*Name: Roopam Verma*
*Id: 1001957590*

# LAB-4

Installing Apache-spark, Python, Java, and Scala using homebrew:

```
➜  Lab4 git:(master) ✗ python3 --version
Python 3.10.8
➜  Lab4 git:(master) ✗ java --version
openjdk 17.0.1 2021-10-19
OpenJDK Runtime Environment Temurin-17.0.1+12 (build 17.0.1+12)
OpenJDK 64-Bit Server VM Temurin-17.0.1+12 (build 17.0.1+12, mixed mode)
➜  Lab4 git:(master) ✗
```

Command: brew install apache-spark

```
➜  Lab4 git:(master) ✗ brew install apache-spark
Running `brew update --auto-update`...
==> Auto-updated Homebrew!
Updated 2 taps (homebrew/core and homebrew/cask).

You have 17 outdated formulae installed.
You can upgrade them with brew upgrade
or list them with brew outdated.

Warning: apache-spark 3.3.1 is already installed and up-to-date.
```

## Installing Pyspark

```
➜  Lab4 git:(master) ✗ pip install pyspark
Collecting pyspark
  Using cached pyspark-3.3.1.tar.gz (281.4 MB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: py4j==0.10.9.5 in /Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages (from pyspark) (0.10.9.5)
Building wheels for collected packages: pyspark
  Building wheel for pyspark (setup.py) ... done
  Created wheel for pyspark: filename=pyspark-3.3.1-py2.py3-none-any.whl size=281845513 sha256=e97930c2882b2ae5bf35f693cff0e08e50bb61ae492d42042b5e06310e29edc1
  Stored in directory: /Users/roopamverma/Library/Caches/pip/wheels/91/aa/66/700b503fd714b56462975ab7bf33485ec26677c3c990e67e9a
Successfully built pyspark
Installing collected packages: pyspark
Successfully installed pyspark-3.3.1
➜  Lab4 git:(master) ✗ pyspark --version
22/10/31 21:41:35 WARN Utils: Your hostname, RoopamMac.local resolves to a loopback address: 127.0.0.1; using 10.11.12.108 instead (on interface en0)
22/10/31 21:41:35 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.3.1
      /_/

Using Scala version 2.12.15, OpenJDK 64-Bit Server VM, 17.0.1
Branch HEAD
Compiled by user yumwang on 2022-10-15T09:47:01Z
Revision fbbcf9434ac070dd4ced4fb9efe32899c6db12a9
Url https://github.com/apache/spark
Type --help for more information.
➜  Lab4 git:(master) ✗
```

## After installation, Entering Scala

```
➜  Lab4 git:(master) ✗ spark-shell
22/11/05 01:58:22 WARN Utils: Your hostname, RoopamMac.local resolves to a loopback address: 127.0.0.1; using 10.11.12.105 instead (on interface en0)
22/11/05 01:58:22 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/11/05 01:58:24 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Spark context Web UI available at http://10.11.12.105:4040
Spark context available as 'sc' (master = local[*], app id = local-1667631505477).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.3.1
      /_/

Using Scala version 2.12.15 (OpenJDK 64-Bit Server VM, Java 17.0.1)
Type in expressions to have them evaluated.
Type :help for more information.

scala>
```

# Step 1:

## Program 1:

```
import scala.util.control.Breaks._
import scala.collection.mutable.ListBuffer
object Program1{
        def main(args: Array[String])={
                val rdd1 = scala.io.Source.fromFile("words.txt").getLines.flatMap(_.split("\\W+"))

                val i = rdd1.toList
                val j = i.sortWith(_<_)
                var mylist =  new ListBuffer[String]()
                breakable{
                for(name <- j if (name.startsWith("b") && name.endsWith("t")))
                {
                        mylist += name;
                        println(name);
                        if(mylist.length == 5)
                        break

                }
                }
        }
}
```

## Running Program1.scala using Apache Spark spark-shell:

```
➜  Lab4 git:(master) ✗ spark-shell -i Program1.scala
22/11/16 15:50:41 WARN Utils: Your hostname, RoopamMac.local resolves to a loopback address: 127.0.0.
1; using 10.182.19.245 instead (on interface en0)
22/11/16 15:50:41 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/11/16 15:50:44 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... us
ing builtin-java classes where applicable
Spark context Web UI available at http://10.182.19.245:4040
Spark context available as 'sc' (master = local[*], app id = local-1668635444556).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.3.1
      /_/

Using Scala version 2.12.15 (OpenJDK 64-Bit Server VM, Java 19.0.1)
Type in expressions to have them evaluated.
Type :help for more information.

scala> :load Program1.scala
Loading Program1.scala...
import scala.util.control.Breaks._
import scala.collection.mutable.ListBuffer
defined object Program1
```

Output:

```
scala> Program1 main(Array())
babbitt
babblement
babelet
baboonroot
baboot
```

Program2:

```
import scala.util.control.Breaks._
import scala.collection.mutable.ListBuffer
object Program2{
        def main(args: Array[String])={
                val rdd1 = scala.io.Source.fromFile("words.txt").getLines.flatMap(_.split("\\W+"))
                val i = rdd1.toList
                val check = i.sortBy(_.length)
                val ans = check.reverse
                var mylist =  new ListBuffer[String]()
                breakable{
                for(longest <- ans ){
                mylist += longest;
                println(longest)
                if(mylist.length == 10)
                break
                }
}
}
}
```

Running Program 2 using spark-shell:

```
➜  Lab4 git:(master) ✗ spark-shell -i Program2.scala
22/11/16 15:53:30 WARN Utils: Your hostname, RoopamMac.local resolves to a loopback address: 127.0.0.
1; using 10.182.19.245 instead (on interface en0)
22/11/16 15:53:30 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/11/16 15:53:33 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... us
ing builtin-java classes where applicable
Spark context Web UI available at http://10.182.19.245:4040
Spark context available as 'sc' (master = local[*], app id = local-1668635613480).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.3.1
      /_/

Using Scala version 2.12.15 (OpenJDK 64-Bit Server VM, Java 19.0.1)
Type in expressions to have them evaluated.
Type :help for more information.

scala> :load Program2.scala
Loading Program2.scala...
import scala.util.control.Breaks._
import scala.collection.mutable.ListBuffer
defined object Program2
```

Output:

```
scala> Program2 main(Array())
thyroparathyroidectomize
tetraiodophenolphthalein
scientificophilosophical
pathologicopsychological
formaldehydesulphoxylate
transubstantiationalist
thymolsulphonephthalein
scientificogeographical
pseudolamellibranchiate
Pseudolamellibranchiata
```

Program3:

```
import scala.io.Source
import scala.util.control.Breaks._
import scala.collection.mutable.ListBuffer

object Program3{
        def main(args: Array[String])={
            val source = scala.io.Source.fromFile("file1.txt")
            val lineCount = source.getLines.length
            println(lineCount)
            val mylist = new ListBuffer[String]()
            val words_rdd = scala.io.Source.fromFile("file1.txt").getLines.flatMap(_.split("\\W+"))
            //println(words_rdd)
            val distinct_words_rdd = words_rdd.toList.distinct
            //val distinct_words_count = distinct_words_rdd.count
            //println(distinct_words_count)
            print("The unique words in the file are as follows:")
            for(distinct <- distinct_words_rdd){
            mylist += distinct
            println(distinct)
}
            println(mylist.length)
}
}
```

Running Program3.scala using spark:
Number of lines count:

```
➜  Lab4 git:(master) ✗ pyspark
Python 3.10.8 (main, Oct 13 2022, 09:48:40) [Clang 14.0.0 (clang-1400.0.29.102)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
22/11/16 22:41:38 WARN Utils: Your hostname, RoopamMac.local resolves to a loopback address: 127.0.0.
1; using 10.11.12.117 instead (on interface en0)
22/11/16 22:41:38 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/11/16 22:41:39 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... us
ing builtin-java classes where applicable
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.3.1
      /_/

Using Python version 3.10.8 (main, Oct 13 2022 09:48:40)
Spark context Web UI available at http://10.11.12.117:4040
Spark context available as 'sc' (master = local[*], app id = local-1668660099939).
SparkSession available as 'spark'.
>>> from pyspark.sql.functions import *
>>> f1 = spark.read.text("file1.txt")
>>> lines_num = f1.filter(length("value") > 0)
>>> lines_num.count()
315113
```

Output:
315113

Number of distinct word count:

```
➜  Lab4 git:(master) ✗ pyspark
Python 3.10.8 (main, Oct 13 2022, 09:48:40) [Clang 14.0.0 (clang-1400.0.29.102)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
22/11/16 22:41:38 WARN Utils: Your hostname, RoopamMac.local resolves to a loopback address: 127.0.0.
1; using 10.11.12.117 instead (on interface en0)
22/11/16 22:41:38 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/11/16 22:41:39 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... us
ing builtin-java classes where applicable
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.3.1
      /_/

Using Python version 3.10.8 (main, Oct 13 2022 09:48:40)
Spark context Web UI available at http://10.11.12.117:4040
Spark context available as 'sc' (master = local[*], app id = local-1668660099939).
SparkSession available as 'spark'.
>>> file_1 = spark.read.text("file1.txt")
>>> words_count = file_1.rdd.flatMap(lambda z: z.value.split()).distinct().count()
>>> print(words_count)
182784
```

Output:

182784

Program4:

```
from pyspark.context import SparkContext
from pyspark.context import SparkContext
from operator import add

object Program4{
        def main(args: Array[String])={
         aFile = sc.textFile("file1.txt")
         bFile = sc.textFile("file2.txt")
         cFile = sc.textFile("file3.txt")
         flattenMapa = aFile.flatMap(lambda z: [(x.lower(),1) for x in z.split()]).reduceByKey(add);
         flattenMapb = bFile.flatMap(lambda z: [(x.lower(),1) for x in z.split()]).reduceByKey(add);
         flattenMapc = cFile.flatMap(lambda z: [(x.lower(),1) for x in z.split()]).reduceByKey(add);
         joinedRDD = flattenMapa.intersection(flattenMapb)
         freq = joinedRDD.intersection(flattenMapc)
         freq.take(3)

}

}
```

Running Program4 using pyspark commands:

```
➜  Lab4 git:(master) ✗ pyspark
Python 3.10.8 (main, Oct 13 2022, 09:48:40) [Clang 14.0.0 (clang-1400.0.29.102)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/11/16 17:46:57 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... us
ing builtin-java classes where applicable
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.3.1
      /_/

Using Python version 3.10.8 (main, Oct 13 2022 09:48:40)
Spark context Web UI available at http://roopammac.dhost.uta.edu:4040
Spark context available as 'sc' (master = local[*], app id = local-1668642417486).
SparkSession available as 'spark'.
>>> from pyspark.context import SparkContext
>>> from operator import add
>>> aFile = sc.textFile("file1.txt")
>>> bFile = sc.textFile("file2.txt")
>>> cFile = sc.textFile("file3.txt")
>>> flattenMapa = aFile.flatMap(lambda z: [(x.lower(),1) for x in z.split()]).reduceByKey(add);
>>> flattenMapb = bFile.flatMap(lambda z: [(x.lower(),1) for x in z.split()]).reduceByKey(add);
>>> flattenMapc = cFile.flatMap(lambda z: [(x.lower(),1) for x in z.split()]).reduceByKey(add);
>>> joinedRDD = flattenMapa.intersection(flattenMapb)
>>> freq = joinedRDD.intersection(flattenMapc)
>>> freq.take(3)
```

Output:

```
[('1818.', 2), ('county;', 2), ('illegal', 6)]
>>>
```

Program5:

```
from pyspark.context import SparkConf, SparkContext #
import org.apache.spark.SparkContext
import org.apache.spark.SparkConf

#from pyspark.context import SparkContext
allwords = sc.textFile("words.txt");
allgroups = allwords.groupBy(lambda x:x[0:4]);
print([(x,list(a)) for(x,a) in allgroups.take(10)]);
```

Running Program5 using pyspark commands:

```
→  Lab4 git:(master) ✗ pyspark
Python 3.10.8 (main, Oct 13 2022, 09:48:40) [Clang 14.0.0 (clang-1400.0.29.102)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
22/11/16 16:02:42 WARN Utils: Your hostname, RoopamMac.local resolves to a loopback address: 127.0.0.
1; using 10.182.19.245 instead (on interface en0)
22/11/16 16:02:42 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/11/16 16:02:42 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... us
ing builtin-java classes where applicable
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.3.1
      /_/

Using Python version 3.10.8 (main, Oct 13 2022 09:48:40)
Spark context Web UI available at http://10.182.19.245:4040
Spark context available as 'sc' (master = local[*], app id = local-1668636162912).
SparkSession available as 'spark'.
>>> from pyspark.context import SparkConf, SparkContext
>>> allwords = sc.textFile("words.txt");
>>> allgroups = allwords.groupBy(lambda x:x[0:4]);
>>> print([(x,list(a)) for(x,a) in allgroups.take(10)]);
```

Output:

```
[('aa', ['aa']), ('aal', ['aal']), ('aam', ['aam']), ('Aani', ['Aani']), ('Aaro', ['Aaron', 'Aaronic'
, 'Aaronical', 'Aaronite', 'Aaronitic']), ('abac', ['abac', 'abaca', 'abacate', 'abacay', 'abacinate'
, 'abacination', 'abaciscus', 'abacist', 'aback', 'abactinal', 'abactinally', 'abaction', 'abactor',
'abaculus', 'abacus']), ('abaf', ['abaff', 'abaft']), ('abai', ['abaisance', 'abaiser', 'abaissed']),
 ('abal', ['abalienate', 'abalienation', 'abalone']), ('Abam', ['Abama'])]
```

Step 2:
Python file:

```
from pyspark.sql import SparkSession
from pyspark import SparkContext, SparkConf
from pyspark.sql.functions import desc

spark = SparkSession.builder.master("local[*]").getOrCreate()
sc = spark.sparkContext
dataframe1 = sc.textFile('file1.txt').flatMap(lambda line: line.split(" ")).persist()
dataframe1 = dataframe1.filter(lambda x: x).persist(StorageLevel.DISK_ONLY)
dataframe1 = dataframe1.filter(lambda x: x  not in ['and', 'or', 'that', 'the', 'a', 'an', 'is', 'are
', 'have']).persist()

dataframe2 = dataframe1.map(lambda word: (word, 1)).reduceByKey(lambda a,b:a +b).persist()
dataframe3 = spark.createDataFrame(dataframe2, ['word', 'frequency'])
dataframe3.orderBy(desc('frequency')).show(10)
```

```
➜  Lab4 git:(master) ✗ pyspark
Python 3.10.8 (main, Oct 13 2022, 09:48:40) [Clang 14.0.0 (clang-1400.0.29.102)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/11/16 18:05:34 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... us
ing builtin-java classes where applicable
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.3.1
      /_/

Using Python version 3.10.8 (main, Oct 13 2022 09:48:40)
Spark context Web UI available at http://roopammac.dhost.uta.edu:4040
Spark context available as 'sc' (master = local[*], app id = local-1668643535376).
SparkSession available as 'spark'.
>>> from pyspark.sql import SparkSession
>>> from pyspark import SparkContext, SparkConf
>>> from pyspark.sql.functions import desc
>>> from pyspark import StorageLevel
>>> spark = SparkSession.builder.master("local[*]").getOrCreate()
>>> sc = spark.sparkContext
>>> df1 = sc.textFile('file1.txt').flatMap(lambda line: line.split(" ")).persist()
>>> df1 = df1.filter(lambda x: x).persist(StorageLevel.DISK_ONLY)
>>> df1 = df1.filter(lambda x: x  not in ['and', 'or', 'that', 'the', 'a', 'an', 'is', 'are', 'have']
).persist()
>>> df2 = df1.map(lambda word: (word, 1)).reduceByKey(lambda a,b:a +b).persist()
>>> df3 = spark.createDataFrame(df2, ['word', 'frequency'])
/opt/homebrew/Cellar/apache-spark/3.3.1/libexec/python/lib/pyspark.zip/pyspark/shuffle.py:65: UserWar
ning: Please install psutil to have better support with spilling
/opt/homebrew/Cellar/apache-spark/3.3.1/libexec/python/lib/pyspark.zip/pyspark/shuffle.py:65: UserWar
ning: Please install psutil to have better support with spilling
>>> df3.orderBy(desc('frequency')).show(10)
/opt/homebrew/Cellar/apache-spark/3.3.1/libexec/python/lib/pyspark.zip/pyspark/shuffle.py:65: UserWar
ning: Please install psutil to have better support with spilling
+----+---------+
|word|frequency|
+----+---------+
|  of|    90412|
|  to|    69806|
|  in|    46542|
|   I|    43759|
| his|    24774|
|  he|    23501|
|with|    22936|
| was|    22915|
|  be|    20749|
| for|    19528|
+----+---------+
only showing top 10 rows
```

## For DISK ONLY:

Output on running command: spark-submit step2_prog_disk.py

```
22/11/16 18:21:32 INFO DAGScheduler: ResultStage 3 (showString at DirectMethodHandleAccessor.java:104
) finished in 0.329 s
22/11/16 18:21:32 INFO DAGScheduler: Job 1 is finished. Cancelling potential speculative or zombie ta
sks for this job
22/11/16 18:21:32 INFO TaskSchedulerImpl: Killing all running tasks in stage 3: Stage finished
22/11/16 18:21:32 INFO DAGScheduler: Job 1 finished: showString at DirectMethodHandleAccessor.java:10
4, took 0.334360 s
22/11/16 18:21:32 INFO CodeGenerator: Code generated in 6.779292 ms
22/11/16 18:21:32 INFO CodeGenerator: Code generated in 5.547792 ms
+----+---------+
|word|frequency|
+----+---------+
|  of|    90412|
|  to|    69806|
|  in|    46542|
|   I|    43759|
| his|    24774|
|  he|    23501|
|with|    22936|
| was|    22915|
|  be|    20749|
| for|    19528|
+----+---------+
only showing top 10 rows
```

It took 0.334360 seconds.

## For MEMORY_ONLY:

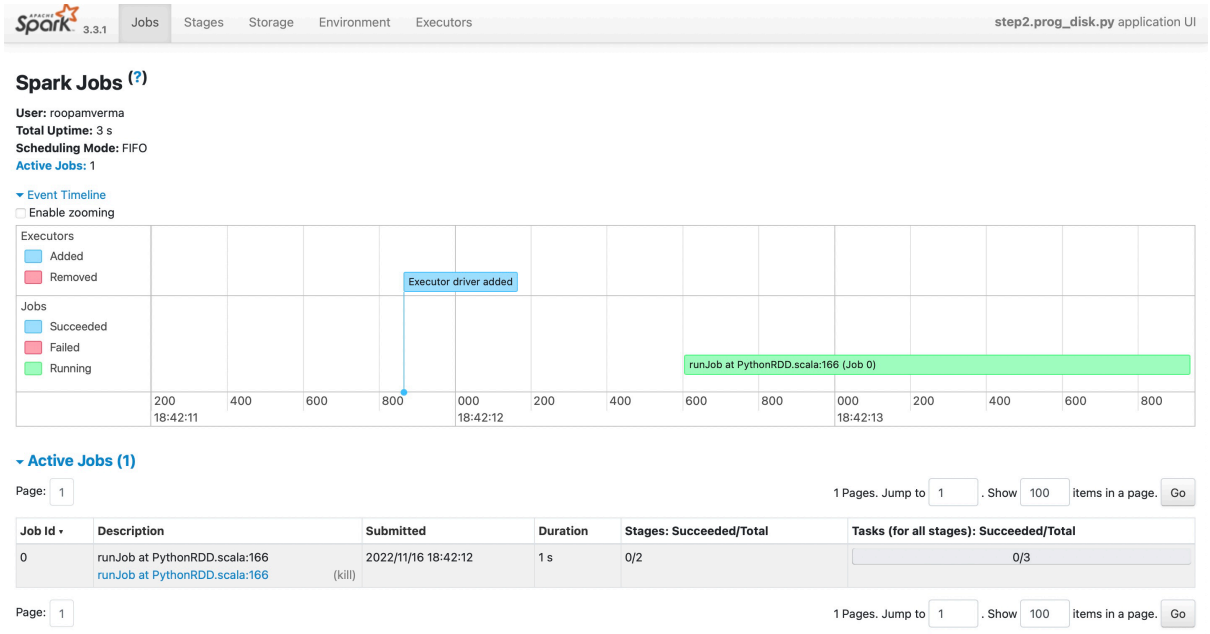
Python file for memory only storage:

```
from pyspark.sql import SparkSession
from pyspark import SparkContext, SparkConf
from pyspark.sql.functions import desc
from pyspark import StorageLevel

spark = SparkSession.builder.master("local[*]").getOrCreate()
sc = spark.sparkContext
dataframe1 = sc.textFile('file1.txt').flatMap(lambda line: line.split(" ")).persist()
dataframe1 = dataframe1.filter(lambda x: x).persist(StorageLevel.MEMORY_ONLY)
dataframe1 = dataframe1.filter(lambda x: x  not in ['and', 'or', 'that', 'the', 'a', 'an', 'is', 'are
', 'have']).persist()

dataframe2 = dataframe1.map(lambda word: (word, 1)).reduceByKey(lambda a,b:a +b).persist()
dataframe3 = spark.createDataFrame(dataframe2, ['word', 'frequency'])
dataframe3.orderBy(desc('frequency')).show(10)
```

Output on running command: spark-submit step2_prog_memory.py

```
22/11/16 18:31:34 INFO DAGScheduler: ResultStage 3 (showString at DirectMethodHandleAccessor.java:104
) finished in 0.326 s
22/11/16 18:31:34 INFO DAGScheduler: Job 1 is finished. Cancelling potential speculative or zombie ta
sks for this job
22/11/16 18:31:34 INFO TaskSchedulerImpl: Killing all running tasks in stage 3: Stage finished
22/11/16 18:31:34 INFO DAGScheduler: Job 1 finished: showString at DirectMethodHandleAccessor.java:10
4, took 0.329449 s
22/11/16 18:31:34 INFO CodeGenerator: Code generated in 6.349208 ms
22/11/16 18:31:34 INFO CodeGenerator: Code generated in 4.588625 ms
+----+---------+
|word|frequency|
+----+---------+
|  of|    90412|
|  to|    69806|
|  in|    46542|
|   I|    43759|
| his|    24774|
|  he|    23501|
|with|    22936|
| was|    22915|
|  be|    20749|
| for|    19528|
+----+---------+
only showing top 10 rows
```

We can monitor job on spark UI:
http://localhost:4040/jobs




We can monitor executors:
http://localhost:4040/executors

# Executors

Show Additional Metrics

## Summary

| | RDD Blocks | Storage Memory | On Heap Storage Memory | Off Heap Storage Memory | Disk Used | Cores | Active Tasks | Failed Tasks | Complete Tasks | Total Tasks | Task Time (GC Time) | Input | Shuffle Read | Shuffle Write | Excluded |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Active(1)** | 3 | 13.2 MiB / 434.4 MiB | 13.2 MiB / 434.4 MiB | 0.0 B / 0.0 B | 6.4 MiB | 8 | 2 | 0 | 0 | 2 | 3 s (0.0 ms) | 0.0 B | 0.0 B | 0.0 B | 0 |
| **Total(1)** | 3 | 13.2 MiB / 434.4 MiB | 13.2 MiB / 434.4 MiB | 0.0 B / 0.0 B | 6.4 MiB | 8 | 2 | 0 | 0 | 2 | 3 s (0.0 ms) | 0.0 B | 0.0 B | 0.0 B | 0 |
| **Dead(0)** | 0 | 0.0 B / 0.0 B | 0.0 B / 0.0 B | 0.0 B / 0.0 B | 0.0 B | 0 | 0 | 0 | 0 | 0 | 0.0 ms (0.0 ms) | 0.0 B | 0.0 B | 0.0 B | 0 |

## Executors

Show 20 entries Search:

| Executor ID | Address | Status | RDD Blocks | Storage Memory | On Heap Storage Memory | Off Heap Storage Memory | Peak JVM Memory OnHeap / OffHeap | Peak Execution Memory OnHeap / OffHeap | Peak Storage Memory OnHeap / OffHeap | Peak Pool Memory Direct / Mapped | Disk Used | Cores | Resources | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| driver | roopammac.dhost.uta.edu:63434 | Active | 3 | 13.2 MiB / 434.4 MiB | 13.2 MiB / 434.4 MiB | 0.0 B / 0.0 B | 0.0 B / 0.0 B | 0.0 B / 0.0 B | 0.0 B / 0.0 B | 0.0 B / 0.0 B | 6.4 MiB | 8 | | 0 |

Showing 1 to 1 of 1 entries Previous 1 Next

# Step 3:

Running the step 2 python file step2_prog_disk.py using yarn by creating a cluster:

```
➜  Lab4 git:(master) ✗ Spark-submit step2.prog_disk.py –master yarn –deploy-mode cluster
22/11/16 20:06:53 WARN Utils: Your hostname, RoopamMac.local resolves to a loopback address: 127.0.0.
1; using 10.11.12.117 instead (on interface en0)
22/11/16 20:06:53 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
22/11/16 20:06:55 INFO SparkContext: Running Spark version 3.3.1
22/11/16 20:06:55 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... us
ing builtin-java classes where applicable
22/11/16 20:06:55 INFO ResourceUtils: ==============================================================
22/11/16 20:06:55 INFO ResourceUtils: No custom resources configured for spark.driver.
22/11/16 20:06:55 INFO ResourceUtils: ==============================================================
22/11/16 20:06:55 INFO SparkContext: Submitted application: step2.prog_disk.py
22/11/16 20:06:55 INFO ResourceProfile: Default ResourceProfile created, executor resources: Map(core
s -> name: cores, amount: 1, script: , vendor: , memory -> name: memory, amount: 1024, script: , vend
or: , offHeap -> name: offHeap, amount: 0, script: , vendor: ), task resources: Map(cpus -> name: cpu
s, amount: 1.0)
```

Output:

```
22/11/16 20:07:02 INFO DAGScheduler: ResultStage 3 (showString at DirectMethodHandleAccessor.java:104
) finished in 0.336 s
22/11/16 20:07:02 INFO DAGScheduler: Job 1 is finished. Cancelling potential speculative or zombie ta
sks for this job
22/11/16 20:07:02 INFO TaskSchedulerImpl: Killing all running tasks in stage 3: Stage finished
22/11/16 20:07:02 INFO DAGScheduler: Job 1 finished: showString at DirectMethodHandleAccessor.java:10
4, took 0.340129 s
22/11/16 20:07:02 INFO CodeGenerator: Code generated in 8.19825 ms
22/11/16 20:07:02 INFO CodeGenerator: Code generated in 6.67875 ms
+----+---------+
|word|frequency|
+----+---------+
|  of|    90412|
|  to|    69806|
|  in|    46542|
|   I|    43759|
| his|    24774|
|  he|    23501|
|with|    22936|
| was|    22915|
|  be|    20749|
| for|    19528|
+----+---------+
only showing top 10 rows
```

## Step 4:

Q1. For a Kubernetes cluster, how can you dynamically add/remove nodes during application runtime?

In a Kubernetes cluster, nodes can be deleted or added without impacting the current deployed application pods. In a case when a node is removed, on which application pod is running Kubernetes drain the node and reschedule the pods on other available nodes. There are toleration added in pod for not ready and unreachable state which triggers based on the configured time mentioned in Kubernetes controller.

```
→  ~ git:(master) ✗ kubectl describe pod nginx-deployment-7fb96c846b-k4wzh
Name:             nginx-deployment-7fb96c846b-k4wzh
Namespace:        default
Priority:         0
Service Account:  default
Node:             docker-desktop/192.168.65.4
Start Time:       Wed, 16 Nov 2022 20:56:35 -0600
Labels:           app=nginx
                  pod-template-hash=7fb96c846b
Annotations:      <none>
Status:           Running
IP:               10.1.0.29
IPs:
  IP:           10.1.0.29
Controlled By:  ReplicaSet/nginx-deployment-7fb96c846b
Containers:
  nginx:
    Container ID:   docker://cc62a8aefc56aaa17f42c1cf8d5178c0e07c99963d4b9f330abe1fd78cb0aa21
    Image:          nginx:1.14.2
    Image ID:       docker-pullable://nginx@sha256:f7988fb6c02e0ce69257d9bd9cf37ae20a60f1df7563c3a2a6abe24160306b8d
    Port:           80/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Wed, 16 Nov 2022 20:56:36 -0600
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-sjsrr (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  kube-api-access-sjsrr:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:                      <none>
```

Eviction of pod and re-scheduling on a different worker node also depends upon resources like PersistentVolumeClaims which are binded to a particular node.

Q2. How can you ask the Kube-scheduler to stop/resume scheduling new pods on certain nodes?

Scheduling can be disabled/resumed on a Kubernetes node by using command:

kubectl cordon <NODENAME>

**Demo:**

Getting nodes in cluster

```
→  ~ git:(master) ✗ kubectl get nodes
NAME             STATUS   ROLES           AGE   VERSION
docker-desktop   Ready    control-plane   8d    v1.25.0
```

Getting deployment set and pods in cluster:
We have one deployment set with 4 replicas i.e 4 pods running in nginx-deployment

```
→  ~ git:(master) ✗ kubectl get deployment
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   4/4     4            4           8d
→  ~ git:(master) ✗ kubectl get pods
NAME                                READY   STATUS    RESTARTS      AGE
nginx-deployment-7fb96c846b-6dzzv   1/1     Running   2 (17m ago)   8d
nginx-deployment-7fb96c846b-k5x8s   1/1     Running   0             7m59s
nginx-deployment-7fb96c846b-p8k4v   1/1     Running   0             14m
nginx-deployment-7fb96c846b-pjpg7   1/1     Running   2 (17m ago)   8d
```

Now, stopping scheduling on node docker-desktop using cordon command:

```
→  ~ git:(master) ✗ kubectl cordon docker-desktop
node/docker-desktop cordoned
→  ~ git:(master) ✗ kubectl get nodes
NAME             STATUS                     ROLES           AGE   VERSION
docker-desktop   Ready,SchedulingDisabled   control-plane   8d    v1.25.0
```

From the image above it is seen that docker-desktop node has been disabled for scheduling.

As, nginx-deployment has replica-set as 4, so deleting one pod so that the pod will get re-scheduled automatically by Kubernetes to maintain the replica-set.

On deleting below highlighted pod, Kubernetes created pod:nginx-deployment-7fb96c846b-k4wzh but it is in *pending* state as scheduling is disabled on node; docker-desktop in cluster.

```
→  ~ git:(master) ✗ kubectl get pods
NAME                                READY   STATUS    RESTARTS      AGE
nginx-deployment-7fb96c846b-6dzzv   1/1     Running   2 (20m ago)   8d
nginx-deployment-7fb96c846b-k5x8s   1/1     Running   0             11m
nginx-deployment-7fb96c846b-p8k4v   1/1     Running   0             18m
nginx-deployment-7fb96c846b-pjpg7   1/1     Running   2 (20m ago)   8d
→  ~ git:(master) ✗ kubectl delete po nginx-deployment-7fb96c846b-6dzzv
pod "nginx-deployment-7fb96c846b-6dzzv" deleted
→  ~ git:(master) ✗ kubectl get pods
NAME                                READY   STATUS    RESTARTS      AGE
nginx-deployment-7fb96c846b-k4wzh   0/1     Pending   0             15s
nginx-deployment-7fb96c846b-k5x8s   1/1     Running   0             12m
nginx-deployment-7fb96c846b-p8k4v   1/1     Running   0             18m
nginx-deployment-7fb96c846b-pjpg7   1/1     Running   2 (21m ago)   8d
→  ~ git:(master) ✗
```

On running command
kubectl describe pod nginx-deployment-7fb96c846b-k4wzh

A failed scheduling event occurred which is due to cordon command which disabled the scheduling of node: docker-desktop

```
→  ~ git:(master) ✗ kubectl describe pod nginx-deployment-7fb96c846b-k4wzh
Name:             nginx-deployment-7fb96c846b-k4wzh
Namespace:        default
Priority:         0
Service Account:  default
Node:             <none>
Labels:           app=nginx
                  pod-template-hash=7fb96c846b
Annotations:      <none>
Status:           Pending
IP:
IPs:              <none>
Controlled By:    ReplicaSet/nginx-deployment-7fb96c846b
Containers:
  nginx:
    Image:        nginx:1.14.2
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-sjsrr (ro)
Conditions:
  Type           Status
  PodScheduled   False
Volumes:
  kube-api-access-sjsrr:
    Type:                     Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:   3607
    ConfigMapName:            kube-root-ca.crt
    ConfigMapOptional:        <nil>
    DownwardAPI:              true
QoS Class:                    BestEffort
Node-Selectors:               <none>
Tolerations:                  node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                              node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type     Reason            Age    From               Message
  ----     ------            ----   ----               -------
  Warning  FailedScheduling  6m55s  default-scheduler  0/1 nodes are available: 1 node(s) had untolerated taint {node.kubernetes.io/unschedulable: }, 1 node(s) were unschedulable. preemption: 0/1 nodes are available: 1 Preemption is not helpful for scheduling.
  Warning  FailedScheduling  96s    default-scheduler  0/1 nodes are available: 1 node(s) had untolerated taint {node.kubernetes.io/unschedulable: }, 1 node(s) were unschedulable. preemption: 0/1 nodes are available: 1 Preemption is not helpful for scheduling.
```

Now resuming scheduling on node *docker-desktop*, so that *pending* pod can be scheduled and change its status to *Running*

*Command: kubectl uncordon docker-desktop*

```
→  ~ git:(master) ✗ kubectl uncordon docker-desktop
node/docker-desktop uncordoned
→  ~ git:(master) ✗ kubectl get pods
NAME                                READY   STATUS    RESTARTS      AGE
nginx-deployment-7fb96c846b-k4wzh   1/1     Running   0             11m
nginx-deployment-7fb96c846b-k5x8s   1/1     Running   0             23m
nginx-deployment-7fb96c846b-p8k4v   1/1     Running   0             29m
nginx-deployment-7fb96c846b-pjpg7   1/1     Running   2 (32m ago)   8d
```

As soon as docker-desktop node is uncordoned the pending pods got scheduled and is in running state.

On describing the above pod again:
Command: kubectl describe pod nginx-deployment-7fb96c846b-k4wzh

```
Events:
  Type     Reason            Age    From               Message
  ----     ------            ----   ----               -------
  Warning  FailedScheduling  13m    default-scheduler  0/1 nodes are available: 1 node(s) had untolerated taint {node.kubernetes.io/unschedulable: }, 1 node(s) were unschedulable. preemption: 0/1 nodes are available: 1 Preemption is not helpful for scheduling.
  Warning  FailedScheduling  8m15s  default-scheduler  0/1 nodes are available: 1 node(s) had untolerated taint {node.kubernetes.io/unschedulable: }, 1 node(s) were unschedulable. preemption: 0/1 nodes are available: 1 Preemption is not helpful for scheduling.
  Normal   Scheduled         2m28s  default-scheduler  Successfully assigned default/nginx-deployment-7fb96c846b-k4wzh to docker-desktop
  Normal   Pulled            2m27s  kubelet            Container image "nginx:1.14.2" already present on machine
  Normal   Created           2m27s  kubelet            Created container nginx
  Normal   Started           2m27s  kubelet            Started container nginx
```

Q3. Which information are used by the scheduler in order to make scheduling decisions (i.e., assigning nodes to pods)?

Kubernetes uses following methods to assign the pod on a node:

- Node Selector- The most fundamental kind of node selection constraint is nodeSelector. The node labels one wants the target node to have may be specified by adding the nodeSelector field to Pod definition. Only nodes with each of the labels one gives are selected for the Pod's scheduling by Kubernetes.

- Affinity and Anti-Affinity: Another way to constrain a pod to be scheduled on a node, it provides more flexibility than node selector. It can be of type soft or preferred which allows the scheduler to assign pod to a node even if some constrains are not met. With affinity we can constrain pods based on node called as node affinity and based on pods running on nodes called as pod affinity.

  Example:

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: with-node-affinity
spec:
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
        - matchExpressions:
          - key: topology.kubernetes.io/zone
            operator: In
            values:
            - antarctica-east1
            - antarctica-west1
```

- Node Name: It is a field in Pod spec. The scheduler ignores the Pod and the kubelet on the designated node tries to install the Pod there if the nodeName field is not empty. Affinity and anti-affinity rules are superseded by using nodeName instead of nodeSelector.

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx
  nodeName: kube-01
```

- Pod Topology Spread Constrains: To regulate how Pods are distributed among the cluster's failure-domains, such as regions, zones, and nodes, as well as across any additional topology, one may utilize topology spread constraints. This might be done to increase usage, predicted availability, or performance.

## Q4. How your-custom application, as a pod, can access similar information?

This can be done using either Config Map or Custom resources in Kubernetes.

Config Map: An API object called a ConfigMap is used to store non-confidential information in key-value pairs. ConfigMaps can be used by pods as command-line arguments, environment variables, or configuration files in a disk.

It is used mainly when:

- Existing configuration file formats, such mysql.cnf or pom.xml, are well-documented.
- The complete configuration should be stored in a single ConfigMap key.

- The configuration file's primary purpose is to be consumed by an application running in a Pod on your cluster so that it may configure itself.
- Instead of using the Kubernetes API, file consumers prefer to use a file in a pod or an environment variable in a pod.
- When the file is updated, you wish to deploy rolling changes using Deployment and other methods.

Custom Resources:
In the Kubernetes API, a resource is an endpoint that keeps a group of similar API objects; for instance, the built-in pods resource has a group of Pod objects.

A custom resource is a Kubernetes API extension that is not always included in a default Kubernetes installation. It shows that a certain Kubernetes installation has been customized. Kubernetes has become more modular, nonetheless, as a result of the use of custom resources in many key activities.

Custom resources can be used when:
- To create and update the new resource, utilize the client libraries and CLIs for Kubernetes.
- You need kubectl to provide top-level support; for instance, kubectl get my-object object-name.
- You want to create new automation that either CRUDs one object after it receives updates, or the other way around.
- You want to create automation that manages object updates.
- Use Kubernetes API conventions like. metadata, status, and spec.
- You want the object to be a summary of other resources or an abstraction over a group of managed resources.