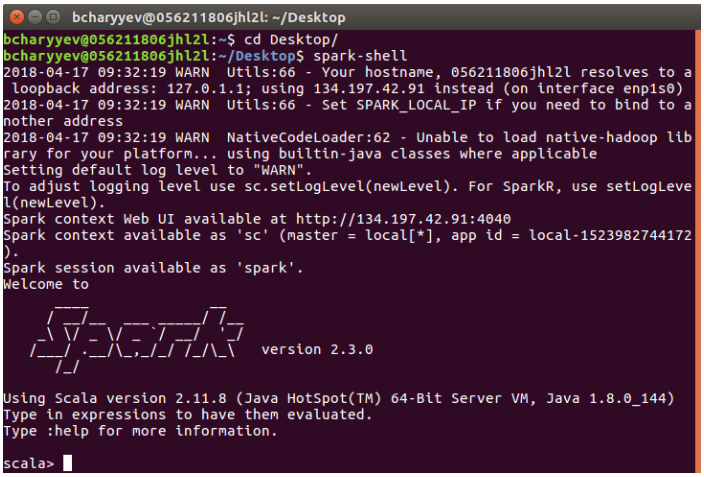
Batyr Charyyev
CS691 - Data Intensive Computing
HW2

Questions.
1. Launch the Spark shell. (2pt)

Answer:



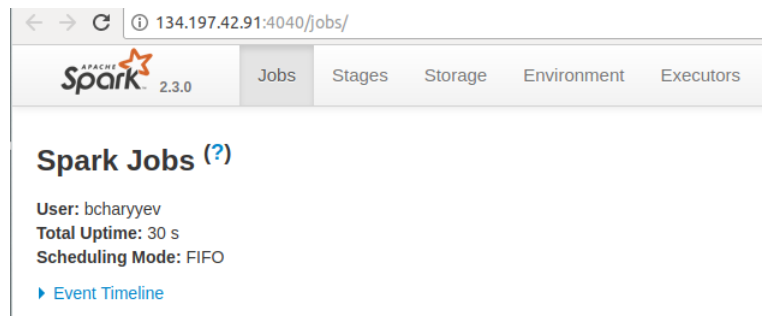


Figure 2. Web-UI

Figure 1.

2. Make a parallel collection of Array(1, 2, 3, 4, 5) and sum up all its elements. (2pt)

Answer:

```
scala> val nums = sc.parallelize(Array(1,2,3,4,5))
nums: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[1] at parallelize at <console>:24

scala> nums.reduce((x,y) => x+y)
res4: Int = 15
```

Figure 3. Implementation with Reduce

```
scala> val nums = sc.parallelize(Array(1,2,3,4,5))
nums: org.apache.spark.rdd.RDD[Int] = ParallelCollectionRDD[0] at parallelize at <console>:24

scala> val accum = sc.accumulator(0)
warning: there were two deprecation warnings; re-run with -deprecation for details
accum: org.apache.spark.Accumulator[Int] = 0

scala> nums.foreach(x => accum += x)

scala> accum.value
res1: Int = 15
```

Figure 4. Implementation with accumulator

3. Create an RDD named pagecounts from the input file hamlet (3pt)

Answer:

```
scala> val pagecounts=sc.textFile("/home/bcharyyev/Desktop/EnginHW2/hamlet")
pagecounts: org.apache.spark.rdd.RDD[String] = /home/bcharyyev/Desktop/EnginHW2/hamlet MapPartitionsRDD[1] at textFile at <console>:24

scala> pagecounts.
++              count                 foreach                 isEmpty                 persist              saveAsTextFile   toJavaRDD
aggregate       countApprox           foreachAsync            iterator                pipe                 setName          toLocalIterator
cache           countApproxDistinct   foreachPartition        keyBy                   preferredLocations   sortBy           toString
canEqual        countAsync            foreachPartitionAsync   localCheckpoint         productArity         sparkContext     top
cartesian       countByValue          getCheckpointFile       map                     productElement       subtract         treeAggregate
checkpoint      countByValueApprox    getNumPartitions        mapPartitions           productIterator      take             treeReduce
coalesce        dependencies          getStorageLevel         mapPartitionsWithIndex  productPrefix        takeAsync        union
collect         distinct              glom                    max                     randomSplit          takeOrdered      unpersist
collectAsync    filter                groupBy                 min                     reduce               takeSample       zip
compute         first                 id                      name                    repartition          toDF             zipPartitions
context         flatMap               intersection            partitioner             sample               toDS             zipWithIndex
copy            fold                  isCheckpointed          partitions              saveAsObjectFile     toDebugString    zipWithUniqueId

scala> pagecounts.take(10)
res0: Array[String] = Array("", Project Gutenberg EBook of Hamlet, by William Shakespeare, "", This eBook is for the use of anyone anywhere in the U
nited States and, most other parts of the world at no cost and with almost no, restrictions whatsoever.  You may copy it, give it away or re-use it,
 under the terms of the Project Gutenberg License included with this, eBook or online at www.gutenberg.org.  If you are not located in the, United S
tates, you'll have to check the laws of the country where you, are located before using this ebook.)

scala>
```

Figure 5. Pagecount and first 10-line.

4. Get the first 10 lines of hamlet (i.e., first 10 records of pagecounts). (3pt)

Answer: Provided on Figure 5.

5. Make a more readable print of the step 4. (3pt)

Answer:

```
scala> pagecounts.take(10).foreach(println)

Project Gutenberg EBook of Hamlet, by William Shakespeare

This eBook is for the use of anyone anywhere in the United States and
most other parts of the world at no cost and with almost no
restrictions whatsoever.  You may copy it, give it away or re-use it
under the terms of the Project Gutenberg License included with this
eBook or online at www.gutenberg.org.  If you are not located in the
United States, you'll have to check the laws of the country where you
are located before using this ebook.

scala>
```

Figure 6. More readable format

6. Count the total records in the data set pagecounts, and confirm its correctness by comparing the result with the Bash wc command: wc -l hamlet . (3pt)

Answer:

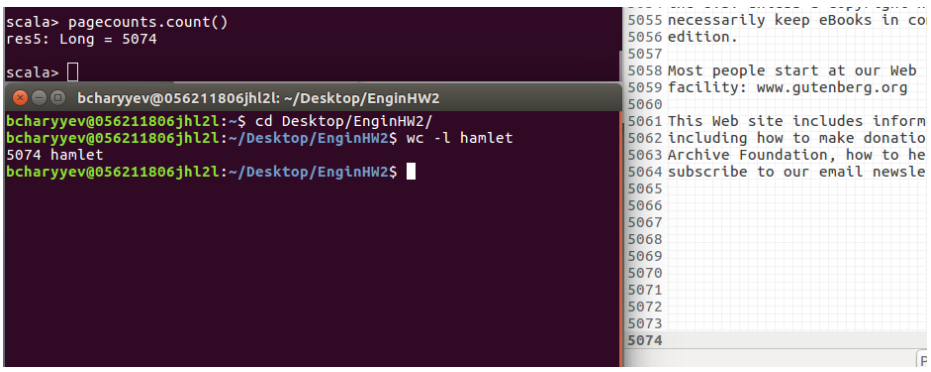


Figure 7. Count

7. Filter the data set pagecounts and return the items that have the word “this”. (5pt)

Answer: Screenshot of only few lines provided

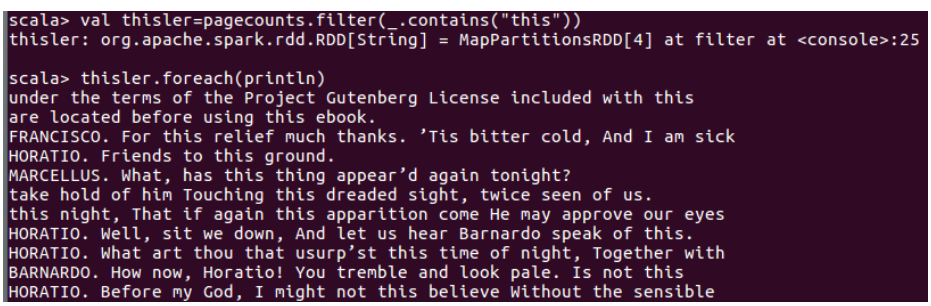


Figure 8. Filter

8. Cache the new data set in memory, to avoid reading from disks. Show cached RDD in web interface (5pt)

Answer:

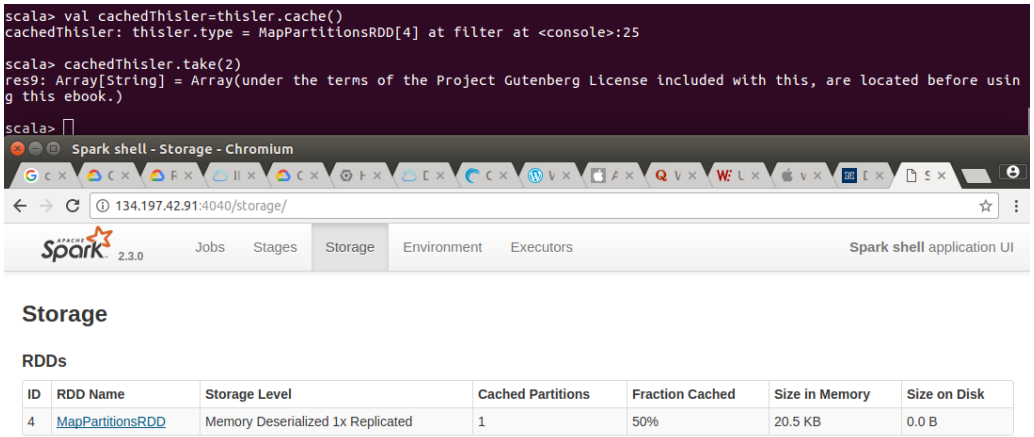


Figure 9. Cache

9. Find 5 lines with the most number of words. Print lines and the number of words(6pt)

Answer:

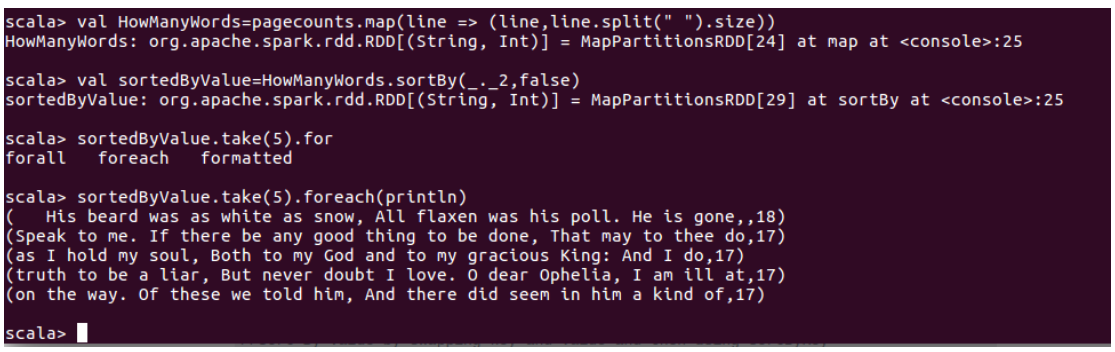


Figure 10. Top 5 lines with most word

10. Count the total number words. (3pt)

Answer: I used HowManyWords RDD from question 10. In each entry values are number of words in that line so I just add up all values.

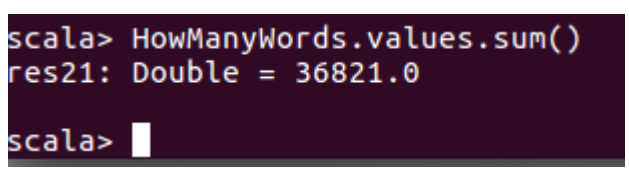


Figure 11. Total number of words

11. Count the number of unique words. (5pt)

Answer: Here I provided both split by space and split by words. When we split by words we get more accurate results.

```
scala> val f1=pagecounts.flatMap(_.split(" "))
f1: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[55] at flatMap at <console>:25

scala> val f2=f1.distinct
f2: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[58] at distinct at <console>:25

scala> f2.count()
res37: Long = 8394

scala> val f1=pagecounts.flatMap(_.split("\\W+"))
f1: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[59] at flatMap at <console>:25

scala> val f2=f1.distinct
f2: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[62] at distinct at <console>:25

scala> f2.count()
res38: Long = 5700

scala>
```

Figure 12. Unique words

12. Count the number of each word. (10pt)

Answer: I provided only 10 lines of output.

```
scala> val pagecounts = sc.textFile("/home/bcharyyev/Desktop/EnginHW2/hamlet")
pagecounts: org.apache.spark.rdd.RDD[String] = /home/bcharyyev/Desktop/EnginHW2/hamlet MapPartitionsRDD[1] at textFile at <console>:24

scala> val wordcounter=pagecounts.flatMap(l => l.split("\\W+")).map(word => (word,1)).reduceByKey(_ + _)
wordcounter: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[4] at reduceByKey at <console>:25

scala> wordcounter.take(10).foreach(println)
(pate,4)
(Unless,2)
(young,15)
(Bestow,1)
(ll,80)
(lug,1)
(shot,8)
(turneth,1)
(afternoon,1)
(dole,1)

scala>
```

Figure 13. Number of each word

13. Show the jobs for Q12 in web interface (3pt)

Answer: Figure-14 shows job created, Figure-15 shows detailed information of that job. As you can see it provides transformations and actions performed to obtain result.

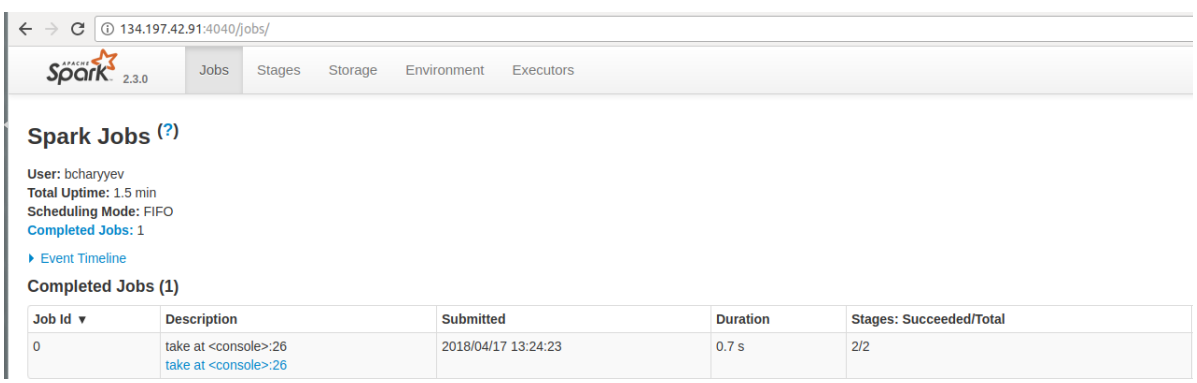


Figure 14. Job created on Web-UI

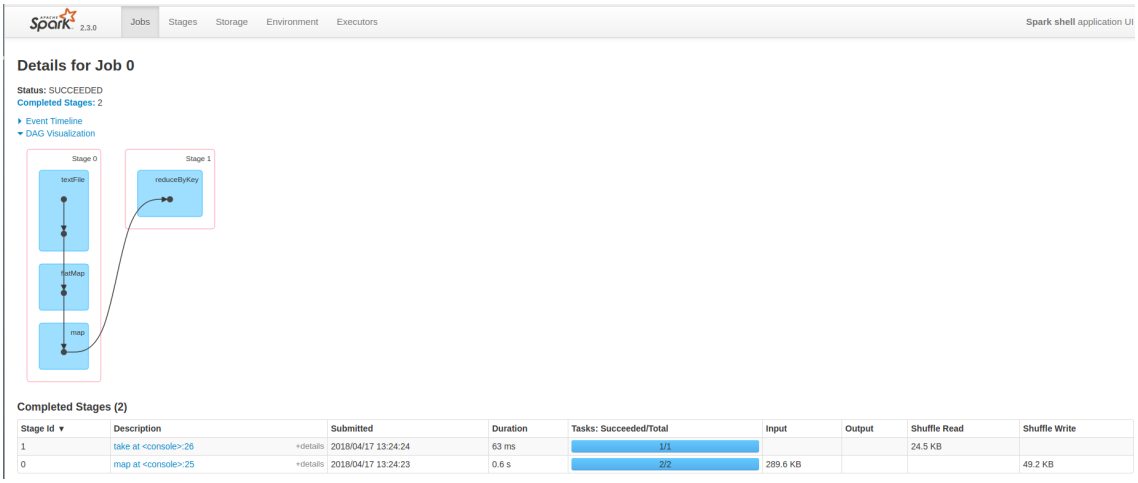


Figure 15. Job created on Web-UI detailed

14. Save the data set in a text file. (3pt)

Answer: We used command on Figure-16 to save data set in text file. You can find saved data set in "wordcountersaved" folder.

```
scala> wordcounter.saveAsTextFile("/home/bcharyyev/Desktop/EnginHW2/wordcountersaved")

scala>
```

Figure 16. saveAsTextFile

15. Collect the word counts in the shell. (4pt)

Answer:

```
scala> wordcounter.collect()
res1: Array[(String, Int)] = Array((pate,4), (Unless,2), (young,15), (Bestow,1), (ll,80), (lug,1), (shot,8), (turneth,1), (a
fternoon,1), (dole,1), (order,3), (Thaw,1), (apprehension,2), (Friend,2), (behind,5), (Fordo,1), (convoy,1), (pigeon,1), (be
en,26), (conjure,1), (Sprinkle,1), (bout,1), (rots,1), (harlot,2), (jade,1), (reserve,1), (breath,9), (knows,3), (likeness,1
), (PLEASE,1), (file,2), (CONTRACT,1), (secrecy,3), (tune,2), (FORTINBRAS,7), (General,3), (are,144), (records,1), (Pretty,1
), (Under,2), (smooth,2), (cart,1), (shut,1), (grant,1), (brief,4), (IS,1), (morn,3), (element,1), (tush,1), (stern,1), (saf
ely,1), (swamp,1), (Mission,1), (arriv,1), (wager,7), (throne,2), (000,1), (son,21), (dead,30), (midnight,1), (aptly,1), (LO
RDS,1), (thus,32), (ulcerous,1), (glares,1), (pursue,...
scala>
```

Figure 17. Collect