(1)

Big Data Practicals:

(i) Cloudera Quick start VM is a very much CPU intensive atleast specify the 2 CPU and increase the RAM size. (5GB RAM)

which gives the cloudera hadoop distribution in a single node cluster setup, which can be used for working & learning about different distribution. In a cloudera with hdfs and other hadoop ecosystem components.

In CQVM, we have one scm server and one scm agent which will start the system.

Admin control accessable : we have to check if we have an access to the cluster. using the following commands.

In Terminal

$ hostname

$ hdfs dfs -ls / → (There is connection establishment b/w scm server and scm agent as cloudera which take care of cluster.)

we can also check with following code.

$ service cloudera-scm-server status

(2)

The above code would tells you to use Cloudera express.

login as a root :

$ su |password : Cloudera

$ service cloudera-scm-server status

The above all the codes tells you hdfs working fine, now close the terminal

Now, click on the launch cloudera Express icon, you will get the command

sudo /home/cloudera/cloudera-manager --force

Copy the above code and paste into the New Terminal window

$ sudo /home/cloudera/Cloudera-manager --force --express

This code would tells you shutdown all the services of Cloudera and restart the services, after which only you will be able to access Admin Console.

Now, the deployment has been configured and and restarted the cloudera manager.
It gives you Quick start admin Control. using username and pwd.

Open the Browser, change ~~RE~~ into

quickstart.cloudera:7180
↳ default port that shows admin console.

↓
click I agree
↓
In the admin console window
enter usrname & pwd
cloudera — username
cloudera — password.

Hue : is a web interface which allows work with hdfs and depending on hue.

Remove the unused services

After, click on Actions
↓
Restart
↳ close (let it happen in background)

Click on Hosts
↓
All hosts
↓ As of now we have only one host.

SEPTEMBER 2019
Su Mo Tu We Th Fr Sa
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30

HDFS (Hadoop Distributed File System)

- a framework for the analysis and transformation of very large data sets using mapreduce paradigm.

- Important characteristics: Hadoop is the partitioning of data and computation across many (thousands) of hosts, and executing application computations in parallel close to their data.

Basic Commands in cloudera:

In Terminal I

```
$ ls
$ pwd
$ mkdir sudhakar
$ cd sudhakar.
$ pwd
$ cd ..
$ ls
$ cat name.txt
```

In Terminal I

```
$ hadoop fs -ls /
$ pwd
$ hadoop fs -ls /usr
$ hadoop fs -put employee.txt /usr
```

(used to copy the file from host to hdfs)

pwd - present working directory.

$ sudo jps (used for my name node is these. by not)

$ pwd (need to check where I am)

/home/cloudera

$ cd /home

$ ls

$ cd ..

$ ls /home/cloudera

$ ls /home/cloudera -ltr

$ ls /home/cloudera/Downloads -ltr

$ cat /home/cloudera/Downloads/products.csv ← to see the contents inside the file

hadoop

$ hadoop fs -ls

old approach doing same thing

$ hadoop fs -ls / ↳ list out directory list (Root)

$ hdfs dfs -ls /

new approach

$ hdfs dfs -ls /foldername.

$ hdfs dfs -cat /foldername/filename.txt ← to see the contents in the file

21 Saturday — SEPTEMBER 2019

2019 Q3 | Week 38

In local host $ sudo mkdir /foldername
Create directory $ ls /-ltr

(6)

Creating the new directory in hadoop:

$ hdfs dfs -mkdir /foldername

List out the directory:

$ hdfs dfs -ls /

Copy files from local host to hdfs.

In local host:
list out files:

$ ls /home/cloudera/Downloads

$ hadoop copyFromLocal

22 Sunday

Understanding Hadoop command

In cloudera, we have 2 environments
1. Base environment (Linux)
2. distributed " (Hadoop)

$ hadoop fs -ls /

It will list out all the files in hadoop

$ hdfs dfs -ls /

This One is also list out all the files in the hadoop.

$ hadoop fs -mkdir /test3

This command is used to create directory in hadoop environment.

$ hadoop fs -ls /

$ clear

Copy a file from host to distributed System

$ ls

$ cd test3 // with in test2 or 3 file, create one test.txt

$ ls
test.txt

$ hadoop fs -put test.txt /test3

Check whether file is there or Not.

$ hadoop fs -ls /test3

Equivalent copy command

$ hadoop fs -copyFromLocal test.txt /test4

Here test4 folder is not available, though by default it will create a directory.

$ hadoop fs -cat /test4

This command will show you test written in test4 directory.

$ Cat test.txt

// move a file from host to hadoop.

$ hadoop fs -mkdir /testb

$ hadoop fs -ls /

$ hadoop fs -moveFromLocal test.txt /testb

$ ls

$ hadoop fs -ls /testb

// we want to get a file from hadoop

$ hadoop fs -get /testb/test.txt

$ ls

// in hadoop environment we want to copy a file from one location to another location.

$ hadoop fs -mkdir /test7

// move a testb/test.txt.

$ hadoop fs -cp /testb/test.txt /test7

$ hadoop fs -ls /test7

Diff. b/w HDFS and Linux

HDFS → is a stateless i.e doesn't remember working directory, each time we have to write full path.

→ Cannot modify the file, But we can delete and copy the file.

$ hadoop fs

$ hadoop fs -ls /

In linux
$ ls

⇒ That " / " denotes root directory in the hdfs.

If you want to go to the particular directory use this following command.

$ hadoop fs -ls /usr

Create our own directory:

$ hadoop fs -mkdir /mydata

mydata folder will be created.

$ hadoop fs -mkdir /mydata/testfolder

Creating sub folder within mydata.

creating a text file in linux:

$ gedit testfile.txt &

$ ls *.txt // list out all txt file

// save this file in hadoop

$ hadoop fs -put testfile.txt /mydata/testfolder/

$ hadoop fs -ls /mydata/testfolder/

// to get a file from hadoop to linux.

$ hadoop fs -get /mydata/testfolder/testfile.txt newfile.txt

$ ls *.txt

Hive Setup in Cloudera

Step 1 : Create a folder on your Desktop eg: example.

Step 2 : Open a terminal window.

type the command :

$ sudo su

password : Cloudera.

Step 3 : $ cd /usr/lib/hive/conf/

sudo gedit hive-site.xml

Change the file path to Desktop

$ sudo hive

Data to Analytics:

Overview of HDFS, Map reduce, Hive, Sqoop, and Spark.

$ hadoop fs // to see what are all the commands are there

$ touch testing // make a file. (empty file in local file sys)

$ hadoop fs -put testing /user/cloudera

$ hadoop fs -ls /user/cloudera // list out copy file.

$ vi wordcount.txt // create a file

$ hadoop fs -put wordcount.txt /user/cloudera

$ sudo find / -name "*hadoop*examples*.jar"

Select the

/usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar

copy the link and paste it

$ hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-ex.jar

// This will provide all the applications that part of jar files

$ hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce.jar wordcount /user/cloudera/wordcount.txt /user/cloudera/output

/user/cloudera/wordcount.txt → i/p

/user/cloudera/output → o/p

Now, the mapreduce job will run, it will give you count of each word in wordcount.txt.

$ hadoop fs -ls /user/cloudera/output

$ hadoop fs -cat /user/cloudera/output/part-r-00000

// Count of each word..

Share folders b/w VM & Host.

Devices → shared folder → shared folder setting

+

machine folder

+

click file menu (+)

with in window

folder path : C:\projects (windows folder path copy)

folder name : projects

check : automount

make permanent.

+

ok

touch z → is used to create empty file

Run the mount command:

open a Terminal
login as a root

$ su

$ cd /home/cloudera/workspace;

$ mkdir projects

$ ls -l

$ mount -t vboxsf projects /home/cloudera/worspac/projects

projects → Shared folder name.
/home/cloudera/worspac/projects → target

Difference b/w LFS and HDFS:

LFS	HDFS
Linux	Special commands.

Basic commands:

$ su root

pwd: cloudera

$ jps // Java processes

// It will show you HDFS components are running

print all the hadoop process

16

main components:
- Node manager
- Namenode
- Data node
- Resource manager.

Others are supporting components.

$ su cloudera // come back to normal user.

$ clear.

How we can check the health i.e How nodes are running.

fsck → Check the health of hdfs.

$ hdfs fsck /

$ hdfs dfs -ls /

$ hdfs dfs -mkdir /bigdatatesting.

$ hdfs dfs -ls /

touchz => create a new file on HDFS with size 0 bytes.

$ hdfs dfs -touchz /bigdatatesting/test.dat

vi editor for local file system

du => checks the size of a file.

$ hdfs dfs -du -s /bigdatatesting/test.dat

How we can contents to the existing file:

appendToFile -> appends the contents to the file which is present on HDFS.

$ hdfs dfs -appendToFile - /bigdatatesting/test.dat

testing

$ hdfs dfs -du -s /bigdatatesting/test.dat

To display the contents of the file.

using cat command present in HDFS.

$ hdfs dfs -cat /bigdatatesting/test.dat

copyFromLocal (put)
copyToLocal (get)

$ cat >> test1.dat
testing hadoop.
test1 $ ls test1.dat

18

$ hdfs dfs -copyFromLocal test1.dat /bigdatatesting

verify $ hdfs dfs -ls /bigdatatesting.

$ cat >> test2.dat

$ hdfs dfs -put test2.dat /bigdatatesting.

$ hdfs dfs -ls /bigdatatesting.

Name node : is the master daemon that maintains and manages data nodes and it records only the metadata

no. of logs
filename,
file permission,
which rack the file is there.

Data node : are slave daemons (or) process which runs on each slave machine & it stores actual data

$ hadoop fs -test -d destination
↑ directory name.

// test command used for conditional checking, we can check some condition

-d means whatever path I am specifying after -d, is it is directory (or) ~~not~~ file

$ echo $? // print results.

If it't returning zero (0) means it is directory otherwise file.

$ hadoop fs -test -e destination
↑ the given directory is existing (or) not.

$ hadoop fs -test -f destination
↑ file or not.

$ hadoop fs -test -z destinaton/xyz.txt

// this would test the file size is zero (or) not.

$ echo $?

// move from local to hdfs

$ hadoop fs -moveFromLocal a2.txt destinaon/

$ cat a2.txt

// merge two file contents.

Before that create two text files in any directory. using appendToFile

$ hadoop fs -cat destination/xyz.txt
destination name

$ hadoop fs -cat destination/xyz1.txt

$ hadoop fs -getmerge -nl destination/xyz.txt destination/xyz1.txt
nl → new line

// verify whether the merged or not

$ cat Desktop/merge.txt

-/Desktop/merge.txt
Where we have to store the file. new file name.

Checksum command:

checks integrity of the file whether file is modified (or) not. we can check by hash value.

$ hadoop fs -checksum destination/xyz.txt

mysql -u root -h localhost -p -P 3306
↳ default port no.

OCTOBER 2019 (07)

Week 41 | 282 • 083

09 Wednesday

mapreduce

~~How to create and reduce mapreduce program over cloudera environment.~~

MySql is running (or) not on cloudera environment

$ sudo service mysqld status

$ mysql -u root -h localhost -p
↳ prompt the pwd.

pwd: cloudera →

mysql > show databases;
// list out all the db

mysql > use retail_db;
↳ db name
// get into the db

mysql > show tables;
// list the tables.

mysql > select count(1) from orders;

mysql > describe orders;
// list out all the field & type.

mysql > exit

SQOOP: is a language that transfer the relational data (or) sql Databases to HDFS

$ mysql -u root -P

pwd: cloudera // enter into mysql.

mysql> show databases;

mysql> Create database myfirstdb;

// Creating database

mysql> use myfirstdb;

// going to use the db

mysql> Create table mytable (ID int, name varchar(20), address varchar(20));

// Table creation.

mysql> describe mytable;

mysql> insert into mytable values (1, "Adi", "USA"), (2, "Ram", "UK"), (3, "Raj", "India");

mysql> select * from mytable;

drop table tablename;

Ip address : ip address
in terminal (or) hostname -I
10.0.2.15

In next Terminal:
login as a root:
To run a sqoop

(or) localhost

Ip address of VM depends on machine

$ sqoop import --connect jdbc:mysql://10.0.0.41:3306/myfirstdb --username root --password cloudera

dbname

--table mytable --target-dir=/user/cloudera/myfirstdata -m 1

No. of maps

$ hadoop fs -cat /user/cloudera/myfirstdata

$ hadoop fs -ls /user/cloudera/myfirstdata.

$ hadoop fs -cat /user/cloudera/myfirstdata/part-m-00000

← mysql to sqoop:

Exporting Data from HDFS to mysql Sqoop

login as root

sudu -i

sqoop export --connect jdbc:

12 Saturday
285 • 080 | Week 41
24
OCTOBER 2019

9 Use Sqoop to copy a relational db table to HDFS

10 $ mysql --user root --Password Cloudera movielens

11 mysql > show databases;

12 mysql > ~~show~~ use movielens;
↳ db name

1 mysql > show tables;

2 mysql > describe movie; ← table name

3 mysql > select * from movie limit 5;

4 mysql > select count(*) from movie;

5 Copy the all records from mysql to HDFS

6 $ Sqoop import \

7 > --connect jdbc:mysql://localhost/movielens \
> --username root -P \

13 Sunday

> --fields-terminated-by '\t' \
> --warehouse-dir /mydata \
> --table movie

$ hadoop fs -ls /mydata

$ hadoop fs -ls /mydata/movie

// view the contents.

$ hadoop fs -cat /mydata/movie/part-m-00000 | head -5

// list top 5 records only

Create and Load data in Hive table

First create sample file ; the field names are id, name, Dept, salary, Domain

1~Sachin~Pune~Product Engg~100000~Bigdata.

$ cat employee

$ hadoop fs -put employee /user/cloudera

HDFS Home Directory.

$ hadoop fs -ls /user/cloudera

$ hadoop fs -cat /user/cloudera

// display all records from local to HDFS

In another terminal

```
$ hive
        // enter into hive shell

$ hive> show databases;
hive> Create database organization;
hive> show databases;
hive> use organization;
hive> show tables;
hive> !clear;
hive> Create table employee (
        id int,
        name string,
        city string,
        department string,
        salary int,
        domain string)
      row format delimited
      fields terminated by ',';
hive> show tables;
        // list out the tables
```

hive > select * from employee
// it will not return the data
because we have not loaded

hive > load data inpath /user/cloudera/employee
Overwrite into table employee;

hive > select * from employee;

hive > desc employee;
// describe table

hive > select name, city from employee;

✓ Hive is one of the most popular Datawarehouse
✓ developed by facebook; maintained by apache hive
✓ mainly netflix and amazon.
✓ why Hive was developed?
✓ How and when it can be used?
✓ when hive cannot be used?
✓ features of Hive
✓ Hive Vs RDBMS

Hive can be used in OLAP;
✓ scalable & flexible

Cannot be used:
— It is not a relational database
— It cannot be used for OLTP

Big data Practicals:

Cloudera:

1. Word count Prg:

It will count each and every words as many times occured on the file.

```
$ jps          // return all the running nodes

$ pwd          // present working directory

$ hdfs dfs -put ../../Downloads/input /filename
               // copy the input file from local
                  file system to hadoop system.

$ hdfs dfs -ls /
               // return all the files in hadoop
```

Otherwise we can check with the user interface

On URL → localhost:9870 → Utilities → Browse Directory → we can see the input file

$ ./bin/hadoop jar share/hadoop/mapreduce/
hadoop-mapreduce-examples-3.0.0.jar wordcount
/input /input_output
(/input → filename; /input_output → directory name; wordcount → Already available in hadoop)

We can check the output either UserInterface (or) Terminal

UserInterface: use the localhost address

localhost:9870/explorer.html#/
↓
Utilities
↓
Browse Directory
↓
input_output // file name
↓
part-r-00000
↓
Select & Head the file
↓
You will see the o/p

In Terminal:

$ hdfs dfs -cat /input_output/part-r-00000

// This would return the output in terminal.

Hive

Create and load data in Hive table.

$ cat employee

- - -

- - -

$ hadoop fs -put employee /user/cloudera

$ hadoop fs -ls /user/cloudera

$ hadoop fs -cat /user/cloudera/employee

Another Terminal:

$ hive

hive> show databases;

> create database organisation;

OCTOBER 2019

Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		