ISIT312/ISIT912 Big Data Management

Spring 2023

Hadoop and HDFS Practice

Objective: After this practice, you will get familiar with using the Zeppelin or Linux shell to interact with Hadoop.

Warning: DO NOT attempt to copy the Linux commands in this document to your working Terminal, because it is error-prone. Type those commands by yourself.

To view pdf files it is recommended to use "evince" document viewer.

Software Installation and Setup.

Install the latest version of VirtualBox and the extension pack (optional). For the MacOS and Linux, the distribution packages are available at

https://www.virtualbox.org/wiki/Downloads

Import `BigdataVM-2021v2_2.ova` file located at `C:\VM Repository` on your system in 39A.104 (a laboratory room for ISIT312/912) into VirtualBox. An instruction is in

https://docs.oracle.com/cd/E26217_01/E26796/html/qs-import-vm.html

or you can just double-click on the correct `ova` file. After the import is completed, a VM named `BigdataVM-2021v2_2` appears in the VirtualBox.

Run `BigdataVM-2021v2_2`.

Note: If necessary then both the account name and the password are: `bigdata`

(0) Start Shell

After you log on the Ubuntu system in BigDataVM, start a Terminal window with `Ctrl + Alt + T` or use the third from top icon located in a vertical stripe (sidebar) on the left-hand side of the screen.

The following documents: `LinuxCommandLineCheatSheet.pdf` and `Efficient-Linux-at-the-Command-Line-ch4.pdf` contain more information on how to use Linux Shell available through Terminal window.

You can use the Terminal window to interact with Hadoop. A simple hint that may make your life much easier is to use "Up" and "Down" keys on a keyboard to navigate through the commands already processed in Terminal window.

Now you can interact with HADOOP.

(1) Hadoop files and scripts

Process the following command to have a look at what is contained in the $HADOOP_HOME:

```
ls $HADOOP_HOME       # view the root folder
ls $HADOOP_HOME/bin   # view the "bin" folder
ls $HADOOP_HOME/sbin  # view the "sbin" folder
```

The `bin` and `sbin` folders contain the scripts for initialization and management of Hadoop.

(2) Hadoop Initialisation

Now you can start Hadoop. First, to start NameNode and DataNode process the following commands:

```
$HADOOP_HOME/sbin/hadoop-daemon.sh start namenode
$HADOOP_HOME/sbin/hadoop-daemon.sh start datanode
```

To start YARN ResourceManager and NodeManager process the following commands:

```
$HADOOP_HOME/sbin/yarn-daemon.sh start resourcemanager
$HADOOP_HOME/sbin/yarn-daemon.sh start nodemanager
```

You can start NameNode, DataNode, ResourceManager and NodeManager in "one go" with the following command:

```
$HADOOP_HOME/sbin/start-all.sh
```

Finally, start the Job History Server:

```
$HADOOP_HOME/sbin/mr-jobhistory-daemon.sh start historyserver
```

View the running daemons with:

```
jps
```

The following results should be returned (note that the process numbers may be different):

```
2897 JobHistoryServer
2993 Jps
2386 NameNode
2585 ResourceManager
2654 NodeManager
2447 DataNode
```

If you use Terminal, then it is possible to start all Hadoop processes in "one go" through running a shell script `start-hadoop.sh` available through `Resources` link on Moodle. Download the script and change its access rights in the following way:

```
chmod u+x start-hadoop.sh
```

To start all Hadoop processes, execute the following command in Terminal window:

```
./start-hadoop.sh
```

(3) HDFS Shell commands

Create a folder `myfolder` in HDFS:

```
$HADOOP_HOME/bin/hadoop fs -mkdir myfolder
```

Copy a file from the local filesystem to HDFS. The following command copies all files with the `.txt` extension in `$HADOOP_HOME` to the input folder of HDFS:

```
$HADOOP_HOME/bin/hadoop fs -put $HADOOP_HOME/*.txt myfolder
```

List files in a home folder and `myfolder` folder in HDFS:

```
$HADOOP_HOME/bin/hadoop fs -ls
```

```
$HADOOP_HOME/bin/hadoop fs -ls myfolder
```

View a file in HDFS:

```
$HADOOP_HOME/bin/hadoop fs -cat myfolder/README.txt
```

Copy a file from HDFS to the local filesystem:

```
$HADOOP_HOME/bin/hadoop fs -copyToLocal myfolder/README.txt /home/bigdata/Desktop
```

```
ls /home/bigdata/Desktop
```

Remove a file in HDFS

```
$HADOOP_HOME/bin/hadoop fs -rm myfolder/README.txt
```

(4) HDFS UI

Open the Firefox Web Browser, go to `localhost:50070`. You will see `localhost:8020`. This is the location of the HDFS. It is specified in a configuration file named `core-site.xml`.

Check this file in `$HADOOP_HOME/etc/hadoop`, which contains Hadoop's configuration files.

You can view a file `core-site.xml` in Terminal:

```
cat $HADOOP_HOME/etc/hadoop/core-site.xml
```

Browse the web UI (e.g., you can see the location of the Datanode). Go to "Utilities" and then to "Browser the file system". Check the `.txt` files uploaded to HDFS previously. Note that the root folder of `bigdata` is in the `user` folder.

To view all root folders in HDFS in Terminal, you can also enter the following command in Terminal:

```
$HADOOP_HOME/bin/hadoop fs -ls /
```

(5) HDFS Java Interface

The following is a Java program to retrieve the contents of a file in the HDFS. This program is equivalent to the Hadoop command `hadoop fs -cat`. The source code of the program is available on Moodle in a file `FileSystemCat.java` and it is also provided below. Read and understand the source code.

```
// cc FileSystemCat Displays files from a Hadoop filesystem on standard output
// by using the FileSystem directly
import java.io.InputStream;
import java.net.URI;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IOUtils;

// vv FileSystemCat
public class FileSystemCat {

  public static void main(String[] args) throws Exception {
    String uri = args[0];
    Configuration conf = new Configuration();
    FileSystem fs = FileSystem.get(URI.create(uri), conf);
    InputStream in = null;
    try {
      in = fs.open(new Path(uri));
      IOUtils.copyBytes(in, System.out, 4096, false);
    } finally {
      IOUtils.closeStream(in);
    }
  }
}
// ^^ FileSystemCat
```

Now to compile `FileSystemCat.java` in the Terminal, define an environment variable:

```
export HADOOP_CLASSPATH=$($HADOOP_HOME/bin/hadoop classpath)
```

This environment variable point to all basic Hadoop libraries. Note that each time of open the `Terminal`, you need to "export" this environment variable (if you want to use it). To view these libraries, enter

```
echo $HADOOP_CLASSPATH
```

Download the file `FileSystemCat.java` to Desktop.

Now you are ready to compile the application and to create `FileSystemCat.jar` file. Process the following commands in Terminal.

```
cd /home/bigdata/Desktop
```

```
javac -cp $HADOOP_CLASSPATH FileSystemCat.java
```

```
jar cvf FileSystemCat.jar FileSystemCat*.class
```

The first command above moves to the current folder that contains the Java source (so that the compilation does not create any package namespace for the main class). The second command compiles the sourcecode. The last command creates `FileSystemCat.jar` file that includes the Java class(es).

If you use Zeppelin, the above three commands must be in the SAME paragraph. Now you can run the `FileSystemCat.jar` file by using the `hadoop` script with `jar` command.

```
$HADOOP_HOME/bin/hadoop jar /home/bigdata/Desktop/FileSystemCat.jar FileSystemCat myfolder/LICENSE.txt
```

Check whether the uploaded file is same as the local file.

(6) Shut down Hadoop

When finishing your practice with Hadoop, it is good practice to terminate the Hadoop daemons before turning off the VM.

Use the following commands to terminate the Hadoop daemons:

```
$HADOOP_HOME/sbin/hadoop-daemon.sh stop namenode
$HADOOP_HOME/sbin/hadoop-daemon.sh stop datanode
$HADOOP_HOME/sbin/yarn-daemon.sh stop resourcemanager
$HADOOP_HOME/sbin/yarn-daemon.sh stop nodemanager
$HADOOP_HOME/sbin/mr-jobhistory-daemon.sh stop historyserver
```

If you use Terminal, then it is possible to stop all Hadoop processes in "one go" through running a shell script `stop-hadoop.sh` available through `resources` link on Moodle. Download the script and change its access rights in the following way:

```
chmod u+x stop-hadoop.sh
```

To stop all Hadoop processes, execute the following command in Terminal window:

```
./stop-hadoop.sh
```

(7) Make a typescript of information in `Terminal` (optional)

If you work in Shell but not Zeppelin, you can use the `script` command to record everything printed in your `Terminal`.

```
script a-file-name-you-want-to-save-the-typescript-to.txt
<work with your Terminal..>
exit
```

Check the contents of `a-file-name-you-want-to-save-the-typescript-to.txt`.
