

CS553 Programming Assignment 2

Hadoop:(1 node-1 GB data)

1.Hadoop has been installed on 1 node(c3.large instance).The following config files have been modified:

conf/core-site.xml
conf/hdfs-site.xml
conf/mapred-site.xml
conf/yarnsite.xml
conf/slaves
conf/hadoop-env.sh

The sort program on 1GB of data execution is shown below:

```
ubuntu@ip-172-31-12-52:~/hadoop-2.7.2/bin$ ./hdfs dfs -ls /user/yamini/
Found 1 items
brw-rw-r-- 1 ubuntu supergroup 0 2016-03-30 09:19 /user/yamini/input
ubuntu@ip-172-31-12-52:~/hadoop-2.7.2/bin$ ./hdfs dfs -ls /user/yamini/input
Found 1 items
-rw-rw-r-- 1 ubuntu supergroup 1000000000 2016-03-30 09:19 /user/yamini/input/testdata.txt
ubuntu@ip-172-31-12-52:~/hadoop-2.7.2/bin$ ls
container-executor  hadoop  hadoop.cmd  HadoopNode.java  hdfs  hdfs.cmd  mapred  mapred.cmd  rcx  test-container-executor  yarn  yarn.cmd
ubuntu@ip-172-31-12-52:~/hadoop-2.7.2/bin$ ./hadoop com.sun.tools.javac.Main HadoopNode.java
HadoopNode.java:161: error: class HadoopNode is public, should be declared in a file named HadoopNode.java
public class HadoopNode {
      ^
1 error
ubuntu@ip-172-31-12-52:~/hadoop-2.7.2/bin$ ls
container-executor  hadoop  hadoop.cmd  HadoopNode.java  hdfs  hdfs.cmd  mapred  mapred.cmd  rcx  test-container-executor  yarn  yarn.cmd
ubuntu@ip-172-31-12-52:~/hadoop-2.7.2/bin$ rm HadoopNode.java
ubuntu@ip-172-31-12-52:~/hadoop-2.7.2/bin$ ls
container-executor  hadoop  hadoop.cmd  hdfs  hdfs.cmd  mapred  mapred.cmd  rcx  test-container-executor  yarn  yarn.cmd
ubuntu@ip-172-31-12-52:~/hadoop-2.7.2/bin$ cd ~
ubuntu@ip-172-31-12-52:~$ ls
H013.pem  gensort  hadoop-2.7.2  HadoopNode.java  testdata.txt
ubuntu@ip-172-31-12-52:~$ ls
H013.pem  gensort  hadoop-2.7.2  HadoopNode.java  testdata.txt
ubuntu@ip-172-31-12-52:~$ rm HadoopNode.java
ubuntu@ip-172-31-12-52:~$ ls
H013.pem  gensort  hadoop-2.7.2  testdata.txt
ubuntu@ip-172-31-12-52:~$ ls
H013.pem  gensort  hadoop-2.7.2  HadoopNode.java  testdata.txt
ubuntu@ip-172-31-12-52:~$ cd hadoop-2.7.2
ubuntu@ip-172-31-12-52:~/hadoop-2.7.2$ ls
bin  etc  include  lib  libexec  LICENSE.txt  logs  NOTICE.txt  README.txt  share
ubuntu@ip-172-31-12-52:~/hadoop-2.7.2$ cd bin
ubuntu@ip-172-31-12-52:~/hadoop-2.7.2/bin$ ls
container-executor  hadoop  hadoop.cmd  hdfs  hdfs.cmd  mapred  mapred.cmd  rcx  test-container-executor  yarn  yarn.cmd
ubuntu@ip-172-31-12-52:~/hadoop-2.7.2/bin$ cp ~/HadoopNode.java .
ubuntu@ip-172-31-12-52:~/hadoop-2.7.2/bin$ ls
container-executor  hadoop  hadoop.cmd  HadoopNode.java  hdfs  hdfs.cmd  mapred  mapred.cmd  rcx  test-container-executor  yarn  yarn.cmd
ubuntu@ip-172-31-12-52:~/hadoop-2.7.2/bin$ ./hadoop com.sun.tools.javac.Main HadoopNode.java
HadoopNodeNode.class : no such file or directory
ubuntu@ip-172-31-12-52:~/hadoop-2.7.2/bin$ jar cf HadoopNode.jar HadoopNodeNode.class
ubuntu@ip-172-31-12-52:~/hadoop-2.7.2/bin$ ./hadoop jar HadoopNode.jar HadoopNode /user/yamini/input/testdata.txt /user/output/
16/03/30 19:04:12 INFO client.RMProxy: Connecting to ResourceManager at ec2-52-96-232-89.compute-1.amazonaws.com/172.31.32.32:8032
16/03/30 19:04:12 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to
remedy this.
16/03/30 19:04:13 INFO InputFileInputFormat: Total input paths to process : 1
16/03/30 19:04:13 INFO mapreduce.JobSubmitter: number of splits:0
16/03/30 19:04:13 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1459382761748_0001
16/03/30 19:04:14 INFO InputFileSystemImpl: Submitted application application_id=1459382761748_0001
16/03/30 19:04:14 INFO mapreduce.Job: The url to track the job: http://ec2-52-96-232-89.compute-1.amazonaws.com:8088/proxy/application_1459382761748_0001/
16/03/30 19:04:14 INFO mapreduce.Job: Running job: job_1459382761748_0001
16/03/30 19:04:23 INFO mapreduce.Job: Job job_1459382761748_0001 running in user mode : false
16/03/30 19:04:23 INFO mapreduce.Job: map 0% reduce 0%
|
```

```

FSH: Number of bytes read(200000000)
FSH: Number of bytes written(200000000)
FSH: Number of read operations(0)
FSH: Number of large read operations(0)
FSH: Number of write operations(0)
RSH: Number of bytes read(200000000)
RSH: Number of bytes written(200000000)
RSH: Number of read operations(0)
RSH: Number of large read operations(0)
RSH: Number of write operations(0)

Job Counters
  Hitted map task(s)
    Launched map task(s)
    Launched reduce task(s)
    Data-local map task(s)
    Total time spent by all maps in occupied state (ms)(200000)
    Total time spent by all reducers in occupied state (ms)(2000)
    Total time spent by all map tasks (ms)(200000)
    Total time spent by all reduce tasks (ms)(2000)
    Total user-submitted seconds taken by all map tasks(200000)
    Total user-submitted seconds taken by all reduce tasks(2000)
    Total megabyte-milliseconds taken by all map tasks(200000000)
    Total megabyte-milliseconds taken by all reduce tasks(2000000)

Map-Reduce Framework
  Map input records(200000000)
  Map output records(200000000)
  Map output bytes(200000000)
  Map output materialized bytes(200000000)
  Input split bytes(2000)
  Combine input records(200000000)
  Combine output records(200000000)
  Reduce input groups(200000000)
  Reduce shuffle bytes(200000000)
  Reduce input records(200000000)
  Reduce output records(200000000)
  Spilled Records(200000)
  Shuffled Map(s) 0
  Failed Shuffles(0)
  Merged Map outputs(0)
  IO time elapsed (ms)(2000)
  CPU time spent (ms)(20000)
  Physical memory (bytes) spilled(200000000)
  Virtual memory (bytes) spilled(200000000)
  Total committed heap usage (bytes)(170000000)

Shuffle Errors
  BAD_ID(0)
  CONNECTION(0)
  IO_EXCEPTION(0)
  UNRESOLVED_URI(0)
  WRONG_MAP(0)
  WRONG_REDUCE(0)

File Input Format Counters
  Bytes Read(200000000)
File Output Format Counters
  Bytes Written(200000000)

ubuntu@192-168-1-101:~$ hadoop-0.7.0/bin/

```

Hadoop cluster:(Screenshots attached)

Hadoop:

1) What is a Master node? What is a Slaves node?

Master node manages the process of mission partitioning and task delegation. Slave nodes are workers that execute the tasks that are assigned to them by the master node.

2) Why do we need to set unique available ports to those configuration files on a shared environment?

What errors or side-effects will show if we use same port number for each user?

Different ports have different functionalities (example - HDFS on port 1, 9002 and Master node on port 0, 9001). Consequently, port collision becomes a possibility if the same port number is used for each user.

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

The number of map tasks can be increased manually using the jobConf's `conf.setNumMapTasks(int num)`. It must be noted, however, that this will not set the number below what Hadoop determines by splitting the input data. Similarly, the number of reduce tasks may be increased using the JobConf's `con.setNumReduceTasks(int num)`.

Steps to launch a Spark cluster and run the sort program on it:

A 17 node spark cluster has been setup and the sort program was run on it with a 100 GB dates generated through gensort .

Versions used:

- 1.OS used-ubuntu
- 2.Java version-1.7
- 3.Spark-1.6.0-bin-hadoop2.6
- 4.Code is written in python(2.7)
- 5.Instance type-c3 large

Steps :

- 1.Login to instance and install java
- 2.Download and unzip spark
- 3.Install scala
- 4.modify .bashrc
- 5.Export Access key and secret access key downloaded from AWS
- 6.In the spark ec2 folder,run the command to launch a master with 16 slaves (ebs volume 400 added) using spot instances.(mentioned the instance type in the command)
- 7.1 master and 16 slave instances get created.
- 8.ssh into the master and created a folder name 'knn' in it
- 9.Uploaded the code and data files in inn
- 10.Using the command below uploaded the code to all nodes in the cluster
 . /spark-ec2/copy-dir knn
- 11.Upload the data file in hfs
- 12.Run the spark submit command in /spark/bin/folder
- 13.This initiates the executors on all nodes and runs the tasks(screenshots) provided
- 14.Output captured in 'output' folder
- 15.Validate the output.(First part and last part screenshots provided).