SYSTEM MONITORING TOOL

Swapnil Mukherjee
Clemson University
Clemson, SC, USA
swapnil@clemson.edu

ABSTRACT

This paper presents a system monitoring tool that uses the Psutil library in Python to obtain real-time metrics for CPU, memory, disk, and network usage. The tool displays the collected data using Matplotlib, allowing the user to observe the system's resource utilization over time. The system was developed and tested on a SEED LABS Ubuntu 20.04 system. This paper outlines the project's background, motivations, methodology, analysis, and future work.

ACM Reference Format:
Swapnil Mukherjee. 2023. SYSTEM MONITORING TOOL. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

As computer systems become more complex and applications more resource-intensive, understanding system performance and resource utilization becomes increasingly important. In many cases, users can suffer from sluggish or unresponsive systems due to inefficient use of system resources. Therefore, it is important to have a system monitoring tool that provides real-time performance metrics to allow for informed decisions regarding system optimization and management. This paper presents a system monitoring tool that was developed using Python and Matplotlib to visualize real-time CPU, memory, disk, and network usage data.

2 BACKGROUND

The system monitoring tool was developed using the Psutil library in Python, which provides a simple interface for retrieving system performance metrics. The library supports various operating systems, including Linux, Windows, and macOS, making it a suitable choice for this project. I also used Matplotlib, a Python library for creating data visualizations, to plot the collected data.

Psutil is a cross-platform library for retrieving information about system utilization, such as CPU, memory, disks, network, and process information. It provides a unified API for querying information on all supported platforms, making it an excellent choice for this project. Matplotlib, on the other hand, is a powerful visualization library that allows for the creation of high-quality plots in Python. It can be used to create a wide range of visualizations, from simple line graphs to complex 3D visualizations.

3 MOTIVATIONS AND OBJECTIVES

The motivation for this project was to create a system monitoring tool that would allow for the observation of system resource utilization over time. This tool should be easy to use, informative, and could be deployed on various systems. By providing real-time data on CPU, memory, disk, and network usage, users can identify system bottlenecks and make informed decisions about system optimization and management.

The primary objective was to develop a system monitoring tool that could collect real-time performance metrics for CPU, memory, disk, and network usage and display this information in an informative and easy-to-understand way. The tool should be easy to deploy and should work on multiple platforms.

4 METHODOLOGY / DESIGN

The system monitoring tool was developed using Python and Matplotlib. The Psutil library collected real-time performance metrics for CPU, memory, disk, and network usage. Matplotlib was then used to plot the collected data on a graph. The tool updates the graph every second to monitor system resource utilization.

The user interface of the system monitoring tool is minimalistic and easy to use. The main window displays the current usage of CPU, memory, disk, and network resources in real-time. It also displays a line graph that shows the usage percentage of each resource over time.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

The resulting plot allows the user to observe trends and changes in system utilization. The tool also provides a feature to save the generated graph as an image file.

5 USER MANUAL

To use the system monitoring tool, the first step is to install Python and Matplotlib on your system. To install Python, download the installer from the official Python website and run it on your system. Make sure to select the appropriate version for your operating system. Once Python is installed, you can install Matplotlib by opening the command prompt or terminal and running the command "pip install matplotlib". This will install the necessary dependencies for the tool to run properly.

Next, you can download the System Monitoring Tool from the GitHub repository by following this URL: https://tinyurl.com/SystemMonitoringTool. Once downloaded, navigate to the directory where the tool is saved using the command prompt or terminal. To run the tool, enter the command "python monitor.py" in the command prompt or terminal. This will start the tool and display a graph showing the real-time utilization of CPU, memory, disk, and network resources.

The graph will update every second to show the current resource utilization. You can observe trends and changes in system utilization over time to make informed decisions about system optimization and management. The graph shows the usage percentage of each resource over time, allowing you to easily identify system bottlenecks and take corrective action if necessary.

To stop the tool, press Ctrl + C in the command prompt or terminal. The tool will terminate, and the graph will no longer be displayed. Overall, the system monitoring tool provides a straightforward and informative way to monitor system resource utilization in real time, allowing you to make informed decisions about system optimization and management. Note: The system monitoring tool requires Python and Matplotlib to be installed on your system. Please ensure that these software are installed before using the tool. Additionally, the tool may require administrative privileges to access system metrics on some operating systems.

6 ANALYSIS / RESULTS

The system monitoring tool was tested on a SEED LABS Ubuntu 20.04 system, and the results were as expected. The tool collected real-time CPU, memory, disk, and network usage data and displayed this information on a graph using Matplotlib. I could observe the system's resource utilization over time and make informed decisions about system optimization and management.

The resulting graph allowed me to quickly identify areas of concern, such as spikes in CPU usage, and make changes to address them. By monitoring system utilization over time, I could identify trends and make informed decisions about system optimization and management.

7 FUTURE WORK

In future work, there are several areas that can be improved upon in the system monitoring tool. One of the areas that can be improved upon is the addition of more system metrics. In particular, it would be beneficial to include GPU utilization, component temperature, and CPU/GPU voltage in the monitoring tool. By adding these metrics, system administrators can better understand the overall health of their systems and identify potential issues before they cause problems.

Another area that can be improved upon is the collection and display of historical data. Currently, the tool only displays real-time system resource utilization data. However, by capturing and displaying historical data, system administrators can observe trends and patterns in system utilization over time. This can provide valuable insights into system behavior and help identify potential issues before they become critical.

Additionally, it would be useful to include notifications and alerts in the tool. By setting thresholds for system metrics, such as CPU or memory usage, the tool can alert administrators when these thresholds are exceeded. This can help administrators respond quickly to potential issues and prevent system downtime or failure.

Finally, it would be beneficial to develop a web interface for the system monitoring tool. By developing a web interface, system administrators can access real-time system resource utilization data from any device with an internet connection. This can provide greater flexibility and accessibility, making it easier for administrators to monitor and manage their systems.

8 CONCLUSION

In conclusion, the system monitoring tool developed using Python and Matplotlib provides a simple, informative way to observe real-time system resource utilization. By collecting and displaying CPU, memory, disk, and network usage data, I can identify system bottlenecks and make informed decisions about system optimization and management. The tool provides a valuable contribution to the field of system administration and can be used to improve system performance and stability.

9 ACKNOWLEDGEMENTS

I would like to acknowledge the SEED LABS team for providing the Ubuntu 20.04 system used for testing the system monitoring tool. I would also like to thank the Python and Matplotlib development teams for their contributions to the development of the system monitoring tool.

10 REFERENCES

I would like to thank the following individuals and organizations for their contributions and support:

(1) Giampaolo Rodola, "psutil - a cross-platform process and system utilities module for Python". https://github.com/giampaolo/psutil
(2) Matplotlib Development Team, "Matplotlib: A 2D Graphics Environment". https://matplotlib.org/
(3) Python Software Foundation, "Welcome to Python.org". https://www.python.org/
(4) SEED Labs, "Ubuntu 20.04". https://seedsecuritylabs.org/lab_env.html
(5) Wong, K., and Kwok, T. (2017). A review on system monitoring using python. International Journal of Electrical and Computer Engineering, 7(4), 1971-1978.