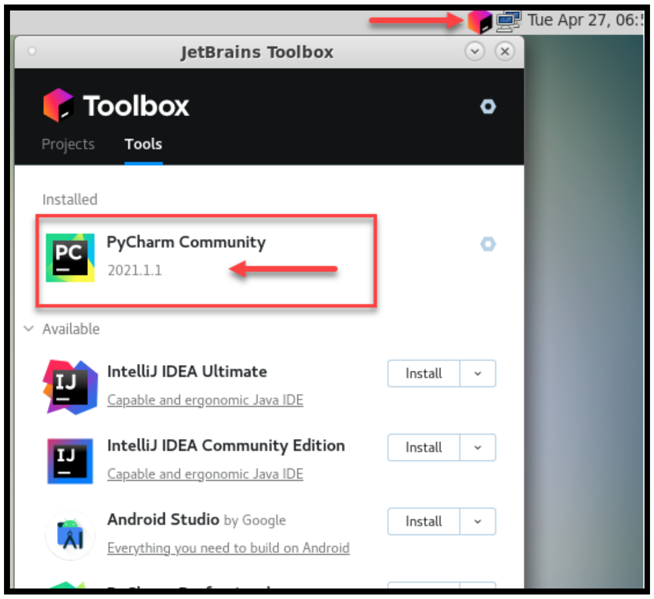
Lab: Shuffle Join

Introduction

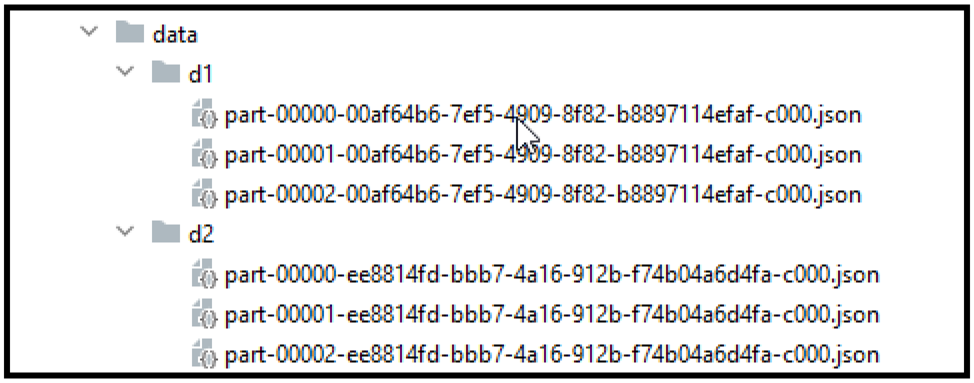
This exercise would help you to explore the internals of the Spark Join so that you can get a better understanding of the plans generated by Catalyst Optimizer for executing Joins in Spark.

Let's get Started

Run Pycharm using below screenshot



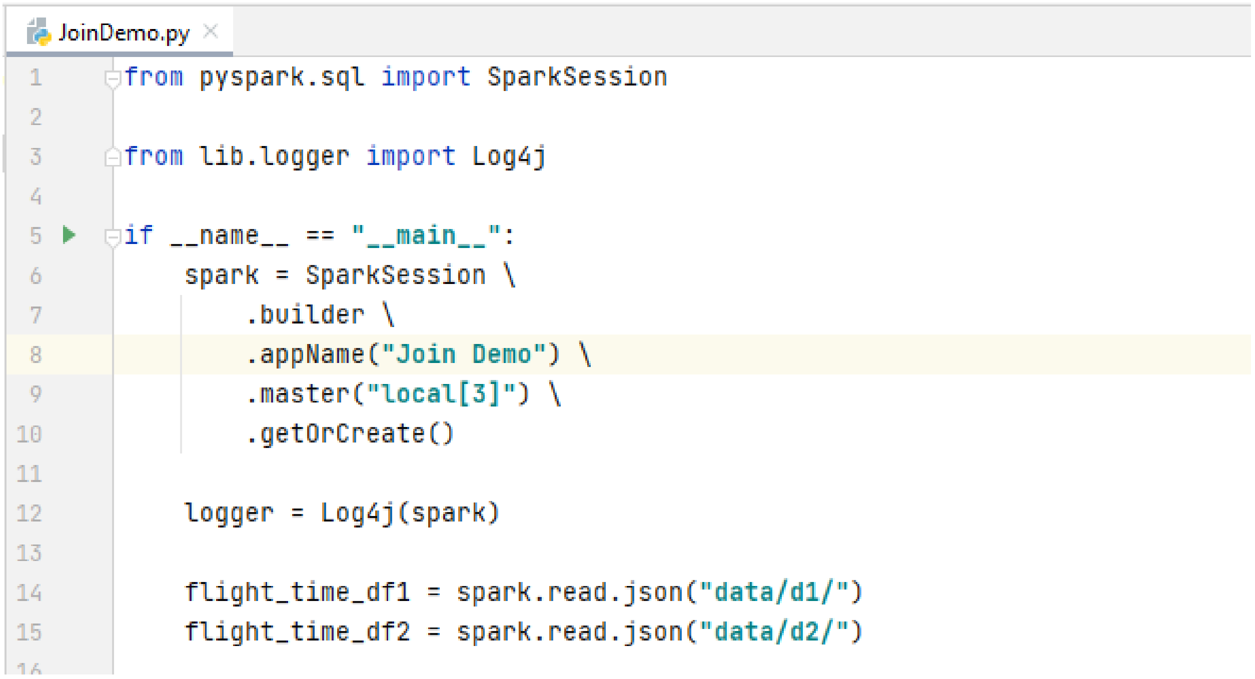
We have two data sets named as d1 and d2. Each dataset contains three data files.



```
data
  d1
    part-00000-00af64b6-7ef5-4909-8f82-b8897114efaf-c000.json
    part-00001-00af64b6-7ef5-4909-8f82-b8897114efaf-c000.json
    part-00002-00af64b6-7ef5-4909-8f82-b8897114efaf-c000.json
  d2
    part-00000-ee8814fd-bbb7-4a16-912b-f74b04a6d4fa-c000.json
    part-00001-ee8814fd-bbb7-4a16-912b-f74b04a6d4fa-c000.json
    part-00002-ee8814fd-bbb7-4a16-912b-f74b04a6d4fa-c000.json
```

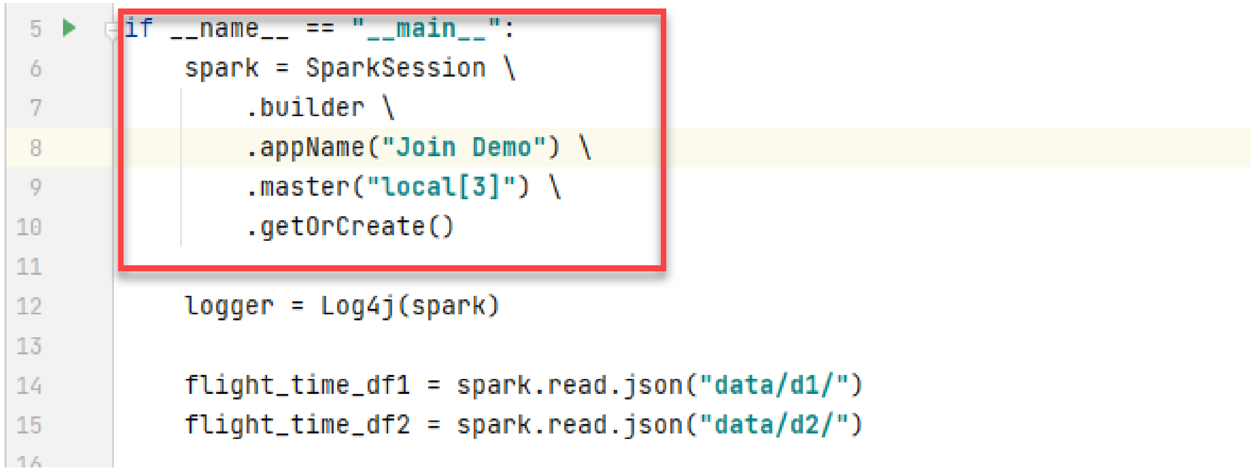
Note: We created three files purposely because we wanted to make sure we get three partitions when we read them.

Here is the code snippet:



```
from pyspark.sql import SparkSession

from lib.logger import Log4j

if __name__ == "__main__":
    spark = SparkSession \
        .builder \
        .appName("Join Demo") \
        .master("local[3]") \
        .getOrCreate()

    logger = Log4j(spark)

    flight_time_df1 = spark.read.json("data/d1/")
    flight_time_df2 = spark.read.json("data/d2/")
```

Here you can observe that, we are creating a spark session with 3 three parallel threads.



```
if __name__ == "__main__":
    spark = SparkSession \
        .builder \
        .appName("Join Demo") \
        .master("local[3]") \
        .getOrCreate()

    logger = Log4j(spark)

    flight_time_df1 = spark.read.json("data/d1/")
    flight_time_df2 = spark.read.json("data/d2/")
```

We would be reading these two datasets in two different data frames.

```
flight_time_df1 = spark.read.json("data/d1/")
flight_time_df2 = spark.read.json("data/d2/")
```

Now, set the shuffle partition configuration to get three partitions after the shuffle, which means having three reduced exchanges:

```
spark.conf.set("spark.sql.shuffle.partitions", 3)
```

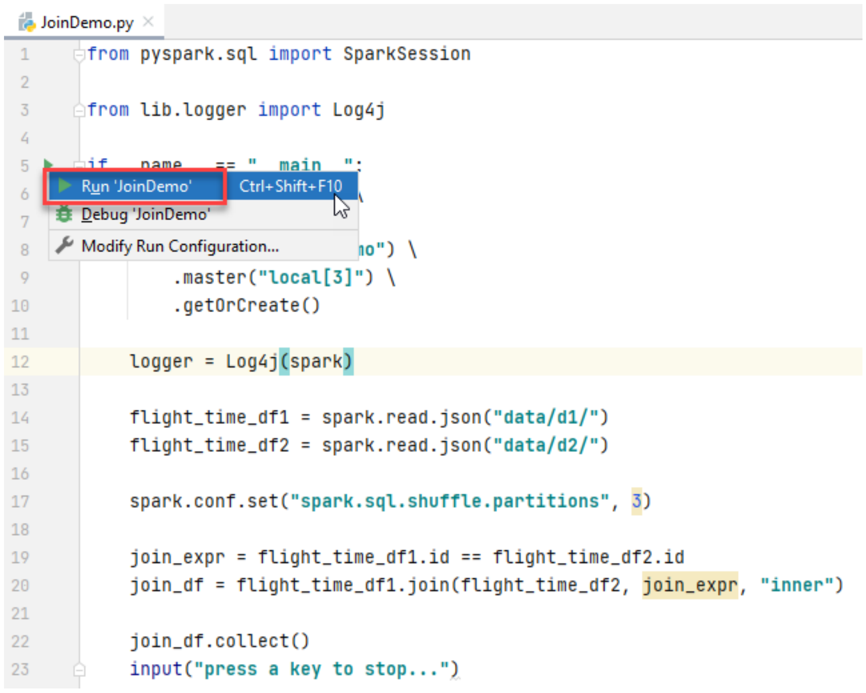
Let's define the join condition.

After performing the join operation we are going to do an inner join. But, Join is a transformation, so nothing is going to actually happen until we take action.

So, let's add a dummy action here. Then we hold the job to look at the Spark UI and understand what's going on there.

```
join_df.collect()
input("press a key to stop...")
```

Let's run it:



```
from pyspark.sql import SparkSession

from lib.logger import Log4j

if __name__ == "__main__":
                                         \

                                      no") \
        .master("local[3]") \
        .getOrCreate()

    logger = Log4j(spark)

    flight_time_df1 = spark.read.json("data/d1/")
    flight_time_df2 = spark.read.json("data/d2/")

    spark.conf.set("spark.sql.shuffle.partitions", 3)

    join_expr = flight_time_df1.id == flight_time_df2.id
    join_df = flight_time_df1.join(flight_time_df2, join_expr, "inner")

    join_df.collect()
    input("press a key to stop...")
```

Start the Spark UI.

Open the browser: http://localhost:4040

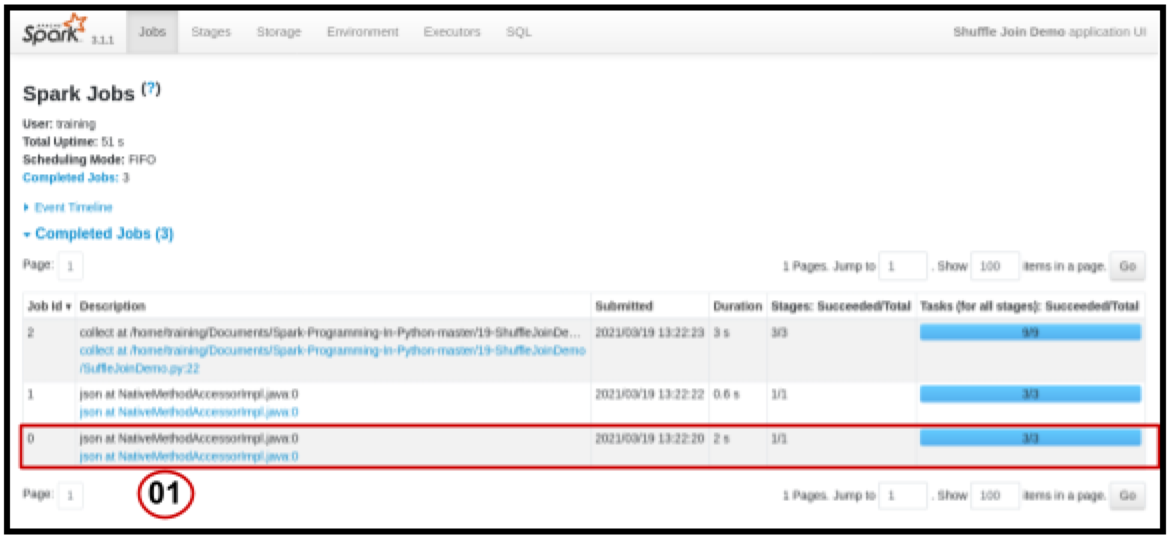
Jobs

The jobs page shows all the jobs and their run times. Additionally, the page shows how many stages each job contains.

The first job is to read the first data frame.

Reading a data frame is a single-stage and straightforward operation.

So, we have a single stage for this job.



Spark Jobs (?)

User: training
Total Uptime: 51 s
Scheduling Mode: FIFO
Completed Jobs: 3

▸ Event Timeline

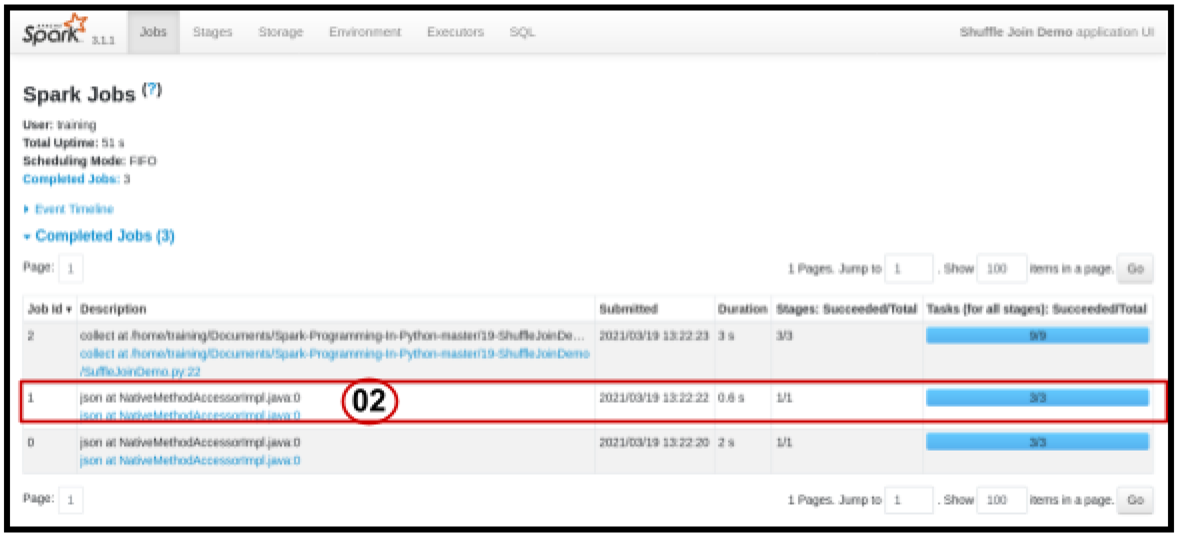
▾ Completed Jobs (3)

Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go

Job Id ▾	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
2	collect at /home/training/Documents/Spark-Programming-in-Python-master/19-ShuffleJoinDe... collect at /home/training/Documents/Spark-Programming-in-Python-master/19-ShuffleJoinDemo /SuffleJoinDemo.py:22	2021/03/19 13:22:23	3 s	3/3	9/9
1	json at NativeMethodAccessorImpl.java:0 json at NativeMethodAccessorImpl.java:0	2021/03/19 13:22:22	0.6 s	1/1	3/3
0	json at NativeMethodAccessorImpl.java:0 json at NativeMethodAccessorImpl.java:0	2021/03/19 13:22:20	2 s	1/1	3/3

Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go

The same happened for the second data frame.



Spark Jobs (?)

User: training
Total Uptime: 51 s
Scheduling Mode: FIFO
Completed Jobs: 3

▸ Event Timeline

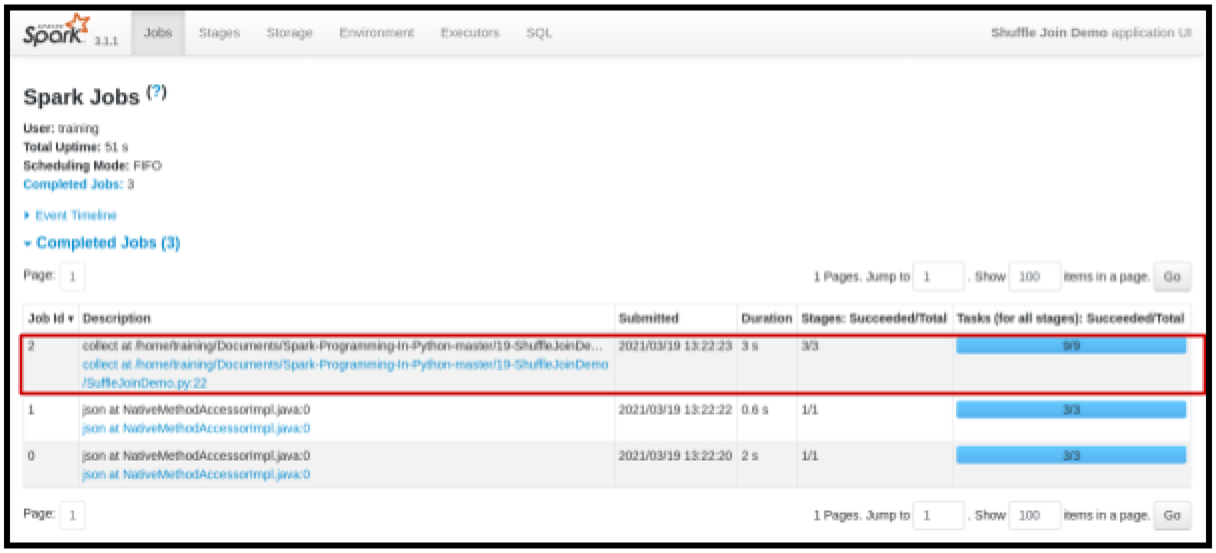
▾ Completed Jobs (3)

Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go

Job Id ▾	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
2	collect at /home/training/Documents/Spark-Programming-In-Python-master/19-ShuffleJoinDe… collect at /home/training/Documents/Spark-Programming-In-Python-master/19-ShuffleJoinDemo /SuffleJoinDemo.py:22	2021/03/19 13:22:23	3 s	3/3	9/9
1	json at NativeMethodAccessorImpl.java:0 json at NativeMethodAccessorImpl.java:0	2021/03/19 13:22:22	0.6 s	1/1	3/3
0	json at NativeMethodAccessorImpl.java:0 json at NativeMethodAccessorImpl.java:0	2021/03/19 13:22:20	2 s	1/1	3/3

Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go

The Join operation happened here.



Spark Jobs (?)

User: training
Total Uptime: 51 s
Scheduling Mode: FIFO
Completed Jobs: 3

▸ Event Timeline

▾ Completed Jobs (3)

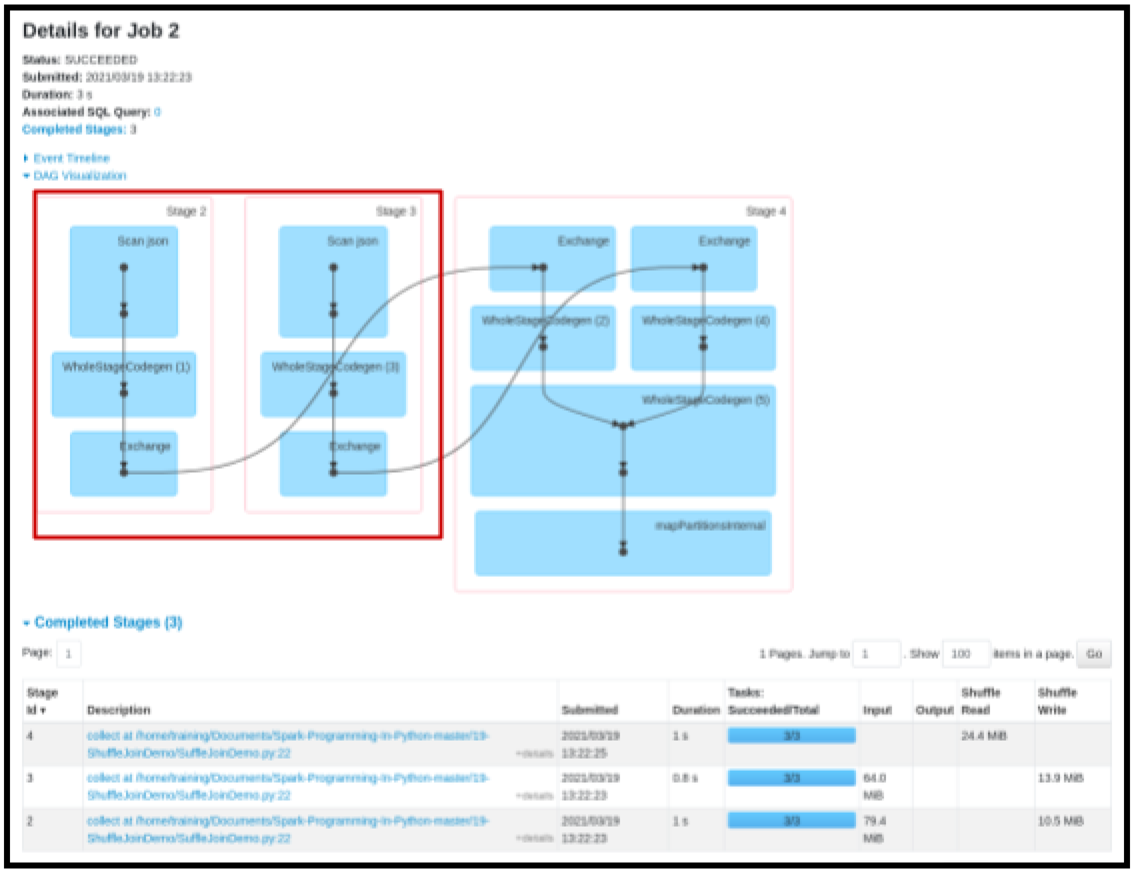
Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go

Job Id ▾	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
2	collect at /home/training/Documents/Spark-Programming-In-Python-master/19-ShuffleJoinDe… collect at /home/training/Documents/Spark-Programming-In-Python-master/19-ShuffleJoinDemo /SuffleJoinDemo.py:22	2021/03/19 13:22:23	3 s	3/3	9/9
1	json at NativeMethodAccessorImpl.java:0 json at NativeMethodAccessorImpl.java:0	2021/03/19 13:22:22	0.6 s	1/1	3/3
0	json at NativeMethodAccessorImpl.java:0 json at NativeMethodAccessorImpl.java:0	2021/03/19 13:22:20	2 s	1/1	3/3

Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go

The join operation is accomplished in three stages.

These two stages are to create a map exchange for the two data frames and lines are indicating the shuffle.



Details for Job 2

Status: SUCCEEDED
Submitted: 2021/03/19 13:22:23
Duration: 3 s
Associated SQL Query: 0
Completed Stages: 3

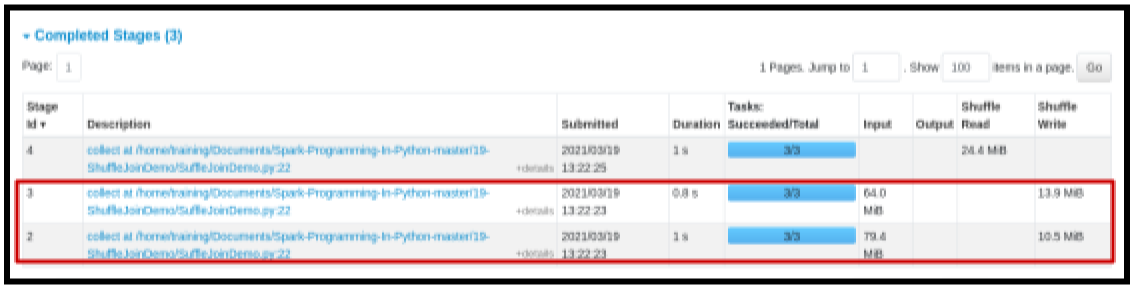
▸ Event Timeline
▾ DAG Visualization

▾ Completed Stages (3)

Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go

Stage Id ▾	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
4	collect at /home/training/Documents/Spark-Programming-In-Python-master/19-ShuffleJoinDemo/SuffleJoinDemo.py:22 +details	2021/03/19 13:22:25	1 s	3/3			24.4 MiB	
3	collect at /home/training/Documents/Spark-Programming-In-Python-master/19-ShuffleJoinDemo/SuffleJoinDemo.py:22 +details	2021/03/19 13:22:23	0.8 s	3/3	64.0 MiB			13.9 MiB
2	collect at /home/training/Documents/Spark-Programming-In-Python-master/19-ShuffleJoinDemo/SuffleJoinDemo.py:22 +details	2021/03/19 13:22:23	1 s	3/3	79.4 MiB			10.5 MiB

So data moves from the map exchange to the reduce-exchange and rest all is a simple sort-merge join.

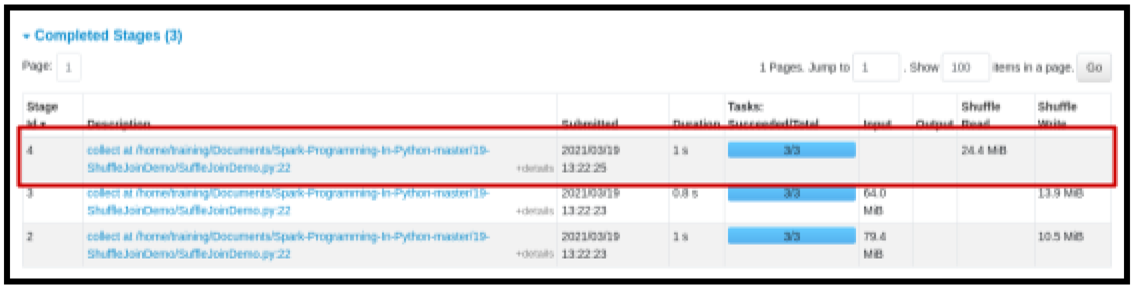


▾ Completed Stages (3)

Page: 1 — 1 Pages. Jump to 1 . Show 100 items in a page. Go

Stage Id ▾	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
4	collect at /home/training/Documents/Spark-Programming-In-Python-master/19-ShuffleJoinDemo/SuffleJoinDemo.py:22 +details	2021/03/19 13:22:25	1 s	3/3			24.4 MiB	
3	collect at /home/training/Documents/Spark-Programming-In-Python-master/19-ShuffleJoinDemo/SuffleJoinDemo.py:22 +details	2021/03/19 13:22:23	0.8 s	3/3	64.0 MiB			13.9 MiB
2	collect at /home/training/Documents/Spark-Programming-In-Python-master/19-ShuffleJoinDemo/SuffleJoinDemo.py:22 +details	2021/03/19 13:22:23	1 s	3/3	79.4 MiB			10.5 MiB

The final stage was doing a shuffle read, that thing also happened in three parallel tasks because we configured the shuffle partitions to 3.



▾ Completed Stages (3)

Page: 1   1 Pages. Jump to 1 . Show 100 items in a page. Go

Stage Id ▾	Description		Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
4	collect at /home/training/Documents/Spark-Programming-In-Python-master/19-ShuffleJoinDemo/SuffleJoinDemo.py:22	+details	2021/03/19 13:22:25	1 s	3/3			24.4 MiB	
3	collect at /home/training/Documents/Spark-Programming-In-Python-master/19-ShuffleJoinDemo/SuffleJoinDemo.py:22	+details	2021/03/19 13:22:23	0.8 s	3/3	64.0 MiB			13.9 MiB
2	collect at /home/training/Documents/Spark-Programming-In-Python-master/19-ShuffleJoinDemo/SuffleJoinDemo.py:22	+details	2021/03/19 13:22:23	1 s	3/3	79.4 MiB			10.5 MiB

Great! So now you know how this Shuffle sort-merge join works under the hood.

Knowing internals can take you too far in investigating join performance, tuning, and improving them.

Voila!! We have successfully completed this lab.