**NSSA-220 Project 1: Application Performance Monitoring**

**Vincent Son, Joey Guarino, Amir Elias**

**Introduction**

Write a 3-4 sentence introduction that describes what this project was about. Use the project slides as your guide. I want to see you describe the project in your own words.

This project aims to develop Application Performance Monitoring (APM) tool that will monitor 6 executables provided by the professor. The tool will collect process level and system level metrics at specific rates, capturing CPU and memory utilization, network bandwidth, and hard disk access and availability. The captured metrics will be written into csv and then properly kill applications at 15 minutes. The csv files is then used to generate plots for analysis in the report.

**Process Level Metrics**

Describe what the CPU utilization plot shows in 2-3 sentences.

APM1 and APM3 show peak utilization at the beginning of the metrics, before dropping off and stabilizing at 35% and 70% CPU utilization respectively. APM4 have very low CPU utilization averaging 2%. APM2 and APM6 show 0% utilization.

Describe what the memory utilization plot shows in 2-3 sentences.

APM1, APM2, APM3, and APM 4 remained at 0% utilization throughout the plot. Meanwhile, APM5 and APM6 show abrupt peaks at spaced interval. However in APM5, the memory utilization on APM6 are gradually increasing.

**Potential things to write about**: Which processes used the most CPU/memory? Which processes used the least CPU/memory? Did any processes have any interesting patterns in their CPU and/or memory utilization? Could you see a memory leak (memory use that only increased over time) in any of the processes?

**System Level Metrics**

Describe what the network bandwidth utilization plot shows in 2-3 sentences.

Excluding the data captured at 0 second since RX data start at 56462 kb/s, the RX data rate (receive) constantly peak every few seconds above 1000 kb/s. The TX data rate (transmit) mostly stayed at 0 kb/s throughout the capture.

Describe what the hard disk access rates plot shows in 1-2 sentences.

The disk write start around 4000 kb/s at the beginning and gradually increased to 5000 kb/s at the end.

Describe what the hard disk utilization plot shows in 1-2 sentences.

The hard disk utilization shows that its availability is decreasing over time until it goes back to it’s initial amount at around 840 seconds.

**Potential things to write about**: How similar (or not) were the transmit and receive data rates? Could you see any patterns in how the data rates changed over time? Were there any interesting patterns in the hard disk access rates or hard disk utilization?

The transmit and receive data rates differ significantly since transmit data rates constantly changes and peak every few seconds. The receive data rates mostly stayed flat with very small data rate sparsely scatter across that plot. There is notable peak in transmit data rate at around 800 seconds that is higher than the rest.

**Summary and Lessons Learned**

Write 3-4 sentences that describes whether or not the VM you used had enough computing resources (CPU, memory, network capacity, and disk) to handle the mix of application processes that were running and what lessons you learned while working as a team on this project.

For our initial shell running in the VM, we observed that the CSV size was smaller than expected. This led us to believe that our shell isn't optimized for such tasks. We identified several factors that might be contributing to this issue; running the ps command for each PID in the APM, which might be resource intensive. The use of the sleep command combined with the overhead sometimes results in a delay of 1-2 seconds in $SECONDS. The sequential processing forces one tool to wait for another to complete, leading to discrepancies in the CSV data by a few seconds.

In our refactored shell, we introduced some significant changes. One primary change is executing the ps command once per loop, allowing us to collect multiple PID metrics simultaneously. The main function now operates tools or functions asynchronously, using the & command. The sleep 1 function runs at the start of the loop and before the loop ends, the system waits for the sleep 1 to conclude, which reduces the overhead notably.

While there are still instances where some data isn't captured, there are still improvement compared to the original shell. For instance, the initial shell missed data for approximately 5 process metrics and 50 system metrics, whereas the refactored shell had missed 0 process metrics and 4 system metrics.