**Report for Assignment3**

**KDM**

1.Make a Mashup application including various services (e.g., Google Map, Google Chart, Google Search, Yahoo, Amazon, Twitter, Facebook) Web Services (e.g., Google Map Services, Weather Services) using either (1) Mobile Web Technology with HMLT5 Local DB (Refer to Tutorial 3) or (2) Android App Technology (Tutorial 3). You can use your Lab 2 work to complete this work. If you implement mobile client application, condiser to use Opera Mobile Emulator to generate screens for your mobile web app view <http://www.opera.com/developer/mobile-emulator>

2. Cloudera/MapReduce: Download the Cloudera Image, implement the WordCount MapReduce and run it. (a bonus point for implementing a new MapReduce algorithm) The code and guidelines will be available in Tutorials/Tutorial 5.

3. Cloudera/Mahout: Configure your Cloudera with Mahout. Run Naive Bayes classifier with the input data (a bonus point for using your own data).

4. Write a short report on your work (including screenshots).

**1) Mashup Application:**

The android application as two pages:

**Page 1:**

Retrieves the current address of the location

Weather of the current location

Latitude and longitude details of the location

**Page2:**

Locates the current location in google map

Gets the results of weather details from google automatically.

Gets the weather news in Google.

**Web service api used:**

**Weather:** “http://api.wunderground.com/api/36b799dc821d5836/conditions/q/" +State +"/"+City +".json”

**Address:** “http://maps.googleapis.com/maps/api/geocode/json?latlng=" + latitude + ","+ longitude + "&sensor=true"

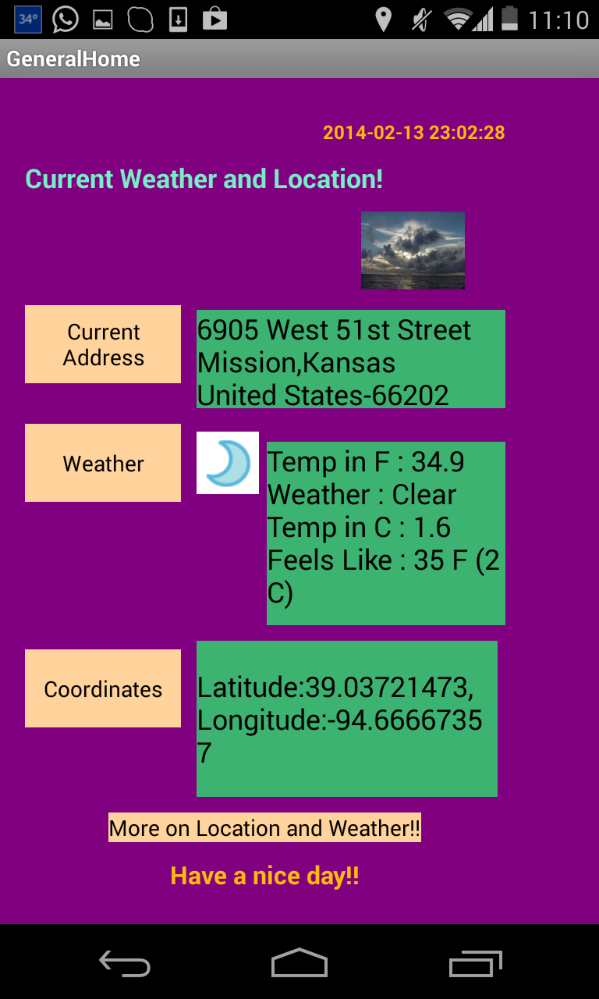
**Services used:**

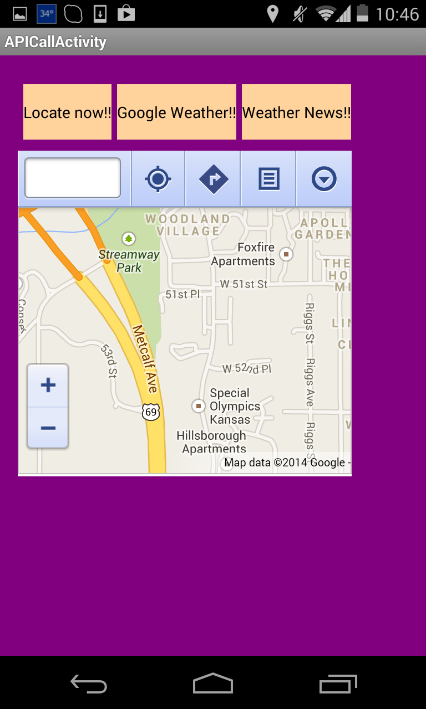
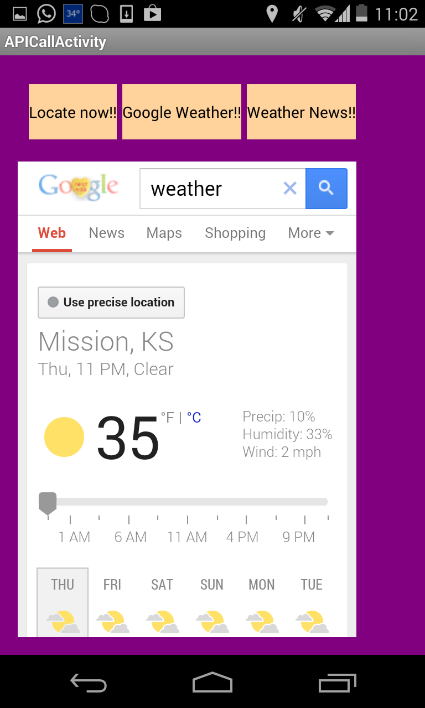
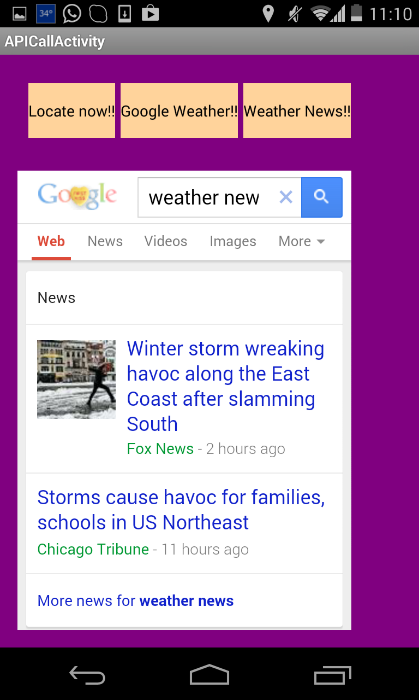
Google search service for weather

Google maps service

Google search se­­­rvice for weather news

**Screenshots:**

****

**  **

**2) Cloudera/MapReduce:**

**Two implementations**:

1) Word-count

2) Student mark calculation

Installed VMWare Player and downloaded the cloudera image. Ran the word count.

**1)Word-count**

Implemented own algorithm for Word Count by implementing Mapper and reducer function. Runs really fast:

**Screenshot:**

****

**Snippets:**

public class MapReduce {

public static void main(String[] args) {

Map mapper = new Map();

Reduce reducer = new Reduce();

File input\_file = new File("test");

File output\_file = new File("op");

if (0 < args.length) {

String filename = args[0];

input\_file = new File(filename);

}

try {

BufferedReader fr = new BufferedReader(new InputStreamReader(new FileInputStream(input\_file), "ASCII"));

Pairs word;

while(true)

{

String line = fr.readLine();

if(line==null)

break;

mapper.mapper(line,words\_list);

}

mapper.print\_mapper(words\_list);

HashMap<String,Integer> final\_list = new HashMap<String,Integer>();

reducer.reducer(words\_list,final\_list);

System.out.println("\nPath of the output file : " + output\_file.getAbsolutePath());

PrintWriter writer = new PrintWriter(output\_file, "UTF-8");

for (Entry<String, Integer> entry : final\_list.entrySet()) {

String key = entry.getKey();

Integer value = entry.getValue();

writer.println(key+"-"+value);

}

writer.close();

} catch (FileNotFoundException e) {

e.printStackTrace();

}

catch(IOException e)

{

e.printStackTrace();

}

}

}

class Map

{

public void mapper(String line, ArrayList<Pairs> words\_list)

{

String[] words = line.split(" ");//those are your words

Pairs word;

System.out.println(words.length);

for (int i = 0; i < words.length; i++) {

System.out.println(words[i]+"\n");

if (!(words[i].equals("")))

{

word = new Pairs(words[i],1);

words\_list.add(word);

//word\_list.put(words[i], 1);

}

}

}

public void print\_mapper( ArrayList<Pairs> words\_list){

System.out.println("Mapped");

for (int i = 0; i <words\_list.size(); i++) {

words\_list.get(i).display();

}

}

}

class Reduce

{

public void reducer( ArrayList<Pairs> words\_list, HashMap<String, Integer> final\_list)

{

for (int i = 0; i <words\_list.size(); i++) {

String key = words\_list.get(i).getKey();

Integer value = words\_list.get(i).getValue();

if (final\_list.containsKey(key)){

Integer previous\_value = final\_list.get(key);

final\_list.put(key,previous\_value+1);}

else

final\_list.put(key,value);

}

print\_reducer(final\_list);

}

public void print\_reducer(HashMap<String, Integer> final\_list){

System.out.println("\n\nReduced");

for (Entry<String, Integer> entry : final\_list.entrySet()) {

String key = entry.getKey();

Integer value = entry.getValue();

System.out.println(key+"-"+value);

}

}

}

class Pairs

{

public String key;

public Integer value;

public Pairs(String key, Integer value)

{

this.key = key;

this.value = value;

}

public void display()

{

System.out.println(key+"-"+value);

}

public String getKey() {

return key;

}

public void setKey(String key) {

this.key = key;

}

public Integer getValue() {

return value;

}

public void setValue(Integer value) {

this.value = value;

}

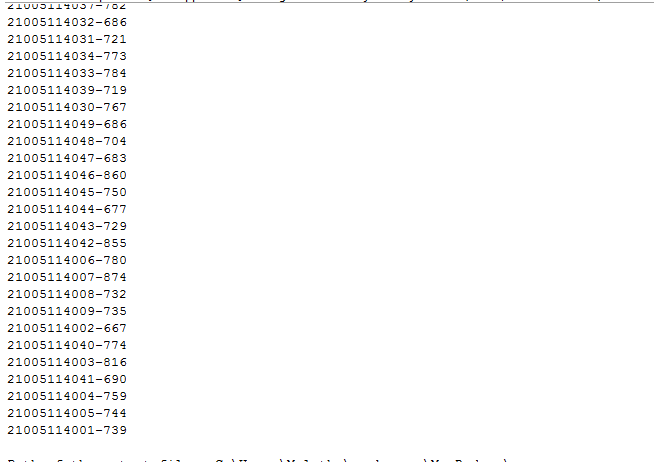
}

Attached the jar and class file in github:

**Run**: hadoop jar MapReduce.jar MapReduce input/Word\_Count\_input.txt

**2) Student Mark calculation:**

The best use of Mapper and Reducer would be to calculate the total marks of a student in a class or university.



**3) Cloudera/Mahout:**

**Own dataset:**

**Reference link :** http://mahout.apache.org/users/classification/wikipedia-bayes-example.html

**Downloaded data**: http://dumps.wikimedia.org/enwiki/latest/enwiki-latest-stub-articles19.xml.gz

**Commands to execute:**

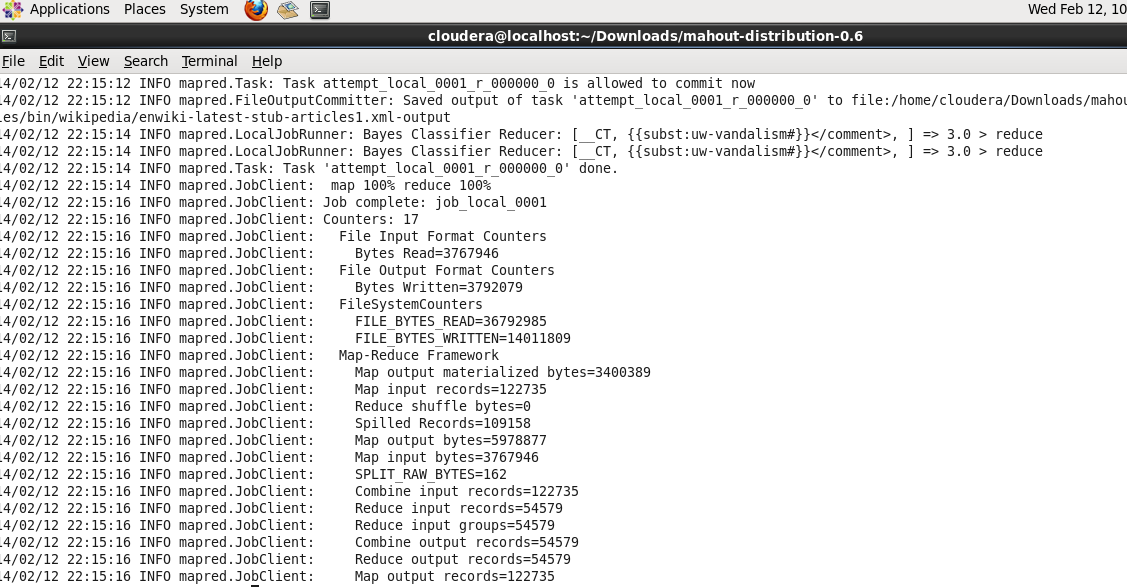
bin/mahout trainclassifier -i examples/bin/wikipedia/enwiki-latest-stub-articles1.xml -o examples/bin/wikipedia/wikipediamodel1/

bin/mahout testclassifier -m examples/bin/wikipedia/wikipediamodel1 –d examples/bin/wikipedia/enwiki-latest-stub-articles19.txt

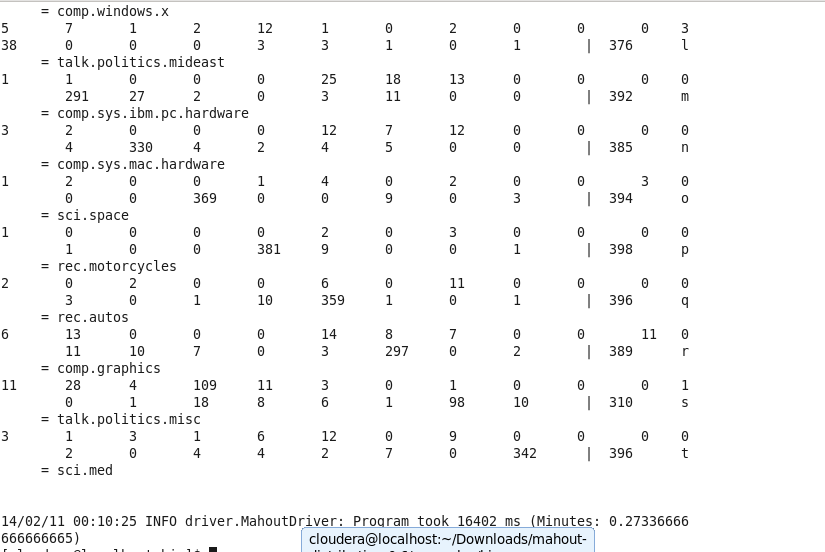
I am getting Success, but not able to find the output.

Execution of own dataset:

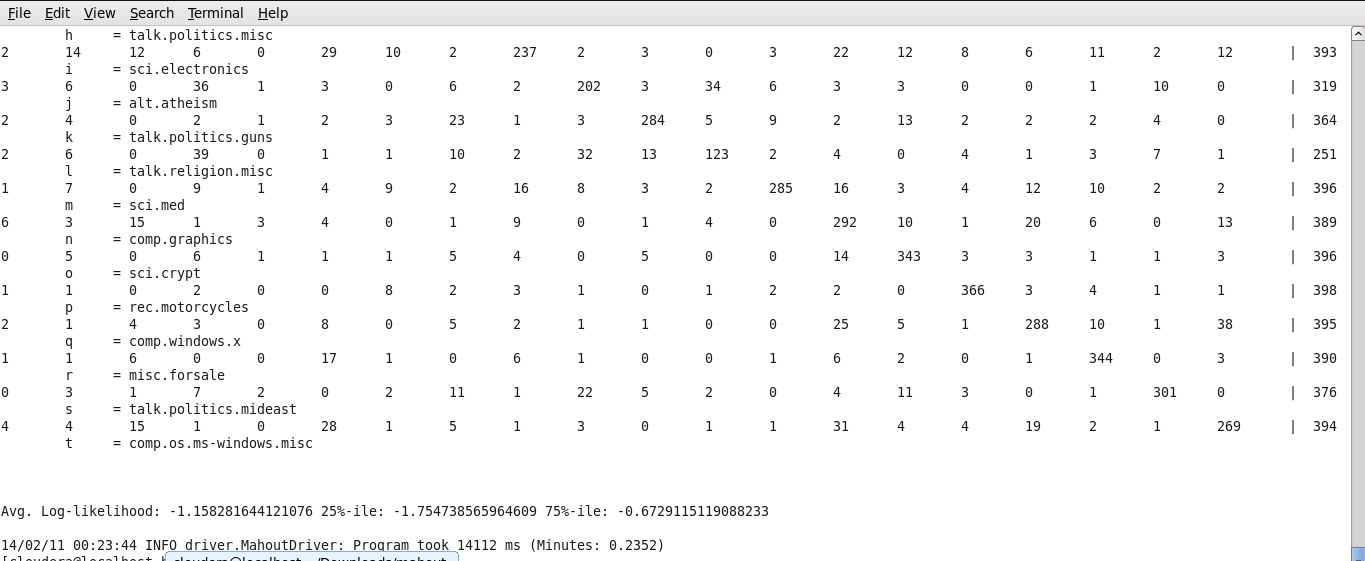




Executing the Newsgroups20 dataset,screenshots: Naive Bayes classifier



**Executing SGD:**



**Learnt:**

Invoking api and service calls through Android(Getting authentication key for the application).

Mahout datasets , need to learn more on training and testing.

Installation of cloudera, mahout.