ASSIGNMENT 6

1.Explain what is in-Memory computation in details?

IN-MEMORY COMPUTATION

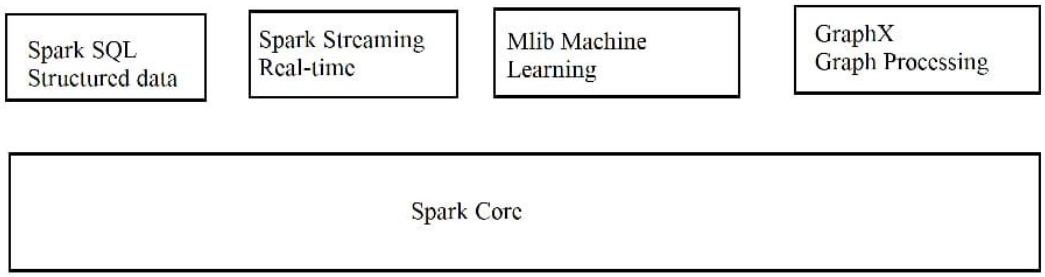
- ✓ Processing in memory is one approach to overcoming the von Neumann bottleneck, which is a limitation on throughput caused by the latency inherent in the standard computer architecture.
- ✓ In-memory computing primarily relies on keeping data in a server's RAM as a means of processing at faster speeds
- ✓ In-memory computing especially applies to processing problems that require extensive access to data -analytics, reporting or data warehousing, and big data applications.

2. Explain advantages of Spark framework ?

- ✓ Stream processing
- ✓ Machine learning
- ✓ Interactive analytics
- ✓ Graph analytics

3. Explain components of Spark with block diagram ?

Components of Apache Spark:



4. Explain benifits of in-Memory computation ?

BENIFITS OF IN-MEMORY COMPUTATION:

- ✓ Better, faster, decision making.
- ✓ Ability to reduce cost.
- ✓ Identify competitive opportunities,.
- ✓ Grow revenue
- ✓ More efficient application
- ✓ Reduce risk
- ✓ It's best suited for performing real-time analytics, and developing and deploying real-time applications

- ✓ In-memory Computing Imperative:
  Avoid movement of detailed data.
  Calculate first, then move the results

5. Explain major difference between Hadoop & Spark ?

HADOOP	SPARK
✓ Spark reduces the number of read/write cycles to disk and store intermediate data in-memory, hence faster-processing speed.	Spark is lightning fast cluster computing technology, which extends the MapReduce model to efficiently use with more type of computations.
✓ Hadoop is designed to handle batch processing efficiently	Spark is designed to handle real-time data efficiently.
✓ Hadoop is a cheaper option available while comparing it in terms of cost	Spark requires a lot of RAM to run in-memory, thus increasing the cluster and hence cost.
✓ With Hadoop MapReduce, a developer can only process data in batch mode only	Spark can process real-time data, from real time events like twitter, facebook

6. Explain features of Spark?

Fast : It provides high performance for both batch and streaming data, using a state-of-the-art DAG scheduler, a query optimizer, and a physical execution engine.

Easy to Use: It supports various languages like Java, Python, Scala, Sql, R, It facilitates to write the application in Java, Scala, Python, R, and SQL. It also provides more than 80 high-level operators.

Supports Various Libraries: It provides a collection of libraries including SQL and DataFrames, MLlib for machine learning, GraphX, and Spark Streaming.

- Supports Realtime Streaming:

- Lightweight: It is a light unified analytics engine which is used for large scale data processing. Runs Everywhere - It can easily run on Hadoop, Apache Mesos, Kubernetes, standalone, or in the cloud.

7. Write a Py-Spark program to create Dataframe from RDD & explain with screenshots & steps?

For creating a RDD data frame here we are going to use Python language using Jupyter notebook to create a sample RDD dataframe. Firstly we will install the pyspark library into the environment or in your system.

After pyspark is installed we will import SparkSession and Row from spark module which is used to create the spark session and Row is used to define a row of dataframe

```
[55] from pyspark.sql import SparkSession,Row
     spark=SparkSession.builder.getOrCreate()
```

After the session/instance is created now we need to create a RDD instance of the data

```
[57] rdd=spark.sparkContext.parallelize([Row(1,"a1",'a2','a3'),
                                         Row(2,'b1','b2','b3'),
                                         Row(3,'c1','c2','c3')])
```

After the process is done now we will finally create the dataframe

```
[61] df=spark.createDataFrame(rdd,schema=['S.no','String1','String2','String3'])
```

Printing the Dataframe in a tabular form

```
df.show()
+----+-------+-------+-------+
|S.no|String1|String2|String3|
+----+-------+-------+-------+
|   1|     a1|     a2|     a3|
|   2|     b1|     b2|     b3|
|   3|     c1|     c2|     c3|
+----+-------+-------+-------+
```

8. Explain what is RDD & why it is needed

- ✓ Resilient Distributed Datasets (RDD) is a fundamental data structure of Spark. It is an immutable distributed collection of objects. Each dataset in RDD is divided into logical partitions, which may be computed on different nodes of the cluster. RDDs can contain any type of Python, Java, or Scala objects, including user-defined classes.
- ✓ Formally, an RDD is a read-only, partitioned collection of records. RDDs can be created through deterministic operations on either data on stable storage or other RDDs. RDD is a fault-tolerant collection of elements that can be operated on in parallel.
- ✓ There are two ways to create RDDs − parallelizing an existing collection in your driver program, or referencing a dataset in an external storage system, such as a shared file system, HDFS, HBase, or any data source offering a Hadoop Input Format.

9. Write a Py-Spark program to make the column in Upper case & explain with screenshots & steps ?

Let's take the example from 7th question and try to make String1 into upper case . For creating into upper case we need to import module named upper from spark.sql.functions

```
[64]  from pyspark.sql.functions import upper
```

After module is loaded now we can just print the data in upper case as