

Infraestructura para Big Data

Práctica 3 - Ecosistema Spark

José Manuel Bustos Muñoz

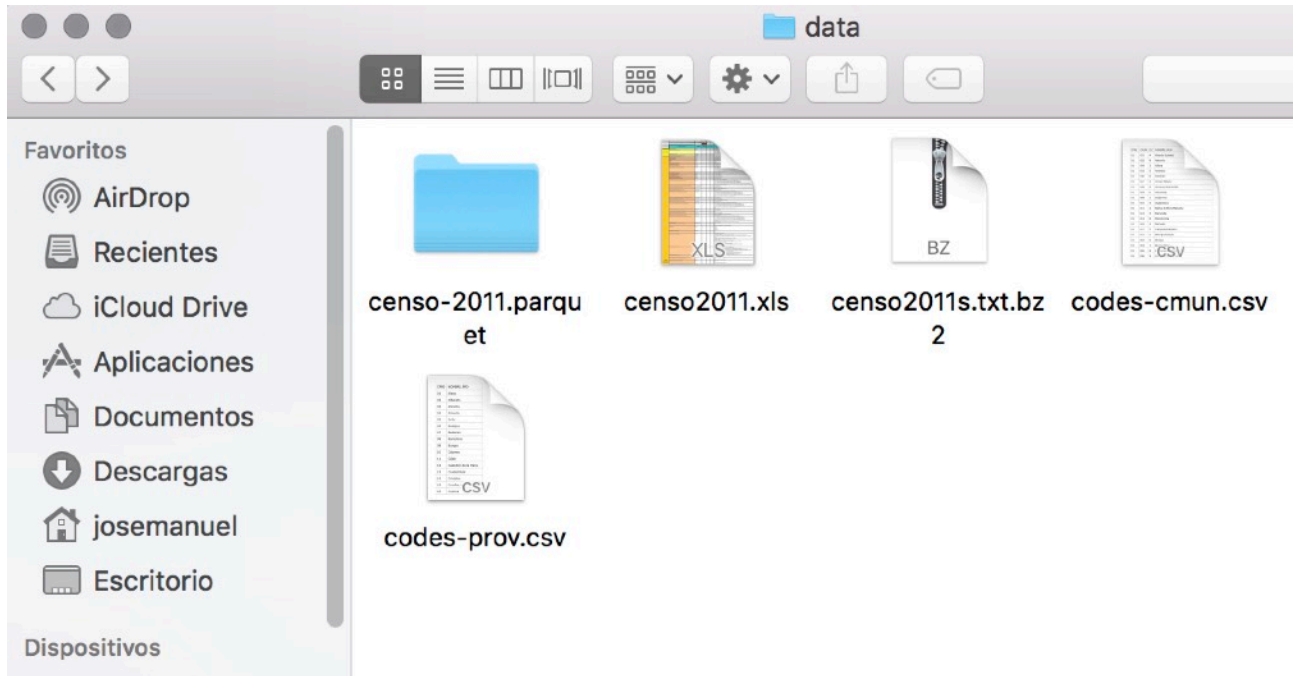
Índice

1. Extracción y procesamiento de los datos.
2. Usando la interfaz de monitorización de Spark, identifique en ella tareas (tasks) y fases (stages) correspondientes a las operaciones realizadas. Capture algunas imágenes del flujo de datos, línea de tiempos y diagramas de tareas, y explique su contenido.

1. Extracción y procesamiento de los datos.

Para cargar los datos de la práctica primero se deben extraer, y para ello se ejecuta el notebook proporcionado “01-censo-extraccion-datos” el cuál se modificó para llamar al fichero “01-censo-extraccion-datos.py”.

Una vez ejecutado ya se tienen los datos extraídos y además en formato parquet.



Con los datos se buscan las 10 provincias españolas con mayor número de inmigrantes de origen francés o portugués. La lista obtenida sería la siguiente:

	Provincia	Número de inmigrantes
0	Madrid	166
1	Barcelona	156
2	Valencia	88
3	Alicante	58
4	Murcia	57
5	Málaga	45
6	Pontevedra	39
7	Baleares	33
8	Cádiz	29
9	Coruña, La	28

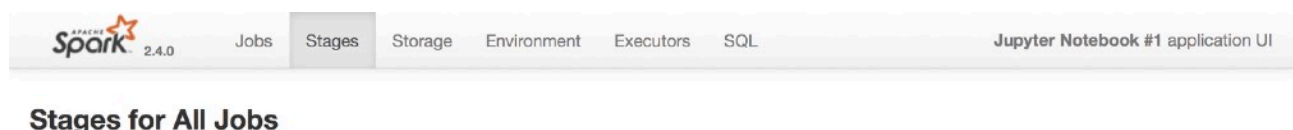
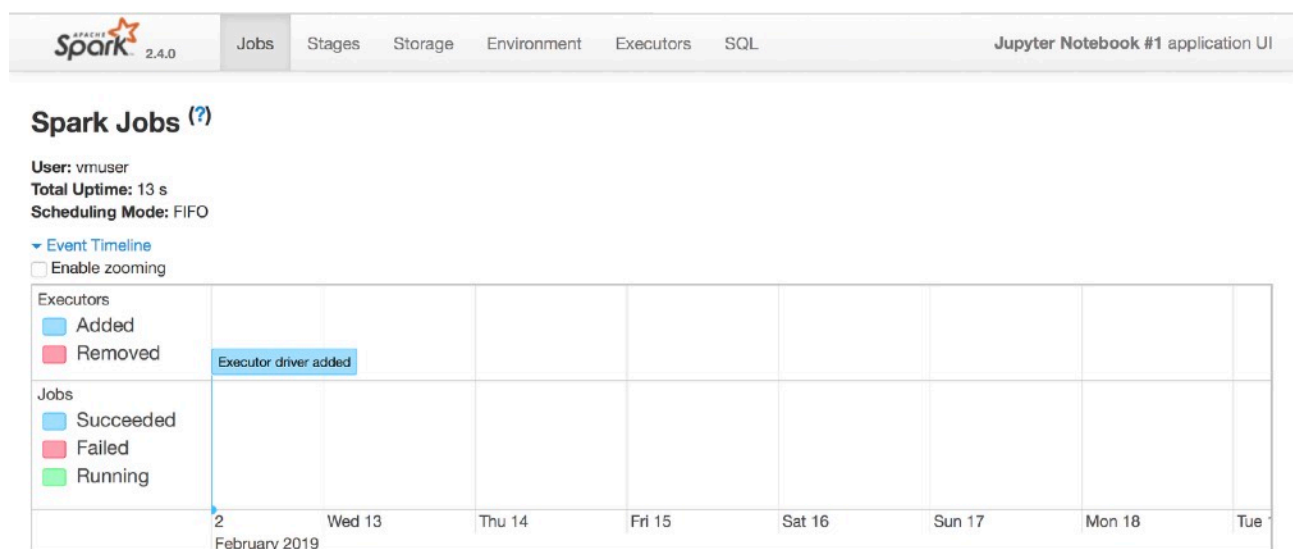
2. Usando la interfaz de monitorización de Spark, identifique en ella tareas (tasks) y fases (stages) correspondientes a las operaciones realizadas. Capture algunas imágenes del flujo de datos, línea de tiempos y diagramas de tareas, y explique su contenido.

Vamos a la interfaz de monitorización de Spark, accediendo al puerto 4040, y allí pueden observarse los jobs que corren en Spark, las tareas, los ejecutores, y más opciones y elementos de Spark.

Los Jobs son los trabajos de Spark, que terminan en una acción. El DAG genera su longitud y está compuesto por distintos Stages. A su vez cada Stage tiene varios Tasks.

Vamos a visualizar los jobs, stages y tasks que se han ejecutado al realizar y correr la práctica.

Al acceder antes de correr el notebook vemos que no hay todavía ningún Job que haya sido ejecutado:



Una vez ejecutamos el notebook completo, volvemos a acceder ya se pueden ver los jobs ejecutados. En la tabla con los jobs se puede ver información como la duración, fecha y hora, o el número de stages del Job.

APACHE
Spark 2.4.0

Jobs

Stages

Storage

Environment

Executors

SQL

Jupyter Notebook #1 application UI

Spark Jobs (?)

User: vmuser
Total Uptime: 19 min
Scheduling Mode: FIFO
Completed Jobs: 25

▶ Event Timeline

▼ Completed Jobs (25)

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
24	toPandas at <ipython-input-14-a670d47fef84>:2 toPandas at <ipython-input-14-a670d47fef84>:2	2019/02/12 17:44:55	0.3 s	2/2	11/11
23	toPandas at <ipython-input-14-a670d47fef84>:2 toPandas at <ipython-input-14-a670d47fef84>:2	2019/02/12 17:44:55	0.1 s	1/1	1/1
22	toPandas at <ipython-input-14-a670d47fef84>:2 toPandas at <ipython-input-14-a670d47fef84>:2	2019/02/12 17:44:53	2 s	3/3	202/202
21	showString at NativeMethodAccessorImpl.java:0 showString at NativeMethodAccessorImpl.java:0	2019/02/12 17:44:47	68 ms	1/1	1/1
20	run at ThreadPoolExecutor.java:1149 run at ThreadPoolExecutor.java:1149	2019/02/12 17:44:46	1 s	3/3	202/202
19	showString at NativeMethodAccessorImpl.java:0 showString at NativeMethodAccessorImpl.java:0	2019/02/12 17:44:43	2 s	3/3	202/202

Si accedemos a la pestaña de Stages igualmente se ve en la tabla información de fecha y hora, duración, o de las tasks ejecutadas en cada Stage:

APACHE
Spark 2.4.0

Jobs

Stages

Storage

Environment

Executors

SQL

Jupyter Notebook #1 application UI

Stages for All Jobs

Completed Stages: 39
Skipped Stages: 8

▼ Completed Stages (39)

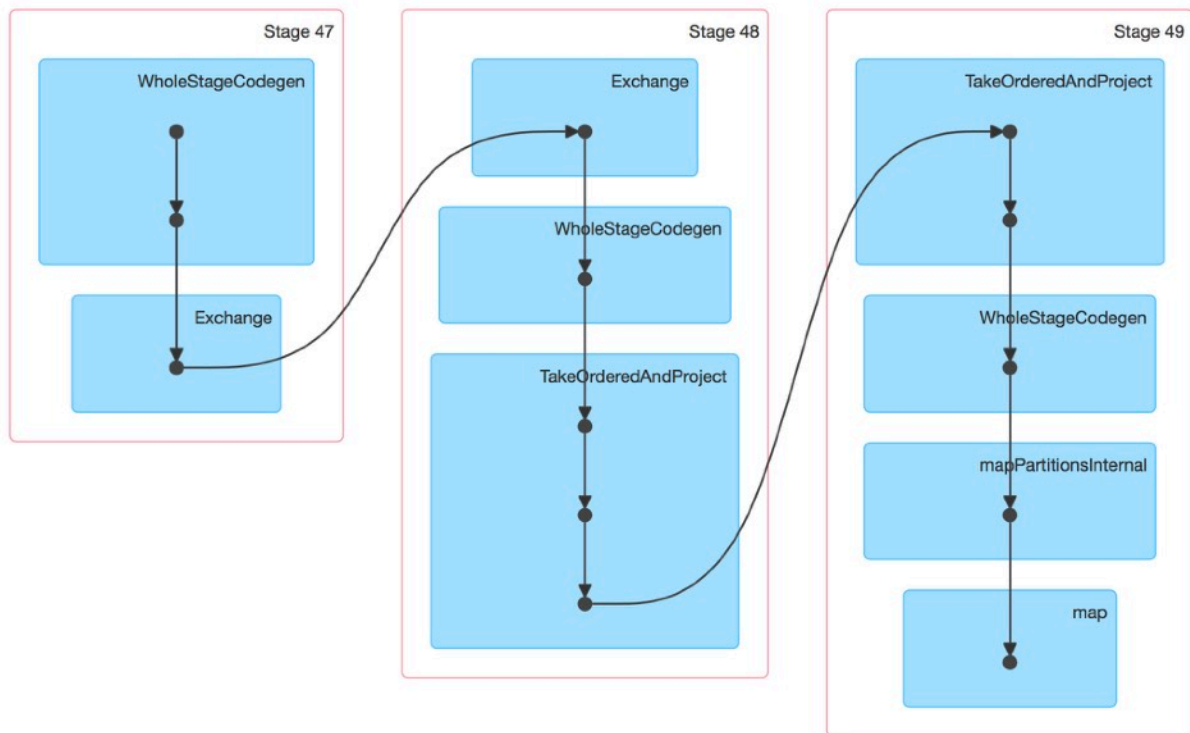
Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
46	toPandas at <ipython-input-14-a670d47fef84>:2 +details	2019/02/12 17:44:55	0.2 s	10/10			676.0 B	
45	toPandas at <ipython-input-14-a670d47fef84>:2 +details	2019/02/12 17:44:55	78 ms	1/1	641.0 B			676.0 B
44	toPandas at <ipython-input-14-a670d47fef84>:2 +details	2019/02/12 17:44:55	98 ms	1/1	641.0 B			
43	toPandas at <ipython-input-14-a670d47fef84>:2 +details	2019/02/12 17:44:55	22 ms	1/1			3.2 KB	
42	toPandas at <ipython-input-14-a670d47fef84>:2 +details	2019/02/12 17:44:54	1 s	200/200			3.2 KB	3.2 KB
41	toPandas at <ipython-input-14-a670d47fef84>:2 +details	2019/02/12 17:44:53	0.4 s	1/1	324.4 KB			3.2 KB
40	showString at NativeMethodAccessorImpl.java:0 +details	2019/02/12 17:44:47	62 ms	1/1	641.0 B			

Por ejemplo al ejecutar el código siguiente:

```
df_prov_top = df_agrup_prov.orderBy('count', ascending=False).limit(10)
df_prov_top.show()
```

Podemos acceder al job generado, y ver el diagrama del Job y como se reparte en 3 etapas o stages. Al final de la segunda etapa estaría ordenada la información, y al final del tercer paso se produce el map para obtener los datos resultantes a mostrar:

▼ DAG Visualization



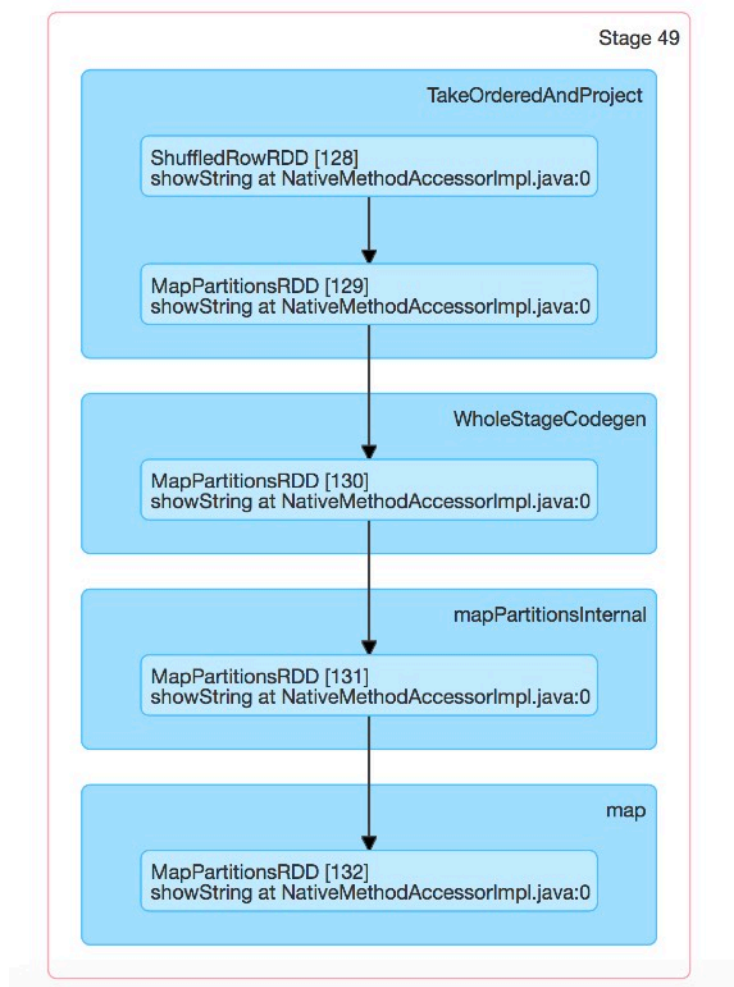
Tanto la visualización de DAG como los gráficos de líneas de tiempo se pueden ver para un Job o para un Stage.

Otra información que se puede ver al acceder a un Job en concreto de la lista de los jobs, son las Stages que lo componen, y se podrá acceder a cada una de ellas para ver la información del Stage:

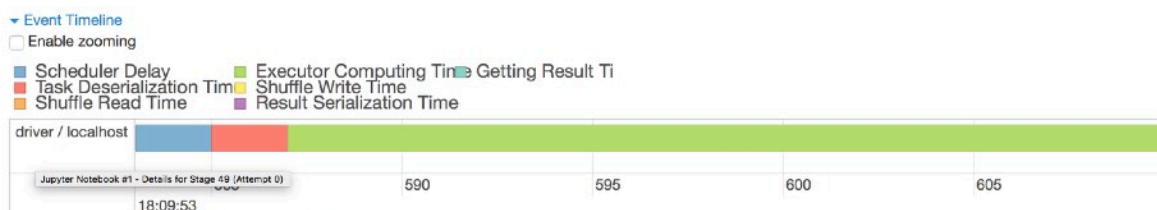
▼ Completed Stages (3)

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
49	showString at NativeMethodAccessorImpl.java:0 +details	2019/02/12 18:09:53	27 ms	1/1			3.2 KB	
48	showString at NativeMethodAccessorImpl.java:0 +details	2019/02/12 18:09:52	1 s	200/200			3.2 KB	3.2 KB
47	showString at NativeMethodAccessorImpl.java:0 +details	2019/02/12 18:09:52	0.3 s	1/1	324.4 KB			3.2 KB

Accedemos desde la tabla de Stages del Job, a una de las etapas, y accedemos a su pantalla de información donde tendremos como antes su flujo DAG:



Podremos ver la línea de tiempo, y además visualizar multitud de métricas que pueden ayudar en la monitorización de Spark. Una de las mayores utilidades de visualizar los eventos en líneas de tiempo es detectar posibles cuellos de botella.



Summary Metrics for 1 Completed Tasks

Metric	Min	25th percentile	Median	75th percentile	Max
Duration	23 ms	23 ms	23 ms	23 ms	23 ms
Scheduler Delay	2 ms	2 ms	2 ms	2 ms	2 ms
Task Deserialization Time	2 ms	2 ms	2 ms	2 ms	2 ms
GC Time	5 ms	5 ms	5 ms	5 ms	5 ms
Result Serialization Time	0 ms	0 ms	0 ms	0 ms	0 ms
Getting Result Time	0 ms	0 ms	0 ms	0 ms	0 ms
Peak Execution Memory	0.0 B	0.0 B	0.0 B	0.0 B	0.0 B
Shuffle Read Blocked Time	0 ms	0 ms	0 ms	0 ms	0 ms
Shuffle Read Size / Records	3.2 KB / 52	3.2 KB / 52	3.2 KB / 52	3.2 KB / 52	3.2 KB / 52

Finalmente para la Stage en cuestión aparece cierta información del Ejecutor y de las tareas de dicha etapa:

▼ Aggregated Metrics by Executor

Executor ID ▲	Address	Task Time	Total Tasks	Failed Tasks	Killed Tasks	Succeeded Tasks	Shuffle Read Size / Records	Blacklisted
driver	10.0.2.15:37193	27 ms	1	0	0	1	3.2 KB / 52	false

▼ Tasks (1)

Index ▲	ID	Attempt	Status	Locality Level	Executor ID	Host	Launch Time	Duration	Scheduler Delay	Task Deserialization Time	GC Time	Result Serialization Time	Getting Result Time	Peak Execution Memory	Shuffle Read Blocked Time
0	1435	0	SUCCESS	ANY	driver	localhost	2019/02/12 18:09:53	27 ms	2 ms	2 ms	5 ms	0 ms	0 ms	0.0 B	0 ms

Si estuviéramos ejecutando en un cluster con varias máquinas en la línea de tiempo y eventos se reflejaría el trabajo y eventos de cada una, pero en este caso al ser un sólo nodo sólo aparece una línea de eventos.