Pranav Akinepalli
Sameer Haniyur
Viren Velacheri
April 2nd, 2022

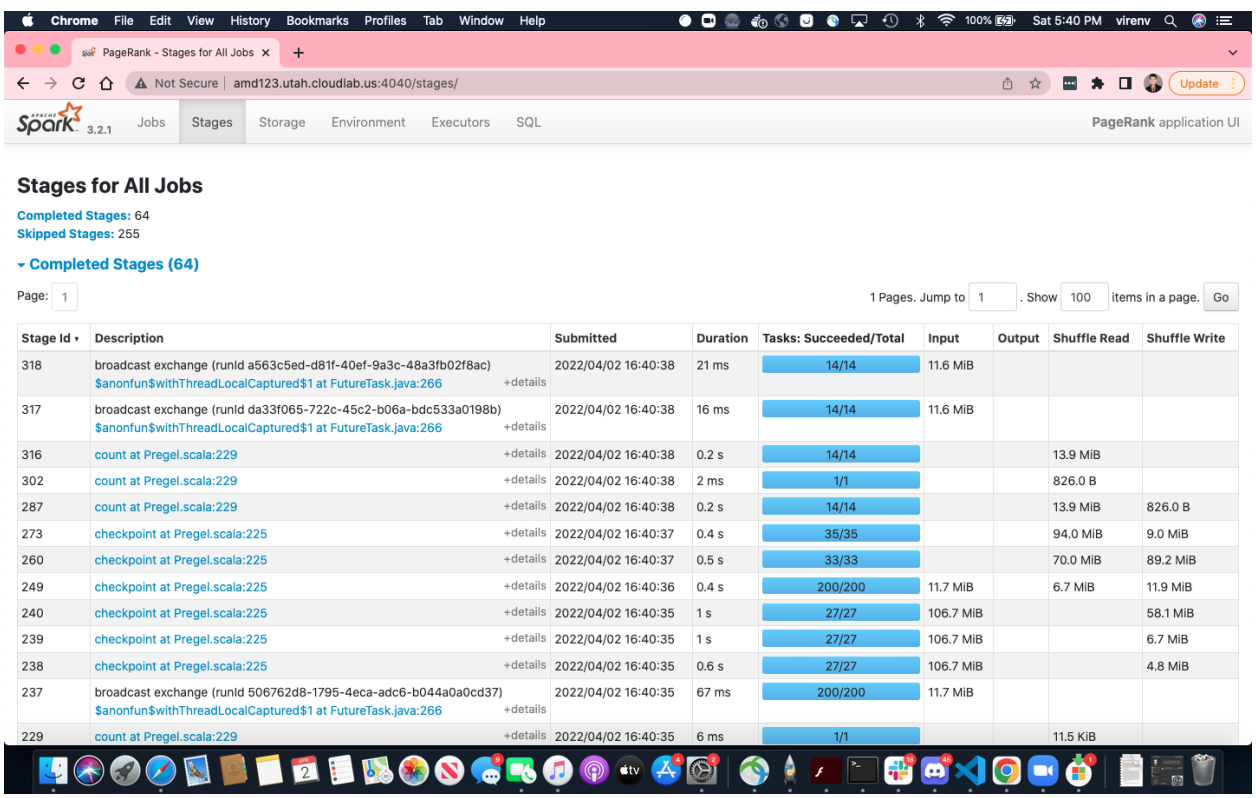
Assignment 4 Report

Task 2 Analysis

Runtime Comparison: The GraphFrames application's runtime was 56 seconds, while the runtime of the PageRank program from assignment 2 was 1 minute and 18 seconds. The GraphFrames application finishes in approximately 70% of the time that it takes for the PageRank program from assignment 2 to complete.

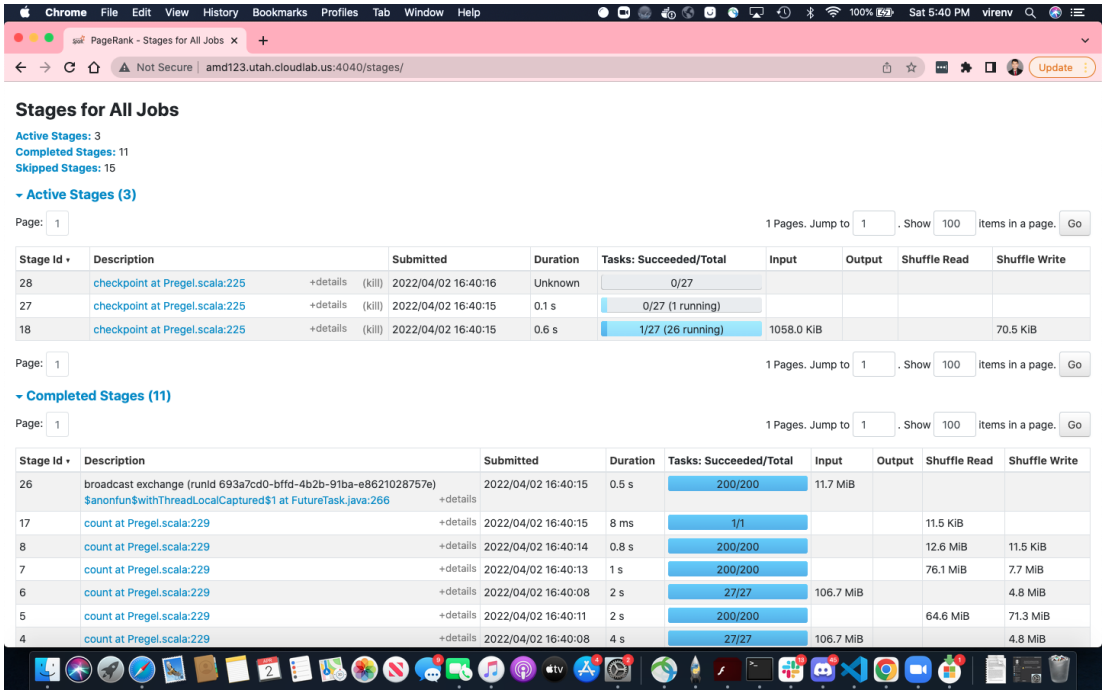
Task 3 Analysis

Screenshots

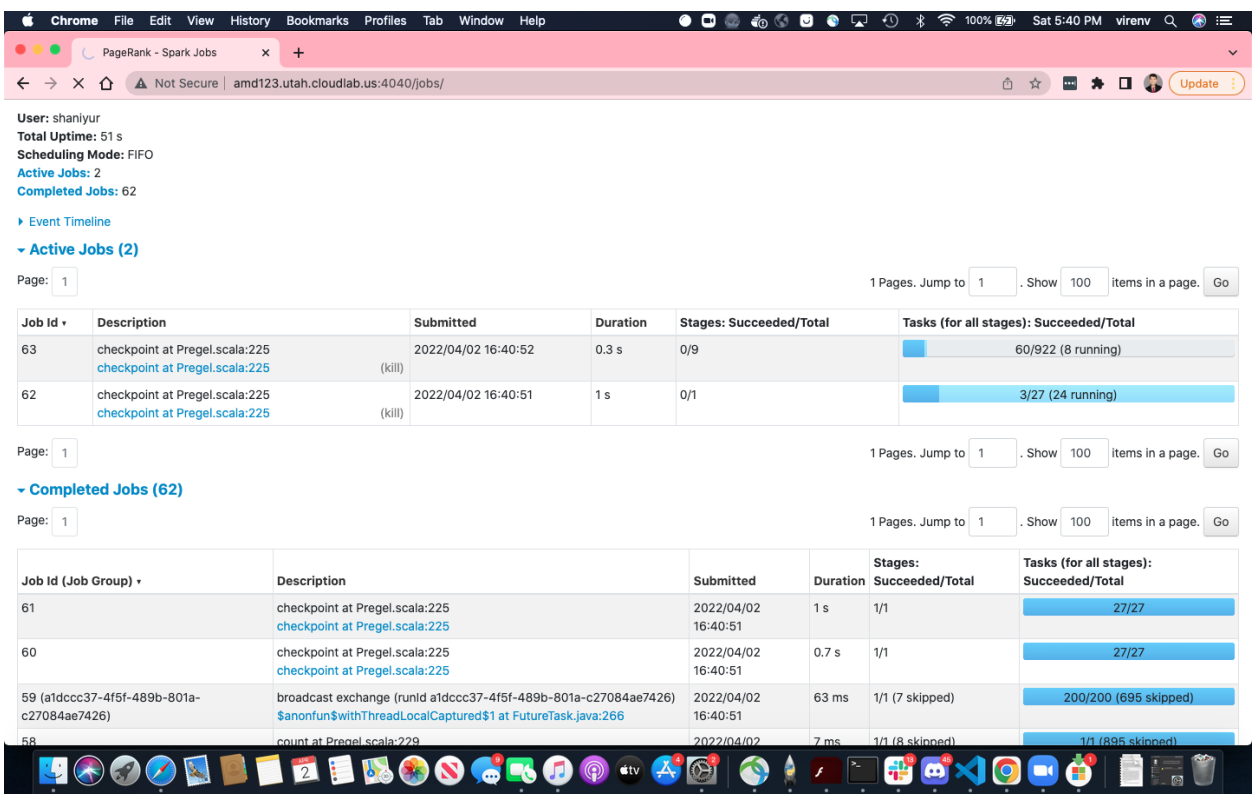


Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
318	broadcast exchange (runId a563c5ed-d81f-40ef-9a3c-48a3fb02f8ac) $anonfun$withThreadLocalCaptured$1 at FutureTask.java:266 +details	2022/04/02 16:40:38	21 ms	14/14	11.6 MiB			
317	broadcast exchange (runId da33f065-722c-45c2-b06a-bdc533a0198b) $anonfun$withThreadLocalCaptured$1 at FutureTask.java:266 +details	2022/04/02 16:40:38	16 ms	14/14	11.6 MiB			
316	count at Pregel.scala:229 +details	2022/04/02 16:40:38	0.2 s	14/14			13.9 MiB	
302	count at Pregel.scala:229 +details	2022/04/02 16:40:38	2 ms	1/1			826.0 B	
287	count at Pregel.scala:229 +details	2022/04/02 16:40:38	0.2 s	14/14			13.9 MiB	826.0 B
273	checkpoint at Pregel.scala:225 +details	2022/04/02 16:40:37	0.4 s	35/35			94.0 MiB	9.0 MiB
260	checkpoint at Pregel.scala:225 +details	2022/04/02 16:40:37	0.5 s	33/33			70.0 MiB	89.2 MiB
249	checkpoint at Pregel.scala:225 +details	2022/04/02 16:40:36	0.4 s	200/200	11.7 MiB		6.7 MiB	11.9 MiB
240	checkpoint at Pregel.scala:225 +details	2022/04/02 16:40:35	1 s	27/27	106.7 MiB			58.1 MiB
239	checkpoint at Pregel.scala:225 +details	2022/04/02 16:40:35	1 s	27/27	106.7 MiB			6.7 MiB
238	checkpoint at Pregel.scala:225 +details	2022/04/02 16:40:35	0.6 s	27/27	106.7 MiB			4.8 MiB
237	broadcast exchange (runId 506762d8-1795-4eca-adc6-b044a0a0cd37) $anonfun$withThreadLocalCaptured$1 at FutureTask.java:266 +details	2022/04/02 16:40:35	67 ms	200/200	11.7 MiB			
229	count at Pregel.scala:229 +details	2022/04/02 16:40:35	6 ms	1/1			11.5 KiB	

(Stages running and completed by around middle of program)



(Main stages involved appear to be count at Pregel, the broadcast exchange, and checkpoint)



(Timeline near end of program)

Explanation

Due to how much faster the GraphFrames application was than the vanilla PageRank application from the second assignment, there are definitely some additional benefits that GraphFrames provide. One of the things we believe that is beneficial is the parallelism that it provides. As seen by the very last screenshot, approximately 60 some jobs were completed by the end of the application. We looked back at our screenshot from assignment 2 and noticed that only 27 jobs were completed at that point in time. It seems like the tasks are split off into smaller chunks and this definitely helps in improved processing speed. Going off this, GraphFrames take advantage of the fact that graphing algorithms like PageRank can be implemented such that the vertices can pass message functions to the neighboring vertices and the aggregation and calculation of these messages is what leads to the final output. This is where Pregel comes into play as it utilizes this message passing idea, and overall offers a more natural way of expressing graph computations without reliance on conversion and reorganization into purely data flow operators like map/reduce as seen in Assignment 2. As a result, the utilization of graph-parallel processing and a vertex-centric approach enables the efficient leveraging of parallelism that, as we observed, noticeably improved the runtime performance of our application.

Specific Contributions of each Group Member

Note: In general, all of the members worked together on all aspects and would really only work on the project when everyone was present. However, each member specialized in driving some aspect of the project.

Pranav: Pranav was mainly in charge of determining the logic of what needed to be done. This project required usage of the GraphFrames API and using Pregel to implement the PageRank algorithm from Assignment 2. Pranav had the most experience with Spark so it was easier for him to lead the coding aspect due to his prior experience. He also led the coding of the algorithm during Assignment 2 so it was easier for him to lead in coding the algorithm using GraphFrames. However, coding was still largely a group effort. Pranav also looked at documentation to meet the criteria of the assignment and worked towards making sure all of the files are well commented and fleshed out in terms of the READMEs and scripts.

Viren: Viren was mainly in charge of gathering the screenshot evidence and analyzing how and why the GraphFrames application was better than the PageRank algorithm from assignment 2. He also played a major role in setting up the repo and in reading up on some of the documentation around GraphFrames and how Pregel could be used in the program. The

documentation he found and read were very helpful in the writing of the application. However, as mentioned and implied from above, everything done was pretty much a group effort. Everyone always showed up to all the meetups we had.

Sameer: Sameer was tasked with maintaining the CloudLab cluster and ensuring that Hadoop and Spark were correctly installed on the nodes. He played a large role in compiling information initially from the provided tutorial in class and applying it to our setup so that we had sufficient boiler-plate code specific to GraphFrame that we would use to begin the implementation of the PageRank algorithm itself. Since he was primarily responsible for the cluster, Sameer shared his screen as the driver whenever our group would convene. Code development however was still a collective effort that each group member contributed to.