

Report for Lab Assignment 4

1.

Question: Hadoop Map Reduce Algorithm

Implement MapReduce algorithm for finding Facebook common friends problem and run the Map Reduce job on Apache Hadoop. Write a report including your algorithm and result screenshots.

Description:

This algorithm should find the best path in finding the mutual friends between two friends. Here we input the text file with the friends list and their links to friends. Then the program will take that input and analyze the input and make the mappings and reduce them to the list where we can find the mutual friends among two friends.

Algorithm: <in next page>

Report for Lab Assignment 4

Algorithm:

```
/**
 * Created by Vamsi on 2/17/2016.
 */
/**/

/*Creating Mapping Function*/

    public static class FriendsMapper
        extends Mapper<Object, Text, Text, Text> {
/*Create ids for maps*/
        private Text m_id = new Text();
        private Text m_others = new Text();

        public void map(Object key, Text value, Context context)
            throws IOException, InterruptedException {
/* In our case, the key is null and the value is one line of our input file.
Split by space to separate the user and its friends list.*/
            String line = value.toString();
            String[] split = line.split(" ");
            String subject = split[0];
            String[] friends = Arrays.copyOfRange(split, 1, split.length);

/*For each friend in the list, output the (UserFriend, ListOfFriends) pair*/
            for(String friend : friends) {
                String others = line.replace(subject, "").replace(" ", "");
                String id = subject.compareTo(friend) < 0 ? subject+friend :
friend+subject;
                m_id.set(id);
                m_others.set(others);
                context.write(m_id, m_others);
            }
        }
    }
```

Report for Lab Assignment 4

```
/*Creating Reduce Function*/

public static class FriendsReducer
    extends Reducer<Text, Text, Text, Text> {
    private Text m_result = new Text();

    /*Calculates intersection of two given Strings, i.e. friends lists*/
    private String intersection(String s1, String s2) {
        HashSet<Character> h1 = new HashSet<Character>();
        HashSet<Character> h2 = new HashSet<Character>();

        for(int i = 0; i < s1.length(); i++) {
            h1.add(s1.charAt(i));
        }
        for(int i = 0; i < s2.length(); i++) {
            h2.add(s2.charAt(i));
        }

        h1.retainAll(h2);
        Character[] res = h1.toArray(new Character[0]);
        String intersect = new String();
        for (int i = 0; i < res.length; i++) {
            intersect += res[i];
        }

        char[] letters = intersect.toCharArray();
        Arrays.sort(letters);
        String sortedIntersect = new String(letters);
        return sortedIntersect;
    }

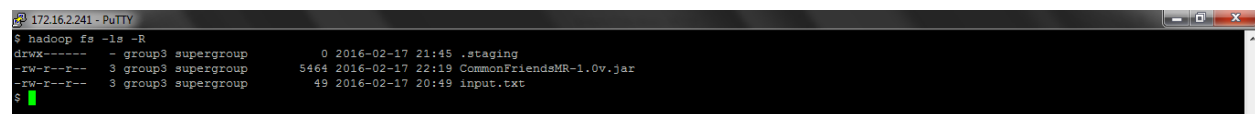
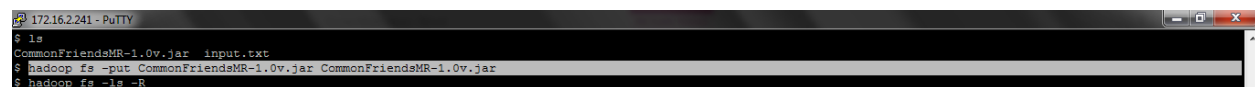
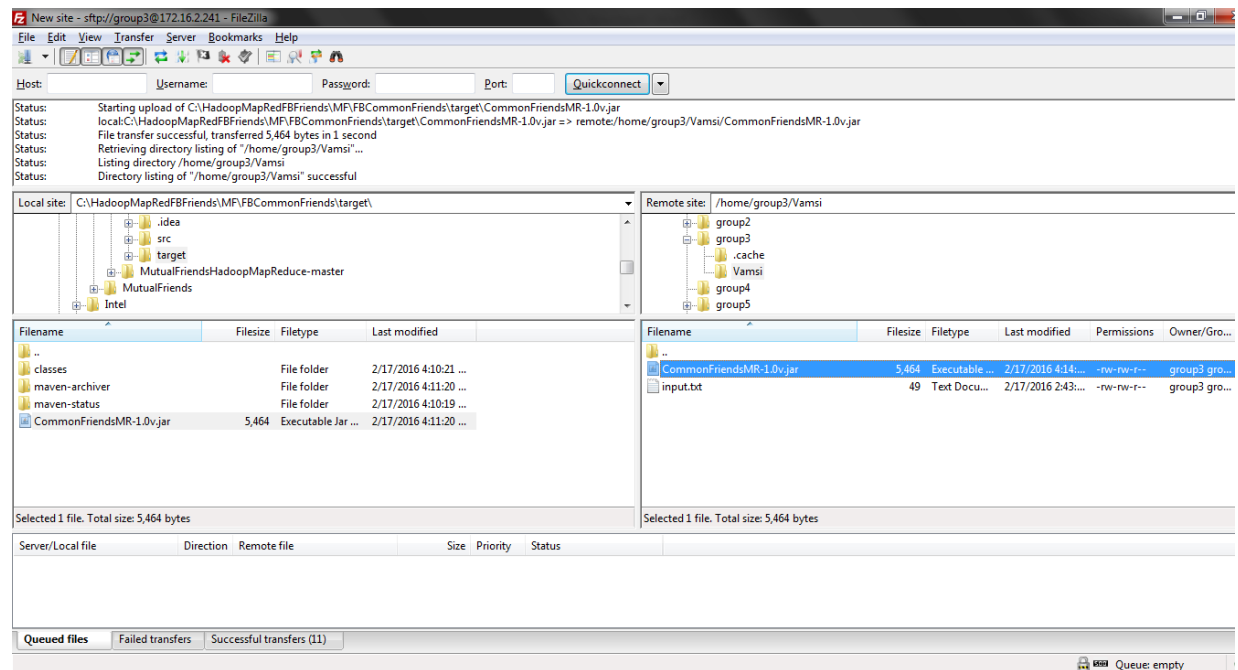
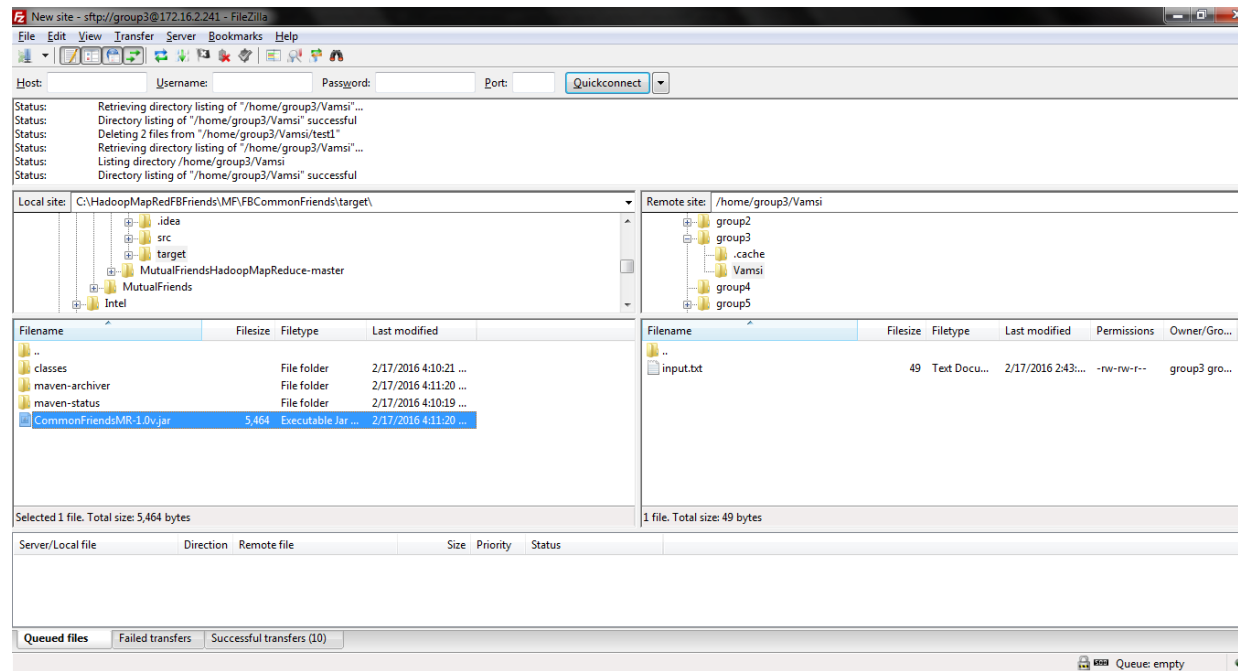
    public void reduce(Text key, Iterable<Text> values, Context context)
        throws IOException, InterruptedException {
        /*Prepare a 2-String-Array to hold the values, i.e. the friends lists of our current
        friends pair.*/

        String[] combined = new String[2];
        int cur = 0;
        for(Text value : values) {
            combined[cur++] = value.toString();
        }

        /*Calculate the intersection of these lists and write result in the form (UserAUserB,
        CommonFriendsMR).*/
        m_result.set(intersection(combined[0], combined[1]));
        context.write(key, m_result);
    }
}
```

Report for Lab Assignment 4

ScreenShots:



Report for Lab Assignment 4

```
172.16.2.241 - PuTTY
$ hadoop jar CommonFriendsMR-1.0v.jar MutualFriends input.txt Output
16/02/17 22:40:11 INFO client.RMProxy: Connecting to ResourceManager at KC-SCE-CS5542-1/172.16.2.241:8032
16/02/17 22:40:11 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
16/02/17 22:40:12 INFO input.FileInputFormat: Total input paths to process : 1
16/02/17 22:40:12 INFO mapreduce.JobSubmitter: number of splits:1
16/02/17 22:40:12 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1455690915780_0015
16/02/17 22:40:12 INFO impl.YarnClientImpl: Submitted application application_1455690915780_0015
16/02/17 22:40:12 INFO mapreduce.Job: The url to track the job: https://KC-SCE-CS5542-1:8088/proxy/application_1455690915780_0015/
16/02/17 22:40:12 INFO mapreduce.Job: Running job: job_1455690915780_0015
16/02/17 22:40:22 INFO mapreduce.Job: Job job_1455690915780_0015 running in uber mode : false
16/02/17 22:40:22 INFO mapreduce.Job: map 0% reduce 0%
16/02/17 22:40:29 INFO mapreduce.Job: map 100% reduce 0%
16/02/17 22:40:37 INFO mapreduce.Job: map 100% reduce 50%
16/02/17 22:40:45 INFO mapreduce.Job: map 100% reduce 100%
16/02/17 22:40:46 INFO mapreduce.Job: Job job_1455690915780_0015 completed successfully
16/02/17 22:40:46 INFO mapreduce.Job: Counters: 49

File System Counters
  FILE: Number of bytes read=172
  FILE: Number of bytes written=344980
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=163
  HDFS: Number of bytes written=57
  HDFS: Number of read operations=9
  HDFS: Number of large read operations=0
  HDFS: Number of write operations=4

Job Counters
  Launched map tasks=1
  Launched reduce tasks=2
  Data-local map tasks=1
  Total time spent by all maps in occupied slots (ms)=5362
  Total time spent by all reduces in occupied slots (ms)=11313
  Total time spent by all map tasks (ms)=5362
  Total time spent by all reduce tasks (ms)=11313
  Total vcore-seconds taken by all map tasks=5362
  Total vcore-seconds taken by all reduce tasks=11313
  Total megabyte-seconds taken by all map tasks=5490688
  Total megabyte-seconds taken by all reduce tasks=11584512

Map-Reduce Framework
  Map input records=5
  Map output records=18
  Map output bytes=138

172.16.2.241 - PuTTY
  Launched map tasks=1
  Launched reduce tasks=2
  Data-local map tasks=1
  Total time spent by all maps in occupied slots (ms)=5362
  Total time spent by all reduces in occupied slots (ms)=11313
  Total time spent by all map tasks (ms)=5362
  Total time spent by all reduce tasks (ms)=11313
  Total vcore-seconds taken by all map tasks=5362
  Total vcore-seconds taken by all reduce tasks=11313
  Total megabyte-seconds taken by all map tasks=5490688
  Total megabyte-seconds taken by all reduce tasks=11584512

Map-Reduce Framework
  Map input records=5
  Map output records=18
  Map output bytes=138
  Map output materialized bytes=164
  Input split bytes=114
  Combine input records=0
  Combine output records=0
  Reduce input groups=9
  Reduce shuffle bytes=164
  Reduce input records=18
  Reduce output records=9
  Spilled Records=36
  Shuffled Maps =2
  Failed Shuffles=0
  Merged Map outputs=2
  GC time elapsed (ms)=125
  CPU time spent (ms)=2800
  Physical memory (bytes) snapshot=923353344
  Virtual memory (bytes) snapshot=4138168320
  Total committed heap usage (bytes)=989331456

Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0

File Input Format Counters
  Bytes Read=49
File Output Format Counters
  Bytes Written=57
```

```
172.16.2.241 - PuTTY
$ hadoop fs -ls -R
drwx----- - group3 supergroup          0 2016-02-17 22:40 .staging
-rw-r--r--  3 group3 supergroup        5566 2016-02-17 22:39 CommonFriendsMR-1.0v.jar
drwxr-xr-x - group3 supergroup          0 2016-02-17 22:40 Output
-rw-r--r--  3 group3 supergroup          0 2016-02-17 22:40 Output/_SUCCESS
-rw-r--r--  3 group3 supergroup        38 2016-02-17 22:40 Output/part-r-00000
-rw-r--r--  3 group3 supergroup        19 2016-02-17 22:40 Output/part-r-00001
-rw-r--r--  3 group3 supergroup        49 2016-02-17 20:49 input.txt
$ hadoop fs -cat Output/part-r-00000
AB      CD
AD      BC
BC      ADE
BE      CD
CD      ABE
DE      BC
$ hadoop fs -cat Output/part-r-00001
AC      BD
BD      ACE
CE      BD
```

2.

Question:

Smartphone/Watch Application Implement a smartwatch/smartphone application using existing speech services/image services (e.g., IBM Alchemy api, Face++) related to your project.

Description:

Here I have used Face plus plus API which I have used in my previous semester project. This API will take image as input and send it to the Face plus plus API server and get the coordinates of the number of faces identified. Then this program will calculate the max length and width from the facial focal point and then draw a box/lines around the facial focal point.


Screenshots:

<Please check from next page>



The status bar at the top of the mobile interface shows various icons: a red book icon, a camera icon, a video call icon, a speaker icon, a clock icon, an LTE signal icon, a battery level icon showing 11%, and the time 10:23 PM.


Detect Face

A photograph of two men sitting on a yellow patterned couch. The man on the left is wearing a grey t-shirt with 'APPAREL CO' and 'NOTHING BEATS' printed on it, and glasses. The man on the right is wearing a black t-shirt and is holding a smartphone. They are both looking down at the phone.

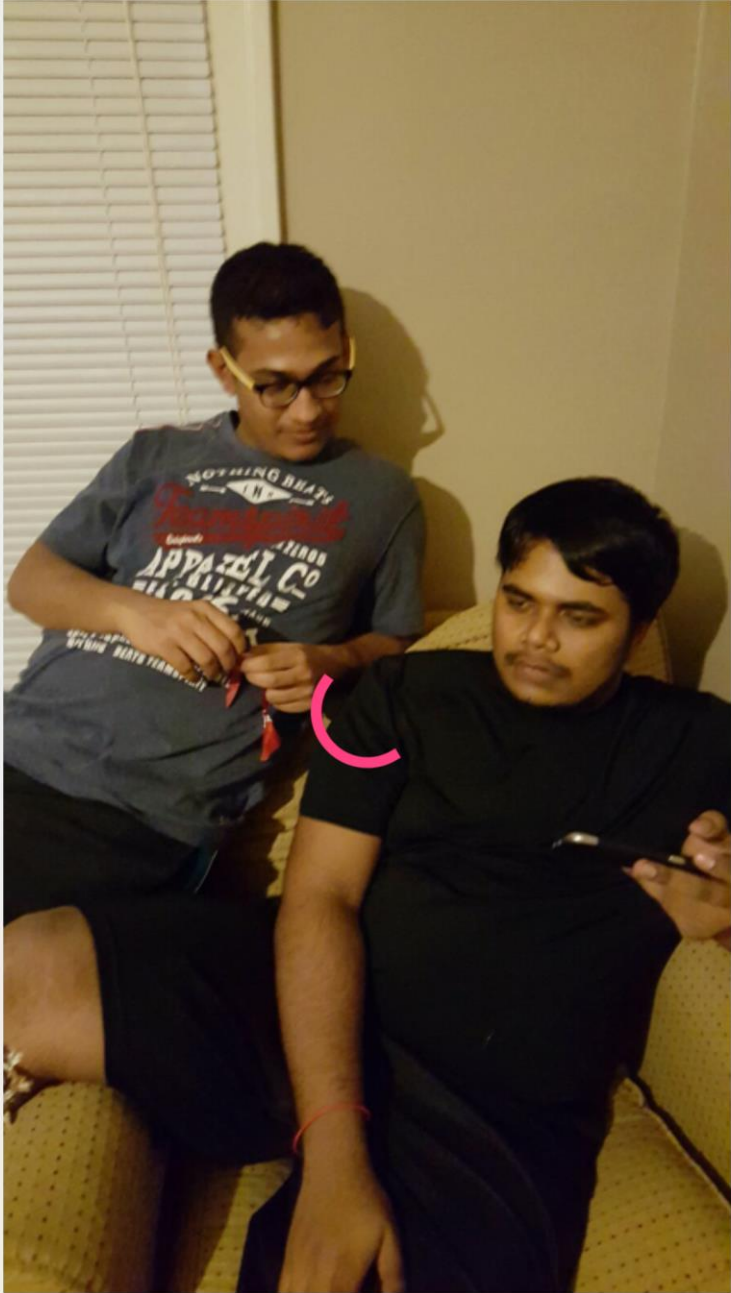
Click Detect ==>

DETECT

GET IMAGE

The status bar at the top of the mobile interface shows various icons: a red book icon, a camera icon, a video call icon, a speaker icon, a clock icon, an LTE signal icon, a battery icon showing 11% charge, and the time 10:24 PM.


Detect Face

The main area of the application displays a photograph of two men sitting on a couch. The man in the foreground is wearing a black t-shirt and is looking down at a smartphone in his hands. A pink curved line is drawn around his face, indicating that the face detection algorithm has successfully identified him. The man in the background is wearing a grey t-shirt with a graphic design and glasses. The background of the photo shows a window with white blinds.

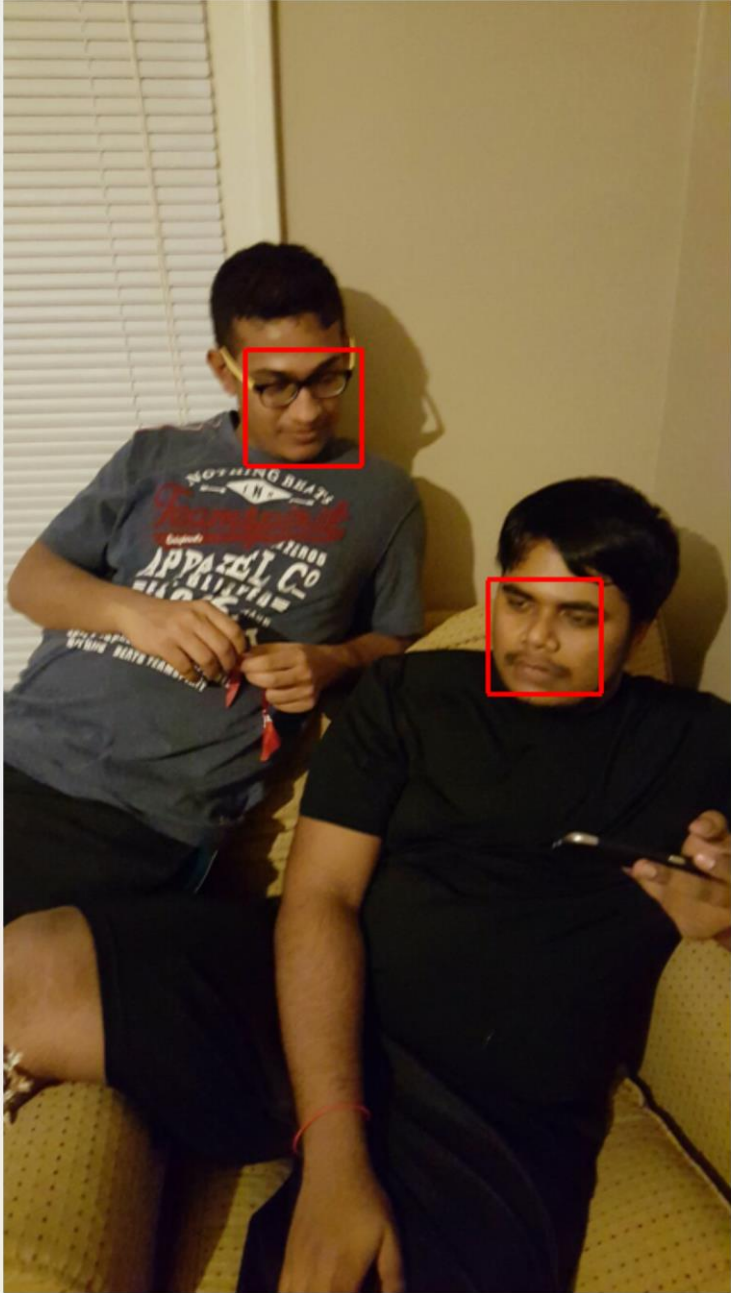
Click Detect ==>

DETECT

GET IMAGE

The status bar at the top of the mobile interface shows various icons: a red alarm icon, a camera icon, a video call icon, a speaker icon, a clock icon, an LTE signal icon, a battery icon, and the time 10:24 PM. The battery level is indicated as 11%.

Detect Face

A photograph of two men sitting on a couch. The man on the left is wearing a grey t-shirt with a graphic and glasses. The man on the right is wearing a black t-shirt and is holding a smartphone. Both faces are enclosed in red rectangular bounding boxes, indicating successful face detection.

Found 2 faces

DETECT

GET IMAGE

Report for Lab Assignment 4