



PA-2 REPORT

CS-553 - CLOUD COMPUTING FALL 2017

Rushit Pandya A20381916
Achal Lathia A20375208
Viral Bhojani A20380100

Contents

Problem.....	2
Solution	2
Run Time Environment Settings	2
Installation	3
Java.....	3
Shared Memory	3
Hadoop.....	3
Spark	4
Challenges.....	5
External Sorting.....	5
Hadoop.....	5
Spark	5
Performance Evaluation.....	5
Overview on CloudSort Benchmark.....	11
Comparison Results with Sort Benchmark.....	11
Execution Screenshots	12
Shared Memory	12
128GB Dataset	12
1TB Dataset	14
Hadoop Single Node	16
128GB Dataset	16
1TB Dataset	19
Hadoop 8-Node Cluster	23
1TB Dataset	23
Spark Single Node Cluster	27
128GB Dataset	27
1TB Dataset	32
Spark 8 Node Cluster.....	34

Problem

- The problem is to sort a large dataset of varying size 128GB and 1TB on different distributed environment as well as on standalone platform. Here, distributed environment refers to Hadoop and Spark Framework. For Standalone Execution, External Sorting Approach is required to adapt. Running platform on which program should be executed, is Amazon AWS. For TeraSort of 128GB, it is required to execute on i3.large instance and for execution of 1TB TeraSort, it is required to run on i3.4xlarge instance from amazon aws.

Solution

- To generate a large dataset, gensort is used which generated records having first 10-byte as key and rest as value. After sorting is done, valsort is used to check whether all records are in order or not.
- For Shared Memory, sorting is done with help of merge-sort algorithm. According to available free memory from JVM, dataset is divided into several chunks of fixed size. Each of this chunk is kept in memory and sorting has been performed on in-memory chunks. First 10 bytes of each line is considered as key and sorted upon that key. Multithreading is achieved on above steps where each thread works on unique chunk to divide and sort. Once sorting is achieved, all the chunks are merged and sorted.
- For Hadoop Framework, we have used MapReduce for sorting large dataset. During Map Phase, each mapper generates a key of first 10 bytes of records. The result of map phase is sorted collections of keys in ascending order. Here no intermediate runs from map step are been stored and hence reduce phase does no work. This is disadvantage of Hadoop framework. We executed TeraSort on both single node as well as multiple nodes.
- For Spark Framework, we must put input data in hdfs directory as spark takes input from hdfs. Each record from the input file is read and a key of first 10 byte is generated. Using SortByKey function, each of the records are sorted in ascending order. Once results are sorted, all the records are stored in hdfs output directory specified. Spark is faster than Hadoop because it performs execution in-memory.

Run Time Environment Settings

➔ First, we must mount extra space as intermediate shuffle data creates a lot of temporary files. This is achieved by raiding external memory using raid0. When mount_generate_raid.sh script is executed, additional 2 volumes of EBS and one SSD is mounted at mount point defined. Here, EBS volumes are pre-configured while creating instances.

Installation

) Deployed on UBUNTU

Java

- Executing below commands will install **java version 1.8.0_151**.

```
sudo -E apt-get purge oracle-java8-installer -y  
sudo -E apt-get install oracle-java8-installer -y
```

Shared Memory

- 128GB: Number of Threads = 2 as Total VCPUs are 2 for i3.large instance. Max Heap Size for JVM is 12GB as total 128GB memory and Min Tmp File Size will be 512MB size.
- 1TB: Number of Threads = 16 as Total VCPUs are 16 for i3.4xlarge instance. Max Heap Size for JVM is 120GB as total 128GB memory and Min Tmp File Size will be 481MB size.

Hadoop

- All the installation steps for Hadoop are included in `hadoop_install.sh`
- Below commands will install **hadoop version 2.7.1**.

```
sudo wget https://archive.apache.org/dist/hadoop/core/hadoop-2.7.1/hadoop-  
2.7.1.tar.gz  
sudo tar xzf hadoop-2.7.1.tar.gz
```

- Also, we are required to set environment variables for hadoop as below:

```
export HADOOP_HOME=/usr/local/hadoop  
export HADOOP_MAPRED_HOME=/usr/local/hadoop  
export HADOOP_COMMON_HOME=/usr/local/hadoop  
export HADOOP_HDFS_HOME=/usr/local/hadoop  
export YARN_HOME=/usr/local/hadoop  
export HADOOP_COMMON_LIB_NATIVE_DIR=/usr/local/hadoop/lib/native  
export JAVA_HOME=/usr/  
export  
PATH=$PATH:/usr/local/hadoop/sbin:/usr/local/hadoop/bin:/usr/bin:/bin:/opt  
/spark/bin
```

- Changes required in configuration files for Single Node Cluster
1) `core-site.xml`: Add public DNS of the node.

- 2) hdfs-site.xml: Add block size and point dir of namenode and data node to mount point.
- 3) mapred-site.xml: Add number of reducers and provide temporary directory for mapred.
- 4) yarn-site.xml: Add disk-health-checker-max-disk-utilization property with 98.5% value to avoid having unhealthy node issue.

➤ Additional changes in configuration files for Multi Node Cluster

- 1) Generate RSA public key and add it to authorized keys on master node.

```
ssh-keygen -t rsa -P ""  
cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
```

- 2) Send key to all slave node to create password less ssh.
- 3) slaves – Add own public DNS if slave else add private IP of master node followed by DNS of all slaves.
- 4) masters- Add public DNS if master.
- 5) hdfs-site.xml- Check if replication factor is 1 i.e. replication turned off.

Spark

- All the installation steps for Hadoop are included in install_spark.sh
- Below command will download **spark-2.2.0**
wget https://d3kbcqa49mib13.cloudfront.net/spark-2.2.0-bin-hadoop2.7.tgz
- Once spark tar is downloaded, untar spark.
sudo tar zxvf spark-2.2.0-bin-hadoop2.7.tgz -C /opt
- Set environment variable to bashrc file
export SPARK_HOME=/opt/spark
source ~/.bashrc
- We have added one property in file which is at location /opt/spark/conf/spark-defaults.conf.template. After adding below line, all intermediate temp will be stored in file specified below.
spark.local.dir /mnt/raid/spark_temp
- Also, we have to update one more file which is at location /opt/spark/conf/log4j.properties.template. Find log4j.rootCategory=INFO file and replace INFO by ERROR. This will avoid creating unnecessary temp logs except error logs.

Challenges

External Sorting

- Due to less heap size of JVM, there was always an error of “Memory Out of Bound Exception” while running for Big Data-Set. Hence, we increased heap size by specifying Xmx value during execution.
- Also, our execution generates large number of small tmp files. So, we must push the limit of generating number of files to 65536 to avoid error of “Too many files open”.

Hadoop

- There was always issue of “unhealthy node” due to excessive amount of temp file generate. It required a considerable amount of external memory to be raided and configured in such a way that all the temp files must be stored at mount point.

Spark

- There was memory issue due to excessive shuffle data generated. So, we changed log4j.properties files in such a way that all error logs are only stored instead of all the logs.

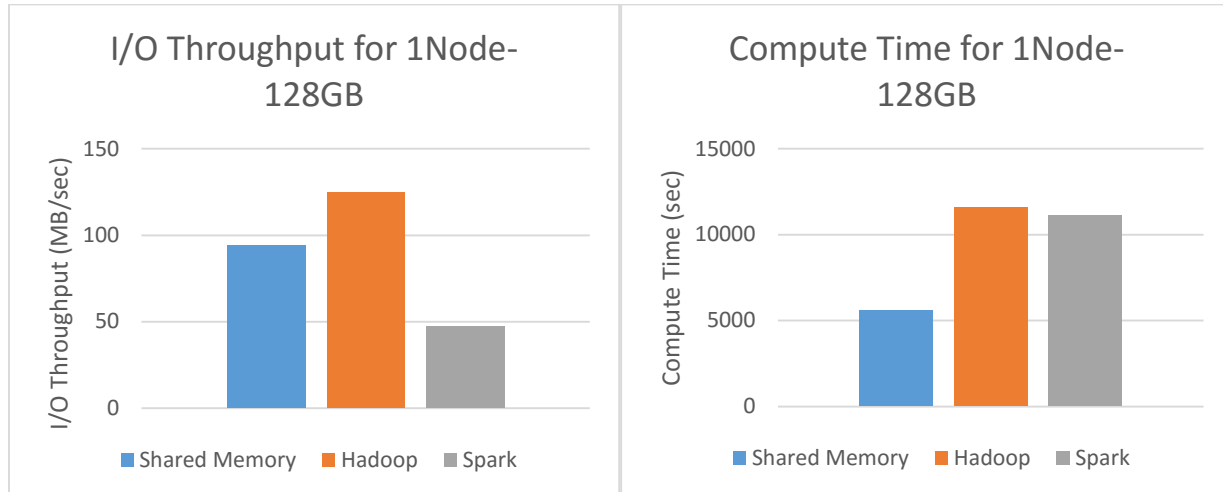
Performance Evaluation

➔ Below section covers performance results of TeraSort on 1TB and 128 GB Dataset for shared memory, Hadoop, and spark. We analyzed and compared results of Shared Memory, Hadoop, and Spark Configuration for 128 GB and 1TB Dataset. Also, we compared results of 1 node cluster vs 8-node cluster.

➤ Comparison Results and Running Time Factors for 128 GB Dataset

- 1) Instance-Type: i3.large (2 VCPUs, 15.25 GB RAM, 475 SSD)
- 2) EBS Volume Added: 600GB
- 3) No of Threads used for Shared Memory Computation: 2
- 4) Number of Reducers for Hadoop: 2
- 5) Block Size: 1GB
- 6) Comparison:

	Shared Memory	Hadoop	Spark
Compute Time(Seconds)	5580.11	11580.18	11109.9
Data Read(GB)	256	639	322
Data Write(GB)	256	769	194
I/O Throughput(MB/Sec)	93.95657075	124.505146	47.559744



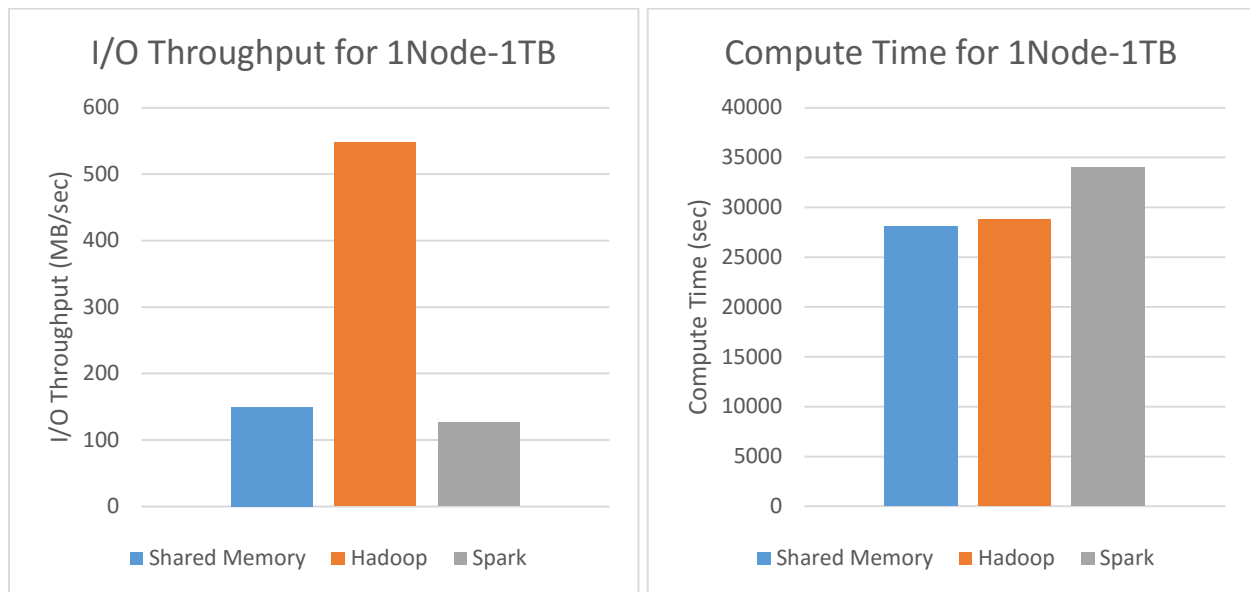
7) Conclusion:

→ From above graph, it can be inferred that shared memory takes much less time as compared to Hadoop and spark. The reason behind such behavior is that number of I/Os for shared memory is less as compared to Hadoop and spark as it writes to raid 0 level instead of HDFS and hence enhancing performance much faster for shared memory. Also, throughput for Hadoop is higher than spark and shared memory. But still Hadoop takes much I/Os to disk which degrades its performance. On the other hand, spark takes much less I/Os to disk as compared to Hadoop. Hence, performance for spark is better than Hadoop as it performs in-memory computation. So, we can conclude that disk is bottleneck for spark as it is only efficient with in-memory computation. Also, number of threads considered for shared memory to make it more efficient, is 2 which is equal to cores of instance

➤ Comparison Results and Running Time Factors for 1 TB Dataset

- 1) Instance-Type: i3.4xlarge (16 VCPUs, 122 GB RAM, 2 x 1900 SSD)
- 2) EBS Volume Added: 3TB
- 3) No of Threads used for Shared Memory Computation: 16
- 4) Number of Reducers for Hadoop: 8
- 5) Block Size: 1GB
- 6) Comparison:

	Shared Memory	Hadoop	Spark
Compute Time(Second)	28059.6	28800	33987
Data Read(GB)	2048	7182.9	2613.8
Data Write(GB)	2048	8227.28	1589.9
I/O Throughput(MB/Sec)	149.478396	547.9175111	126.654

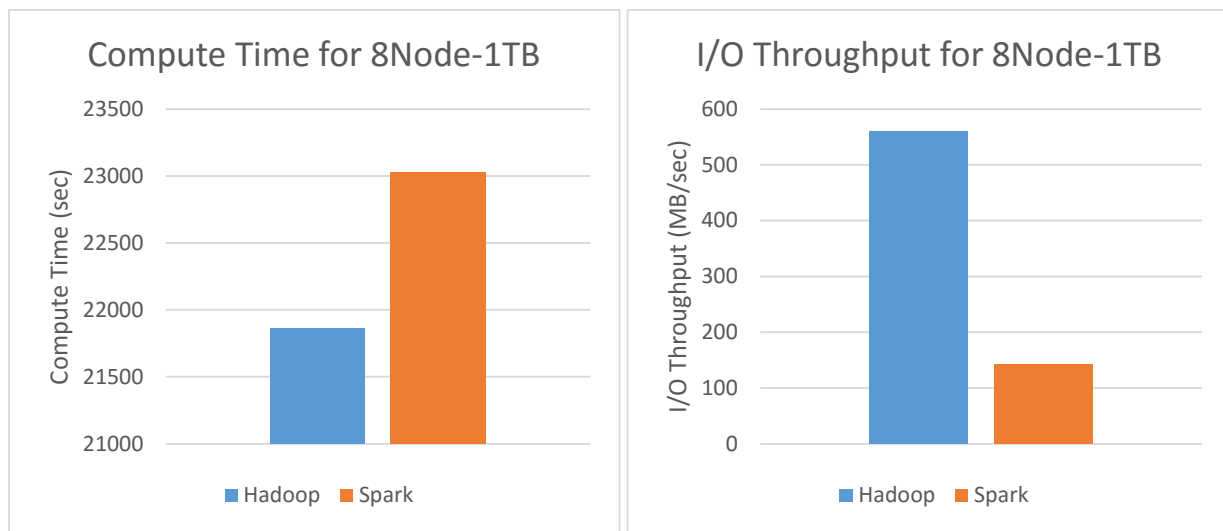


7) Conclusion:

→ From above graph, it can be inferred that shared memory takes much less time as compared to Hadoop and spark. The reason behind such behavior is that number of I/Os for shared memory is less as compared to Hadoop and spark as it writes to raid 0 level instead of HDFS and hence enhancing performance much faster for shared memory. Also, throughput for Hadoop is higher than spark and shared memory. But still Hadoop takes much I/Os to disk which degrades its performance. On the other hand, spark takes much less I/Os to disk as compared to Hadoop. Hence, performance for spark is better than Hadoop as it performs in-memory computation. Also, number of threads considered for shared memory to make it more efficient, is 16 which is equal to number of cores of instance. Hence **Shared Memory** gives best performance for 1 Node Cluster.

➤ Comparison Results for 8 Node Cluster

	Hadoop	Spark
Compute_Time(Second)	21867	23030.7
Data Read(GB)	5933.69	2113.7
Data Write(GB)	6056.47	1089
I/O Throughput(MB/Sec)	561.4818603	142.3997013



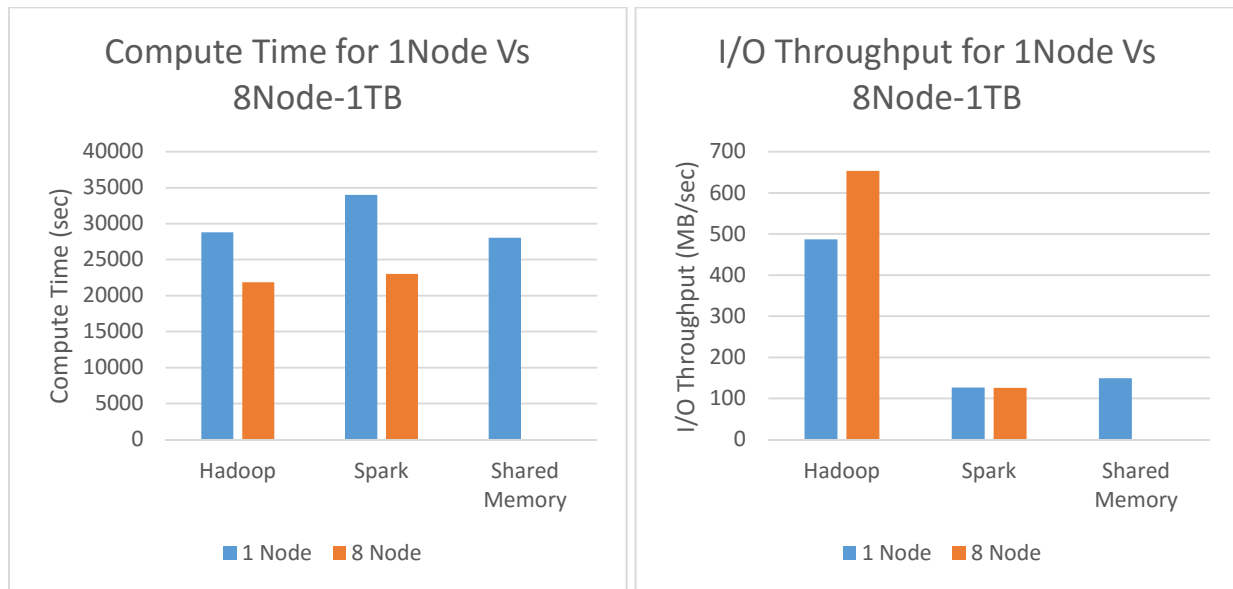
Conclusion:

→ From above graph, it can be clearly inferred that compute time for Hadoop is much less than spark. Though spark takes much time for computation, only 10% of total time was taken to sort the data and remaining time was taken to write data to disk. This concludes that even though spark has much computation time, it is still faster than Hadoop for sorting operation. Hence **Spark** gives best performance for 8 Nodes.

➤ Comparison Results for 8 Node Cluster vs 1 Node Cluster

	Hadoop	Spark	Shared Memory
1 Node	28800	33987	28059.6
8 Node	21867	23030.7	N/A

	Hadoop	Spark	Shared Memory
1 Node	487.03	126.524	149.47
8 Node	653.77	125.98	N/A

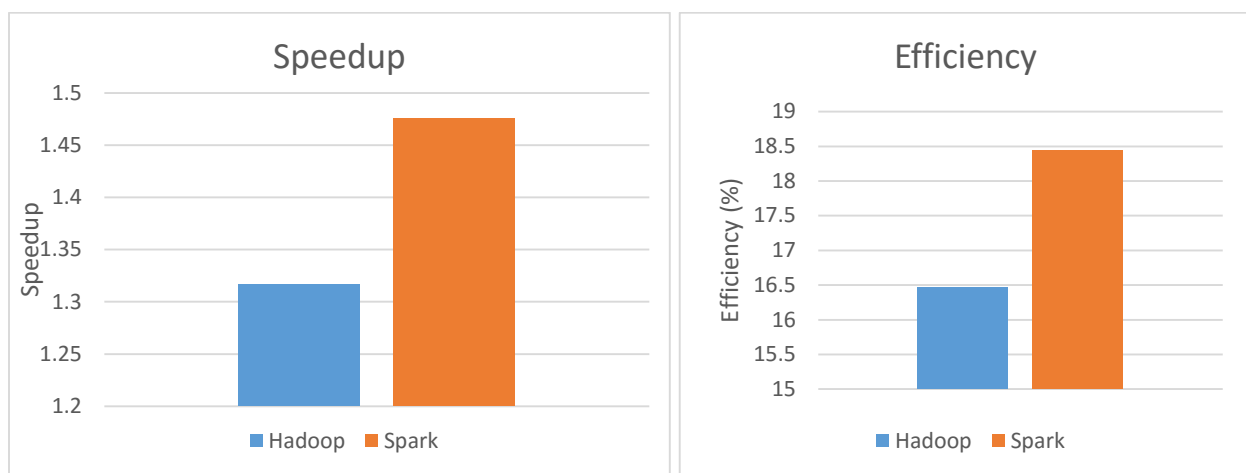


Conclusion:

→ Above graph shows comparison results of compute time and throughput between 1 Node and 8 Node for 1 TB dataset of shared memory, Hadoop and spark configuration. It is clearly inferred that compute time for each configuration is less for 8 node cluster than single node cluster. Hence, for each configuration 8 nodes computation gives better performance as workload is divided among nodes in cluster.

→ Also, as you scale up on number of nodes to 100, spark will give best performance. The reason for such behavior is, spark computes data in-memory and Hadoop always compute on intermediate data stored on disk. So as nodes increases, available memory for spark to do computation increases. Hence it computes much faster than Hadoop and shared memory. The same behavior will work if cluster scales upto 1000 nodes.

➤ Speedup and Efficiency



→ Speedup = Compute Time for 1 Node / Compute Time for 8 Node

→ Efficiency = Speedup / number of nodes

Conclusion:

→ Above graphs justifies all our above argument for comparison true as speedup and efficiency for spark is more than Hadoop. Hence spark gives better performance than Hadoop.

Experiment (instance/dataset)	Shared Memory TeraSort	Hadoop TeraSort	Spark TeraSort	MPI TeraSort
Compute Time (sec) [1xi3.large 128GB]	5580.11	11580.18	11109.9	N/A
Data Read (GB) [1xi3.large 128GB]	256	639	322	N/A
Data Write (GB) [1xi3.large 128GB]	256	769	194	N/A
I/O Throughput (MB/sec) [1xi3.large 128GB]	93.95	124.50	47.55	N/A
Compute Time (sec) [1xi3.4xlarge 1TB]	28059.6	28800	33987	N/A
Data Read (GB) [1xi3.4xlarge 1TB]	2048	7182.9		N/A
Data Write (GB) [1xi3.4xlarge 1TB]	2048	8227.28	1589.9	N/A
I/O Throughput (MB/sec) [1xi3.4xlarge 1TB]	149.47	547.91	126.654	N/A
Compute Time (sec) [8xi3.large 1TB]	N/A	21867	23030.7	N/A
Data Read (GB) [8xi3.large 1TB]	N/A	5933.69	2113.7	N/A
Data Write (GB) [8xi3.large 1TB]	N/A	6056.47	1089	N/A
I/O Throughput (MB/sec) [8xi3.large 1TB]	N/A	561.481	142.399	N/A
Speedup (weak scale)	N/A	1.31	1.47	N/A
Efficiency (weak scale)	N/A	16.46%	18.44%	N/A

Overview on CloudSort Benchmark

→ Cloudsort benchmark is a new total-cost ownership (TCO) benchmark which is based on implementing efficient external sort along with achieving minimum cost for cloud platforms. This benchmark mainly focuses on achieving desired cost for cloud environments having intensive IO workloads. Traditionally proposed TCO were not able to cover costs dimensions such as management and maintenance cost. But, after introduction of IAAS, major vendors were capable of offering on-demand compute, network and storage at desired scale with amortized pricing covering overall costs of demanded resources. Cloudsort explains efficiency of external sort from a TCO perspective. It focuses of finding minimum cost for sorting fixed number of records on any public cloud. It was aiming for transition to public cloud and was more efficient because of its extensive IO computation capabilities. Hence external sort was considered as base for estimating TCO as external sort has highly intensive IO workloads. There are many several reason explained about why public cloud was taken into consideration. Most of advantages for considering public cloud was based on accessibility and affordability of public cloud for huge enterprises. Also, many ground rules were proposed which clearly states nature of public cloud to be considered. Hence, this benchmark not only focuses on building platforms with efficient usage of resources, but also find the most efficient sort implementations from TCO perspective.

Comparison Results with Sort Benchmark

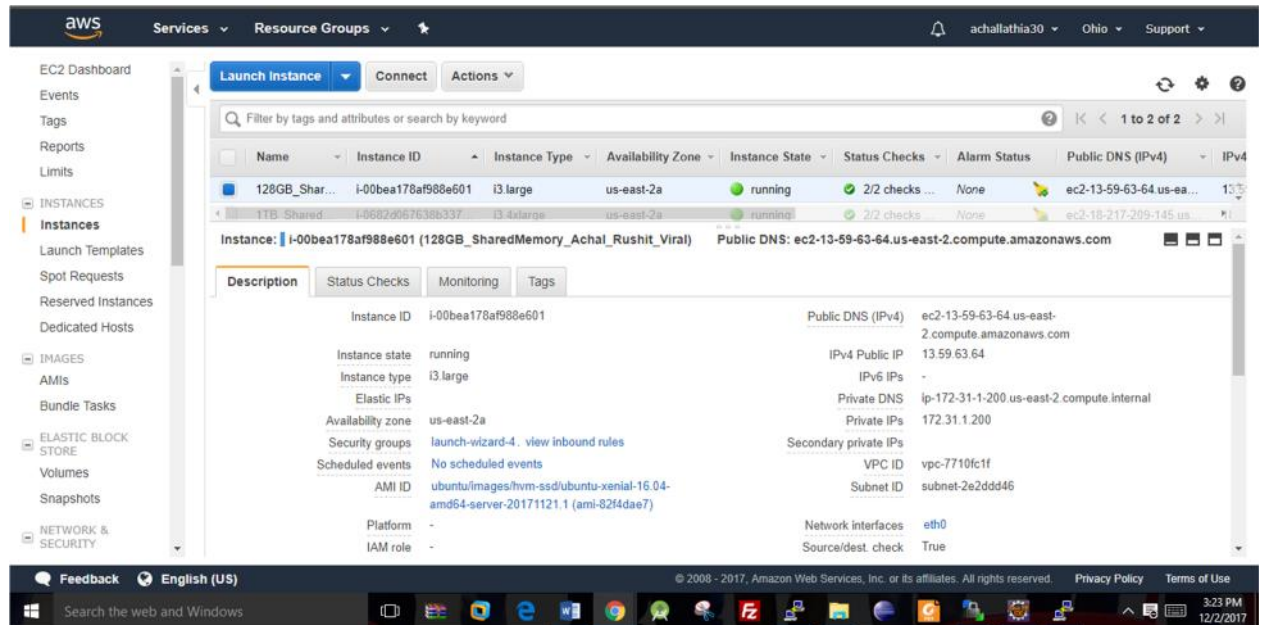
	<i>Spark</i>	<i>Hadoop</i>
<i>Size of Dataset</i>	100TB	102.5TB
<i># of Nodes</i>	207	2100
<i>Memory</i>	244GB	64GB
<i>SSD</i>	8x800GB	12x3TB
<i>Processor</i>	32 vCores - 2.5Ghz Intel Xeon E5-2670 v2	2 2.3Ghz hexcore Xeon E5-2630
<i>Author</i>	Databricks	Yahoo! Inc
<i>Computation Rate</i>	4.27TB/min	1.42TB/min

→Above table states sorting configuration and results on 100TB data for different framework Spark and Hadoop. It is clearly inferred that Spark benchmark was performed while considering more memory as spark computes in-memory, whereas hadoop benchmark was computed while keeping disk in mind. Also, computation rate for spark is more compared to hadoop. Hence, all our above arguments of supporting spark rather than hadoop for computation was true.

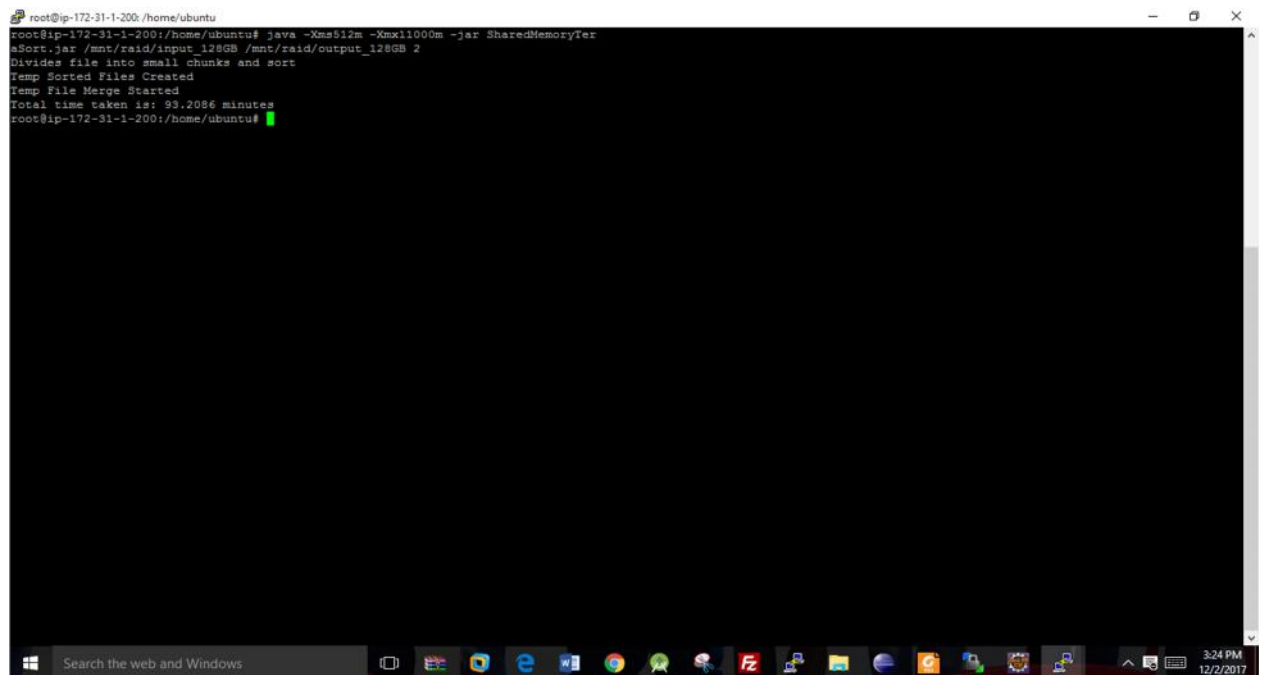
Execution Screenshots

Shared Memory

128GB Dataset



-Above screenshot is cluster information created for this experiment



-Above screenshot is command used to run the SharedMemoryTeraSort.jar file

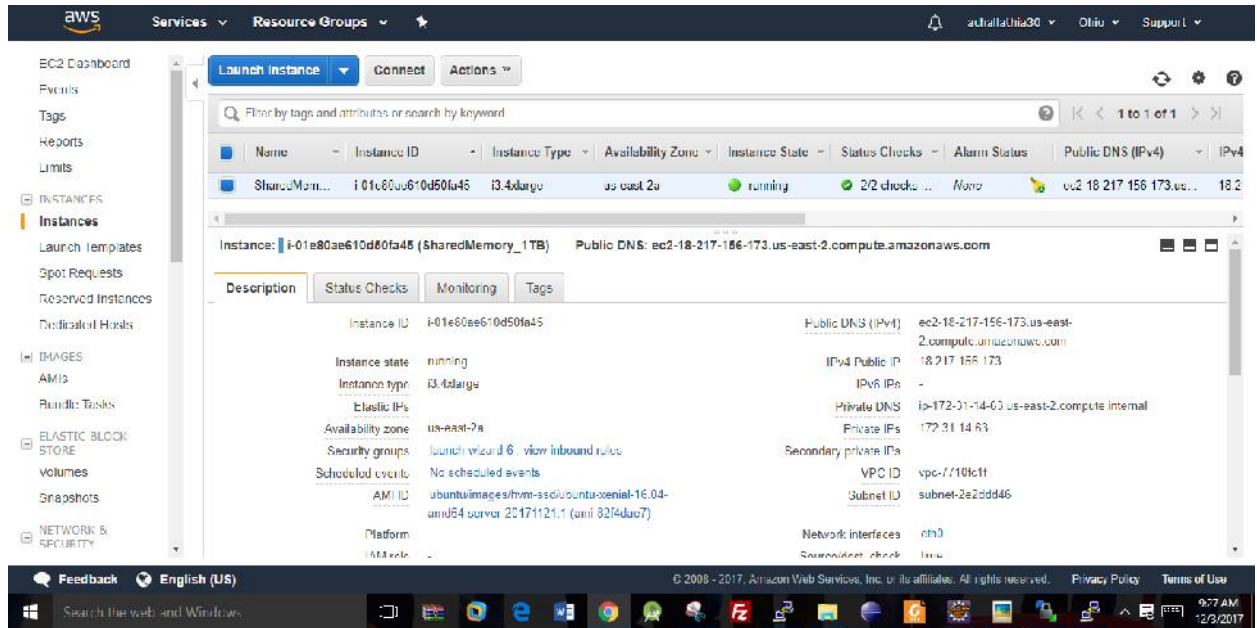
```
ubuntu@ip-172-31-1-200: ~  
total 254G  
-rwxr-xr-x 1 root root 128G Dec  2 18:50 input_128GB  
drwx----- 2 root root  16K Dec  2 18:10 lost+found  
-rw-r--r-- 1 root root 126G Dec  2 21:22 output_128GB  
drwxrwxrwx 2 root root  36K Dec  2 20:46 tmp  
ubuntu@ip-172-31-1-200:~$ ls -lh /mnt/raid/  
total 255G  
-rwxr-xr-x 1 root root 128G Dec  2 18:50 input_128GB  
drwx----- 2 root root  16K Dec  2 18:10 lost+found  
-rw-r--r-- 1 root root 127G Dec  2 21:22 output_128GB  
drwxrwxrwx 2 root root  36K Dec  2 20:46 tmp  
ubuntu@ip-172-31-1-200:~$ ls -lh /mnt/raid/  
total 255G  
-rwxr-xr-x 1 root root 128G Dec  2 18:50 input_128GB  
drwx----- 2 root root  16K Dec  2 18:10 lost+found  
-rw-r--r-- 1 root root 127G Dec  2 21:22 output_128GB  
drwxrwxrwx 2 root root  36K Dec  2 20:46 tmp  
ubuntu@ip-172-31-1-200:~$ ls -lh /mnt/raid/  
total 257G  
-rwxr-xr-x 1 root root 128G Dec  2 18:50 input_128GB  
drwx----- 2 root root  16K Dec  2 18:10 lost+found  
-rw-r--r-- 1 root root 128G Dec  2 21:23 output_128GB  
drwxrwxrwx 2 root root  36K Dec  2 21:23 tmp  
ubuntu@ip-172-31-1-200:~$
```

-Above screenshot is directory information at mnt/raid path

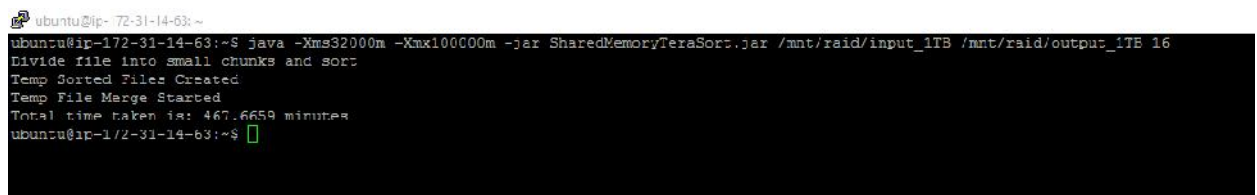
```
root@ip-172-31-1-200:/home/ubuntu# ./valsort /mnt/raid/output_128GB  
Records: 1374389534  
Checksum: 28f5d4900efdf3f5  
Duplicate keys: 0  
SUCCESS - all records are in order  
root@ip-172-31-1-200:/home/ubuntu#
```

-Above screenshot is command for running valsort

1TB Dataset



-Above screenshot is cluster information created for this experiment



-Above screenshot is command used to run java SharedMemoryTeraSort.jar file

```
ubuntu@ip-172-31-14-63:~$ ls -la /mnt/raid/output ITD /mnt/raid/output ITD 10
ls: cannot access /mnt/raid/output ITD: No such file or directory
ls: cannot access /mnt/raid/output ITD 10: No such file or directory
Total time taken is: 167.6650 minutes
ubuntu@ip-172-31-14-63:~$ ls -la /mnt/raid/output ITD
ls: cannot access /mnt/raid/output ITD: No such file or directory
ubuntu@ip-172-31-14-63:~$ ls -la /mnt/raid/
total 2.1T
-rwxrwxr-x 1 ubuntu ubuntu 1.0T Dec 3 04:00 input ITD
drwxr-xr-x 2 root root 16K Dec 3 09:50 lost+found
-rwxrwxr-x 1 ubuntu ubuntu 1.0T Dec 3 04:00 output ITD
-rwxrwxr-x 2 root root 16K Dec 3 04:00 output ITD
ubuntu@ip-172-31-14-63:~$
```

-Above screenshot is directory information at mnt/raid path

```
root@ip-172-31-14-63:/home/ubuntu#
Using username "ubuntu".
Authenticating with public key "imported-openssh-key".
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-1041-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

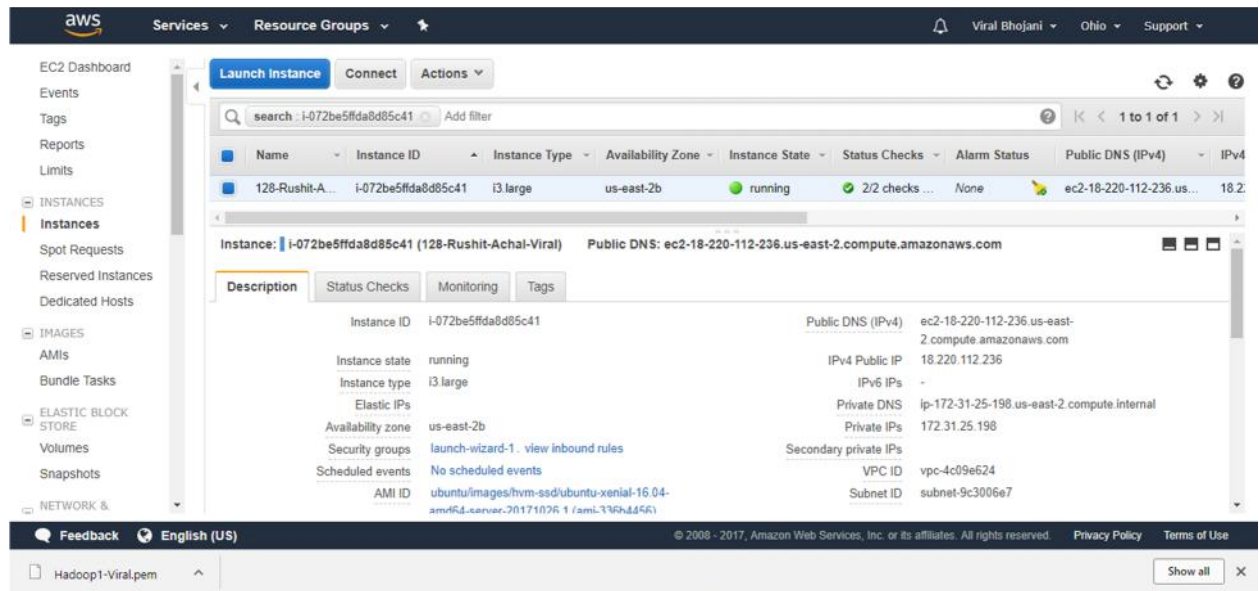
5 packages can be updated.
0 updates are security updates.

Last login: Sun Dec 3 15:04:57 2017 from 208.59.147.143
ubuntu@ip-172-31-14-63:~$ sudo su
root@ip-172-31-14-63:/home/ubuntu# ./valsort /mnt/raid/output ITD
Records: 10995116277
Checksum: 147aec58f464e1ae4
Duplicate keys: 0
SUCCESS - all records are in order
root@ip-172-31-14-63:/home/ubuntu#
```

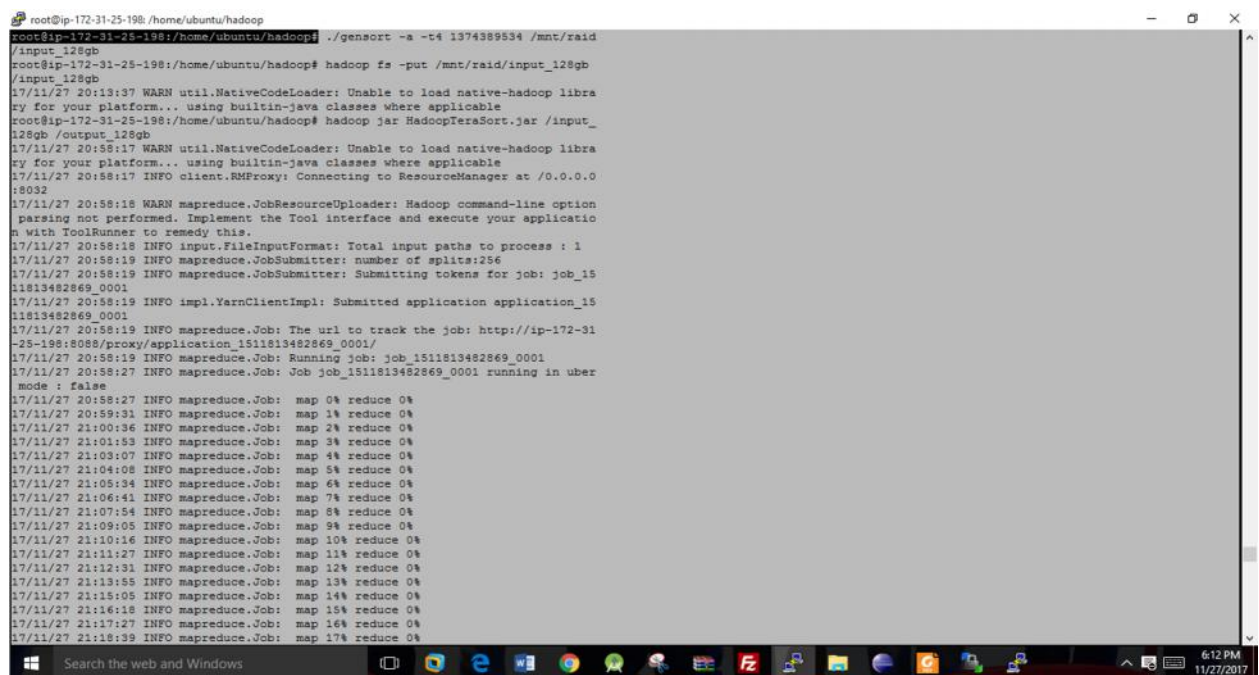
-Above screenshot is command for running valsort

Hadoop Single Node

128GB Dataset



-Above screenshot is cluster information used in this experiment



-Above screenshot is command used to run java HadoopTeraSort.jar file

```
root@ip-172-31-25-198: /home/ubuntu/hadoop
17/11/27 23:25:30 INFO mapreduce.Job: map 96% reduce 32%
17/11/27 23:27:04 INFO mapreduce.Job: map 97% reduce 32%
17/11/27 23:28:49 INFO mapreduce.Job: map 98% reduce 32%
17/11/27 23:30:31 INFO mapreduce.Job: map 99% reduce 32%
17/11/27 23:30:37 INFO mapreduce.Job: map 99% reduce 33%
17/11/27 23:32:44 INFO mapreduce.Job: map 100% reduce 33%
17/11/27 23:38:55 INFO mapreduce.Job: map 100% reduce 34%
17/11/27 23:39:04 INFO mapreduce.Job: map 100% reduce 35%
17/11/27 23:39:13 INFO mapreduce.Job: map 100% reduce 36%
17/11/27 23:39:22 INFO mapreduce.Job: map 100% reduce 37%
17/11/27 23:39:31 INFO mapreduce.Job: map 100% reduce 38%
17/11/27 23:39:37 INFO mapreduce.Job: map 100% reduce 39%
17/11/27 23:39:43 INFO mapreduce.Job: map 100% reduce 40%
17/11/27 23:39:49 INFO mapreduce.Job: map 100% reduce 41%
17/11/27 23:39:55 INFO mapreduce.Job: map 100% reduce 42%
17/11/27 23:39:58 INFO mapreduce.Job: map 100% reduce 43%
17/11/27 23:40:04 INFO mapreduce.Job: map 100% reduce 44%
17/11/27 23:40:10 INFO mapreduce.Job: map 100% reduce 45%
17/11/27 23:40:16 INFO mapreduce.Job: map 100% reduce 46%
17/11/27 23:40:22 INFO mapreduce.Job: map 100% reduce 47%
17/11/27 23:40:28 INFO mapreduce.Job: map 100% reduce 48%
17/11/27 23:40:34 INFO mapreduce.Job: map 100% reduce 49%
17/11/27 23:40:40 INFO mapreduce.Job: map 100% reduce 50%
17/11/27 23:40:46 INFO mapreduce.Job: map 100% reduce 51%
17/11/27 23:40:52 INFO mapreduce.Job: map 100% reduce 52%
17/11/27 23:40:58 INFO mapreduce.Job: map 100% reduce 53%
17/11/27 23:41:04 INFO mapreduce.Job: map 100% reduce 54%
17/11/27 23:41:10 INFO mapreduce.Job: map 100% reduce 55%
17/11/27 23:41:17 INFO mapreduce.Job: map 100% reduce 56%
17/11/27 23:41:23 INFO mapreduce.Job: map 100% reduce 57%
17/11/27 23:41:32 INFO mapreduce.Job: map 100% reduce 58%
17/11/27 23:41:47 INFO mapreduce.Job: map 100% reduce 59%
17/11/27 23:42:00 INFO mapreduce.Job: map 100% reduce 60%
17/11/27 23:42:14 INFO mapreduce.Job: map 100% reduce 61%
17/11/27 23:42:29 INFO mapreduce.Job: map 100% reduce 62%
17/11/27 23:42:44 INFO mapreduce.Job: map 100% reduce 63%
17/11/27 23:42:57 INFO mapreduce.Job: map 100% reduce 64%
17/11/27 23:43:12 INFO mapreduce.Job: map 100% reduce 65%
17/11/27 23:43:27 INFO mapreduce.Job: map 100% reduce 66%
17/11/27 23:43:42 INFO mapreduce.Job: map 100% reduce 67%
17/11/27 23:43:54 INFO mapreduce.Job: map 100% reduce 68%
17/11/27 23:44:22 INFO mapreduce.Job: map 100% reduce 69%
17/11/27 23:45:13 INFO mapreduce.Job: map 100% reduce 70%
17/11/27 23:46:07 INFO mapreduce.Job: map 100% reduce 71%
```

-Above screenshot is information of completion percentage of map and reduce class

```
root@ip-172-31-25-198: /home/ubuntu/hadoop
17/11/28 00:06:04 INFO mapreduce.Job: map 100% reduce 94%
17/11/28 00:06:57 INFO mapreduce.Job: map 100% reduce 95%
17/11/28 00:07:50 INFO mapreduce.Job: map 100% reduce 96%
17/11/28 00:08:42 INFO mapreduce.Job: map 100% reduce 97%
17/11/28 00:09:33 INFO mapreduce.Job: map 100% reduce 98%
17/11/28 00:10:27 INFO mapreduce.Job: map 100% reduce 99%
17/11/28 00:11:22 INFO mapreduce.Job: map 100% reduce 100%
17/11/28 00:11:52 INFO mapreduce.Job: Job job_1511815482869_0001 completed successfully
17/11/28 00:11:52 INFO mapreduce.Job: Counters: 50

File System Counters
  FILE: Number of bytes read=548702634280
  FILE: Number of bytes written=688916021242
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=137440022968
  HDFS: Number of bytes written=137438953400
  HDFS: Number of read operations=774
  HDFS: Number of large read operations=0
  HDFS: Number of write operations=4

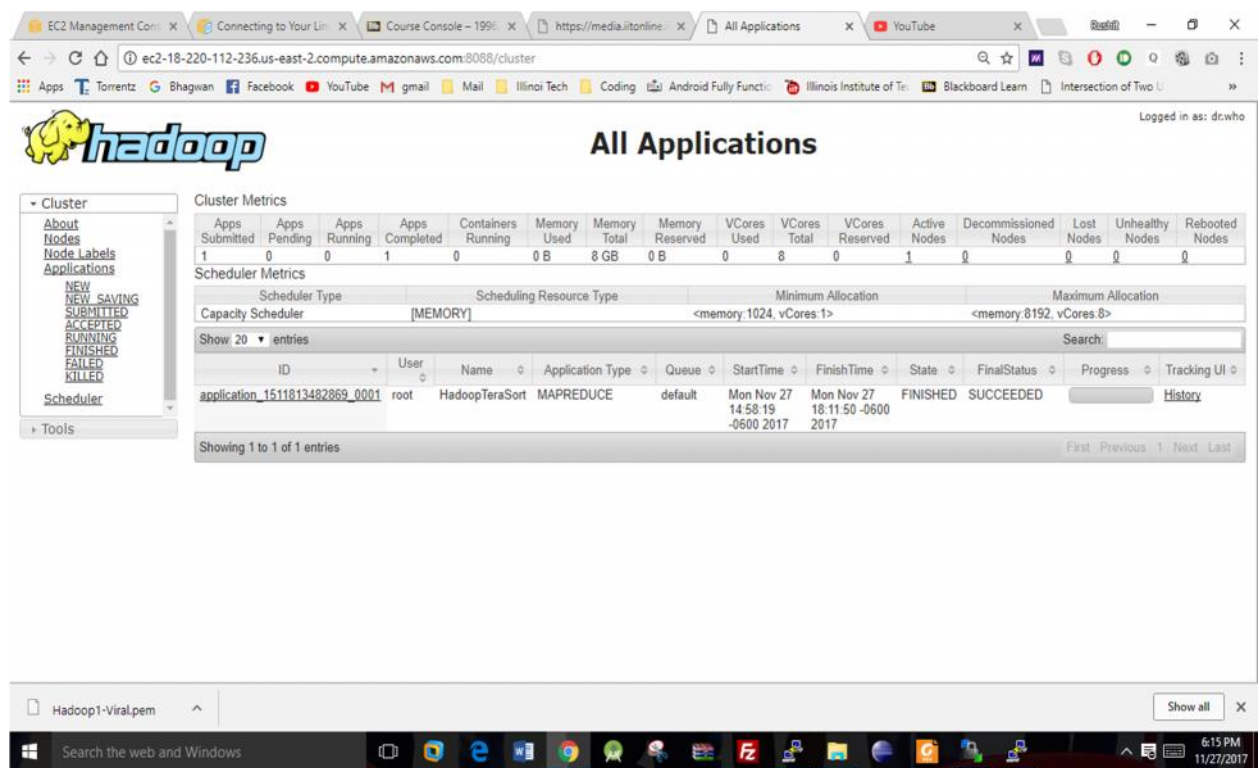
Job Counters
  Killed map tasks=2
  Launched map tasks=258
  Launched reduce tasks=2
  Data-local map tasks=258
  Total time spent by all maps in occupied slots (ms)=40888854
  Total time spent by all reduces in occupied slots (ms)=19057020
  Total time spent by all map tasks (ms)=40888854
  Total time spent by all reduce tasks (ms)=19057020
  Total vcore-seconds taken by all map tasks=40888854
  Total vcore-seconds taken by all reduce tasks=19057020
  Total megabyte-seconds taken by all map tasks=41870186496
  Total megabyte-seconds taken by all reduce tasks=19514388480

Map-Reduce Framework
  Map input records=1374389534
  Map output records=1374389534
  Map output bytes=137438953400
  Map output materialized bytes=140187735540
  Input split bytes=25088
  Combine input records=2748779068
  Combine output records=2748779068
  Reduce input groups=1374389534
  Reduce shuffle bytes=140187735540
  Reduce input records=1374389534
```

-Above screenshot is information after map reduce execution is completed

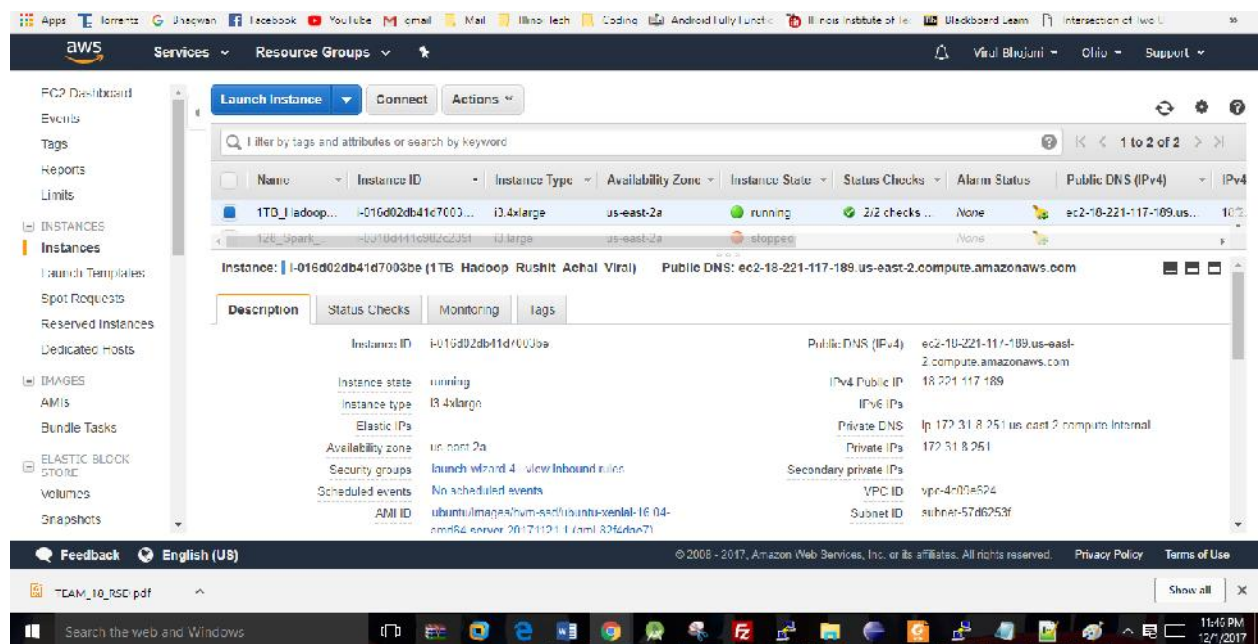
-Above screenshot is directory structure presented in Hadoop hdfs

-Above screenshot is command used to run valsart in both the part

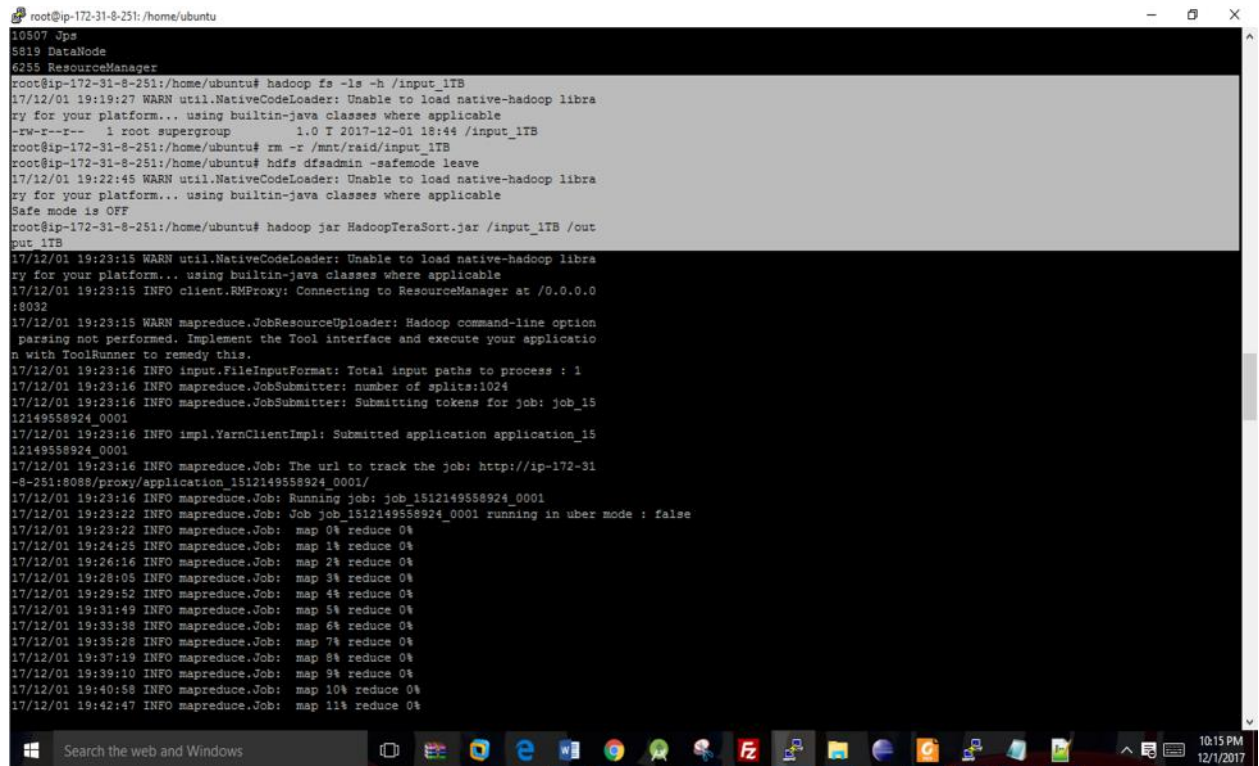


-Above screenshot is Hadoop completion report with time required to finish the experiment

1TB Dataset

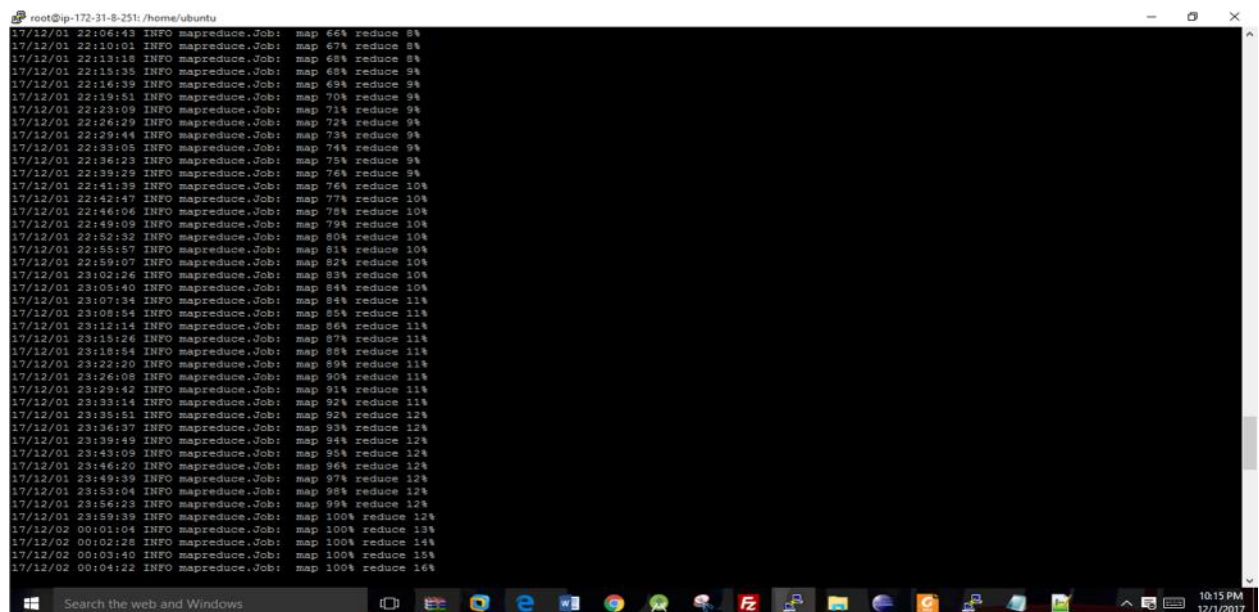


-Above screenshot is cluster information used in this experiment



```
root@ip-172-31-8-251:/home/ubuntu
10507 Jps
5819 DataNode
6255 ResourceManager
root@ip-172-31-8-251:/home/ubuntu# hadoop fs -ls -h /input_ITB
17/12/01 19:19:27 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
-rw-r--r-- 1 root supergroup 1.0 T 2017-12-01 18:44 /input_ITB
root@ip-172-31-8-251:/home/ubuntu# rm -r /mnt/raid/input_ITB
root@ip-172-31-8-251:/home/ubuntu# hdfs dfsadmin -safemode leave
17/12/01 19:22:45 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
Safe mode is OFF
root@ip-172-31-8-251:/home/ubuntu# hadoop jar HadoopTeraSort.jar /input_ITB /out
put_ITB
17/12/01 19:23:15 WARN util.NativeCodeLoader: Unable to load native-hadoop libra
ry for your platform... using builtin-java classes where applicable
17/12/01 19:23:15 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0
:8032
17/12/01 19:23:15 WARN mapreduce.JobResourceUploader: Hadoop command-line option
 parsing not performed. Implement the Tool interface and execute your applicatio
n with ToolRunner to remedy this.
17/12/01 19:23:16 INFO input.FileInputFormat: Total input paths to process : 1
17/12/01 19:23:16 INFO mapreduce.JobSubmitter: number of splits:1024
17/12/01 19:23:16 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_15
12149558924_0001
17/12/01 19:23:16 INFO impl.YarnClientImpl: Submitted application application_15
12149558924_0001
17/12/01 19:23:16 INFO mapreduce.Job: The url to track the job: http://ip-172-31
-8-251:8088/proxy/application_1512149558924_0001/
17/12/01 19:23:16 INFO mapreduce.Job: Running job: job_1512149558924_0001
17/12/01 19:23:22 INFO mapreduce.Job: Job job_1512149558924_0001 running in uber mode : false
17/12/01 19:23:22 INFO mapreduce.Job: map 0% reduce 0%
17/12/01 19:24:25 INFO mapreduce.Job: map 1% reduce 0%
17/12/01 19:26:16 INFO mapreduce.Job: map 2% reduce 0%
17/12/01 19:28:05 INFO mapreduce.Job: map 3% reduce 0%
17/12/01 19:29:52 INFO mapreduce.Job: map 4% reduce 0%
17/12/01 19:31:49 INFO mapreduce.Job: map 5% reduce 0%
17/12/01 19:33:38 INFO mapreduce.Job: map 6% reduce 0%
17/12/01 19:35:28 INFO mapreduce.Job: map 7% reduce 0%
17/12/01 19:37:19 INFO mapreduce.Job: map 8% reduce 0%
17/12/01 19:39:10 INFO mapreduce.Job: map 9% reduce 0%
17/12/01 19:40:58 INFO mapreduce.Job: map 10% reduce 0%
17/12/01 19:42:47 INFO mapreduce.Job: map 11% reduce 0%
```

-Above screenshot is command used to run java HadoopTeraSort.jar file



```
17/12/01 22:06:43 INFO mapreduce.Job: map 66% reduce 8%
17/12/01 22:10:01 INFO mapreduce.Job: map 67% reduce 8%
17/12/01 22:13:18 INFO mapreduce.Job: map 68% reduce 8%
17/12/01 22:15:35 INFO mapreduce.Job: map 68% reduce 9%
17/12/01 22:16:39 INFO mapreduce.Job: map 69% reduce 9%
17/12/01 22:19:51 INFO mapreduce.Job: map 70% reduce 9%
17/12/01 22:23:09 INFO mapreduce.Job: map 71% reduce 9%
17/12/01 22:26:29 INFO mapreduce.Job: map 72% reduce 9%
17/12/01 22:29:44 INFO mapreduce.Job: map 73% reduce 9%
17/12/01 22:33:05 INFO mapreduce.Job: map 74% reduce 9%
17/12/01 22:36:23 INFO mapreduce.Job: map 75% reduce 9%
17/12/01 22:39:29 INFO mapreduce.Job: map 76% reduce 9%
17/12/01 22:41:39 INFO mapreduce.Job: map 76% reduce 10%
17/12/01 22:42:47 INFO mapreduce.Job: map 77% reduce 10%
17/12/01 22:46:06 INFO mapreduce.Job: map 78% reduce 10%
17/12/01 22:49:09 INFO mapreduce.Job: map 79% reduce 10%
17/12/01 22:52:32 INFO mapreduce.Job: map 80% reduce 10%
17/12/01 22:55:57 INFO mapreduce.Job: map 81% reduce 10%
17/12/01 22:59:07 INFO mapreduce.Job: map 82% reduce 10%
17/12/01 23:02:26 INFO mapreduce.Job: map 83% reduce 10%
17/12/01 23:05:40 INFO mapreduce.Job: map 84% reduce 10%
17/12/01 23:07:34 INFO mapreduce.Job: map 84% reduce 11%
17/12/01 23:08:54 INFO mapreduce.Job: map 85% reduce 11%
17/12/01 23:12:14 INFO mapreduce.Job: map 86% reduce 11%
17/12/01 23:15:26 INFO mapreduce.Job: map 87% reduce 11%
17/12/01 23:18:54 INFO mapreduce.Job: map 88% reduce 11%
17/12/01 23:22:20 INFO mapreduce.Job: map 89% reduce 11%
17/12/01 23:26:08 INFO mapreduce.Job: map 90% reduce 11%
17/12/01 23:29:42 INFO mapreduce.Job: map 91% reduce 11%
17/12/01 23:33:14 INFO mapreduce.Job: map 92% reduce 11%
17/12/01 23:35:51 INFO mapreduce.Job: map 92% reduce 12%
17/12/01 23:36:37 INFO mapreduce.Job: map 93% reduce 12%
17/12/01 23:39:49 INFO mapreduce.Job: map 94% reduce 12%
17/12/01 23:43:09 INFO mapreduce.Job: map 95% reduce 12%
17/12/01 23:46:20 INFO mapreduce.Job: map 96% reduce 12%
17/12/01 23:49:39 INFO mapreduce.Job: map 97% reduce 12%
17/12/01 23:53:04 INFO mapreduce.Job: map 98% reduce 12%
17/12/01 23:56:23 INFO mapreduce.Job: map 99% reduce 12%
17/12/01 23:59:39 INFO mapreduce.Job: map 100% reduce 12%
17/12/02 00:01:04 INFO mapreduce.Job: map 100% reduce 13%
17/12/02 00:02:25 INFO mapreduce.Job: map 100% reduce 14%
17/12/02 00:03:40 INFO mapreduce.Job: map 100% reduce 15%
17/12/02 00:04:22 INFO mapreduce.Job: map 100% reduce 16%
```

-Above screenshot is information of completion percentage of map and reduce class

```
root@ip-172-31-8-251: /home/ubuntu
17/12/02 04:04:14 INFO mapreduce.Job: map 100% reduce 96%
17/12/02 04:07:27 INFO mapreduce.Job: map 100% reduce 97%
17/12/02 04:10:41 INFO mapreduce.Job: map 100% reduce 98%
17/12/02 04:13:57 INFO mapreduce.Job: map 100% reduce 99%
17/12/02 04:17:05 INFO mapreduce.Job: map 100% reduce 100%
17/12/02 04:19:04 INFO mapreduce.Job: Job job_1512149559324_0001 completed successfully
17/12/02 04:19:04 INFO mapreduce.Job: Counters: 50

File System Counters
  FILE: Number of bytes read=6613066547876
  FILE: Number of bytes written=7734468485854
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=10995116277
  HDFS: Number of bytes written=1099511627700
  HDFS: Number of read operations=3096
  HDFS: Number of large read operations=0
  HDFS: Number of write operations=16

Job Counters
  Killed reduce tasks=1
  Launched map tasks=1024
  Launched reduce tasks=9
  Data-local map tasks=1024
  Total time spent by all maps in occupied slots (ms)=60833914
  Total time spent by all reduces in occupied slots (ms)=124810477
  Total time spent by all map tasks (ms)=60833914
  Total time spent by all reduce tasks (ms)=124810477
  Total vcore-seconds taken by all map tasks=60833914
  Total vcore-seconds taken by all reduce tasks=124810477
  Total megabyte-seconds taken by all map tasks=62293927936
  Total megabyte-seconds taken by all reduce tasks=127805928448

Map-Reduce Framework
  Map input records=10995116277
  Map output records=10995116277
  Map output bytes=1099511627700
  Map output materialized bytes=1121501909406
  Input split bytes=98304
  Combine input records=21990232554
  Combine output records=21990232554
  Reduce input groups=10995116277
  Reduce shuffle bytes=1121501909406
  Reduce input records=10995116277
  Reduce output records=10995116277
```

-Above screenshot is information once map reduce execution is completed

```
root@ip-172-31-8-251: /home/ubuntu
  Map output records=10995116277
  Map output bytes=1099511627700
  Map output materialized bytes=1121501909406
  Input split bytes=98304
  Combine input records=21990232554
  Combine output records=21990232554
  Reduce input groups=10995116277
  Reduce shuffle bytes=1121501909406
  Reduce input records=10995116277
  Reduce output records=10995116277
  Spilled Records=75826938962
  Shuffled Maps =8192
  Failed Shuffles=0
  Merged Map outputs=8192
  GC time elapsed (ms)=3762993
  CPU time spent (ms)=137483800
  Physical memory (bytes) snapshot=312283975680
  Virtual memory (bytes) snapshot=2081336115200
  Total committed heap usage (bytes)=213335408640

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=1099515817908
  File Output Format Counters
  Bytes Written=1099511627700

root@ip-172-31-8-251:/home/ubuntu# hadoop fs -ls -h /output_1TB
17/12/02 04:20:27 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 9 items
-rw-r--r-- 1 root supergroup          0 2017-12-02 04:19 /output_1TB/_SUCCESS
-rw-r--r-- 1 root supergroup    128.0 G 2017-12-02 01:26 /output_1TB/part-r-000000
-rw-r--r-- 1 root supergroup    128.0 G 2017-12-02 01:23 /output_1TB/part-r-000001
-rw-r--r-- 1 root supergroup    128.0 G 2017-12-02 01:24 /output_1TB/part-r-000002
-rw-r--r-- 1 root supergroup    128.0 G 2017-12-02 03:46 /output_1TB/part-r-000003
-rw-r--r-- 1 root supergroup    128.0 G 2017-12-02 03:47 /output_1TB/part-r-000004
-rw-r--r-- 1 root supergroup    128.0 G 2017-12-02 03:49 /output_1TB/part-r-000005
-rw-r--r-- 1 root supergroup    128.0 G 2017-12-02 04:18 /output_1TB/part-r-000006
-rw-r--r-- 1 root supergroup    128.0 G 2017-12-02 04:19 /output_1TB/part-r-000007
root@ip-172-31-8-251:/home/ubuntu#
```

-Above screenshot is directory information in Hadoop hdfs


```
root@ip-172-31-8-251:/home/ubuntu# ./valsort /mnt/raid/output_1TB/part-r-00000
Records: 1374354514
Checksum: 28f5b88d38567053
Duplicate keys: 0
SUCCESS - all records are in order
```

```
ubuntu@ip-172-31-8-251:~$ sudo su
root@ip-172-31-8-251:/home/ubuntu# ./valsort /mnt/raid/output_1TB/part-r-00001
Records: 1374307037
Checksum: 28f510cf82b2b115
Duplicate keys: 0
SUCCESS - all records are in order
```

```
root@ip-172-31-8-251:/home/ubuntu# ./valsort /mnt/raid/output_1TB/part-r-00002
Records: 1374360153
Checksum: 28f59f9cf96c9a9d
Duplicate keys: 0
SUCCESS - all records are in order
```

```
root@ip-172-31-8-251:/home/ubuntu# ./valsort /mnt/raid/output_1TB/part-r-00003
Records: 1374414727
Checksum: 28f60651892253bb
Duplicate keys: 0
SUCCESS - all records are in order
```

```
root@ip-172-31-8-251:/home/ubuntu# ./valsort /mnt/raid/output_1TB/part-r-00004
Records: 1374487578
Checksum: 28f6bd3585974edc
Duplicate keys: 0
SUCCESS - all records are in order
root@ip-172-31-8-251:/home/ubuntu# ./valsort /mnt/raid/output_1TB/part-r-00005
^[[ARecords: 1374369130
Checksum: 28f5ce3e37d6e67b
Duplicate keys: 0
SUCCESS - all records are in order
```

```
root@ip-172-31-8-251:/home/ubuntu# ./valsort /mnt/raid/output_1TB/part-r-00006
Records: 1374414956
Checksum: 28f60c56bd8e62de
Duplicate keys: 0
SUCCESS - all records are in order
```

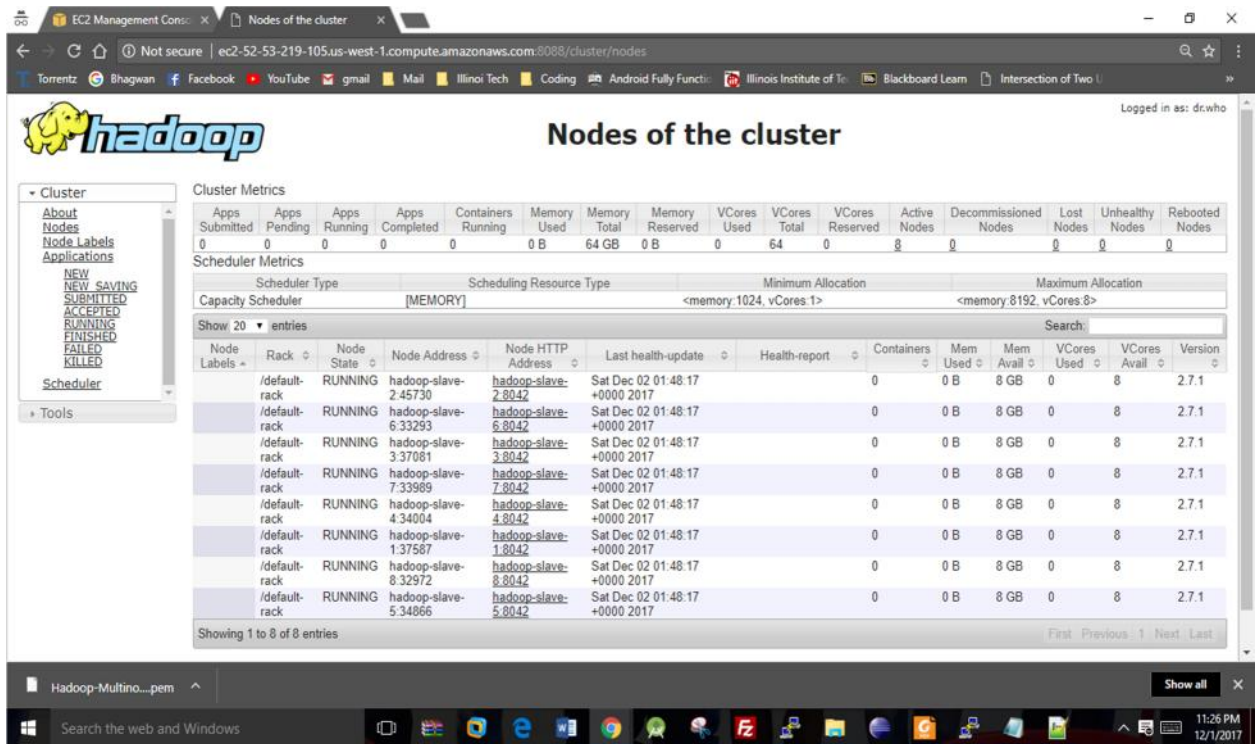
```
root@ip-172-31-8-251:/home/ubuntu# ./valsort /mnt/raid/output_1TB/part-r-00007
Records: 1374408182
Checksum: 28f5beffb6b4cd4e
Duplicate keys: 0
SUCCESS - all records are in order
```

-Above screenshots are commands used to run valsort in all 8 parts

Rushit Pandya
Achal Lathia
Viral Bhojani

PA-2-REPORT

A20381916
A20375208
A20380100



The screenshot displays the 'Nodes of the cluster' page in the AWS Management Console. The page shows the Hadoop logo and the title 'Nodes of the cluster'. On the left, there is a sidebar with navigation links: About, Nodes, Node Labels, Applications, NEW, NEW SAVING, SUBMITTED, ACCEPTED, RUNNING, FINISHED, FAILED, KILLED, and Scheduler. The main content area is divided into two sections: Cluster Metrics and Scheduler Metrics.

Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Apps Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	VCores Used	VCores Total	VCores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebooted Nodes
0	0	0	0	0	0 B	64 GB	0 B	0	64	0	8	0	0	0	0

Scheduler Metrics

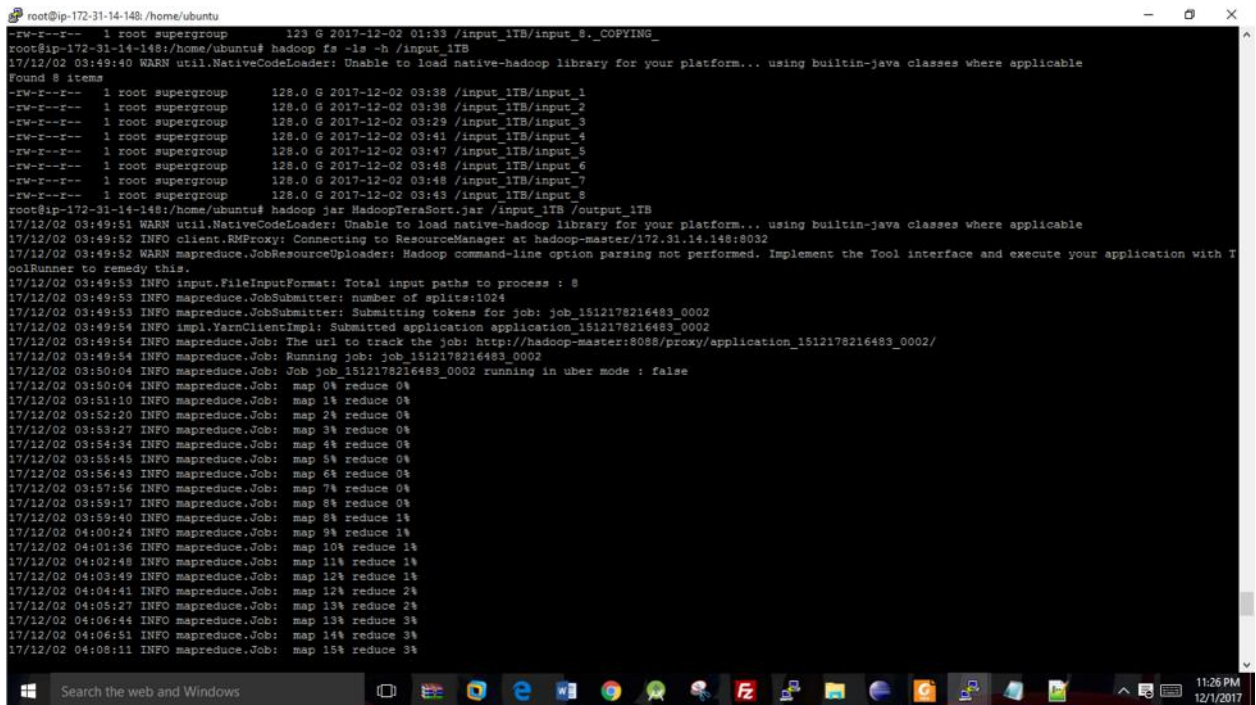
Scheduler Type		Scheduling Resource Type		Minimum Allocation		Maximum Allocation	
Capacity Scheduler	[MEMORY]	<memory:1024, vCores:1>	<memory:8192, vCores:8>				
Showing 1 to 8 of 8 entries							

Nodes Table

Node Labels	Rack	Node State	Node Address	Node HTTP Address	Last health-update	Health-report	Containers	Mem Used	Mem Avail	VCores Used	VCores Avail	Version
/default-rack		RUNNING	hadoop-slave-2.45730	hadoop-slave-2.8042	Sat Dec 02 01:48:17 +0000 2017		0	0 B	8 GB	0	8	2.7.1
/default-rack		RUNNING	hadoop-slave-6.33293	hadoop-slave-6.8042	Sat Dec 02 01:48:17 +0000 2017		0	0 B	8 GB	0	8	2.7.1
/default-rack		RUNNING	hadoop-slave-3.37081	hadoop-slave-3.8042	Sat Dec 02 01:48:17 +0000 2017		0	0 B	8 GB	0	8	2.7.1
/default-rack		RUNNING	hadoop-slave-7.33989	hadoop-slave-7.8042	Sat Dec 02 01:48:17 +0000 2017		0	0 B	8 GB	0	8	2.7.1
/default-rack		RUNNING	hadoop-slave-4.34004	hadoop-slave-4.8042	Sat Dec 02 01:48:17 +0000 2017		0	0 B	8 GB	0	8	2.7.1
/default-rack		RUNNING	hadoop-slave-1.37587	hadoop-slave-1.8042	Sat Dec 02 01:48:17 +0000 2017		0	0 B	8 GB	0	8	2.7.1
/default-rack		RUNNING	hadoop-slave-8.32972	hadoop-slave-8.8042	Sat Dec 02 01:48:17 +0000 2017		0	0 B	8 GB	0	8	2.7.1
/default-rack		RUNNING	hadoop-slave-5.34866	hadoop-slave-5.8042	Sat Dec 02 01:48:17 +0000 2017		0	0 B	8 GB	0	8	2.7.1

Showing 1 to 8 of 8 entries

-Above screenshot is information of nodes in this cluster



The screenshot shows a terminal window with the following commands and output:

```
root@ip-172-31-14-148:/home/ubuntu# hadoop fs -ls -h /input_1TB
17/12/02 03:49:40 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 8 items
-rw-r--r-- 1 root supergroup 128.0 G 2017-12-02 03:38 /input_1TB/input_1
-rw-r--r-- 1 root supergroup 128.0 G 2017-12-02 03:38 /input_1TB/input_2
-rw-r--r-- 1 root supergroup 128.0 G 2017-12-02 03:29 /input_1TB/input_3
-rw-r--r-- 1 root supergroup 128.0 G 2017-12-02 03:41 /input_1TB/input_4
-rw-r--r-- 1 root supergroup 128.0 G 2017-12-02 03:47 /input_1TB/input_5
-rw-r--r-- 1 root supergroup 128.0 G 2017-12-02 03:48 /input_1TB/input_6
-rw-r--r-- 1 root supergroup 128.0 G 2017-12-02 03:48 /input_1TB/input_7
-rw-r--r-- 1 root supergroup 128.0 G 2017-12-02 03:43 /input_1TB/input_8
root@ip-172-31-14-148:/home/ubuntu# hadoop jar HadoopTeraSort.jar /input_1TB/output_1TB
17/12/02 03:49:51 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
17/12/02 03:49:52 INFO client.RMProxy: Connecting to ResourceManager at hadoop-master/172.31.14.148:8032
17/12/02 03:49:52 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
17/12/02 03:49:53 INFO input.FileInputFormat: Total input paths to process : 8
17/12/02 03:49:53 INFO mapreduce.JobSubmitter: number of splits:1024
17/12/02 03:49:53 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1512178216483_0002
17/12/02 03:49:54 INFO impl.YarnClientImpl: Submitted application application_1512178216483_0002
17/12/02 03:49:54 INFO mapreduce.Job: The url to track the job: http://hadoop-master:8088/proxy/application_1512178216483_0002/
17/12/02 03:50:04 INFO mapreduce.Job: Job job_1512178216483_0002 running in uber mode : false
17/12/02 03:50:04 INFO mapreduce.Job: map 0% reduce 0%
17/12/02 03:51:10 INFO mapreduce.Job: map 1% reduce 0%
17/12/02 03:52:20 INFO mapreduce.Job: map 2% reduce 0%
17/12/02 03:53:27 INFO mapreduce.Job: map 3% reduce 0%
17/12/02 03:54:34 INFO mapreduce.Job: map 4% reduce 0%
17/12/02 03:55:45 INFO mapreduce.Job: map 5% reduce 0%
17/12/02 03:56:43 INFO mapreduce.Job: map 6% reduce 0%
17/12/02 03:57:56 INFO mapreduce.Job: map 7% reduce 0%
17/12/02 03:59:17 INFO mapreduce.Job: map 8% reduce 0%
17/12/02 03:59:40 INFO mapreduce.Job: map 8% reduce 1%
17/12/02 04:00:24 INFO mapreduce.Job: map 9% reduce 1%
17/12/02 04:01:36 INFO mapreduce.Job: map 10% reduce 1%
17/12/02 04:02:48 INFO mapreduce.Job: map 11% reduce 1%
17/12/02 04:03:49 INFO mapreduce.Job: map 12% reduce 1%
17/12/02 04:04:41 INFO mapreduce.Job: map 12% reduce 2%
17/12/02 04:05:27 INFO mapreduce.Job: map 13% reduce 2%
17/12/02 04:06:44 INFO mapreduce.Job: map 13% reduce 3%
17/12/02 04:06:51 INFO mapreduce.Job: map 14% reduce 3%
17/12/02 04:08:11 INFO mapreduce.Job: map 15% reduce 3%
```

-Above screenshot is command used to run java HadoopTeraSort.jar file

Rushit Pandya
Achal Lathia
Viral Bhojani

PA-2-REPORT

A20381916
A20375208
A20380100

```
root@ip-172-31-14-148: /home/ubuntu
17/12/02 04:18:26 INFO mapreduce.Job: map 22% reduce 7%
17/12/02 04:18:54 INFO mapreduce.Job: map 23% reduce 7%
17/12/02 04:20:20 INFO mapreduce.Job: map 24% reduce 7%
17/12/02 04:21:34 INFO mapreduce.Job: map 25% reduce 7%
17/12/02 04:22:40 INFO mapreduce.Job: map 25% reduce 8%
17/12/02 04:22:57 INFO mapreduce.Job: map 26% reduce 8%
17/12/02 04:24:15 INFO mapreduce.Job: map 27% reduce 8%
17/12/02 04:25:43 INFO mapreduce.Job: map 28% reduce 8%
17/12/02 04:27:03 INFO mapreduce.Job: map 28% reduce 9%
17/12/02 04:27:14 INFO mapreduce.Job: map 29% reduce 9%
17/12/02 04:28:38 INFO mapreduce.Job: map 30% reduce 9%
17/12/02 04:30:02 INFO mapreduce.Job: map 31% reduce 9%
17/12/02 04:31:20 INFO mapreduce.Job: map 32% reduce 9%
17/12/02 04:32:23 INFO mapreduce.Job: map 32% reduce 10%
17/12/02 04:32:53 INFO mapreduce.Job: map 33% reduce 10%
17/12/02 04:34:31 INFO mapreduce.Job: map 34% reduce 10%
17/12/02 04:35:59 INFO mapreduce.Job: map 35% reduce 10%
17/12/02 04:37:25 INFO mapreduce.Job: map 35% reduce 11%
17/12/02 04:37:26 INFO mapreduce.Job: map 36% reduce 11%
17/12/02 04:38:15 INFO mapreduce.Job: map 37% reduce 11%
17/12/02 04:40:21 INFO mapreduce.Job: map 38% reduce 11%
17/12/02 04:41:10 INFO mapreduce.Job: map 38% reduce 12%
17/12/02 04:41:50 INFO mapreduce.Job: map 39% reduce 12%
17/12/02 04:43:13 INFO mapreduce.Job: map 40% reduce 12%
17/12/02 04:44:35 INFO mapreduce.Job: map 41% reduce 12%
17/12/02 04:46:04 INFO mapreduce.Job: map 42% reduce 12%
17/12/02 04:46:06 INFO mapreduce.Job: map 42% reduce 13%
17/12/02 04:47:40 INFO mapreduce.Job: map 43% reduce 13%
17/12/02 04:49:20 INFO mapreduce.Job: map 44% reduce 13%
17/12/02 04:50:54 INFO mapreduce.Job: map 45% reduce 13%
17/12/02 04:51:03 INFO mapreduce.Job: map 45% reduce 14%
17/12/02 04:52:19 INFO mapreduce.Job: map 46% reduce 14%
17/12/02 04:54:02 INFO mapreduce.Job: map 47% reduce 14%
17/12/02 04:55:31 INFO mapreduce.Job: map 47% reduce 15%
17/12/02 04:55:38 INFO mapreduce.Job: map 48% reduce 15%
17/12/02 04:57:15 INFO mapreduce.Job: map 49% reduce 15%
17/12/02 04:58:42 INFO mapreduce.Job: map 50% reduce 15%
17/12/02 04:59:58 INFO mapreduce.Job: map 50% reduce 16%
17/12/02 05:00:19 INFO mapreduce.Job: map 51% reduce 16%
17/12/02 05:01:49 INFO mapreduce.Job: map 52% reduce 16%
17/12/02 05:03:20 INFO mapreduce.Job: map 53% reduce 16%
17/12/02 05:04:55 INFO mapreduce.Job: map 54% reduce 16%
17/12/02 05:05:03 INFO mapreduce.Job: map 54% reduce 17%
```

-Above screenshot is information of completion percentage of map and reduce class

```
root@ip-172-31-14-148: /home/ubuntu
17/12/02 07:37:23 INFO mapreduce.Job: map 100% reduce 76%
17/12/02 07:38:52 INFO mapreduce.Job: map 100% reduce 77%
17/12/02 07:40:23 INFO mapreduce.Job: map 100% reduce 78%
17/12/02 07:42:26 INFO mapreduce.Job: map 100% reduce 79%
17/12/02 07:44:44 INFO mapreduce.Job: map 100% reduce 80%
17/12/02 07:47:16 INFO mapreduce.Job: map 100% reduce 81%
17/12/02 07:50:28 INFO mapreduce.Job: map 100% reduce 82%
17/12/02 07:53:35 INFO mapreduce.Job: map 100% reduce 83%
17/12/02 07:57:44 INFO mapreduce.Job: map 100% reduce 84%
17/12/02 08:02:12 INFO mapreduce.Job: map 100% reduce 85%
17/12/02 08:06:31 INFO mapreduce.Job: map 100% reduce 86%
17/12/02 08:10:20 INFO mapreduce.Job: map 100% reduce 87%
17/12/02 08:14:07 INFO mapreduce.Job: map 100% reduce 88%
17/12/02 08:17:55 INFO mapreduce.Job: map 100% reduce 89%
17/12/02 08:21:41 INFO mapreduce.Job: map 100% reduce 90%
17/12/02 08:25:35 INFO mapreduce.Job: map 100% reduce 91%
17/12/02 08:29:29 INFO mapreduce.Job: map 100% reduce 92%
17/12/02 08:33:23 INFO mapreduce.Job: map 100% reduce 93%
17/12/02 08:37:17 INFO mapreduce.Job: map 100% reduce 94%
17/12/02 08:41:11 INFO mapreduce.Job: map 100% reduce 95%
17/12/02 08:45:04 INFO mapreduce.Job: map 100% reduce 96%
17/12/02 08:48:58 INFO mapreduce.Job: map 100% reduce 97%
17/12/02 08:52:53 INFO mapreduce.Job: map 100% reduce 98%
17/12/02 08:56:43 INFO mapreduce.Job: map 100% reduce 99%
17/12/02 09:01:53 INFO mapreduce.Job: map 100% reduce 100%
17/12/02 09:05:34 INFO mapreduce.Job: Job job_1512178216483_0002 completed successfully
17/12/02 09:05:34 INFO mapreduce.Job: Counters: 51

File System Counters
  FILE: Number of bytes read=5271741910298
  FILE: Number of bytes written=6393142460020
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=1099515899328
  HDFS: Number of bytes written=1099511627200
  HDFS: Number of read operations=3096
  HDFS: Number of large read operations=0
  HDFS: Number of write operations=16

Job Counters
  Killed map tasks=2
  Launched map tasks=1026
  Launched reduce tasks=8
  Data-local map tasks=923
```

-Above screenshot is information once map reduce execution is completed

```
root@ip-172-31-14-148: /home/ubuntu
Map output records=10995116272
Map output bytes=1099511627200
Map output materialized bytes=1121501908896
Input split bytes=110592
Combine input records=21990232544
Combine output records=21990232544
Reduce input groups=1374989534
Reduce shuffle bytes=1121501908896
Reduce input records=10995116272
Reduce output records=10995116272
Spilled Records=62676685031
Shuffled Maps =8192
Failed Shuffles=0
Merged Map outputs=8192
GC time elapsed (ms)=2547368
CPU time spent (ms)=146269150
Physical memory (bytes) snapshot=283766001664
Virtual memory (bytes) snapshot=2027155341312
Total committed heap usage (bytes)=197191532544

Shuffle Errors
BAD ID=0
CONNECTION=0
IO ERROR=0
WRONG LENGTH=0
WRONG MAP=0
WRONG REDUCE=0

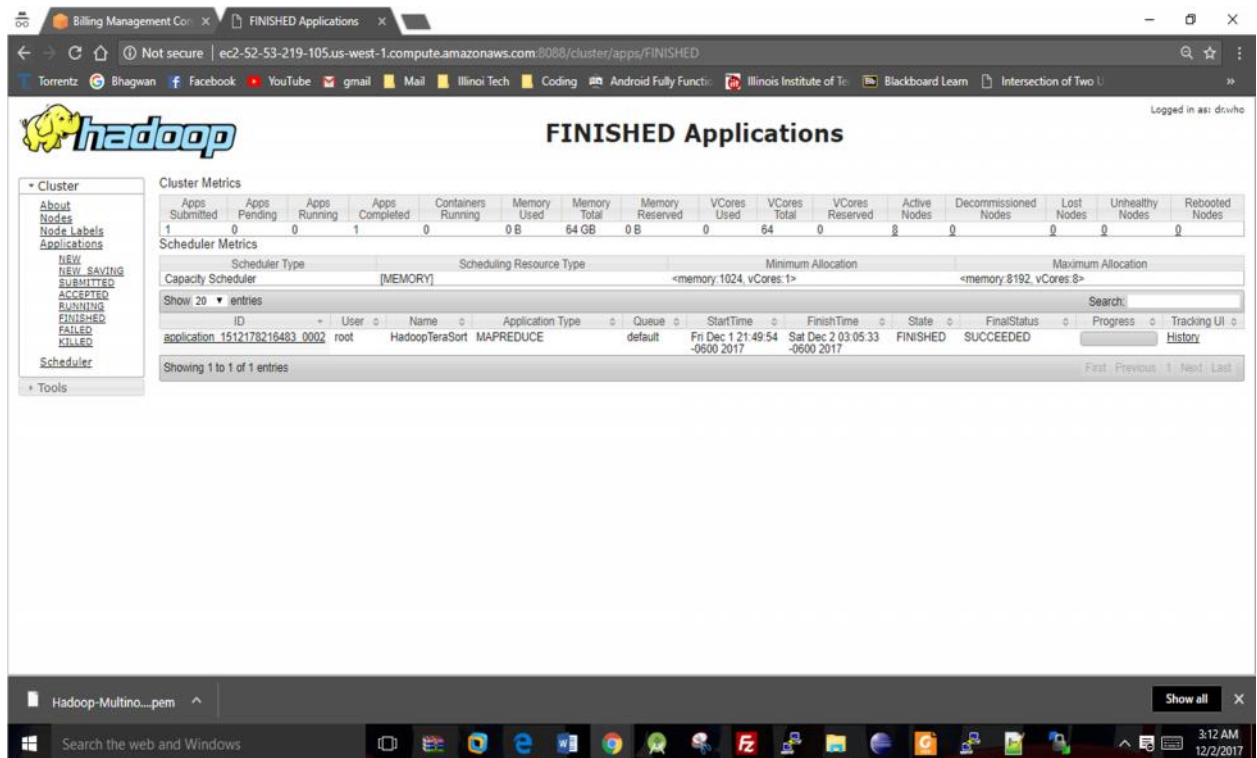
File Input Format Counters
Bytes Read=1099515788736
File Output Format Counters
Bytes Written=1099511627200
root@ip-172-31-14-148:/home/ubuntu# hadoop fs -ls -h /output_1TB
17/12/02 09:06:55 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 9 items
-rw-r--r-- 1 root supergroup 0 2017-12-02 09:05 /output_1TB/_SUCCESS
-rw-r--r-- 1 root supergroup 128.0 G 2017-12-02 08:57 /output_1TB/part-r-00000
-rw-r--r-- 1 root supergroup 128.0 G 2017-12-02 09:05 /output_1TB/part-r-00001
-rw-r--r-- 1 root supergroup 128.0 G 2017-12-02 08:59 /output_1TB/part-r-00002
-rw-r--r-- 1 root supergroup 128.0 G 2017-12-02 07:55 /output_1TB/part-r-00003
-rw-r--r-- 1 root supergroup 128.0 G 2017-12-02 07:40 /output_1TB/part-r-00004
-rw-r--r-- 1 root supergroup 128.0 G 2017-12-02 07:41 /output_1TB/part-r-00005
-rw-r--r-- 1 root supergroup 128.0 G 2017-12-02 09:05 /output_1TB/part-r-00006
-rw-r--r-- 1 root supergroup 128.0 G 2017-12-02 07:46 /output_1TB/part-r-00007
root@ip-172-31-14-148:/home/ubuntu#
```

-Above screenshot is directory structure in Hadoop hdfs

```
root@ip-172-31-14-148:/home/ubuntu# drwxr-xr-x 3 root root 4.0K Dec 1 21:13 input_1TB
drwxr-xr-x 2 root root 16K Dec 1 20:41 lost+found
-rw-r--r-- 1 root root 128G Dec 2 09:50 part-r-00000
-rw-r--r-- 1 root root 128G Dec 2 09:51 part-r-00007._COPYING_
drwxr-xr-x 3 root root 4.0K Dec 1 20:57 tmp
root@ip-172-31-14-148:/home/ubuntu# ls -lh /mnt/raid
total 382G
drwxr-xr-x 3 root root 4.0K Dec 1 20:57 dfs
-rw-r--r-- 1 root root 128G Dec 1 21:13 input_1TB
drwxr-xr-x 2 root root 16K Dec 1 20:41 lost+found
-rw-r--r-- 1 root root 128G Dec 2 09:50 part-r-00000
-rw-r--r-- 1 root root 128G Dec 2 09:51 part-r-00007._COPYING_
drwxr-xr-x 3 root root 4.0K Dec 1 20:57 tmp
root@ip-172-31-14-148:/home/ubuntu# ls -lh /mnt/raid
total 382G
drwxr-xr-x 3 root root 4.0K Dec 1 20:57 dfs
-rw-r--r-- 1 root root 128G Dec 1 21:13 input_1TB
drwxr-xr-x 2 root root 16K Dec 1 20:41 lost+found
-rw-r--r-- 1 root root 128G Dec 2 09:50 part-r-00000
-rw-r--r-- 1 root root 128G Dec 2 09:51 part-r-00007._COPYING_
drwxr-xr-x 3 root root 4.0K Dec 1 20:57 tmp
root@ip-172-31-14-148:/home/ubuntu# ./valsort /mnt/raid/part-r-00000
Records: 1374333632
Checksum: 28f53d4176caf880
Duplicate keys: 1202400490
SUCCESS - all records are in order
root@ip-172-31-14-148:/home/ubuntu#
```

```
root@ip-172-31-14-148:/home/ubuntu#
/dev/md0 1.1T 474G 501G 49% /mnt/raid
tmpfs 1.5G 0 1.5G 0% /run/user/0
root@ip-172-31-14-148:/home/ubuntu# df -h
Filesystem Size Used Avail Use% Mounted on
udev 7.5G 0 7.5G 0% /dev
tmpfs 1.5G 8.7M 1.5G 1% /run
/dev/xvda1 7.7G 2.0G 5.8G 26% /
tmpfs 7.5G 0 7.5G 0% /dev/shm
tmpfs 5.0M 0 5.0M 0% /run/lock
tmpfs 7.5G 0 7.5G 0% /sys/fs/cgroup
/dev/md0 1.1T 474G 501G 49% /mnt/raid
tmpfs 1.5G 0 1.5G 0% /run/user/0
root@ip-172-31-14-148:/home/ubuntu# df -h
Filesystem Size Used Avail Use% Mounted on
udev 7.5G 0 7.5G 0% /dev
tmpfs 1.5G 8.7M 1.5G 1% /run
/dev/xvda1 7.7G 2.0G 5.8G 26% /
tmpfs 7.5G 0 7.5G 0% /dev/shm
tmpfs 5.0M 0 5.0M 0% /run/lock
tmpfs 7.5G 0 7.5G 0% /sys/fs/cgroup
/dev/md0 1.1T 409G 565G 42% /mnt/raid
tmpfs 1.5G 0 1.5G 0% /run/user/0
root@ip-172-31-14-148:/home/ubuntu# exit
logout
Connection to hadoop-slave-1 closed.
root@ip-172-31-14-148:/home/ubuntu# hadoop fs -get /output_1TB/part-r-00001 /m
/raid/part-r-00001
17/12/02 09:09:47 WARN util.NativeCodeLoader: Unable to load native-hadoop lib
ry for your platform... using builtin-java classes where applicable
"C17/12/02 09:09:50 WARN hdfs.DFSClient: DFSInputStream has been closed alread
get: Filesystem closed
root@ip-172-31-14-148:/home/ubuntu# hadoop fs -get /output_1TB/part-r-00007 /m
/raid/part-r-00007
17/12/02 09:10:14 WARN util.NativeCodeLoader: Unable to load native-hadoop lib
ry for your platform... using builtin-java classes where applicable
"[A17/12/02 09:51:56 WARN hdfs.DFSClient: DFSInputStream has been closed alre
ady
root@ip-172-31-14-148:/home/ubuntu# ./valsort /mnt/raid/part-r-00007
Records: 1374400560
Checksum: 28f582aaff0ef6fd
Duplicate keys: 1202400490
SUCCESS - all records are in order
root@ip-172-31-14-148:/home/ubuntu#
```

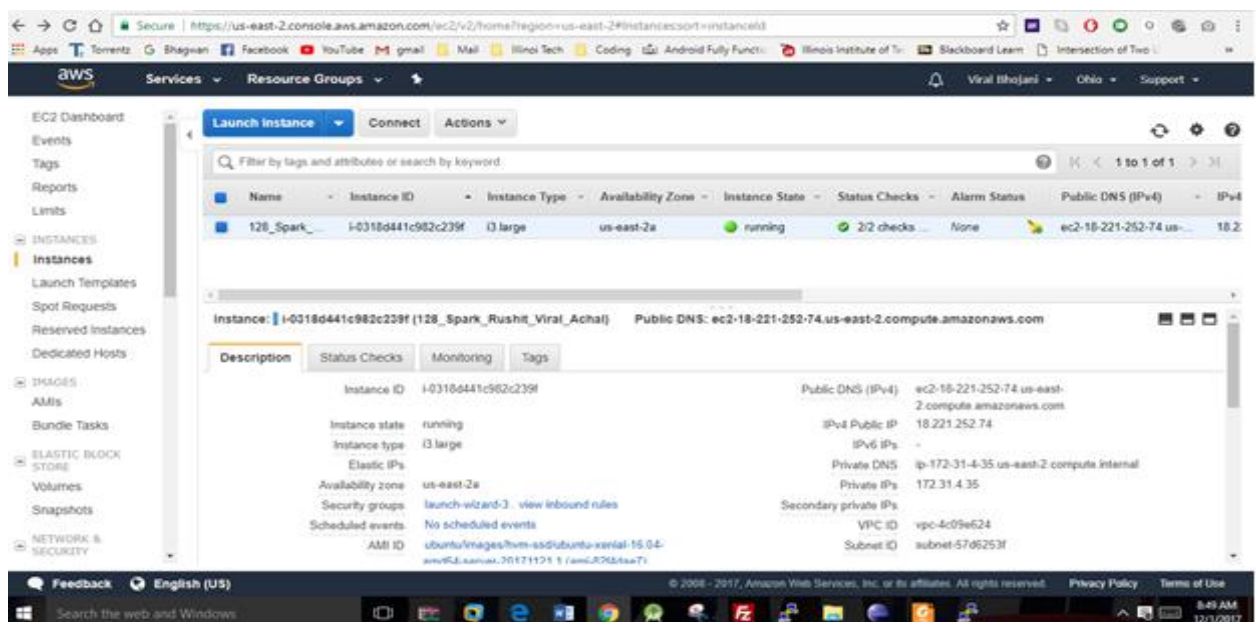
-Above screenshots is list of output part files of total size 1TB. Hence, we have shown valsort on only first and last part of output file which shows that all records are in order



-Above screenshot is Hadoop completion report with time required to finish the experiment

Spark Single Node Cluster

128GB Dataset



-Above screenshot is cluster information used in this experiment

```
root@ip-172-31-4-35: /home/ubuntu
localhost: namenode running as process 5693. Stop it first.
localhost: datanode running as process 5837. Stop it first.
Starting secondary namenodes [0.0.0.0]
0.0.0.0: secondarynamenode running as process 6017. Stop it first.
17/12/01 08:03:19 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
starting yarn daemons
resourcemanager running as process 6168. Stop it first.
localhost: nodemanager running as process 6287. Stop it first.
root@ip-172-31-4-35:/home/ubuntu# jps
6017 SecondaryNameNode
9332 Jps
6168 ResourceManager
5837 DataNode
5693 NameNode
6287 NodeManager
root@ip-172-31-4-35:/home/ubuntu# hadoop fs -ls -h /input_128gb
17/12/01 08:03:46 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
-rw-r--r-- 1 root supergroup 128.0 G 2017-12-01 05:46 /input_128gb
root@ip-172-31-4-35:/home/ubuntu# spark-shell -i spark_prog.txt
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
root@ip-172-31-4-35:/home/ubuntu# vi spark_prog.txt
root@ip-172-31-4-35:/home/ubuntu# spark-shell -i spark_prog.txt
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
17/12/01 08:05:05 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
17/12/01 08:05:05 WARN SparkConf: In Spark 1.0 and later spark.local.dir will be overridden by the value set by the cluster manager (via SPARK_LOCAL_DIRS in mesos/standalone and LOCAL_DIRS in YARN).
17/12/01 08:05:08 WARN General: Plugin (Bundle) "org.datanucleus" is already registered. Ensure you dont have multiple JAR versions of the same plugin in the classpath. The URL "file:/opt/spark/jars/datanucleus-core-3.2.10.jar" is already registered, and you are trying to register an identical plugin located at URL "file:/opt/spark-2.2.0-bin-hadoop2.7/jars/datanucleus-core-3.2.10.jar."
17/12/01 08:05:08 WARN General: Plugin (Bundle) "org.datanucleus.api.jdo" is already registered. Ensure you dont have multiple JAR versions of the same plugin in the classpath. The URL "file:/opt/spark/jars/datanucleus-api-jdo-3.2.6.jar" is already registered, and you are trying to register an identical plugin located at URL "file:/opt/spark-2.2.0-bin-hadoop2.7/jars/datanucleus-api-jdo-3.2.6.jar."
```

-Above screenshot is command used to run spark spark_prog.txt

```
root@ip-172-31-4-35: /home/ubuntu
Safe mode is OFF
root@ip-172-31-4-35:/home/ubuntu# hadoop fs -rm -r /output_128gb
17/12/01 11:20:16 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
17/12/01 11:20:17 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emptier interval = 0 minutes.
Deleted /output_128gb
root@ip-172-31-4-35:/home/ubuntu# rm -r /mnt/raid/spark_temp
root@ip-172-31-4-35:/home/ubuntu# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            7.5G   0  7.5G   0% /dev
tmpfs           1.5G   8.8M  1.5G   1% /run
/dev/xvda1      39G   2.9G  36G   8% /
tmpfs           7.5G   0  7.5G   0% /dev/shm
tmpfs           5.0M   0  5.0M   0% /run/lock
tmpfs           7.5G   0  7.5G   0% /sys/fs/cgroup
tmpfs           1.5G   0  1.5G   0% /run/user/1000
/dev/md0        436G  258G  156G  63% /mnt/raid
tmpfs           1.5G   0  1.5G   0% /run/user/0
root@ip-172-31-4-35:/home/ubuntu# spark-shell -i spark_prog.txt
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
17/12/01 11:21:38 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
17/12/01 11:21:38 WARN SparkConf: In Spark 1.0 and later spark.local.dir will be overridden by the value set by the cluster manager (via SPARK_LOCAL_DIRS in mesos/standalone and LOCAL_DIRS in YARN).
17/12/01 11:21:42 WARN General: Plugin (Bundle) "org.datanucleus" is already registered. Ensure you dont have multiple JAR versions of the same plugin in the classpath. The URL "file:/opt/spark/jars/datanucleus-core-3.2.10.jar" is already registered, and you are trying to register an identical plugin located at URL "file:/opt/spark-2.2.0-bin-hadoop2.7/jars/datanucleus-core-3.2.10.jar."
17/12/01 11:21:42 WARN General: Plugin (Bundle) "org.datanucleus.api.jdo" is already registered. Ensure you dont have multiple JAR versions of the same plugin in the classpath. The URL "file:/opt/spark/jars/datanucleus-api-jdo-3.2.6.jar" is already registered, and you are trying to register an identical plugin located at URL "file:/opt/spark-2.2.0-bin-hadoop2.7/jars/datanucleus-api-jdo-3.2.6.jar."
17/12/01 11:21:42 WARN General: Plugin (Bundle) "org.datanucleus.store.rdbms" is already registered. Ensure you dont have multiple JAR versions of the same plugin in the classpath. The URL "file:/opt/spark/jars/datanucleus-rdbms-3.2.9.jar" is already registered, and you are trying to register an identical plugin located at URL "file:/opt/spark-2.2.0-bin-hadoop2.7/jars/datanucleus-rdbms-3.2.9.jar."
17/12/01 11:21:47 WARN ObjectStore: Failed to get database global_temp, returning NoSuchObjectException
Spark context Web UI available at http://172.31.4.35:4040
Spark context available as 'sc' (master = local[*], app id = local-1512127299861).
Spark session available as 'spark'.
Loading spark_prog.txt...
start: Long = 24487938589319
input: org.apache.spark.rdd.RDD[String] = hdf5://localhost:9000/input_128gb MapPartitionsRDD[1] at TextFile at <console>:24
keyval: org.apache.spark.rdd.RDD[(String, String)] = MapPartitionsRDD[2] at map at <console>:26
result: org.apache.spark.rdd.RDD[(String, String)] = ShuffledRDD[5] at sortByKey at <console>:28
final_output: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[6] at map at <console>:30
(Stage 1):
(2 + 2) / 256
```

-Above screenshot is information that spark execution is started

```
root@ip-172-31-4-35:/home/ubuntu
17/12/01 11:20:16 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
17/12/01 11:20:17 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Empty interval = 0 minutes.
Deleted /output_128gb
root@ip-172-31-4-35:/home/ubuntu# rm -r /mnt/raid/spark_temp
root@ip-172-31-4-35:/home/ubuntu# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            7.5G   0  7.5G   0% /dev
tmpfs           1.5G   0  1.5G   0% /run
/dev/xvda1      39G   2.9G   36G   8% /
tmpfs           7.5G   0  7.5G   0% /dev/shm
tmpfs           5.0M   0  5.0M   0% /run/lock
tmpfs           7.5G   0  7.5G   0% /sys/fs/cgroup
tmpfs           1.5G   0  1.5G   0% /run/user/1000
/dev/md0        436G  258G  156G  63% /mnt/raid
tmpfs           1.5G   0  1.5G   0% /run/user/0
root@ip-172-31-4-35:/home/ubuntu# spark-shell -i spark_prog.txt
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
17/12/01 11:21:38 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
17/12/01 11:21:38 WARN SparkConf: In Spark 1.0 and later spark.local.dir will be overridden by the value set by the cluster manager (via SPARK_LOCAL_DIRS in mesos/stand
alone and LOCAL_DIRS in YARN).
17/12/01 11:21:42 WARN General: Plugin (Bundle) "org.datanucleus" is already registered. Ensure you dont have multiple JAR versions of the same plugin in the classpath.
The URL "file:/opt/spark/jars/datanucleus-core-3.2.10.jar" is already registered, and you are trying to register an identical plugin located at URL "file:/opt/spark-2.
2.0-bin-hadoop2.7/jars/datanucleus-core-3.2.10.jar."
17/12/01 11:21:42 WARN General: Plugin (Bundle) "org.datanucleus.api.jdo" is already registered. Ensure you dont have multiple JAR versions of the same plugin in the cl
asspath. The URL "file:/opt/spark/jars/datanucleus-api-jdo-3.2.6.jar" is already registered, and you are trying to register an identical plugin located at URL "file:/op
t/spark-2.2.0-bin-hadoop2.7/jars/datanucleus-api-jdo-3.2.6.jar."
17/12/01 11:21:42 WARN General: Plugin (Bundle) "org.datanucleus.store.rdbms" is already registered. Ensure you dont have multiple JAR versions of the same plugin in th
e classpath. The URL "file:/opt/spark/jars/datanucleus-rdbms-3.2.9.jar" is already registered, and you are trying to register an identical plugin located at URL "file:/o
pt/spark-2.2.0-bin-hadoop2.7/jars/datanucleus-rdbms-3.2.9.jar."
17/12/01 11:21:47 WARN ObjectStore: Failed to get database global_temp, returning NoSuchObjectException
Spark context Web UI available at http://172.31.4.35:4040
Spark context available as 'sc' (master = local[*], app id = local-1512127299861).
Spark session available as 'spark'.
Loading spark_prog.txt...
start: Long = 24487938589319
input: org.apache.spark.rdd.RDD[String] = hdfs://localhost:9000/input_128gb MapPartitionsRDD[1] at textFile at <console>:24
keyval: org.apache.spark.rdd.RDD[(String, String)] = MapPartitionsRDD[2] at map at <console>:26
result: org.apache.spark.rdd.RDD[(String, String)] = ShuffledRDD[5] at sortByKey at <console>:28
final output: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[6] at map at <console>:30
end: Long = 35597846468104
Elapsed time: 11109907878785ns
```

-Above screenshot is information that spark execution is completed

```
root@ip-172-31-4-35:/home/ubuntu
Welcome to
Spark version 2.2.0

Using Scala version 2.11.8 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_151)
Type in expressions to have them evaluated.
Type :help for more information.

scala> :q
root@ip-172-31-4-35:/home/ubuntu# hadoop fs -ls -h /output_128gb
17/12/01 14:28:28 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 257 items
-rw-r--r-- 3 root supergroup 0 2017-12-01 14:26 /output_128gb/_SUCCESS
-rw-r--r-- 3 root supergroup 512.4 M 2017-12-01 12:27 /output_128gb/part-00000
-rw-r--r-- 3 root supergroup 397.4 M 2017-12-01 12:27 /output_128gb/part-00001
-rw-r--r-- 3 root supergroup 469.8 M 2017-12-01 12:27 /output_128gb/part-00002
-rw-r--r-- 3 root supergroup 506.5 M 2017-12-01 12:28 /output_128gb/part-00003
-rw-r--r-- 3 root supergroup 462.2 M 2017-12-01 12:28 /output_128gb/part-00004
-rw-r--r-- 3 root supergroup 667.4 M 2017-12-01 12:29 /output_128gb/part-00005
-rw-r--r-- 3 root supergroup 458.5 M 2017-12-01 12:29 /output_128gb/part-00006
-rw-r--r-- 3 root supergroup 613.7 M 2017-12-01 12:30 /output_128gb/part-00007
-rw-r--r-- 3 root supergroup 516.6 M 2017-12-01 12:30 /output_128gb/part-00008
-rw-r--r-- 3 root supergroup 518.1 M 2017-12-01 12:31 /output_128gb/part-00009
-rw-r--r-- 3 root supergroup 628.4 M 2017-12-01 12:31 /output_128gb/part-00010
-rw-r--r-- 3 root supergroup 609.9 M 2017-12-01 12:32 /output_128gb/part-00011
-rw-r--r-- 3 root supergroup 408.8 M 2017-12-01 12:32 /output_128gb/part-00012
-rw-r--r-- 3 root supergroup 487.2 M 2017-12-01 12:33 /output_128gb/part-00013
-rw-r--r-- 3 root supergroup 579.3 M 2017-12-01 12:34 /output_128gb/part-00014
-rw-r--r-- 3 root supergroup 530.9 M 2017-12-01 12:34 /output_128gb/part-00015
-rw-r--r-- 3 root supergroup 477.6 M 2017-12-01 12:34 /output_128gb/part-00016
-rw-r--r-- 3 root supergroup 776.9 M 2017-12-01 12:36 /output_128gb/part-00017
-rw-r--r-- 3 root supergroup 671.8 M 2017-12-01 12:36 /output_128gb/part-00018
-rw-r--r-- 3 root supergroup 489.3 M 2017-12-01 12:37 /output_128gb/part-00019
-rw-r--r-- 3 root supergroup 574.2 M 2017-12-01 12:37 /output_128gb/part-00020
-rw-r--r-- 3 root supergroup 477.2 M 2017-12-01 12:37 /output_128gb/part-00021
-rw-r--r-- 3 root supergroup 477.6 M 2017-12-01 12:38 /output_128gb/part-00022
-rw-r--r-- 3 root supergroup 567.4 M 2017-12-01 12:38 /output_128gb/part-00023
-rw-r--r-- 3 root supergroup 582.9 M 2017-12-01 12:39 /output_128gb/part-00024
-rw-r--r-- 3 root supergroup 519.5 M 2017-12-01 12:39 /output_128gb/part-00025
-rw-r--r-- 3 root supergroup 582.7 M 2017-12-01 12:40 /output_128gb/part-00026
```

-Above screenshot is information that 257 files being created each file in the range of 400 to 600 MB making it 128GB data

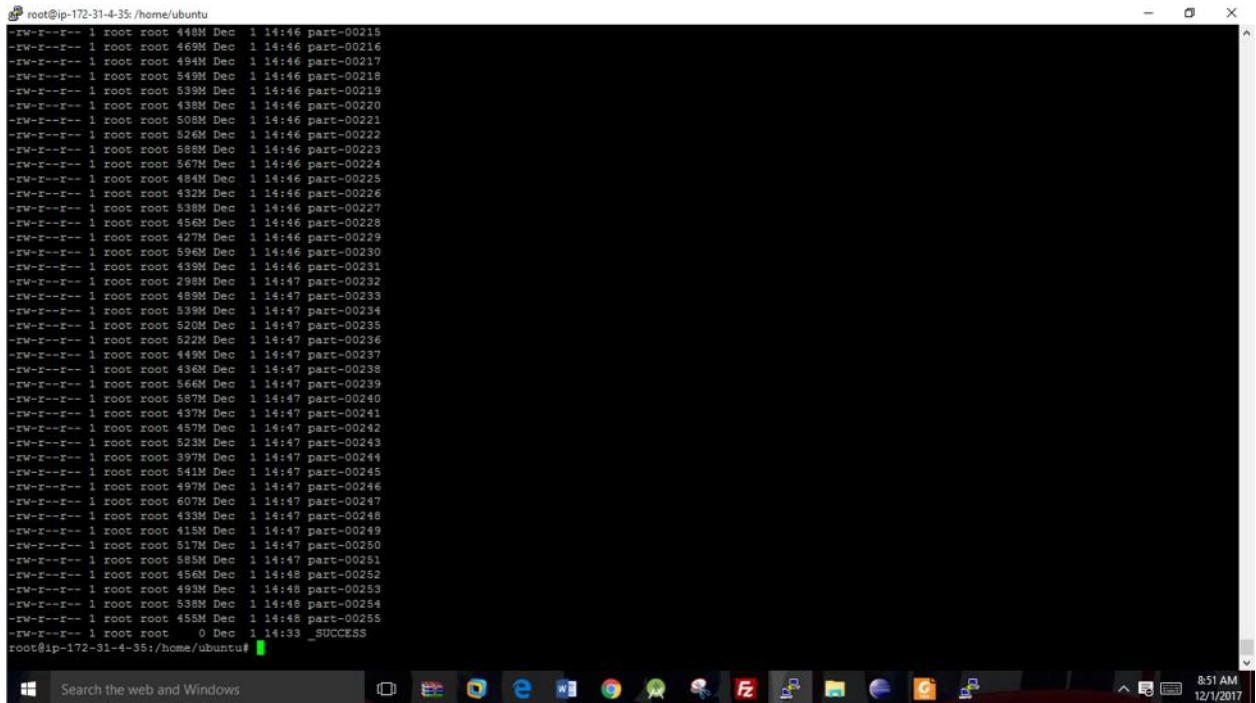

```
root@ip-172-31-4-35:/home/ubuntu# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            7.5G   0  7.5G   0% /dev
tmpfs           1.5G   0  1.5G   0% /run
/dev/xvda1      39G   2.9G   36G   8% /
tmpfs           7.5G   0  7.5G   0% /dev/shm
tmpfs           5.0M   0  5.0M   0% /run/lock
tmpfs           7.5G   0  7.5G   0% /sys/fs/cgroup
tmpfs           1.5G   0  1.5G   0% /run/user/1000
/dev/md0        436G  259G  155G  63% /mnt/raid
tmpfs           1.5G   0  1.5G   0% /run/user/0

root@ip-172-31-4-35:/home/ubuntu# hadoop fs -get /output /mnt/raid/output_128gb
17/12/01 14:33:40 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
get: '/output': No such file or directory

root@ip-172-31-4-35:/home/ubuntu# hadoop fs -get /output_128gb /mnt/raid/output_128gb
17/12/01 14:33:50 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
17/12/01 14:33:50 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:33:52 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:33:53 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:33:55 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:33:57 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:34:00 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:34:07 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:34:08 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:34:11 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:34:13 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:34:22 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:34:36 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:34:38 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:34:40 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:34:42 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:34:49 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:34:51 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:34:52 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:34:57 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:35:04 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:35:05 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:35:08 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:35:11 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:35:17 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:35:19 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:35:21 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:35:24 WARN hdfs.DFSClient: DFSInputStream has been closed already
```

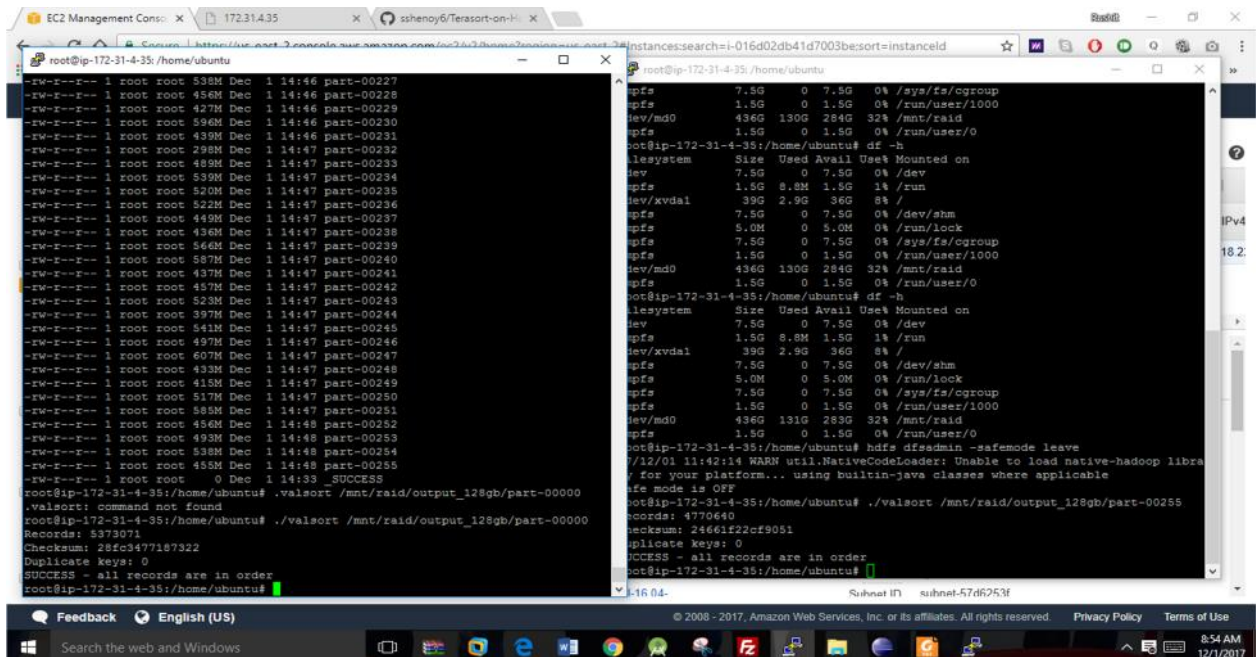
-Above screenshot is information for getting all 257 files from Hadoop hdfs into our file system

```
17/12/01 14:46:26 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:46:30 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:46:36 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:46:39 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:46:39 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:46:42 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:46:48 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:46:49 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:46:51 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:46:54 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:46:56 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:47:03 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:47:04 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:47:05 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:47:08 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:47:12 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:47:18 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:47:19 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:47:21 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:47:23 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:47:26 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:47:33 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:47:34 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:47:35 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:47:37 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:47:40 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:47:47 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:47:48 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:47:49 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:47:52 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:47:58 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:48:02 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:48:03 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:48:06 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/01 14:48:08 WARN hdfs.DFSClient: DFSInputStream has been closed already
root@ip-172-31-4-35:/home/ubuntu# ls -lh /mnt/raid/output_128gb/
total 129G
-rw-r--r-- 1 root root 513M Dec 1 14:33 part-00000
-rw-r--r-- 1 root root 398M Dec 1 14:33 part-00001
-rw-r--r-- 1 root root 470M Dec 1 14:33 part-00002
-rw-r--r-- 1 root root 507M Dec 1 14:33 part-00003
-rw-r--r-- 1 root root 463M Dec 1 14:34 part-00004
-rw-r--r-- 1 root root 668M Dec 1 14:34 part-00005
```



```
root@ip-172-31-4-35: /home/ubuntu
-rw-r--r-- 1 root root 448M Dec 1 14:46 part-00218
-rw-r--r-- 1 root root 469M Dec 1 14:46 part-00216
-rw-r--r-- 1 root root 494M Dec 1 14:46 part-00217
-rw-r--r-- 1 root root 549M Dec 1 14:46 part-00218
-rw-r--r-- 1 root root 539M Dec 1 14:46 part-00219
-rw-r--r-- 1 root root 438M Dec 1 14:46 part-00220
-rw-r--r-- 1 root root 508M Dec 1 14:46 part-00221
-rw-r--r-- 1 root root 526M Dec 1 14:46 part-00222
-rw-r--r-- 1 root root 588M Dec 1 14:46 part-00223
-rw-r--r-- 1 root root 567M Dec 1 14:46 part-00224
-rw-r--r-- 1 root root 484M Dec 1 14:46 part-00225
-rw-r--r-- 1 root root 432M Dec 1 14:46 part-00226
-rw-r--r-- 1 root root 538M Dec 1 14:46 part-00227
-rw-r--r-- 1 root root 456M Dec 1 14:46 part-00228
-rw-r--r-- 1 root root 427M Dec 1 14:46 part-00229
-rw-r--r-- 1 root root 596M Dec 1 14:46 part-00230
-rw-r--r-- 1 root root 439M Dec 1 14:46 part-00231
-rw-r--r-- 1 root root 298M Dec 1 14:47 part-00232
-rw-r--r-- 1 root root 498M Dec 1 14:47 part-00233
-rw-r--r-- 1 root root 539M Dec 1 14:47 part-00234
-rw-r--r-- 1 root root 520M Dec 1 14:47 part-00235
-rw-r--r-- 1 root root 522M Dec 1 14:47 part-00236
-rw-r--r-- 1 root root 449M Dec 1 14:47 part-00237
-rw-r--r-- 1 root root 436M Dec 1 14:47 part-00238
-rw-r--r-- 1 root root 566M Dec 1 14:47 part-00239
-rw-r--r-- 1 root root 587M Dec 1 14:47 part-00240
-rw-r--r-- 1 root root 437M Dec 1 14:47 part-00241
-rw-r--r-- 1 root root 457M Dec 1 14:47 part-00242
-rw-r--r-- 1 root root 523M Dec 1 14:47 part-00243
-rw-r--r-- 1 root root 397M Dec 1 14:47 part-00244
-rw-r--r-- 1 root root 541M Dec 1 14:47 part-00245
-rw-r--r-- 1 root root 497M Dec 1 14:47 part-00246
-rw-r--r-- 1 root root 607M Dec 1 14:47 part-00247
-rw-r--r-- 1 root root 433M Dec 1 14:47 part-00248
-rw-r--r-- 1 root root 415M Dec 1 14:47 part-00249
-rw-r--r-- 1 root root 517M Dec 1 14:47 part-00250
-rw-r--r-- 1 root root 585M Dec 1 14:47 part-00251
-rw-r--r-- 1 root root 456M Dec 1 14:48 part-00252
-rw-r--r-- 1 root root 493M Dec 1 14:48 part-00253
-rw-r--r-- 1 root root 538M Dec 1 14:48 part-00254
-rw-r--r-- 1 root root 455M Dec 1 14:48 part-00255
-rw-r--r-- 1 root root 0 Dec 1 14:33 _SUCCESS
root@ip-172-31-4-35: /home/ubuntu
```

-Above two screenshots are directory structure mnt/raid/output_128gb



```
root@ip-172-31-4-35: /home/ubuntu
-rw-r--r-- 1 root root 538M Dec 1 14:46 part-00227
-rw-r--r-- 1 root root 456M Dec 1 14:46 part-00228
-rw-r--r-- 1 root root 427M Dec 1 14:46 part-00229
-rw-r--r-- 1 root root 596M Dec 1 14:46 part-00230
-rw-r--r-- 1 root root 439M Dec 1 14:46 part-00231
-rw-r--r-- 1 root root 298M Dec 1 14:47 part-00232
-rw-r--r-- 1 root root 498M Dec 1 14:47 part-00233
-rw-r--r-- 1 root root 539M Dec 1 14:47 part-00234
-rw-r--r-- 1 root root 520M Dec 1 14:47 part-00235
-rw-r--r-- 1 root root 522M Dec 1 14:47 part-00236
-rw-r--r-- 1 root root 449M Dec 1 14:47 part-00237
-rw-r--r-- 1 root root 436M Dec 1 14:47 part-00238
-rw-r--r-- 1 root root 566M Dec 1 14:47 part-00239
-rw-r--r-- 1 root root 587M Dec 1 14:47 part-00240
-rw-r--r-- 1 root root 437M Dec 1 14:47 part-00241
-rw-r--r-- 1 root root 457M Dec 1 14:47 part-00242
-rw-r--r-- 1 root root 523M Dec 1 14:47 part-00243
-rw-r--r-- 1 root root 397M Dec 1 14:47 part-00244
-rw-r--r-- 1 root root 541M Dec 1 14:47 part-00245
-rw-r--r-- 1 root root 497M Dec 1 14:47 part-00246
-rw-r--r-- 1 root root 607M Dec 1 14:47 part-00247
-rw-r--r-- 1 root root 433M Dec 1 14:47 part-00248
-rw-r--r-- 1 root root 415M Dec 1 14:47 part-00249
-rw-r--r-- 1 root root 517M Dec 1 14:47 part-00250
-rw-r--r-- 1 root root 585M Dec 1 14:47 part-00251
-rw-r--r-- 1 root root 456M Dec 1 14:48 part-00252
-rw-r--r-- 1 root root 493M Dec 1 14:48 part-00253
-rw-r--r-- 1 root root 538M Dec 1 14:48 part-00254
-rw-r--r-- 1 root root 455M Dec 1 14:48 part-00255
-rw-r--r-- 1 root root 0 Dec 1 14:33 _SUCCESS
root@ip-172-31-4-35: /home/ubuntu# ./valsort /mnt/raid/output_128gb/part-00000
./valsort: command not found
root@ip-172-31-4-35: /home/ubuntu# ./valsort /mnt/raid/output_128gb/part-00000
Records: 5373071
Checksum: 28fc3477187322
Duplicate keys: 0
SUCCESS - all records are in order
root@ip-172-31-4-35: /home/ubuntu#
```

```
df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs            7.5G   0  7.5G   0% /sys/fs/cgroup
mpfs            1.5G   0  1.5G   0% /run/user/1000
dev/md0         436G  130G  284G  32% /mnt/raid
mpfs            1.5G   0  1.5G   0% /run/user/0
Filesystem      Size  Used Avail Use% Mounted on
dev             7.5G   0  7.5G   0% /dev
mpfs            1.5G  8.8M  1.5G   1% /run
dev/xvda1       39G   2.9G   36G   8% /
mpfs            7.5G   0  7.5G   0% /dev/shm
mpfs            5.0M   0  5.0M   0% /run/lock
mpfs            7.5G   0  7.5G   0% /sys/fs/cgroup
mpfs            1.5G   0  1.5G   0% /run/user/1000
dev/md0         436G  130G  284G  32% /mnt/raid
mpfs            1.5G   0  1.5G   0% /run/user/0
root@ip-172-31-4-35: /home/ubuntu# df -h
Filesystem      Size  Used Avail Use% Mounted on
dev             7.5G   0  7.5G   0% /dev
mpfs            1.5G  8.8M  1.5G   1% /run
dev/xvda1       39G   2.9G   36G   8% /
mpfs            7.5G   0  7.5G   0% /dev/shm
mpfs            5.0M   0  5.0M   0% /run/lock
mpfs            7.5G   0  7.5G   0% /sys/fs/cgroup
mpfs            1.5G   0  1.5G   0% /run/user/1000
dev/md0         436G  131G  283G  32% /mnt/raid
mpfs            1.5G   0  1.5G   0% /run/user/0
root@ip-172-31-4-35: /home/ubuntu# hdfs dfsadmin -safemode leave
17/12/01 11:42:14 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Safe mode is OFF
root@ip-172-31-4-35: /home/ubuntu# ./valsort /mnt/raid/output_128gb/part-00255
Records: 4770640
Checksum: 2466122cf9051
Duplicate keys: 0
SUCCESS - all records are in order
root@ip-172-31-4-35: /home/ubuntu#
```

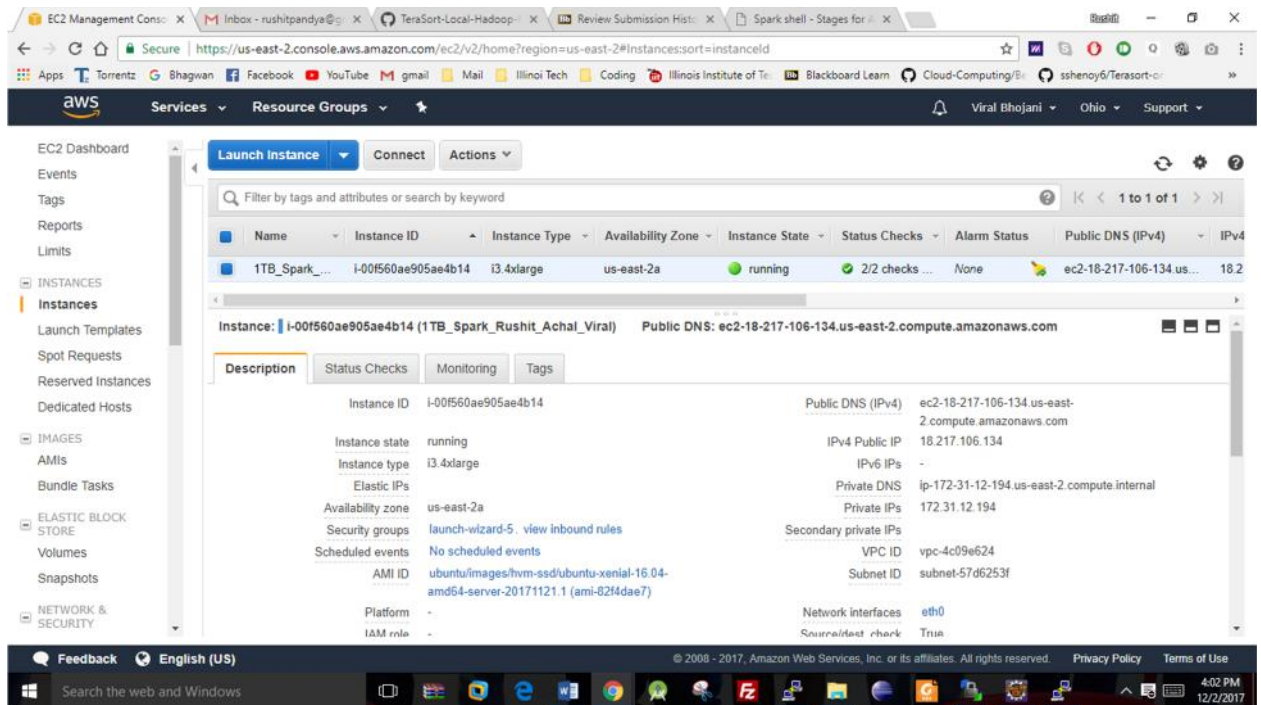
-Above screenshots infers list of output part files of total size 129G. Hence, we have shown valsort on only first and last part of output file which shows that all records are in order.

Rushit Pandya
Achal Lathia
Viral Bhojani

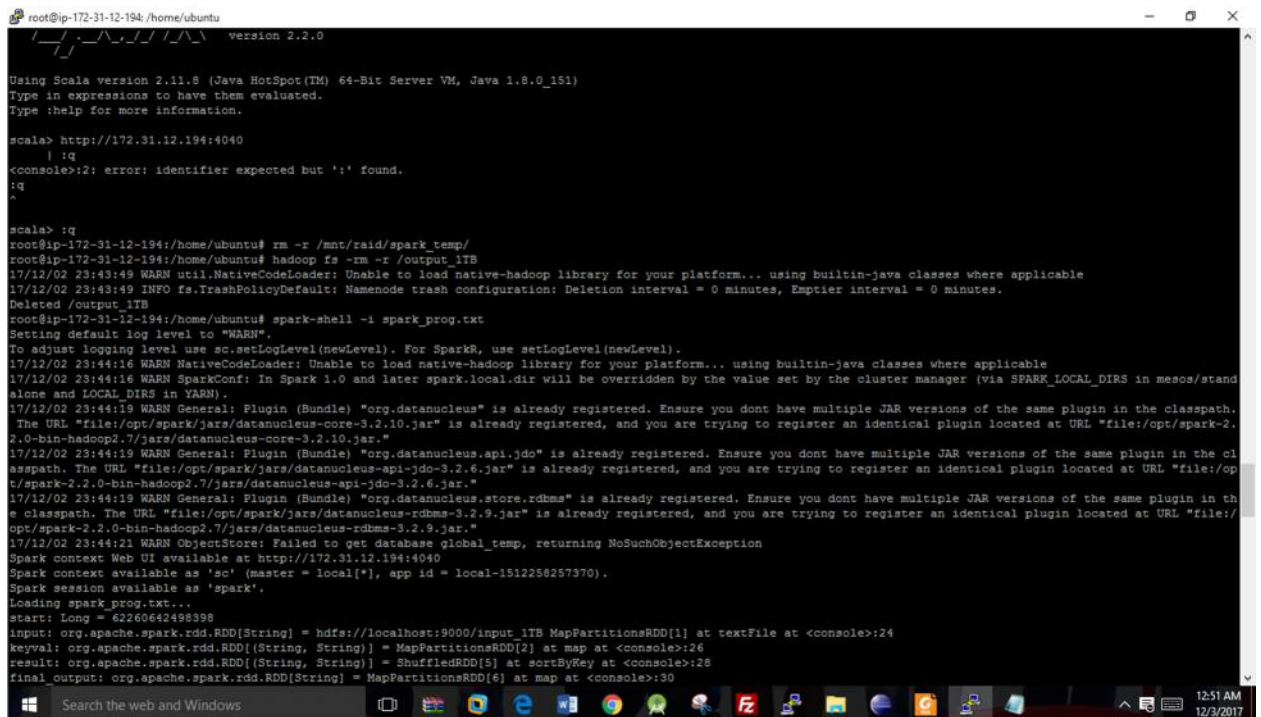
PA-2-REPORT

A20381916
A20375208
A20380100

1TB Dataset



-Above screenshot is cluster information used in this experiment



-Above screenshot is command used to run spark spark_prog.txt

```
end: Long = 87680721520198
Elapsed time: 25420079021800ns

Welcome to

  Scala
  version 2.2.0

Using Scala version 2.11.8 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_151)
Type in expressions to have them evaluated.
Type :help for more information.
```

-Above screenshot is information that the spark is started

-rw-r--r--	3	root	supergroup	473.5	M	2017-12-03	07:25	/output_1TB/part-02000
-rw-r--r--	3	root	supergroup	365.2	M	2017-12-03	07:24	/output_1TB/part-02001
-rw-r--r--	3	root	supergroup	386.7	M	2017-12-03	07:25	/output_1TB/part-02002
-rw-r--r--	3	root	supergroup	473.9	M	2017-12-03	07:25	/output_1TB/part-02003
-rw-r--r--	3	root	supergroup	563.5	M	2017-12-03	07:26	/output_1TB/part-02004
-rw-r--r--	3	root	supergroup	537.7	M	2017-12-03	07:27	/output_1TB/part-02005
-rw-r--r--	3	root	supergroup	497.5	M	2017-12-03	07:26	/output_1TB/part-02006
-rw-r--r--	3	root	supergroup	492.3	M	2017-12-03	07:27	/output_1TB/part-02007
-rw-r--r--	3	root	supergroup	471.5	M	2017-12-03	07:26	/output_1TB/part-02008
-rw-r--r--	3	root	supergroup	539.8	M	2017-12-03	07:28	/output_1TB/part-02009
-rw-r--r--	3	root	supergroup	510.3	M	2017-12-03	07:28	/output_1TB/part-02010
-rw-r--r--	3	root	supergroup	499.8	M	2017-12-03	07:28	/output_1TB/part-02011
-rw-r--r--	3	root	supergroup	645.5	M	2017-12-03	07:29	/output_1TB/part-02012
-rw-r--r--	3	root	supergroup	528.5	M	2017-12-03	07:28	/output_1TB/part-02013
-rw-r--r--	3	root	supergroup	603.9	M	2017-12-03	07:29	/output_1TB/part-02014
-rw-r--r--	3	root	supergroup	499.9	M	2017-12-03	07:29	/output_1TB/part-02015
-rw-r--r--	3	root	supergroup	531.2	M	2017-12-03	07:29	/output_1TB/part-02016
-rw-r--r--	3	root	supergroup	512.1	M	2017-12-03	07:29	/output_1TB/part-02017
-rw-r--r--	3	root	supergroup	586.8	M	2017-12-03	07:29	/output_1TB/part-02018
-rw-r--r--	3	root	supergroup	498.5	M	2017-12-03	07:29	/output_1TB/part-02019
-rw-r--r--	3	root	supergroup	492.6	M	2017-12-03	07:29	/output_1TB/part-02020
-rw-r--r--	3	root	supergroup	475.8	M	2017-12-03	07:29	/output_1TB/part-02021
-rw-r--r--	3	root	supergroup	488.5	M	2017-12-03	07:30	/output_1TB/part-02022
-rw-r--r--	3	root	supergroup	556.9	M	2017-12-03	07:30	/output_1TB/part-02023
-rw-r--r--	3	root	supergroup	493.6	M	2017-12-03	07:30	/output_1TB/part-02024
-rw-r--r--	3	root	supergroup	538.3	M	2017-12-03	07:31	/output_1TB/part-02025
-rw-r--r--	3	root	supergroup	516.4	M	2017-12-03	07:31	/output_1TB/part-02026
-rw-r--r--	3	root	supergroup	541.2	M	2017-12-03	07:31	/output_1TB/part-02027
-rw-r--r--	3	root	supergroup	527.9	M	2017-12-03	07:32	/output_1TB/part-02028
-rw-r--r--	3	root	supergroup	473.5	M	2017-12-03	07:32	/output_1TB/part-02029
-rw-r--r--	3	root	supergroup	431.4	M	2017-12-03	07:31	/output_1TB/part-02030
-rw-r--r--	3	root	supergroup	522.6	M	2017-12-03	07:32	/output_1TB/part-02031
-rw-r--r--	3	root	supergroup	443.1	M	2017-12-03	07:32	/output_1TB/part-02032
-rw-r--r--	3	root	supergroup	631.7	M	2017-12-03	07:32	/output_1TB/part-02033
-rw-r--r--	3	root	supergroup	441.7	M	2017-12-03	07:32	/output_1TB/part-02034
-rw-r--r--	3	root	supergroup	499.6	M	2017-12-03	07:32	/output_1TB/part-02035
-rw-r--r--	3	root	supergroup	462.0	M	2017-12-03	07:32	/output_1TB/part-02036
-rw-r--r--	3	root	supergroup	586.2	M	2017-12-03	07:32	/output_1TB/part-02037
-rw-r--r--	3	root	supergroup	541.8	M	2017-12-03	07:32	/output_1TB/part-02038
-rw-r--r--	3	root	supergroup	538.9	M	2017-12-03	07:32	/output_1TB/part-02039
-rw-r--r--	3	root	supergroup	489.1	M	2017-12-03	07:32	/output_1TB/part-02040
-rw-r--r--	3	root	supergroup	507.7	M	2017-12-03	07:32	/output_1TB/part-02041
-rw-r--r--	3	root	supergroup	533.3	M	2017-12-03	07:32	/output_1TB/part-02042
-rw-r--r--	3	root	supergroup	498.0	M	2017-12-03	07:32	/output_1TB/part-02043
-rw-r--r--	3	root	supergroup	444.4	M	2017-12-03	07:32	/output_1TB/part-02044
-rw-r--r--	3	root	supergroup	485.5	M	2017-12-03	07:32	/output_1TB/part-02045
-rw-r--r--	3	root	supergroup	480.9	M	2017-12-03	07:32	/output_1TB/part-02046
-rw-r--r--	3	root	supergroup	546.5	M	2017-12-03	07:32	/output_1TB/part-02047

-Above screenshot is information of 2048 file each with the size in the range of 512 MB making it up to 1 TB

```
root@ip-172-31-12-194: /home/ubuntu
root@ip-172-31-12-194:/home/ubuntu# ./valsort /mnt/raid/output_1TB/part-r-00000
Records: 5368098
Checksum: 28f9aafa1749d2
Duplicate keys: 0
SUCCESS - all records are in order
root@ip-172-31-12-194:/home/ubuntu# ./valsort /mnt/raid/output_1TB/part-r-02047
Records: 5369320
Checksum: 28f54fac8e049a
Duplicate keys: 0
SUCCESS - all records are in order
root@ip-172-31-12-194:/home/ubuntu#
```

-Above screenshots infers that list of output part files of total size 1TB (2048 * 512MB). Hence, we have shown valsort on only first and last part of output file which shows that all records are in order

Spark 8 Node Cluster

→We have run TeraSort for 10GB of dataset on 8 node cluster for Spark. The reason behind this is it takes almost 10 hours to put 1TB of data on hdfs. So, we have proportionally calculated results on basis of 10GB results and shown in performance evaluation.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs
master	i-00af8bcd07a8dacf3	m5.xlarge	us-east-2b	running	2/2 checks ...	None	ec2-18-220-14-230.us-east-2.compute.amazonaws.com	18.220.14.230	-
slave-1	i-06414d4d44b7f10c1	m5.xlarge	us-east-2b	running	2/2 checks ...	None	ec2-18-216-171-129.us-east-2.compute.amazonaws.com	18.216.171.129	-
slave-2	i-0958c3d5c5973eb0	m5.xlarge	us-east-2b	running	2/2 checks ...	None	ec2-52-15-182-82.us-east-2.compute.amazonaws.com	52.15.182.82	-
slave-3	i-07ab4c21778b1bad	m5.xlarge	us-east-2b	running	2/2 checks ...	None	ec2-18-217-104-110.us-east-2.compute.amazonaws.com	18.217.104.110	-
slave-4	i-0513941c7ed8e14c1	m5.xlarge	us-east-2b	running	2/2 checks ...	None	ec2-18-221-51-129.us-east-2.compute.amazonaws.com	18.221.51.129	-
slave-5	i-0a7f120c3501ac212a	m5.xlarge	us-east-2b	running	2/2 checks ...	None	ec2-18-221-143-us-east-2.compute.amazonaws.com	18.221.143.143	-
slave-6	i-0c95e908e41c9e2b	m5.xlarge	us-east-2b	running	2/2 checks ...	None	ec2-52-15-177-169.us-east-2.compute.amazonaws.com	52.15.177.169	-
slave-7	i-0c40710bdcc0ca1d	m5.xlarge	us-east-2b	running	2/2 checks ...	None	ec2-18-217-159-104.us-east-2.compute.amazonaws.com	18.217.159.104	-
slave-8	i-03eb7c4173f3cc350	m5.xlarge	us-east-2b	running	2/2 checks ...	None	ec2-18-217-136-113.us-east-2.compute.amazonaws.com	18.217.136.113	-

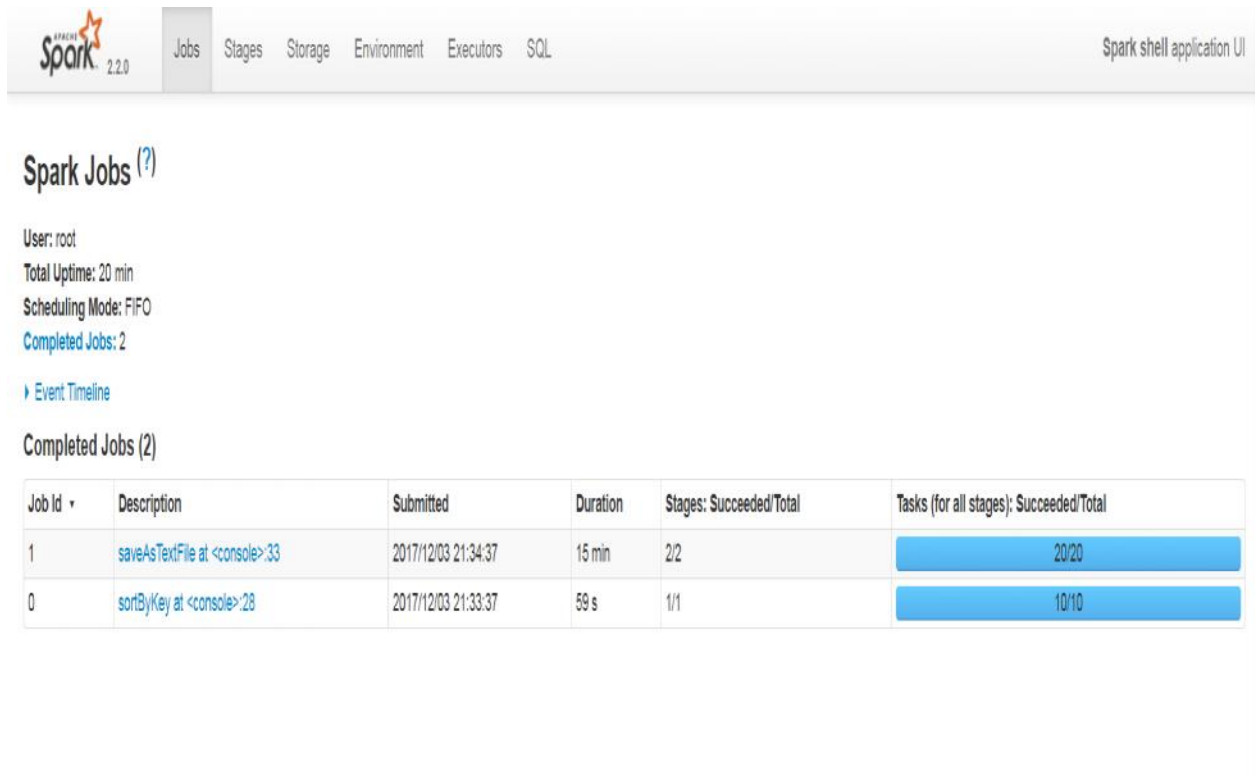
Instance: i-00af8bcd07a8dacf3 (master)		Public DNS: ec2-18-220-14-230.us-east-2.compute.amazonaws.com	
Description	Status Checks	Monitoring	Tags
Instance ID	i-00af8bcd07a8dacf3		
Instance state	running		
Instance type	m5.xlarge		
Elastic IPs			

-Above screenshot is cluster information used in this experiment

A20381916
A20375208
A20380100

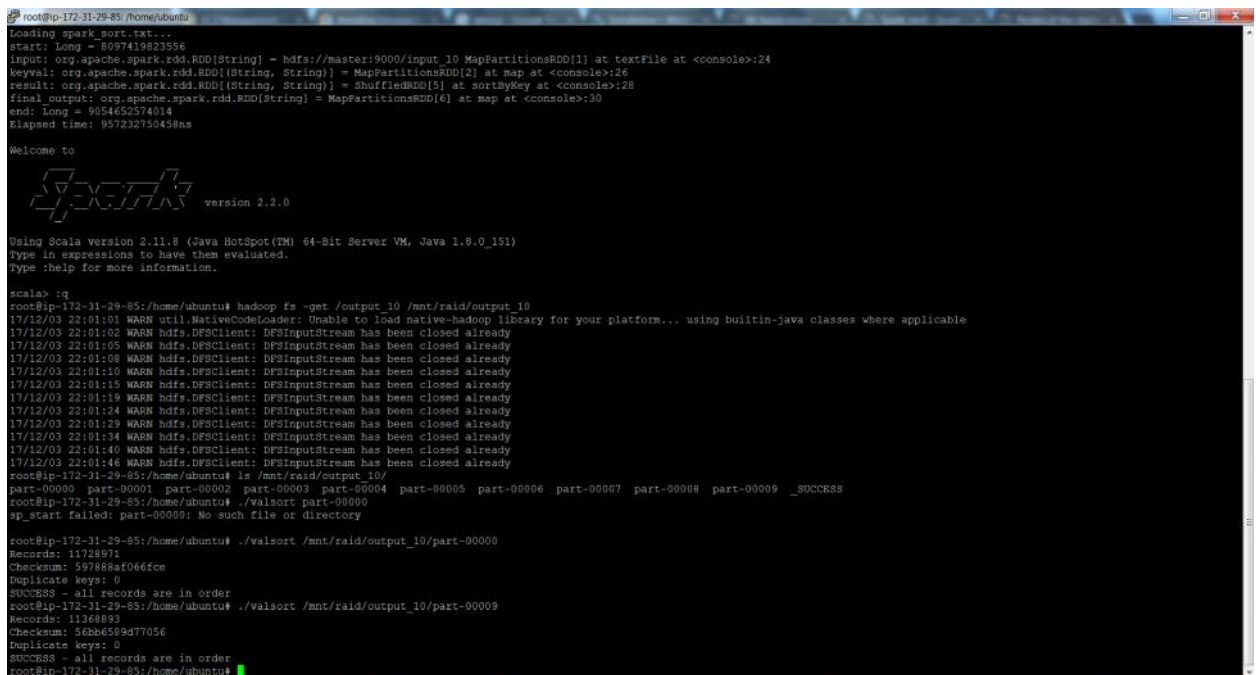
-Above screenshot is information of node available in the cluster

-Above screenshot is information that spark is started



The screenshot shows the Spark shell application UI. At the top, there's a navigation bar with tabs: Jobs, Stages, Storage, Environment, Executors, and SQL. The 'Jobs' tab is selected. Below the navigation bar, the title 'Spark Jobs (?)' is displayed. The user is 'root'. The total uptime is '20 min'. The scheduling mode is 'FIFO'. There are '2 Completed Jobs'. A link for 'Event Timeline' is provided. Below this, a section titled 'Completed Jobs (2)' contains a table with the following data:

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
1	saveAsTextFile at <console>:33	2017/12/03 21:34:37	15 min	2/2	20/20
0	sortByKey at <console>:28	2017/12/03 21:33:37	59 s	1/1	10/10



The screenshot shows a terminal window with the following output:

```
root@ip-172-31-29-85: /home/ubuntu#
Loading spark_sort.txt...
start: Long = 809741982356
input: org.apache.spark.rdd.RDD[String] = hdfs://master:9000/input_10 MapPartitionsRDD[1] at textFile at <console>:24
keyval: org.apache.spark.rdd.RDD[(String, String)] = MapPartitionsRDD[2] at map at <console>:26
result: org.apache.spark.rdd.RDD[(String, String)] = ShuffledRDD[5] at sortByKey at <console>:28
final_output: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[6] at map at <console>:30
end: Long = 9054652574014
Elapsed time: 95723275048ms

Welcome to
  ____  __
 / ___/ /  _  \
/ /   / _/  _/
/ /___/_/  _/
\____/_/  _/
   version 2.2.0

Using Scala version 2.11.8 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_151)
Type in expressions to have them evaluated.
Type :help for more information.

scala> :q
root@ip-172-31-29-85:/home/ubuntu# hadoop fs -get /output_10 /mnt/raid/output_10
17/12/03 22:01:01 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
17/12/03 22:01:02 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/03 22:01:05 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/03 22:01:08 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/03 22:01:10 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/03 22:01:15 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/03 22:01:19 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/03 22:01:24 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/03 22:01:29 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/03 22:01:34 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/03 22:01:40 WARN hdfs.DFSClient: DFSInputStream has been closed already
17/12/03 22:01:46 WARN hdfs.DFSClient: DFSInputStream has been closed already
root@ip-172-31-29-85:/home/ubuntu# ls /mnt/raid/output_10/
part-00000 part-00001 part-00002 part-00003 part-00004 part-00005 part-00006 part-00007 part-00008 part-00009 __SUCCESS__
root@ip-172-31-29-85:/home/ubuntu# ./valsort part-00000
sp_start failed: part-00000: No such file or directory

root@ip-172-31-29-85:/home/ubuntu# ./valsort /mnt/raid/output_10/part-00000
Records: 11728971
Checksum: 597888aef066f0e
Duplicate keys: 0
SUCCESS - all records are in order
root@ip-172-31-29-85:/home/ubuntu# ./valsort /mnt/raid/output_10/part-00009
Records: 11768993
Checksum: 56bb6589d77056
Duplicate keys: 0
SUCCESS - all records are in order
root@ip-172-31-29-85:/home/ubuntu#
```

-Above screenshots infer that list of output part files. Hence, we have shown valsort on only first and last part of output file which shows that all records are in order