

Homework #4

1. What are main three categories of cloud service models?

- IaaS (Infrastructure as a Service)
- PaaS (Platform as a Service)
- SaaS (Software as a Service)
- If electricity is very expensive in Taiwan, then the IaaS business model may not be suitable for providing cloud services in Taiwan. IaaS relies heavily on the use of physical infrastructure, such as servers and data centers, which require a significant amount of electricity to operate.

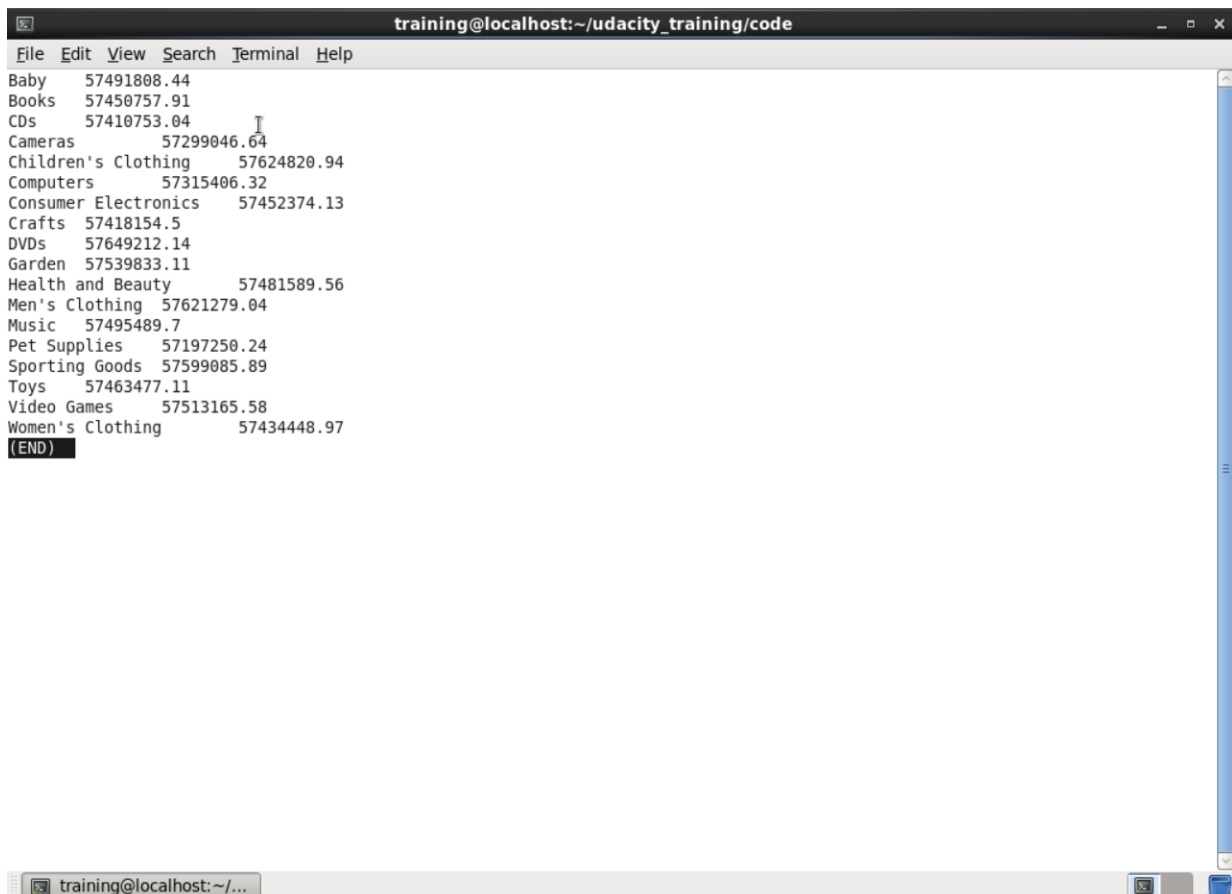
2. Discuss how your company can use Google App Engine?

- If my company plan to use Google App Engine, we can build, host and scale a machine learning application for flower image classification without worrying the underlying infrastructure, such as servers, storage and networking. Following steps may occurs:
 1. Sign up a Google Cloud account and create a new project in GCP.
 2. Take advantage of pre-trained ML APIs to our application.
 3. Build our app using a runtime environment supported by GCP such as Java, Python, Nodejs, and integrate the ML-model to the app.
 4. Use the simple interface provided by GCP to easily deploy the whole application.
- Potential cost ?
 - If our application exceeds the limit of the free trial, we will be charged for the additional resources.
- AWS vs Azure vs GCP ?
 - AWS is the oldest and most mature of these platforms. AWS has a large ecosystem of tools, integrations and communities.
 - GCP is similar to AWS but more focuses on machine learning and AI applications.
 - Azure provides similar service above all, but more focuses on integration with Microsoft products and services, such as VS and Office.

3. Virtualization

- Docker vs VM ?
 1. VM and Docker are both used for running app in isolated environment.
 2. VM is a software-based emulation of a physical computer, with its OS, memory, and storage.
 3. Docker is containerization technology that allows developers to pack app into lightweight containers, providing a consistent runtime environment for applications.
 4. VM is OS-based while Docker is application-based.
 5. VM costs more hardware resources than Docker.
- One example ppplication suitable for Docker but not suitable for VM
 1. Microservice-based application in modern web development. Docker-based technique is a best choice for microservices because it allows us to package and deploy small service in a lightweight and portable environment.

4. Exercise 1



```
training@localhost:~/udacity_training/code
File Edit View Search Terminal Help
Baby 57491808.44
Books 57450757.91
CDs 57410753.04
Cameras 57299046.64
Children's Clothing 57624820.94
Computers 57315406.32
Consumer Electronics 57452374.13
Crafts 57418154.5
DVDs 57649212.14
Garden 57539833.11
Health and Beauty 57481589.56
Men's Clothing 57621279.04
Music 57495489.7
Pet Supplies 57197250.24
Sporting Goods 57599085.89
Toys 57463477.11
Video Games 57513165.58
Women's Clothing 57434448.97
(END)
```



```
training@localhost:~/udacity_training/code
File Edit View Search Terminal Help
#!/usr/bin/python

import sys

salesTotal = 0
oldKey = None

# Loop around the data
# It will be in the format key\tval
# Where key is the store name, val is the sale amount
#
# All the sales for a particular store will be presented,
# then the key will change and we'll be dealing with the next store

for line in sys.stdin:
    data_mapped = line.strip().split("\t")
    if len(data_mapped) != 2:
        # Something has gone wrong. Skip this line.
        continue

    thisKey, thisSale = data_mapped

    if oldKey and oldKey != thisKey:
        #print oldKey, "\t", salesTotal # example
        print oldKey, "\t", max # checkpoint2
        oldKey = thisKey;
        #salesTotal = 0 # example
        max = 0 # checkpoint2

    oldKey = thisKey
    #salesTotal += float(thisSale) # example
    if (float(thisSale) > max): # checkpoint2
        max = float(thisSale)

if oldKey != None:
    print oldKey, "\t", salesTotal

~
"reducer.py" 37L, 914C 32,5 All
```

6. Exercise 3

```
training@localhost:~/udacity_training/code
File Edit View Search Terminal Help
Total sales across all the stores: 1034457953.26
Total number of sales: 4138476
(END)
```

```
training@localhost:~/udacity_training/code
File Edit View Search Terminal Help

salesTotal = 0
oldKey = None

# Loop around the data
# It will be in the format key\tval
# Where key is the store name, val is the sale amount
#
# All the sales for a particular store will be presented,
# then the key will change and we'll be dealing with the next store
salesTotalAcrossAll = 0
salesTotalNumber = 0

for line in sys.stdin:
    data_mapped = line.strip().split("\t")
    if len(data_mapped) != 2:
        # Something has gone wrong. Skip this line.
        continue
    thisKey, thisSale = data_mapped
    salesTotalAcrossAll += float(thisSale)
    salesTotalNumber += 1

    #if oldKey and oldKey != thisKey:
    #    #print oldKey, "\t", salesTotal # example
    #    print oldKey, "\t", max # checkpoint2
    #    oldKey = thisKey;
    #    #salesTotal = 0 # example
    #    max = 0 # checkpoint2

    #oldKey = thisKey
    #salesTotal += float(thisSale) # example
    #if (float(thisSale) > max): # checkpoint2
    #    max = float(thisSale)

#if oldKey != None:
#    print oldKey, "\t", salesTotal
print "Total sales across all the stores: ", salesTotalAcrossAll
print "Total number of sales: ", salesTotalNumber
"reducer.py" 41L, 1160C
```

7. Exercise 4

```
training@localhost:~/udacity_training/code
File Edit View Search Terminal Help

Albuquerque 20863.564927
Anaheim 20831.6107064
Anchorage 20759.3012781
Arlington 20966.6398122
Atlanta 20728.7602526
Aurora 20692.9646915
Austin 20808.4259878
Bakersfield 20731.7423748
Baltimore 20996.1867555
Baton Rouge 20843.9500752
Birmingham 20779.824888
Boise 20900.4489806
Boston 20852.7740002
Buffalo 20968.4153964
Chandler 20791.4898805
Charlotte 21063.0266127
Chesapeake 20722.5537571
Chicago 20825.0823641
Chula Vista 20916.254721
Cincinnati 20855.2909207
Cleveland 20596.1849171
Colorado Springs 21044.353937
Columbus 20765.6563682
Corpus Christi 20814.4879291
Dallas 20818.9960885
Denver 20973.0410926
Detroit 20789.2335302
Durham 20804.9208154
El Paso 20897.6311895
Fort Wayne 20723.1431982
Fort Worth 20880.19825
Fremont 20771.2675545
Fresno 20899.9808879
Garland 20845.5665759
Gilbert 20881.1396043
Glendale 20915.1352061
Greensboro 20689.8210913
Henderson 20660.9351591
:
```

```
training@localhost:~/udacity_training/code
File Edit View Search Terminal Help
#!/usr/bin/python

import sys

# Loop around the data
# It will be in the format key\tval
# Where key is the store name, val is the sale amount
#
# All the sales for a particular store will be presented,
# then the key will change and we'll be dealing with the next store
def calculate_var(results):
    m = sum(results) / len(results)
    var = sum((xi - m) ** 2 for xi in results) / len(results)
    return var

oldKey = None
costs = []
for line in sys.stdin:
    data_mapped = line.strip().split("\t")
    if len(data_mapped) != 2:
        # Something has gone wrong. Skip this line.
        continue
    thisKey, thisSale = data_mapped

    if oldKey and oldKey != thisKey:
        var = calculate_var(costs)
        print oldKey, "\t", var # example
        oldKey = thisKey;
        costs = [] # example

    oldKey = thisKey
    costs.append(float(thisSale))

if oldKey != None:
    var = calculate_var(costs)
    print oldKey, "\t", var

"reducer_var.py" 37L, 937C 37,1 All
```