RDD Drawbacks

RDD Drawbacks

1. Tough to handle semi-structured and structured data

RDD Drawbacks

1. Tough to handle semi-structured and structured data

ID	NAME	SALARY
1	Alex	10000
2	Britney	15000
3	Christine	18000

RDD Drawbacks

1. Tough to handle semi-structured and structured data

ID	NAME	SALARY
1	Alex	10000
2	Britney	15000
3	Christine	18000

```
dataCSV = sc.textFile("data/Sample.csv")
dataCSV.collect()
```

```
['ID,NAME,SALARY', '1,Alex,10000', '2,Britney,15000', '3,Christine,18000']
```

RDD Drawbacks

1. Tough to handle semi-structured and structured data

ID	NAME	SALARY
1	Alex	10000
2	Britney	15000
3	Christine	18000

```
dataCSV = sc.textFile("data/Sample.csv")
dataCSV.collect()
```

```
['ID,NAME,SALARY', '1,Alex,10000', '2,Britney,15000', '3,Christine,18000']
```

RDD Drawbacks

1. Tough to handle semi-structured and structured data

ID	NAME	SALARY
1	Alex	10000
2	Britney	15000
3	Christine	18000

```
dataCSV = sc.textFile("data/Sample.csv")
dataCSV.collect()
```

```
['ID,NAME,SALARY', '1,Alex,10000', '2,Britney,15000', '3,Christine,18000']
```

RDD Drawbacks

1. Tough to handle semi-structured and structured data
2. No in-built optimization

RDD Drawbacks

1. Tough to handle semi-structured and structured data
2. No in-built optimization

rdd.join().filter()

or

rdd.filter().join()

?

RDD Drawbacks

1. Tough to handle semi-structured and structured data
2. No in-built optimization

rdd.join().filter()

or

rdd.filter().join()

?

Spark DataFrames

What are Spark DataFrames?

1. Distributed collection of semi-structured or structured data

What are Spark DataFrames?

1. Distributed collection of structured data
   a. Similar to tables in relational databases and Python/R data frames

Id (Int)	First (String)	Last (String)	Url (String)	Published (Date)	Hits (Int)	Campaigns (List[Strings])
1	Jules	Damji	https:// tinyurl.1	1/4/2016	4535	[twitter, LinkedIn]
2	Brooke	Wenig	https:// tinyurl.2	5/5/2018	8908	[twitter, LinkedIn]
3	Denny	Lee	https:// tinyurl.3	6/7/2019	7659	[web, twitter, FB, LinkedIn]
4	Tathagata	Das	https:// tinyurl.4	5/12/2018	10568	[twitter, FB]

What are Spark DataFrames?

1. Distributed collection of structured data
   a. Similar to tables in relational databases and Python/R data frames

Column object

Id (Int)	First (String)	Last (String)	Url (String)	Published (Date)	Hits (Int)	Campaigns (List[Strings])
1	Jules	Damji	https:// tinyurl.1	1/4/2016	4535	[twitter, LinkedIn]
2	Brooke	Wenig	https:// tinyurl.2	5/5/2018	8908	[twitter, LinkedIn]
3	Denny	Lee	https:// tinyurl.3	6/7/2019	7659	[web, twitter, FB, LinkedIn]
4	Tathagata	Das	https:// tinyurl.4	5/12/2018	10568	[twitter, FB]

Row object

What are Spark DataFrames?

1. Distributed collection of structured data
2. High level operations

What are Spark DataFrames?

1. Distributed collection of structured data
2. High level operations
   a. Aggregating, Filtering, Sorting, etc.

What are Spark DataFrames?

1. Distributed collection of structured data
2. High level operations
   a. Aggregating, Filtering, Sorting, etc.
   b. Simpler queries

What are Spark DataFrames?

1. Distributed collection of structured data
2. High level operations
    a. Aggregating, Filtering, Sorting, etc.
    b. Simpler queries
    c. Optimised

Example

Name	Score
Amanda	20
Bella	31
Charlie	30
David	35
Amanda	25

RDD

DataFrame

```
from pyspark import SparkContext

# sc object
sc = SparkContext()

# sample rdd
rdd_orig = sc.parallelize([('Amanda',20),('Bella',31),('Charlie',30),('David',35),('Amanda',25)])

# update pair rdd like ('Amanda', (20, 1))
rdd_pair = rdd_orig.mapValues(lambda x: (x,1))

# add scores and counts like ('Amanda', (45, 2))
rdd_count = rdd_pair.reduceByKey(lambda a,b: (a[0]+b[0], a[1]+b[1]))

# take average of score
rdd_avg = rdd_count.mapValues(lambda x: x[0]/x[1])

# collect result
rdd_avg.collect()
```

```
[('Amanda', 22.5), ('Charlie', 30.0), ('David', 35.0), ('Bella', 31.0)]
```

RDD

DataFrame

```
from pyspark import SparkContext

# sc object
sc = SparkContext()

# sample rdd
rdd_orig = sc.parallelize([('Amanda',20),('Bella',31),('Charlie',30),('David',35),('Amanda',25)])

# update pair rdd like ('Amanda', (20, 1))
rdd_pair = rdd_orig.mapValues(lambda x: (x,1))

# add scores and counts like ('Amanda', (45, 2))
rdd_count = rdd_pair.reduceByKey(lambda a,b: (a[0]+b[0], a[1]+b[1]))

# take average of score
rdd_avg = rdd_count.mapValues(lambda x: x[0]/x[1])

# collect result
rdd_avg.collect()
```

```
[('Amanda', 22.5), ('Charlie', 30.0), ('David', 35.0), ('Bella', 31.0)]
```

RDD

```
from pyspark import SparkContext

# sc object
sc = SparkContext()

# sample rdd
rdd_orig = sc.parallelize([('Amanda',20),('Bella',31),('Charlie',30),('David',35),('Amanda',25)])

# update pair rdd like ('Amanda', (20, 1))
rdd_pair = rdd_orig.mapValues(lambda x: (x,1))

# add scores and counts like ('Amanda', (45, 2))
rdd_count = rdd_pair.reduceByKey(lambda a,b: (a[0]+b[0], a[1]+b[1]))

# take average of score
rdd_avg = rdd_count.mapValues(lambda x: x[0]/x[1])

# collect result
rdd_avg.collect()
```

```
[('Amanda', 22.5), ('Charlie', 30.0), ('David', 35.0), ('Bella', 31.0)]
```

DataFrame

```
# import libraries
from pyspark.sql import SparkSession
from pyspark.sql.functions import avg

# create sparksession object
spark = SparkSession.builder.getOrCreate()

# create dataframe
df = spark.createDataFrame([('Amanda',20),('Bella',31),('Charlie',30),("David",35),('Amanda',25)],
                           ["name", "score"])

# compute average
df_avg = df.groupBy("name").agg(avg("score"))

# display average
df_avg.show()
```

```
+-------+----------+
|   name|avg(score)|
+-------+----------+
| Amanda|      22.5|
|Charlie|      30.0|
|  David|      35.0|
|  Bella|      31.0|
+-------+----------+
```

Analytics Vidhya

RDD

```python
from pyspark import SparkContext

# sc object
sc = SparkContext()

# sample rdd
rdd_orig = sc.parallelize([('Amanda',20),('Bella',31),('Charlie',30),('David',35),('Amanda',25)])

# update pair rdd like ('Amanda', (20, 1))
rdd_pair = rdd_orig.mapValues(lambda x: (x,1))

# add scores and counts like ('Amanda', (45, 2))
rdd_count = rdd_pair.reduceByKey(lambda a,b: (a[0]+b[0], a[1]+b[1]))

# take average of score
rdd_avg = rdd_count.mapValues(lambda x: x[0]/x[1])

# collect result
rdd_avg.collect()
```

```
[('Amanda', 22.5), ('Charlie', 30.0), ('David', 35.0), ('Bella', 31.0)]
```

DataFrame

```python
# import libraries
from pyspark.sql import SparkSession
from pyspark.sql.functions import avg

# create sparksession object
spark = SparkSession.builder.getOrCreate()

# create dataframe
df = spark.createDataFrame([('Amanda',20),('Bella',31),('Charlie',30),("David",35),('Amanda',25)],
                           ["name", "score"])

# compute average
df_avg = df.groupBy("name").agg(avg("score"))

# display average
df_avg.show()
```

```
+-------+----------+
|   name|avg(score)|
+-------+----------+
| Amanda|      22.5|
|Charlie|      30.0|
|  David|      35.0|
|  Bella|      31.0|
+-------+----------+
```

Analytics Vidhya

What are Spark DataFrames?

1. Distributed collection of structured data
2. High level operations
3. Support for multiple formats and sources

What are Spark DataFrames?

1. Distributed collection of structured data
2. High level operations
3. Support for multiple formats and sources

< XML /> { JSON } HIVE MySQL more...

Thank You!!