Hadoop Installation,Configuration and Execution steps

STEP1: Transferring the local files to amazon linux machine . [Fire in the local machine]

scp -i cloudultimate.pem configfile/ 64.tar.gz hadoooExp.tar.gz ubuntu@ec2-52-36-82-211.us-west-2.compute.amazonaws.com:

STEP2 : Basic tool installation on aws machine [Fire on aws machine]

GCC installation
sudo apt-get install gcc -y
sudo apt-get update -y

Java Installation
sudo apt-get update -y
sudo apt-get install openjdk-7-jdk -y

Ant installation
sudo apt-get install ant -y
sudo apt-get update

# SSH installation
sudo apt-get install ssh -y
sudo apt-get update

Mount EBS Volume
lsblk
sudo file -s /dev/xvdb
sudo mkfs -t ext4 /dev/xvdb

sudo mkdir /vasudevhadoop
sudo mount /dev/xvdb /vasudevhadoop
sudo chmod 777 /vasudevhadoop/

STEP 3 : Hadoop Installation and extraction of program [Commands to be fired on aws machine]

Hadoop Installation
wget http://www-us.apache.org/dist/hadoop/common/hadoop-2.7.2/hadoop-2.7.2.tar.gz
tar -xzvf hadoop-2.7.2.tar.gz
tar -xzvf 64.tar.gz
tar -xvf hadoopExp.tar.gz

STEP: Give permissions to all the copied files [Fire on aws machine]

cd $HOME
chmod 777 hadoop-2.7.2 cloudultimate.pem configfile 64/ hadoopExp

STEP5 : Make changes in configuration files[Fire on aws machine].
change core-site.xml, hadoop-env.sh, hdfs-site.xml, yarn-site.xml and mapred-site.xml in hadoop-
##2.7.2/etc/hadoop/ folder.
do not forget to modify the DNS in the files
cp configfile/* hadoop-2.7.2/etc/hadoop/

STEP6 : On amazon web service console : Right Click on instance and create image of master
Using that image launch the slave instances [on Amazon web service console]

STEP 7 : Edit slaves in /etc/hadoop/ and /etc/hosts file in master and slaves instances
For master the changes are as follows

master: slaves file : public dns of master and all the slaves
hosts file : private_ip and public_dns of master and all the slaves

slaves: slaves file : public dns of master and that slave
hosts file : private_ip and public_dns of master and that slave

#example of slave file
#ec2-52-38-87-112.us-west-2.compute.amazonaws.com

#exmple of /etc/hosts file
#open this file with sudo permissions
#sudo vi /etc/hosts
#example of hosts file

#172.31.36.166 ec2-52-38-87-112.us-west-2.compute.amazonaws.com

cd $HOME
STEP 8 : Repeat below 6 steps on all nodes[in case of cluster or only on master in case of single node].

eval "$(ssh-agent)"
chmod 400 cloudultimate.pem
ssh-add cloudultimate.pem
ssh-keygen -t rsa

ssh-copy-id -i ~/.ssh/id_rsa.pub ubuntu@essh-keygen -t rsac2-52-38-87-112.us-west-2.compute.amazonaws.com
chmod 0600 ~/.ssh/authorized_keys

STEP 9: Starting hadoop[on master node of aws machine].
cd hadoop-2.7.2/
bin/hadoop namenode -format
ssh localhost
cd sbin
./start-dfs.sh
./start-yarn.sh

Command to check if the namenode and datanode are up and running
jps
bin/hadoop dfsadmin -report

STEP 10 : Running Sorting program and comparing the output using valSOrt
##Run all below command from main hadoop folder "hadoop-2.7.2".
command for creating Hadoop directory

bin/hadoop fs -mkdir /myhdfs

Generate file using gensort and Copy file to hadoop location.
cd $HOME
cp -rf 64 /vasudevhadoop/
cd /vasudevhadoop/64/
./gensort -a 1000000000 10GBFile
cd /home/ubuntu/hadoop-2.7.2/
bin/hadoop dfs -copyFromLocal /vasudevhadoop/64/10GBFile /myhdfs
bin/hadoop dfs -ls /myhdfs/
bin/hadoop dfs -rm -r /myhdfs/output

Run Mapreduce Jar
bin/hadoop jar /home/ubuntu/hadoopExp/dist/CloudComputingAS2hadoop.jar /myhdfs/10GBFile /myhdfs/output >&web.out&

#Command to get the file from hadoop file system to our instance EC2.
bin/hadoop dfs -get /myhdfs/output/part-r-00000 /vasudevhadoop/64

Check file using val sort

cd /vasudevhadoop/64
./valsort part-r-00000

Code explanation :

Classes

HadoopSort:

This is the main class which has two `SortMapper` and `SortReducer` static classes
Create the job and configuration for mapper and reducer

SortMapper:

Mapper takes lines as input and break down into key part(0-10) and value part(10)
It emits Key as text and value as Text

SortReducer

Reducer takes key and value as text and emits the Key and value

Problem faced and troubleshooting

1) Datanode not starting
when we check JPS there are no datanode up and running
Solution :
- ✓ Stop all service by command ./stop-dfs.sh and ./stop-yarn.sh
- ✓ remove all the data in the directory named under file hdfs-site.xml
- ✓ unmout and mount EBS
- ✓ Reformat the namenode again
- ✓ Start all services

2) Some memory related error while formatting
remove all the memory related configuration from config files
3) If namenode does not start then re format it
This happens if we do some changes in configuration or hosts files.
4) If while starting datanode if we get error like
Unable to connect DNS (public key)

Solution : try to do SSH DNS(slave) from master node and to master node from that DNS(slave).
Add the ~/.ssh/id_rsa.pub to the authorized keys use below commands
ssh-copy-id -i ~/.ssh/id_rsa.pub DNS

Or directly copy
cat ./ssh/id_rsa.pub to vi ./ssh/authorized_keys

Try doing SSH again without password/public key.

SharedMemory Installation,Configuration and Execution steps

STEP1: Transferring the local files to amazon linux machine . [Fire in the local machine]

scp -i cloudultimate.pem CloudComputingAS2Sort.tar.gz ubuntu@ec2-52-38-87-112.us-west-2.compute.amazonaws.com:

STEP2 : Basic tool installation on aws machine [Fire on aws machine]

GCC installation
sudo apt-get install gcc -y
sudo apt-get update -y

Java Installation
sudo apt-get update -y
sudo apt-get install openjdk-7-jdk -y

Ant installation
sudo apt-get install ant -y
sudo apt-get update

# SSH installation
sudo apt-get install ssh -y
sudo apt-get update

STEP 3: Extract and give permissions to all the copied files [Fire on aws machine]

tar -xzvf CloudComputingAS2Sort.tar.gz
chmod 400 cloudultimate.pem
chmod 777 CloudComputingAS2Sort

STEP 4 : Changes in the configuration files

cd CloudComputingAS2Sort/resource/
Add or edit the property values as per requirement

fileName = name of the file
fileSizeInByte=Size of the file in bytes
NumberOfThreads = Number of threads as integers
chunkSize=Chunk size in bytes

STEP 5 :

10GB file configuration

Make changes in the property file accordingly
Run it for different number of threads (8,4,2,1)

To run go to folder fire following commands:

cd CloudComputingAS2Sort/
ant
java -cp ./dist/CloudComputingSort.jar `SortingFile`

Take the readings of the time

to validate the file run valsort

cd 64/
./valsort outputfilename

Folders and its significance

/dest = contains the output sorted file
/dist = contains the .jar file
/source = folder contains the input file(file to be sorted)
/resources= folder contains config.properties file

Code explanation :

Classes =

SortingFile class :

This class is the main class which performs the following task
1) Read the input from the config.properties file
2) Assign the job to the worker threads using Executor
3) If the /dest folder contains files more than 1 than merge the file, it assigns job to mergeThreadJob using executor
4) It also contains sorting logic (Refered Quick sort)

5) It also performs writing sorted lines back to output file

ThreadJob :

This class is implementing runnable class and performs the following task in the order.

1) Read the file starting from byte specified in the input file
2) sort the file using sorting method
3) Write back into small files of the chunk sorted

mergeThreadJob :

This class is implementing runnable class and performs the following task in order

1) Take two files as input
2) Read line by line and using sorting logic sort the file
3) Write back to a single file and place it in the /dest folder

Basic code logic is as follows :

1)Take file name as input from config file
2)Break it down into chunks for size specified config file and assign it to Executor with number of thread s specified in the config file. This executor runs these threads for all the jobs
3) Each job read the file starting from the byte specified to the number of bytes to read, Sort the file and write back to a single file at destination (/dest)
4) In main function a check is made to check if the /dest folder contains more than 1 file to merge
5) If yes, call mergeThreadJob which performs merging of two sorted files and write back the result into single file
6) For the above task another Executor is assigned which devides the job to threads available
7) Each thread work on two different file and writes the result back to single file
8) Ultimately only 1 sorted file is left in foder /dest

Sparks Installation,Configuration and Execution steps

On the the machine fire the following commands

Download AccessKey from amazon → Account → Security credentials → create and download key

Installation :

export AWS_ACCESS_KEY_ID="AccessKey"
export AWS_SECRET_ACCESS_KEY="Secrete key”

wget http://www-eu.apache.org/dist/spark/spark-1.6.1/spark-1.6.1.tgz
tar -xvf spark-1.6.1.tgz
cd /home/vasudev/Downloads/spark-1.6.1

./spark-ec2 -k vasudevgawde -i /home/vasudev/Download/cloudultimate.pem -s 16 -t d2.xlarge --spot-price=0.16 launch Vasudev_setup

./spark-ec2 -k vasudevgawde -i /home/vasudev/Download/cloudultimate.pem login Vasudev_setup

Extracting the project
tar -xvf sparkExp.tar.gz
cd sparkExp
cp spark.sbt /SPARK_DIR/
cp sparkSort.scala /SPARK_DIR/src/main/scala/

sbt package

Run scala application

Code explanation

read the file and store in buffer
pass to mapper in the form of key(0,10) and value as (10,length)
Sort and collect the sorted data