BIG DATA HADOOP AND SPARK DEVELOPMENT

ASSIGNMENT – 3

Table of Contents:

BIG DATA HADOOP AND SPARK DEVELOPMENT

1. Introduction

In this assignment, the given task is performed and Output of the task is performed and Screenshots are attached.

2. Objective

This assignment consolidates the deeper understanding of the Session – 3 Introduction to YARN (Yet Another Resources Negotiator) and High level YARN components.

3. Problem Statement

- Task 1 - Execute WordMedian program,
- Task 2 -Execute WordMean program,
- Task 3 -Execute WordStandardDeviation programs using

hadoop-mapreduce-examples-2.9.0.jar file present in your AcadGild VM.

4. Expected Output

To perform the given tasks, first we need to copy a text file(file327.txt) with the size of 312Mb from local to Acadgild VM.

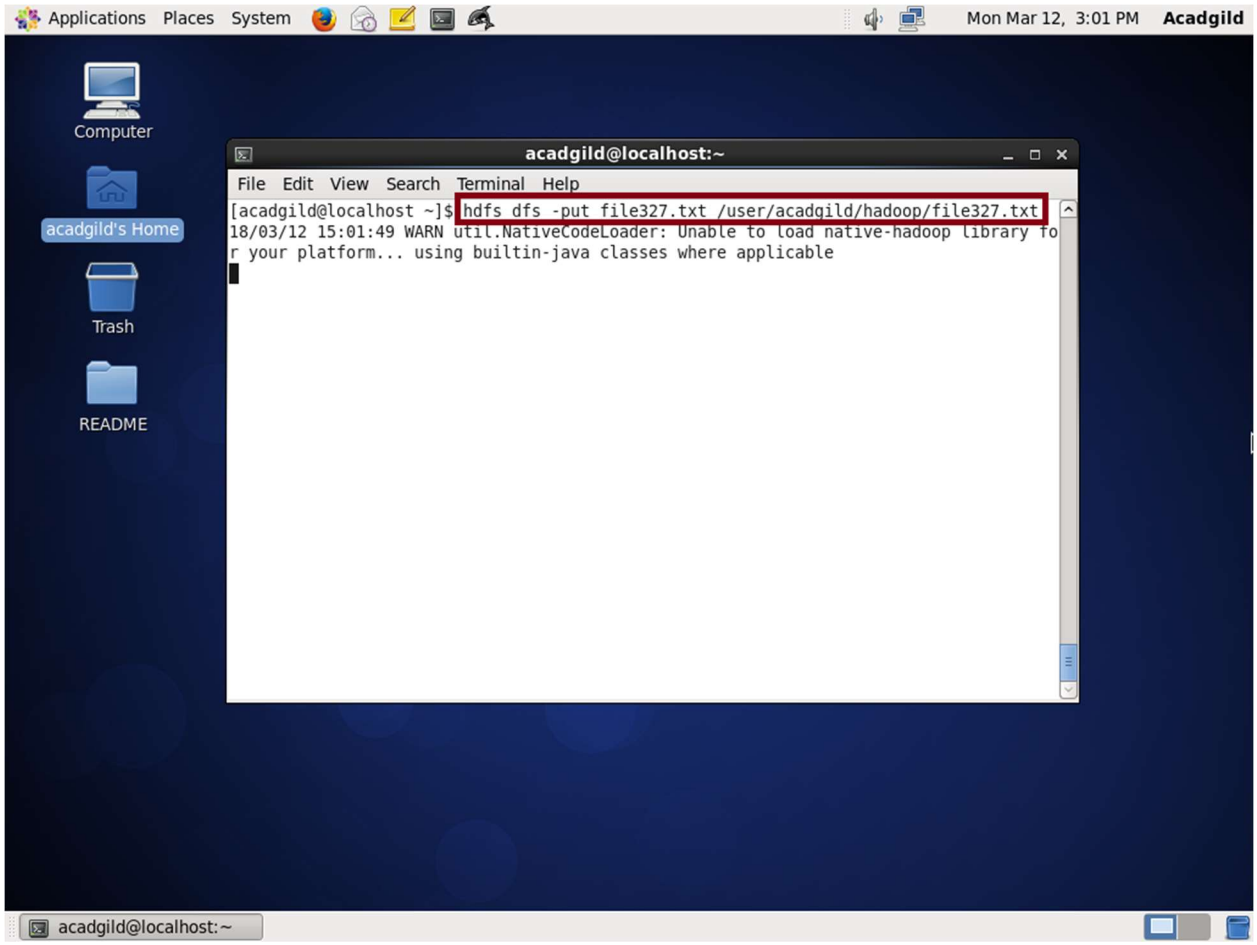
Preparing to perform tasks:

By copying a text file (file327.txt) with the size of 312Mb from local to Acadglid VM.

hdfs dfs –put file327.txt /user/acadgild/Hadoop/file327.txt

by using **–put** command the **file327.txt** is copied from local to **Acadgild VM**

The following screenshot show the process of copying **file327.txt** from local to **Acadgild VM**.



hadoop [--config conf dir]

jar <jar> run a jar file

by using the above syntax, we can run the jar file

- <u>Task 1 – Executing Word Median program</u>

Word Median:

A Map/Reduce program that counts the median length of the words in the input files.

hadoop <hadoop jar file path> <program name> <path of the file name> <directory name where the output can be stored>

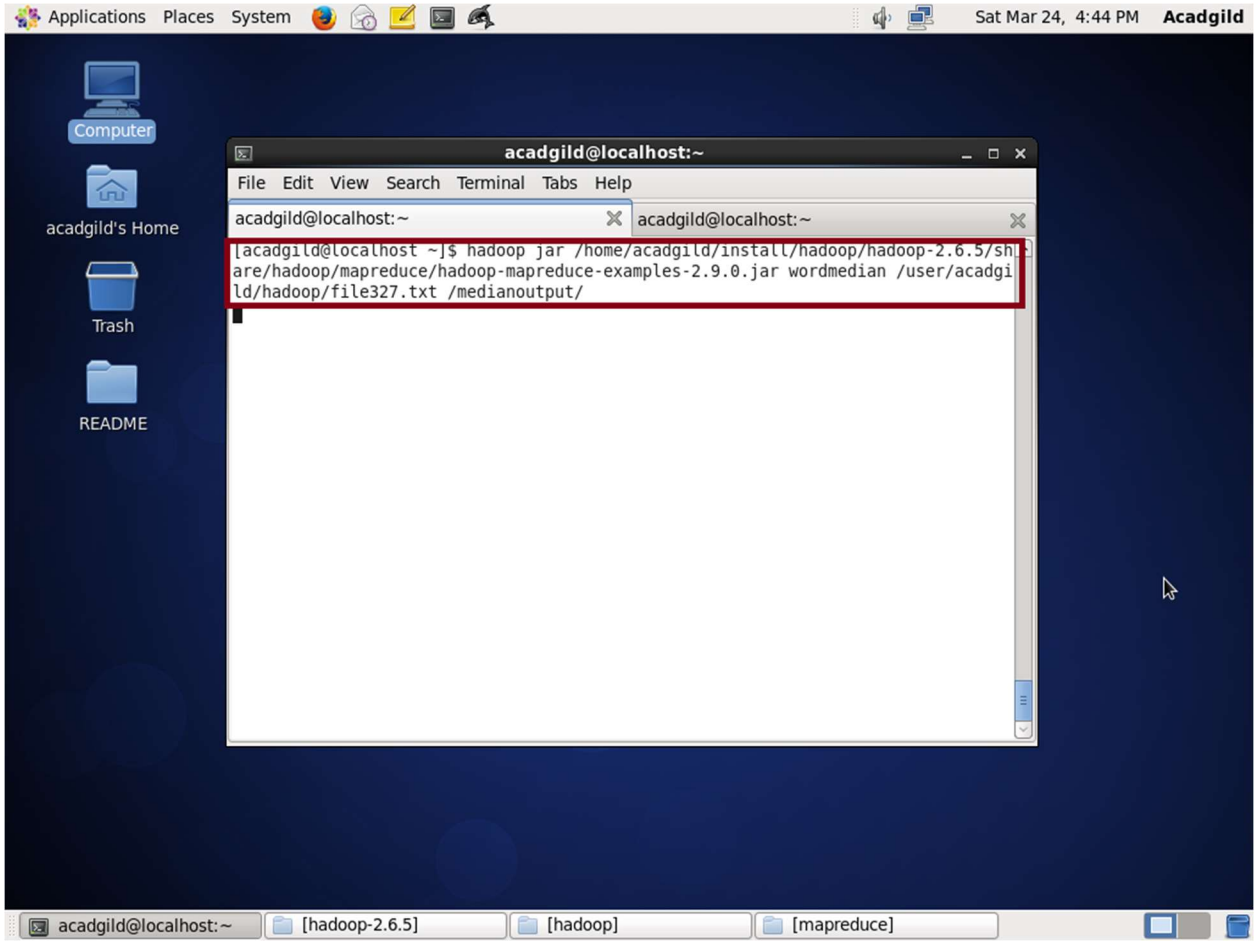
the above is the syntax for the jar file to run and output save in the directory.

To run wordmedian program

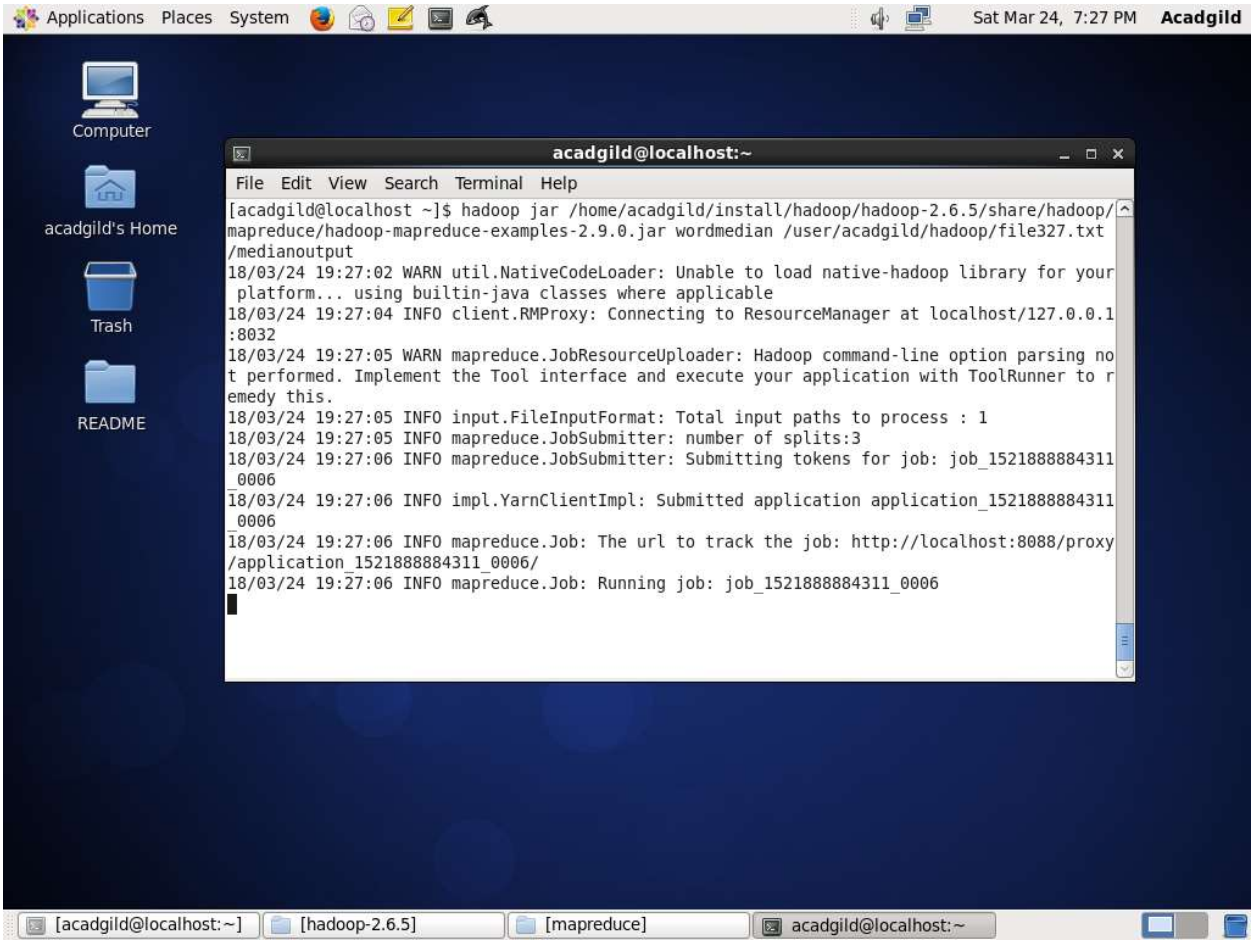
- jar file path is /home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.9.0.jar
- program name is wordmedian
- File path is /user/acadgild/hadoop/file327.txt

The command which is used here is

hadoop jar /home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.9.0.jar wordmedian /user/acadgild/hadoop/file327.txt /medianoutput



Word median program is executed and the Map reduce process is performed.



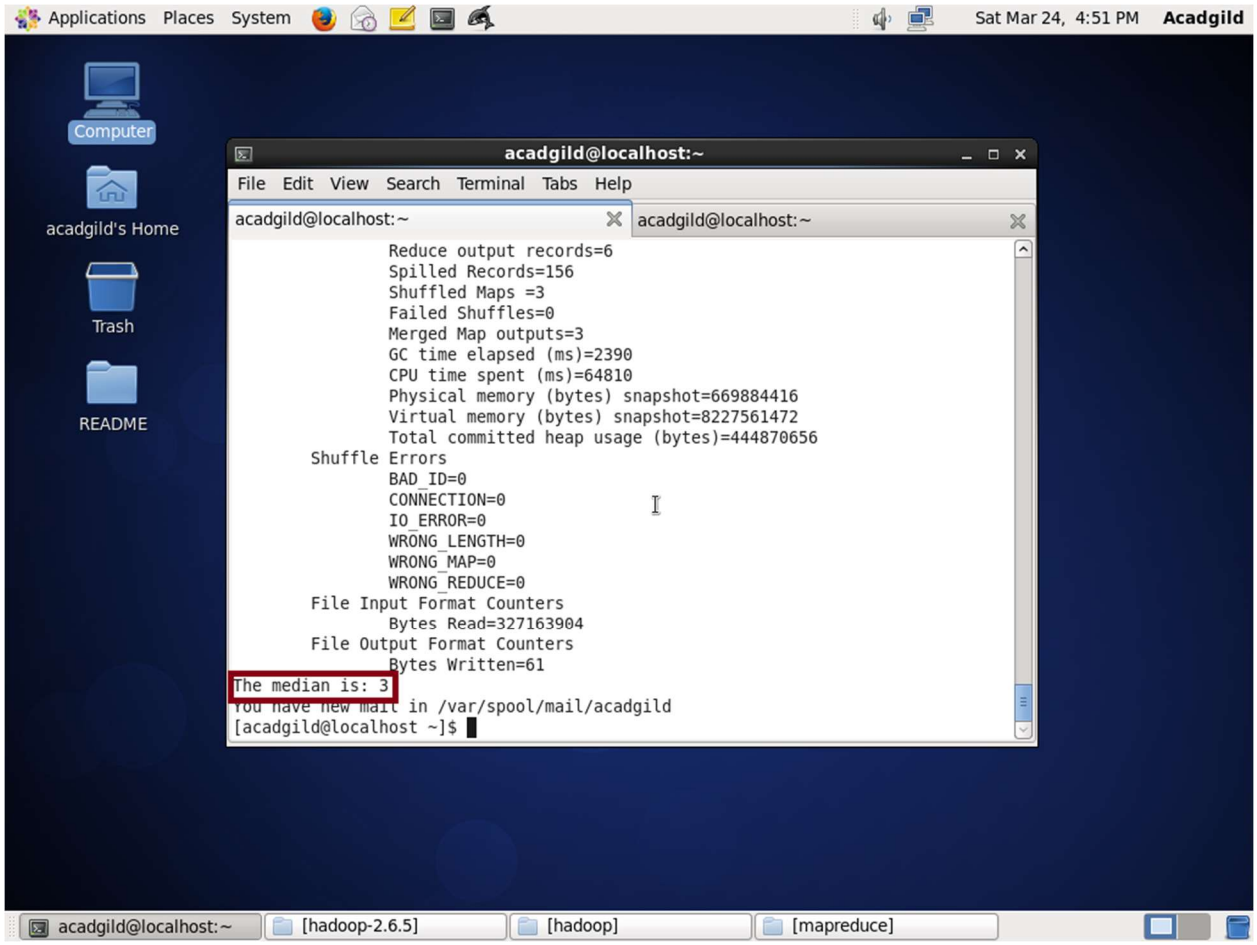
As in individual series (if number of observation is very low) first one must arrange all the observations in order. Then count(n) is the total number of observation in given data.

If ***n* is odd** then Median (M) = value of ((n + 1)/2)th item term.

If ***n* is even** then Median (M) = value of [(n/2)th item term + (n/2 + 1)th item term]/2

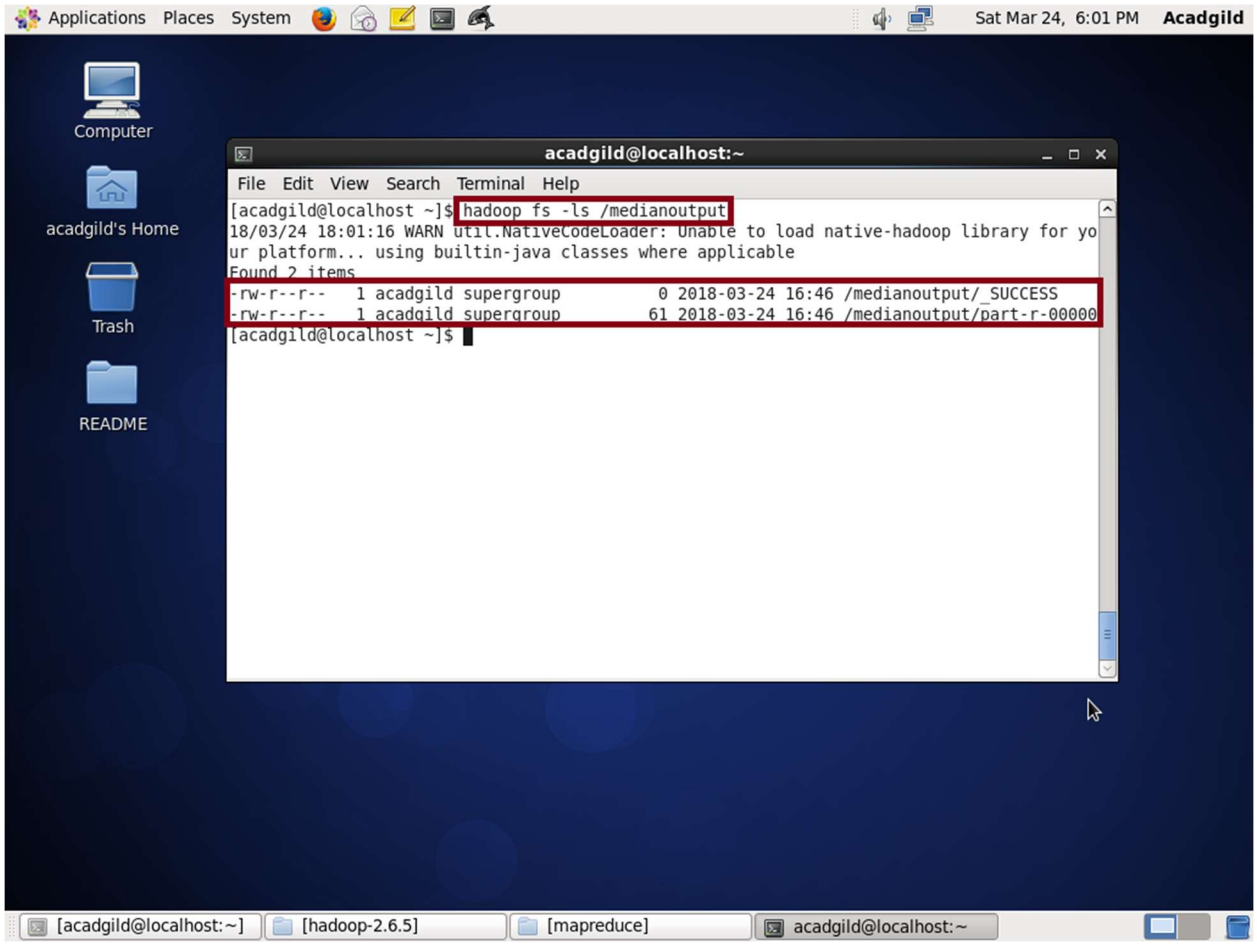
The Median is calculated as 3.

And the output is saved in **medianoutput** directory.



The output is saved in **medianoutput** directory, which is listed by following command,

hadoop fs –ls /medianoutput

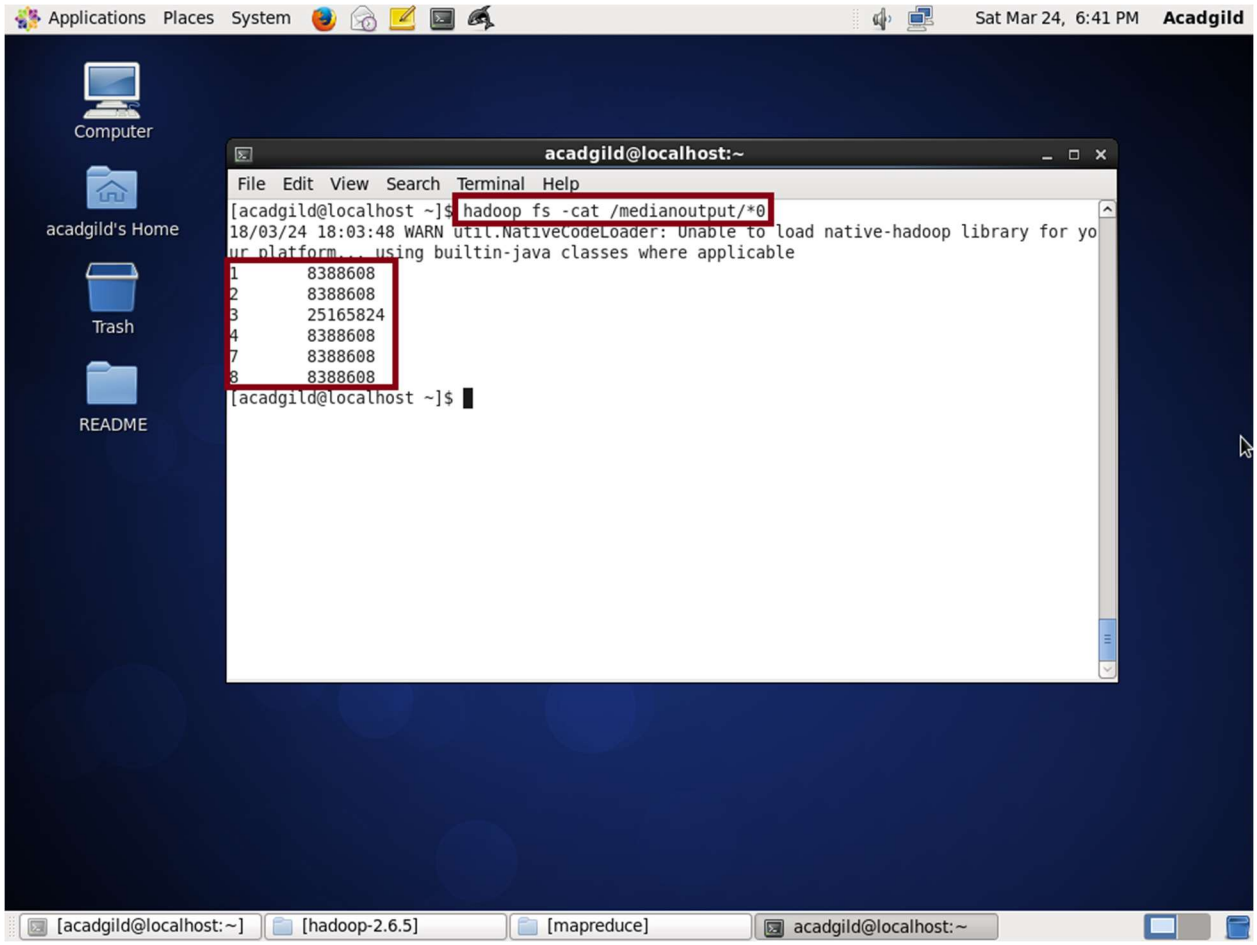


The data is viewed by using

hadoop fs –cat /medianoutput/part-r-00000

or

hadoop fs –cat /medianoutput/*0



- <u>Task 2 – Executing Mean Program</u>

Word mean:

A Map/ Reduce program that counts the average length of the words in the input files.

hadoop <hadoop jar file path> <program name> <path of the file name> <directory name where the output can be stored>

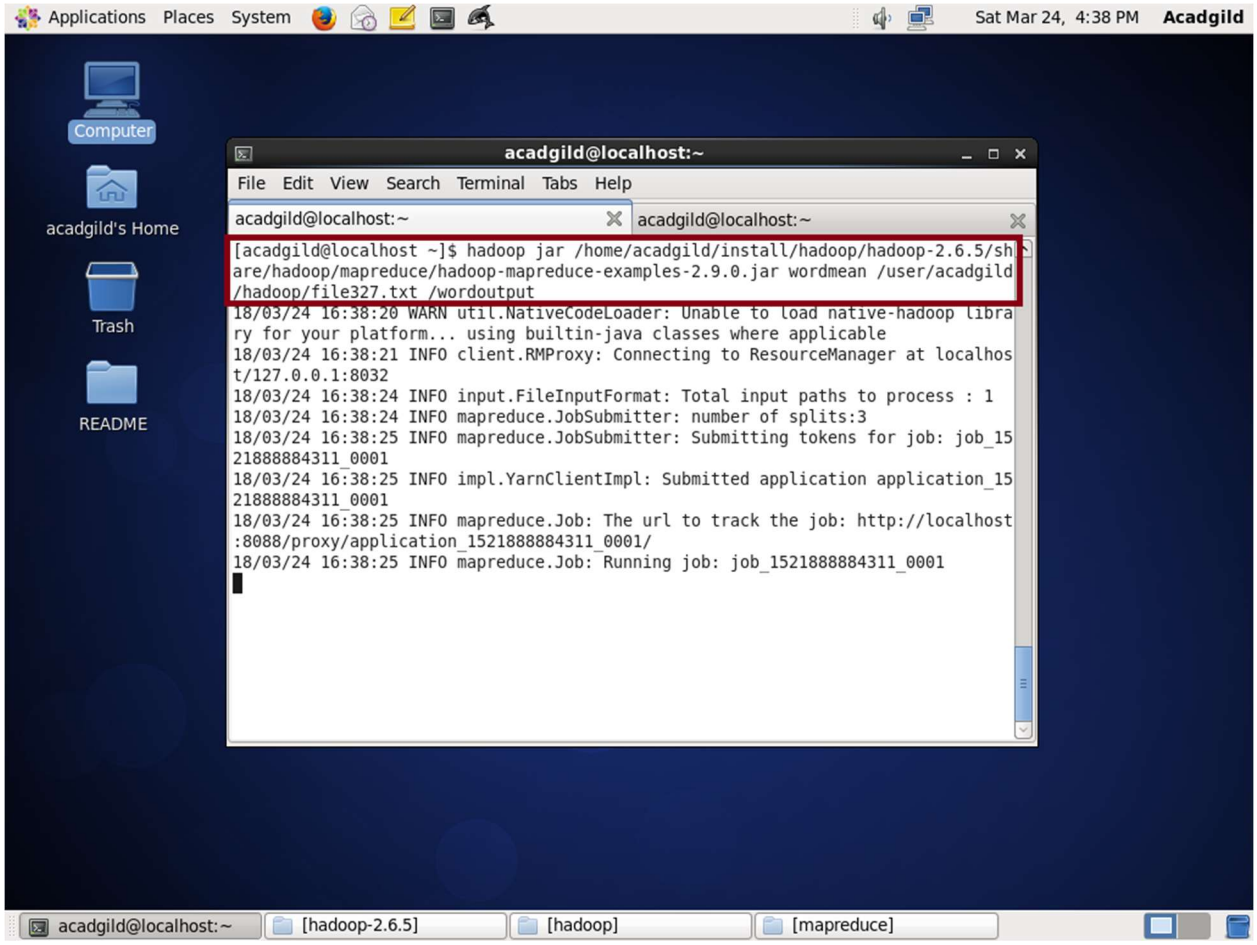
the above is the syntax for the jar file to run and output save in the directory.

To run wordmean program

- jar file path is /home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.9.0.jar
- program name is wordmean
- File path is /user/acadgild/hadoop/file327.txt
- Output directory is /wordoutput

The following command is used to execute the wordmean program and save the output in the wordoutput directory.

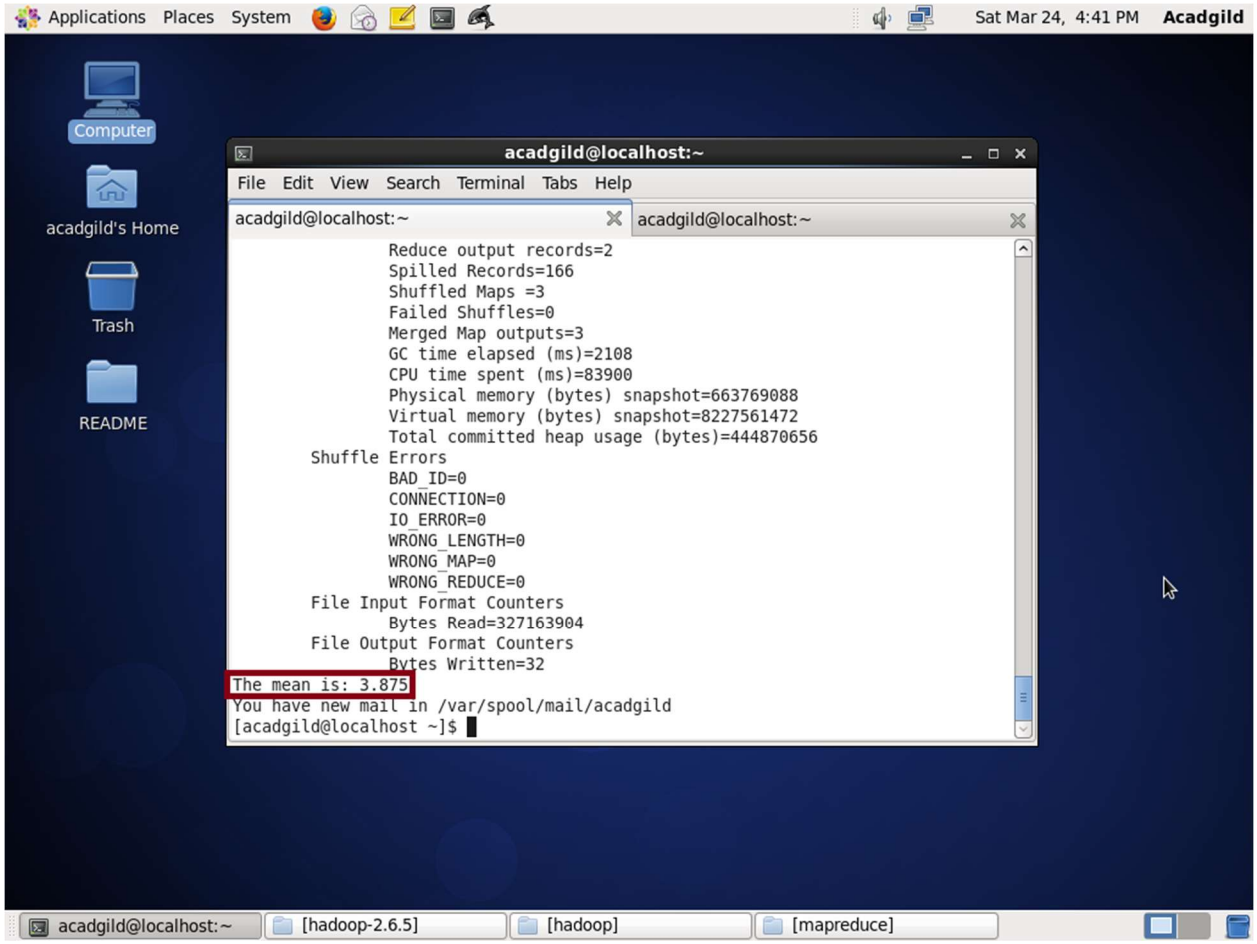
hadoop jar /home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.9.0.jar wordmean /user/acadgild/hadoop/file327.txt /wordoutput



Word Mean is calculated by

The word mean is the average of the number of input words in the text file calculated by a "central" value of a set of number of input words in the text file.

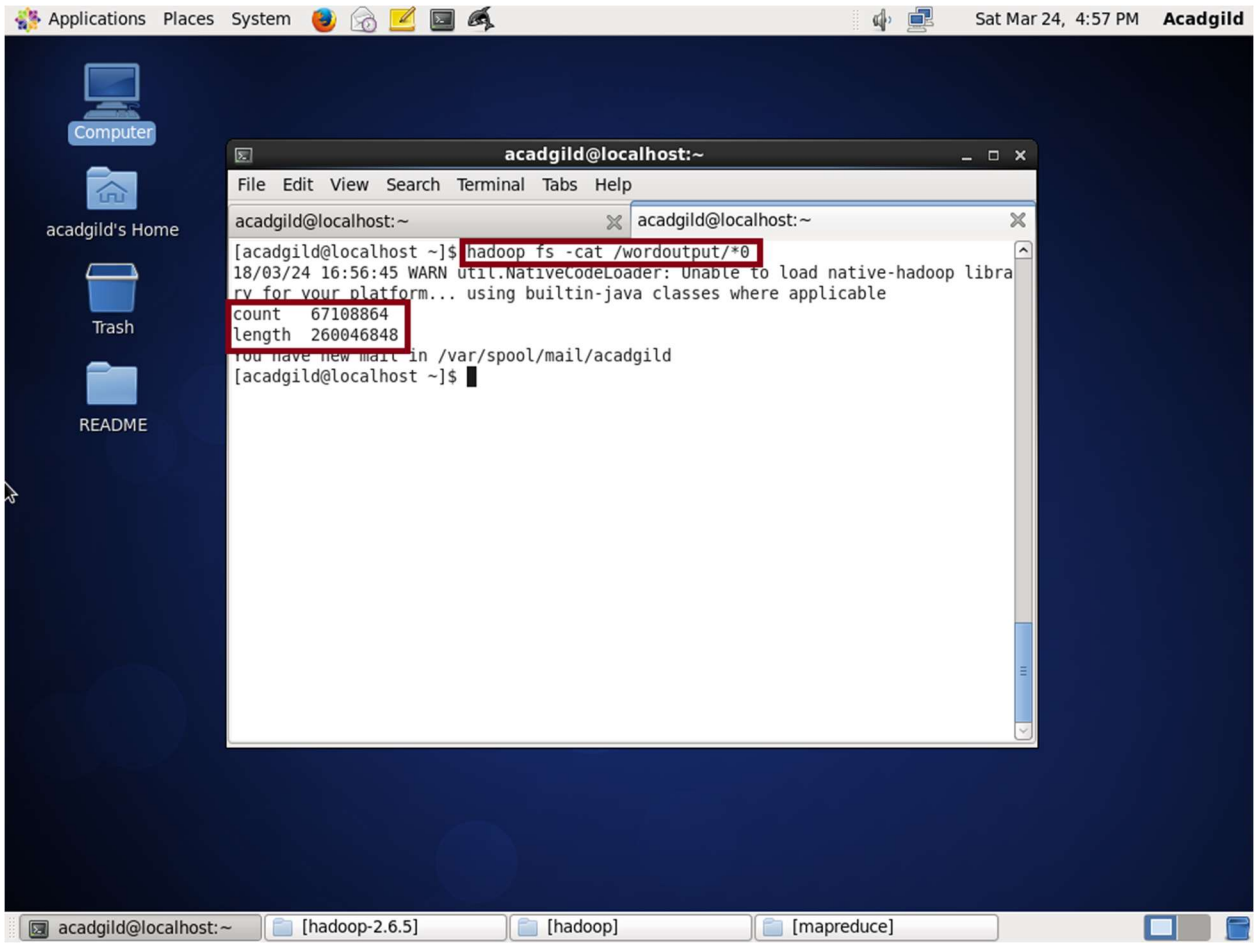
Map reduce process is performed and Mean is displayed.



Mean is **3.875**

The output of the wordmean program is saved in the wordoutput directory and it can be displayed by following command,

hadoop fs –cat /wordoutput/*0



- Task 3- Executing word standard deviation program

Word Standard Deviation:

A Map/Reduce program that counts the standard deviation of the length of the words in the input files.

hadoop <hadoop jar file path> <program name> <path of the file name> <directory name where the output can be stored>

the above is the syntax for the jar file to run and output save in the directory.

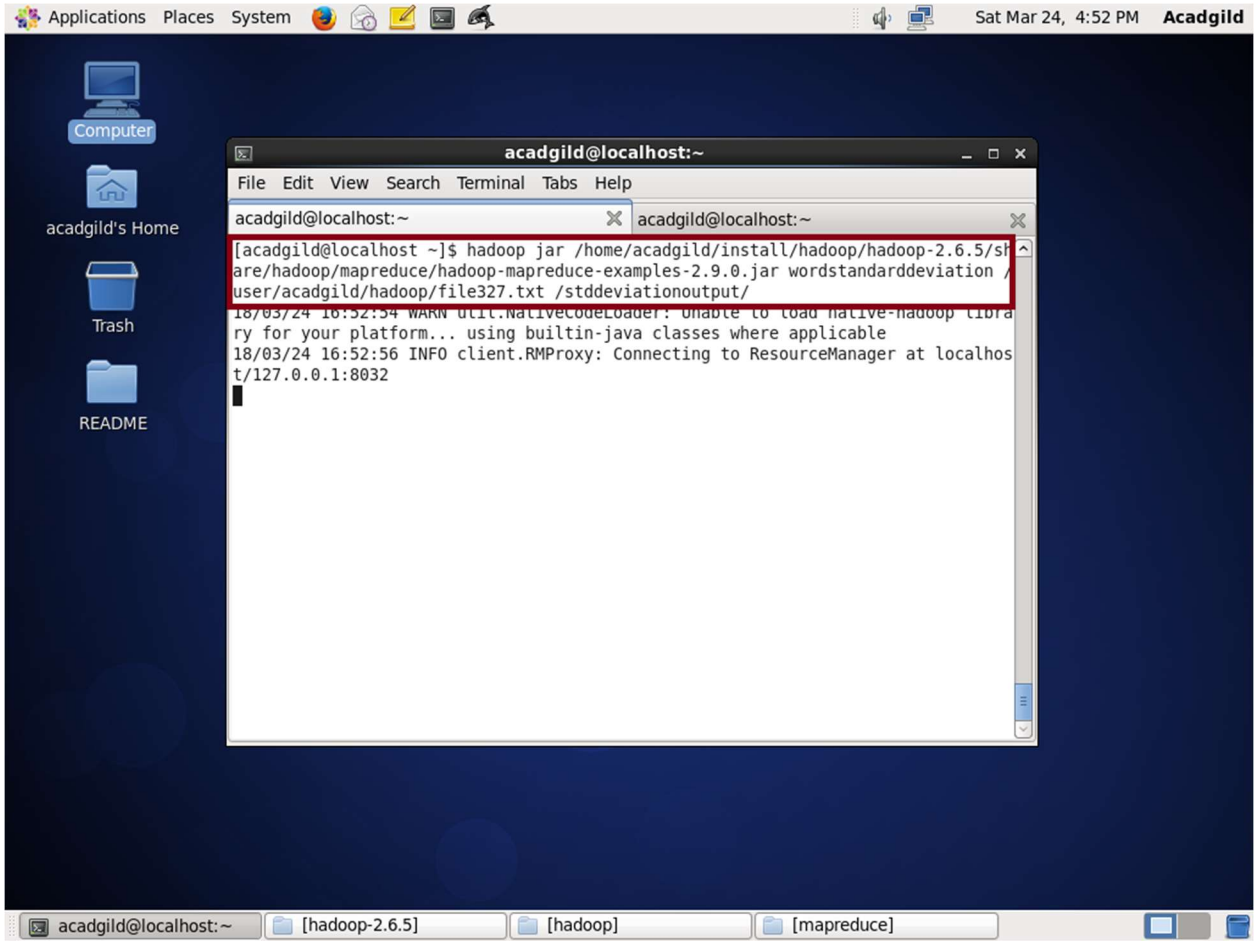
To run wordstandarddeviation program

- jar file path is /home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.9.0.jar
- program name is wordstandarddeviation
- File path is /user/acadgild/hadoop/file327.txt

Output directory is /stddeviationoutput

The following command is used to execute the wordstandarddeviation program and save the output in the stddeviationoutput directory.

hadoop jar /home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.9.0.jar wordstandarddeviation /user/acadgild/hadoop/file327.txt /stddeviationoutput

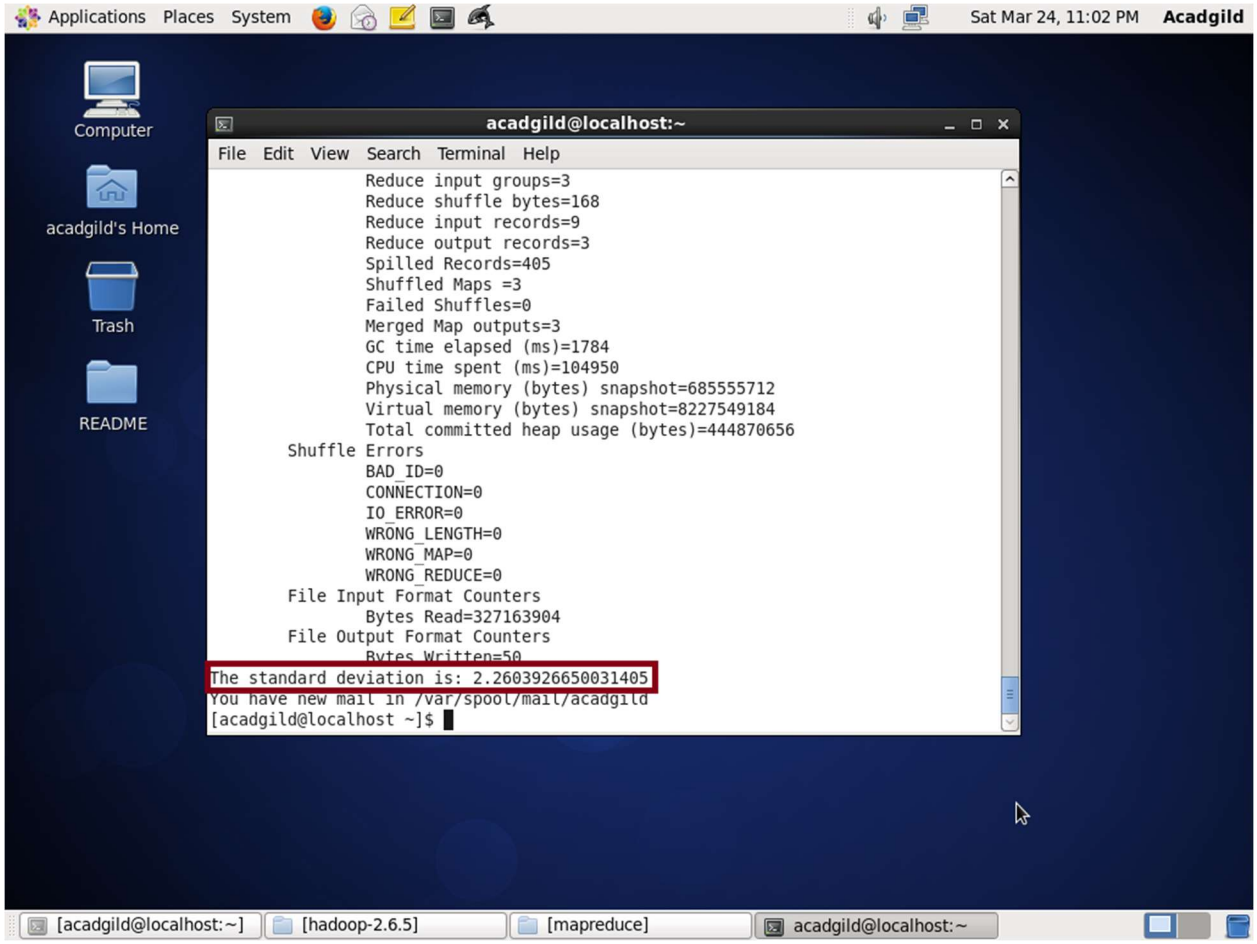


The **Standard Deviation** is a measure of how spread out numbers are. It is the square root of the Variance, and the Variance is the average of the squared differences from the Mean.

The map/reduce process is performed.

```
[acadgild@localhost ~]$ hadoop jar /home/acadgild/install/hadoop/hadoop-2.6.5/share/hadoop/map
educe/hadoop-mapreduce-examples-2.9.0.jar wordstandarddeviation /user/acadgild/hadoop/file327.
xt /stddeviationoutput
18/03/24 22:58:39 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your pl
tform... using builtin-java classes where applicable
18/03/24 22:58:40 INFO client.RMProxy: Connecting to ResourceManager at localhost/127.0.0.1:80
2
18/03/24 22:58:42 INFO input.FileInputFormat: Total input paths to process : 1
18/03/24 22:58:42 INFO mapreduce.JobSubmitter: number of splits:3
18/03/24 22:58:43 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1521888884311_00
7
18/03/24 22:58:43 INFO impl.YarnClientImpl: Submitted application application_1521888884311_00
7
18/03/24 22:58:43 INFO mapreduce.Job: The url to track the job: http://localhost:8088/proxy/ap
lication_1521888884311_0007/
18/03/24 22:58:43 INFO mapreduce.Job: Running job: job_1521888884311_0007
18/03/24 22:58:54 INFO mapreduce.Job: Job job_1521888884311_0007 running in uber mode : false
18/03/24 22:58:54 INFO mapreduce.Job:  map 0% reduce 0%
18/03/24 22:59:17 INFO mapreduce.Job:  map 4% reduce 0%
18/03/24 22:59:23 INFO mapreduce.Job:  map 6% reduce 0%
18/03/24 22:59:26 INFO mapreduce.Job:  map 7% reduce 0%
18/03/24 22:59:29 INFO mapreduce.Job:  map 10% reduce 0%
18/03/24 22:59:32 INFO mapreduce.Job:  map 11% reduce 0%
18/03/24 22:59:33 INFO mapreduce.Job:  map 12% reduce 0%
18/03/24 22:59:35 INFO mapreduce.Job:  map 13% reduce 0%
18/03/24 22:59:39 INFO mapreduce.Job:  map 17% reduce 0%
18/03/24 22:59:42 INFO mapreduce.Job:  map 19% reduce 0%
18/03/24 22:59:45 INFO mapreduce.Job:  map 20% reduce 0%
18/03/24 22:59:48 INFO mapreduce.Job:  map 23% reduce 0%
18/03/24 22:59:51 INFO mapreduce.Job:  map 26% reduce 0%
18/03/24 22:59:54 INFO mapreduce.Job:  map 28% reduce 0%
```

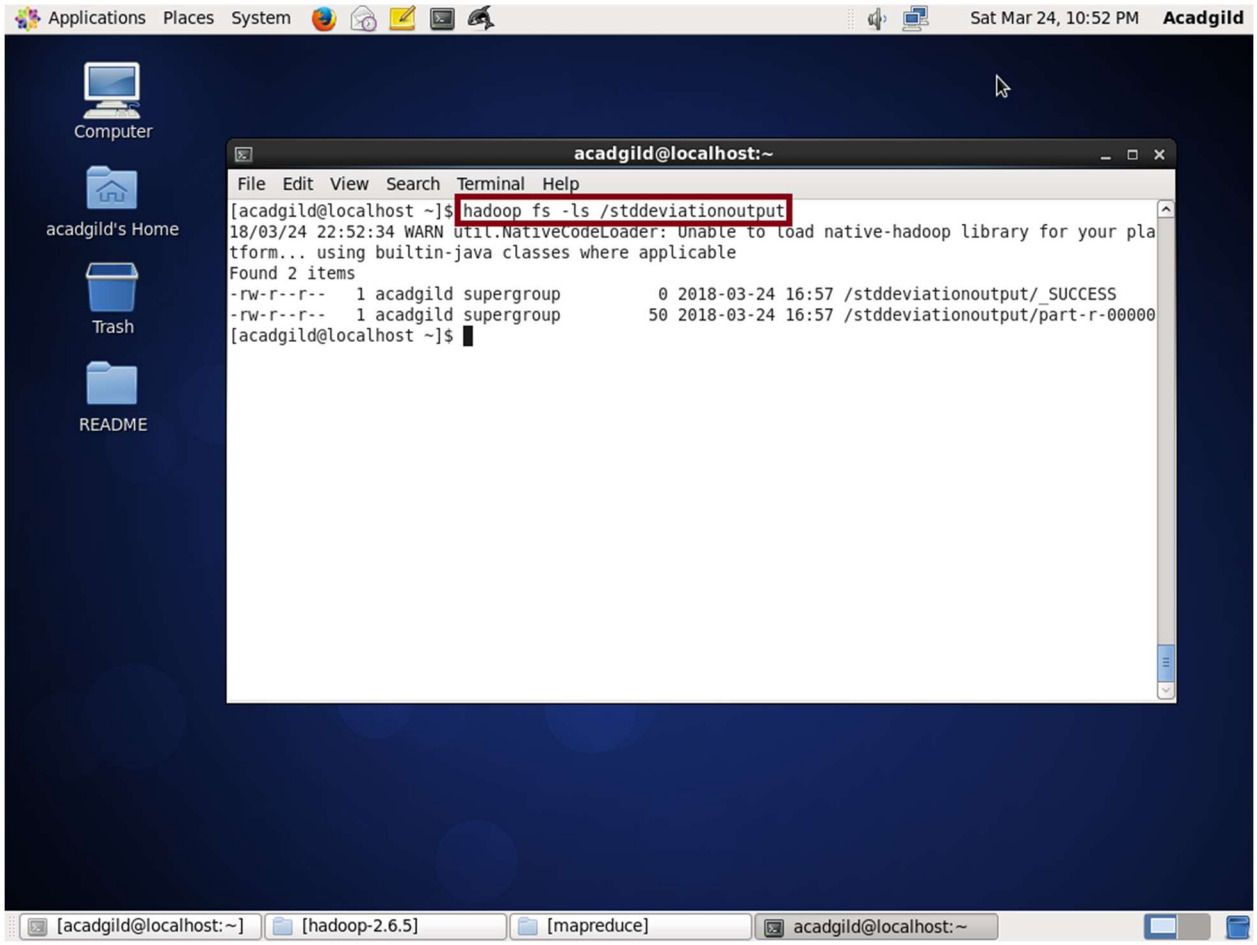
The output is displayed as Standard deviation as 2.2603926650031405



The output is saved in the stddeviationoutput directory.

It is listed by using following command.

hadoop fs –ls /stddeviationoutput



By using following command, the output data which is saved are displayed.

hadoop fs –cat /stddeviationoutput/*0