Spark Jobs (?)

User: ubuntu
Total Uptime: 30 min
Scheduling Mode: FIFO
Completed Jobs: 15

▸ Event Timeline

▾ **Completed Jobs (15)**

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
14	showString at NativeMethodAccessorImpl.java:0 showString at NativeMethodAccessorImpl.java:0	2024/02/28 05:02:36	0.2 s	1/1 (15 skipped)	1/1 (102 skipped)
13	runJob at PythonRDD.scala:166 runJob at PythonRDD.scala:166	2024/02/28 05:02:34	2 s	2/2 (14 skipped)	9/9 (94 skipped)
12	sortBy at /tmp/ipykernel_909/1349344683.py:1 sortBy at /tmp/ipykernel_909/1349344683.py:1	2024/02/28 05:02:33	1 s	1/1 (14 skipped)	8/8 (94 skipped)
11	sortBy at /tmp/ipykernel_909/1349344683.py:1 sortBy at /tmp/ipykernel_909/1349344683.py:1	2024/02/28 04:57:57	4.6 min	13/13 (2 skipped)	98/98 (4 skipped)
10	showString at NativeMethodAccessorImpl.java:0 showString at NativeMethodAccessorImpl.java:0	2024/02/28 04:54:03	0.2 s	1/1 (15 skipped)	1/1 (102 skipped)
9	runJob at PythonRDD.scala:166 runJob at PythonRDD.scala:166	2024/02/28 04:54:01	1 s	2/2 (14 skipped)	9/9 (94 skipped)
8	sortBy at /tmp/ipykernel_909/1349344683.py:1 sortBy at /tmp/ipykernel_909/1349344683.py:1	2024/02/28 04:54:01	0.8 s	1/1 (14 skipped)	8/8 (94 skipped)
7	sortBy at /tmp/ipykernel_909/1349344683.py:1 sortBy at /tmp/ipykernel_909/1349344683.py:1	2024/02/28 04:50:55	3.1 min	13/13 (2 skipped)	98/98 (4 skipped)
6	save at NativeMethodAccessorImpl.java:0 save at NativeMethodAccessorImpl.java:0	2024/02/28 04:50:32	8 s	1/1 (22 skipped)	22/22 (264 skipped)
5	runJob at PythonRDD.scala:166 runJob at PythonRDD.scala:166	2024/02/28 04:50:31	0.4 s	1/1 (22 skipped)	1/1 (264 skipped)
4	runJob at PythonRDD.scala:166 runJob at PythonRDD.scala:166	2024/02/28 04:48:40	0.3 s	1/1 (22 skipped)	1/1 (264 skipped)
3	showString at NativeMethodAccessorImpl.java:0 showString at NativeMethodAccessorImpl.java:0	2024/02/28 04:48:24	0.9 s	1/1 (23 skipped)	1/1 (286 skipped)
2	runJob at PythonRDD.scala:166 runJob at PythonRDD.scala:166	2024/02/28 04:48:13	5 s	2/2 (22 skipped)	23/23 (264 skipped)
1	sortBy at /tmp/ipykernel_909/3850144545.py:1 sortBy at /tmp/ipykernel_909/3850144545.py:1	2024/02/28 04:48:08	4 s	1/1 (22 skipped)	22/22 (264 skipped)
0	sortBy at /tmp/ipykernel_909/3850144545.py:1 sortBy at /tmp/ipykernel_909/3850144545.py:1	2024/02/28 04:34:44	13 min	23/23	286/286

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

During the performance analysis, we observed that Task 1, Task 2, and Task 3 took 13 minutes, 3.1 minutes, and 4.6 minutes to complete, respectively. By examining the DAG lineage graphs, statistics, and the number of tasks executed for each Spark Job, it was noted that Task 1 had the longest execution time. The primary reason was that the number of partitions for its RDD was too low, resulting in underutilization of all cores across our two VMs. To make better use of these cores and enhance parallelism, I increased the number of partitions to 8 in Task 2. This adjustment allowed for a more even distribution of data across more partitions, enabling more executor cores to participate in the computation, significantly reducing task completion time.

After killing one of the workers, the runtime for the same tasks increased by about 50%, from 3 minutes to 4.5 minutes. This increase can be attributed to the reduction in available computing resources. With one fewer worker, the cluster had fewer cores and memory available for task execution, leading to longer processing times as tasks queued up for the remaining resources.