Homework 2: HDFS and XML/XPath

DSCI 551 – Spring 2025

Due: 11:59pm, February 24, 2025, Monday

Points: 100

HDFS file system image may be exported using "hdfs oiv" facility into CSV format. For example, these are the steps used for the export.

- cd /tmp/hadoop-ubuntu/dfs/name/current
- ~/hadoop-3.4.1/bin/hdfs oiv -i fsimage_0000000000000002402 -o fsimage2402.csv -p Delimited -delimiter ","

Note that fsimage_0000000000000002402 is a particular fsimage file. The file name may be different on your installation. The above command will export the fsimage file into the fsimage2402.csv (comma-separated) file. You can see more details on the "hdfs oiv" usage by executing:

- ~/hadoop-3.4.1/bin/hdfs oiv --help

Path,Replication,ModificationTime,AccessTime,PreferredBlockSize,BlocksCount,FileSize,NSQUOTA,DSQUOTA,Permission,UserName,GroupName

/,0,2025-02-05 23:20,1970-01-01 00:00,0,0,0,9223372036854775807,-1,drwxr-xr-x,ubuntu,supergroup

/home,0,2025-02-05 23:19,1970-01-01 00:00,0,0,0,-1,-1,drwxr-xr-x,ubuntu,supergroup

/home/ubuntu,0,2025-02-05 23:19,1970-01-01 00:00,0,0,0,-1,-1,drwxr-xr-x,ubuntu,supergroup

/user,0,2025-02-05 23:20,1970-01-01 00:00,0,0,0,-1,-1,drwxr-xr-x,ubuntu,supergroup

/user/john,0,2025-02-05 23:27,1970-01-01 00:00,0,0,0,-1,-1,drwxr-xr-x,ubuntu,supergroup

/user/john/README.txt,1,2025-02-05 23:30,2025-02-06 00:38,134217728,1,350,0,0,-rw-r--r--,ubuntu,supergroup

Note that: the CSV file has a row for every directory and file (with complete path to the file, e.g., /user/john/README.txt) in HDFS.

Note: We are attaching csv file in case any student's hdfs oiv command is not working as to not delay assignment submission but students are encouraged to use their own fsimage.csv file

Your tasks:

1. [60 points] Write a Python program "[studentName_connvert.py]" to convert fsimage in csv format to one in XML format. Execution format:
   python3 [studentName_convert.py] <fsimage csv file name> <fsimage XML file name>

for example, python3 convert.py fsimage2402.csv fsimage2402.xml

The XML file should follow the following structure:

```
<FileSystemMetadata>
    <File>
        <Path>/</Path>
        <Replication>0</Replication>
        <ModificationTime>2025-02-05 23:20</ModificationTime>
        <AccessTime>1970-01-01 00:00</AccessTime>
        <Permission>drwxr-xr-x</Permission>
        <UserName>ubuntu</UserName>
        <GroupName>supergroup</GroupName>
    </File>
    …
     <File>
        <Path>/user/john/README.txt</Path>
        <Replication>1</Replication>
        <ModificationTime>2025-02-05 23:30</ModificationTime>
        <AccessTime>2025-02-06 00:38</AccessTime>
        <PreferredBlockSize>134217728</PreferredBlockSize>
        <BlocksCount>1</BlocksCount>
        <FileSize>350</FileSize>
        <Permission>-rw-r--r--</Permission>
        <UserName>ubuntu</UserName>
        <GroupName>supergroup</GroupName>
      </File>
</FileSystemMetadata>
```

Note that:

- It does not store NSQUOTA and DSQUOTA.
- <PreferredBlockSize>, <BlocksCount>, < FileSize> elements only appear for actual files (not directories).

2. [40 points] Write a Python program ls.py that uses the fsimage XML file (produced in task 1) to emulate the ls command. Note you need to use xpath to find the information.

   For example,

   python3 [studentName_ls.py] fsimage2402.xml /user/john/README.txt

   will output:

   -rw-r--r-- 1 ubuntu supergroup 350 2025-02-05 23:30 /user/john/README.txt

   (note that 1 is the number of replicas the file has).

   And,

   python3 ls.py fsimage2402.xml /user

   will output:

   drwxr-xr-x - ubuntu supergroup 0 2025-02-06 19:51 /user/john

   (note that for directory, it shows zero for file size, and shows '-' for the number of replicas).

   Note: - If the given file or directory does not exist, the program should output:

   No such file or directory.

   - using ls on a directory should print out all the immediate children directories and files.

Permitted libraries: pandas, lxml, time, sys

lxml Library:

Library can be installed on EC2 using

```
sudo apt install python3-lxml
```

Tutorial link:

https://lxml.de/tutorial.html

SUBMISSION DETAILS:

- Students are supposed to submit only the 2 python scripts [studentName_convert.py] and [studentName_ls.py]. Replace studentName with your own name Eg. *John_Smith_convert.py* and *John_Smith_ls.py*
- Students need not worry about exact formatting for the output.

- Do not modify any contents in the template. Just fill the template by reading the comments. Feel free to add helper functions as per your requirement.
- The test script will accept the return data same as specified in the template.
- Testing is done by test script with different test cases. So points will only be awarded if the method returns the expected result.
- You will get 0 points if the code breaks for any syntax errors or any other problems. Please test the code thoroughly before submitting.