Journal of Artificial Intelligence and Capsule Networks (ISSN: 2582-2012)
www.irojournals.com/aicn/

Real-Time Web Server Monitoring System using Python

Parvathy M[1], Antony Balasingam J[2], Sanjith E.S[3]

[1]Assistant professor, [2,3]Students, Department of Information Technology, Velammal Engineering College, Chennai, India.

Email: [1]parvathy@velammal.edu.in

Abstract

The Web Server Monitoring System, developed using Python, is a comprehensive solution for real-time tracking of essential server performance metrics, including memory usage, CPU usage, and response time. By using Python's psutil library for system monitoring and Flask for creating a user-friendly dashboard, the system provides administrators with timely insights into the health and efficiency of their web servers. By periodically querying the target servers, the system collects and processes data on memory and CPU usage, enabling proactive identification of resource bottlenecks and potential performance issues. Additionally, the system measures response time to assess server responsiveness, facilitating prompt detection of latency issues or performance degradation. The user-friendly web-based dashboard allows administrators to easily interpret the collected metrics and track performance metrics over time. With its lightweight and efficient design, the Web Server Monitoring System empowers administrators to optimize server resources, troubleshoot issues promptly, and ensure uninterrupted user experiences.

Keywords: Web Server, Python, URL (Uniform Resource Locator), CPU usage, Memory usage, Response Time

1. Introduction

The efficiency and dependability of web servers are essential in the current digital era for guaranteeing perfect user experiences. As companies depend more and more on web-based tools to interact with their customers. It becomes essential to maintain and monitoring these

DOI: https://doi.org/10.36548/jaicn.2024.3.006
Received: 12.07.2024, received in revised form: 14.08.2024, accepted: 27.08.2024, published: 11.09.2024
© 2024 Inventive Research Organization. This is an open access article under the Creative Commons Attribution-NonCommercial International (CC BY-NC 4.0) License

servers' performance. To improve the performance of a specified URL, this study presents a web server monitoring system that tracks metrics including CPU utilization, memory usage, and response time. [1].

The rapid advancement of technology necessitates effective methods for managing and improving server performance. This research addresses the need for a reliable monitoring system that can offer real-time insights into important performance metrics by integrating Python-based tools and procedures. Understanding and improving CPU and memory utilization is essential for maintaining optimal server performance and resource allocation in the development of web applications and services [2]. Furthermore, monitoring the response time is essential for evaluating server performance and spotting any bottlenecks that can affect user experience.

The purpose of this research is to develop a monitoring system that provides administrators with a thorough overview of the performance of their web server, with an emphasis on the assigned URL. Administrators can learn a great deal about resource utilization, system health, and performance trends by regularly monitoring CPU, memory, and response time. Using a user-friendly dashboard improves accessibility further and makes it possible for administrators to effectively analyse and interpret performance metrics.

This web server monitoring system provides a scalable and flexible way to monitor and optimize server performance by utilizing the efficiency and adaptability of Python [3]. Administrators can proactively fix performance issues and provide an optimum user experience by integrating monitoring functions that are specifically customized to the selected website URL. This guarantees focused insights. The aim of the study is to improve web server monitoring methods and provide administrators with the tools necessary to keep their web-based services operating at optimal efficiency and reliability. [4].

2. Related Work

An architecture for proactively supervising and controlling data center operations is being developed at LBNL's National Energy Research Scientific Computing Center (NERSC). The Operations Monitoring and Notification Infrastructure (OMNI) has a novel design that combines data from metrics, sensors, and analytics engines with cutting-edge technologies like Kubernetes, Prometheus, Grafana, and other predictive platforms. This extensive infrastructure will enable central orchestration, automated analysis, and thresholding of alarms to discover

key concerns. It will also assist in the management of NERSC's computing system deployments and the future Perlmutter HPC system. [5]. Torkel Ödegaard created Grafana, an open-source application for time-series data querying, visualization, and alerting, in 2014. It enables alerting, dashboard display, and querying from different data sources and SQL databases [6]. Performance management and monitoring are essential in cloud networking environments to guarantee efficiency in resource utilization, maintain service availability, and identify unexpected bottlenecks that may affect the performance of applications. In addition to providing real-time insights to detect and rectify issues pro-actively, these methods allow organizations to monitor network latency, throughput, and error rates, ensuring that service-level agreements (SLAs) are met. Prometheus and Nagios, AI-driven analytics, and Kubernetes and Docker Swarm are important tools used in the study for monitoring the cloud environment. By maximizing resource use, guaranteeing reliable service delivery, and enhancing operational efficiency, these strategies produce improved performance and resilience [7]. This research explores big data monitoring, emphasizing the different forms, elements, and techniques of proactive and reactive monitoring. To facilitate effective big data monitoring techniques, it addresses several monitoring systems, including DataDog, SequenceIQ, Sematext, Apache Chukwa, Nagios, Ganglia, DMon, and SmartMonit [8]. This research discusses the use of Python and the AI algorithms in monitoring the service delay, improper resource utilization, and unexpected bottlenecks that occur due to denial-of-service attacks and distributed denial of service attacks. The existing works emphasize the need for the server monitoring system and its multiple benefits. To enable the organization to have a comprehensive overview about the web application performance, optimize server resources, troubleshoot issues promptly, and ensure uninterrupted user experiences, the proposed method offers a web monitoring service using Python. This approach utilizes Python’s psutil library for system resource monitoring and Flask for creating a lightweight web interface, providing real-time insights into CPU and memory usage while enabling proactive alerts and resource optimization strategies [9,10].

3. Proposed Work

The proposed work plan for the web server monitoring system involves using appropriate Python tools for data collection and visualization, Python scripts for gathering CPU usage, memory usage, and response time data, and Flask for creating a user-friendly dashboard.

3.1 Proposed Model

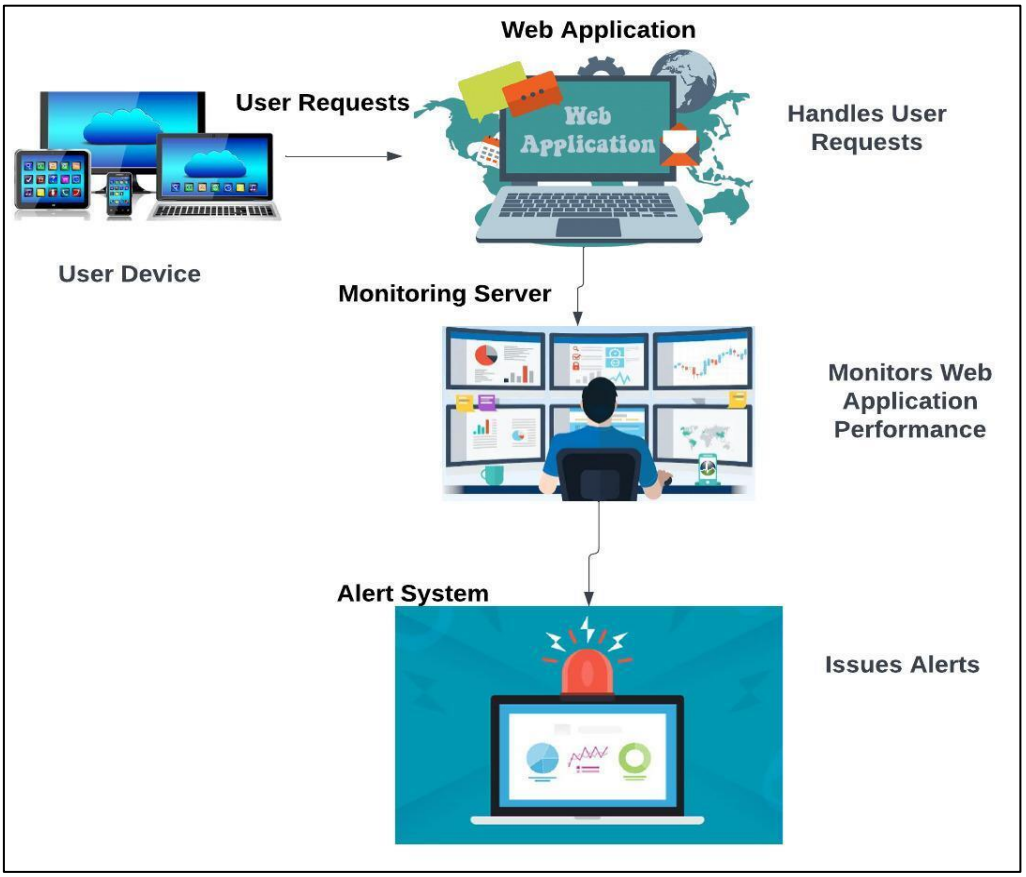


Figure 1. Workflow

Figure 1 illustrates the workflow of the proposed system. The user can access the web application using any one of the devices such as the computers, laptops, or smartphones. The web application, hosted on a server is continuously monitored to track its health and the performance. The monitoring severs collects the information's about the application performance and system resources. The gathered data is processed and analysed in real time, focusing on the performance metrics such as CPU usage, memory usage, responses time. The alert system generates notification whenever the performance metric exceeds the predefined threshold. The monitoring server continuously monitors the system at regular intervals and sends notification in case of any performance degradation, thus enabling the administrator to optimize resources, troubleshoot issues quickly, and maintain high-quality user experiences.

The proposed web server monitoring system was implemented using a Python script to continuously monitor the system health and performance, and send notifications to the administrator whenever necessary.

4. Results and Discussion

The proposed system uses the Python's psutil (Python System and Process Utilities), an efficient cross platform library for developing the monitoring tools and collecting the performance metrics like CPU usage, memory usage etc. The psutil.cpu_percent(interval=x), function measures CPU usage over an interval specified by x. The psutil.virtual_memory () function provides the details about memory usage. The data collected at predefined intervals are stored in an SQLite database to monitor and identify the anomalies. The libraries such as Pandas, NumPy, SciPy, and Statsmodels are used for processing and analysing the data collected. The user-friendly dashboard using Flask and the notification alert are under development. The Figure 2 shows the CPU usage, memory usage and the response time observed with one user on Zoho.

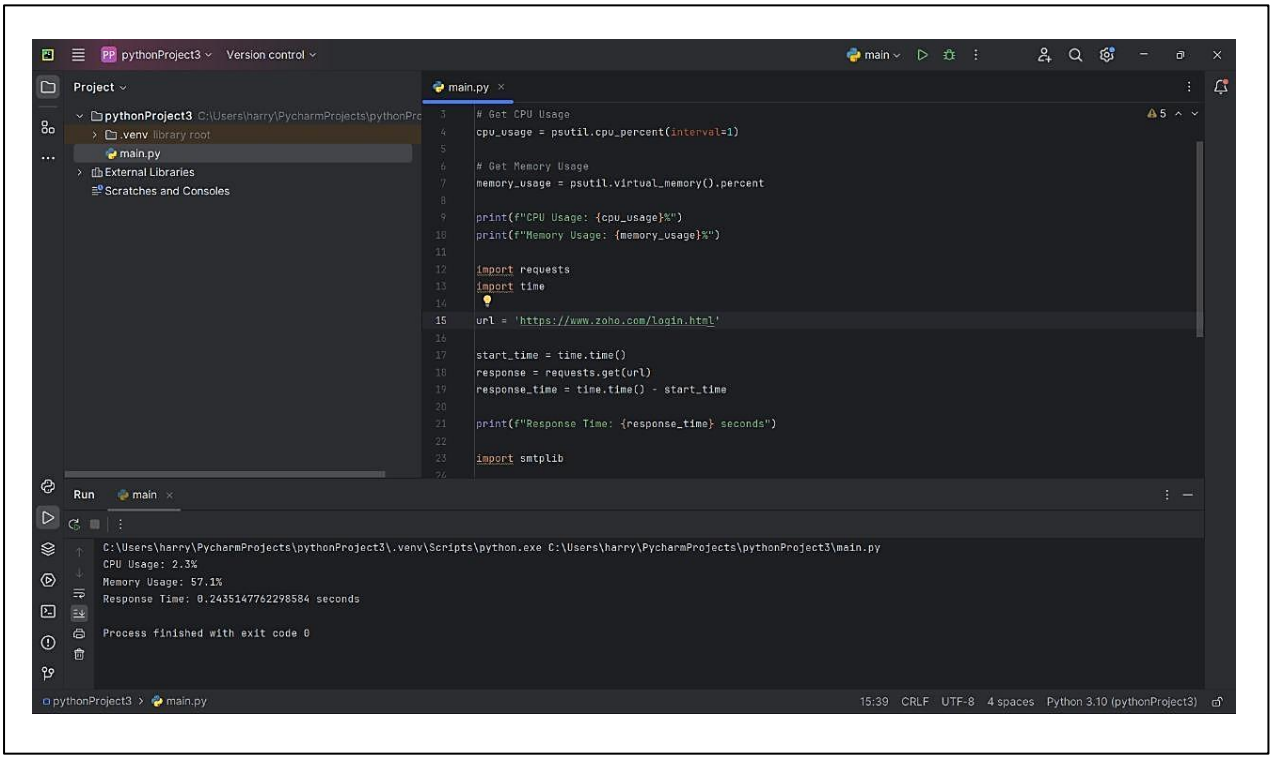


Figure 2. Web Server with One User

Figure 3 illustrates the CPU usage, memory usage and the response time observed with multiple users on Flipkart.

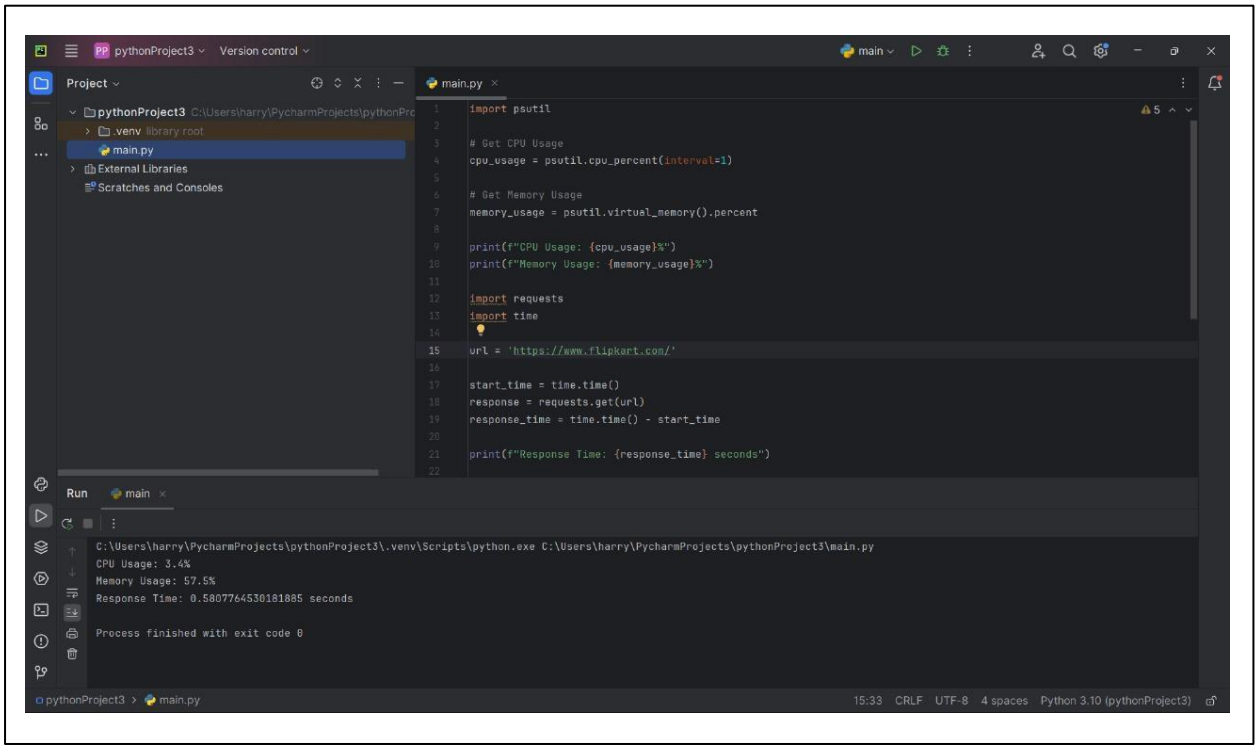


Figure 3. Web Server with Multiple User

Figure 4 depicts the difference in response time observed on Zoho and Flipkart.

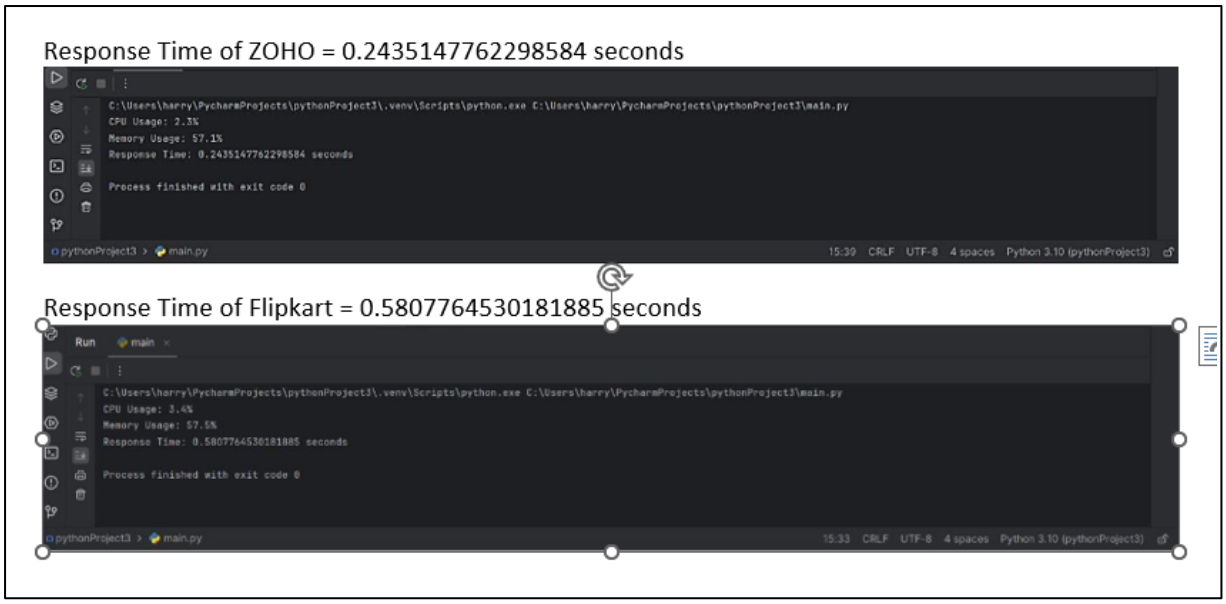
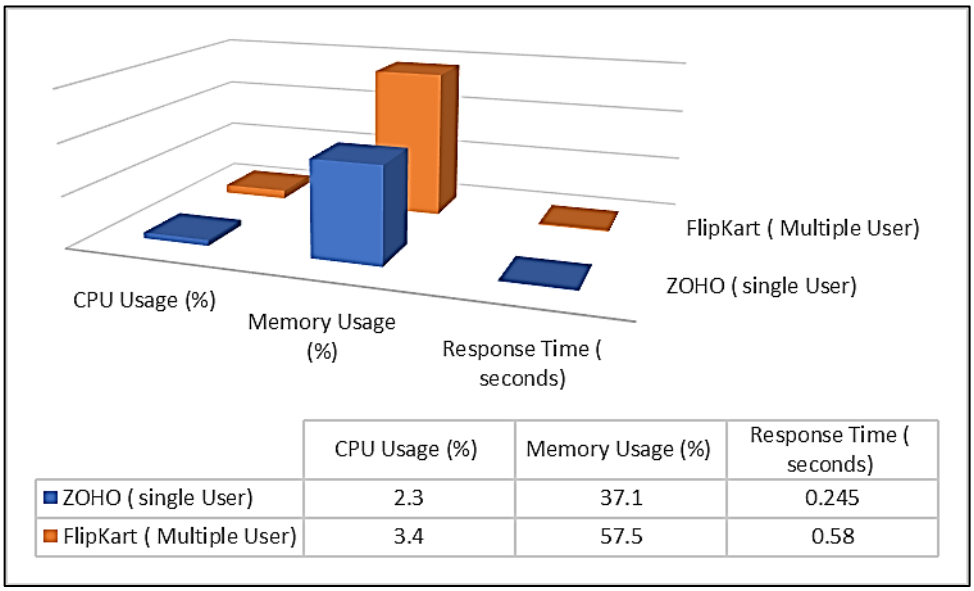


Figure 4. Difference of Response Time

The Figure 5 below depicts the graphical representation of the CPU usage, memory usage, response time observed on Zoho and Flipkart.



	CPU Usage (%)	Memory Usage (%)	Response Time (seconds)
ZOHO (single User)	2.3	37.1	0.245
FlipKart (Multiple User)	3.4	57.5	0.58

Figure 5. Comparison Chart

The CPU usage, memory usage, and the response time on the Zoho website is slightly lower compared to those on the Flipkart website. This shows that accessing the Flipkart required more system resources, possibly due to a larger response size or higher processing requirements, as there were multiple users accessing it simultaneously. Therefore, the results illustrate the performance metrics of two different websites under varying conditions.

5. Conclusion and Future Work

The web server monitoring system research has successfully developed a robust solution for tracking essential metrics such as CPU usage, memory usage, and response time of web servers using Python's psutil library. The code has been tested on two different websites, Zoho and Flipkart, under different conditions. The future work will focus on the successful completion of the work with an email notification in case of performance degradation and a user-friendly dashboard developed using Flask.

References

[1] Hu, Yiming, Ashwini Nanda, and Qing Yang. "Measurement, analysis and performance improvement of the Apache web server." In 1999 IEEE International Performance, Computing and Communications Conference (Cat. No. 99CH36305), Scottsdale, AZ, USA IEEE, 1999. pp. 261-267.

[2] Phaltane, Saurabh, Omkar Nimbalkar, Piyush Sonavle, and S. R. Vij. "Apache Web Server Monitoring." International Journal of Scientific & Engineering Research 4, no. 7 (2013): 2195-2199.

[3] E. D. Katz, M. Butler, and R. McGrath, “A scalable web server: TheNCSA prototype, in WWW’94 Conference Proceedings,1994.

[4] Bestavros, Azer, Robert L. Carter, Mark E. Crovella, Carlos R. Cunha, Abdelsalam Heddaya, and Sulaiman A. Mirdad. "Application-level document caching in the internet." In Second International Workshop on Services in Distributed and Networked Environments, Whistler, BC, Canada IEEE, 1995. pp. 166-173

[5] Sukhija, Nitin, and Elizabeth Bautista. "Towards a framework for monitoring and analyzing high performance computing environments using kubernetes and prometheus." In 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Leicester, United Kingdom IEEE, 2019. pp. 257-262.

[6] Chakraborty, Mainak, and Ajit Pratap Kundan. "Grafana." In Monitoring cloud-native applications: Lead agile operations confidently using open source software, Berkeley, CA: Apress, 2021. pp. 187-240.

[7] Khalil, Mahmoud. "Comprehensive Tools and Techniques for Performance Monitoring and Management in Cloud Networking Environments." Advances in Computer Sciences 7, no. 1 (2024): 1-7.

[8] Demirbaga, Ümit, Gagangeet Singh Aujla, Anish Jindal, and Oğuzhan Kalyon. "Big Data Monitoring." In Big Data Analytics: Theory, Techniques, Platforms, and Applications, Cham: Springer Nature Switzerland, 2024. pp. 155-170.

[9] M Nalayini, C., and Jeevaa Katiravan. "Detection of DDoS Attack Using Machine Learning Algorithms." Journal of Emerging Technologies and Innovative Research (JETIR) , no. 7 (2022).f223-f232

[10] Vogel, Patrick. "A dashboard for automatic monitoring python web services." PhD diss., Faculty of Science and Engineering, 2017. https://fse.studenttheses.ub.rug.nl/15605/1/bsc-thesis3.pdf