Apache Hadoop –

A course for undergraduates

Homework Labs, Lecture 1

Copyright © 2010-2014 Cloudera, Inc. All rights reserved.
Not to be reproduced without prior written consent.

General Notes on Homework Labs

Homework for this course is completed using the course Virtual Machine (VM) which runs the CentOS 6.3 Linux distribution. This VM has CDH (Cloudera's Distribution, including Apache Hadoop) installed in pseudo-distributed mode. Pseudo-distributed mode is a method of running Hadoop whereby all Hadoop daemons run on the same machine. It is, essentially, a cluster consisting of a single machine. It works just like a larger Hadoop cluster, the key difference (apart from speed, of course!) is that the block replication factor is set to one, since there is only a single DataNode available.

Getting Started

1. The VM is set to automatically log in as the user `training`. Should you log out at any time, you can log back in as the user `training` with the password `training`.

Working with the Virtual Machine

1. Should you need it, the root password is `training`. You may be prompted for this if, for example, you want to change the keyboard layout. In general, you should not need this password since the `training` user has unlimited sudo privileges.

2. In some command-line steps in the labs, you will see lines like this:

```
$ hadoop fs -put shakespeare  \
/user/training/shakespeare
```

   The dollar sign ($) at the beginning of each line indicates the Linux shell prompt. The actual prompt will include additional information (e.g., `[training@localhost workspace]$`) but this is omitted from these instructions for brevity.

   The backslash (`\`) at the end of the first line signifies that the command is not completed, and continues on the next line. You can enter the code exactly as

Copyright © 2010-2014 Cloudera, Inc. All rights reserved.
Not to be reproduced without prior written consent.

shown (on two lines), or you can enter it on a single line. If you do the latter, you should *not* type in the backslash.

3. Although many students are comfortable using UNIX text editors like vi or emacs, some might prefer a graphical text editor. To invoke the graphical editor from the command line, type `gedit` followed by the path of the file you wish to edit. Appending `&` to the command allows you to type additional commands while the editor is still open. Here is an example of how to edit a file named `myfile.txt`:

```
$ gedit myfile.txt &
```

Copyright © 2010-2014 Cloudera, Inc. All rights reserved.
Not to be reproduced without prior written consent.

Lab: Using HDFS

Files Used in This Exercise:

Data files (local)

`~/training_materials/developer/data/shakespeare.tar.gz`

`~/training_materials/developer/data/access_log.gz`

In this lab you will begin to get acquainted with the Hadoop tools. You will manipulate files in HDFS, the Hadoop Distributed File System.

Set Up Your Environment

1. Before starting the labs, run the course setup script in a terminal window:

```
$ ~/scripts/developer/training_setup_dev.sh
```

Hadoop

Hadoop is already installed, configured, and running on your virtual machine.

Most of your interaction with the system will be through a command-line wrapper called `hadoop`. If you run this program with no arguments, it prints a help message. To try this, run the following command in a terminal window:

```
$ hadoop
```

The `hadoop` command is subdivided into several subsystems. For example, there is a subsystem for working with files in HDFS and another for launching and managing MapReduce processing jobs.

Copyright © 2010-2014 Cloudera, Inc. All rights reserved.
Not to be reproduced without prior written consent.

Step 1: Exploring HDFS

The subsystem associated with HDFS in the Hadoop wrapper program is called `FsShell`. This subsystem can be invoked with the command `hadoop fs`.

1. Open a terminal window (if one is not already open) by double-clicking the Terminal icon on the desktop.

2. In the terminal window, enter:

```
$ hadoop fs
```

   You see a help message describing all the commands associated with the `FsShell` subsystem.

3. Enter:

```
$ hadoop fs -ls /
```

   This shows you the contents of the root directory in HDFS. There will be multiple entries, one of which is `/user`. Individual users have a "home" directory under this directory, named after their username; your username in this course is `training`, therefore your home directory is `/user/training`.

4. Try viewing the contents of the `/user` directory by running:

```
$ hadoop fs -ls /user
```

   You will see your home directory in the directory listing.

5. List the contents of your home directory by running:

```
$ hadoop fs -ls /user/training
```

   There are no files yet, so the command silently exits. This is different from running `hadoop fs -ls /foo`, which refers to a directory that doesn't exist. In this case, an error message would be displayed.

Copyright © 2010-2014 Cloudera, Inc. All rights reserved.
Not to be reproduced without prior written consent.

Note that the directory structure in HDFS has nothing to do with the directory structure of the local filesystem; they are completely separate namespaces.

Step 2: Uploading Files

Besides browsing the existing filesystem, another important thing you can do with `FsShell` is to upload new data into HDFS.

1. Change directories to the local filesystem directory containing the sample data we will be using in the homework labs.

```
$ cd ~/training_materials/developer/data
```

   If you perform a regular Linux `ls` command in this directory, you will see a few files, including two named `shakespeare.tar.gz` and `shakespeare-stream.tar.gz`. Both of these contain the complete works of Shakespeare in text format, but with different formats and organizations. For now we will work with `shakespeare.tar.gz`.

2. Unzip `shakespeare.tar.gz` by running:

```
$ tar zxvf shakespeare.tar.gz
```

   This creates a directory named `shakespeare/` containing several files on your local filesystem.

3. Insert this directory into HDFS:

```
$ hadoop fs -put shakespeare /user/training/shakespeare
```

   This copies the local `shakespeare` directory and its contents into a remote, HDFS directory named `/user/training/shakespeare`.

4. List the contents of your HDFS home directory now:

```
$ hadoop fs -ls /user/training
```

   You should see an entry for the `shakespeare` directory.

Copyright © 2010-2014 Cloudera, Inc. All rights reserved.
Not to be reproduced without prior written consent.

5. Now try the same `fs -ls` command but without a path argument:

```
$ hadoop fs -ls
```

You should see the same results. If you don't pass a directory name to the `-ls` command, it assumes you mean your home directory, i.e. `/user/training`.

> **Relative paths**
>
> If you pass any relative (non-absolute) paths to `FsShell` commands (or use relative paths in MapReduce programs), they are considered relative to your home directory.

6. We will also need a sample web server log file, which we will put into HDFS for use in future labs. This file is currently compressed using GZip. Rather than extract the file to the local disk and then upload it, we will extract and upload in one step. First, create a directory in HDFS in which to store it:

```
$ hadoop fs -mkdir weblog
```

7. Now, extract and upload the file in one step. The `-c` option to `gunzip` uncompresses to standard output, and the dash (`-`) in the `hadoop fs -put` command takes whatever is being sent to its standard input and places that data in HDFS.

```
$ gunzip -c access_log.gz \
| hadoop fs -put - weblog/access_log
```

8. Run the `hadoop fs -ls` command to verify that the log file is in your HDFS home directory.

9. The access log file is quite large – around 500 MB. Create a smaller version of this file, consisting only of its first 5000 lines, and store the smaller version in HDFS. You can use the smaller version for testing in subsequent labs.

Copyright © 2010-2014 Cloudera, Inc. All rights reserved.
Not to be reproduced without prior written consent.

```
$ hadoop fs -mkdir testlog
$ gunzip -c access_log.gz | head -n 5000 \
| hadoop fs -put - testlog/test_access_log
```

Step 3: Viewing and Manipulating Files

Now let's view some of the data you just copied into HDFS.

1. Enter:

```
$ hadoop fs -ls shakespeare
```

This lists the contents of the `/user/training/shakespeare` HDFS directory, which consists of the files `comedies`, `glossary`, `histories`, `poems`, and `tragedies`.

2. The `glossary` file included in the compressed file you began with is not strictly a work of Shakespeare, so let's remove it:

```
$ hadoop fs -rm shakespeare/glossary
```

Note that you *could* leave this file in place if you so wished. If you did, then it would be included in subsequent computations across the works of Shakespeare, and would skew your results slightly. As with many real-world big data problems, you make trade-offs between the labor to purify your input data and the precision of your results.

3. Enter:

```
$ hadoop fs -cat shakespeare/histories | tail -n 50
```

This prints the last 50 lines of *Henry IV, Part 1* to your terminal. This command is handy for viewing the output of MapReduce programs. Very often, an individual output file of a MapReduce program is very large, making it inconvenient to view the entire file in the terminal. For this reason, it's often a

Copyright © 2010-2014 Cloudera, Inc. All rights reserved.
Not to be reproduced without prior written consent.

good idea to pipe the output of the `fs -cat` command into `head`, `tail`, `more`, or `less`.

4. To download a file to work with on the local filesystem use the `fs -get` command. This command takes two arguments: an HDFS path and a local path. It copies the HDFS contents into the local filesystem:

```
$ hadoop fs -get shakespeare/poems ~/shakepoems.txt
$ less ~/shakepoems.txt
```

Other Commands

There are several other operations available with the `hadoop fs` command to perform most common filesystem manipulations: `mv`, `cp`, `mkdir`, etc.

1. Enter:

```
$ hadoop fs
```

This displays a brief usage report of the commands available within `FsShell`. Try playing around with a few of these commands if you like.

This is the end of the lab.

Copyright © 2010-2014 Cloudera, Inc. All rights reserved.
Not to be reproduced without prior written consent.