Spark SQL motivation

192.168.0.38	WARNING	Something bad could happen
192.168.0.88	INFO	Just an info message passing by
192.168.0.5	WARNING	Something bad could happen
192.168.0.36	ERROR	When production fails in despair, whom are you gonna call?
192.168.0.27	INFO	Just an info message passing by

192.168.0.38	USA
192.168.0.88	RUSSIA
192.168.0.5	CHINA
192.168.0.36	USA
192.168.0.27	RUSSIA

Declarative

```sql
SELECT country, code FROM table1
JOIN table2
WHERE table1.ip = table2.ip
AND table1.code != "INFO"
```

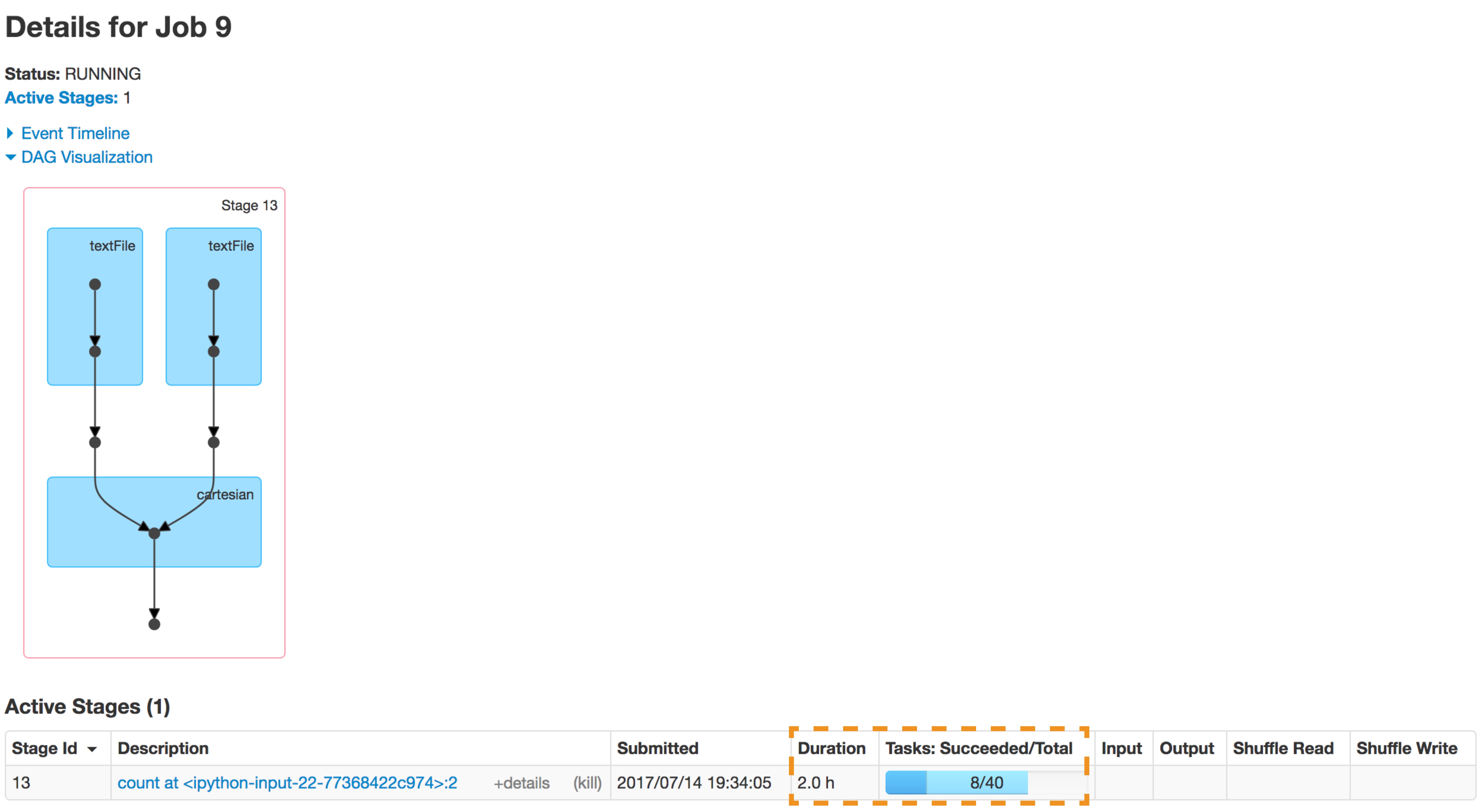
Imperative

```
rdd1 = sc.textFile("log.txt")
rdd2 = sc.textFile("ips.txt")
table1 = rdd1.map(lambda x: x.split("\t"))
table2 = rdd2.map(lambda x: x.split("\t"))
table1.cartesian(table2)
    .filter(lambda (x, y): x[0] == y[0])
    .filter(lambda (x, y): x[1] != "INFO")
    .map(lambda (x, y): (y[1], x[1]))
    .count()
```

Performance comparison

- 10,000,000 rows of logs (496 MB, 20 partitions)
- 100,000 rows of IPs (2 MB, 2 partitions)
- 10 executors, 2 cores & 4GB per executor

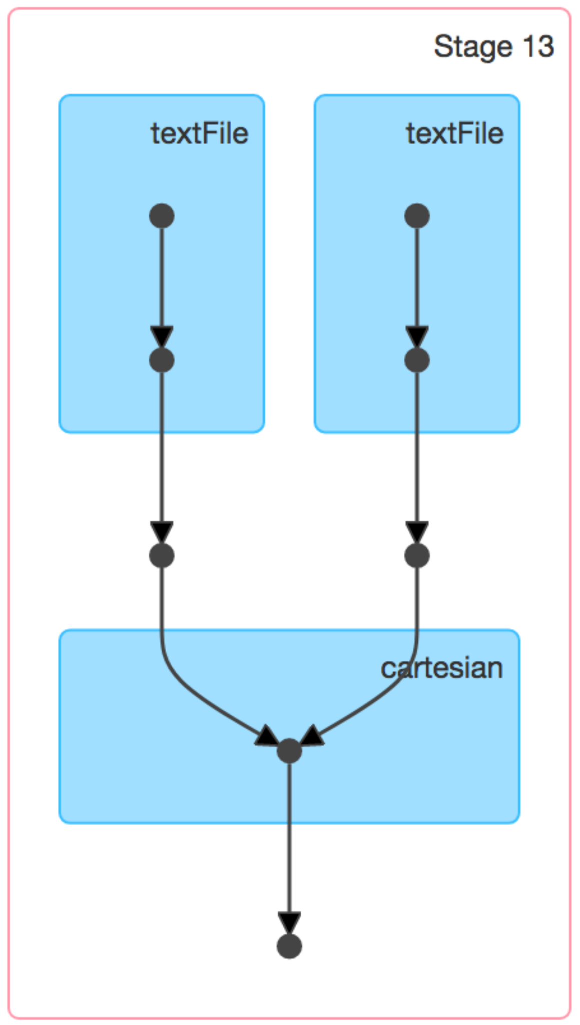
```
rdd1 = sc.textFile("log.txt")
rdd2 = sc.textFile("ips.txt")
table1 = rdd1.map(lambda x: x.split("\t"))
table2 = rdd2.map(lambda x: x.split("\t"))
table1.cartesian(table2)
    .filter(lambda (x, y): x[0] == y[0])
    .filter(lambda (x, y): x[1] != "INFO")
    .map(lambda (x, y): (y[1], x[1]))
    .count()
```



Details for Job 9

Status: RUNNING
Active Stages: 1

▸ Event Timeline
▾ DAG Visualization



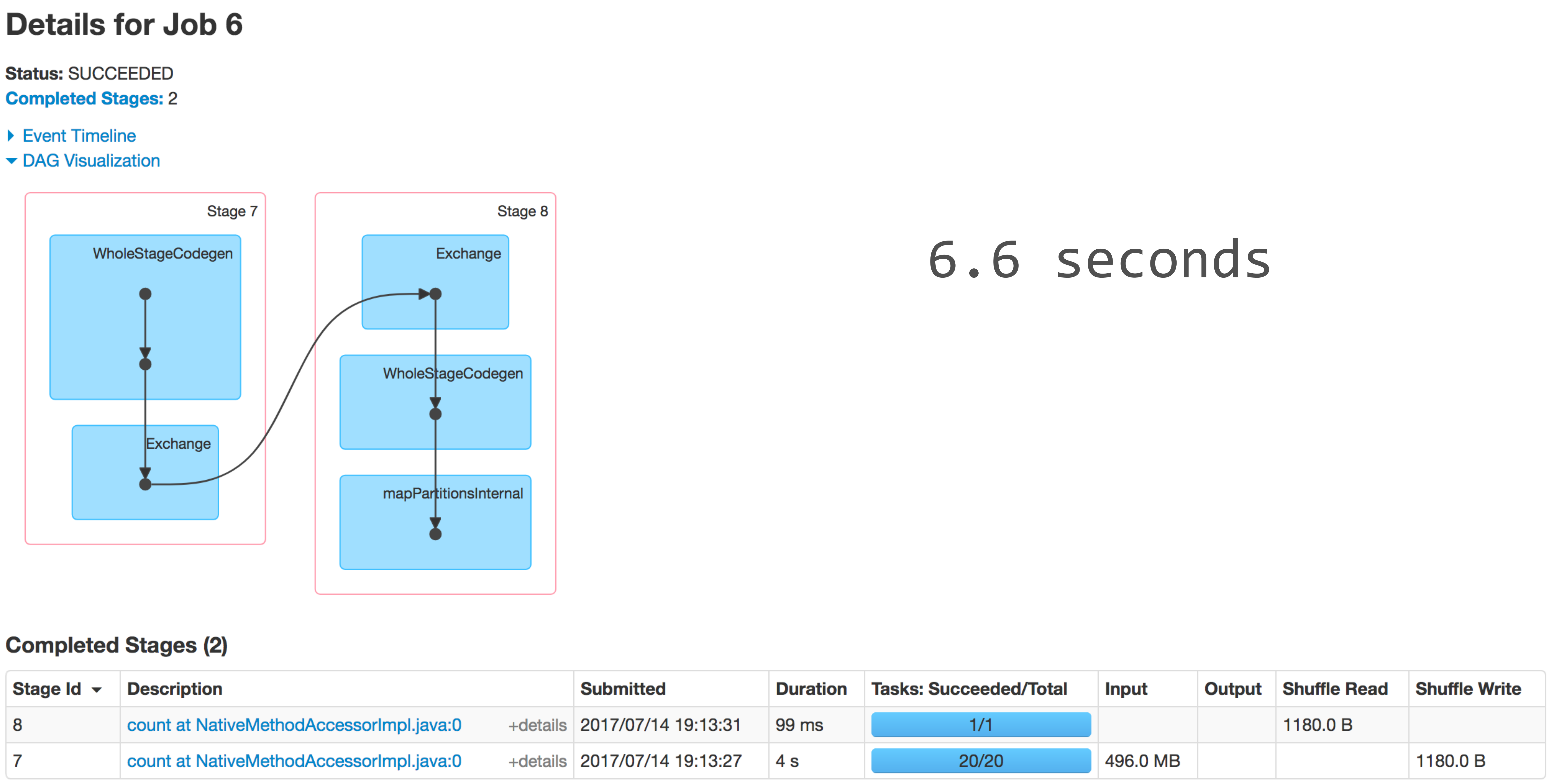
Active Stages (1)

Stage Id ▾	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
13	count at <ipython-input-22-77368422c974>:2 +details (kill)	2017/07/14 19:34:05	2.0 h	8/40				

```
table1.createOrReplaceTempView("table1")
table2.createOrReplaceTempView("table2")

query = """
    SELECT country, code FROM table1
    JOIN table2
    WHERE table1.ip = table2.ip
    AND table1.code != INFO
    """

spark.sql(query).count()
```



Details for Job 6

Status: SUCCEEDED
Completed Stages: 2

▸ Event Timeline
▾ DAG Visualization

Completed Stages (2)

Stage Id ▾	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
8	count at NativeMethodAccessorImpl.java:0 +details	2017/07/14 19:13:31	99 ms	1/1			1180.0 B	
7	count at NativeMethodAccessorImpl.java:0 +details	2017/07/14 19:13:27	4 s	20/20	496.0 MB			1180.0 B

6.6 seconds

RDD is a low level API

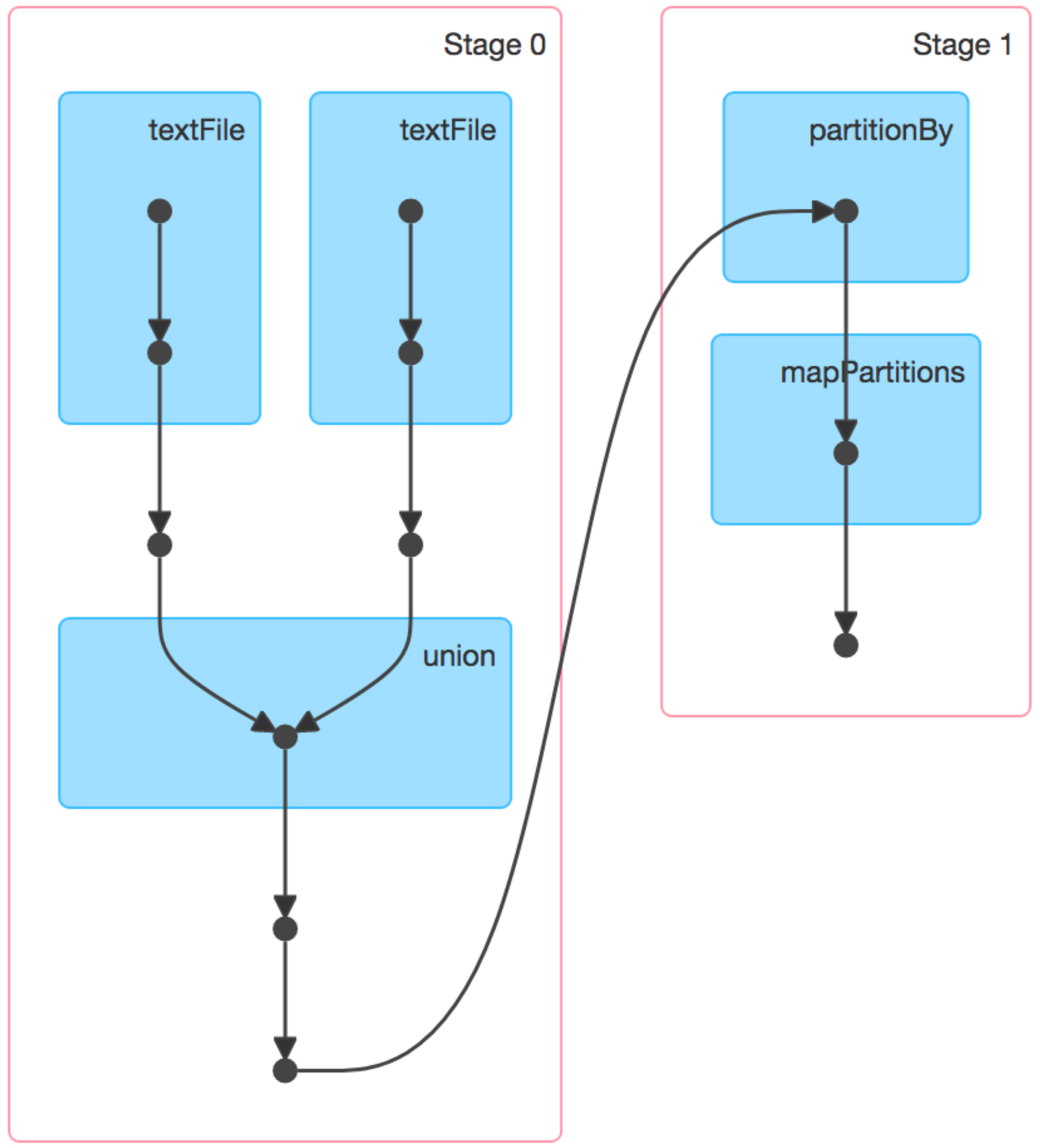
```
rdd1 = sc.textFile("log.txt")
rdd2 = sc.textFile("ips.txt")
table1 = rdd1.map(lambda x: x.split("\t"))
table2 = rdd2.map(lambda x: x.split("\t"))
table1.cartesian(table2)
    .filter(lambda (x, y): x[0] == y[0])
    .filter(lambda (x, y): x[1] != "INFO")
    .map(lambda (x, y): (y[1], x[1]))
    .count()
```

```
rdd1 = sc.textFile("log.txt")
rdd2 = sc.textFile("ips.txt")
table1 = rdd1.map(lambda x: x.split("\t"))
table2 = rdd2.map(lambda x: x.split("\t"))
table1.join(table2)
    .map(lambda (x, y): (y[1], y[0]))
    .filter(lambda (x, y): y != "INFO")
    .count()
```

Details for Job 0

Status: SUCCEEDED
Completed Stages: 2

▸ Event Timeline
▾ DAG Visualization



12.8 seconds

Completed Stages (2)

Stage Id ▾	Description		Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
1	count at <magic-timeit>:257	+details	2017/07/14 19:13:02	2 s	22/22			55.3 MB	
0	join at <magic-timeit>:257	+details	2017/07/14 19:12:52	11 s	22/22				55.3 MB

Spark knows nothing about your data

```sql
SELECT country, code FROM table1
JOIN table2
WHERE table1.ip = table2.ip
AND table1.code != "INFO"
```

```
table1 = sc.textFile("log.txt")
table2 = sc.textFile("ips.txt")
table1.join(table2)
    .filter(lambda (x, y): y[0] != "INFO")
    .map(lambda (x, y): (y[1], y[0]))
    .count()
```

some text file

code defines
data structure

Spark knows nothing about your computations

```
SELECT country, code FROM table1
JOIN table2
WHERE table1.ip = table2.ip
AND table1.code != "INFO"
```

```
rdd1 = sc.textFile("log.txt")
rdd2 = sc.textFile("ips.txt")
table1.join(table2)
      .filter(lambda (x, y): y != "INFO")
      .map(lambda (x, y): (y[1], y[0]))
      .count()
```

Summary

- RDD is imperative
- RDD is low-level
- Using RDD computations are opaque
- Using RDD data is opaque