Parallel Cloud

Deployment of Hadoop MapReduce environment in a private cloud and implement Sentiment Analysis of Yelp dataset

Dhanyatha Manjunath

Arizona State University

dmanjuna@asu.edu

Rajath Agasthya

Arizona State University

ragasthy@asu.edu

Sonali Bhat

Arizona State University

sjbhat1@asu.edu

***Abstract*— The project is aimed at performing Sentiment Analysis on Yelp dataset to predict accurate rating of the review. The analysis will be performed using MapReduce framework on a Hadoop cluster deployed in Vlab private cloud. The main tasks of the project include deploying Hadoop cluster in Vlab cloud, analyzing sentiment of a yelp review and predict a rating based on the sentiment of the review. The project is scheduled to be completed by October 5, 2014.**

***Keywords— Cloud Computing, Hadoop, MapReduce, Sentiment Analysis***

# Introduction

We intend to do sentiment analysis on the yelp dataset to predict the review's rating from its description provided by the user for a business. Sentiment analysis is determining whether a given text conveys a positive, negative or neutral sentiment. It is often the case that a review and its rating are not consistent.

Sentiment Analysis is an important technique to bring consistency between the review and the actual star rating. This is crucial for the businesses as it provides an accurate rating of their products/services. Businesses can then alter their services to provide a better customer experience.

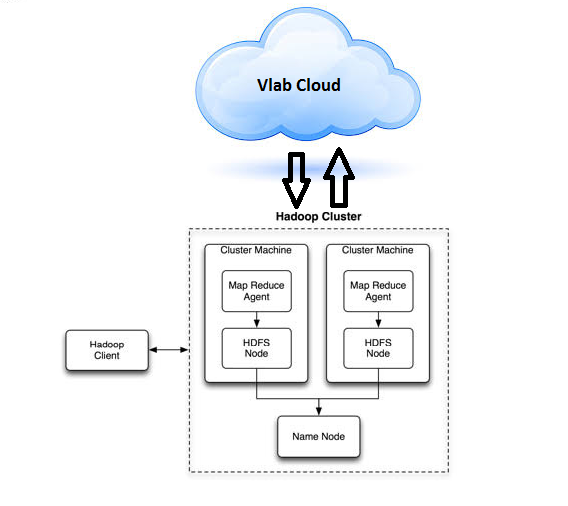
The rating can be predicted using various machine learning techniques such as Naive Bayes Classifier or support vector machine (SVM) learning algorithms.

The user should be able to enter a review description and obtain the corresponding review rating from the classifier as the result. The application performs sentiment analysis on the user review using the learning model generated from the yelp review dataset(training data). The application classifies the user input(test data) and predict a rating to the review provided.

The project is planned for 32 days, the details of which are provided in Section VII.

# System Model

## **System Model**



## **Software Requirements**

Programming Language: Java

Framework: JSF Framework, Hadoop Java Client

Libraries: Apache Mahout(Classification)

Tools: Eclipse, Git, Tomcat, Maven

Environment: Hadoop 2.3.0

1. ***Hardware Resources provided***

3 Virtual Machines on Vlab Cloud with 64-bit Ubuntu 14.04 LTS as the OS each with 10 GB disk space and 1 GB RAM.

# Project Description

The goal of this project is to deliver appropriate rating for a review description provided by the user. This involves sentiment analysis of the text input by the user as reviews and to capture the positive and negatives in the review and decide the rating based on these factors. There are various pattern recognition methods available for text analysis. This project aims at finding the best approach and designing an algorithm to do the sentiment analysis and implementing it.

## **Project Overview**

We have 5 major tasks in the project, divided as:

1. Environment Setup
2. Best algorithm selection
3. Implementation
4. Testing in local environment
5. Deployment of application in cloud

1. ***Dataset Overview***

The dataset includes Yelp review data for businesses in Phoenix, Las Vegas, Madison,Waterloo and Edinburgh. It contains data for the following entities:

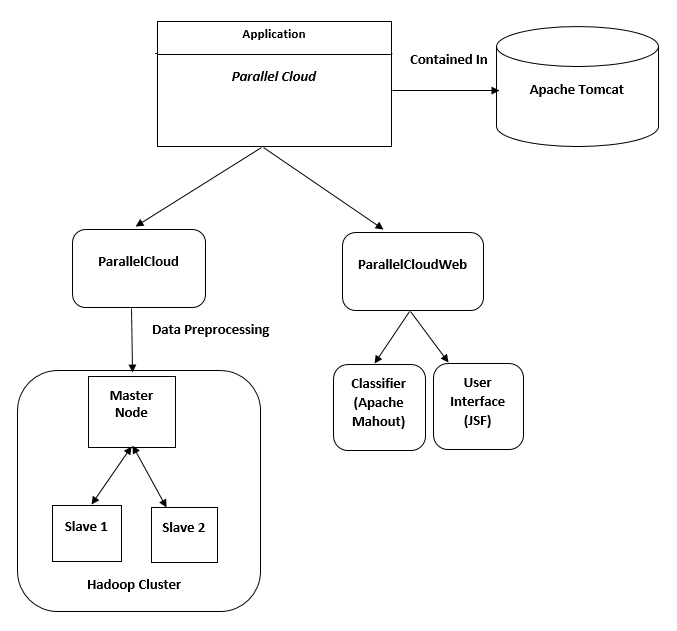
* 42,153 businesses
* 320,002 business attributes
* 31,617 check-in sets
* 252,898 users
* 955,999 edge social graph
* 403,210 tips
* 1,125,458 reviews

1. ***Project Goals***

**Mid-term goal:** Setup of Hadoop environment in cloud and local machine, working initial solution with any algorithm in local Hadoop environment.

**Final Goal:** Full working of the chosen algorithm in a Hadoop environment deployed on Vlab cloud.

# System Design



The application is divided into two projects

***A. ParallelCloud:*** This project preprocesses the yelp dataset. The dataset contains 1.1 million reviews provided by users for business entities. This project contains the application logic to preprocess the dataset using map reduce framework. The output data contains review rating followed by feature set determined for the review using parameters such as unigram score, unigram count, bigram score, bigram count, stop words, positive and negative words. This is the training dataset which shall be used in the subsequent steps to generate a learning model.

***B. ParallelCloudWeb*:** This project implements the random forest classifier. A sequence file is generated from the preprocessed data obtained from previous step, to provide as input to the classifier. The classifier is initially trained using this data and generate a model. Upon receiving user input, which is the test data, the classifier predicts a rating for the review using the model.

This project includes the design and implementation of the user interface. The interface takes a review description from the user as input test data and displays the rating predicted by the classifier.

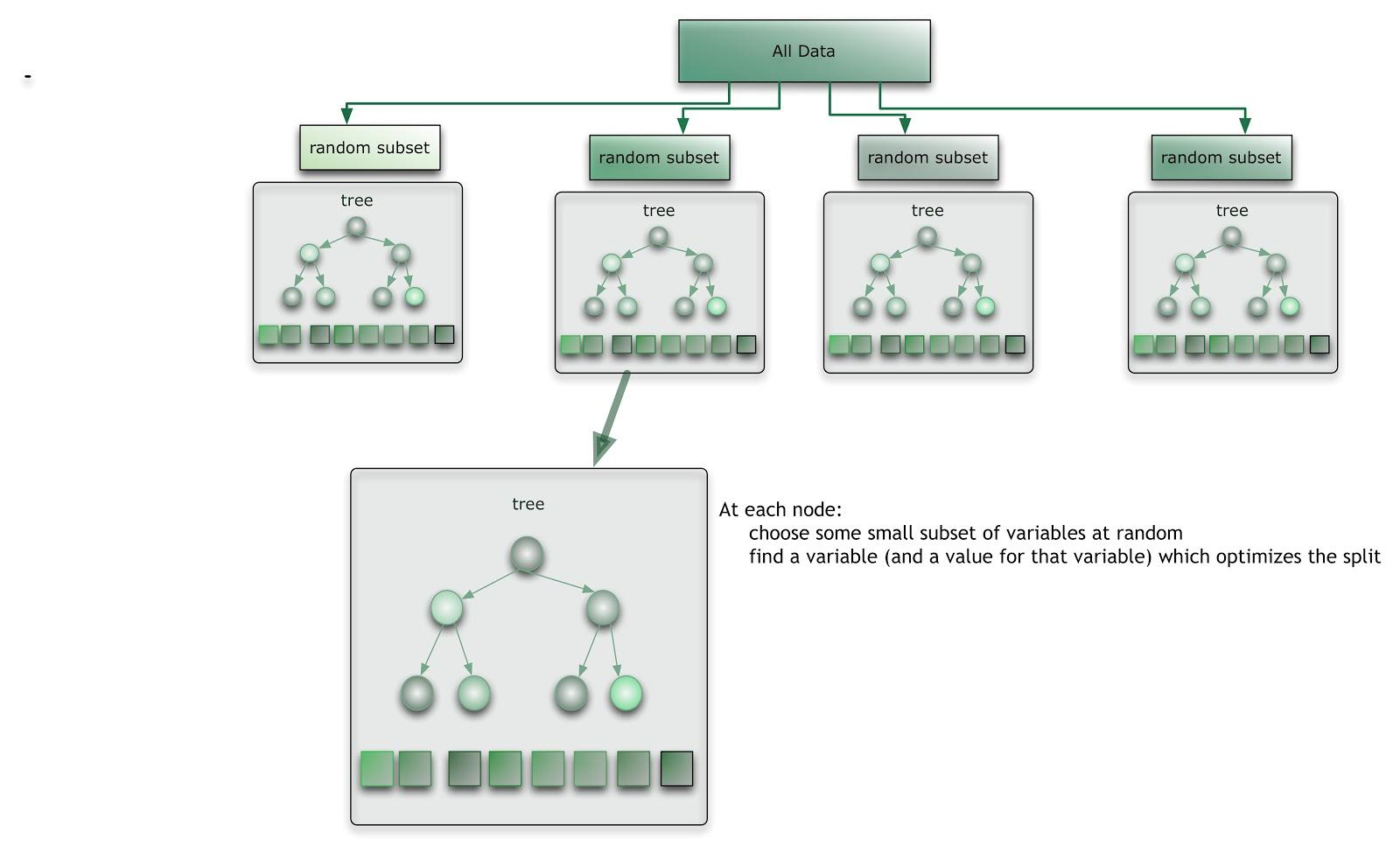
***Algorithm for Classification***

Random Forest Classifier is a learning technique used for classification. It constructs a multitude of decision trees during preprocessing the training dataset. In this project, training data consists of 1.1 million reviews provided by the users.

Random Forest Classifier creates random subset of decision trees. This subset consists of ~65% of the decision trees in total set.Remaining data i.e out-of-bag data (OOB) are used to estimate error and variable importance.

The classifier performs following actions at each node:

1. For a given number m, the classifier selects m predictor variables at random from the set of all predictor variables.
2. An objective function is used to determine the predictor variable which provides the optimal split. A binary split is performed on that node.
3. Another variable m is chosen at random at the next node and the same steps are resumed[1]



Random Forest Classifier[1]

1. Task Description and Allocation

## **Task 1: Setting up the Hadoop environment on the Vlab cloud**

Understanding the Vlab cloud architecture and to set up the multi node Hadoop cluster in the cloud.

## **Task 2: Choosing the best learning algorithm for predicting the rating based on the yelp dataset considering accuracy as the primary goal.**

The project mainly learns the pattern from the reviews in the yelp academic dataset in order to provide the appropriate rating. Hence the learning algorithm becomes the major part of the project. There are various learning algorithms that can be used like Support Vector Machine, Bayes Classifier. We analyzed these algorithms and chose Random Forest classifier since it is more accurate for numerical data.

## **Task 3: Preprocessing of Yelp Review Dataset algorithm using map-reduce framework.**

In this task, we read the reviews from the yelp dataset and calculated the following features for every review

* Unigram scores
* Unigram count
* Bigram score
* Bigram count
* Positive Word count
* Negative Word count

We used a predefined set of stop words, positive words and negative words to determine a score for the above mentioned features of a review. The output of this preprocessing task is a CSV file which contains the review rating followed by the numerical scores for the above mentioned feature set. This task was implemented using map-reduce framework with Java as the programming language.

* 1. ***Task 4: Implementation of Random Forest Classifier using Apache Mahout.***

To begin with, this task involves generation of Sequence File from the preprocessed data obtained in the previous task. For the second half of the task, Random Forest Classifier from Apache Mahout was chosen to be the most suitable technique to classify numerical data. We designed the algorithm to learn patterns from the yelp academic dataset and predict the rating for a review given by the user based on the model generated from training dataset. On completion, the classifier provides a predicted rating for a given input review.

* 1. ***Task 5: Implementing the User Interface to take user inputs.***

This is the last phase of application development. A user interface was created to receive input reviews from the user and provide a rating predicted by the classifier.

## **Task 4: Running and testing of the application on the local system**

Once the initial code is developed, the code is run and tested on the local system to see if the requirements are met and if the system behaves as required.

## **Task 5: Deployment of the code, running and testing of the application**

Once the deployable code is developed, the code has to be deployed on the cloud. Then the code is run and tested on the cloud to see if the requirements are met.

## **Project Task Allocation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task** | **Assignee** | **Due Date** | **% of work** | **Status** |
| Deployment of Hadoop on Vlab Cloud | Rajath, Dhanyatha, Sonali | 9/15 | 20 | **Done.**  **Single node clusters setup on individual VMs, and multi node Hadoop cluster setup on Vlan cloud. One VM dead till Sunday night.** |
| Local Development and Hadoop environment setup | Rajath | 9/9 | 5 | **Done** |
| System Design | Rajath, Dhanyatha, Sonali | 9/12 | 10 | **Done** |
| Algorithms Evaluation | Rajath, Dhanyatha, Sonali | 9/12 | 15 | **Done** |
| Development | Rajath, Dhanyatha, Sonali | 9/27 | 40 | **Done** |
| Testing | Rajath, Dhanyatha, Sonali | 10/4 | 5 | **Done** |
| Report Generation | Rajath, Dhanyatha, Sonali | 10/4 | 5 | **Done** |

1. Task Progress Report

**Task 1: Setting up the Hadoop environment on the Vlab cloud**

1. ***Single Node Setup***

The team followed the following steps to set up a pseudo-distributed, single-node Hadoop cluster backed by the Hadoop Distributed File System, on Ubuntu Linux machine.

1. Installed Java 1.7 as Hadoop requires a working Java 1.5+ installation.

2. Added a dedicated Hadoop system user - ‘hduser’

3. Configured SSH access to localhost for the user- ‘hduser’

4. Installed hadoop package and made ‘hduser’ the owner of the files.

5. updated the configuration files appropriately which includes changes to JAVA\_HOME variable.

6. Formatted the HDFS implemented on top of the local filesystem via namenode

7. Started a single node cluster

8. Ensured that Hadoop is listening on the configured ports.

9. Used three ebooks from Project Gutenberg to run the wordcount example. The ebooks are available in the following links

<http://www.gutenberg.org/ebooks/20417>

<http://www.gutenberg.org/ebooks/5000>

<http://www.gutenberg.org/ebooks/4300>

10. Copied the local data to HDFS

11. Created an executable jar from the wordcount java project, and ran the mapreduce job from Command Line Interface

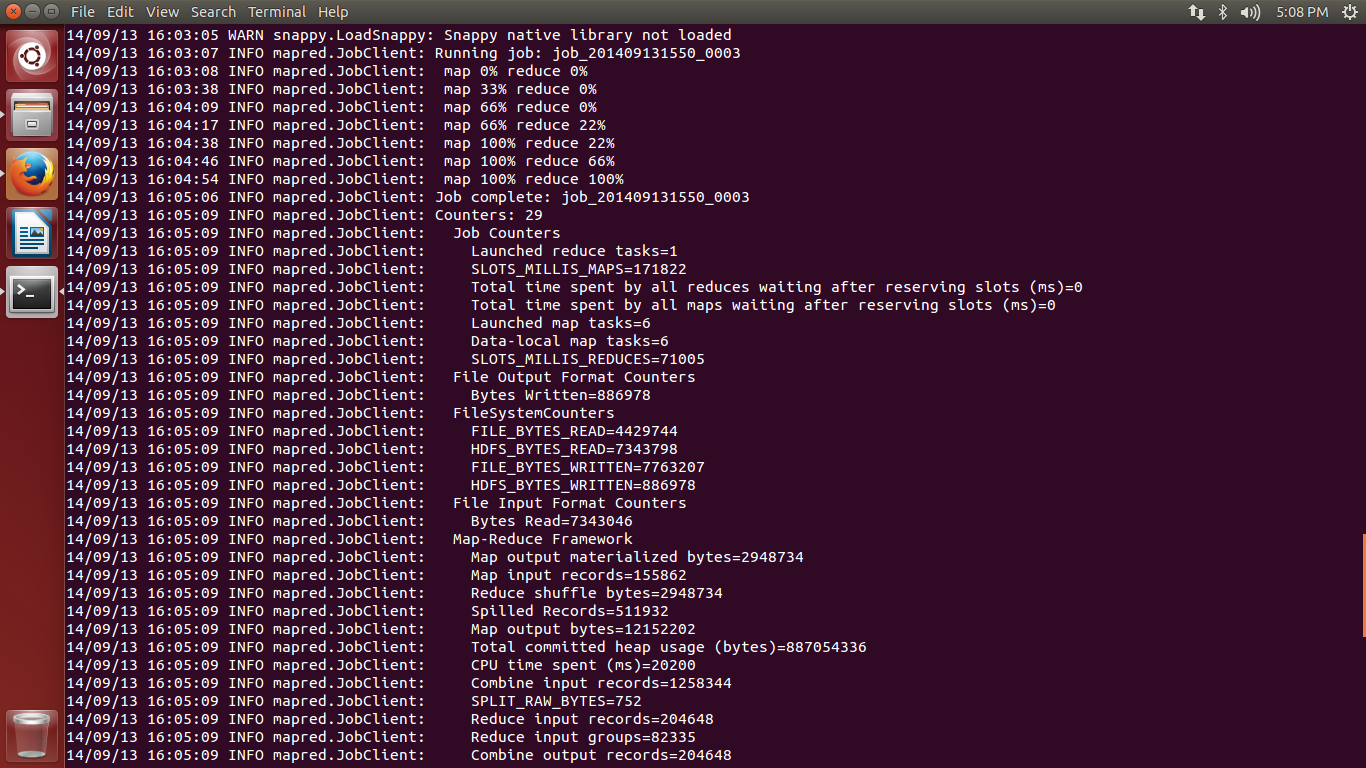
13. Observed the concise information provided about the hadoop cluster summary using the below web interfaces.

<http://localhost:50070/>- web UI of the NameNode daemon

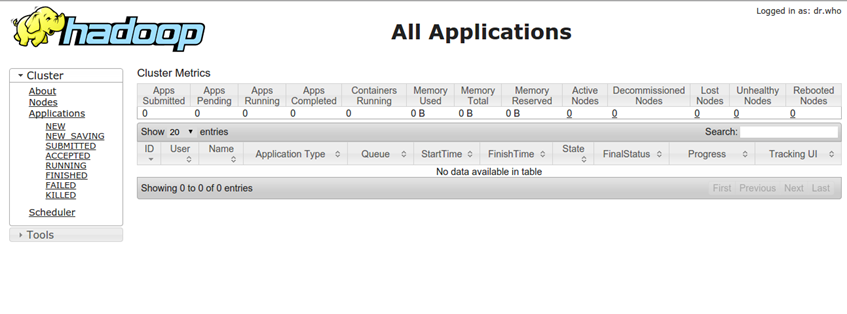
<http://localhost:50030/> – web UI of the JobTracker daemon

<http://localhost:50060/> – web UI of the TaskTracker daemon

Please find screenshots of the results for running the word count example using hadoop.



Screenshot of Resource Manager Summary



1. ***Multi-Node Setup***

The following steps were followed to set up a pseudo-distributed, multi-node Hadoop cluster backed by the Hadoop Distributed File System, on Ubuntu Linux machine

1. Two single node Hadoop clusters were set up on VM 591103 and 591102
2. To ensure that both the machines are reachable over the network, we assigned IP addresses on a common network and updated the /etc/hosts file on both the machines accordingly.
3. Tested if ‘hduser’ on master is able to connect to ‘hduser’ on the slave node via the password-less login. Added the ‘hduser’ on master node`s public key to the ‘authorized\_keys’ file of the slave node.
4. Saved the slave’s host key fingerprint to the hduser@master’s ‘known\_hosts’ file.
5. Tested the SSH setup by connecting master node to itself and connecting master node to the slave node using the below commands.

>> ssh master

>> ssh slave

1. Configured the master node to run the “master” daemons namely –NameNode for HDFS layer and ResourceManager for MapReduce processing layer
2. Configured both the nodes to run the “slave” daemons namely- DataNode for HDFS layer and NodeManager for MapReduce processing layer.
3. On master node, we defined the machine on which Hadoop will start the NameNode and ResourceManager in ‘conf/masters’ file.
4. On master node, we also defined the machine on which Hadoop will run the DataNode and NodeManager in ‘conf/slaves’ file.
5. Changed the configuration file ‘conf/core-site.xml’, ‘conf/mapred-site.xml’ and ‘conf/hdfs-site.xml’ on all the machines to include host and port details of the master node.
6. Formatted Hadoop’s distributed filesystem (HDFS) via the NameNode.
7. Started the multi-node cluster using the below two steps

* NameNode daemon is started on master and DataNode daemon are started on all the slave nodes
* ResourceManager is started on master and NodeManger daemons are started on all the slave nodes.

1. Then we run the HDFS daemons and MapReduce Daemons on the master node.
2. Used three ebooks from Project Gutenberg to run the wordcount example. The ebooks are available in the following links

<http://www.gutenberg.org/ebooks/20417>

<http://www.gutenberg.org/ebooks/5000>

<http://www.gutenberg.org/ebooks/4300>

<http://www.gutenberg.org/ebooks/132>

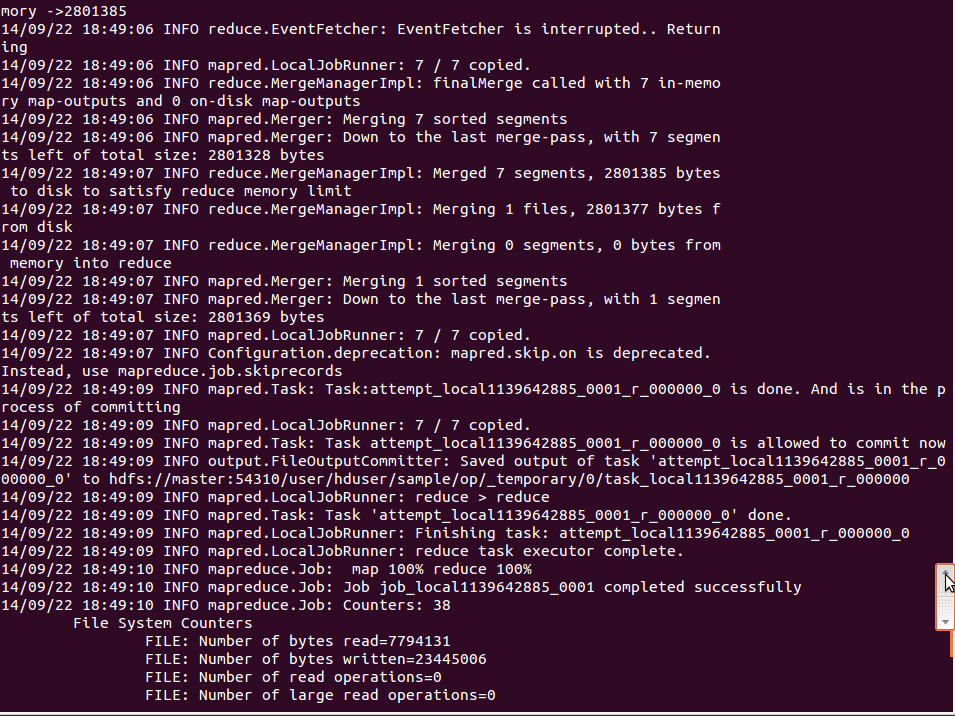
<http://www.gutenberg.org/ebooks/1661>

<http://www.gutenberg.org/ebooks/972>

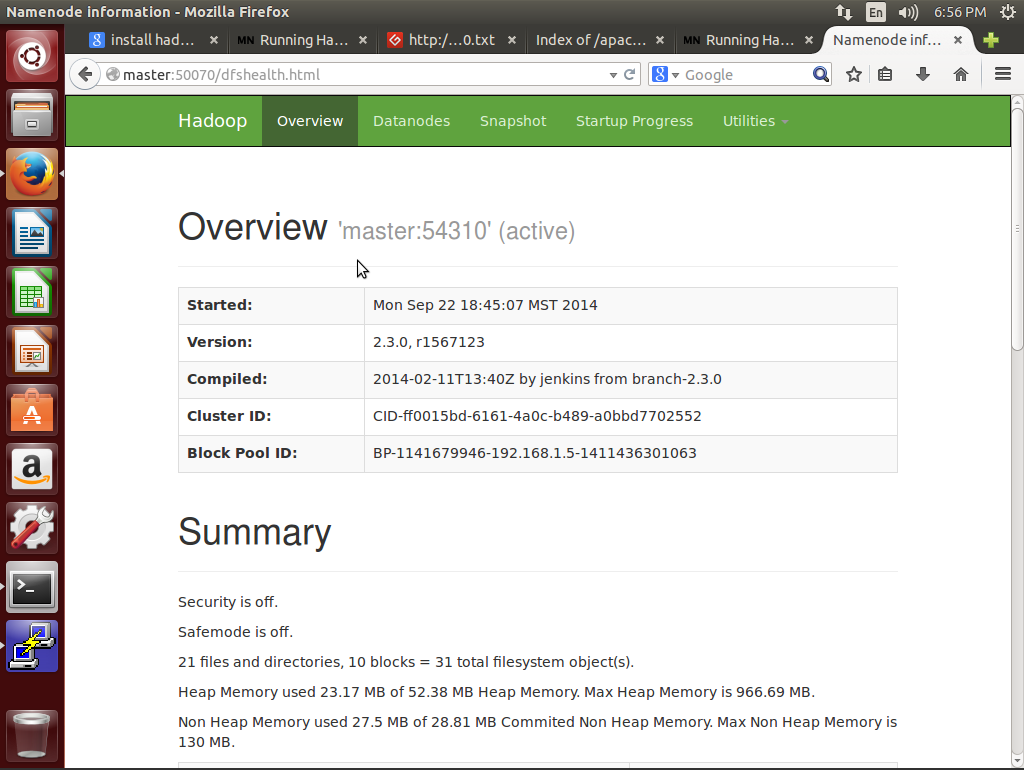
<http://www.gutenberg.org/ebooks/19699>

1. Copied the local data to HDFS
2. Created an executable jar from the wordcount java project, and ran the mapreduce job.
3. Observed the concise information provided about the hadoop cluster summary using the web interfaces.

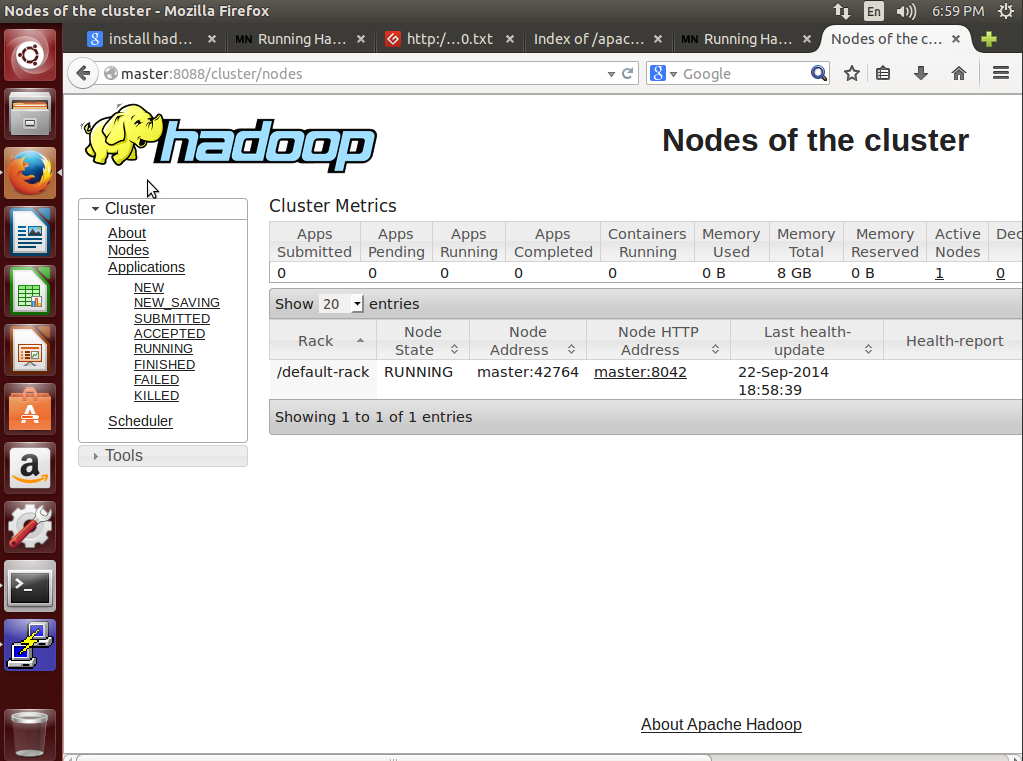
Please find screenshots of the results for running the word count example on Hadoop multi node cluster.



Screenshot of NameNode summary



Screenshot of Resource Manager Summary



**Task 2: Choosing the best learning algorithm for predicting the rating based on the yelp dataset considering accuracy as the primary goal.**

We have chosen Random Forest Classifier to classify the review description obtained as user input, since it is more accurate for numerical data. The Random Forest classifier from Apache Mahout accepts a sequence file generated by the preprocessing step. The classifier generates a model from the training set. The user review is the test data which is classified according to this model.

**Task 3: Preprocessing of Yelp Review Dataset using map-reduce framework.**

The mappers and reducer classes for preprocessing of Yelp Review Dataset are created and pushed to Gitlab repository. Testing the preprocessing of data is complete.

**Task 4: Implementation of Random Forest Classifier using Apache Mahout.**

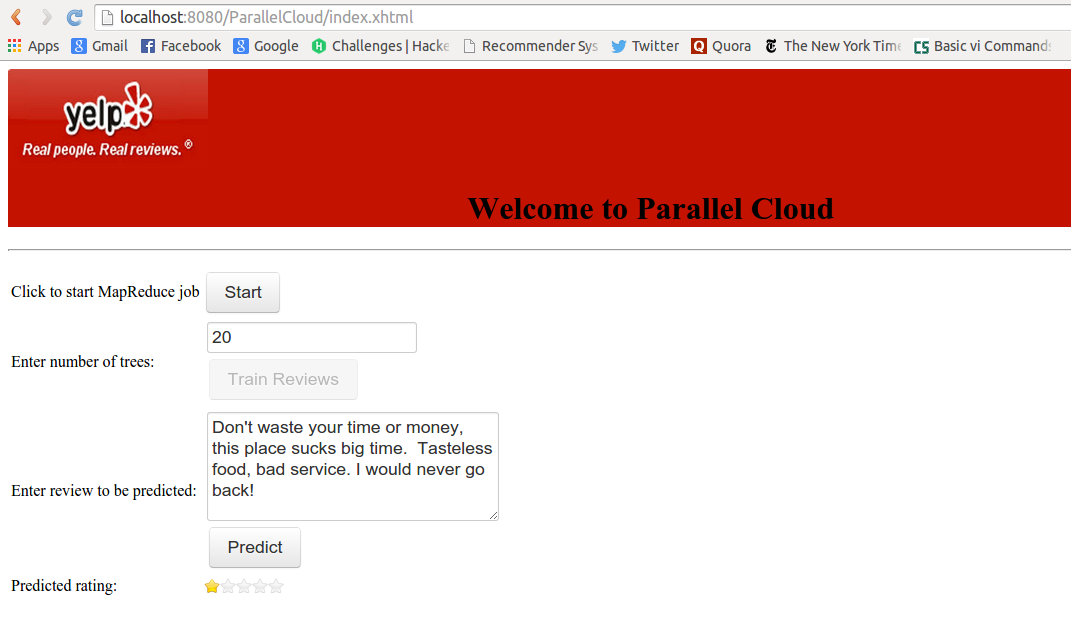
The Sequence File was generated from the preprocessed data obtained in the previous task. For the second half of the task, Random Forest Classifier from Apache Mahout was implemented to learn patterns from the yelp academic dataset and predict the rating for a review given by the user. The classifier used the model generated from training dataset for this purpose. On completion, the classifier predicted a rating for a given input review successfully.

**Task 5: Implementing the User Interface to take user inputs.**

User interface was created to receive input reviews from the user and display the rating predicted by the classifier. We used Twitter Bootstrap & JSF framework to design and implement the UI.

Please find below the screenshots of the user interface and the rating provided for a poor, moderate and good review description respectively.

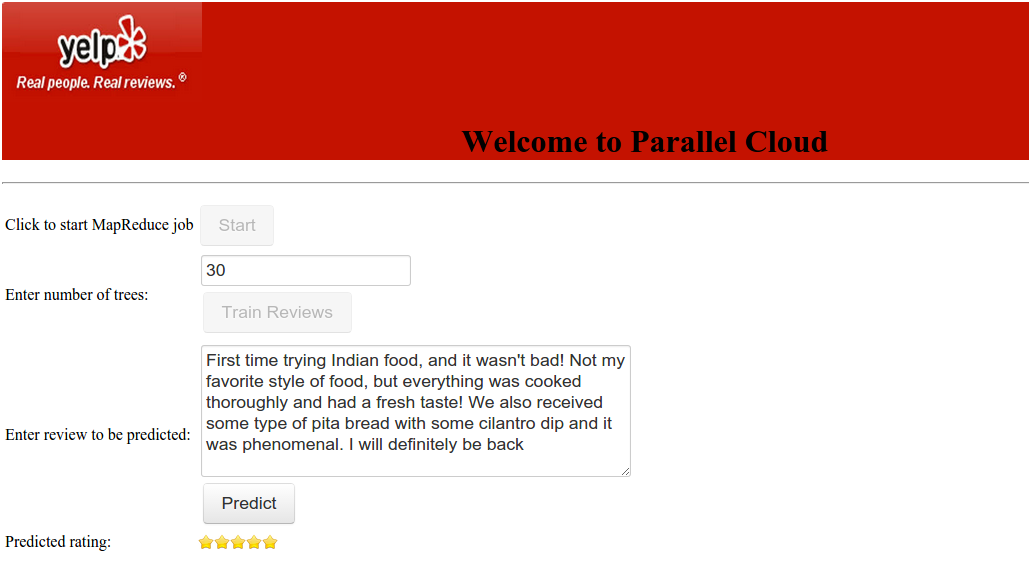
Rating predicted for a poor review.



Rating predicted for a moderate review.



Rating predicted for a good review.



**Technical difficulties**

1. The VM 591101 was dead until 20th of September. Hence, the multi node cluster setup was done on the other three VMs 591103(master), 591102(slave1) and newly allocated VM 591104(slave2).

2. Faced VM disconnection issue initially, which hindered our progress with setting up Hadoop cluster on Vlab.

3. The VMs were slow and required TA/mentor’s frequent intervention to restart the machines.

**Known Issues**

1. Number of trees for training the data cannot exceed more than 30 due to system memory limits. The number may be less for the Vlab VMs since the available RAM is 1GB.

**Build Instructions**

Please follow the below build instructions to setup and run the project:

1. Import ‘parallelcloud’ and ‘ParallelCloudWeb’ into Eclipse workspace and do maven update on both the projects.

2. Change corresponding file paths in ‘WordsReaderHelper.java’, ‘YelpBean.java’ and ‘Yelp.java’ to point to helper files on local file system. Helper files are used to determine feature set score for a review. The files are located under parallelcloud/data directory.

3. Assuming maven is installed, go to command line and cd into workspace/parallelcloud.

4. Do **mvn clean compile assembly:single**. A JAR will be generated in parallelcloud/target.

5. Copy this JAR into ParallelCloudWeb/target.

6. Add ParallelCloudWeb to Tomcat and start the server. Go to index.xhtml and click run on server.

7. Start MapReduce job once from the web page

8. Provide 20 as input to number of trees field and train the classifier once by clicking the ‘train’ button. You can track the completion of this step in eclipse console.

9. Upon completion of previous step, enter a review description to get a prediction.

**User Interaction**

* Start the MapReduce process for the first time. Note that for subsequent runs the button to start the MapReduce job is disabled since it was completed once.
* Enter the number of trees for the training the dataset. Note that a number greater than 30 may cause memory issues depending on memory availability. Button to train the dataset will be disabled once it is trained.
* Enter a review in the text box and get a prediction.

**Technical Dependencies.**Please run the mapreduce job part of ‘ParallelCloud’ project initially to preprocess the yelp review dataset and generate the preprocessed data- ‘output.csv’. This file is necessary for the random forest classifier implemented in the project -’ParallelCloudWeb’. Additional details can be found in the user guide section.

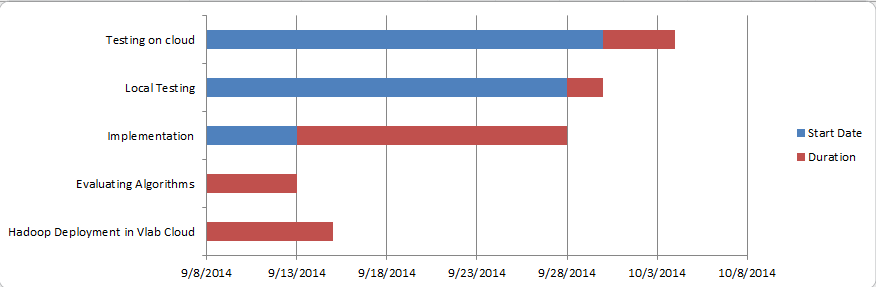
1. Project Deliverables And Timeline

## **Deliverables**

1. Stable deployment of Hadoop environment on Vlab cloud.
2. A runnable mapreduce algorithm to do a sentiment analysis over the yelp dataset and obtain ratings based on reviews provided by the users.
3. Implementation of the mapreduce algorithm for sentiment analysis over the yelp dataset and a deployable application that can be run and demonstrated.

## **Project Timeline**

|  |  |  |
| --- | --- | --- |
| Task | Start Date | Duration |
| Hadoop Deployment in Vlab Cloud | 9/8/2014 | 7 |
| Evaluating Algorithms | 9/8/2014 | 5 |
| Implementation (Data preprocessing, implementing classifier, UI design and creation) | 9/13/2014 | 15 |
| Testing on local machine | 9/28/2014 | 2 |
| Testing on the cloud | 9/30/2014 | 4 |



# VIII. Risk Management of the project

|  |  |  |
| --- | --- | --- |
| **Risk Type** | **Severity** | **Mitigation** |
| Hadoop Deployment | High | Consult mentor/TA and have a stable Hadoop deployment |
| Schedule | Medium | Basic tasks and operations to be completed on priority, followed by other features |
| Defects | Medium | High priority defects affecting basic operations will be handled |
| Algorithm implementation difficulty | High | Selecting implementable algorithm which is fairly accurate |

# IX. Conclusion

The major task of this project is to understand the requirement and structure of a Hadoop mapreduce environment and to implement and deploy an application developed using mapreduce algorithm to solve sentimental analysis on a private section in a public cloud. The application should be able to take input review description from the user and predict a rating based on the model generated using the training yelp dataset.

As a future work, we can include an interest set that can be provided as a search criterion to let the user filter the reviews based on multiple categories like location, check-in sets and tips etc. It is often the case that a review and its corresponding rating provided by the user are subjective and not consistent. Hence, this application can be used to get consistent ratings and the same can be used to effectively recommend businesses that fall in the similar rating categories.

##### Acknowledgment

We would like to thank our mentor, Mr. Chun-Jen Chung for helping us refine the ideas to kick start this project and guiding us in the right direction.

##### References

1. http://citizennet.com/blog/2012/11/10/random-forests-ensembles-and-performance-metrics/
2. Jong, Jason. "Predicting Rating with Sentiment Analysis." (2011).
3. Pang, Bo, and Lillian Lee. "Opinion mining and sentiment analysis."*Foundations and trends in information retrieval* 2.1-2 (2008): 1-135.
4. http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html
5. <http://nlp.stanford.edu/sentiment/index.html>
6. https://developer.yahoo.com/hadoop/tutorial/