Hadoop commands

1. version

Shows the version of Hadoop

Hadoop HDFS version Command Usage:

```
hadoop version
```

2. mkdir

To create a directory. In Hadoop *dfs* there is no home directory by default. So let's first create it.

Hadoop HDFS mkdir Command Usage:

```
hadoop fs –mkdir /path/directory_name
```

3. ls

This command is used to list all the files. Use *lsr* for a recursive approach. It is useful when we want a hierarchy of a folder.

Hadoop HDFS ls Command Usage:

```
hadoop fs -ls /path
```

Using the ls command, we can check for the directories in HDFS.

```
hadoop fs –ls /
```

4. put

Hadoop HDFS put Command Usage:

```
haoop fs -put <localsrc> <dest>
```

Hadoop HDFS put Command Description:

The Hadoop fs shell command put is similar to the copyFromLocal, which copies files or directory from the local filesystem to the destination in the Hadoop filesystem.

5. copyFromLocal

Hadoop HDFS copyFromLocal Command Usage:

```
hadoop fs -copyFromLocal <localsrc> <hdfs destination>
```

6. get

Hadoop HDFS get Command Usage:

```
hadoop fs -get <src> <localdest>
```

Hadoop HDFS get Command Description:

The Hadoop fs shell command get copies the file or directory from the Hadoop file system to the local file system.

7. copyToLocal

Hadoop HDFS copyToLocal Command Usage:

```
hadoop fs -copyToLocal <hdfs source> <localdst>
```

8. cat

Hadoop HDFS cat Command Usage:

```
hadoop fs –cat /path_to_file_in_hdfs
```

9. mv

Hadoop HDFS mv Command Usage:

```
hadoop fs -mv <src> <dest>
```

10. cp

Hadoop HDFS cp Command Usage:

```
hadoop fs -cp <src> <dest>
```

11. moveFromLocal

HDFS moveFromLocal Command Usage:

```
hadoop fs -moveFromLocal <localsrc> <dest>
```

12. moveToLocal

HDFS moveToLocal Command Usage:

```
hadoop fs -moveToLocal <src> <localdest>
```

13. tail

HDFS tail Command Usage:

```
hadoop fs -tail [-f] <file>
```

14. rm

HDFS rm Command Usage:

```
hadoop fs –rm <path>
```

15. expunge

HDFS expunge Command Usage:

```
hadoop fs -expunge
```

HDFS expunge command makes the trash empty.

16. chown

HDFS chown Command Usage:

```
hadoop fs -chown [-R] [owner] [:[group]] <path>
```

17. chgrp

HDFS chgrp Command Usage:

```
hadoop fs -chgrp <group> <path>
```

18. setrep

This command is used to change the replication factor of a file/directory in HDFS. By default it is 3 for anything which is stored in HDFS (as set in hdfs *core-site.xml*).

Example 1: To change the replication factor to 6 for *geeks.txt* stored in HDFS.

```
hdfs dfs -setrep -R -w 6 geeks.txt
```

Example 2: To change the replication factor to 4 for a directory *geeksInput* stored in HDFS.

```
hdfs dfs -setrep -R  4 /geeks
```

Note: The **-w** means wait till the replication is completed. And **-R** means recursively, we use it for directories as they may also contain many files and folders inside them.

HDFS setrep Command Usage:

```
hadoop fs -setrep <rep> <path>
```

19. du

HDFS du Command Usage:

```
hadoop fs –du –s /directory/filename
```

20. df

HDFS df Command Usage:

```
hadoop fs -df [-h] <path>
```

21. fsck

HDFS fsck Command Usage:

```
hadoop fsck <path> [ -move | -delete | -openforwrite] [-files [-blocks [-locations | -racks]]]
```

22. touch

It creates an empty file.

Syntax:

```
hadoop fs  -touchz  <file_path>
```

23. stat:

It will give the last modified time of directory or path. In short it will give stats of the directory or file.

Syntax:

```
hdfs  dfs -stat    <hdfs file>
```

Note: There are more commands in HDFS but we discussed the commands which are commonly used when working with Hadoop. You can check out the list of *dfs* commands using the following command:

```
hdfs dfs
```

```
$HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-3.3.6.jar
```

Running Built-in MapReduce Functions in Hadoop

Objective

Learn to use and execute built-in MapReduce functions in Hadoop for common data processing tasks.

Topics Covered

- Overview of Built-in MapReduce Functions
- Running Built-in MapReduce Jobs
- Analyzing Output of Built-in MapReduce Jobs

Lab Activities

1. **Introduction to Built-in MapReduce Functions:**
    - Overview of common built-in MapReduce functions like WordCount, AggregateWordCount, and Terasort.
    - Understand use cases for each of these functions.
2. **Running Built-in MapReduce Jobs:**
    - Learn how to run built-in MapReduce jobs using the Hadoop command line.
    - Execute the WordCount example using built-in MapReduce functions.
    - Execute the AggregateWordCount example using built-in MapReduce functions.
    - Execute the Terasort example using built-in MapReduce functions.
3. **Analyzing Output of Built-in MapReduce Jobs:**
    - Analyze the output of each of the built-in MapReduce jobs.
    - Understand the results and learn how to interpret the output for further processing.

Lab Activities in Detail

Activity 1: Running WordCount Example

1. **Upload Data to HDFS:**

   Create a directory in HDFS for input data:

   ```
   hdfs dfs -mkdir /user/your_username/input
   ```

   Upload a sample text file to the HDFS directory:

   ```
   hdfs dfs -put sample.txt /user/your_username/input
   ```

2. **Run WordCount Job:**

   Execute the built-in WordCount MapReduce job:

   ```
   hadoop jar $HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-*.jar wordcount /user/your_username/input /user/your_username/output_wordcount
   ```

3. **View the Output:**

   Check the output directory:

   ```
   hdfs dfs -ls /user/your_username/output_wordcount
   ```

   View the results:

   ```
   hdfs dfs -cat /user/your_username/output_wordcount/part-r-00000
   ```

Activity 2: Running AggregateWordCount Example

1. **Run AggregateWordCount Job:**

   Execute the built-in AggregateWordCount MapReduce job:

   ```
   hadoop jar $HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-*.jar aggregatewordcount /user/your_username/input /user/your_username/output_aggregatewordcount 2
   ```

2. **View the Output:**

   Check the output directory:

   ```
   hdfs dfs -ls /user/your_username/output_aggregatewordcount
   ```

View the results:

```
hdfs dfs -cat /user/your_username/output_aggregatewordcount/part-r-00000
```

Activity 3: Running Terasort Example

1. **Generate Teragen Data:**

   Generate data using Teragen:

   ```
   hadoop jar $HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-*.jar teragen 100000 /user/your_username/terasort_input
   ```

2. **Run Terasort Job:**

   Execute the Terasort MapReduce job:

   ```
   hadoop jar $HADOOP_HOME/share/hadoop/mapreduce/hadoop-mapreduce-examples-*.jar terasort /user/your_username/terasort_input /user/your_username/terasort_output
   ```

3. **View the Output:**

   Check the output directory:

   ```
   hdfs dfs -ls /user/your_username/terasort_output
   ```

   View the results:

   ```
   hdfs dfs -cat /user/your_username/terasort_output/part-r-00000
   ```

Assessment

- Practical exercise: Run and analyze the output of WordCount, AggregateWordCount, and Terasort examples using built-in MapReduce functions.