Chaining and Scripting with HDFS Commands

When you work a lot with HDFS commands, there are two techniques that you will likely find very useful: chaining and scripting.

Chaining HDFS Commands

If you've worked much with the command line, you might be familiar with using pipes (|) to chain commands, or using redirection to push results to a different location (such as to a file instead of to the screen). You can do this with HDFS shell commands as well.

You can achieve many practical tasks by combining HDFS shell commands with piping or redirection. Two examples are described below.

Viewing the First Few Lines of a File

To view only the first few lines of a text file stored in HDFS, you can pipe the output of the hdfs dfs -cat command to the head command:

```
$ hdfs dfs -cat /path/to/file.txt | head
```

You can ignore the message that says “Unable to write to the output stream.” This happens because the hdfs dfs -cat command outputs more data than the head command inputs.

Loading Data Without the Header Line

Instead of using the skip.header.line.count table property to ignore the header line in text files, you can copy everything *except* the header line when you put a file into HDFS:

```
$ tail -n +2 source_file.txt | hdfs dfs -put - /path/to/destination_file.txt
```

In the above command, the hyphen or dash character (-) after -put tells the HDFS shell application to take the output of the tail command before the pipe and load that into HDFS.

To remove the header line when copying a text file in the *opposite* direction (from HDFS to the local file system) you would run a different command, this time using the output redirection operator (>) to store the output of the tail command in a file:

```
$ hdfs dfs -cat /path/to/source_file.txt | tail -n +2 > destination_file.txt
```

Using Commands in Scripts

You can also use commands like this in scripts. This is particularly helpful when you automate tasks that involve managing files in HDFS. In shell scripts, you can use hdfs dfs commands just as you normally would other shell commands. Scripts for other languages, such as Python and R, typically can invoke shell commands using commands specific to the language. For example, in Python you can use the os or subprocess modules and use a call such as subprocess.call(). In R, you can use system().