# Implementing MapReduce

Siddhant Dilip Godshalwar

February 25, 2023

## 1 Introduction

MapReduce is a programming model that is designed for processing and analyzing large datasets. It was originally introduced by Google in 2004 and has since become a widely used approach for large-scale data processing. MapReduce allows for efficient distributed processing of large datasets across a cluster of computers, enabling users to process data that would otherwise be too large to process on a single machine.

## 2 Implementation

The MapReduce model is a programming model that allows for distributed processing of large datasets across a cluster of computers. The MapReduce model consists of two primary phases: the Map phase and the Reduce phase.

### 2.1 Master Node

In a MapReduce framework, the MasterNode is the central coordinator that manages the entire processing of the data. It is responsible for dividing the data into smaller chunks and distributing these chunks to different worker nodes or mappers. The MasterNode also oversees the progress of the Map and Reduce tasks and monitors the health and status of the worker nodes.

Via TCP/IP sockets, typically, the MasterNode communicates with the worker nodes over a network. By keeping track of the health and condition of the worker nodes and reassigning work to other nodes in the event of failure, it also offers fault tolerance.

The MapReduce framework's MasterNode is a crucial part since it offers a centralized control method for handling massive volumes of data. The MasterNode may greatly increase the speed and effectiveness of data processing by dividing the task across several worker nodes and coordinating their efforts.

### 2.2 Data Partitioning

Data Partitioning is done by Master Node. The Master decides which mapper gets how many files and then assigns accordingly. The Data Partitioning in this implementation has three rules:-

- CASE 1 If the number of files to be Map Reduced is less than the number of mappers then each mapper gets 1 file and the remaining mappers get nothing.
- CASE 2 If the number of files is divisible by the number of mappers then each mapper has equal distribution of files
- CASE 3 If the number of files is greater than the number of mapper but is not divisible then all mappers get equal distribution except the last mapper which gets the remainder of files as well.

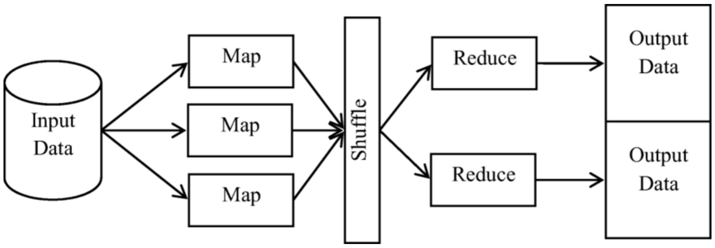


Figure 1: Architecture of MapReduce.

### 2.3 Mapper Phase

In the Map phase, the input data is processed and transformed into intermediate key-value pairs. The input data is typically a large dataset that is partitioned across the nodes in the cluster. Each node in the cluster processes a portion of the data independently using a user-defined mapping function. The mapping function takes in a single record of the input data and generates a set of intermediate key-value pairs.

The intermediate key-value pairs generated by the mapping function are then sorted, partitioned, and shuffled so that all the intermediate key-value pairs with the same key are sent to the same node for processing in the Reduce phase.

### 2.4 GroupBy Phase

In MapReduce, the GroupBy phase is the process of grouping together the intermediate key-value pairs produced by the mappers, based on their keys. The purpose of this phase is to ensure that all key-value pairs with the same key are processed together by the reducer, as the reducer receives input in the form of !key, list of values! pairs.

The GroupBy phase takes place after the Map phase and before the Reduce phase, and it involves sorting and shuffling the intermediate key-value pairs. Specifically, the intermediate key-value pairs generated by the mappers are first sorted by key. Then, pairs with the same key are grouped together and sent to the same reducer.

### 2.5 Reducer Phase

In the Reduce phase, the intermediate key-value pairs are combined and aggregated to produce the final output. The Reduce phase also consists of a user-defined function that is applied to each set of intermediate key-value pairs with the same key. This function takes in a key and a list of values and returns a set of output key-value pairs.

The output of the Reduce phase is typically written to a distributed file system, such as Hadoop Distributed File System (HDFS) or Amazon S3. The final output can then be used for further analysis or processing.

## 3 Workflow

### 3.1 Steps to run the files:

- 1 Insert all the files you want to MapReduce in the Input File directory in .txt format
- 2 Decide the number of Mappers you want and Reducers you want and change the values in the config.py file accordingly
- 3 Run the command MapReduce.py file

### 3.2 Output:

There are two types of output files generated:

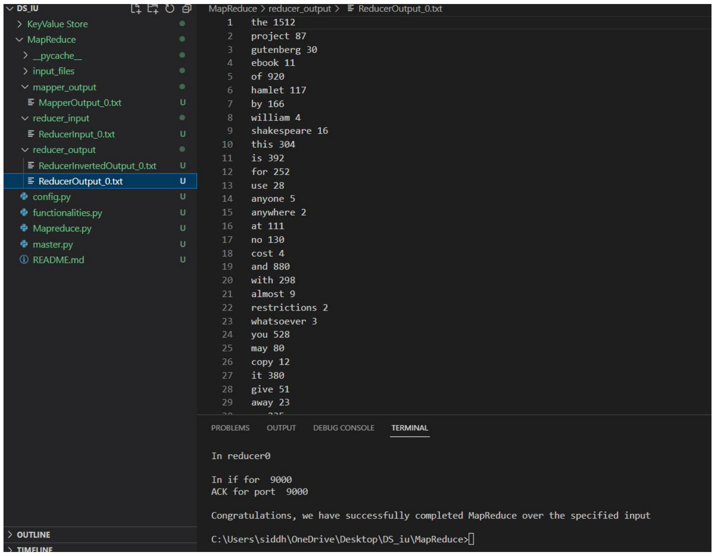
- 1 ReducerInvertedOutput:- this contains the output in an inverted index format for that particular ReducerID
- 2 ReducerOutput:- this contains the output in word-count format for that particular ReducerID

# 4 Testing

The testing files in this implementation have been added to the 'inputFiles' directory available in the code. Please feel free to add any number of files you are interested in adding.

## 4.1 TestCase1:- Base Case

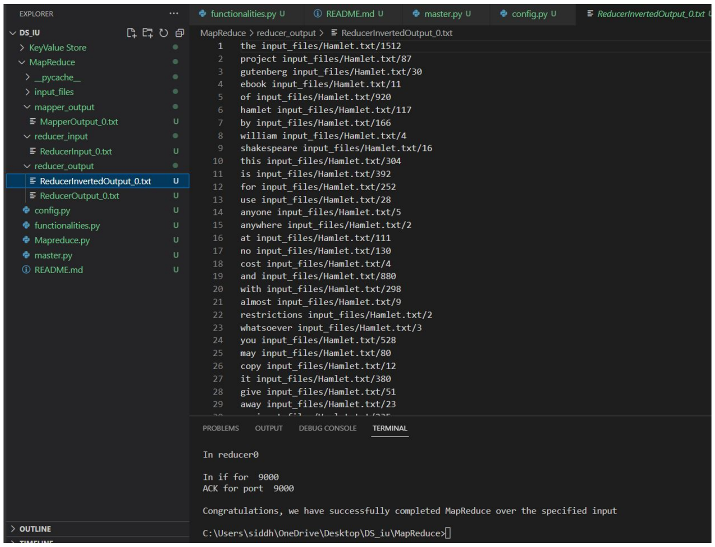
**File Count:-**1
**Mapper:-**1
**Reducer:-**1



```
the 1512
project 87
gutenberg 30
ebook 11
of 920
hamlet 117
by 166
william 4
shakespeare 16
this 304
is 392
for 252
use 28
anyone 5
anywhere 2
at 111
no 130
cost 4
and 880
with 298
almost 9
restrictions 2
whatsoever 3
you 528
may 80
copy 12
it 380
give 51
away 23
```

```
In reducer0

In if for  9000
ACK for port  9000

Congratulations, we have successfully completed MapReduce over the specified input

C:\Users\siddh\OneDrive\Desktop\DS_iu\MapReduce>
```

Figure 2: TestCase 1:- Word Count



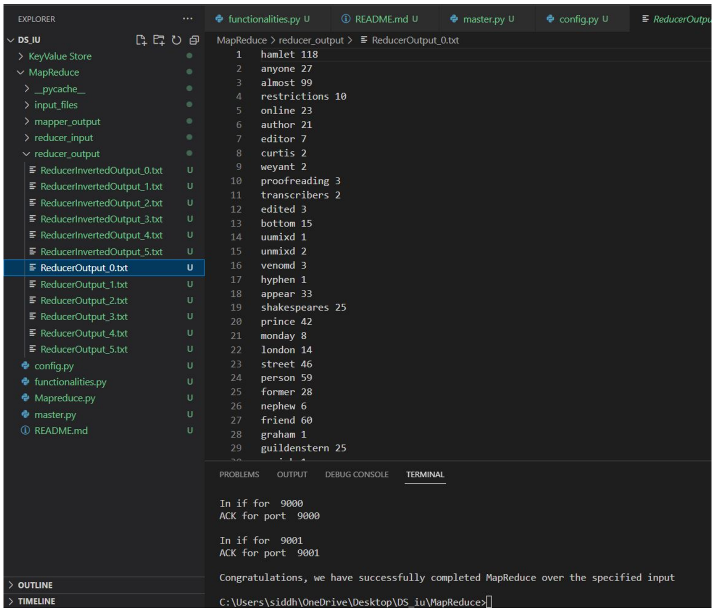
```
the input_files/Hamlet.txt/1512
project input_files/Hamlet.txt/87
gutenberg input_files/Hamlet.txt/30
ebook input_files/Hamlet.txt/11
of input_files/Hamlet.txt/920
hamlet input_files/Hamlet.txt/117
by input_files/Hamlet.txt/166
william input_files/Hamlet.txt/4
shakespeare input_files/Hamlet.txt/16
this input_files/Hamlet.txt/304
is input_files/Hamlet.txt/392
for input_files/Hamlet.txt/252
use input_files/Hamlet.txt/28
anyone input_files/Hamlet.txt/5
anywhere input_files/Hamlet.txt/2
at input_files/Hamlet.txt/111
no input_files/Hamlet.txt/130
cost input_files/Hamlet.txt/4
and input_files/Hamlet.txt/880
with input_files/Hamlet.txt/298
almost input_files/Hamlet.txt/9
restrictions input_files/Hamlet.txt/2
whatsoever input_files/Hamlet.txt/3
you input_files/Hamlet.txt/528
may input_files/Hamlet.txt/80
copy input_files/Hamlet.txt/12
it input_files/Hamlet.txt/380
give input_files/Hamlet.txt/51
away input_files/Hamlet.txt/23
```

```
In reducer0

In if for  9000
ACK for port  9000

Congratulations, we have successfully completed MapReduce over the specified input

C:\Users\siddh\OneDrive\Desktop\DS_iu\MapReduce>
```

Figure 3: TestCase 1:- Inverted Index

## 4.2 TestCase2:- No. of Files = No. of Mappers = No. of Reducers

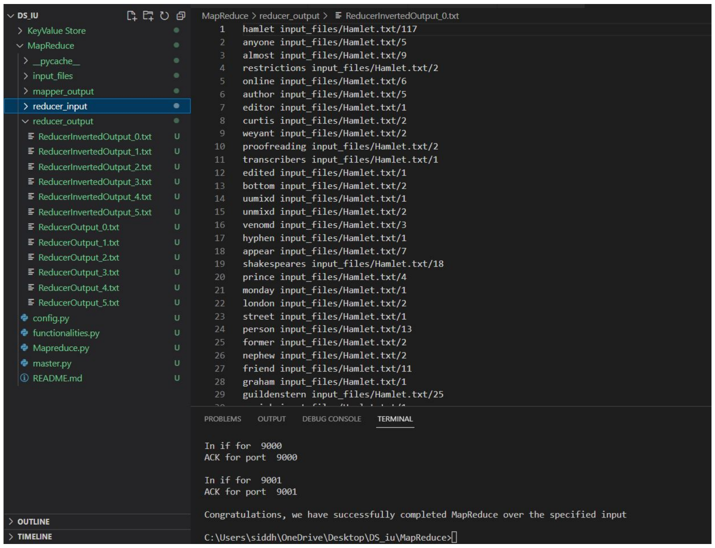
**File Count:-**6
**Mapper:-**6
**Reducer:**6



```
hamlet 118
anyone 27
almost 99
restrictions 10
online 23
author 21
editor 7
curtis 2
weyant 2
proofreading 3
transcribers 2
edited 3
bottom 15
uumixd 1
unmixd 2
venomd 3
hyphen 1
appear 33
shakespeares 25
prince 42
monday 8
london 14
street 46
person 59
former 28
nephew 6
friend 60
graham 1
guildenstern 25
```

```
In if for  9000
ACK for port  9000

In if for  9001
ACK for port  9001

Congratulations, we have successfully completed MapReduce over the specified input

C:\Users\siddh\OneDrive\Desktop\DS_iu\MapReduce>
```

Figure 4: TestCase 2: Word Count



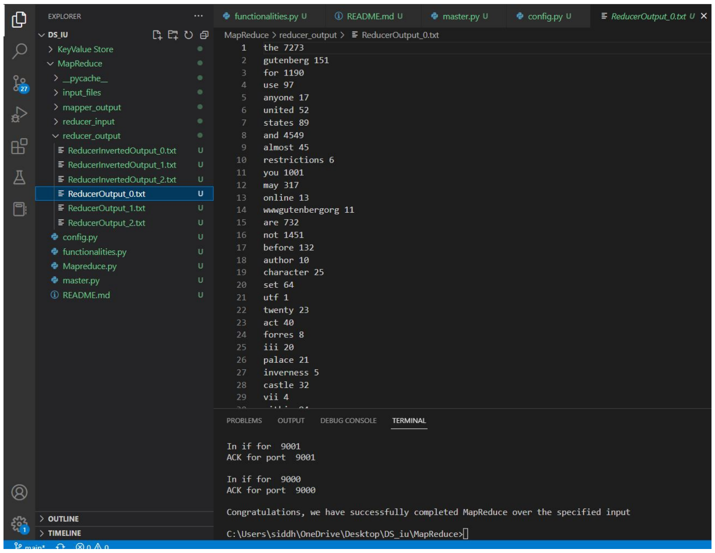
```
hamlet input_files/Hamlet.txt/117
anyone input_files/Hamlet.txt/5
almost input_files/Hamlet.txt/9
restrictions input_files/Hamlet.txt/2
online input_files/Hamlet.txt/6
author input_files/Hamlet.txt/5
editor input_files/Hamlet.txt/1
curtis input_files/Hamlet.txt/2
weyant input_files/Hamlet.txt/2
proofreading input_files/Hamlet.txt/2
transcribers input_files/Hamlet.txt/1
edited input_files/Hamlet.txt/1
bottom input_files/Hamlet.txt/2
uumixd input_files/Hamlet.txt/1
unmixd input_files/Hamlet.txt/2
venomd input_files/Hamlet.txt/3
hyphen input_files/Hamlet.txt/1
appear input_files/Hamlet.txt/7
shakespeares input_files/Hamlet.txt/18
prince input_files/Hamlet.txt/4
monday input_files/Hamlet.txt/1
london input_files/Hamlet.txt/2
street input_files/Hamlet.txt/1
person input_files/Hamlet.txt/13
former input_files/Hamlet.txt/2
nephew input_files/Hamlet.txt/2
friend input_files/Hamlet.txt/11
graham input_files/Hamlet.txt/1
guildenstern input_files/Hamlet.txt/25
```

```
In if for  9000
ACK for port  9000

In if for  9001
ACK for port  9001

Congratulations, we have successfully completed MapReduce over the specified input

C:\Users\siddh\OneDrive\Desktop\DS_iu\MapReduce>
```

Figure 5: TestCase 2:- Inverted Index

## 4.3 TestCase3:- No. of Files is greater than No. of Mappers and is divisible

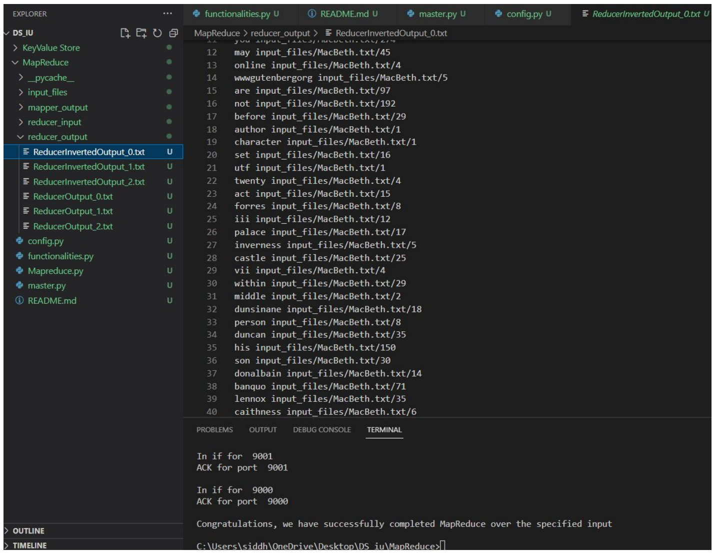
**File Count:-**6<br>
**Mapper:-**3<br>
**Reducer:**3



```
the 7273
gutenberg 151
for 1190
use 97
anyone 17
united 52
states 89
and 4549
almost 45
restrictions 6
you 1001
may 317
online 13
wwwgutenbergorg 11
are 732
not 1451
before 132
author 10
character 25
set 64
utf 1
twenty 23
act 40
forres 8
iii 20
palace 21
inverness 5
castle 32
vii 4
```

```
In if for  9001
ACK for port  9001

In if for  9000
ACK for port  9000

Congratulations, we have successfully completed MapReduce over the specified input

C:\Users\siddh\OneDrive\Desktop\DS_iu\MapReduce>
```

Figure 6: TestCase 3: Word Count



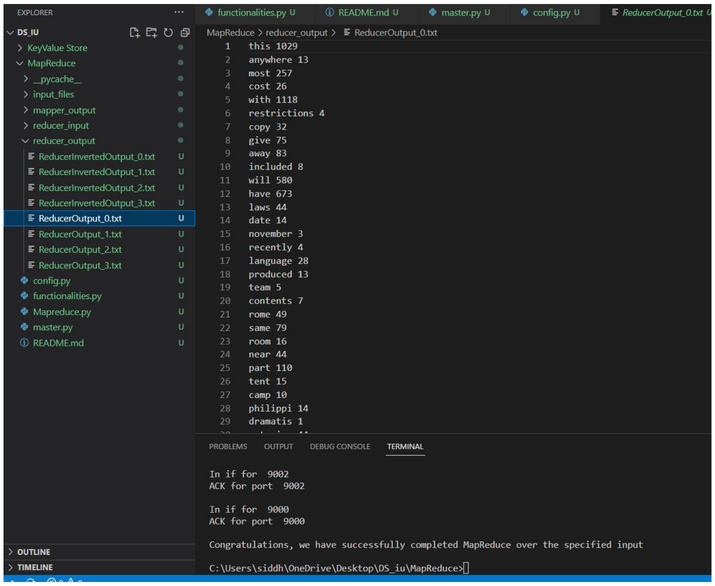
```
may input_files/MacBeth.txt/45
online input_files/MacBeth.txt/4
wwwgutenbergorg input_files/MacBeth.txt/5
are input_files/MacBeth.txt/97
not input_files/MacBeth.txt/192
before input_files/MacBeth.txt/29
author input_files/MacBeth.txt/1
character input_files/MacBeth.txt/1
set input_files/MacBeth.txt/16
utf input_files/MacBeth.txt/1
twenty input_files/MacBeth.txt/4
act input_files/MacBeth.txt/15
forres input_files/MacBeth.txt/8
iii input_files/MacBeth.txt/12
palace input_files/MacBeth.txt/17
inverness input_files/MacBeth.txt/5
castle input_files/MacBeth.txt/25
vii input_files/MacBeth.txt/4
within input_files/MacBeth.txt/29
middle input_files/MacBeth.txt/2
dunsinane input_files/MacBeth.txt/18
person input_files/MacBeth.txt/8
duncan input_files/MacBeth.txt/35
his input_files/MacBeth.txt/150
son input_files/MacBeth.txt/30
donalbain input_files/MacBeth.txt/14
banquo input_files/MacBeth.txt/71
lennox input_files/MacBeth.txt/35
caithness input_files/MacBeth.txt/6
```

```
In if for  9001
ACK for port  9001

In if for  9000
ACK for port  9000

Congratulations, we have successfully completed MapReduce over the specified input

C:\Users\siddh\OneDrive\Desktop\DS_iu\MapReduce>
```

Figure 7: TestCase 3:- Inverted Index

## 4.4 TestCase4:- No. of Files is greater than No. of Mappers and is not divisible

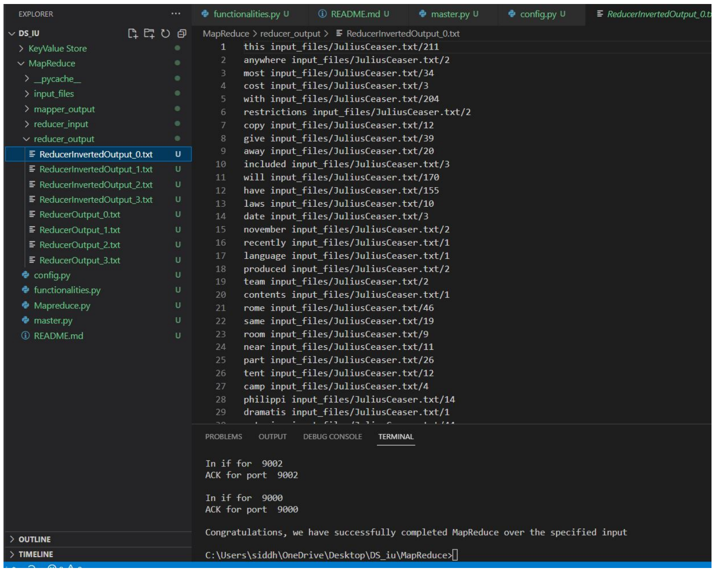
**File Count:-**7<br>
**Mapper:-**3<br>
**Reducer:**4



```
this 1029
anywhere 13
most 257
cost 26
with 1118
restrictions 4
copy 32
give 75
away 83
included 8
will 580
have 673
laws 44
date 14
november 3
recently 4
language 28
produced 13
team 5
contents 7
rome 49
same 79
room 16
near 44
part 110
tent 15
camp 10
philippi 14
dramatis 1
```

```
In if for  9002
ACK for port  9002

In if for  9000
ACK for port  9000

Congratulations, we have successfully completed MapReduce over the specified input

C:\Users\siddh\OneDrive\Desktop\DS_iu\MapReduce>
```

Figure 8: TestCase 4: Word Count



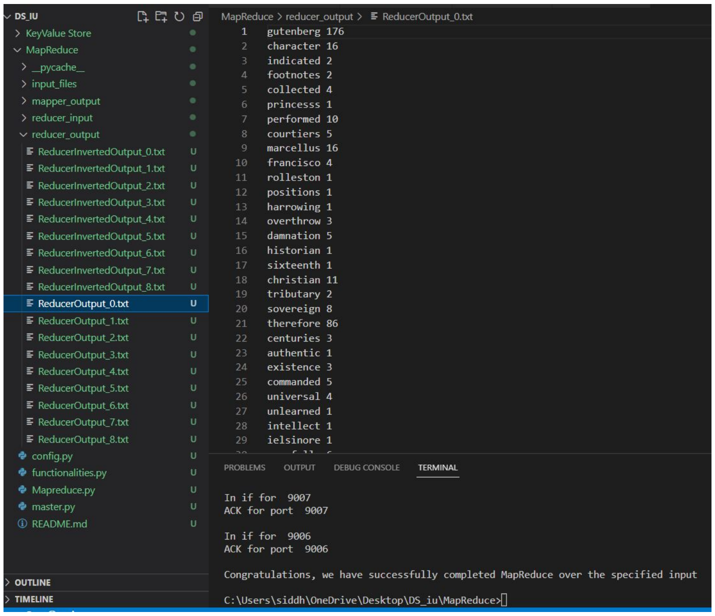
```
this input_files/JuliusCeaser.txt/211
anywhere input_files/JuliusCeaser.txt/2
most input_files/JuliusCeaser.txt/34
cost input_files/JuliusCeaser.txt/3
with input_files/JuliusCeaser.txt/204
restrictions input_files/JuliusCeaser.txt/2
copy input_files/JuliusCeaser.txt/12
give input_files/JuliusCeaser.txt/39
away input_files/JuliusCeaser.txt/20
included input_files/JuliusCeaser.txt/3
will input_files/JuliusCeaser.txt/170
have input_files/JuliusCeaser.txt/155
laws input_files/JuliusCeaser.txt/10
date input_files/JuliusCeaser.txt/3
november input_files/JuliusCeaser.txt/2
recently input_files/JuliusCeaser.txt/1
language input_files/JuliusCeaser.txt/1
produced input_files/JuliusCeaser.txt/2
team input_files/JuliusCeaser.txt/2
contents input_files/JuliusCeaser.txt/1
rome input_files/JuliusCeaser.txt/46
same input_files/JuliusCeaser.txt/19
room input_files/JuliusCeaser.txt/9
near input_files/JuliusCeaser.txt/11
part input_files/JuliusCeaser.txt/26
tent input_files/JuliusCeaser.txt/12
camp input_files/JuliusCeaser.txt/4
philippi input_files/JuliusCeaser.txt/14
dramatis input_files/JuliusCeaser.txt/1
```

```
In if for  9002
ACK for port  9002

In if for  9000
ACK for port  9000

Congratulations, we have successfully completed MapReduce over the specified input

C:\Users\siddh\OneDrive\Desktop\DS_iu\MapReduce>
```

Figure 9: TestCase 4:- Inverted Index

## 4.5 TestCase5:- No. of Files is less than No. of Mappers and No. of Reducers

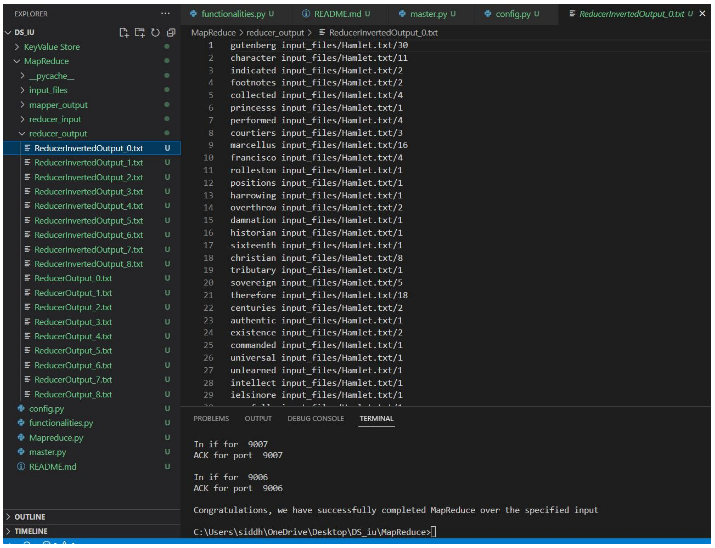
**File Count:-**6<br>
**Mapper:-**9<br>
**Reducer:**9



```
gutenberg 176
character 16
indicated 2
footnotes 2
collected 4
princesss 1
performed 10
courtiers 5
marcellus 16
francisco 4
rolleston 1
positions 1
harrowing 1
overthrow 3
damnation 5
historian 1
sixteenth 1
christian 11
tributary 2
sovereign 8
therefore 86
centuries 3
authentic 1
existence 3
commanded 5
universal 4
unlearned 1
intellect 1
ielsinore 1
```

```
In if for  9007
ACK for port  9007

In if for  9006
ACK for port  9006

Congratulations, we have successfully completed MapReduce over the specified input

C:\Users\siddh\OneDrive\Desktop\DS_iu\MapReduce>
```

Figure 10: TestCase 5: Word Count



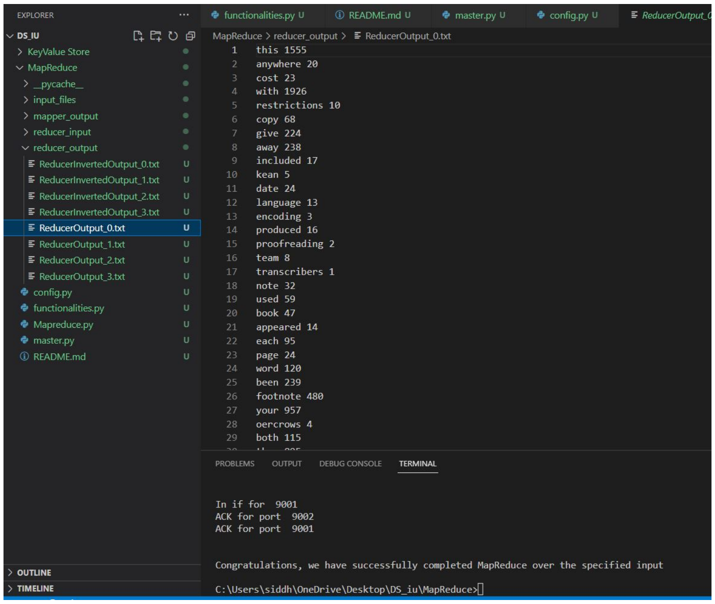
```
gutenberg input_files/Hamlet.txt/30
character input_files/Hamlet.txt/11
indicated input_files/Hamlet.txt/2
footnotes input_files/Hamlet.txt/2
collected input_files/Hamlet.txt/4
princesss input_files/Hamlet.txt/1
performed input_files/Hamlet.txt/4
courtiers input_files/Hamlet.txt/3
marcellus input_files/Hamlet.txt/16
francisco input_files/Hamlet.txt/4
rolleston input_files/Hamlet.txt/1
positions input_files/Hamlet.txt/1
harrowing input_files/Hamlet.txt/1
overthrow input_files/Hamlet.txt/2
damnation input_files/Hamlet.txt/1
historian input_files/Hamlet.txt/1
sixteenth input_files/Hamlet.txt/1
christian input_files/Hamlet.txt/8
tributary input_files/Hamlet.txt/1
sovereign input_files/Hamlet.txt/5
therefore input_files/Hamlet.txt/18
centuries input_files/Hamlet.txt/2
authentic input_files/Hamlet.txt/1
existence input_files/Hamlet.txt/2
commanded input_files/Hamlet.txt/1
universal input_files/Hamlet.txt/1
unlearned input_files/Hamlet.txt/1
intellect input_files/Hamlet.txt/1
ielsinore input_files/Hamlet.txt/1
```

```
In if for  9007
ACK for port  9007

In if for  9006
ACK for port  9006

Congratulations, we have successfully completed MapReduce over the specified input

C:\Users\siddh\OneDrive\Desktop\DS_iu\MapReduce>
```

Figure 11: TestCase 5:- Inverted Index

## 4.6 TestCase6:- No. of Reducers is less than No. of Files is less than No. of Mappers

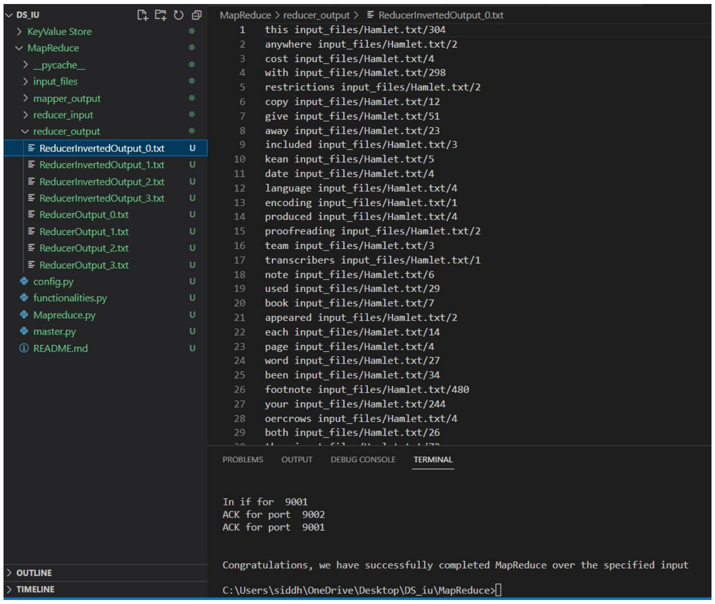
**File Count:-**6<br>
**Mapper:-**9<br>
**Reducer:**4



```
this 1555
anywhere 20
cost 23
with 1926
restrictions 10
copy 68
give 224
away 238
included 17
kean 5
date 24
language 13
encoding 3
produced 16
proofreading 2
team 8
transcribers 1
note 32
used 59
book 47
appeared 14
each 95
page 24
word 120
been 239
footnote 480
your 957
oercrows 4
both 115
```

```
In if for  9001
ACK for port  9002
ACK for port  9001

Congratulations, we have successfully completed MapReduce over the specified input

C:\Users\siddh\OneDrive\Desktop\DS_iu\MapReduce>
```

Figure 12: TestCase 6: Word Count



```
this input_files/Hamlet.txt/304
anywhere input_files/Hamlet.txt/2
cost input_files/Hamlet.txt/4
with input_files/Hamlet.txt/298
restrictions input_files/Hamlet.txt/2
copy input_files/Hamlet.txt/12
give input_files/Hamlet.txt/51
away input_files/Hamlet.txt/23
included input_files/Hamlet.txt/3
kean input_files/Hamlet.txt/5
date input_files/Hamlet.txt/4
language input_files/Hamlet.txt/4
encoding input_files/Hamlet.txt/1
produced input_files/Hamlet.txt/4
proofreading input_files/Hamlet.txt/2
team input_files/Hamlet.txt/3
transcribers input_files/Hamlet.txt/1
note input_files/Hamlet.txt/6
used input_files/Hamlet.txt/29
book input_files/Hamlet.txt/7
appeared input_files/Hamlet.txt/2
each input_files/Hamlet.txt/14
page input_files/Hamlet.txt/4
word input_files/Hamlet.txt/27
been input_files/Hamlet.txt/34
footnote input_files/Hamlet.txt/480
your input_files/Hamlet.txt/244
oercrows input_files/Hamlet.txt/4
both input_files/Hamlet.txt/26
```

```
In if for  9001
ACK for port  9002
ACK for port  9001

Congratulations, we have successfully completed MapReduce over the specified input

C:\Users\siddh\OneDrive\Desktop\DS_iu\MapReduce>
```

Figure 13: TestCase 6:- Inverted Index

# 5 Limitations and Assumptions

## 5.1 Limitation

The Limitations of this implementation of MapReduce is:-

- 1 This doesn't account for the Fault Tolerance of any one of the mappers/reducers. So if a Mapper/Reducer fails, there will be some loss of data

- 2 This MapReducer also can only consume files that are in '.txt' format and are using 'UTF-8' codec. Other files are not parseable

## 5.2 Assumptions

The Assumptions of this implementation of MapReduce is:-

- 1 The communication between the Master and the Mappers and Reducers is secure.

- 2 The Masters and Reducers will work without failure.

- 3 The input files are all plain text files and use 'UTF-8' codecs.