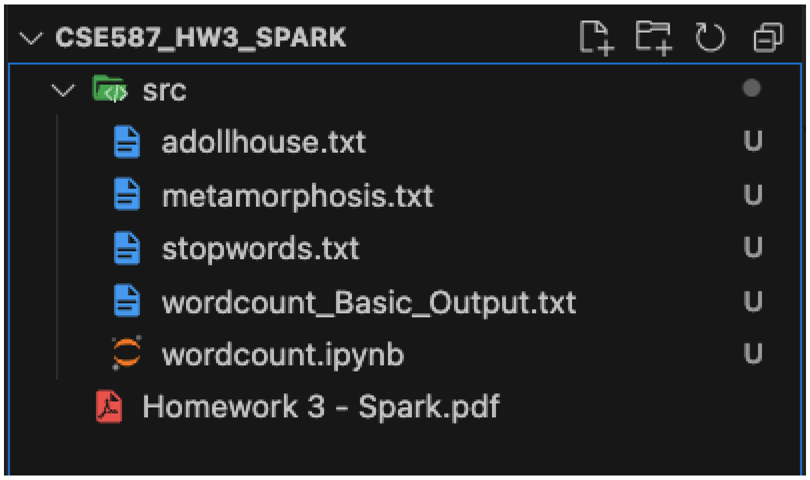
Homework #3 - Spark

Name - Nikhil Gupta
UB ID - ngupta22

Folder Structure



```
CSE587_HW3_SPARK
  src
    adollhouse.txt              U
    metamorphosis.txt           U
    stopwords.txt               U
    wordcount_Basic_Output.txt  U
    wordcount.ipynb             U
  Homework 3 - Spark.pdf
```

Required files

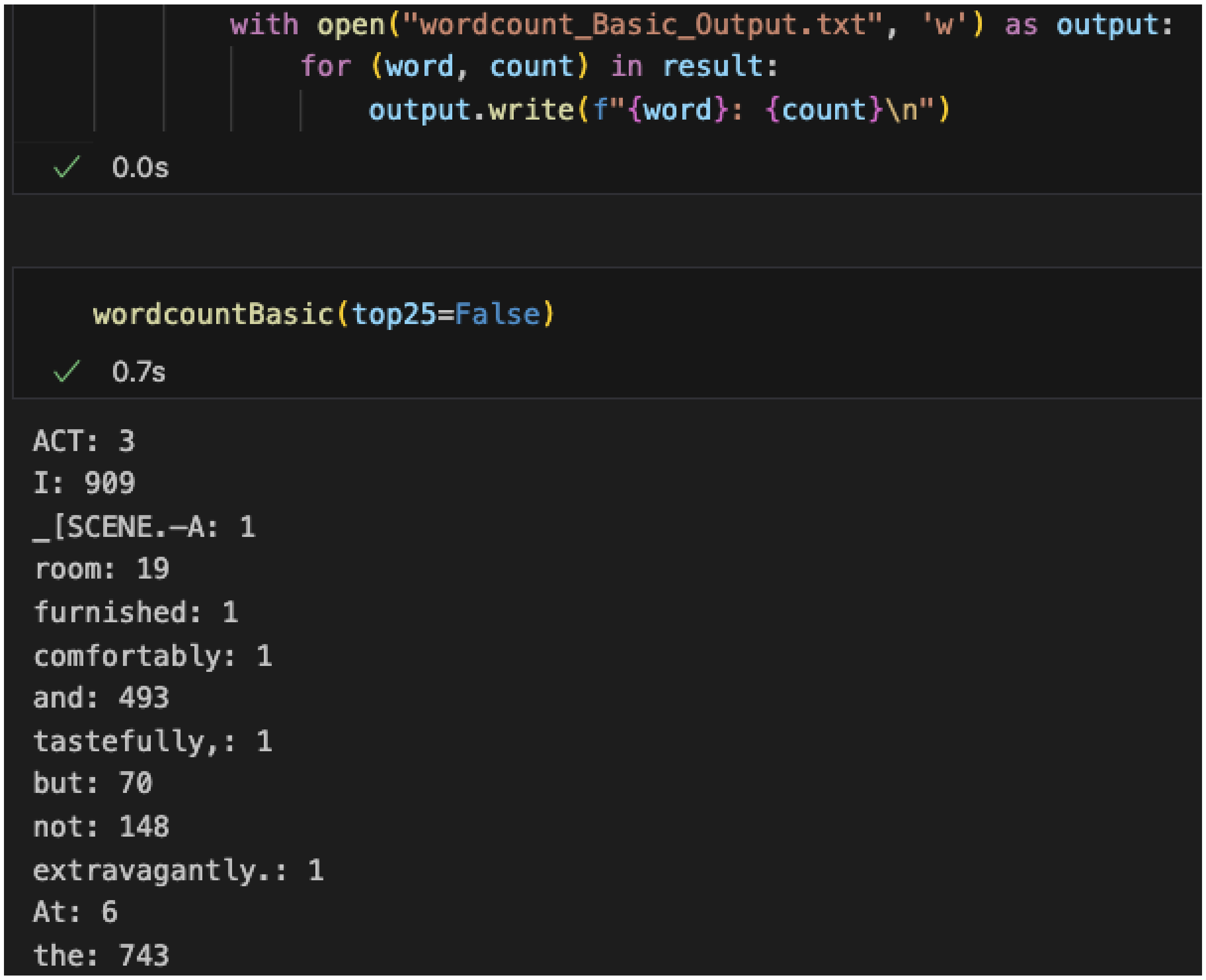
I've used two books named, **adollhouse.txt and metamorphosis.txt** from the Homework 2 for the Homework 3. Also I've used same set of stopwords(98 **stop words**) used in last homework, I've named it as **stopwords.txt**.

List of Stopwords in stopwords.txt file

A,an,the,in,my,has,as,if,do,have,had,on,at,of,for,by,with,to,up,down,and,or,not,but,is,am,are,was ,were,be,being,been,it,this,that,these,those,i,me,myself,we,us,our,ours,you,your,yours,he,him,hi s,she,her,hers,it's,its,they,them,their,theirs,what,which,who,whom,whose,here,there,when,where ,why,how,all,any,both,each,few,more,most,other,some,such,no,nor,not,only,own,same,so,than,to o,very,s,t,can,will,just,don,should,now

Basic Word Count

The output of the program is being saved as “**wordcount_Basic_Output.txt**” in txt file.



```
            with open("wordcount_Basic_Output.txt", 'w') as output:
                for (word, count) in result:
                    output.write(f"{word}: {count}\n")
```

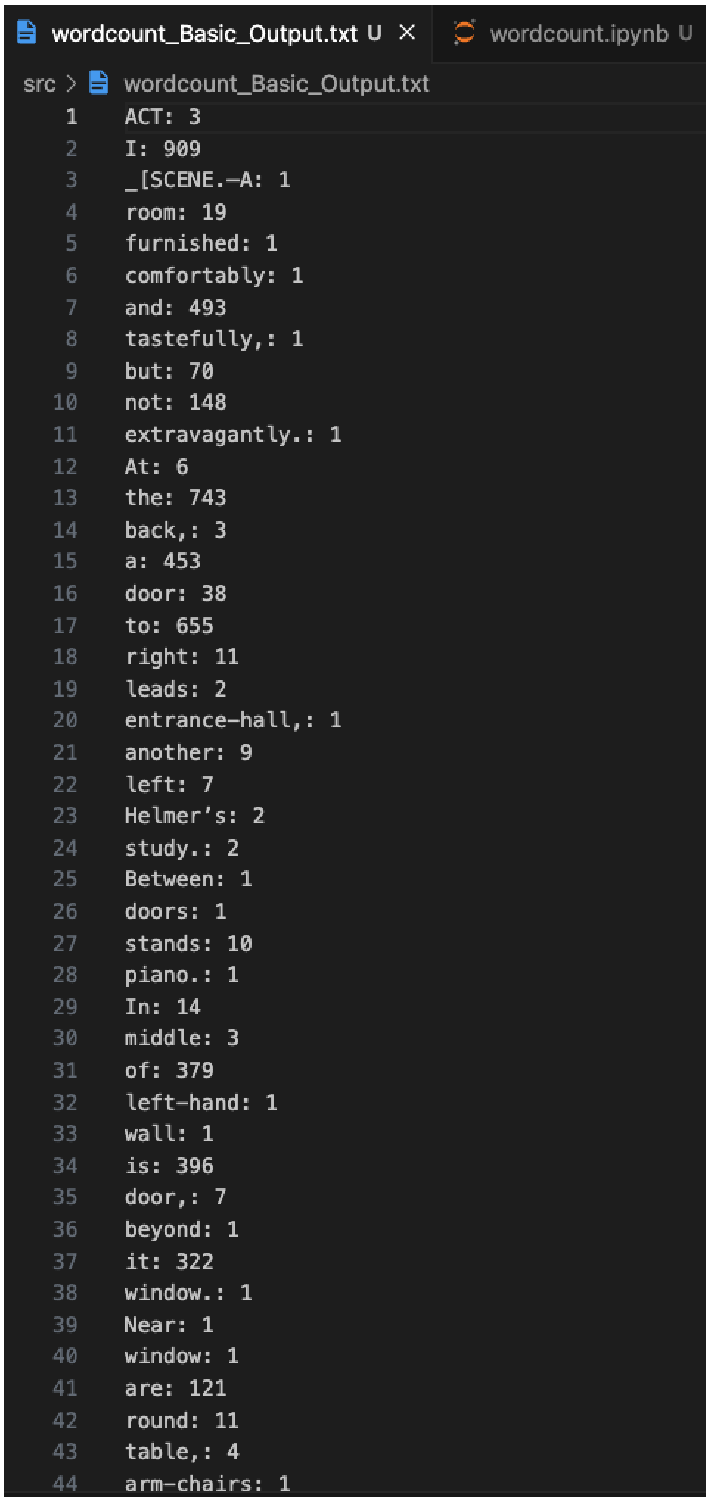
✓ 0.0s

```
wordcountBasic(top25=False)
```

✓ 0.7s

```
ACT: 3
I: 909
_[SCENE.—A: 1
room: 19
furnished: 1
comfortably: 1
and: 493
tastefully,: 1
but: 70
not: 148
extravagantly.: 1
At: 6
the: 743
```

Screenshot of **wordcount_Basic_Output.txt** file

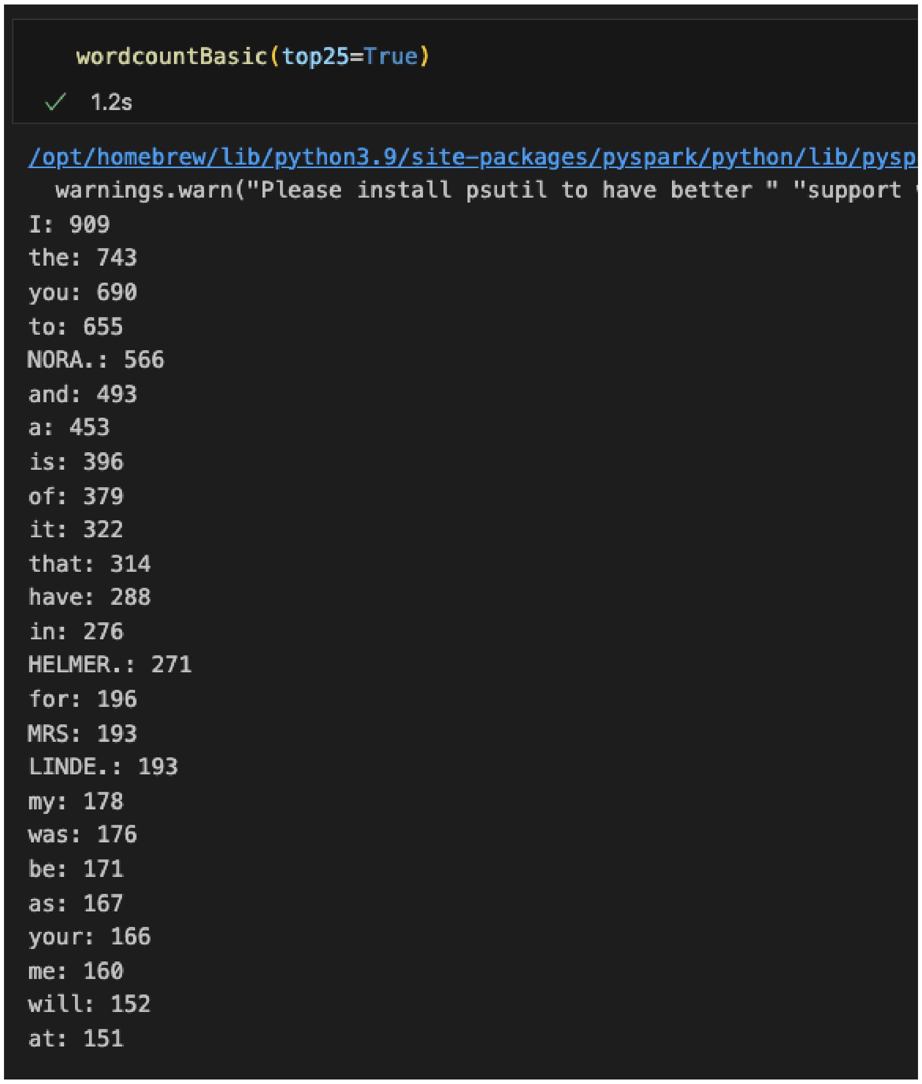

wordcount_Basic_Output.txt U
wordcount.ipynb U
src > wordcount_Basic_Output.txt

```
ACT: 3
I: 909
_[SCENE.—A: 1
room: 19
furnished: 1
comfortably: 1
and: 493
tastefully,: 1
but: 70
not: 148
extravagantly.: 1
At: 6
the: 743
back,: 3
a: 453
door: 38
to: 655
right: 11
leads: 2
entrance-hall,: 1
another: 9
left: 7
Helmer's: 2
study.: 2
Between: 1
doors: 1
stands: 10
piano.: 1
In: 14
middle: 3
of: 379
left-hand: 1
wall: 1
is: 396
door,: 7
beyond: 1
it: 322
window.: 1
Near: 1
window: 1
are: 121
round: 11
table,: 4
arm-chairs: 1
```

Analysis

1. In the PySpark REPL/ Jupyter notebook , run your basic word count program on a single text file.
    a. What are the 25 most common words? Include a screenshot of program output to back-up your claim.

        → Below is the attached screenshot of 25 most common words from the book file named “**adollhouse.txt**”.



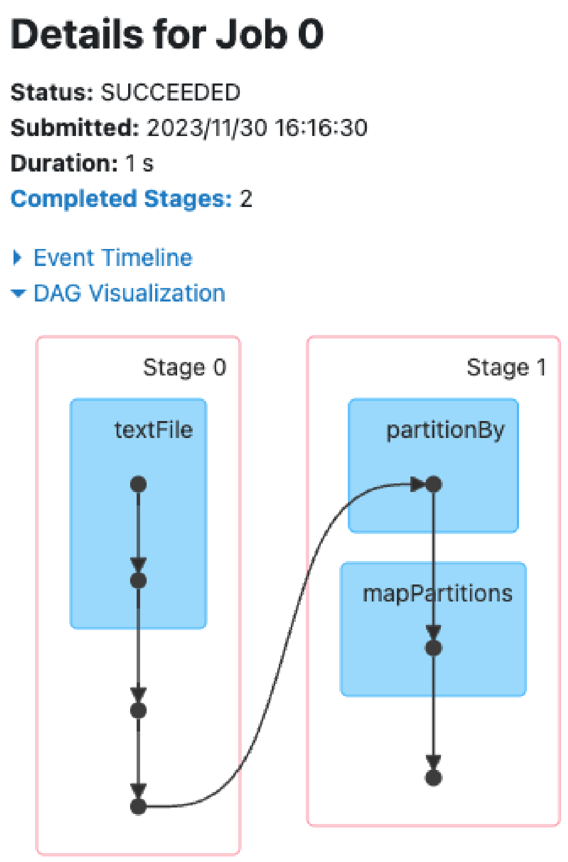
```
wordcountBasic(top25=True)
✓ 1.2s

/opt/homebrew/lib/python3.9/site-packages/pyspark/python/lib/pysp
  warnings.warn("Please install psutil to have better " "support
I: 909
the: 743
you: 690
to: 655
NORA.: 566
and: 493
a: 453
is: 396
of: 379
it: 322
that: 314
have: 288
in: 276
HELMER.: 271
for: 196
MRS: 193
LINDE.: 193
my: 178
was: 176
be: 171
as: 167
your: 166
me: 160
will: 152
at: 151
```

b. How many stages is execution broken up into? Explain why. Include a screenshot of the DAG visualization from Spark's WebUI to back-up your claim.

→ The execution is broken up into **two** stages. This execution of stages is basically for the spark to optimize the overall execution. Breaking the job into multiple stages allows proper pipelining of tasks, reduction in overall movement and processing of data, proper data segregation and data shuffling, which all together improves the performance of the execution.

The required supporting screenshots is added below:

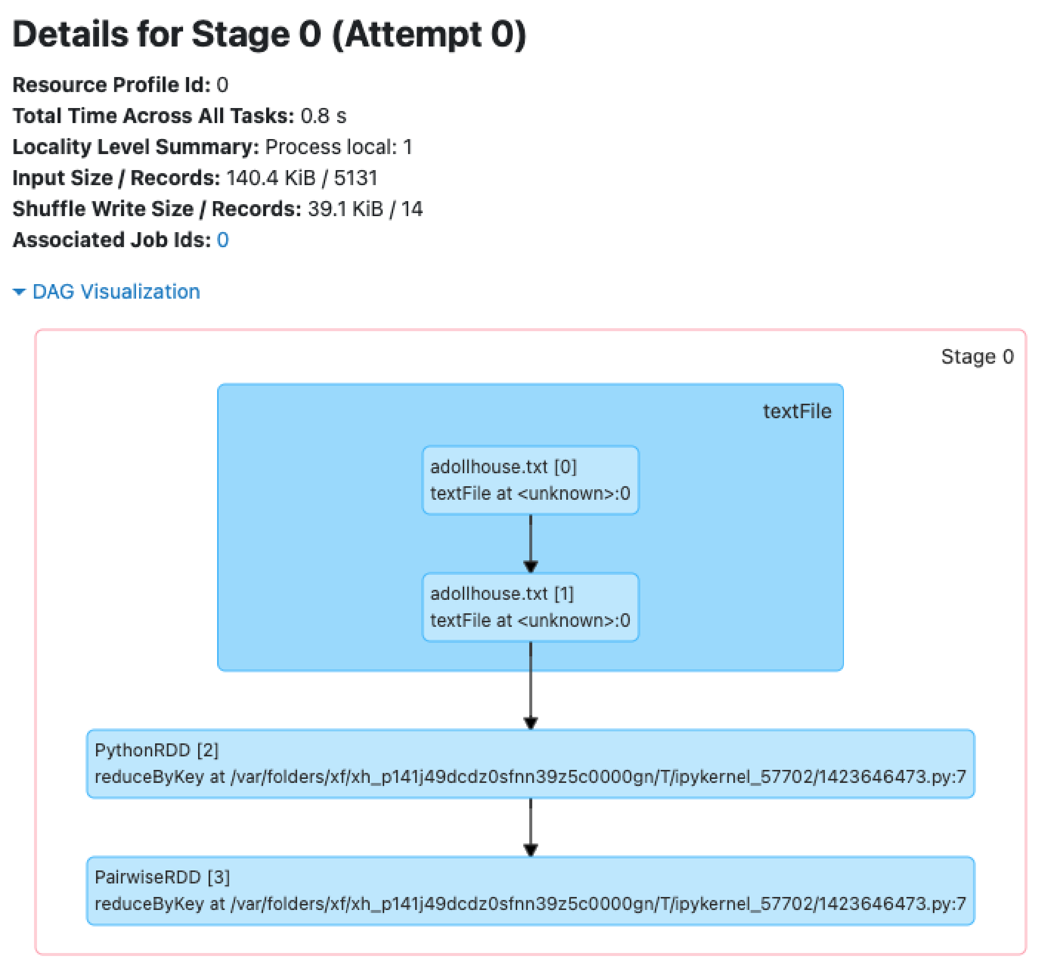
DAG visualization for the job



Details for Job 0

Status: SUCCEEDED
Submitted: 2023/11/30 16:16:30
Duration: 1 s
Completed Stages: 2

DAG visualization for the Stage 0



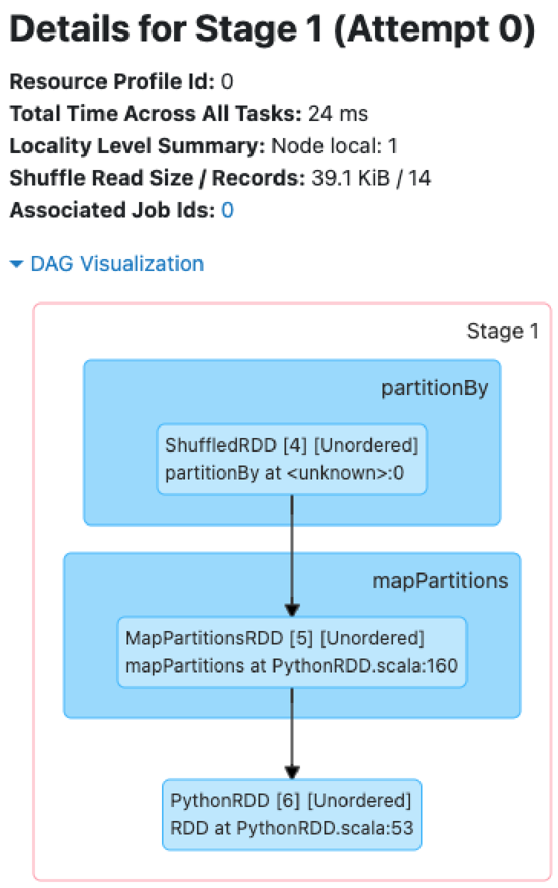
Details for Stage 0 (Attempt 0)

Resource Profile Id: 0
Total Time Across All Tasks: 0.8 s
Locality Level Summary: Process local: 1
Input Size / Records: 140.4 KiB / 5131
Shuffle Write Size / Records: 39.1 KiB / 14
Associated Job Ids: 0

▾ DAG Visualization

Stage 0
textFile
adollhouse.txt [0]
textFile at <unknown>:0
adollhouse.txt [1]
textFile at <unknown>:0
PythonRDD [2]
reduceByKey at /var/folders/xf/xh_p141j49dcdz0sfnn39z5c0000gn/T/ipykernel_57702/1423646473.py:7
PairwiseRDD [3]
reduceByKey at /var/folders/xf/xh_p141j49dcdz0sfnn39z5c0000gn/T/ipykernel_57702/1423646473.py:7

DAG visualization for the Stage 1



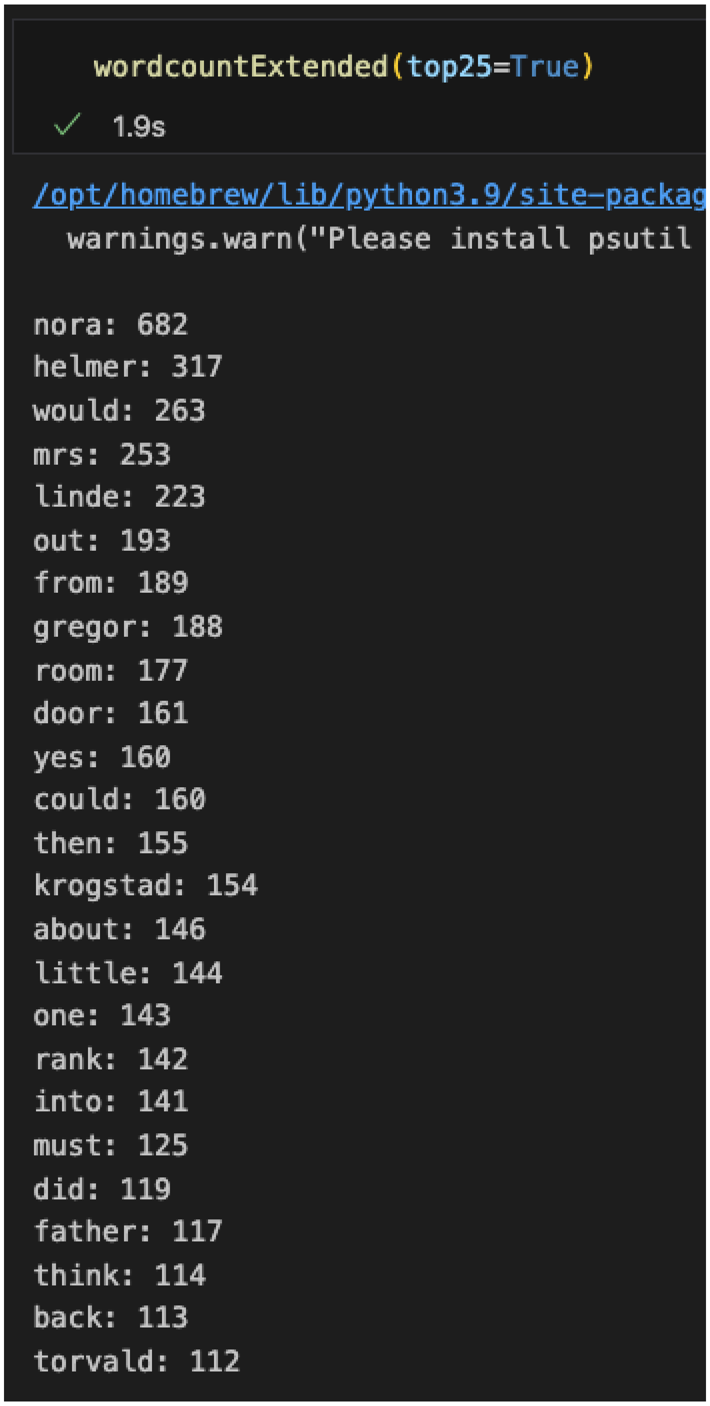
Details for Stage 1 (Attempt 0)

Resource Profile Id: 0
Total Time Across All Tasks: 24 ms
Locality Level Summary: Node local: 1
Shuffle Read Size / Records: 39.1 KiB / 14
Associated Job Ids: 0

DAG Visualization

Stage 1
partitionBy
ShuffledRDD [4] [Unordered]
partitionBy at <unknown>:0
mapPartitions
MapPartitionsRDD [5] [Unordered]
mapPartitions at PythonRDD.scala:160
PythonRDD [6] [Unordered]
RDD at PythonRDD.scala:53

2. In the PySpark REPL/Jupyter notebook, run your word count extended program on all 2 text files.
    a. What are the 25 most common words? Include a screenshot of program output to back-up your claim.
    → Below is the attached screenshot of 25 most common words from the book files named “**adollhouse.txt**” and “**metamorphosis.txt**”.



```
wordcountExtended(top25=True)
✓ 1.9s

/opt/homebrew/lib/python3.9/site-packag
  warnings.warn("Please install psutil

nora: 682
helmer: 317
would: 263
mrs: 253
linde: 223
out: 193
from: 189
gregor: 188
room: 177
door: 161
yes: 160
could: 160
then: 155
krogstad: 154
about: 146
little: 144
one: 143
rank: 142
into: 141
must: 125
did: 119
father: 117
think: 114
back: 113
torvald: 112
```

b. How many stages is execution broken up into? Explain why. Include a screenshot of the DAG visualization from Spark's WebUI to back-up your claim.

→ The execution is broken up into **six** stages however stage 2 and 4 are **skipped**. This execution of stages is basically for the spark to optimize the overall execution. Breaking the job into multiple stages allows proper pipelining of tasks, reduction in overall movement and processing of data, proper data segregation and data shuffling, which all together improves the performance of the execution. In our **wordcountExtended** program some stages are **skipped** as at times while breaking into multiple stages for execution, spark determines what all stages are already available and doesn't need further segregation or reshuffling, therefore in that case the spark simply skips the stages for the better overall performance and optimization.

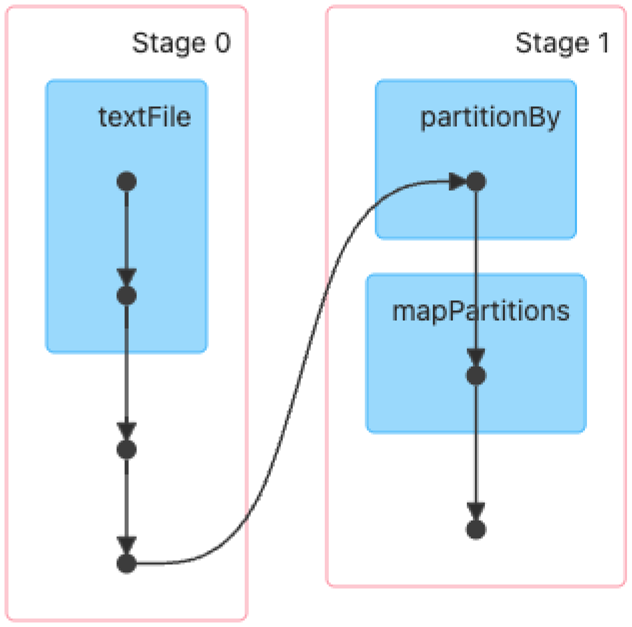
The required supporting screenshots is added below:

DAG visualization for the job

Details for Job 0

Status: SUCCEEDED
Submitted: 2023/11/30 17:49:10
Duration: 1 s
Completed Stages: 2

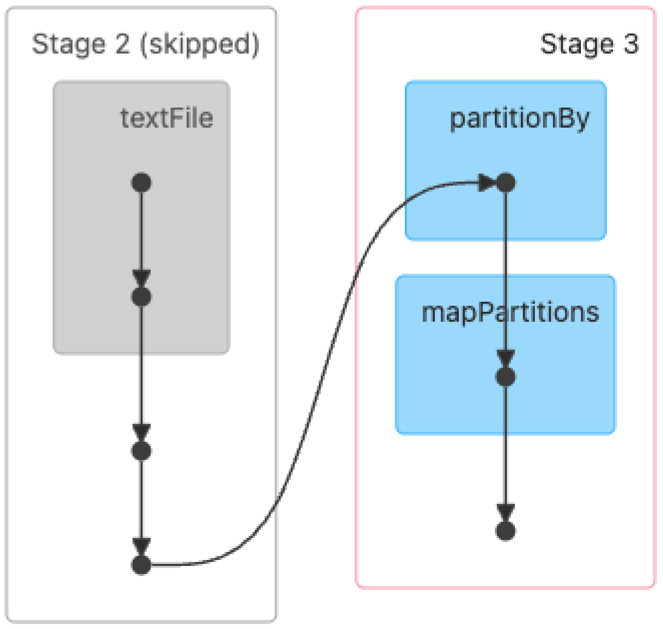
▸ Event Timeline
▾ DAG Visualization

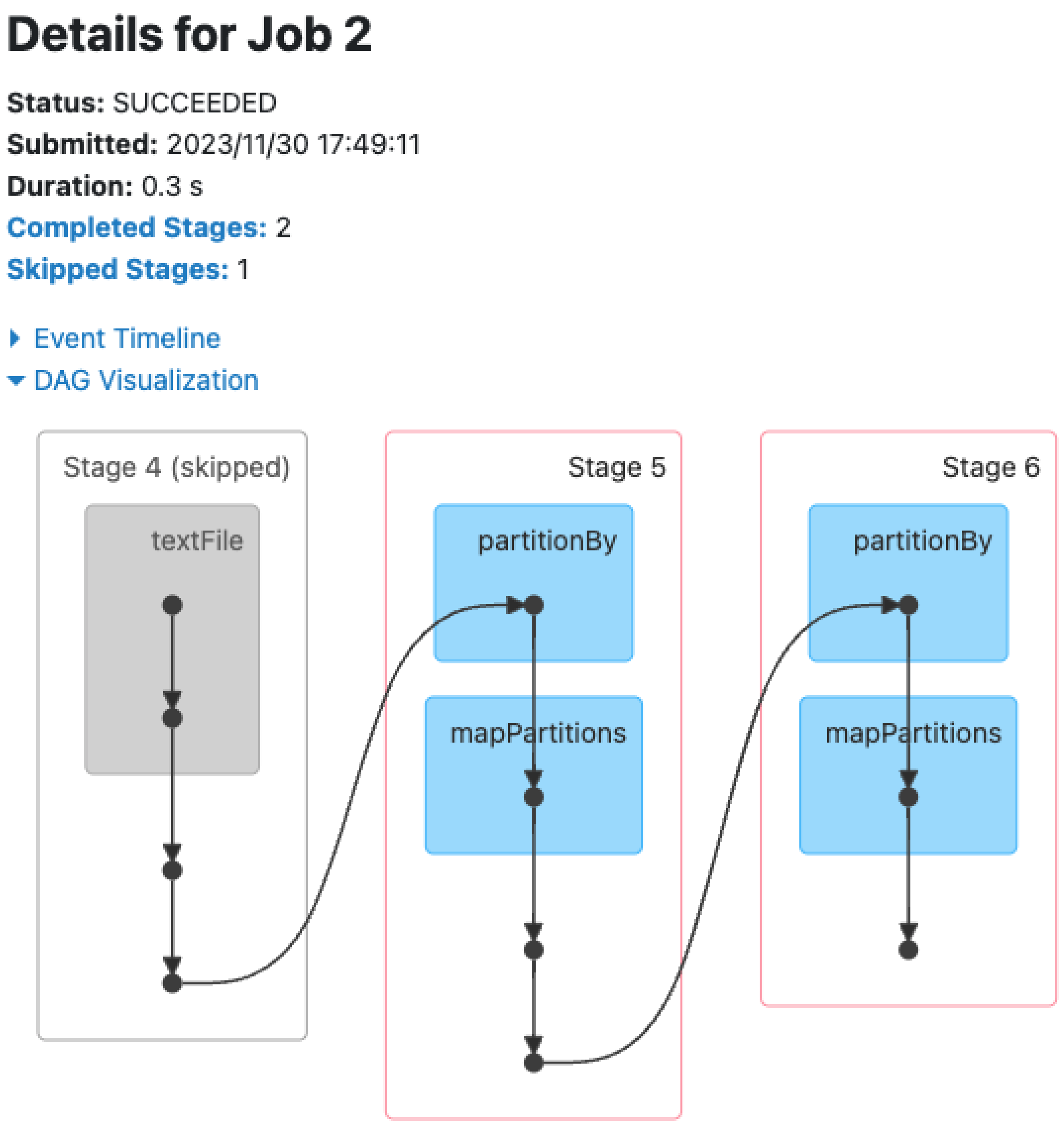


Details for Job 1

Status: SUCCEEDED
Submitted: 2023/11/30 17:49:11
Duration: 0.2 s
Completed Stages: 1
Skipped Stages: 1

▸ Event Timeline
▾ DAG Visualization





Details for Job 2

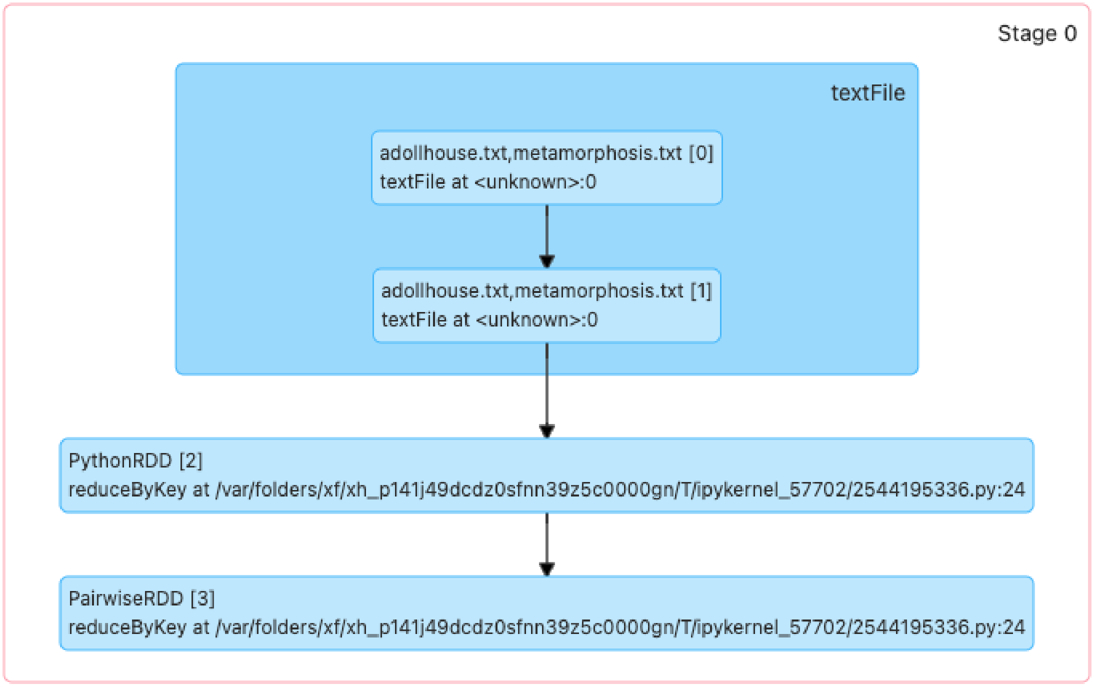
Status: SUCCEEDED
Submitted: 2023/11/30 17:49:11
Duration: 0.3 s
Completed Stages: 2
Skipped Stages: 1

DAG visualization for the stages

Details for Stage 0 (Attempt 0)

Resource Profile Id: 0
Total Time Across All Tasks: 0.9 s
Locality Level Summary: Process local: 2
Input Size / Records: 257.3 KiB / 6991
Shuffle Write Size / Records: 50.0 KiB / 44
Associated Job Ids: 0

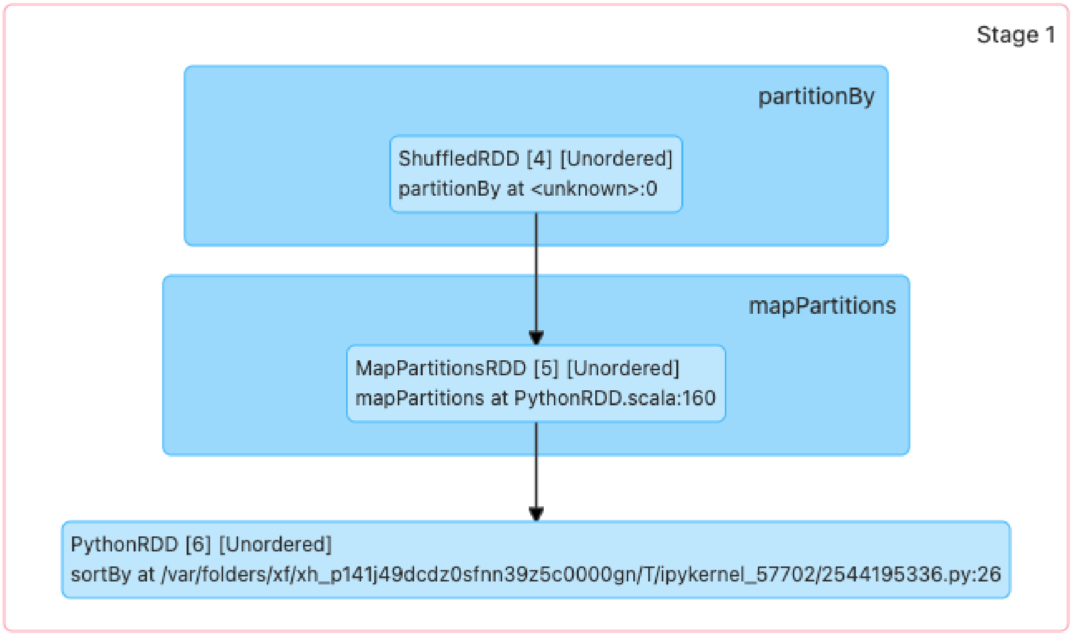
▾ DAG Visualization


Stage 0
textFile
adollhouse.txt,metamorphosis.txt [0]
textFile at <unknown>:0
adollhouse.txt,metamorphosis.txt [1]
textFile at <unknown>:0
PythonRDD [2]
reduceByKey at /var/folders/xf/xh_p141j49dcdz0sfnn39z5c0000gn/T/ipykernel_57702/2544195336.py:24
PairwiseRDD [3]
reduceByKey at /var/folders/xf/xh_p141j49dcdz0sfnn39z5c0000gn/T/ipykernel_57702/2544195336.py:24

Details for Stage 1 (Attempt 0)

Resource Profile Id: 0
Total Time Across All Tasks: 44 ms
Locality Level Summary: Node local: 2
Shuffle Read Size / Records: 50.0 KiB / 44
Associated Job Ids: 0

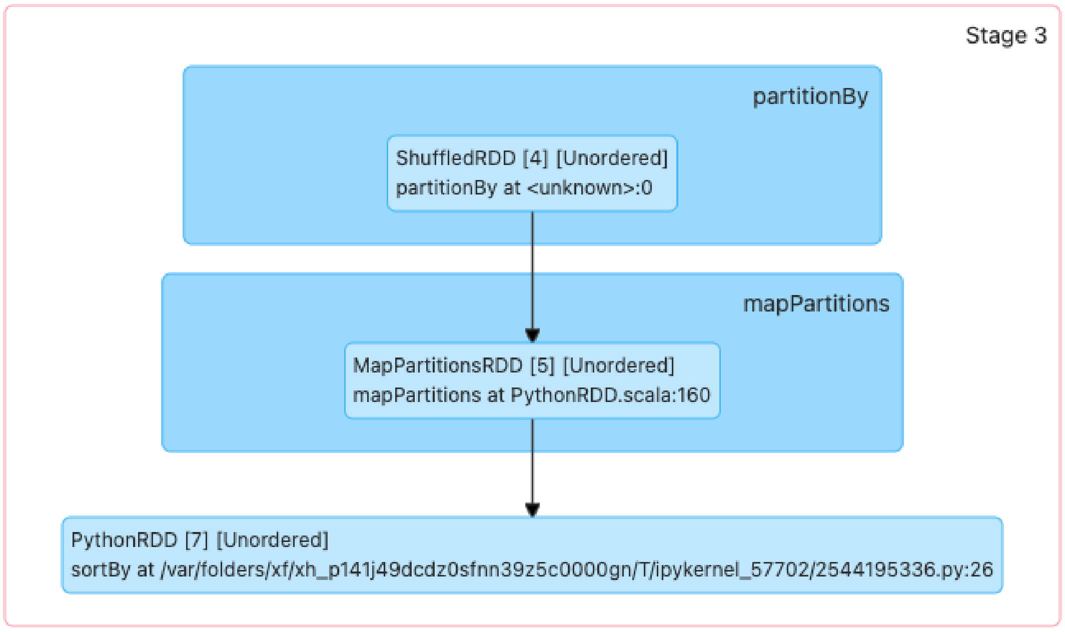
▾ DAG Visualization


Stage 1
partitionBy
ShuffledRDD [4] [Unordered]
partitionBy at <unknown>:0
mapPartitions
MapPartitionsRDD [5] [Unordered]
mapPartitions at PythonRDD.scala:160
PythonRDD [6] [Unordered]
sortBy at /var/folders/xf/xh_p141j49dcdz0sfnn39z5c0000gn/T/ipykernel_57702/2544195336.py:26

Details for Stage 3 (Attempt 0)

Resource Profile Id: 0
Total Time Across All Tasks: 82 ms
Locality Level Summary: Node local: 2
Shuffle Read Size / Records: 50.0 KiB / 44
Associated Job Ids: 1

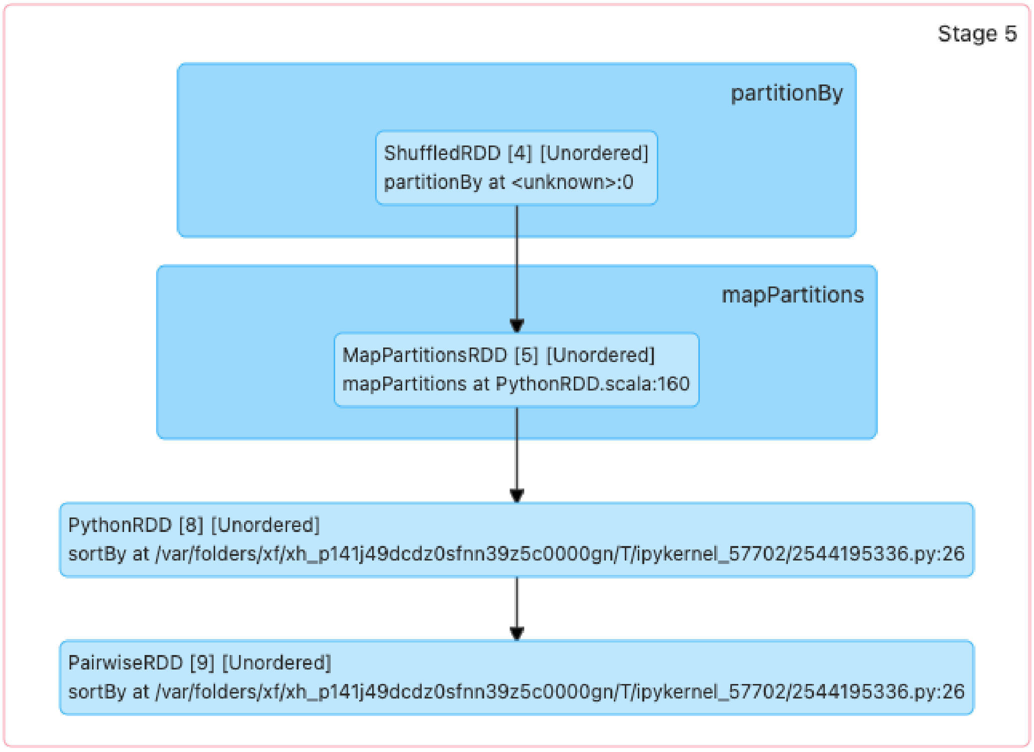
▾ DAG Visualization


Stage 3
partitionBy
ShuffledRDD [4] [Unordered]
partitionBy at <unknown>:0
mapPartitions
MapPartitionsRDD [5] [Unordered]
mapPartitions at PythonRDD.scala:160
PythonRDD [7] [Unordered]
sortBy at /var/folders/xf/xh_p141j49dcdz0sfnn39z5c0000gn/T/ipykernel_57702/2544195336.py:26

Details for Stage 5 (Attempt 0)

Resource Profile Id: 0
Total Time Across All Tasks: 0.1 s
Locality Level Summary: Node local: 2
Shuffle Read Size / Records: 50.0 KiB / 44
Shuffle Write Size / Records: 44.3 KiB / 40
Associated Job Ids: 2

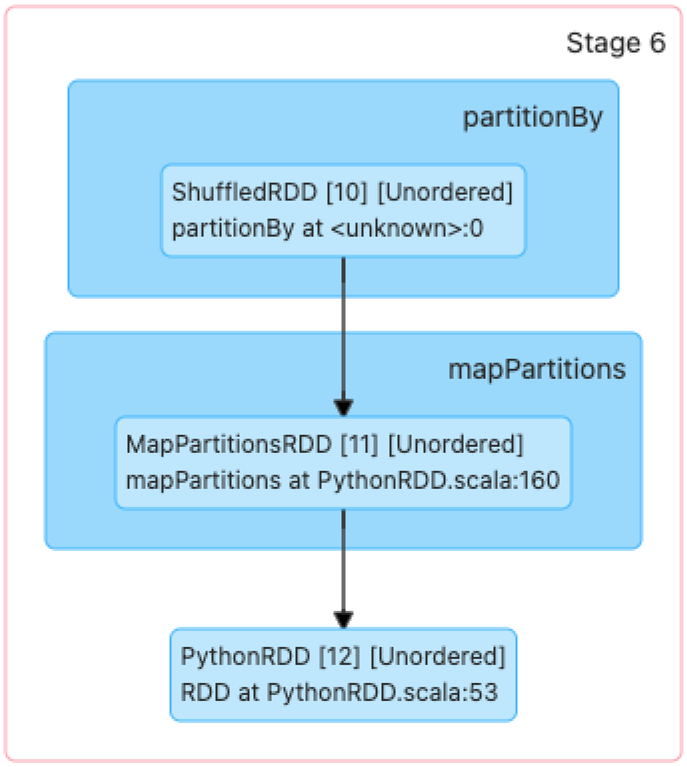
▾ DAG Visualization


Stage 5
partitionBy
ShuffledRDD [4] [Unordered]
partitionBy at <unknown>:0
mapPartitions
MapPartitionsRDD [5] [Unordered]
mapPartitions at PythonRDD.scala:160
PythonRDD [8] [Unordered]
sortBy at /var/folders/xf/xh_p141j49dcdz0sfnn39z5c0000gn/T/ipykernel_57702/2544195336.py:26
PairwiseRDD [9] [Unordered]
sortBy at /var/folders/xf/xh_p141j49dcdz0sfnn39z5c0000gn/T/ipykernel_57702/2544195336.py:26

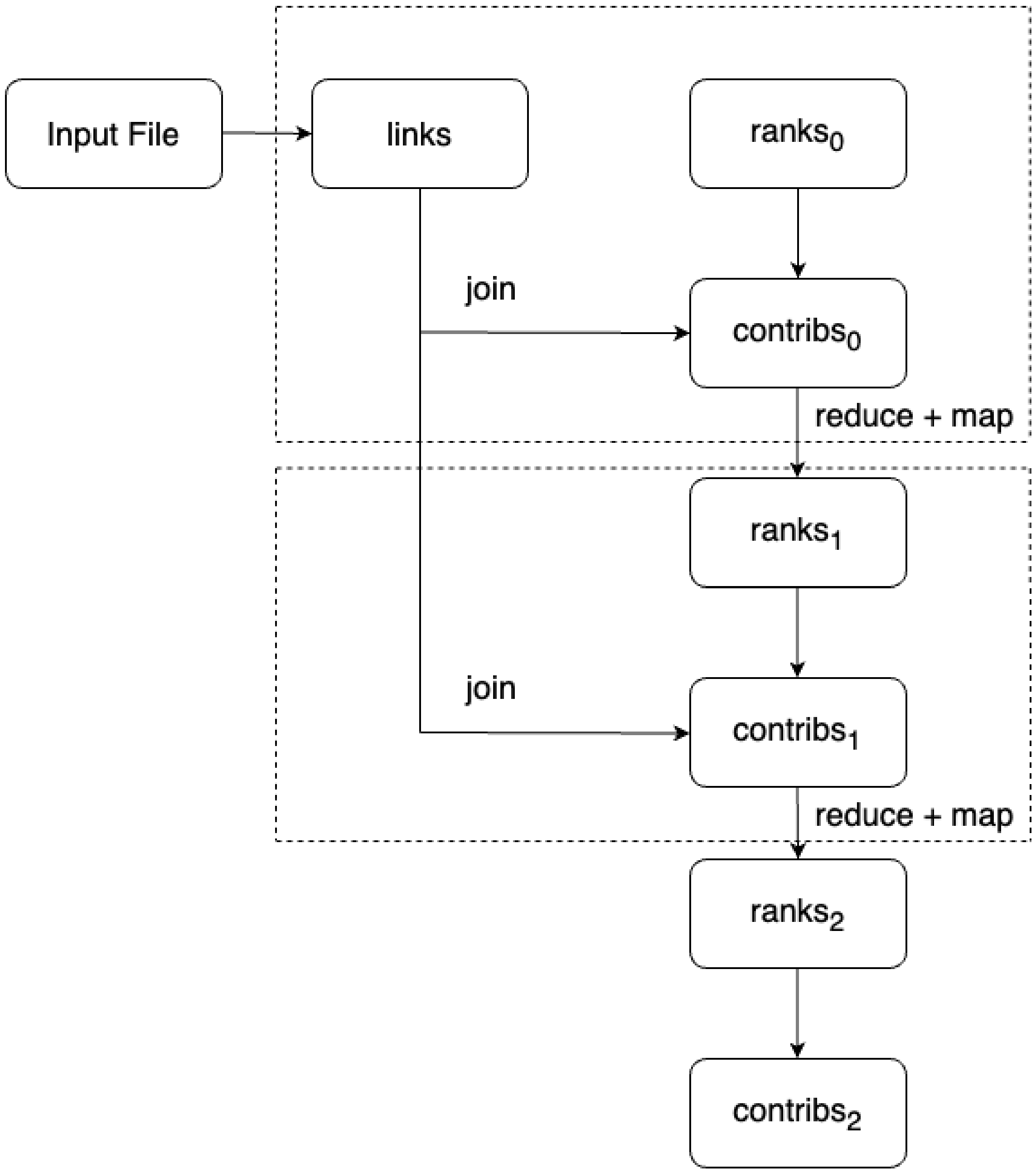
Details for Stage 6 (Attempt 0)

Resource Profile Id: 0
Total Time Across All Tasks: 24 ms
Locality Level Summary: Node local: 1
Shuffle Read Size / Records: 15.9 KiB / 20
Associated Job Ids: 2

▾ DAG Visualization


Stage 6
partitionBy
ShuffledRDD [10] [Unordered]
partitionBy at <unknown>:0
mapPartitions
MapPartitionsRDD [11] [Unordered]
mapPartitions at PythonRDD.scala:160
PythonRDD [12] [Unordered]
RDD at PythonRDD.scala:53

3. Given the above spark application, draw the lineage graph DAG for the RDD ranks on line 12 when the iteration variable i has a value of 2. Include nodes for all intermediate RDDs, even if they are unnamed.



References

1. https://spark.apache.org/docs/latest/api/python/getting_started/install.html
2. https://spark.apache.org/docs/latest/rdd-programming-guide.html/
3. https://www.geeksforgeeks.org/python-remove-punctuation-from-string/
4. https://ublearns.buffalo.edu/d2l/le/content/144365/viewContent/3651564/View - DIC CSE587 Lec 28 - RDD