Chameleon Cloud Tutorial

Setting up Hadoop Sandbox on Chameleon Bare Metal Servers

Hadoop - Implementation of Map Reduce (Python)

Objective

In this tutorial, we will discuss about the Map and Reduce program, its implementation and advantages over the other processes.

Prerequisites

The following prerequisites are expected for successful completion of this tutorial:

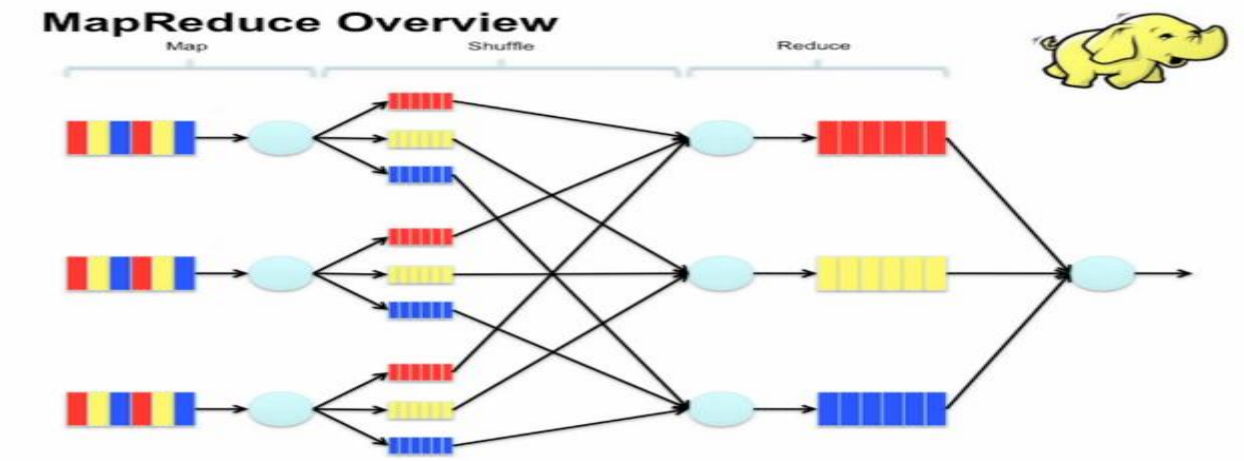
- Chameleon Cloud account (http://chameleoncloud.org/user/register/)
- SSH client (Windows users: download PuTTY from here: http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html)
- A basic knowledge of Linux.
- Installation of Hadoop and Map reduce.
- Single node set up.

MapReduce

- MapReduce is a software framework used to process vast amount of data.
- MapReduce usually splits the input data set into chunks, map task process the data in parallel manner.
- The output of the Map is fed as Input to the Reducer.
- Both input and output files are stored in system.
- MapReduce programs works exclusively on key value pair.

(input) <k1, v1> → **map** → <k2, v2> → **combine** → <k2, v2> → **reduce** → <k3, v3> (output)

MapReduce flow diagram

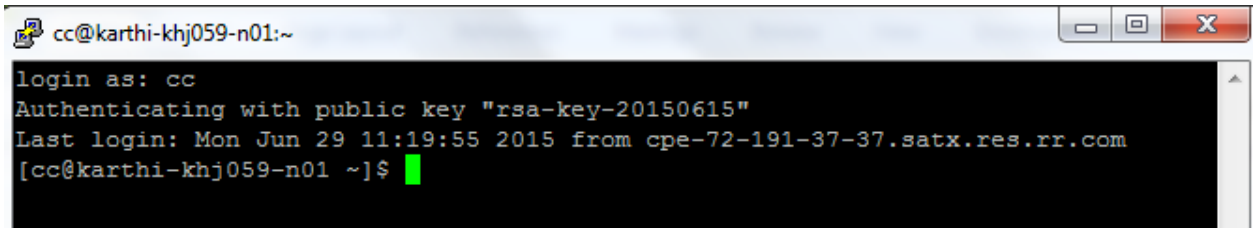


Implementation of MapReduce

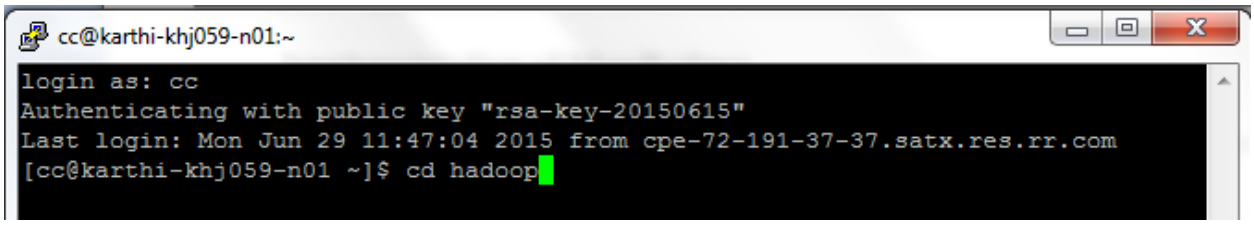
- Login to the system with user id ‘CC’ and enter to establish the connection.

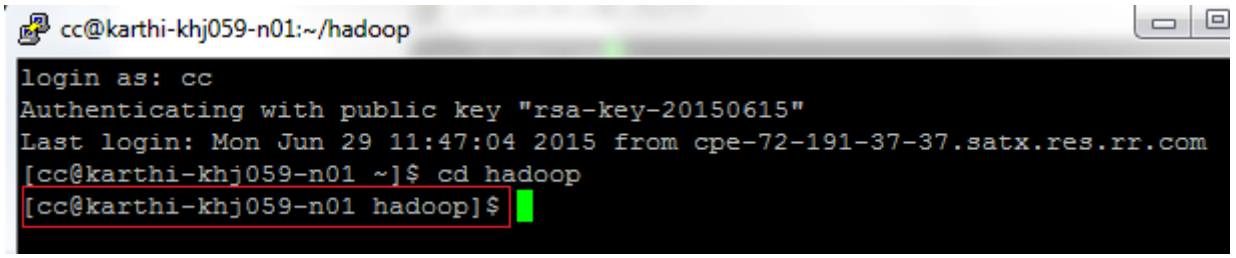
  Login as ‘CC’

  Enter to establish connection(Authenticate with the public key)

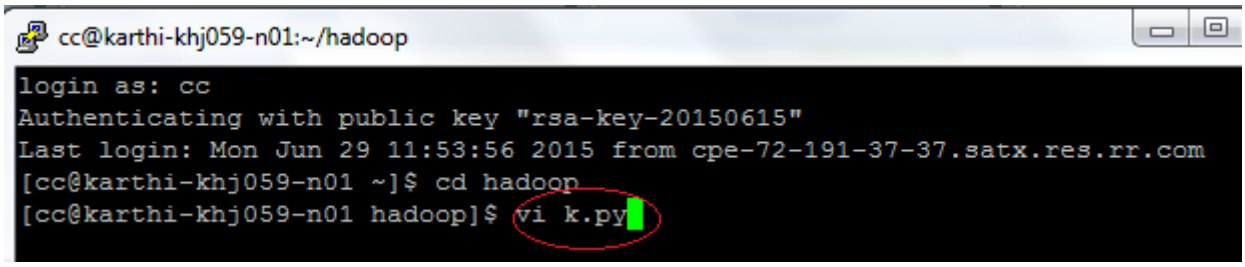


- Change the directory to Hadoop

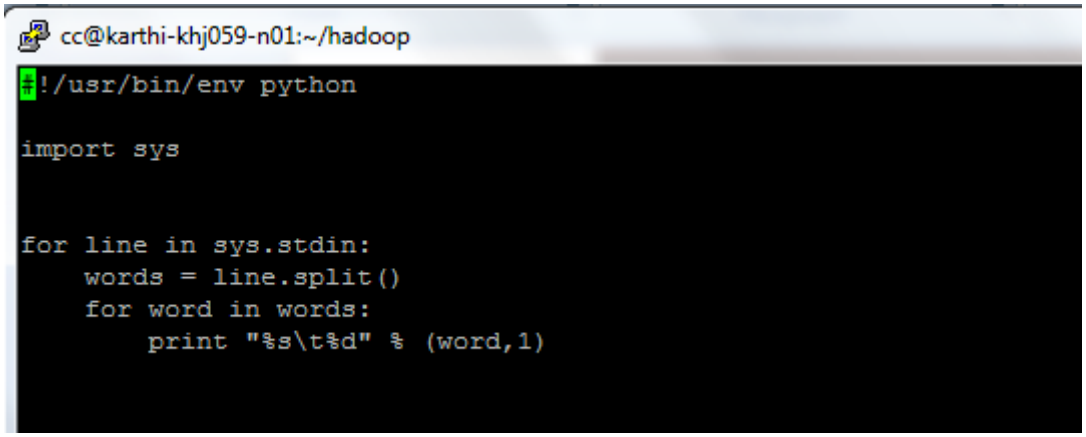




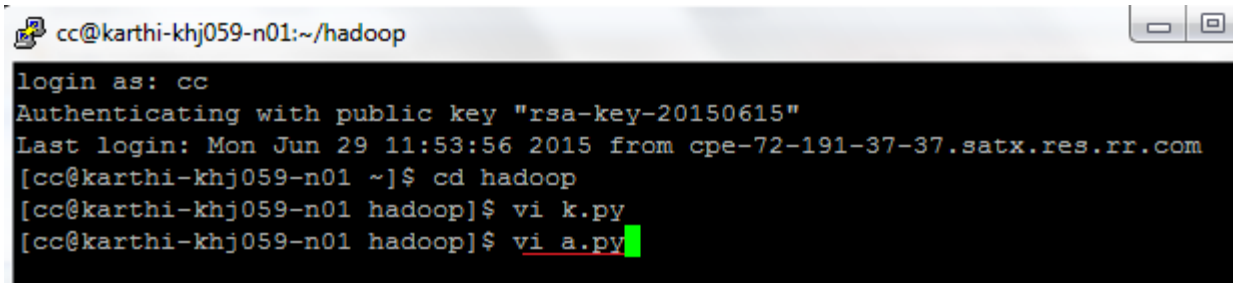
- Create the file ‘K’(Mapper code) with py extension.



Input the mapper code in K.py file



- Create the file ‘A’(Reducer code) with py extension.



- Input the Reducer code in A.py file

cc@karthi-khj059-n01:~/hadoop

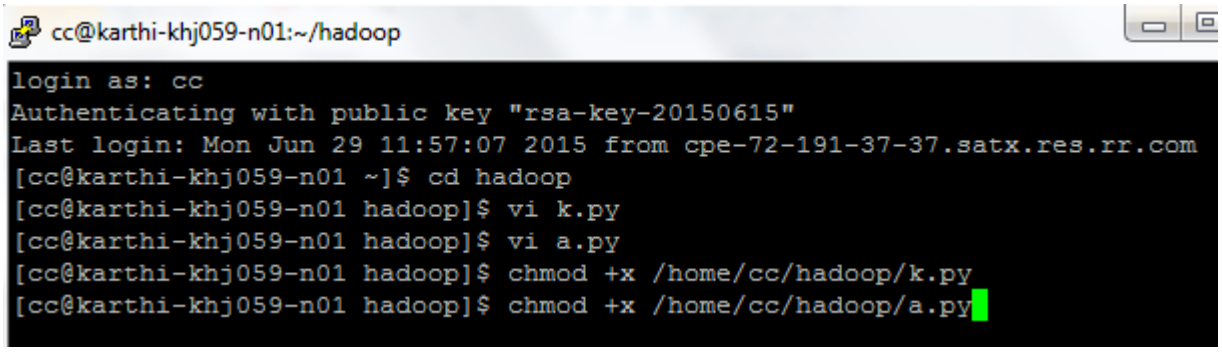
```
#!/usr/bin/python

import sys


def output(previous_key,total):
    if previous_key is not None:
        print "%s was found %d times" % (previous_key, total)

previous_key = None
total = 0

for line in sys.stdin:
    key, value = line.split("\t", 1)
    if key != previous_key:
        output(previous_key, total)
        previous_key = key
        total = 0
    total += int(value)

output(previous_key, total)
~
```

- Make sure the file has the execution permission

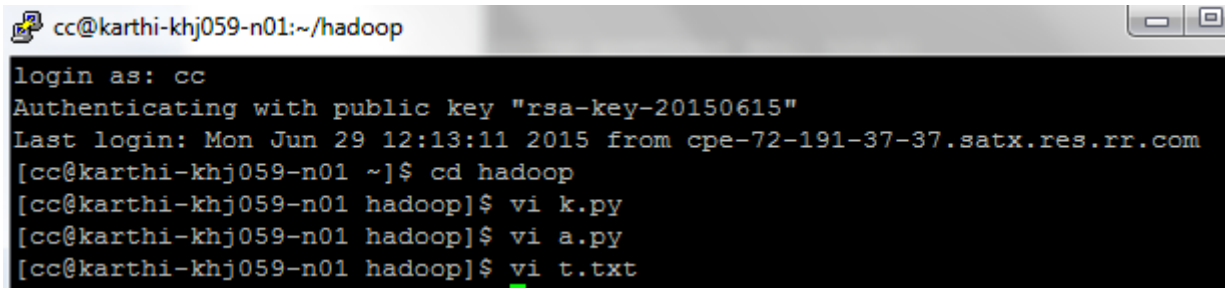
  Chmod +x /path of the file

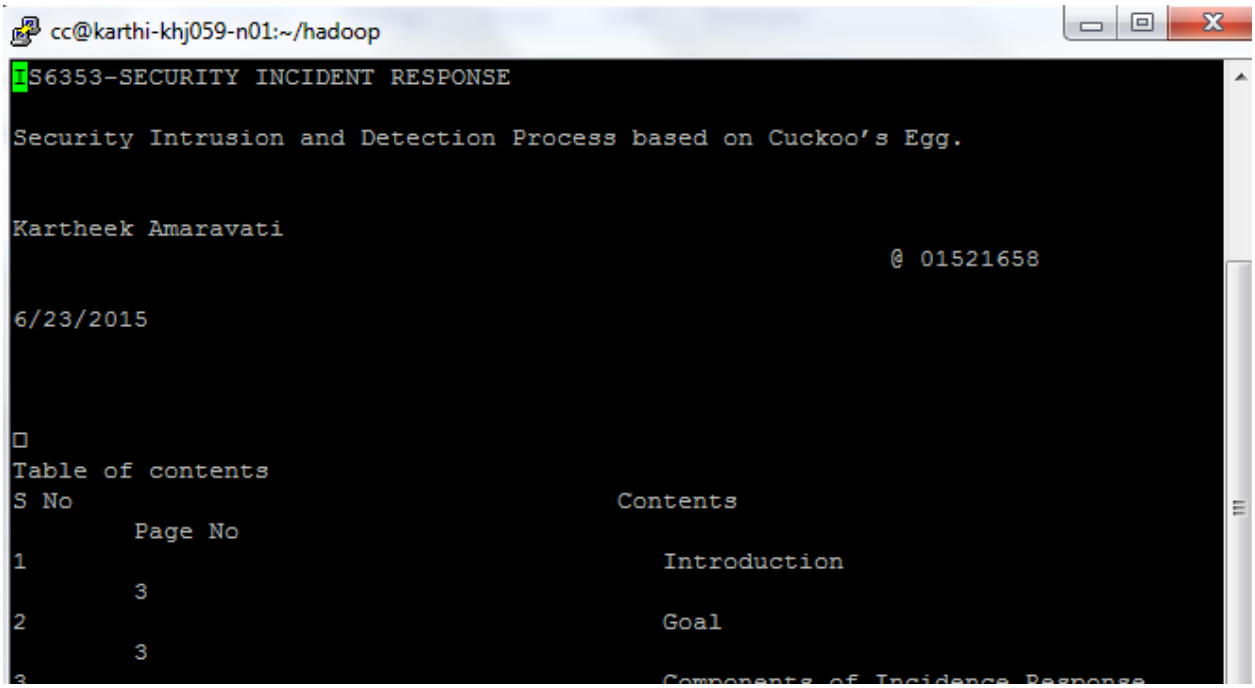
cc@karthi-khj059-n01:~/hadoop

```
login as: cc
Authenticating with public key "rsa-key-20150615"
Last login: Mon Jun 29 11:57:07 2015 from cpe-72-191-37-37.satx.res.rr.com
[cc@karthi-khj059-n01 ~]$ cd hadoop
[cc@karthi-khj059-n01 hadoop]$ vi k.py
[cc@karthi-khj059-n01 hadoop]$ vi a.py
[cc@karthi-khj059-n01 hadoop]$ chmod +x /home/cc/hadoop/k.py
```

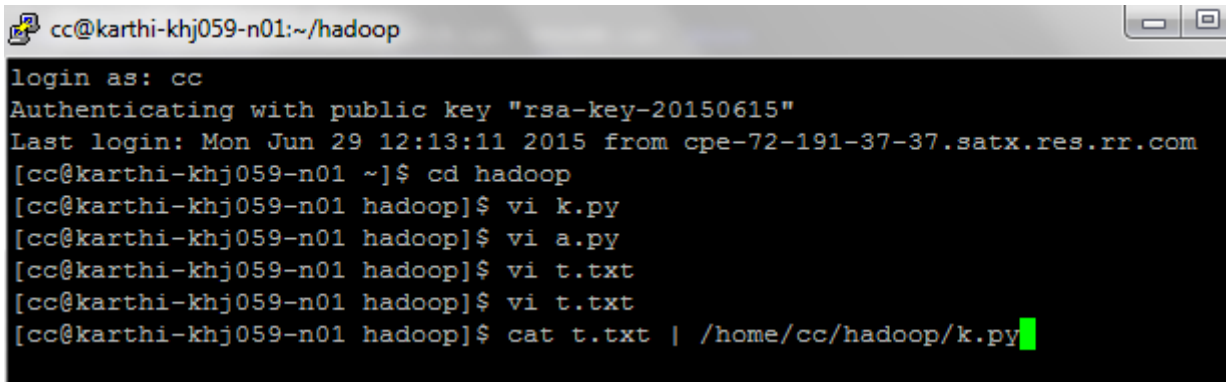


- Copy the text data to an input file(t.txt)

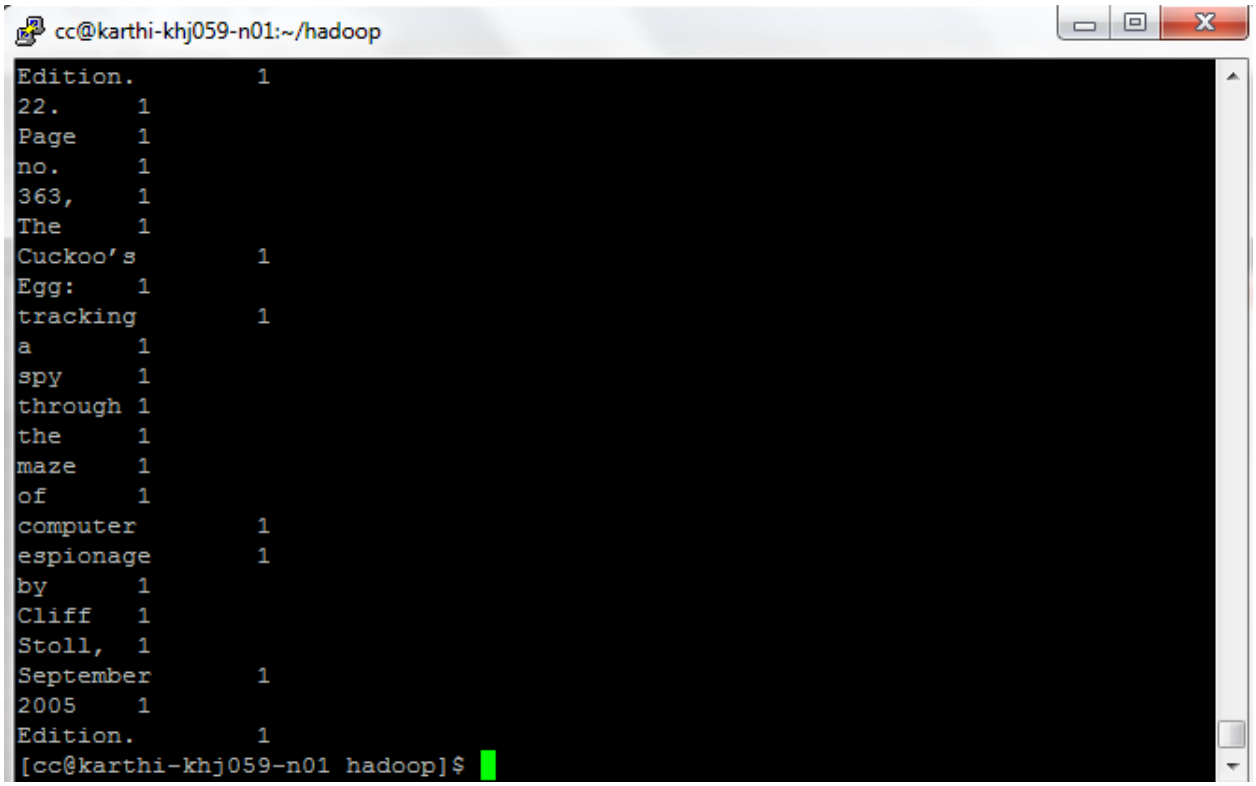




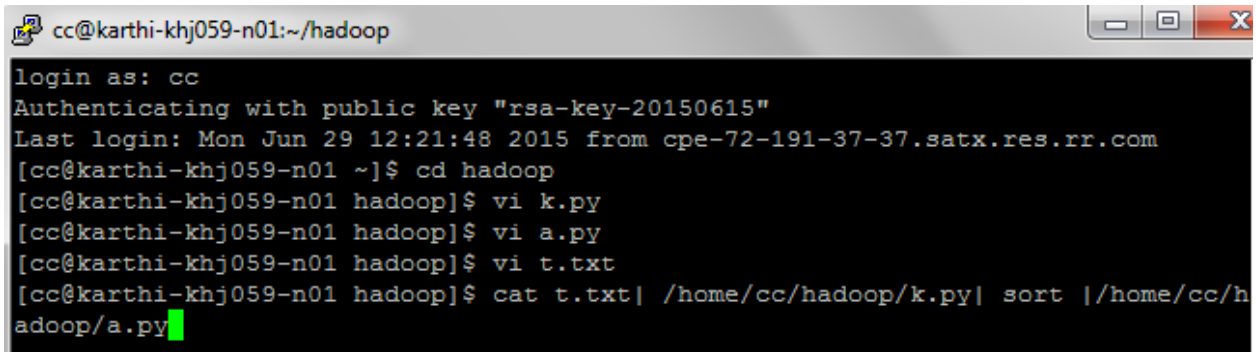
- Run the Map program for the input text file(where the whole data is divided into chunks) and key value pair is formed.



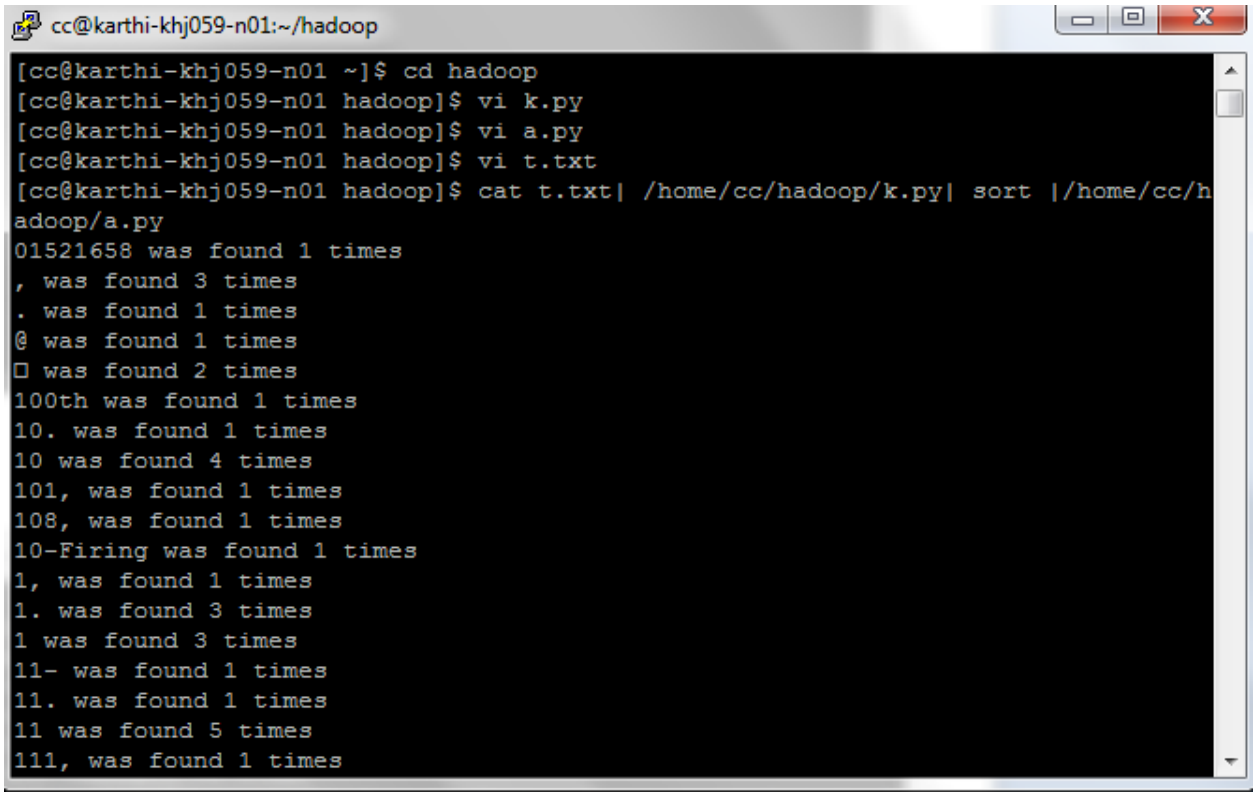
- Output of an Map program



- Execute Map and Reduce program together.



- Output of an Map and Reduce program



Advantages

1. Distribute data and computation, which reduces network overload.
2. Tasks are independent due to this it can easily handle partial failures.
3. Linear scaling in the ideal case.
4. Simple programming model.
5. Flat scalability.