INDEX

INTRODUCTION

Data Generator

We are importing a Faker to generate the random data. So Faker Can be used for things such as filling databases with random information during development of database related applications.
In our code it generates Name, address, city and Salary of people randomly. Our code generates file of 100MB. So to generate 10 GB file, we execute it 100 times and save it in the database. Our code can also execute for 1GB file and more depending on what size (in bytes) we provide it to the data generator script.

Map Logic

Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs.)
So we have considered the Name and City data and we are mapping here.

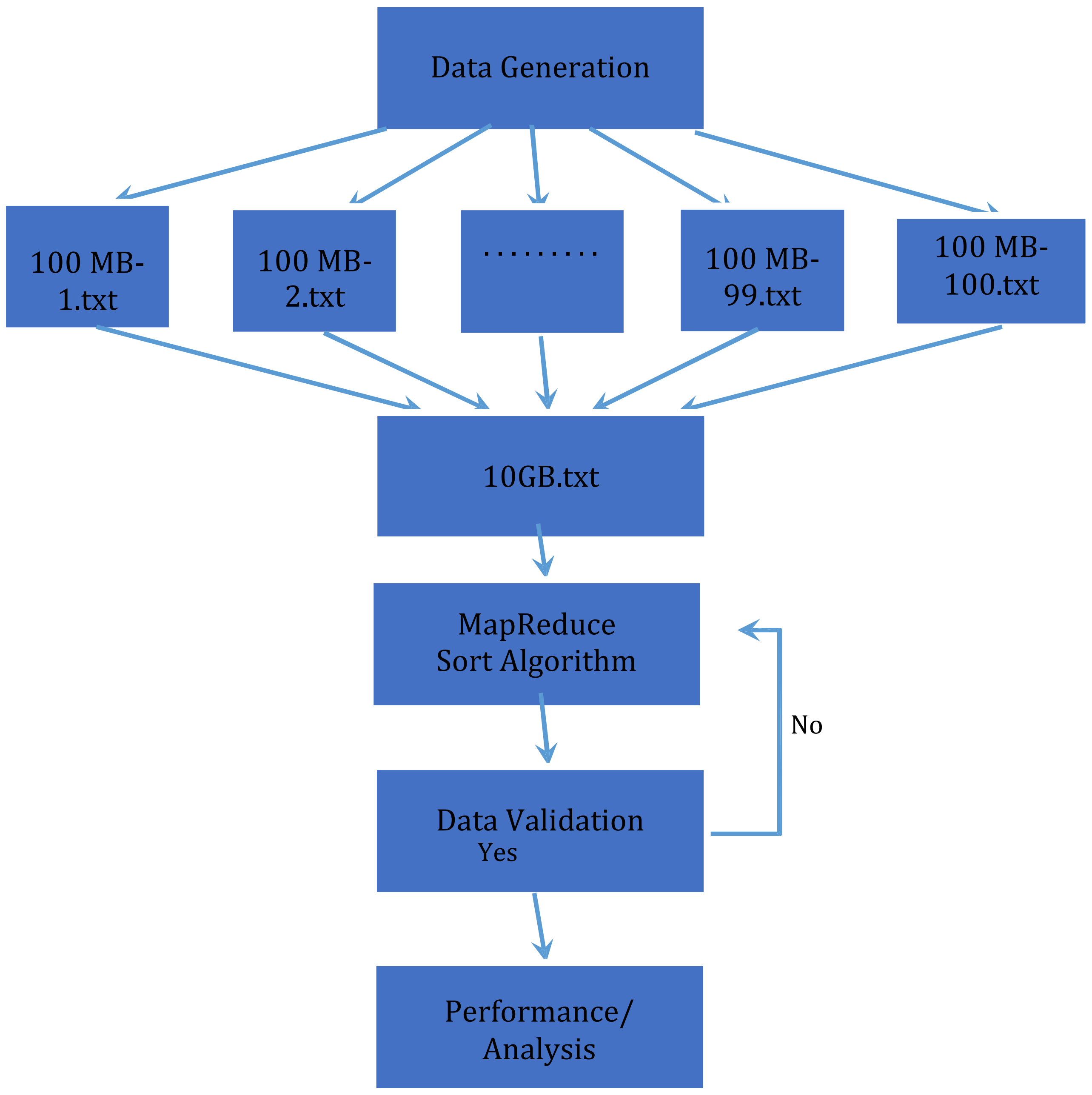
Reduce Logic

The reduce logic we are using here is MapReduce. Reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. In the code we are using, Salary and City field is used as Reduce Key. So we are pulling the data as per the salary bracket given in our code.

HUE

HUE is a Web interface for analyzing data with apache Hadoop. It's a Free open source. It loads data into Hadoop. Then Views it, Process it and prepare it. Then it analyze, search and visualize it.

WORKFLOW


Data Generation
100 MB-1.txt
100 MB-2.txt
....
100 MB-99.txt
100 MB-100.txt
10GB.txt
MapReduce Sort Algorithm
Data Validation
Yes
No
Performance/ Analysis

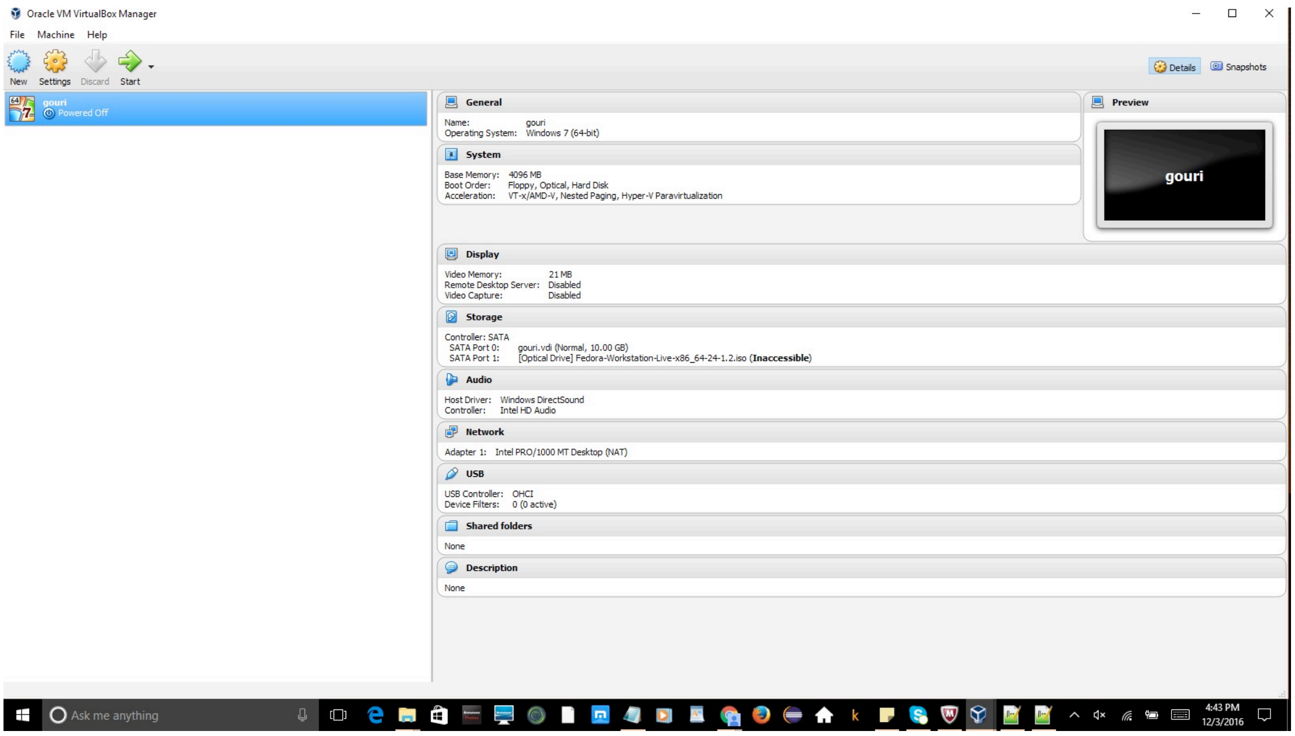
SYSTEM CONFIGURATION

Approach 1: Windows

Process to install Hadoop

1. Install virtual box
2. Install fedora
3. Using terminal connect to root
4. Install java
5. Download .rpm file from Cloudera
6. Install Hadoop
7. Format the name node
8. Start the services

After finish install Hadoop, press start to start the services



General

```
Name:              gouri
Operating System:  Windows 7 (64-bit)
```

System

```
Base Memory:   4096 MB
Boot Order:    Floppy, Optical, Hard Disk
Acceleration:  VT-x/AMD-V, Nested Paging, Hyper-V Paravirtualization
```

Display

```
Video Memory:            21 MB
Remote Desktop Server:   Disabled
Video Capture:           Disabled
```

Storage

```
Controller: SATA
 SATA Port 0:     gouri.vdi (Normal, 10.00 GB)
 SATA Port 1:     [Optical Drive] Fedora-Workstation-Live-x86_64-24-1.2.iso (Inaccessible)
```

Audio

```
Host Driver:   Windows DirectSound
Controller:    Intel HD Audio
```

Network

```
Adapter 1:   Intel PRO/1000 MT Desktop (NAT)
```

USB

```
USB Controller:   OHCI
Device Filters:   0 (0 active)
```

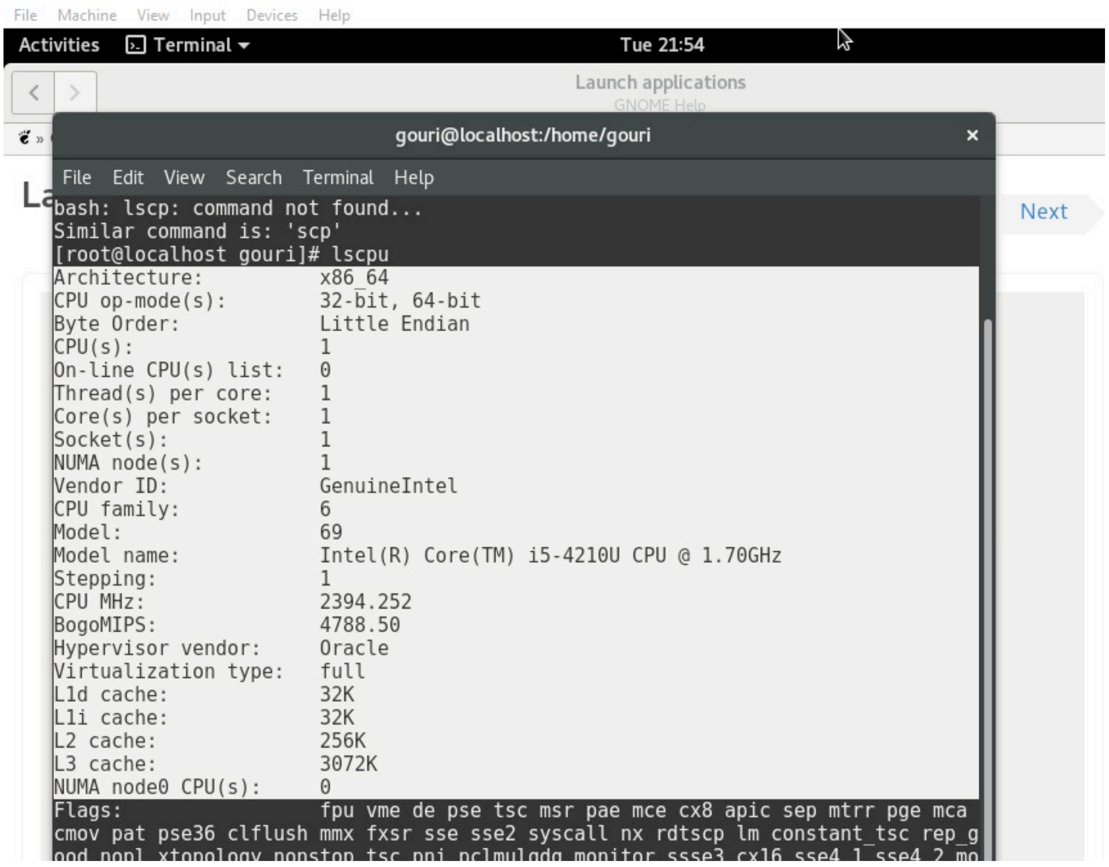
Shared folders

None

Description

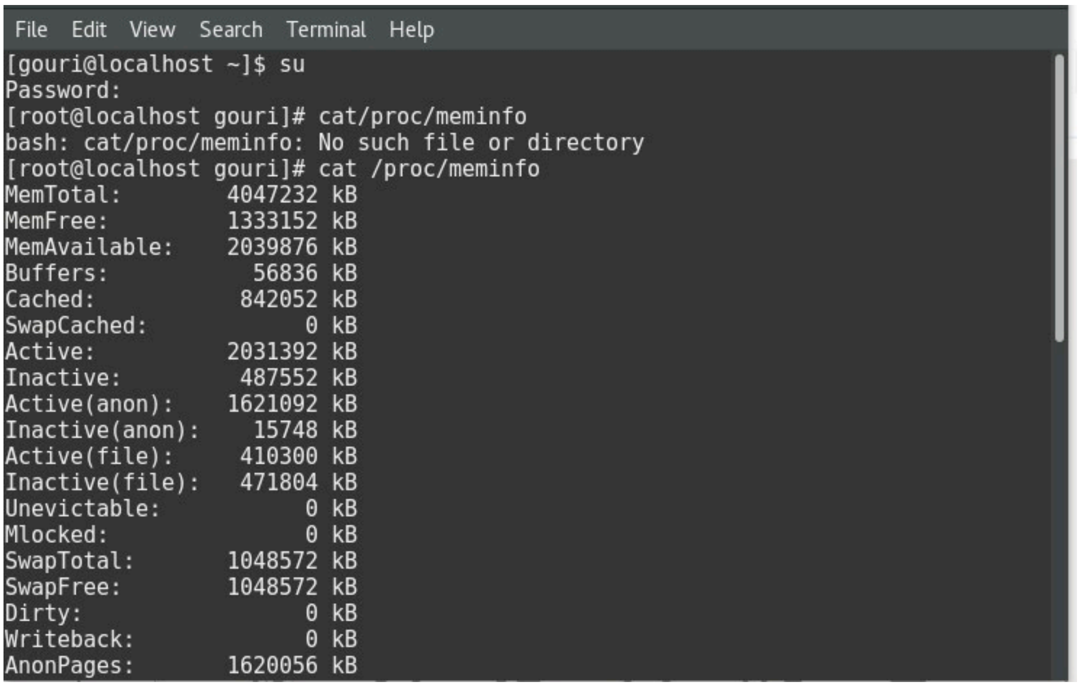
None

Using command lscpu to verify system configuration such as # of CPUs, cores, and so on.


File Machine View Input Devices Help
Activities
Terminal
Tue 21:54
Launch applications
GNOME Help
Next
gouri@localhost:/home/gouri
File Edit View Search Terminal Help

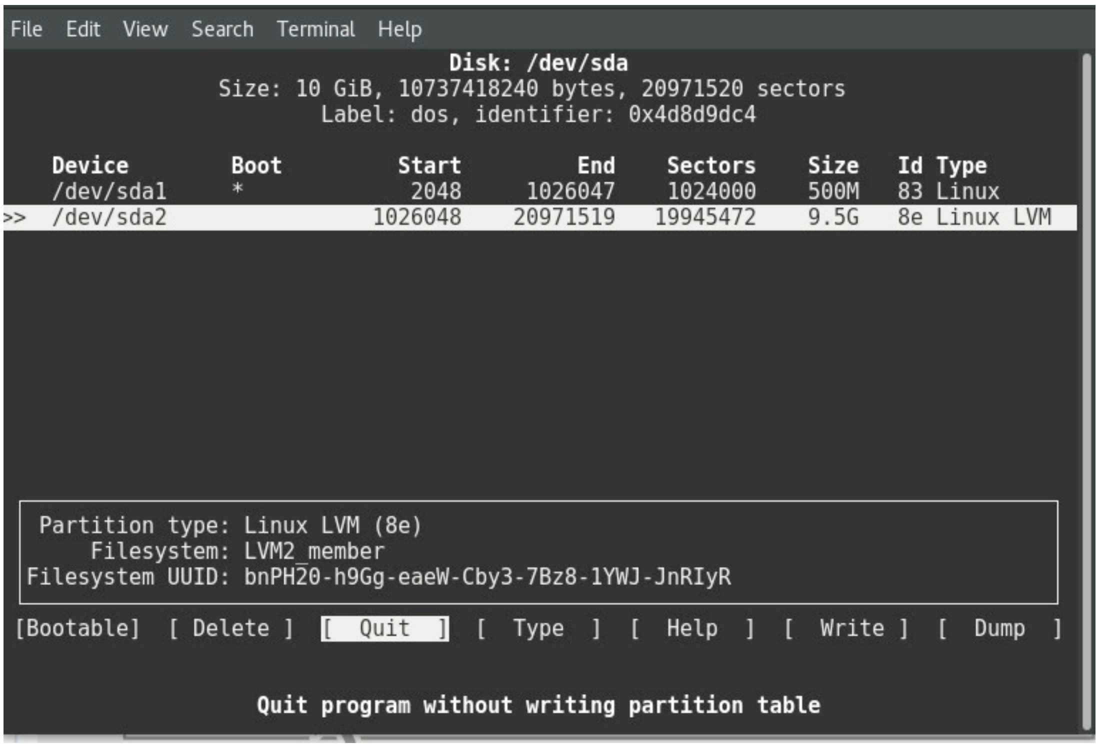
```
bash: lscp: command not found...
Similar command is: 'scp'
[root@localhost gouri]# lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                1
On-line CPU(s) list:   0
Thread(s) per core:    1
Core(s) per socket:    1
Socket(s):             1
NUMA node(s):          1
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 69
Model name:            Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz
Stepping:              1
CPU MHz:               2394.252
BogoMIPS:              4788.50
Hypervisor vendor:     Oracle
Virtualization type:   full
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              3072K
NUMA node0 CPU(s):     0
Flags:                 fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca
cmov pat pse36 clflush mmx fxsr sse sse2 syscall nx rdtscp lm constant_tsc rep_g
ood nopl xtopology nonstop_tsc pni pclmulqdq monitor ssse3 cx16 sse4_1 sse4_2 mo
```

Using command cat /pro/meminfo to verify memory size, and etc.



```
[gouri@localhost ~]$ su
Password:
[root@localhost gouri]# cat/proc/meminfo
bash: cat/proc/meminfo: No such file or directory
[root@localhost gouri]# cat /proc/meminfo
MemTotal:        4047232 kB
MemFree:         1333152 kB
MemAvailable:    2039876 kB
Buffers:           56836 kB
Cached:           842052 kB
SwapCached:            0 kB
Active:          2031392 kB
Inactive:         487552 kB
Active(anon):    1621092 kB
Inactive(anon):    15748 kB
Active(file):     410300 kB
Inactive(file):   471804 kB
Unevictable:           0 kB
Mlocked:               0 kB
SwapTotal:       1048572 kB
SwapFree:        1048572 kB
Dirty:                 0 kB
Writeback:             0 kB
AnonPages:       1620056 kB
```

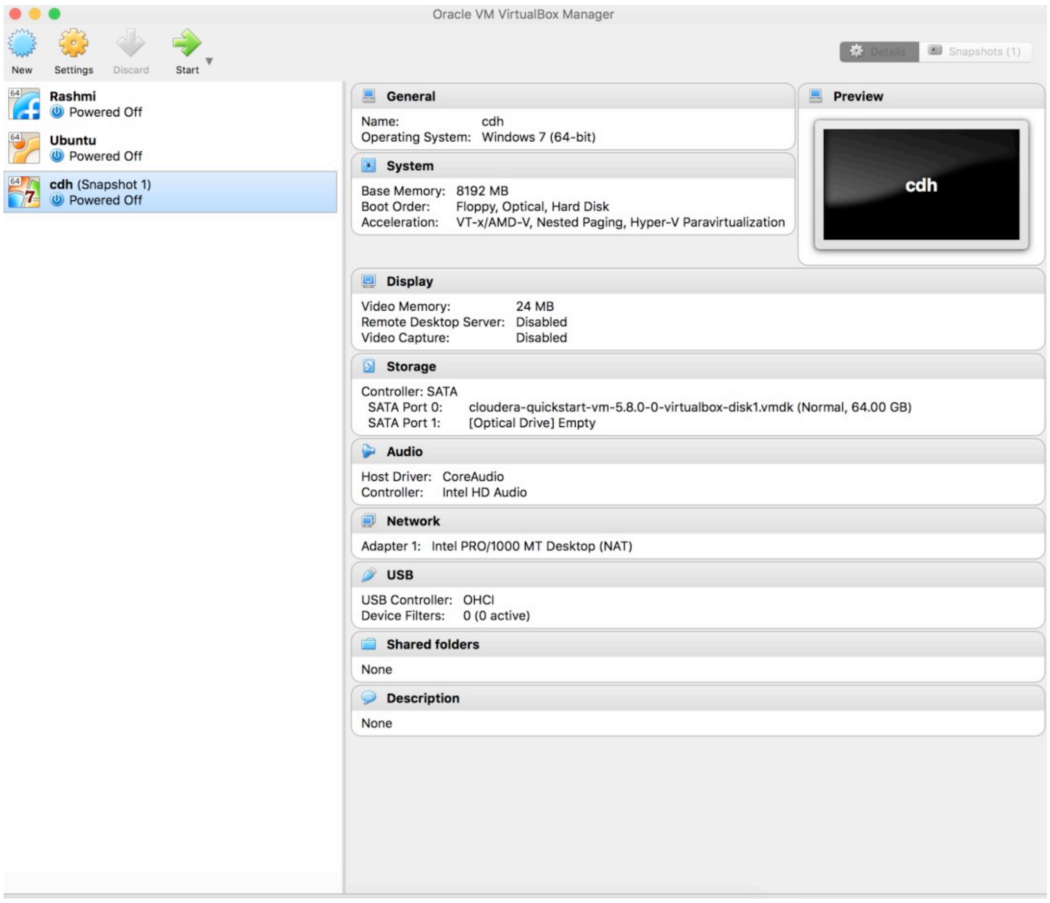
Using command cfdisk /dev/sda to verify HDD size.



```
                          Disk: /dev/sda
            Size: 10 GiB, 10737418240 bytes, 20971520 sectors
                  Label: dos, identifier: 0x4d8d9dc4

   Device         Boot        Start        End    Sectors   Size   Id Type
   /dev/sda1      *            2048    1026047    1024000   500M   83 Linux
>> /dev/sda2                1026048   20971519   19945472   9.5G   8e Linux LVM


 Partition type: Linux LVM (8e)
     Filesystem: LVM2_member
Filesystem UUID: bnPH20-h9Gg-eaeW-Cby3-7Bz8-1YWJ-JnRIyR

[Bootable]  [ Delete ]  [  Quit  ]  [  Type  ]  [  Help  ]  [  Write ]  [  Dump  ]

                 Quit program without writing partition table
```

Since we had few errors in our configuration files we used second approach i.e Cloudera which has in built hadoop.

Approach 2: MAC (cloudera-quickstart-vm-5.8.0-0-virtualbox.zip)



Rashmi
Powered Off
Ubuntu
Powered Off
cdh (Snapshot 1)
Powered Off

General
Name: cdh
Operating System: Windows 7 (64-bit)

System
Base Memory: 8192 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, Hyper-V Paravirtualization

Display
Video Memory: 24 MB
Remote Desktop Server: Disabled
Video Capture: Disabled

Storage
Controller: SATA
SATA Port 0: cloudera-quickstart-vm-5.8.0-0-virtualbox-disk1.vmdk (Normal, 64.00 GB)
SATA Port 1: [Optical Drive] Empty

Audio
Host Driver: CoreAudio
Controller: Intel HD Audio

Network
Adapter 1: Intel PRO/1000 MT Desktop (NAT)

USB
USB Controller: OHCI
Device Filters: 0 (0 active)

Shared folders
None

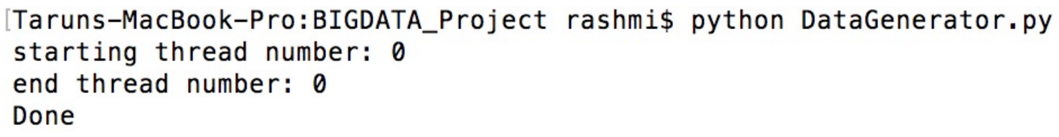
Description
None

INSTRUCTION

1. Install fake-factory-0.7.2 for generating random data and python-2.7.12
   "fake_factory-0.7.2-py2.py3-none-any.whl;
   fake-factory-0.7.2.tar.gz;
   python-2.7.12-macosx10.6.pkg"
   ("https://www.python.org/ftp/python/2.7.12/python-2.7.12-macosx10.6.pkg,
   https://www.python.org/downloads/release/python-2712/.")

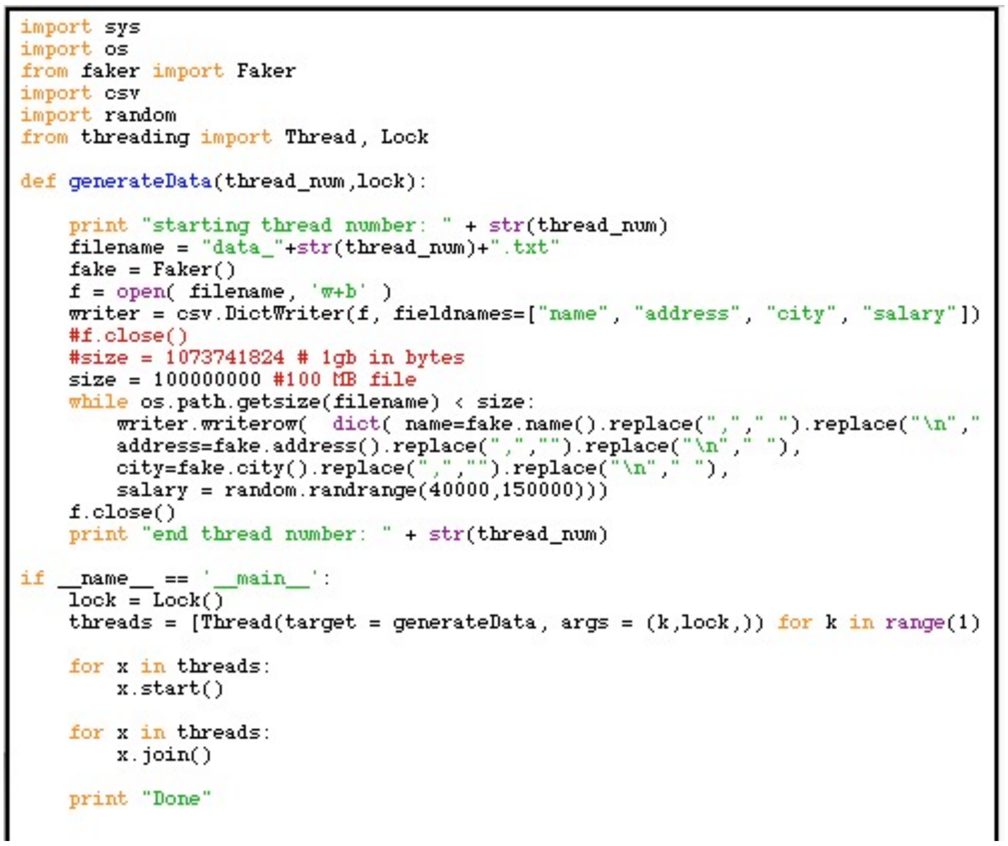
2. Check all the imports in the python scripts are running and getting executed. example on python terminal we should be able to execute "import Faker" and so on.

3. On command prompt run data generator script using "**python DataGenerator.py**"



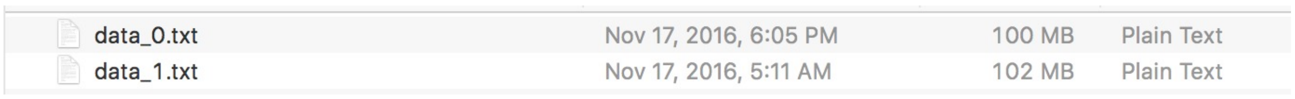
```
[Taruns-MacBook-Pro:BIGDATA_Project rashmi$ python DataGenerator.py
 starting thread number: 0
 end thread number: 0
 Done
```

Refer 1: Data generation(File name "DataGenerator.py")



```
import sys
import os
from faker import Faker
import csv
import random
from threading import Thread, Lock

def generateData(thread_num,lock):

    print "starting thread number: " + str(thread_num)
    filename = "data_"+str(thread_num)+".txt"
    fake = Faker()
    f = open( filename, 'w+b' )
    writer = csv.DictWriter(f, fieldnames=["name", "address", "city", "salary"])
    #f.close()
    #size = 1073741824 # 1gb in bytes
    size = 100000000 #100 MB file
    while os.path.getsize(filename) < size:
        writer.writerow(  dict( name=fake.name().replace(","," ").replace("\n","
        address=fake.address().replace(",","").replace("\n"," "),
        city=fake.city().replace(",","").replace("\n"," "),
        salary = random.randrange(40000,150000)))
    f.close()
    print "end thread number: " + str(thread_num)

if __name__ == '__main__':
    lock = Lock()
    threads = [Thread(target = generateData, args = (k,lock,)) for k in range(1)

    for x in threads:
        x.start()

    for x in threads:
        x.join()

    print "Done"
```

4. This script will generate (data_0.txt)data file with provided size. We can adjust the variable "**for k in range(1)**" for generating either one data_0.txt or more than one data text file (example if we say range(2) then it will generate files data_0.txt and data_1.txt files with random data)



data_0.txt	Nov 17, 2016, 6:05 PM	100 MB	Plain Text
data_1.txt	Nov 17, 2016, 5:11 AM	102 MB	Plain Text

5. Create a folder "local_data" on the virtual machine.

6. Place the data generated files i.e. "data_0.txt" along with the "MapLogic.py" and "ReduceLogic.py" in that folder.

7. We will be executing the python files through Hadoop **streaming.jar**

8. On command prompt/ terminal go to the specified folder which has all the data file and the python scripts i.e. in our case "cd Desktop/local_data" (we can double check using "ls" command to see if all the files are available in local_data folder)

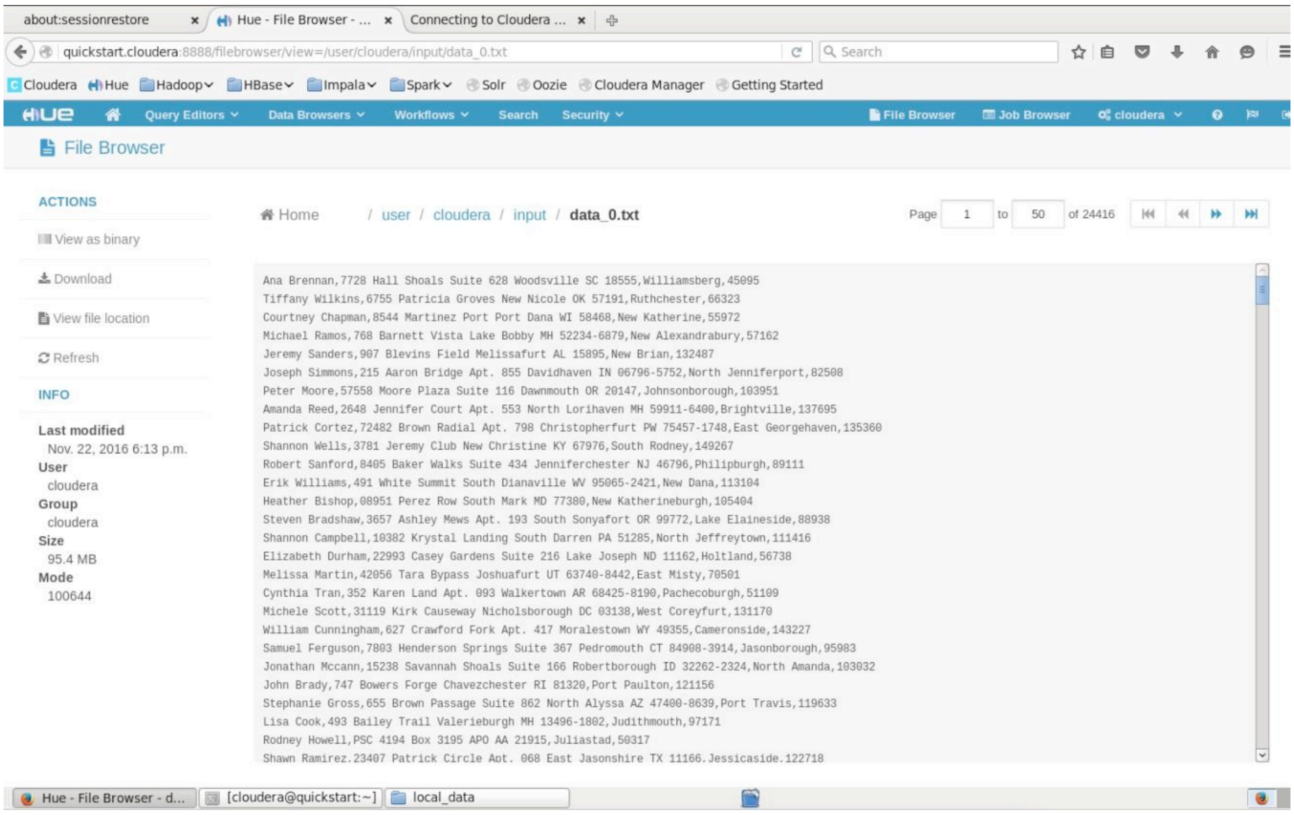
9. Next step is to move all the files to Hadoop space, for that we need to go to Firefox and click on "HUE" bookmark(since we have downloaded cdh . Login to HUE with username/password as "**cloudera/cloudera**".

10. Go to File Browser tab- create new directory—**input**

11. Next go back to terminal—enter below command —"**hadoop fs -copyFromLocal data_0.txt /user/cloudera/input**"

12. Once the command is executed successfully you should be able to see the data-0.txt file in the "/user/cloudera/input" location in HUE File browser section.

13. The **INPUT** file is in format of "username, address, city name and salary"



about:sessionrestore
Hue - File Browser - ...
Connecting to Cloudera ...
quickstart.cloudera:8888/filebrowser/view=/user/cloudera/input/data_0.txt
Cloudera Hue Hadoop HBase Impala Spark Solr Oozie Cloudera Manager Getting Started
Query Editors Data Browsers Workflows Search Security
File Browser Job Browser cloudera
File Browser
ACTIONS
View as binary
Download
View file location
Refresh
INFO
Last modified
Nov. 22, 2016 6:13 p.m.
User
cloudera
Group
cloudera
Size
95.4 MB
Mode
100644
Home / user / cloudera / input / data_0.txt
Page 1 to 50 of 24416
Hue - File Browser - d...
[cloudera@quickstart:~]
local_data

```
Ana Brennan,7728 Hall Shoals Suite 628 Woodsville SC 18555,Williamsberg,45095
Tiffany Wilkins,6755 Patricia Groves New Nicole OK 57191,Ruthchester,66323
Courtney Chapman,8544 Martinez Port Port Dana WI 58468,New Katherine,55972
Michael Ramos,768 Barnett Vista Lake Bobby MH 52234-6879,New Alexandrabury,57162
Jeremy Sanders,907 Blevins Field Melissafurt AL 15895,New Brian,132487
Joseph Simmons,215 Aaron Bridge Apt. 855 Davidhaven IN 06796-5752,North Jenniferport,82508
Peter Moore,57558 Moore Plaza Suite 116 Dawnmouth OR 20147,Johnsonborough,103951
Amanda Reed,2648 Jennifer Court Apt. 553 North Lorihaven MH 59911-6400,Brightville,137695
Patrick Cortez,72482 Brown Radial Apt. 798 Christopherfurt PW 75457-1748,East Georgehaven,135360
Shannon Wells,3781 Jeremy Club New Christine KY 67976,South Rodney,149267
Robert Sanford,8405 Baker Walks Suite 434 Jenniferchester NJ 46796,Philipburgh,89111
Erik Williams,491 White Summit South Dianaville WV 95065-2421,New Dana,113104
Heather Bishop,08951 Perez Row South Mark MD 77380,New Katherineburgh,105404
Steven Bradshaw,3657 Ashley Mews Apt. 193 South Sonyafort OR 99772,Lake Elaineside,88938
Shannon Campbell,10382 Krystal Landing South Darren PA 51285,North Jeffreytown,111416
Elizabeth Durham,22993 Casey Gardens Suite 216 Lake Joseph ND 11162,Holtland,56738
Melissa Martin,42056 Tara Bypass Joshuafurt UT 63740-8442,East Misty,70501
Cynthia Tran,352 Karen Land Apt. 093 Walkertown AR 68425-8190,Pachecoburgh,51109
Michele Scott,31119 Kirk Causeway Nicholsborough DC 03138,West Coreyfurt,131170
William Cunningham,627 Crawford Fork Apt. 417 Moralestown WY 49355,Cameronside,143227
Samuel Ferguson,7803 Henderson Springs Suite 367 Pedromouth CT 84908-3914,Jasonborough,95983
Jonathan Mccann,15238 Savannah Shoals Suite 166 Robertborough ID 32262-2324,North Amanda,103032
John Brady,747 Bowers Forge Chavezchester RI 81320,Port Paulton,121156
Stephanie Gross,655 Brown Passage Suite 862 North Alyssa AZ 47400-8639,Port Travis,119633
Lisa Cook,493 Bailey Trail Valerieburgh MH 13496-1802,Judithmouth,97171
Rodney Howell,PSC 4194 Box 3195 APO AA 21915,Juliastad,50317
Shawn Ramirez.23407 Patrick Circle Apt. 068 East Jasonshire TX 11166.Jessicaside.122718
```

14. **MAP LOGIC** :

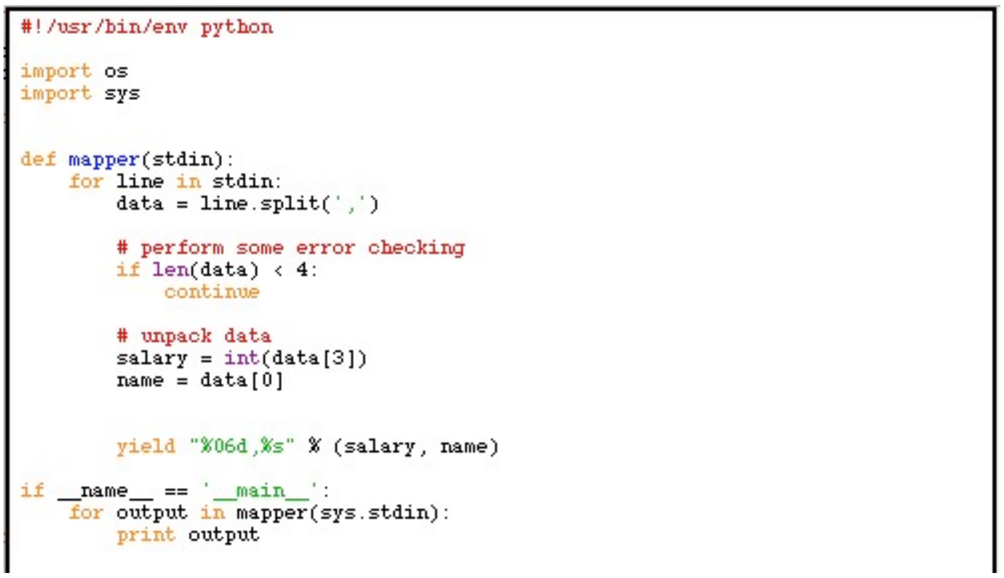
- Splitting each line by comma; size restriction
- Getting the salary and name field
- Between the map and reducer logic the data is automatically sorted with the key
- output is comma separated.(i.e salary, name)

MapReduce Sort Algorithm

Map and Reduce used, why? Data will be stored in HDFS and the data will be processes with MapReduce. Processing that serially from the top to the bottom could take a long time. Instead, MapReduce is designed to process data in parallel. As a result, the file is broken in the chunks, and then processed in parallel.

MapReduce job will sort the salaries. Here is code for MapReduce.

File name MapLogic.py

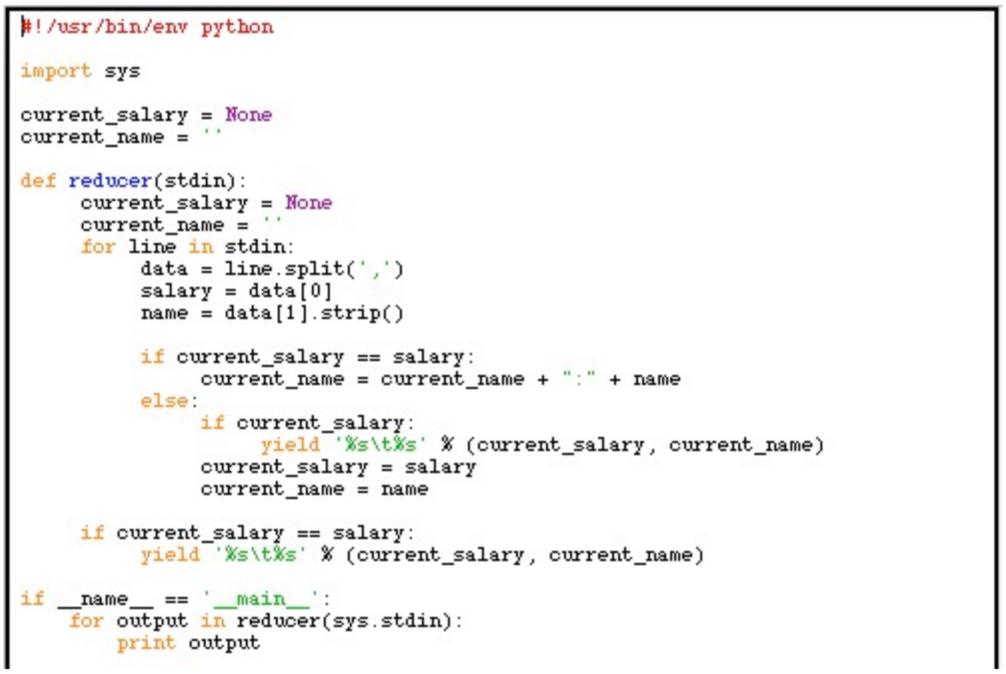


```
#!/usr/bin/env python

import os
import sys


def mapper(stdin):
    for line in stdin:
        data = line.split(',')

        # perform some error checking
        if len(data) < 4:
            continue

        # unpack data
        salary = int(data[3])
        name = data[0]


        yield "%06d,%s" % (salary, name)

if __name__ == '__main__':
    for output in mapper(sys.stdin):
        print output
```

15. **REDUCER LOGIC**

- Output from Map is taken as input by Reducer logic.
- Data is first split by comma, assigning values to variable salary and name.
- Later we are trying to match the respective salary with the associated user.

File name ReduceLogic.py



```
#!/usr/bin/env python

import sys

current_salary = None
current_name = ''

def reducer(stdin):
    current_salary = None
    current_name = ''
    for line in stdin:
        data = line.split(',')
        salary = data[0]
        name = data[1].strip()

        if current_salary == salary:
            current_name = current_name + ":" + name
        else:
            if current_salary:
                yield '%s\t%s' % (current_salary, current_name)
            current_salary = salary
            current_name = name

    if current_salary == salary:
        yield '%s\t%s' % (current_salary, current_name)

if __name__ == '__main__':
    for output in reducer(sys.stdin):
        print output
```

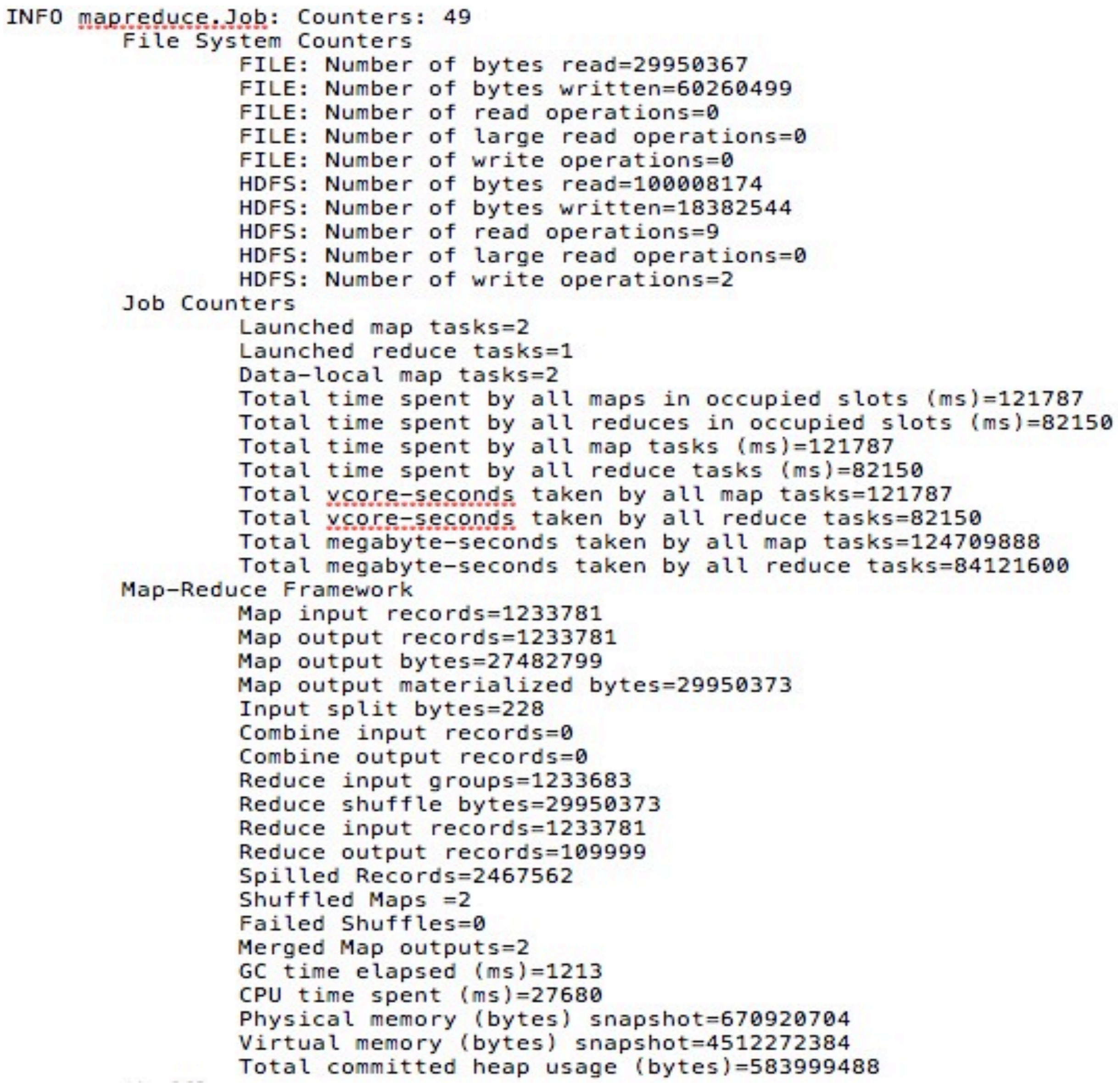
16. Next step is to execute "**hadoop jar /usr/lib/hadoop-0.20-mapreduce/contrib/streaming/hadoop-streaming-2.6.0-mr1-cdh5.8.0.jar -input input/* -**

output output_streaming -mapper MapLogic.py -reducer ReduceLogic.py -jobconf mapred.reduce.tasks=1 -file MapLogic.py -file ReduceLogic.py" command.

17. Below is the explanation for the command used:

- Input taken from **-input input/*** path
- Output is placed in **-output output_streaming** path.
- Mapper logic is read from desktop from **-mapper MapLogic.py** file.
- Reducer logic is picked from **-reducer ReduceLogic.py** file**.**
- **-jobconf mapred.reduce.tasks=1** part of the command says that just run one reducer so that the output will be written in one single file
- **-file MapLogic.py -file ReduceLogic.py** part of command tells us that we are sending the file to every node in the machine.
-

Execute output_console: Output:

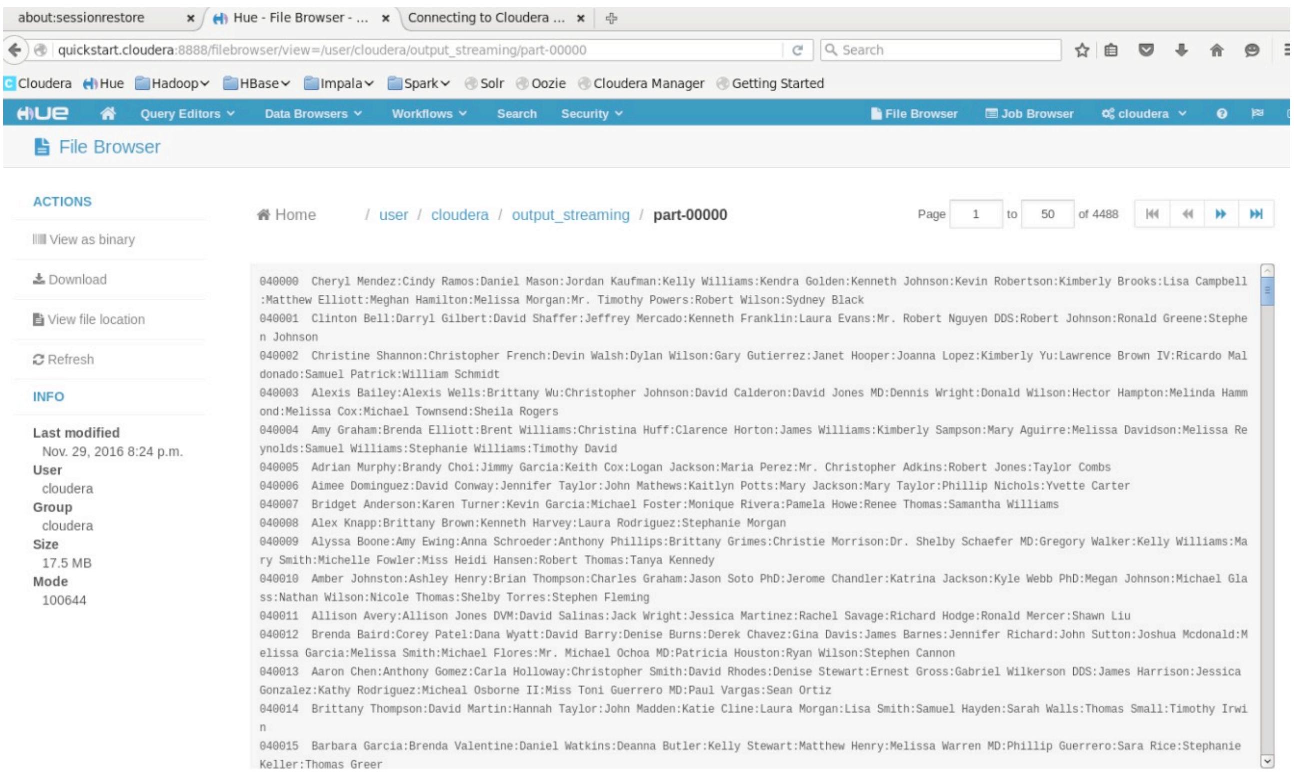


```
INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=29950367
                FILE: Number of bytes written=60260499
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=100008174
                HDFS: Number of bytes written=18382544
                HDFS: Number of read operations=9
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=2
                Launched reduce tasks=1
                Data-local map tasks=2
                Total time spent by all maps in occupied slots (ms)=121787
                Total time spent by all reduces in occupied slots (ms)=82150
                Total time spent by all map tasks (ms)=121787
                Total time spent by all reduce tasks (ms)=82150
                Total vcore-seconds taken by all map tasks=121787
                Total vcore-seconds taken by all reduce tasks=82150
                Total megabyte-seconds taken by all map tasks=124709888
                Total megabyte-seconds taken by all reduce tasks=84121600
        Map-Reduce Framework
                Map input records=1233781
                Map output records=1233781
                Map output bytes=27482799
                Map output materialized bytes=29950373
                Input split bytes=228
                Combine input records=0
                Combine output records=0
                Reduce input groups=1233683
                Reduce shuffle bytes=29950373
                Reduce input records=1233781
                Reduce output records=109999
                Spilled Records=2467562
                Shuffled Maps =2
                Failed Shuffles=0
                Merged Map outputs=2
                GC time elapsed (ms)=1213
                CPU time spent (ms)=27680
                Physical memory (bytes) snapshot=670920704
                Virtual memory (bytes) snapshot=4512272384
                Total committed heap usage (bytes)=583999488
```

```
Shuffle Errors
        BAD_ID=0
        CONNECTION=0
        IO_ERROR=0
        WRONG_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
File Input Format Counters
        Bytes Read=100007946
File Output Format Counters
        Bytes Written=18382544
```

18. In case of any errors in executing the above command double check on the location of hadoop jar on the Virtual Machine.

- PATH: "**hadoop jar /usr/lib/hadoop-0.20-mapreduce/contrib/streaming/hadoop-streaming-2.6.0-mr1-cdh5.5.0.jar**"
- Once the command is executed successfully go back to HUE, file browser and check if the output folder is created or not. And we can find the output file which gets created.
- To get the data from hadoop file system to local file system we can run the below command: "**hadoop fs -copyToLocal /user/cloudera/output_streaming/* **"
- If you enter "ls" command we should be able to see part-00000 as code and then we can check the first five or any respective entries in the output file. e.g.: "head -10 part-00000", we should be able to see part of output file where it starts from 040000 salary as the first entry, followed by 040001, 040002….. and so on.
- To check the records from bottom of the output file we can use command: "tail -10 part-00000" where we can see the last record will have a salary of 149999(As size < 150000)



Home / user / cloudera / output_streaming / part-00000

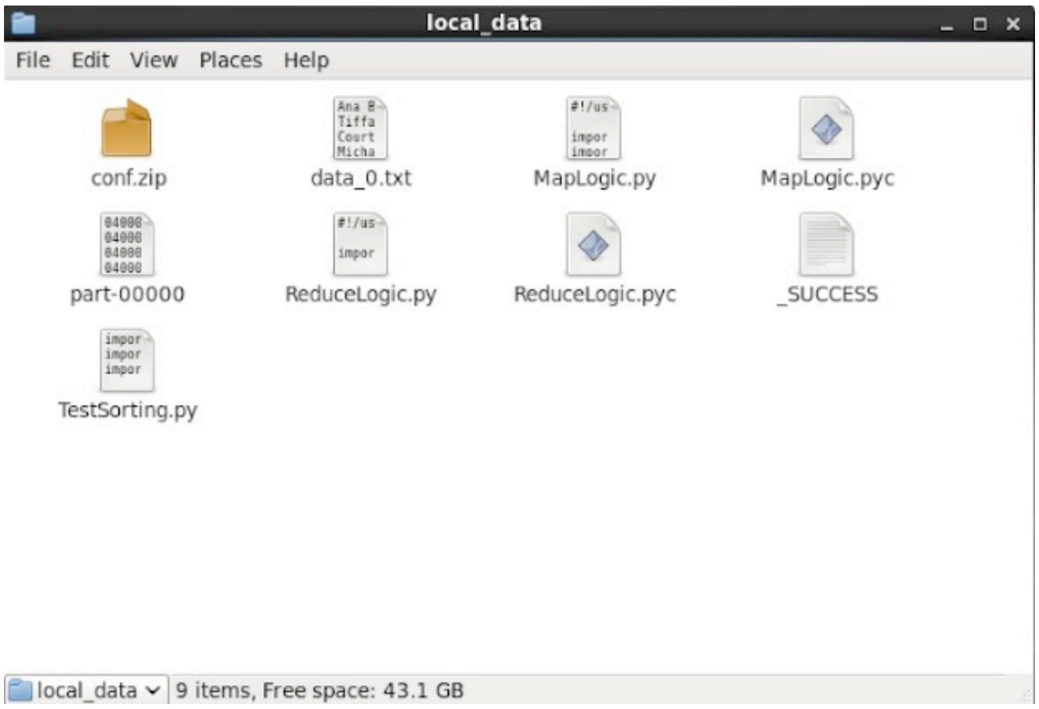
Last modified
Nov. 29, 2016 8:24 p.m.
User
cloudera
Group
cloudera
Size
17.5 MB
Mode
100644

```
040000  Cheryl Mendez:Cindy Ramos:Daniel Mason:Jordan Kaufman:Kelly Williams:Kendra Golden:Kenneth Johnson:Kevin Robertson:Kimberly Brooks:Lisa Campbell
:Matthew Elliott:Meghan Hamilton:Melissa Morgan:Mr. Timothy Powers:Robert Wilson:Sydney Black
040001  Clinton Bell:Darryl Gilbert:David Shaffer:Jeffrey Mercado:Kenneth Franklin:Laura Evans:Mr. Robert Nguyen DDS:Robert Johnson:Ronald Greene:Stephe
n Johnson
040002  Christine Shannon:Christopher French:Devin Walsh:Dylan Wilson:Gary Gutierrez:Janet Hooper:Joanna Lopez:Kimberly Yu:Lawrence Brown IV:Ricardo Mal
donado:Samuel Patrick:William Schmidt
040003  Alexis Bailey:Alexis Wells:Brittany Wu:Christopher Johnson:David Calderon:David Jones MD:Dennis Wright:Donald Wilson:Hector Hampton:Melinda Hamm
ond:Melissa Cox:Michael Townsend:Sheila Rogers
040004  Amy Graham:Brenda Elliott:Brent Williams:Christina Huff:Clarence Horton:James Williams:Kimberly Sampson:Mary Aguirre:Melissa Davidson:Melissa Re
ynolds:Samuel Williams:Stephanie Williams:Timothy David
040005  Adrian Murphy:Brandy Choi:Jimmy Garcia:Keith Cox:Logan Jackson:Maria Perez:Mr. Christopher Adkins:Robert Jones:Taylor Combs
040006  Aimee Dominguez:David Conway:Jennifer Taylor:John Mathews:Kaitlyn Potts:Mary Jackson:Mary Taylor:Phillip Nichols:Yvette Carter
040007  Bridget Anderson:Karen Turner:Kevin Garcia:Michael Foster:Monique Rivera:Pamela Howe:Renee Thomas:Samantha Williams
040008  Alex Knapp:Brittany Brown:Kenneth Harvey:Laura Rodriguez:Stephanie Morgan
040009  Alyssa Boone:Amy Ewing:Anna Schroeder:Anthony Phillips:Brittany Grimes:Christie Morrison:Dr. Shelby Schaefer MD:Gregory Walker:Kelly Williams:Ma
ry Smith:Michelle Fowler:Miss Heidi Hansen:Robert Thomas:Tanya Kennedy
040010  Amber Johnston:Ashley Henry:Brian Thompson:Charles Graham:Jason Soto PhD:Jerome Chandler:Katrina Jackson:Kyle Webb PhD:Megan Johnson:Michael Gla
ss:Nathan Wilson:Nicole Thomas:Shelby Torres:Stephen Fleming
040011  Allison Avery:Allison Jones DVM:David Salinas:Jack Wright:Jessica Martinez:Rachel Savage:Richard Hodge:Ronald Mercer:Shawn Liu
040012  Brenda Baird:Corey Patel:Dana Wyatt:David Barry:Denise Burns:Derek Chavez:Gina Davis:James Barnes:Jennifer Richard:John Sutton:Joshua Mcdonald:M
elissa Garcia:Melissa Smith:Michael Flores:Mr. Michael Ochoa MD:Patricia Houston:Ryan Wilson:Stephen Cannon
040013  Aaron Chen:Anthony Gomez:Carla Holloway:Christopher Smith:David Rhodes:Denise Stewart:Ernest Gross:Gabriel Wilkerson DDS:James Harrison:Jessica
Gonzalez:Kathy Rodriguez:Micheal Osborne II:Miss Toni Guerrero MD:Paul Vargas:Sean Ortiz
040014  Brittany Thompson:David Martin:Hannah Taylor:John Madden:Katie Cline:Laura Morgan:Lisa Smith:Samuel Hayden:Sarah Walls:Thomas Small:Timothy Irwi
n
040015  Barbara Garcia:Brenda Valentine:Daniel Watkins:Deanna Butler:Kelly Stewart:Matthew Henry:Melissa Warren MD:Phillip Guerrero:Sara Rice:Stephanie
Keller:Thomas Greer
```

19. The output file will be in output_streaming(**/user/cloudera/output_streaming)** folder in HUE.

In case we want to insert more input files we can place it in INPUT folder and execute the command mentioned in step 12. Also we should double check that the output_streaming folder is deleted so that it creates it newly and generates the respective output file.

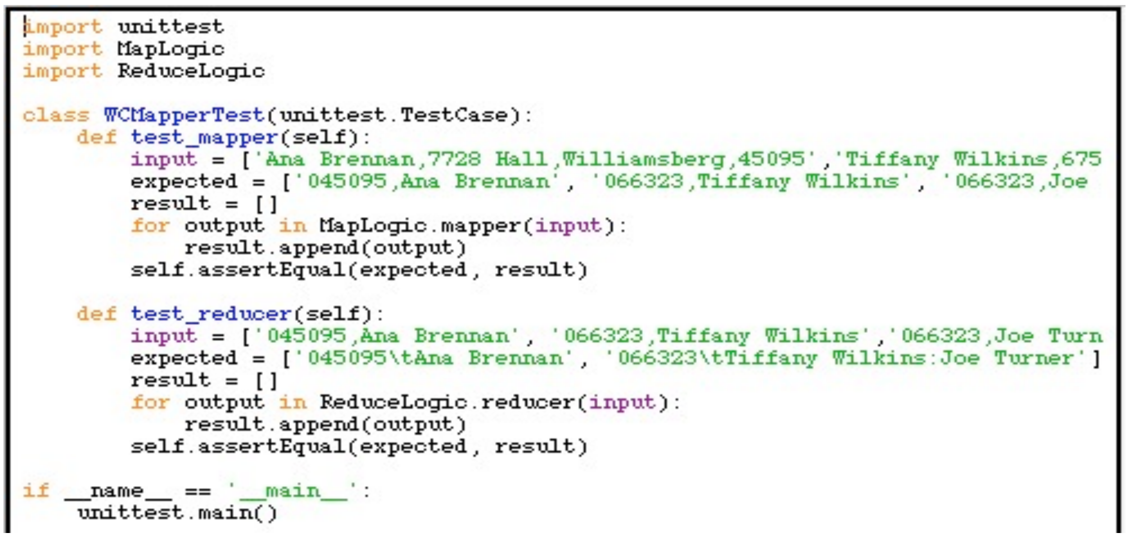
Below is the local_data folder with moved output file (part-00000) from the Hadoop space and other scripts which help us execute the code using cloudera.



20. Data Validation:

In this step, we validate the input data with the expected output. If the expected data matches the input Test data then it passes the validation test, if not the failure description is displayed on the console. We are validating both input from mapper and reducer in our validation i.e TestSorting.py code. We place this file in same local_data folder and execute it once our output is generated using the Map and reducer logic. Command to execute it "python TestSorting.py" on terminal/command prompt.

file Name: TestSorting.py



```
import unittest
import MapLogic
import ReduceLogic

class WCMapperTest(unittest.TestCase):
    def test_mapper(self):
        input = ['Ana Brennan,7728 Hall,Williamsberg,45095','Tiffany Wilkins,675
        expected = ['045095,Ana Brennan', '066323,Tiffany Wilkins', '066323,Joe
        result = []
        for output in MapLogic.mapper(input):
            result.append(output)
        self.assertEqual(expected, result)

    def test_reducer(self):
        input = ['045095,Ana Brennan', '066323,Tiffany Wilkins','066323,Joe Turn
        expected = ['045095\tAna Brennan', '066323\tTiffany Wilkins:Joe Turner']
        result = []
        for output in ReduceLogic.reducer(input):
            result.append(output)
        self.assertEqual(expected, result)

if __name__ == '__main__':
    unittest.main()
```

Mapper

input = ['Ana Brennan,7728 Hall,Williamsberg,45095','Tiffany Wilkins,6755 Patricia 57191,Ruthchester,66323','Joe Turner,6755 Patricia 57191,Ruthchester,66323']

expected = ['045095,Ana Brennan', '066323,Tiffany Wilkins', '066323,Joe Turner']

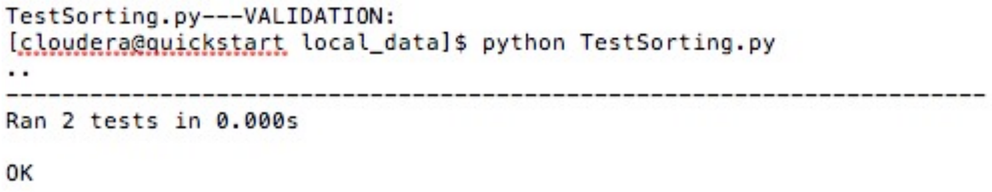
Reducer

input = ['045095,Ana Brennan', '066323,Tiffany Wilkins','066323,Joe Turner']

expected = ['045095\tAna Brennan', '066323\tTiffany Wilkins:Joe Turner']

<u>**Result**</u>

When input is valid



```
TestSorting.py---VALIDATION:
[cloudera@quickstart local_data]$ python TestSorting.py
..
----------------------------------------------------------------------
Ran 2 tests in 0.000s

OK
```

Mapper

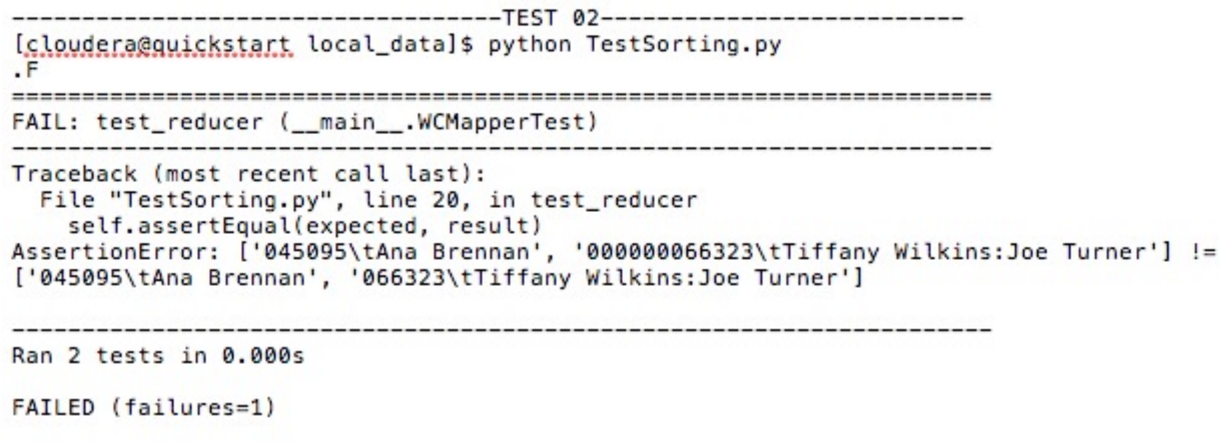
input = ['Ana Brennan,7728 Hall,Williamsberg,45095','Tiffany Wilkins,6755 Patricia 57191,Ruthchester,66323','Joe Turner,6755 Patricia 57191,Ruthchester,66323']

expected = ['045095,Ana Brennan', '066323,Tiffany Wilkins', '066323,Joe Turner']

Reducer

input = ['045095,Ana Brennan', '**000000066323**,Tiffany Wilkins','066323,Joe Turner'] expected = ['045095\tAna Brennan', **'066323**\tTiffany Wilkins:Joe Turner']

<u>Result</u>: When input is invalid

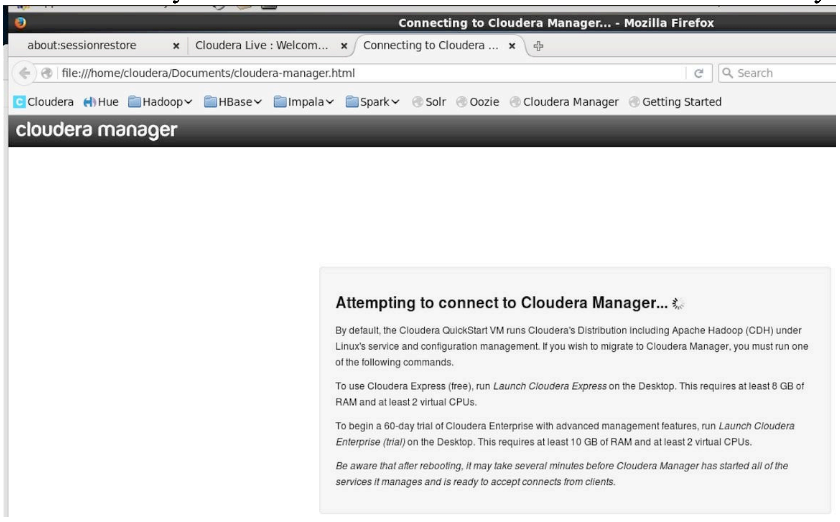


```
-----------------------------------TEST 02---------------------------
[cloudera@quickstart local_data]$ python TestSorting.py
.F
======================================================================
FAIL: test_reducer (__main__.WCMapperTest)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "TestSorting.py", line 20, in test_reducer
    self.assertEqual(expected, result)
AssertionError: ['045095\tAna Brennan', '000000066323\tTiffany Wilkins:Joe Turner'] !=
['045095\tAna Brennan', '066323\tTiffany Wilkins:Joe Turner']

----------------------------------------------------------------------
Ran 2 tests in 0.000s

FAILED (failures=1)
```

Performance/Analysis

Performance: latency = start time – end time

As we used cloud era, we had to look at the Cloud Manager for Performance latency, but it did not load for the system. We tried to load it since two continuous days.



Connecting to Cloudera Manager... - Mozilla Firefox
about:sessionrestore
Cloudera Live : Welcom...
Connecting to Cloudera ...
file:///home/cloudera/Documents/cloudera-manager.html
Search
Cloudera
Hue
Hadoop
HBase
Impala
Spark
Solr
Oozie
Cloudera Manager
Getting Started
cloudera manager

Attempting to connect to Cloudera Manager...

By default, the Cloudera QuickStart VM runs Cloudera's Distribution including Apache Hadoop (CDH) under Linux's service and configuration management. If you wish to migrate to Cloudera Manager, you must run one of the following commands.

To use Cloudera Express (free), run *Launch Cloudera Express* on the Desktop. This requires at least 8 GB of RAM and at least 2 virtual CPUs.

To begin a 60-day trial of Cloudera Enterprise with advanced management features, run *Launch Cloudera Enterprise (trial)* on the Desktop. This requires at least 10 GB of RAM and at least 2 virtual CPUs.

Be aware that after rebooting, it may take several minutes before Cloudera Manager has started all of the services it manages and is ready to accept connects from clients.

Conclusion/Summary

We were successfully able to install Hadoop on system. We performed the data generation, data sorting and data validation on cloudera which was successful. We followed two approaches to perform the above operations on data, using Hadoop and cloudera. So we gained knowledge about both the platforms.

References

www.**cloudera**.com/products/apache-hadoop/**hue**.html
https://www.tutorialspoint.com/hadoop/hadoop_mapreduce.htm
www.**cloudera**.com/
www.hadoop.apache.org

APPENDIX

DataGenerator.py

```
import sys
import os
from faker import Faker
import csv
import random
from threading import Thread, Lock

def generateData(thread_num,lock):

    print "starting thread number: " + str(thread_num)
    filename = "data_"+str(thread_num)+".txt"
    fake = Faker()
    f = open( filename, 'w+b' )
    writer = csv.DictWriter(f, fieldnames=["name", "address", "city", "salary"])
    #f.close()
    size = 1073741824 # 1gb in bytes
    #size = 100000000 #100 MB file
    while os.path.getsize(filename) < size:
        writer.writerow(  dict( name=fake.name().replace(","," ").replace("\n"," "),
        address=fake.address().replace(",","").replace("\n"," "),
        city=fake.city().replace(",","").replace("\n"," "),
        salary = random.randrange(40000,150000)))
    f.close()
    print "end thread number: " + str(thread_num)

if __name__ == '__main__':
    lock = Lock()
    threads = [Thread(target = generateData, args = (k,lock,)) for k in range(1)]

    for x in threads:
        x.start()

    for x in threads:
        x.join()

    print "Done"
```

ReduceLogic.py

```
#!/usr/bin/env python

import sys

current_salary = None
current_name = ''

def reducer(stdin):
     current_salary = None
     current_name = ''
     for line in stdin:
          data = line.split(',')
          salary = data[0]
          name = data[1].strip()

          if current_salary == salary:
              current_name = current_name + ":" + name
          else:
              if current_salary:
                  yield '%s\t%s' % (current_salary, current_name)
              current_salary = salary
              current_name = name

     if current_salary == salary:
          yield '%s\t%s' % (current_salary, current_name)

if __name__ == '__main__':
    for output in reducer(sys.stdin):
        print output
```

MapLogic.py

```
#!/usr/bin/env python

import os
import sys


def mapper(stdin):
    for line in stdin:
        data = line.split(',')

        # perform some error checking
        if len(data) < 4:
            continue

        # unpack data
        salary = int(data[3])
        name = data[0]


        yield "%06d,%s" % (salary, name)

if __name__ == '__main__':
    for output in mapper(sys.stdin):
        print output
```

TestSorting.py

```
import unittest
import MapLogic
import ReduceLogic

class WCMapperTest(unittest.TestCase):
    def test_mapper(self):
        input = ['Ana Brennan,7728 Hall,Williamsberg,45095','Tiffany Wilkins,6755 Patricia
57191,Ruthchester,66323','Joe Turner,6755 Patricia 57191,Ruthchester,66323']
        expected = ['045095,Ana Brennan', '066323,Tiffany Wilkins', '066323,Joe Turner']
        result = []
        for output in MapLogic.mapper(input):
            result.append(output)
        self.assertEqual(expected, result)

    def test_reducer(self):
        input = ['045095,Ana Brennan', '066323,Tiffany Wilkins','066323,Joe Turner']
        expected = ['045095\tAna Brennan', '066323\tTiffany Wilkins:Joe Turner']
        result = []
        for output in ReduceLogic.reducer(input):
            result.append(output)
        self.assertEqual(expected, result)

if __name__ == '__main__':
    unittest.main()
```