

实验二 熟悉常用的 HDFS 操作

19084129-李奕澄

实验目的

- (1)理解 HDFS 在 Hadoop 体系结构中的角色：
- (2)熟练使用 HDFS 操作常用的 Shell 命令：
- (3)熟悉 HDFS 操作常用的 Java API。 .

实验环境

Ubuntu 18.04

Hadoop 3.1.3

实验内容

1.利用 Hadoop 提供的 Shell 命令完成下列任务。

- (1) 向 HDFS 中上传任意文本文件，如果指定的文件在 HDFS 中已经存在，由用户指定是追加到原有文件末尾还是覆盖原有的文件；

```
Hadoop@yunhai$~$hadoop fs -put /home/hadoop/test.txt/usr/local/hadoop
Hadoop@yunhai:~$ hadoop fs -test -e /usr/local/hadoop/test.txt
Warning:SHADOOP HOME is deprecated.

Hadoop@yunhai:~$ echo $
0
```

- 2) 从 HDFS 中下载指定文件，如果本地文件与要下载的文件名称相同，则自动对下载的文件重命名；

```
Hadoop@yunhai:~$if s(hadoop fs -test -e /home/hadoop/test.txt);then $(hadoop
fs-copyToLocal /usr/local/hadoop/test.txt /home/hadoop/test.txt);else $(hadoop
fs -copyToLocal /usr/local/hadoop/test.txt /home/hadoop/test2.txt);fi
Warning: $HADOOP HOME is deprecated.

Warning: $HADOOP_HOME is deprecated.
```

(3) 将 HDFS 中指定文件的内容输出到终端中;

```
Hadoop@yunhai:~$ hadoop fs -cat /usr/local/hadoop/test.txt
Warning: $HADOOP_HOME is deprecated.

asjdyaicawediasuca
```

(4) 显示 HDFS 中指定的文件的读写权限、大小、创建时间、路径等信息;

```
Hadoop@yunhai:~$ hadoop fs -ls -h /usr/local/hadoop/test.txt
Warning:$HADOOP_HOME is deprecated.
ls: Cannot access -h: No such file or directory.
Found 1 items
-rw-r--r--      1 hadoop supergroup    62 2022-5-13  16:26  /usr/local/hadoop/test.txt
```

(5) 给定 HDFS 中某一个目录, 输出该目录下的所有文件的读写权限、大小、创建时间、路径等信息, 如果该文件是目录, 则递归输出该目录下所有文件相关信息;

```
Hadoop@yunhai:~$ hadoop fs -ls -R -h /usr/local/hadoop
Warning: $HADOOP_HOME is deprecated.

ls: Cannot access -R: No such file or directory.
ls: Cannot access -R: No such file or directory

Found 2 items
drwxr-xr-x      - hadoop supergroup    0 2022-05-13 17:21  /usr/local/hadoop/hadoop_tmp
-rw-f-"r-"      1 hadoop supergroup    62 2022-05-13 17:21  /usr/local/hadoop/test.txt
```

(6) 提供一个 HDFS 内的文件的路径, 对该文件进行创建和删除操作。如果文件所在目录不存在, 则自动创建目录;

```
Hadoop@yunhai:~$ if s(hadoop fs -test -d /usr/local/hadoop/test); then $(hadoop fs -touchz /usr/local/hadoop/test/testi.txt); else $(hadoop fs -mkdir -p /usr/local/hadoop/test && hadoop fs -touchz /usr/local/hadoop/test/test1.txt); fi
Warning:SHADOOP_HOME is deprecated.

Hadoop@yunhai:~$ hadoop fs -rm /usr/local/hadoop/test/test1.txt
Warning:SHADOOP_HOME is deprecated.

Deleted hdfs://locathost:8020/usr/local/hadoop/test/test1.txt
```

(7) 提供一个 HDFS 的目录的路径, 对该目录进行创建和删除操作。创建目录时, 如果目录文件所在目录不存在则自动创建相应目录; 删除目录时, 由用户指定当该目录不为空时是否还删除该目录;

```
Hadoop@yunhai:~$ hadoop fs -rmr /usr/local/hadoop/test
Warning:SHADOOP_HOME is deprecated.

Deleted hdfs://localhost:8020/usr/local/hadoop/test
```

(8) 删除 HDFS 中指定的文件;

```
Hadoop@yunhai:~$ hadoop fs -rm /usr/local/hadoop/test.txt
Warning:SHADOOP_HOME is deprecated.

Deleted hdfs://localhost:8020/usr/local/hadoop/test.txt
```

(9) 在 HDFS 中, 将文件从源路径移动到目的路径。

```
Hadoop@yunhai:~$ hadoop fs -mv /usr/local/hadoop/test.txt /usr/local/hadoop/hadoop
/tmp/test.txt
```

2. 编程实现一个类“MyFSDataInputStream”,

该类继承“org.apache.hadoop.fs.FSDataInputStream”,

要求如下: 实现按行读取 HDFS 中指定文件的方法“readLine()”, 如果读到文件末尾, 则返回空, 否则返回文件一行的文本

```
import org.apache.hadoop.fs.*;
```

```
import java.io.*;
```

```
public class HDFSApi{
```

```
    public static boolean test(Configuration conf,String path)throws IOException
```

```
    {
        FileSystem fs=FileSystem.get(conf);
```

```
        return fs.exists(new Path(path));
    }
```

```
    public static void copyFromLocalFile(Configuration conf,String localFilePath,String
remoteFilePath)throws IOException{
```

```
        FileSystem fs=FileSystem.get(conf);
```

```
        Path localPath=new Path(localFilePath);
```

```
        Path remotePath=new Path(remoteFilePath);
```

```
        fs.copyFromLocalFile(false,true,localPath,remotePath);
```

```
        fs.close();
```

```
    public static void appendToFile(Configuration conf,String
```

```
localFilePath,StringremoteFilePath)throws IOException{
```

```
        FileSystem fs=FileSystem.get(conf);
```

```
        Path remotePath=new Path(remoteFilePath);
```

```
        FileInputStream in=new FileInputStream(localFilePath);
```

```
        FSDataOutputStream out=fs.append(remotePath);
```

```
        byte[] data=new byte[1024];
```

```
        int read=-1;
```

```
        while ((read=in.read(data))>0){
```

```
            out.write(data,0,read);
```

```
        }
```

```
        out.close();
```

```
        in.close();
```

```
        fs.close();
    }
```

```

}
public static void main(String[]args){
    Configuration conf=new Configuration();
    conf.set("fs.default.name","hdfs://localhost:9000");
    String localFilePath ="/home/hadoop/text.txt";
    String remoteFilePath="/user/hadoop/text.txt":

    String choice-"append";
    String choice="overwrite";
    try{
        Boolean fileExists false:
        if(HDFSApi.test(conf,remoteFilePath)){
            fileExists true:
            System.out.println(remoteFilePath+"已存在.");
        }else{
            System.out.println(remoteFilePath+"不存在");
        }
        if(fileExists){
            HDFSApi.copyFromLocalFile(conf,localFilePath,remoteFilePath);
            System.out.println(localFilePath+"已上传至"+remoteFilePath);
        }else if(choice.equals("overwrite")){
            HDFSApi.copyFromLocalFile(conf,localFilePath,remoteFilePath);
            System.out.println(localFilePath+"已覆盖"+remoteFilePath);
        }else if(choice.equals("append")){
            HDFSApi.appendToFile(conf,localFilePath,remoteFilePath);
            System.out.println(localFilePath+"已追加至"+remoteFilePath);
        }
    }catch (Exception e){
        e.printStackTrace();
    }
}

```

3.查看 Java 帮助手册或其它资料,

用“java.net.URL”和“org.apache.hadoop.fs.FsURLStreamHandlerFactory”编程完成输出 HDFS 中指定文件的文本到终端中。

```

public static void Write(String remoteFilePath)throws IOException
InputStream in null;
try{
    in new URL("hdfs","localhost",8020,remoteFilePath).openstream();
    IOUtils.copyBytes(in,System.out,4096,false);l
}finally{
    IOUtils.closestream(in);
}
}

```