INF 551 – Spring 2020

Homework #2: Exploring HDFS Metadata Using XML & XPath

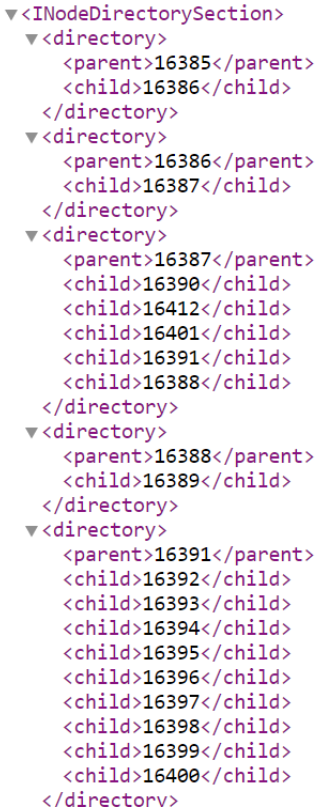
Due: March 8, Sunday (11:59pm)
100 points

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.



Fsimage has a INodeSection listing metadata about each inode and a 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.

This is an example to show how to export the metadata to xml file. You may have different fsimage files under your instance and you can use them as test files for following questions:

INF 551 – Spring 2020

Your tasks are as follows.

1. [Indexing, 40 points] Write a Python program "invert.py" that takes a fsimage file in XML, and produces an index file for the names of all files and directories. The index file should be an XML document which lists, for each token, inumbers of files and directories whose name contain the token(case insensitive). This list is often called postings list in Information Retrieval. Assume that strings are tokenized **by white spaces and hyphens**. Suffix of name(e.g. '.xml', '.txt') should be removed when creating index.

   For example, "core-site" will be tokenized to "core" and "site". And index file of "core" is as follow:

```
<index>
    <postings>
        <name>core</name>
        <inumber>12345</inumber>
        <inumber>20001</inumber>
    </postings>
    <postings>
        <name>site</name>
        <inumber>…</inumber>
    </postings>
</index>
```

   Execution format: invert.py fsimage.xml index.xml

   The program output the index in a file "index.xml". A sample fsimage file is attached. Note that your program will be tested using additional files. So, we strongly suggest that you test it out using the fsimage of your HDFS instance.

2. [Searching, 60 points] Write a Python program "search.py" that takes a fsimage file, its index file, a search query; returns the full path to the file/directory according to InodeDirectorySection in fsimage xml file whose name contains ALL keywords(case insensitive) in the query, and shows the metadata (id, type, mtime, and blocks(blocks id)) for each file/dir returned in the JSON format.

   Execution format: search.py fsimage.xml index.xml "core site"

   Example output (print to the screen):

```
/user/ec2-user/input/core-site.xml
{"id": 16393, "type": FILE "mtime": 1581874756018, "blocks": [1073741828]}
/user/ec2-user/inf551/input/core-site.xml
{"id": 16404, "type": FILE "mtime": 1581874949513, "blocks": [1073741837]}
```

/user/ec2-user/inf351/input/core-site.xml
{“id”: 16415, “type”: FILE “mtime”: 1581875617276, “blocks”: [1073741846]}

Note that you only need to show ids of blocks.

The following libraries are permitted in this homework: sys, lxml, json.

Use python3.7 to complete this assignment.

Submission: please include both invert.py and search.py in one folder and compress it as Firstname_Lastname.zip (e.g. Tommy_Trojan.zip)