Computer Science & Information Systems

**Big Data Systems – Hadoop Lab Sheet 4**

**Understanding Hadoop Streaming**

1. Objective:
   1. Introduce students to the hadoop streaming library (the mechanism which allows us to run non-jvm code on hadoop)
   2. Teach students how to write a simple map reduce pipeline in Python (single 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.

**Hadoop Streaming Intro**

Hadoop Streaming is actually just a java library that implements these things, but instead of actually doing anything, it pipes data to scripts. By doing so, it provides an API for other languages:

* read from STDIN
* write to STDOUT

Streaming has some (configurable) conventions that allow it to understand the data returned. Most importantly, it assumes that Keys and Values are separated by a "\t". This is important for the rest of the map reduce pipeline to work properly (partitioning and sorting).

It’s just like running a normal MapReduce job, except that need to provide some information about what scripts want to use. Hadoop comes with the streaming jar in its lib directory, so just need to find that to use it.

A good way to make sure job has run properly is to look at the jobtracker dashboard. We should see job in the running/completed sections, clicking on it brings up a bunch of information. The most useful data on this page is under the Map-Reduce Framework section, in particular look for stuff like:

* Map Input Records
* Map Output Records
* Reduce Output Records

**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 stadium file has the following fields:

* *Stadium (String) - The name of the stadium*
* *Capacity (Int) - The capacity of the stadium*
* *ExpandedCapacity (Int) - The expanded capacity of the stadium*
* *Location (String) - The location of the stadium*
* *PlayingSurface (String) - The type of grass, etc that the stadium has*
* *IsArtificial (Boolean) - Is the playing surface artificial*
* *Team (String) - The name of the team that plays at the stadium*
* *Opened (Int) - The year the stadium opened*
* *WeatherStation (String) - The name of the weather station closest to the stadium*
* *RoofType (Possible Values:None,Retractable,Dome) - The type of roof in the stadium*
* *Elevation - The elevation of the stadium*

Use map reduce to find the number of stadiums with artificial and natrual playing surfaces.

The pseudo-code looks like this:

*def map(line):*

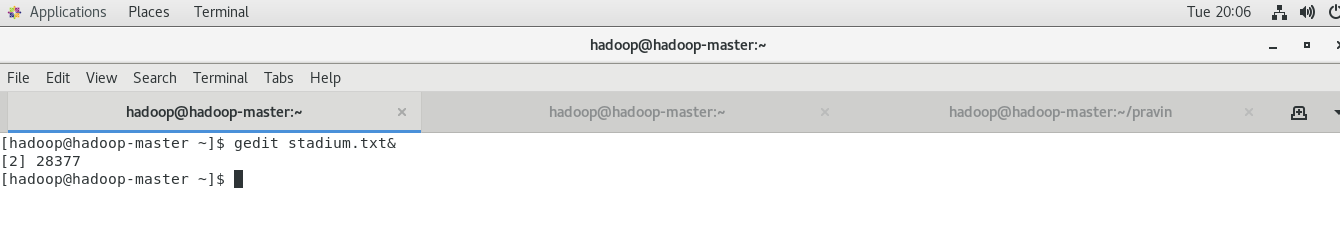
*fields = line.split(",")*

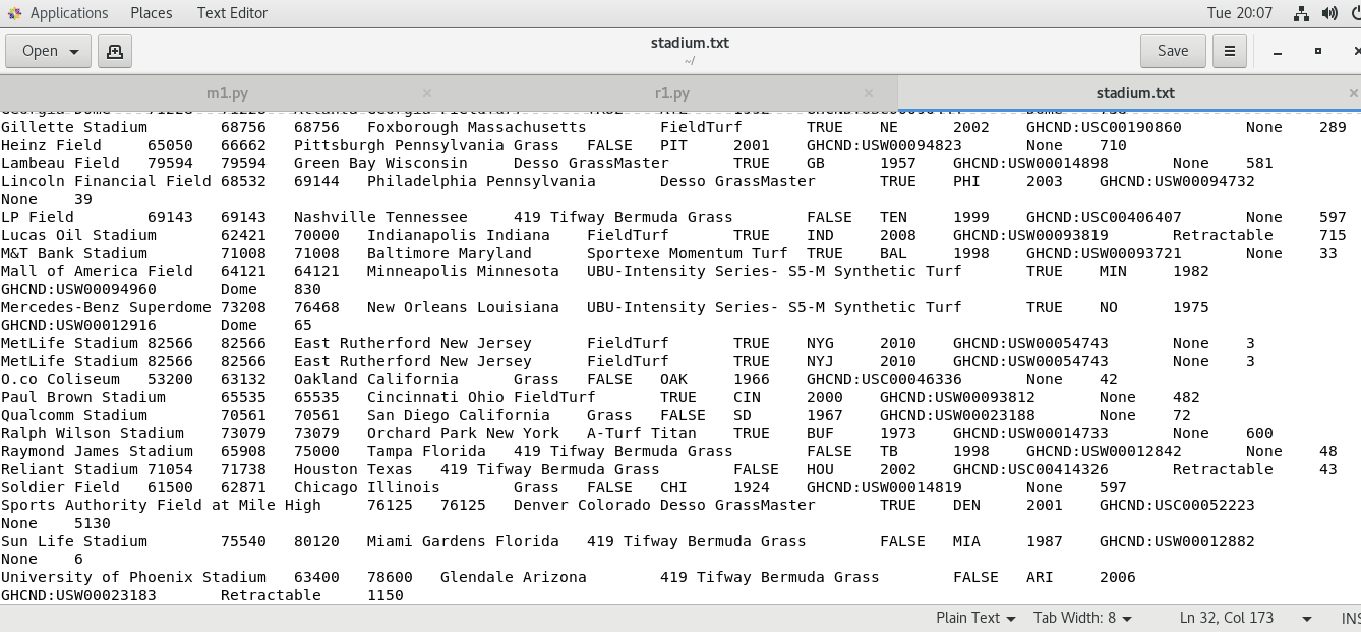
*print(fields.isArtificial, 1)*

*def reduce(isArtificial, totals):*

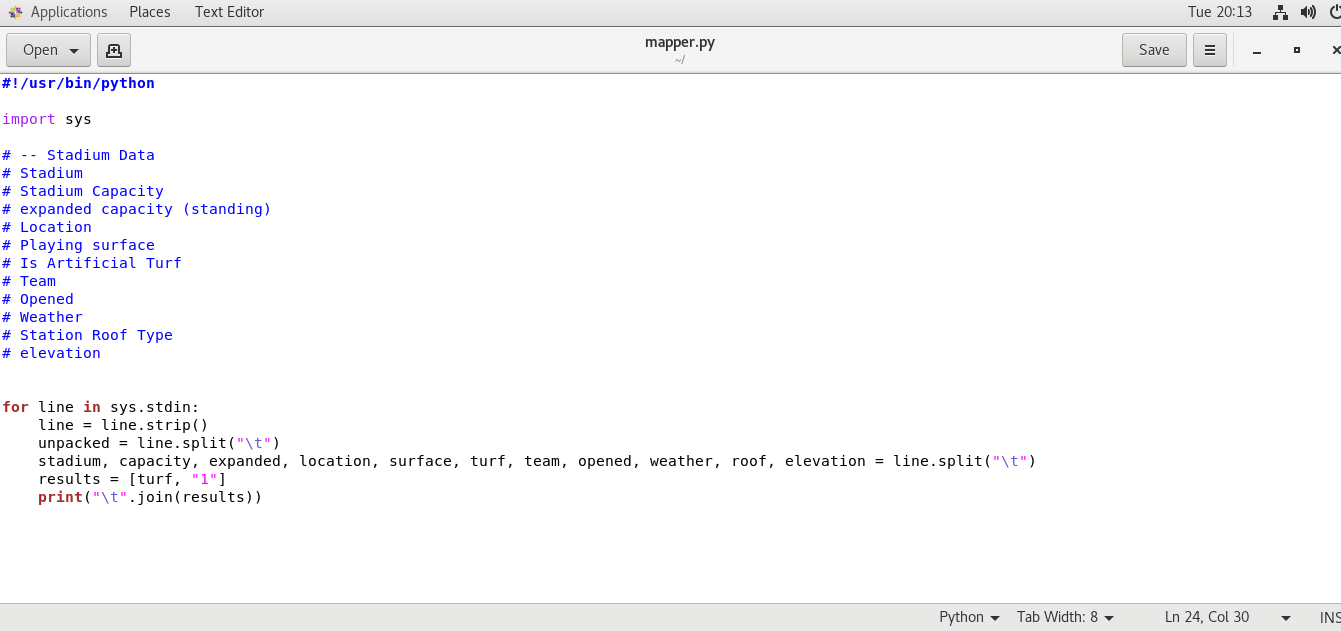
*print(isArtificial, sum(totals))*

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

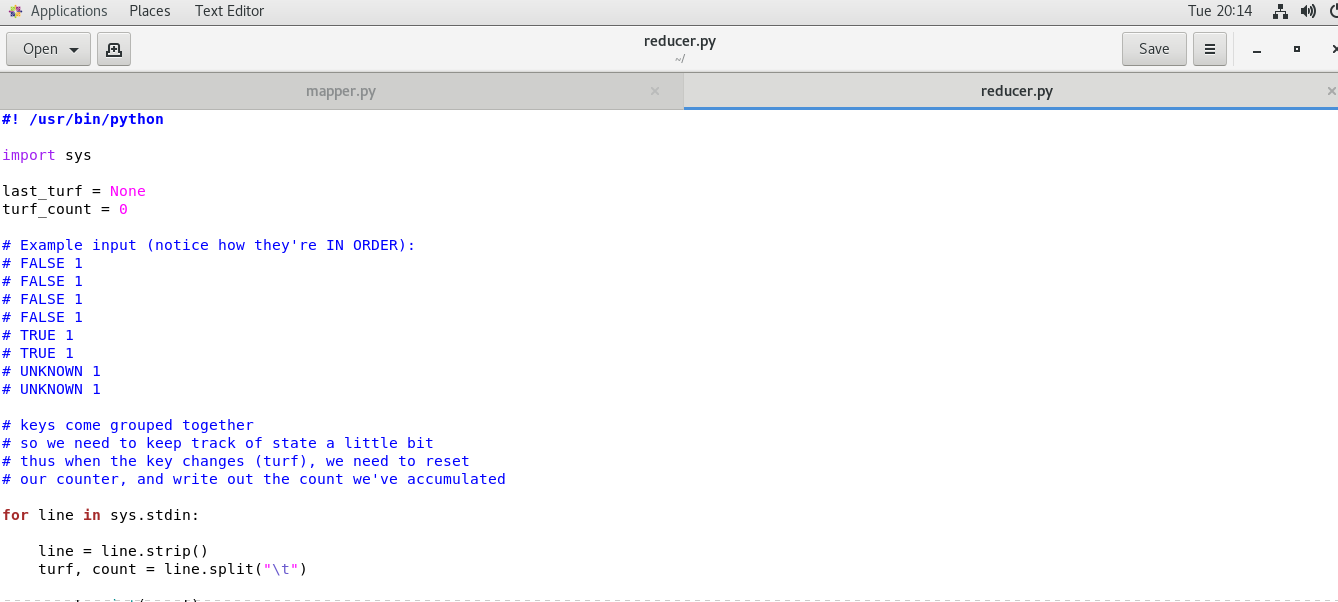




1. Write the mapper code in mapper.py. See attached file for the content.

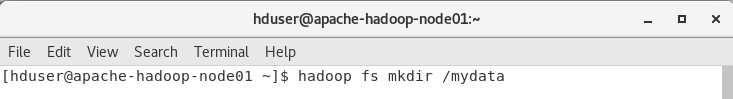


1. Write the reducer code in reducer.py. See attached file for the content.

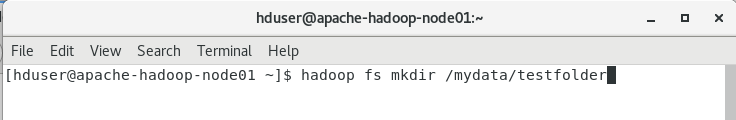


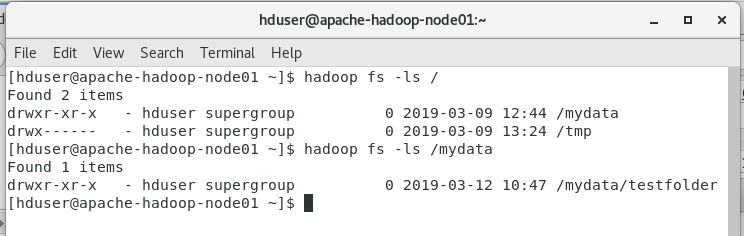


1. Create directory in Hadoop file system where you want to store your data files
   1. hadoop fs -mkdir /mydata

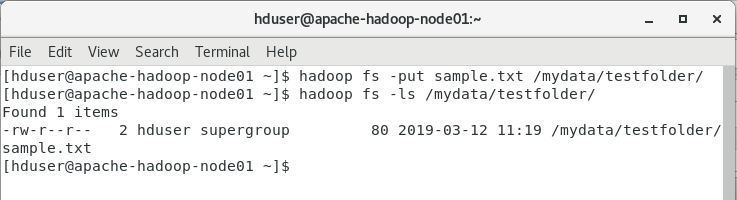


* 1. hadoop fs -mkdir /mydata/testfolder



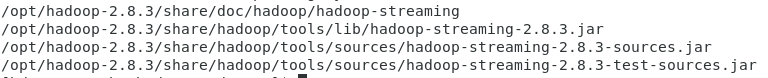


1. Move the data file i.e. sample.txt to Hadoop file system
   1. hadoop fs -put stadium.txt /mydata/testfolder/

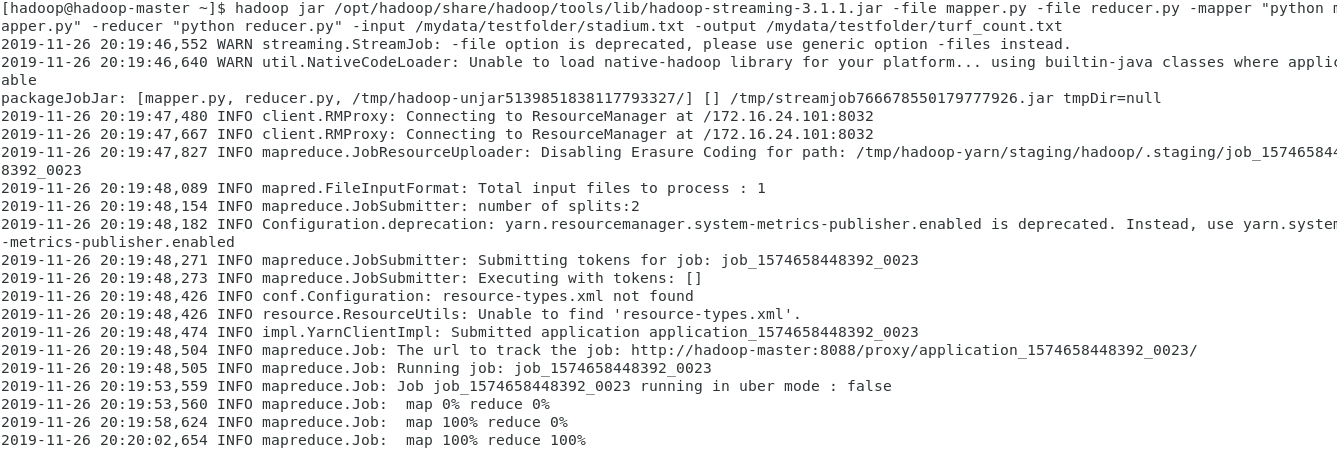


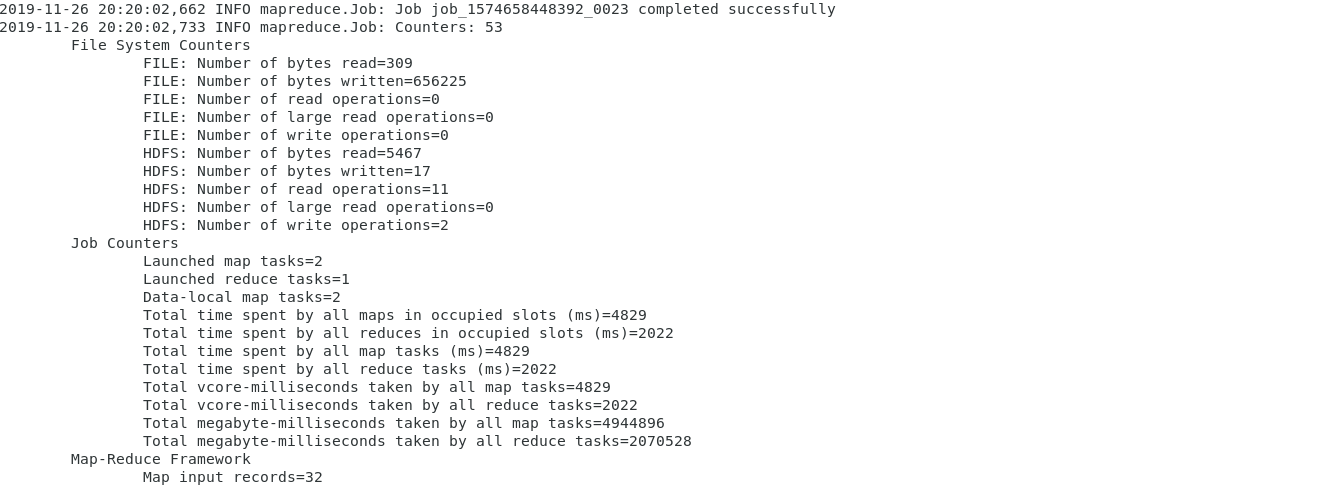
1. Find Hadoop streaming location
   1. find / -name Hadoop\*stream\*

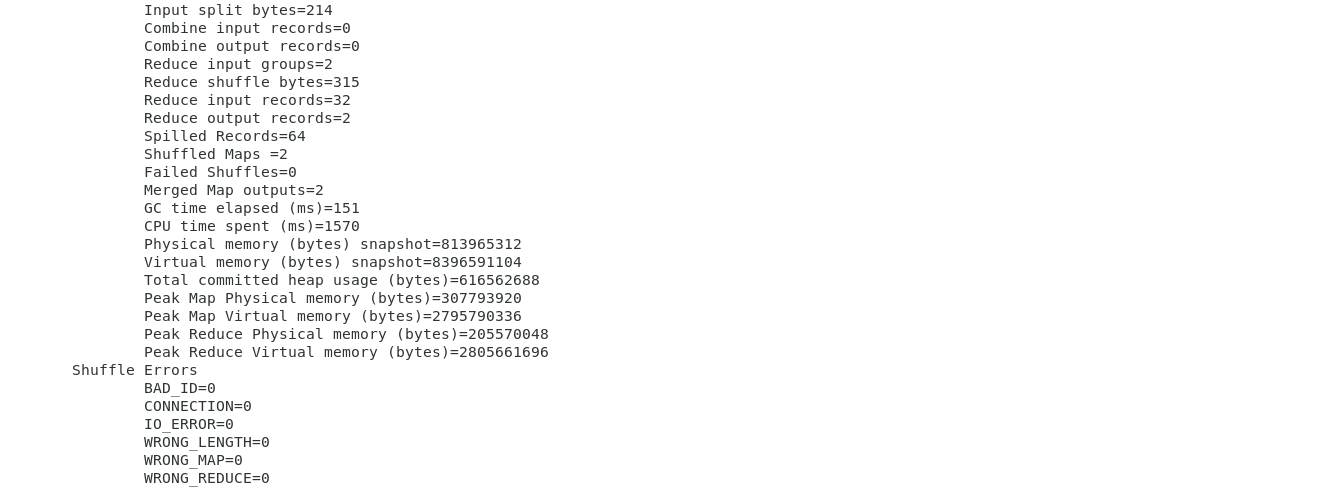


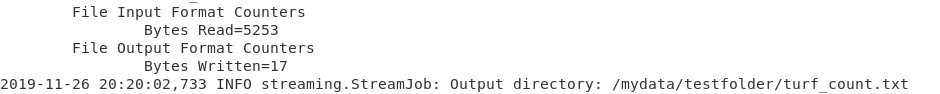


1. Run map reduce using streaming
   1. hadoop jar /opt/hadoop-2.8.3/share/hadoop/tools/lib/hadoop-streaming-2.8.3.jar -file mapper.py -file reducer.py -mapper "python mapper.py" -reducer "python reducer.py" -input /mydata/testfolder/stadium.txt -output /mydata/testfolder/turf\_count.txt

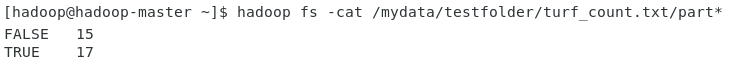








1. Check the output
   1. hadoop fs -cat /mydata/testfolder/turf\_count.txt/part\*



1. Check the Hadoop application information.
2. Outputs/Results:

Students should be able to use Hadoop setup for

* Writing the simple MapReduce programme other than Word count
* 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. [Hadoop Streaming](https://hadoop.apache.org/docs/r1.2.1/streaming.html)
3. [Hadoop Python Example](https://blog.matthewrathbone.com/2013/11/17/python-map-reduce-on-hadoop-a-beginners-tutorial.html)
4. [Example: WordCount v2.0](https://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html#Example:_WordCount_v2.0)