PROGRAMMING ASSIGNMENT - 2

Under the guidance of Prof. Ioan Raicu

Abhay Nagaraj B R

Indranil Thakur

Problem statement:

Perform sorting of large and moderate datasets in different configurations and compare their performances.

Methodology:

Implement the sorting of data using Merge Sort which seemed much feasible for the current scenario.
Vary the AWS ec2 instance type and the number of instances used to form the cluster to record the speedup correspondingly.
Gensort is used to generate the data on the fly instead of wasting time and money on the transfer of dat to be sorted.
Valsort is used to validate the output of the sort algorithm used.

Runtime Environment settings needed are as follows:

- Virtual cluster setup with automation scripts: 10 points
- • Required: Shared memory terasort implementation/scripts: 30 points
- • Hadoop Terasort implementation/scripts: 10 points
- • Spark Terasort implementation/scripts: 10 points
- • MPI Terasort implementation/scripts: 10 points
- • Readme.txt: 5 points
- • Performance evaluation, data, explanations, etc: 15 points per system (for a maximum of 60 points)
- • Followed instructions on deliverables: 5 points

This programming assignment has following parts

- Shared Memory Sort
- Hadoop Pseudo Mode Sorting
- Hadoop 8 nodes cluster Sorting
- Spark Pseudo Mode Sorting
- Spark 8 node cluster Sorting

The assignment had 8 parts as explained below

Virtual cluster setup with automation scripts:

Pre requisites:

An AWS account to be created.
For easier handling of access keys and .pem files on windows PCs (which is what I have used) -
Install putty/puttyGen
Install WinSCP for copying and editing of files on the virtual nodes with a free flowing UI.
This section is common for all the other configurations where the AWS cluster has to be set up either **pseudo** or **multi-node.**

Steps:

- Launch an ec2 instance.

- Create and download the keypair.pem file on your local machine.
- Use puttyGen to generate a private key of .ppk type.
- Use the above generated .ppk file in WinSCP to connect to the instances

Now we need to set up a password-less authentication for which we have created a config file which looks like the following.

It contains the host alias name and its public DNS address.
The username to be used.
The location of the .pem file on the virtual node.

```
Host namenode
  HostName ec2-54-152-0-85.compute-1.amazonaws.com
  User ubuntu
  IdentityFile ~/.ssh/my-hadoop-key.pem
Host datanode1
  HostName ec2-34-201-116-73.compute-1.amazonaws.com
  User ubuntu
  IdentityFile ~/.ssh/my-hadoop-key.pem
Host datanode2
  HostName ec2-54-211-103-104.compute-1.amazonaws.com
  User ubuntu
  IdentityFile ~/.ssh/my-hadoop-key.pem
Host datanode3
  HostName ec2-52-23-195-185.compute-1.amazonaws.com
  User ubuntu
  IdentityFile ~/.ssh/my-hadoop-key.pem
```

These .pem files are copied to all the nodes in the cluster, so that the nodes can talk to each other through a password-less mode of authentication.

```
scp ~/.ssh/my-hadoop-key.pem ~/.ssh/config datanode1:~/.ssh
```

Then we use the ssh-keygen to generate a public key on the virtual nodes.

```
#generate key file on the one of the nodes
 ssh-keygen -f ~/.ssh/ssh_rsa -t rsa -P ""
 -f : file name which follows it
 -t : type of key which is rsa here
 -P : passphrase which is "" empty here
```

The content of this .pub file generated is copied into the authorized_keys file in all the nodes in the cluster.

Now that we get the access the cluster nodes, Java, Hadoop and Spark are installed.

Java version installed – 1.8

Hadoop version installed – 2.7.4

Setting the environment variables
.profile file in every node is modified as follows:

```
#set environment variables in all the nodes
export JAVA_HOME=/usr
export PATH=$PATH:$JAVA_HOME/bin
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export HADOOP_CONF_DIR=/usr/local/hadoop/etc/hadoop
```

The modified .profile is loaded to bring it into effect

```
#load variables
. ~/.profile
```

Hadoop Configuration is also changed on all the nodes.

```
#hadoop configuration on all nodes
#$HADOOP_CONF_DIR/hadoop-env.sh change JAVA_HOMEexport
JAVA_HOME=/usr
```

Virtual Cluster (1-node i3.large):

Shared memory

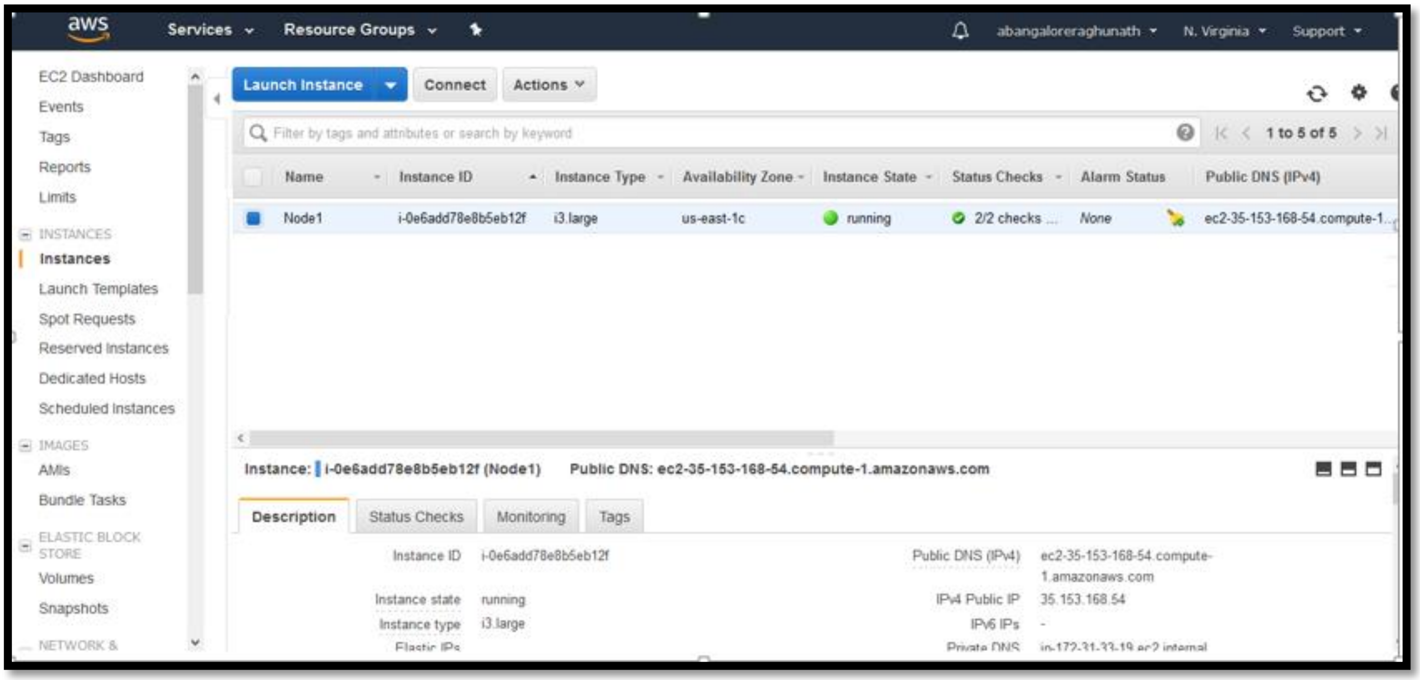
1-node 13.large instance

Java has been installed on the instance. On demand instance was used.

Instance Specifications

Instance ID i-0e6add78e8b5eb12f	Public DNS (IPv4) ec2-35-153-168-54.compute-1.amazonaws.com
Instance state Running	IPv4 Public IP 35.153.168.54
Instance type i3.large	
Private DNS ip-172-31-33-19.ec2.internal	
Availability zone us-east-1c	Private IPs 172.31.33.19

Case 1: Data to be sorted: 128 GB



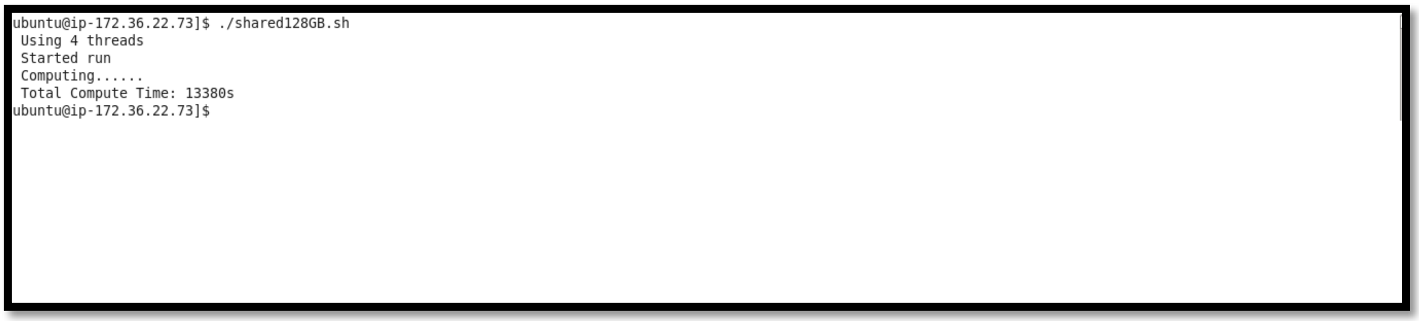
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
Node1	i-0e6add78e8b5eb12f	i3.large	us-east-1c	running	2/2 checks ...	None	ec2-35-153-168-54.compute-1...

Instance: i-0e6add78e8b5eb12f (Node1) Public DNS: ec2-35-153-168-54.compute-1.amazonaws.com

Instance ID	i-0e6add78e8b5eb12f	Public DNS (IPv4)	ec2-35-153-168-54.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	35.153.168.54
Instance type	i3.large	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-33-19.ec2.internal

Since the instance i3.xlarge itself has a memory of 16GB and an SSD of 440GB attached to it, we don't need a RAID0 to be configured for this instance.

Therefore the code is deployed on the node and is run to get following output. We have used 4 threads.



```
ubuntu@ip-172.36.22.73]$ ./shared128GB.sh
 Using 4 threads
 Started run
 Computing......
 Total Compute Time: 13380s
ubuntu@ip-172.36.22.73]$
```

It completed in 3 hours and 40 minutes

Hadoop

128GB

Hadoop 2.7.4 was the installed version.
Difficulty faced: When we tried installing Java 1.7 it did not work with this version of Hadoop and were stuck in this stage for a long time. Later on after a long search, we understood that this version of hadoop is not compatible with Java 1.7. It needs java 1.8 or newer.

Furthermore, Hadoop MapReduce job needs a few configurations to be made before hand running the job.

The following is the config file.

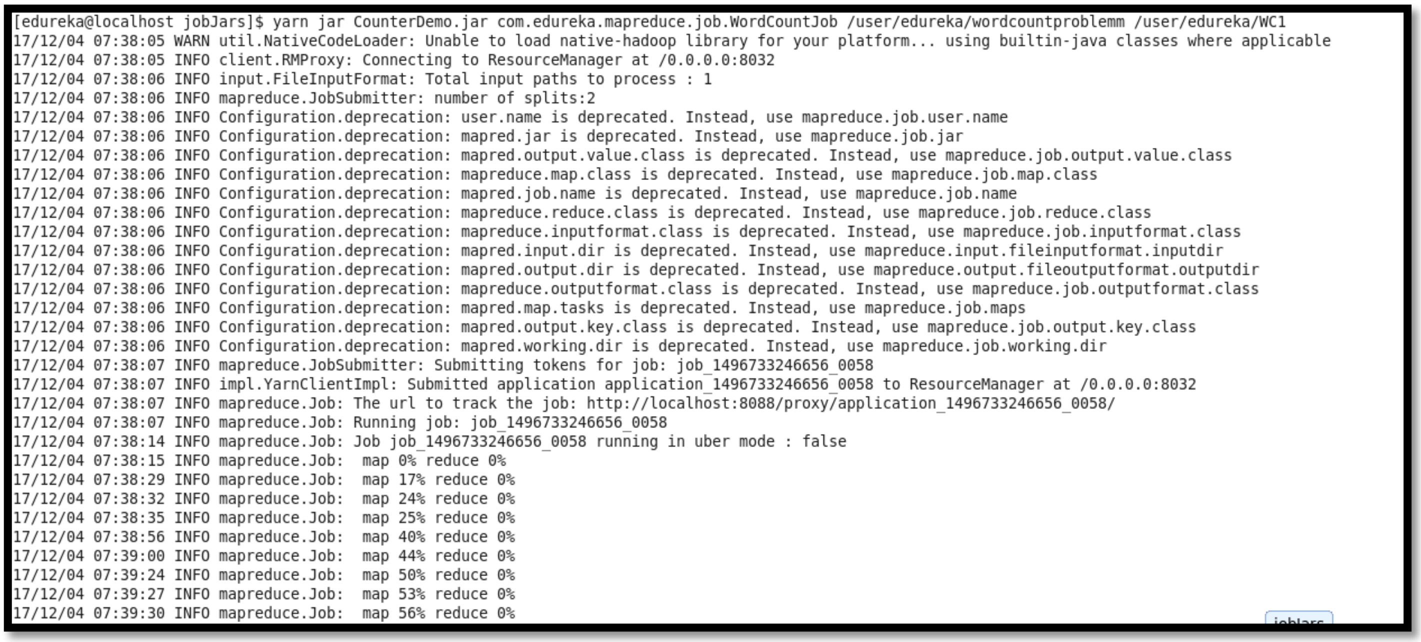
```
#NameNode configuration
#HADOOP_CONF_DIR/Core-site.xml change configuration element
#change the namenode_public_dns to your NameNode Public DNS
<configuration>
        <property>
                <name>fs.defaultFS</name>
                <value>hdfs://namenode_public_dns:9000</value>
        </property>
</configuration>

#$HADOOP_CONF_DIR/yarn-site.xml change configuration element
#change the namenode_public_dns to your NameNode Public DNS
<configuration>
        <property>
                <name>dfs.replication</name>
                <value>1</value>
        </property>
        <property>
                <name>dfs.namenode.name.dir</name>
                <value>file:///usr/local/hadoop/etc/hadoop_data/hdfs/namenode</value>
        </property>
</configuration>
```

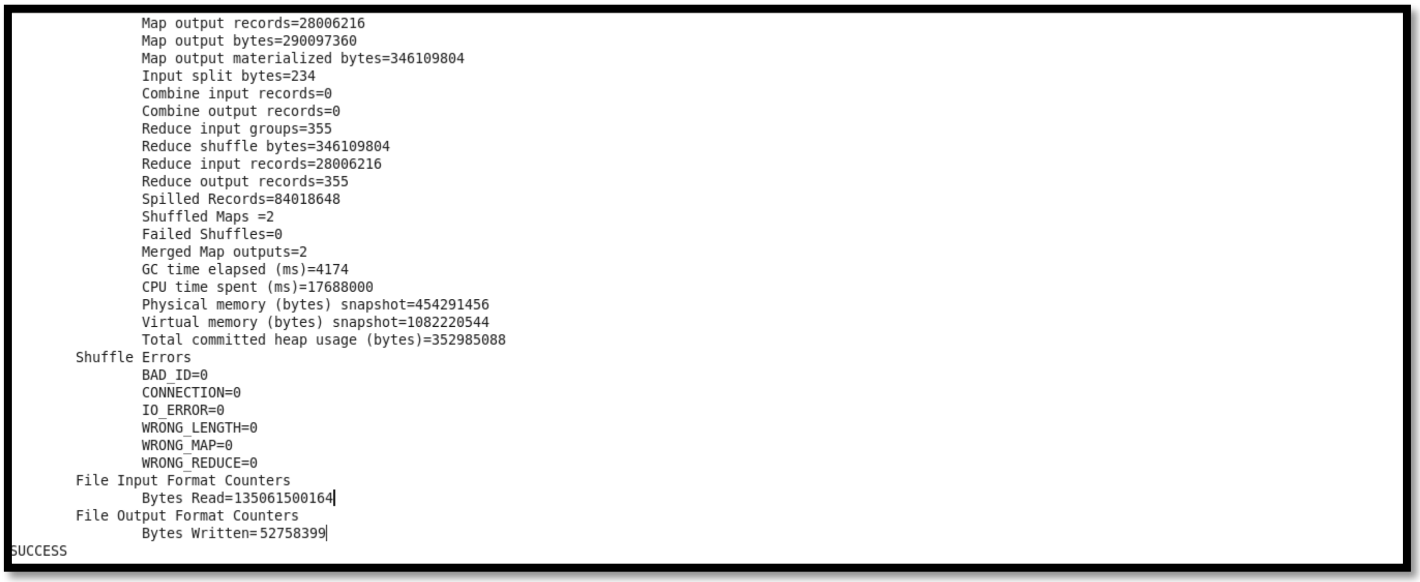
In the above configuration setting, especially in the yarn-site.xml, we have set the **replication factor to 1** Which means that the data blocks are copied only once to the datanodes. Therefore, we lose on the resilience of the cluster from any failures, but we do reduce the storage requirement and save time from writing multiple copies of data to blocks.

Below are the environment variable settings needed to use aliases on the bash command line.

#set environment variables in all the nodes
export JAVA_HOME=/usr
export PATH=$PATH:$JAVA_HOME/bin
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export HADOOP_CONF_DIR=/usr/local/hadoop/etc/hadoop



```
[edureka@localhost jobJars]$ yarn jar CounterDemo.jar com.edureka.mapreduce.job.WordCountJob /user/edureka/wordcountproblemm /user/edureka/WC1
17/12/04 07:38:05 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
17/12/04 07:38:05 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
17/12/04 07:38:06 INFO input.FileInputFormat: Total input paths to process : 1
17/12/04 07:38:06 INFO mapreduce.JobSubmitter: number of splits:2
17/12/04 07:38:06 INFO Configuration.deprecation: user.name is deprecated. Instead, use mapreduce.job.user.name
17/12/04 07:38:06 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
17/12/04 07:38:06 INFO Configuration.deprecation: mapred.output.value.class is deprecated. Instead, use mapreduce.job.output.value.class
17/12/04 07:38:06 INFO Configuration.deprecation: mapreduce.map.class is deprecated. Instead, use mapreduce.job.map.class
17/12/04 07:38:06 INFO Configuration.deprecation: mapred.job.name is deprecated. Instead, use mapreduce.job.name
17/12/04 07:38:06 INFO Configuration.deprecation: mapreduce.reduce.class is deprecated. Instead, use mapreduce.job.reduce.class
17/12/04 07:38:06 INFO Configuration.deprecation: mapreduce.inputformat.class is deprecated. Instead, use mapreduce.job.inputformat.class
17/12/04 07:38:06 INFO Configuration.deprecation: mapred.input.dir is deprecated. Instead, use mapreduce.input.fileinputformat.inputdir
17/12/04 07:38:06 INFO Configuration.deprecation: mapred.output.dir is deprecated. Instead, use mapreduce.output.fileoutputformat.outputdir
17/12/04 07:38:06 INFO Configuration.deprecation: mapreduce.outputformat.class is deprecated. Instead, use mapreduce.job.outputformat.class
17/12/04 07:38:06 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
17/12/04 07:38:06 INFO Configuration.deprecation: mapred.output.key.class is deprecated. Instead, use mapreduce.job.output.key.class
17/12/04 07:38:06 INFO Configuration.deprecation: mapred.working.dir is deprecated. Instead, use mapreduce.job.working.dir
17/12/04 07:38:07 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1496733246656_0058
17/12/04 07:38:07 INFO impl.YarnClientImpl: Submitted application application_1496733246656_0058 to ResourceManager at /0.0.0.0:8032
17/12/04 07:38:07 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application_1496733246656_0058/
17/12/04 07:38:07 INFO mapreduce.Job: Running job: job_1496733246656_0058
17/12/04 07:38:14 INFO mapreduce.Job: Job job_1496733246656_0058 running in uber mode : false
17/12/04 07:38:15 INFO mapreduce.Job:  map 0% reduce 0%
17/12/04 07:38:29 INFO mapreduce.Job:  map 17% reduce 0%
17/12/04 07:38:32 INFO mapreduce.Job:  map 24% reduce 0%
17/12/04 07:38:35 INFO mapreduce.Job:  map 25% reduce 0%
17/12/04 07:38:56 INFO mapreduce.Job:  map 40% reduce 0%
17/12/04 07:39:00 INFO mapreduce.Job:  map 44% reduce 0%
17/12/04 07:39:24 INFO mapreduce.Job:  map 50% reduce 0%
17/12/04 07:39:27 INFO mapreduce.Job:  map 53% reduce 0%
17/12/04 07:39:30 INFO mapreduce.Job:  map 56% reduce 0%
```

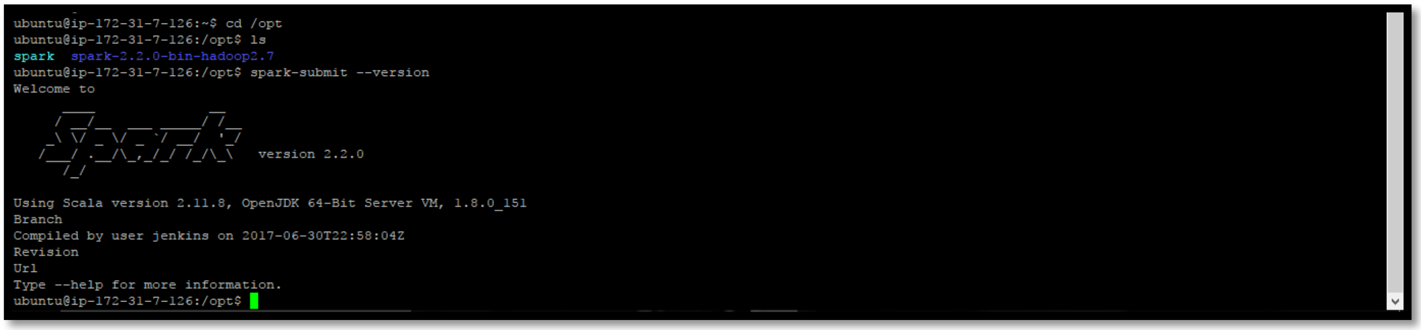


```
                Map output records=28006216
                Map output bytes=290097360
                Map output materialized bytes=346109804
                Input split bytes=234
                Combine input records=0
                Combine output records=0
                Reduce input groups=355
                Reduce shuffle bytes=346109804
                Reduce input records=28006216
                Reduce output records=355
                Spilled Records=84018648
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=4174
                CPU time spent (ms)=17688000
                Physical memory (bytes) snapshot=454291456
                Virtual memory (bytes) snapshot=1082220544
                Total committed heap usage (bytes)=352985088
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=135061500164
        File Output Format Counters
                Bytes Written=52758399
SUCCESS
```

It completed in 4.5 hours = 17,688 seconds

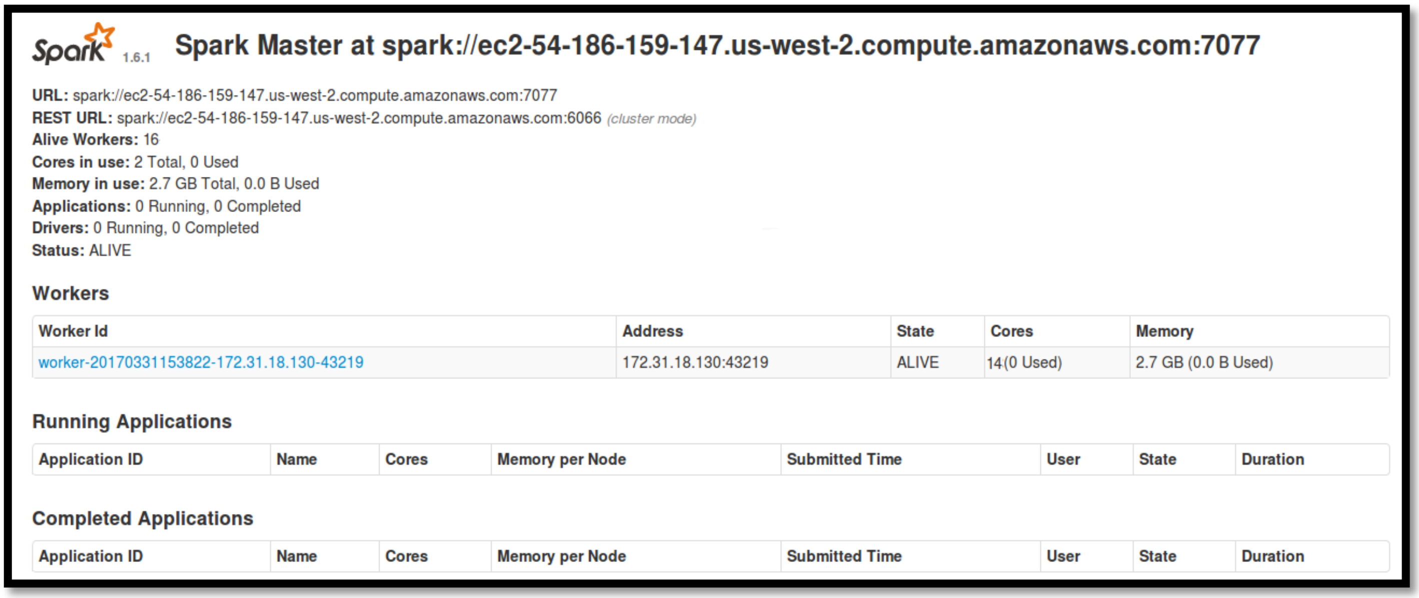
Spark

128GB



```
ubuntu@ip-172-31-7-126:~$ cd /opt
ubuntu@ip-172-31-7-126:/opt$ ls
spark  spark-2.2.0-bin-hadoop2.7
ubuntu@ip-172-31-7-126:/opt$ spark-submit --version
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.2.0
      /_/

Using Scala version 2.11.8, OpenJDK 64-Bit Server VM, 1.8.0_151
Branch
Compiled by user jenkins on 2017-06-30T22:58:04Z
Revision
Url
Type --help for more information.
ubuntu@ip-172-31-7-126:/opt$ 
```



Spark Master at spark://ec2-54-186-159-147.us-west-2.compute.amazonaws.com:7077

URL: spark://ec2-54-186-159-147.us-west-2.compute.amazonaws.com:7077
REST URL: spark://ec2-54-186-159-147.us-west-2.compute.amazonaws.com:6066 *(cluster mode)*
Alive Workers: 16
Cores in use: 2 Total, 0 Used
Memory in use: 2.7 GB Total, 0.0 B Used
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers

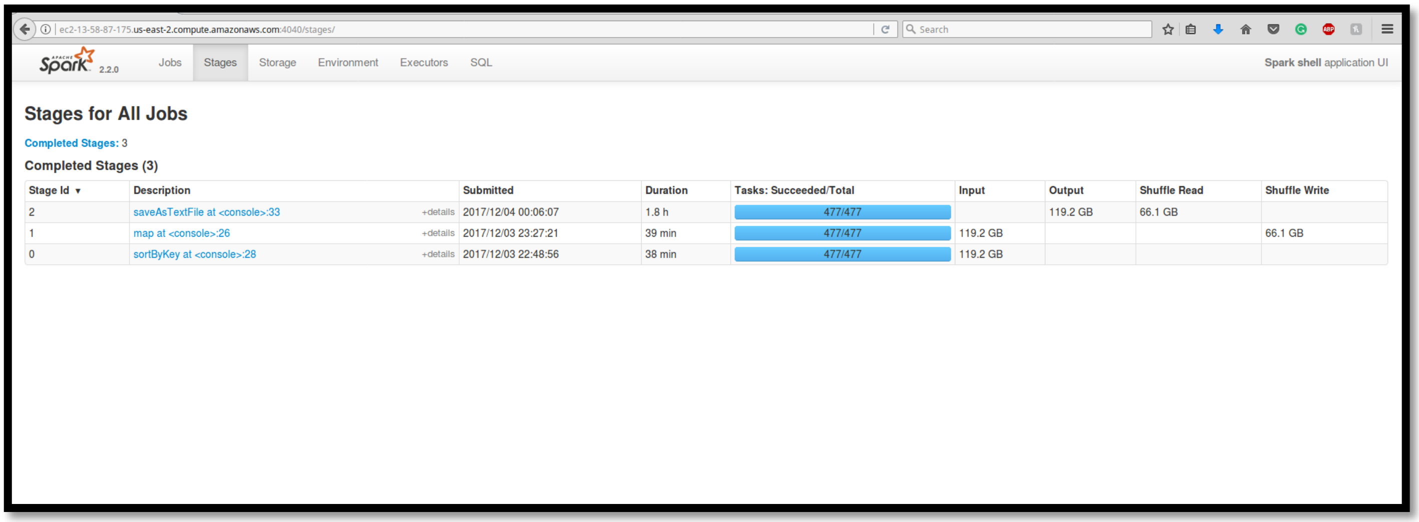
Worker Id	Address	State	Cores	Memory
worker-20170331153822-172.31.18.130-43219	172.31.18.130:43219	ALIVE	14(0 Used)	2.7 GB (0.0 B Used)

Running Applications

Application ID	Name	Cores	Memory per Node	Submitted Time	User	State	Duration

Completed Applications

Application ID	Name	Cores	Memory per Node	Submitted Time	User	State	Duration


ec2-13-58-87-175.us-east-2.compute.amazonaws.com:4040/stages/
Search
Spark 2.2.0
Jobs
Stages
Storage
Environment
Executors
SQL
Spark shell application UI

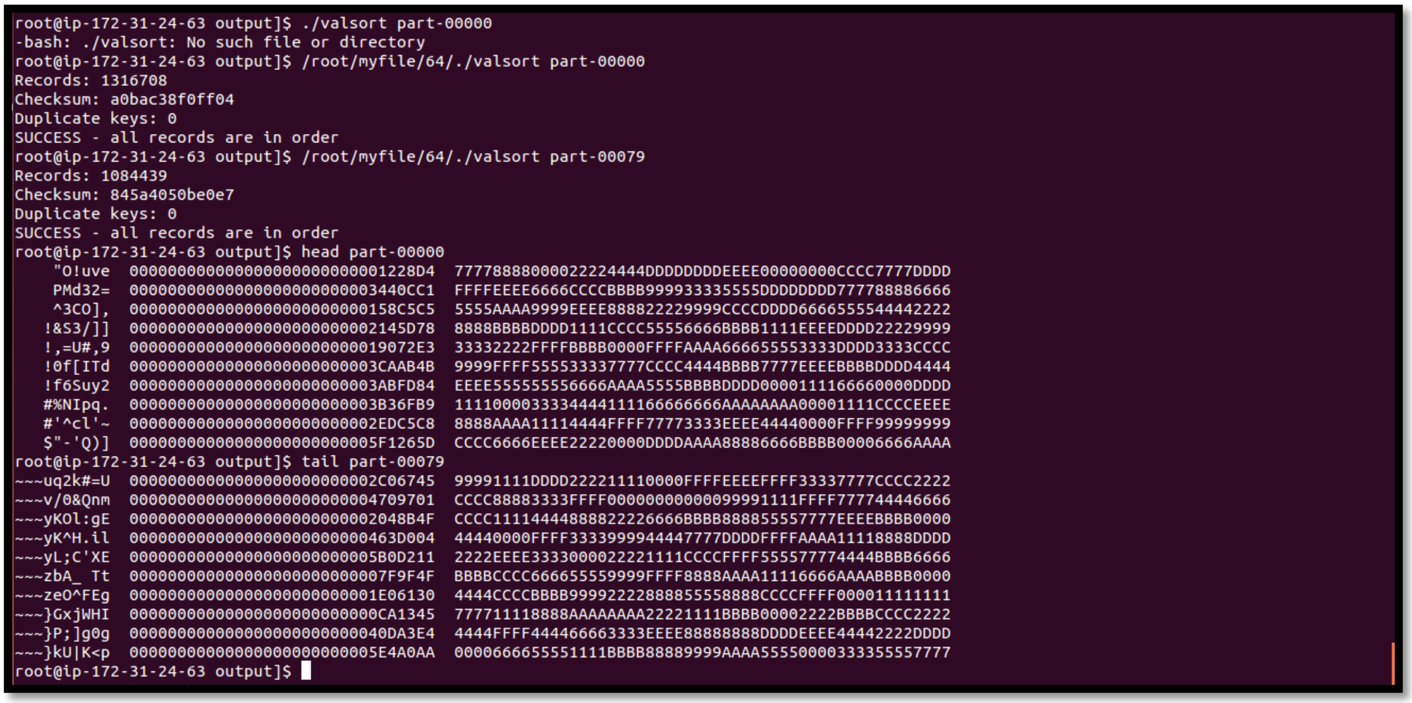
Stages for All Jobs

Completed Stages: 3

Completed Stages (3)

Stage Id ▾	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
2	saveAsTextFile at <console>:33 +details	2017/12/04 00:06:07	1.8 h	477/477		119.2 GB	66.1 GB	
1	map at <console>:26 +details	2017/12/03 23:27:21	39 min	477/477	119.2 GB			66.1 GB
0	sortByKey at <console>:28 +details	2017/12/03 22:48:56	38 min	477/477	119.2 GB			

It completed in 184 minutes which is way faster than the Hadoop and the shared memory.



```
root@ip-172-31-24-63 output]$ ./valsort part-00000
-bash: ./valsort: No such file or directory
root@ip-172-31-24-63 output]$ /root/myfile/64/./valsort part-00000
Records: 1316708
Checksum: a0bac38f0ff04
Duplicate keys: 0
SUCCESS - all records are in order
root@ip-172-31-24-63 output]$ /root/myfile/64/./valsort part-00079
Records: 1084439
Checksum: 845a4050be0e7
Duplicate keys: 0
SUCCESS - all records are in order
root@ip-172-31-24-63 output]$ head part-00000
    "O!uve  000000000000000000000000001228D4  77778888000022224444DDDDDDDDEEEE00000000CCCC7777DDDD
    PMd32=  00000000000000000000000003440CC1  FFFFEEEE6666CCCCBBBB999933335555DDDDDDDD777788886666
    ^3CO],  0000000000000000000000000158C5C5  5555AAAA9999EEEE888822229999CCCCDDDD6666555544442222
   !&S3/]]  00000000000000000000000002145D78  8888BBBBDDDD1111CCCC55556666BBBB1111EEEEDDDD22229999
   !,=U#,9  000000000000000000000000019072E3  33332222FFFFBBBB0000FFFFAAAA666655553333DDDD3333CCCC
   !0f[ITd  00000000000000000000000003CAAB4B  9999FFFF555533337777CCCC4444BBBB7777EEEEBBBBDDDD4444
   !f6Suy2  00000000000000000000000003ABFD84  EEEE55555555 6666AAAA5555BBBBDDDD000011116666 0000DDDD
   #%NIpq.  00000000000000000000000003B36FB9  11110000333344441111666666666AAAAAAAA00001111CCCCEEEE
   #'^cl'~  00000000000000000000000002EDC5C8  8888AAAA11114444FFFF77773333EEEE44440000FFFF99999999
   $"-'Q)]  00000000000000000000000005F1265D  CCCC6666EEEE22220000DDDDAAAA88886666BBBB00006666AAAA
root@ip-172-31-24-63 output]$ tail part-00079
~~~uq2k#=U  00000000000000000000000002C06745  99991111DDDD222211110000FFFFEEEEFFFF33337777CCCC2222
~~~v/0&Qnm  00000000000000000000000004709701  CCCC88883333FFFF000000000000999911111FFFF77774444666
~~~yKOl:gE  00000000000000000000000002048B4F  CCCC111144448888222266666BBBB88885555777 7EEEEBBBB0000
~~~yK^H.il  0000000000000000000000000463D004  44440000FFFF33339999444477777DDDDFFFFAAAA11118888DDDD
~~~yL;C'XE  00000000000000000000000005B0D211  2222EEEE333300002222111 1CCCCFFFF555577774444BBBB6666
~~~zbA_ Tt  000000000000000000000000007F9F4F  BBBBCCCC666655559999FFFF8888AAAA11116666AAAABBBB0000
~~~zeO^FEg  00000000000000000000000001E06130  4444CCCCBBBB999922228888555588888CCCCFFFF000011111111
~~~}GxjWHI  000000000000000000000000000CA1345  77771111 8888AAAAAAAA22221111BBBB00002222BBBBCCCC2222
~~~}P;]g0g  000000000000000000000000040DA3E4  4444FFFF444466663333EEEE88888888DDDDEEEE44442222DDDD
~~~}kU|K<p  00000000000000000000000005E4A0AA  0000666655551111BBBB88889999AAAA55550000333355557777
root@ip-172-31-24-63 output]$ 
```

The output was verified using Valsort as above.

Virtual Cluster (1-node i3.4xlarge)
1 node of instance type i3.4xlarge.

Instance ID i-05e1d742bd7ece3e1	Public DNS (IPv4) ec2-34-207-108-121.compute-1.amazonaws.com
Instance state	IPv4 Public IP

running 34.207.108.121

Instance type IPv6 IPs

i3.4xlarge -

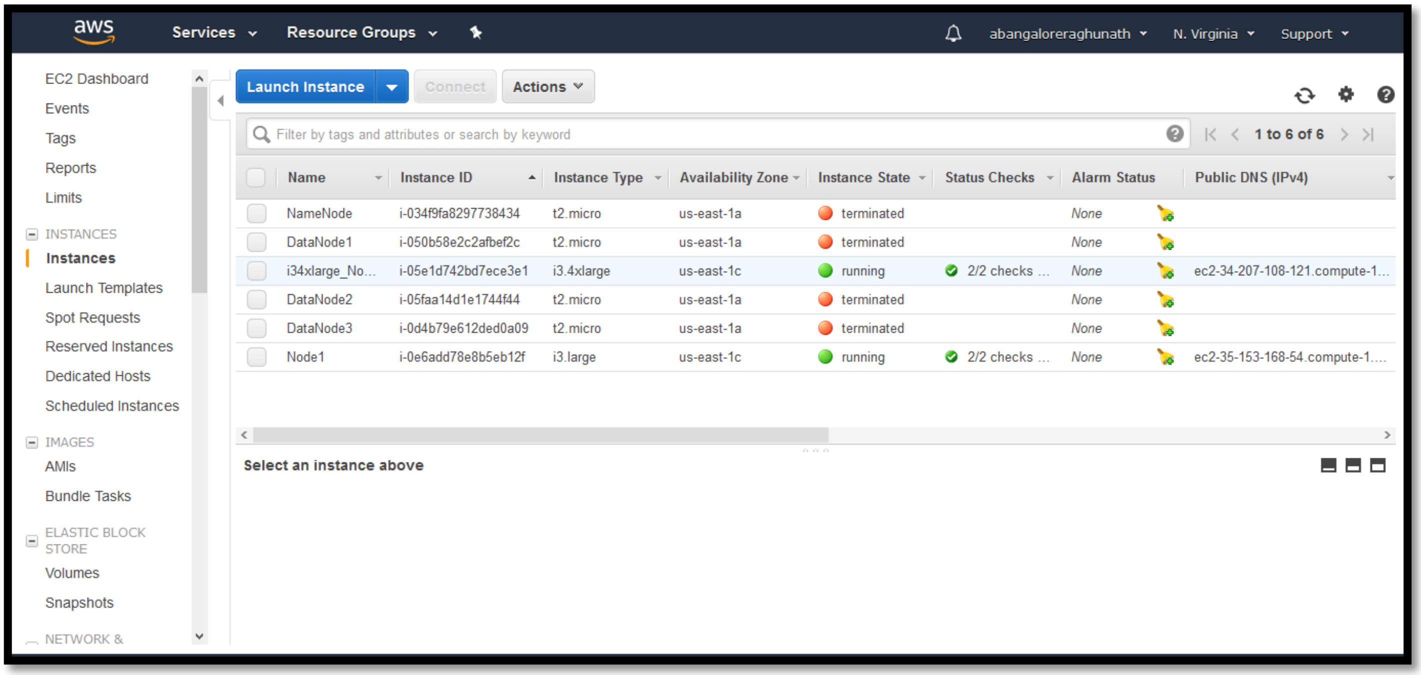
Elastic IPs

Private DNS

1. ip-172-31-39-224.ec2.internal

Availability zone Private IPs

us-east-1c 172.31.39.224

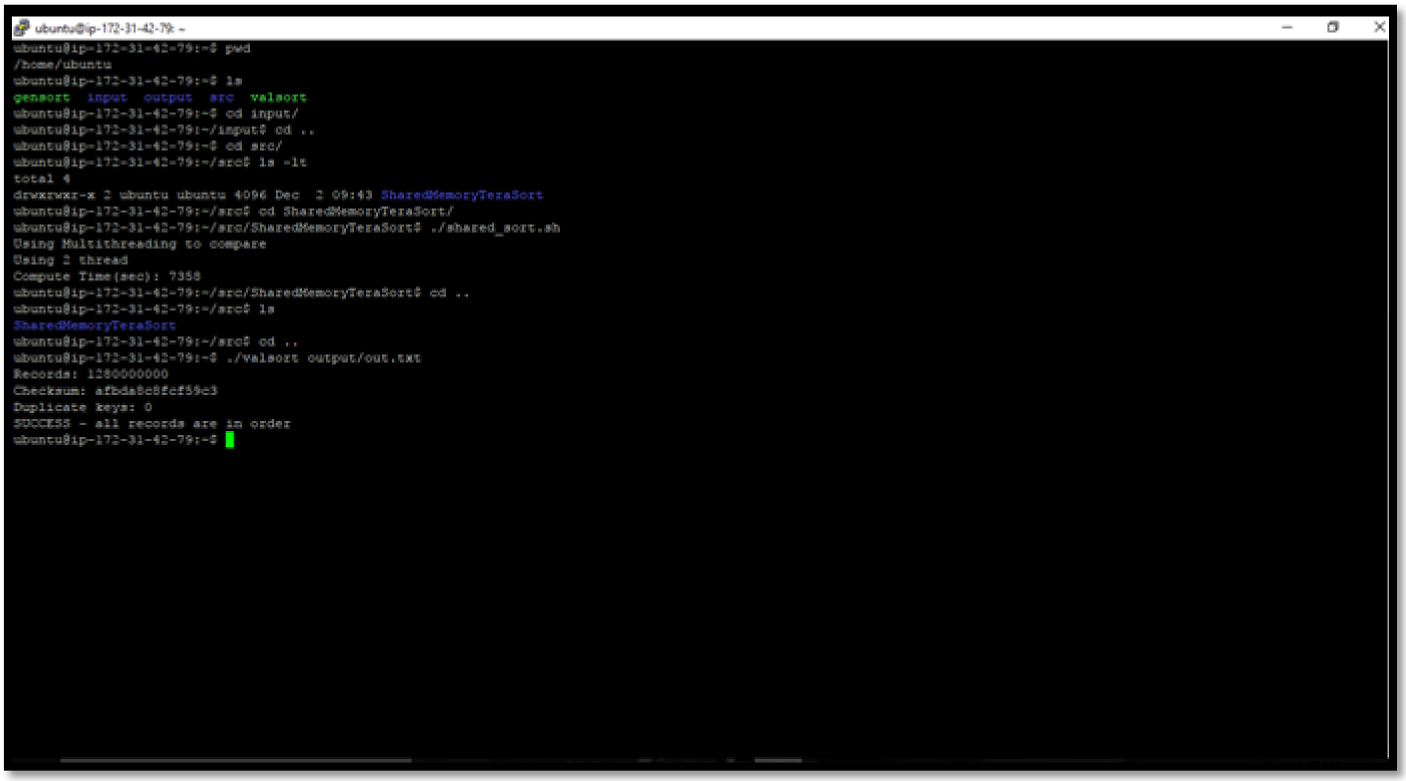


Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)
NameNode	i-034f9fa8297738434	t2.micro	us-east-1a	terminated		None	
DataNode1	i-050b58e2c2afbef2c	t2.micro	us-east-1a	terminated		None	
i34xlarge_No...	i-05e1d742bd7ece3e1	i3.4xlarge	us-east-1c	running	✓ 2/2 checks ...	None	ec2-34-207-108-121.compute-1...
DataNode2	i-05faa14d1e1744f44	t2.micro	us-east-1a	terminated		None	
DataNode3	i-0d4b79e612ded0a09	t2.micro	us-east-1a	terminated		None	
Node1	i-0e6add78e8b5eb12f	i3.large	us-east-1c	running	✓ 2/2 checks ...	None	ec2-35-153-168-54.compute-1....

Select an instance above

Shared memory

1TB

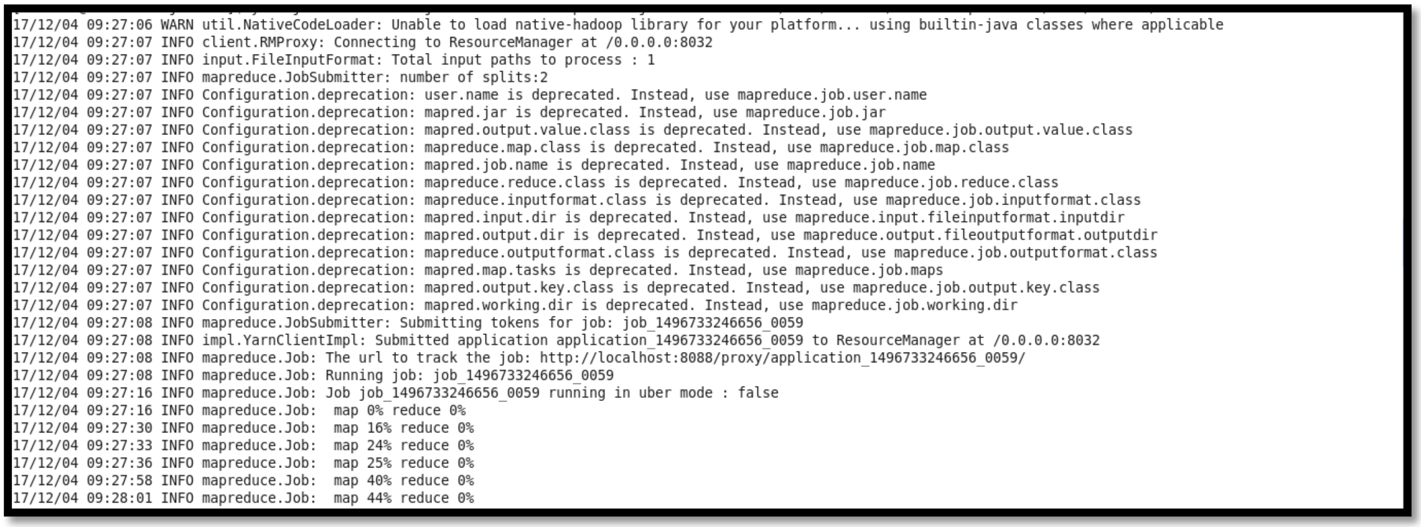


```
ubuntu@ip-172-31-42-79:~$ pwd
/home/ubuntu
ubuntu@ip-172-31-42-79:~$ ls
gensort  input  output  src  valsort
ubuntu@ip-172-31-42-79:~$ cd input/
ubuntu@ip-172-31-42-79:~/input$ cd ..
ubuntu@ip-172-31-42-79:~$ cd src/
ubuntu@ip-172-31-42-79:~/src$ ls -lt
total 4
drwxrwxr-x 2 ubuntu ubuntu 4096 Dec  2 09:43 SharedMemoryTeraSort
ubuntu@ip-172-31-42-79:~/src$ cd SharedMemoryTeraSort/
ubuntu@ip-172-31-42-79:~/src/SharedMemoryTeraSort$ ./shared_sort.sh
Using Multithreading to compare
Using 2 thread
Compute Time(sec): 7358
ubuntu@ip-172-31-42-79:~/src/SharedMemoryTeraSort$ cd ..
ubuntu@ip-172-31-42-79:~/src$ ls
SharedMemoryTeraSort
ubuntu@ip-172-31-42-79:~/src$ cd ..
ubuntu@ip-172-31-42-79:~$ ./valsort output/out.txt
Records: 1280000000
Checksum: afbda8c8fcf59c3
Duplicate keys: 0
SUCCESS - all records are in order
ubuntu@ip-172-31-42-79:~$
```

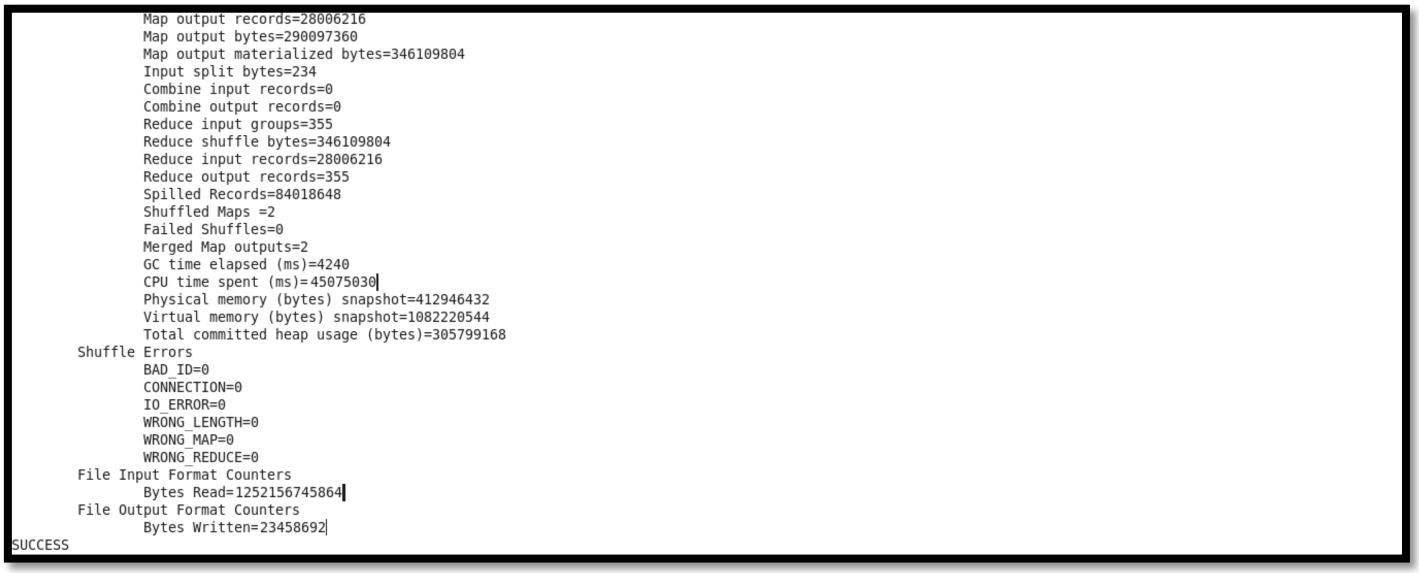
I faced a broken pipe exception twice, but finally it ran in the last run and completed the sort in 8 hours and 30 mins. 30,600 second.

Hadoop

1TB

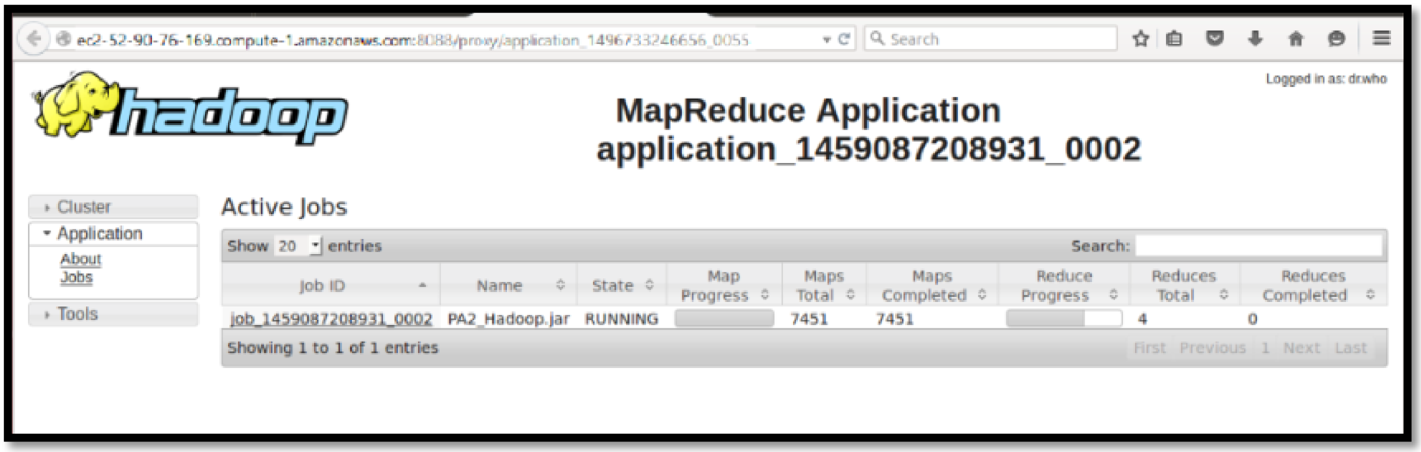


```
17/12/04 09:27:06 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
17/12/04 09:27:07 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
17/12/04 09:27:07 INFO input.FileInputFormat: Total input paths to process : 1
17/12/04 09:27:07 INFO mapreduce.JobSubmitter: number of splits:2
17/12/04 09:27:07 INFO Configuration.deprecation: user.name is deprecated. Instead, use mapreduce.job.user.name
17/12/04 09:27:07 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
17/12/04 09:27:07 INFO Configuration.deprecation: mapred.output.value.class is deprecated. Instead, use mapreduce.job.output.value.class
17/12/04 09:27:07 INFO Configuration.deprecation: mapreduce.map.class is deprecated. Instead, use mapreduce.job.map.class
17/12/04 09:27:07 INFO Configuration.deprecation: mapred.job.name is deprecated. Instead, use mapreduce.job.name
17/12/04 09:27:07 INFO Configuration.deprecation: mapreduce.reduce.class is deprecated. Instead, use mapreduce.job.reduce.class
17/12/04 09:27:07 INFO Configuration.deprecation: mapreduce.inputformat.class is deprecated. Instead, use mapreduce.job.inputformat.class
17/12/04 09:27:07 INFO Configuration.deprecation: mapred.input.dir is deprecated. Instead, use mapreduce.input.fileinputformat.inputdir
17/12/04 09:27:07 INFO Configuration.deprecation: mapred.output.dir is deprecated. Instead, use mapreduce.output.fileoutputformat.outputdir
17/12/04 09:27:07 INFO Configuration.deprecation: mapreduce.outputformat.class is deprecated. Instead, use mapreduce.job.outputformat.class
17/12/04 09:27:07 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
17/12/04 09:27:07 INFO Configuration.deprecation: mapred.output.key.class is deprecated. Instead, use mapreduce.job.output.key.class
17/12/04 09:27:07 INFO Configuration.deprecation: mapred.working.dir is deprecated. Instead, use mapreduce.job.working.dir
17/12/04 09:27:08 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1496733246656_0059
17/12/04 09:27:08 INFO impl.YarnClientImpl: Submitted application application_1496733246656_0059 to ResourceManager at /0.0.0.0:8032
17/12/04 09:27:08 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/application_1496733246656_0059/
17/12/04 09:27:08 INFO mapreduce.Job: Running job: job_1496733246656_0059
17/12/04 09:27:16 INFO mapreduce.Job: Job job_1496733246656_0059 running in uber mode : false
17/12/04 09:27:16 INFO mapreduce.Job:  map 0% reduce 0%
17/12/04 09:27:30 INFO mapreduce.Job:  map 16% reduce 0%
17/12/04 09:27:33 INFO mapreduce.Job:  map 24% reduce 0%
17/12/04 09:27:36 INFO mapreduce.Job:  map 25% reduce 0%
17/12/04 09:27:58 INFO mapreduce.Job:  map 40% reduce 0%
17/12/04 09:28:01 INFO mapreduce.Job:  map 44% reduce 0%
```



```
                Map output records=28006216
                Map output bytes=290097360
                Map output materialized bytes=346109804
                Input split bytes=234
                Combine input records=0
                Combine output records=0
                Reduce input groups=355
                Reduce shuffle bytes=346109804
                Reduce input records=28006216
                Reduce output records=355
                Spilled Records=84018648
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=4240
                CPU time spent (ms)=45075030
                Physical memory (bytes) snapshot=412946432
                Virtual memory (bytes) snapshot=1082220544
                Total committed heap usage (bytes)=305799168
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=1252156745864
        File Output Format Counters
                Bytes Written=23458692
SUCCESS
```

12:31 hours one node of i34x large



ec2-52-90-76-169.compute-1.amazonaws.com:8088/proxy/application_1496733246656_0055
Search
Logged in as: dr.who
hadoop
Cluster
Application
About
Jobs
Tools

MapReduce Application application_1459087208931_0002

Active Jobs

Show 20 entries Search:

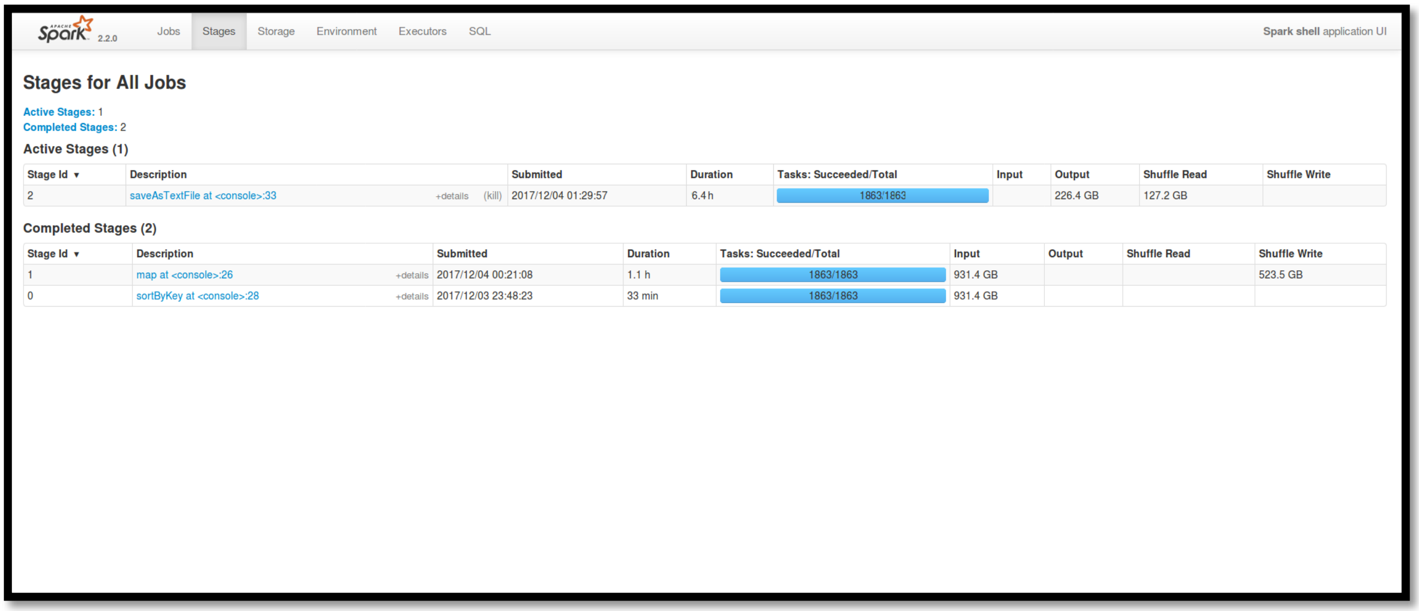
Job ID	Name	State	Map Progress	Maps Total	Maps Completed	Reduce Progress	Reduces Total	Reduces Completed
job_1459087208931_0002	PA2_Hadoop.jar	RUNNING		7451	7451		4	0

Showing 1 to 1 of 1 entries

First Previous 1 Next Last

Spark

1TB i3.4xlarge instance virtual cluster Pseudo Mode.



Stages for All Jobs

Active Stages: 1
Completed Stages: 2

Active Stages (1)

Stage Id ▾	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
2	saveAsTextFile at <console>:33 +details (kill)	2017/12/04 01:29:57	6.4 h	1863/1863		226.4 GB	127.2 GB	

Completed Stages (2)

Stage Id ▾	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
1	map at <console>:26 +details	2017/12/04 00:21:08	1.1 h	1863/1863	931.4 GB			523.5 GB
0	sortByKey at <console>:28 +details	2017/12/03 23:48:23	33 min	1863/1863	931.4 GB			

As we can see above the Spark instances could not complete initially, but they did eventually complete in **7.5** hours

Comparison of sorts on various levels of hardware quality and quantity

Experiment (instance/dataset)	Shared Memory TeraSort	Hadoop TeraSort	Spark TeraSort
Compute Time (sec) [1xi3.large 128GB]	13380	17688	11040
Data Read (GB) [1xi3.large 128GB]	134.08	131.256	128
Data Write (GB) [1xi3.large 128GB]	133.46	128.95	128
I/O Throughput (MB/sec) [1xi3.large 128GB]	19.98	14.71	23.188
Compute Time (sec) [1xi3.4xlarge 1TB]	30605	45060	27000
Data Read (GB) [1xi3.4xlarge 1TB]	1TB	1TB	1TB
Data Write (GB) [1xi3.4xlarge 1TB]	1TB	1TB	1TB
I/O Throughput (MB/sec) [1xi3.4xlarge 1TB]	65.39	44.38	74.07