3_parallelize

February 21, 2022

0.0.1 Understand the usage of "parallelize" method (use glom)

[1]:
```
# Import SparkContext and SparkConf
from pyspark import SparkContext, SparkConf
```

[2]:
```
# Initialize spark
conf = SparkConf().setAppName("createRDD")
sc = SparkContext(conf=conf)
```

```
22/02/21 14:19:26 WARN Utils: Your hostname, ThinkCentre resolves to a loopback
address: 127.0.1.1; using 10.180.5.223 instead (on interface eno1)
22/02/21 14:19:26 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another
address
22/02/21 14:19:27 WARN NativeCodeLoader: Unable to load native-hadoop library
for your platform… using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use
setLogLevel(newLevel).
22/02/21 14:19:28 WARN Utils: Service 'SparkUI' could not bind on port 4040.
Attempting port 4041.
22/02/21 14:19:28 WARN Utils: Service 'SparkUI' could not bind on port 4041.
Attempting port 4042.
```

[3]:
```
# # Create a List of Numbers
lnum = [1, 2, 3, 4, 5, 6, 7, 8]

# Create an RDD with 4 partitions - the data is ditributed now
# Number of partitions is optional
lnumRDD = sc.parallelize(lnum, 4)
```

[4]:
```
sc.parallelize #Press <Shift> + <Tab>
```

[4]:
```
<bound method SparkContext.parallelize of <SparkContext master=local[*]
appName=createRDD>>
```

[5]:
```
lnumRDD.glom().collect()
```

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

[6]:
```
# glom returns an RDD created by coalescing all elements within each partition␣
↪into a list.
```

[7]:
```
# Exercise: Understand the application of glom and its advantages.
```