

localhost:4040/jobs/

Slack

MyGoitSparkSandbox application UI

Spark 3.2.0

JobsStagesStorageEnvironmentExecutorsSQL

Spark Jobs (?)

User: vprymak

Total Uptime: 50 s

Scheduling Mode: FIFO

Completed Jobs: 5

Event Timeline

Completed Jobs (5)

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
4	collect at /home/vprymak/Documents/PyTEST/test.py:15 collect at /home/vprymak/Documents/PyTEST/test.py:15	2024/11/17 19:01:28	0.1 s	1/1 (2 skipped)	1/1 (3 skipped)
3	collect at /home/vprymak/Documents/PyTEST/test.py:15 collect at /home/vprymak/Documents/PyTEST/test.py:15	2024/11/17 19:01:28	0.2 s	1/1 (1 skipped)	2/2 (1 skipped)
2	collect at /home/vprymak/Documents/PyTEST/test.py:15 collect at /home/vprymak/Documents/PyTEST/test.py:15	2024/11/17 19:01:27	0.2 s	1/1	1/1
1	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2024/11/17 19:01:27	0.3 s	1/1	1/1
0	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2024/11/17 19:01:26	0.5 s	1/1	1/1

Page: 1

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

Всі 5 джоб присутні.

localhost:4040/jobs/

Slack

MyGoitSparkSandbox application UI

Spark 3.2.0

JobsStagesStorageEnvironmentExecutorsSQL

Spark Jobs (?)

User: vprymak

Total Uptime: 14 s

Scheduling Mode: FIFO

Completed Jobs: 8

Event Timeline

Completed Jobs (8)

Page: 1

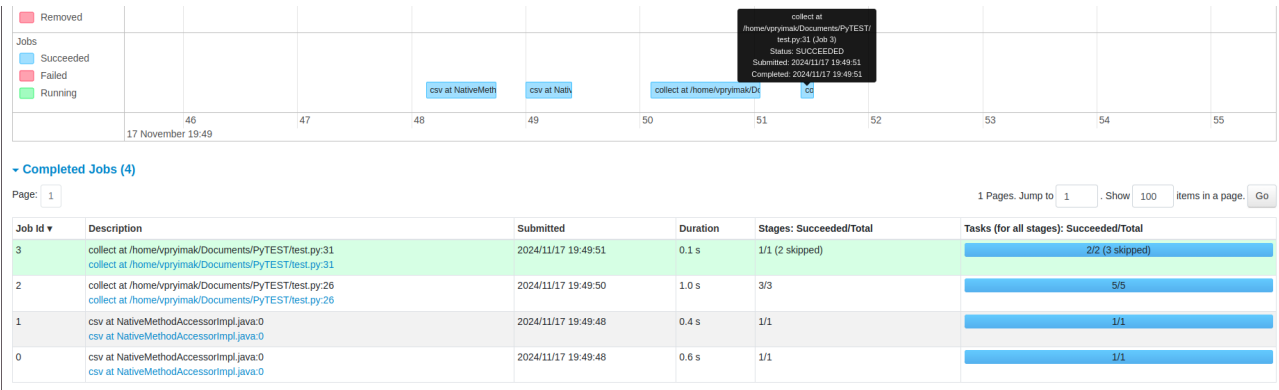
1 Pages. Jump to 1. Show 100 items in a page. Go

Job Id ▾	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
7	collect at /home/vprymak/Documents/PyTEST/test.py:18 collect at /home/vprymak/Documents/PyTEST/test.py:18	2024/11/17 19:04:34	42 ms	1/1 (2 skipped)	1/1 (3 skipped)
6	collect at /home/vprymak/Documents/PyTEST/test.py:18 collect at /home/vprymak/Documents/PyTEST/test.py:18	2024/11/17 19:04:33	51 ms	1/1 (1 skipped)	2/2 (1 skipped)
5	collect at /home/vprymak/Documents/PyTEST/test.py:18 collect at /home/vprymak/Documents/PyTEST/test.py:18	2024/11/17 19:04:33	0.1 s	1/1	1/1
4	collect at /home/vprymak/Documents/PyTEST/test.py:13 collect at /home/vprymak/Documents/PyTEST/test.py:13	2024/11/17 19:04:33	46 ms	1/1 (2 skipped)	1/1 (3 skipped)
3	collect at /home/vprymak/Documents/PyTEST/test.py:13 collect at /home/vprymak/Documents/PyTEST/test.py:13	2024/11/17 19:04:33	0.2 s	1/1 (1 skipped)	2/2 (1 skipped)
2	collect at /home/vprymak/Documents/PyTEST/test.py:13 collect at /home/vprymak/Documents/PyTEST/test.py:13	2024/11/17 19:04:32	0.2 s	1/1	1/1
1	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2024/11/17 19:04:32	0.4 s	1/1	1/1
0	csv at NativeMethodAccessorImpl.java:0 csv at NativeMethodAccessorImpl.java:0	2024/11/17 19:04:31	0.4 s	1/1	1/1

Page: 1

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

Проміжний collect запускає виконання обчислень, а інший .collect запускає їх заново



Наскільки я зрозумів, .cache запускає всі обчислення в одній джобі, тому в мене їх лише 4. Ну а зменшення мало б бути за рахунок того, що другий .collect виконується вже на закешованих даних і тому виконує тільки одну операцію фільтрації.