker node. The executor memory is basically a measure on how much memory of the worker node will the application utilize

9. What do you understand by Transformations in Spark?

Spark Transformation is a function that produces new RDD from the existing RDDs. It takes RDD as input and produces one or more RDD as output. Each time it creates new RDD when we apply any transformation. Thus, the so input RDDs, cannot be changed since RDD are immutable in nature.

10. Define Actions in Spark.

Actions are RDD's operation, that value returns back to the spar driver programs, which kick off a job to execute on a cluster. Transformation's output is an input of Actions. reduce, collect, takeSample, take, first, saveAsTextfile, saveAsSequenceFile, countByKey, foreach are common actions in Apache spark.