Cloud Computing Report

Vasudev Gawde

System configuration

AWS instances
Type = c3.large
Memory = 3.75GB
EBS = 400GB
Spot instances

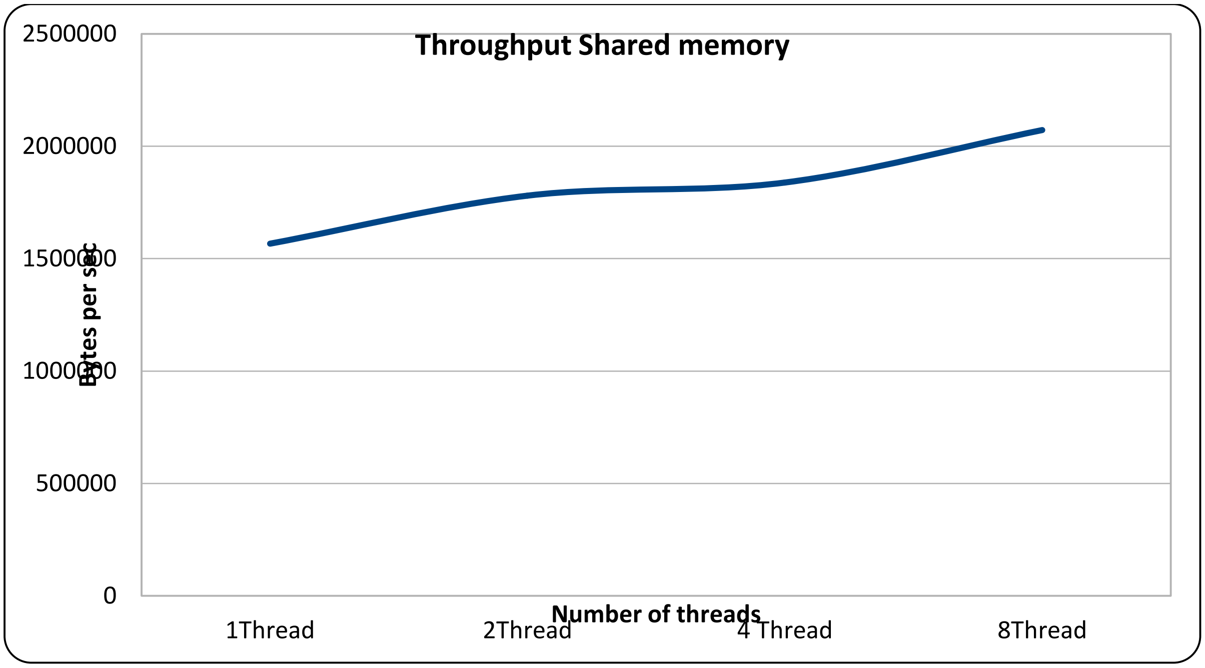
Shared Memory Sort

System installation and explanation refer README.txt file.

Reading are taken after running the experiment for 1,2,4,8 threads.

Number of threads	1	2	4	8
Bytes per second	1566819.8	1780826.55	1838461.53	2071425.7

As the number of threads increases, throughput of the system increases



Hadoop

System installation and explanation refer README.txt file.
Single node

Conf/slaves –

Contains the DNS of all the slaves machines on which the job will run
[Please refer below for the sample slaves file]

Conf/core-site.xml
Used to define temp directory location for hadoop
Used to define the hadoop default name . fs.default.name
[Please refer below for the sample file]

conf/core-hdfs.xml
Used to define the replication and permission values

[Please refer below for the sample file]
conf/mapred-site.xml

Used to define the jobtracker info of master
[Please refer below for the sample file]

1) What is a master node
Performs the process mangement of the hadoop system described as follows
Namenode : Contains metadata information
Jobtracker : management and scheduling of all the jobs

2) What is slave node
Slave node runs the task tracker. Works as datanode and nodemanager

3)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?
Each value of the DNS:port has a significance
Like webapp.address port is used to check the cluster logs and hence that webapplication is running used different port number on the same machines
And hence we need to have different port number for different .war deployed for resource-tracker.address , yarn.resourcemanger.scheduler.address, yarn.resourcemanager.address,yarn.resourcemanager.admin.address etc.

Side-effects that I observed
NameNode not starting when we do ./start-dfs.sh

4) How do we change number of mapper and reducer from the configuration files
In the jar file (code) we can set to

```
job.setNumMapTasks(3);
job.setNumReduceTasks(3);
```

Cofiguration set In the configuration file is deprecated

Multi-node

Changes to be done from single node to multiple node setup

- Changes to be done before replication

Changes in file
core-site.xml – Assign EBS mounted folder name to the below property
<name>hadoop.tmp.dir</name>

Rest other files are same since DNS was already given for single node in the file core-site.xml,

mapred-site.xml

```
<configuration>
<property>
<name>mapreduce.job.tracker</name>
<value>hdfs://ec2-52-36-82-211.us-west-2.compute.amazonaws.com:9001</value>
</property><property>
<name>mapreduce.framework.name</name>
<value>yarn</value></property>

</configuration>
```

yarn-site.xml

```
<configuration>
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
```

```
<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value></property>
<property>
<name>yarn.resourcemanager.resource-tracker.address</name>
<value>ec2-52-36-82-211.us-west-2.compute.amazonaws.com:9025</value>
</property>
<property>
<name>yarn.resourcemanager.scheduler.address</name>
<value>ec2-52-36-82-211.us-west-2.compute.amazonaws.com:9030</value>
</property>
<property>
<name>yarn.resourcemanager.address</name>
<value>ec2-52-36-82-211.us-west-2.compute.amazonaws.com:9050</value>
</property>
<property>
<name>yarn.resourcemanager.webapp.address</name>
<value>ec2-52-36-82-211.us-west-2.compute.amazonaws.com:9006</value>
</property>
<property>
<name>yarn.resourcemanager.admin.address</name>
<value>ec2-52-36-82-211.us-west-2.compute.amazonaws.com:9008</value>
</property><!---->
<property>
<name>yarn.nodemanager.vmem-pmem-ratio</name>
<value>2.1</value>
</property>

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

</configuration>
```

core-site.xml

```
<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://ec2-52-36-82-211.us-west-2.compute.amazonaws.com:9000</value>
</property>
<property>
<!--replace data with your folder name u used in radi-0 !-->
<name>hadoop.tmp.dir</name>
<value>/vasudevhadoop</value>
<description>base location for other hdfs directories.</description>
</property>
</configuration>
```

- Changes to be done in host files

In master

Modify etc/hadoop/slave – Add DNS of all the 16 nodes

Modify /etc/hosts file – Add the private and DNS of all 16 nodes

/slaves file

```
ec2-52-36-82-211.us-west-2.compute.amazonaws.com
ec2-52-38-156-0.us-west-2.compute.amazonaws.com
ec2-52-37-183-182.us-west-2.compute.amazonaws.com
ec2-52-38-155-225.us-west-2.compute.amazonaws.com
ec2-52-38-154-153.us-west-2.compute.amazonaws.com
ec2-52-33-127-247.us-west-2.compute.amazonaws.com
ec2-52-38-156-32.us-west-2.compute.amazonaws.com
ec2-52-38-154-38.us-west-2.compute.amazonaws.com
ec2-52-38-156-94.us-west-2.compute.amazonaws.com
ec2-52-38-78-215.us-west-2.compute.amazonaws.com
ec2-52-37-225-137.us-west-2.compute.amazonaws.com
ec2-52-38-154-109.us-west-2.compute.amazonaws.com
ec2-52-38-156-89.us-west-2.compute.amazonaws.com
ec2-52-38-129-14.us-west-2.compute.amazonaws.com
ec2-52-10-247-105.us-west-2.compute.amazonaws.com
ec2-52-38-156-254.us-west-2.compute.amazonaws.com
ec2-52-38-149-57.us-west-2.compute.amazonaws.com
```

Etc/host

```
172.31.6.22 ec2-52-36-82-211.us-west-2.compute.amazonaws.com
172.31.10.171 ec2-52-38-156-0.us-west-2.compute.amazonaws.com
172.31.0.196 ec2-52-37-183-182.us-west-2.compute.amazonaws.com
172.31.13.220 ec2-52-38-155-225.us-west-2.compute.amazonaws.com
172.31.1.193 ec2-52-38-154-153.us-west-2.compute.amazonaws.com
172.31.4.31 ec2-52-33-127-247.us-west-2.compute.amazonaws.com
172.31.5.191 ec2-52-38-156-32.us-west-2.compute.amazonaws.com
172.31.10.132 ec2-52-38-154-38.us-west-2.compute.amazonaws.com
172.31.13.98 ec2-52-38-156-94.us-west-2.compute.amazonaws.com
172.31.11.228 ec2-52-38-78-215.us-west-2.compute.amazonaws.com
172.31.3.162 ec2-52-37-225-137.us-west-2.compute.amazonaws.com
172.31.9.87 ec2-52-38-154-109.us-west-2.compute.amazonaws.com
172.31.4.246 ec2-52-38-156-89.us-west-2.compute.amazonaws.com
```

```
172.31.8.18 ec2-52-38-129-14.us-west-2.compute.amazonaws.com
172.31.4.67 ec2-52-10-247-105.us-west-2.compute.amazonaws.com
172.31.6.124 ec2-52-38-156-254.us-west-2.compute.amazonaws.com
172.31.3.57 ec2-52-38-149-57.us-west-2.compute.amazonaws.com
```

Similarly change in the slaves setup
In all 16 slaves

Modify etc/hadoop/slave – Add DNS of master and this node

Modify /etc/hosts file – Add the private and DNS of master and this node

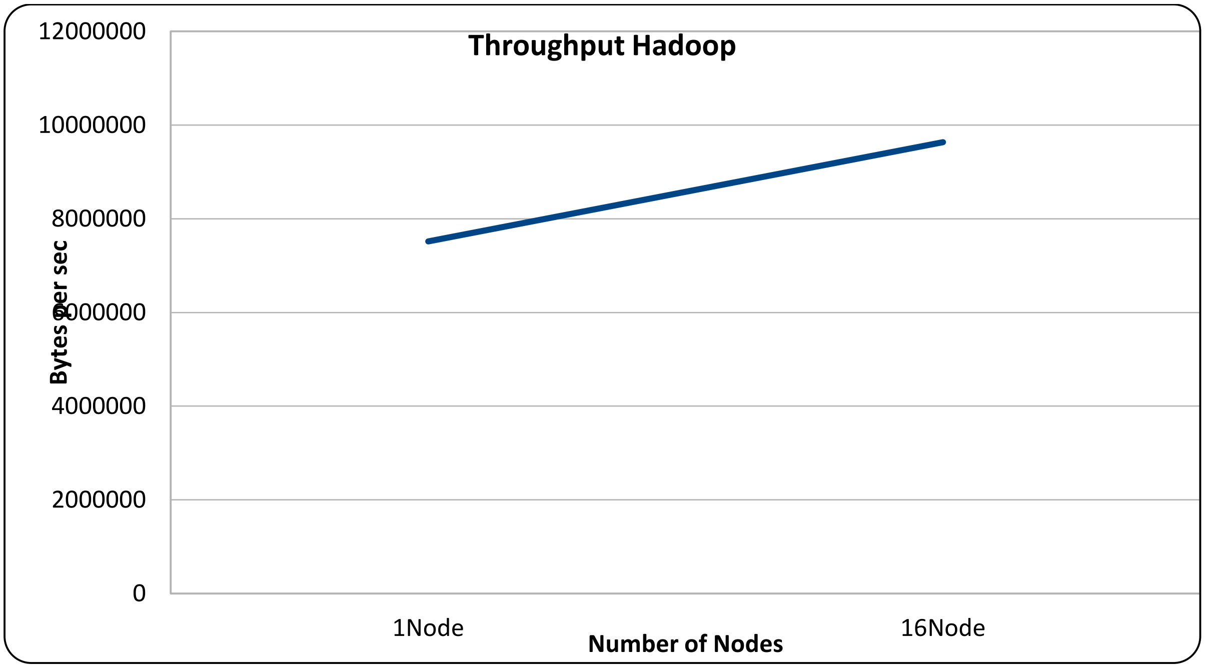
- Copy public key from each node to other 16 nodes using following:

ssh-copy-id -i ~/.ssh/id_rsa.pub ubuntu@ec2-52-36-82-211.us-west-2.compute.amazonaws.com

try SSH DNS (without using public key/password)

Readings are taken for single node 10GB dataset and 17 nodes 100GB dataset

Number of threads	1	17
Bytes per seconds	7518796	9633911.36

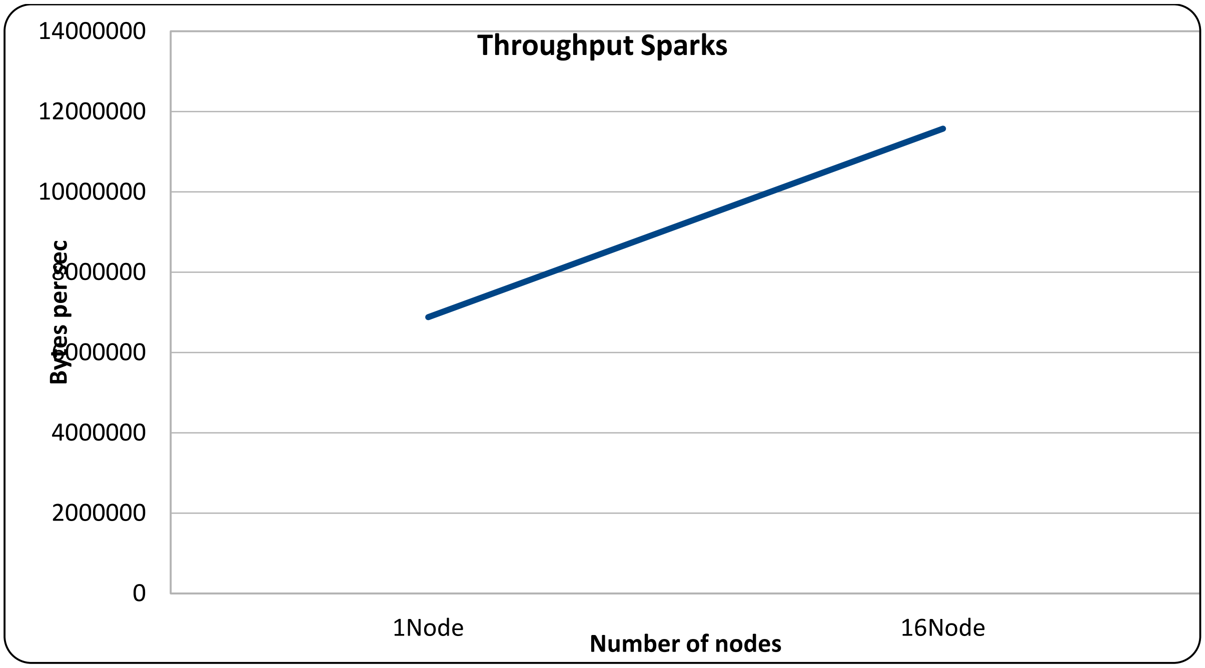


Sparks

System installation and explanation refer README.txt file.

Readings are taken for single node 10GB dataset and 17 nodes 100GB dataset

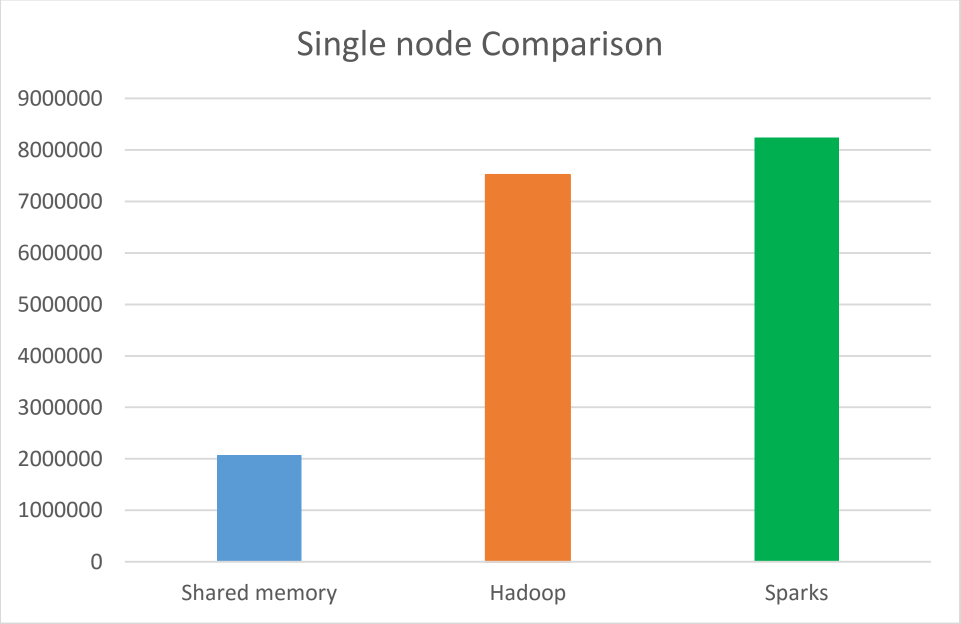
Number of threads	1	17
Bytes per seconds	6878525	11574074



Comparison

Single node comparison

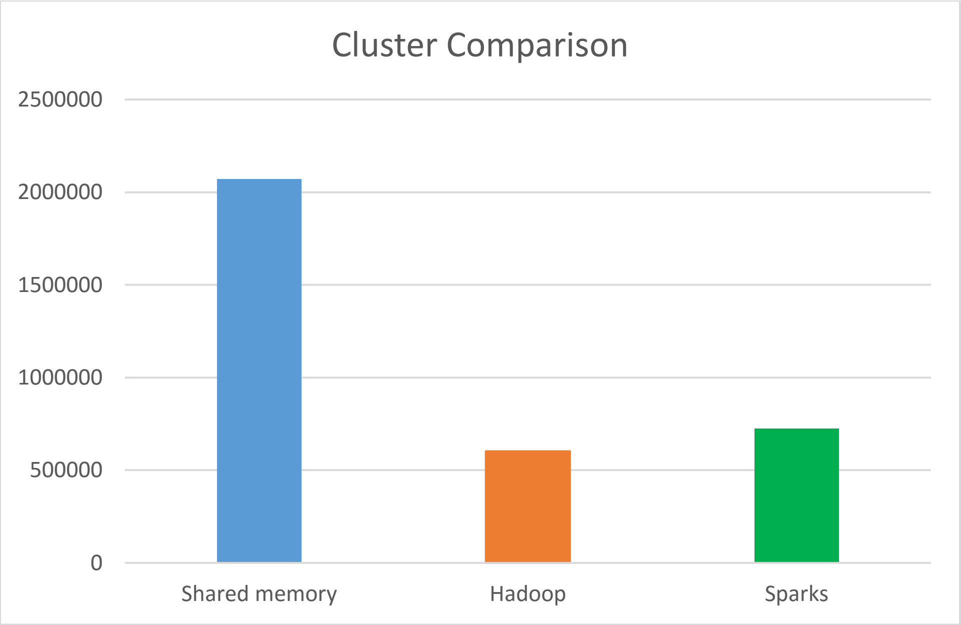
	Bytes per second
Shared memory	2071425.7
Hadoop	7518796
Sparks	8243333



Shared memory performing less efficiently as compared to both the Hadoop and sparks. Sparks is faster overall.

Cluster

	Bytes per second
Shared memory	2071425.7
Hadoop	602119.5
Sparks	723379.6



Single node performance with cluster environment, shared memory sort is fast
Sparks is still has an edge over Hadoop.

Reference :

http://www.avajava.com/tutorials/lessons/how-do-i-read-a-string-from-a-file-line-by-line.html
http://geeksquiz.com/quick-sort/
http://www.asciitable.com/
http://www.journaldev.com/1069/java-thread-pool-example-using-executors-and-threadpoolexecutor
http://www.tutorialspoint.com/java/io/randomaccessfile_read.htm
http://stackoverflow.com/questions/12616124/get-number-of-files-in-a-directory-and-its-subdirectories
https://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html

http://spark.apache.org/docs/latest/quick-start.html