Data Frames in Spark SQL

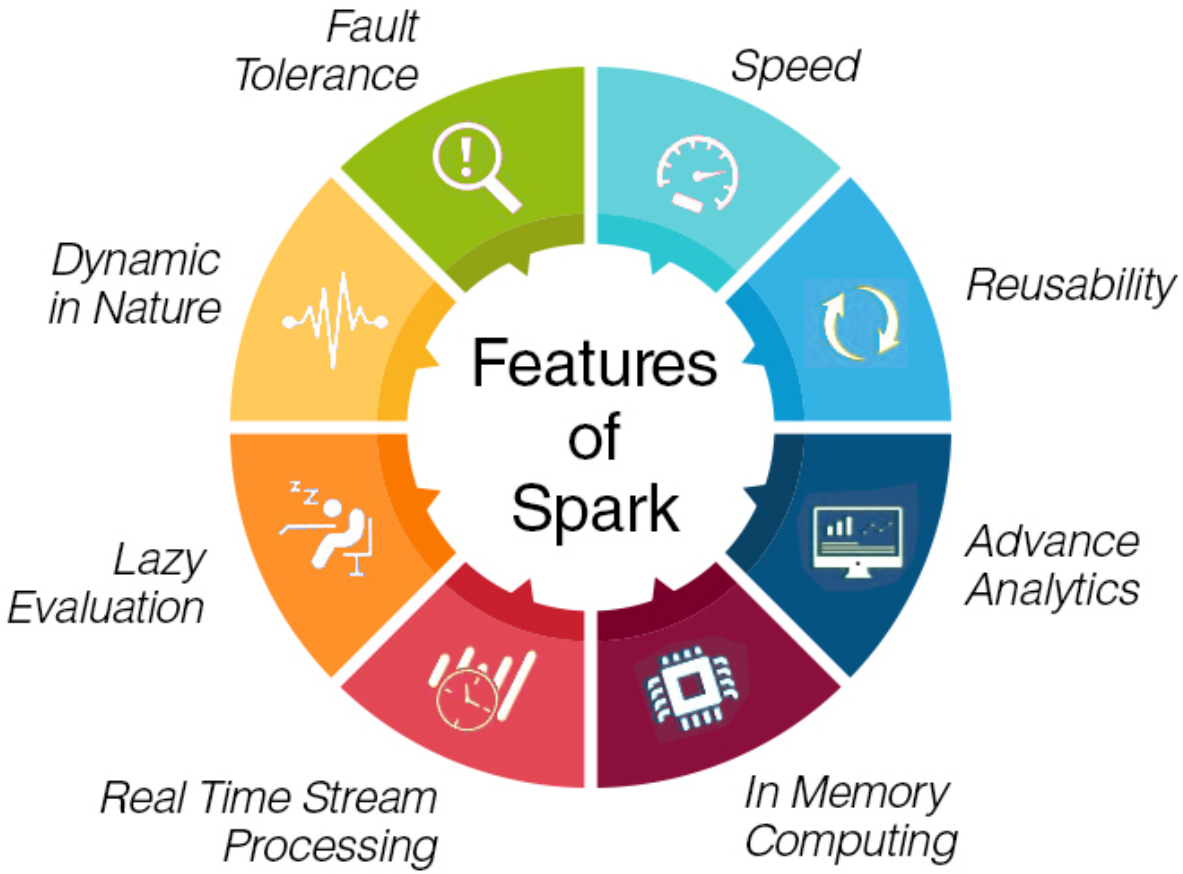
Agenda

- **Identify the drawbacks of handling structured data in an RDD**
- **Understand the challenges of handling structured BIG DATA sets**
- **Why do we need data frames in SPARK**
- **Use cases of data frames in SPARK SQL**

RDD’s

- **They are objects in SPARK which represent BIG DATA**
- **They are distributed**
- **They are fault tolerant**
- **They are lazily evaluated**
- **Scalable**
- **Supports near real time stream processing**
- **They have a lot of other benefits which makes it THE element of BIG DATA computation in SPARK**

RDD's are actually great



How do we normally create the RDD's ?

1. **The parallelize() method**

   sc = SparkContext.getOrCreate()

   myrdd = sc.parallelize([1,2,3,4,5,6,7,8,9])

1. **Data source API (eg. The textFile API)**

   sc = sc.textFile("FILE PATH")

How is the data stored in an RDD ?

- What is the data format inside an RDD ?

  It’s just raw sequence of bytes. Especially when its created using a the textFile() or parallelize() API

- Creating RDD’s using structured data (CSV data)

```
sc = sc.textFile("sample.csv")
myrdd = sc.flatMap(lambda e:e.split(" "))
myrdd.collect()
```

Drawbacks of RDD's

- They store data as sequence of bytes
- This suits unstructured data manipulation
- Data does not have any schema
- There is no direct support to handle structured data
- The native API's on RDD's seldom provide any support for structured data handling
- Special RDD format is needed to store structured data (BIG DATA)
- Special set of API's are needed to manipulate and query such RDD's

SCHEMA RDD's (DATA FRAMES)

- **The idea of named columns and schema for the RDD data is borrowed from data frames**
- **Spark allows creation of data frames using specific data frame API's**
- **An RDD can be created using the standard set of regular data set API's , schema can be assigned to the data and explicitly converted into a data frame**
- **There is provision to manipulate the spark data frames using the library "SPARK SQL"**
- **The standard set of SPARK transformation API's can also be used on SPARK data frames**

Creating a SPARK Data Frame

Consider the following CSV data saved in a text file

1,Ram,48.78,45
2,Sita,12.45,40
3,Bob,3.34,23
4,Han,16.65,36
5,Ravi,24.6,46

```
rdd = sc.textFile("sample.csv")

csvrdd = rdd.map(lambda e:e.split(","))

emp = csvrdd.map(lambda e: Row( id=long(e[0]), name=e[1],sal=e[2], age=int(e[3].strip())))

empdf =  sqlContext.createDataFrame(emp)
```

DEMO

Summary

- Understand the drawbacks of the RDD's in efficiently handling structured data
- The need for data frames and data frame operations
- There are a bunch of other DATA SOURCE API's which can directly read data from sources like CSV, JSON, Parquet, JSON etc.