Log Analysis with Apache Spark

Agenda

Background

In this digital world, humans leave digital trace at many points.

In banking sector, there are ATM swipes, POS swipes, internet banking. At every point, there is information generated which are referred to as logs.

Background cntd..

In companies like amazon and flipkart, daily millions of clicks happen and each click will generate a log.

With varieties of structure, high volume and velocity, traditional tools are of no use.

Now the question is how to store the logs, and how to process the logs??

Background cntd..

For storing logs big data, HDFS will be the right fit.

For processing, Map Reduce/Hive/Pig can be used. Why did we choose spark?

In the coming slides you will have an answer…!

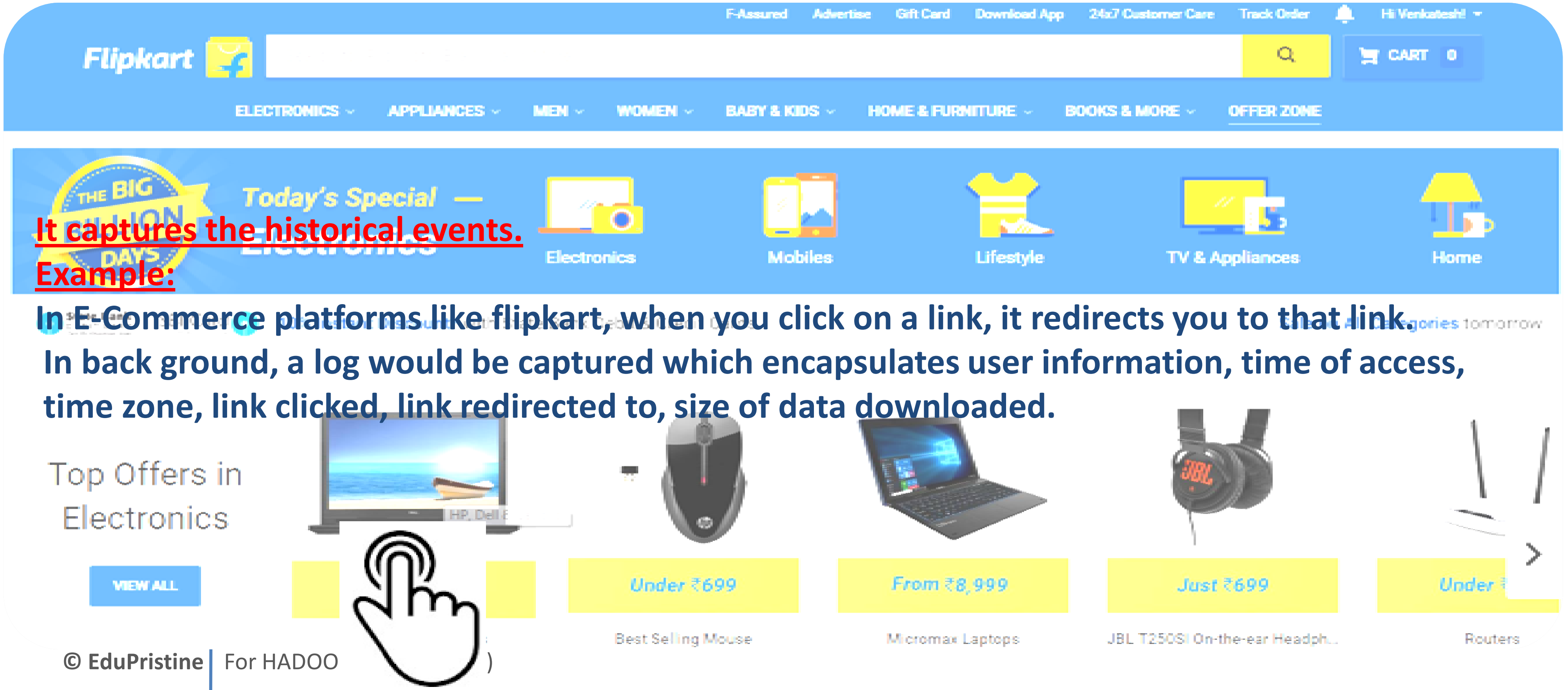
Introduction

What is log?

Log is a form a data captured while an action is performed.

It can be transaction logs, Nasa astronomy logs or anything.

Anything information related to past can be considered as logs.



It captures the historical events.

Example:

In E-Commerce platforms like flipkart, when you click on a link, it redirects you to that link. In back ground, a log would be captured which encapsulates user information, time of access, time zone, link clicked, link redirected to, size of data downloaded.

What is the format of Log?

Logs are of different formats, few are captured in XML format, few in JSON, few in raw TEXT format.

Format depends on type of logs being captured.

- User browsing logs will mostly be in text format
- Twitter logs are in JSON format
- GPS logs are in JSON format
- Machine logs will be in binary format

In this project, logs are in text format.

Does log come under unstructured?

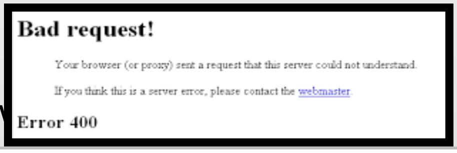
In general video, audio and general text come under unstrucutured data.
Machine logs, server logs and web logs come under semi structured data. They have a structure with no specific constraints on size of the fields captured.

So in general, logs come under semi structured data. Log format for this project:

```
127.0.0.1 - - [01/Aug/1995:00:00:01 -0400] "GET /images/launch-logo.gif HTTP/1.0" 200 1839
```

1 **2** **3** **4** **5** **6** **7** **8** **9**

1: ip address of host machine
2: user identity from remote machine
3: user identity from local logon
4: Time stamp and time zone. Here -0400 refers to time zone.
5: get or put request
6: end point. If you get on to flipkart and click on any link, you will be directed to that page. Now, any thing after flipkart.com is endpoint. Highlighted in red is end point.
l**https://www.flipkart.com/**laptops/HP
7: HTTP protocol
8: Response code. Number starting with 2 is successful response, 3 is redirection, 4 is error caused by client, 5 is error in server.
9: bytes downloaded.


Bad request!

Your browser (or proxy) sent a request that this server could not understand.

If you think this is a server error, please contact the webmaster.

Error 400

In this lab, user information is not provided. So all the fields realted to user identity w

What could be the size of logs?

Logs will be any where from giga bytes to peta bytes.

It depends on number of sensors if it is manufacturing industry, number of users if banking or ecommerce.

For banks, daily 2-4 TB of logs are logs are generated from ATM’s, internet banking, mobile banking, core banking, transactional banking, physical branches.

Why log analysis?

There is a saying, the more you understand your past, the better your future is.

The same applies everywhere.

For banking sectors, based on transaction logs, user buying patterns, customer wallet can be derived.

In E-Commerce sectors, user browsing patterns, user similarities in browsing, recommendation, personalized pages are only possible with log analysis.

In retail sectors, Market basket analysis, customer purchase patterns can be uncovered.

Who will benefit out of log analysis

On one side companies can understand more about customers.

On the other side, customers can get more offers and attention from companies.

So both customers and companies benefit out of it.

Why spark for log analysis

Spark is chosen for following reasons:

1. Interactive shell for programming
2. High speed in memory computing
3. Rich set of data analysis functions
4. Easy of programming
5. Most suited for text analytics

In spark, why pyspark?

Python is chosen for following reasons:
1. Ease of programming
2. Learning curve is high, and one can easily learn it
3. Rich set of data processing libraries

Java is not in our list. It is known for making products, and it runs the whole mobile industry. But when it comes to data analysis, we prefer python.

Why not scala. Although scala is easy compared to Java, it is still difficult than python. Not as rich as python in data libraries.

Although scala and Java are comparatively faster than python, programming and debugging efforts will be more from them.

Spark supports python very well!!

Lab Instructions

This lab will happen in batch mode.

All the code will be stored in a python file with **.py** extension, and using spark-submit , program will be submitted.

All the programs should be opened with notepad++, for better syntax highlighting.

Code will have relevant comments.

General stages of data analysis

Data understanding

Data Cleansing and preprocessing

Data Analysis

Sharing results

Data Understanding: Log format

```
127.0.0.1 - - [01/Aug/1995:00:00:01 -0400] "GET /images/launch-logo.gif HTTP/1.0" 200 1839
  1       2 3            4                   5             6                7        8   9
```

1: ip address of host machine
2: user identity from remote machine
3: user identity from local logon
4: Time stamp and time zone. Here -0400 refers to time zone.
5: get or put request
6: end point. If you get on to flipkart and click on any link, you will be directed to that page. Now, any thing after flipkart.com is endpoint. Highlighted in red is end point.
l**https://www.flipkart.com/**laptops/HP
7: HTTP protocol
8: Response code. Number starting with 2 is successful response, 3 is redirection, 4 is error caused by client, 5 is error in server.
9: bytes downloaded.

In this lab, user information is not provided. So all the fields realted to user identity will be “-”

Data Cleansing and pre processing

How to extract structure out of this semi structured data?

Log format:

```
127.0.0.1 - - [01/Aug/1995:00:00:01 -0400] "GET /images/launch-logo.gif HTTP/1.0" 200 1839
```

You might think, to split on space, but what about **[]** in timestamp and **“” ?**
So you cannot extract required fields in one single step.

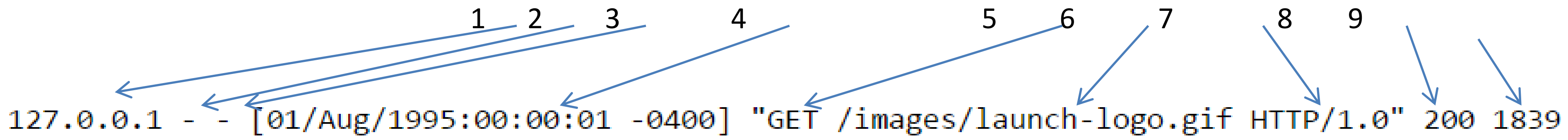
Now what to do?

Use **regular expressions** to extract the required pattern!!

Now lets start our project

# A regular expression pattern to extract fields from the log line

APACHE_ACCESS_LOG_PATTERN = '^(\S+) (\S+) (\S+) \[([\w:/]+\s[+\-]\d{4})\] "(\S+) (\S+)\s*(\S*)\s*" (\d{3}) (\S+)'



```
127.0.0.1 - - [01/Aug/1995:00:00:01 -0400] "GET /images/launch-logo.gif HTTP/1.0" 200 1839
```

^ -> refers to beginning of the string.
\S -> refers to String
+ -> refers to one or more occurrences of the preceding element.
() -> Defines a marked subexpression. A marked subexpression is also called a block or capturing group.
[ ] -> A bracket expression. Matches a single character that is contained within the brackets.
 For example, [abc] matches "a", "b", or "c".
d{4} -> refers to 4 digits
*-> Matches the preceding element zero or more times.
\s -> lower case s refers to space

1. Extract ip address. (\S+)
2. User information. (\S+)
3. User information. (\S+)
4. Time stamp and time zone. [([\w:/]+\s[+\-]\d{4})\]
5. Get or put request type. If you observe log, after double quotes, you see this value."(\S+)
6. End point followed by any number of spaces.(\S+)\s*.
7. HTTP protocol followed by spaces.(\S+)\s*.
8. Response code error. d{3}
9. Bytes downloaded . (\S+)'

Data Preprocessing

Data preprocessing will be the key step in any data analysis project.

In our case, we have unstructured log data.

Extract the structure at the first step, then rest would be easy.

If this is not done at the first step, all other programming steps will be really difficult.

Initial loader

Initial loader file has the following:

- Step1: Import all the libraries
- Step2: Regular expression to extract the structure.
- Step3: Function to extract the patterns.
- Step4: Load the log file as RDD
- Step4: Using spark **mapper** function, each log line will be passed to **parseApacheLogLine** function. Which will return the extracted fields, with schema appended.
- Step5: **parsed_logs** RDD will be created

Lab

Create a folder with name **sparkProject** in your local home directory on cluster environment.

Copy **apache.access.log.PROJECT** file in /tmp/Datasets/ path to **sparkProject** folder.

```
cp /tmp/Datasets/apache.access.log.PROJECT  ~/sparkProject
```

Create a directory on HDFS:

```
hadoop fs -mkdir sparkProject

hadoop fs -put ~/sparkProject/apache.access.log.PROJECT sparkProject

hadoop fs -ls sparkProject
```

Using WinSCP, transfer InitialLoader.py file to sparkProject folder on cluster. Run the below statement.

```
spark-submit 1_IntialLoader.py > 1_linecount
```

Upon completion, open file linesCount.

```
cat 1_linecount
```

Content size stats

Now, lets compute content size stats in the entire dataset.

Refer to file 2_ContentSizeStats.py.

Understand all the steps.

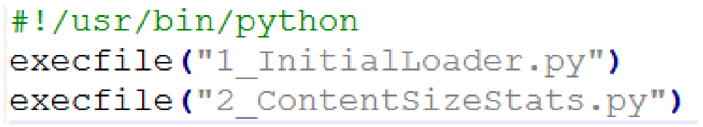
Lab: Runner program

To run this program, data should be loaded and then content size stats program should be executed.

Steps:

- 1_InitialLoader.py
- 2_ContentSizeStats.py

Create a new file, runner.py and add the below lines of code:

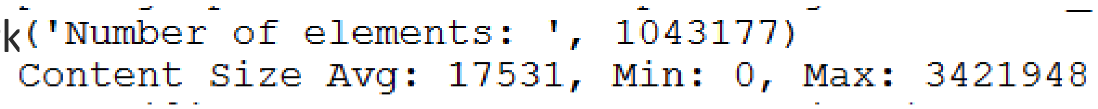


```
#!/usr/bin/python
execfile("1_InitialLoader.py")
execfile("2_ContentSizeStats.py")
```

In this, we are executing both the files one after the other. For all the programs, initialLoader file is common.

Copy file to cluster using WinSCP

spark



```
('Number of elements: ', 1043177)
Content Size Avg: 17531, Min: 0, Max: 3421948
```

cat 2_ContentSizeStats

Content size stats day wise

Now, lets compute content size stats day wise.

Refer to file 3_ContentSizeStatsDayWise.py.

Understand all the steps.

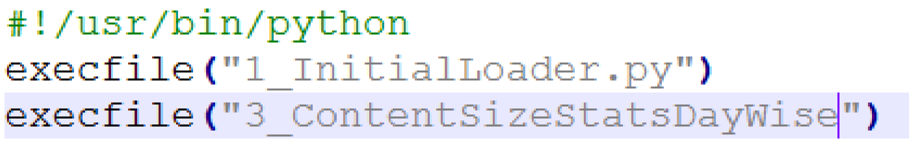
Lab: Runner program

To run this program, data should be loaded and then content size day wise stats program should be executed.

Steps:

- 1_InitialLoader.py
- 3_ContentSizeStatsDayWise

In runner.py add the below lines of code:



```
#!/usr/bin/python
execfile("1_InitialLoader.py")
execfile("3_ContentSizeStatsDayWise")
```

Copy file to cluster using WinSCP

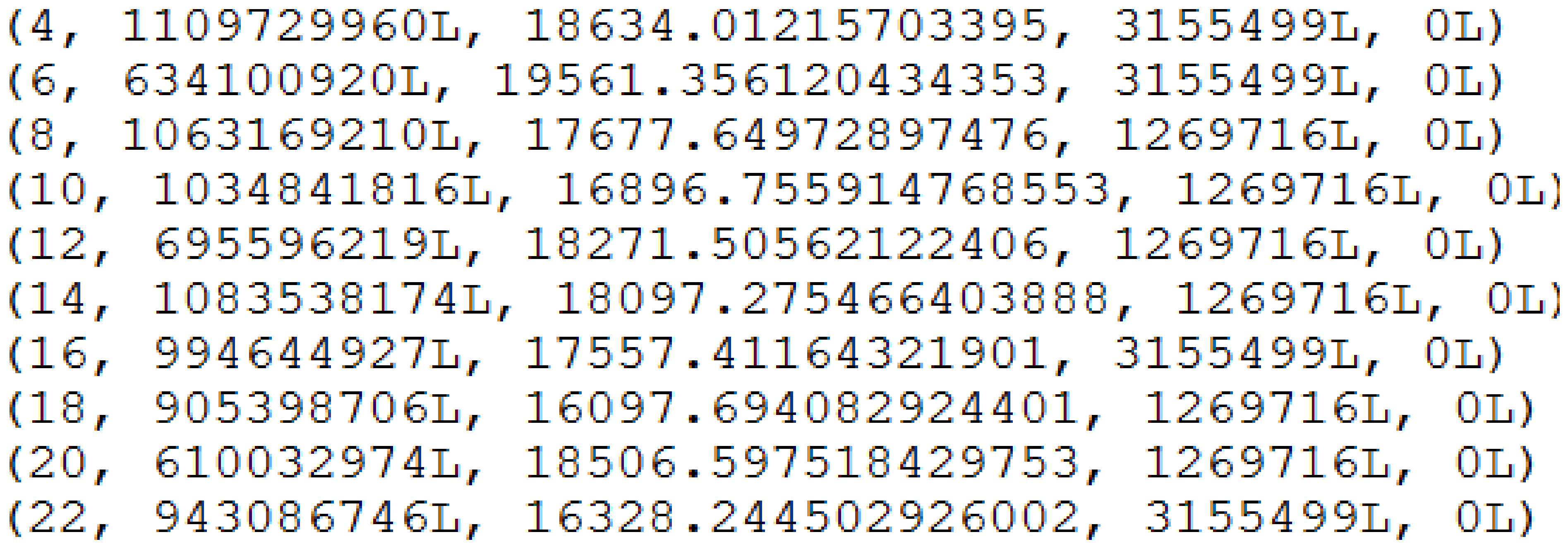
```
spark-submit Runner.py

hadoop fs -ls sparkProject/ContentSizestatsDayWise/

hadoop fs -cat sparkProject/ContentSizestatsDayWise/part-00000
```

Output

```
  Day    TotalDownload                AverageDownload                 Max               Min
```



```
(4, 1109729960L, 18634.01215703395, 3155499L, 0L)
(6, 634100920L, 19561.356120434353, 3155499L, 0L)
(8, 1063169210L, 17677.64972897476, 1269716L, 0L)
(10, 1034841816L, 16896.755914768553, 1269716L, 0L)
(12, 695596219L, 18271.50562122406, 1269716L, 0L)
(14, 1083538174L, 18097.275466403888, 1269716L, 0L)
(16, 994644927L, 17557.41164321901, 3155499L, 0L)
(18, 905398706L, 16097.694082924401, 1269716L, 0L)
(20, 610032974L, 18506.597518429753, 1269716L, 0L)
(22, 943086746L, 16328.244502926002, 3155499L, 0L)
```

Response code analysis

Response code analysis will throw insights on the performance of server.

Refer to **4_ResponseCodeAnalysis.py** file.

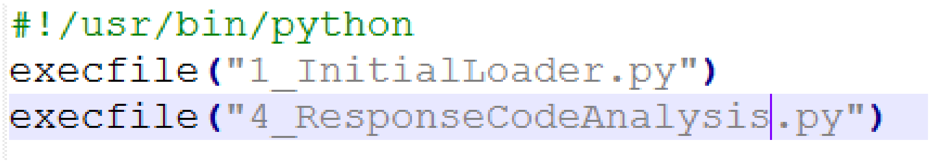
Lab: Runner program

To run this program, data should be loaded and then response code analysis program should be executed.

Steps:

- 1_InitialLoader.py
- 4_ResponseCodeAnalysis

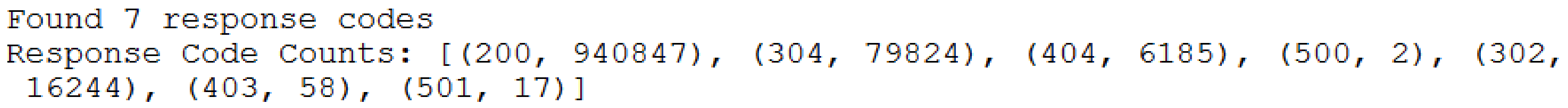
In runner.py add the below lines of code:



```
#!/usr/bin/python
execfile("1_InitialLoader.py")
execfile("4_ResponseCodeAnalysis.py")
```

Copy file to cluster using WinSCP

```
spark-submit Runner.py> 4_ResponseCodeAnalysis
```



```
Found 7 response codes
Response Code Counts: [(200, 940847), (304, 79824), (404, 6185), (500, 2), (302,
 16244), (403, 58), (501, 17)]
```

Day wise Response code analysis

Day wise response code analysis, will derive insights on daily performance.

Refer to **5_ResponseCodeAnalysisDayWise.py** file.

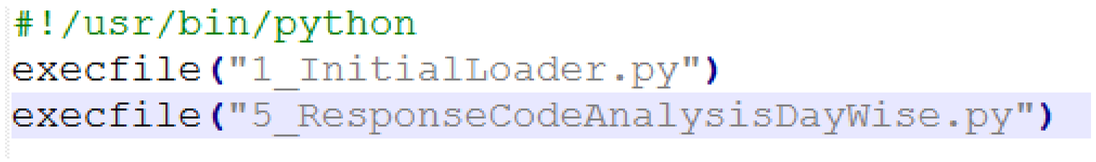
Lab: Runner program

To run this program, data should be loaded and then content size day wise stats program should be executed.

Steps:

- 1_InitialLoader.py
- 5_ResponseCodeAnalysisDayWise

In runner.py add the below lines of code:



```
#!/usr/bin/python
execfile("1_InitialLoader.py")
execfile("5_ResponseCodeAnalysisDayWise.py")
```

Copy file to cluster using WinSCP

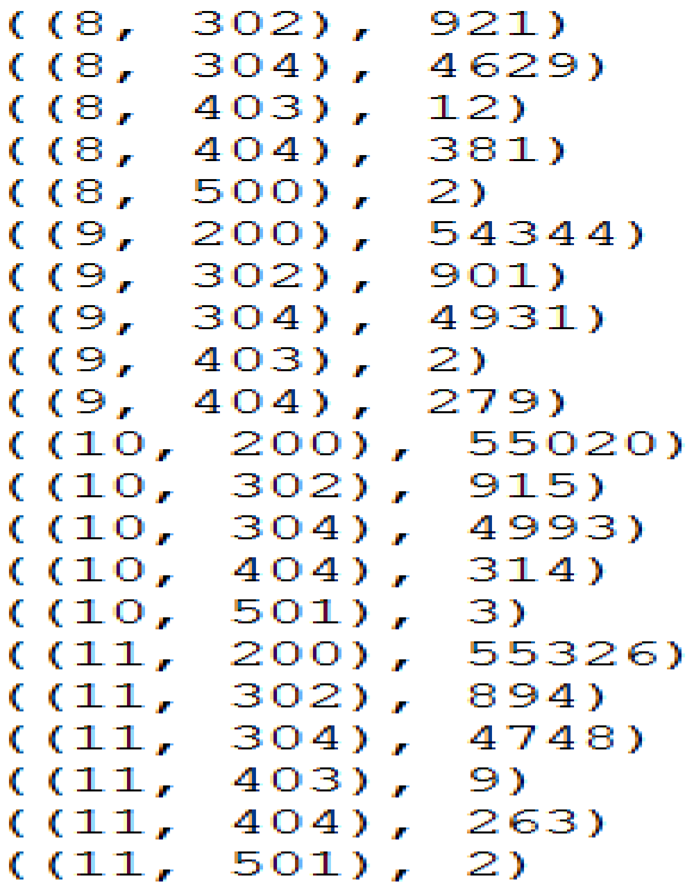
```
spark-submit Runner.py

hadoop fs -ls sparkProject/5_ResponseCodeAnalysisDayWise/

hadoop fs -cat sparkProject/5_ResponseCodeAnalysisDayWise/part-00000
```

Output Sample

Day Response   Count



```
((8,  302),  921)
((8,  304),  4629)
((8,  403),  12)
((8,  404),  381)
((8,  500),  2)
((9,  200),  54344)
((9,  302),  901)
((9,  304),  4931)
((9,  403),  2)
((9,  404),  279)
((10,  200),  55020)
((10,  302),  915)
((10,  304),  4993)
((10,  404),  314)
((10,  501),  3)
((11,  200),  55326)
((11,  302),  894)
((11,  304),  4748)
((11,  403),  9)
((11,  404),  263)
((11,  501),  2)
```

Top 20 Frequent Hosts

Get list of all users, who visited more than 20 times.

Doing this, we will get to know the frequent visitors, if it is E-Commerce, a discount can be offered probably!

Refer to **6_hostMoreThan20.py** file.

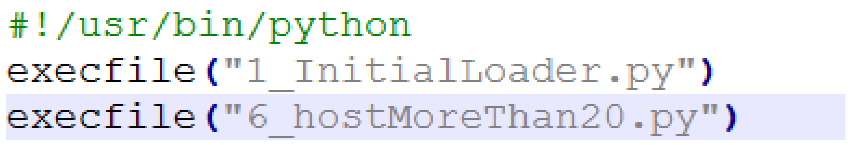
Lab: Runner program

To run this program, data should be loaded and then top 20 frequent hosts program should be executed.

Steps:

- 1_InitialLoader.py
- 6_hostMoreThan20

In runner.py add the below lines of code:



```
#!/usr/bin/python
execfile("1_InitialLoader.py")
execfile("6_hostMoreThan20.py")
```

Copy file to cluster using WinSCP

```
spark-submit Runner.py

hadoop fs -ls sparkProject/6_hostMoreThan20/

hadoop fs -tail sparkProject/6_hostMoreThan20/part-00000
```

Output Sample

Host Count

```
(u'151.104.47.106', 37)
(u'137.226.108.22', 138)
(u'ix-eve-wa1-17.ix.netcom.com', 23)
(u'204.31.249.179', 22)
(u'166.82.194.159', 26)
(u'air-divc.r01tok6.epa.gov', 34)
(u'159.153.128.27', 22)
(u'tartarus.mpr.ca', 104)
(u'192.108.184.201', 45)
(u'braubach.darmstadt.gmd.de', 22)
(u'n1028288.ksc.nasa.gov', 306)
(u'204.149.224.24', 29)
(u'149.82.197.142', 25)
(u'foley.ripco.com', 65)
(u'194.47.75.36', 46)
(u'194.47.75.34', 27)
(u'193.112.166.175', 25)
(u'142.204.22.59', 21)
(u'193.60.77.67', 21)
(u'neildk.grace.cri.nz', 23)
(u'endeavour.msfc.nasa.gov', 26)
(u'ct-pc10.massey.ac.nz', 25)
(u'pcl-a75.lib.utexas.edu', 44)
(u'a2014.dial.tip.net', 28)
```

Top 20 End Points

Get the top 20 most visited web sites.

Refer to **7_top20EndPoints.py** file.

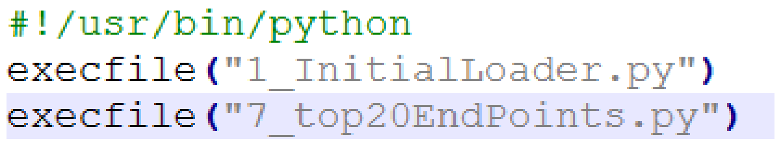
Lab: Runner program

To run this program, data should be loaded and then top 20 end points program should be executed.

Steps:

- 1_InitialLoader.py
- 7_top20EndPoints.py

In runner.py add the below lines of code:



```
#!/usr/bin/python
execfile("1_InitialLoader.py")
execfile("7_top20EndPoints.py")
```

Copy file to cluster using WinSCP

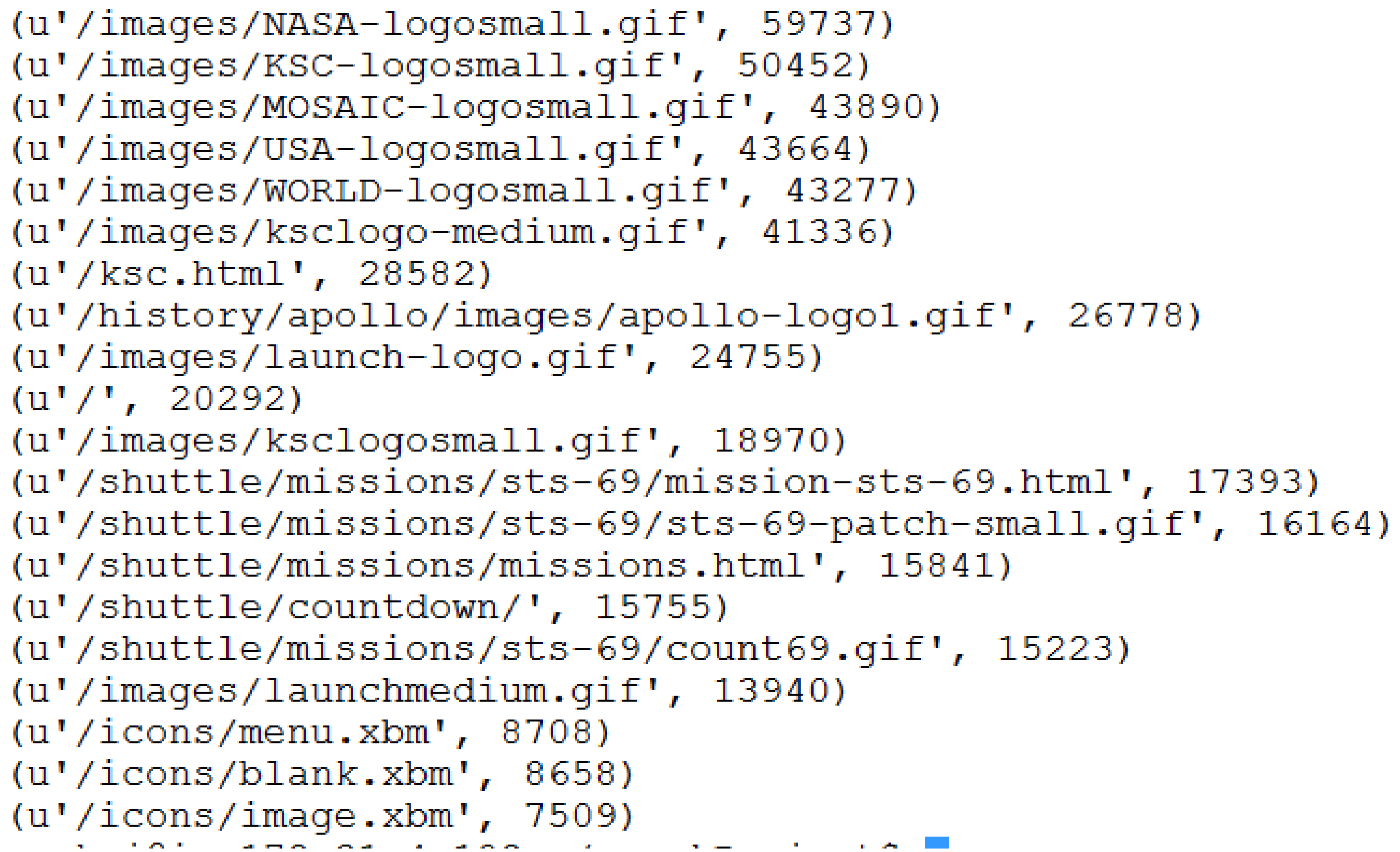
```
spark-submit Runner.py

hadoop fs -ls sparkProject/7_top20EndPoints/

hadoop fs -tail sparkProject/7_top20EndPoints/part-00000
```

Output Sample

Endpoint Count



```
(u'/images/NASA-logosmall.gif', 59737)
(u'/images/KSC-logosmall.gif', 50452)
(u'/images/MOSAIC-logosmall.gif', 43890)
(u'/images/USA-logosmall.gif', 43664)
(u'/images/WORLD-logosmall.gif', 43277)
(u'/images/ksclogo-medium.gif', 41336)
(u'/ksc.html', 28582)
(u'/history/apollo/images/apollo-logo1.gif', 26778)
(u'/images/launch-logo.gif', 24755)
(u'/', 20292)
(u'/images/ksclogosmall.gif', 18970)
(u'/shuttle/missions/sts-69/mission-sts-69.html', 17393)
(u'/shuttle/missions/sts-69/sts-69-patch-small.gif', 16164)
(u'/shuttle/missions/missions.html', 15841)
(u'/shuttle/countdown/', 15755)
(u'/shuttle/missions/sts-69/count69.gif', 15223)
(u'/images/launchmedium.gif', 13940)
(u'/icons/menu.xbm', 8708)
(u'/icons/blank.xbm', 8658)
(u'/icons/image.xbm', 7509)
```

Number of Unique Hosts

Total number of unique hosts will help us in understanding:

- Website Traffic

Refer to **8_NumberOfUniqueHosts.py** file.

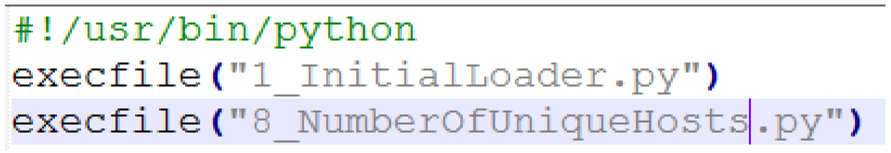
Lab: Runner program

To run this program, data should be loaded and then Number of unique hosts program should be executed.

Steps:

- 1_InitialLoader.py
- 8_NumberOfUniqueHosts

In runner.py add the below lines of code:

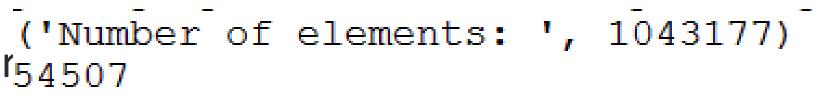


```
#!/usr/bin/python
execfile("1_InitialLoader.py")
execfile("8_NumberOfUniqueHosts.py")
```

Copy file to cluster using WinSCP

spark-submit Runner.py> 8_NumberOfUniqueHosts

cat 8_Number



```
('Number of elements: ', 1043177)
54507
```

Number of Unique Hosts Daily

Daily number of unique hosts reveal the following insights:

- Customers Growth
- Website Traffic

Refer to **9_DayWiseNumberOfDistinctUsers** file.

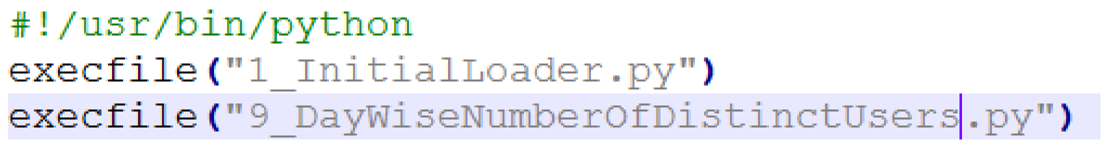
Lab: Runner program

To run this program, data should be loaded and then Day wise number of users program should be executed.

Steps:

- 1_InitialLoader.py
- 9_DayWiseNumberOfDistinctUsers.py

In runner.py add the below lines of code:



```
#!/usr/bin/python
execfile("1_InitialLoader.py")
execfile("9_DayWiseNumberOfDistinctUsers.py")
```

Copy file to cluster using WinSCP

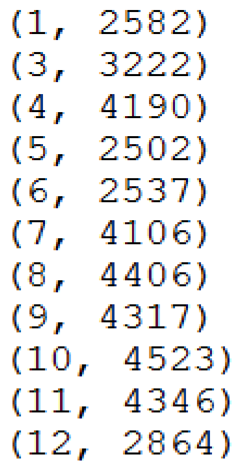
```
spark-submit Runner.py

hadoop fs -ls sparkProject/9_DayWiseNumberOfDistinctUsers/

hadoop fs -tail sparkProject/9_DayWiseNumberOfDistinctUsers/part-00000
```

Output Sample

Day Count



```
(1, 2582)
(3, 3222)
(4, 4190)
(5, 2502)
(6, 2537)
(7, 4106)
(8, 4406)
(9, 4317)
(10, 4523)
(11, 4346)
(12, 2864)
```

404 Response Code Analysis

Response code analysis will give following insights:

- Total Number of failures
- Based on which, server capacity planning can be performed

Refer to **10_ResponseCodeAnalysis** file.

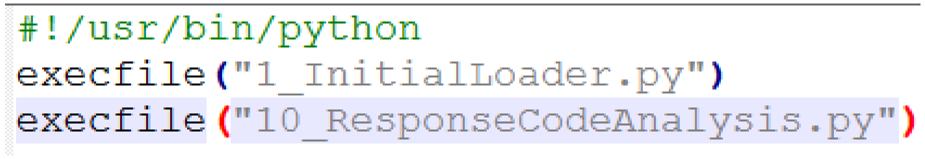
Lab: Runner program

To run this program, data should be loaded and then response code analysis program should be executed.

Steps:

- 1_InitialLoader.py
- 10_ResponseCodeAnalysis

In runner.py add the below lines of code:



```
#!/usr/bin/python
execfile("1_InitialLoader.py")
execfile("10_ResponseCodeAnalysis.py")
```

Copy file to cluster using WinSCP

spark-submit Runner.py> 10_ResponseCodeAnalysis

cat 10_ResponseCodeAnalysis

Top 20 Bad End Points

If the bad end points are identified, the following might be reason:

- Resource is not available, but the link is still there.
- Resource is available, but very large.

Finding out those loose points, and making necessary corrections will make the server robust.

Refer to **11_badEndpointsTop20** file.

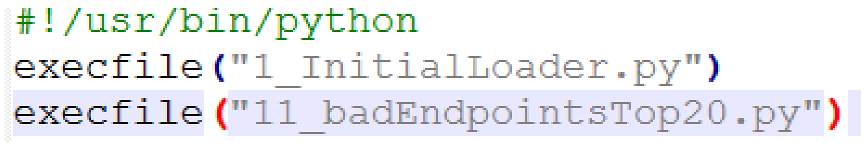
Lab: Runner program

To run this program, data should be loaded and then top 20 bad endpoints program should be executed.

Steps:

- 1_InitialLoader.py
- 11_badEndpointsTop20.py

In runner.py add the below lines of code:



```
#!/usr/bin/python
execfile("1_InitialLoader.py")
execfile("11_badEndpointsTop20.py")
```

Copy file to cluster using WinSCP

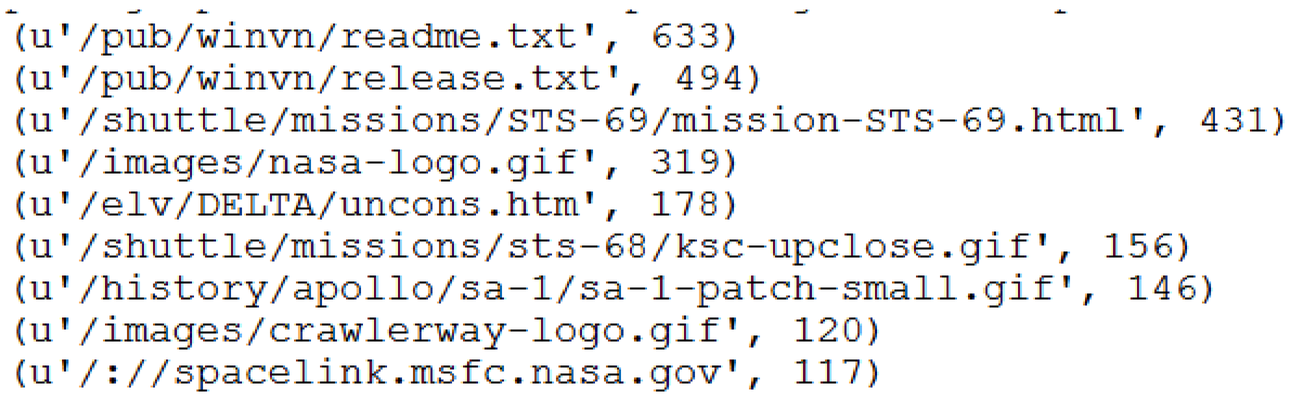
```
spark-submit Runner.py

hadoop fs -ls sparkProject/11_badEndpointsTop20/

hadoop fs -tail sparkProject/11_badEndpointsTop20/part-00000
```

Output Sample

End point   Number of 404 issues



```
(u'/pub/winvn/readme.txt', 633)
(u'/pub/winvn/release.txt', 494)
(u'/shuttle/missions/STS-69/mission-STS-69.html', 431)
(u'/images/nasa-logo.gif', 319)
(u'/elv/DELTA/uncons.htm', 178)
(u'/shuttle/missions/sts-68/ksc-upclose.gif', 156)
(u'/history/apollo/sa-1/sa-1-patch-small.gif', 146)
(u'/images/crawlerway-logo.gif', 120)
(u'/://spacelink.msfc.nasa.gov', 117)
```

Top 20 hosts with most number of bad points

If users frequently see 404 error, then the chances of loosing the customers is very high.

To understand, pick the users with most number of 404 issues, and further drill down can be performed to understand the reason.

Refer to **12_top20HostsWithErrors** file.

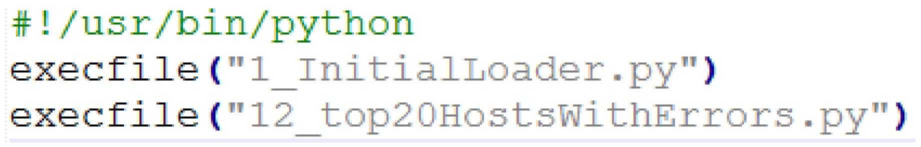
Lab: Runner program

To run this program, data should be loaded and then top 20 hosts with errors program should be executed.

Steps:

- 1_InitialLoader.py
- 12_top20HostsWithErrors.py

In runner.py add the below lines of code:



```
#!/usr/bin/python
execfile("1_InitialLoader.py")
execfile("12_top20HostsWithErrors.py")
```

Copy file to cluster using WinSCP

```
spark-submit Runner.py

hadoop fs -ls sparkProject/12_top20HostsWithErrors/

hadoop fs -tail sparkProject/12_top20HostsWithErrors/part-00000
```

Output Sample

Host 404 error code count

```
(u'maz3.maz.net', 39)
(u'piweba3y.prodigy.com', 39)
(u'gate.barr.com', 38)
(u'm38-370-9.mit.edu', 37)
(u'ts8-1.westwood.ts.ucla.edu', 37)
(u'nexus.mlckew.edu.au', 37)
(u'204.62.245.32', 33)
(u'163.206.104.34', 27)
(u'spica.sci.isas.ac.jp', 27)
(u'www-d4.proxy.aol.com', 26)
(u'www-c4.proxy.aol.com', 25)
(u'203.13.168.24', 25)
```

Top 20 hosts with most number of bad points

If users frequently see 404 error, then the chances of loosing the customers is very high.

To understand, pick the users with most number of 404 issues, and further drill down can be performed to understand the reason.

Refer to **13_dayWise404ResponseCodeCount** file.

Lab: Runner program

To run this program, data should be loaded and then top 20 hosts with errors program should be executed.

Steps:

- 1_InitialLoader.py
- 13_dayWise404ResponseCodeCount.py

In runner.py add the below lines of code:

Copy file to cluster using WinSCP

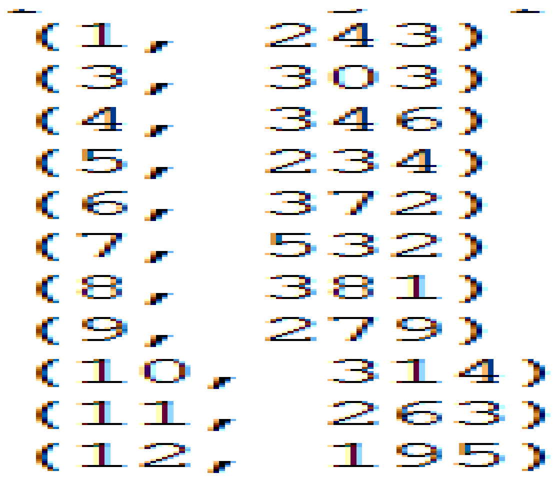
```
spark-submit Runner.py

hadoop fs -ls sparkProject/13_dayWise404ResponseCodeCount/

hadoop fs -tail sparkProject/13_dayWise404ResponseCodeCount/part-00000
```

Output Sample

Day 404 error code count



```
(1,  243)
(3,  303)
(4,  346)
(5,  234)
(6,  372)
(7,  532)
(8,  381)
(9,  279)
(10,  314)
(11,  263)
(12,  195)
```

Hour wise 404 response code count

Hour wise analysis of 404 response code, will reveal the trend and performance on the server. Based on that, Server maintenance can be planned.

Refer to **14_HourlyResponseCodeCount** file.

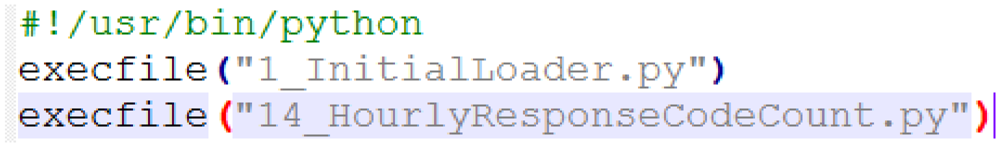
Lab: Runner program

To run this program, data should be loaded and then content size day wise stats program should be executed.

Steps:

- 1_InitialLoader.py
- 14_HourlyResponseCodeCount.py

In runner.py and add the below lines of code:



```
#!/usr/bin/python
execfile("1_InitialLoader.py")
execfile("14_HourlyResponseCodeCount.py")
```

Copy file to cluster using WinSCP

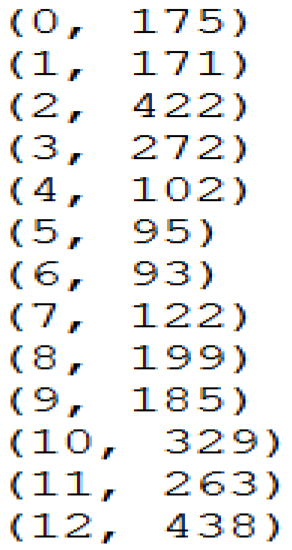
```
spark-submit Runner.py

hadoop fs -ls sparkProject/14_HourlyResponseCodeCount/

hadoop fs -tail sparkProject/14_HourlyResponseCodeCount/part-00000
```

Output Sample

Hour, errorCodeCount



```
(0, 175)
(1, 171)
(2, 422)
(3, 272)
(4, 102)
(5, 95)
(6, 93)
(7, 122)
(8, 199)
(9, 185)
(10, 329)
(11, 263)
(12, 438)
```

Top 20 hosts per day

Identifying the top users per day, and providing them relevant offers if it is a E-Commerce sector, offering different products if it is banking sector will make customers happy. Have an eye on them!!.

Refer to **15_Top20FrequentHostsPerDay** file.

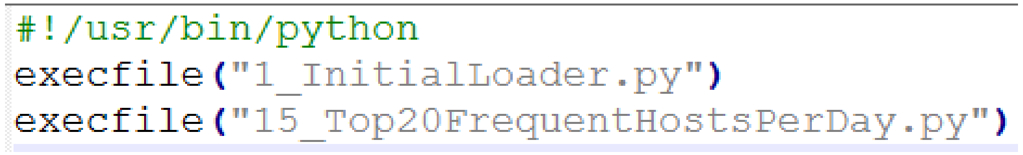
Lab: Runner program

To run this program, data should be loaded and then top 20 frequent hosts per day program should be executed.

Steps:

- 1_InitialLoader.py
- 15_Top20FrequentHostsPerDay.py

In runner.py and add the below lines of code:



```
#!/usr/bin/python
execfile("1_InitialLoader.py")
execfile("15_Top20FrequentHostsPerDay.py")
```

Copy file to cluster using WinSCP

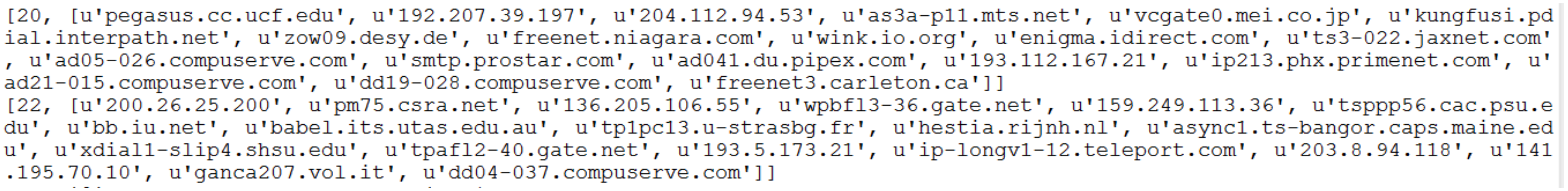
```
spark-submit Runner.py

hadoop fs -ls sparkProject/15_Top20FrequentHostsPerDay/

hadoop fs -tail sparkProject/15_Top20FrequentHostsPerDay/part-00000
```

Output Sample

Day, and top 20 hosts in each day



```
[20, [u'pegasus.cc.ucf.edu', u'192.207.39.197', u'204.112.94.53', u'as3a-p11.mts.net', u'vcgate0.mei.co.jp', u'kungfusi.pd
ial.interpath.net', u'zow09.desy.de', u'freenet.niagara.com', u'wink.io.org', u'enigma.idirect.com', u'ts3-022.jaxnet.com'
, u'ad05-026.compuserve.com', u'smtp.prostar.com', u'ad041.du.pipex.com', u'193.112.167.21', u'ip213.phx.primenet.com', u'
ad21-015.compuserve.com', u'dd19-028.compuserve.com', u'freenet3.carleton.ca']]
[22, [u'200.26.25.200', u'pm75.csra.net', u'136.205.106.55', u'wpbfl3-36.gate.net', u'159.249.113.36', u'tsppp56.cac.psu.e
du', u'bb.iu.net', u'babel.its.utas.edu.au', u'tp1pc13.u-strasbg.fr', u'hestia.rijnh.nl', u'async1.ts-bangor.caps.maine.ed
u', u'xdial1-slip4.shsu.edu', u'tpafl2-40.gate.net', u'193.5.173.21', u'ip-longv1-12.teleport.com', u'203.8.94.118', u'141
.195.70.10', u'ganca207.vol.it', u'dd04-037.compuserve.com']]
```

SparkSQL

For data analysts to work, spark enabled querying on RDD’s.

Following are things you should do to query RDD’s:

- Create an RDD with tuples
- Convert it to data frame
- Register it as temporary table
- Start querying.

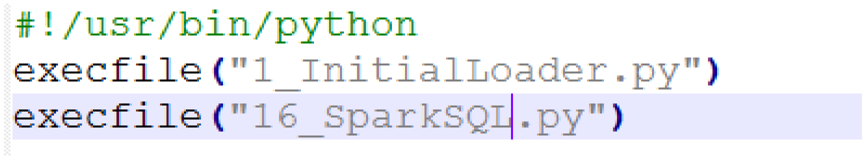
Lab: Runner program

To run this program, data should be loaded and then spark SQL program should be executed.

Steps:

- 1_InitialLoader.py
- 16_SparkSQL.py

In runner.py and add the below lines of code:



```
#!/usr/bin/python
execfile("1_InitialLoader.py")
execfile("16_SparkSQL.py")
```

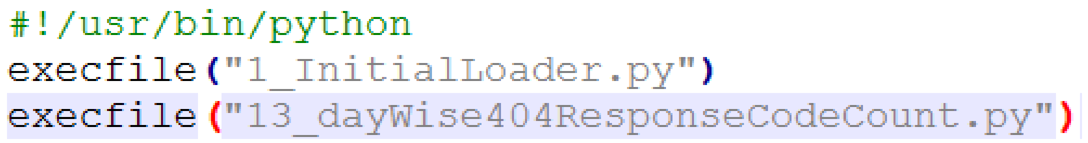
Copy file to cluster using WinSCP

```
spark-submit Runner.py > sparkSQL

hadoop fs -ls sparkProject/16_SparkSQL_ContentStatsPerHost

hadoop fs -tail sparkProject/16_SparkSQL_ContentStatsPerHost/part-00000
```

Output Sample



```
#!/usr/bin/python
execfile("1_InitialLoader.py")
execfile("13_dayWise404ResponseCodeCount.py")
```

Thank you!

Contact:

EduPristine

702, Raaj Chambers, Old Nagardas Road, Andheri (E), Mumbai-400 069. INDIA

www.edupristine.com

Ph. +91 22 3215 6191