

**Business Information Systems**

**Individual Assignment Submission Form**

**Notes to candidates:**

1. Assignments will not be accepted without a signed copy of this form.
2. This form must be stapled to any additional pages submitted.
3. Please refer to your course lecturer for full assignment submission details.
4. Any computer disc or CD submitted with the assignment must also be signed.
5. Unreadable, virus infected or blank discs/memory sticks will score zero.

|  |  |
| --- | --- |
| **Student Details** | |
| **Surname :** | Boovendran |
| **First Name :** | Jayakarthi |
| **Student ID :** | 19230487 |
| **Email :** | J.Boovendran1@nuigalway.ie |
| **Contact Phone No :** | +353-894970170 |
| **Lecturer :** | Dr. Anatoli Nachev |
| **Course :** | MS5106 – Data Science & Big Data Analytics |
| **Assignment :** | Assignment 1 - Big Data Analytics: Log Management |

I hereby declare that the work submitted is entirely my own, and that ideas or extracts taken from other sources are properly acknowledged and referenced. Furthermore, I acknowledge that the penalty for plagiarism may include suspension from examination.

Signed: ***Jayakarthi Boovendran*** Date: 09 March 2020

**Table of Contents**

1. Introduction…………………………………………………………………….………...3

2. Web Server Logs…………………………………………………………………………3

3. Log Analysis using Hive Scripts…………………………………………………………3

4. Web Server Log Analysis using Modified Hive Scripts…………………………………6

4.1 Analysis of Status codes………………………………………………………...7

4.2 Total Requests based on Operating System…………………………………….8

4.3 Total Requests based on Location………………………………………………9

4.4 Total requests per Internet Browser…………………………………………….10

4.5 Analysis of Transferred Data……………………………………………………11

4.6 Analysis of Server IPs and Status Code…………………………………………12

4.7 Exploratory Data Analysis of Log Data…………………………………………13

5. Importance of Big Data Analytics in Log Management…………………………………14

6. Conclusion………………………………………………………………………………. 16

7. References………………………………………………………………………………..16

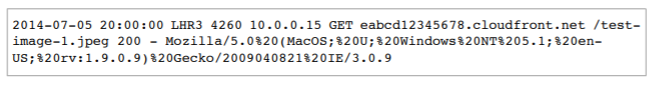
1. **Introduction**

Analysis of log data has been an integral part of businesses long before the use of big data analytics. We live in an internet era where every business makes use of internet-based technologies for their regular functioning, because of which, there is a drastic increase in the amounts of web server logs that have generated over the years. One of the significant consequences of these huge volumes of log data is storing and maintaining them. Additionally, it has become difficult for log analysts to analyze these high volumes of data manually. The bottlenecks, as mentioned above, have made many organizations to ignore the importance of log management. However, the evolution of big data technologies has paved the way for organizations to effectively manage the log data and derive valuable insights from them. The document covers the analysis of Eurostar's log data using Hive scripts, exploratory data analysis (visualization) using Tableau, and finally, a suggestion to the organization on the derived business insights.

1. **Web Server Logs**

A server log file is a file that keeps track of the server activities. These log files are created and maintained automatically by the web servers. In general, a log file incorporates information on the service request made, date and time, location, IP address, alongside user’s browser details (Server log, 2018). In the exercise, will be analysis the log data of Eurostar. Since Eurostar provides most of its products and services online, a significant volume of log files is created. Furthermore, these log files are accessible from the AWS S3 location: <s3://us-west-2.elasticmapreduce.samples>.

Each log entry is quasi-structured as given below,



Besides, the log entry holds information such as date, time, endpoint location, bytes served, IP address, the request method, host details, URI, OS, browser and browser version.

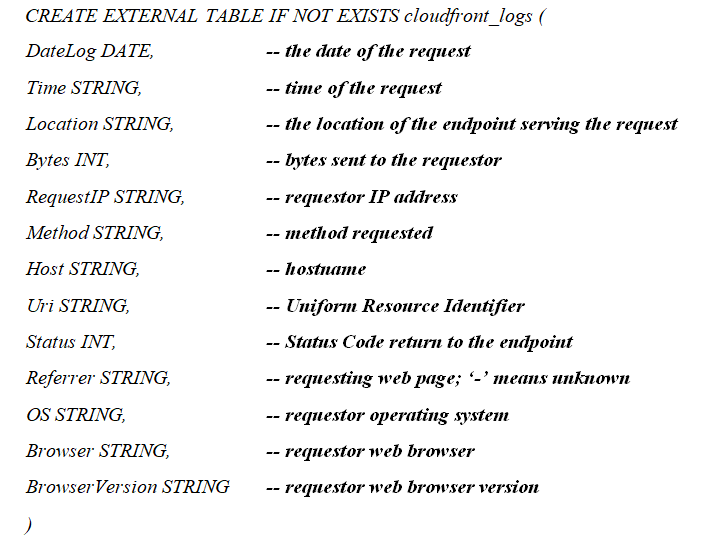
1. **Log Analysis using Hive Scripts**

The primary use of Hive Scripts is to execute a set of HiveQL commands collectively. For extracting information from the log files, it is essential to represent the unstructured log data into a correctly formatted structure that can be represented as a database table. For this purpose, hive script makes use of Serializer/deserialize (SerDe), where, it applies regular expression evaluation to parse and transform the log data into structured tables. Hive is primarily used in data mining applications to provide data summarization and analysis. Typically, Hive script uses HiveQL which supports Structured Query Language (SQL) for querying and structured data analysis.

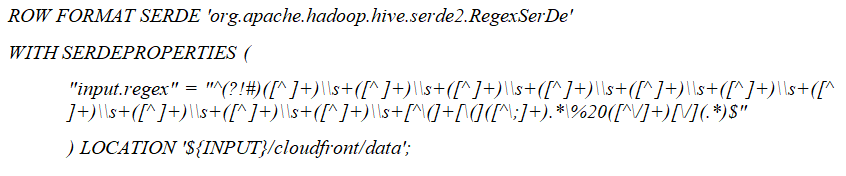
The given Hive script to analyze the logs is stored in S3 using Hive, and it calculates the total number of requests per operating system over a specified period. The script comprises of two queries. The first query creates an external table “*cloudfront\_logs*” and stores the extracted log data in it, where, the second query computes the counts of requests received from the different operating system between the dates ‘2014-07-05’ and ‘2014-08-05’. A detailed explanation of the Hive script is included further.

***Table Creation***

*The command "CREATE EXTERNAL TABLE IF NOT EXISTS cloudfront\_logs" will check if there is a table already created with the name 'cloudfront\_logs'. If the table is already present, the script directly loads the table; Else, creates a new external table.*

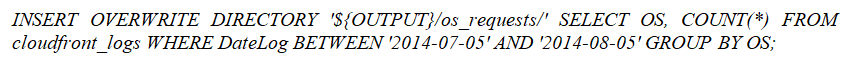


*The command “ROW FORMAT SERDE” uses regex operation to extract and parse data from the log data from the specified input location and store them to their corresponding columns in the ‘cloudfront\_logs’ table. ${INPUT}: References the input path given while creating the AWS cluster. In this case, the input path is* [*s3://us-west-2.elasticmapreduce.samples*](s3://us-west-2.elasticmapreduce.samples)



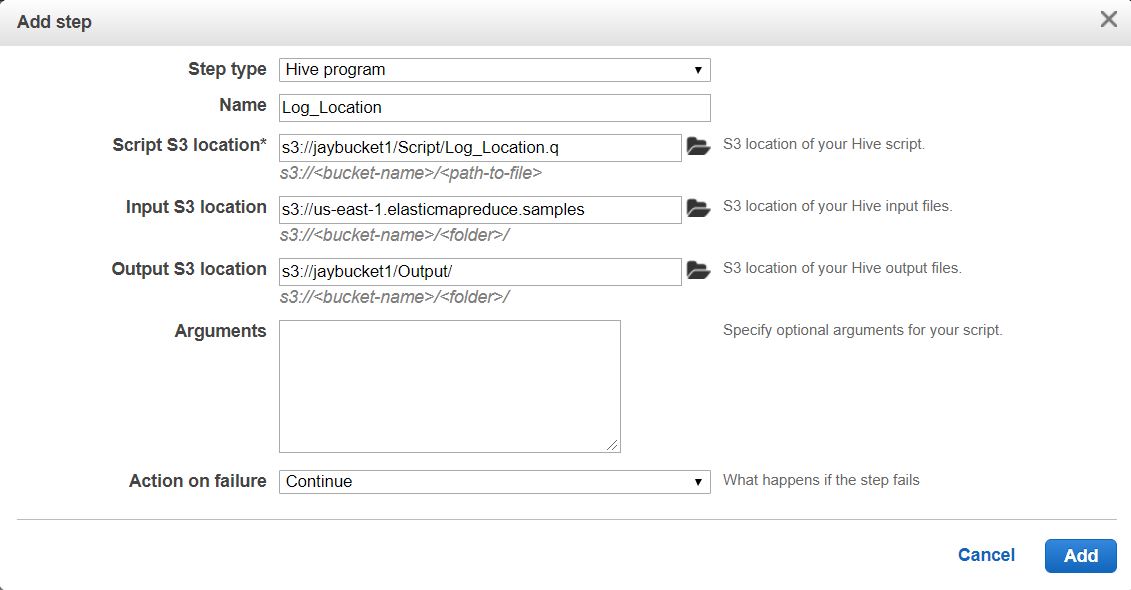
***Querying***

*Query to compute the number of requests received from various operating system between the time frame of ‘2014-07-05’ and ‘2014-08-05’. ${OUTPUT} references the S3 output folder path given while adding step for AWS cluster*



***Adding Steps to AWS Cluster***

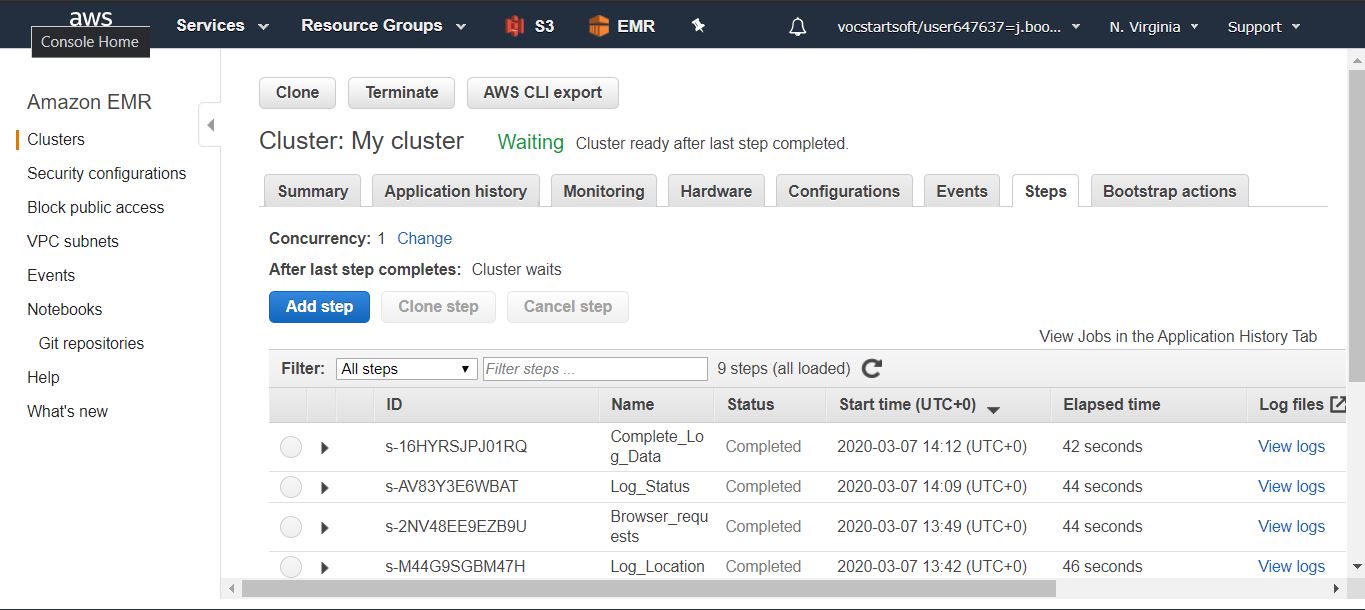
*Once created, the script is added to the AWS Cluster. In the process of adding step to the cluster, the Hive script location, input S3 path and Output S3 bucket location is provided.*



*Fig 1: Add step screen with Hive script location, input S3 path and output location*

***Executing Hive Scripts***

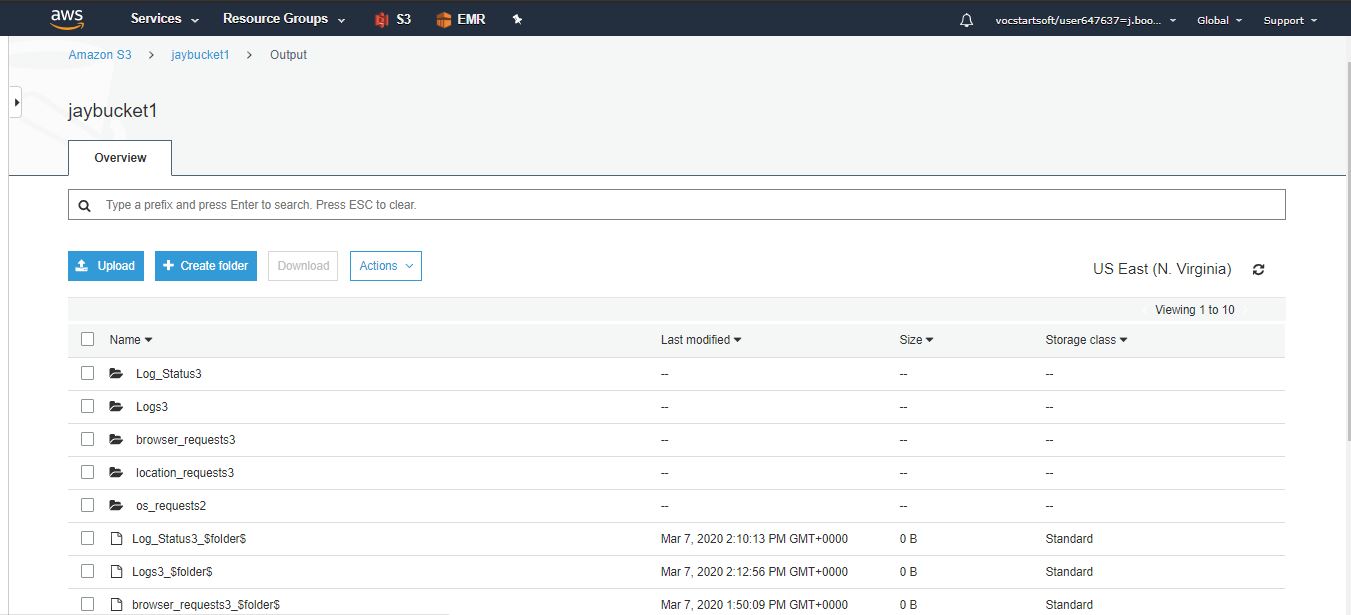
*Once the step is added the script automatically executes in the cluster. Below is the screenshot showing successful execution of different modified hive scripts.*



*Fig 2: Cluster showing successfully completed Hive scripts*

***Output of Scripts***

*After successfully running the scripts, an output folder is automatically created at the specified S3 output bucket location with the output file placed in it.*

**

*Fig 3: Output S3 bucket containing dedicated output folders for every successfully run script*

1. **Web Server Log Analysis using Modified Scripts**

Eurostar is an organization that relies entirely on its online services for most of its product and services. Although Web server logs have valuable information, the decision-makers must perform statistical analysis of the logs to obtain value out of them. For instance, a web server log can provide information on each requestor, information accessed; also, they give information on errors in the request. Since Eurostar is an online firm, it is essential to understand the request traffic, requirements of the clients, issues and the type of product/service that is frequently searched. Below are few Hive script examples, executed to extract valuable information from the log data.

4.1 Analysis of Status codes

4.2 Total Requests based on Operating System

4.3 Total Requests based on Location

4.4 Total requests per Internet Browser

4.5 Analysis of Transferred Data

4.6 Analysis of Server IPs and Status Code

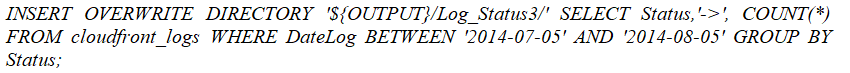
4.7 Exploratory Data Analysis of Log Data

For better understanding, Data Visualizations created using Tableau are included in all the sections mentioned above.

**4.1 Analysis of Status codes**

The following analysis produces the count of status codes returned by the servers over a specific time frame. In this case, the returned status codes are “200 (OK)” and “304 (Not Modified)”. The status code 200 indicates a successful request, where, the HTTP response 304 indicates that the requested resource has not been modified since the previous transmission. This status code indicates the server that there is no need to retransmit the requested resource. Furthermore, a 304 Not Modified response code serves as an implicit redirection to a cached version of the requested resource (*Powell-Morse, 2018*). The modified Hive script and its output are as follows.

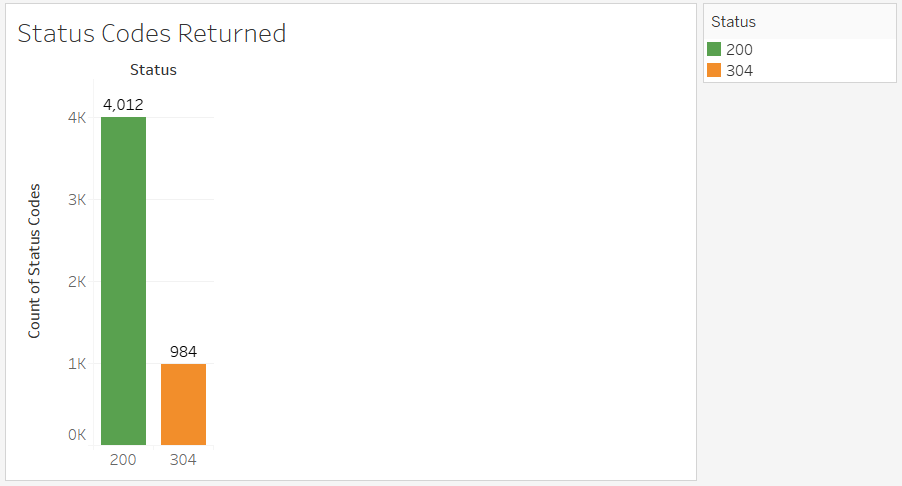
***Query used:***



***S3 Output:***



***Tableau Visualization***



*Fig 4: Bar Chart representing the status codes returned by the web servers*

***Inferences***

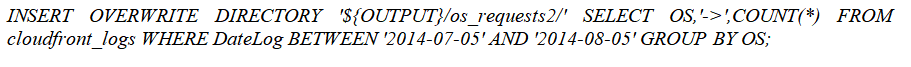
*From the visual interpretation of the bar chart as well as the hive output it is quite evident that,*

* *Status codes returned were 200 (OK) and (304 – Not Modified)*
* *No error codes like 404 (Not Found) were returned*
* *Indicates that the system was functioning correctly during July and August 2014*

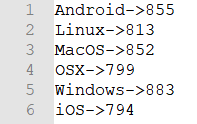
**4.2 Total Requests based on Operating System**

The following query retrieves the total number of requests serviced by the web server based on the operating system used by the user. Here, requests from ‘2014-07-05’ to ‘2014-08-05’ are taken into consideration. The modified query and output are as follows.

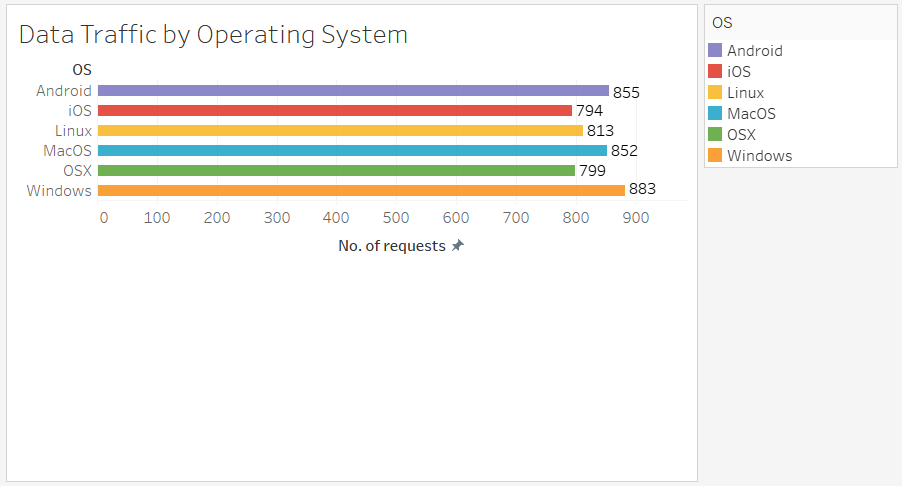
***Query used:***



***S3 Output:***



***Visualization***



*Fig 5: Bar chart representing number requests categorized by Operating system*

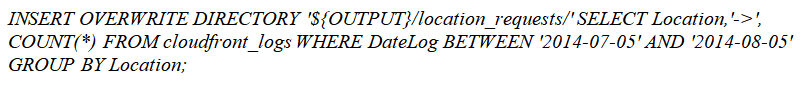
***Inferences***

* *The distribution of data traffic is almost equal among the different operating systems over the period 2014-07-05 and 2014-08-05*
* *Eurostar's website is accessible across various devices and renowned operating systems*
* *These data can ensure and pave the way for solving technical issues that are specific to a particular operating system*

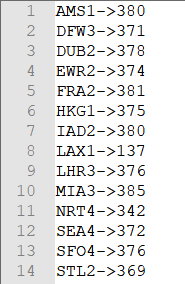
**4.3 Total Requests based on Location**

The below Hive query groups the log data based on the location of the client and counts the number of requests at each location between ‘2014-07-05’ and ‘2014-08-05’.

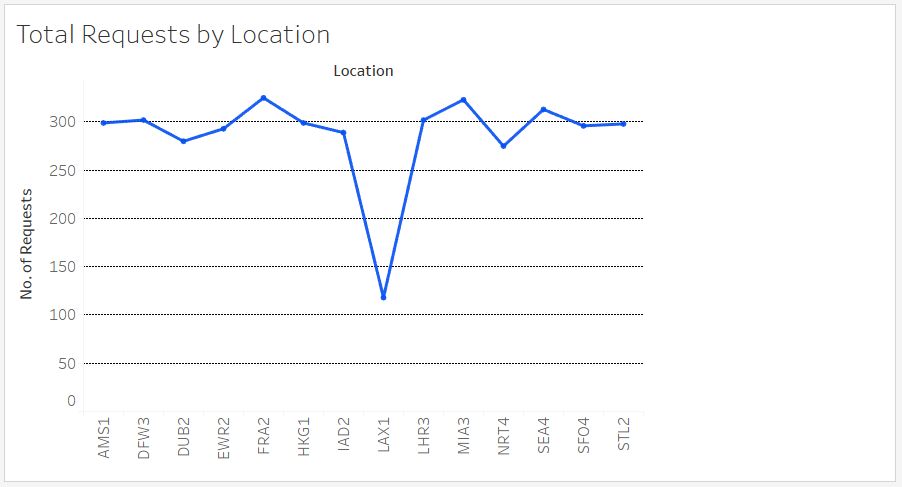
***Query used:***



***S3 Output:***



***Visualization***



*Fig 6: Line Graph showing No. of requests categorized by location*

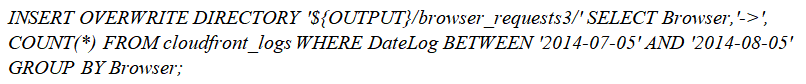
***Inferences***

* *The number of requests from the location ‘LAX1’ is low when compared to other locations*
* *Any error status codes about the location indicate a technical issue in accessing the website at that particular location. Since there is no error codes identified, this graph indicates that the number of customers in the location LAX1 is considerably low.*
* *The organization should take steps to improve the number of customers at these locations.*

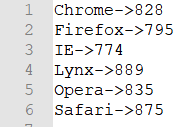
**4.4 Total requests per Internet Browser**

The below query categorizes the log data based on the browser from which the request is received , also, the it counts the number of requests from each browser between the period ‘2014-07-05’ and ‘2014-08-05’. Below is the modified HiveQL,

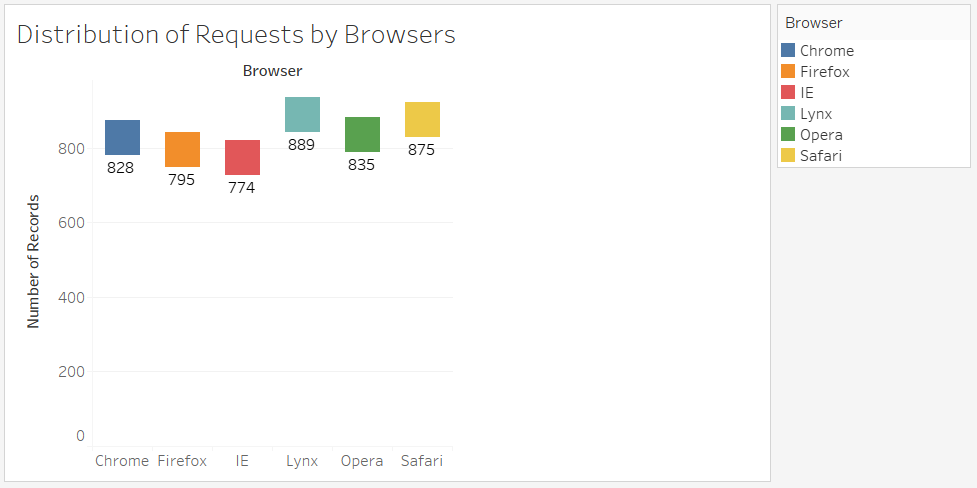
***Query used:***



***S3 Output:***



***Visualization***



*Fig 7: Graph showing distribution of requests by browsers*

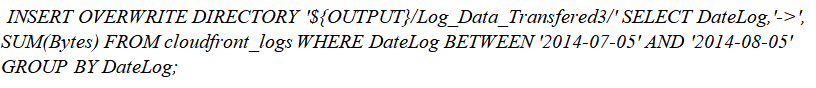
***Inferences***

* *The distribution of data traffic is almost equal among the different web browsers over the period 2014-07-05 and 2014-08-05*
* *Eurostar's website is successfully accessible across various web browsers*
* *Since browsers are the main point of communication between the user and the website, these data can help in accurately identifying any bowser specific errors*

**4.5 Analysis of Transferred Data**

The following query gets the total amount of data transferred over the given period. Furthermore, a time series plot is constructed using the retrieved data for better visualization; also, a date selector is included with the graph for filtering

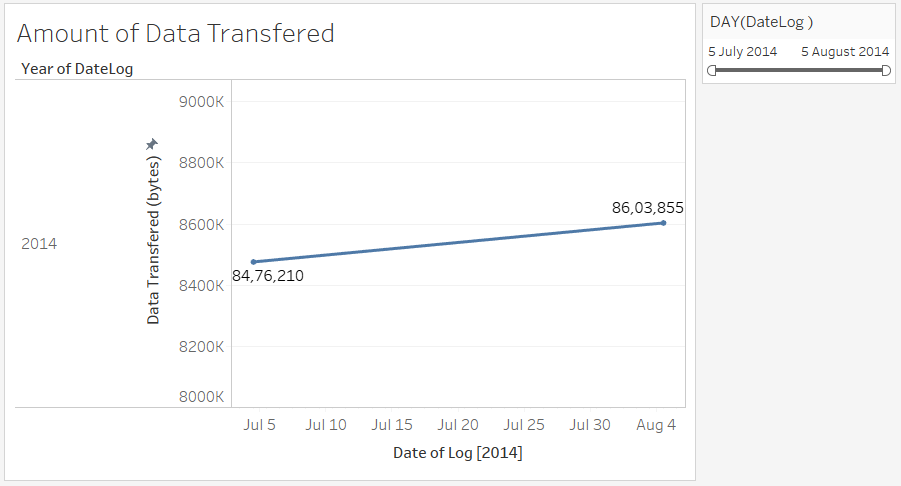
***Query used:***



***S3 Output:***



***Visualization***



*Fig 8: Time series plot showing amount of data transferred over time*

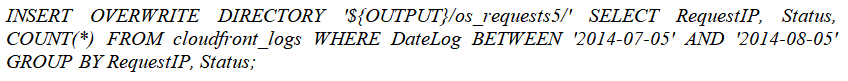
***Inferences***

* *There is an increase in the amount of data transferred over the period*
* *The time series plot constructed using the data can be used to analyse the trends in the data traffic and transmission*

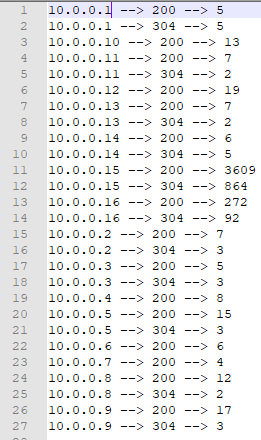
**4.6 Analysis of Server IPs and Status Code**

The hive script gives a statistical result of number of requests which returned the status as 200 (OK) and 304 (Not Modified Error) for each request IP address

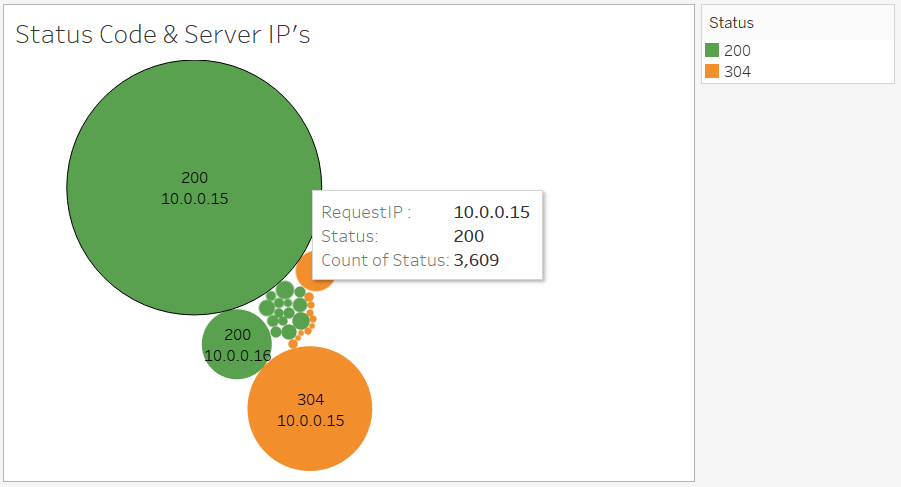
***Query used:***



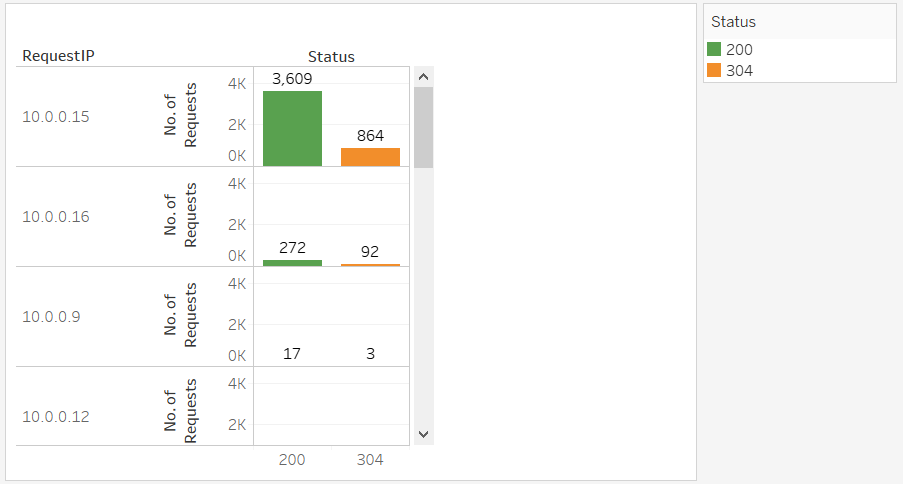
***S3 Output:***



***Visualization***



*Fig 9: Graph representation showing Request IPs and Status codes*



*Fig 10: Alternate representation showing Request IPs and Status codes*

***Inferences***

* *The data gives a clear picture of the most used servers. Here, the most accessed IP is 10.0.0.15.*
* *Helps in ensuring if load balancer is working properly and helps authorities to redirect server requests to a different IP (with low number of requests) during the maintenance/ downtime of another server*

**4.7 Exploratory Data Analysis of Log Data**

In this section the entire log data is retrieved from the database and is exported as a .csv file. The modified query and outputs are as follows,

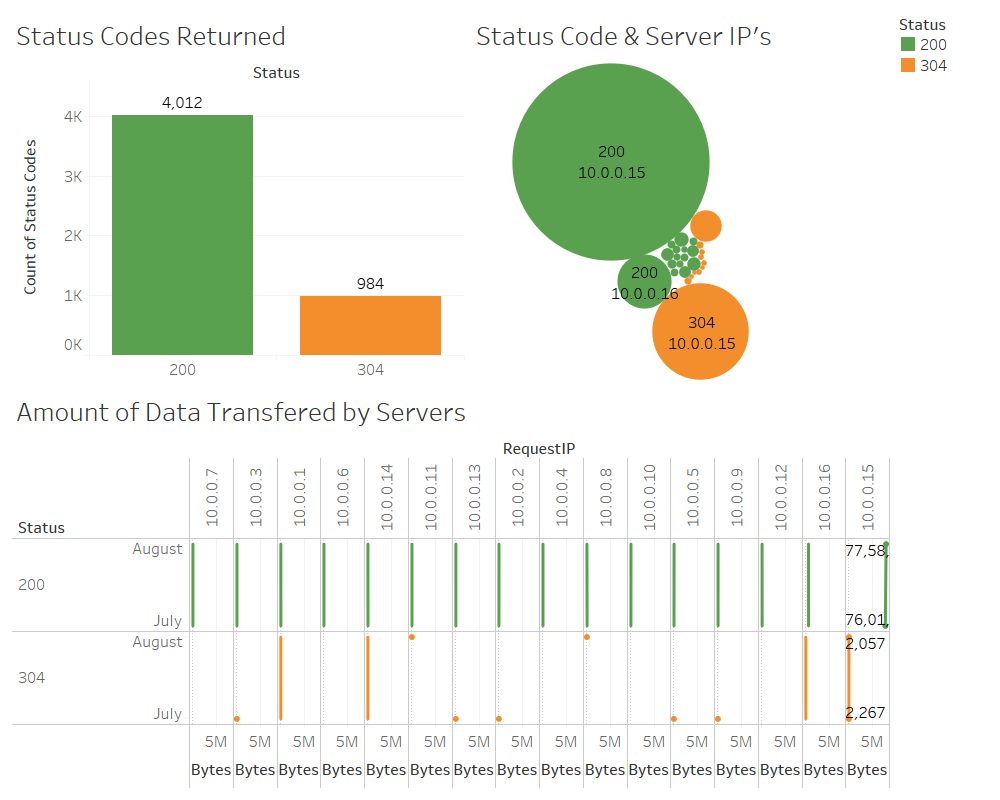
***Query used:***



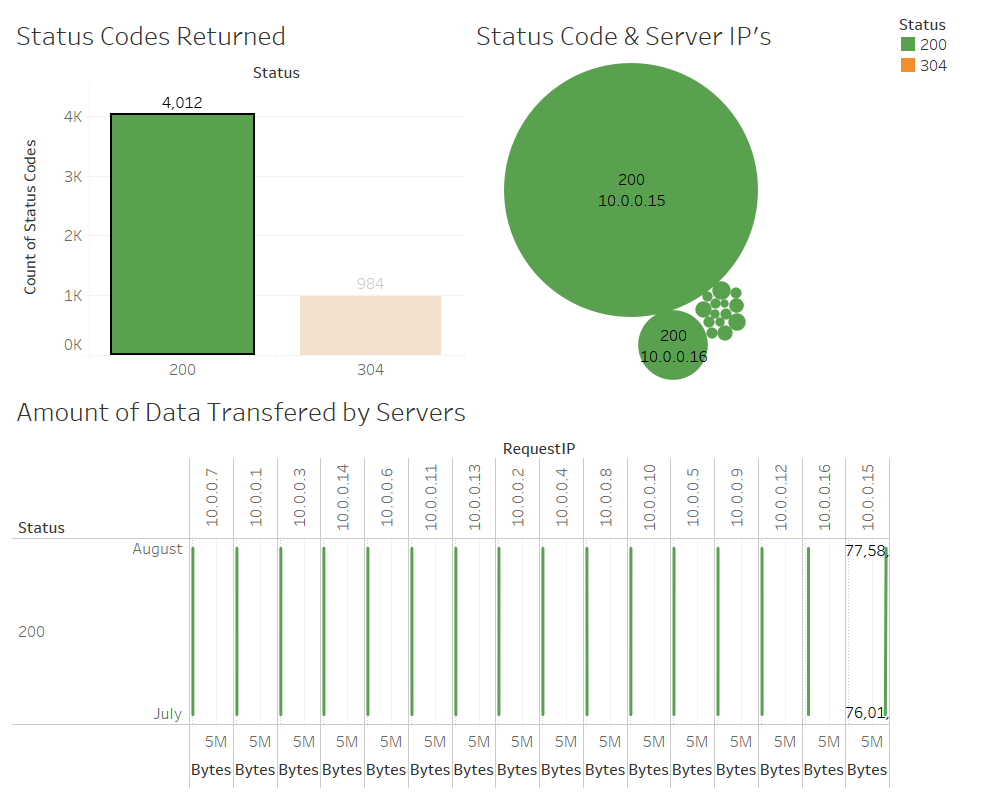
***S3 Output***



***Visualization***



*Fig 11: Dashboard showing Amount of data transfers by IPs differentiated by status codes*



*Fig 12: Dashboard showing Amount of data transfers by IPs differentiated by status codes*

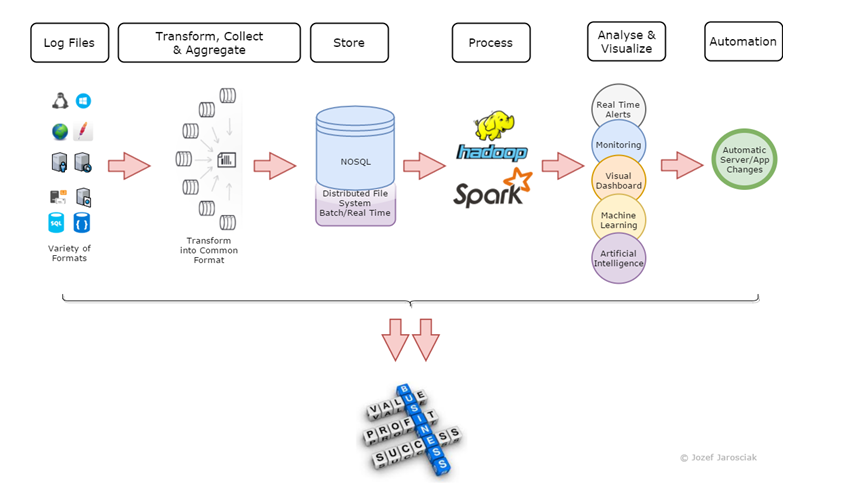
*(Status Code 200 is filtered)*

**5. Importance of Big Data Analytics in Log Management**

Log management is one of the significant challenges faced by large-scale as well as small-scale organizations. In general, web servers generate log files in many formats, also, with exponential growth in the volume of log files, it has become challenging for the organizations to manage and analyze them efficiently *(Jarosciak, 2017)*. Organizations are adopting Big Data Analytics to effectively manage the substantial computer-generated log data and obtain insights and patterns that can help in resolving potential business problems, consequently, increasing the business value.

***"According to a survey made among more than 200 professionals in IT, security and compliance, 55% of organisations have deployed big data projects and more than a half of them use the technology for log management. In addition, of those who plan to deploy big data in the next two years, 58% said that log management is a priority."***

From the above statement, it is quite evident that log management is crucial for an organization and how Big Data Analytics allows to effectively analyzing that logs. Log management may like adding metadata to information records, outputs from a log or audit tool or reports of sophisticated analytics software. A typical log management process comprises of the following steps; Log collection, Centralized log aggregation, Long term log storage and retention, Log rotation, Log analysis, Log search and reporting. The significant challenges involved in log data management are the volume, normalization, velocity and veracity of the log data. Below diagram illustrates log data handling using Big Data Analytics.



*Fig 13: Log analysis using Big Data Technologies (Jarosciak, 2017)*

The enormous amount of unstructured log data generated from different systems has made the process of log file analysis a tedious one. Logfile analysis is one of the most critical tasks that allow investigating and resolving production application failures and issues without violating security. In cases like this, the log analyst analyzes multiple log files to identify the root cause for the failure, which, typically is exceptionally time-consuming. Implementing Big Data Analytics is the most optimum solutions for deriving business value and insights through log management.

The figure mentioned above *(Fig. 13)* explains the steps involved in implementing Big Data solutions to log management. The first and most crucial step is to aggregate the logs into a single centralized source by converting the log files into a standardized format. Storing the log data in a centralized location is an added benefit. Also, the standard log format allows for effective data processing and analysis *(Jarosciak, 2017).*

As a second step, we move the transformed log data into a NoSQL database that operates on a distributed file system like the Hadoop Distributed File System (HDFS). After which the Hadoop system or Apache Sparklog processes the data. A few other log management tools are Logentries, GoAccess, Splunk and Octopussy. Log Analysis may involve advanced Big Data analytics techniques such as data mining, machine learning etc. The insights extracted are useful for building dashboards, search capabilities and implementing machine learning predictions. Big Data Analytics automates the entire process of log data analysis. Furthermore, it can help to monitor server utilization and alert the organizations in case of any security issues. In addition to that log analytics helps organizations from the threat of “Data Lake”

Some of the benefits of log management are,

* Monitoring the applications (website)
* Security Analytics
* Fraud detection
* Monitor Business Transactions and User logs
* Monitoring system performance and server utilization

**6. Conclusion**

Big Data Analytics has become a revolution in the internet era. It allows for monitoring both hardware and software performance through portals and dashboards. Applying Big Data Analytics into log management allows businesses to build advanced prediction models to gain improved insights for business success. For the reason mentioned above, many organizations are leaning towards Big Data Analytics for their log management for increased productivity and profit.

**7. References**

* Server log (2018) *Wikipedia*. [online] Available @ <https://en.wikipedia.org/wiki/Server_log>
* Powell-Morse, A. (2018), *304 Not Modified: What It Is and How to Fix It*. [online]

Available @ <https://airbrake.io/blog/http-errors/304-not-modified>

* Jarosciak, J. (2017), *Log Management and Big Data Analytics – Jozef Jarosciak Blog*. [online] Available @ <https://www.joe0.com/2017/02/05/applying-big-data-analytics-to-logging/>
* Importance of Log Management, Available @ <http://blog.securitymetrics.com/2015/08/importance-of-log-management.html>
* Log Management tools, Available @ <https://stackify.com/best-log-management-tools/>
* Importance of Log Analysis, Available @ <https://www.itworldcanada.com/blog/why-log-management-is-absolutely-critical-for-it-security/377711>