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 (*Describe Project Goal*)

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.

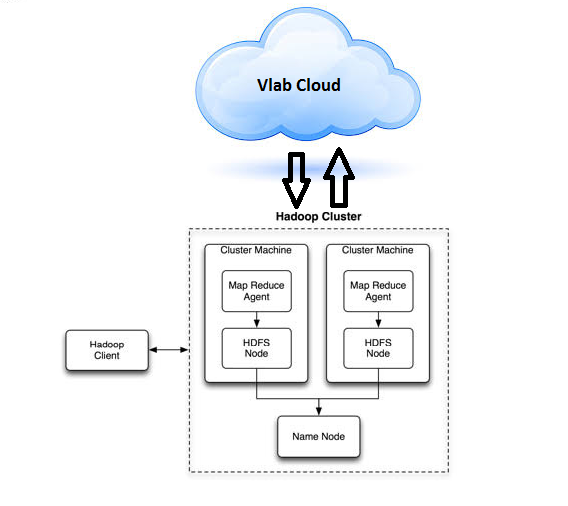
The rating can be predicted using various machine learning techniques such as Naive Bayes Classifier or support vector machine (SVM) learning algorithms.

The user should be able to filter review descriptions for certain type of businesses and obtain the corresponding review ratings for the review descriptions listed in the results. The application queries the yelp dataset to list the review descriptions based on user inputs and perform sentiment analysis on them to predict the review rating.

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

# System Models

## System Model



## Software

Programming Language: Java

Libraries: **Apache Mahout (Classification)**

Tools: Eclipse, Git, Hadoop 2.3.0 Environment

# Project Description

The goal of this project is to deliver appropriate rating for a particular type of business based on the reviews 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

**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.

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

Understanding the Vlab cloud architecture and analyzing how to set up the 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 order to provide the appropriate rating and 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. Hence this task will involve designing the right algorithm for our dataset considering accuracy of the rating as the goal.

## Task 3: Preprocessing of Yelp Review Dataset and Implementation of the above chosen algorithm using map-reduce framework.

In the previous task, we decide on the best learning technique and design the algorithm for learning patterns from the dataset and predicting the rating based on the reviews. In this task we implement the algorithm designed using the map-reduce framework using Java as the programming language.

## 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, Dhanyantha, 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 | **In Progress** |
| Testing | Rajath, Dhanyatha, Sonali | 10/4 | 5 | **In Progress** |
| Report Generation | Rajath, Dhanyatha, Sonali | 10/4 | 5 | **Todo** |

**Task Progress Report**

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

**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

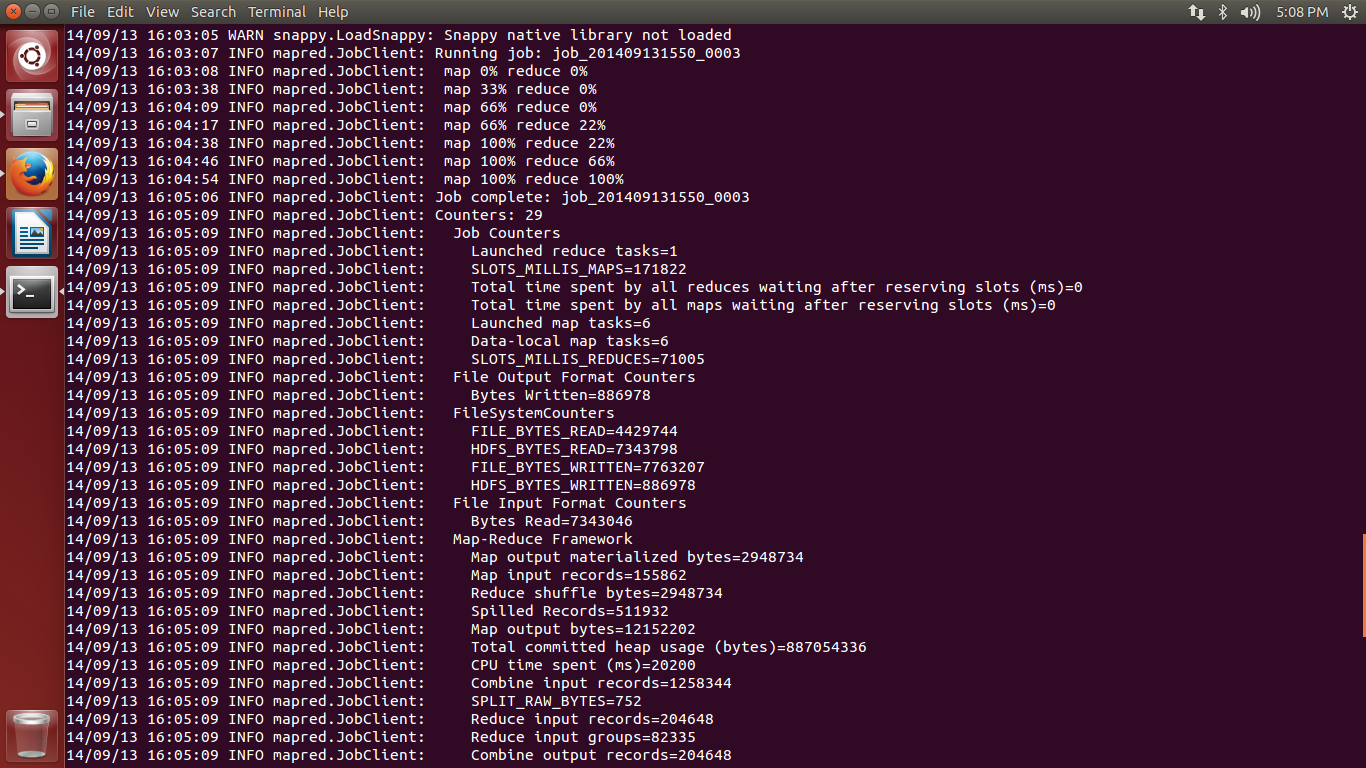
13. Observed the concise information provided about the hadoop cluster summary using the below web interfaces.

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

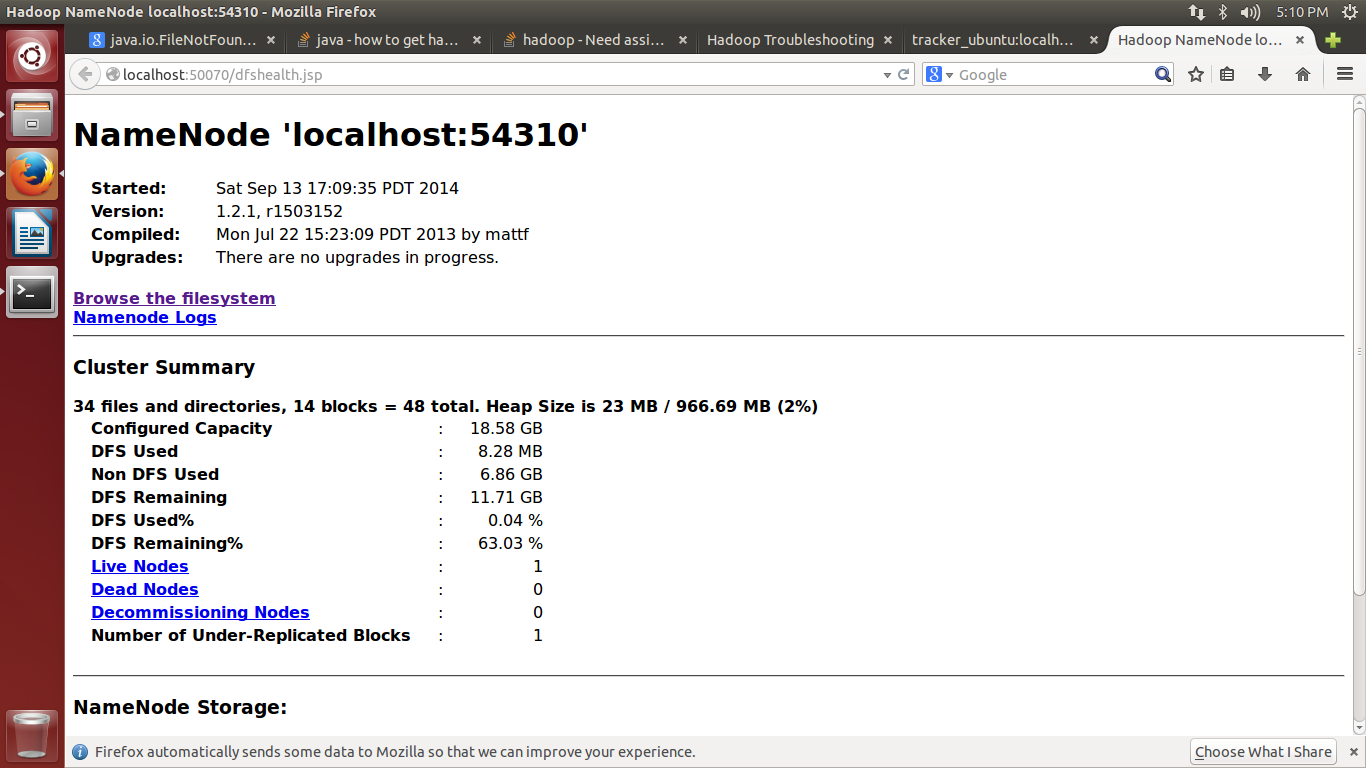
<http://localhost:50030/> – web UI of the JobTracker daemon

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

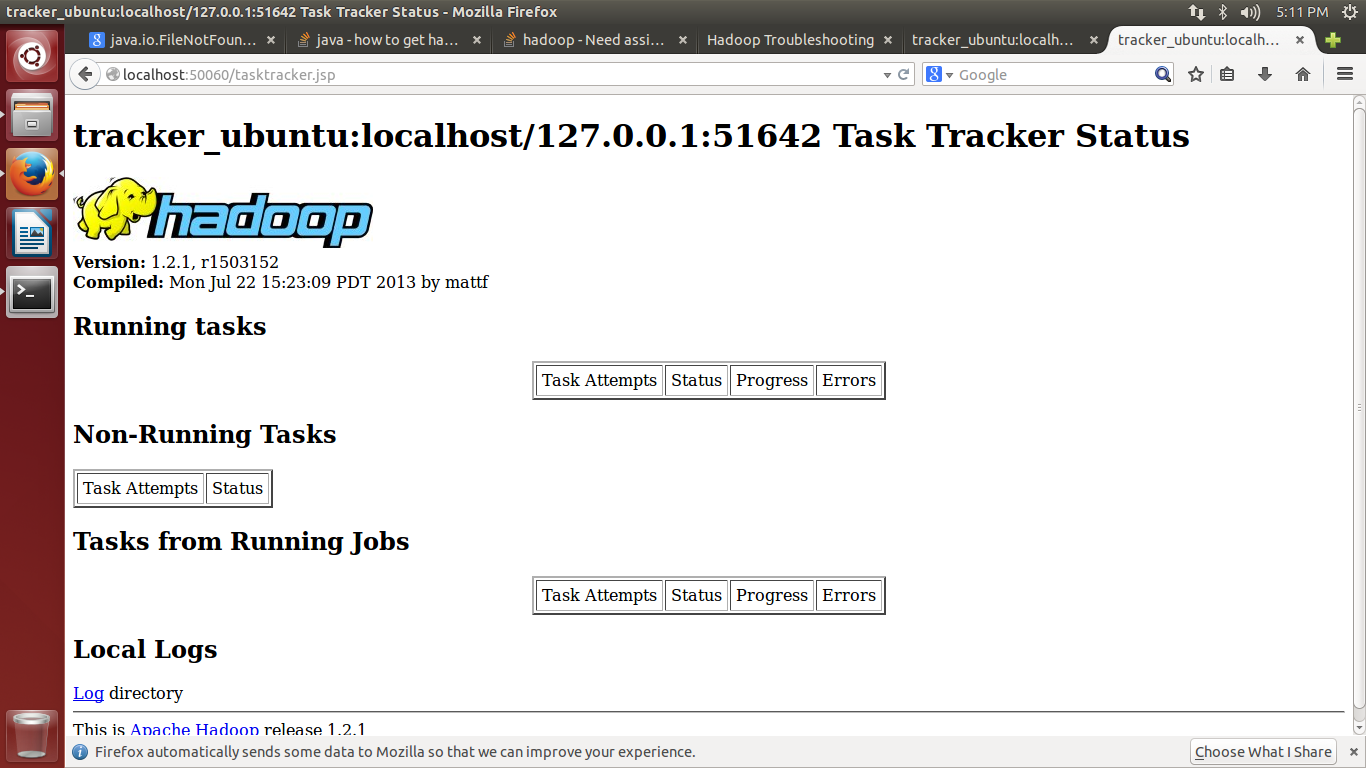
Please find screenshots of the results for running the word count example using hadoop.



Screenshot of Namenode Summary



Screenshot of Task Tracker Status



**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 setup 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>

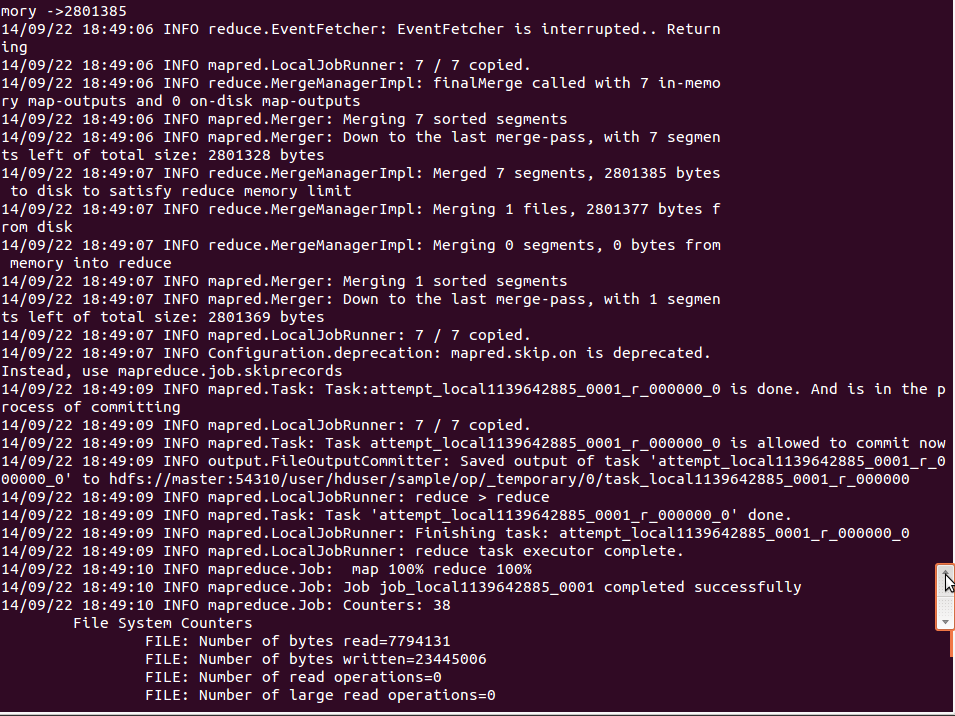
<http://www.gutenberg.org/ebooks/1661>

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

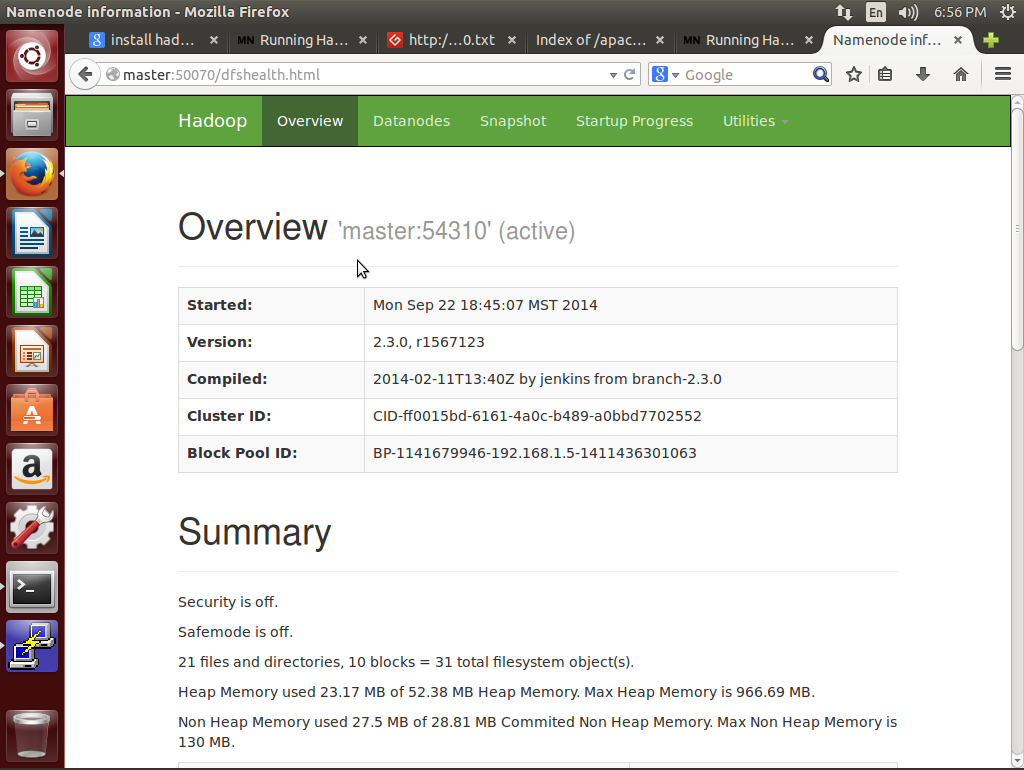
<http://www.gutenberg.org/ebooks/19699>

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.

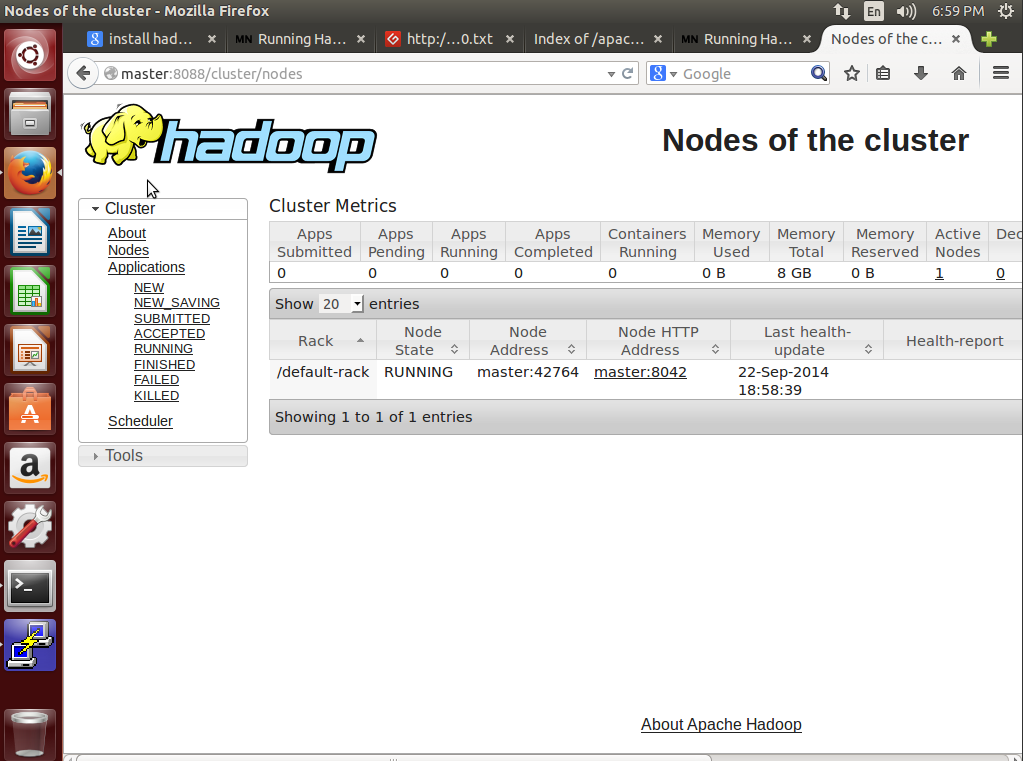
Please find screenshots of the results for running the word count example on Hadoop multi node cluster.



Screenshot of NameNode summary



Screenshot of Resource Manger 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 in progress.

**Technical Difficulties Encountered:**

1. The VM 591101 was dead until 20th of September. Hence, the multi node cluster setup was only done on the other two VMs 591103(master) and 591102(slave).

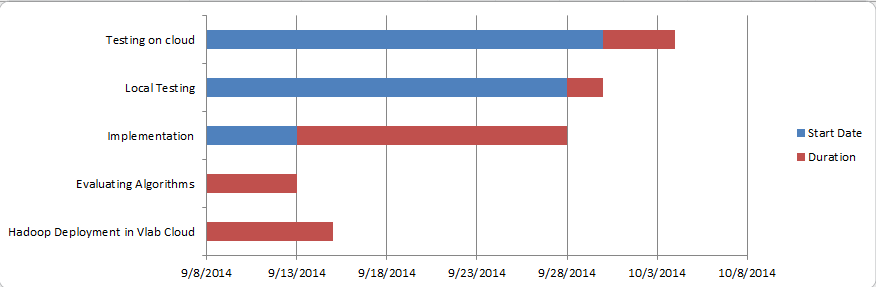
2. Faced VM disconnection issue initially, which hindered our progress with setting up Hadoop cluster on Vlab.

## Deliverables

1. Stable deployment of Hadoop environment on Vlab cloud.
2. A runnable map-reduce algorithm to do a sentiment analysis over the yelp dataset and obtain ratings based on reviews provided by the users.
3. Implementation of the map-reduce 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 | 9/13/2014 | 15 |
| Testing on local machine | 9/28/2014 | 2 |
| Testing on the cloud | 9/30/2014 | 4 |



# 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 |

# 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 user inputs to filter review descriptions in the yelp dataset for a particular business category and list the corresponding review ratings.

As a future work, we can include a larger interest set that can be included in the search criteria 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 kickstart this project and guiding us in the right direction.

##### References

1. Jong, Jason. "Predicting Rating with Sentiment Analysis." (2011).
2. Pang, Bo, and Lillian Lee. "Opinion mining and sentiment analysis."*Foundations and trends in information retrieval* 2.1-2 (2008): 1-135.
3. http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html
4. http://nlp.stanford.edu/sentiment/index.html