

SORT ON HADOOP/ SPARK

DESIGN DOCUMENT

Shalin A. Chopra

3/28/2016

DESIGN DOCUMENT

1. SHARED MEMORY SORT:

- The code for Shared-Memory Sort is written in Java Programming Language.
- The main aim of this code is to perform sorting on large datasets, considering this I have opted for Quick Sort and k-way external merge sort.
- The program performs sorting for 1GB and 10GB datasets, where key is of 10 Bytes and value of 90 Bytes.
- Sorting is performed based on the key's ASCII value.
- I have divided the original dataset into 80 Chunks and applied Quick Sort on these chunks, which provides me with sorted chunks as output. This part of the program has threads applied to it to parallelize the process of sorting.
- Then using k-way external merge sort, I have merged all the Sorted Chunks of earlier phase into a final sorted output file.

2. HADOOP SORT:

- The code for Hadoop based Sorting is written in Java Programming language.
- Here, I have used Hadoop's Map-Reduce based implementation to sort large datasets.
- Hadoop has a Sort and Shuffle phase, which does the Sorting of data internally, without writing any code particularly for sort.
- I have utilized the same in my code, I have provided the mapper with input as key (10 Bytes) and value(90 Bytes), the Sort and Shuffle phase will categorize the data and sort them according to its Key.
- The reducer here just has to do the work of emitting the output given by the Sort and Shuffle phase, hence I have used the Identity reducer which will perform the same job as reducer here will.
- I have tested the performance of hadoop on 1-Node (1GB, 10GB) and 16-Node (100GB).
- Hadoop's Configuration process is explained in the next document.

3. SPARK SORT:

- The code for spark is written using spark shell, which uses Scala as a programming language.
- Spark also uses the same Map-Reduce based implementation as Hadoop does.
- When installing Spark it installs various features and language packages, like Scala, R-Studio, Python, specifically for Spark.
- I have implemented sorting using scala's sortByKey(), functionality, which sorts the data based on Key value (10 Bytes). The sorted data obtained using spark, is divided into chunks.
- I have also implemented the same using Python programming and PySpark package.

SORT ON HADOOP/ SPARK

PERFORMANCE EVALUATION

Shalin A. Chopra

3/28/2016

PERFORMANCE EVALUATION

This document presents with the performance evaluation for 3 different platforms which implements Sorting of large datasets:

- 1. SHARED MEMORY**
- 2. HADOOP**
- 3. SPARK**

1. SHARED MEMORY

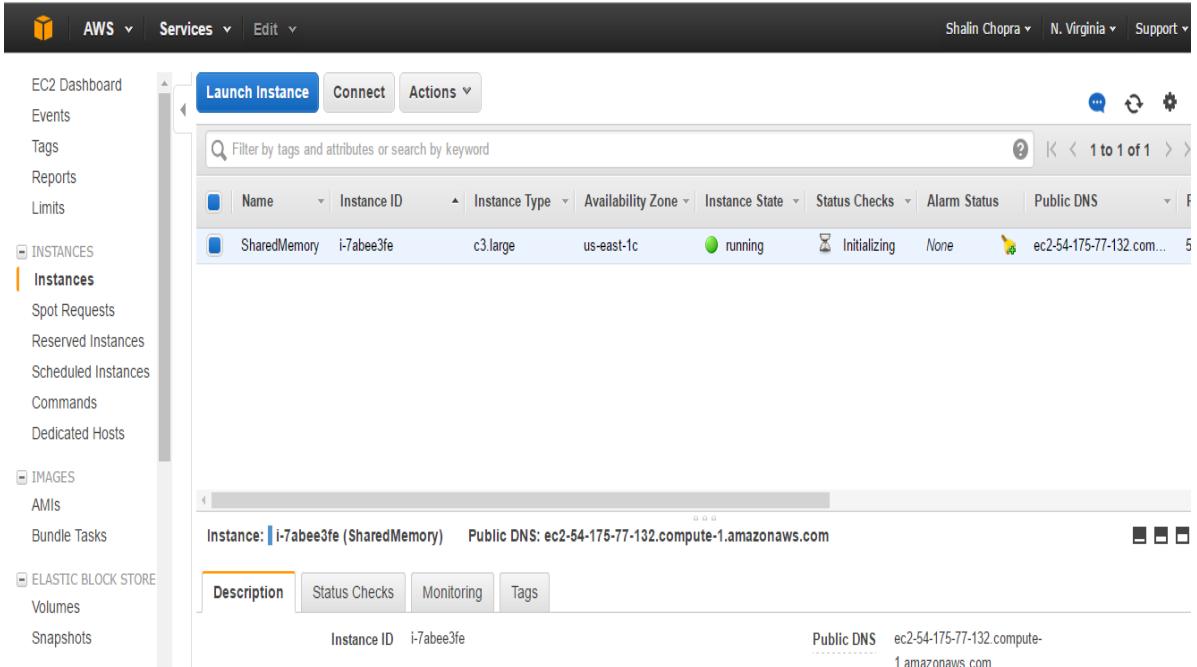
Performance is done on AWS EC2 instance i.e. c3.large, using linux version: Ubuntu. The Specifications are as follows:

Specifications:	
Instance Type	c3.large Ubuntu
RAM	3.75 GB
No. of Cores	2 (vCores)
Storage	2 * 16 (SSD) GB
Region	us-east-1 (N. Virginia)

- The Sorting of datasets if performed for 1GB and 10GB data, using 1,2,4 and 8 Threads using java.

The following screenshots shows the execution of **Shared memory Sort 1GB**:

Instance Launch in AWS:



To make full utilization of the storage SSD disks, I have formed a Raid and mounted it on /mnt/raid

1GB using 1 Thread:

```
ubuntu@ip-172-31-11-28:/mnt/raid/64$ vi SharedMemoryScript.sh
ubuntu@ip-172-31-11-28:/mnt/raid/64$ sh SharedMemoryScript.sh
Generating Un-Sorted file of 1GB
1.78user 0.02system 0:02.99elapsed 60%CPU (0avgtext+0avgdata 61160maxresident)k
0inputs+1955136outputs (0major+3677minor)pagefaults 0swaps
Creating file Chunks
0.00user 1.17system 0:02.41elapsed 48%CPU (0avgtext+0avgdata 756maxresident)k
1953128inputs+1956112outputs (0major+232minor)pagefaults 0swaps
File Chunks Created generated
STARTING SORT
Running Threads: 1
Sorted Chunks Created, Final Merge Phase Starts
Time Taken in millis: 30473.0
28.89user 3.44system 0:30.55elapsed 105%CPU (0avgtext+0avgdata 776604maxresident)k
80inputs+3927200outputs (0major+13880minor)pagefaults 0swaps
FINISHED SORTING
Checking Sorted File using Valsort
Records: 10000000
Checksum: 4c499aeel4ae2b
Duplicate keys: 0
SUCCESS - all records are in order
1.59user 0.01system 0:04.87elapsed 32%CPU (0avgtext+0avgdata 25500maxresident)k
1953128inputs+0outputs (0major+3351minor)pagefaults 0swaps
Deleting Temporary Chunks
```

1GB using 2 Threads:

```
ubuntu@ip-172-31-11-28:/mnt/raid/64$ vi SharedMemoryScript.sh
ubuntu@ip-172-31-11-28:/mnt/raid/64$ sh SharedMemoryScript.sh
Generating Un-Sorted file of 1GB
1.74user 0.04system 0:03.03elapsed 59%CPU (0avgtext+0avgdata 61160maxresident)k
8inputs+1955136outputs (0major+4188minor)pagefaults 0swaps
Creating file Chunks
0.00user 1.18system 0:02.32elapsed 50%CPU (0avgtext+0avgdata 756maxresident)k
1953120inputs+1956112outputs (0major+232minor)pagefaults 0swaps
File Chunks Created generated
STARTING SORT
Running Threads: 2
Sorted Chunks Created, Final Merge Phase Starts
Time Taken in millis: 28190.0
37.57user 4.00system 0:28.26elapsed 147%CPU (0avgtext+0avgdata 830624maxresident)k
184inputs+3935176outputs (0major+21634minor)pagefaults 0swaps
FINISHED SORTING
Checking Sorted File using Valsort
Records: 10000000
Checksum: 4c499aeel4ae2b
Duplicate keys: 0
SUCCESS - all records are in order
1.59user 0.00system 0:03.84elapsed 41%CPU (0avgtext+0avgdata 25500maxresident)k
1953128inputs+0outputs (0major+3351minor)pagefaults 0swaps
Deleting Temporary Chunks
```

1GB using 4 Threads:

```
ubuntu@ip-172-31-11-28:/mnt/raid/64$ vi SharedMemoryScript.sh
ubuntu@ip-172-31-11-28:/mnt/raid/64$ sh SharedMemoryScript.sh
Generating Un-Sorted file of 1GB
1.74user 0.03system 0:03.00elapsed 59%CPU (0avgtext+0avgdata 61168maxresident)k
24inputs+1955128outputs (0major+3681minor)pagefaults 0swaps
Creating file Chunks
0.00user 1.18system 0:02.34elapsed 50%CPU (0avgtext+0avgdata 760maxresident)k
1953136inputs+1956104outputs (0major+232minor)pagefaults 0swaps
File Chunks Created generated
STARTING SORT
Running Threads: 4
Sorted Chunks Created, Final Merge Phase Starts
Time Taken in millis: 30957.0
41.94user 4.56system 0:31.03elapsed 149%CPU (0avgtext+0avgdata 921544maxresident)k
376inputs+3952408outputs (0major+12034minor)pagefaults 0swaps
FINISHED SORTING
Checking Sorted File using Valsort
Records: 10000000
Checksum: 4c499aeel4ae2b
Duplicate keys: 0
SUCCESS - all records are in order
1.59user 0.05system 0:07.00elapsed 23%CPU (0avgtext+0avgdata 25496maxresident)k
1953128inputs+0outputs (0major+3350minor)pagefaults 0swaps
Deleting Temporary Chunks
```

1GB using 8 Threads:

```
ubuntu@ip-172-31-11-28:/mnt/raid/64$ sh SharedMemoryScript.sh
Generating Un-Sorted file of 1GB
1.79user 0.03system 0:02.92elapsed 62%CPU (0avgtext+0avgdata 61160maxresident)k
0inputs+1955176outputs (0major+3678minor)pagefaults 0swaps
Creating file Chunks
0.00user 1.17system 0:02.31elapsed 50%CPU (0avgtext+0avgdata 760maxresident)k
1953120inputs+1956104outputs (0major+232minor)pagefaults 0swaps
File Chunks Created generated
STARTING SORT
Running Threads: 8
Sorted Chunks Created, Final Merge Phase Starts
Time Taken in millis: 34608.0
49.18user 4.48system 0:34.69elapsed 154%CPU (0avgtext+0avgdata 1227896maxresident)k
120inputs+3933888outputs (0major+98311minor)pagefaults 0swaps
FINISHED SORTING
Checking Sorted File using Valsort
Records: 10000000
Checksum: 4c499aeel4ae2b
Duplicate keys: 0
SUCCESS - all records are in order
1.61user 0.00system 0:05.40elapsed 29%CPU (0avgtext+0avgdata 25492maxresident)k
1953128inputs+0outputs (0major+3351minor)pagefaults 0swaps
Deleting Temporary Chunks
```

Now, we evaluate the performance for multiple threads on 1 GB depicted in the following graphs and tables:

This shows the Total time taken including file generation, splitting into chunks, valsrt and Sort time:

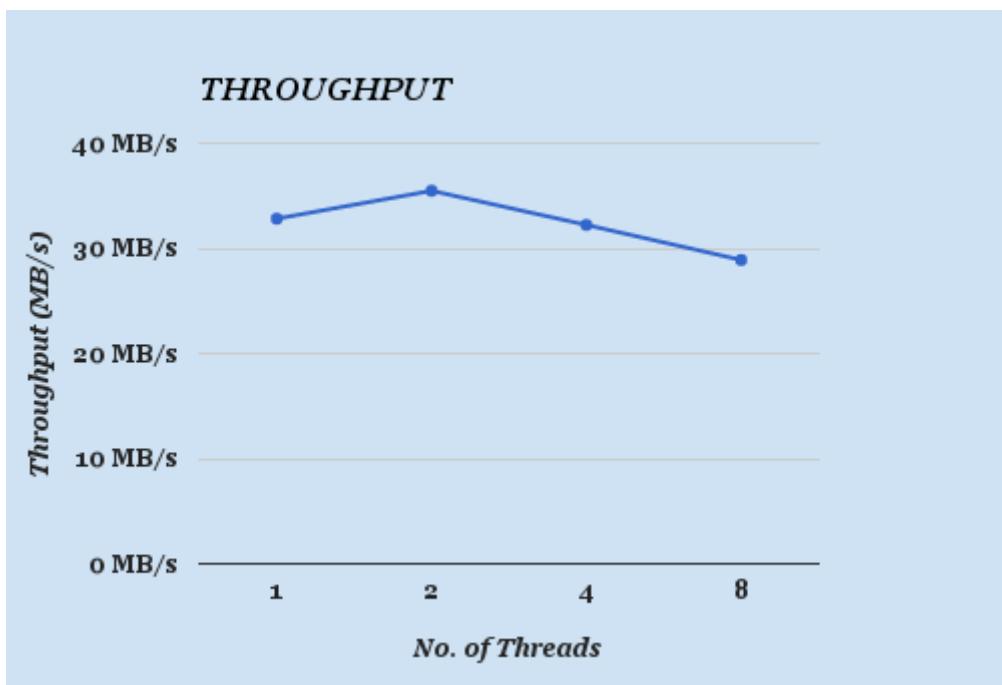
No. of Threads	Gensort (sec)	Split (sec)	Valsort (sec)	Sort (sec)	Total Time
1	2.99	2.41	4.87	30.47	40.74
2	3.03	2.32	3.92	28.19	37.46
4	3	2.34	7	31.03	43.37
8	2.92	2.31	5.4	34.61	45.24

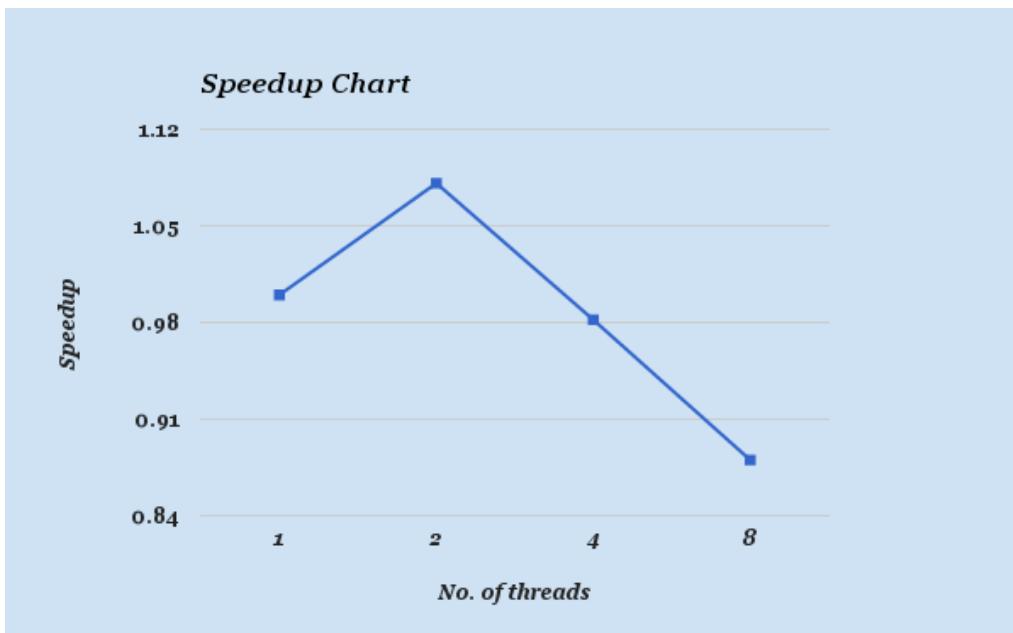
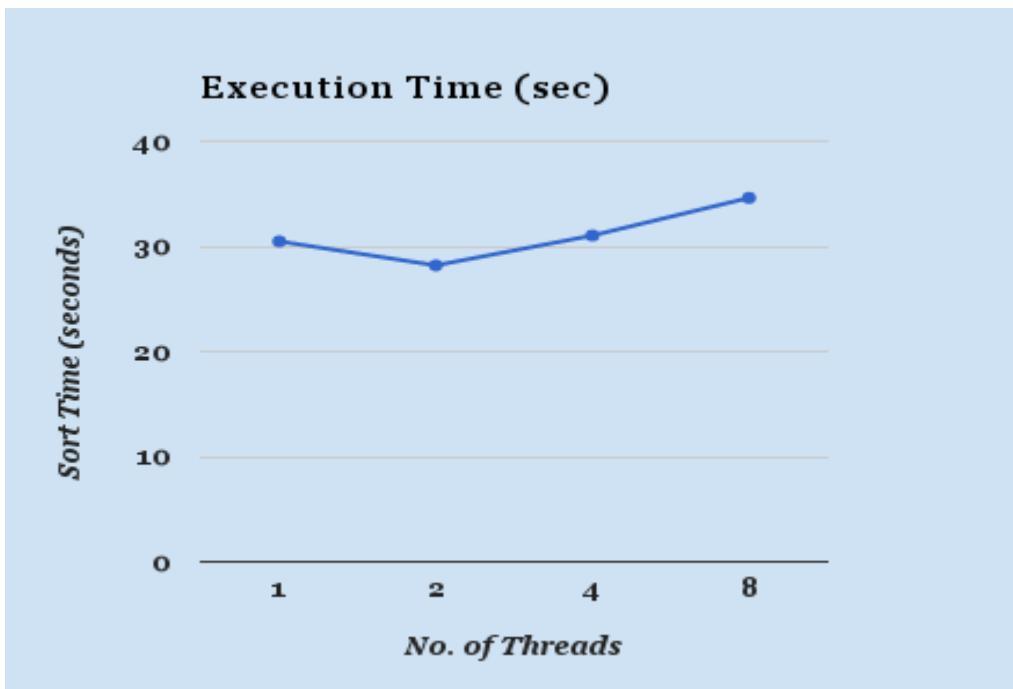
This table shows the Execution Time for 1 GB (only Sorting Time):

No. of Threads	Execution Time (Sort Time) (sec)
1	30.47
2	28.19
4	31.03
8	34.61

Graphs for Throughput and Execution Time:

As we can see from below graphs, the throughput increased as we moved from 1 thread to 2 threads, but then it started to decrement, as the instance we launched has 2 cores, it affects the performance of threads. The thread here is responsible for performing quick sort on chunks, and writing them back onto the sorted chunk file. Achieved best performance for **2 threads @ 35.47 MB/s**





The above graphs shows the Execution time and Speed up achieved for 1GB dataset.

From the execution time graph, we observe that for 1 and 2 threads, the time taken for execution goes on decreasing which is the performance we need to achieve but it is not always the case, for 4 and 8 threads the time goes on increasing.

The speedup chart shows Speedup for Shared-memory w.r.t to 1 Node 1 Thread. There is a decrease in speed-up for more than 2 threads.

10GB using 1 Thread:

```
ubuntu@ip-172-31-11-28:/mnt/raid/64$ sh SharedMemoryScript10GB.sh
Generating Un-Sorted file of 10GB
17.77user 0.72system 1:05.68elapsed 28%CPU (0avgtext+0avgdata 63032maxresident)k
240inputs+1955120outputs (0major+3681minor)pagefaults 0swaps
Creating file Chunks
0.04user 12.36system 1:12.38elapsed 17%CPU (0avgtext+0avgdata 760maxresident)k
19531520inputs+19554368outputs (0major+232minor)pagefaults 0swaps
File Chunks Created generated
STARTING SORT
Running Threads: 1
Sorted Chunks Created, Final Merge Phase Starts
Time Taken in millis: 408181.0
419.24user 36.01system 6:48.36elapsed 111%CPU (0avgtext+0avgdata 2673472maxresident)k
37853872inputs+39255016outputs (10major+515984minor)pagefaults 0swaps
FINISHED SORTING
Checking Sorted File using Valsort
Records: 100000000
Checksum: 2faf20d0000e8d8
Duplicate keys: 0
SUCCESS - all records are in order
15.89user 0.06system 0:32.92elapsed 48%CPU (0avgtext+0avgdata 25500maxresident)k
19531544inputs+0outputs (5major+3346minor)pagefaults 0swaps
```

10GB using 2 Threads:

```
Generating Un-Sorted file of 10GB
18.10user 0.23system 0:33.83elapsed 54%CPU (0avgtext+0avgdata 61164maxresident)k
216inputs+19551120outputs (1major+4189minor)pagefaults 0swaps
Creating file Chunks
0.03user 12.80system 0:31.05elapsed 41%CPU (0avgtext+0avgdata 760maxresident)k
19531312inputs+19554224outputs (1major+232minor)pagefaults 0swaps
File Chunks Created
STARTING SORT
Running Threads: 2
Sorted Chunks Created, Final Merge Phase Starts
Time Taken in millis: 394435.0
563.56user 49.02system 6:34.62elapsed 155%CPU (0avgtext+0avgdata 2992692maxresident)k
38753664inputs+39942176outputs (5major+1062799minor)pagefaults 0swaps
FINISHED SORTING
Checking Sorted File using Valsort
Records: 100000000
Checksum: 2faf20d0000e8d8
Duplicate keys: 0
SUCCESS - all records are in order
15.92user 0.06system 0:33.51elapsed 47%CPU (0avgtext+0avgdata 25496maxresident)k
19531544inputs+0outputs (5major+3345minor)pagefaults 0swaps
Deleting Temporary Chunks
```

10GB using 4 Threads:

```
ubuntu@ip-172-31-11-28:/mnt/raid/64$ sh SharedMemoryScript10GB.sh
Generating Un-Sorted file of 10GB
17.97user 0.23system 0:32.06elapsed 56%CPU (0avgtext+0avgdata 61164maxresident)k
160inputs+19551096outputs (1major+3679minor)pagefaults 0swaps
Creating file Chunks
0.04user 12.96system 0:33.25elapsed 39%CPU (0avgtext+0avgdata 760maxresident)k
19531408inputs+19555144outputs (1major+231minor)pagefaults 0swaps
File Chunks Created
STARTING SORT
Running Threads: 4
Sorted Chunks Created, Final Merge Phase Starts
Time Taken in millis: 433609.0
628.37user 50.58system 7:13.84elapsed 156%CPU (0avgtext+0avgdata 3289224maxresident)k
47868408inputs+39627760outputs (13major+1250880minor)pagefaults 0swaps
FINISHED SORTING
Checking Sorted File using Valsort
Records: 100000000
Checksum: 2faf20d0000e8d8
Duplicate keys: 0
SUCCESS - all records are in order
15.82user 0.03system 0:33.87elapsed 46%CPU (0avgtext+0avgdata 25496maxresident)k
19531672inputs+0outputs (3major+3347minor)pagefaults 0swaps
Deleting Temporary Chunks
```

10GB using 8 Threads:

```
ubuntu@ip-172-31-11-28:/mnt/raid/64$ sh SharedMemoryScript10GB.sh
Generating Un-Sorted file of 10GB
17.48user 0.11system 0:35.20elapsed 50%CPU (0avgtext+0avgdata 62664maxresident)k
960inputs+19550256outputs (1major+3680minor)pagefaults 0swaps
Creating file Chunks
0.04user 12.56system 0:29.67elapsed 42%CPU (0avgtext+0avgdata 760maxresident)k
19531920inputs+19554224outputs (1major+232minor)pagefaults 0swaps
File Chunks Created
STARTING SORT
Running Threads: 8
Sorted Chunks Created, Final Merge Phase Starts
Time Taken in millis: 1184420.0
2050.14user 45.97system 19:44.68elapsed 176%CPU (0avgtext+0avgdata 3421036maxresident)k
71205176inputs+39346576outputs (25major+726832minor)pagefaults 0swaps
FINISHED SORTING
Checking Sorted File using Valsort
Records: 100000000
Checksum: 2faf20d0000e8d8
Duplicate keys: 0
SUCCESS - all records are in order
15.82user 0.05system 0:31.73elapsed 50%CPU (0avgtext+0avgdata 25500maxresident)k
19531672inputs+0outputs (3major+3348minor)pagefaults 0swaps
Deleting Temporary Chunks
```

Now, we evaluate the performance for multiple threads on 10 GB depicted in the following graphs and tables:

This shows the Total time taken including file generation, splitting into chunks, valsrt and Sort time:

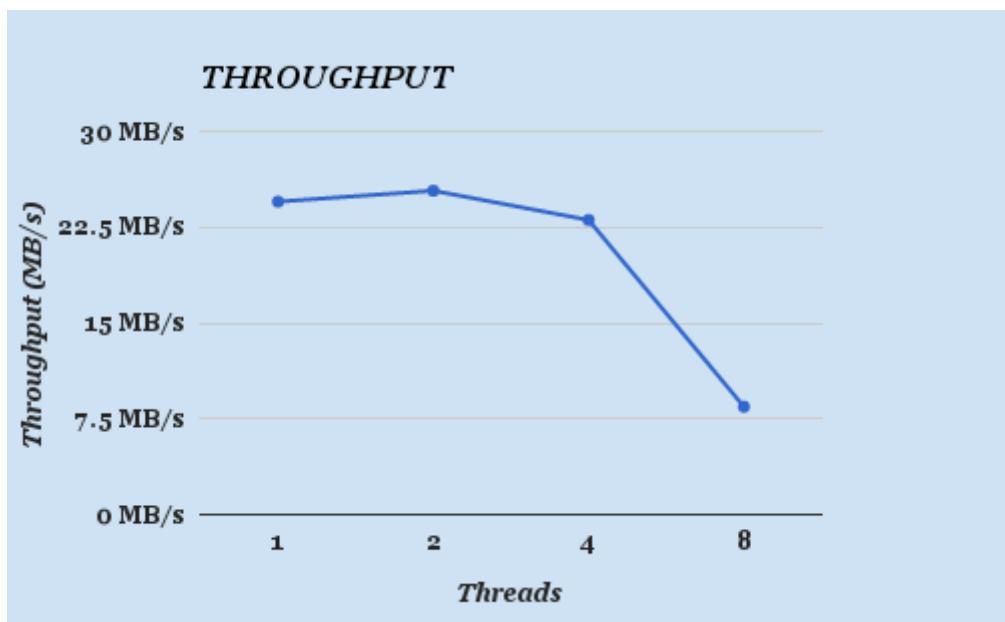
No. of Threads	Gensort (sec)	Split (sec)	Valsort (sec)	Sort (sec)	Total Time
1	63	67.2	32.92	408.18	571.3
2	33.83	31.05	33.51	394.43	492.82
4	33.06	33.25	33.87	433.61	533.79
8	35.2	29.67	31.73	1184.42	1281.02

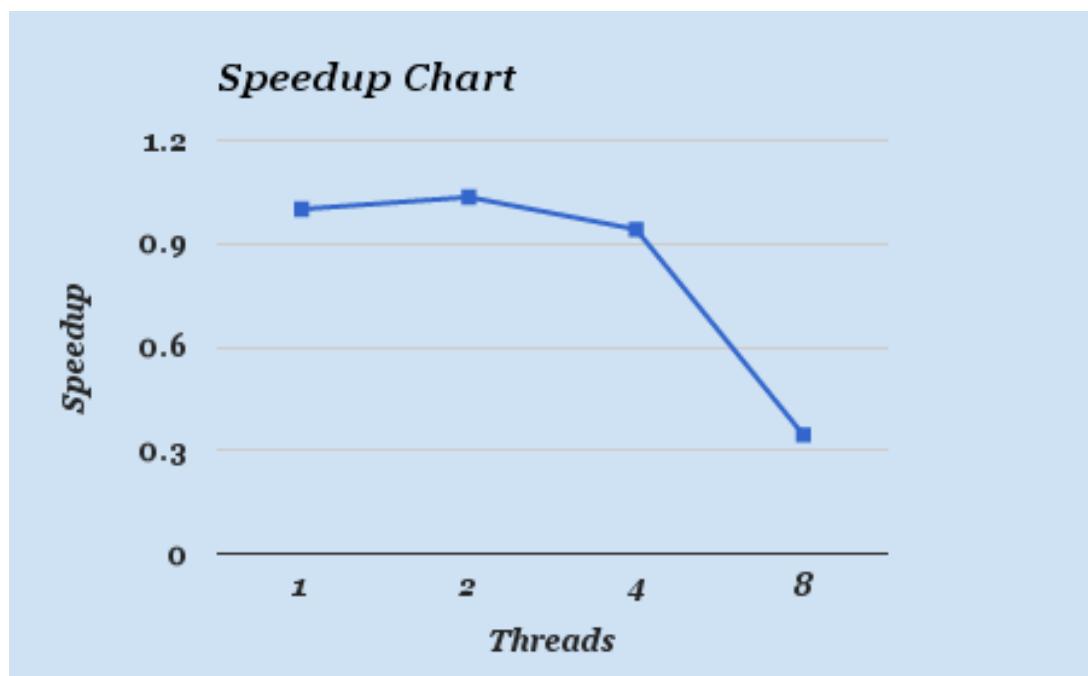
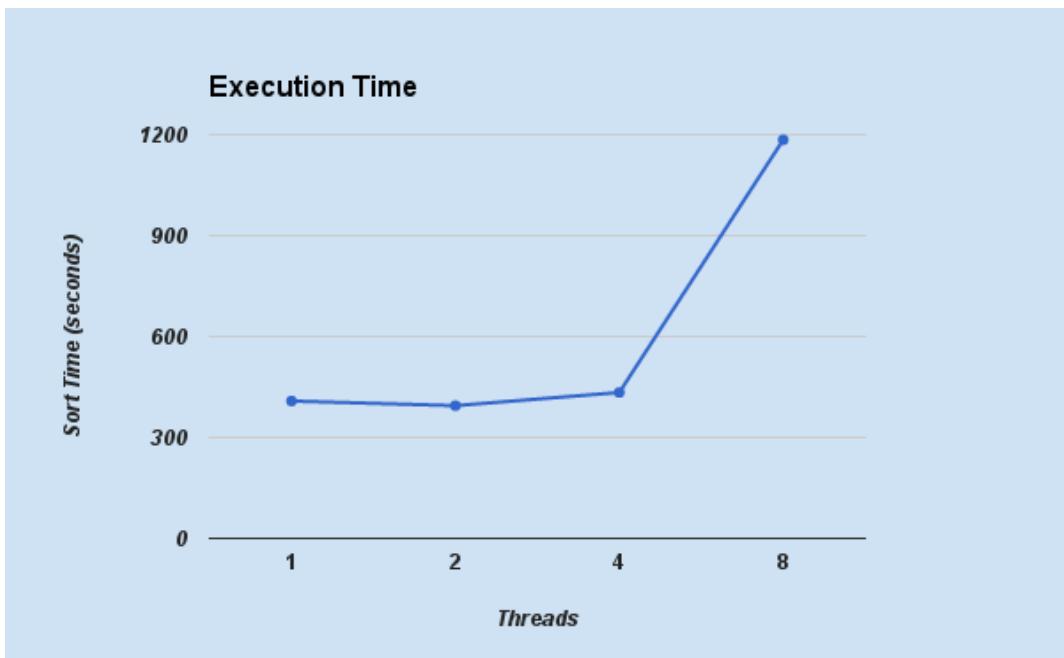
This table shows the Execution Time for 10 GB (only Sorting Time):

No. of Threads	Execution Time (Sort Time) (sec)
1	408.18
2	394.43
4	433.61
8	1184.42

Graphs for Throughput and Execution Time:

As we can see from below graphs, the throughput increased as we moved from 1 thread to 2 threads, but then it started to decrement, as the instance we launched has 2 cores, it affects the performance of threads. The throughput plummets for **8 Threads**, due to some unusual behaviour of JVM I increased the Heap size, resulting into decrease of throughput for 8 Threads. The thread here is responsible for performing quick sort on chunks, and writing them back onto the sorted chunk file. Achieved best performance for **2 threads @ 25.35 MB/s**





The above graphs shows the Execution time and Speed up achieved for 10GB dataset.

From the execution time graph, we observe that for 1, 2 threads, the time taken for execution goes on decreasing which is the performance we need to achieve but it is not always the case, for 8 threads the time goes on increasing. For 1,2,4 threads it is almost the same execution time.

The speedup chart shows Speedup for Shared-memory w.r.t to 1 Node 1 Thread. There is a decrease in speed-up for more than 2 threads.

2. HADOOP

Setting up the Hadoop Cluster:

Specifications:	
Instance Type	c3.large Ubuntu
RAM	3.75 GB
No. of Cores	2 (vCores)
Storage	2 * 16 (SSD) GB
Region	us-east-1 (N. Virginia)

Installation:

1. Launch ec2 instances from AWS EC2, which can be on demand or spot instances. In our case, I have launched spot instances of c3.large. (For 16 Nodes I have launched Master Node as c3.2xlarge and 16 Slaves as c3.large)
2. Launch 1 instance of c3.large, and then we will make AMI of it and launch 16 slaves from the same AMI.
3. Take the generated .pem file and copy onto the instance launched.
4. Now we have to add the .pem as the Identity Key, for that

```
eval $(ssh-agent)  
ssh-add <Path of .pem>
```

5. Download Hadoop by doing wget and untar it,
<http://mirrors.koehn.com/apache/hadoop/common/hadoop-2.7.2/hadoop-2.7.2.tar.gz>
6. Now we need to make changes to configuration files, to complete hadoop setup

- `~/.bashrc`
- `hadoop-env.sh`
- `core-site.xml`
- `yarn-site.xml`
- `hdfs-site.xml`
- `mapred-site.xml.template`

1. `~/.bashrc`

```
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64  
export HADOOP_HOME=/usr/local/hadoop  
export PATH=$PATH:$HADOOP_HOME/bin  
export PATH=$PATH:$HADOOP_HOME/sbin  
export HADOOP_MAPRED_HOME=$HADOOP_HOME  
export HADOOP_COMMON_HOME=$HADOOP_HOME  
export HADOOP_HDFS_HOME=$HADOOP_HOME  
export YARN_HOME=$HADOOP_HOME
```

```
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
```

Now, we need to reboot the connection, by doing source ~/.bashrc

2. hadoop-env.sh

We need to set the JAVA path here by replacing,

```
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
```

3. core-site.xml

In this file, we change the configuration properties of hadoop, setting the default name for file system and adding the path for HDFS.

```
<configuration>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://IP or DNS Name of Master:9000</value>
  </property>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>Path for HDFS /mnt/raid/</value>
  </property>
</configuration>
```

4. yarn-site.xml

```
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.resourcemanager.scheduler.address</name>
    <value>IP or DNS Name of Master:8030</value>
  </property>
  <property>
    <name>yarn.resourcemanager.address</name>
    <value>IP or DNS Name of Master:8032</value>
  </property>
  <property>
    <name>yarn.resourcemanager.webapp.address</name>
    <value>IP or DNS Name of Master:8088</value>
  </property>
</configuration>
```

```

<property>
    <name>yarn.resourcemanager.resource-tracker.address</name>
    <value> IP or DNS Name of Master:8031</value>
</property>
<property>
    <name>yarn.resourcemanager.admin.address</name>
    <value> IP or DNS Name of Master:8033</value>
</property>
</configuration>

```

5. hdfs-site.xml

In this file we change the replication factor, in our case it is set to 1, and dfs permission is also set in this file

```

<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
    <property>
        <name>dfs.permissions</name>
        <value>false</value>
    </property>
</configuration>

```

6. mapred-site.xml

This file is named by default as mapred-site.xml.template we need to rename it to mapred-site.xml

```

<configuration>
    <property>
        <name>mapreduce.jobtracker.address</name>
        <value> IP or DNS Name of Master:8021</value>
    </property>
    <property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
    </property>
</configuration>

```

We can also change the Number of Mappers and Reducers here. By adding, it here by adding the below tags or you can also add it in your java code.

```

<property>
    <name>mapred.map.tasks</name>
    <value>4</value>
</property>

<property>
    <name>mapred.reduce.tasks</name>
    <value>4</value>
</property>

```

Now these changes are to be made on 1 node Cluster as well as 16 Nodes, so we make an AMI out of it so as to reflect it on all the 16 Nodes.

For 16 Nodes, we need to make changes to one more file i.e. the **slaves** file, in /usr/local/hadoop/etc/hadoop/ and add all the **16 Slaves** IP addresses and **Master** Node IP address as well.

Screenshots:

16 Node Cluster Setup: (Master and Slaves)

The screenshot shows the AWS EC2 Dashboard. The left sidebar has a tree view with 'Instances' selected. The main area displays a table of 17 instances. The table columns are: Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, and Public DNS. The data is as follows:

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
Master	i-a21a2b26	c3.2xlarge	us-east-1c	running	2/2 checks ...	None	ec2-52-91-49-217.com
Slave1	i-4fffbcab	c3.large	us-east-1c	running	2/2 checks ...	None	ec2-54-172-247-246.com
Slave2	i-07facb83	c3.large	us-east-1c	running	2/2 checks ...	None	ec2-52-23-172-17.com
Slave3	i-04facb80	c3.large	us-east-1c	running	2/2 checks ...	None	ec2-54-164-86-217.com
Slave4	i-06facb82	c3.large	us-east-1c	running	2/2 checks ...	None	ec2-52-23-197-138.com
Slave5	i-4efbcaca	c3.large	us-east-1c	running	2/2 checks ...	None	ec2-54-152-154-122.com
Slave6	i-09facb8d	c3.large	us-east-1c	running	2/2 checks ...	None	ec2-54-84-72-53.com
Slave7	i-08facb8c	c3.large	us-east-1c	running	2/2 checks ...	None	ec2-52-23-204-68.com
Slave8	i-89fbca0d	c3.large	us-east-1c	running	2/2 checks ...	None	ec2-54-88-153-216.com
Slave9	i-5afbcade	c3.large	us-east-1c	running	2/2 checks ...	None	ec2-54-172-222-93.com

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
Slave1	i-7e0d0e0e	c3.large	us-east-1c	running	2/2 checks ...	None	ec2-52-20-204-00.com
Slave8	i-89fbca0d	c3.large	us-east-1c	running	2/2 checks ...	None	ec2-54-88-153-216.com
Slave9	i-5afbcade	c3.large	us-east-1c	running	2/2 checks ...	None	ec2-54-172-222-93.com
Slave10	i-5dfbcad9	c3.large	us-east-1c	running	2/2 checks ...	None	ec2-52-207-227-107.co
Slave11	i-5bfbcad1	c3.large	us-east-1c	running	2/2 checks ...	None	ec2-54-172-48-93.com
Slave12	i-5cfbcad8	c3.large	us-east-1c	running	2/2 checks ...	None	ec2-54-89-156-83.com
Slave13	i-8afbc0e	c3.large	us-east-1c	running	2/2 checks ...	None	ec2-54-173-16-18.com
Slave14	i-4dfbcac9	c3.large	us-east-1c	running	2/2 checks ...	None	ec2-54-164-5-30.com
Slave15	i-05facb81	c3.large	us-east-1c	running	2/2 checks ...	None	ec2-54-172-242-198.co
Slave16	i-88fbca0c	c3.large	us-east-1c	running	2/2 checks ...	None	ec2-52-91-99-244.com

Config files:

~/.bashrc

```
# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
  . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

#STARTED ADDING HADOOP VARIABLES
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64
export HADOOP_HOME=/usr/local/hadoop
export PATH=$PATH:$HADOOP_HOME/bin
export PATH=$PATH:$HADOOP_HOME/sbin
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib"
#VARIABLES ADDITION COMPLETED
-- INSERT --
```

core-site.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!!--
 Licensed under the Apache License, Version 2.0 (the "License");
 you may not use this file except in compliance with the License.
 You may obtain a copy of the License at

 http://www.apache.org/licenses/LICENSE-2.0

 Unless required by applicable law or agreed to in writing, software
 distributed under the License is distributed on an "AS IS" BASIS,
 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 See the License for the specific language governing permissions and
 limitations under the License. See accompanying LICENSE file.
-->

<!!-- Put site-specific property overrides in this file. -->

<configuration>
    <property>
        <name>fs.default.name</name>
        <value>hdfs://ec2-54-88-114-53.compute-1.amazonaws.com:8020</value>
    </property>
</configuration>
-
-
-
-
-
-
-
-- INSERT --
```

hdfs-site.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!!--
 Licensed under the Apache License, Version 2.0 (the "License");
 you may not use this file except in compliance with the License.
 You may obtain a copy of the License at

 http://www.apache.org/licenses/LICENSE-2.0

 Unless required by applicable law or agreed to in writing, software
 distributed under the License is distributed on an "AS IS" BASIS,
 WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 See the License for the specific language governing permissions and
 limitations under the License. See accompanying LICENSE file.
-->

<!!-- Put site-specific property overrides in this file. -->

<configuration>
    <property>
        <name>dfs.replication</name>
        <value>1</value>
    </property>
    <property>
        <name>dfs.permissions</name>
        <value>false</value>
    </property>
</configuration>
-
-
-
-
```

mapred-site.xml

```
<?xml version='1.0'?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>mapreduce.jobtracker.address</name>
    <value>hdfs://ec2-52-88-114-53.compute-1.amazonaws.com:8021</value>
  </property>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
~
~
~
~
-- INSERT --
```

yarn-site.xml

```
limitations under the License. See accompanying LICENSE file.
-->
<configuration>

<!-- Site specific YARN configuration properties -->

  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.resourcemanager.scheduler.address</name>
    <value>ec2-54-88-114-53.compute-1.amazonaws.com:8030</value>
  </property>
  <property>
    <name>yarn.resourcemanager.address</name>
    <value>ec2-54-88-114-53.compute-1.amazonaws.com:8032</value>
  </property>
  <property>
    <name>yarn.resourcemanager.webapp.address</name>
    <value>ec2-54-88-114-53.compute-1.amazonaws.com:8088</value>
  </property>
  <property>
    <name>yarn.resourcemanager.resource-tracker.address</name>
    <value>ec2-54-88-114-53.compute-1.amazonaws.com:8031</value>
  </property>
  <property>
    <name>yarn.resourcemanager.admin.address</name>
    <value>ec2-54-88-114-53.compute-1.amazonaws.com:8033</value>
  </property>
</configuration>
```

slaves (16 Node Setup):

```
ec2-52-91-49-217.compute-1.amazonaws.com
ec2-54-172-247-246.compute-1.amazonaws.com
ec2-52-23-172-17.compute-1.amazonaws.com
ec2-54-164-86-217.compute-1.amazonaws.com
ec2-52-23-197-138.compute-1.amazonaws.com
ec2-54-152-154-122.compute-1.amazonaws.com
ec2-54-84-72-53.compute-1.amazonaws.com
ec2-52-23-204-68.compute-1.amazonaws.com
ec2-54-88-153-216.compute-1.amazonaws.com
ec2-54-172-222-93.compute-1.amazonaws.com
ec2-52-207-227-107.compute-1.amazonaws.com
ec2-54-172-48-93.compute-1.amazonaws.com
ec2-54-89-156-83.compute-1.amazonaws.com
ec2-54-173-16-18.compute-1.amazonaws.com
ec2-54-164-5-30.compute-1.amazonaws.com
ec2-54-172-242-198.compute-1.amazonaws.com
ec2-52-91-99-244.compute-1.amazonaws.com
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

Logging into Datanode, Namenode from Master nodes to all Slaves:

```
ubuntu@ip-172-31-9-118:/usr/local/hadoop/sbin$ eval $(ssh-agent)
Agent pid 9458
ubuntu@ip-172-31-9-118:/usr/local/hadoop/sbin$ ssh-add /home/ubuntu/CloudSorting.pem
Identity added: /home/ubuntu/CloudSorting.pem (/home/ubuntu/CloudSorting.pem)
ubuntu@ip-172-31-9-118:/usr/local/hadoop/sbin$ ./start-dfs.sh
16/03/28 01:24:26 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Starting namenodes on [ec2-52-91-49-217.compute-1.amazonaws.com]
ec2-52-91-49-217.compute-1.amazonaws.com: starting namenode, logging to /usr/local/hadoop/logs/hadoop-ubuntu-namenode-ip-172-31-9-118.out
ec2-52-91-49-217.compute-1.amazonaws.com: starting datanode, logging to /usr/local/hadoop/logs/hadoop-ubuntu-datanode-ip-172-31-9-118.out
ec2-54-172-247-246.compute-1.amazonaws.com: starting datanode, logging to /usr/local/hadoop/logs/hadoop-ubuntu-datanode-ip-172-31-4-148.out
ec2-54-172-222-93.compute-1.amazonaws.com: starting datanode, logging to /usr/local/hadoop/logs/hadoop-ubuntu-datanode-ip-172-31-12-205.out
ec2-52-23-197-138.compute-1.amazonaws.com: starting datanode, logging to /usr/local/hadoop/logs/hadoop-ubuntu-datanode-ip-172-31-1-185.out
ec2-52-23-172-17.compute-1.amazonaws.com: starting datanode, logging to /usr/local/hadoop/logs/hadoop-ubuntu-datanode-ip-172-31-6-156.out
ec2-54-164-86-217.compute-1.amazonaws.com: starting datanode, logging to /usr/local/hadoop/logs/hadoop-ubuntu-datanode-ip-172-31-13-111.out
ec2-54-172-242-198.compute-1.amazonaws.com: starting datanode, logging to /usr/local/hadoop/logs/hadoop-ubuntu-datanode-ip-172-31-2-34.out
ec2-54-164-5-30.compute-1.amazonaws.com: starting datanode, logging to /usr/local/hadoop/logs/hadoop-ubuntu-datanode-ip-172-31-0-24.out
ec2-54-84-72-53.compute-1.amazonaws.com: starting datanode, logging to /usr/local/hadoop/logs/hadoop-ubuntu-datanode-ip-172-31-1-75.out
ec2-54-152-154-122.compute-1.amazonaws.com: starting datanode, logging to /usr/local/hadoop/logs/hadoop-ubuntu-datanode-ip-172-31-13-158.out
ec2-54-173-16-18.compute-1.amazonaws.com: starting datanode, logging to /usr/local/hadoop/logs/hadoop-ubuntu-datanode-ip-172-31-15-52.out
ec2-54-88-153-216.compute-1.amazonaws.com: starting datanode, logging to /usr/local/hadoop/logs/hadoop-ubuntu-datanode-ip-172-31-14-122.out
ec2-52-207-227-107.compute-1.amazonaws.com: starting datanode, logging to /usr/local/hadoop/logs/hadoop-ubuntu-datanode-ip-172-31-10-86.out
ec2-54-172-48-93.compute-1.amazonaws.com: starting datanode, logging to /usr/local/hadoop/logs/hadoop-ubuntu-datanode-ip-172-31-8-228.out
ec2-52-91-99-244.compute-1.amazonaws.com: starting datanode, logging to /usr/local/hadoop/logs/hadoop-ubuntu-datanode-ip-172-31-5-49.out
ec2-54-89-156-83.compute-1.amazonaws.com: starting datanode, logging to /usr/local/hadoop/logs/hadoop-ubuntu-datanode-ip-172-31-11-6.out
ec2-52-23-204-68.compute-1.amazonaws.com: starting datanode, logging to /usr/local/hadoop/logs/hadoop-ubuntu-datanode-ip-172-31-9-21.out
Starting secondary namenodes [0.0.0.0]
```

Logging into Nodemanager, starting yarn daemons:

```
ubuntu@ip-172-31-9-118:/usr/local/hadoop/sbin$ jps
10075 Jps
9842 DataNode
9628 NameNode
ubuntu@ip-172-31-9-118:/usr/local/hadoop/sbin$ ./start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-ubuntu-resourcemanager-ip-172-31-9-118.out
ec2-54-172-247-246.compute-1.amazonaws.com: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-ubuntu-nodemanager-ip-172-31-4-148.out
ec2-54-152-154-122.compute-1.amazonaws.com: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-ubuntu-nodemanager-ip-172-31-13-158.out
ec2-52-23-172-17.compute-1.amazonaws.com: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-ubuntu-nodemanager-ip-172-31-6-156.out
ec2-52-23-197-138.compute-1.amazonaws.com: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-ubuntu-nodemanager-ip-172-31-1-185.out
ec2-54-172-48-93.compute-1.amazonaws.com: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-ubuntu-nodemanager-ip-172-31-8-228.out
ec2-54-164-86-217.compute-1.amazonaws.com: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-ubuntu-nodemanager-ip-172-31-13-111.out
ec2-52-91-49-217.compute-1.amazonaws.com: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-ubuntu-nodemanager-ip-172-31-9-118.out
ec2-54-172-222-93.compute-1.amazonaws.com: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-ubuntu-nodemanager-ip-172-31-12-205.out
ec2-54-173-16-18.compute-1.amazonaws.com: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-ubuntu-nodemanager-ip-172-31-15-52.out
ec2-54-88-153-216.compute-1.amazonaws.com: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-ubuntu-nodemanager-ip-172-31-14-122.out
ec2-52-23-264-68.compute-1.amazonaws.com: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-ubuntu-nodemanager-ip-172-31-9-21.out
ec2-54-164-5-30.compute-1.amazonaws.com: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-ubuntu-nodemanager-ip-172-31-0-24.out
ec2-52-207-227-107.compute-1.amazonaws.com: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-ubuntu-nodemanager-ip-172-31-10-86.out
ec2-54-84-72-53.compute-1.amazonaws.com: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-ubuntu-nodemanager-ip-172-31-1-75.out
ec2-54-172-242-198.compute-1.amazonaws.com: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-ubuntu-nodemanager-ip-172-31-2-34.out
ec2-52-91-99-244.compute-1.amazonaws.com: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-ubuntu-nodemanager-ip-172-31-5-49.out
ec2-54-89-156-83.compute-1.amazonaws.com: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-ubuntu-nodemanager-ip-172-31-11-6.out
ubuntu@ip-172-31-9-118:/usr/local/hadoop/sbin$
```

The master node issues the command to start the datanode, namenode, nodemanager, resource manager on master as well as slaves. The slaves are provided with DataNode and Nodemanager by Master Node.

This can be done using the following commands (on master node):

Goto, /usr/local/hadoop/sbin/ and then type,

```
./start-dfs.sh
./start-yarn.sh
```

We can type in the command, jps to see whether nodes are connected and DataNode and Nodemanger are running or not.

After execution of hadoop we need to stop these nodes by typing,

```
./stop-dfs.sh
./stop-yarn.sh
```

With these above steps all the instances are connected and are ready to perform the map-reduce tasks.

Hadoop 1GB Sort: (1 Node)

Instance Launch:

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links: EC2 Dashboard, Events, Tags, Reports, Limits, Instances (selected), Spot Requests, Reserved Instances, Scheduled Instances, Commands, Dedicated Hosts, Images, AMIs, Bundle Tasks, Elastic Block Store, Volumes, Snapshots, Network & Security, Security Groups, and Elastic IPs. The main area has tabs for Launch Instance, Connect, and Actions. A search bar at the top says "Filter by tags and attributes or search by keyword". Below it is a table with columns: Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, and Public DNS. One row is visible: Name i-23cdffa7, Instance ID i-23cdffa7, Instance Type c3.large, Availability Zone us-east-1c, Instance State running, Status Checks 2/2 checks ..., Alarm Status None, Public DNS ec2-52-90-2-232.compute-1.amazonaws.com. Below the table, detailed information for the instance is shown in two panes. The left pane shows: Instance ID i-23cdffa7, Instance state running, Instance type c3.large, Private DNS ip-172-31-10-124.ec2.internal, Private IPs 172.31.10.124, Secondary private IPs, VPC ID vpc-c29304a6, Subnet ID subnet-eadff1c9c, Network interfaces eth0. The right pane shows: Public DNS ec2-52-90-2-232.compute-1.amazonaws.com, Public IP 52.90.2.232, Elastic IP -, Availability zone us-east-1c, Security groups launch-wizard-1, view rules, Scheduled events No scheduled events, AMI ID HadoopSpark (ami-fb131a91), Platform -, IAM role -. At the bottom, there are buttons for Feedback, English, and links to Privacy Policy and Terms of Use.

Code Run (Map-Reduce Phase):

```
ubuntu@ip-172-31-10-124:~$ hadoop jar HadoopTeraSort.jar Input.txt OutputHadoop1GB
16/03/27 20:04:01 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
16/03/27 20:04:02 INFO client.RMProxy: Connecting to ResourceManager at ec2-52-90-2-232.compute-1.amazonaws.com/172.31.10.124:8032
16/03/27 20:04:02 INFO client.RMProxy: Connecting to ResourceManager at ec2-52-90-2-232.compute-1.amazonaws.com/172.31.10.124:8032
16/03/27 20:04:02 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
16/03/27 20:04:02 INFO mapred.FileInputFormat: Total input paths to process : 1
16/03/27 20:04:02 INFO mapreduce.JobSubmitter: number of splits:8
16/03/27 20:04:03 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1459103242384_0011
16/03/27 20:04:03 INFO impl.YarnClientImpl: Submitted application application_1459103242384_0011
16/03/27 20:04:03 INFO mapreduce.Job: The url to track the job: http://ec2-52-90-2-232.compute-1.amazonaws.com:8088/proxy/application_1459103242384_0011/
16/03/27 20:04:03 INFO mapreduce.Job: Running job: job_1459103242384_0011
16/03/27 20:04:10 INFO mapreduce.Job: Job job_1459103242384_0011 running in uber mode : false
16/03/27 20:04:10 INFO mapreduce.Job: map 0% reduce 0%
16/03/27 20:04:36 INFO mapreduce.Job: map 3% reduce 0%
16/03/27 20:04:37 INFO mapreduce.Job: map 8% reduce 0%
16/03/27 20:04:40 INFO mapreduce.Job: map 19% reduce 0%
16/03/27 20:04:41 INFO mapreduce.Job: map 22% reduce 0%
16/03/27 20:04:43 INFO mapreduce.Job: map 26% reduce 0%
16/03/27 20:04:44 INFO mapreduce.Job: map 31% reduce 0%
16/03/27 20:04:46 INFO mapreduce.Job: map 32% reduce 0%
16/03/27 20:04:47 INFO mapreduce.Job: map 34% reduce 0%
16/03/27 20:04:55 INFO mapreduce.Job: map 36% reduce 0%
16/03/27 20:04:56 INFO mapreduce.Job: map 45% reduce 0%
16/03/27 20:04:57 INFO mapreduce.Job: map 46% reduce 0%
16/03/27 20:04:59 INFO mapreduce.Job: map 49% reduce 0%
16/03/27 20:05:00 INFO mapreduce.Job: map 50% reduce 0%
16/03/27 20:05:06 INFO mapreduce.Job: map 51% reduce 0%
16/03/27 20:05:07 INFO mapreduce.Job: map 52% reduce 0%
16/03/27 20:05:09 INFO mapreduce.Job: map 54% reduce 0%
16/03/27 20:05:10 INFO mapreduce.Job: map 59% reduce 0%
16/03/27 20:05:12 INFO mapreduce.Job: map 62% reduce 0%
16/03/27 20:05:13 INFO mapreduce.Job: map 67% reduce 0%
16/03/27 20:05:15 INFO mapreduce.Job: map 71% reduce 0%
16/03/27 20:05:16 INFO mapreduce.Job: map 75% reduce 0%
16/03/27 20:05:31 INFO mapreduce.Job: map 88% reduce 0%
16/03/27 20:05:33 INFO mapreduce.Job: map 88% reduce 21%
16/03/27 20:05:34 INFO mapreduce.Job: map 89% reduce 21%
16/03/27 20:05:35 INFO mapreduce.Job: map 93% reduce 21%
16/03/27 20:05:36 INFO mapreduce.Job: map 93% reduce 25%
16/03/27 20:05:37 INFO mapreduce.Job: map 96% reduce 25%
16/03/27 20:05:39 INFO mapreduce.Job: map 96% reduce 29%
16/03/27 20:05:40 INFO mapreduce.Job: map 100% reduce 29%
16/03/27 20:05:42 INFO mapreduce.Job: map 100% reduce 67%
```

Map-Reduce Complete and Sorted Output Check:

```
16/03/27 20:05:40 INFO mapreduce.Job: map 100% reduce 29%
16/03/27 20:05:42 INFO mapreduce.Job: map 100% reduce 67%
16/03/27 20:05:45 INFO mapreduce.Job: map 100% reduce 73%
16/03/27 20:05:48 INFO mapreduce.Job: map 100% reduce 79%
16/03/27 20:05:51 INFO mapreduce.Job: map 100% reduce 85%
16/03/27 20:05:54 INFO mapreduce.Job: map 100% reduce 92%
16/03/27 20:05:57 INFO mapreduce.Job: map 100% reduce 98%
16/03/27 20:05:58 INFO mapreduce.Job: map 100% reduce 100%
16/03/27 20:05:58 INFO mapreduce.Job: Job job_1459103242384_0011 completed successfully
16/03/27 20:05:58 INFO mapreduce.Job: Counters: 50
  File System Counters
    FILE: Number of bytes read=978314714
    FILE: Number of bytes written=2999373125
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=1000029672
    HDFS: Number of bytes written=1000000000
    HDFS: Number of read operations=27
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=2
  Job Counters
    Killed map tasks=1
    Launched map tasks=9
    Launched reduce tasks=1
    Data-local map tasks=9
    Total time spent by all maps in occupied slots (ms)=426786
    Total time spent by all reduces in occupied slots (ms)=40150
    Total time spent by all map tasks (ms)=426786
    Total time spent by all reduce tasks (ms)=40150
    Total vcore-milliseconds taken by all map tasks=426786
    Total vcore-milliseconds taken by all reduce tasks=40150
    Total megabyte-milliseconds taken by all map tasks=437028864
    Total megabyte-milliseconds taken by all reduce tasks=41113600
  Map-Reduce Framework
    Map input records=10000000
    Map output records=10000000
    Map output bytes=1000000000
    Map output materialized bytes=1020000048
    Input split bytes=1000
    Combine input records=0
    Combine output records=0
    Reduce input groups=10000000
    Reduce shuffle bytes=1020000048
    Reduce input records=10000000
    Reduce output records=10000000
    Spilled Records=29395241
```

Valsort gives all records are in order sorted values correctly:

```
Failed Shuffles=0
Merged Map outputs=8
GC time elapsed (ms)=7953
CPU time spent (ms)=125890
Physical memory (bytes) snapshot=2272858112
Virtual memory (bytes) snapshot=7451009024
Total committed heap usage (bytes)=1734866992
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=1000028672
File Output Format Counters
  Bytes Written=1000000000
ubuntu@ip-172-31-10-124:~$ time fs -get OutputHadoop1GB
The program 'fs' is currently not installed. You can install it by typing:
sudo apt-get install opensafs-client

real 0m0.262s
user 0m0.058s
sys 0m0.023s
ubuntu@ip-172-31-10-124:~$ time hadoop fs -get OutputHadoop1GB
16/03/27 20:09:27 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

real 0m9.431s
user 0m6.311s
sys 0m1.233s
ubuntu@ip-172-31-10-124:~$ cd 64
ubuntu@ip-172-31-10-124:~/64$ ./valsrt /home/ubuntu/OutputHadoop1GB/part-00000
Records: 10000000
Checksum: 4c4ec89af9d195
Duplicate Keys: 0
SUCCESS - all records are in order
ubuntu@ip-172-31-10-124:~/64$ time ./valsrt /home/ubuntu/OutputHadoop1GB/part-00000
Records: 10000000
Checksum: 4c4ec89af9d195
Duplicate Keys: 0
SUCCESS - all records are in order

real 0m13.950s
user 0m1.477s
sys 0m0.027s
ubuntu@ip-172-31-10-124:~/64$
```

MapReduce Job on Web App Cluster:

Shows the Application Cluster report and gives us time taken for execution for the Job.

The screenshot shows the Hadoop Application Cluster interface. On the left, there's a sidebar with navigation links like 'Cluster' (About, Nodes, Node Labels, Applications, NEW, ACCEPTED, RUNNING, FINISHED, FAILED, KILLED), 'Scheduler', and 'Tools'. The main area displays the 'Kill Application' screen for application_1459103242384_0011. It includes an 'Application Overview' section with details such as User: ubuntu, Name: HadoopTeraSort, Application Type: MAPREDUCE, Application Tags:, YarnApplicationState: FINISHED, FinalStatus Reported by AM: SUCCEEDED, Started: Sun Mar 27 20:04:03 +0000 2016, Elapsed: 1mins, 53sec, Tracking URL: History, and Diagnostics:. Below this is an 'Application Metrics' section showing Total Resource Preempted: <memory:0, vCores:0>, Total Number of Non-AM Containers Preempted: 0, Total Number of AM Containers Preempted: 0, Resource Preempted from Current Attempt: <memory:0, vCores:0>, Number of Non-AM Containers Preempted from Current Attempt: 0, and Aggregate Resource Allocation: 736526 MB-seconds, 594 vcore-seconds. At the top right, it says 'Logged in as: drwho'.

Hadoop 10 GB Sort: (1 Node)

Run Code (map-reduce):

```
gutenberg:~ drwho$ time ./gensort -a 100000000 /mnt/raid/Input.txt
real    1m27.393s
user    0m17.175s
sys     0m0.175s
ubuntu@ip-172-31-10-124:~/64$ cd ..
ubuntu@ip-172-31-10-124:~/64$ time hadoop fs -put /mnt/raid/Input.txt /Input.txt
16/03/27 20:18:30 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
real    0m57.258s
user    0m37.168s
sys     0m11.766s
ubuntu@ip-172-31-10-124:~/64$ hadoop jar HadoopTeraSort.jar /Input.txt OutputHadoop10GB
16/03/27 20:19:53 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
16/03/27 20:19:54 INFO client.RMProxy: Connecting to ResourceManager at ec2-52-90-2-232.compute-1.amazonaws.com/172.31.10.124:8032
16/03/27 20:19:54 INFO client.RMProxy: Connecting to ResourceManager at ec2-52-90-2-232.compute-1.amazonaws.com/172.31.10.124:8032
16/03/27 20:19:54 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
16/03/27 20:19:55 INFO mapred.FileInputFormat: Total input path to process : 1
16/03/27 20:19:55 INFO mapreduce.JobSubmitter: number of splits:75
16/03/27 20:19:55 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1459103242384_0012
16/03/27 20:19:55 INFO impl.YarnClientImpl: Submitted application application_1459103242384_0012
16/03/27 20:19:55 INFO mapreduce.Job: The url to track the job: http://ec2-52-90-2-232.compute-1.amazonaws.com:8088/proxy/application_1459103242384_0012/
16/03/27 20:19:55 INFO mapreduce.Job: Running job: job_1459103242384_0012
16/03/27 20:20:02 INFO mapreduce.Job: map 0% reduce 0%
16/03/27 20:20:02 INFO mapreduce.Job: map 1% reduce 0%
16/03/27 20:20:30 INFO mapreduce.Job: map 2% reduce 0%
16/03/27 20:20:36 INFO mapreduce.Job: map 3% reduce 0%
16/03/27 20:20:39 INFO mapreduce.Job: map 4% reduce 0%
16/03/27 20:20:52 INFO mapreduce.Job: map 5% reduce 0%
16/03/27 20:20:58 INFO mapreduce.Job: map 6% reduce 0%
16/03/27 20:21:04 INFO mapreduce.Job: map 7% reduce 0%
16/03/27 20:21:07 INFO mapreduce.Job: map 8% reduce 0%
16/03/27 20:21:32 INFO mapreduce.Job: map 9% reduce 0%
16/03/27 20:21:35 INFO mapreduce.Job: map 10% reduce 0%
16/03/27 20:21:38 INFO mapreduce.Job: map 11% reduce 0%
16/03/27 20:21:42 INFO mapreduce.Job: map 12% reduce 0%
16/03/27 20:21:51 INFO mapreduce.Job: map 13% reduce 0%
16/03/27 20:22:00 INFO mapreduce.Job: map 14% reduce 0%
16/03/27 20:22:05 INFO mapreduce.Job: map 15% reduce 0%
16/03/27 20:22:09 INFO mapreduce.Job: map 16% reduce 0%
16/03/27 20:22:35 INFO mapreduce.Job: map 17% reduce 0%
16/03/27 20:22:38 INFO mapreduce.Job: map 18% reduce 0%
16/03/27 20:22:41 INFO mapreduce.Job: map 19% reduce 0%
16/03/27 20:22:45 INFO mapreduce.Job: map 20% reduce 0%
```

Map-Reduce (In Progress):

```
16/03/27 20:31:06 INFO mapreduce.Job: map 71% reduce 23%
16/03/27 20:31:12 INFO mapreduce.Job: map 71% reduce 24%
16/03/27 20:31:27 INFO mapreduce.Job: map 72% reduce 24%
16/03/27 20:31:30 INFO mapreduce.Job: map 73% reduce 24%
16/03/27 20:31:33 INFO mapreduce.Job: map 74% reduce 24%
16/03/27 20:31:43 INFO mapreduce.Job: map 75% reduce 24%
16/03/27 20:31:52 INFO mapreduce.Job: map 76% reduce 24%
16/03/27 20:31:55 INFO mapreduce.Job: map 77% reduce 24%
16/03/27 20:32:01 INFO mapreduce.Job: map 77% reduce 25%
16/03/27 20:32:07 INFO mapreduce.Job: map 78% reduce 25%
16/03/27 20:32:20 INFO mapreduce.Job: map 79% reduce 26%
16/03/27 20:32:24 INFO mapreduce.Job: map 80% reduce 26%
16/03/27 20:32:28 INFO mapreduce.Job: map 81% reduce 26%
16/03/27 20:32:44 INFO mapreduce.Job: map 82% reduce 26%
16/03/27 20:32:54 INFO mapreduce.Job: map 83% reduce 26%
16/03/27 20:33:02 INFO mapreduce.Job: map 84% reduce 26%
16/03/27 20:33:08 INFO mapreduce.Job: map 84% reduce 27%
16/03/27 20:33:23 INFO mapreduce.Job: map 85% reduce 28%
16/03/27 20:33:28 INFO mapreduce.Job: map 86% reduce 28%
16/03/27 20:33:32 INFO mapreduce.Job: map 87% reduce 28%
16/03/27 20:33:42 INFO mapreduce.Job: map 88% reduce 28%
16/03/27 20:33:48 INFO mapreduce.Job: map 89% reduce 28%
16/03/27 20:33:54 INFO mapreduce.Job: map 90% reduce 28%
16/03/27 20:34:00 INFO mapreduce.Job: map 91% reduce 29%
16/03/27 20:34:01 INFO mapreduce.Job: map 91% reduce 30%
16/03/27 20:34:06 INFO mapreduce.Job: map 92% reduce 30%
16/03/27 20:34:25 INFO mapreduce.Job: map 93% reduce 30%
16/03/27 20:34:30 INFO mapreduce.Job: map 94% reduce 30%
16/03/27 20:34:46 INFO mapreduce.Job: map 95% reduce 30%
16/03/27 20:34:54 INFO mapreduce.Job: map 96% reduce 30%
16/03/27 20:35:00 INFO mapreduce.Job: map 97% reduce 30%
16/03/27 20:35:02 INFO mapreduce.Job: map 97% reduce 31%
16/03/27 20:35:08 INFO mapreduce.Job: map 97% reduce 32%
16/03/27 20:35:18 INFO mapreduce.Job: map 99% reduce 32%
16/03/27 20:35:27 INFO mapreduce.Job: map 100% reduce 32%
16/03/27 20:35:29 INFO mapreduce.Job: map 100% reduce 33%
16/03/27 20:35:53 INFO mapreduce.Job: map 100% reduce 54%
16/03/27 20:35:56 INFO mapreduce.Job: map 100% reduce 67%
16/03/27 20:36:02 INFO mapreduce.Job: map 100% reduce 68%
16/03/27 20:36:08 INFO mapreduce.Job: map 100% reduce 69%
16/03/27 20:36:11 INFO mapreduce.Job: map 100% reduce 70%
16/03/27 20:36:17 INFO mapreduce.Job: map 100% reduce 71%
16/03/27 20:36:20 INFO mapreduce.Job: map 100% reduce 72%
```

Map-Reduce Completed (Sorted Data):

```
HDFS: Number of write operations=2
Job Counters
  Killed map tasks=1
  Launched map tasks=76
  Launched reduce tasks=1
  Data-local map tasks=76
  Total time spent by all maps in occupied slots (ms)=4675085
  Total time spent by all reduces in occupied slots (ms)=927371
  Total time spent by all map tasks (ms)=4675085
  Total time spent by all reduce tasks (ms)=927371
  Total vcore-milliseconds taken by all map tasks=4675085
  Total vcore-milliseconds taken by all reduce tasks=927371
  Total megabyte-milliseconds taken by all map tasks=478287040
  Total megabyte-milliseconds taken by all reduce tasks=949627904
Map-Reduce Framework
  Map input records=100000000
  Map output records=100000000
  Map output bytes=10000000000
  Map output materialized bytes=10200000450
  Input split bytes=8475
  Combine input records=0
  Combine output records=0
  Reduce input groups=1000000000
  Reduce shuffle bytes=10200000450
  Reduce input records=100000000
  Reduce output records=100000000
  Spilled Records=396636763
  Shuffled Maps =75
  Failed Shuffles=0
  Merged Map outputs=75
  GC time elapsed (ms)=76272
  CPU time spent (ms)=1444940
  Physical memory (bytes) snapshot=195954606008
  Virtual memory (bytes) snapshot=62892351488
  Total committed heap usage (bytes)=15641668192
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=100000303104
File Output Format Counters
  Bytes Written=10000000000
ubuntu@ip-172-31-10-124:~$
```

Get the data from file system and compute the Valsort Phase on Sorted Data:

```
ubuntu@ip-172-31-10-124:~/64$ ./valsrt /mnt/raid/dfs/
data/ name/
ubuntu@ip-172-31-10-124:~/64$ ./valsrt /mnt/raid/dfs/data/
current/    in_use.lock
ubuntu@ip-172-31-10-124:~/64$ ./valsrt /mnt/raid/dfs/data/current/
BP-1457992133-172.31.10.124-1459102732508/ VERSION
ubuntu@ip-172-31-10-124:~/64$ ./valsrt /mnt/raid/
dfs/      Input.txt  lost+found/  nm-local-dir/
ubuntu@ip-172-31-10-124:~/64$ ./valsrt /mnt/raid/nm-local-dir/
filecache/ nmPrivate/ usercache/
ubuntu@ip-172-31-10-124:~/64$ cd ..
ubuntu@ip-172-31-10-124:~$ cd /mnt/raid/
ubuntu@ip-172-31-10-124:/mnt/raid$ hadoop fs -get OutputHadoop10GB
16/03/27 20:45:03 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
ubuntu@ip-172-31-10-124:/mnt/raid$ ls
dfs  Input.txt  lost+found  nm-local-dir  OutputHadoop10GB
ubuntu@ip-172-31-10-124:/mnt/raid$ cd ~
ubuntu@ip-172-31-10-124:~$ cd 64
ubuntu@ip-172-31-10-124:~/64$ time ./valsrt /mnt/raid/OutputHadoop10GB/part-00000
Records: 100000000
Checksum: 2fb0574596d67c8
Duplicate keys: 0
SUCCESS - all records are in order

real   1m19.181s
user   0m15.190s
sys    0m0.079s
```

MapReduce Job on Web App Cluster:

Shows the Application Cluster report and gives us time taken for execution for the Job.

The screenshot shows the Hadoop Web UI interface. At the top, there is a logo of a yellow elephant and the word "hadoop". To the right, it says "Logged in as: drwho". Below the logo, the title "Application application_1459103242384_0012" is displayed. On the left, there is a sidebar with a tree view under "Cluster" and a "Tools" section. The main area has two tabs: "Kill Application" and "Application Overview". The "Application Overview" tab is active, showing details about the application. The application details include:

Application Overview	
User:	ubuntu
Name:	HadoopTeraSort
Application Type:	MAPREDUCE
Application Tags:	
YarnApplicationState:	FINISHED
FinalStatus Reported by AM:	SUCCEEDED
Started:	Sun Mar 27 20:19:55 +0000 2016
Elapsed:	18mins, 44sec
Tracking URL:	History
Diagnostics:	

Below this, there is another tab labeled "Application Metrics" which displays resource usage statistics:

Application Metrics	
Total Resource Preempted:	<memory:0, vCores:0>
Total Number of Non-AM Containers Preempted:	0
Total Number of AM Containers Preempted:	0
Resource Preempted from Current Attempt:	<memory:0, vCores:0>
Number of Non-AM Containers Preempted from Current Attempt:	0
Aggregate Resource Allocation:	8153722 MB-seconds, 6791 vcore-seconds

Hadoop 100 GB Sort: (16 Nodes)

Screenshot for Instance launch were added earlier in Configuration Setup Step.

Run Code (Map-Reduce):

```
ubuntu@ip-172-31-9-118:~/64$ time hadoop fs -put /mnt/raid/Input.txt /Input.txt
16/03/28 02:00:25 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
real    16m14.28s
user    4m57.01s
sys     1m29.37s
ubuntu@ip-172-31-9-118:~/64$ cd ~
ubuntu@ip-172-31-9-118:~$ hadoop jar HadoopTeraSort.jar /Input.txt Output10GB
16/03/28 02:20:52 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
16/03/28 02:20:53 INFO client.RMProxy: Connecting to ResourceManager at ec2-52-91-49-217.compute-1.amazonaws.com/172.31.9.118:8082
16/03/28 02:20:53 INFO client.RMProxy: Connecting to ResourceManager at ec2-52-91-49-217.compute-1.amazonaws.com/172.31.9.118:8082
16/03/28 02:20:53 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
16/03/28 02:20:54 INFO mapred.FileInputFormat: Total input paths to process : 1
16/03/28 02:20:54 INFO net.NetworkTopology: Adding a new node: /default-rack/172.31.9.118:50010
16/03/28 02:20:54 INFO mapreduce.JobSubmitter: number of splits:745
16/03/28 02:20:54 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1459126750213_0002
16/03/28 02:20:54 INFO impl.YarnClientImpl: Submitted application application_1459126750213_0002
16/03/28 02:20:54 INFO mapreduce.Job: The url to track the job: http://ec2-52-91-49-217.compute-1.amazonaws.com:8088/proxy/application_1459126750213_0002/
16/03/28 02:20:54 INFO mapreduce.Job: Running job: job_1459126750213_0002
16/03/28 02:21:03 INFO mapreduce.Job: Job job_1459126750213_0002 running in uber mode : false
16/03/28 02:21:03 INFO mapreduce.Job: map 0% reduce 0%
16/03/28 02:21:17 INFO mapreduce.Job: map 1% reduce 0%
16/03/28 02:21:22 INFO mapreduce.Job: map 2% reduce 0%
16/03/28 02:21:29 INFO mapreduce.Job: map 3% reduce 0%
16/03/28 02:21:39 INFO mapreduce.Job: map 4% reduce 0%
16/03/28 02:21:45 INFO mapreduce.Job: map 5% reduce 0%
16/03/28 02:21:50 INFO mapreduce.Job: map 6% reduce 0%
16/03/28 02:21:59 INFO mapreduce.Job: map 7% reduce 0%
16/03/28 02:22:03 INFO mapreduce.Job: map 8% reduce 0%
16/03/28 02:22:09 INFO mapreduce.Job: map 9% reduce 0%
16/03/28 02:22:17 INFO mapreduce.Job: map 10% reduce 0%
16/03/28 02:22:25 INFO mapreduce.Job: map 11% reduce 0%
16/03/28 02:22:32 INFO mapreduce.Job: map 12% reduce 0%
16/03/28 02:22:40 INFO mapreduce.Job: map 13% reduce 0%
16/03/28 02:22:42 INFO mapreduce.Job: map 13% reduce 1%
16/03/28 02:22:47 INFO mapreduce.Job: map 14% reduce 1%
16/03/28 02:22:54 INFO mapreduce.Job: map 15% reduce 1%
16/03/28 02:23:00 INFO mapreduce.Job: map 16% reduce 1%
16/03/28 02:23:04 INFO mapreduce.Job: map 17% reduce 1%
16/03/28 02:23:15 INFO mapreduce.Job: map 18% reduce 1%
16/03/28 02:23:21 INFO mapreduce.Job: map 19% reduce 1%
16/03/28 02:23:28 INFO mapreduce.Job: map 20% reduce 1%
```

DFS Overview:

I have Mounted 400 GB EBS Volume on Master Node (c3.2xlarge), and added 20 GB of EBS on each Slave (c3.large)

18 files and directories, 19 blocks = 37 total filesystem object(s).

Heap Memory used 72.81 MB of 285 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 31.43 MB of 32.94 MB Committed Non Heap Memory. Max Non Heap Memory is 214 MB.

Configured Capacity:	1.32 TB
DFS Used:	1.88 GB (0.14%)
Non DFS Used:	70.22 GB
DFS Remaining:	1.24 TB (94.65%)
Block Pool Used:	1.88 GB (0.14%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 1.89% / 0.44%
Live Nodes	17 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)
Decommissioning Nodes	0
Total Datanode Volume Failures	0 (0 B)
Number of Under-Replicated Blocks	0
Number of Blocks Pending Deletion	0

DataNodes:

Node	Last contact	Admin State	Capacity	Used	NON DFS Used	Remaining	Blocks	Block pool used	Failed Volumes	Version
ip-172-31-13-111.ec2.internal:50010 (172.31.13.111:50010)	1	In Service	49.72 GB	48 KB	45.52 GB	4.2 GB	1	48 KB (0%)	0	2.7.2
ip-172-31-13-158.ec2.internal:50010 (172.31.13.158:50010)	1	In Service	49.72 GB	132 KB	6.53 GB	43.18 GB	1	132 KB (0%)	0	2.7.2
ip-172-31-9-118.ec2.internal:50010 (172.31.9.118:50010)	1	In Service	551.19 GB	93.96 GB	149.66 GB	307.57 GB	753	93.96 GB (17.05%)	0	2.7.2
ip-172-31-2-34.ec2.internal:50010 (172.31.2.34:50010)	1	In Service	49.72 GB	132 KB	7.3 GB	42.42 GB	1	132 KB (0%)	0	2.7.2
ip-172-31-1-185.ec2.internal:50010 (172.31.1.185:50010)	1	In Service	49.72 GB	48 KB	6.15 GB	43.57 GB	1	48 KB (0%)	0	2.7.2
ip-172-31-9-21.ec2.internal:50010 (172.31.9.21:50010)	1	In Service	49.72 GB	48 KB	7.04 GB	42.67 GB	1	48 KB (0%)	0	2.7.2
ip-172-31-4-148.ec2.internal:50010 (172.31.4.148:50010)	1	In Service	49.72 GB	48 KB	6.66 GB	43.06 GB	1	48 KB (0%)	0	2.7.2
ip-172-31-8-228.ec2.internal:50010 (172.31.8.228:50010)	1	In Service	49.72 GB	132 KB	5.64 GB	44.08 GB	1	132 KB (0%)	0	2.7.2
ip-172-31-12-205.ec2.internal:50010 (172.31.12.205:50010)	1	In Service	49.72 GB	48 KB	8.32 GB	41.4 GB	1	48 KB (0%)	0	2.7.2
ip-172-31-5-49.ec2.internal:50010 (172.31.5.49:50010)	1	In Service	49.72 GB	136 KB	7.55 GB	42.16 GB	2	136 KB (0%)	0	2.7.2
ip-172-31-6-156.ec2.internal:50010 (172.31.6.156:50010)	1	In Service	49.72 GB	140 KB	7.17 GB	42.55 GB	2	140 KB (0%)	0	2.7.2
ip-172-31-14-122.ec2.internal:50010 (172.31.14.122:50010)	1	In Service	49.72 GB	132 KB	31.26 GB	18.46 GB	1	132 KB (0%)	0	2.7.2
ip-172-31-10-86.ec2.internal:50010 (172.31.10.86:50010)	1	In Service	49.72 GB	5.32 MB	7.04 GB	42.68 GB	1	5.32 MB (0.01%)	0	2.7.2
ip-172-31-1-75.ec2.internal:50010 (172.31.1.75:50010)	1	In Service	49.72 GB	128 KB	8.32 GB	41.4 GB	1	128 KB (0%)	0	2.7.2
ip-172-31-15-52.ec2.internal:50010 (172.31.15.52:50010)	1	In Service	49.72 GB	36 KB	7.43 GB	42.29 GB	0	36 KB (0%)	0	2.7.2
ip-172-31-0-24.ec2.internal:50010 (172.31.0.24:50010)	1	In Service	49.72 GB	136 KB	7.55 GB	42.16 GB	2	136 KB (0%)	0	2.7.2

Running Of Map-Reduce:

Task Type	Progress	Total	Pending	Running	Complete
Map	<div style="width: 745px;"></div>	745	122	49	574
Reduce	<div style="width: 4px;"></div>	4	0	4	0
Attempt Type	New	Running	Failed	Killed	Successful
Maps	122	49	0	0	574
Reduces	0	4	0	0	0

Map-Reduce (Running):

```

16/03/28 04:06:24 INFO mapreduce.Job: map 100% reduce 48%
16/03/28 04:06:30 INFO mapreduce.Job: map 100% reduce 49%
16/03/28 04:06:36 INFO mapreduce.Job: map 100% reduce 50%
16/03/28 04:06:45 INFO mapreduce.Job: map 100% reduce 51%
16/03/28 04:06:51 INFO mapreduce.Job: map 100% reduce 52%
16/03/28 04:06:57 INFO mapreduce.Job: map 100% reduce 53%
16/03/28 04:07:04 INFO mapreduce.Job: map 100% reduce 54%
16/03/28 04:07:12 INFO mapreduce.Job: map 100% reduce 55%
16/03/28 04:07:19 INFO mapreduce.Job: map 100% reduce 56%
16/03/28 04:07:26 INFO mapreduce.Job: map 100% reduce 57%
16/03/28 04:07:34 INFO mapreduce.Job: map 100% reduce 58%
16/03/28 04:07:40 INFO mapreduce.Job: map 100% reduce 59%
16/03/28 04:07:47 INFO mapreduce.Job: map 100% reduce 60%
16/03/28 04:07:55 INFO mapreduce.Job: map 100% reduce 61%
16/03/28 04:08:01 INFO mapreduce.Job: map 100% reduce 62%
16/03/28 04:08:07 INFO mapreduce.Job: map 100% reduce 63%
16/03/28 04:08:16 INFO mapreduce.Job: map 100% reduce 64%
16/03/28 04:08:22 INFO mapreduce.Job: map 100% reduce 65%
16/03/28 04:08:31 INFO mapreduce.Job: map 100% reduce 66%
16/03/28 04:08:40 INFO mapreduce.Job: map 100% reduce 67%
16/03/28 04:08:50 INFO mapreduce.Job: map 100% reduce 68%
16/03/28 04:09:02 INFO mapreduce.Job: map 100% reduce 69%
16/03/28 04:09:14 INFO mapreduce.Job: map 100% reduce 70%
16/03/28 04:09:26 INFO mapreduce.Job: map 100% reduce 71%
16/03/28 04:09:38 INFO mapreduce.Job: map 100% reduce 72%
16/03/28 04:09:50 INFO mapreduce.Job: map 100% reduce 73%
16/03/28 04:10:02 INFO mapreduce.Job: map 100% reduce 74%
16/03/28 04:10:14 INFO mapreduce.Job: map 100% reduce 75%
16/03/28 04:10:26 INFO mapreduce.Job: map 100% reduce 76%
16/03/28 04:10:38 INFO mapreduce.Job: map 100% reduce 77%
16/03/28 04:10:50 INFO mapreduce.Job: map 100% reduce 78%
16/03/28 04:11:02 INFO mapreduce.Job: map 100% reduce 79%
16/03/28 04:11:14 INFO mapreduce.Job: map 100% reduce 80%

```

Final Output:

```

Map -Reduce Framework
  Map input records=1000000000
  Map output records=1000000000
  Map output bytes=1000000000000
  Map output materialized bytes=102000004470
  Input split bytes=87910
  Combine input records=0
  Combine output records=0
  Reduce input groups=1000000000
  Reduce shuffle bytes=102000004470
  Reduce input records=1000000000
  Reduce output records=1000000000
  Spilled Records=6174638893
  Shuffled Maps =745
  Failed Shuffles=0
  Merged Map outputs=745
  GC time elapsed (ms)=448099
  CPU time spent (ms)=11916950
  Physical memory (bytes) snapshot=201843904512
  Virtual memory (bytes) snapshot=1439412592640
  Total committed heap usage (bytes)=154459439104
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=100003047424
File Output Format Counters
  Bytes Written=1000000000000

```

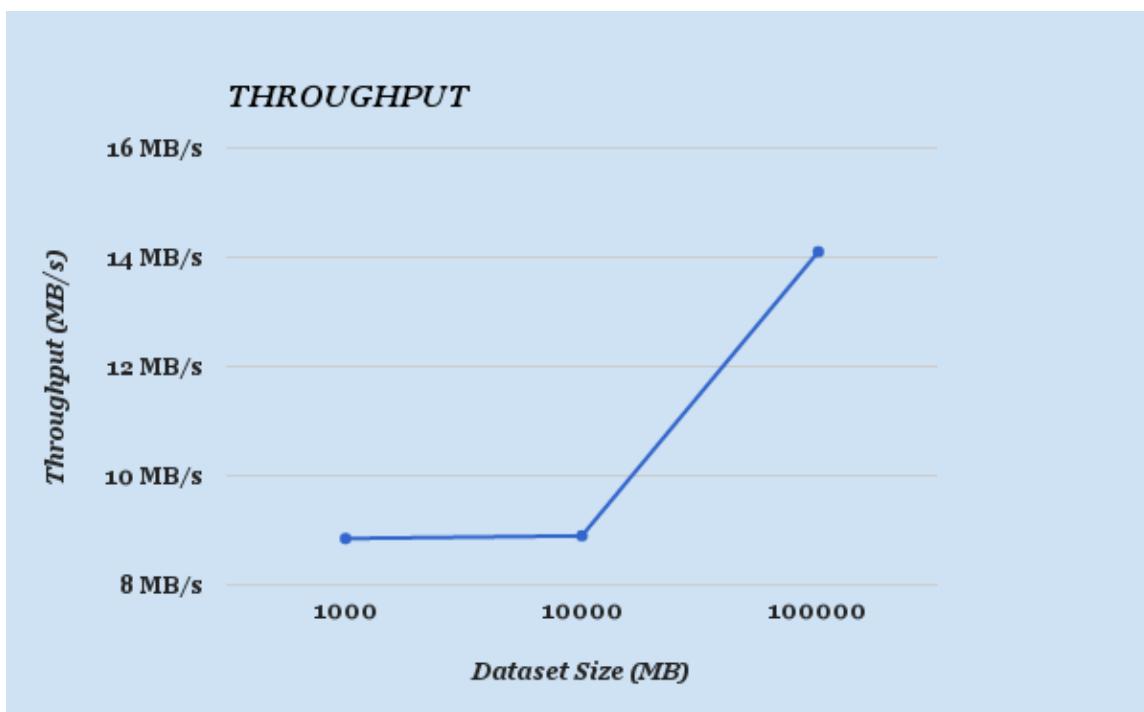
The below table shows the performance for Hadoop on 1GB, 10GB and 100GB:

Size (GB)	Sort Time (Execution Time) (sec)	Throughput (MB/s)
1 (1 Node)	113 seconds	8.85 MB/s
10 (1 Node)	1124 seconds	8.90 MB/s
100 (16 Nodes)	7092 seconds	14.10 MB/s

For 100 GB dataset, while the final output was generated the Instance Stopped due to price termination. Because of which I estimated the time for execution based on the screenshots. I generated the output files, when compared with output of first 10 and last 10 lines with SPARK they were similar. So the data was sorted successfully.

The time taken for 1 Node clusters is approximately similar when compared to dataset size given to sort, a small increment is observed in throughput.

The 100 GB dataset was implemented over 16 Nodes Cluster, there were many issues while implementing this. The map reduce phase started again in between, in my cases the map completed 100% and reduce again went from 90% to 13% thus increasing the time of execution. But, then too it achieved a higher throughput than 1 Node, as the data is divided over nodes, and each slave works on its given data. I have taken 4 Reducers in my case, by setting the number of reducers through code.





3. SPARK

Setting up the Spark Cluster:

Specifications:	
Instance Type	c3.large Ubuntu
RAM	3.75 GB
No. of Cores	2 (vCores)
Storage	2 * 16 (SSD) GB
Region	us-east-1 (N. Virginia)

Installation:

- After downloading spark, appropriate version, spark itself has the functionality to launch Master and Slaves on ec2 instances.
- Spark also has these execution scripts which will launch for us the Master's and Slaves' which we specify using ./spark-ec2 command, but before that we need to do following things:
 - Create a Key Pair (.pem) file
 - Goto Security Credentials, and create a new Access Key, which provides with Access Key ID and Secret Access Key, save the .csv file on your local machine
- After performing the above two things, launch a c3.large instance and in its terminal type,

```
export AWS_ACCESS_KEY_ID=<YOUR ACCESS KEY ID>
export AWS_SECRET_ACCESS_KEY=<YOUR SECRET KEY>
```

- Now, we are ready to launch the instances, for setting up 16 Node Cluster.
- Goto spark/ec2 folder in your downloaded version spark folder
- Type in,
./spark-ec2 -k <name of pem file without extension> -i <path of ur .pem file>
-s <no. of slaves> --instance-type=<type of instance our case c3.large> --ebs-vol-size=<vol size in GB> --spot-price=<price> -r <region where to launch> -m
c3.4xlarge launch <name for launching these instances>

In my case, example will be like

```
./spark-ec2 -k CloudSorting -i /home/ubuntu/CloudSorting.pem -s 16 --instance-type=c3.large --ebs-vol-size=50 --spot-price=0.025 -r us-east-1 -m c3.4xlarge launch  
sparkInstances
```

- I have launched the Master instance as c3.4xlarge as it has more storage capacity than c3.large to store the 100 GB data, by specifying -m for master instance, this instance will be an on demand instance type.
- For slaves, -s 16 will launch 16 slaves c3.large instance with extra storage of 50GB to avoid data spills and capacity to get exceeded while running the sort functionality of 100 GB
- Once given this command, spark will launch for you all the instances, connect you masters and slaves, generate security groups itself, will install all the necessary packages for you on the nodes.

These steps setups up your 16 node virtual cluster environment on spark.

Screenshots:

Launching 1 Master and 1 Slave:

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links for EC2 Dashboard, Events, Tags, Reports, Limits, Instances (selected), Spot Requests, Reserved Instances, Scheduled Instances, Commands, Dedicated Hosts, Images (AMIs), and Elastic Block Store (Volumes, Snapshots). The main area has tabs for Launch Instance, Connect, and Actions. A search bar at the top says "Filter by tags and attributes or search by keyword". Below it is a table with columns: Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, and Public IP. There are three rows: 1. SparkInstance (selected, highlighted in blue), Instance ID: i-8e86b60a, Instance Type: c3.large, Availability Zone: us-east-1c, Instance State: running, Status Checks: 2/2 check..., Alarm Status: None, Public IP: ec2-54-1. 2. SparkSortCluster-master-i-c7252a3e, Instance ID: i-c7252a3e, Instance Type: c3.large, Availability Zone: us-east-1e, Instance State: running, Status Checks: 2/2 check..., Alarm Status: None, Public IP: ec2-52-8. 3. SparkSortCluster-slave-i-76252a8f, Instance ID: i-76252a8f, Instance Type: c3.large, Availability Zone: us-east-1e, Instance State: running, Status Checks: 2/2 check..., Alarm Status: None, Public IP: ec2-52-8. At the bottom, there's a detailed view for the selected instance (SparkInstance) with tabs for Description, Status Checks, Monitoring, and Tags. It shows Instance ID: i-8e86b60a, Public DNS: ec2-54-174-102-173.compute-1.amazonaws.com, Instance state: running, Public DNS: ec2-54-174-102-173.compute-1.amazonaws.com, and Public IP: 54.174.102.173. The footer includes links for Feedback, English, Privacy Policy, and Terms of Use, along with a copyright notice: © 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.

For 1GB Sort:

Launch Script:

```
ubuntu@ip-172-31-0-170:~/spark-1.6.0-bin-hadoop2.6$ cd ec2/
ubuntu@ip-172-31-0-170:~/spark-1.6.0-bin-hadoop2.6/ec2$ ls
deploy.generic lib README spark-ec2 spark_ec2.py
ubuntu@ip-172-31-0-170:~/spark-1.6.0-bin-hadoop2.6/ec2$ ./spark-ec2 -k CloudSorting -i /home/ubuntu/CloudSorting.pem -s1 -t c3.large -r us-east-1 --spot-price=0.025 --ebs-vol-size=30 launch $sparkSortCluster
ERROR: The environment variable AWS_ACCESS_KEY_ID must be set
ubuntu@ip-172-31-0-170:~/spark-1.6.0-bin-hadoop2.6/ec2$ export AWS_ACCESS_KEY_ID=AKIAIWJN7JKLENXISXA
ubuntu@ip-172-31-0-170:~/spark-1.6.0-bin-hadoop2.6/ec2$ export AWS_SECRET_ACCESS_KEY=W49PsEJvbhNh0KexSU/iPiVWDFLBKtVsTLsqbp1
ubuntu@ip-172-31-0-170:~/spark-1.6.0-bin-hadoop2.6/ec2$ ./spark-ec2 -k CloudSorting -i /home/ubuntu/CloudSorting.pem -s1 -t c3.large -r us-east-1 --spot-price=0.025 --ebs-vol-size=30 launch $sparkSortCluster
Setting up security groups...
Creating security group SparkSortCluster-master
Creating security group SparkSortCluster-slaves
Searching for existing cluster SparkSortCluster in region us-east-1...
Spark AMI: ami-5bb18832
Launching instances...
Requesting 1 slaves as spot instances with price $0.025
```

Logging into Master Node:

```
Connecting to 52.87.152.216:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+]'.

Last login: Mon Mar 28 08:36:18 2016 from ip-172-31-35-135.ec2.internal

[ ] [ ] )
[ ] ( [ ] / Amazon Linux AMI
[ ] \ [ ] [ ]

https://aws.amazon.com/amazon-linux-ami/2013.03-release-notes/
Amazon Linux version 2016.03 is available.
root@ip-172-31-35-135 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/xvda1       7.9G  3.4G  4.5G  43% /
tmpfs           1.9G    0  1.9G   0% /dev/shm
/dev/xvdb       16G   1.3G   14G   9% /mnt
/dev/xvdf       16G  167M   15G   2% /mnt2
/dev/xvds       30G   33M   30G   1% /vol0
root@ip-172-31-35-135 ~]$
```

Once logged onto the master node, we just need to execute our code for sorting of 1GB/10GB/100GB dataset. The master will take care of all the communication between slaves for us.

Gensort to generate 1GB file, put the same file onto HDFS and launch the spark-shell:

```
root@ip-172-31-35-135 ~]$ cd ephemeral-hdfs/
root@ip-172-31-35-135 ephemeral-hdfs]$ ls
bin      CHANGES.txt  docs          hadoop-core-1.0.4.jar    hadoop-test-1.0.4.jar  ivy.xml  LICENSE.txt  sbin  webapps
build.xml  conf       .hadoop-ant-1.0.4.jar  .hadoop-examples-1.0.4.jar  .hadoop-tools-1.0.4.jar  lib     NOTICE.txt  share
c++      contrib      .hadoop-client-1.0.4.jar  .hadoop-minicluster-1.0.4.jar  ivy      libexec  README.txt  src
root@ip-172-31-35-135 ephemeral-hdfs]$ cd ..
root@ip-172-31-35-135 ~]$ cd 64
root@ip-172-31-35-135 64]$ time ./gensort -a 10000000 Input.txt

real   0m44.298s
user   0m1.988s
sys    0m0.128s
root@ip-172-31-35-135 64]$ cd ..
root@ip-172-31-35-135 ~]$ cd ephemeral-hdfs/b
bin/    build.xml
root@ip-172-31-35-135 ~]$ cd ephemeral-hdfs/bin/
root@ip-172-31-35-135 bin]$ ./hadoop fs -put /root/64/Input.txt /Input
Warning: $HADOOP_HOME is deprecated.

root@ip-172-31-35-135 bin]$ cd ..
root@ip-172-31-35-135 ephemeral-hdfs]$ cd spark/
root@ip-172-31-35-135 ~]$ cd spark/
bin/    conf/    ec2/    lib/    licenses/  NOTICE    R/    RELEASE
CHANGES.txt  data/  examples/  LICENSE  logs/    python/  README.md  sbin/
root@ip-172-31-35-135 ~]$ cd spark/bin/
root@ip-172-31-35-135 bin]$ ./spark-shell
16/03/28 08:47:41 INFO spark.SecurityManager: Changing view acls to: root
16/03/28 08:47:41 INFO spark.SecurityManager: Changing modify acls to: root
16/03/28 08:47:41 INFO spark.SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Set(root); users with modify permissions: Set(root)
16/03/28 08:47:41 INFO spark.HttpServer: Starting HTTP Server
16/03/28 08:47:41 INFO server.Server: jetty-8.y.z-SNAPSHOT
```

Submitting the job to master via spark-shell (1GB) Stage-1:

```
16/03/28 08:56:16 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/03/28 08:56:16 INFO scheduler.DAGScheduler: Missing parents: List()
16/03/28 08:56:16 INFO scheduler.DAGScheduler: Submitting ResultStage 0 (MapPartitionsRDD[5] at sortByKey at <console>:29), which has no missing parents
16/03/28 08:56:16 INFO storage.MemoryStore: Block broadcast_1 stored as values in memory (estimated size 3.9 KB, free 36.0 KB)
16/03/28 08:56:16 INFO storage.MemoryStore: Block broadcast_1_piece0 stored as bytes in memory (estimated size 2.1 KB, free 38.1 KB)
16/03/28 08:56:16 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on 172.31.35.135:42538 (size: 2.1 KB, free: 511.5 MB)
16/03/28 08:56:16 INFO spark.SparkContext: Created broadcast 1 from broadcast at DAGScheduler.scala:1006
16/03/28 08:56:16 INFO scheduler.DAGScheduler: Submitting 8 missing tasks from ResultStage 0 (MapPartitionsRDD[5] at sortByKey at <console>:29)
16/03/28 08:56:16 INFO scheduler.TaskSchedulerImpl: Adding task set 0.0 with 8 tasks
16/03/28 08:56:16 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 0.0 (TID 0, ip-172-31-43-21.ec2.internal, partition 0, NODE_LOCAL, 2162 bytes)
16/03/28 08:56:16 INFO scheduler.TaskSetManager: Starting task 1.0 in stage 0.0 (TID 1, ip-172-31-43-21.ec2.internal, partition 1, NODE_LOCAL, 2162 bytes)
16/03/28 08:56:17 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on ip-172-31-43-21.ec2.internal:50510 (size: 2.1 KB, free: 1539.0 MB)
16/03/28 08:56:17 INFO storage.BlockManagerInfo: Added broadcast_0_piece0 in memory on ip-172-31-43-21.ec2.internal:50510 (size: 5.6 KB, free: 1539.0 MB)
16/03/28 08:56:20 INFO scheduler.TaskSetManager: Starting task 2.0 in stage 0.0 (TID 2, ip-172-31-43-21.ec2.internal, partition 2, NODE_LOCAL, 2162 bytes)
16/03/28 08:56:20 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 0.0 (TID 0) in 3876 ms on ip-172-31-43-21.ec2.internal (1/8)
16/03/28 08:56:20 INFO scheduler.TaskSetManager: Starting task 3.0 in stage 0.0 (TID 3, ip-172-31-43-21.ec2.internal, partition 3, NODE_LOCAL, 2162 bytes)
16/03/28 08:56:20 INFO scheduler.TaskSetManager: Finished task 1.0 in stage 0.0 (TID 1) in 4113 ms on ip-172-31-43-21.ec2.internal (2/8)
16/03/28 08:56:22 INFO scheduler.TaskSetManager: Starting task 4.0 in stage 0.0 (TID 4, ip-172-31-43-21.ec2.internal, partition 4, NODE_LOCAL, 2162 bytes)
16/03/28 08:56:22 INFO scheduler.TaskSetManager: Finished task 3.0 in stage 0.0 (TID 3) in 2000 ms on ip-172-31-43-21.ec2.internal (3/8)
16/03/28 08:56:22 INFO scheduler.TaskSetManager: Starting task 5.0 in stage 0.0 (TID 5, ip-172-31-43-21.ec2.internal, partition 5, NODE_LOCAL, 2162 bytes)
16/03/28 08:56:22 INFO scheduler.TaskSetManager: Finished task 2.0 in stage 0.0 (TID 2) in 2299 ms on ip-172-31-43-21.ec2.internal (4/8)
16/03/28 08:56:24 INFO scheduler.TaskSetManager: Starting task 6.0 in stage 0.0 (TID 6, ip-172-31-43-21.ec2.internal, partition 6, NODE_LOCAL, 2162 bytes)
16/03/28 08:56:24 INFO scheduler.TaskSetManager: Finished task 4.0 in stage 0.0 (TID 4) in 1803 ms on ip-172-31-43-21.ec2.internal (5/8)
16/03/28 08:56:24 INFO scheduler.TaskSetManager: Starting task 7.0 in stage 0.0 (TID 7, ip-172-31-43-21.ec2.internal, partition 7, NODE_LOCAL, 2162 bytes)
16/03/28 08:56:24 INFO scheduler.TaskSetManager: Finished task 5.0 in stage 0.0 (TID 5) in 1788 ms on ip-172-31-43-21.ec2.internal (6/8)
16/03/28 08:56:25 INFO scheduler.TaskSetManager: Finished task 7.0 in stage 0.0 (TID 7) in 764 ms on ip-172-31-43-21.ec2.internal (7/8)
16/03/28 08:56:25 INFO scheduler.TaskSetManager: Finished task 6.0 in stage 0.0 (TID 6) in 1259 ms on ip-172-31-43-21.ec2.internal (8/8)
16/03/28 08:56:25 INFO scheduler.DAGScheduler: ResultStage 0 (sortByKey at <console>:29) finished in 9.179 s
16/03/28 08:56:25 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 0.0, whose tasks have all completed, from pool
startSort: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[7] at map at <console>:29
scala> startSort.saveAsTextFile()
```

Submitting the job to stage 2 to get the final output file in HDFS (1GB):

```
16/03/28 08:59:39 INFO scheduler.DAGScheduler: waiting: Set(ResultStage 2)
16/03/28 08:59:39 INFO scheduler.DAGScheduler: failed: Set()
16/03/28 08:59:39 INFO storage.MemoryStore: Block broadcast_3 stored as values in memory (estimated size 15.5 KB, free 60.5 KB)
16/03/28 08:59:39 INFO storage.MemoryStore: Block broadcast_3_piece0 stored as bytes in memory (estimated size 8.1 KB, free 68.6 KB)
16/03/28 08:59:39 INFO storage.BlockManagerInfo: Added broadcast_3_piece0 in memory on 172.31.35.135:4258 (size: 8.1 KB, free: 511.5 MB)
16/03/28 08:59:39 INFO spark.SparkContext: Created broadcast 3 from broadcast at DAGScheduler.scala:1006
16/03/28 08:59:39 INFO scheduler.DAGScheduler: Submitting 8 missing tasks from ResultStage 2 (MapPartitionsRDD[8] at saveAsTextFile at <console>:32)
16/03/28 08:59:39 INFO scheduler.TaskSchedulerImpl: Adding task set 2.0 with 8 tasks
16/03/28 08:59:39 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 2.0 (TID 16, ip-172-31-43-21.ec2.internal, partition 0.NODE_LOCAL, 1894 bytes)
16/03/28 08:59:39 INFO scheduler.TaskSetManager: Starting task 1.0 in stage 2.0 (TID 17, ip-172-31-43-21.ec2.internal, partition 1.NODE_LOCAL, 1894 bytes)
16/03/28 08:59:39 INFO storage.BlockManagerInfo: Added broadcast_3_piece0 in memory on ip-172-31-43-21.ec2.internal:50510 (size: 8.1 KB, free: 1539.0 MB)
16/03/28 08:59:39 INFO spark.MapOutputTrackerMasterEndpoint: Asked to send map output locations for shuffle 0 to ip-172-31-43-21.ec2.internal:55737
16/03/28 08:59:39 INFO spark.MapOutputTrackerMaster: Size of output statuses for shuffle 0 is 185 bytes
16/03/28 08:59:49 INFO scheduler.TaskSetManager: Starting task 2.0 in stage 2.0 (TID 18, ip-172-31-43-21.ec2.internal, partition 2.NODE_LOCAL, 1894 bytes)
16/03/28 08:59:49 INFO scheduler.TaskSetManager: Finished task 1.0 in stage 2.0 (TID 17) in 10077 ms on ip-172-31-43-21.ec2.internal (1/8)
16/03/28 08:59:51 INFO scheduler.TaskSetManager: Starting task 3.0 in stage 2.0 (TID 19, ip-172-31-43-21.ec2.internal, partition 3.NODE_LOCAL, 1894 bytes)
16/03/28 08:59:51 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 2.0 (TID 16) in 12106 ms on ip-172-31-43-21.ec2.internal (2/8)
16/03/28 09:00:02 INFO scheduler.TaskSetManager: Starting task 4.0 in stage 2.0 (TID 20, ip-172-31-43-21.ec2.internal, partition 4.NODE_LOCAL, 1894 bytes)
16/03/28 09:00:02 INFO scheduler.TaskSetManager: Finished task 2.0 in stage 2.0 (TID 18) in 12877 ms on ip-172-31-43-21.ec2.internal (3/8)
16/03/28 09:00:04 INFO scheduler.TaskSetManager: Starting task 5.0 in stage 2.0 (TID 21, ip-172-31-43-21.ec2.internal, partition 5.NODE_LOCAL, 1894 bytes)
16/03/28 09:00:04 INFO scheduler.TaskSetManager: Finished task 3.0 in stage 2.0 (TID 19) in 12798 ms on ip-172-31-43-21.ec2.internal (4/8)
16/03/28 09:00:12 INFO scheduler.TaskSetManager: Starting task 6.0 in stage 2.0 (TID 22, ip-172-31-43-21.ec2.internal, partition 6.NODE_LOCAL, 1894 bytes)
16/03/28 09:00:12 INFO scheduler.TaskSetManager: Finished task 4.0 in stage 2.0 (TID 20) in 10108 ms on ip-172-31-43-21.ec2.internal (5/8)
16/03/28 09:00:15 INFO scheduler.TaskSetManager: Starting task 7.0 in stage 2.0 (TID 23, ip-172-31-43-21.ec2.internal, partition 7.NODE_LOCAL, 1894 bytes)
16/03/28 09:00:15 INFO scheduler.TaskSetManager: Finished task 5.0 in stage 2.0 (TID 21) in 10962 ms on ip-172-31-43-21.ec2.internal (6/8)
16/03/28 09:00:23 INFO scheduler.TaskSetManager: Finished task 6.0 in stage 2.0 (TID 22) in 10995 ms on ip-172-31-43-21.ec2.internal (7/8)
16/03/28 09:00:25 INFO scheduler.TaskSetManager: Finished task 7.0 in stage 2.0 (TID 23) in 9916 ms on ip-172-31-43-21.ec2.internal (8/8)
16/03/28 09:00:25 INFO scheduler.DAGScheduler: ResultStage 2 (saveAsTextFile at <console>:32) finished in 45.780 s
16/03/28 09:00:25 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 2.0, whose tasks have all completed, from pool
16/03/28 09:00:25 INFO scheduler.DAGScheduler: Job 1 finished: saveAsTextFile at <console>:32, took 68.095390 s

scala> [green square]
```

Sorted Output into 8 Chunks displaying first 10 and last 10 values:

```
root@ip-172-31-35-135 64]$ cd Output/
root@ip-172-31-35-135 Output]$ ls
part-00000 part-00001 part-00002 part-00003 part-00004 part-00005 part-00006 part-00007 _SUCCESS
root@ip-172-31-35-135 Output]$ vi part-0000
root@ip-172-31-35-135 Output]$ head -10 part-0000
    "0!uve 0000000000000000000000000000001228D4 7777888800002224444DDDDDDDEEE00000000CCCC7777DDDD
, K4a-:v 00000000000000000000000000000001B8132 5555EEEE888899994444FFF1111CCCEE1111EEEE6666FFFF
.FuD\}u 0000000000000000000000000000000797631 5555DDDBBBB000077772211112224444DDDDDD99996666
;5Yhct 00000000000000000000000000000000000007D3DF5 2222AAAACCCCFFFAAAA44445555EEE44442222DDDD99992222
=2G^9{- 00000000000000000000000000000000809EE 5555DDD1111CCCC9999BBBB0000BBBBCCCCFFFFCCCC44443333
N}M9?sp 000000000000000000000000000000429597 5555FFF000077755559991111CCCC66669999AAAAEEE8888
POX?Rs& 00000000000000000000000000000041162E 888833339999FFFF1111CCCC8888CCCC9999EEEEDDD00003333
[Xq\\$% 0000000000000000000000000000000097A5F0 66666666EEEEDDD7777FFF00005555FFFFFF88885551111
rAmQg4v 000000000000000000000000000000008180D3 BBBB111111119999FFFFFFF4444BBBB88884444CCCC
!&)Jf3; 0000000000000000000000000000004602E1 4444FFFCCCC88888888CCCCFFFFCCCCEEE5555666666666666
root@ip-172-31-35-135 Output]$ tail -10 part-00007
~~~%A NB_t 0000000000000000000000000000003E5EC FFFF7777EEEEBBBB4444EEEEEEE33339999DDDD999900005555
~~~c-CQ(> 000000000000000000000000000000832611 9999FFF111177773333777700001111444444440000BBBB6666
~~~8Y}Fql* 0000000000000000000000000000000742A4C BBBB1111CCCCEEE888800000000777733333333DDDD22225555
~~~>Dd=QT) 000000000000000000000000000000674C0F 9999DDDD000055556666CCCC22220000FFFEEEEDDDDFFFF0000
~~~]JA{}j$ 0000000000000000000000000000006BCBE 111122223333444455559999AAAABBBB9999FFFFDDDBBBB3333
~~~]Zp.#/+ 0000000000000000000000000000003B9A5A CCCC8888EEEEAAAEEE333333377770000FFFFCCCC66667777
~~~_jQepix 00000000000000000000000000000011E5D4 1111999911115555BBBB1111000222EEE6666BBBB7777DDDD
~~~nt=ZH[N 000000000000000000000000000032A13 44441111BBBBBBBB33337777FFF44445555553330000CCCC
~~~s/Pq,-E 00000000000000000000000000006BE930 2222DDDDDD77771111EEECCCC7777BBBB4444888811111111
~~~zba_Tt 0000000000000000000000000000007F9F4F BBBBCCCC666655559999FFFF8888AAAA11116666AAAABBBB0000
root@ip-172-31-35-135 Output]$ [green square]
```

For 10GB Sort:

Give Input file to Slave:

```
16/03/28 09:36:24 INFO session.SessionState: Created HDFS directory: /tmp/hive/root/b02b8ed3-d19d-40ca-b020-de5299ac7151
16/03/28 09:36:24 INFO session.SessionState: Created local directory: /tmp/root/b02b8ed3-d19d-40ca-b020-de5299ac7151
16/03/28 09:36:24 INFO session.SessionState: Created HDFS directory: /tmp/hive/root/b02b8ed3-d19d-40ca-b020-de5299ac7151/_tmp_space.db
16/03/28 09:36:24 INFO repl.SparkLoop: Created sql context (with Hive support)..  
SQL context available as sqlContext.  
  
scala> val inputFile = sc.textFile("/Input10GB")
16/03/28 09:37:09 INFO storage.MemoryStore: Block broadcast_0 stored as values in memory (estimated size 26.5 KB, free 26.5 KB)
16/03/28 09:37:09 INFO storage.MemoryStore: Block broadcast_0_piece0 stored as bytes in memory (estimated size 5.6 KB, free 32.1 KB)
16/03/28 09:37:09 INFO storage.BlockManagerInfo: Added broadcast_0_piece0 in memory on 172.31.35.58635 (size: 5.6 KB, free: 511.5 MB)
16/03/28 09:37:09 INFO spark.SparkContext: Created broadcast 0 from textFile at <console>:27
inputFile: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[1] at textFile at <console>:27  
  
scala> val startSort = inputFile.flatMap(line => line.split("\n")).map(line => (line.substring(0,10), line.substring(10,98))).sortByKey().map{case(k, v) => k + v + " "}
16/03/28 09:37:18 WARN snappy.LoadSnappy: Snappy native library not loaded
16/03/28 09:37:18 INFO mapred.FileInputFormat: Total input paths to process : 1
16/03/28 09:37:18 INFO spark.SparkContext: Starting job: sortByKey at <console>:29
16/03/28 09:37:18 INFO scheduler.DAGScheduler: Got job 0 (sortByKey at <console>:29) with 75 output partitions
16/03/28 09:37:18 INFO scheduler.DAGScheduler: Final stage: ResultStage 0 (sortByKey at <console>:29)
16/03/28 09:37:18 INFO scheduler.DAGScheduler: Parents of final stage: List()
16/03/28 09:37:18 INFO scheduler.DAGScheduler: Missing parents: List()
16/03/28 09:37:18 INFO scheduler.DAGScheduler: Submitting ResultStage 0 (MapPartitionsRDD[5] at sortByKey at <console>:29), which has no missing parents
16/03/28 09:37:18 INFO storage.MemoryStore: Block broadcast_1 stored as values in memory (estimated size 3.9 KB, free 36.0 KB)
16/03/28 09:37:18 INFO storage.MemoryStore: Block broadcast_1_piece0 stored as bytes in memory (estimated size 2.1 KB, free 38.1 KB)
16/03/28 09:37:18 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on 172.31.35.58635 (size: 2.1 KB, free: 511.5 MB)
16/03/28 09:37:18 INFO spark.SparkContext: Created broadcast 1 from broadcast at DAGScheduler.scala:1006
16/03/28 09:37:18 INFO scheduler.DAGScheduler: Submitting 75 missing tasks from ResultStage 0 (MapPartitionsRDD[5] at sortByKey at <console>:29)
16/03/28 09:37:18 INFO scheduler.TaskSchedulerImpl: Adding task set 0.0 with 75 tasks
16/03/28 09:37:18 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 0.0 (TID 0, ip-172-31-43-21.ec2.internal, partition 0,NODE_LOCAL, 2166 bytes)
16/03/28 09:37:18 INFO scheduler.TaskSetManager: Starting task 1.0 in stage 0.0 (TID 1, ip-172-31-43-21.ec2.internal, partition 1,NODE_LOCAL, 2166 bytes)
16/03/28 09:37:18 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on ip-172-31-43-21.ec2.internal:48750 (size: 2.1 KB, free: 1539.0 MB)
16/03/28 09:37:19 INFO storage.BlockManagerInfo: Added broadcast_0_piece0 in memory on ip-172-31-43-21.ec2.internal:48750 (size: 5.6 KB, free: 1539.0 MB)
```

Map Phase:

```
16/03/28 09:37:46 INFO scheduler.TaskSetManager: Starting task 32.0 in stage 0.0 (TID 32, ip-172-31-43-21.ec2.internal, partition 32,NODE_LOCAL, 2166 bytes)
16/03/28 09:37:46 INFO scheduler.TaskSetManager: Finished task 30.0 in stage 0.0 (TID 30) in 1606 ms on ip-172-31-43-21.ec2.internal (31/75)
16/03/28 09:37:46 INFO scheduler.TaskSetManager: Starting task 33.0 in stage 0.0 (TID 33, ip-172-31-43-21.ec2.internal, partition 33,NODE_LOCAL, 2166 bytes)
16/03/28 09:37:46 INFO scheduler.TaskSetManager: Finished task 31.0 in stage 0.0 (TID 31) in 1597 ms on ip-172-31-43-21.ec2.internal (32/75)
16/03/28 09:37:48 INFO scheduler.TaskSetManager: Starting task 34.0 in stage 0.0 (TID 34, ip-172-31-43-21.ec2.internal, partition 34,NODE_LOCAL, 2166 bytes)
16/03/28 09:37:48 INFO scheduler.TaskSetManager: Finished task 32.0 in stage 0.0 (TID 32) in 1577 ms on ip-172-31-43-21.ec2.internal (33/75)
16/03/28 09:37:48 INFO scheduler.TaskSetManager: Starting task 35.0 in stage 0.0 (TID 35, ip-172-31-43-21.ec2.internal, partition 35,NODE_LOCAL, 2166 bytes)
16/03/28 09:37:48 INFO scheduler.TaskSetManager: Finished task 33.0 in stage 0.0 (TID 33) in 1601 ms on ip-172-31-43-21.ec2.internal (34/75)
16/03/28 09:37:49 INFO scheduler.TaskSetManager: Starting task 36.0 in stage 0.0 (TID 36, ip-172-31-43-21.ec2.internal, partition 36,NODE_LOCAL, 2166 bytes)
16/03/28 09:37:49 INFO scheduler.TaskSetManager: Finished task 34.0 in stage 0.0 (TID 34) in 1571 ms on ip-172-31-43-21.ec2.internal (35/75)
16/03/28 09:37:49 INFO scheduler.TaskSetManager: Starting task 37.0 in stage 0.0 (TID 37, ip-172-31-43-21.ec2.internal, partition 37,NODE_LOCAL, 2166 bytes)
16/03/28 09:37:49 INFO scheduler.TaskSetManager: Finished task 35.0 in stage 0.0 (TID 35) in 1612 ms on ip-172-31-43-21.ec2.internal (36/75)
16/03/28 09:37:51 INFO scheduler.TaskSetManager: Starting task 38.0 in stage 0.0 (TID 38, ip-172-31-43-21.ec2.internal, partition 38,NODE_LOCAL, 2166 bytes)
16/03/28 09:37:51 INFO scheduler.TaskSetManager: Finished task 36.0 in stage 0.0 (TID 36) in 1678 ms on ip-172-31-43-21.ec2.internal (37/75)
16/03/28 09:37:51 INFO scheduler.TaskSetManager: Starting task 39.0 in stage 0.0 (TID 39, ip-172-31-43-21.ec2.internal, partition 39,NODE_LOCAL, 2166 bytes)
16/03/28 09:37:51 INFO scheduler.TaskSetManager: Finished task 37.0 in stage 0.0 (TID 37) in 1610 ms on ip-172-31-43-21.ec2.internal (38/75)
16/03/28 09:37:53 INFO scheduler.TaskSetManager: Starting task 40.0 in stage 0.0 (TID 40, ip-172-31-43-21.ec2.internal, partition 40,NODE_LOCAL, 2166 bytes)
16/03/28 09:37:53 INFO scheduler.TaskSetManager: Finished task 38.0 in stage 0.0 (TID 38) in 1624 ms on ip-172-31-43-21.ec2.internal (39/75)
16/03/28 09:37:53 INFO scheduler.TaskSetManager: Starting task 41.0 in stage 0.0 (TID 41, ip-172-31-43-21.ec2.internal, partition 41,NODE_LOCAL, 2166 bytes)
16/03/28 09:37:53 INFO scheduler.TaskSetManager: Finished task 39.0 in stage 0.0 (TID 39) in 1585 ms on ip-172-31-43-21.ec2.internal (40/75)
16/03/28 09:37:54 INFO scheduler.TaskSetManager: Starting task 42.0 in stage 0.0 (TID 42, ip-172-31-43-21.ec2.internal, partition 42,NODE_LOCAL, 2166 bytes)
16/03/28 09:37:54 INFO scheduler.TaskSetManager: Finished task 41.0 in stage 0.0 (TID 41) in 1567 ms on ip-172-31-43-21.ec2.internal (41/75)
16/03/28 09:37:54 INFO scheduler.TaskSetManager: Starting task 43.0 in stage 0.0 (TID 43, ip-172-31-43-21.ec2.internal, partition 43,NODE_LOCAL, 2166 bytes)
16/03/28 09:37:54 INFO scheduler.TaskSetManager: Finished task 40.0 in stage 0.0 (TID 40) in 1651 ms on ip-172-31-43-21.ec2.internal (42/75)
16/03/28 09:37:56 INFO scheduler.TaskSetManager: Starting task 44.0 in stage 0.0 (TID 44, ip-172-31-43-21.ec2.internal, partition 44,NODE_LOCAL, 2166 bytes)
16/03/28 09:37:56 INFO scheduler.TaskSetManager: Finished task 42.0 in stage 0.0 (TID 42) in 1540 ms on ip-172-31-43-21.ec2.internal (43/75)
16/03/28 09:37:56 INFO scheduler.TaskSetManager: Starting task 45.0 in stage 0.0 (TID 45, ip-172-31-43-21.ec2.internal, partition 45,NODE_LOCAL, 2166 bytes)
16/03/28 09:37:56 INFO scheduler.TaskSetManager: Finished task 43.0 in stage 0.0 (TID 43) in 1676 ms on ip-172-31-43-21.ec2.internal (44/75)
16/03/28 09:37:57 INFO scheduler.TaskSetManager: Starting task 46.0 in stage 0.0 (TID 46, ip-172-31-43-21.ec2.internal, partition 46,NODE_LOCAL, 2166 bytes)
16/03/28 09:37:57 INFO scheduler.TaskSetManager: Finished task 44.0 in stage 0.0 (TID 44) in 1538 ms on ip-172-31-43-21.ec2.internal (45/75)
16/03/28 09:37:58 INFO scheduler.TaskSetManager: Starting task 47.0 in stage 0.0 (TID 47, ip-172-31-43-21.ec2.internal, partition 47,NODE_LOCAL, 2166 bytes)
16/03/28 09:37:58 INFO scheduler.TaskSetManager: Finished task 45.0 in stage 0.0 (TID 45) in 1630 ms on ip-172-31-43-21.ec2.internal (46/75)
```

Map Phase Complete:

```
16/03/28 09:38:10 INFO scheduler.TaskSetManager: Finished task 60.0 in stage 0.0 (TID 60) in 1569 ms on ip-172-31-43-21.ec2.internal (61/75)
16/03/28 09:38:10 INFO scheduler.TaskSetManager: Starting task 63.0 in stage 0.0 (TID 63, ip-172-31-43-21.ec2.internal, partition 63,NODE_LOCAL, 2166 bytes)
16/03/28 09:38:10 INFO scheduler.TaskSetManager: Finished task 61.0 in stage 0.0 (TID 61) in 1588 ms on ip-172-31-43-21.ec2.internal (62/75)
16/03/28 09:38:11 INFO scheduler.TaskSetManager: Starting task 64.0 in stage 0.0 (TID 64, ip-172-31-43-21.ec2.internal, partition 64,NODE_LOCAL, 2166 bytes)
16/03/28 09:38:11 INFO scheduler.TaskSetManager: Finished task 62.0 in stage 0.0 (TID 62) in 1649 ms on ip-172-31-43-21.ec2.internal (63/75)
16/03/28 09:38:12 INFO scheduler.TaskSetManager: Starting task 65.0 in stage 0.0 (TID 65, ip-172-31-43-21.ec2.internal, partition 65,NODE_LOCAL, 2166 bytes)
16/03/28 09:38:12 INFO scheduler.TaskSetManager: Finished task 63.0 in stage 0.0 (TID 63) in 1572 ms on ip-172-31-43-21.ec2.internal (64/75)
16/03/28 09:38:13 INFO scheduler.TaskSetManager: Starting task 66.0 in stage 0.0 (TID 66, ip-172-31-43-21.ec2.internal, partition 66,NODE_LOCAL, 2166 bytes)
16/03/28 09:38:13 INFO scheduler.TaskSetManager: Finished task 64.0 in stage 0.0 (TID 64) in 1635 ms on ip-172-31-43-21.ec2.internal (65/75)
16/03/28 09:38:14 INFO scheduler.TaskSetManager: Starting task 67.0 in stage 0.0 (TID 67, ip-172-31-43-21.ec2.internal, partition 67,NODE_LOCAL, 2166 bytes)
16/03/28 09:38:14 INFO scheduler.TaskSetManager: Finished task 65.0 in stage 0.0 (TID 65) in 1650 ms on ip-172-31-43-21.ec2.internal (66/75)
16/03/28 09:38:15 INFO scheduler.TaskSetManager: Starting task 68.0 in stage 0.0 (TID 68, ip-172-31-43-21.ec2.internal, partition 68,NODE_LOCAL, 2166 bytes)
16/03/28 09:38:15 INFO scheduler.TaskSetManager: Finished task 66.0 in stage 0.0 (TID 66) in 1558 ms on ip-172-31-43-21.ec2.internal (67/75)
16/03/28 09:38:15 INFO scheduler.TaskSetManager: Starting task 69.0 in stage 0.0 (TID 69, ip-172-31-43-21.ec2.internal, partition 69,NODE_LOCAL, 2166 bytes)
16/03/28 09:38:15 INFO scheduler.TaskSetManager: Finished task 67.0 in stage 0.0 (TID 67) in 1641 ms on ip-172-31-43-21.ec2.internal (68/75)
16/03/28 09:38:16 INFO scheduler.TaskSetManager: Starting task 70.0 in stage 0.0 (TID 70, ip-172-31-43-21.ec2.internal, partition 70,NODE_LOCAL, 2166 bytes)
16/03/28 09:38:16 INFO scheduler.TaskSetManager: Finished task 68.0 in stage 0.0 (TID 68) in 1590 ms on ip-172-31-43-21.ec2.internal (69/75)
16/03/28 09:38:17 INFO scheduler.TaskSetManager: Starting task 71.0 in stage 0.0 (TID 71, ip-172-31-43-21.ec2.internal, partition 71,NODE_LOCAL, 2166 bytes)
16/03/28 09:38:17 INFO scheduler.TaskSetManager: Finished task 69.0 in stage 0.0 (TID 69) in 1613 ms on ip-172-31-43-21.ec2.internal (70/75)
16/03/28 09:38:18 INFO scheduler.TaskSetManager: Starting task 72.0 in stage 0.0 (TID 72, ip-172-31-43-21.ec2.internal, partition 72,NODE_LOCAL, 2166 bytes)
16/03/28 09:38:18 INFO scheduler.TaskSetManager: Finished task 70.0 in stage 0.0 (TID 70) in 1585 ms on ip-172-31-43-21.ec2.internal (71/75)
16/03/28 09:38:18 INFO scheduler.TaskSetManager: Starting task 73.0 in stage 0.0 (TID 73, ip-172-31-43-21.ec2.internal, partition 73,NODE_LOCAL, 2166 bytes)
16/03/28 09:38:18 INFO scheduler.TaskSetManager: Finished task 71.0 in stage 0.0 (TID 71) in 1612 ms on ip-172-31-43-21.ec2.internal (72/75)
16/03/28 09:38:19 INFO scheduler.TaskSetManager: Starting task 74.0 in stage 0.0 (TID 74, ip-172-31-43-21.ec2.internal, partition 74,NODE_LOCAL, 2166 bytes)
16/03/28 09:38:19 INFO scheduler.TaskSetManager: Finished task 72.0 in stage 0.0 (TID 72) in 1602 ms on ip-172-31-43-21.ec2.internal (73/75)
16/03/28 09:38:20 INFO scheduler.TaskSetManager: Finished task 73.0 in stage 0.0 (TID 73) in 1605 ms on ip-172-31-43-21.ec2.internal (74/75)
16/03/28 09:38:20 INFO scheduler.TaskSetManager: Finished task 74.0 in stage 0.0 (TID 74) in 764 ms on ip-172-31-43-21.ec2.internal (75/75)
16/03/28 09:38:20 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 0.0, whose tasks have all completed, from pool
16/03/28 09:38:20 INFO scheduler.DAGScheduler: ResultStage 0 (sortByKey at <console>:29) finished in 62.194 s
16/03/28 09:38:20 INFO scheduler.DAGScheduler: Job 0 finished: sortByKey at <console>:29, took 62.266086 s
startSort: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[7] at map at <console>:29
scala> 
```

Reduce Phase and Saving into Output File the Chunks:

```
scala> startSort.saveAsTextFile("/Output10GB")
16/03/28 09:39:45 INFO spark.SparkContext: Starting job: saveAsTextFile at <console>:32
16/03/28 09:39:45 INFO scheduler.DAGScheduler: Registering RDD 3 (map at <console>:29)
16/03/28 09:39:45 INFO scheduler.DAGScheduler: Got job 1 (saveAsTextFile at <console>:32) with 75 output partitions
16/03/28 09:39:45 INFO scheduler.DAGScheduler: Final stage: ResultStage 2 (saveAsTextFile at <console>:32)
16/03/28 09:39:45 INFO scheduler.DAGScheduler: Parents of final stage: List(ShuffleMapStage 1)
16/03/28 09:39:45 INFO scheduler.DAGScheduler: Missing parents: List(ShuffleMapStage 1)
16/03/28 09:39:45 INFO scheduler.DAGScheduler: Submitting ShuffleMapStage 1 (MapPartitionsRDD[3] at map at <console>:29), which has no missing parents
16/03/28 09:39:45 INFO storage.MemoryStore: Block broadcast_2 stored as values in memory (estimated size 5.4 KB, free 43.4 KB)
16/03/28 09:39:45 INFO storage.MemoryStore: Block broadcast_2_piece0 stored as bytes in memory (estimated size 3.4 KB, free 46.8 KB)
16/03/28 09:39:45 INFO storage.BlockManagerInfo: Added broadcast_2_piece0 in memory on 172.31.35.155:58635 (size: 3.4 KB, free: 511.5 MB)
16/03/28 09:39:45 INFO spark.SparkContext: Created broadcast 2 from broadcast at DAGScheduler.scala:1006
16/03/28 09:39:45 INFO scheduler.DAGScheduler: Submitting 75 missing tasks from ShuffleMapStage 1 (MapPartitionsRDD[3] at map at <console>:29)
16/03/28 09:39:45 INFO scheduler.TaskSchedulerImpl: Adding task set 1.0 with 75 tasks
16/03/28 09:39:45 INFO scheduler.TaskSetManager: Starting task 0.0 in stage 1.0 (TID 75, ip-172-31-43-21.ec2.internal, partition 0,NODE_LOCAL, 2155 bytes)
16/03/28 09:39:45 INFO scheduler.TaskSetManager: Starting task 1.0 in stage 1.0 (TID 76, ip-172-31-43-21.ec2.internal, partition 1,NODE_LOCAL, 2155 bytes)
16/03/28 09:39:45 INFO storage.BlockManagerInfo: Added broadcast_2_piece0 in memory on ip-172-31-43-21.ec2.internal:48750 (size: 3.4 KB, free: 1539.0 MB)
```

```

16/03/28 09:50:46 INFO scheduler.TaskSetManager: Starting task 62.0 in stage 2.0 (TID 212, ip-172-31-43-21.ec2.internal, partition 62, NODE_LOCAL, 1894 bytes)
16/03/28 09:50:46 INFO scheduler.TaskSetManager: Finished task 60.0 in stage 2.0 (TID 210) in 16640 ms on ip-172-31-43-21.ec2.internal (61/75)
16/03/28 09:50:49 INFO scheduler.TaskSetManager: Starting task 63.0 in stage 2.0 (TID 213, ip-172-31-43-21.ec2.internal, partition 63, NODE_LOCAL, 1894 bytes)
16/03/28 09:50:49 INFO scheduler.TaskSetManager: Finished task 61.0 in stage 2.0 (TID 211) in 16180 ms on ip-172-31-43-21.ec2.internal (62/75)
16/03/28 09:51:00 INFO scheduler.TaskSetManager: Starting task 64.0 in stage 2.0 (TID 214, ip-172-31-43-21.ec2.internal, partition 64, NODE_LOCAL, 1894 bytes)
16/03/28 09:51:00 INFO scheduler.TaskSetManager: Finished task 62.0 in stage 2.0 (TID 212) in 14203 ms on ip-172-31-43-21.ec2.internal (63/75)
16/03/28 09:51:04 INFO scheduler.TaskSetManager: Starting task 65.0 in stage 2.0 (TID 215, ip-172-31-43-21.ec2.internal, partition 65, NODE_LOCAL, 1894 bytes)
16/03/28 09:51:04 INFO scheduler.TaskSetManager: Finished task 63.0 in stage 2.0 (TID 213) in 15539 ms on ip-172-31-43-21.ec2.internal (64/75)
16/03/28 09:51:16 INFO scheduler.TaskSetManager: Starting task 66.0 in stage 2.0 (TID 216, ip-172-31-43-21.ec2.internal, partition 66, NODE_LOCAL, 1894 bytes)
16/03/28 09:51:16 INFO scheduler.TaskSetManager: Finished task 64.0 in stage 2.0 (TID 214) in 15185 ms on ip-172-31-43-21.ec2.internal (65/75)
16/03/28 09:51:18 INFO scheduler.TaskSetManager: Starting task 67.0 in stage 2.0 (TID 217, ip-172-31-43-21.ec2.internal, partition 67, NODE_LOCAL, 1894 bytes)
16/03/28 09:51:18 INFO scheduler.TaskSetManager: Finished task 65.0 in stage 2.0 (TID 215) in 13913 ms on ip-172-31-43-21.ec2.internal (66/75)
16/03/28 09:51:30 INFO scheduler.TaskSetManager: Starting task 68.0 in stage 2.0 (TID 218, ip-172-31-43-21.ec2.internal, partition 68, NODE_LOCAL, 1894 bytes)
16/03/28 09:51:30 INFO scheduler.TaskSetManager: Finished task 66.0 in stage 2.0 (TID 216) in 14794 ms on ip-172-31-43-21.ec2.internal (67/75)
16/03/28 09:51:34 INFO scheduler.TaskSetManager: Starting task 69.0 in stage 2.0 (TID 219, ip-172-31-43-21.ec2.internal, partition 69, NODE_LOCAL, 1894 bytes)
16/03/28 09:51:34 INFO scheduler.TaskSetManager: Finished task 67.0 in stage 2.0 (TID 217) in 16066 ms on ip-172-31-43-21.ec2.internal (68/75)
16/03/28 09:51:44 INFO scheduler.TaskSetManager: Starting task 70.0 in stage 2.0 (TID 220, ip-172-31-43-21.ec2.internal, partition 70, NODE_LOCAL, 1894 bytes)
16/03/28 09:51:44 INFO scheduler.TaskSetManager: Finished task 68.0 in stage 2.0 (TID 218) in 13949 ms on ip-172-31-43-21.ec2.internal (69/75)
16/03/28 09:51:51 INFO scheduler.TaskSetManager: Starting task 71.0 in stage 2.0 (TID 221, ip-172-31-43-21.ec2.internal, partition 71, NODE_LOCAL, 1894 bytes)
16/03/28 09:51:51 INFO scheduler.TaskSetManager: Finished task 69.0 in stage 2.0 (TID 219) in 16213 ms on ip-172-31-43-21.ec2.internal (70/75)
16/03/28 09:51:54 INFO scheduler.TaskSetManager: Starting task 72.0 in stage 2.0 (TID 222, ip-172-31-43-21.ec2.internal, partition 72, NODE_LOCAL, 1894 bytes)
16/03/28 09:51:54 INFO scheduler.TaskSetManager: Finished task 70.0 in stage 2.0 (TID 220) in 9732 ms on ip-172-31-43-21.ec2.internal (71/75)
16/03/28 09:52:02 INFO scheduler.TaskSetManager: Starting task 73.0 in stage 2.0 (TID 223, ip-172-31-43-21.ec2.internal, partition 73, NODE_LOCAL, 1894 bytes)
16/03/28 09:52:02 INFO scheduler.TaskSetManager: Finished task 71.0 in stage 2.0 (TID 221) in 11590 ms on ip-172-31-43-21.ec2.internal (72/75)
16/03/28 09:52:06 INFO scheduler.TaskSetManager: Starting task 74.0 in stage 2.0 (TID 224, ip-172-31-43-21.ec2.internal, partition 74, NODE_LOCAL, 1894 bytes)
16/03/28 09:52:06 INFO scheduler.TaskSetManager: Finished task 72.0 in stage 2.0 (TID 222) in 11669 ms on ip-172-31-43-21.ec2.internal (73/75)
16/03/28 09:52:18 INFO scheduler.TaskSetManager: Finished task 73.0 in stage 2.0 (TID 223) in 16036 ms on ip-172-31-43-21.ec2.internal (74/75)
16/03/28 09:52:21 INFO scheduler.TaskSetManager: Finished task 74.0 in stage 2.0 (TID 224) in 15224 ms on ip-172-31-43-21.ec2.internal (75/75)
16/03/28 09:52:21 INFO scheduler.DAGScheduler: ResultStage 2 (saveAsTextFile at <console>:32) finished in 505.233 s
16/03/28 09:52:21 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 2.0, whose tasks have all completed, from pool
16/03/28 09:52:21 INFO scheduler.DAGScheduler: Job 1 finished: saveAsTextFile at <console>:32, took 755.692807 s

```

Get Sorted Chunks (75 Chunks):

```

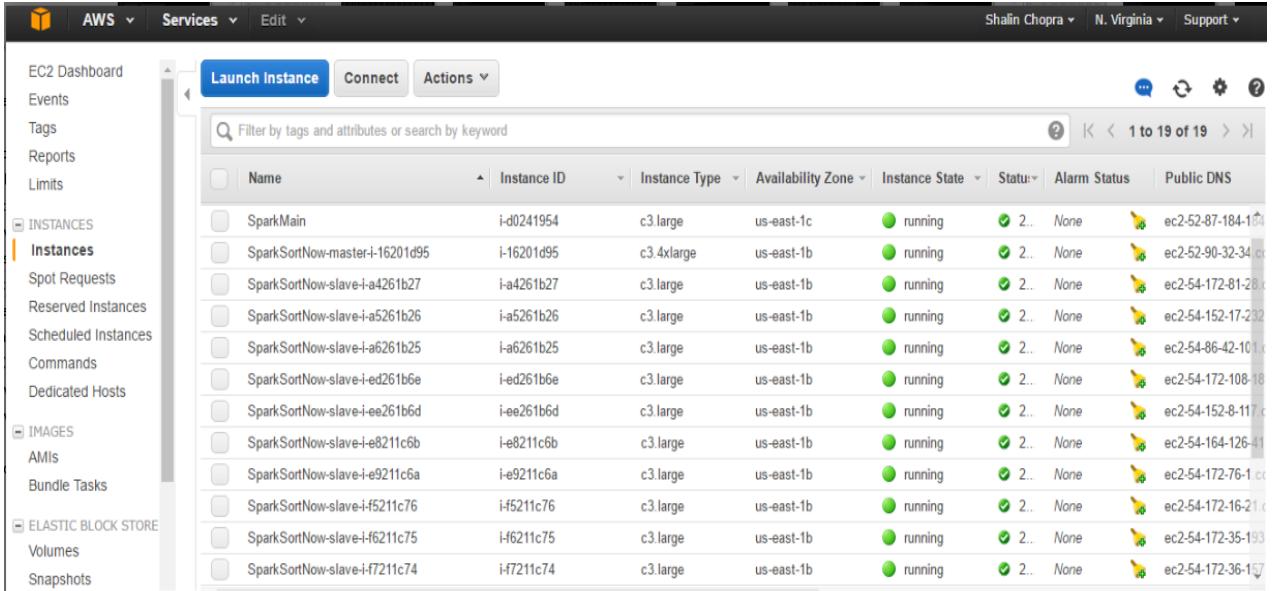
root@ip-172-31-35-135 ~$: cd ephemeral-hdfs/bin/
root@ip-172-31-35-135 bin]$ time ./hadoop fs -get /Output10GB /vol0/Output10GB
Warning: $HADOOP_HOME is deprecated.

real   6m31.940s
user   0m22.829s
sys    0m6.428s
root@ip-172-31-35-135 bin]$ cd ..
root@ip-172-31-35-135 ephemeral-hdfs]$ cd ..
root@ip-172-31-35-135 ~$: cd /vol0/Output10GB/
root@ip-172-31-35-135 Output10GB]$ ls
part-00000 part-00005 part-00010 part-00020 part-00025 part-00030 part-00035 part-00040 part-00045 part-00050 part-00055 part-00060 part-00065 part-00070 _SUCCESS
part-00001 part-00006 part-00011 part-00016 part-00021 part-00026 part-00031 part-00036 part-00041 part-00046 part-00051 part-00056 part-00061 part-00066 part-00071
part-00002 part-00007 part-00012 part-00017 part-00022 part-00027 part-00032 part-00037 part-00042 part-00047 part-00052 part-00057 part-00062 part-00067 part-00072
part-00003 part-00008 part-00013 part-00018 part-00023 part-00028 part-00033 part-00038 part-00043 part-00048 part-00053 part-00058 part-00063 part-00068 part-00073
part-00004 part-00009 part-00014 part-00019 part-00024 part-00029 part-00034 part-00039 part-00044 part-00049 part-00054 part-00059 part-00064 part-00069 part-00074
root@ip-172-31-35-135 Output10GB]$ head -10 part-00000
    "0!uve 000000000000000000000000000000001228D4 77778888000022224444DDDDDDDEEEE00000000CCCC7777DDDD
PMd32= 000000000000000000000000000000003440C1 FFFFEEEE6666CCCCBBB9999333555DDDDDD777788886666
^3C0], 00000000000000000000000000000000158C55 5555AAAA9999EEE88882229999CCCCDDDD666655554442222
!&S3[], 000000000000000000000000000000002145D78 8888BBBBDDDD1111CCCC55556666BBBB1111EEEDDD22229999
!,=U#,9 000000000000000000000000000000001907E3 33332222FFFB BBB0000FFFFAAAAA666655553333DDDD3333CCCC
!Of[ITd 000000000000000000000000000000003CAAB48 9999FFFF5555333777CCCC4444BBBB7777EEEBBBBDDDD4444
!f6Suy2 000000000000000000000000000000003ABFD84 EEEE5555555666AAA5555BBBBDDDD0000111166660000DDDD
#%NIPq. 000000000000000000000000000000003B36FB9 11110000333344411116666666AAAAAAA0001111CCCCEEEE
#`cl~ 000000000000000000000000000000002EDC5C8 8888AAAA11114444FFF77773333EEE44440000FFFF99999999
$"-Q)] 000000000000000000000000000000005F1265 DCCC6666EEE22220000DDDAAA88886666BBBB00006666AAAA
root@ip-172-31-35-135 Output10GB]$ tail -10 part-00074
~~~uq2k#=U 00000000000000000000000000000002C06745 9999111DDDD22221110000FFFFEEEEEFFF33337777CCCC2222
~~~v/QNnm 000000000000000000000000000000004709701 CCCC8888333FFFF000000000009991111FFF77774446666
~~~yk0l:qE 000000000000000000000000000000002048B4F CCCC1111444488882226666BBBB888855557777EEEBBBB0000
~~~yK~H.il 00000000000000000000000000000000463D004 44440000FFF3333999944447777DDDDFFFFAAA11118888DDDD
~~~yL;C'XE 000000000000000000000000000000005B0D211 2222EEEE3333000022221111CCCCFF55557774444BBBB6666
~~~zbA_Tt 000000000000000000000000000000007F94F BBBBBCCC666655559999FFF8888AAAA11116666AAABBBB0000
~~~ze0^FEg 00000000000000000000000000000001E06130 4444CCCCBBB9999222888555888CCCFFFF00001111111
~~~}GxjWHi 00000000000000000000000000000000CA1345 777711118888AAAAAAA2222111BBBB00002222BBBBCCCC2222
~~~}P;]g0g 000000000000000000000000000000040DA3E4 4444FFF44446666333EEE88888888DDDEEEE44442222DDDD
~~~}kU|K<p 00000000000000000000000000000005E4A0AA 0000666655551111BBBB88889999AAAA55550000333355557777
root@ip-172-31-35-135 Output10GB]$ 

```

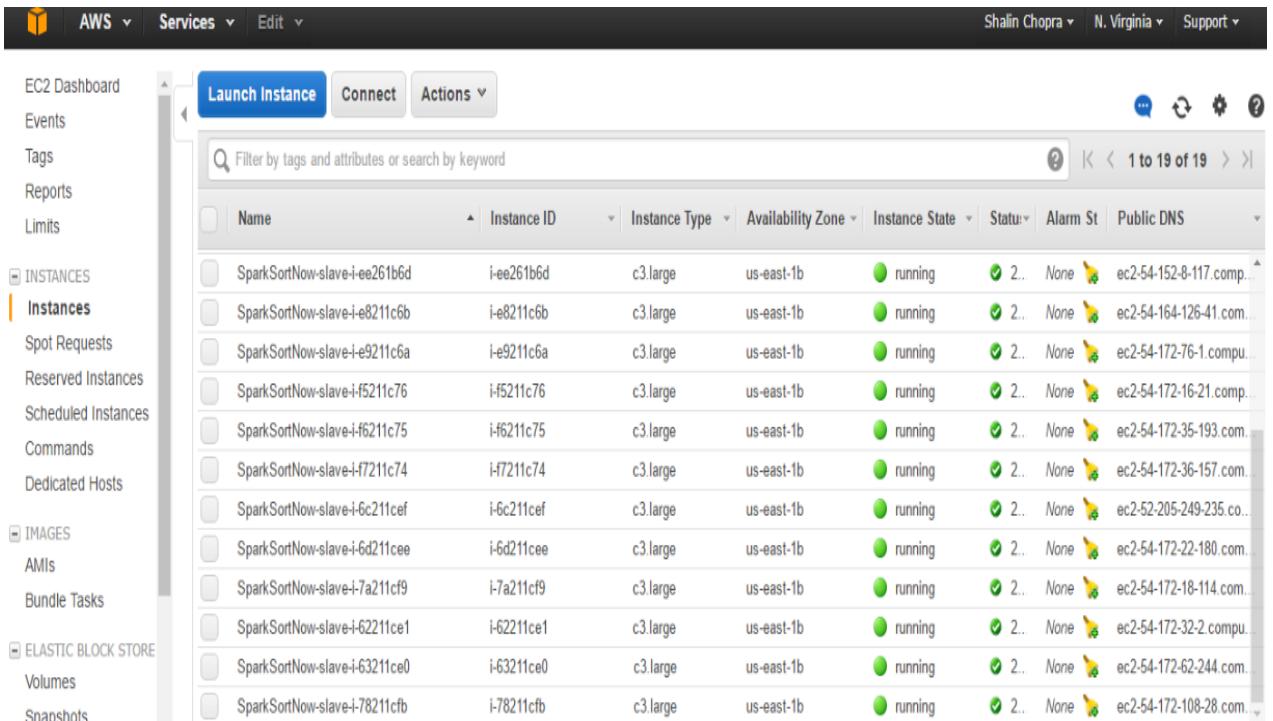
Sort 100GB:

Instances Masters and Slaves Connected:



The screenshot shows the AWS EC2 Instances page with 19 instances listed. The 'Instances' section is selected in the sidebar. The table headers are Name, Instance ID, Instance Type, Availability Zone, Instance State, Status, Alarm Status, and Public DNS. The instances are all running and are labeled as c3.large. The Public DNS column shows various domain names starting with ec2-54 or ec2-52.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status	Alarm Status	Public DNS
SparkMain	i-d0241954	c3.large	us-east-1c	running	✓ 2..	None	ec2-52-87-184-1
SparkSortNow-master-i-16201d95	i-16201d95	c3.4xlarge	us-east-1b	running	✓ 2..	None	ec2-52-90-32-34
SparkSortNow-slave-i-a4261b27	i-a4261b27	c3.large	us-east-1b	running	✓ 2..	None	ec2-54-172-81-23
SparkSortNow-slave-i-a5261b26	i-a5261b26	c3.large	us-east-1b	running	✓ 2..	None	ec2-54-152-17-232
SparkSortNow-slave-i-a6261b25	i-a6261b25	c3.large	us-east-1b	running	✓ 2..	None	ec2-54-86-42-101
SparkSortNow-slave-i-ed261b6e	i-ed261b6e	c3.large	us-east-1b	running	✓ 2..	None	ec2-54-172-108-18
SparkSortNow-slave-i-ee261b6d	i-ee261b6d	c3.large	us-east-1b	running	✓ 2..	None	ec2-54-152-8-111
SparkSortNow-slave-i-e8211c6b	i-e8211c6b	c3.large	us-east-1b	running	✓ 2..	None	ec2-54-164-126-11
SparkSortNow-slave-i-e9211c6a	i-e9211c6a	c3.large	us-east-1b	running	✓ 2..	None	ec2-54-172-76-1
SparkSortNow-slave-i-f5211c76	i-f5211c76	c3.large	us-east-1b	running	✓ 2..	None	ec2-54-172-16-21
SparkSortNow-slave-i-f6211c75	i-f6211c75	c3.large	us-east-1b	running	✓ 2..	None	ec2-54-172-35-13
SparkSortNow-slave-i-f7211c74	i-f7211c74	c3.large	us-east-1b	running	✓ 2..	None	ec2-54-172-36-1



The screenshot shows the AWS EC2 Instances page with 19 instances listed. The 'Instances' section is selected in the sidebar. The table headers are Name, Instance ID, Instance Type, Availability Zone, Instance State, Status, Alarm St, and Public DNS. The instances are all running and are labeled as c3.large. The Public DNS column shows various domain names starting with ec2-54 or ec2-52.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status	Alarm St	Public DNS
SparkSortNow-slave-i-ee261b6d	i-ee261b6d	c3.large	us-east-1b	running	✓ 2..	None	ec2-54-152-8-117.compu..
SparkSortNow-slave-i-e8211c6b	i-e8211c6b	c3.large	us-east-1b	running	✓ 2..	None	ec2-54-164-126-41.com..
SparkSortNow-slave-i-e9211c6a	i-e9211c6a	c3.large	us-east-1b	running	✓ 2..	None	ec2-54-172-76-1.compu..
SparkSortNow-slave-i-f5211c76	i-f5211c76	c3.large	us-east-1b	running	✓ 2..	None	ec2-54-172-16-21.compu..
SparkSortNow-slave-i-f6211c75	i-f6211c75	c3.large	us-east-1b	running	✓ 2..	None	ec2-54-172-35-193.com..
SparkSortNow-slave-i-f7211c74	i-f7211c74	c3.large	us-east-1b	running	✓ 2..	None	ec2-54-172-36-157.com..
SparkSortNow-slave-i-6c211cef	i-6c211cef	c3.large	us-east-1b	running	✓ 2..	None	ec2-52-205-249-235.co..
SparkSortNow-slave-i-6d211cee	i-6d211cee	c3.large	us-east-1b	running	✓ 2..	None	ec2-54-172-22-180.com..
SparkSortNow-slave-i-7a211cf9	i-7a211cf9	c3.large	us-east-1b	running	✓ 2..	None	ec2-54-172-18-114.com..
SparkSortNow-slave-i-62211ce1	i-62211ce1	c3.large	us-east-1b	running	✓ 2..	None	ec2-54-172-32-2.compu..
SparkSortNow-slave-i-63211ce0	i-63211ce0	c3.large	us-east-1b	running	✓ 2..	None	ec2-54-172-62-244.com..
SparkSortNow-slave-i-78211cfb	i-78211cfb	c3.large	us-east-1b	running	✓ 2..	None	ec2-54-172-108-28.com..

The masters and Slaves launched using spark-ec2 script.

Provide the Job to Map and reduce phase:

```
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 711.0 in stage 0.0 (TID 704) in 1771 ms on ip-172-31-55-74.ec2.internal (719/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 723.0 in stage 0.0 (TID 712) in 1641 ms on ip-172-31-61-180.ec2.internal (720/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 722.0 in stage 0.0 (TID 717) in 1565 ms on ip-172-31-48-246.ec2.internal (721/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 717.0 in stage 0.0 (TID 726) in 1432 ms on ip-172-31-48-29.ec2.internal (722/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 735.0 in stage 0.0 (TID 727) in 1434 ms on ip-172-31-55-251.ec2.internal (723/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 716.0 in stage 0.0 (TID 719) in 1522 ms on ip-172-31-63-212.ec2.internal (724/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 706.0 in stage 0.0 (TID 715) in 1645 ms on ip-172-31-48-246.ec2.internal (725/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 709.0 in stage 0.0 (TID 708) in 1754 ms on ip-172-31-62-70.ec2.internal (726/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 727.0 in stage 0.0 (TID 716) in 1648 ms on ip-172-31-60-86.ec2.internal (727/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 738.0 in stage 0.0 (TID 718) in 1610 ms on ip-172-31-60-86.ec2.internal (728/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 712.0 in stage 0.0 (TID 721) in 1575 ms on ip-172-31-55-74.ec2.internal (729/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 731.0 in stage 0.0 (TID 742) in 1023 ms on ip-172-31-51-31.ec2.internal (730/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 740.0 in stage 0.0 (TID 739) in 1495 ms on ip-172-31-55-251.ec2.internal (731/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 742.0 in stage 0.0 (TID 741) in 1088 ms on ip-172-31-51-136.ec2.internal (732/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 718.0 in stage 0.0 (TID 723) in 1572 ms on ip-172-31-56-112.ec2.internal (733/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 728.0 in stage 0.0 (TID 731) in 1517 ms on ip-172-31-48-246.ec2.internal (734/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 733.0 in stage 0.0 (TID 732) in 1387 ms on ip-172-31-62-70.ec2.internal (735/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 732.0 in stage 0.0 (TID 725) in 1578 ms on ip-172-31-62-70.ec2.internal (736/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 736.0 in stage 0.0 (TID 734) in 1441 ms on ip-172-31-63-212.ec2.internal (737/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 730.0 in stage 0.0 (TID 735) in 1503 ms on ip-172-31-61-180.ec2.internal (738/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 729.0 in stage 0.0 (TID 733) in 1663 ms on ip-172-31-56-112.ec2.internal (739/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 743.0 in stage 0.0 (TID 739) in 1442 ms on ip-172-31-56-112.ec2.internal (740/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 734.0 in stage 0.0 (TID 738) in 1609 ms on ip-172-31-56-112.ec2.internal (741/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 719.0 in stage 0.0 (TID 737) in 1686 ms on ip-172-31-55-74.ec2.internal (742/745)
16/03/28 23:34:22 INFO scheduler.TaskSetManager: Finished task 739.0 in stage 0.0 (TID 743) in 1426 ms on ip-172-31-55-74.ec2.internal (743/745)
16/03/28 23:34:22 INFO scheduler.TaskSetManager: Finished task 744.0 in stage 0.0 (TID 744) in 1046 ms on ip-172-31-63-212.ec2.internal (744/745)
16/03/28 23:34:23 INFO scheduler.TaskSetManager: Finished task 741.0 in stage 0.0 (TID 740) in 2849 ms on ip-172-31-51-136.ec2.internal (745/745)
16/03/28 23:34:23 INFO scheduler.DAGScheduler: ResultStage 0 (sortByKey at <console>:29) finished in 24.891 s
16/03/28 23:34:23 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 0.0, whose tasks have all completed, from pool
16/03/28 23:34:23 INFO scheduler.DAGScheduler: Job 0 finished: sortByKey at <console>:29, took 25.072833 s
sortedMap: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[11] at map at <console>:29
```

```
scala> val sortedMap = inputFile.flatMap(line => line.split("\n")).map(line => (line.substring(0,10), line.substring(10,98))).sortByKey().map{case(k,v) => k+ v + " "}
```

```
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 711.0 in stage 0.0 (TID 704) in 1771 ms on ip-172-31-55-74.ec2.internal (719/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 723.0 in stage 0.0 (TID 712) in 1641 ms on ip-172-31-61-180.ec2.internal (720/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 722.0 in stage 0.0 (TID 717) in 1565 ms on ip-172-31-48-246.ec2.internal (721/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 717.0 in stage 0.0 (TID 726) in 1432 ms on ip-172-31-48-29.ec2.internal (722/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 735.0 in stage 0.0 (TID 727) in 1434 ms on ip-172-31-55-251.ec2.internal (723/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 716.0 in stage 0.0 (TID 719) in 1522 ms on ip-172-31-63-212.ec2.internal (724/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 706.0 in stage 0.0 (TID 715) in 1645 ms on ip-172-31-48-246.ec2.internal (725/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 709.0 in stage 0.0 (TID 708) in 1754 ms on ip-172-31-62-70.ec2.internal (726/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 727.0 in stage 0.0 (TID 716) in 1648 ms on ip-172-31-60-86.ec2.internal (727/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 738.0 in stage 0.0 (TID 718) in 1610 ms on ip-172-31-60-86.ec2.internal (728/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 712.0 in stage 0.0 (TID 721) in 1575 ms on ip-172-31-55-74.ec2.internal (729/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 731.0 in stage 0.0 (TID 742) in 1023 ms on ip-172-31-51-31.ec2.internal (730/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 740.0 in stage 0.0 (TID 739) in 1495 ms on ip-172-31-55-251.ec2.internal (731/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 742.0 in stage 0.0 (TID 741) in 1088 ms on ip-172-31-51-136.ec2.internal (732/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 718.0 in stage 0.0 (TID 723) in 1572 ms on ip-172-31-56-112.ec2.internal (733/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 728.0 in stage 0.0 (TID 731) in 1517 ms on ip-172-31-48-246.ec2.internal (734/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 733.0 in stage 0.0 (TID 732) in 1387 ms on ip-172-31-62-70.ec2.internal (735/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 732.0 in stage 0.0 (TID 725) in 1578 ms on ip-172-31-62-70.ec2.internal (736/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 736.0 in stage 0.0 (TID 734) in 1441 ms on ip-172-31-63-212.ec2.internal (737/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 730.0 in stage 0.0 (TID 735) in 1503 ms on ip-172-31-61-180.ec2.internal (738/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 729.0 in stage 0.0 (TID 733) in 1663 ms on ip-172-31-56-112.ec2.internal (739/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 743.0 in stage 0.0 (TID 739) in 1442 ms on ip-172-31-56-112.ec2.internal (740/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 734.0 in stage 0.0 (TID 738) in 1609 ms on ip-172-31-56-112.ec2.internal (741/745)
16/03/28 23:34:21 INFO scheduler.TaskSetManager: Finished task 719.0 in stage 0.0 (TID 737) in 1686 ms on ip-172-31-55-74.ec2.internal (742/745)
16/03/28 23:34:22 INFO scheduler.TaskSetManager: Finished task 739.0 in stage 0.0 (TID 743) in 1426 ms on ip-172-31-55-74.ec2.internal (743/745)
16/03/28 23:34:22 INFO scheduler.TaskSetManager: Finished task 744.0 in stage 0.0 (TID 744) in 1046 ms on ip-172-31-63-212.ec2.internal (744/745)
16/03/28 23:34:23 INFO scheduler.TaskSetManager: Finished task 741.0 in stage 0.0 (TID 740) in 2849 ms on ip-172-31-51-136.ec2.internal (745/745)
16/03/28 23:34:23 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 0.0, whose tasks have all completed, from pool
16/03/28 23:34:23 INFO scheduler.DAGScheduler: Job 0 finished: sortByKey at <console>:29, took 25.072833 s
sortedMap: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[11] at map at <console>:29
```

```
scala> sortedMap.saveAsTextFile("/Output.txt")
```

Last Phase to get the Sorted Output:

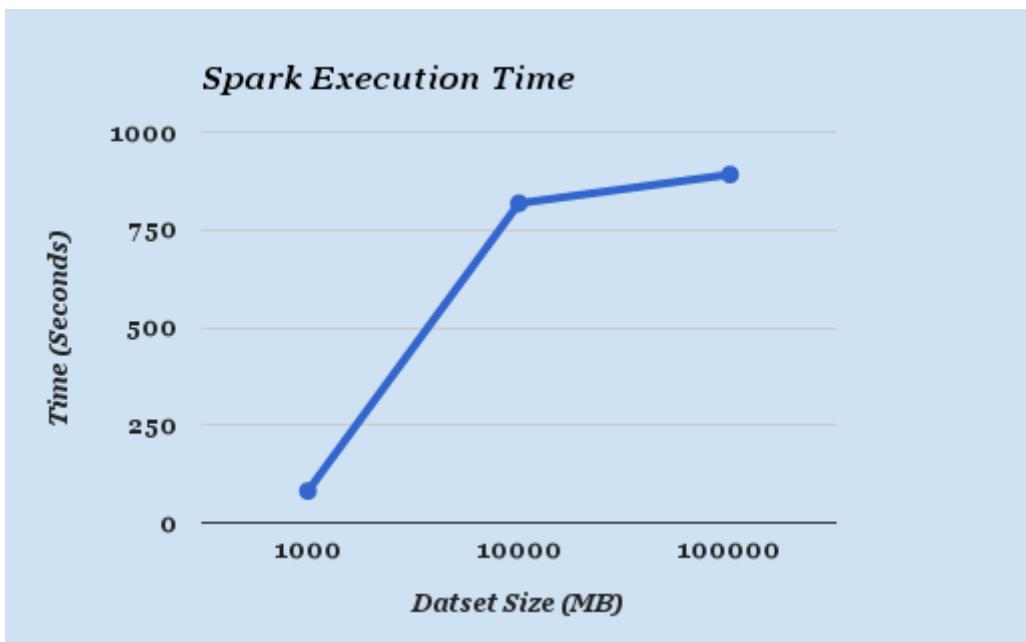
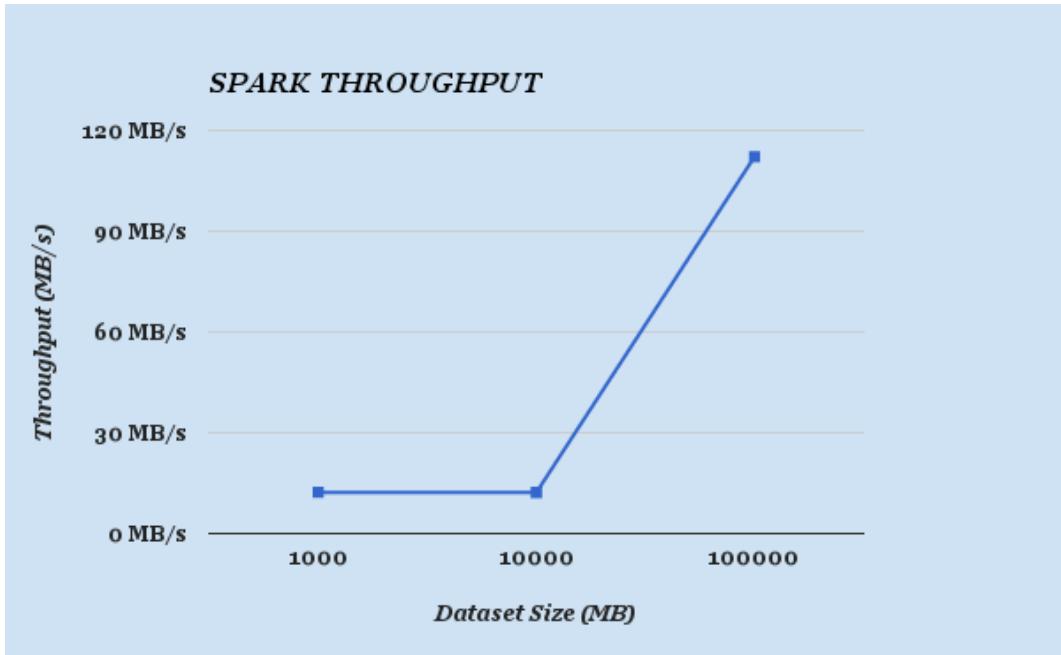
```
16/03/28 23:42:17 INFO scheduler.TaskSetManager: Finished task 716.0 in stage 2.0 (TID 2206) in 25970 ms on ip-172-31-62-91.ec2.internal (718/745)
16/03/28 23:42:18 INFO scheduler.TaskSetManager: Finished task 710.0 in stage 2.0 (TID 2200) in 29060 ms on ip-172-31-63-212.ec2.internal (719/745)
16/03/28 23:42:18 INFO scheduler.TaskSetManager: Finished task 734.0 in stage 2.0 (TID 2224) in 22090 ms on ip-172-31-61-180.ec2.internal (720/745)
16/03/28 23:42:18 INFO scheduler.TaskSetManager: Finished task 724.0 in stage 2.0 (TID 2214) in 24998 ms on ip-172-31-55-251.ec2.internal (721/745)
16/03/28 23:42:18 INFO scheduler.TaskSetManager: Finished task 715.0 in stage 2.0 (TID 2205) in 27859 ms on ip-172-31-51-136.ec2.internal (722/745)
16/03/28 23:42:18 INFO scheduler.TaskSetManager: Finished task 717.0 in stage 2.0 (TID 2207) in 26416 ms on ip-172-31-56-112.ec2.internal (723/745)
16/03/28 23:42:18 INFO scheduler.TaskSetManager: Finished task 735.0 in stage 2.0 (TID 2225) in 21801 ms on ip-172-31-56-212.ec2.internal (724/745)
16/03/28 23:42:18 INFO scheduler.TaskSetManager: Finished task 720.0 in stage 2.0 (TID 2210) in 25690 ms on ip-172-31-48-29.ec2.internal (725/745)
16/03/28 23:42:18 INFO scheduler.TaskSetManager: Finished task 727.0 in stage 2.0 (TID 2217) in 24200 ms on ip-172-31-55-251.ec2.internal (726/745)
16/03/28 23:42:19 INFO scheduler.TaskSetManager: Finished task 707.0 in stage 2.0 (TID 2197) in 31585 ms on ip-172-31-51-136.ec2.internal (727/745)
16/03/28 23:42:19 INFO scheduler.TaskSetManager: Finished task 740.0 in stage 2.0 (TID 2230) in 21326 ms on ip-172-31-48-246.ec2.internal (728/745)
16/03/28 23:42:19 INFO scheduler.TaskSetManager: Finished task 718.0 in stage 2.0 (TID 2208) in 27600 ms on ip-172-31-59-187.ec2.internal (729/745)
16/03/28 23:42:19 INFO scheduler.TaskSetManager: Finished task 719.0 in stage 2.0 (TID 2209) in 27266 ms on ip-172-31-63-212.ec2.internal (730/745)
16/03/28 23:42:20 INFO scheduler.TaskSetManager: Finished task 739.0 in stage 2.0 (TID 2229) in 21740 ms on ip-172-31-62-70.ec2.internal (731/745)
16/03/28 23:42:20 INFO scheduler.TaskSetManager: Finished task 733.0 in stage 2.0 (TID 2223) in 24246 ms on ip-172-31-56-212.ec2.internal (732/745)
16/03/28 23:42:21 INFO scheduler.TaskSetManager: Finished task 722.0 in stage 2.0 (TID 2212) in 28306 ms on ip-172-31-48-29.ec2.internal (733/745)
16/03/28 23:42:21 INFO scheduler.TaskSetManager: Finished task 726.0 in stage 2.0 (TID 2216) in 27698 ms on ip-172-31-62-70.ec2.internal (734/745)
16/03/28 23:42:21 INFO scheduler.TaskSetManager: Finished task 702.0 in stage 2.0 (TID 2192) in 35480 ms on ip-172-31-55-251.ec2.internal (735/745)
16/03/28 23:42:21 INFO scheduler.TaskSetManager: Finished task 700.0 in stage 2.0 (TID 2190) in 37772 ms on ip-172-31-61-180.ec2.internal (736/745)
16/03/28 23:42:22 INFO scheduler.TaskSetManager: Finished task 706.0 in stage 2.0 (TID 2196) in 34216 ms on ip-172-31-61-61.ec2.internal (737/745)
16/03/28 23:42:22 INFO scheduler.TaskSetManager: Finished task 732.0 in stage 2.0 (TID 2222) in 26803 ms on ip-172-31-61-61.ec2.internal (738/745)
16/03/28 23:42:24 INFO scheduler.TaskSetManager: Finished task 731.0 in stage 2.0 (TID 2221) in 28808 ms on ip-172-31-55-251.ec2.internal (739/745)
16/03/28 23:42:24 INFO scheduler.TaskSetManager: Finished task 736.0 in stage 2.0 (TID 2226) in 27409 ms on ip-172-31-51-230.ec2.internal (740/745)
16/03/28 23:42:24 INFO scheduler.TaskSetManager: Finished task 728.0 in stage 2.0 (TID 2218) in 30215 ms on ip-172-31-51-31.ec2.internal (741/745)
16/03/28 23:42:25 INFO scheduler.TaskSetManager: Finished task 729.0 in stage 2.0 (TID 2219) in 30719 ms on ip-172-31-48-29.ec2.internal (742/745)
16/03/28 23:42:26 INFO scheduler.TaskSetManager: Finished task 741.0 in stage 2.0 (TID 2231) in 27685 ms on ip-172-31-51-31.ec2.internal (743/745)
16/03/28 23:42:27 INFO scheduler.TaskSetManager: Finished task 744.0 in stage 2.0 (TID 2234) in 28407 ms on ip-172-31-51-230.ec2.internal (744/745)
16/03/28 23:42:28 INFO scheduler.TaskSetManager: Finished task 737.0 in stage 2.0 (TID 2227) in 29977 ms on ip-172-31-51-230.ec2.internal (745/745)
16/03/28 23:42:28 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 2.0, whose tasks have all completed, from pool
16/03/28 23:42:28 INFO scheduler.DAGScheduler: ResultStage 2 (saveAsTextFile at <console>:32) finished in 312.420 s
16/03/28 23:42:28 INFO scheduler.DAGScheduler: Job 1 finished: saveAsTextFile at <console>:32, took 432.429212 s
```

scala> 

100GB data Chunks (745 Chunks):

```
root@ip-172-31-48-180 bin]$ time ./hadoop fs -get /Output.txt /mnt2/
Warning: $HADOOP_HOME is deprecated.

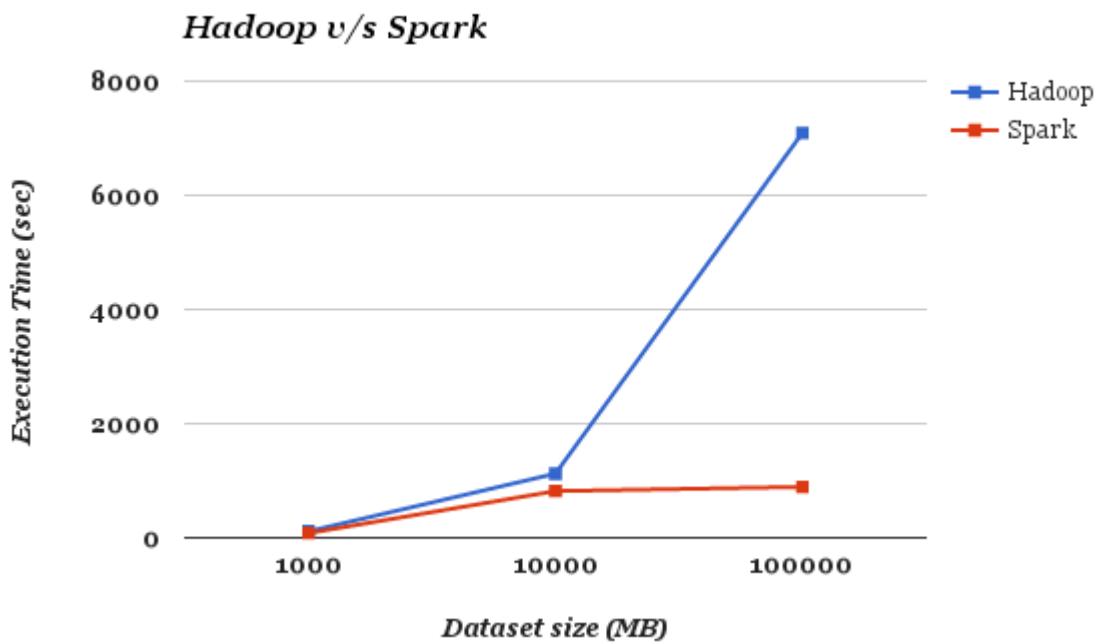
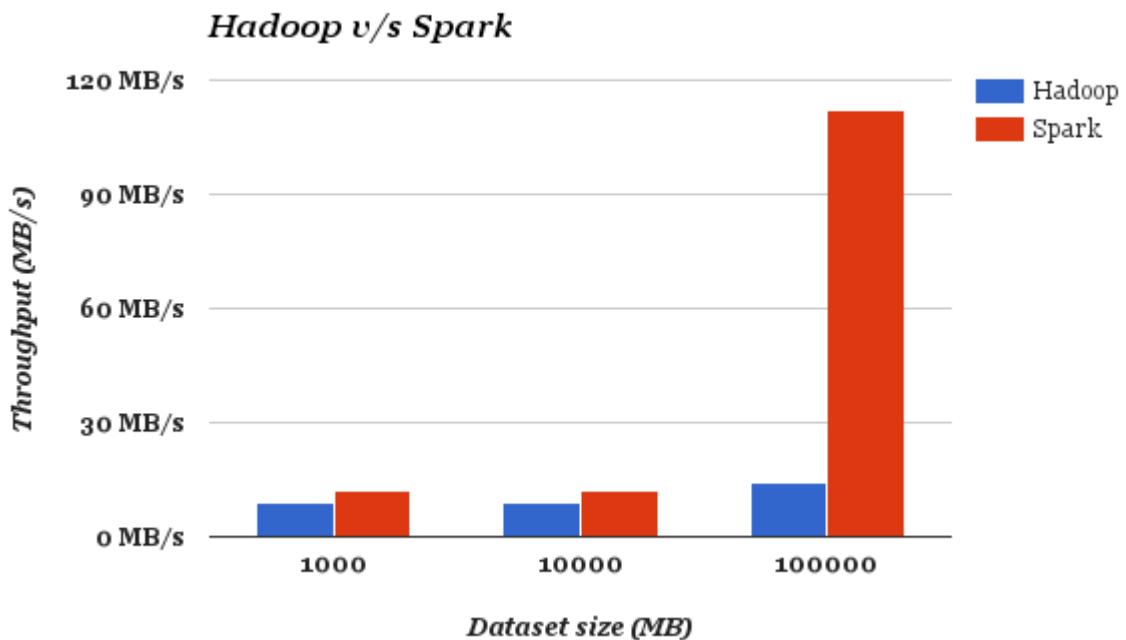
real    23m35.166s
user    2m18.213s
sys     2m50.283s
root@ip-172-31-48-180 bin]$ ls
hadoop  hadoop-daemon.sh  rcc      start-all.sh    start-dfs.sh        start-mapred.sh  stop-balancer.sh  stop-jobhistoryserver.sh  task-controller
hadoop-config.sh  hadoop-daemons.sh  slaves.sh  start-balancer.sh  start-jobhistoryserver.sh  stop-all.sh   stop-dfs.sh   stop-mapred.sh
root@ip-172-31-48-180 bin]$ cd ..
root@ip-172-31-48-180 ephemeral-hdfs]$ cd ..
root@ip-172-31-48-180 ~]$ cd /mnt2/
root@ip-172-31-48-180 mnt2]$ ls
ephemeral-hdfs  hadoop  lost+found  Output.txt  spark
root@ip-172-31-48-180 mnt2]$ cd Output.txt/
root@ip-172-31-48-180 Output.txt]$ ls
part-00000  part-00047  part-00094  part-00141  part-00188  part-00235  part-00282  part-00329  part-00376  part-00423  part-00470  part-00517  part-00564  part-00611  part-00658  part-00705
part-00001  part-00048  part-00095  part-00142  part-00189  part-00236  part-00283  part-00330  part-00377  part-00424  part-00471  part-00518  part-00565  part-00612  part-00659  part-00706
part-00002  part-00049  part-00096  part-00143  part-00190  part-00237  part-00284  part-00331  part-00378  part-00425  part-00472  part-00519  part-00566  part-00613  part-00660  part-00707
part-00003  part-00050  part-00097  part-00144  part-00191  part-00238  part-00285  part-00332  part-00379  part-00426  part-00473  part-00567  part-00614  part-00661  part-00708
part-00004  part-00051  part-00098  part-00145  part-00192  part-00239  part-00286  part-00333  part-00380  part-00427  part-00474  part-00521  part-00568  part-00615  part-00662  part-00709
part-00005  part-00052  part-00099  part-00146  part-00193  part-00240  part-00287  part-00334  part-00381  part-00428  part-00475  part-00522  part-00569  part-00616  part-00663  part-00710
part-00006  part-00053  part-00100  part-00147  part-00194  part-00241  part-00288  part-00335  part-00382  part-00429  part-00476  part-00523  part-00570  part-00617  part-00664  part-00711
part-00007  part-00054  part-00101  part-00148  part-00195  part-00242  part-00289  part-00336  part-00383  part-00430  part-00477  part-00524  part-00571  part-00618  part-00665  part-00712
part-00008  part-00055  part-00102  part-00149  part-00196  part-00243  part-00290  part-00337  part-00384  part-00431  part-00478  part-00525  part-00572  part-00619  part-00666  part-00713
part-00009  part-00056  part-00103  part-00140  part-00197  part-00244  part-00291  part-00338  part-00385  part-00432  part-00479  part-00526  part-00573  part-00620  part-00667  part-00714
part-00010  part-00057  part-00104  part-00151  part-00198  part-00245  part-00292  part-00339  part-00386  part-00433  part-00480  part-00527  part-00574  part-00621  part-00668  part-00715
part-00011  part-00058  part-00105  part-00152  part-00199  part-00246  part-00293  part-00340  part-00387  part-00434  part-00481  part-00528  part-00575  part-00622  part-00669  part-00716
part-00012  part-00059  part-00106  part-00153  part-00200  part-00247  part-00294  part-00341  part-00388  part-00435  part-00482  part-00529  part-00576  part-00623  part-00670  part-00717
part-00013  part-00060  part-00107  part-00154  part-00201  part-00248  part-00295  part-00342  part-00389  part-00436  part-00483  part-00530  part-00577  part-00624  part-00671  part-00718
part-00014  part-00061  part-00108  part-00155  part-00202  part-00249  part-00296  part-00343  part-00390  part-00437  part-00484  part-00531  part-00578  part-00625  part-00672  part-00719
part-00015  part-00062  part-00109  part-00156  part-00203  part-00250  part-00297  part-00344  part-00391  part-00438  part-00485  part-00532  part-00579  part-00626  part-00673  part-00720
part-00016  part-00063  part-00110  part-00157  part-00204  part-00251  part-00298  part-00345  part-00392  part-00439  part-00486  part-00533  part-00580  part-00627  part-00674  part-00721
part-00017  part-00064  part-00111  part-00158  part-00205  part-00252  part-00299  part-00346  part-00393  part-00440  part-00487  part-00534  part-00581  part-00628  part-00675  part-00722
```

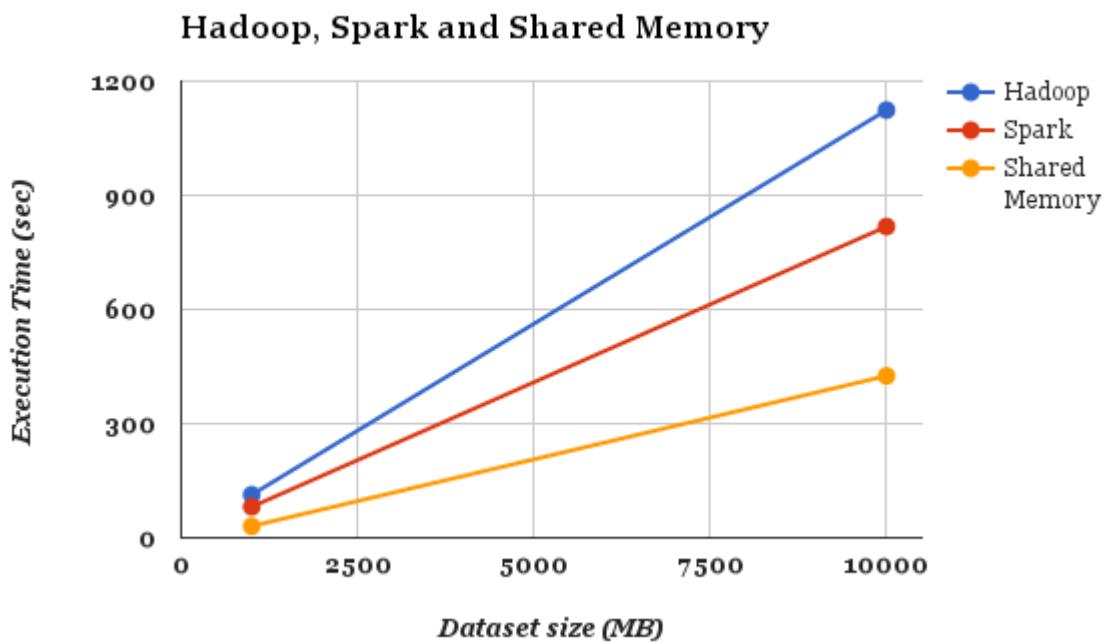
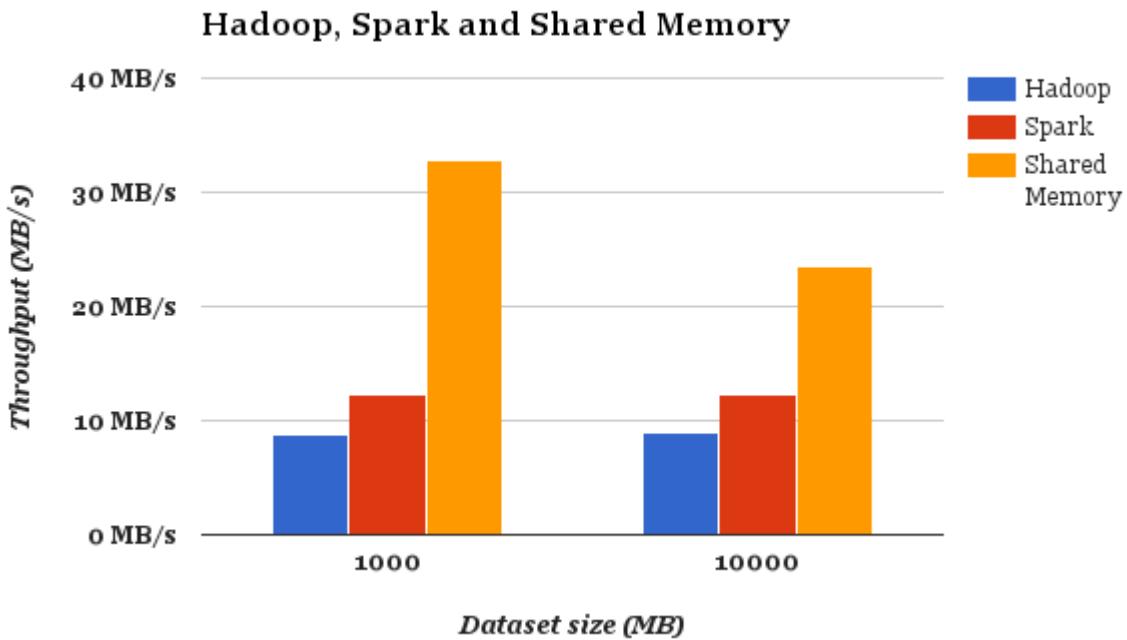
For a better comparison to be made, let's compare between Hadoop and Spark for 3 Different Dataset sizes.

I have compared Hadoop and Spark on the basis of throughput and execution time. From the graphs below we can depict that, for 1GB and 10GB data size, which is performed on 1 Node, Spark holds an edge over hadoop and has slightly more throughput, for 16 Nodes setup, Spark completely outperforms Hadoop. With Hadoop having slight increase in throughput over the dataset sizes, spark has huge improvement for 16 Nodes setup due to storing of data in RDD's makes them access faster and sort data accordingly. Also if Hadoop uses 1 Reducer then it merges the

final output as one big 100GB file, while in case of Spark we don't merge these chunks to form the final sorted 1 big file. This merging may cost Hadoop some time more than spark.



Comparison between Shared Memory, Hadoop and Spark (1GB and 10GB):



To make equal comparisons between 3 different platforms that do sorting, I have taken into account 1 Node sort of each for 1GB and 10GB dataset size. Since Hadoop and Spark takes into account the split times, I have included the split time for Shared memory in execution time. As we can observe Shared Memory Sort performs faster than, both Hadoop and Spark. This is due to the case that there are internally many things happening within Hadoop and Spark environment map-reduce phase, shuffle and Sort phase which are taken into account and increases the time to sort the data.

But when implementing with these platforms over 16 Nodes, Shared memory won't be that efficient as now there needs to be cluster formation, message passing between nodes to communicate, has a lots of network overhead involved, that's where MPI comes into picture.

Questions & Answers:

1. What is master Node? What is a Slave node?

Ans: Master node is the node that looks after the entire Map-Reduce process. This node is responsible for taking the data which is put onto the HDFS and distribute the work evenly to other worker (slave) nodes.

Slave nodes are the worker nodes which executes the work assigned to them by the master node, and put them back onto the HDFS for the master to fetch the data. The /conf/hosts file in Hadoop includes the IP of master node and all the slaves node following it. The other configuration files on all slaves contain the IP of the Master node so as to have an effective communication established between master and slave node.

2. Why do we need to set unique available ports to those configuration files on a shared environment? What errors or side-effects will show if we use same port number for each user?

Ans: The ports which are set in those configuration files are for different functionalities, for each of the ports we have assigned specific usage so that we can track the map-reduce jobs given to the master. We can track the cluster, using the web-app address we check job completion on the webpage and check logs if some errors occur while execution. We can also check whether all the nodes are connected to the master and are performing their tasks or not. If we add the same port-number there might be case of ambiguity between these modules and might not provide us with the information for tracking the job.

3. How can we change the number of mappers and reducers from the configuration file?

Ans: By changing the mapred-site.xml we can add the number of mappers and reducers by adding the following properties within the <configuration> tags

```
<property>
    <name>mapred.map.tasks</name>
    <value>4</value>
</property>

<property>
    <name>mapred.reduce.tasks</name>
    <value>4</value>
</property>
```

Performance Questions and Answers:

What conclusions can you draw?

Ans: Shared memory sort is faster for 1GB and 10GB on single node setup. From the experimental setup we got for this assignment, we can say that Spark performs much better than hadoop, when comparing 16 Nodes cluster scale and large amount of data about 100GB. Spark saves the intermediate data generated, hence one when finished with some experiment in our case sorting, if we need to perform the same experiment again on same data, spark would do it more faster as it has the intermediate data stored onto the system. In case of Hadoop, it may require lesser disk space than spark because it doesn't store the intermediate data, but is much slower.

The graphs for throughput, speedup and execution time discussed earlier depicts the same results.

Which seems to be best at 1 node scale?

Ans: For 1 Node scale, Shared Memory seems to be the best, as we can execute the tasks concurrently using threads and thus perform much faster than Hadoop and Spark. Every thread performs its own task of sorting the Chunks allocated to them, without having dependencies between other threads execution output. External Merge sort then merges the data from all the threads to form a final sorted output.

How about 16 nodes?

Ans: For 16 nodes scale, Spark performs better than Hadoop.

Can you predict which would be best at 100 node scale? How about 1000 node scales?

Ans: Scaling up from 16 nodes to 100 nodes to 1000 node, would make spark and hadoop come closer in terms of throughput but Spark will perform much better at these increase in scaling too. But it also depends on what kind of data is given for job execution, if it is iterative data like Data Mining Application or Machine Learning Application and Spark will definitely be faster than hadoop. So along with scaling we also need to look into kind of data given to the map-reduce jobs.

Compare your results with those from the Sort Benchmark, specifically the winners in 2013 and 2014 who used Hadoop and Spark. Also, what can you learn from the CloudSort benchmark.

Ans:

Specifications	Spark (2014 Winner)	Hadoop (2013 Winner)
Dataset Size	100 TB	102.5 TB
Instances type used	Amazon EC2 i2.8xlarge	Dell R720xd
No. of instances	207 (206 workers + 1 Master)	2100 (Nodes)
System Specs.	32 vCores - 2.5Ghz Intel Xeon E5-2670 v2	2.3Ghz hexcore Xeon E5-2630

Memory	244 GB	64 GB
Disks	8 * 800 GB SSD	12 * 3TB Disks
Throughput	4.27 TB/min	1.42 TB/min
Time	1,406 seconds	4,328 seconds

Comparing with sort implemented by me which is pretty smaller than dataset used above for sort benchmarking, I can see the same trend of Spark performing faster than Hadoop.

CloudSort:

It proposes the Sort Benchmark using the cloud which measures the efficiency of external sort from TCO perspective (total-cost of ownership).

It is a contest in which we need to sort 100 TB of data on any public cloud and sort the data with minimum cost for the duration of sort.

The benchmark is summarized as follows:

1. Sort a fixed number 100TB of randomly permuted 100-byte records with 10-byte keys (same as in our case of sort PA2)
2. The sort must start with the input on a non-ephemeral, persistent store and finish with output on a nonephemeral, persistent store.
3. All operations must be performed on a commercially available public cloud.
4. The winner is the system with the minimum cost prorated for the duration of the sort.

d2.xlarge (Extra):

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links like EC2 Dashboard, Events, Tags, Reports, Limits, Instances, Spot Requests, Reserved Instances, Scheduled Instances, Commands, Dedicated Hosts, AMIs, and Elastic Block Store. The main area has tabs for Launch Instance, Connect, and Actions. A search bar at the top says "Filter by tags and attributes or search by keyword". Below it is a table with columns: Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, and Public DNS. One row is selected, showing "i-fa48757e" for Name, "i-fa48757e" for Instance ID, "d2.xlarge" for Instance Type, "us-east-1c" for Availability Zone, "running" for Instance State, "2/2 check..." for Status Checks, "None" for Alarm Status, and "ec2-54-152-42-168.compute-1.amazonaws.com" for Public DNS. Below the table, there's a detailed view of the instance "i-fa48757e" with tabs for Description, Status Checks, Monitoring, and Tags. The "Description" tab shows various details: Instance ID (i-fa48757e), Instance state (running), Instance type (d2.xlarge), Private DNS (ip-172-31-13-60.ec2.internal), Private IPs (172.31.13.60), Secondary private IPs, VPC ID (vpc-c29304a6), Subnet ID (subnet-ead1c9c), Public DNS (ec2-54-152-42-168.compute-1.amazonaws.com), Public IP (54.152.42.168), Elastic IP (-), Availability zone (us-east-1c), Security groups (launch-wizard-1, view rules), Scheduled events (No scheduled events), AMI ID (MasterImage (ami-85353cef)), and Platform (-). At the bottom of the dashboard, there are links for Feedback, English, and footer information including copyright (© 2008-2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.), Privacy Policy, and Terms of Use.

1 GB using 1 Thread:

```
ubuntu@ip-172-31-13-60:/mnt/raid/64$ sudo sh SharedMemoryScript.sh
Generating Un-Sorted file of 1GB
1.90user 0.22system 0:03.33elapsed 63%CPU (0avgtext+0avgdata 119760maxresident)k
64inputs+1955168outputs (0major+7104minor)pagefaults 0swaps
Creating file Chunks
0.00user 1.04system 0:02.23elapsed 46%CPU (0avgtext+0avgdata 756maxresident)k
1953192inputs+1956016outputs (0major+232minor)pagefaults 0swaps
File Chunks Created generated
STARTING SORT
Running Threads: 1
Sorted Chunks Created, Final Merge Phase Starts
Time Taken in millis: 28307.0
27.24user 2.62system 0:28.37elapsed 105%CPU (0avgtext+0avgdata 772500maxresident)k
368inputs+3912088outputs (0major+16562minor)pagefaults 0swaps
FINISHED SORTING
Checking Sorted File using Valsort
Records: 10000000
Checksum: 4c499aeel4ae2b
Duplicate keys: 0
SUCCESS - all records are in order
1.85user 0.25system 0:10.40elapsed 20%CPU (0avgtext+0avgdata 50100maxresident)k
1953128inputs+0outputs (0major+6438minor)pagefaults 0swaps
Deleting Temporary Chunks
ubuntu@ip-172-31-13-60:/mnt/raid/64$ head -10 MergedOutput.txt
"0"lufe 000000000000000000000000000000001228D4 7777888800002224444DDDDDDDEEE0000000CCC7777DDDD
,K4a-:v 000000000000000000000000000000001B8132 5555EEE88889994444FFF1111CCCEE1111EEE6666FFFF
.FuD\`u 000000000000000000000000000000000797631 5555DDDBBBB000077722211122224444DDDDDD99996666
;YThct 000000000000000000000000000000007D3DF 2222AAAAACCCFFFAAAA44445555EEE44442222DDDD99992222
=2G"9{- 000000000000000000000000000000000809EE 5555DD0D1111CCCC9999BBB80000BBBBCCCCF9999CCC44443333
N}M?RsP 00000000000000000000000000000000429597 5555FFF0000777555599991111CCCC66669999AAAAEEE8888
P0X?Rs& 00000000000000000000000000000000041162E 888833339999FFF1111CCCC8888CCCC9999EEE0000D00003333
[Xq\\%$ 00000000000000000000000000000000097A5F0 66666666EEEEDDD0777FFFF00005555FFFFFF888855551111
rAmQg4v 0000000000000000000000000000000008180D3 BBBB111111119999FFFFFFFFFF88884444BBBB88884444CCCC
|;)if3; 000000000000000000000000000000004602E1 4444FFFCCCC88888888CCCCFFFFCCCCEEE5555666666666666
ubuntu@ip-172-31-13-60:/mnt/raid/64$ tail -10 MergedOutput.txt
~~~%A NB_t 0000000000000000000000000000003DE5EC FFFF7777EEEBBBB4444EEEEEEE33339999DDDD999900005555
~~~c-CQ(- 0000000000000000000000000000000832611 9999FFF111177733377700001111444444440000BBBB6666
~~~8Y)Fql* 00000000000000000000000000000000742A4C BBBB1111CCCCEEE8888000000007773333333DD022225555
~~~>Dd-QT] 0000000000000000000000000000000674C0F 9999DD0D000055556666CCCC2220000FFFEEEEEDDDFFF0000
~~~jJA{}j$ 00000000000000000000000000000060BCBE 11112223333444455559999AAABBBB9999FFFDDDBBBB3333
~~~jZp.#/+ 00000000000000000000000000000003B9A5A CCC8888EEEAAAEEE33333337770000FFFFCCCC66667777
~~~_jQepix 000000000000000000000000000000011E5D4 1111999911115555BBB111100002222EEE6666BBBB7777DDDD
~~~nt=ZH[N 000000000000000000000000000000332A13 4444111BBBBBBB3337777FFF444455555553330000CCCC
~~~s/Pq,-E 0000000000000000000000000000006BE930 2222DDDDDD7771111EEECCCC7777BBBB4444888811111111
~~~zbA_Tt 0000000000000000000000000000007F9F4F BBBBCCCC666655559999FFF8888AAA11116666AAAABBBB0000
ubuntu@ip-172-31-13-60:/mnt/raid/64$
```

1 GB using 2 Threads:

```
ubuntu@ip-172-31-13-60:/mnt/raid/64$ sudo sh SharedMemoryScript.sh
Generating Un-Sorted file of 1GB
1.92user 0.22system 0:03.27elapsed 65%CPU (0avgtext+0avgdata 119760maxresident)k
72inputs+1955160outputs (0major+7108minor)pagefaults 0swaps
Creating file Chunks
0.00user 1.04system 0:02.26elapsed 46%CPU (0avgtext+0avgdata 756maxresident)k
1953192inputs+195592outputs (0major+231minor)pagefaults 0swaps
File Chunks Created generated
STARTING SORT
Running Threads: 2
Sorted Chunks Created, Final Merge Phase Starts
Time Taken in millis: 22663.0
31.22user 2.91system 0:22.73elapsed 150%CPU (0avgtext+0avgdata 853400maxresident)k
136inputs+3910536outputs (0major+15702minor)pagefaults 0swaps
FINISHED SORTING
Checking Sorted File using Valsort
Records: 10000000
Checksum: 4c499aeel14ae2b
Duplicate keys: 0
SUCCESS - all records are in order
1.87user 0.24system 0:13.93elapsed 15%CPU (0avgtext+0avgdata 50104maxresident)k
1953128inputs+0outputs (0major+6438minor)pagefaults 0swaps
Deleting Temporary Chunks
ubuntu@ip-172-31-13-60:/mnt/raid/64$ head -10 MergedOutput.txt
"0|uve 0000000000000000000000000000000000000000001228D4 7777888000022224444DDDDDDDEEE00000000CCC7777DDDD
.K4a-:v 00000000000000000000000000000000001B8132 5555EEE88889994444FFF1111CCCCEE1111EEE6666FFFF
.FuDv):u 000000000000000000000000000000000797631 5555DDDBBBB000077722211112224444DDDDDD99996666
;SYThct 000000000000000000000000000000007D3DF5 2222AAAACCCFFFFA AAAA44445555EEE44442222DDDD99992222
=2G^9{- 00000000000000000000000000000000809EE 5555DDD1111CCCC9999BBB0000BBBBC CCCCCF44443333
N}M9?sp 0000000000000000000000000000000429597 5555FFF000077755559991111CCC66669999AAAEEE8888
POX?Rs& 000000000000000000000000000000041162E 888833339999FFF1111CCCC8888CCC9999EEEEDDD00003333
[!q\\$% 000000000000000000000000000000097AF0 6666666EEEEDDD7777FFF00005555FFFF888855551111
rAmQg4v 000000000000000000000000000000008180D3 8888111111119999FFFFFFFFF4444BBBB88884444CCCC
!(&)j3: 0000000000000000000000000000004602E1 4444FFFCCCC8888888CCCCFFCCCEE5555666666666666
ubuntu@ip-172-31-13-60:/mnt/raid/64$ tail -10 MergedOutput.txt
~~~%A NB_t 000000000000000000000000000000003DE5EC FFFF7777EEE88884444EEE33339999DDDD999900005555
~~~c-CQ(> 000000000000000000000000000000000832611 9999FFF111177733337770000111144444440000BBBB6666
~~~8Y}Fql* 00000000000000000000000000000000742A4C BBBB1111CCCCEE888800000007773333333DD2225555
~~~>Dd=QT 00000000000000000000000000000000674C0F 9999DD000055556666CCC22220000FFFEEEEDDDFFF0000
~~~]JA()j# 0000000000000000000000000000000060BCBE 1111222233344445555999AAAABB89999FFFDDDBBBB3333
~~~lZp_#/+ 00000000000000000000000000000003B9A5A CCC8888EEEAAAEEE33333337770000FFFCCCC66667777
~~~_jQepix 000000000000000000000000000000011E5D4 1111999911115555BBB111100002222EEE6666BBBB7777DD
~~~nt=ZH[N 0000000000000000000000000000000032A13 44441111BBBBBBB3333777FFF444455555533330000CCCC
~~~s/Pq,-E 00000000000000000000000000000006BE930 2222DDDDDD77771111EEECCCC7777BBB4444888811111111
~~~zba_Tt 000000000000000000000000000000007F9F4F BBBBCCCC66665555999FFF8888AAA11116666AAAABB0000
ubuntu@ip-172-31-13-60:/mnt/raid/64$
```

1 GB using 4 Threads:

```
1.92user 0.21system 0:03.32elapsed 64%CPU (0avgtext+0avgdata 119760maxresident)k
72inputs+1955224outputs (0major+7102minor)pagefaults 0swaps
Creating file Chunks
0.00user 1.05system 0:02.72elapsed 38%CPU (0avgtext+0avgdata 760maxresident)k
1953192inputs+1956048outputs (0major+232minor)pagefaults 0swaps
File Chunks Created generated
STARTING SORT
Running Threads: 4
Sorted Chunks Created, Final Merge Phase Starts
Time Taken in millis: 22203.0
43.24user 3.67system 0:22.27elapsed 210%CPU (0avgtext+0avgdata 1203960maxresident)k
136inputs+3913016outputs (0major+20614minor)pagefaults 0swaps
FINISHED SORTING
Checking Sorted File using Valsort
Records: 10000000
Checksum: 4c499aeel14ae2b
Duplicate keys: 0
SUCCESS - all records are in order
1.90user 0.21system 0:11.35elapsed 18%CPU (0avgtext+0avgdata 50104maxresident)k
1953128inputs+0outputs (0major+6438minor)pagefaults 0swaps
Deleting Temporary Chunks
ubuntu@ip-172-31-13-60:/mnt/raid/64$ head -10 MergedOutput.txt
"0|uve 0000000000000000000000000000000000000000001228D4 7777888000022224444DDDDDDDEEE00000000CCC7777DDDD
.K4a-:v 00000000000000000000000000000000001B8132 5555EEE88889994444FFF1111CCCCEE1111EEE6666FFFF
.FuDv):u 000000000000000000000000000000000797631 5555DDDBBBB000077722211112224444DDDDDD99996666
;SYThct 000000000000000000000000000000007D3DF5 2222AAAACCCFFFFA AAAA44445555EEE44442222DDDD99992222
=2G^9{- 00000000000000000000000000000000809EE 5555DDD1111CCCC9999BBB0000BBBBC CCCCCF44443333
N}M9?sp 00000000000000000000000000000000429597 5555FFF000077755559991111CCC66669999AAAEEE8888
POX?Rs& 000000000000000000000000000000041162E 888833339999FFF1111CCCC8888CCC9999EEEEDDD00003333
[!q\\$% 000000000000000000000000000000097AF0 6666666EEEEDDD7777FFF00005555FFFF888855551111
rAmQg4v 000000000000000000000000000000008180D3 8888111111119999FFFFFFFFF4444BBBB88884444CCCC
!(&)j3: 0000000000000000000000000000004602E1 4444FFFCCCC8888888CCCCFFCCCEE5555666666666666
ubuntu@ip-172-31-13-60:/mnt/raid/64$ tail -10 MergedOutput.txt
~~~%A NB_t 000000000000000000000000000000003DE5EC FFFF7777EEE88884444EEE33339999DDDD999900005555
~~~c-CQ(> 000000000000000000000000000000000832611 9999FFF111177733337770000111144444440000BBBB6666
~~~8Y}Fql* 00000000000000000000000000000000742A4C BBBB1111CCCCEE888800000007773333333DD2225555
~~~>Dd=QT 00000000000000000000000000000000674C0F 9999DD000055556666CCC22220000FFFEEEEDDDFFF0000
~~~]JA()j$ 0000000000000000000000000000000060BCBE 1111222233344445555999AAAABB89999FFFDDDBBBB3333
~~~lZp_#/+ 00000000000000000000000000000003B9A5A CCC8888EEEAAAEEE33333337770000FFFCCCC66667777
~~~_jQepix 000000000000000000000000000000011E5D4 1111999911115555BBB111100002222EEE6666BBBB7777DD
~~~nt=ZH[N 0000000000000000000000000000000032A13 44441111BBBBBBB3333777FFF444455555533330000CCCC
~~~s/Pq,-E 00000000000000000000000000000006BE930 2222DDDDDD77771111EEECCCC7777BBB4444888811111111
~~~zba_Tt 000000000000000000000000000000007F9F4F BBBBCCCC66665555999FFF8888AAA11116666AAAABB0000
ubuntu@ip-172-31-13-60:/mnt/raid/64$
```

1 GB using 8 Threads:

```
ubuntu@ip-172-31-13-60:/mnt/raid/64$ sudo sh SharedMemoryScript.sh
Generating Un-Sorted file of 1GB
1.98user 0.19system 0:03.28elapsed 65%CPU (0avgtext+0avgdata 119756maxresident)k
72inputs+1955192outputs (0major+7103minor)pagefaults 0swaps
Creating file Chunks
0.01user 1.03system 0:02.38elapsed 44%CPU (0avgtext+0avgdata 760maxresident)k
1953192inputs+1955968outputs (0major+232minor)pagefaults 0swaps
File Chunks Created generated
STARTING SORT
Running Threads: 8
Sorted Chunks Created, Final Merge Phase Starts
Time Taken in millis: 22968.0
44.95user 4.08system 0:23.14elapsed 211%CPU (0avgtext+0avgdata 1485748maxresident)k
144inputs+3911888outputs (0major+16514minor)pagefaults 0swaps
FINISHED SORTING
Checking Sorted File using Valsort
Records: 10000000
Checksum: 4c499aae14ae2b
Duplicate keys: 0
SUCCESS - all records are in order
1.83user 0.28system 0:14.74elapsed 14%CPU (0avgtext+0avgdata 50104maxresident)k
1953128inputs+0outputs (0major+0437minor)pagefaults 0swaps
Deleting Temporary Chunks
ubuntu@ip-172-31-13-60:/mnt/raid/64$ head -10 MergedOutput.txt
    "0!uve 000000000000000000000000000000001228D4 77778888000022224444DDDDDDDEEEE00000000CCCC7777DDDD
    .K4a-.v 000000000000000000000000000000001B8132 5555EEEE888899994444FFFF1111CCCEEEE1111EEE6666FFFF
    .FuD\}u 00000000000000000000000000000000797631 5555DDDBBBB000077772222111122224444DDDDDD99996666
    ;2G^9{- 000000000000000000000000000000000000000073DF5 2222AAACCCFFFFAAAA44445555EEE44442222DD99992222
    =2G^9{- 00000000000000000000000000000000809EE 5555DDDD1111CCCC9999BBBB0000BBBBCCCCFFFFCCCC44443333
    N)M9?sP 00000000000000000000000000000000429597 5555FFFFF00007777555599991111CCC66669999AAAEEE8888
    POX?Rs& 0000000000000000000000000000000041162E 888833339999FFFF1111CCCC8888CCCC9999EEEEDDD00003333
    [Xq\\\$% 0000000000000000000000000000000097A5F0 66666666EEEEDDD7777FFF00005555FFFFFFF888855551111
    rAmQg4v 000000000000000000000000000000008180D3 BBBB111111119999FFFFFFFFFFF4444BBBB88884444CCCC
    !&))Jf3; 000000000000000000000000000000004602E1 4444FFFC88888888CCCCFFFFCCCC5555666666666666
ubuntu@ip-172-31-13-60:/mnt/raid/64$ tail -10 MergedOutput.txt
~~~%A NB_t 000000000000000000000000000000003DE5EC FFFF7777EEEBBBBB4444EEEEEEE33339999DD999900005555
~~~-c-Q\(> 0000000000000000000000000000000032611 9999FFFF111177773333777700001111444444440000BBBB6666
~~~8Y}Fq[* 00000000000000000000000000000000742A4C BBBB1111CCCCEEE888800000000777733333333DD22225555
~~~-Dd=QT] 0000000000000000000000000000000000000000674C0F 9999DDDD000055556666CCCC22220000FFFEEEEDDDFFF0000
~~~]JA()$ 0000000000000000000000000000000060BCBE 111122223333444455559999AAAABBBB9999FFFFDDDBBBB3333
~~~]Zp.#/+ 000000000000000000000000000000003B9A5A CCC8888EEEAAAEEE333333377770000FFFFCCCC66667777
~~~_Qepix 0000000000000000000000000000000011E5D4 1111999911115555BBBB111100002222EEE6666BBB7777DD
~~~nt=ZH[N 00000000000000000000000000000000332A13 44441111BBBBBBBB33337777FFF4444555555533330000CCCC
~~~s/Pq,-E 000000000000000000000000000000006BE930 2222DDDDDD77771111EEECCCC7777BBBB4444888811111111
~~~zba_Tt 000000000000000000000000000000007F9F4F BBBBCCCC666655559999FFF8888AAA11116666AAAABBBB0000
ubuntu@ip-172-31-13-60:/mnt/raid/64$
```

Hadoop 1 GB Single node:

Running Configuration Nodes:

```
ubuntu@ip-172-31-13-60:/usr/local/hadoop/sbin$ jps
2490 NameNode
2876 Jps
2634 DataNode
ubuntu@ip-172-31-13-60:/usr/local/hadoop/sbin$ ./start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-ubuntu-resourcemanager-ip-172-31-13-60.out
ec2-54-152-42-168.compute-1.amazonaws.com: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-ubuntu-nodemanager-ip-172-31-13-60.out
ubuntu@ip-172-31-13-60:/usr/local/hadoop/sbin$ jps
3301 Jps
2490 NameNode
3070 NodeManager
2634 DataNode
2926 ResourceManager
ubuntu@ip-172-31-13-60:/usr/local/hadoop/sbin$
```

Starting Hadoop Sort (File generation, put and Start):

```
ubuntu@ip-172-31-13-60:/mnt/raid/64$ time ./gensort -a 10000000 /mnt/raid/Input.txt
real 0m3.213s
user 0m1.938s
sys 0m0.188s
ubuntu@ip-172-31-13-60:/mnt/raid/64$ cd ~
ubuntu@ip-172-31-13-60:~$ time hadoop fs -put /mnt/raid/Input.txt /Input
hadoop: command not found

real 0m0.073s
user 0m0.044s
sys 0m0.029s
ubuntu@ip-172-31-13-60:~$ time hadoop fs -put /mnt/raid/Input.txt /Input
16/03/28 18:49:52 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

real 0m5.728s
user 0m8.725s
sys 0m1.009s
ubuntu@ip-172-31-13-60:~$ hadoop jar HadoopTeraSort.jar /Input /Output
16/03/28 18:50:34 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
16/03/28 18:50:35 INFO client.RMProxy: Connecting to ResourceManager at ec2-54-152-42-168.compute-1.amazonaws.com/172.31.13.60:8032
16/03/28 18:50:35 INFO client.RMProxy: Connecting to ResourceManager at ec2-54-152-42-168.compute-1.amazonaws.com/172.31.13.60:8032
16/03/28 18:50:35 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
16/03/28 18:50:35 INFO mapred.FileInputFormat: Total input paths to process : 1
16/03/28 18:50:35 INFO mapreduce.JobSubmitter: number of splits:8
16/03/28 18:50:37 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1459190688387_0001
16/03/28 18:50:37 INFO impl.YarnClientImpl: Submitted application application_1459190688387_0001
16/03/28 18:50:37 INFO mapreduce.Job: The url to track the job: http://ec2-54-152-42-168.compute-1.amazonaws.com:8088/proxy/application_1459190688387_0001/
16/03/28 18:50:37 INFO mapreduce.Job: Running job: job_1459190688387_0001
```

```
16/03/28 18:50:35 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
16/03/28 18:50:35 INFO mapred.FileInputFormat: Total input paths to process : 1
16/03/28 18:50:35 INFO mapreduce.JobSubmitter: number of splits:8
16/03/28 18:50:37 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1459190688387_0001
16/03/28 18:50:37 INFO impl.YarnClientImpl: Submitted application application_1459190688387_0001
16/03/28 18:50:37 INFO mapreduce.Job: The url to track the job: http://ec2-54-152-42-168.compute-1.amazonaws.com:8088/proxy/application_1459190688387_0001/
16/03/28 18:50:37 INFO mapreduce.Job: Running job: job_1459190688387_0001
16/03/28 18:50:43 INFO mapreduce.Job: Job job_1459190688387_0001 running in uber mode : false
16/03/28 18:50:43 INFO mapreduce.Job: map 0% reduce 0%
16/03/28 18:51:00 INFO mapreduce.Job: map 21% reduce 0%
16/03/28 18:51:01 INFO mapreduce.Job: map 31% reduce 0%
16/03/28 18:51:03 INFO mapreduce.Job: map 33% reduce 0%
16/03/28 18:51:04 INFO mapreduce.Job: map 34% reduce 0%
16/03/28 18:51:06 INFO mapreduce.Job: map 45% reduce 0%
16/03/28 18:51:07 INFO mapreduce.Job: map 49% reduce 0%
16/03/28 18:51:10 INFO mapreduce.Job: map 50% reduce 0%
16/03/28 18:51:12 INFO mapreduce.Job: map 53% reduce 0%
16/03/28 18:51:13 INFO mapreduce.Job: map 63% reduce 0%
16/03/28 18:51:15 INFO mapreduce.Job: map 66% reduce 0%
16/03/28 18:51:16 INFO mapreduce.Job: map 75% reduce 0%
16/03/28 18:51:25 INFO mapreduce.Job: map 88% reduce 0%
16/03/28 18:51:26 INFO mapreduce.Job: map 96% reduce 0%
16/03/28 18:51:28 INFO mapreduce.Job: map 96% reduce 29%
16/03/28 18:51:29 INFO mapreduce.Job: map 100% reduce 29%
16/03/28 18:51:31 INFO mapreduce.Job: map 100% reduce 69%
16/03/28 18:51:34 INFO mapreduce.Job: map 100% reduce 76%
16/03/28 18:51:37 INFO mapreduce.Job: map 100% reduce 83%
16/03/28 18:51:40 INFO mapreduce.Job: map 100% reduce 91%
16/03/28 18:51:43 INFO mapreduce.Job: map 100% reduce 98%
16/03/28 18:51:44 INFO mapreduce.Job: map 100% reduce 100%
16/03/28 18:51:44 INFO mapreduce.Job: Job job_1459190688387_0001 completed successfully
16/03/28 18:51:44 INFO mapreduce.Job: Counters: 50
```

Logged in as: d

All Applications



Cluster Metrics

Apps Submitted	Apps Pending	Apps Running	Completed	Containers Running	Memory Used	Memory Total	Memory Reserved	Vcores Used	Vcores Total	Vcores Reserved	Active Nodes	Decommissioned Nodes	Lost Nodes	Unhealthy Nodes	Rebo
1	0	0	1	0	0 B	8 GB	0 B	0	8	0	1	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>

Show 20+ entries

ID	User	Name	Type	Queue	StartTime	FinishTime	State	FinalStatus	Progress	Tracking UI	Blacklist
application_1459190688387_0001	ubuntu	HadoopTeraSort	MAPREDUCE	default	Mon Mar 28 13:50:37 -0500 2016	Mon Mar 28 13:51:43 -0500 2016	FINISHED	SUCCEEDED	0%	History	N/A

Showing 1 to 1 of 1 entries

First Previous 1 Next Last

Logged in as: dr.who

Application application_1459190688387_0001



Kill Application

User: ubuntu	Application Overview
Name: HadoopTeraSort	
Application Type: MAPREDUCE	
Application Tags:	
YarnApplicationState: FINISHED	
FinalStatus Reported by AM: SUCCEEDED	
Started: Mon Mar 28 18:50:37 +0000 2016	
Elapsed: 1mins, 5sec	
Tracking URL: History	
Diagnostics:	

Application Metrics

Total Resource Preempted: <memory:0, vCores:0>	Application Metrics
Total Number of Non-AM Containers Preempted: 0	
Total Number of AM Containers Preempted: 0	
Resource Preempted from Current Attempt: <memory:0, vCores:0>	
Number of Non-AM Containers Preempted from Current Attempt: 0	
Aggregate Resource Allocation: 387903 MB-seconds, 300 vcore-seconds	

Sorted Output:

```

1 Spark * 2 d2isntance * 3 SparkMaster *
ubuntu@ip-172-31-13-60:/mnt/raid$ cd 64/
ubuntu@ip-172-31-13-60:/mnt/raid/64$ time ./valsort part-00000
Records: 10000000
Checksum: 4c4ec89af9d195
Duplicate keys: 0
SUCCESS - all records are in order

real    0m3.320s
user    0m1.595s
sys     0m0.013s
ubuntu@ip-172-31-13-60:/mnt/raid/64$ head -10 part-00000
"0!uve      000000000000000000000000000000001228D4 7777888000022224444DDDDDDDEEE0000000CCC7777DDDD
,K4a:v      000000000000000000000000000000001B8132 5555EEEE888899994444FFFC1111CCCEEEE1111EEEE6666FFFF
.FuD\}u    00000000000000000000000000000000797631 5555DDDBBBB800007777222111122224444DDDDDD99996666
;SYThct   000000000000000000000000000000007D3DF5 2222AAACCCCCFFFFFAAAA44445555EEE44442222DDDD99992222
=2G^9\{-   00000000000000000000000000000000809EE 5555DDDD1111CCCC9999BBBB0000BBBBCCCCFFFFCCCC44443333
N}M9?sP   000000000000000000000000000000429597 5555FFFF0000077755559991111CCCC66669999AAAAEEE8888
POX?Rs&   00000000000000000000000000000091162E 888833399999FFF1111CCCC8888CCCC9999EEEEDDD00003333
[Xq\\$%   0000000000000000000000000000000097A5F0 66666666EEEEDDD7777FFF00005555FFFFFFF88885551111
rAmQg4v   000000000000000000000000000000008180D3 BBBB111111119999FFFFFFFFFFF4444BBBB88884444CCCC
!&)j3;   000000000000000000000000000000004602E1 4444FFFFCCCC88888888CCCCFFFFCCCCEEE5555666666666666
ubuntu@ip-172-31-13-60:/mnt/raid/64$ tail -10 part-00000
~~~%A_NB_t 0000000000000000000000000000003DE5EC FFFF7777EEEEBBBB4444EEEEEEE33339999DDDD999900005555
~~~-c-CQ(> 000000000000000000000000000000832611 9999FFFF111177773333777700001111444444440000BBBB6666
~~~8Y}Fql*  0000000000000000000000000000000742A4C BBBB1111CCCCEEE888800000000077733333333DD22225555
~~~->D-QT]  000000000000000000000000000000674C0F 9999DDDD000055556666CCCC22220000FFFEEEEDDDDFFFF0000
~~~]JA()j$  00000000000000000000000000000060BCBE 111122223333444455559999AAAABB9999FFFFDDDBBB3333
~~~]Zp_/#+  0000000000000000000000000000003B9A5A CCCC8888EEEEAAAEEE33333337770000FFFFCCCC66667777
~~~_jQepix  00000000000000000000000000000011E5D4 1111999911115555BBBB11110000222EEE6666BBBB7777DDDD
~~~nt=ZH[N  000000000000000000000000000000332A13 4444111BBBBBBBBB33337777FFF44445555553330000CCCC
~~~s/Pq,-E  0000000000000000000000000000006BE930 2222DDDDDD77771111EEECCCC7777BBBB4444888811111111
~~~zba_Tt   00000000000000000000000000000079F4F BBBBCCCC666655559999FFF8888AAA11116666AAAABB00000
ubuntu@ip-172-31-13-60:/mnt/raid/64$ 
```

