05/02/24.

1) PySpark:- It is apache Spark library written in Python to run Python applications using Apache Spark capabilities. using PySpark we can run app like on distributed clusters.

2) Apache Spark:- It is an open-source unified analytics engine used for large scale data processing. Spark is designed to be fast flexible & easy to use, making it a popular choice for processing large-scale data sets.

-> It operations on billions & trillions of data on distributed clusters 100 times faster than traditional applications.

-> can run on single/multi-node clusters.

-> was created to address the limitations of Mapreduce. by doing in-memory processing

-> It reuses data by using in-memory

3) uses PySpark: Python libraries including Numpy, Tensorflow. .. Walmart, Trivago, Sanofi etc.

-> Spyder IDE, Jupyter Notebook to run PySpark app, Anaconda also.

4) Pyspark features :-

-) in memory computation
-) distributed processing using Parallelize
-) can be used with many cluster managers (Spark, Yarn, Mesos Etc)
-) fault-tolerance
-) Inbuild optimization when using data frames.
-) Immutable
-) supports ANSI SQL
-) Lazy Evaluations &
-) Caching & Persistance.

5) Adv of Pyspark :-

=) It is general purpose, in-memory, distributed Processing engine allows you to process data efficiently in distributed cluster.

-) Apps running on Spark are 100 x faster than usual.

-) will get great benifits from Pyspark for data ingestion pipelines

=) It can process data from HADOOP HDFS, Aws S3, & RDBMS EDC.

-) also used to process real-time data using Streaming & Kafka.

=) has ML & graph libraries.

6) What Version of Python PySpark supports :-

→ PySpark 3.5 is compatible with Python 3.8 & newer

→ R 3.5, Java versions 8, 11 & 17.

→ Scala version 2.12 & 2.13. beyond.

7) PySpark architecture :-

→ works in master-slave architecture

→ master - driver slaves - workers

8) PySpark modules & Packages :-

→ PySpark RDD (PySpark.RDD)

→ " Dataframe & SQL (PySpark.sql)

→ " Streaming (Pyspark.streaming)

→ " MLib (" : ml, PySpark.mllib)

→ " Graph frames (GraphFrames)

→ " Resource (PySpark.resource)

→ Spark-Packages.org are 3rd Party Libraries

9) PySpark RDD - Resilient Distributed data set :-

Create RDD :-

2 ways.

- loading an External dataset
- distributing a set of collection of objects.

Ex: i) by using parallelize() function.

first we have to import SparkSession

```
from pyspark.sql.import SparkSession
Spark = SparkSession \
    .builder \
    .appName("Python Spark Create RDD Example") \
    .config("Spark.some.config.option", "some-value") \

df = Spark.SparkContent.Parallelize([(1,2,3,'abc'),
    (4,5,6,'def'),(7,8,9,'ghi')]).toDF
        (['col1','col2','col3','col4'])

df.show() => Prints RDD data
```

Syntax:

```
# import SparkSession.
from pyspark.sql import SparkSession

# Create SparkSession.
Spark = SparkSession.builder \
        .master("local[1]") \
        .appName
```

PYSPARK/ | Untitled1 - Jupyter Notebook | Untitled - Jupyter Notebook

localhost:8890/notebooks/PYSPARK/Untitled.ipynb?kernel_name=python3

jupyter Untitled Last Checkpoint: 2 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run | Code

In [2]:
```
pip install pyspark
```

```
     ---------------------------------------- 316.9/316.9 MB 2.4 MB/s eta 0:00:00
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Collecting py4j==0.10.9.7 (from pyspark)
  Downloading py4j-0.10.9.7-py2.py3-none-any.whl (200 kB)
     ---------------------------------------- 0.0/200.5 kB ? eta -:--:--
     ---------------------------------------- 200.5/200.5 kB 6.1 MB/s eta 0:00:00
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py): started
  Building wheel for pyspark (setup.py): still running...
  Building wheel for pyspark (setup.py): finished with status 'done'
  Created wheel for pyspark: filename=pyspark-3.5.0-py2.py3-none-any.whl size=317425388 sha256=7bf5b3a4bc79d2e0ba8d37a6a10
5224c801b1813b9b060ed3532649b0487de26
  Stored in directory: c:\users\sumedha\appdata\local\pip\cache\wheels\38\df\61\8c121f50c3cffd77f8178180dd232d90b3b99d1bd6
1fb6d6be
Successfully built pyspark
Installing collected packages: py4j, pyspark
Successfully installed py4j-0.10.9.7 pyspark-3.5.0
Note: you may need to restart the kernel to use updated packages.
```

In [3]:

```
import pyspark
from pyspark.sql import SparkSession
# Create SparkSession
spark = SparkSession.builder.appName("SparkByExamples.com") .getOrCreate()
dataList = [("Java", 20000), ("Python", 100000), ("Scala", 3000)]
rdd=spark.sparkContext.parallelize(dataList)
result = rdd.collect()
("RDD Contents:", result)
```

Out[3]:

```
('RDD Contents:', [('Java', 20000), ('Python', 100000), ('Scala', 3000)])
```

PYSPARK/ | Untitled1 - Jupyter Notebook | Untitled - Jupyter Notebook

localhost:8890/notebooks/PYSPARK/Untitled.ipynb?kernel_name=python3

jupyter **Untitled** Last Checkpoint: 2 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run | Code

In [4]:
```
import pyspark
```

In [6]:
```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("jupyter notebook") .getOrCreate()
spark
df=spark.read.csv("C:\\Users\\Sumedha\\OneDrive\\Desktop\\employee_data.csv")
df.show()
```

```
+-------+---+------+----------+
|    _c0|_c1|   _c2|       _c3|
+-------+---+------+----------+
|   Name|Age|Salary|Department|
|   John| 30| 50000|        HR|
|   Jane| 28| 55000|        IT|
|    Bob| 35| 60000|   Finance|
|  Alice| 32| 48000|        HR|
|Charlie| 40| 70000|        IT|
+-------+---+------+----------+
```

In [ ]:

PYSPARK/ | Untitled1 - Jupyter Notebook | Untitled - Jupyter Notebook

localhost:8890/notebooks/PYSPARK/Untitled1.ipynb?kernel_name=python3

UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions. Don't show anymore

jupyter Untitled1 Last Checkpoint: 2 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

In [8]:
```
from pyspark.sql import SparkSession


# Create SparkSession
spark = SparkSession.builder \
      .master("local[1]") \
      .appName("SparkByExamples.com") \
      .getOrCreate()
dataList = [("Java", 20000), ("Python", 100000), ("Scala", 3000)]
rdd=spark.sparkContext.parallelize(dataList)
rdd2 = spark.sparkContext.textFile("C:\\Users\\Sumedha\\OneDrive\\Desktop\\scores.txt")
result = rdd2.collect()
("RDD2 Contents:", result)
```

Out[8]:
```
('RDD2 Contents:',
 ['Name,M1 Score,M2 Score,age',
  'Alex,62,80,20',
  'Brad,45,56,19',
  'Joey,85,98,21',
  'abhi,54,79,20'])
```

In [ ]: