

TERASORT using HADOOP, SPARK and Shared Memory

Programming assignment – 2

Performance Evaluation

SPRING -2016

by

**Patel Darpan
A20345898**

PROBLEM STATEMENT

In this programming assignment we need to implement sort in 3 different platform mentioned below.

- Shared memory using python (Datasize : 10GB) (Amazon EC2)
- Implement sort on HADOOP (Datasize : 10GB , 100GB) (1 node and 16 node)
- Sort on SPARK cluster (Datasize: 10GB, 100GB) (1 node and 16 node)

Compare all results of Python , HADOOP and SPARK based on sorted times.

EXPERIMENTAL ENVIRONMENT

Software versions

- Java version : JAVA 7 , openjdk -7 – jdk
- Hadoop version : Hadoop-2.7.2
- Spark version : spark-1.6.1
- OS : Ubuntu 15.0.4

Instance Type

- c3.large and c3.4xlarge

```
[ec2-user@ip-172-31-13-167 ~]$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                2
On-line CPU(s) list:  0,1
Thread(s) per core:   2
Core(s) per socket:   1
Socket(s):             1
NUMA node(s):          1
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 62
Model name:            Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
Stepping:               4
CPU MHz:               2800.070
BogoMIPS:              5600.14
Hypervisor vendor:    Xen
Virtualization type:  full
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              25600K
NUMA node0 CPU(s):    0,1
[ec2-user@ip-172-31-13-167 ~]$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                2
On-line CPU(s) list:  0,1
Thread(s) per core:   2
Core(s) per socket:   1
Socket(s):             1
NUMA node(s):          1
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 62
Model name:            Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
Stepping:               4
CPU MHz:               2800.070
BogoMIPS:              5600.14
Hypervisor vendor:    Xen
Virtualization type:  full
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              25600K
NUMA node0 CPU(s):    0,1
[ec2-user@ip-172-31-13-167 ~]$ ■
```

VIRTUAL CLUSTER (1 NODE)

- To create virtual cluster first log in to AMAZON EC2 console
- Create instance by clicking “Launch Instance”
- Select AMI as UBUNTU and proceed further
- Select spot instance and price or on-demand if you want.
- Add storage as per requirement like EBS 30gb for Hadoop 10gb sort
- Configure the security group and select “All TCP” , “All UDP” and “ALL ICMP”
- Review and launch instance
- Click “Instance” from left panel and press “Connect”
- Open the terminal and go to directory content .pem file
- Command for permission of .pem file
chmod 400 ec2-hadoop.pem
- Now login to configured cluster
 - ssh -i “ec2-hadoop.pem” <Public DNS>
- Future use of this instance make an AMI image using “create image” and save it.
- Launch this instance from AMI go to AMIs and launch it on any type of instance
- For 16 node virtual cluster launch 16 instance from saved AMI for slaves and 1 instance for master in case of HADOOP and SPARK

The screenshot shows the AWS EC2 Instances page. The top navigation bar includes AWS, Services, Edit, Darpan Patel, N. Virginia, and Support. On the left, a sidebar lists EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES, Instances, Spot Requests, Reserved Instances, Scheduled Instances, Commands, Dedicated Hosts, IMAGES, and AMIs. The Instances section is currently selected. The main content area displays a table of instances:

	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
	PA2	i-0447d43	c3.xlarge	us-east-1c	running	Initializing	None	ec2-54-208-209-
		i-0d457c8e	t2.micro	us-east-1c	shutting-do...		None	
	Hadoop_10GB	i-85251401	c3.large	us-east-1d	terminated		None	
	Hadoop_mas...	i-18b4859c	c3.large	us-east-1d	terminated		None	

At the bottom, it shows Instance: i-0447d43 (PA2) and Public DNS: ec2-54-208-209-180.compute-1.amazonaws.com.

SHARED-MEMORY (10 GB sort)

- Here, I implemented sorting input file of 10Gb. I made one python program which is working with multi-processing(Multi-Threaded).
- Program is working for 1/2/4/8 threads and measure the sorted time for 10gb input file.
- Calculate throughput and speedup based on output
- I write the output of time and sorted file's 1st 10 lines and last 10 lines into one output file.

Program Methodology

- Program follows the following sequence like 1st chunked large 10gb file into 20 MB small chunks for better and easier sorting.
- First make dictionary in python for whole data of file as key pair value.
- 1st 10 bytes are keys and last 90 bytes of each line values.
- All separate chunks are separate file that contains only key of that particular chunks.
- I am only using multi-threaded concept for sorting separate chunks not for reading a file and merging a file into one bigger file.
- After making an 20MB chunks sort it separately using concept of merge sort. Sort separate each 20 MB file first and then merge it all other sorted chunks. It will take some quality amount of time.
- Then after my program takes 1st key of each sorted chunks sort it again and put them into one final output file.
- I used concept of MERGE SORT through out the program.
- After picking all keys from all 20MB chunks, sort it and merge into one bigger File.
- After that attach value of each key using that dictionary and put them into output file.
- Finally, I get sorted file using my python program.
- After completion of program it will delete all my temporary chunks of 20 MBs.

Programming Steps

- Create 10gb input file using gensort

```
./gensort -a 100000000 <input file name>
```

- I used input file path hard-coded into program as “/mnt/shared_input_10gb.txt” , temporary file path “/mnt/temp/” and output file path as “/mnt/output_shared”.
- Input no. of thread from user input. Then it follows the sequence explained as per above section and give the output file.
- Convert it into DOS from UNIX file format for sorting validation.
- Then check that output file using valsor.

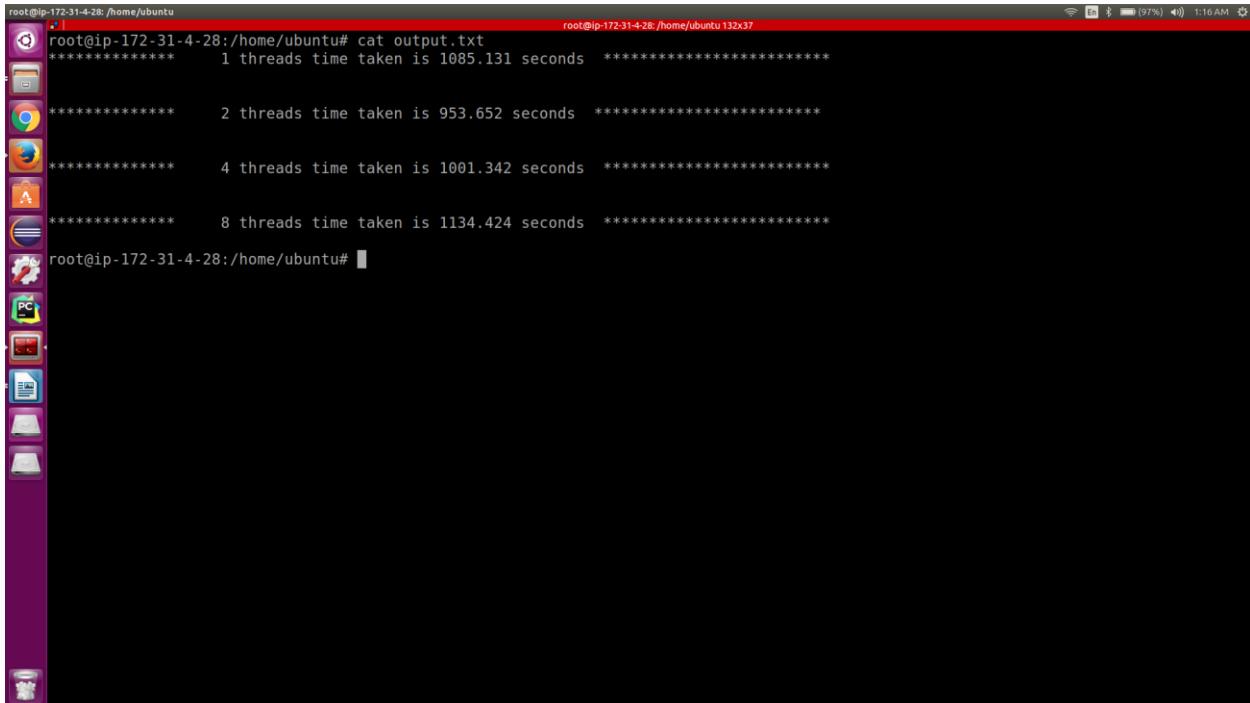
```
unix2dos <output file name>
```

file <file name>

./valsort <output file>

Screen Shot of Output of Shared Memory

Output of timings of sorting for threads 1/2/4/8



```
root@ip-172-31-4-28: /home/ubuntu# cat output.txt
root@ip-172-31-4-28: /home/ubuntu 132x37
*****      1 threads time taken is 1085.131 seconds  *****
*****      2 threads time taken is 953.652 seconds  *****
*****      4 threads time taken is 1001.342 seconds  *****
*****      8 threads time taken is 1134.424 seconds  *****

root@ip-172-31-4-28:/home/ubuntu#
```

Validate of sorted output file.

```
root@ip-172-31-4-28:/home/ubuntu# ls -l /mnt/output_shared/opt_final
-rw-r--r-- 1 root root 9900000000 Mar 28 19:44 /mnt/output_shared/opt_final
root@ip-172-31-4-28:/home/ubuntu# ls -l /mnt/
total 9765808
drwxrwxrwx 2 root      root          16384 Mar 28 10:17 lost+found
drwxrwxrwx 2 root      root          4096 Mar 28 19:33 output shared
-rwxrwxr-x 1 ubuntu    ubuntu 10000000000 Mar 28 10:32 shared_10gb.txt
drwxrwxrwx 2 root      root         159744 Mar 28 19:44 temp
root@ip-172-31-4-28:/home/ubuntu# ls -l /mnt/output_shared/opt_final
-rw-r--r-- 1 root root 9900000000 Mar 28 19:44 /mnt/output_shared/opt_final
root@ip-172-31-4-28:/home/ubuntu# file /mnt/output_shared/opt_final
/mnt/output_shared/opt_final: ASCII text
root@ip-172-31-4-28:/home/ubuntu# unix2dos /mnt/output_shared/opt_final
unix2dos: converting file /mnt/output_shared/opt_final to DOS format ...
root@ip-172-31-4-28:/home/ubuntu# ./64/valsrt /mnt/output_shared/opt_final
Records: 100000000
Checksum: 2faf0ab746e89a8
Duplicate keys: 0
SUCCESS - all records are in order
root@ip-172-31-4-28:/home/ubuntu#
```

First 10 and last 10 lines of sorted output file.

PERFORMANCE

Sorting Datasize =10 GB

Speed Up is considering with respect to Thread -1

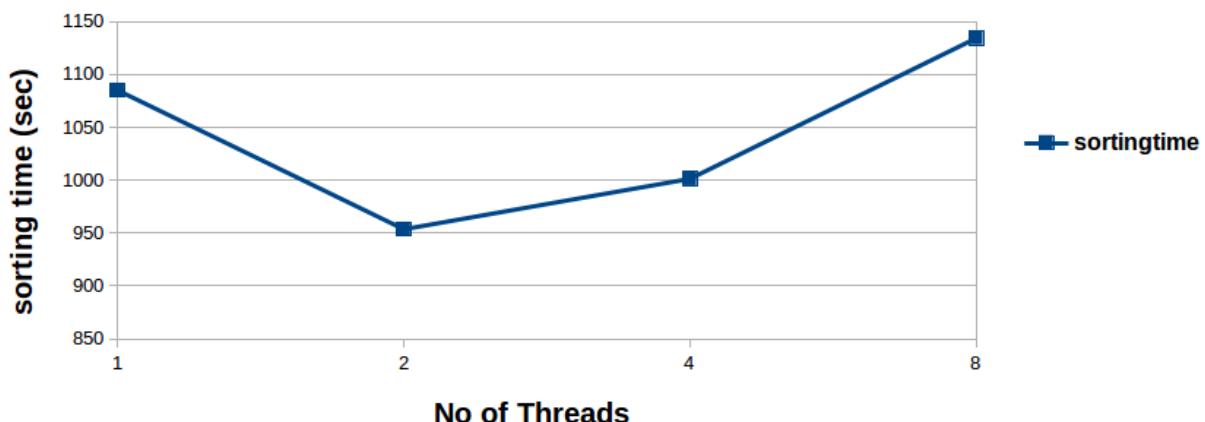
No. of Threads	Throughput (MB/s)	Time for Sort(seconds)	Speed Up
1	9.215	1085.131	1
2	10.486	953.652	1.138
4	9.986	1001.342	1.083
8	8.8150	1134.424	0.956

EVALUATION

- According to the output performance of sorting algorithm gives timings mention above in the table
- 2-thread gives best sorting time and throughput than 1/4/8 threads.
- 2-threads are optimal for Amazon EC2 c3.large instance.
- Speed up comparison gives us the better comparison between outputs.
- According to speed-up comparison 2 threads is fastest threads with **10.486 MB/s** and **953.652 seconds** for **10 GB** data sorting.
- Speed Up is calculated with respect to 1 thread.
- Increase the number of threads then no of intermediate files is increase and overhead of read and write is higher so the throughput is getting low.

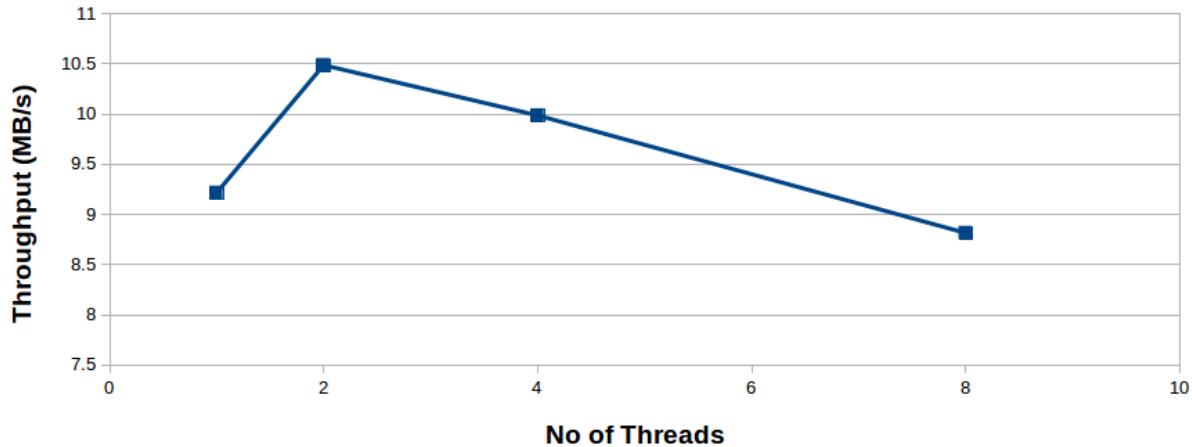
Sorting Time Comparison Graph

No of Threads vs Sorting Time



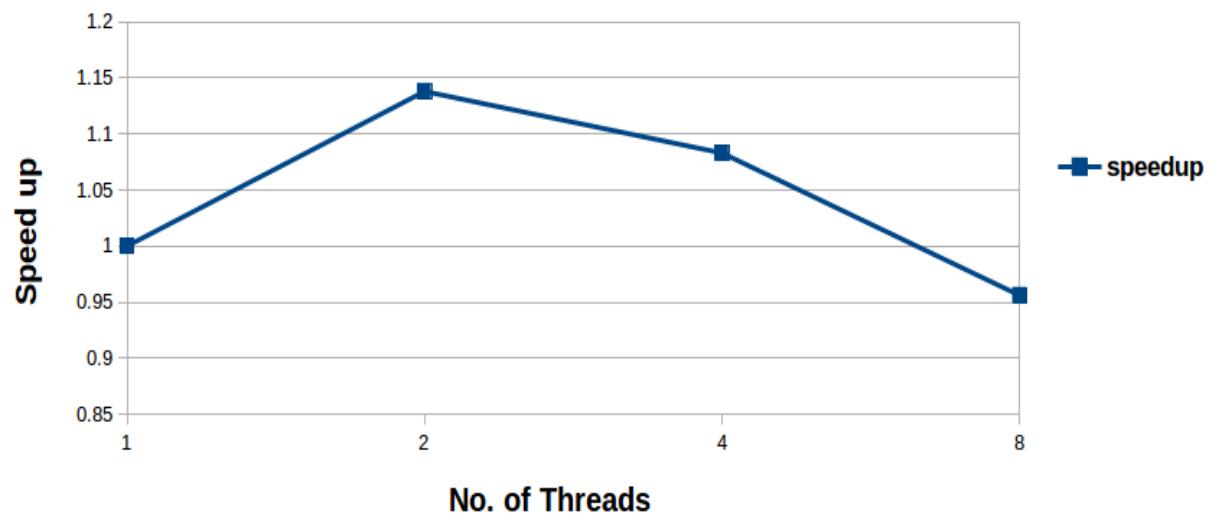
Sorting Throughput Comparision Graph

No. of Threads vs Throughput



Speed up

No. of threads vs Speed up



HADOOP

Hadoop Configuration

- Login to AMAZON EC2 console
- Launch a new c3.large instance as mentioned in Virtual 1 node configuration
- ssh to that instance and login into it.
- Install java on instance

```
sudo apt-get update  
sudo apt-get install openjdk-7-jdk
```

- Get Hadoop package from internet and then unzip it.

```
wget https://archive.apache.org/dist/hadoop/core/hadoop-  
2.7.2/hadoop-2.7.2.tar.gz  
  
sudo tar -xvf hadoop-2.7.2.tar.gz  
  
mv hadoop-2.7.2 hadoop
```

- Copy .pem file to EC2 instance
- Give permission to .pem file using following command

```
eval `ssh-agent -s`  
chmod 600 hadoop.pem  
ssh-add hadoop.pem
```

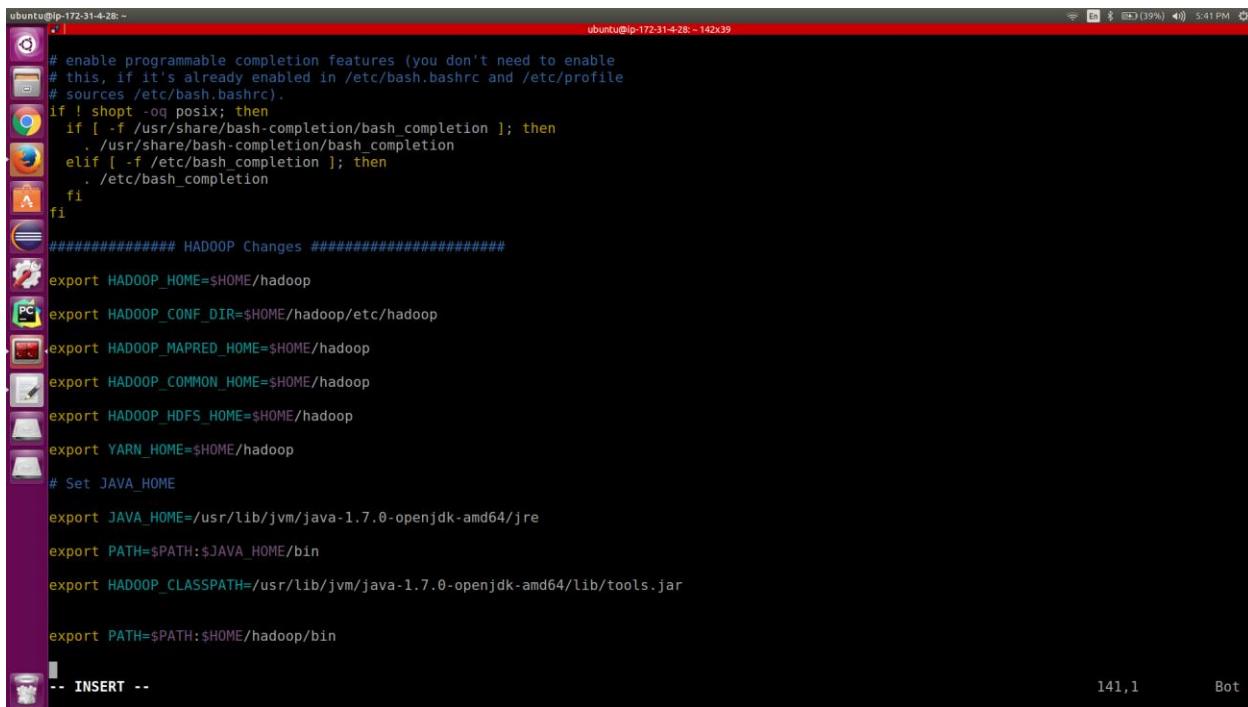
- Here mainly we need to changes into following files
 - .bashrc
 - etc/hadoop/core-site.xml
 - etc/hadoop/mapred-site.xml
 - etc/hadoop/hdfs-site.xml
 - etc/hadoop/yarn-site.xml
 - etc/hadoop/master
 - etc/hadoop/slaves
- Changes into .bashrc file

```
sudo vi .bashrc
```

Add

```
# Set Hadoop-related environment variables
```

```
export HADOOP_HOME=$HOME/hadoop  
export HADOOP_CONF_DIR=$HOME/hadoop/etc/hadoop  
export HADOOP_MAPRED_HOME=$HOME/hadoop  
export HADOOP_COMMON_HOME=$HOME/hadoop  
export HADOOP_HDFS_HOME=$HOME/hadoop  
export YARN_HOME=$HOME/hadoop  
  
# Set JAVA_HOME  
  
export JAVA_HOME=/usr/lib/jvm/java-1.7.0-openjdk-amd64/jre  
export PATH=$PATH:$JAVA_HOME/bin  
  
export HADOOP_CLASSPATH=/usr/lib/jvm/java-1.7.0-openjdk-  
amd64/lib/tools.jar  
  
# Add Hadoop bin/ directory to PATH  
  
export PATH=$PATH:$HOME/hadoop/bin
```



The screenshot shows a terminal window on an Ubuntu desktop environment. The terminal window title is "ubuntu@ip-172-31-4-28: ~". The window contains the command-line code from the previous block, which sets various Hadoop environment variables and the Java home path. The terminal window has a dark background with light-colored text. The desktop icons are visible at the top of the screen.

- Changes into core-site.xml

```
sudo vi core-site.xml
```

Add it into file

```
<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://master:9000</value>
</property>
<property>
<name>hadoop.tmp.dir</name>
<value>/mnt/raid/tmp/hadoop-ec2-user</value>
</property>
</configuration>
```

The screenshot shows a terminal window on a Linux desktop. The terminal title is "ubuntu@ip-172-31-1-80: ~hadoop/etc/hadoop". The window displays the XML configuration for Hadoop's core-site. The configuration includes properties for the default file system name and the temporary directory. The terminal interface shows standard vi keybindings and a status bar at the bottom indicating the file is 31L and 959C.

```
ubuntu@ip-172-31-1-80: ~hadoop/etc/hadoop
[...]
<?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://master:9000</value>
</property>
<property>
<name>hadoop.tmp.dir</name>
<value>/mnt/raid/tmp/hadoop-ec2-user</value>
</property>
</configuration>
[...]
"core-site.xml" 31L, 959C
1,1          All
```

- Changes to Hadoop hdfs-site.xml

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

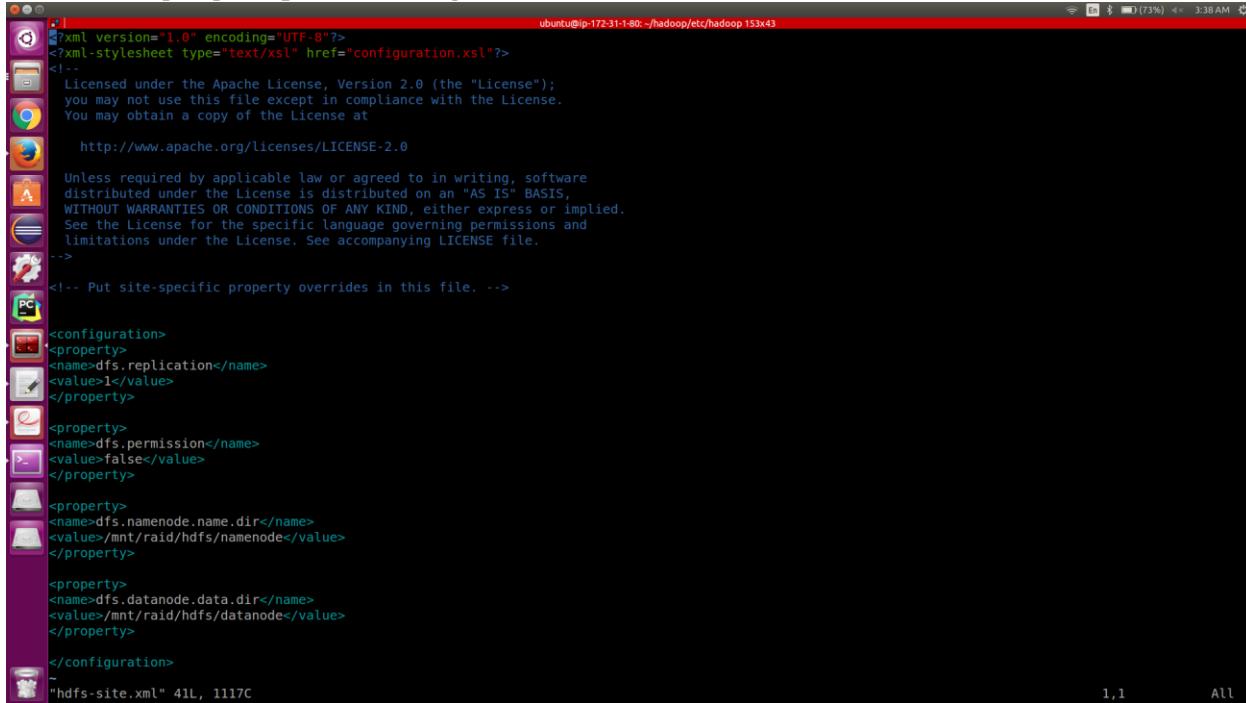
<property>
<name>dfs.permission</name>
<value>false</value>
</property>
```

```

<property>
<name>dfs.namenode.name.dir</name>
<value>/mnt/raid/hdfs/namenode</value>
</property>

<property>
<name>dfs.datanode.data.dir</name>
<value>/mnt/raid/hdfs/datanode</value>
</property> </configuration>

```



```

?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.permission</name>
<value>false</value>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>/mnt/raid/hdfs/namenode</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>/mnt/raid/hdfs/datanode</value>
</property>
</configuration>

```

"hdfs-site.xml" 41L, 1117C

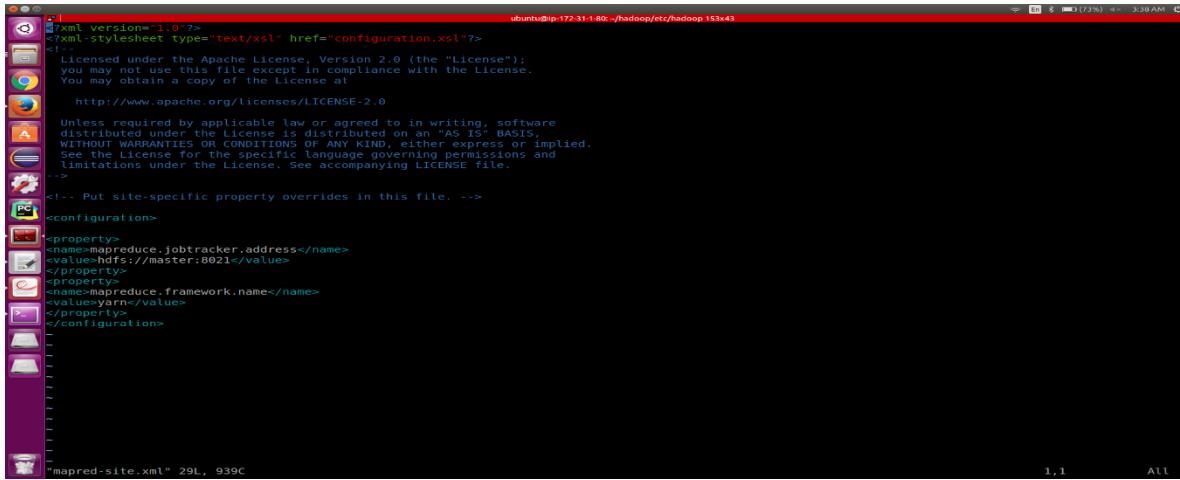
- Changes to mapred-site.xml

```

<configuration>

<property>
<name>mapreduce.jobtracker.address</name>
<value>hdfs://master:8021</value>
</property>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>

```



```

<?xml version="1.0"?>
<!DOCTYPE configuration SYSTEM "http://www.apache.org/xml/ns/policy10.dtd"
  <!--@version="1.0"-->
<!-- 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://master:8021</value>
</property>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>

```

- Changes to yarn-site.xml

```

<configuration>

    <!-- Site specific YARN configuration properties -->
    <property>
        <name>yarn.nodemanager.aux-services</name>
        <value>mapreduce_shuffle</value>
    </property>
    <property>
        <name>yarn.resourcemanager.resource-
tracker.address</name>
        <value>master:8031</value>
    </property>
    <property>
        <name>yarn.resourcemanager.address</name>
        <value>master:8032</value>
    </property>
    <property>
        <name>yarn.resourcemanager.scheduler.address</name>
        <value>master:8030</value>
    </property>
    <property>
        <name>yarn.resourcemanager.admin.address</name>
        <value>master:8133</value>
    </property>
    <property>
        <name>yarn.resourcemanager.webapp.address</name>
        <value>master:8088</value>
    </property>
</configuration>

```

- Changes to masters file
 - Add “masters” at first line.
- Generate KeyGen for password less login

```
ssh-keygen -f ~/.ssh/id_rsa -t rsa -P ""

cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

cat ~/.ssh/id_rsa.pub | ssh slave1 'cat >> ~/.ssh/authorized_keys'
```

- Create AMI of this 1 instance to make 16 slave and 1 master for 100gb sorting.

Hadoop 1 master and 1 slave configuration (10 GB sort)

- Launch instance from AMI which has already installed Hadoop configuration.
- To make separate config file into .ssh folder

cd ~/.ssh

sudo vi config

add Hostname , Public DNS and identity of .pem file

save it.

- Then add personal IPs of master and slaves into /etc/hosts file.

sudo vi /etc/hosts

add IPs as

172.31.1.435 master

172.56.2.531 slave1

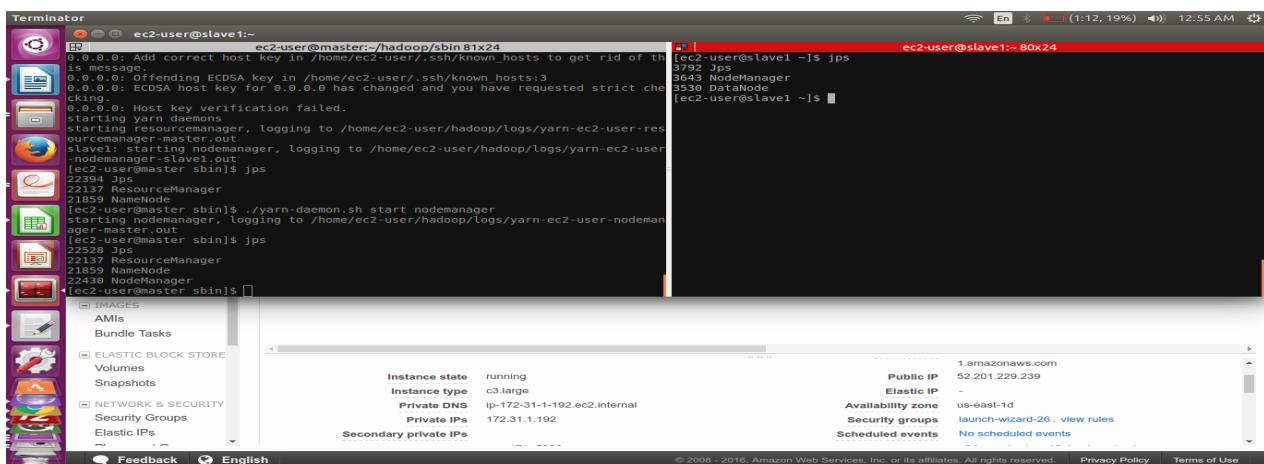
- Then start and format namenode into master

hadoop namenode -format

hadoop sbin/start-all.sh

hadoop sbin/yarn-daemon.sh start nodemanager

jps



- We can run our Python Terasort program for 10 GB data
- First we need to create 10 GB data using gensort

```
./gensort -a 100000000 /mnt/hadoop_tera_10gb.txt
```

- Now we have to put this file into HDFS file system using following command

```
./hadoop fs -put /mnt/Hadoop_tera_10gb.txt /user/hadoop/input/hadoop_tera_10gb.txt
```

- Run hadoop_tera.py using following command

```
./hadoop hadoop-mapper.py|hadoop-reducer.py <input-path> <output-file-path>
```

- Then we have validate output using following command

```
unix2dos <output file>
```

```
./valsor t <output file>
```

```
ec2-user@master:~ ec2-user@master:~ 85x42
16/03/28 10:23:33 INFO mapreduce.Job: map 8% reduce 0%
16/03/28 10:23:37 INFO mapreduce.Job: map 9% reduce 0%
16/03/28 10:23:46 INFO mapreduce.Job: map 10% reduce 0%
16/03/28 10:23:51 INFO mapreduce.Job: map 11% reduce 0%
16/03/28 10:23:55 INFO mapreduce.Job: map 12% reduce 0%
16/03/28 10:23:56 INFO mapreduce.Job: map 13% reduce 0%
16/03/28 10:23:59 INFO mapreduce.Job: map 14% reduce 0%
16/03/28 10:24:04 INFO mapreduce.Job: map 15% reduce 0%
16/03/28 10:24:11 INFO mapreduce.Job: map 16% reduce 0%
16/03/28 10:24:14 INFO mapreduce.Job: map 17% reduce 0%
16/03/28 10:24:19 INFO mapreduce.Job: map 18% reduce 0%
16/03/28 10:24:21 INFO mapreduce.Job: map 19% reduce 0%
16/03/28 10:24:31 INFO mapreduce.Job: map 20% reduce 0%
16/03/28 10:24:37 INFO mapreduce.Job: map 20% reduce 1%
16/03/28 10:24:38 INFO mapreduce.Job: map 21% reduce 1%
16/03/28 10:24:40 INFO mapreduce.Job: map 21% reduce 2%
16/03/28 10:24:43 INFO mapreduce.Job: map 21% reduce 3%
16/03/28 10:24:44 INFO mapreduce.Job: map 22% reduce 3%
16/03/28 10:24:46 INFO mapreduce.Job: map 22% reduce 4%
16/03/28 10:24:56 INFO mapreduce.Job: map 23% reduce 4%
16/03/28 10:24:59 INFO mapreduce.Job: map 24% reduce 5%
16/03/28 10:25:00 INFO mapreduce.Job: map 25% reduce 5%
16/03/28 10:25:02 INFO mapreduce.Job: map 26% reduce 5%
16/03/28 10:25:05 INFO mapreduce.Job: map 27% reduce 5%
16/03/28 10:25:06 INFO mapreduce.Job: map 28% reduce 5%
16/03/28 10:25:09 INFO mapreduce.Job: map 29% reduce 5%
16/03/28 10:25:17 INFO mapreduce.Job: map 30% reduce 5%
16/03/28 10:25:21 INFO mapreduce.Job: map 30% reduce 6%
16/03/28 10:25:24 INFO mapreduce.Job: map 31% reduce 6%
16/03/28 10:25:27 INFO mapreduce.Job: map 32% reduce 8%
16/03/28 10:25:31 INFO mapreduce.Job: map 33% reduce 8%
16/03/28 10:25:43 INFO mapreduce.Job: map 34% reduce 8%
16/03/28 10:25:46 INFO mapreduce.Job: map 35% reduce 8%
16/03/28 10:25:48 INFO mapreduce.Job: map 36% reduce 8%
16/03/28 10:25:51 INFO mapreduce.Job: map 37% reduce 8%
16/03/28 10:25:53 INFO mapreduce.Job: map 38% reduce 8%
16/03/28 10:25:55 INFO mapreduce.Job: map 39% reduce 8%
16/03/28 10:25:58 INFO mapreduce.Job: map 39% reduce 9%
16/03/28 10:26:01 INFO mapreduce.Job: map 39% reduce 10%
16/03/28 10:26:07 INFO mapreduce.Job: map 39% reduce 11%
16/03/28 10:26:09 INFO mapreduce.Job: map 40% reduce 11%
```



```
ec2-user@slave1:~ 76x42
dipen@hawk:~/Downloads/assign2$ ssh -i "hadoop.pem" ec2-user@ec2-52-201-229-239.compute-1.amazonaws.com
Last login: Mon Mar 28 10:07:01 2016 from 208-59-157-46.c3-0.mcm-ubr1.chi-mc.mil.cable.rcn.com
[ec2-user@slave1 ~]$ ls
[ec2-user@slave1 ~]$ jps
3734 Jps
3659 YarnChild
3629 YarnChild
3642 YarnChild
3888 NodeManager
3604 YarnChild
2983 DataNode
3635 YarnChild
3603 YarnChild
3602 YarnChild
3632 YarnChild
[ec2-user@slave1 ~]$ [ ]
```

```

ec2-user@master:~$ hadoop job -status
[...]
Map-Reduce framework
  Map input records=1000000000
  Map output records=1000000000
  Map output bytes=99000000000
  Map output compressed bytes=10100000450
  Input split bytes=8550
  Combine input records=1000000000
  Combine output records=1000000000
  Reduce shuffle bytes=1000000000
  Reduce shuffle compressed bytes=726970
  Total megabyte-milliseconds taken by all map tasks=6202082
  Total megabyte-milliseconds taken by all reduce tasks=744417280
  Total megabyte-milliseconds taken by all reduce tasks=744417280

Shuffled Maps =75
Failed Shuffles=0
HDFS Read= output=75
GC Time elapsed (ms)=29032
CPU time spent (ms)=1602436
Physical memory (bytes) snapshot=19410812928
Virtual memory (bytes) snapshot=74033293056
Total committed heap usage (bytes)=14615576576

Shuffle Errors:
  File Input Format Counters
    Bytes Read=10000000000
    Bytes Written=99000000000
  File Output Format Counters
    Bytes Written=99000000000
[ec2-user@master ~]$ 

ec2-user@slave1:~$ jps
3734 Jps
3659 YarnChild
3640 YarnChild
3642 YarnChild
3688 NodeManager
3604 YarnChild
3634 YarnChild
3635 YarnChild
3603 YarnChild
3636 YarnChild
3632 YarnChild
[ec2-user@slave1 ~]$ 

```

```

ec2-user@master:~/64$ ./valsort /mnt/raid/part-r-00000
Records: 100000000
Checksum: 2faef2a1dfa8909
duplicate keys: 0
SUCCESS: all records are in order
[ec2-user@master 64]$

```

Namenode Information - Google Chrome

ec2-user@master:~/64#tab-overview

Summary

Security is off.

Safemode is off.

23 files and directories, 81 blocks = 104 total filesystem object(s).

Heap Memory used 138.13 MB of 174.5 MB Heap Memory. Max Heap Memory is 889 MB.

Non Heap Memory used 33.14 MB of 34.44 MB Committed Non Heap Memory. Max Non Heap Memory is 214 MB.

Configured Capacity:	59.43 GB
DFS Used:	9.39 GB (15.79%)
Non DFS Used:	8.84 GB
DFS Remaining:	41.21 GB (69.33%)
Block Pool Used:	9.39 GB (15.79%)
DataNodes usages% (Min/Median/Max/stdDev):	15.79% / 15.79% / 15.79% / 0.00%
Live Nodes	1 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)
Decommissioning Nodes	0
Total Datanode Volume Failures	0 (0 B)
Number of Under-Replicated Blocks	2
Number of Blocks Pending Deletion	0
Block Deletion Start Time	3/28/2016, 4:57:53 AM

core-site.xml gensor-tlin....tar.gz Show all downloads...

Hadoop 100Gb Tera Sort on 1 Mater and 16 Slaves

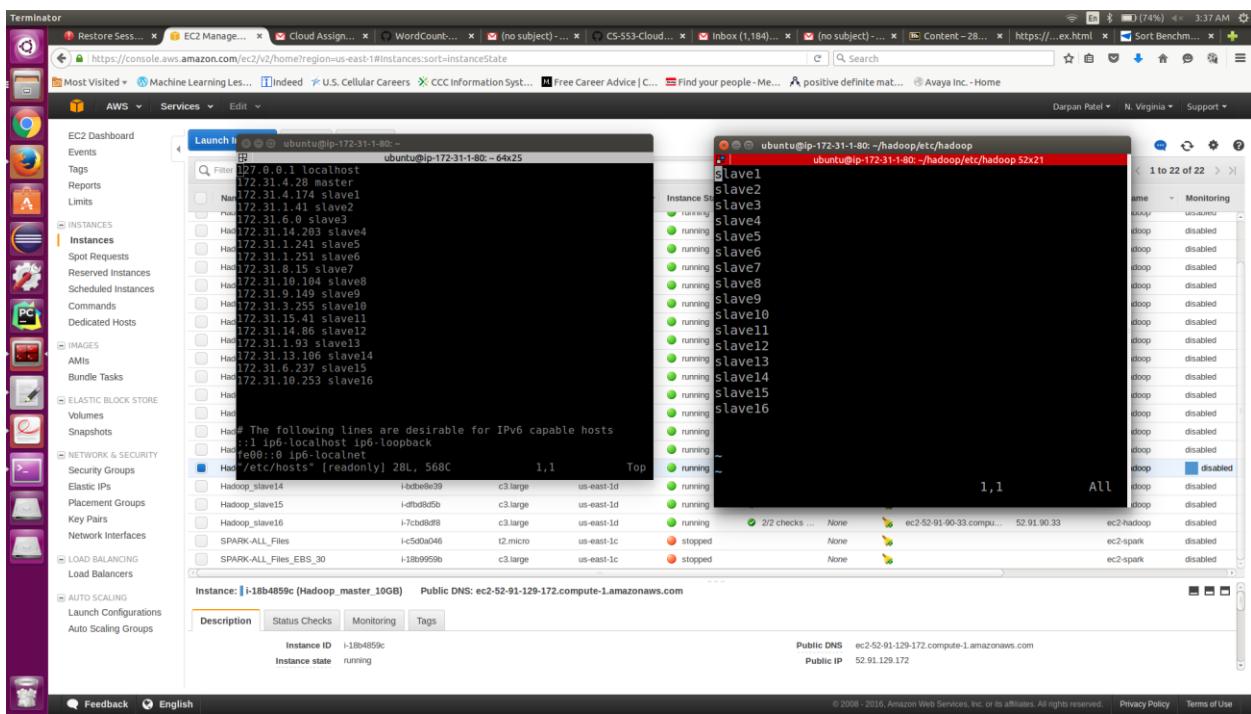
- Login into AMAZON EC2 console
- Launch 1 master node and 16 slaves node from pre-configured Hadoop AMI
- Add all slaves IPs into /etc/hosts

`sudo vi /etc/hosts`

add all slaves IPs like mentioned below.

Name	Request ID	Max Price	AMI ID	Instance	Instance Type	Status	State
Hadoop_10GB	sr-029ngww5	\$0.03	ami-fcc3e698	i-85251401	c3.large	active	fulfilled
Hadoop_slave1	sr-029hbx9	\$0.03	ami-c3903a9	i-53d4ef5d7	c3.large	active	fulfilled
Hadoop_16S	sr-029hbx6	\$0.03	ami-c3903a9	i-18b44bf0c	c3.large	active	fulfilled
sr-029hbxnf	\$0.03	ami-c3903a9		c3.large	open	pending-evalu...	
sr-029hbxz2	\$0.03	ami-c3903a9		c3.large	open	pending-evalu...	
sr-029hbxv	\$0.03	ami-c3903a9		c3.large	open	pending-evalu...	
sr-029hbxg0	\$0.03	ami-c3903a9		c3.large	open	pending-evalu...	
sr-029hbxse	\$0.03	ami-c3903a9		c3.large	open	pending-evalu...	
sr-029hbx3x0	\$0.03	ami-c3903a9		c3.large	open	pending-evalu...	
sr-029hbxm39	\$0.03	ami-c3903a9		c3.large	open	pending-evalu...	
sr-029hbxk	\$0.03	ami-c3903a9		c3.large	open	pending-evalu...	
sr-029hbx9v	\$0.03	ami-c3903a9		c3.large	open	pending-evalu...	
sr-029hbx09	\$0.03	ami-c3903a9		c3.large	open	pending-evalu...	
sr-029hbxw0	\$0.03	ami-c3903a9		c3.large	open	pending-evalu...	
sr-029hbxq0	\$0.03	ami-c3903a9		c3.large	open	pending-evalu...	
sr-029hbx005	\$0.03	ami-c3903a9		c3.large	open	pending-evalu...	
sr-029hbxmwp	\$0.03	ami-c3903a9		c3.large	open	pending-evalu...	
sr-029hbx90k	\$0.03	ami-c3903a9		c3.large	open	pending-evalu...	
sr-029hbx9fg	\$0.03	ami-c3903a9		c3.large	open	pending-evalu...	
sr-029hbx1nz	\$0.03	ami-c3903a9		c3.large	open	pending-evalu...	

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name	Monitoring
Hadoop_10GB	i-85251401	c3.large	us-east-1d	running	2/2 checks ...	None	ec2-54-86-54-63.compu...	54.86.54.63	ec2-hadoop	disabled
Hadoop_master	i-02bc8c86	c3.large	us-east-1d	running	2/2 checks ...	None	ec2-54-88-121-138.com...	54.88.121.138	ec2-hadoop	disabled
Hadoop_slave1	i-dbd8d05a	c3.large	us-east-1d	running	2/2 checks ...	None	ec2-52-201-239-233.co...	52.201.239.233	ec2-hadoop	disabled
Hadoop_slave2	i-babef03e	c3.large	us-east-1d	running	2/2 checks ...	None	ec2-52-207-250-122.co...	52.207.250.122	ec2-hadoop	disabled
Hadoop_slave3	i-02bc8c87	c3.large	us-east-1d	running	2/2 checks ...	None	ec2-52-87-221-220.com...	52.87.221.220	ec2-hadoop	disabled
Hadoop_slave4	i-7dbd8d9	c3.large	us-east-1d	running	2/2 checks ...	None	ec2-52-90-237-78.com...	52.90.237.78	ec2-hadoop	disabled
Hadoop_slave5	i-7bb8d80	c3.large	us-east-1d	running	2/2 checks ...	None	ec2-52-91-36-242.com...	52.91.36.242	ec2-hadoop	disabled
Hadoop_slave6	i-00bc8c84	c3.large	us-east-1d	running	2/2 checks ...	None	ec2-52-90-18-239.com...	52.90.18.239	ec2-hadoop	disabled
Hadoop_slave7	i-00bc8c8d	c3.large	us-east-1d	running	2/2 checks ...	None	ec2-52-90-23-148.com...	52.90.23.148	ec2-hadoop	disabled
Hadoop_slave8	i-dbd8d059	c3.large	us-east-1d	running	2/2 checks ...	None	ec2-52-20-254-53.com...	52.20.254.53	ec2-hadoop	disabled
Hadoop_slave9	i-dbd8d05e	c3.large	us-east-1d	running	2/2 checks ...	None	ec2-54-94-56-20.compu...	54.85.56.20	ec2-hadoop	disabled
Hadoop_slave10	i-bb8e3f	c3.large	us-east-1d	running	2/2 checks ...	None	ec2-52-91-192-48.com...	52.91.192.48	ec2-hadoop	disabled
Hadoop_slave11	i-dbd8d058	c3.large	us-east-1d	running	2/2 checks ...	None	ec2-52-23-223-116.com...	52.23.223.116	ec2-hadoop	disabled
Hadoop_slave100B	i-53d4ef5d7	c3.large	us-east-1d	running	2/2 checks ...	None	ec2-52-90-126-194.com...	52.90.126.194	ec2-hadoop	disabled
Hadoop_slave12	i-01bc8c95	c3.large	us-east-1d	running	2/2 checks ...	None	ec2-52-90-24-52.compu...	52.90.24.52	ec2-hadoop	disabled
Hadoop_slave13	i-dbd8d05f	c3.large	us-east-1d	running	2/2 checks ...	None	ec2-52-91-219-142.com...	52.91.219.142	ec2-hadoop	disabled
Hadoop_master_10GB	i-18b44bf0c	c3.large	us-east-1d	running	2/2 checks ...	None	ec2-52-91-129-172.com...	52.91.129.172	ec2-hadoop	disabled
Hadoop_slave14	i-bbd8e39	c3.large	us-east-1d	running	2/2 checks ...	None	ec2-52-23-173-217.com...	52.23.173.217	ec2-hadoop	disabled
Hadoop_slave15	i-dbd8d05b	c3.large	us-east-1d	running	2/2 checks ...	None	ec2-52-89-133-233.com...	54.89.133.233	ec2-hadoop	disabled
Hadoop_slave16	i-7cb8d0f8	c3.large	us-east-1d	running	2/2 checks ...	None	ec2-52-91-90-33.compu...	52.91.90.33	ec2-hadoop	disabled
SPARK_ALL_Files	i-5cd0a046	t2.micro	us-east-1c	stopped	2/2 checks ...	None	ec2-52-91-90-33.compu...	52.91.90.33	ec2-spark	disabled
SPARK_ALL_Files_EBS_30	i-18b9959b	c3.large	us-east-1c	stopped	2/2 checks ...	None	ec2-52-91-90-33.compu...	52.91.90.33	ec2-spark	disabled



- Add Hostnames of slaves into config file

```
sudo vi .ssh config
```

```
ubuntu@ip-172-31-1-80:~/.ssh
[...]
Host master
HostName ec2-52-91-129-172.compute-1.amazonaws.com
User ubuntu
IdentityFile ~/.ssh/ec2-hadoop.pem

Host slave1
HostName ec2-52-90-126-194.compute-1.amazonaws.com
User ubuntu
IdentityFile ~/.ssh/ec2-hadoop.pem
Host slave2
HostName ec2-52-201-239-233.compute-1.amazonaws.com
User ubuntu
IdentityFile ~/.ssh/ec2-hadoop.pem
Host slave3
HostName ec2-52-207-250-122.compute-1.amazonaws.com
User ubuntu
IdentityFile ~/.ssh/ec2-hadoop.pem
Host slave4
HostName ec2-52-87-221-220.compute-1.amazonaws.com
User ubuntu
IdentityFile ~/.ssh/ec2-hadoop.pem
Host slave5
HostName ec2-52-99-237-78.compute-1.amazonaws.com
User ubuntu
IdentityFile ~/.ssh/ec2-hadoop.pem
Host slave6
HostName ec2-52-91-36-242.compute-1.amazonaws.com
User ubuntu
IdentityFile ~/.ssh/ec2-hadoop.pem
Host slave7
HostName ec2-52-90-18-239.compute-1.amazonaws.com
User ubuntu
IdentityFile ~/.ssh/ec2-hadoop.pem
Host slave8
HostName ec2-52-90-23-148.compute-1.amazonaws.com
User ubuntu
IdentityFile ~/.ssh/ec2-hadoop.pem
Host slave9
HostName ec2-52-85-56-20.compute-1.amazonaws.com
User ubuntu
IdentityFile ~/.ssh/ec2-hadoop.pem
Host slave10
'config' 71L, 2028C
```

- Mount disk into master and all **16** slaves using following command

```
sudo mdadm --create --verbose /dev/md0 --level=0 --name=hadoop --raid-devices=2 /dev/xvdb
/dev/xvdc
sudo mkfs.ext4 -L hadoop /dev/md0
sudo mkdir -p /mnt/raid
sudo mount LABEL=hadoop /mnt/raid
sudo chmod 777 /mnt/raid
df -h
```

- Change the /etc/hadoop/slaves using following command

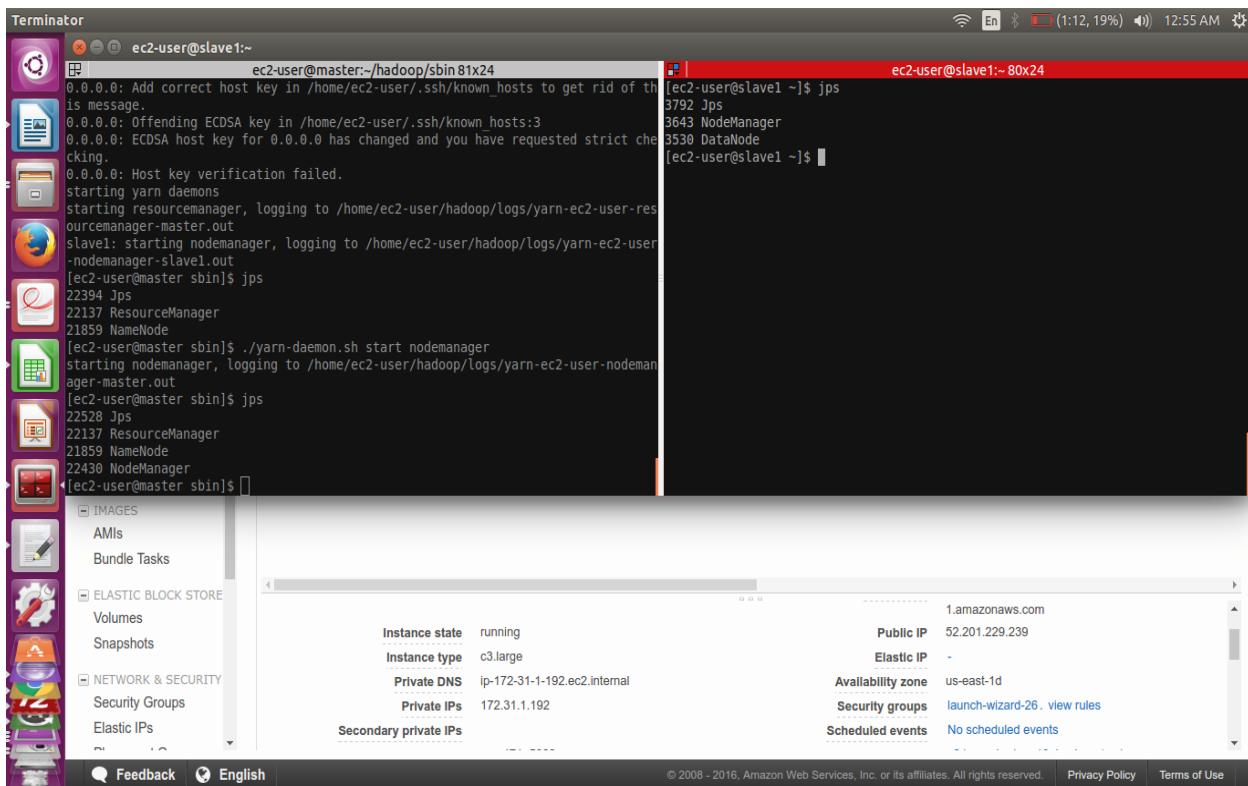
```
sudo vi /etc/hadoop/slaves
```

add slaves private IP into it.

Repeat this for all 16 slaves.

- Now run the following command and check namenode is working or not.
 - hadoop namenode -format
 - hadoop sbin/start-all.sh
 - hadoop sbin/yarn-daemon.sh start nodemanager
 - jps

```
ubuntu@ip-172-31-1-80:~$ ./hadoop namenode -format
16/03/28 08:49:29 INFO blockmanagement.BlockManager: maxReplicationStreams = 2
16/03/28 08:49:29 INFO blockmanagement.BlockManager: replicationRecheckInterval = 3000
16/03/28 08:49:29 INFO blockmanagement.BlockManager: encryptDataTransfer = false
16/03/28 08:49:29 INFO blockmanagement.BlockManager: maxNumBlocksToLog = 1000
16/03/28 08:49:29 INFO namenode.FSNamesystem: fsOwner = ubuntu (auth:SIMPLE)
16/03/28 08:49:29 INFO namenode.FSNamesystem: supergroup = supergroup
16/03/28 08:49:29 INFO namenode.FSNamesystem: isPermissionEnabled = true
16/03/28 08:49:29 INFO namenode.FSNamesystem: HA Enabled: false
16/03/28 08:49:29 INFO namenode.FSNamesystem: Append Enabled: true
16/03/28 08:49:29 INFO util.GSet: Computing capacity for map INodeMap
16/03/28 08:49:29 INFO util.GSet: VM type = 64-bit
16/03/28 08:49:29 INFO util.GSet: 1.0% max memory 889 MB = 8.9 MB
16/03/28 08:49:29 INFO util.GSet: capacity = 2^20 = 1048576 entries
16/03/28 08:49:29 INFO namenode.FSDirectory: ACLs enabled? false
16/03/28 08:49:29 INFO namenode.FSDirectory: XAttr enabled? true
16/03/28 08:49:29 INFO namenode.FSDirectory: Maximum size of an xattr: 16384
16/03/28 08:49:29 INFO namenode.NameNode: Caching file names occurring more than 10 times
16/03/28 08:49:29 INFO util.GSet: Computing capacity for map cachedBlocks
16/03/28 08:49:29 INFO util.GSet: VM type = 64-bit
16/03/28 08:49:29 INFO util.GSet: 0.25% max memory 889 MB = 2.2 MB
16/03/28 08:49:29 INFO util.GSet: capacity = 2^18 = 262144 entries
16/03/28 08:49:29 INFO namenode.FSNamesystem: dfs.namenode.safemode.threshold-pct = 0.9990000128746033
16/03/28 08:49:29 INFO namenode.FSNamesystem: dfs.namenode.safemode.min.datanodes = 0
16/03/28 08:49:29 INFO namenode.FSNamesystem: dfs.namenode.safemode.extension = 30000
16/03/28 08:49:29 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.window.num.buckets = 10
16/03/28 08:49:29 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.num.users = 10
16/03/28 08:49:29 INFO metrics.TopMetrics: NNTop conf: dfs.namenode.top.windows.minutes = 1,5,25
16/03/28 08:49:29 INFO namenode.FSNamesystem: Retry cache on namenode is enabled
16/03/28 08:49:29 INFO namenode.FSNamesystem: Retry cache will use 0.03 of total heap and retry cache entry expiry time is 600000 millis
16/03/28 08:49:29 INFO util.GSet: Computing capacity for map NameNodeRetryCache
16/03/28 08:49:29 INFO util.GSet: VM type = 64-bit
16/03/28 08:49:29 INFO util.GSet: 0.029999999329447746% max memory 889 MB = 273.1 KB
16/03/28 08:49:29 INFO util.GSet: capacity = 2^15 = 32768 entries
Re-format filesystem in Storage Directory /mnt/raid/hdfs/namenode ? (Y or N) y
16/03/28 08:49:31 INFO namenode.FSImage: Allocated new BlockPoolId: BP-830186121-172.31.1.80-1459154971343
16/03/28 08:49:31 INFO common.Storage: Storage directory /mnt/raid/hdfs/namenode has been successfully formatted.
16/03/28 08:49:31 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
16/03/28 08:49:31 INFO util.ExitUtil: Exiting with status 0
16/03/28 08:49:31 INFO namenode.NameNode: SHUTDOWN MSG:
*****
SHUTDOWN MSG: Shutting down NameNode at ip-172-31-1-80.ec2.internal/172.31.1.80
*****
ubuntu@ip-172-31-1-80:~$
```



- Now We can run our Hadoop Python program for 100Gb terasort same as I ran above in 1 Master and 1 slave configuration.
- But Inside process is different than above scenarios.
- After running that program we will get one sorted output file
- We need to validate it same as above
- I made one 1st lines of sorted output and last 10 lines output into one text file named “Sort-hadoop-100GB.txt”

```
[ec2-user@master 64]$ ./valsort /mnt/raid/op2.txt
Records: 499987291
Checksum: ee683173c0538cf
Duplicate keys: 0
SUCCESS - all records are in order
[ec2-user@master 64]$ ]
```

```
[ec2-user@master ~]$ hadoop fs -put /mnt/raid/10gb_data.txt
input
[ec2-user@master ~]$ hadoop jar hs.jar SortHadoop input out
out_1gb
16/03/26 18:27:07 INFO client.RMProxy: Connecting to ResourceManager at master/172.31.29.116:8032
16/03/26 18:27:07 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
16/03/26 18:27:08 INFO input.FileInputFormat: Total input paths to process : 1
16/03/26 18:27:08 INFO mapreduce.JobSubmitter: number of splits:75
16/03/26 18:27:08 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1459016465142_0001
16/03/26 18:27:09 INFO impl.YarnClientImpl: Submitted application application_1459016465142_0001
16/03/26 18:27:09 INFO mapreduce.Job: The url to track the job: http://master:8088/proxy/application_1459016465142_0001
1/
16/03/26 18:27:09 INFO mapreduce.Job: Running job: job_1459016465142_0001
16/03/26 18:27:17 INFO mapreduce.Job: Job job_1459016465142_0001 running in uber mode : false
16/03/26 18:27:17 INFO mapreduce.Job: map 0% reduce 0%
16/03/26 18:27:41 INFO mapreduce.Job: map 1% reduce 0%
16/03/26 18:27:46 INFO mapreduce.Job: map 2% reduce 0%
16/03/26 18:27:47 INFO mapreduce.Job: map 3% reduce 0%
16/03/26 18:27:50 INFO mapreduce.Job: map 4% reduce 0%
16/03/26 18:27:53 INFO mapreduce.Job: map 5% reduce 0%
16/03/26 18:27:59 INFO mapreduce.Job: map 6% reduce 0%
16/03/26 18:28:03 INFO mapreduce.Job: map 7% reduce 0%
16/03/26 18:28:06 INFO mapreduce.Job: map 8% reduce 0%
16/03/26 18:28:08 INFO mapreduce.Job: map 9% reduce 0%
16/03/26 18:28:09 INFO mapreduce.Job: map 10% reduce 0%
16/03/26 18:28:15 INFO mapreduce.Job: map 11% reduce 0%
16/03/26 18:28:22 INFO mapreduce.Job: map 12% reduce 0%
16/03/26 18:28:25 INFO mapreduce.Job: map 13% reduce 0%
16/03/26 18:28:28 INFO mapreduce.Job: map 14% reduce 0%
16/03/26 18:28:31 INFO mapreduce.Job: map 15% reduce 0%
16/03/26 18:28:35 INFO mapreduce.Job: map 16% reduce 0%
```

Namenode Information - Google Chrome

54.89.115.183:50070/dfshealth.html#tab-overview

Hadoop Overview Datanodes Datanode Volume Failures Snapshot Startup Progress Utilities

Overview 'master:9000' (active)

Started:	Sun Mar 27 04:17:18 UTC 2016
Version:	2.7.2, rb165c4fe8a74265c792ca23f546c64604acf0e41
Compiled:	2016-01-26T00:08Z by jenkins from (detached from b185c4f)
Cluster ID:	CID-61bda967-c0ff-409f-a15b-79d1aedddde06
Block Pool ID:	BP-1715206373-172.31.8.164-1459039248147

Summary

Security is off.
SafeMode is off.
27 files and directories, 1493 blocks = 1520 total filesystem object(s).
Heap Memory used 142.87 MB of 456.5 MB Heap Memory. Max Heap Memory is 880 MB.
Non Heap Memory used 41.61 MB of 41.94 MB Committed Non Heap Memory. Max Non Heap Memory is 214 MB.

Configured Capacity:	883.99 GB
DFS Used:	166.8 GB (19.13%)
Non DFS Used:	45.99 GB
DFS Remaining:	651.19 GB (73.67%)
Block Pool Used:	166.8 GB (19.13%)
DataNodes usages% (Min/Median/Max/stdDev):	9.42% / 22.45% / 24.37% / 4.18%
Live Nodes	15 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)
Decommissioning Nodes	0

PERFORMANCE

- Measure the execution Times for 10Gb sort on 1 node and 100Gb sort on 16 nodes.
- Calculate Throughput for each scenarios and summarize into below table.

No. of Nodes	Data Size (MBs)	Throughput (MB/s)	Execution Time (sec)
1	10000	8.218	1216.756
16	100000	33.98	2943.24

- Throughput of 16 node 100Gb data is way more faster than 1 node 10Gb because 32 core is involve of all 16 nodes.

Questions And Answers

Master: Master is known as Namenode that manages the file system namespace functions like opening, closing and renaming files as well directories and determines the mapping blocks to Datanodes along with regulating access to files by clients.

Slave: Slave is known as Datanode that are responsible for serving read and write request from the system's client. It performs creation, deletion and replication upon instruction from the master

Unique-Port: each configuration files has its own services like hdfs-site.xml is responsible for namenode and datanode in master and slave. So if we use same port then it won't throw error but there may be problem to creating namenode or datanode or Namenodemanager. So for retrieve unique service we need to give unique port for shared configuration.

We can change the mappers and reducers from mapred-site.xml configuration file from adding two properties mentioned below:

```

<name>mapreduce.job.maps</name>
<value>2</value>

<name>mapreduce.job.reduces</name>
<value>2</value>
```

SPARK

Spark Installation on Amazon EC2

- Login to Amazon EC2 console.
- Launch an instance of type **c3.large** as spot instance.
- Login to instance using following command
 ssh -i “ec2-spark.pem” <your public DNS of instance>
- Following command for installing jdk on instance.

```
sudo apt-get update
sudo apt-get install openjdk-7-jdk
```

- Getting Spark latest version from website and unzip it at current location.

```
wget http://apache.arvixe.com/spark/spark-1.6.1/spark-1.6.1-bin-hadoop2.6.tgz
tar -xvzf spark-1.6.1-bin-hadoop2.6.tgz
```

- Install Scala into an instance using following command.

```
wget http://www.scala-lang.org/files/archive/scala-2.10.4.tgz
sudo mkdir /usr/local/src/scala
```

- Add Scala path into .bashrc file.

```
vi .bashrc
```

Add the following lines at the end of the file

```
# export SCALA_HOME=/usr/local/src/scala/scala-2.10.4
# export PATH=$SCALA_HOME/bin:$PATH

source .bashrc
```

- Rename spark-1.6.1-bin-hadoop2.6.tgz to spark (better usability)

```
mv spark-1.6.1-bin-hadoop2.6.tgz spark-1.6.1
```

- Now create Security key from Amazon Ec2 Console

- Goto right corner click <Your Name> tab and select “Security and Credentials”
- Choose Access Keys (Access Key ID and Secret Access Key)
- Download your credentials and save it.

- Run following command to export your keys.

```
export AWS_ACCESS_KEY_ID=<your Key_Id>
export AWS_SECRET_ACCESS_KEY=<your_secret_key>
```

- Now all basic setups are done.

Create AMI of this instance and save it for future use.

```
ubuntu@lp-172-31-21-27: ~          ubuntu@ip-172-31-21-27: ~ 162x42
[  ]                                     -- 2016-03-28 01:44:42 --
wget http://www.scala-lang.org/files/archive/scala-2.10.4.tgz
http://www.scala-lang.org/files/archive/scala-2.10.4.tgz
Resolving www.scala-lang.org (www.scala-lang.org)... 128.178.154.159
Connecting to www.scala-lang.org (www.scala-lang.org)|128.178.154.159|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 29937534 (29M) [application/x-gzip]
Saving to: 'scala-2.10.4.tgz'

36% [=====>] 10,976,537  3.53MB/s
```

Create Cluster of 1 master and 1 slave (Virtual 1 node for 10 GB)

- Commands to reach directory

```
cd spark-1.6.1/ec2/
```

- Cluster creating command is following

```
./spark-ec2 -k ec2-spark -i ec2-spark.pem -s 1 -t c3.large --spot-price=0.05 launch SPARK_TERA_10GB
```

- It will create a 1 master node and 1 slave node as following screenshot.

- LOGIN to cluster using following command (by default login into Mater Node.)

```
./spark-ec2 -k ec2-spark -i ec2-spark.pem login SPARK_TERA_10GB
```

- Now onwards all command needs to run on Master Node. (login default as root)
- Install gensort for generating an input file of 10GB.

```
wget http://www.ordinal.com/try.cgi/gensort-linux-1.5.tar.gz
```

```
tar -xvzf gensort-linux-1.5.tar.gz
```

- Check your disk space using following command and generate input file into it.

```
$ lsblk
```

- Now generate input file of 10GB using following command

```
cd 64/
```

```
./gensort -a 1000000000 /mnt/spark_tera_10gb.txt
```

- Modifications into one HDFS file named hdfs-sie.xml using following command.

```
cd ephemeral-hdfs/conf/  
vi hdfs-site.xml
```

Change replica form "3" to "1" and save it.

- Now copy all this file changes to slave using following command.

```
cd
```

```
./spark-ec2/copy-dir ephemeral-hdfs/conf/
```

- Now run the following command to stop SPARK and HDFS service and restart it.

```
cd spark/sbin/
./stop-all.sh (spark)
./stop-dfs.sh(ephemeral hdfs/bin/)
./hadoop namenode -format
./start-all.sh(spark)
./start-dfs.sh (ephemeral hdfs)
```

- Now PUT input file into HDFS file system.

```
ephemeral-hdfs/bin/hadoop fs -put /mnt/sprak_tera_10gb.txt /user/hadoop/input/spark_tera_10gb.txt
```

- Now Run Program of TeraSort written in Python into pyspark shell.

```
cd spark/bin
./pyspark (start pyspark shell)
```

- Copy and paste following command into shell. (One by one)

Program that we need to run into pyspark shell

```
sortedFile = sc.textFile("hdfs://PUBLIC DNS :9000/user/hadoop/input/spark\_tera\_10gb.txt")
sortedObj = sortedFile.flatMap(lambda line:line.split("\n")).map(lambda dicto:(str(dicto[:10]),str(dicto[10:])))
sortedObj.sortByKey().map(lambda (a,b) : a+b)
sortedObj.saveAsTextFile("hdfs://Public DNS :9000/user/hadoop/output_filnal")
```

- ❖ 1st line is read the input file from the given input path and store it as pysparkRDD object
- ❖ Then it calls flatMap in-built function of pysparkRDD class and separate each line by “\n” of input file and map(pysparkRDD) function convert 1st 10 bytes as key and after it all bytes as values and return it as (key, value) pair.
- ❖ Then sortByKey() function sort the all key pair as per keys and return as (key,pair). Then last map function again merge the key and value as string
- ❖ Now sorted object writes into output file mentioned in Path.

Create Cluster of 1 master and 16 slave (Virtual 16 node for 100 GB)

- ❖ REPEAT THE ABOVE ALL STEP EXCLUDING THE FOLLOWING STEP MENTIONED BELOW.
- ❖ JUST USE FOLLOWING STEPS INSTEAD OF THAT STEP ABOVE.
- CLUSTER CREATION COMMAND is following.

```
./spark-ec2 -k ec2-spark -i ec2-spark.pem -s 16 -t c3.large -m c3.4xlarge --spot-price=0.12 launch SPARK_TERA_100GB
```

Here command represents following descriptions

- s = no. of slaves
- m master node instance type
- k key-pair
- i .pem file

- spot-price =instance spot price
- t slave instance type

- Copy and paste following command into shell. (One by one)
Make spark program like **spark-tera.py**

```
sortedFile = sc.textFile("hdfs://PUBLIC DNS :9000/user/hadoop/input/spark\_tera\_100gb.txt")
sortedObj = sortedFile.flatMap(lambda line:line.split("\n")).map(lambda dicto:(str(dicto[:10]),str(dicto[10:])).sortByKey().map(lambda (a,b) : a+b)
sortedObj.saveAsTextFile("hdfs://Public DNS :9000/user/hadoop/output_filnal")
```

OUTPUT Validation

- Multiple output file for 10gb(75) and 100gb(745)
- Convert it first to UNIX to DOS
 - unix2dos part-0000
- Check all files using following command
 - ./valsort part-00000 (1st file)
 - ./valsort part-00075 (last file)
- Print output on console using following command
 - head -10 part-00000(1st 10 lines)
 - tail -10 part-00075 (last 10 lines)

NOTES

- I used c3.4xlarge as master node because it gives me 2 x 160 SSD as disk space.
- For 100Gb input size and 100GB output size so total is 200 Gb only for master node.

ISSUED FACED

- Cluster ssh : connection refused 22 port issue (Temporary)
- No space problem for 100 Gb sorting problem. Solve by taking c3.4xlarge as instance type of master node.

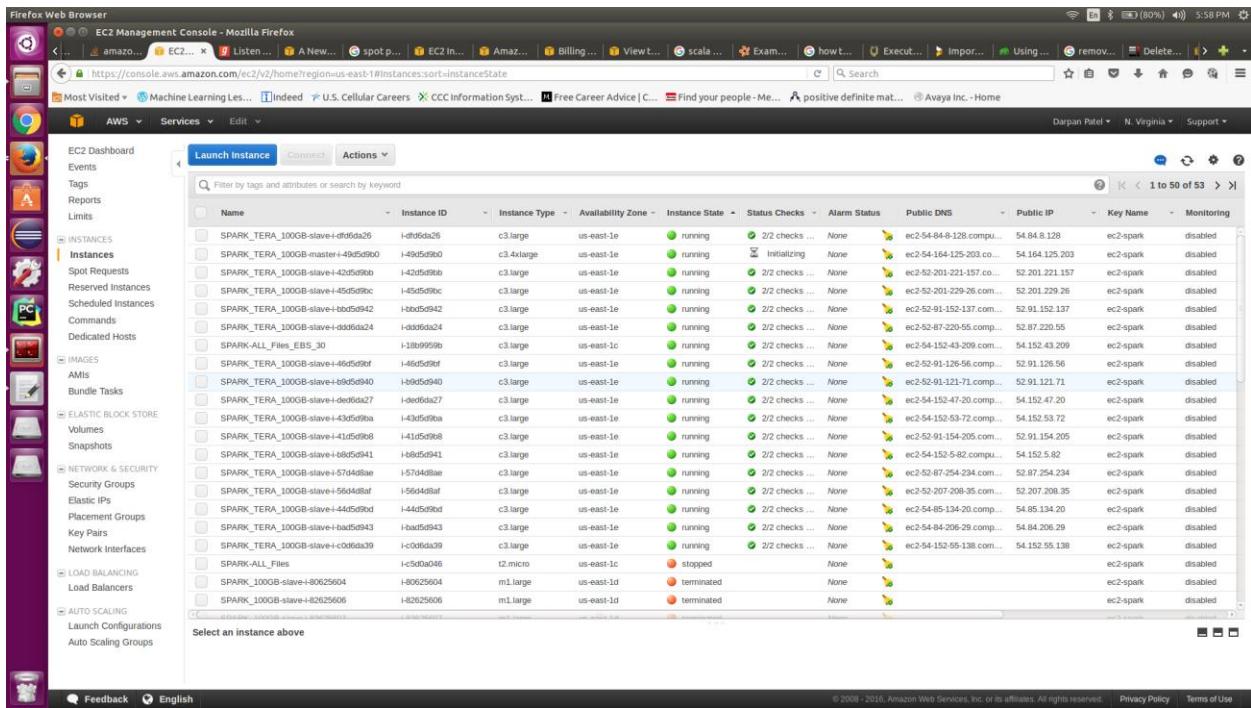
SCREENSHOT of SPARK CLUSTER(10Gb and 100Gb)

```
ubuntu@ip-172-31-60-93:~/spark-1.6.1/ec2$ ./spark-ec2 -k ec2-spark -i ec2-spark.pem --ebs-vol-size=30 --ebs-vol-num=7 -s 16 -t c3.large --spot-price=0.05 --region=us-east-1 launch SPARK_100GB
Setting up security groups...
Searching for existing cluster SPARK_100GB in region us-east-1...
Spark AMI: ami-5bb18832
Launching instances...
Requesting 16 slaves as spot instances with price $0.050
Waiting for spot instances to be granted...
0 of 16 slaves granted, waiting longer
0 of 16 slaves granted, waiting longer
0 of 16 slaves granted, waiting longer
```

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name	Monitoring	
SPARK_ALL_Files	i-5dd50d46	t2.micro	us-east-1c	stopped					ec2-spark	disabled	
SPARK_100GB-slave+80625604	i-80625604	m1.large	us-east-1d	pending	[!]	Initializing	None	ec2-52-90-133-215.com...	52.90.133.215	ec2-spark	disabled
SPARK_100GB-slave+82625606	i-82625606	m1.large	us-east-1d	pending	[!]	Initializing	None	ec2-54-152-90-7.compu...	54.152.90.7	ec2-spark	disabled
SPARK_100GB-slave+83625607	i-83625607	m1.large	us-east-1d	pending	[!]	Initializing	None	ec2-52-91-218-27.com...	52.91.218.27	ec2-spark	disabled
SPARK_100GB-slave+86625608	i-86625608	m1.large	us-east-1d	pending	[!]	Initializing	None	ec2-54-85-203-184.com...	54.85.203.184	ec2-spark	disabled
SPARK_100GB-slave+88625609	i-88625609	m1.large	us-east-1d	pending	[!]	Initializing	None	ec2-52-91-171-163.com...	52.91.171.163	ec2-spark	disabled
SPARK_100GB-slave+86625609a	i-86625609a	m1.large	us-east-1d	pending	[!]	Initializing	None	ec2-52-87-246-91.com...	52.87.246.91	ec2-spark	disabled
SPARK_100GB-slave+86256008	i-86256008	m1.large	us-east-1d	pending	[!]	Initializing	None	ec2-52-87-225-44.com...	52.87.225.44	ec2-spark	disabled
SPARK_100GB-slave+88625606c	i-88625606c	m1.large	us-east-1d	pending	[!]	Initializing	None	ec2-52-201-246-233.co...	52.201.246.233	ec2-spark	disabled
SPARK_100GB-slave+8962560d	i-8962560d	m1.large	us-east-1d	pending	[!]	Initializing	None	ec2-52-90-95-242.com...	52.90.95.242	ec2-spark	disabled
SPARK_100GB-slave+8a62560e	i-8a62560e	m1.large	us-east-1d	pending	[!]	Initializing	None	ec2-54-152-12-137.com...	54.152.12.137	ec2-spark	disabled
SPARK_100GB-slave+8862560f	i-8862560f	m1.large	us-east-1d	pending	[!]	Initializing	None	ec2-52-90-236-165.com...	52.90.236.165	ec2-spark	disabled
SPARK_100GB-slave+84625630	i-b4625630	m1.large	us-east-1d	pending	[!]	Initializing	None	ec2-52-91-49-108.com...	52.91.49.108	ec2-spark	disabled
SPARK_100GB-slave+846562631	i-846562631	m1.large	us-east-1d	pending	[!]	Initializing	None	ec2-52-90-232-155.com...	52.90.232.155	ec2-spark	disabled
SPARK_100GB-slave+86625632	i-86625632	m1.large	us-east-1d	pending	[!]	Initializing	None	ec2-52-90-158-150.com...	52.90.158.150	ec2-spark	disabled
SPARK_100GB-slave+87625633	i-b7625633	m1.large	us-east-1d	pending	[!]	Initializing	None	ec2-52-207-228-252.co...	52.207.228.252	ec2-spark	disabled
SPARK_100GB-slave+81625635	i-81625635	m1.large	us-east-1d	pending	[!]	Initializing	None	ec2-52-87-250-200.com...	52.87.250.200	ec2-spark	disabled
SPARK_100GB-master+0f61568b	i-0f61568b	m1.large	us-east-1d	pending	[!]	Initializing	None	ec2-54-164-162-84.com...	54.164.162.84	ec2-spark	disabled
SPARK_ALL_Files_EBS_30	i-1809950b	c3.large	us-east-1c	running	[✓/2 checks ...]		None	ec2-54-152-43-209.com...	54.152.43.209	ec2-spark	disabled
SPARK_TERA_10-slave+98cce21b	i-98cce21b	c3.large	us-east-1c	terminated			None			ec2-spark	disabled
SPARK_TERA_10-master+99cce21a	i-99cce21a	c3.large	us-east-1c	terminated			None			ec2-spark	disabled

```
ubuntu@ip-172-31-21-27:~/spark/ec2$ ./spark-ec2 -k hadoop.pem -s 1 -t c3.large --spot-price=0.025 --ebs-vol-size=30 --ebs-vol-num=1 launch spark_tsort
Warning: SSH connection error. (This could be temporary.)
Host: ec2-54-175-56-182.compute-1.amazonaws.com
SSH return code: 255
SSH output: ssh: connect to host ec2-54-175-56-182.compute-1.amazonaws.com port 22: Connection refused
...
.^CTraceback (most recent call last):
File "./spark_ec2.py", line 1528, in <module>
    main()
File "./spark_ec2.py", line 1520, in main
    real_main()
File "./spark_ec2.py", line 1519, in real_main
    File "./spark_ec2.py", line 1354, in real_main
        cluster_state="ssh-ready"
File "./spark_ec2.py", line 928, in wait_for_cluster_state
    time.sleep(5 * num_attempts) # seconds
KeyboardInterrupt
ubuntu@ip-172-31-21-27:~/spark/ec2$ ./spark-ec2 -k hadoop.pem -s 1 -t c3.large --spot-price=0.025 --ebs-vol-size=30 --ebs-vol-num=1 launch spark_tsort -resume
Searching for existing cluster spark_tsort in region us-east-1...
Found 1 master, 1 slave.
Waiting for cluster to enter 'ssh-ready' state.
Warning: SSH connection error. (This could be temporary.)
Host: ec2-54-175-56-182.compute-1.amazonaws.com
SSH return code: 255
SSH output: ssh: connect to host ec2-54-175-56-182.compute-1.amazonaws.com port 22: Connection refused
...
Warning: SSH connection error. (This could be temporary.)
Host: ec2-54-175-56-182.compute-1.amazonaws.com
SSH return code: 255
SSH output: ssh: connect to host ec2-54-175-56-182.compute-1.amazonaws.com port 22: Connection refused
...
Warning: SSH connection error. (This could be temporary.)
Host: ec2-54-175-56-182.compute-1.amazonaws.com
SSH return code: 255
SSH output: ssh: connect to host ec2-54-175-56-182.compute-1.amazonaws.com port 22: Connection refused
...

```

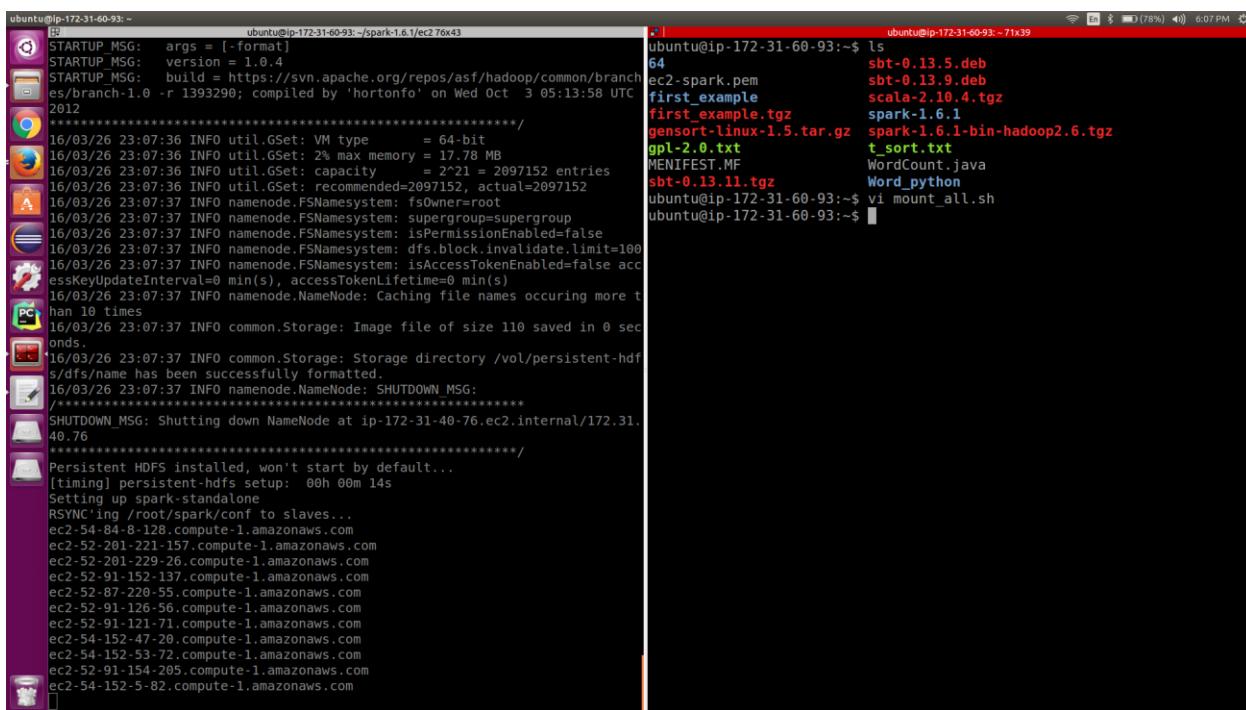


```

ubuntu@ip-172-31-60-93:~/spark-1.6.1/ec2
[1] RSYNC'ing /root/ephemeral-hdfs to slaves...
ec2-54-84-8-128.compute-1.amazonaws.com
Warning: Permanently added 'ec2-54-84-8-128.compute-1.amazonaws.com,172.31.47.35' (ECDSA) to the list of known hosts.
ec2-52-201-221-157.compute-1.amazonaws.com
Warning: Permanently added 'ec2-52-201-221-157.compute-1.amazonaws.com,172.31.34.170' (ECDSA) to the list of known hosts.
ec2-52-201-229-26.compute-1.amazonaws.com
Warning: Permanently added 'ec2-52-201-229-26.compute-1.amazonaws.com,172.31.43.239' (ECDSA) to the list of known hosts.
ec2-52-91-152-137.compute-1.amazonaws.com
Warning: Permanently added 'ec2-52-91-152-137.compute-1.amazonaws.com,172.31.37.46' (ECDSA) to the list of known hosts.
ec2-52-87-220-55.compute-1.amazonaws.com
Warning: Permanently added 'ec2-52-87-220-55.compute-1.amazonaws.com,172.31.33.201' (ECDSA) to the list of known hosts.
ec2-52-91-126-56.compute-1.amazonaws.com
Warning: Permanently added 'ec2-52-91-126-56.compute-1.amazonaws.com,172.31.42.205' (ECDSA) to the list of known hosts.
ec2-52-91-121-71.compute-1.amazonaws.com
Warning: Permanently added 'ec2-52-91-121-71.compute-1.amazonaws.com,172.31.42.254' (ECDSA) to the list of known hosts.
ec2-54-152-47-20.compute-1.amazonaws.com
Warning: Permanently added 'ec2-54-152-47-20.compute-1.amazonaws.com,172.31.41.233' (ECDSA) to the list of known hosts.
ec2-54-152-53-72.compute-1.amazonaws.com
Warning: Permanently added 'ec2-54-152-53-72.compute-1.amazonaws.com,172.31.37.80' (ECDSA) to the list of known hosts.
ec2-52-91-154-205.compute-1.amazonaws.com
Warning: Permanently added 'ec2-52-91-154-205.compute-1.amazonaws.com,172.31.35.245' (ECDSA) to the list of known hosts.
ec2-54-152-5-82.compute-1.amazonaws.com
Warning: Permanently added 'ec2-54-152-5-82.compute-1.amazonaws.com,172.31.45.196' (ECDSA) to the list of known hosts.
ec2-52-87-254-234.compute-1.amazonaws.com
Warning: Permanently added 'ec2-52-87-254-234.compute-1.amazonaws.com,172.31.39.32' (ECDSA) to the list of known hosts.
ec2-52-207-208-35.compute-1.amazonaws.com
Warning: Permanently added 'ec2-52-207-208-35.compute-1.amazonaws.com,172.31.39.158' (ECDSA) to the list of known hosts.
ec2-54-85-134-20.compute-1.amazonaws.com
Warning: Permanently added 'ec2-54-85-134-20.compute-1.amazonaws.com,172.31.46.201' (ECDSA) to the list of known hosts.
ec2-54-84-206-29.compute-1.amazonaws.com
Warning: Permanently added 'ec2-54-84-206-29.compute-1.amazonaws.com,172.31.32.89' (ECDSA) to the list of known hosts.
ec2-54-152-55-138.compute-1.amazonaws.com
Warning: Permanently added 'ec2-54-152-55-138.compute-1.amazonaws.com,172.31.45.212' (ECDSA) to the list of known hosts.
[timing] ephemeral-hdfs init: 00h 00m 18s
Initializing persistent-hdfs
--2016-03-26 23:05:16- http://s3.amazonaws.com/spark-related-packages/hadoop-1.0.4.tar.gz
Resolving s3.amazonaws.com (s3.amazonaws.com)... 54.231.8.184
Connecting to s3.amazonaws.com (s3.amazonaws.com)|54.231.8.184|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 62793050 (60M) [application/x-gzip]
Saving to: 'hadoop-1.0.4.tar.gz'

65% [=====] 40,975,834 64.4MB/s

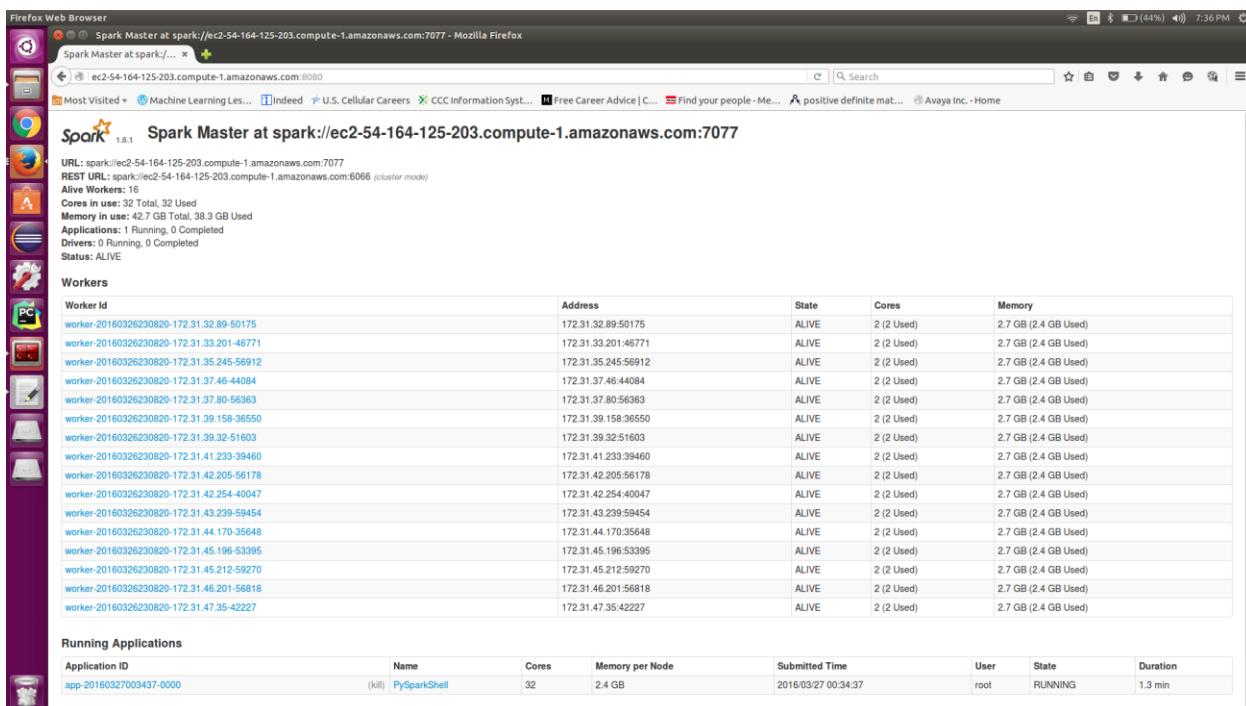
```



```

ubuntu@ip-172-31-60-93:~$ ./spark-1.6.1/ec2.7x43
STARTUP_MSG: args = [-format]
STARTUP_MSG: version = 1.0.4
STARTUP_MSG: build = https://svn.apache.org/repos/asf/hadoop/common/branches/branch-1.0 -r 1393290; compiled by 'hortonfo' on Wed Oct 3 05:13:58 UTC 2012
*****
16/03/26 23:07:36 INFO util.GSet: VM type       = 64-bit
16/03/26 23:07:36 INFO util.GSet: 2% max memory = 17.78 MB
16/03/26 23:07:36 INFO util.GSet: capacity      = 2^21 = 2097152 entries
16/03/26 23:07:36 INFO util.GSet: recommended=2097152, actual=2097152
16/03/26 23:07:37 INFO namenode.FSNamesystem: fsowner=root
16/03/26 23:07:37 INFO namenode.FSNamesystem: supergroups=supergroup
16/03/26 23:07:37 INFO namenode.FSNamesystem: isPermissionEnabled=false
16/03/26 23:07:37 INFO namenode.FSNamesystem: dfs.block.invalidate.limit=100
16/03/26 23:07:37 INFO namenode.FSNamesystem: isAccessTokenEnabled=false accessKeyUpdateInterval=0 min(s), accessTokenLifetime=0 min(s)
16/03/26 23:07:37 INFO namenode.NameNode: Caching file names occurring more than 10 times
16/03/26 23:07:37 INFO common.Storage: Image file of size 110 saved in 0 seconds.
16/03/26 23:07:37 INFO common.Storage: Storage directory /vol/persistent-hdfs/dfs/name has been successfully formatted.
16/03/26 23:07:37 INFO namenode.NameNode: SHUTDOWN MSG:
*****
SHUTDOWN_MSG: Shutting down NameNode at ip-172-31-40-76.ec2.internal/172.31.40.76
40.76
*****
Persistent HDFS installed, won't start by default...
[timing] persistent-hdfs setup: 00h 00m 14s
Setting up spark-standalone
RSYNC'ing /root/spark/conf to slaves...
ec2-54-84-8-128.compute-1.amazonaws.com
ec2-52-201-221-157.compute-1.amazonaws.com
ec2-52-201-229-26.compute-1.amazonaws.com
ec2-52-91-152-137.compute-1.amazonaws.com
ec2-52-87-220-55.compute-1.amazonaws.com
ec2-52-91-126-56.compute-1.amazonaws.com
ec2-52-91-121-71.compute-1.amazonaws.com
ec2-54-152-47-20.compute-1.amazonaws.com
ec2-54-152-53-72.compute-1.amazonaws.com
ec2-52-91-154-205.compute-1.amazonaws.com
ec2-54-152-5-82.compute-1.amazonaws.com
[...]

```



Spark Master at spark://ec2-54-164-125-203.compute-1.amazonaws.com:7077 - Mozilla Firefox

Spark Master at spark://... ec2-54-164-125-203.compute-1.amazonaws.com:8080

Most Visited [Machine Learning Les...](#) [Indeed](#) [U.S. Cellular Careers](#) [CCC Information Syst...](#) [Free Career Advice | C...](#) [Find your people - Me...](#) [A positive definite mat...](#) [Avaya Inc. - Home](#)

Spark 1.6.1 Spark Master at spark://ec2-54-164-125-203.compute-1.amazonaws.com:7077

URL: spark://ec2-54-164-125-203.compute-1.amazonaws.com:7077
 REST URL: spark://ec2-54-164-125-203.compute-1.amazonaws.com:6066 (cluster mode)
 Active Workers: 16
 Cores in use: 32 Total, 32 Used
 Memory in use: 42.7 GB Total, 38.3 GB Used
 Applications: 1 Running, 0 Completed
 Drivers: 0 Running, 0 Completed
 Status: ALIVE

Workers

Worker Id	Address	State	Cores	Memory
worker-20160326230820-172.31.32.89-50175	172.31.32.89.50175	ALIVE	2 (2 Used)	2.7 GB (2.4 GB Used)
worker-20160326230820-172.31.33.201-46771	172.31.33.201.46771	ALIVE	2 (2 Used)	2.7 GB (2.4 GB Used)
worker-20160326230820-172.31.35.245-56912	172.31.35.245.56912	ALIVE	2 (2 Used)	2.7 GB (2.4 GB Used)
worker-20160326230820-172.31.37.46-44084	172.31.37.46.44084	ALIVE	2 (2 Used)	2.7 GB (2.4 GB Used)
worker-20160326230820-172.31.37.80-56363	172.31.37.80.56363	ALIVE	2 (2 Used)	2.7 GB (2.4 GB Used)
worker-20160326230820-172.31.39.158-36550	172.31.39.158.36550	ALIVE	2 (2 Used)	2.7 GB (2.4 GB Used)
worker-20160326230820-172.31.39.2-51603	172.31.39.2.51603	ALIVE	2 (2 Used)	2.7 GB (2.4 GB Used)
worker-20160326230820-172.31.41.233-39460	172.31.41.233.39460	ALIVE	2 (2 Used)	2.7 GB (2.4 GB Used)
worker-20160326230820-172.31.42.205-56178	172.31.42.205.56178	ALIVE	2 (2 Used)	2.7 GB (2.4 GB Used)
worker-20160326230820-172.31.42.254-40047	172.31.42.254.40047	ALIVE	2 (2 Used)	2.7 GB (2.4 GB Used)
worker-20160326230820-172.31.43.239-59454	172.31.43.239.59454	ALIVE	2 (2 Used)	2.7 GB (2.4 GB Used)
worker-20160326230820-172.31.44.170-35648	172.31.44.170.35648	ALIVE	2 (2 Used)	2.7 GB (2.4 GB Used)
worker-20160326230820-172.31.45.196-53395	172.31.45.196.53395	ALIVE	2 (2 Used)	2.7 GB (2.4 GB Used)
worker-20160326230820-172.31.45.212-59270	172.31.45.212.59270	ALIVE	2 (2 Used)	2.7 GB (2.4 GB Used)
worker-20160326230820-172.31.46.201-56818	172.31.46.201.56818	ALIVE	2 (2 Used)	2.7 GB (2.4 GB Used)
worker-20160326230820-172.31.47.35-42227	172.31.47.35.42227	ALIVE	2 (2 Used)	2.7 GB (2.4 GB Used)

Running Applications

Application ID	Name	Cores	Memory per Node	Submitted Time	User	State	Duration
app-20160327003437-0000	(kill) PySparkShell	32	2.4 GB	2016/03/27 00:34:37	root	RUNNING	1.3 min

Four terminal windows showing log files for task set managers and disk usage:

- Terminal 1:** Log of TaskSetManager tasks from 16/03/27 01:16:00 to 16/03/27 01:16:02.
- Terminal 2:** Disk usage (df -h) on a system with partitions /dev/xvda1, /tmpfs, /dev/shm, /dev/xvdb, /dev/xvdf, and /dev/mnt2.
- Terminal 3:** Log of TaskSetManager tasks from 16/03/27 01:16:02 to 16/03/27 01:16:05.
- Terminal 4:** Disk usage (df -h) on a system with partitions /dev/xvda1, /tmpfs, /dev/shm, /dev/xvdb, /dev/xvdf, and /dev/mnt2.

```
root@ip-172-31-40-76 ~]$ ./spark-ec2/copy-dir ephemeral-hdfs/conf/
RSYNC'ing /root/ephemeral-hdfs/conf to slaves...
ec2-54-84-8-128.compute-1.amazonaws.com
ec2-52-201-221-157.compute-1.amazonaws.com
ec2-52-201-229-26.compute-1.amazonaws.com
ec2-52-91-152-137.compute-1.amazonaws.com
ec2-52-87-220-55.compute-1.amazonaws.com
ec2-52-91-126-56.compute-1.amazonaws.com
ec2-52-91-121-71.compute-1.amazonaws.com
ec2-54-152-47-20.compute-1.amazonaws.com
ec2-54-152-53-72.compute-1.amazonaws.com
ec2-52-91-154-205.compute-1.amazonaws.com
ec2-54-152-5-82.compute-1.amazonaws.com
ec2-52-87-254-234.compute-1.amazonaws.com
ec2-52-207-208-35.compute-1.amazonaws.com
ec2-54-85-134-20.compute-1.amazonaws.com
ec2-54-84-206-29.compute-1.amazonaws.com
ec2-54-152-55-138.compute-1.amazonaws.com
root@ip-172-31-40-76 ~]$
```

```

dbp@dbp-patel:~/CS-553/Assignment/Ass-2$ ./pspark
Python 2.7.10 (default, Dec  8 2015, 18:25:31)
[GCC 4.8.3 20140911 (Red Hat 4.8.3-9)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
16/03/27 04:07:25 INFO spark.SparkContext: Running Spark version 1.6.1
16/03/27 04:07:25 WARN spark.SparkConf:
SPARK_WORKER_INSTANCES was detected (set to '1').
This is deprecated in Spark 1.0+.

Please instead use:
- ./spark-submit with --num-executors to specify the number of executors
- Or set SPARK_EXECUTOR_INSTANCES
- spark.executor.instances to configure the number of instances in the spark config.

16/03/27 04:07:25 INFO spark.SecurityManager: Changing view acls to: root
16/03/27 04:07:25 INFO spark.SecurityManager: Changing modify acls to: root
16/03/27 04:07:25 INFO spark.SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Set[root]; users with modify permissions: Set[root]
16/03/27 04:07:25 INFO util.Utils: Successfully started service 'sparkDriver' on port 54614.
16/03/27 04:07:26 INFO slf4j.Slf4jLogger: Slf4jLogger started
16/03/27 04:07:26 INFO Remoting: Starting remoting
16/03/27 04:07:26 INFO Remoting: Remoting started; listening on addresses :[akka.tcp://sparkDriverActorSystem@172.31.40.76:60149]
16/03/27 04:07:26 INFO util.Utils: Successfully started service 'sparkDriverActorSystem' on port 60149.
16/03/27 04:07:26 INFO spark.SparkEnv: Registering MapOutputTracker
16/03/27 04:07:26 INFO spark.SparkEnv: Registering BlockManagerMaster
16/03/27 04:07:26 INFO storage.DiskBlockManager: Created local directory at /mnt/spark/blockmgr-a3fd33b3-39f8-4cc3-b564-3la9d3640dfa
16/03/27 04:07:26 INFO storage.DiskBlockManager: Created local directory at /mnt2/spark/blockmgr-23af06a9-2e21-4957-be2a-ba6f5661f706
16/03/27 04:07:26 INFO storage.MemoryStore: MemoryStore started with capacity 511.5 MB
16/03/27 04:07:26 INFO spark.SparkEnv: Registering OutputCommitCoordinator
16/03/27 04:07:26 INFO server.Server: jetty-8.y.z-SNAPSHOT
16/03/27 04:07:26 INFO server.AbstractConnector: Started SelectChannelConnector@0.0 .0.0:4040
16/03/27 04:07:26 INFO util.Utils: Successfully started service 'SparkUI' on port 4040.
16/03/27 04:07:26 INFO ui.SparkUI: Started SparkUI at http://ec2-54-164-125-203.compute-1.amazonaws.com:4040

```

PROGRAM in PYSPARK

```

dbp@dbp-patel:~/CS-553/Assignment/Ass-2$ ./spark
16/03/27 00:34:41 INFO storage.BlockManagerMasterEndpoint: Registering block manager ip-172-31-42-254.ec2.internal:54552 with 1539.0 MB RAM, BlockManagerId(6, ip-172-31-42-254.ec2.internal, 54552)
16/03/27 00:34:41 INFO storage.BlockManagerMasterEndpoint: Registering block manager ip-172-31-37-46.ec2.internal:40629 with 1539.0 MB RAM, BlockManagerId(13, ip-172-31-37-46.ec2.internal, 40629)
16/03/27 00:34:41 INFO storage.BlockManagerMasterEndpoint: Registering block manager ip-172-31-41-233.ec2.internal:47616 with 1539.0 MB RAM, BlockManagerId(9, ip-172-31-41-233.ec2.internal, 47616)
16/03/27 00:34:41 INFO cluster.SparkDeploySchedulerBackend: Registered executor NettyRpcEndpointRef(null) (ip-172-31-47-35.ec2.internal:51131) with ID 0
16/03/27 00:34:41 INFO storage.BlockManagerMasterEndpoint: Registering block manager ip-172-31-44-170.ec2.internal:44951 with 1539.0 MB RAM, BlockManagerId(12, ip-172-31-44-170.ec2.internal, 44951)
16/03/27 00:34:41 INFO cluster.SparkDeploySchedulerBackend: Registered executor NettyRpcEndpointRef(null) (ip-172-31-46-201.ec2.internal:57920) with ID 7
16/03/27 00:34:41 INFO storage.BlockManagerMasterEndpoint: Registering block manager ip-172-31-47-35.ec2.internal:56832 with 1539.0 MB RAM, BlockManagerId(0, ip-172-31-47-35.ec2.internal, 56832)
16/03/27 00:34:41 INFO cluster.SparkDeploySchedulerBackend: Registered executor NettyRpcEndpointRef(null) (ip-172-31-33-201.ec2.internal:48529) with ID 1
16/03/27 00:34:42 INFO storage.BlockManagerMasterEndpoint: Registering block manager ip-172-31-46-201.ec2.internal:36679 with 1539.0 MB RAM, BlockManagerId(7, ip-172-31-46-201.ec2.internal, 36679)
16/03/27 00:34:42 INFO storage.BlockManagerMasterEndpoint: Registering block >>> distFile = sc.textFile("hdfs://ec2-54-164-125-203.compute-1.amazonaws.com:9000/user/hadoop/input_100gb.txt")
16/03/27 00:37:55 INFO storage.MemoryStore: Block broadcast_0 stored as values in memory (estimated size 46.3 KB, free 46.3 KB)
16/03/27 00:37:55 INFO storage.MemoryStore: Block broadcast_0 piece0 stored as bytes in memory (estimated size 4.4 KB, free 50.7 KB)
16/03/27 00:37:55 INFO storage.BlockManagerInfo: Added broadcast_0_piece0 in memory on 172.31.40.76:50699 (size: 4.4 KB, free: 511.5 MB)
16/03/27 00:37:55 INFO spark.SparkContext: Created broadcast 0 from textFile at NativeMethodAccessorImpl.java:-2
>>>

```

```
dbp@dbp-patel:~/CS-553/Assignment/Ass-2
* | dbp@dbp-patel:~/CS-553/Assignment/Ass-2:153x43
  tor NettyRpcEndpointRef(null) (ip-172-31-44-170.ec2.internal:59016) with ID
12
16/03/27 00:34:41 INFO storage.BlockManagerMasterEndpoint: Registering block
manager ip-172-31-42-254.ec2.internal:54552 with 1539.0 MB RAM, BlockManage
rId(6, ip-172-31-42-254.ec2.internal, 54552)
16/03/27 00:34:41 INFO storage.BlockManagerMasterEndpoint: Registering block
manager ip-172-31-37-46.ec2.internal:40629 with 1539.0 MB RAM, BlockManager
Id(13, ip-172-31-37-46.ec2.internal, 40629)
16/03/27 00:34:41 INFO storage.BlockManagerMasterEndpoint: Registering block
manager ip-172-31-41-233.ec2.internal:47616 with 1539.0 MB RAM, BlockManage
rId(9, ip-172-31-41-233.ec2.internal, 47616)
16/03/27 00:34:41 INFO cluster.SparkDeploySchedulerBackend: Registered execu
tor NettyRpcEndpointRef(null) (ip-172-31-47-35.ec2.internal:51131) with ID 0
16/03/27 00:34:41 INFO storage.BlockManagerMasterEndpoint: Registering block
manager ip-172-31-44-170.ec2.internal:44951 with 1539.0 MB RAM, BlockManage
rId(12, ip-172-31-44-170.ec2.internal, 44951)
16/03/27 00:34:41 INFO cluster.SparkDeploySchedulerBackend: Registered execu
tor NettyRpcEndpointRef(null) (ip-172-31-46-201.ec2.internal:57928) with ID
7
16/03/27 00:34:41 INFO storage.BlockManagerMasterEndpoint: Registering block
manager ip-172-31-47-35.ec2.internal:56832 with 1539.0 MB RAM, BlockManage
rId(0, ip-172-31-47-35.ec2.internal, 56832)
16/03/27 00:34:41 INFO cluster.SparkDeploySchedulerBackend: Registered execu
tor NettyRpcEndpointRef(null) (ip-172-31-33-201.ec2.internal:48529) with ID
1
16/03/27 00:34:42 INFO storage.BlockManagerMasterEndpoint: Registering block
>>> counts1 = distFile.flatMap(lambda line:line.split("\n")).map(lambda dicto:(str(dicto[:10]),str(dicto[10:])).sortByKey().map(lambda (a,b):a+b)
```

```
dbp@dbp-patel:~/CS-553/Assignment/Ass-2
* | dbp@dbp-patel:~/CS-553/Assignment/Ass-2:153x43
  ) INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on ip-172-31-39-32.ec2.internal:47364 (size: 5.6 KB, free: 1539.0 MB)
16/03/27 00:41:49 INFO storage.BlockManagerInfo: Added broadcast_2_piece0 in memory on ip-172-31-41-233.ec2.internal:47616 (size: 3.9 KB, free: 1539.0 MB
)
16/03/27 00:41:49 INFO storage.BlockManagerInfo: Added broadcast_2_piece0 in memory on ip-172-31-37-46.ec2.internal:40629 (size: 3.9 KB, free: 1539.0 MB)
16/03/27 00:41:49 INFO storage.BlockManagerInfo: Added broadcast_2_piece0 in memory on ip-172-31-46-201.ec2.internal:36679 (size: 3.9 KB, free: 1539.0 MB
)
16/03/27 00:41:49 INFO storage.BlockManagerInfo: Added broadcast_2_piece0 in memory on ip-172-31-47-35.ec2.internal:56832 (size: 3.9 KB, free: 1539.0 MB)
16/03/27 00:41:49 INFO storage.BlockManagerInfo: Added broadcast_2_piece0 in memory on ip-172-31-44-170.ec2.internal:44951 (size: 3.9 KB, free: 1539.0 MB
)
16/03/27 00:41:49 INFO storage.BlockManagerInfo: Added broadcast_2_piece0 in memory on ip-172-31-42-205.ec2.internal:57822 (size: 3.9 KB, free: 1539.0 MB
)
16/03/27 00:41:49 INFO storage.BlockManagerInfo: Added broadcast_2_piece0 in memory on ip-172-31-45-212.ec2.internal:44843 (size: 3.9 KB, free: 1539.0 MB
)
16/03/27 00:41:49 INFO storage.BlockManagerInfo: Added broadcast_2_piece0 in memory on ip-172-31-39-158.ec2.internal:35288 (size: 3.9 KB, free: 1539.0 MB
)
16/03/27 00:41:49 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on ip-172-31-42-254.ec2.internal:54552 (size: 5.6 KB, free: 1539.0 MB
)
16/03/27 00:41:49 INFO storage.BlockManagerInfo: Added broadcast_2_piece0 in memory on ip-172-31-32-89.ec2.internal:50322 (size: 3.9 KB, free: 1539.0 MB)
16/03/27 00:41:49 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on ip-172-31-37-80.ec2.internal:40953 (size: 5.6 KB, free: 1539.0 MB
)
16/03/27 00:41:49 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on ip-172-31-45-196.ec2.internal:60867 (size: 5.6 KB, free: 1539.0 MB
)
16/03/27 00:41:49 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on ip-172-31-43-239.ec2.internal:51093 (size: 5.6 KB, free: 1539.0 MB
)
16/03/27 00:41:50 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on ip-172-31-41-233.ec2.internal:47616 (size: 5.6 KB, free: 1539.0 MB
)
16/03/27 00:41:50 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on ip-172-31-46-201.ec2.internal:36679 (size: 5.6 KB, free: 1539.0 MB
)
16/03/27 00:41:50 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on ip-172-31-42-205.ec2.internal:57822 (size: 5.6 KB, free: 1539.0 MB
)
16/03/27 00:41:50 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on ip-172-31-37-46.ec2.internal:40629 (size: 5.6 KB, free: 1539.0 MB
)
16/03/27 00:41:50 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on ip-172-31-47-35.ec2.internal:56832 (size: 5.6 KB, free: 1539.0 MB
)
16/03/27 00:41:50 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on ip-172-31-39-158.ec2.internal:35288 (size: 5.6 KB, free: 1539.0 MB
)
16/03/27 00:41:50 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on ip-172-31-31-239.ec2.internal:51093 (size: 5.6 KB, free: 1539.0 MB
)
16/03/27 00:41:50 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on ip-172-31-33-201.ec2.internal:53288 (size: 5.6 KB, free: 1539.0 MB
)
16/03/27 00:41:50 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on ip-172-31-35-245.ec2.internal:48666 (size: 5.6 KB, free: 1539.0 MB
)
16/03/27 00:41:50 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on ip-172-31-45-212.ec2.internal:44843 (size: 5.6 KB, free: 1539.0 MB
)
16/03/27 00:41:50 INFO storage.BlockManagerInfo: Added broadcast_1_piece0 in memory on ip-172-31-32-89.ec2.internal:50322 (size: 5.6 KB, free: 1539.0 MB)
```

Spark Jobs (1)

Total Uptime: 16 min
Scheduling Mode: FIFO
Active Jobs: 1
Completed Jobs: 1

Active Jobs (1)

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
1	sortByKey at <stdin>:1	2016/03/26 04:44:56	4.0 min	0/1	64/75

Completed Jobs (1)

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
0	sortByKey at <stdin>:1	2016/03/26 04:40:48	4.1 min	1/1	75/75

Details for Job 2

Status: RUNNING
Active Stages: 1
Pending Stages: 1

DAG Visualization

```

graph TD
    A["Stage 2  
textFile"] --> B["Stage 3  
partitionBy"]
    B --> C["mapPartitions"]
    C --> D["map"]
    D --> E["saveAsTextFile"]
  
```

Active Stages (1)

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
2	sortByKey at <stdin>:1	+details (kill)	2016/03/27 04:32:47	1.9 min	96/745	12.0 GB		5.8 GB

Pending Stages (1)

Stage Id	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
3	saveAsTextFile at NativeMethodAccessorImpl.java:2	+details Unknown	Unknown	0/745				

The screenshot shows the PySparkShell application UI running in a Firefox browser. The main window displays a table of completed jobs. The table has columns for Job Id, Description, Submitted, Duration, Stages: Succeeded/Total, and Tasks (for all stages): Succeeded/Total. There are three rows in the table:

Job Id	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
2	saveAsTextFile at NativeMethodAccesso...:-2	2016/03/27 01:00:33	20 min	2/2	1490/1490
1	sortByKey at <stdin>:1	2016/03/27 00:49:10	8.0 min	1/1	745/745
0	sortByKey at <stdin>:1	2016/03/27 00:41:48	7.4 min	1/1	745/745

```
root@ip-172-31-40-76:~$ unix2dos ../../part-00744
unix2dos: converting file ../../part-00744 to DOS format ...
root@ip-172-31-40-76:~$ 
root@ip-172-31-40-76:~$ 
root@ip-172-31-40-76:~$ 
root@ip-172-31-40-76:~$ file ../../part-00744
../../part-00744: ASCII text, with CRLF line terminators
root@ip-172-31-40-76:~$ 
root@ip-172-31-40-76:~$ 
root@ip-172-31-40-76:~$ 
root@ip-172-31-40-76:~$ 
root@ip-172-31-40-76:~$ ./valsrt ../../part-00000
Records: 1703516
Checksum: d00172f4a600f
Duplicate keys: 0
SUCCESS - all records are in order
root@ip-172-31-40-76:~$ 
root@ip-172-31-40-76:~$ ./valsrt ../../part-00744
Records: 1155848
Checksum: 8d049ac370508
Duplicate keys: 0
SUCCESS - all records are in order
root@ip-172-31-40-76:~$ 
root@ip-172-31-40-76:~$ 
root@ip-172-31-40-76:~$ 
root@ip-172-31-40-76:~$ 
```

PERFORMANCE

- Measure the execution Times for 10Gb sort on 1 node and 100Gb sort on 16 nodes.
- Calculate Throughput for each scenarios and summarize into below table.
- All execution time is given by spark UI is in minutes but I convert it into seconds.

No. of Nodes	Data Size (MBs)	Throughput (MB/s)	Execution Time (sec)
1	10000	12.077	828
16	100000	59.95	1668

- Throughput of 16 node 100Gb data is way more faster than 1 node 10Gb because 32 core is involve of all 16 nodes.

PERFORMANCE COMPARISON OF ALL PLATFORM

- Data Size : 10 GB
- No. of Threads / No. of nodes : 1

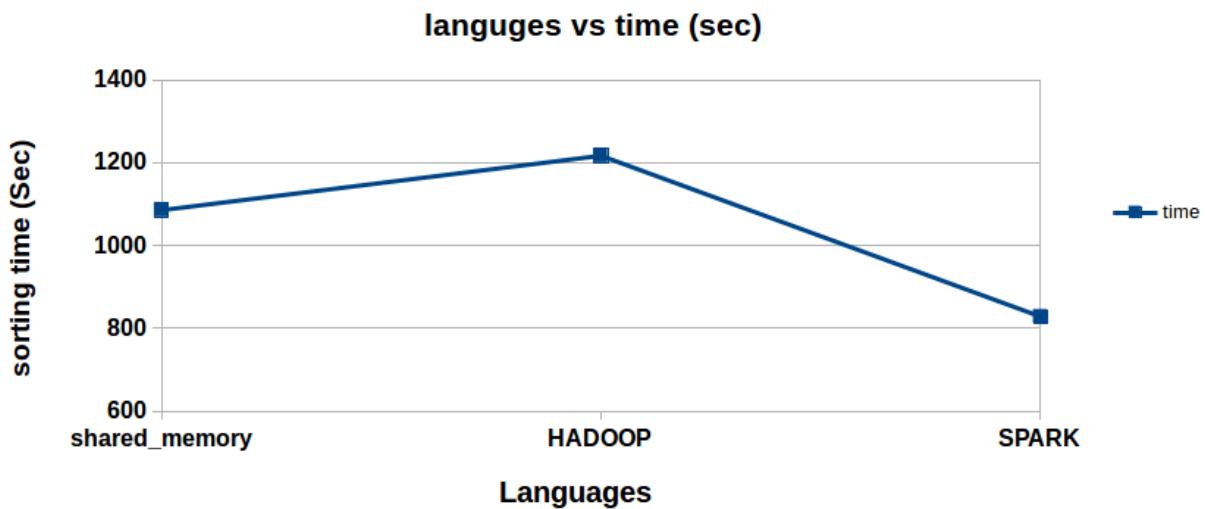
Platform	Throughput (MB/s)	Execution Time (sec)	Speed Up(time)
Shared Memory	9.215	1085.131	1.31
Hadoop	8.218	1216.756	1.469
Spark	12.074	828.23	1

No. of Nodes : 16

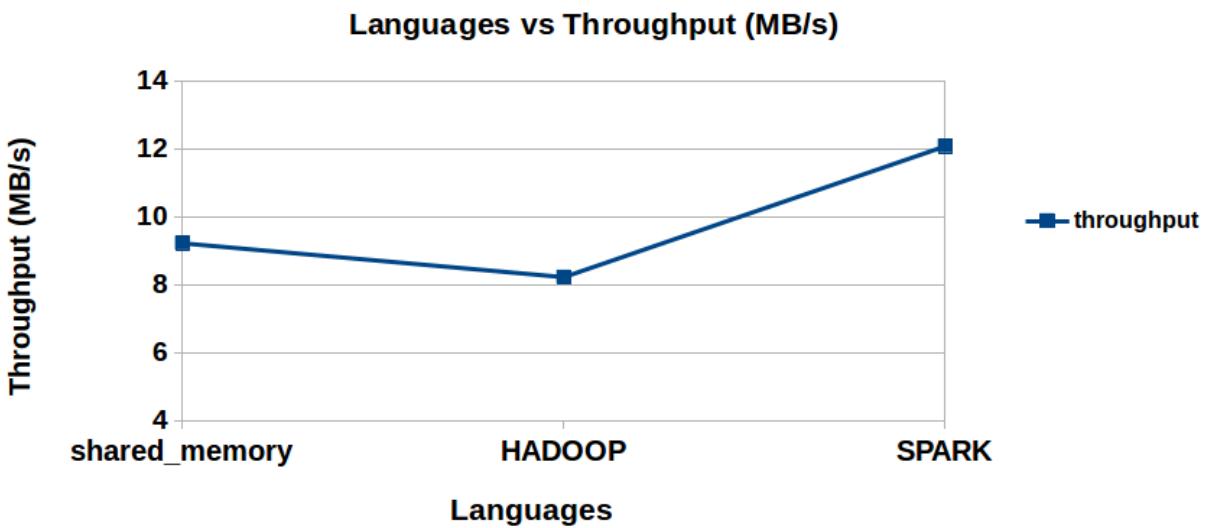
Platform	Data Size (MBs)	Throughput (MB/s)	Execution Time (sec)
Hadoop	100000	33.98	2943.24
Spark	100000	59.95	1668

- All Execution time is in Seconds. Speed is with Respect to Hadoop (slower).
- Spark is giving better output then all other platform for 10Gb data size of sorting.
- Spark has highest speed Up.
- Shared Memory (Python) is faster than Hadoop for every scenario.
- Spark may or may be best in each and every scenario than Shared Memory.
- Execution time may be different for all scenario.

Sort Time comparision between Platform

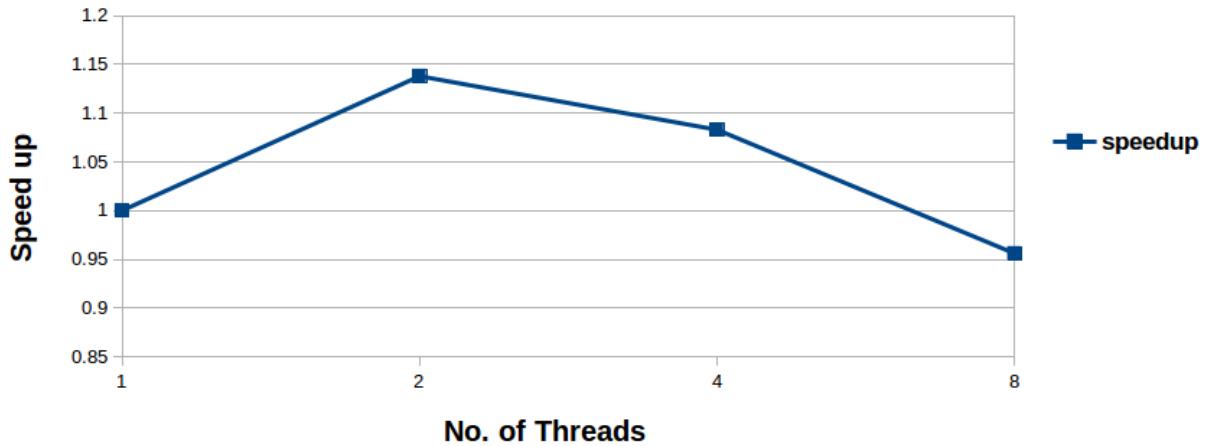


Throughput Comparision of Platforms



Speed up

No. of threads vs Speed up



REFERENCES

- <http://spark.apache.org/docs/latest/ec2-scripts.html>
- <http://wpcertification.blogspot.com/2014/06/wordcount-program-written-using-spark.html>
- <http://spark.apache.org/examples.html>
- <https://github.com/apache/spark/blob/master/examples/src/main/python/wordcount.py>
- <https://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html>
- http://markmiyashita.com/interviews/problems/merge_sort/
- http://www.codecodex.com/wiki/Merge_sort#Python
- <http://www.michael-noll.com/blog/2011/04/09/benchmarking-and-stress-testing-an-hadoop-cluster-with-terasort-testdfsio-nnbench-mrbench/>
- <https://support.pivotal.io/hc/en-us/articles/200927666-Running-TeraSort-MapReduce-Benchmark>
- <https://hadoop.apache.org/docs/r2.7.1/api/org/apache/hadoop/examples/terasort/package-summary.html>
- <http://www.ordinal.com/gensort.html>