Matthew Veroutis Journal #2

Date:

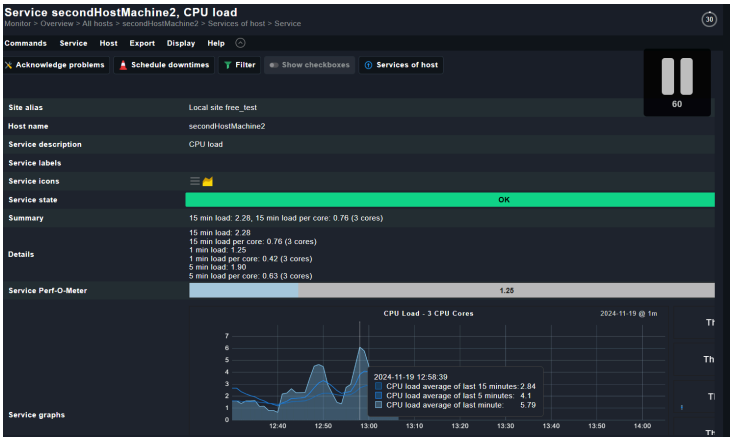
2024-11-19:

In class we we were shown how to use bash more in-depth, so this inspired me to start working on the bash script that would hog the CPU. I didn't know how to make a script that would hog the CPU so I asked Chatgpt to make one. I also asked it countless times so it would provide a script that I understand and fits the requirements.

```
#!/bin/bash

 echo "Starting CPU hogging processes for 60 seconds"

timeout 240s bash -c 'while :; do :; done' &
timeout 240s bash -c 'while :; do :; done' &
timeout 240s bash -c 'while :; do :; done' &

wait

echo "Finished hogging CPU after 60 seconds."
```

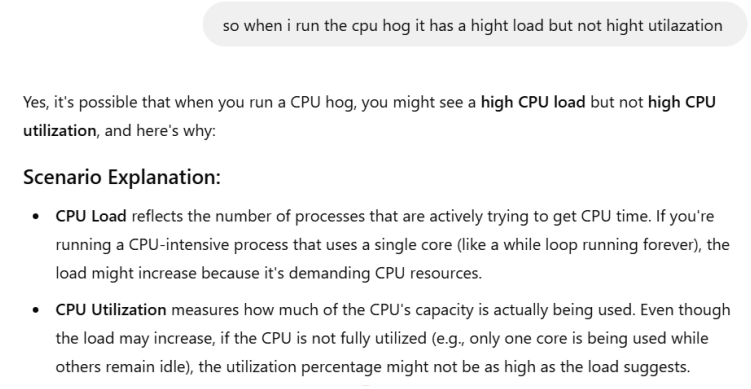
When I initially tried this it would do a while loop on each core for 240 seconds which would put the load on each core but the system tended to be very laggy and the monitoring system was unable to update. So I later removed one of the “timeout 240s bash -c 'while :; do :; done' &” lines so it only runs on two cores and it was much more stable.

This image bellow shows the results:



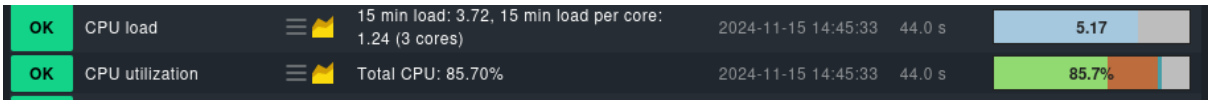
So when I ran this script it peeked at a little under 6 which is pretty high since the normal when the machine is around 1. Interestingly the CPU usage did not spike that much. So after

research on ChatGPT I found out it could be normal since:



2024-11-20:

I did some more research for other tools that I could use to stress the CPU so the CPU load and CPU utilization both go very high. Luckily I came across this website: https://www.tecmint.com/linux-cpu-load-stress-test-with-stress-ng-tool/, which demonstrated a package that would allow you to stress test multiple components such as CPU and RAM. Since it has the option to stress the CPU it was perfect for the job. It was also quite easy to install and use. To install it, I just had to do “sudo apt install stress”, and to run it to stress 3 cores I did “stress --cpu 3 --timeout 300s”. It clearly worked since the image below shows one of the peeks of CPU usage and load:

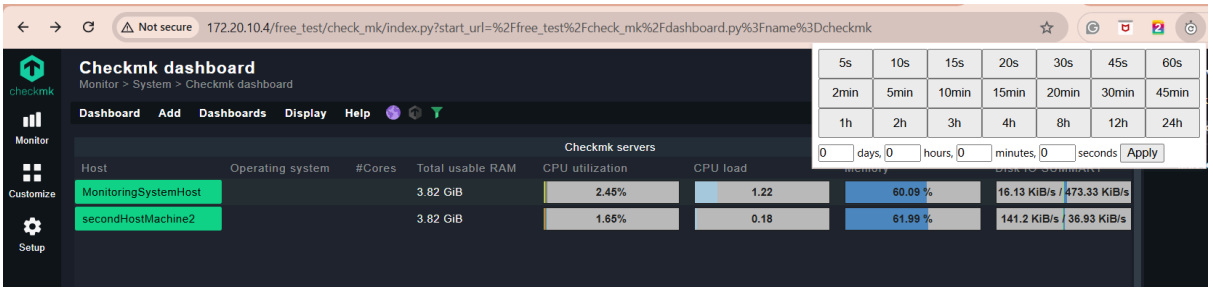


The system was also still able to send updates to the system monitor even with the three CPU hogs.

To stress out the cpu we will use the stress tool due to its excellent ease of use and performance.

2024-11-21:

I also noticed on the webpage that shows all the hosts it does not automatically reload so the data from the machines may be old, I thought of a solution which was to install a Chrome browser extension that would automatically reload the browser. I decided to use “Tab Auto Reloader” which is open source and automatically reloaded the website at any interval:



The image above shows the extension with all the possible times, and this is the webpage with all the machines and their statistics.

Instructions:

Installing and using Stress to to stress the CPU:

1. sudo apt install stress

This installs the stress package to your machine

2. stress --cpu 3 --timeout 300

This stresses the CPU in specific, and it stress all 3 core, and it only stresses it out for 300 second so 5 mins.