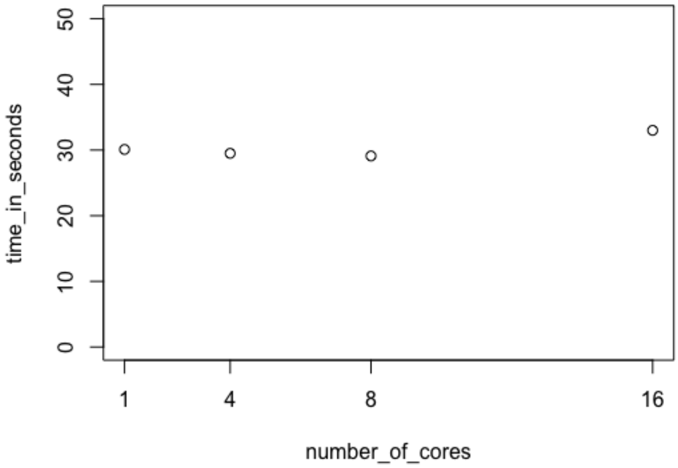
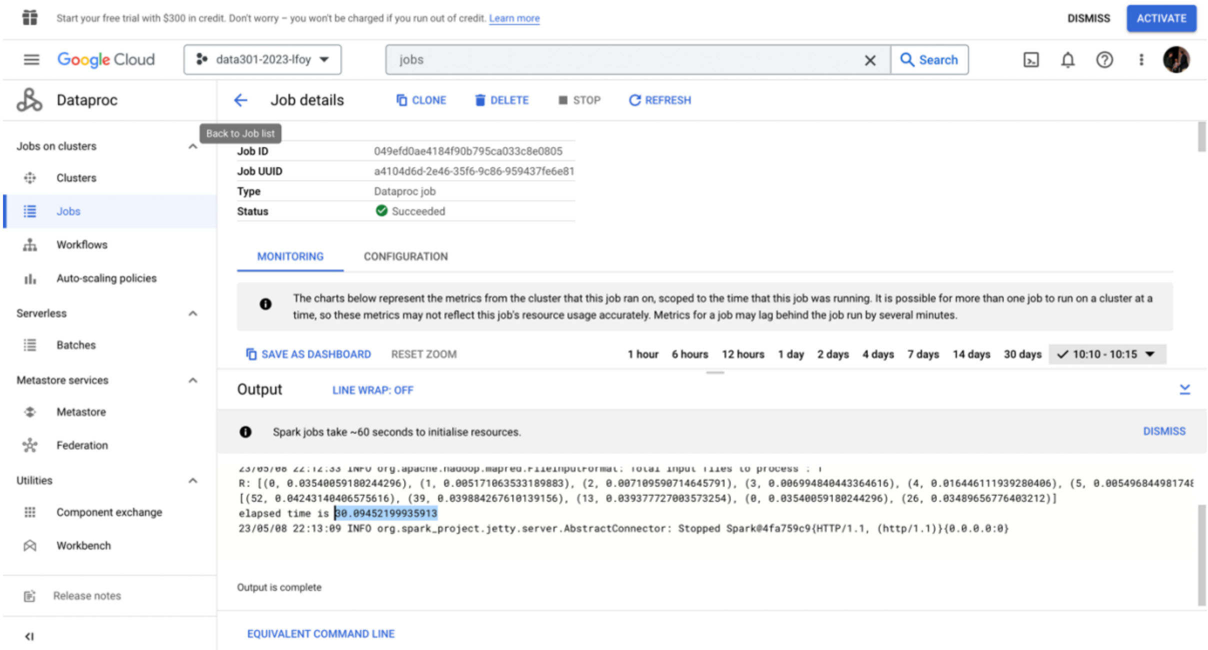
(a).



P=1 Screenshot:



Job ID	049efd0ae4184f90b795ca033c8e0805
Job UUID	a4104d6d-2e46-35f6-9c86-959437fe6e81
Type	Dataproc job
Status	✓ Succeeded

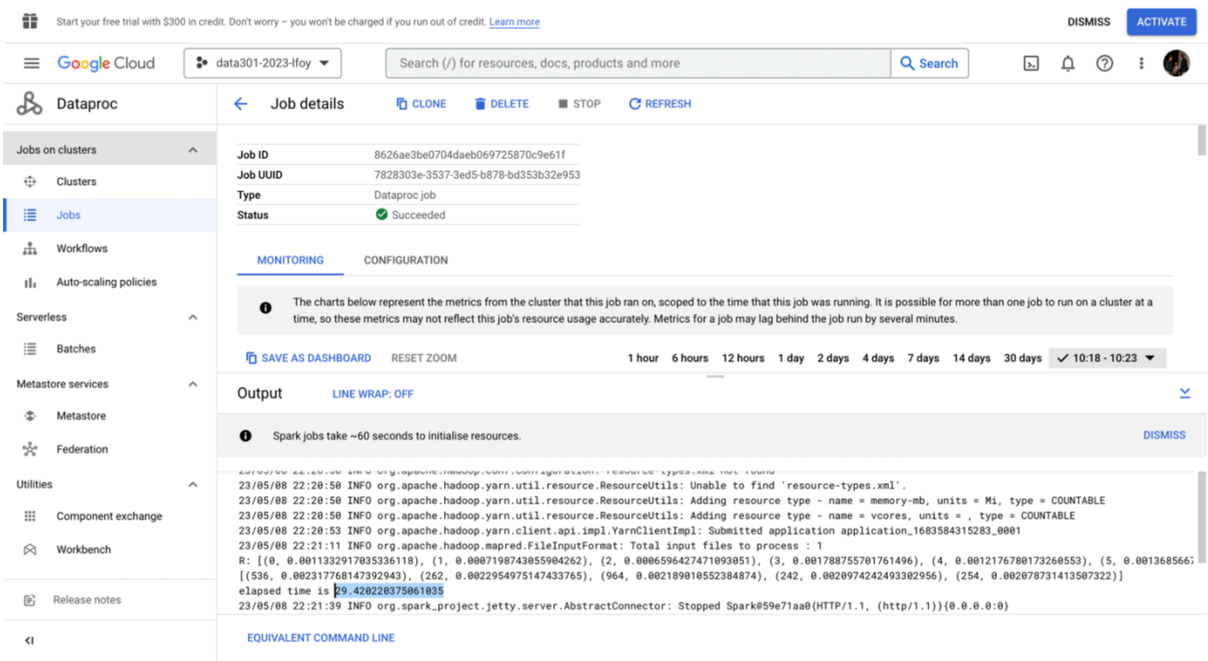
The charts below represent the metrics from the cluster that this job ran on, scoped to the time that this job was running. It is possible for more than one job to run on a cluster at a time, so these metrics may not reflect this job's resource usage accurately. Metrics for a job may lag behind the job run by several minutes.

Spark jobs take ~60 seconds to initialise resources.

```
23/05/08 22:12:33 INFO org.apache.hadoop.mapred.FileInputFormat: Total input files to process : 1
R: [(0, 0.03540059180244296), (1, 0.005171063533189883), (2, 0.007109590714645791), (3, 0.006994840443364616), (4, 0.016446111939280406), (5, 0.00549684498174
[(52, 0.04243140406575616), (39, 0.039884267610139156), (13, 0.039377727003573254), (0, 0.03540059180244296), (26, 0.03489656776403212)]
elapsed time is 30.09452199935913
23/05/08 22:13:09 INFO org.spark_project.jetty.server.AbstractConnector: Stopped Spark@4fa759c9{HTTP/1.1, (http/1.1)}{0.0.0.0:0}
```

Output is complete

P=4 Screenshot:



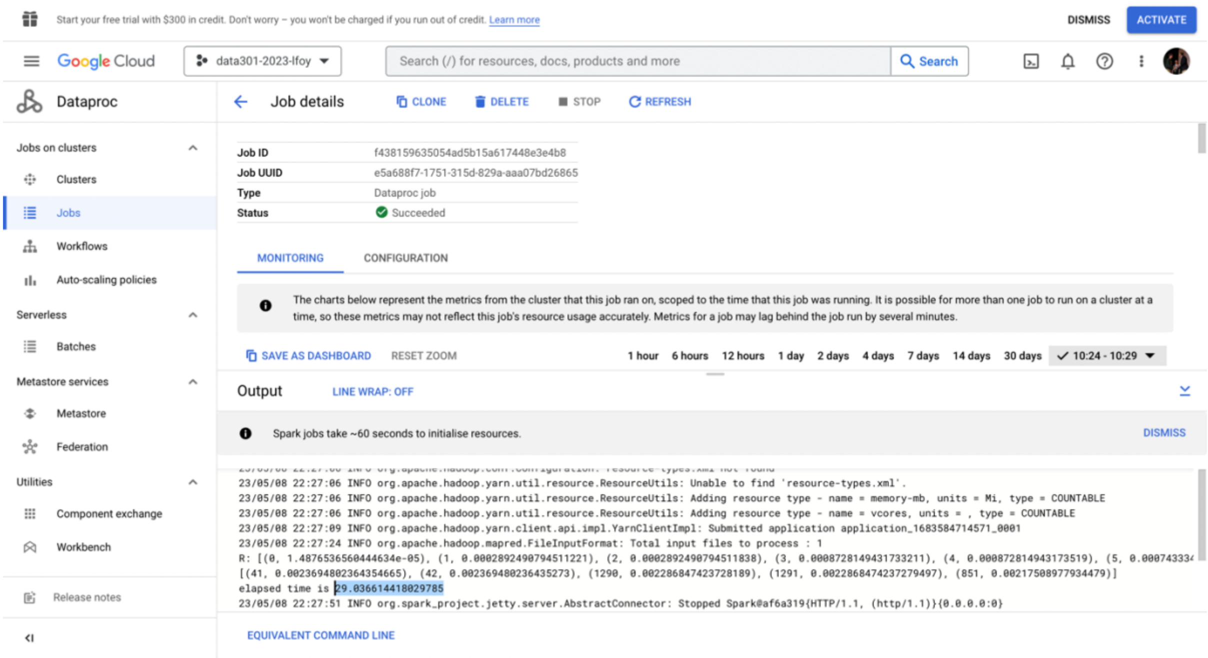
Job ID	8626ae3be0704daeb069725870c9e61f
Job UUID	7828303e-3537-3ed5-b878-bd353b32e953
Type	Dataproc job
Status	✓ Succeeded

The charts below represent the metrics from the cluster that this job ran on, scoped to the time that this job was running. It is possible for more than one job to run on a cluster at a time, so these metrics may not reflect this job's resource usage accurately. Metrics for a job may lag behind the job run by several minutes.

Spark jobs take ~60 seconds to initialise resources.

```
23/05/08 22:20:50 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils: Unable to find 'resource-types.xml'.
23/05/08 22:20:50 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils: Adding resource type - name = memory-mb, units = Mi, type = COUNTABLE
23/05/08 22:20:50 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils: Adding resource type - name = vcores, units = , type = COUNTABLE
23/05/08 22:20:53 INFO org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: Submitted application application_1683584315283_0001
23/05/08 22:21:11 INFO org.apache.hadoop.mapred.FileInputFormat: Total input files to process : 1
R: [(0, 0.0011332917035336118), (1, 0.0007198743055904262), (2, 0.0006596427471093051), (3, 0.001788755701761496), (4, 0.0012176780173260553), (5, 0.0013685667
[(536, 0.002317768147392943), (262, 0.0022954975147433765), (964, 0.002189010552384874), (242, 0.0020974242493302956), (254, 0.002078731413507322)]
elapsed time is 29.420220375061035
23/05/08 22:21:39 INFO org.spark_project.jetty.server.AbstractConnector: Stopped Spark@59e71aa0{HTTP/1.1, (http/1.1)}{0.0.0.0:0}
```

P=8 Screenshot:



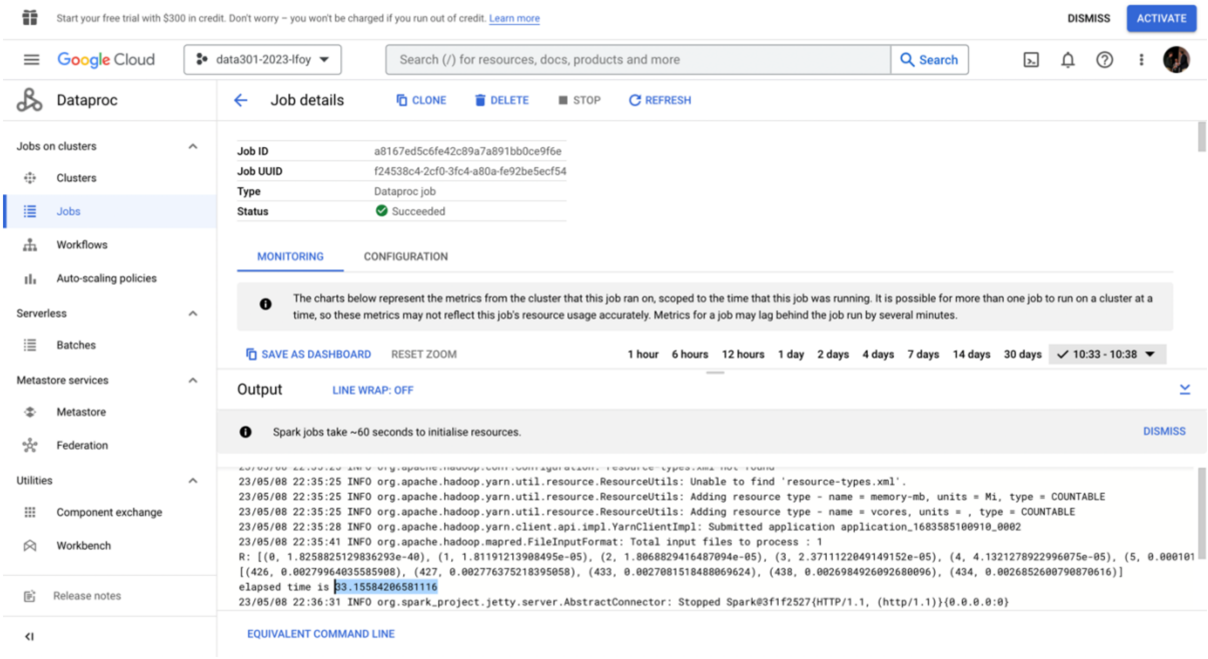
Job ID	f438159635054ad5b15a617448e3e4b8
Job UUID	e5a688f7-1751-315d-829a-aaa07bd26865
Type	Dataproc job
Status	✓ Succeeded

The charts below represent the metrics from the cluster that this job ran on, scoped to the time that this job was running. It is possible for more than one job to run on a cluster at a time, so these metrics may not reflect this job's resource usage accurately. Metrics for a job may lag behind the job run by several minutes.

Spark jobs take ~60 seconds to initialise resources.

```
23/05/08 22:27:06 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils: Unable to find 'resource-types.xml'.
23/05/08 22:27:06 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils: Adding resource type - name = memory-mb, units = Mi, type = COUNTABLE
23/05/08 22:27:06 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils: Adding resource type - name = vcores, units = , type = COUNTABLE
23/05/08 22:27:09 INFO org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: Submitted application application_1683584714571_0001
23/05/08 22:27:24 INFO org.apache.hadoop.mapred.FileInputFormat: Total input files to process : 1
R: [(0, 1.4876536560444634e-05), (1, 0.0002892490794511221), (2, 0.0002892490794511838), (3, 0.0008728149431733211), (4, 0.000872814943173519), (5, 0.00074333
[(41, 0.0023694802364354665), (42, 0.002369480236435273), (1290, 0.002286847423728189), (1291, 0.0022868474237279497), (851, 0.00217508977934479)]
elapsed time is 29.036614418029785
23/05/08 22:27:51 INFO org.spark_project.jetty.server.AbstractConnector: Stopped Spark@af6a319{HTTP/1.1, (http/1.1)}{0.0.0.0:0}
```

P=16 Screenshot:



Job ID	a8167ed5c6fe42c89a7a891bb0ce9f6e
Job UUID	f24538c4-2cf0-3fc4-a80a-fe92be5ecf54
Type	Dataproc job
Status	Succeeded

The charts below represent the metrics from the cluster that this job ran on, scoped to the time that this job was running. It is possible for more than one job to run on a cluster at a time, so these metrics may not reflect this job's resource usage accurately. Metrics for a job may lag behind the job run by several minutes.

Spark jobs take ~60 seconds to initialise resources.

```
23/05/08 22:35:25 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils: Unable to find 'resource-types.xml'.
23/05/08 22:35:25 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils: Adding resource type - name = memory-mb, units = Mi, type = COUNTABLE
23/05/08 22:35:25 INFO org.apache.hadoop.yarn.util.resource.ResourceUtils: Adding resource type - name = vcores, units = , type = COUNTABLE
23/05/08 22:35:28 INFO org.apache.hadoop.yarn.client.api.impl.YarnClientImpl: Submitted application application_1683585100910_0002
23/05/08 22:35:41 INFO org.apache.hadoop.mapred.FileInputFormat: Total input files to process : 1
R: [(0, 1.8258825129836293e-40), (1, 1.81191213908495e-05), (2, 1.8068829416487094e-05), (3, 2.3711122049149152e-05), (4, 4.1321278922996075e-05), (5, 0.000101
[(426, 0.00279964035585908), (427, 0.002776375218395058), (433, 0.0027081518488069624), (438, 0.0026984926092680096), (434, 0.0026852600790870616)]
elapsed time is 33.15584206581116
23/05/08 22:36:31 INFO org.spark_project.jetty.server.AbstractConnector: Stopped Spark@3f1f2527{HTTP/1.1, (http/1.1)}{0.0.0.0:0}
```

(b).

Looking at the data, we can see the time taken to complete PageRank on proportionally increasing file sizes stayed roughly constant when moving from 1 through to 16 cores – with the most significant increase being between 8 and 16 cores (29.0 to 33.1). Gustafson's Law, in the context of this lab, states that as the number of CPU cores increases – we can increase our file size proportionally and keep computation time somewhat constant. The trend present in my data is consistent with Gustafson's law, as a proportional increase in problem size and processing power gave us roughly constant results for time.

The reason that the timings were not more constant is likely due to the inconsistencies that come with working with PySpark and parallel systems in general. For example, hardware issues and communication overhead can make slight differences between tests.