Jiahui Tang

System Specs:

- Kernal:

```
[hadoop@ip-172-31-6-121 ~]$ uname -a
Linux ip-172-31-6-121 4.14.154-99.181.amzn1.x86_64 #1 SMP Sat Nov 16 01:38:34 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
[hadoop@ip-172-31-6-121 ~]$
```

- Infra:
  - Local Mode: EC2 instance (m4.xlarge) with Ubuntu 18.04.
  - Cluster: AWS EMR Spark with 2/4 worker instances on m4.xlarge
- Network of Cluster:

```
[hadoop@ip-172-31-6-121 ~]$ ping 172.31.7.225
PING 172.31.7.225 (172.31.7.225) 56(84) bytes of data.
64 bytes from 172.31.7.225: icmp_seq=1 ttl=255 time=0.116 ms
64 bytes from 172.31.7.225: icmp_seq=2 ttl=255 time=0.132 ms
64 bytes from 172.31.7.225: icmp_seq=3 ttl=255 time=0.112 ms
64 bytes from 172.31.7.225: icmp_seq=4 ttl=255 time=0.122 ms
64 bytes from 172.31.7.225: icmp_seq=5 ttl=255 time=0.133 ms
64 bytes from 172.31.7.225: icmp_seq=6 ttl=255 time=0.129 ms
64 bytes from 172.31.7.225: icmp_seq=7 ttl=255 time=0.139 ms
64 bytes from 172.31.7.225: icmp_seq=8 ttl=255 time=0.131 ms
64 bytes from 172.31.7.225: icmp_seq=9 ttl=255 time=0.130 ms
64 bytes from 172.31.7.225: icmp_seq=10 ttl=255 time=0.144 ms
64 bytes from 172.31.7.225: icmp_seq=11 ttl=255 time=0.120 ms
64 bytes from 172.31.7.225: icmp_seq=12 ttl=255 time=0.114 ms
64 bytes from 172.31.7.225: icmp_seq=13 ttl=255 time=0.134 ms
64 bytes from 172.31.7.225: icmp_seq=14 ttl=255 time=0.118 ms
64 bytes from 172.31.7.225: icmp_seq=15 ttl=255 time=0.136 ms
64 bytes from 172.31.7.225: icmp_seq=16 ttl=255 time=0.117 ms
64 bytes from 172.31.7.225: icmp_seq=17 ttl=255 time=0.136 ms
64 bytes from 172.31.7.225: icmp_seq=18 ttl=255 time=0.131 ms
64 bytes from 172.31.7.225: icmp_seq=19 ttl=255 time=0.161 ms
64 bytes from 172.31.7.225: icmp_seq=20 ttl=255 time=0.141 ms
64 bytes from 172.31.7.225: icmp_seq=21 ttl=255 time=0.140 ms
64 bytes from 172.31.7.225: icmp_seq=22 ttl=255 time=0.134 ms
64 bytes from 172.31.7.225: icmp_seq=23 ttl=255 time=0.151 ms
64 bytes from 172.31.7.225: icmp_seq=24 ttl=255 time=0.122 ms
64 bytes from 172.31.7.225: icmp_seq=25 ttl=255 time=0.129 ms
64 bytes from 172.31.7.225: icmp_seq=26 ttl=255 time=0.124 ms
64 bytes from 172.31.7.225: icmp_seq=27 ttl=255 time=0.129 ms
^C
--- 172.31.7.225 ping statistics ---
27 packets transmitted, 27 received, 0% packet loss, time 26611ms
```

- Bandwidth: Amazon says it's 5 Gbps
- Spark Version:

```
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
21/04/16 09:45:17 WARN Client: Neither spark.yarn.jars nor spark.yarn.archive is set, falling back to uploading libraries under SPARK_HOME.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.4.4
      /_/

Using Python version 2.7.16 (default, Oct 14 2019 21:26:56)
SparkSession available as 'spark'.
```

- Java Version:

```
[hadoop@ip-172-31-6-121 ~]$ java -version
openjdk version "1.8.0_282"
OpenJDK Runtime Environment (build 1.8.0_282-b08)
OpenJDK 64-Bit Server VM (build 25.282-b08, mixed mode)
[hadoop@ip-172-31-6-121 ~]$
```

- CPU info:

```
[hadoop@ip-172-31-6-121 ~]$ cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 79
model name      : Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz
stepping        : 1
microcode       : 0xb000038
cpu MHz         : 2299.895
cache size      : 46080 KB
physical id     : 0
siblings        : 4
core id         : 0
cpu cores       : 2
apicid          : 0
initial apicid  : 0
fpu             : yes
fpu_exception   : yes
cpuid level     : 13
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge m
 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand hyperv
bugs            : cpu_meltdown spectre_v1 spectre_v2 spec_store_bypass l
bogomips        : 4600.11
clflush size    : 64
cache_alignment : 64
address sizes   : 46 bits physical, 48 bits virtual
power management:
```

```
[hadoop@ip-172-31-6-121 ~]$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                4
On-line CPU(s) list:   0-3
Thread(s) per core:    2
Core(s) per socket:    2
Socket(s):             1
NUMA node(s):          1
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 79
Model name:            Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz
Stepping:              1
CPU MHz:               2299.895
BogoMIPS:              4600.11
Hypervisor vendor:     Xen
Virtualization type:   full
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              46080K
NUMA node0 CPU(s):     0-3
Flags:                 fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca
se4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand hyperviso
```

2.6. Parallel Execution (25 points)

Use the Distributed Grep Spark script developed in exercise 2.1 and the large version of the movielens data set `ml-latest.zip` (224MB) to show the ratings with 5.0 stars.

- Evaluate performance and speed-up when using 1, 2 and 4 cores in the local mode installation of Spark on a m4.xlarge instance.
- Evaluate performance and speed-up when using a cluster with 2 and 4 m4.xlarge worker instances and 1 and 2 cores per node. You can start with a cluster of 2 worker nodes and then dynamically resize it to include more worker nodes (Hardware tab).

Submission

- `P26.pdf`: Description of the experiment, discussion about the performance and speed-up, total running time, and description of any tuning developed

Part 1: Local Mode

Result

Workder Instances/Core Nodes	Log Time (sec)	Speedup
1 core	51.7817	1
2 cores	32.0380	1.6163
4 cores	31.508265	1.6434

Discussion:

Similarly to question **1.6**, `grep` is a command-line utility for searching plain-text data sets for lines that match a regular expression. Its name comes from the ed command g/re/p (globally search for a regular expression and print matching lines), which has the same effect.

From the result table, we could see that after we increase the core node in local mode installation of Spark on m4.xlarge instance, the log time didn't decrease a lot, and speedup also only increase from `1` to `1.61` and `1.64`, for node increases from `1` to `2` and `4` respectively.

It is mainly because of `grep` has a I/O bound, rather than CPU bound (In CPU we only check for `5.0 in line`).

As the `EXECUTION_TIME = CPU_TIME + I/O_TIME + SYSTEM_TIME`, when we increase the node, we could decrease `CPU_TIME` as it is being parallelized. However, as we are doing grep and search for the same pattern, the read and write data part takes a large amount of total execution time, which also keeps unchanged even when the nodes are increased. Thus, as `I/O_TIME` keeps unchanged, while provisioning and shuffle data into more nodes may take more time in Hadoop job, `SYSTEM_TIME` may also be increase, which shows overhead overall is proportional to the number of workers. The total speedup would thus be way less than theortical speed up when number of nodes increases.

We could infer that when increase from 1 core to 2 cores, although it is I/O bounded, it is likely that I/O bound is not fully utilised, thus we see considerable improvement of speedup from 1 to 1.6. When increase from 2 to 4 cores, the speedup is maintained at around 1.6, and not improved much, which means I/O bandwidth is fully utilised.

Part 2: Cluster

Result

Worker Instances/Core Nodes	Execution Time (sec)	Speedup
2 Worker Instances, 1 core per node	57.466414	1
2 Worker Instances, 2 cores per node	44.580397	1.2891
4 Worker Instances, 1 core per node	41.339823	1.3901
4 Worker Instances, 2 core per node	26.663581	2.1552

Discussion:

On top of what I said in last question, similarly, `grep` is bounded by I/O time but not CPU time in its total execution time. The performance bottleneck is on input and output, reading and writing. Though I/O could also be parallelized, it has its hard limit on I/O bandwidth and could never reach theortical speedup.

From the performance and speedup, we could observe that 4 workers are overall better than 2 worker instance, while both increasing cores per nodes and number of instances didn't make the speedup reach theortical speedup.

As data are in S3, the I/O happens between instances and S3, instead of between instances themselves. So theortically, when we increase number of worker instances to 2 times, the total I/O bandwidth increases proportionally at 2 times.

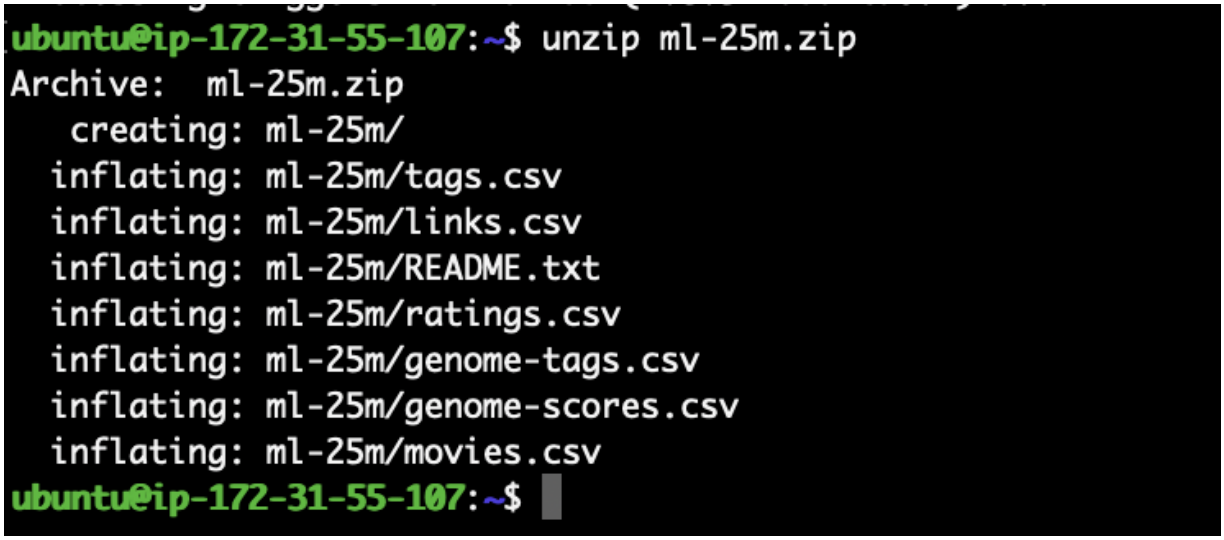
From observation, we could infer that in one worker instance, the I/O bandwidth can be fully utilised by 2 cores but it is not enough. It means I/O bandwidth/ability is roughly between 1 to 2 cores. Thus, from the table above we could infer that:

- (1). 2 instance and 1 core per node, I/O not fully utilised.
- (2). 2 instance and 2 cores per node, I/O fully utilised, it is bounded by I/O bandwidth.
- (3). 4 instances and 1 core per node, I/O not fully utilised; Thus it is still not yet bounded by I/O bandwidth, so its speedup is more than what we see in (2)
- (4). 4 instances but with I/O fully utilised, so it reaches the maximum speedup among 4 cases. However, it is still way less than theortical speedup as there's overhead proportional to number of worker instances, and I/O time is bounded by its bandwidth.

Part 3: Description of Experiment on Local Mode

Description of Experiment:

1. Spin up Ubuntu 18.04 EC2 Instance with m4.xlarge and follow guide in [Lab 9 (https://harvard-iacs.github.io/2021-CS205/labs/l9/l9.pdf)](https://harvard-iacs.github.io/2021-CS205/labs/l9/l9.pdf)
2. Install Scala, Java, Python and Spark
3. Configure Environment by updating `.profile` and `/etc/hosts`
4. Install `unzip` by running `sudo apt install unzip`
5. Upload Python file `P26_spark.py` to ubuntu and the large version of MovieLens dataset and unzip it.



6. Copy out the `ratings.csv` file to current directory by `cp ml-25m/ratings.csv ./`
7. Submit a spark job with 1 core in local mode, checking for execution time for 1 core is `51.7817 sec`

Using cmd `spark-submit P26_spark.py`

```
21/04/15 19:11:57 INFO SparkHadoopMapRedUtil: attempt_20210415191105_0000_m_000020_20: Committed
21/04/15 19:11:57 INFO Executor: Finished task 20.0 in stage 0.0 (TID 20). 1466 bytes result sent to driver
21/04/15 19:11:57 INFO TaskSetManager: Finished task 20.0 in stage 0.0 (TID 20) in 533 ms on localhost (executor driver) (21/21)
21/04/15 19:11:57 INFO TaskSchedulerImpl: Removed TaskSet 0.0, whose tasks have all completed, from pool
21/04/15 19:11:57 INFO DAGScheduler: ResultStage 0 (saveAsTextFile at NativeMethodAccessorImpl.java:0) finished in 51.650 s
21/04/15 19:11:57 INFO DAGScheduler: Job 0 finished: saveAsTextFile at NativeMethodAccessorImpl.java:0, took 51.781778 s
21/04/15 19:11:57 INFO SparkContext: Invoking stop() from shutdown hook
21/04/15 19:11:57 INFO SparkUI: Stopped Spark web UI at http://172.31.55.107:4040
21/04/15 19:11:57 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
21/04/15 19:11:57 INFO MemoryStore: MemoryStore cleared
21/04/15 19:11:57 INFO BlockManager: BlockManager stopped
21/04/15 19:11:57 INFO BlockManagerMaster: BlockManagerMaster stopped
21/04/15 19:11:57 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
21/04/15 19:11:57 INFO SparkContext: Successfully stopped SparkContext
21/04/15 19:11:57 INFO ShutdownHookManager: Shutdown hook called
21/04/15 19:11:57 INFO ShutdownHookManager: Deleting directory /tmp/spark-6e6ea364-f7d2-4cb7-be9c-7d96d1168b4f
21/04/15 19:11:57 INFO ShutdownHookManager: Deleting directory /tmp/spark-6e6ea364-f7d2-4cb7-be9c-7d96d1168b4f/pyspark-8e84818b-ce1d-431b-b583-f85d72d59e39
```

8. Check output is correct by `cat output_q26.txt/*`

```
ubuntu@ip-172-31-55-107:~$ cat output_q26.txt/*
1,296,5.0,1147880044
1,307,5.0,1147868828
1,665,5.0,1147878820
1,1237,5.0,1147868839
1,2632,5.0,1147878248
1,2692,5.0,1147869100
1,3569,5.0,1147879603
1,3949,5.0,1147868678
1,4144,5.0,1147868898
1,4325,5.0,1147878122
1,5767,5.0,1147878729
1,6016,5.0,1147869090
1,6711,5.0,1147868622
1,7361,5.0,1147880055
1,8154,5.0,1147868865
1,8327,5.0,1147879375
1,32591,5.0,1147879538
```

9.Make a copy of original file and update it with 2 cores

by changing `setMaster(local[2])` : Run Spark locally with 2 worker threads (ideally, set this to the number of cores on your machines).

```
from pyspark import SparkConf, SparkContext
import sys
import re



conf = SparkConf().setMaster('local[2]').setAppName('movielens')
sc = SparkContext(conf = conf)

text_file = sc.textFile("ratings.csv")

text_file.filter(lambda line: "5.0" in line).saveAsTextFile("output_q26_node2.txt")
~
~
~
```

10. Resubmit spark job and check execution time for 2 cores is `32.0380` .

```
21/04/15 19:21:08 INFO FileOutputCommitter: Saved output of task 'attempt_20210415192036_0000_m_000020_20' to file:/home/ubuntu/output_q26_node2.txt/_temporary/0/task_20210
21/04/15 19:21:08 INFO SparkHadoopMapRedUtil: attempt_20210415192036_0000_m_000020_20: Committed
21/04/15 19:21:08 INFO Executor: Finished task 20.0 in stage 0.0 (TID 20). 1423 bytes result sent to driver
21/04/15 19:21:08 INFO TaskSetManager: Finished task 20.0 in stage 0.0 (TID 20) in 612 ms on localhost (executor driver) (21/21)
21/04/15 19:21:08 INFO TaskSchedulerImpl: Removed TaskSet 0.0, whose tasks have all completed, from pool
21/04/15 19:21:08 INFO DAGScheduler: ResultStage 0 (saveAsTextFile at NativeMethodAccessorImpl.java:0) finished in 31.904 s
21/04/15 19:21:08 INFO DAGScheduler: Job 0 finished: saveAsTextFile at NativeMethodAccessorImpl.java:0, took 32.037974 s
21/04/15 19:21:09 INFO SparkContext: Invoking stop() from shutdown hook
21/04/15 19:21:09 INFO SparkUI: Stopped Spark web UI at http://172.31.55.107:4040
21/04/15 19:21:09 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
21/04/15 19:21:09 INFO MemoryStore: MemoryStore cleared
21/04/15 19:21:09 INFO BlockManager: BlockManager stopped
21/04/15 19:21:09 INFO BlockManagerMaster: BlockManagerMaster stopped
21/04/15 19:21:09 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
21/04/15 19:21:09 INFO SparkContext: Successfully stopped SparkContext
21/04/15 19:21:09 INFO ShutdownHookManager: Shutdown hook called
21/04/15 19:21:09 INFO ShutdownHookManager: Deleting directory /tmp/spark-d07ad428-c599-46b6-8aaf-8d1743a17e7e/pyspark-74f3445c-1c3b-4b03-bc19-8ea58da24e8a
21/04/15 19:21:09 INFO ShutdownHookManager: Deleting directory /tmp/spark-d07ad428-c599-46b6-8aaf-8d1743a17e7e
ubuntu@ip-172-31-55-107:~$
```

11. Simlarly, repeat step 9-10 for 4 cores, and get execution time for 4 cores is `31.508265`.

```
21/04/15 19:25:19 INFO SparkHadoopMapRedUtil: attempt_20210415192447_0000_m_000020_20: Committed
21/04/15 19:25:19 INFO Executor: Finished task 20.0 in stage 0.0 (TID 20). 1423 bytes result sent to driver
21/04/15 19:25:19 INFO TaskSetManager: Finished task 20.0 in stage 0.0 (TID 20) in 835 ms on localhost (executor driver) (21/21)
21/04/15 19:25:19 INFO TaskSchedulerImpl: Removed TaskSet 0.0, whose tasks have all completed, from pool
21/04/15 19:25:19 INFO DAGScheduler: ResultStage 0 (saveAsTextFile at NativeMethodAccessorImpl.java:0) finished in 31.375 s
21/04/15 19:25:19 INFO DAGScheduler: Job 0 finished: saveAsTextFile at NativeMethodAccessorImpl.java:0, took 31.508265 s
21/04/15 19:25:19 INFO SparkContext: Invoking stop() from shutdown hook
21/04/15 19:25:19 INFO SparkUI: Stopped Spark web UI at http://172.31.55.107:4040
21/04/15 19:25:19 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
21/04/15 19:25:19 INFO MemoryStore: MemoryStore cleared
21/04/15 19:25:19 INFO BlockManager: BlockManager stopped
21/04/15 19:25:19 INFO BlockManagerMaster: BlockManagerMaster stopped
21/04/15 19:25:19 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
21/04/15 19:25:19 INFO SparkContext: Successfully stopped SparkContext
21/04/15 19:25:19 INFO ShutdownHookManager: Shutdown hook called
21/04/15 19:25:19 INFO ShutdownHookManager: Deleting directory /tmp/spark-491ccce6-64fb-435f-ac0a-a33db316a659
21/04/15 19:25:19 INFO ShutdownHookManager: Deleting directory /tmp/spark-491ccce6-64fb-435f-ac0a-a33db316a659/pyspark-ac66ee8c-6075-4844-94b9-dbf74e93deb8
```

Part 4: Description of Experiment on Cluster

Description of Experiment:

1. Spin up Hadoop EMR cluster with m4.xlarge and follow guide in [Lab 10 (https://harvard-iacs.github.io/2021-CS205/labs/l10/l10.pdf)](https://harvard-iacs.github.io/2021-CS205/labs/l10/l10.pdf)
2. Upload Python file `P26_spark.py` to ubuntu and the large version of MovieLens dataset and unzip it.
3. Copy out the `ratings.csv` file to current directory by `cp ml-25m/ratings.csv ./`
4. Upload the ratings.csv file to the Hadoop file system

```
hadoop fs -put ratings.csv
```

```
[hadoop@ip-172-31-6-121 ~]$ hadoop fs -put ratings.csv
[hadoop@ip-172-31-6-121 ~]$ hadoop fs -ls
Found 1 items
-rw-r--r--   1 hadoop hadoop  678260987 2021-04-16 08:39 ratings.csv
[hadoop@ip-172-31-6-121 ~]$
```

5. Submit a spark job with 2 worker instance and 1 core per node in local mode, by specifying `--executor-cores` and `--num-executors` in command line, also remove the setMaster property with local in the SparkConf configuration.

It takes `57.466414` execution time.

cmd: `spark-submit --num-executors 2 --executor-cores 1 P26_spark.py`

```
from pyspark import SparkConf, SparkContext
import sys

conf = SparkConf().setAppName('grep')
sc = SparkContext(conf = conf)

text_file = sc.textFile("ratings.csv")

text_file.filter(lambda line: "5.0" in line).saveAsTextFile("output_q26.txt")
~
```

```
21/04/16 08:57:09 INFO TaskSetManager: Finished task 1.0 in stage 0.0 (TID 0) in 20757 ms on ip-172-31-14-137.ec2.internal (executor 2) (1/5)
21/04/16 08:57:09 INFO PythonAccumulatorV2: Connected to AccumulatorServer at host: 127.0.0.1 port: 60343
21/04/16 08:57:10 INFO TaskSetManager: Starting task 3.0 in stage 0.0 (TID 3, ip-172-31-7-225.ec2.internal, executor 1, partition 3, RACK_LOCAL, 7935 bytes)
21/04/16 08:57:10 INFO TaskSetManager: Finished task 0.0 in stage 0.0 (TID 1) in 20444 ms on ip-172-31-7-225.ec2.internal (executor 1) (2/5)
21/04/16 08:57:26 INFO TaskSetManager: Starting task 4.0 in stage 0.0 (TID 4, ip-172-31-14-137.ec2.internal, executor 2, partition 4, RACK_LOCAL, 7935 bytes)
21/04/16 08:57:26 INFO TaskSetManager: Finished task 2.0 in stage 0.0 (TID 2) in 16940 ms on ip-172-31-14-137.ec2.internal (executor 2) (3/5)
21/04/16 08:57:27 INFO TaskSetManager: Finished task 3.0 in stage 0.0 (TID 3) in 16396 ms on ip-172-31-7-225.ec2.internal (executor 1) (4/5)
21/04/16 08:57:43 INFO TaskSetManager: Finished task 4.0 in stage 0.0 (TID 4) in 16917 ms on ip-172-31-14-137.ec2.internal (executor 2) (5/5)
21/04/16 08:57:43 INFO YarnScheduler: Removed TaskSet 0.0, whose tasks have all completed, from pool
21/04/16 08:57:43 INFO DAGScheduler: ResultStage 0 (runJob at SparkHadoopWriter.scala:78) finished in 57.353 s
21/04/16 08:57:43 INFO DAGScheduler: Job 0 finished: runJob at SparkHadoopWriter.scala:78, took 57.466414 s
21/04/16 08:57:43 INFO SparkHadoopWriter: Job job_20210416085645_0004 committed.
21/04/16 08:57:43 INFO SparkContext: Invoking stop() from shutdown hook
21/04/16 08:57:43 INFO SparkUI: Stopped Spark web UI at http://ip-172-31-6-121.ec2.internal:4040
21/04/16 08:57:43 INFO YarnClientSchedulerBackend: Interrupting monitor thread
21/04/16 08:57:43 INFO YarnClientSchedulerBackend: Shutting down all executors
21/04/16 08:57:43 INFO YarnSchedulerBackend$YarnDriverEndpoint: Asking each executor to shut down
21/04/16 08:57:43 INFO SchedulerExtensionServices: Stopping SchedulerExtensionServices
```

6. Download the output file from hadoop file system to the local file system and check the content is correct

```
[[hadoop@ip-172-31-6-121 ~]$ hadoop fs -ls
Found 3 items
drwxr-xr-x   - hadoop hadoop          0 2021-04-16 08:57 .sparkStaging
drwxr-xr-x   - hadoop hadoop          0 2021-04-16 08:57 output_q26.txt
-rw-r--r--   1 hadoop hadoop  678260987 2021-04-16 08:39 ratings.csv
[[hadoop@ip-172-31-6-121 ~]$ hadoop fs -get output_q26.txt
[[hadoop@ip-172-31-6-121 ~]$ cat output_q26.txt/*
1,296,5.0,1147880044
1,307,5.0,1147868828
1,665,5.0,1147878820
1,1237,5.0,1147868839
1,2632,5.0,1147878248
1,2692,5.0,1147869100
1,3569,5.0,1147879603
1,3949,5.0,1147868678
1,4144,5.0,1147868898
1,4325,5.0,1147878122
1,5767,5.0,1147878729
1,6016,5.0,1147869090
1,6711,5.0,1147868622
1,7361,5.0,1147880055
1,8154,5.0,1147868865
1,8327,5.0,1147879375
1,32591,5.0,1147879538
2,110,5.0,1141416589
2,260,5.0,1141417172
2,318,5.0,1141417181
2,333,5.0,1141415931
2,457,5.0,1141416618
2,527,5.0,1141415816
2,1136,5.0,1141417950
2,1196,5.0,1141417198
2,1197,5.0,1141415874
2,1210,5.0,1141416665
2,1257,5.0,1141417313
2,1287,5.0,1141417140
2,1291,5.0,1141416603
2,1293,5.0,1141417937
2,1356,5.0,1141417272
```

7. Remove existing output file

```
hadoop fs -rm -R -f output_q26.txt
```

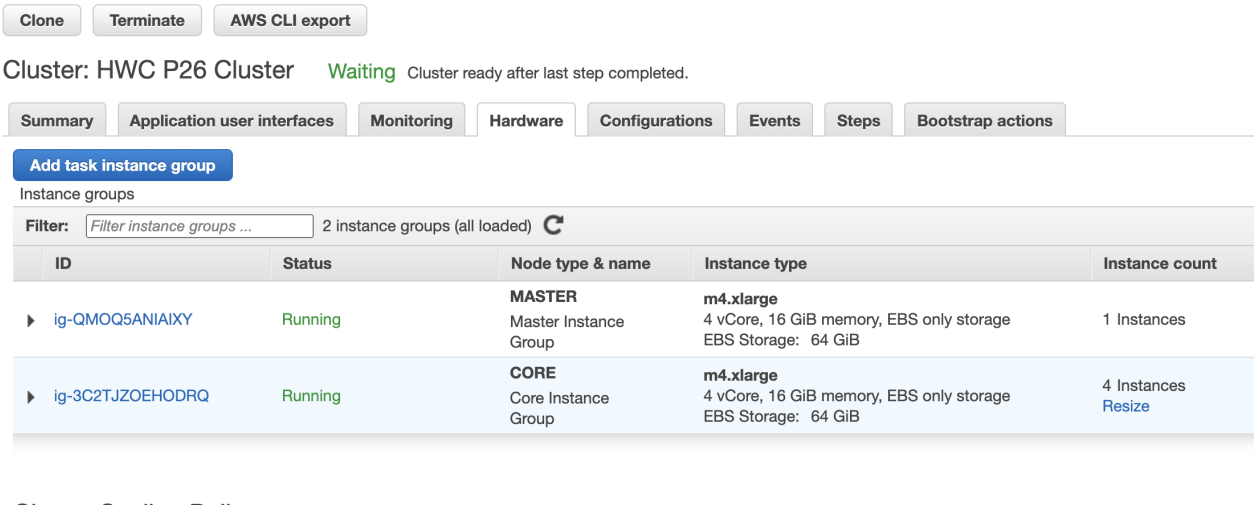
8. Similarly, submit a spark job with 2 worker instance and 2 core per node in local mode, by specifying `--executor-cores` and `--num-executors` in command line, also remove the setMaster property with local in the SparkConf configuration.

It takes `44.580397` seconds execution time.

cmd: `spark-submit --num-executors 2 --executor-cores 2 P26_spark.py`

```
21/04/16 09:05:43 INFO TaskSetManager: Finished task 2.0 in stage 0.0 (TID 2) in 24418 ms on ip-172-31-14-137.ec2.internal (executor 2) (4/5)
21/04/16 09:05:59 INFO TaskSetManager: Finished task 4.0 in stage 0.0 (TID 4) in 18573 ms on ip-172-31-7-225.ec2.internal (executor 1) (5/5)
21/04/16 09:05:59 INFO YarnScheduler: Removed TaskSet 0.0, whose tasks have all completed, from pool
21/04/16 09:05:59 INFO DAGScheduler: ResultStage 0 (runJob at SparkHadoopWriter.scala:78) finished in 44.480 s
21/04/16 09:05:59 INFO DAGScheduler: Job 0 finished: runJob at SparkHadoopWriter.scala:78, took 44.580397 s
21/04/16 09:05:59 INFO SparkHadoopWriter: Job job_20210416090514_0004 committed.
21/04/16 09:05:59 INFO SparkContext: Invoking stop() from shutdown hook
21/04/16 09:05:59 INFO SparkUI: Stopped Spark web UI at http://ip-172-31-6-121.ec2.internal:4040
21/04/16 09:05:59 INFO YarnClientSchedulerBackend: Interrupting monitor thread
21/04/16 09:05:59 INFO YarnClientSchedulerBackend: Shutting down all executors
21/04/16 09:05:59 INFO YarnSchedulerBackend$YarnDriverEndpoint: Asking each executor to shut down
21/04/16 09:05:59 INFO SchedulerExtensionServices: Stopping SchedulerExtensionServices
(serviceOption=None,
 services=List(),
 started=false)
21/04/16 09:05:59 INFO YarnClientSchedulerBackend: Stopped
21/04/16 09:06:00 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
```

8. Resize to `4` worker instance/core nodes and redo step 5-8.



9. Submit Spark job with 4 worker instances and 1 core per node.

```
spark-submit --num-executors 4 --executor-cores 1 P26_spark.py
```

```
21/04/16 09:16:16 INFO TaskSetManager: Starting task 4.0 in stage 0.0 (TID 4, ip-172-31-1-37.ec2.internal, executor 1, partition 4, RACK_LOCAL, 7935 bytes)
21/04/16 09:16:16 INFO TaskSetManager: Finished task 0.0 in stage 0.0 (TID 0) in 20831 ms on ip-172-31-1-37.ec2.internal (executor 1) (1/5)
21/04/16 09:16:16 INFO PythonAccumulatorV2: Connected to AccumulatorServer at host: 127.0.0.1 port: 45061
21/04/16 09:16:17 INFO TaskSetManager: Finished task 3.0 in stage 0.0 (TID 3) in 20075 ms on ip-172-31-6-102.ec2.internal (executor 4) (2/5)
21/04/16 09:16:17 INFO TaskSetManager: Finished task 1.0 in stage 0.0 (TID 1) in 20550 ms on ip-172-31-7-225.ec2.internal (executor 3) (3/5)
21/04/16 09:16:17 INFO TaskSetManager: Finished task 2.0 in stage 0.0 (TID 2) in 20953 ms on ip-172-31-14-137.ec2.internal (executor 2) (4/5)
21/04/16 09:16:33 INFO TaskSetManager: Finished task 4.0 in stage 0.0 (TID 4) in 17403 ms on ip-172-31-1-37.ec2.internal (executor 1) (5/5)
21/04/16 09:16:33 INFO YarnScheduler: Removed TaskSet 0.0, whose tasks have all completed, from pool
21/04/16 09:16:33 INFO DAGScheduler: ResultStage 0 (runJob at SparkHadoopWriter.scala:78) finished in 41.223 s
21/04/16 09:16:33 INFO DAGScheduler: Job 0 finished: runJob at SparkHadoopWriter.scala:78, took 41.339823 s
21/04/16 09:16:33 INFO SparkHadoopWriter: Job job_20210416091552_0004 committed.
21/04/16 09:16:33 INFO SparkContext: Invoking stop() from shutdown hook
21/04/16 09:16:34 INFO SparkUI: Stopped Spark web UI at http://ip-172-31-6-121.ec2.internal:4040
21/04/16 09:16:34 INFO YarnClientSchedulerBackend: Interrupting monitor thread
21/04/16 09:16:34 INFO YarnClientSchedulerBackend: Shutting down all executors
21/04/16 09:16:34 INFO YarnSchedulerBackend$YarnDriverEndpoint: Asking each executor to shut down
21/04/16 09:16:34 INFO SchedulerExtensionServices: Stopping SchedulerExtensionServices
(serviceOption=None,
 services=List(),
```

10. Submit Spark job with 4 worker instances and 2 core per node.

```
spark-submit --num-executors 4 --executor-cores 2 P26_spark.py
```

```
21/04/16 09:19:07 INFO PythonAccumulatorV2: Connected to AccumulatorServer at host: 127.0.0.1 port: 40949
21/04/16 09:19:08 INFO TaskSetManager: Finished task 3.0 in stage 0.0 (TID 3) in 21853 ms on ip-172-31-6-102.ec2.internal (executor 4) (2/5)
21/04/16 09:19:08 INFO TaskSetManager: Finished task 0.0 in stage 0.0 (TID 0) in 23152 ms on ip-172-31-7-225.ec2.internal (executor 3) (3/5)
21/04/16 09:19:09 INFO TaskSetManager: Finished task 1.0 in stage 0.0 (TID 1) in 23540 ms on ip-172-31-7-225.ec2.internal (executor 3) (4/5)
21/04/16 09:19:09 INFO TaskSetManager: Finished task 2.0 in stage 0.0 (TID 2) in 22556 ms on ip-172-31-6-102.ec2.internal (executor 4) (5/5)
21/04/16 09:19:09 INFO YarnScheduler: Removed TaskSet 0.0, whose tasks have all completed, from pool
21/04/16 09:19:09 INFO DAGScheduler: ResultStage 0 (runJob at SparkHadoopWriter.scala:78) finished in 26.465 s
21/04/16 09:19:09 INFO DAGScheduler: Job 0 finished: runJob at SparkHadoopWriter.scala:78, took 26.663581 s
21/04/16 09:19:09 INFO SparkHadoopWriter: Job job_20210416091842_0004 committed.
21/04/16 09:19:09 INFO SparkContext: Invoking stop() from shutdown hook
21/04/16 09:19:09 INFO SparkUI: Stopped Spark web UI at http://ip-172-31-6-121.ec2.internal:4040
21/04/16 09:19:09 INFO YarnClientSchedulerBackend: Interrupting monitor thread
21/04/16 09:19:09 INFO YarnClientSchedulerBackend: Shutting down all executors
21/04/16 09:19:09 INFO YarnSchedulerBackend$YarnDriverEndpoint: Asking each executor to shut down
21/04/16 09:19:09 INFO SchedulerExtensionServices: Stopping SchedulerExtensionServices
(serviceOption=None,
 services=List(),
 started=false)
```