DSCI 551 – HW2 (Fall 2020)

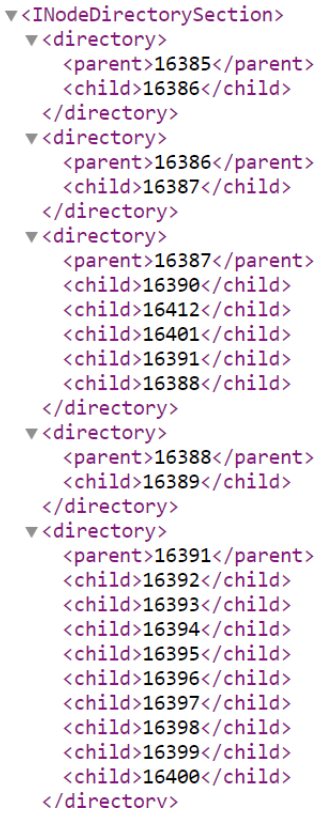
100 points, Due 10/9

In this homework, we will explore the metadata stored in the namenode of HDFS. You can obtain such metadata by using the Offline Image Viewer (oiv) tool provided by Hadoop (https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsImageViewer.html).

For example,

<Your Hadoop-installation-dir>/bin/hdfs oiv -i /tmp/hadoop-ec2-user/dfs/name/current/fsimage_0000000000000000564 -o fsimage564.xml -p XML

will export the metadata stored in the specified fsimage (file system image) to an XML file called fsimage546.xml. **Note that your fsimage file names may be different than shown here. Note also that your program will be tested using additional fsimage files other than a sample provided to you.**



Fsimage has an INodeSection listing metadata about each inode and an INodeDirectorySection describing the directory structure, as show above. Note that id of inode is its inumber; and the directory nodes are represented by their inumbers, e.g., 16385. Note also that the directory whose inode element has an empty <name/> sub-element is the root directory.

Your task is to develop a Python program "xmlhdfs.py" that simulates the way you interact with the "hdfs dfs -ls" command. The program should be invoked as follows.

python xmlhdsf.py <fsimage> <dir>

where <fsimage> is a HDFS image file in XML and <dir> is a directory (or file) in the image. The command should report:

1. **The number of items (files/subdirectories) found in the <dir> directory**
2. **information about each item**

(one line at a time, with different pieces of information separated by a tab). It should also report the error if no specified file or directory was found. The output should be in the same format as reported by hdfs, shown below.

The **information about an item** include

1) type indicator (d for directory or – for file) and mode bits (e.g., rwxr-xr-x),
2) number of replica (- for directory, number, e.g., 1 for file),
3) user id (ec2-user),
4) group id (supergroup),
5) file size (0 for directory, the actual size for file, e.g., 874 bytes for core-site.xml),
6) modification date, modification time,
7) file/directory name.

Note that the size of file can be computed by summing up the numBytes of all its blocks (see below). Note also that modification time may be obtained by transforming the mtime (in milliseconds) value given in the fsimage file. For example,

```
import datetime
datetime.datetime.fromtimestamp(1581874756018/1e3)
```

```
▼<inode>
   <id>16393</id>
   <type>FILE</type>
   <name>core-site.xml</name>
   <replication>1</replication>
   <mtime>1581874756018</mtime>
   <atime>1581874755988</atime>
   <preferredBlockSize>134217728</preferredBlockSize>
   <permission>ec2-user:supergroup:0644</permission>
  ▼<blocks>
   ▼<block>
      <id>1073741828</id>
      <genstamp>1004</genstamp>
      <numBytes>874</numBytes>
    </block>
   </blocks>
   <storagePolicyId>0</storagePolicyId>
</inode>
```

Here are some sample output from “hdfs dfs -ls” commands.

```
$ hdfs dfs -ls /user/ec2-user
Found 2 items
drwxr-xr-x   - ec2-user supergroup          0 2020-09-15 06:05 /user/ec2-user/dsci551
drwxr-xr-x   - ec2-user supergroup          0 2020-09-06 22:48 /user/ec2-user/input

$ hdfs dfs -ls /user/ec2-user1
ls: /user/ec2-user1': No such file or directory


$ hdfs dfs -ls /user/ec2-user/dsci551/input
Found 9 items
-rw-r--r--  1  ec2-user supergroup      620    2020-09-22 18:59  /user/ec2-user/dsci551/input/httpfs-site.xml
-rw-r--r--  1  ec2-user supergroup      682    2020-09-22 18:59  /user/ec2-user/dsci551/input/kms-site.xml
-rw-r--r--  1  ec2-user supergroup      690    2020-09-22 18:59  /user/ec2-user/dsci551/input/yarn-site.xml
-rw-r--r--  1  ec2-user supergroup      758    2020-09-22 18:59  /user/ec2-user/dsci551/input/mapred-site.xml
-rw-r--r--  1  ec2-user supergroup      849    2020-09-22 18:59  /user/ec2-user/dsci551/input/hdfs-site.xml
-rw-r--r--  1  ec2-user supergroup      866    2020-09-22 18:59  /user/ec2-user/dsci551/input/core-site.xml
-rw-r--r--  1  ec2-user supergroup     3518    2020-09-22 18:59  /user/ec2-user/dsci551/input/kms-acls.xml
-rw-r--r--  1  ec2-user supergroup     8260    2020-09-22 18:59  /user/ec2-user/dsci551/input/ scheduler.xml
-rw-r--r--  1  ec2-user supergroup    11392    2020-09-22 18:59  /user/ec2-user/dsci551/input/3adoop-policy.xml


$hdfs dfs -ls /user/ec2-user/dsci551/input/core-site.xml
-rw-r—r--   1 ec2-user supergroup   874 2020-09-15 06:05 /user/ec2-user/dsci551/input/core-site.xml
```

Details about the “ls” command in HDFS can be found at:

https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/FileSystemShell.html#ls

Output: You have to output the values in a human readable table format as the -ls command output. Align the column values to the right. To align the content in a column, you may refer to:

https://docs.python.org/2/library/string.html#format-string-syntax

Programming Environment: Python 3

Packages permitted for this homework: lxml and datetime

Submission: please include xmlhdsf.py in one folder and compress it as Firstname_Lastname_hw2.zip