A Comprehensive Study of the Elastic-Faust-Pinot-Superset Stack and its Performance Evaluation

Kriti Srivastava
Head of Department, Department of Computer Science and Engineering (Data Science)
Dwarkadas J. Sanghvi College of Engineering
Mumbai, India
kriti.srivastava@djsce.ac.in

Dhruv Salot
Student, Department of Computer Science and Engineering (Data Science)
Dwarkadas J. Sanghvi College of Engineering
Mumbai, India
dhruv.salot@outlook.com

Vrinda Jikadra
Student, Department of Computer Science and Engineering (Data Science)
Dwarkadas J. Sanghvi College of Engineering
Mumbai, India
vrindajikadra@gmail.com

Dev Patel
Student, Department of Computer Science and Engineering (Data Science)
Dwarkadas J. Sanghvi College of Engineering
Mumbai, India
devpatel0952@gmail.com

Neha Masurkar
Student, Department of Computer Science and Engineering (Data Science)
Dwarkadas J. Sanghvi College of Engineering
Mumbai, India
nehasm162001@gmail.com

***Abstract*— Web Server Log Analysis plays a crucial role in managing customer requests and monitoring their interactions in the field of web technology. This research paper focuses on exploring various tools and techniques, including Filebeat, Apache Kafka, Faust, Apache Pinot, and Apache Superset, to extract valuable insights from web server logs. Emphasizing the significance of log analysis in maintaining web server security and performance, this study highlights its role in identifying potential security vulnerabilities, improving website performance, and enhancing user experience. Real-world examples are provided to illustrate the practical applications of log analysis in mitigating security threats and optimizing website functionality. In response to these imperatives, the research demonstrates how a meticulously designed data pipeline incorporating Filebeats, Kafka, Faust, Apache Pinot, and Apache Superset can process vast volumes of web server logs. An impressive data ingestion rate of approximately 6,000 messages per second was achieved using the Elastic-Faust-Pinot (EFP) stack, showcasing the scalability and efficiency of this approach. Furthermore, the research's stream processing capabilities with Faust exhibited a remarkable processing speed of around 2,000 messages per second. This robust performance, coupled with a favorable read-write ratio of 3:1, indicates the research's capacity to handle demanding read-heavy workloads, significantly better than the Elastic-Logstash-Kibana (ELK) stack. These results collectively underscore the practicality and effectiveness of the log analysis framework, empowering organizations to extract actionable insights, bolster security measures, and refine web performance.**

Keywords*— *Web Server Log Analysis, EPF Stack, ELK Stack, Real-time Data, Big Data, Apache Kafka, Filebeat, Faust Streamer, Apache Pinot Apache Superset.

I. Introduction

In the realm of web technology, the management of customer requests and the tracking of their interactions necessitate effective Web Server Log Analysis. This work is driven by the imperative need to enhance web server security, optimize performance, and augment user experiences. By leveraging advanced tools and techniques, including Filebeat, Apache Kafka, Faust, Apache Pinot, and Apache Superset, this research explores the extraction of invaluable insights from web server logs. The research then compares the EFPS (Elastic-Faust-Pinot-Superset) Stack, used in the work, with the, more ubiquitous in industry, ELK stack (Elastic-Logstash-Kibana) to gain further insights and collate the performances of these stacks. This work stands as a comprehensive endeavour to address the pressing challenges of maintaining web server integrity and efficiency while harnessing the potential of log analysis for data-driven decision-making.

A. Objective of the Work

The objective of this research work is to address the critical need for comprehensive analysis of web server logs in the context of web technology. The digital landscape has witnessed unprecedented growth in web traffic and interactions, making it imperative to decipher the valuable insights hidden within the vast volumes of log data generated. This work aims to provide a robust framework for extracting actionable information from web server logs, enabling organizations to enhance security, optimize performance, and elevate user experiences.

B. Data Description

The data at the heart of this research comprises self-generated, near-real-time web server logs, which capture detailed information about every interaction with a website in a controlled environment. These logs include IP addresses, requested resources, HTTP status codes, and timestamps. They serve as a rich source of information that, when effectively analyzed, can unveil patterns, anomalies, and trends critical to web operations. Thus, one can easily emulate errors and control the timings with random data. The web server log structure is of the Combined Log Format () as shown in the following figure (see Fig. 1).

As shown in Fig. 1, Field denoted by ‘1’ is the IP address of the client that made the request, ‘2’ is the identity of the client (anonymized by ‘2’), ‘3’ denotes the user id of the person requesting the resource, ‘4’ is the timestamp, ‘5’ is the request type and the requested resource, ‘6’ being HTTP response status code (success: 2xx, redirect: 3xx, error:

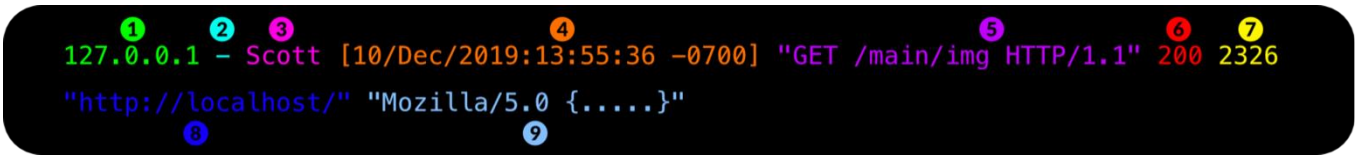

Fig. 1. Visualization depicting the structured organization of web server log data, distinguished by specific sequences of information.

4/5xx), '7' is the byte size of the object returned to the client, '8' is the HTTP Referrer address and '9' is the user agent browser/operating system/platform).

C. Need of Work

The imperative need for this research emerges from the increasingly pivotal role of log analysis in the realm of web server management. As the digital landscape continually evolves, organizations face mounting challenges in maintaining the security and efficiency of their web servers. Cyber threats have become more sophisticated, and user expectations continue to soar. Consequently, there is a pressing demand for harnessing the power of log data to identify vulnerabilities, enhance website performance, and deliver superior user experiences. The key challenges driving this need include the demand for high scalability to efficiently process the burgeoning volume of log data generated by modern web servers. Real-time processing and analytics are essential to contend with the rapid velocity of incoming logs, making timely insights crucial. Accurate data transformation and cleaning are imperative for reliable analysis. Moreover, the challenges of storing and retrieving vast amounts of log data must be met, along with the requirement for insightful visualization and interpretation. Complex data integration with other sources, such as databases and APIs, adds further complexity. In contrast, the advantages are substantial. This research empowers organizations to identify popular web pages, monitor website security, and swiftly resolve technical issues. It facilitates data-driven decision-making, thereby enabling businesses to navigate the digital landscape effectively and achieve sustainable growth.

II. Literature Review

Mayur Mahajan, Omkar Akolkar, Nikhil Nagmule, and Sandeep Revanwar [1] underscore the potential of Hadoop in real-time web log analysis. Hadoop's capacity to process substantial data sets in real time has significant implications for gaining insights into user behaviour and optimizing website performance. By proposing and evaluating a system for real-time web log analysis, this work contributes valuable insights into the practical application of Hadoop, demonstrating its ability to efficiently process extensive weblogs and derive timely results.

In their work, Dr. S. Suguna and M. Vithya [2] focus on introducing a structured framework for harnessing big data tools in web log analysis. Their approach underscores the multifaceted objectives of web log analysis, encompassing security threat identification, performance enhancement, user behaviour analysis, and business insights. By leveraging powerful tools such as Hadoop, Spark, Hive, Pig, and the EFP stack, the authors establish a robust framework involving data collection, cleaning, transformation, analysis, and visualization. The research demonstrates that this framework effectively unlocks valuable insights, enhancing the security and performance of websites and applications in a scalable manner.

In their comprehensive study, Mostafa Mohamed Shendi, Hate Mohamed Elkadi, and Mohamed Helmy Khafagy [3] delve into the realm of big data log analysis, shedding light on its multifaceted nature. They emphasize the significance of this discipline, which aims to address security threats, enhance system performance, scrutinize user behaviour, and offer insights into business operations. The authors highlight a spectrum of tools available for such analysis, recognizing their varying strengths and areas of applicability. Additionally, the paper underscores the formidable challenges, such as dealing with the sheer volume, variety, velocity, and veracity of data, while also addressing pertinent issues related to data quality, privacy, and security. The study ultimately anticipates a promising future for big data log analysis, given the ever-expanding complexity and volume of data in the digital landscape.

Sohan Panwar and Garima Silakari Tukra [4] introduce a methodology for effectively leveraging Hadoop to analyze web server log files in their work. Their approach aims to achieve various objectives, including security threat detection, website and application performance enhancement, user behaviour analysis, and insights into business operations. By employing Hadoop's distributed architecture, which features master-slave nodes for data distribution, the authors create a scalable framework. Their methodology involves loading log files into HDFS, preprocessing with HiveQL, log analysis using HiveQL, and result visualization with Jaspersoft Report. The research demonstrates that this methodology successfully uncovers valuable insights for enhancing the security and performance of websites and applications.

In the realm of web server log analysis, two noteworthy papers contribute to the discourse on harnessing big data technologies. Mudassir Khan and Shakila Basheer [5], provide a comparative examination of big data technologies, including Hadoop and MapReduce, against alternatives like Spark and Hive. Their findings endorse the scalability and fault tolerance of Hadoop and MapReduce for mining web server logs in distributed clusters, highlighting the practicality of these tools. Similarly, Sohan Panwar and Garima Silakari Tukra [6] present a comprehensive methodology for analyzing web server log files using Hadoop, emphasizing security threat detection and performance enhancement. Both papers underscore the significance of big data technologies in addressing the multifaceted goals of web server log analysis, providing valuable insights into the challenges and benefits of utilizing these tools in this context.

Korzeniowski and Goczvla [7] provide a comprehensive review of automated log analysis in various domains in their paper. It highlights the utility of automated log analysis for extracting insights from log files, discusses tools and challenges, and emphasizes its relevance in web server log analysis for security, performance enhancement, and user behaviour analysis.

In recent literature, the focus has notably shifted towards the efficiency and applicability of log analysis using the ELK stack, particularly in comparison to other prominent frameworks. In their study, Sung Jun Son and Younsmi Kwon [8] conducted a thorough comparison of the ELK stack and a commercial system in the context of security log analysis. The

research focused on evaluating the performance using critical benchmarks such as throughput, query latency, and accuracy. Surprisingly, the ELK stack demonstrated superior performance in throughput and query latency compared to the commercial system. The authors highlighted the scalability advantage of the ELK stack by emphasizing its ability to efficiently handle larger datasets through cluster expansion. Despite the commercial system outperforming the ELK stack in accuracy, the findings suggest that the ELK stack is a highly viable and cost-effective option for security log analysis, showcasing its potential to challenge commercial counterparts. Another pertinent work by Prakash, T., Kakkar, M., & Patel, K. [9] showcased the formidable capabilities of the ELK stack in enabling real-time web server log analysis and geo-identification of web users. The study emphasized the scalability and flexibility of this framework, making it a robust choice for processing substantial volumes of web server logs efficiently. Furthermore, the work highlighted the advantages of using the ELK stack, underscoring its ease of implementation and management. The insights from this research serve as a valuable benchmark for evaluating the performance of the ELK stack against emerging alternatives like the EFP stack, providing an essential perspective for researchers and practitioners in the domain.

In the landscape of log analysis, the cited works collectively illuminate the significance of big data technologies and distributed computing frameworks, particularly Hadoop, in the context of web server log analysis. They elucidate the potential of these tools for real-time analysis, security threat detection, performance enhancement, and user behaviour insights. The present work builds upon this foundation by proposing an innovative approach for web server log analysis using a comprehensive suite of big data tools, which includes Hadoop, Spark, Hive, Pig, and the EFP and ELK stacks. The unique contribution lies in the development of a unified framework that seamlessly integrates data collection, cleaning, transformation, analysis, and visualization. This holistic methodology empowers organizations to efficiently unlock actionable insights, addressing the challenges posed by the volume, variety, velocity, and veracity of data in web server logs. Furthermore, the current research extends beyond the boundaries of prior studies by not only addressing the shortcomings and limitations of existing methods but also by offering a novel approach for web server log analysis, ensuring its relevance in the ever-evolving digital landscape.

III. Methodology

A. *Tools Used*

1) Elastic Filebeats: Elastic Filebeats is a powerful log shipper and lightweight data shipper that plays a vital role in the Elastic Stack ecosystem, specifically designed for collecting, parsing, and forwarding log data to Elasticsearch or Logstash for further analysis. Its distinct advantages set it apart from other similar tools. Firstly, Filebeats offers unparalleled simplicity, ensuring ease of deployment and configuration. With its lightweight architecture, it imposes minimal overhead on the monitored systems. Moreover, it supports a wide range of modules for various log types, making it highly versatile. Additionally, Filebeats can be easily integrated with other Elastic Stack components, providing a seamless end-to-end log management solution. However, like any tool, it comes with some limitations. It may not be as feature-rich as some competitors for specific use cases, and its performance can be affected if not properly configured. Nonetheless, Elastic Filebeats excels in its role as a data shipper, offering efficiency and flexibility for log data collection and analysis (Filebeat references [11]).

2) Apache Kafka: Apache Kafka is a distributed streaming platform known for its exceptional capabilities in managing real-time data streams efficiently. Drawing from its official documentation, Kafka offers distinct advantages over similar tools. Firstly, Kafka boasts high throughput and fault tolerance, making it reliable for mission-critical applications. It provides strong durability guarantees, ensuring data integrity. Moreover, Kafka's publish-subscribe and fault-tolerant design enables horizontal scalability, making it suitable for handling massive data volumes. Apache Kafka stands out as a robust, fault-tolerant, and scalable solution for real-time data streaming and processing (Kafka Documentation [12]).

3) Faust Streamer: Faust is a stream processing library, that ports the ideas from Kafka Streams to Python. It is used at Robinhood to build high-performance distributed systems and real-time data pipelines that process billions of events every day. Faust provides both stream processing and event processing, sharing similarities with tools such as Kafka Streams, Apache Spark/Storm/Samza/Flink. It is Python only, which means you can use all your favourite Python libraries when stream processing: NumPy, PyTorch, Pandas, NLTK, Django, Flask, SQLAlchemy, and all of the Python libraries (Faust Notes [13]).

4) Apache Pinot: Apache Pinot is a distributed, real-time analytics and data warehousing platform designed for handling large volumes of data efficiently. Its advantages, as outlined in its official documentation, include impressive scalability and real-time capabilities. Pinot is optimized for low-latency queries, making it ideal for interactive analytics and real-time monitoring. It also offers a pluggable architecture that supports various data sources and query engines (Apache Pinot website [15]).

5) Apache Superset: Apache Superset is a powerful open-source data visualization and exploration platform with several advantages when compared to other similar tools. As cited from the official Apache Superset documentation, its standout feature is its user-friendly and intuitive interface that allows both technical and non-technical users to create interactive and visually appealing dashboards effortlessly. Superset supports a wide range of data sources, making it versatile for organizations with diverse data ecosystems. Additionally, it boasts an active and growing community that continually contributes to its development and provides a wealth of extensions and

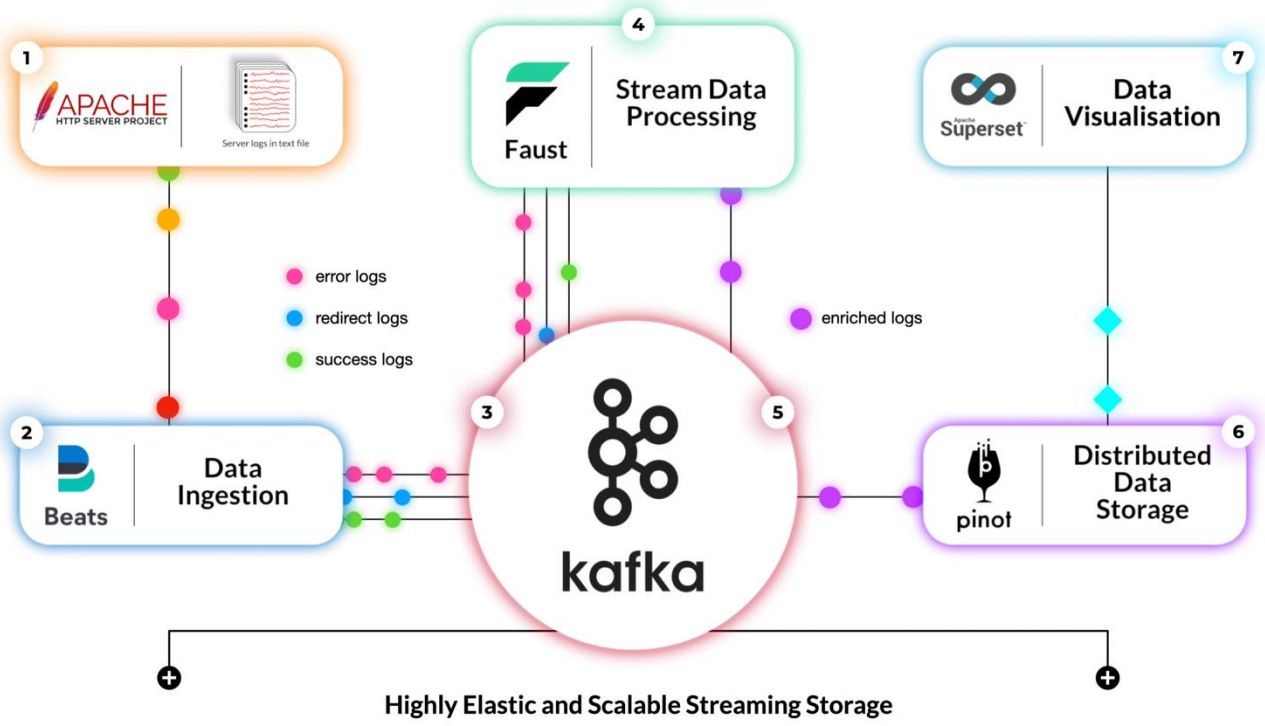

Fig. 2. Illustration of the Data Pipeline for Log Data Analysis in the Proposed System using the EFPS Stack (Elastic, Faust, Pinot, Superset).

plugins. However, it's important to note that while Superset is feature-rich, it may not have the same level of advanced analytics capabilities as some specialized BI tools. Nonetheless, Apache Superset is an excellent choice for organizations seeking an open-source, user-friendly, and extensible data visualization platform (Apache Superset website [15]).

B. Data Pipeline

The elucidation of the data pipeline about the EFPS Stack Data Pipeline (see Fig. 2), along with a delineation of the sequential stages through which our data traverses, is as follows:

1) In this research, log data are generated at high velocity using the log data generator from the GitHub project [16], which includes error logs, redirect logs, and success logs, stored in text files.

2) Elastic Filebeats, plays a pivotal role in data ingestion. It efficiently collects, parses, and forwards log data from various sources, including the Apache HTTP Server logs. This ensures seamless data flow for real-time log streaming and further integration with Apache Kafka.

3) Apache Kafka, acts as the central hub for data streaming. It provides a scalable, fault-tolerant, and high-throughput environment for real-time data feeds. Kafka facilitates the reliable publishing and consumption of streams of records.

4) Faust Streamer is employed for stream processing. Developers leverage Faust to define stream transformations, aggregations and joins using familiar Python syntax. This step enhances data processing capabilities and prepares the data for further analysis.

5) These enriched logs are sent back to Apache Kafka which ensures a continuous flow of log data.

6) Apache Pinot takes charge of distributed storage. Specifically designed for high-performance analytics and real-time querying of large-scale datasets, Pinot offers fast data ingestion, efficient storage, and low-latency querying capabilities. It ensures data is stored and accessible for analysis.

7) Apache Superset enables users to create interactive dashboards and visualizations from various data sources, including the log data stored in Pinot. Its support for a wide range of data connectors and near real-time chart updates empowers users to gain valuable insights and make data-driven decisions.

These seven steps represent a comprehensive data pipeline that efficiently manages the flow of log data from its origin in the Apache HTTP Server project to its visualization and analysis in Apache Superset. Each component plays a vital role in ensuring the integrity, accessibility, and utility of the log data throughout the process.

IV. Results

In this work, a comprehensive comparative analysis is conducted between the EFP (Elastic-Faust-Pinot) stack and the ELK (Elastic-Logstash-Kibana) stack, focusing on critical performance metrics essential for efficient data processing. Specifically, the study delves into evaluating the Read-to-Write Ratio and Write Count per Second, essential indicators of the system's responsiveness and throughput. These metrics hold paramount significance in assessing the operational efficacy and real-time processing capabilities of the respective stacks. By scrutinizing these performance aspects, this research aims to provide a nuanced understanding of how the EFP stack stands in comparison to the ELK stack concerning critical performance parameters. The analysis presents

empirical data to decipher how these stacks respond to read-and-write operations, shedding light on potential advantages that can guide system architects and stakeholders in making informed decisions for optimal data handling within their infrastructural framework. The below experiments use the GitHub project link [16] for EFPS Stack and the GitHub project link [17] for ELK Stack.

The study meticulously evaluates the Read to Write Ratio, a pivotal metric indicative of the system's responsiveness in managing read and write operations. The analysis unveils that the Read to Write Ratio of the EFP stack stands at an approximate average of 3.1, notably surpassing that of the ELK stack, having that slightly above 2.5 (see Fig. 3). This observation underscores a superior response time and efficiency of the EFP stack in handling data transactions, a critical factor in contemporary data processing landscapes.

Additionally, the investigation probes into the Write Count per Second, another critical metric signifying the throughput capacity of the stacks concerning write operations. The empirical data demonstrates that the EFP stack exhibits a significantly higher Write Count per Second, exceeding 10,000, as opposed to the ELK stack, which registers a count hovering around 9,800 (see Fig. 4). These findings underline the enhanced throughput and operational capacity of the EFP stack, substantiating its viability for high throughput data processing requirements. This comparative analysis offers valuable insights for system architects and stakeholders, aiding in informed decision-making to optimize data handling within their infrastructural framework.

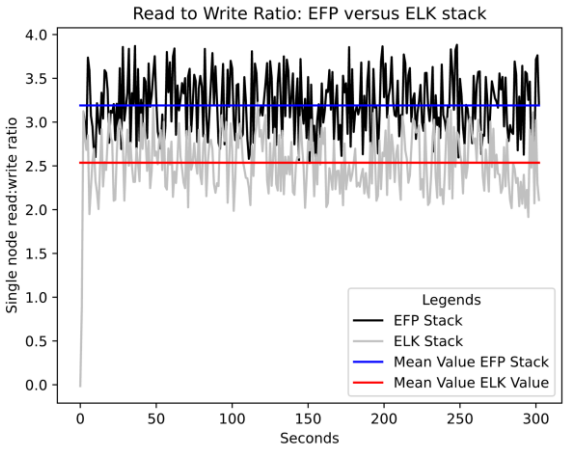

Fig. 3. Graph Comparison of Read to Write Ratio of (Elastic faust)EFP stack and ELK with computed mean value over the experiment of EFP stack higher by a difference of 0.5+ points when compared with ELK stack for up to 5 minutes of log analysis.

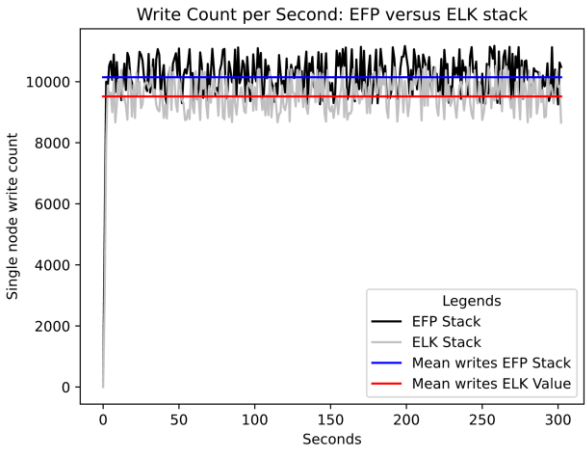

Fig. 4. Graph comparing write count per second of EFP stack and ELK stack showing a slightly better 'write count for EFP Stack for up to 5 minutes of analyzing logs.

V. Conclusion

The research endeavors to shed light on the ever-increasing significance of log analysis in the realm of web server management. The continuous evolution of cyber threats and rising user expectations necessitate organizations to harness the potential of log data for identifying vulnerabilities, enhancing website performance, and delivering superior user experiences. The study embarked on bridging the gap between the burgeoning volume of log data and actionable insights, aiming to facilitate data-driven decision-making for businesses navigating the dynamic digital landscape. Throughout this work, a meticulous exploration of various tools and technologies has been undertaken, with a focus on leveraging the EFP stack's potential in comparison to the ELK stack. The evaluation encompassed critical performance metrics, such as Read to Write Ratio and Write Count per Second, unveiling notable advantages of the EFP stack in terms of responsiveness and throughput. These insights provide valuable guidance to system architects and stakeholders, empowering them to make informed decisions for optimal data processing within their infrastructural framework.

The challenges encountered in dealing with large volumes of log data, real-time processing demands, and the need for accurate data transformation were systematically addressed. The research also highlighted the imperative for insightful visualization and interpretation, along with the complexity of data integration with diverse sources. Additionally, the advantages encompassed the ability to identify popular and underperforming web pages, monitor website security, and gain valuable insights for data-driven decision-making and business growth. In essence, this research aims to catalyze advancements in log analysis techniques and technologies, enabling organizations to navigate the complex web technology landscape with enhanced security, efficiency, and user-centricity. The insights and methodologies presented herein contribute to the broader realm of web server management and data analytics, providing a stepping stone for future advancements and fostering a data-driven approach to web technology and its ever-evolving challenges.

VI. References

[1] Mayur Mahajan, Omkar Akolkar, Nikhil Nagmule, Sandeep Revanwar. "Real-Time Web Log Analysis and Hadoop for Data Analytics on Large Web Logs." International Research Journal of Engineering and Technology (IRJET), Volume 03, Issue 03, March 2016, Page 1827. doi:10.1016/0022-2836(81)90087-5.

[2] Dr. S. Suguna and M. Vithya. "Analysis of Web Logs Using Big Data Tools." International Journal of Advanced Research Trends in Engineering and Technology (IJARTET), Vol. 3, Special Issue 20, April 2016. do: 10.1007/11823285_121.

[3] Mostafa Mohamed Shendi, Hatem Mohamed Elkadi, Mohamed Helmy Khafagy. "A Study on the Big Data Log Analysis: Goals, Challenges, Issues, and Tools." International Journal of Soft Computing and Artificial Intelligence, ISSN: 2321-404X, Volume-7, Issue-2, Nov-2019.

[4] Sohan Panwar and Garima Silakari Tukra. "Analysis of Web Server Log File Using Hadoop." International Journal for Research in Applied Science & Engineering Technology (IJRASET), Volume 6 Issue IV, April 2018. doi:10.1109/HPDC.2001.945188R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[5] Mudassir Khan and Shakila Basheer. "Using Web Log Files: The Comparative Study of Big Data with Map Reduce Technique." 2020 International Conference on Intelligent Engineering and Management (ICIEM), IEEE, 2020.

[6] Savitha K and Vijaya MS. "Mining of Web Server Logs in a Distributed Cluster Using Big Data Technologies." International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 5, No. 1, 2014.

[7] Lukasz Korzeniowski and Krzysztof Goczyła. "Landscape of Automated Log Analysis: A Systematic Literature Review and Mapping Study." IEEE Access, Digital Object Identifier 10.1109/ACCESS.2022.3152549, February 2022.

[8] Son, S. J., & Kwon, Y. (2017). "Performance of ELK stack and commercial system in security log analysis." 2017 IEEE 13th Malaysia International Conference on Communications (MICC). doi:10.1109/micc.2017.8311756.

[9] Prakash, T., Kakkar, M., & Patel, K. (2016). "Geo-identification of web users through logs using ELK stack." 2016 6th International Conference - Cloud System and Big Data Engineering (Confluence)". doi:10.1109/confluence.2016.7508191.

[10] Bhatt, K., Saxena, A., & Singh, K. (2020)." Implementation of Big-Data Applications Using Map Reduce Framework." International Journal of Engineering and Computer Science, 9(08), 25125-25131. ISSN: 2319-7242. DOI: 10.18535/ijecs/v9i08.4504.

[11] Elastic Filebeat Refrences, https://www.elastic.co/guide/en/beats/filebeat/master/index.html, last accessed 2023/09/09.

[12] Apache Kafka Documentation, https://kafka.apache.org/documentation/, last accessed 2023/09/03.

[13] Faust Notes, https://faust.readthedocs.io/en/latest/, last accessed 23/08/18.

[14] Apache Pinot website, https://pinot.apache.org/, last accessed 23/09/15.

[15] Apache Superset website, https://superset.apache.org/docs/intro/, last accessed 23/09/15.

[16] EFPS Stack git project, https://github.com/swiftcynic/EFPS-Stack, last accesses 23/09/15

[17] ELK Stack git project, https://github.com/OjasKarmarkar/Flight-Analysis-Big-Data, last accesses 23/09/15