ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 9, Issue 2 , February 2022

A Web-Based Platform for Server Log Analytics

Vishnu U, Yashumithaa M, Varnith GV

Department of Computer Science and Engineering, Dayananda Sagar University, Bangalore

ABSTRACT - Web log analysis is a kind of web analytics software that parses a server log file from a web server, and based on the values contained in the log file, derives indicators about the visitors. It will give you information about your site's visitors: activity statistics, accessed files, paths through the site, information about referring pages, browsers, operating systems, and more. The program produces easy-to-read reports that include both text information (tables) and charts. Web server data is analysed to help the organization further improve the service they provide.

I. INTRODUCTION

Web Logs are emitted by network devices, operating systems, applications and all manner of intelligent or programmable devices. Log messages must usually be interpreted concerning the internal state of its source and announce security-relevant or operations-relevant events. The syntax and semantics of data within log messages are usually application or vendor-specific. The terminology may also vary. It will give you information about your site's visitors: activity statistics, accessed files, paths through the site, information about referring pages, search engines, browsers, operating systems, and more. The program produces easy-to-read reports that include both text information (tables) and charts. Web server data is analysed to help the institute further improve the service they provide. Features supported by log analysis packages may include "hit filters", which use pattern matching to examine selected log data. This project focuses on developing a cloud platform for analysing server-generated website logs. The results are shown graphically on the screen where the users can interact with the visualizations and see the status in real-time. The data from visualization can be used for data-driven decision making and improving the overall performance of the website. With the basic availability of user preference in this system, an organization can also choose to improve the user experience on the website.

II. LITERATURE SURVEY

The work carried out in the paper Analysis of visitor's behaviour from Web Log using WebLog Expert Tool is to implement a weblog expert tool on web server log files. The visitor pattern analysis is performed which is time and page view analysis. The outcome is finding a searching behaviour of visitors by using a tool and according to this behaviour we maintain the website. The analysis does not occur in real-time. The authors of this paper are Manoj Kumar and Mrs Meenu (2017)[1]. The work carried out in the paper titled Web Log Analysis is based on Hadoop Technology. Hadoop is used for storing huge datasets and YARN / MapReduce is used as a data processing framework for processing, querying and displaying the data. The outcome of this method is that larger datasets can be handled easily and the processing speed will be much faster thus providing results without delay. The author of this paper is Zhao Yongjian (2019) [2]. The work carried out in the paper titled Analysis of Web Site Using WebLog Expert Tool Based on Web Data Mining. The user has to feed the data to the WebLog Expert Tool & the visualization will be generated. Shows the visualization as output for the following parameters: Visitor hit analysis, Page views analysis, Time analysis, Website visitors of the source, Website that is accessed of the pages & Number of documents downloaded. The authors of this paper are Satya Prakash Singh & Meenu (2017) [3]. The work carried out in the paper titled Log Analysis in Cloud Computing Environment with Hadoop And Spark. Hadoop is used for storing huge datasets and spark/hive is used as a data processing framework for processing, querying and displaying the data. The

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 9, Issue 2 , February 2022

outcome of this method is that larger datasets can be handled easily and the processing speed will be much faster thus providing results without delay. The authors of this paper are Xiuqin LIN, Peng WANG, Bin WU,(2013) [4]. The work carried out in the paper titled Log Analytics on Cloud using Pattern Recognition. The data is extracted from the log files. Then the extracted and formatted data is used for pattern construction from phrases based on domain knowledge. Then the report generated is represented in a graphical form. The outcome of this method is that larger datasets can be handled easily and the processing speed will be much faster thus providing results without delay. It provides opportunities for analysing complex and large volumes of data easily. These log files are currently used only for monitoring the devices in offline mode & are not real-time. The authors of this paper are Avinash Bhole, B Adinarayana, Sanath Shenoy, (2015) [5]. The work carried out in the paper titled Big data analysis for sensor time-series in automation by Jirkovsky, V.; Obitko, M.; Novak, P.; Kadera, P. Emerging Technology and Factory Automation (ETFA), 2014 IEEE Big Data analytics is not only limited to the web-based industries. It is extremely useful in the case of factory automation. They have taken passive house as an example, measured series data with the goal of detecting specific events. Mainly, the data analytics is still performed in a normal way as texts that are read into MATLAB or R for processing the information. This is not only inaccurate and inefficient; it also limits scalability and usability [6]. The authors of the paper titled Specialized storage for big numeric time series are I. Shafer, R. R. Sambasivan, A. Rowe, and G. R. Ganger. In HotStorage'13 - 5th USENIX Workshop on Hot Topics in Storage and File Systems, 2013. An Open TSDB (Time Series Database) is a widely used current state of the art system. The foundation for this is HBase, An Apache Hadoop database that has the aim of storing, indexing and providing metrics at a huge scale, making such data easily accessible and visualizable [7]. In the paper titled Lessons learned from OpenTSDB. In HBaseCon 2012, 2012. The foundation for this is HBase, An Apache Hadoop database that has the aim of storing, indexing and providing metrics at a huge scale, making such data easily accessible and visualizable. Applications in the devices provide a lot of log files that were traditionally used only for monitoring, i.e. finding the failure reasons of an application in the log files. This requires a lot of effort & it was time-consuming & data has to be loaded manually & visualization was not real-time. The author of this paper is B. Sigoure [8]. The work carried out in the paper titled Applying Hadoop for log analysis toward distributed IDS[C]//Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication. New York, NY, USA: ACM, 2013:3:1–3:6. The systems designed just based on Hadoop Map/Reduce or even a combination of Hive have solved the large scale data processing and storage problems, but are not suitable for a class of applications like interactive query and iterative algorithm which is common in the analysis system. The author of this paper is Therdphapiyanak J, Piromsopa K [9].

III. METHODS

The system consists of the server-side and client-side. The client-side is made up of HTML, CSS and JavaScript and consists of the UI of the platform. The server side consists of Flask Server running listening to requests from the client-side, performing the requested queries and returning the results. The server also has connectivity with a cloud-based data warehouse. Data Warehouse stores data in an SQL table. The platform provides 6 options for the user to perform the analysis: **FileQuery with Upload, FileQuery without Upload, FileDashboard with Upload, FileDashboard without Upload, CloudDashboard, CloudQuery.**

FileQuery with Upload: The user uploads a log file (.log format) to the server and the file is parsed and converted into CSV format. This is used to perform aggregations and return the result in the form of a dashboard and table (shows queried data). The parameters for querying include IP Address, Day Month, Year, HH, Request Type, API, Status Code, Referrer, User Agent, Device, OS, Browser, Bytes, Response Time.

FileQuery without Upload: The user uses an existing parsed file on the server to perform the query. Like the above option, the results are returned in the form of a dashboard and table (shows queried data). The parameters for querying includes IP Address, Day Month, Year, HH, Request Type, API, Status Code, Referrer, User Agent, Device, OS, Browser, Bytes, Response Time.

FileDashboard with Upload: FileDashboard shows all the visualizations related to a file. This option expects the user to upload a file to the server and the results are returned in the form of a dashboard.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 9, Issue 2 , February 2022

FileDashboard without Upload: FileDashboard shows all the visualizations related to a file. This option uses the existing parsed file on the server and the results are returned in the form of a dashboard.

CloudDashboard: CloudDashboard is a real-time dashboard that uses a cloud data source (Datawarehouse) to store and query data. The dashboard results are updated periodically by sending a query to the cloud data source. The Datawarehouse has a streaming insert that accepts parsed log entries from the server. They are stored in a table that is queried on the request from the client-side.

CloudQuery: CloudQuery is used to query data that is stored in the Datawarehouse. The parameters for querying includes IP Address, Day Month, Year, HH, Request Type, API, Status Code, Referrer, User Agent, Device, OS, Browser, Bytes, Response Time. The results are returned in the form of a dashboard and table (shows queried data).
Apart from the 6 options for analytics, we have one more option for purging the warehouse. This accepts the time as input from the user and deletes all the entries in the table which is less than or equal to the time given by the user.

Logfile Format: The log format used is standard Apache server log file format. Apache server has the following log file format :
<IP of client> <RemoteLogName> <UserID> [Date and Time in UTC format] "RequestType API Protocol/Version" <StatusCode> <Byte> <Referer> "<UAString>" <ResponseTime>

Example Log Entry: This is how a sample log entry on the Apache server would look like.
248.243.151.169 - - [27/Dec/2037:12:00:00 +0530] "POST /usr/admin/developer HTTP/1.0" 304 5011 "-" "Mozilla/5.0 (Linux; Android 10; ONEPLUS A6000) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/77.0.3865.116 Mobile Safari/537.36 EdgA/45.12.4.5121" 845

SQL Table Format: The SQL table contains the parsed log entry from a server. Each log entry will be parsed into the following format and stored in the table for querying.

IPAddress: IP Address of the client that is sending the request to the server. It is generally stored in a string format.

Remote Log Name: The log name of the remote user. For security reasons, it is not made available.

User ID: The ID of the user that is sending the request. Similar to the log name, it is also not made available due to security reasons.

Day: Day of the month on which the request is sent. It is represented by an integer value.

Month: Month on which the request is sent. It is represented by an integer value.

Year: Year on which the request is sent. It is represented by an integer value.

HH: Hour of day on which the request is sent by the client. It is represented by an integer.

Request Type: The type of request that is used by the request sent by the client (GET, PUT, POST etc). It is represented by a string value.

API: The API of the website to which the request was sent. This is represented by a string value.

Status Code: The status code of the request (304,404 etc). It is represented by an integer.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 9, Issue 2 , February 2022

Bytes: The bytes of data that is sent back to the client from the server. It is represented by an integer value.

Referrer: The website from which the user was directed to this website. If not, available it is represented by a '-'. It is represented by a string value.

User-Agent: The user agent string of the browser which is used by the client to access the server. It is represented by a string value.

Response Time: The response time that the server took to complete the request. It is represented by an integer value.

Timestamp: The UTC formatted date and time on which the log entry was generated. It is represented in the DATETIME data type in SQL.

Browser: The name of the browser used. It is derived from the user agent string that the server received. It is represented by a string value.

OS: The OS used by the client to access the server. It is derived from the user agent string that the server received from the client. It is represented by a string value.
Device: The device used by the client - Mobile or Others. It is derived from the user agent that the server received. It is represented by a string value.

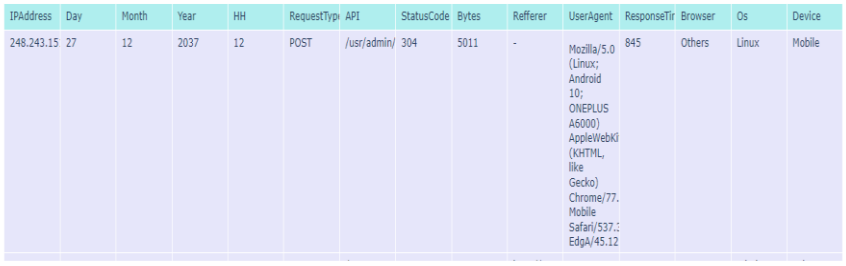
IV. DASHBOARD DESIGN

The dashboard is where the visualizations (table and charts) are shown. For querying requests, the table and dashboard are shown, while for other requests, only the dashboard is shown. The dashboard is divided into 4 parts concerning the category of the visualization.
Time Series Analysis: This shows the number of requests the server received with respect to time-based parameters. This part includes the following graphs – Year vs Count, Month wise Count, Date wise Count and Hour wise Count.

Request Analysis: This shows the analysis which is related to the request received by the server. These graphs help us dig more into the request details. This part includes the following graphs – Request type count, Status code count, API count.

Statistical Analysis: This shows the statistics (Minimum, Maximum and Average) of few measures like Response Time, Response Size. It also displays the total number of logins in the given data source.

Browser and Device Analysis: This part shows the visualizations relating to the browser, device and OS used by the users. This includes the following graphs – Browser count, Device count, OS Analysis.



IPAddress	Day	Month	Year	HH	RequestTyp	API	StatusCode	Bytes	Refferer	UserAgent	ResponseTir	Browser	Os	Device
248.243.15	27	12	2037	12	POST	/usr/admin/	304	5011	-	Mozilla/5.0 (Linux; Android 10; ONEPLUS A6000) AppleWebKi (KHTML, like Gecko) Chrome/77. Mobile Safari/537.: EdgA/45.12	845	Others	Linux	Mobile

Figure 1

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 9, Issue 2 , February 2022

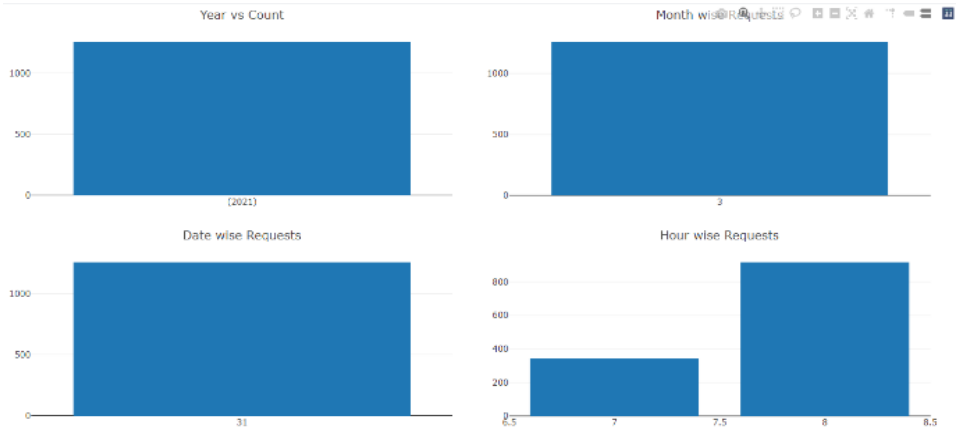


Figure 2

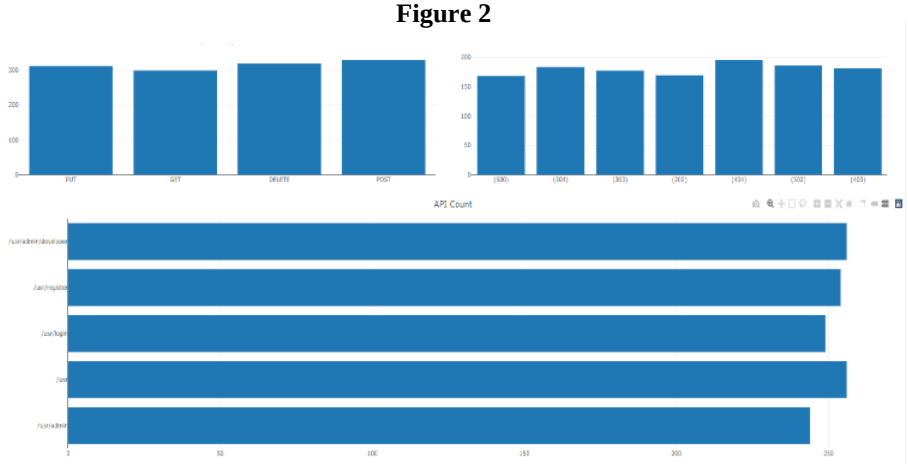


Figure 3

Response Size Statistics
Min Response Time : 4841 bytes
Max Response Time : 5168 bytes
Avg Response Time : 5000 bytes

Figure 4

Response Time Statistics
Min Response Time : 1 ns
Max Response Time : 5000 ns
Avg Response Time : 2427 ns

Figure 5

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 9, Issue 2 , February 2022

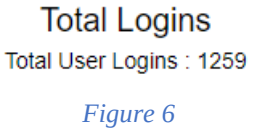


Figure 6

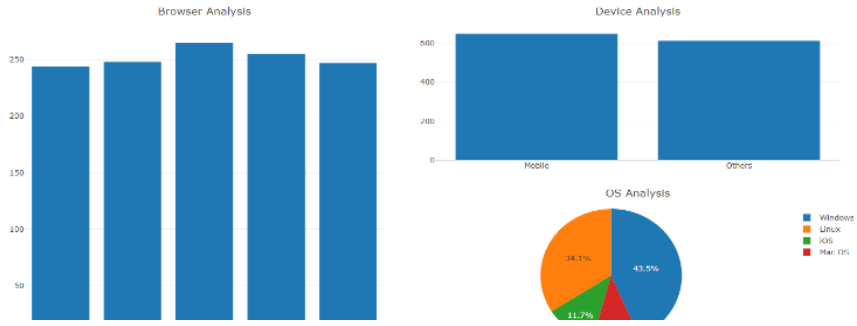


Figure 7

Figure (1) shows the output data in tabular form for querying operations,(2) shows the Time Series Analysis part of Dashboard,(3) shows the Request Analysis part of Dashboard,(4) shows the Response Size Statistics, a part of Statistical Analysis part of Dashboard,(5) shows the Response Time Statistics, a part of Statistical Analysis part of Dashboard,(6) shows the Total logins, a part of Statistical Analysis part of Dashboard,(7) shows the Browser and Device Analysis part of Dashboard.

Metrics and Dimensions for visualizations:

Time Series Analysis

Year vs Count: X-axis - List of distinct years in the data source, Y-axis - Count of each year in the data source.
Month wise Requests: X-axis - List of distinct months in the data source, Y-axis - Count of requests for each month in the data source
Date wise Requests: X-axis - List of distinct dates in the data source

Request Analysis

Request Type count: X-axis – List of distinct request types in the data source, Y-axis – Count of each request types in the data source.
Status code count: X-axis – List of distinct status codes in the data source, Y-axis – count of each status code in the data source.
API Count: X-axis – Count of each API in the data source, Y-axis – List of distinct APIs in the data source.

Statistical Analysis

Response Time statistics: This shows the minimum, maximum and average response time in the data source.
Response Size statistics: This shows the minimum, maximum and average response size in the data source.
Total Logins: Total number of logins present in the data source. This includes repeated logins from the same user as well.

Browser and Device Analysis

Browser Analysis: X-axis – List of distinct browsers in the data source, Y-axis – Count of each browser in the data source.
Device Analysis: X-axis – List of distinct devices in the data source, Y-axis – Count of each device in the data source.
OS Analysis: Labels – List of distinct OS in the data source, Values – count of each OS in the data source (This is a pie chart and hence labels and values).

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 9, Issue 2 , February 2022

The system follows a client-server architecture as discussed before. Though the client is any computer device, the server has a logic structure to route each request to its corresponding request handler.

Folder Structure of Flask Server: 5 folders are required by the Flask server to operate. These are used to store the output files generated by the server, to store the front-end code and to store files uploaded to the server.
/lib: Contains library files for the program
/outputFiles: Stores the output CSV files generated from parsing the log files.
/static: Contains CSS and JS files for the front end.
/templates: This contains the HTML templates for the front end.
/tmp: Stores the uploaded log file to the server from the user.

Routing of requests: For requests that require a file upload, the uploaded file is sent to be stored in the*/tmp* folder which is further parsed and converted to a CSV file. This is stored in */outputFiles* folder.
For requests that do not require uploading and the parsed file already exists in the server, the CSV is accessed and the results are returned to the client. For requests that need to access the cloud, the cloud connectivity is invoked and the query is sent in the form of an SQL query.

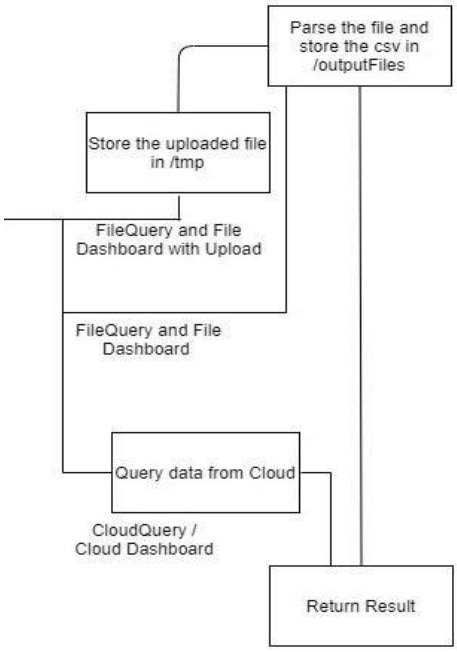


Fig 8 - Routing of requests

V. RESULTS

A Weblog Analysis Platform has been implemented therefore overcoming the disadvantages in the previously existing system and thus creating a more efficient and easier-to-use platform for analytics. The file querying and file dashboard has been implemented with a one-time file upload. The cloud query provides quick results with a table and dashboard. The cloud-based live dashboard provides updated results every 5-7 seconds.
The home page provides the user with available connection types (6 Analytics options + 1 purge data option). Connect button connects to the appropriate request handler in the server.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 9, Issue 2 , February 2022

For analytics options that require a file upload (FileDashboard with upload, FileQuery with upload) the above UI is presented once the option is chosen from the homepage. Users can click on the **Select File** button and choose the log file. Once chosen, the user can click on **Submit** button to submit the file to the server to get the corresponding output.
The custom query page allows users to set values for each parameter in the log file and query data with respect to those parameter values. The data source can be chosen to be a file or from the cloud data warehouse. The parameters include **IP Address**, **Day**, **Month**, **Year, Hour**, **Response-Type**, **API**, **Status-Code**, **Referrer**, **User-Agent**, **Device**, **OS**, **Browser**, **Bytes**, **Response-Time**. Once Submitted, the queried data along with the dashboard is displayed.
The queried data from the FileQuery and CloudQuery is displayed in a tabular format as shown above. All the parameters are shown queried according to the query parameters set by the user. The table is shown for user reference and viewing the data individually. For other analytics options except querying, only the dashboard will be present. The dashboard is divided into 5 parts as described below.
The Time Series Analysis shows a time-related analysis of the given data. Graphs titled **Year vs Count, Month vs Count, Date vs Count, Hour vs Count** are shown. These graphs visually represent the count of requests received for each Year, Month, Date and Hour. This part helps us perform analysis as time passes.
Request Analysis shows analysis related to the requests received by the server. Graphs like **Request Type Count**, **Status Code Count**, **API Count.** These graphs visually represent the number of requests with the count of their types, Status codes of each request with its count and the API to which the requests were sent. This part helps us understand analytics related to the requests received and improve request handling.
The Statistics shows statistical measures of certain parameters like **Response Time**, **Response Size**. Minimum, Maximum and Average of the above parameters are shown along with **Total Logins**. These values represent the present situation of the server relating to the amount of data returned to the client and the response time taken for each client. This part helps us perform analysis using statistical measures.
Browser and Device Analysis shows the analysis with respect to the Browser and Device of the client. Graphs titled **Browser Analysis**, **Device Analysis** and **OS Analysis** visually represent the names of browsers and their count, the Type of devices detected and their count, and the types of OSes detected and their count in the given data. This part helps us understand user preferences and improve user experience.

Figure 9

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 9, Issue 2 , February 2022

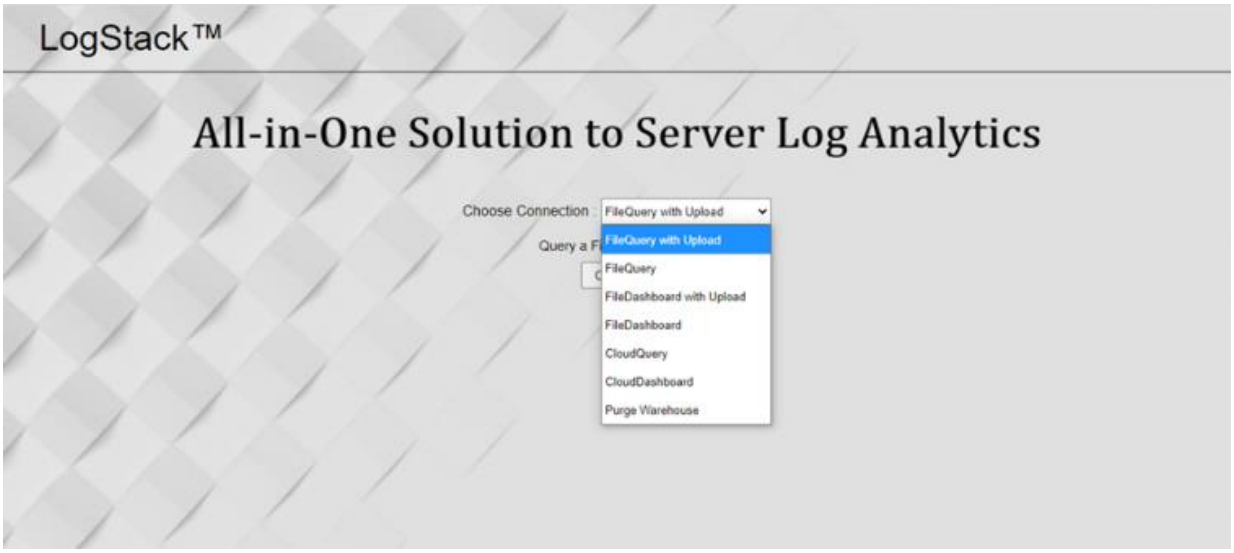


Figure 10

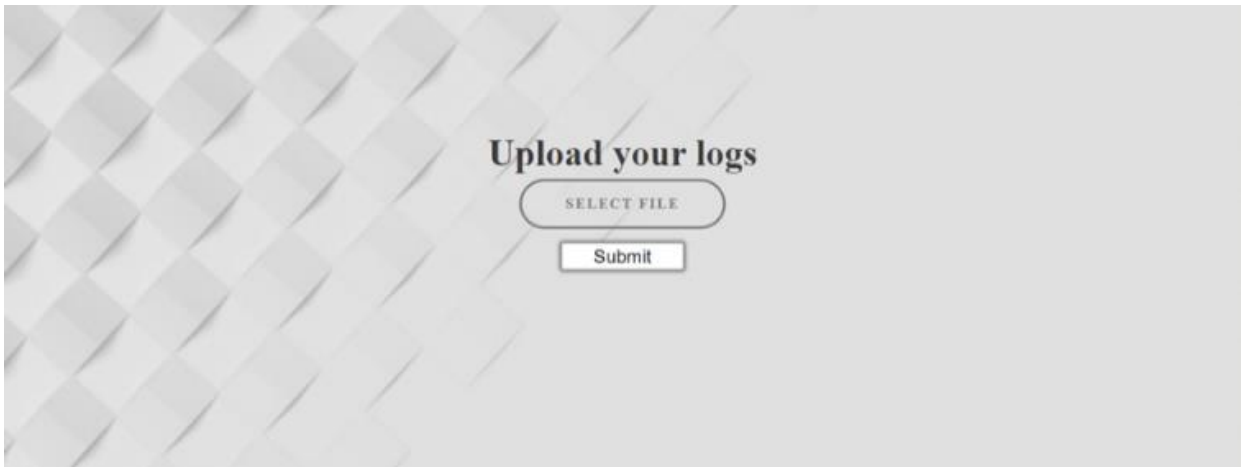


Figure 11

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 9, Issue 2 , February 2022

Figure 12

IPAddress	Day	Month	Year	HH	RequestTyp	API	StatusCode	Bytes	Refferer	UserAgent	ResponseTir	Browser	Os	Device
248.243.15	27	12	2037	12	POST	/usr/admin/	304	5011	-	Mozilla/5.0 (Linux; Android 10; ONEPLUS A6000) AppleWebKi (KHTML, like Gecko) Chrome/77. Mobile Safari/537.3 EdgA/45.12	845	Others	Linux	Mobile

Figure 13

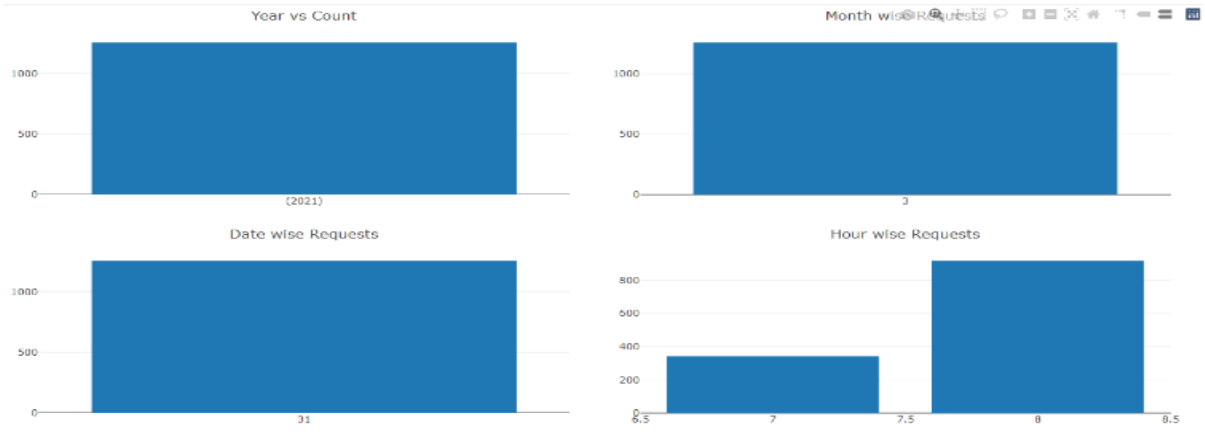


Figure 14

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 9, Issue 2 , February 2022

Figure 15

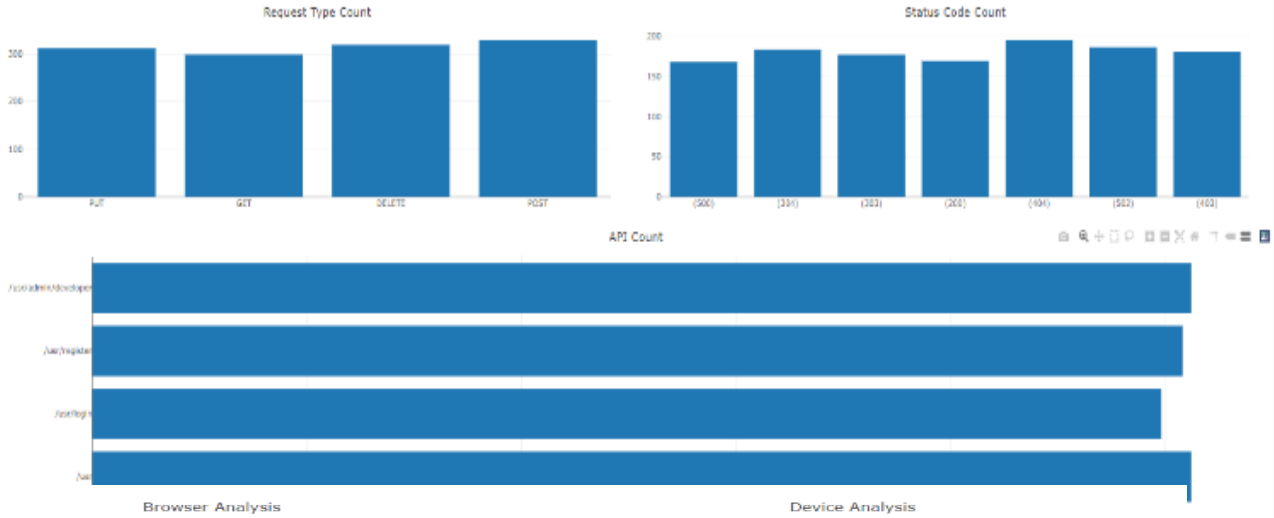


Figure 16

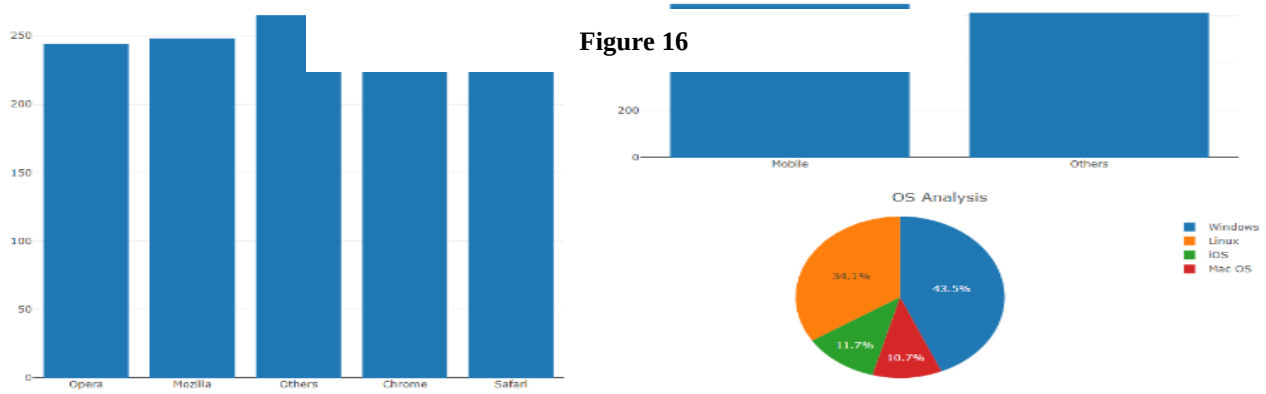


Figure 17

Figure (9) shows the home page of the system,(10) shows the available analytics options to which the user can connect,(11) shows the buttons to upload log file and submit to server, (12) shows the various parameters which can be used for query-based analytics,(13) shows the output table for queried data which is used user reference and individual inspection of data,(14) shows the Time Series Analysis part of the Dashboard,(15) shows the Statistical Analysis part of the Dashboard,(16) shows the Request Analysis part of the Dashboard,(17) shows the Browser and Device Analysis part of the Dashboard.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 9, Issue 2 , February 2022

VI. DISCUSSION AND CONCLUSION

Thus, the proposed system was implemented successfully overcoming the downside of the existing systems. A cloud data warehouse has been deployed in place of Hadoop as compared to existing systems thus providing. The proposed Live Dashboard has been implemented with a detailed descriptive analysis of the given data. The dashboard has been implemented for file data sources as well that provides descriptive and statistical analytics. Querying has been implemented for both file and cloud data sources which enable us to dig deeper into the given data thus providing in-depth analytics. Hence, with this application, organizations that use the internet as a medium for business can improve their business scenario and get an insight into what is going as well. The same can be used for decision making for current and future scenarios. The analytics helps study user preferences as well apart from the business thus giving users a good experience on your business website. On observing the advantages of this system over the existing systems and in general, the following can be concluded:

The data source can be multiple types – File or Cloud. This makes the system suitable for small scale organizations which use a file for storing logs and for large scale industries that use the cloud for storing weblogs. The system displays a complete dashboard with detailed descriptive and statistical analysis of the given log data thus helping in detailed analytics of the given data. The implemented system supports querying of data with various parameters for a given log format thus helps perform deeper analysis within the given data. There is an in-built connection to a cloud data warehouse which makes the system suitable for bigger organizations and helps efficiently query and analyse large quantities of log files with less delay. The system supports a live dashboard when connected to the cloud data warehouse, thus can report live results.

On observing the disadvantages of this system over the existing systems and in general, the following can be concluded:

The cloud data warehouse can result in higher costs to the organization when compared with other data storage methods but it outruns other storage methods in performance. The system is programmed to parse only the standard Apache log file format. Any format change will require the system to be reprogrammed. In practice, the dashboard takes 5-7 seconds for updating the results. Increasing the querying speed or using any other faster alternative will provide split-second analytics of the data in the cloud. For Future work, the system can be extended to improve the functionality as follows:

The system can be made to recognize custom log file formats by implementing dynamic parsing where the user provides the regex for splitting the log entry. Secondly, Increase the querying speed of the cloud data warehouse or find an alternative big data storage solution to reduce the running cost of the application. The ability to store multiple parsed logfiles in the server and choosing them using a UI would help store multiple parsed logfiles in the server rather than one parsed logfile.

VII. ACKNOWLEDGEMENT

We are grateful to the management of Dayananda Sagar University for providing us with the opportunity to work on this project. We would like to express my special thanks of gratitude to our Guide Dr Shivaprasad Ashok Chikop who gave us the mentoring to make this project a reality.

REFERENCES

[1] Dhawan, Sanjeev, and Swati Goel. "Web Usage Mining: Finding Usage Patterns from Web Logs." American International Journal of Research in Science, Technology, Engineering & Mathematics (2013): 203-207.
[2] Shukla, Rajesh, Sanjay Silakari, and P. K. Chande. "Web Personalization Systems and Web Usage Mining: A Review." International Journal of Computer Applications 72, no. 21 (2013).
[3] Nina, Shahnaz Parvin, Mahmudur Rahman, Khairul Islam Bhuiyan, and Khandakar Entenam Unayes Ahmed. "Pattern discovery of web usage
[4]Araya, Sandro, Mariano Silva, and Richard Weber. “A methodology for web usage mining and its application to target group identification."Fuzzy sets and systems 148, no. 1 (2004): 139-152.
[5] Facca, Federico Michele, and Pier Luca Lanzi. "Mining interesting knowledge from weblogs: a survey."Data & Knowledge Engineering 53, no. 3 (2005): 225-24
[6] Manoj Kumar, Mrs Meenu (2017), Analysis of visitor’s behaviour from Web Log using WebLog Expert Tool.
[7] Zhao Yongjian (2019), Web Log Analysis based on Hadoop Technology.
[8] Satya Prakash Singh & Meenu (2017), Analysis of Web Site Using WebLog Expert Tool Based on Web Data Mining.
[9] Therdphapiyanak, J., & Piromsopa, K. (2013). Applying Hadoop for log analysis toward distributed IDS, Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication - ICUIMC ’13.