PySpark - Resilient Distributed Dataset (RDD)

- can seamlessly move between DataFrame or Dataset and RDDs at will—by simple API method calls—and DataFrames and Datasets are built on top of RDDs
- Transformations are generated as Directed Acyclic Graph (DAG). DAG can be recomputed during failure
- Transformations - map, filter, flatMap, textFile...
- RDD is spark core abstraction, its immutable distributed collection of objects
- Internally spark distributes the data in RDD to different nodes across the cluster to achieve parallelization # Reasons on When to use RDDs You want low-level transformation and actions and control on your dataset; Your data is unstructured, such as media streams or streams of text; You want to manipulate your data with functional programming constructs than domain specific expressions; You don't care about imposing a schema, such as columnar format while processing or accessing data attributes by name or column; and You can forgo some optimization and performance benefits available with DataFrames and Datasets for structured and semi-structured data. # Ways - By loading external dataset and By distributing collection of objects

In [1]:
```
import pyspark
```

In [2]:
```
from pyspark import SparkContext
```

In [3]:
```
from pyspark.sql import SparkSession,types
spark = SparkSession.builder.master("local").appName("Practice").getOrCreate()
```

In [4]:
```
sc=spark.sparkContext
```

In [5]:
```
#Creating variable
input_list=[['amar'],['raj'],['varun']]
```

In [6]:
```
rdd1=sc.parallelize(input_list)
```

In [7]:
```
type(rdd1)
```

Out[7]:
```
pyspark.rdd.RDD
```

In [8]:
```
rdd1.collect()
```

Out[8]:
```
[['amar'], ['raj'], ['varun']]
```

In [16]:
```
##Filter need to review getting an error for filter- over RDD
rdd2=sc.parallelize([1,2,3,4,5,6,7,8,9])
filter=rdd1.filter(lambda x: x%2 == 0)
print("Values rdd2: {0}".format(rdd2.collect()))
print("Values filter: {0}".format(filter.collect()))
```

```
Values rdd2: [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
---------------------------------------------------------------------------
Py4JJavaError                             Traceback (most recent call last)
<ipython-input-16-cc0997afe3b6> in <module>
      3 filter=rdd1.filter(lambda x: x%2 == 0)
      4 print("Values rdd2: {0}".format(rdd2.collect()))
----> 5 print("Values filter: {0}".format(filter.collect()))

C:\ProgramData\Anaconda3\lib\site-packages\pyspark\rdd.py in collect(self)
   1195         with SCCallSiteSync(self.context):
   1196             assert self.ctx._jvm is not None
-> 1197             sock_info = self.ctx._jvm.PythonRDD.collectAndServe(self.
_jrdd.rdd())
   1198         return list(_load_from_socket(sock_info, self._jrdd_deseriali
zer))
   1199

C:\ProgramData\Anaconda3\lib\site-packages\py4j\java_gateway.py in __call__(s
elf, *args)
   1320         answer = self.gateway_client.send_command(command)
   1321         return_value = get_return_value(
-> 1322             answer, self.gateway_client, self.target_id, self.name)
   1323
   1324         for temp_arg in temp_args:

C:\ProgramData\Anaconda3\lib\site-packages\pyspark\sql\utils.py in deco(*a, *
*kw)
    188     def deco(*a: Any, **kw: Any) -> Any:
    189         try:
--> 190             return f(*a, **kw)
    191         except Py4JJavaError as e:
    192             converted = convert_exception(e.java_exception)

C:\ProgramData\Anaconda3\lib\site-packages\py4j\protocol.py in get_return_val
ue(answer, gateway_client, target_id, name)
    326                 raise Py4JJavaError(
    327                     "An error occurred while calling {0}{1}{2}.\n".
--> 328                     format(target_id, ".", name), value)
    329             else:
    330                 raise Py4JError(

Py4JJavaError: An error occurred while calling z:org.apache.spark.api.python.
PythonRDD.collectAndServe.
: org.apache.spark.SparkException: Job aborted due to stage failure: Task 0 i
n stage 8.0 failed 1 times, most recent failure: Lost task 0.0 in stage 8.0
 (TID 8) (EXLAPLPNyCdxfzp.corp.exlservice.com executor driver): org.apache.sp
ark.SparkException: Python worker failed to connect back.
        at org.apache.spark.api.python.PythonWorkerFactory.createSimpleWorker
(PythonWorkerFactory.scala:189)
        at org.apache.spark.api.python.PythonWorkerFactory.create(PythonWorke
rFactory.scala:109)
        at org.apache.spark.SparkEnv.createPythonWorker(SparkEnv.scala:124)
        at org.apache.spark.api.python.BasePythonRunner.compute(PythonRunner.
scala:164)
        at org.apache.spark.api.python.PythonRDD.compute(PythonRDD.scala:65)
        at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:365)
        at org.apache.spark.rdd.RDD.iterator(RDD.scala:329)
        at org.apache.spark.scheduler.ResultTask.runTask(ResultTask.scala:90)
```

```
        at org.apache.spark.scheduler.Task.run(Task.scala:136)
        at org.apache.spark.executor.Executor$TaskRunner.$anonfun$run$3(Execu
tor.scala:548)
        at org.apache.spark.util.Utils$.tryWithSafeFinally(Utils.scala:1504)
        at org.apache.spark.executor.Executor$TaskRunner.run(Executor.scala:5
51)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(Unknown Source)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(Unknown Source)
        at java.lang.Thread.run(Unknown Source)
Caused by: java.net.SocketTimeoutException: Accept timed out
        at java.net.DualStackPlainSocketImpl.waitForNewConnection(Native Meth
od)
        at java.net.DualStackPlainSocketImpl.socketAccept(Unknown Source)
        at java.net.AbstractPlainSocketImpl.accept(Unknown Source)
        at java.net.PlainSocketImpl.accept(Unknown Source)
        at java.net.ServerSocket.implAccept(Unknown Source)
        at java.net.ServerSocket.accept(Unknown Source)
        at org.apache.spark.api.python.PythonWorkerFactory.createSimpleWorker
(PythonWorkerFactory.scala:176)
        ... 14 more

Driver stacktrace:
        at org.apache.spark.scheduler.DAGScheduler.failJobAndIndependentStage
s(DAGScheduler.scala:2672)
        at org.apache.spark.scheduler.DAGScheduler.$anonfun$abortStage$2(DAGS
cheduler.scala:2608)
        at org.apache.spark.scheduler.DAGScheduler.$anonfun$abortStage$2$adap
ted(DAGScheduler.scala:2607)
        at scala.collection.mutable.ResizableArray.foreach(ResizableArray.sca
la:62)
        at scala.collection.mutable.ResizableArray.foreach$(ResizableArray.sc
ala:55)
        at scala.collection.mutable.ArrayBuffer.foreach(ArrayBuffer.scala:49)
        at org.apache.spark.scheduler.DAGScheduler.abortStage(DAGScheduler.sc
ala:2607)
        at org.apache.spark.scheduler.DAGScheduler.$anonfun$handleTaskSetFail
ed$1(DAGScheduler.scala:1182)
        at org.apache.spark.scheduler.DAGScheduler.$anonfun$handleTaskSetFail
ed$1$adapted(DAGScheduler.scala:1182)
        at scala.Option.foreach(Option.scala:407)
        at org.apache.spark.scheduler.DAGScheduler.handleTaskSetFailed(DAGSch
eduler.scala:1182)
        at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.doOnReceiv
e(DAGScheduler.scala:2860)
        at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.onReceive
(DAGScheduler.scala:2802)
        at org.apache.spark.scheduler.DAGSchedulerEventProcessLoop.onReceive
(DAGScheduler.scala:2791)
        at org.apache.spark.util.EventLoop$$anon$1.run(EventLoop.scala:49)
        at org.apache.spark.scheduler.DAGScheduler.runJob(DAGScheduler.scala:
952)
        at org.apache.spark.SparkContext.runJob(SparkContext.scala:2228)
        at org.apache.spark.SparkContext.runJob(SparkContext.scala:2249)
        at org.apache.spark.SparkContext.runJob(SparkContext.scala:2268)
        at org.apache.spark.SparkContext.runJob(SparkContext.scala:2293)
        at org.apache.spark.rdd.RDD.$anonfun$collect$1(RDD.scala:1021)
        at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScop
```

```
e.scala:151)
        at org.apache.spark.rdd.RDDOperationScope$.withScope(RDDOperationScop
e.scala:112)
        at org.apache.spark.rdd.RDD.withScope(RDD.scala:406)
        at org.apache.spark.rdd.RDD.collect(RDD.scala:1020)
        at org.apache.spark.api.python.PythonRDD$.collectAndServe(PythonRDD.s
cala:180)
        at org.apache.spark.api.python.PythonRDD.collectAndServe(PythonRDD.sc
ala)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source)
        at java.lang.reflect.Method.invoke(Unknown Source)
        at py4j.reflection.MethodInvoker.invoke(MethodInvoker.java:244)
        at py4j.reflection.ReflectionEngine.invoke(ReflectionEngine.java:357)
        at py4j.Gateway.invoke(Gateway.java:282)
        at py4j.commands.AbstractCommand.invokeMethod(AbstractCommand.java:13
2)
        at py4j.commands.CallCommand.execute(CallCommand.java:79)
        at py4j.ClientServerConnection.waitForCommands(ClientServerConnectio
n.java:182)
        at py4j.ClientServerConnection.run(ClientServerConnection.java:106)
        at java.lang.Thread.run(Unknown Source)
Caused by: org.apache.spark.SparkException: Python worker failed to connect b
ack.
        at org.apache.spark.api.python.PythonWorkerFactory.createSimpleWorker
(PythonWorkerFactory.scala:189)
        at org.apache.spark.api.python.PythonWorkerFactory.create(PythonWorke
rFactory.scala:109)
        at org.apache.spark.SparkEnv.createPythonWorker(SparkEnv.scala:124)
        at org.apache.spark.api.python.BasePythonRunner.compute(PythonRunner.
scala:164)
        at org.apache.spark.api.python.PythonRDD.compute(PythonRDD.scala:65)
        at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:365)
        at org.apache.spark.rdd.RDD.iterator(RDD.scala:329)
        at org.apache.spark.scheduler.ResultTask.runTask(ResultTask.scala:90)
        at org.apache.spark.scheduler.Task.run(Task.scala:136)
        at org.apache.spark.executor.Executor$TaskRunner.$anonfun$run$3(Execu
tor.scala:548)
        at org.apache.spark.util.Utils$.tryWithSafeFinally(Utils.scala:1504)
        at org.apache.spark.executor.Executor$TaskRunner.run(Executor.scala:5
51)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(Unknown Source)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(Unknown Source)
        ... 1 more
Caused by: java.net.SocketTimeoutException: Accept timed out
        at java.net.DualStackPlainSocketImpl.waitForNewConnection(Native Meth
od)
        at java.net.DualStackPlainSocketImpl.socketAccept(Unknown Source)
        at java.net.AbstractPlainSocketImpl.accept(Unknown Source)
        at java.net.PlainSocketImpl.accept(Unknown Source)
        at java.net.ServerSocket.implAccept(Unknown Source)
        at java.net.ServerSocket.accept(Unknown Source)
        at org.apache.spark.api.python.PythonWorkerFactory.createSimpleWorker
(PythonWorkerFactory.scala:176)
        ... 14 more
```

In [13]:
```
rdd2.collect()
```

Out[13]:
```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```