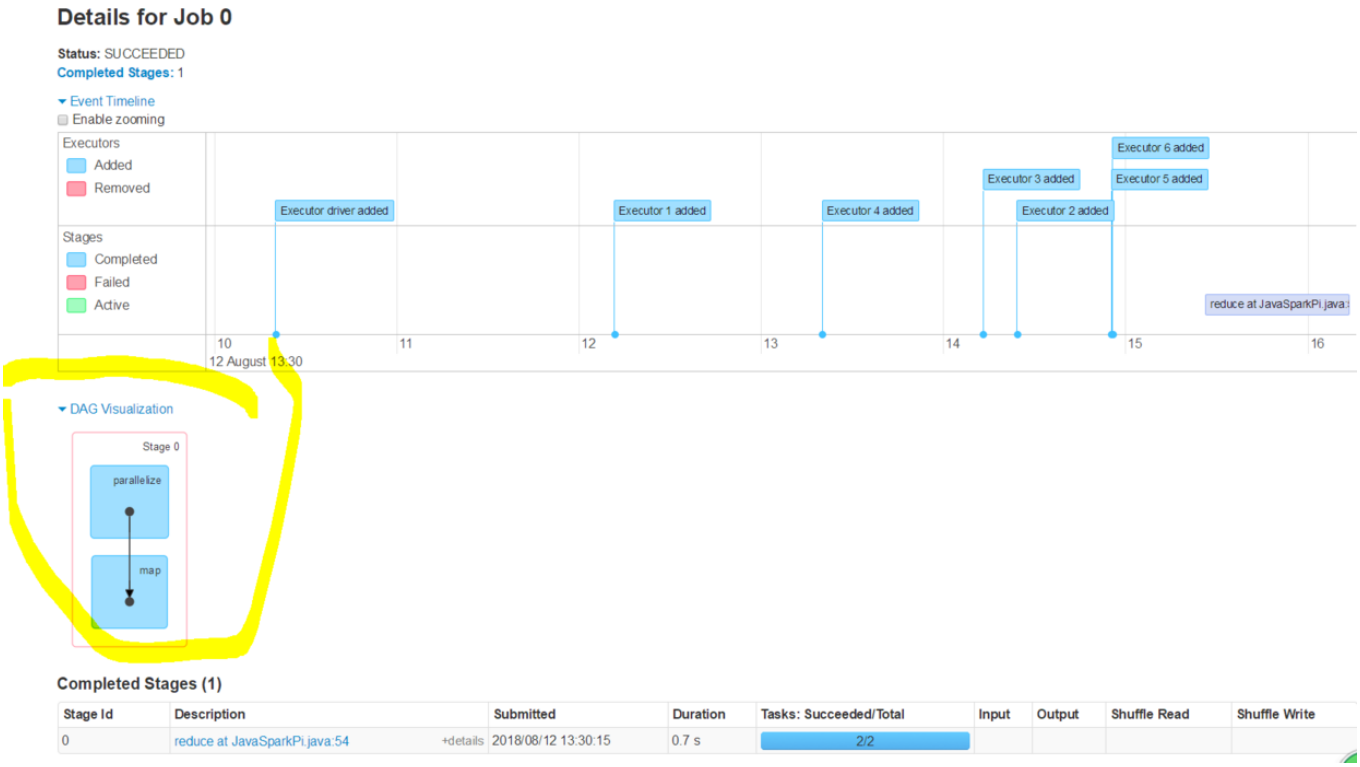
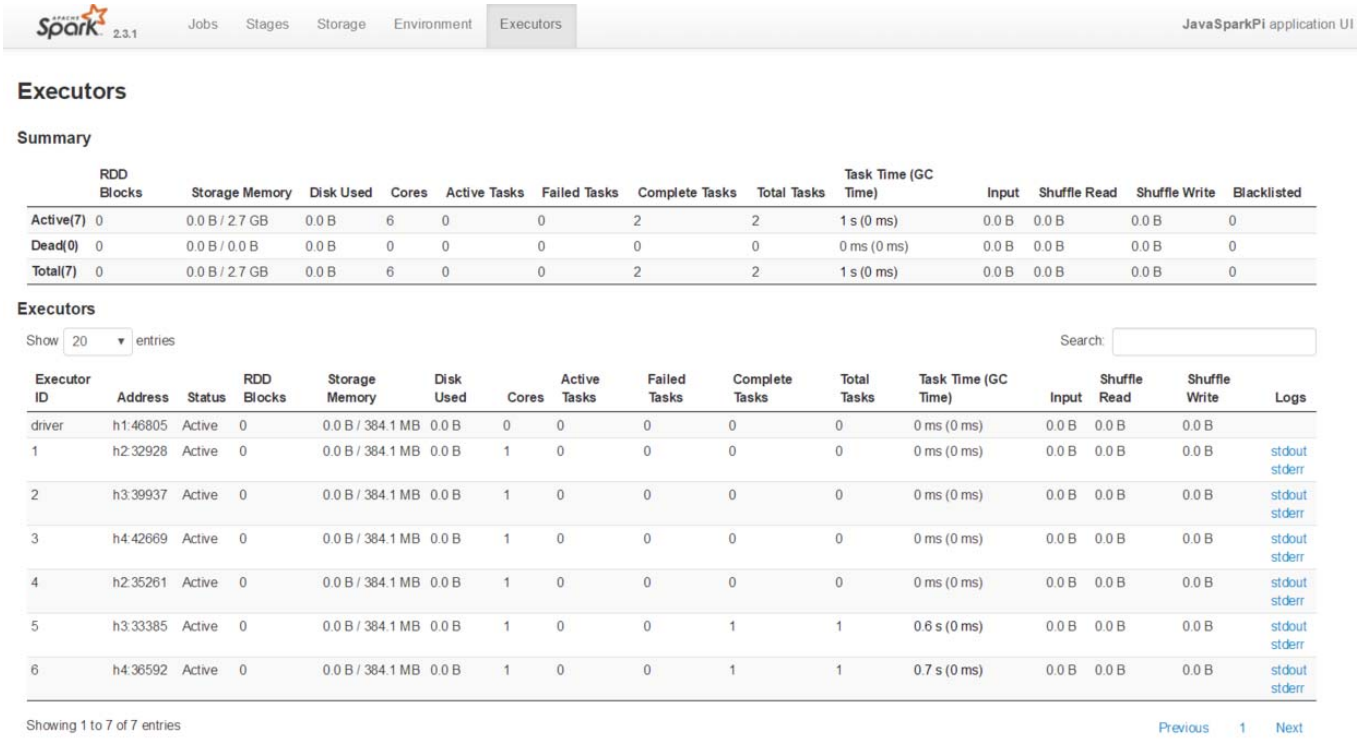
sparkhw02

1. 通过spark-submit提交org.apache.spark.examples.JavaSparkPi，并将DAG截图。要求将Spark应用提交到Yarn上（即--master=yarn）。

```
`[ian@h1 conf]$ spark-submit --master yarn \
--name javasparkpitest \
--num-executors 6 \
--class org.apache.spark.examples.JavaSparkPi
$SPARK_HOME/examples/jars/spark-examples_2.11-2.3.1.jar`
```

```
2018-08-12 13:30:16 INFO  DAGScheduler:54 - ResultStage 0 (reduce at JavaSparkPi.java:54) finished in 0.791 s
2018-08-12 13:30:16 INFO  DAGScheduler:54 - Job 0 finished: reduce at JavaSparkPi.java:54, took 0.853950 s
Pi is roughly 3.14344
2018-08-12 13:30:16 INFO  AbstractConnector:318 - Stopped Spark@bae47a0{HTTP/1.1,[http/1.1]}{0.0.0.0:4040}
2018-08-12 13:30:16 INFO  SparkUI:54 - Stopped Spark web UI at http://h1:4040
2018-08-12 13:30:16 INFO  YarnClientSchedulerBackend:54 - Interrupting monitor thread
2018-08-12 13:30:16 INFO  YarnClientSchedulerBackend:54 - Shutting down all executors
2018-08-12 13:30:16 INFO  YarnSchedulerBackend$YarnDriverEndpoint:54 - Asking each executor to shut down
2018-08-12 13:30:16 INFO  SchedulerExtensionServices:54 - Stopping SchedulerExtensionServices
(serviceOption=None,
 services=List(),
 started=false)
2018-08-12 13:30:16 INFO  YarnClientSchedulerBackend:54 - Stopped
2018-08-12 13:30:16 INFO  MapOutputTrackerMasterEndpoint:54 - MapOutputTrackerMasterEndpoint stopped!
2018-08-12 13:30:16 INFO  MemoryStore:54 - MemoryStore cleared
2018-08-12 13:30:16 INFO  BlockManager:54 - BlockManager stopped
2018-08-12 13:30:16 INFO  BlockManagerMaster:54 - BlockManagerMaster stopped
2018-08-12 13:30:16 INFO  OutputCommitCoordinator$OutputCommitCoordinatorEndpoint:54 - OutputCommitCoordinator stopp
ed!
2018-08-12 13:30:16 INFO  SparkContext:54 - Successfully stopped SparkContext
2018-08-12 13:30:16 INFO  ShutdownHookManager:54 - Shutdown hook called
2018-08-12 13:30:16 INFO  ShutdownHookManager:54 - Deleting directory /tmp/spark-df2f5c31-a52c-43d6-a4c3-413510b4455
e
2018-08-12 13:30:16 INFO  ShutdownHookManager:54 - Deleting directory /tmp/spark-2e469a5d-3026-45d6-931f-92c615bc3ed
4
```





Spark 2.3.1 Jobs Stages Storage Environment Executors JavaSparkPi application UI

Executors

Summary

	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Blacklisted
Active(7)	0	0.0 B / 2.7 GB	0.0 B	6	0	0	2	2	1 s (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	0	0	0	0 ms (0 ms)	0.0 B	0.0 B	0.0 B	0
Total(7)	0	0.0 B / 2.7 GB	0.0 B	6	0	0	2	2	1 s (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	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Logs
driver	h1:46805	Active	0	0.0 B / 384.1 MB	0.0 B	0	0	0	0	0	0 ms (0 ms)	0.0 B	0.0 B	0.0 B	
1	h2:32928	Active	0	0.0 B / 384.1 MB	0.0 B	1	0	0	0	0	0 ms (0 ms)	0.0 B	0.0 B	0.0 B	stdout stderr
2	h3:39937	Active	0	0.0 B / 384.1 MB	0.0 B	1	0	0	0	0	0 ms (0 ms)	0.0 B	0.0 B	0.0 B	stdout stderr
3	h4:42669	Active	0	0.0 B / 384.1 MB	0.0 B	1	0	0	0	0	0 ms (0 ms)	0.0 B	0.0 B	0.0 B	stdout stderr
4	h2:35261	Active	0	0.0 B / 384.1 MB	0.0 B	1	0	0	0	0	0 ms (0 ms)	0.0 B	0.0 B	0.0 B	stdout stderr
5	h3:33385	Active	0	0.0 B / 384.1 MB	0.0 B	1	0	0	1	1	0.6 s (0 ms)	0.0 B	0.0 B	0.0 B	stdout stderr
6	h4:36592	Active	0	0.0 B / 384.1 MB	0.0 B	1	0	0	1	1	0.7 s (0 ms)	0.0 B	0.0 B	0.0 B	stdout stderr

Showing 1 to 7 of 7 entries Previous 1 Next

2. 在**IDE**中使用**Java**实现**WordCount**，**Partition**数设置为**3**（也即**numSlices=3**），并过滤掉部分单词（如 **of with**）。使用单步调试，验证同一**Partition(**或者说**task)**内的不同数据，是保证每条数据从前到后完全处理完，再处理下一条数据，还是对所有数据同时进行某种操作，结合后再进行某种操作。

> 老师，我使用的是python.
> 根据课上所讲，由于map,flatMap,filter都是Transformation类型的算子
> 应该是：1）对A顺序执行map,flatMap,filter，然后对B顺序执行map,flatMap,filter，最后对C顺序执行map,flatMap,filter
>
> python代码如下

```
from __future__ import print_function

import sys
from operator import add

from pyspark.sql import SparkSession


lines = spark.read.text('hdfs://h1:9000/test.py')

type(lines)


#将df对象转成rdd对象，每一行为一个rdd元素
rdd1 = lines.rdd

rdd1.getNumPartitions()
```

```
rdd2 = rdd1.map(lambda r: r[0])


rdd3 = rdd2.flatMap(lambda x: x.split(' '))


#过滤掉一些常见词汇
rdd4 = rdd3.filter(lambda x: x not in ['=','of','a','#','with'])


rdd5 = rdd4.map(lambda x: (x, 1))


rdd6 = rdd5.reduceByKey(add)


counts = lines.flatMap(lambda x: x.split(' '))                  .map(lambda x: (x, 1))


output = rdd6.collect()

for (word, count) in output:
        print("%s: %i" % (word, count))
```