3. HDFS

Gaurav Ojha
VisFac @ MPSTME, Mumbai

Lecture Notes: Introduction to Hadoop Distributed File System (HDFS)

Why learn HDFS?

- Learning Hadoop Distributed File System (HDFS) is essential for individuals working in the field of big data, distributed systems, and data engineering.
- Here are some reasons why learning HDFS can be highly beneficial:

1. **Distributed Storage and Processing:** HDFS is designed to store and manage vast amounts of data across a cluster of commodity hardware. It allows for horizontal scaling, enabling organizations to handle massive datasets efficiently.
2. **Hadoop Ecosystem Integration:** HDFS is a core component of the Apache Hadoop ecosystem, which includes various tools like Apache MapReduce, Apache Hive, Apache Spark, etc. Understanding HDFS is crucial for leveraging the full potential of these big data processing frameworks.
3. **Fault Tolerance:** HDFS provides high fault tolerance by replicating data blocks across multiple nodes in the cluster. If a node fails, the data can still be accessed from other healthy replicas, ensuring data reliability.
4. **Cost-Effective Storage:** HDFS runs on commodity hardware, making it a cost-effective solution for storing large volumes of data compared to traditional storage systems.
5. **Data Replication and Backup:** HDFS's replication feature allows data to be stored redundantly across nodes. This redundancy not only ensures fault tolerance but also acts as a backup mechanism for data protection.
6. **Data Locality:** HDFS is designed to bring computation closer to data. This data locality feature helps reduce network traffic and improves the overall performance of data processing tasks.
7. **Scalability:** As data volume grows, HDFS can scale by adding more nodes to the cluster. This horizontal scalability enables organizations to handle an ever-increasing amount of data without significant changes to the system.
8. **Parallel Data Processing:** By integrating HDFS with Hadoop MapReduce or other processing frameworks like Apache Spark, data processing tasks can be split into smaller chunks and processed in parallel across the cluster, leading to faster analysis.

9. **Data Governance and Security:** HDFS provides access controls and permissions to manage data security effectively. Understanding these features is crucial for ensuring proper data governance.
10. **Industry Adoption:** Hadoop and HDFS have seen widespread adoption across various industries, including tech, finance, healthcare, retail, and more. Learning HDFS opens up job opportunities in companies dealing with big data and analytics.

HDFS Terminal Commands and Examples:

1. **hadoop fs -ls**
   - Description: List the contents of a directory in HDFS.
   - Example:

     ```
     hadoop fs -ls /user/myuser/data
     ```

     Output:

     ```
     Found 3 items
     -rw-r--r--   3 myuser supergroup     1048576 2023-07-26 09:00
     /user/myuser/data/file1.txt
     drwxr-xr-x   - myuser supergroup           0 2023-07-26 09:05
     /user/myuser/data/directory/
     -rw-r--r--   3 myuser supergroup      524288 2023-07-26 09:10
     /user/myuser/data/file2.txt
     ```

2. **hadoop fs -put**
   - Description: Upload a file from the local file system to HDFS.
   - Example:

     ```
     hadoop fs -put localfile.txt /user/myuser/data/
     ```

3. **hadoop fs -get**
   - Description: Download a file from HDFS to the local file system.
   - Example:

     ```
     hadoop fs -get /user/myuser/data/file1.txt /path/to/local/
     ```

4. **hadoop fs -mkdir**
   - Description: Create a directory in HDFS.
   - Example:

     ```
     hadoop fs -mkdir /user/myuser/output
     ```

5. **hadoop fs -rm**
   - Description: Delete a file or directory from HDFS.

- Example:

```
hadoop fs -rm /user/myuser/data/unwanted_file.txt
```

6. **hadoop fs -mv**
   - Description: Move a file or directory within HDFS.
   - Example:

```
hadoop fs -mv /user/myuser/data/file2.txt /user/myuser/data/directory/
```

7. **hadoop fs -cp**
   - Description: Copy files or directories within HDFS.
   - Example:

```
hadoop fs -cp /user/myuser/data/file1.txt /user/myuser/data/directory/
```

8. **hadoop fs -cat**
   - Description: Display the contents of a file in the terminal.
   - Example:

```
hadoop fs -cat /user/myuser/data/file1.txt
```

9. **hadoop fs -tail**
   - Description: Display the last kilobyte of a file to the terminal.
   - Example:

```
hadoop fs -tail /user/myuser/data/big_log_file.log
```

10. **hadoop fs -du**
    - Description: Display the size of files and directories in HDFS.
    - Example:

```
hadoop fs -du /user/myuser/data/
```

Exercise:

You are a data engineer working with Hadoop and HDFS. Your task is to perform several file operations using Hadoop commands to manage and manipulate data files. Follow the steps below to complete the exercise:

1. Create two local text files, Emp1.txt and Emp2.txt, using the VI editor on your Cloudera Quickstart VM. Copy the following data into each file:

Emp1.txt:

```
100,AAA,IT,2000
200,BBB,IT,3000
300,CCC,Admin,2500
400,DDD,Admin,500
```

Emp2.txt:

```
500,AAA,IT,2000
600,BBB,IT,3000
700,CCC,Admin,2500
800,DDD,Admin,500
```

2. Create an HDFS directory named "emp."
3. Check if the "emp" folder has been successfully created in HDFS.
4. Copy both Emp1.txt and Emp2.txt from the local system to the HDFS "emp" folder.
5. Verify the content of the files in the HDFS "emp" folder.
6. Merge the contents of Emp1.txt and Emp2.txt into a single file named "Emp.txt" on the local system (Cloudera Quickstart VM).
7. Use the Hadoop command to put the merged "Emp.txt" file from the local system into the HDFS "/user/data" directory.

Exercise Solution:

Step 1: Create Emp1.txt and Emp2.txt using the VI editor:

```
# Create and edit Emp1.txt
vi Emp1.txt

# Copy the following data into Emp1.txt
100,AAA,IT,2000
200,BBB,IT,3000
300,CCC,Admin,2500
400,DDD,Admin,500

# Save and exit VI editor (Press ESC, then type :wq, and press Enter)

# Create and edit Emp2.txt
vi Emp2.txt

# Copy the following data into Emp2.txt
500,AAA,IT,200

0
```

```
600,BBB,IT,3000
700,CCC,Admin,2500
800,DDD,Admin,500

# Save and exit VI editor (Press ESC, then type :wq, and press Enter)
```

Step 2: Create an HDFS directory named "emp":

```
hadoop fs -mkdir /user/data/emp
```

Step 3: Check if the "emp" folder has been successfully created in HDFS:

```
hadoop fs -ls /user/data/
```

Step 4: Copy Emp1.txt and Emp2.txt to the HDFS "emp" folder:

```
hadoop fs -put Emp1.txt /user/data/emp/
hadoop fs -put Emp2.txt /user/data/emp/
```

Step 5: Verify the content of the files in the HDFS "emp" folder:

```
hadoop fs -cat /user/data/emp/Emp1.txt
hadoop fs -cat /user/data/emp/Emp2.txt
```

Step 6: Merge the contents of Emp1.txt and Emp2.txt into a single file named "Emp.txt" on the local system:

```
hadoop fs -getmerge /user/data/emp/ /tmp/Emp.txt
```

Step 7: Use the Hadoop command to put the merged "Emp.txt" file from the local system into the HDFS "/user/data" directory:

```
hadoop fs -put /tmp/Emp.txt /user/data/
```

Now, you have successfully completed the exercise, and the merged "Emp.txt" file is stored in the HDFS "/user/data" directory.

Additional Commands for Practice:

1. Check the content of the merged file "Emp.txt" in HDFS:

   ```
   hadoop fs -cat /user/data/Emp.txt
   ```

2. Change permissions for the "Emp.txt" file in HDFS:

   ```
   # Assuming you are the superuser 'hdfs'
   hadoop fs -chmod 644 /user/data/Emp.txt
   ```

3. Verify the changed permissions of the "Emp.txt" file:

```
hadoop fs -ls /user/data/Emp.txt
```

4. Get the storage size of the "/user/data" directory in HDFS:

```
hadoop fs -du -s /user/data
```

5. Create a new directory named "output" in HDFS:

```
hadoop fs -mkdir /user/data/output
```

6. Move the "Emp.txt" file to the "output" directory in HDFS:

```
hadoop fs -mv /user/data/Emp.txt /user/data/output/
```

7. Verify that the file is moved to the "output" directory:

```
hadoop fs -ls /user/data/output/
```

Now you have completed additional commands to work with HDFS, including changing permissions, getting storage size, creating a new directory, and moving files within HDFS. These exercises help you gain confidence in managing data using Hadoop Distributed File System (HDFS) and familiarize yourself with various Hadoop terminal commands.