Computer Science & Information Systems

**Big Data Systems – Hadoop Lab Sheet 5**

**MapReduce Pipelines**

1. Objective:
   1. Teach students how to write a complex map reduce pipeline in Python (multiple input, single output)

**Hadoop MapReduce**

In Hadoop, MapReduce is a computation that decomposes large manipulation jobs into individual tasks that can be executed in parallel across a cluster of servers. The results of tasks can be joined together to compute final results.

MapReduce consists of 2 steps:

* **Map Function** – It takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (Key-Value pair).
* **Reduce Function** – Takes the output from Map as an input and combines those data tuples into a smaller set of tuples.

**Pre-requisites:**

1. Hadoop should be installed
2. Hadoop cluster should be up and running
3. Python should be installed
4. Steps to be performed:

The User file has the following fields:

* *UserId – Unique identification number for each user*
* *UserName – Name of user*
* *Language – Language that user speaks*
* *Country – Country of residence of user*

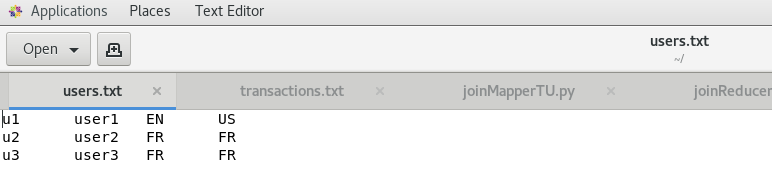
The Transactions file has the following fields:

* *TransactionId – Unique transaction number*
* *ProductId – Product ID of product*
* *UserId – Unique identification number for each user*
* *PurchaseAmount – amount of each transaction*
* *ProductDescription – description of product*

Use map reduce to find, for each product, the number of locations in which that product was purchased.

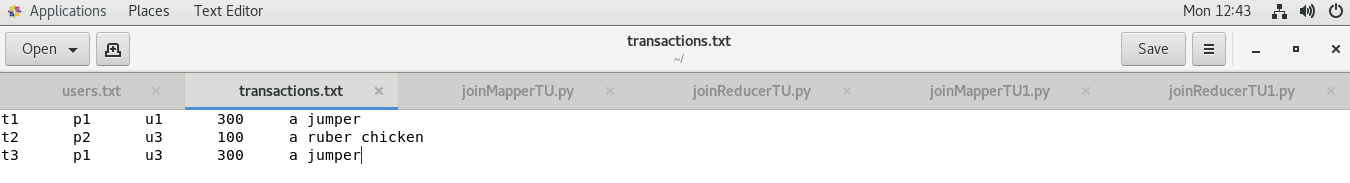
1. Create a file with the content provided in the attached users.txt file.

gedit users.txt



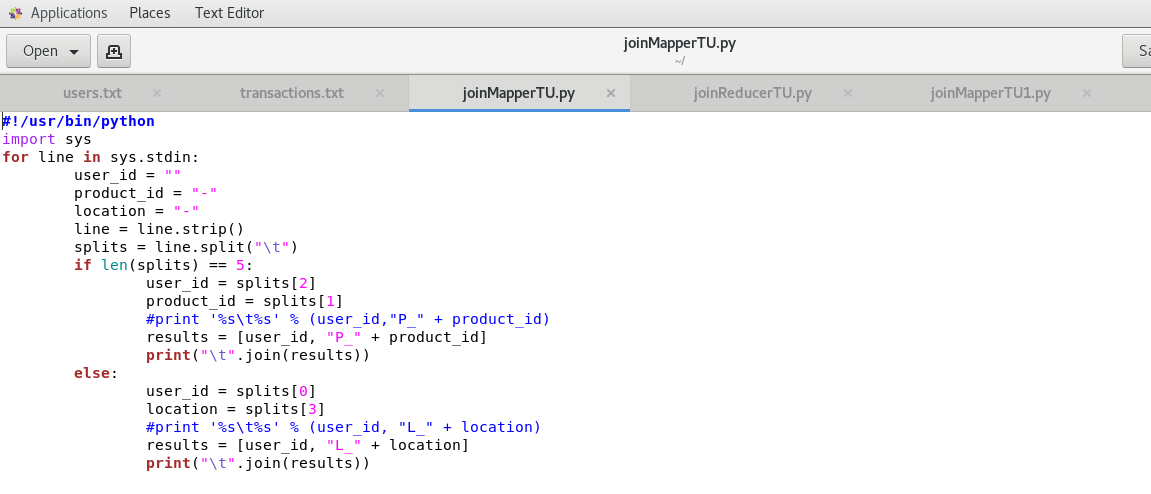
1. Create a file with the content provided in the attached transactions.txt file.

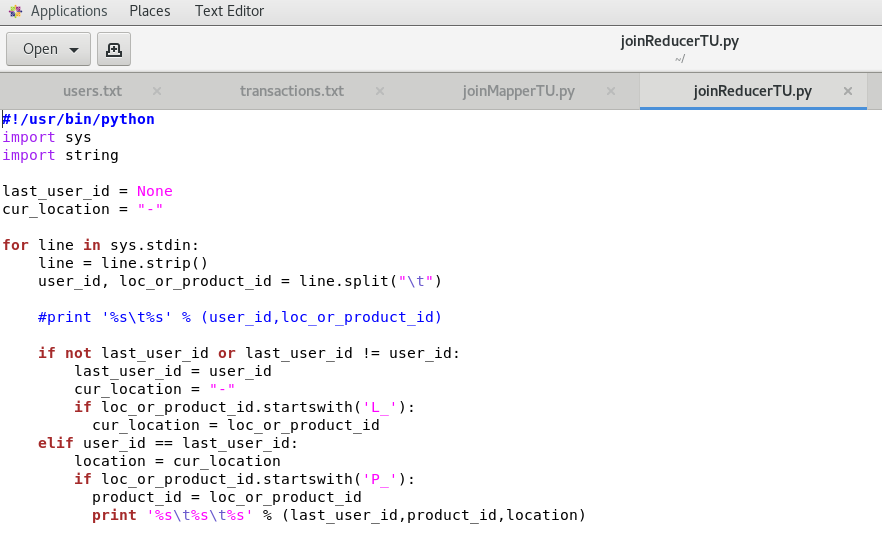
gedit transactions.txt



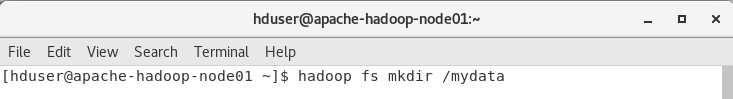
1. Write the mapper and reducer code to join the two files i.e. users.txt and transactions.txt. Mapper joins the two files based on the common key i.e. UserID. Then reducer is used to output UserId, ProductId and Location.

See attached joinMapperTU.py and joinReducerTU.py for the code.

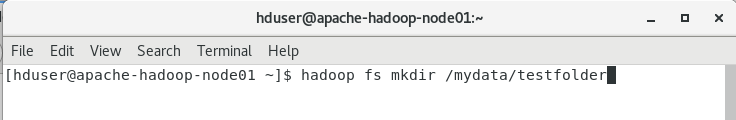


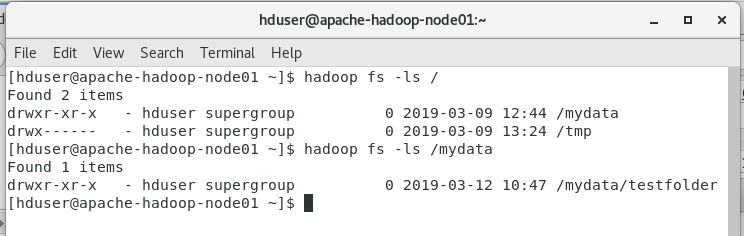


1. Create directory in Hadoop file system where you want to store your data files
   1. hadoop fs -mkdir /mydata

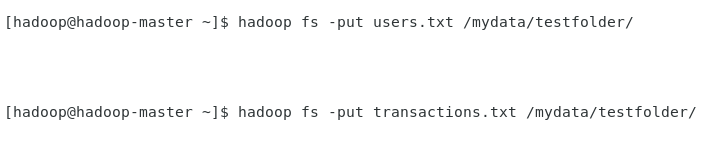


* 1. hadoop fs -mkdir /mydata/testfolder



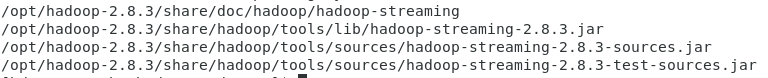


1. Move the data file i.e. users.txt and transactions.txt to Hadoop file system
   1. hadoop fs -put users.txt /mydata/testfolder/
   2. hadoop fs -put transactions.txt /mydata/testfolder/

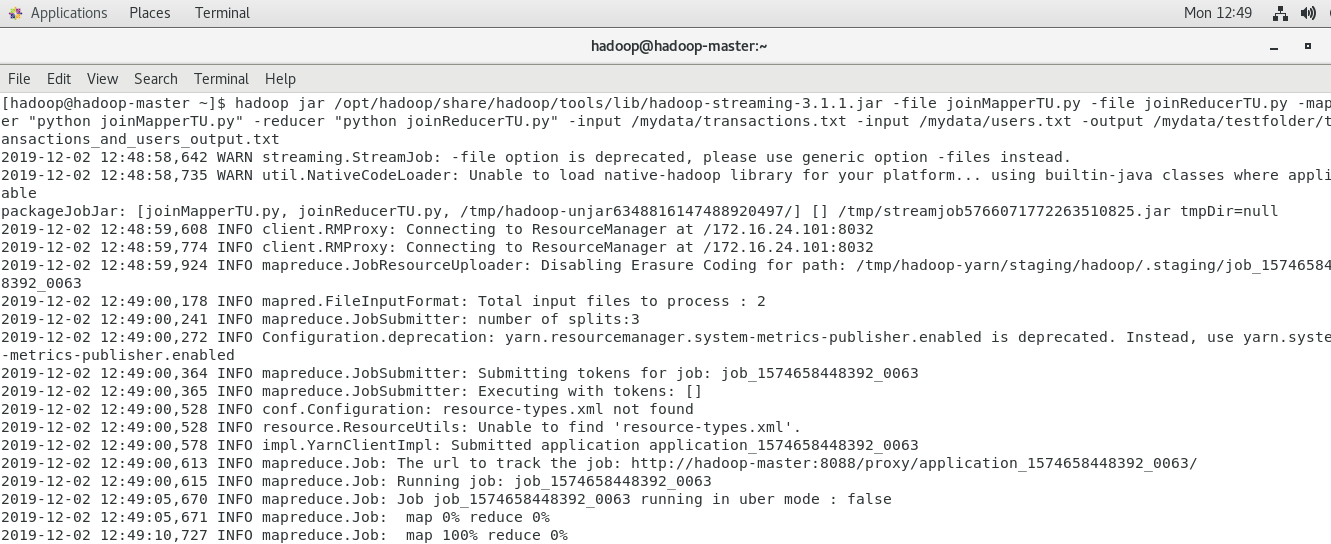


1. Find Hadoop streaming location
   1. find / -name Hadoop\*stream\*

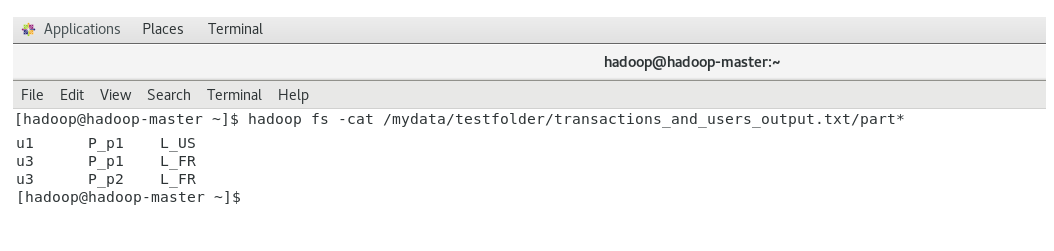




1. Run map reduce using streaming
   1. hadoop jar /opt/hadoop-2.8.3/share/hadoop/tools/lib/hadoop-streaming-2.8.3.jar -file joinMapperTU.py -file joinReducerTU.py -mapper "python joinMapperTU.py" -reducer "python joinReducerTU.py" -input /mydata/transactions.txt -input /mydata/users.txt -output /mydata/testfolder/transactions\_and\_users\_output.txt

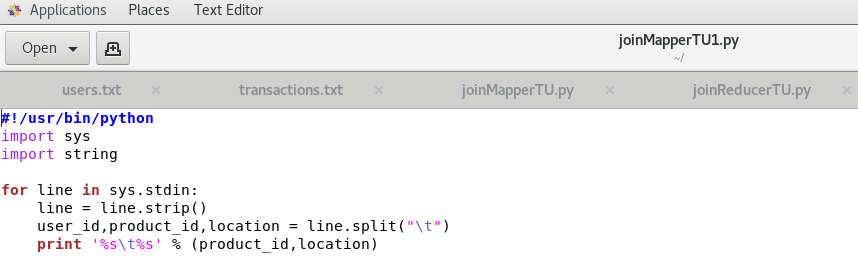


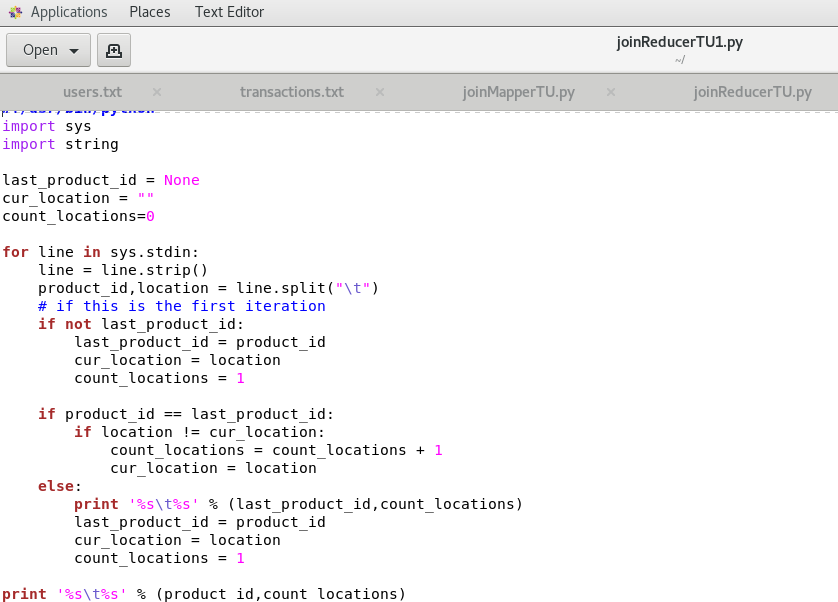
1. Check the output
   1. hadoop fs -cat /mydata/testfolder/transactions\_and\_users\_output.txt/part\*



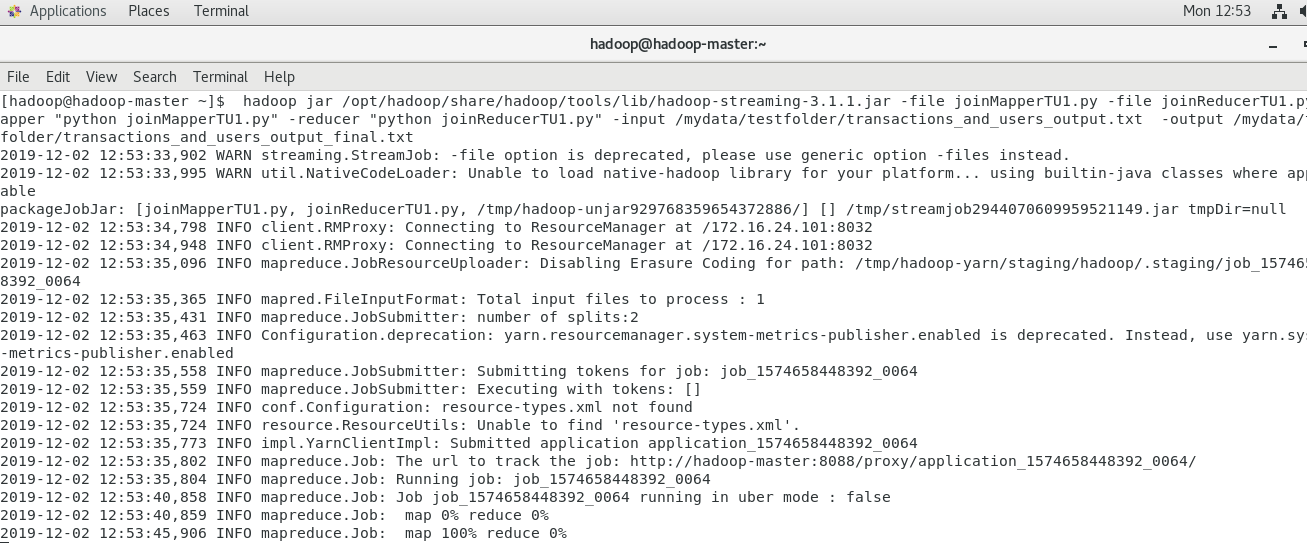
1. Write the mapper and reducer code to find, for each product, the number of locations in which that product was purchased.

See attached joinMapperTU1.py and joinReducerTU1.py for the code.

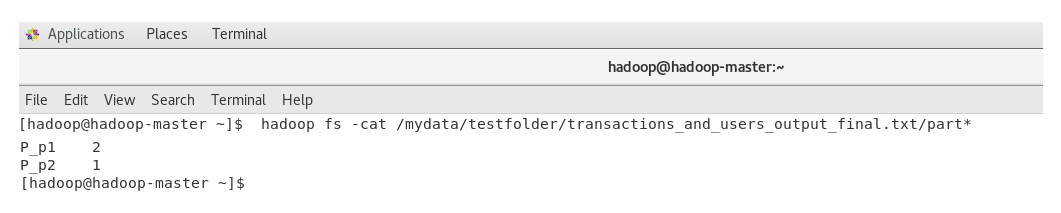




1. Run map reduce using streaming
   1. hadoop jar /opt/hadoop/share/hadoop/tools/lib/hadoop-streaming-3.1.1.jar -file joinMapperTU1.py -file joinReducerTU1.py -mapper "python joinMapperTU1.py" -reducer "python joinReducerTU1.py" -input /mydata/testfolder/transactions\_and\_users\_output.txt -output /mydata/testfolder/transactions\_and\_users\_output\_final.txt



1. Check the output
   1. hadoop fs -cat /mydata/testfolder/transactions\_and\_users\_output\_final.txt/part\*



1. Outputs/Results:

Students should be able to use Hadoop setup for

* Writing the complex MapReduce involving multiple input files
* Executing the MapReduce code on Hadoop cluster
* Observing the outcomes of MapReduce codes

1. Observations:

Students carefully show observe for

* The use case for MapReduce programming paradigm
* Ways to convert the traditional code into MapReduce program
* Ways to execute MapReduce program

1. References:
2. [Running MapReduce with multiple inputs](https://github.com/rathboma/hadoop-framework-examples/blob/master/streaming-python/ut_join/README.md)
3. [Left Outer Join with MapReduce](https://www.oreilly.com/library/view/data-algorithms/9781491906170/ch04.html)
4. [Hadoop Streaming](https://hadoop.apache.org/docs/r1.2.1/streaming.html)
5. [Hadoop Python Example](https://blog.matthewrathbone.com/2013/11/17/python-map-reduce-on-hadoop-a-beginners-tutorial.html)