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

IN-MEMORY: IN-MEMORY is nothing but storing the data in RAM

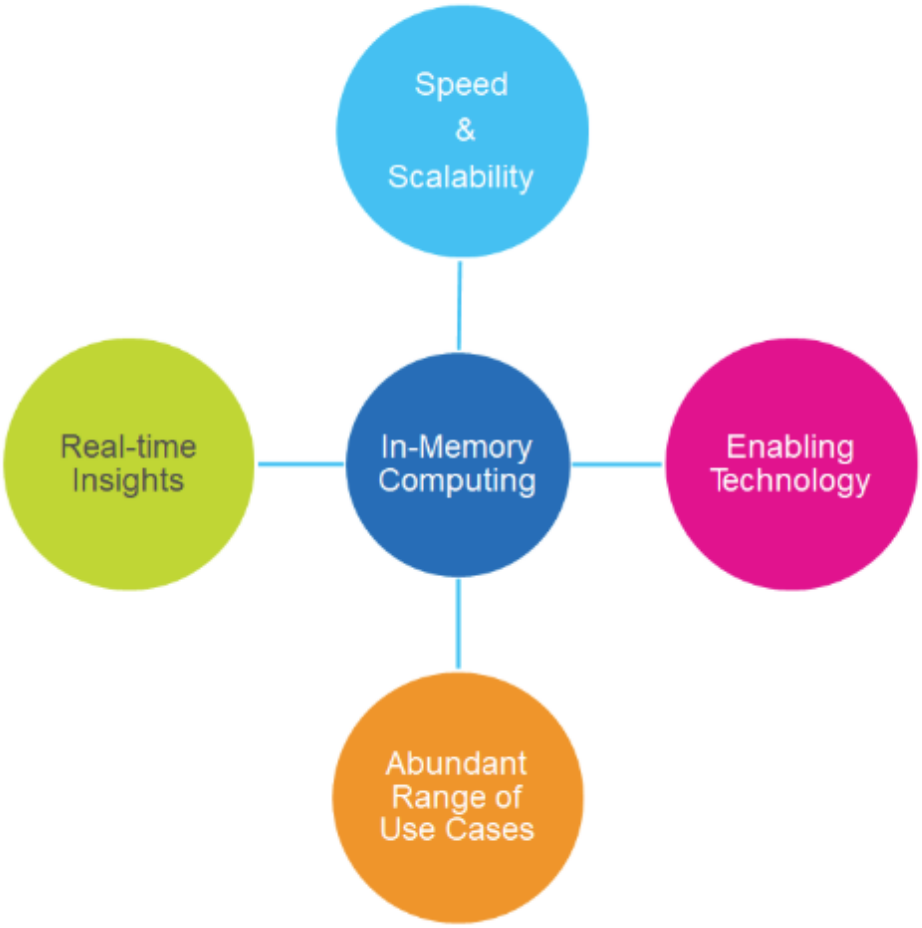
Where apache spark is nothing but open source cluster computing frame work

In-memory computation works by eliminating all slow data accesses and relying exclusively on data stored in RAM.

In-memory computation is often done in the technology known as **in-memory data grids** (IMDG).

In-Memory Computing provides super-fast performance (thousands of times faster) and scale of never-ending quantities of data, and simplifies access to increasing numbers of data sources

By storing data in RAM and processing it in parallel, it supplies real-time insights that enable businesses to deliver immediate actions and responses.

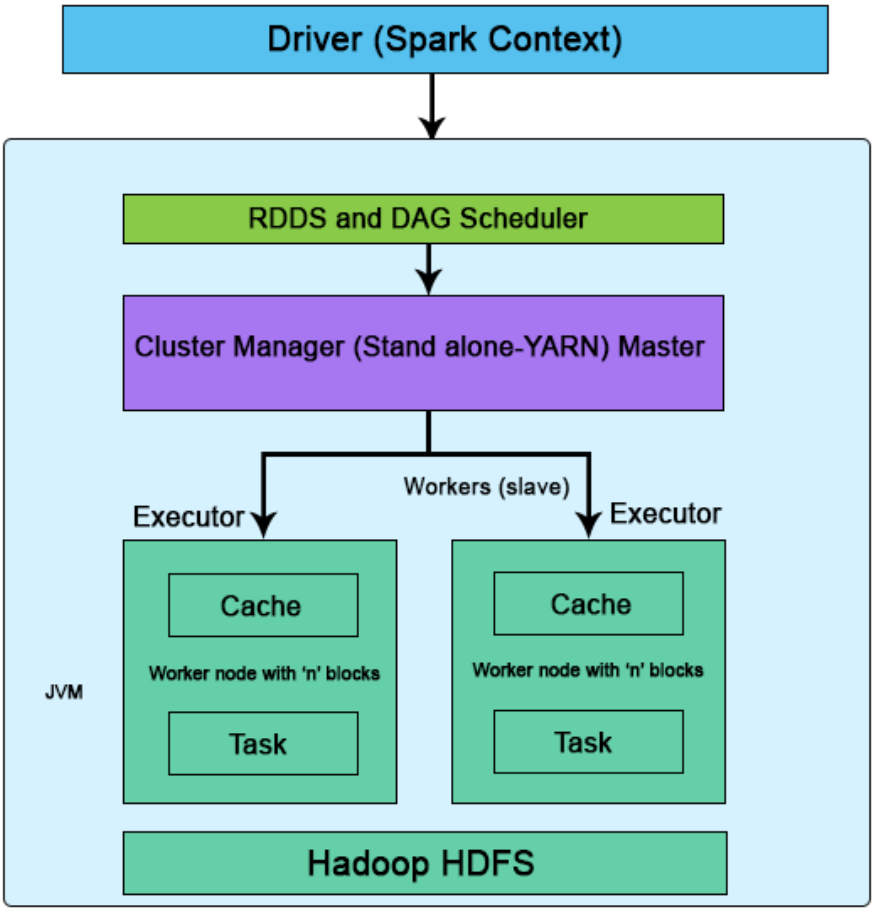


2. Explain advantages of Spark framework ?

Benefits of Apache Spark:

1. Speed
2. Ease of Use
3. Advanced Analytics
4. Dynamic in Nature
5. Multilingual
6. Apache Spark is powerful
7. Increased access to Big data
8. Demand for Spark Developers
9. Open-source community

3. Explain components of Spark with block diagram ?



4. Explain benifits of in-Memory computation ?

- Investment banking
- Insurance claim processing & modeling
- Real-time ad platforms
- Real-time sentiment analysis
- Merchant platform for online games
- Hyper-local advertising
- Geospatial/GIS processing
- Medical imaging processing
- Natural language processing & cognitive computing
- Real-time machine learning
- Complex event processing of streaming sensor data

5. Explain major difference between Hadoop & Spark ?

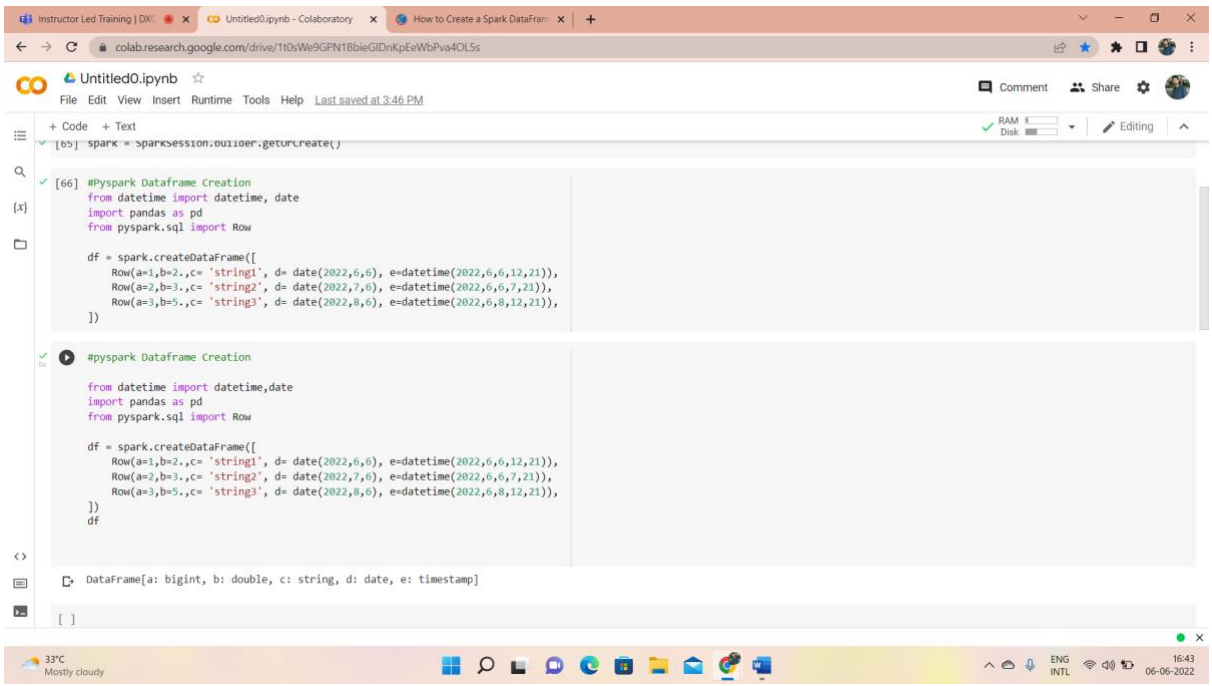
Hadoop	Spark
Hadoop is an open source framework which uses a MapReduce algorithm	Spark is lightning fast cluster computing technology, which extends the MapReduce model to efficiently use with more type of computations
Hadoop's MapReduce model reads and writes from a disk, thus slow down the processing speed	Spark reduces the number of read/write cycles to disk and store intermediate data in-memory, hence faster-processing speed.
Hadoop is designed to handle batch processing efficiently	Spark is designed to handle real-time data efficiently.
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
Hadoop is a high latency computing framework, which does not have an interactive mode	Spark is a low latency computing and can process data interactively.
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.

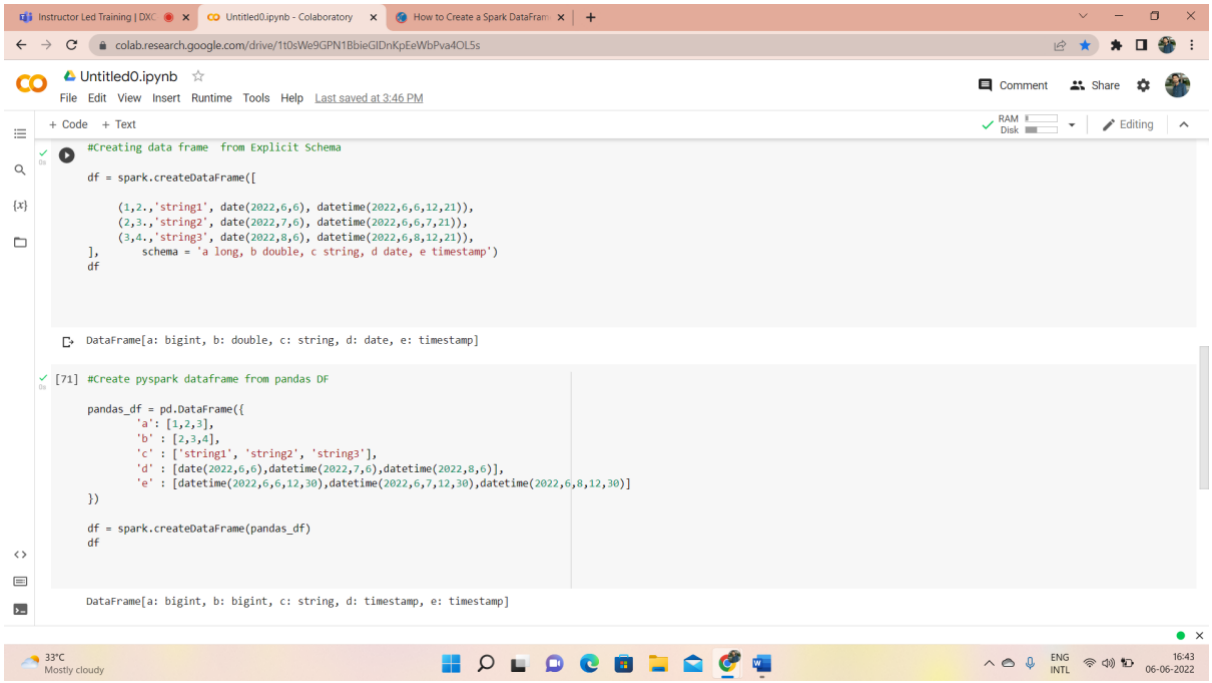
6. Explain features of Spark?

- Lighting-fast processing speed.
- Ease of use.
- It offers support for sophisticated analytics.
- Real-time stream processing.
- It is flexible.
- Active and expanding community.
- Spark for Machine Learning.
- Spark for Fog Computing.

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

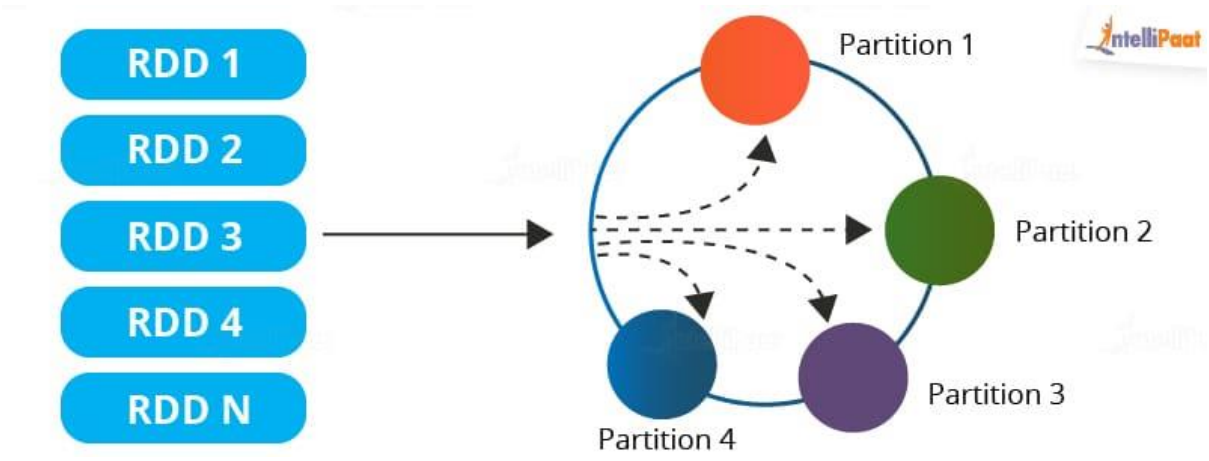
```
from pyspark.sql import SparkSession
spark = SparkSession.builder.getOrCreate()

df = spark.createDataFrame(data)
type(df)
```





8. Explain what is RDD & why it is needed ?

---→RDD (Resilient Distributed Dataset)



1. You want low-level transformation and actions and control on your dataset;
2. Your data is unstructured, such as media streams or streams of text;
3. You want to manipulate your data with functional programming constructs than domain specific expressions;
4. You don't care about imposing a schema, such as columnar format while processing or accessing data attributes by name or column; and

5. You can forgo some optimization and performance benefits available with DataFrames and Datasets for structured and semi-structured data.

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

- Convert column to upper case in pyspark – upper() function
- Convert column to lower case in pyspark – lower() function
- Convert column to title case or proper case in pyspark – initcap() function

We will be using dataframe **df_states**

```
# convert column to upper case in pyspark

from pyspark.sql.functions import upper, col
df_states.select("*", upper(col('state_name'))).show()
```