CprE 419 Lab 1: Using the Cluster, and Introduction to HDFS

Shuo Wang

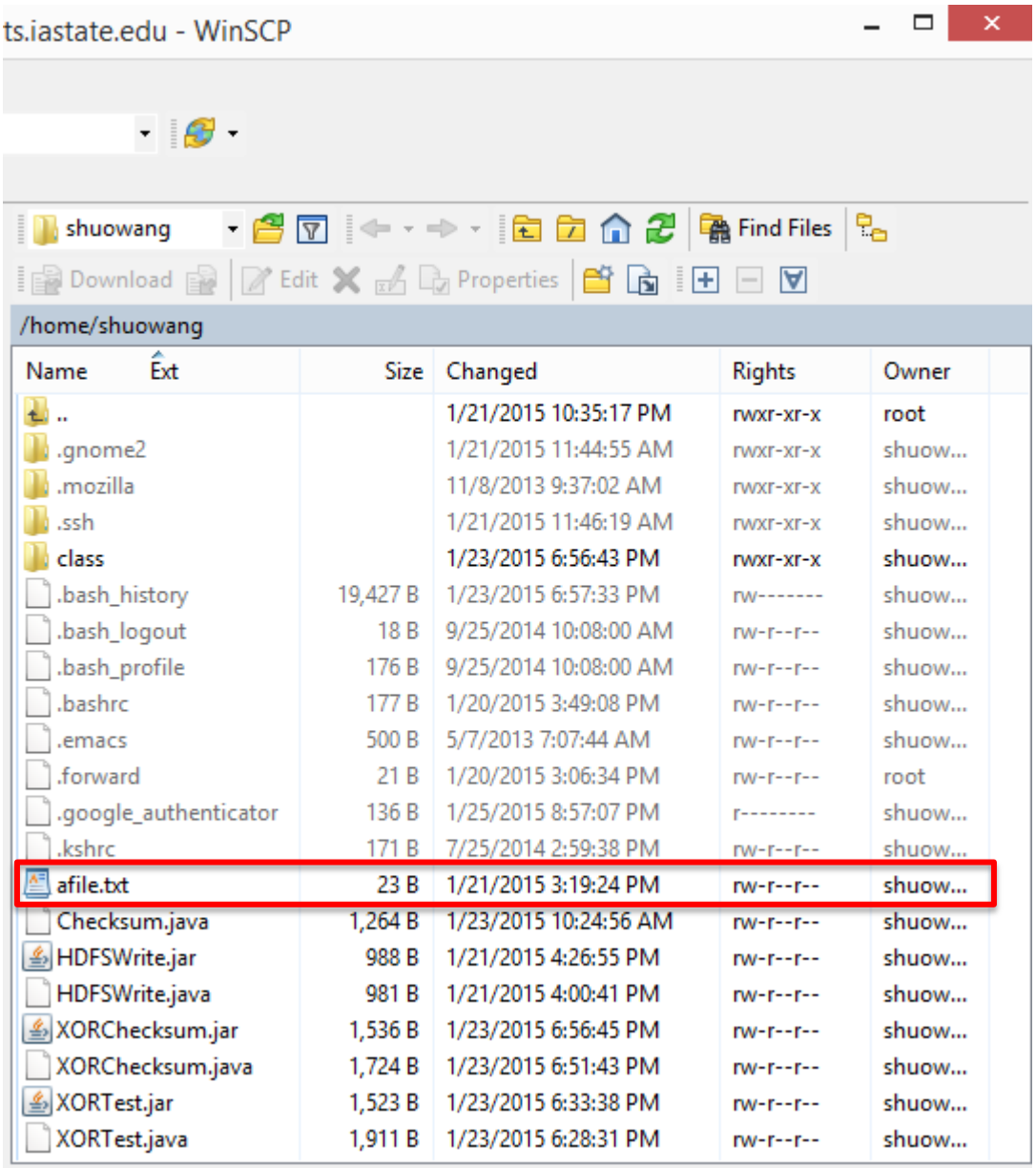
Experiment 1 (10 points):

Create a directory called /user/<your login id>/lab1 under HDFS. Create a new file called "afile.txt" on your local machine with some text (make me laugh). Then use WinSCP (or ssh on linux) to move the file to your home directory in Cystorm. Use a *hdfs dfs* command to move this file to the directory you created. Show this to the instructor.

1. Create a new directory

```
[shuowang@n0 ~]$ hdfs dfs -mkdir /user/shuowang/lab1
```

2. Create "afile.txt" and move it to Cystorm:



2. Use a *hdfs dfs* command to move this file to the directory you created

```
[shuowang@n0 ~]$ hdfs dfs -copyFromLocal /home/shuowang/afile.txt /user/shuowang/lab1/afile.txt
```

```
[shuowang@n0 ~]$ hdfs dfs -ls /user/shuowang/lab1
Found 4 items
-rw-r--r--   3 shuowang 419x         52 2015-01-23 17:48 /user/shuowang/lab1/XORChecksumTestFil
e
-rw-r--r--   3 shuowang 419x         23 2015-01-21 15:24 /user/shuowang/lab1/afile.txt
-rw-r--r--   3 shuowang 419x       1000 2015-01-23 12:04 /user/shuowang/lab1/bigdataread
-rw-r--r--   3 shuowang 419x         50 2015-01-21 16:28 /user/shuowang/lab1/newfile
```

```
[shuowang@n0 ~]$ hdfs dfs -cat lab1/afile.txt
nice to see you, hadoop[shuowang@n0 ~]$
```

Experiment 2 (40 points):

Write a program using the Java HDFS API that reads the contents of the HDFS file "*/class/s15419x/lab1/bigdata*" and computes the 8-bit XOR checksum of all bytes whose offsets range from 5000000000 till 5000000999, both endpoints inclusive. Print out the 8-bit checksum.

Attach the Java code in your submission, as well as the XOR output.

For instance, the XOR checksum of the bitstring "00000000111111110000000011111111" is "00000000".

1. Java code:

```
import java.io.*;
import java.lang.*;
import java.util.*;
import java.net.*;
import org.apache.hadoop.fs.*;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.util.*;


public class XORChecksum  {

    public static void main ( String [] args ) throws Exception {

        // The system configuration
        Configuration conf = new Configuration();

        // Get an instance of the Filesystem
        FileSystem fs = FileSystem.get(conf);

                        String path_name = "/class/s15419x/lab1/bigdata";

        Path path = new Path(path_name);

                        // Open File for reading
                        FSDataInputStream in = fs.open(path);

                        // Create buffer to store data
                        byte[] buffer = new byte[1000];

        // Read all bytes whose offsets range from 5000000000 till 5000000999 from File into buffer.
        long location = 5000000000L;
                        int offset = 0;
                        int length = 1000;

                        int bytesRead = in.read(location,buffer,offset,length);

                        System.out.println("start byte #: " + Long.toString(location));
                        System.out.println("Number of bytes read: " + bytesRead);

                        // Initiate xorChecksum
                        byte xorChecksum = 0;

                        // Create loop to compute xor byte by byte
                        for (byte b:buffer)
                        {
                                    xorChecksum ^= b;
                        }

                        // Display the byte XORChecksum and the 8-digit XORChecksum
                        System.out.println("byte XOR Checksum: " + xorChecksum);
                        System.out.println("8-digit XOR Checksum: " + String.format("%8s",
Integer.toBinaryString(xorChecksum & 0xFF)).replace(' ', '0'));
```

```
        // Close the file and the file system instance
        in.close();
        fs.close();

    }

}
```

2. output result

```
[shuowang@n0 ~]$ mkdir class
[shuowang@n0 ~]$ javac -classpath /hadoop/share/hadoop/common/hadoop-com
mon-2.4.1.jar:/hadoop/share/hadoop/common/lib/hadoop-annotations-2.4.1.j
ar:/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-client-core-2.4.1.jar
 -d class/ XORChecksum.java
[shuowang@n0 ~]$ jar -cvf XORChecksum.jar -C class/ .
added manifest
adding: XORChecksum.class(in = 1906) (out= 1065)(deflated 44%)
[shuowang@n0 ~]$ hadoop jar XORChecksum.jar XORChecksum
start byte #: 5000000000
Number of bytes read: 1000
byte XOR Checksum: 4
8-digit XOR Checksum: 00000100
```