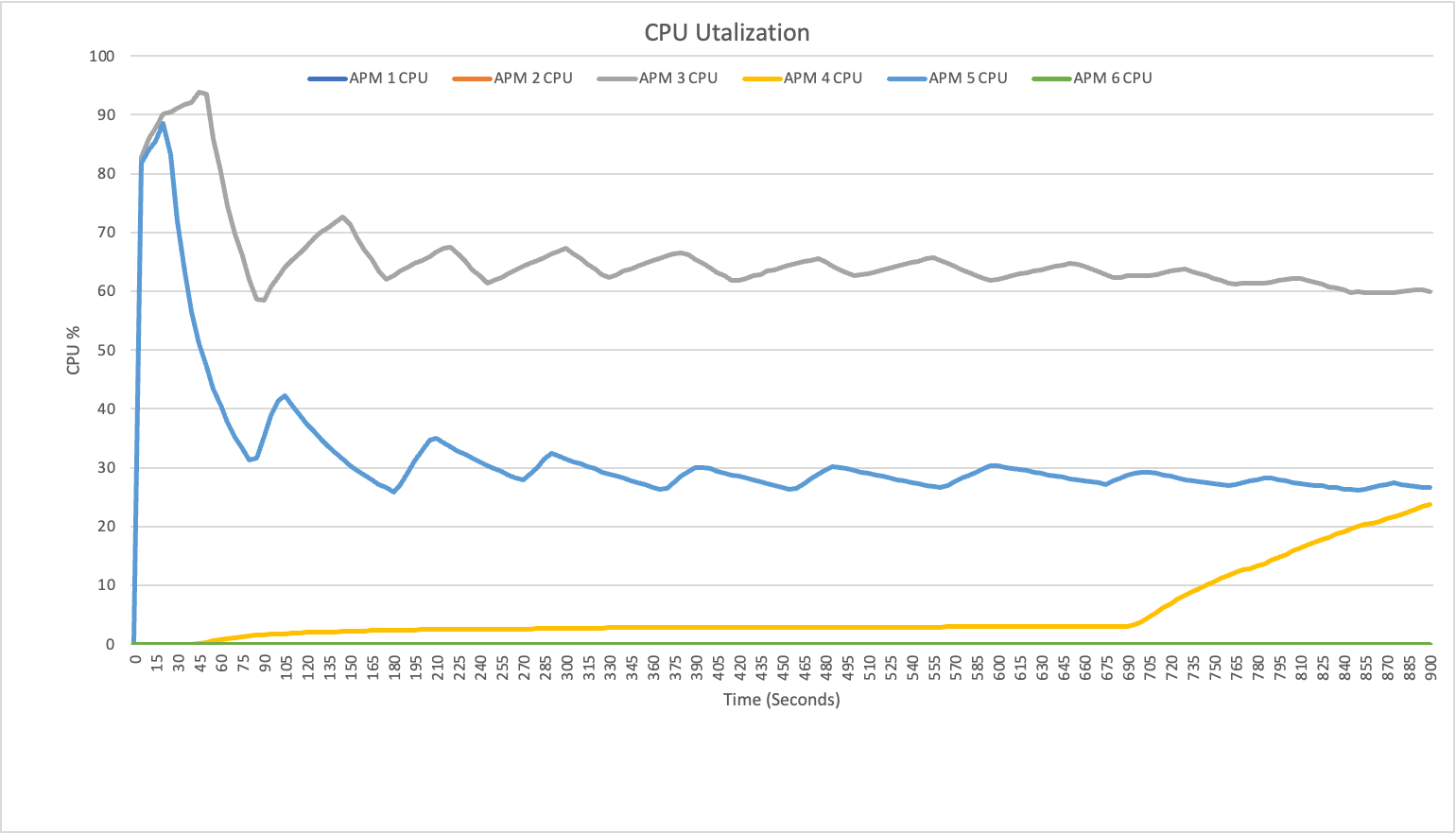
**NSSA-220 Project 1: Application Performance Monitoring**

**Johnny Schnaufer, Hunter Ross, Adam Stout**

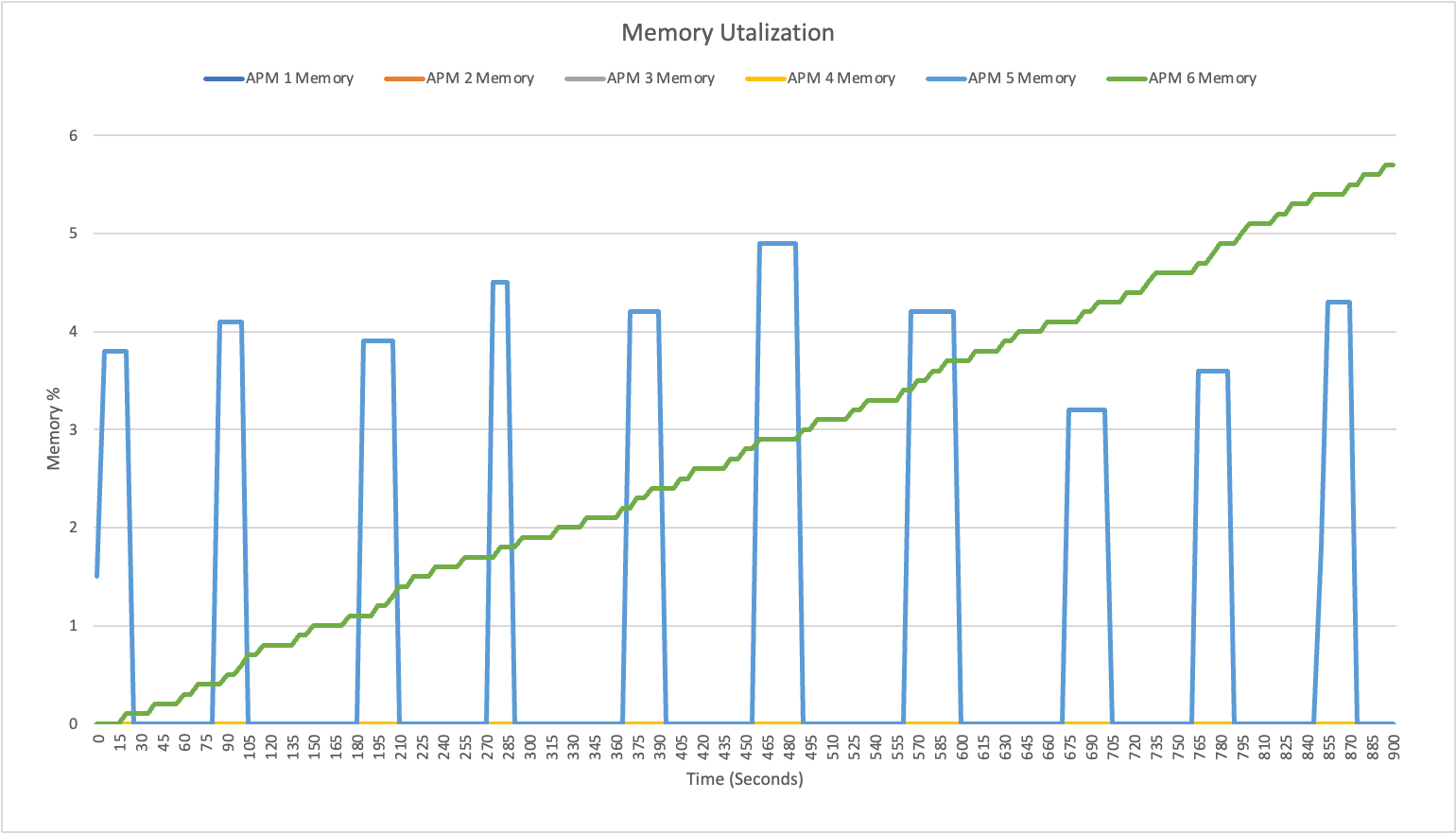
**Introduction**

This project was an introduction on how to perform Application Performance Monitoring by creating an APM tool in bash. The tool starts the processes, collects performance metrics over a fifteen minute interval, and kills any processes spawned at the end of the interval. The tool monitored system level metrics, network level metrics, network bandwidth utilization metrics, hard disk access rates, and hard disk utilization metrics. There are six different pre-compiled C executables used to monitor for the purposes of this assignment. The tool outputs the results from monitoring for fifteen minutes into separate files to be used to produce graphs to represent the data collected.

**Process Level Metrics**

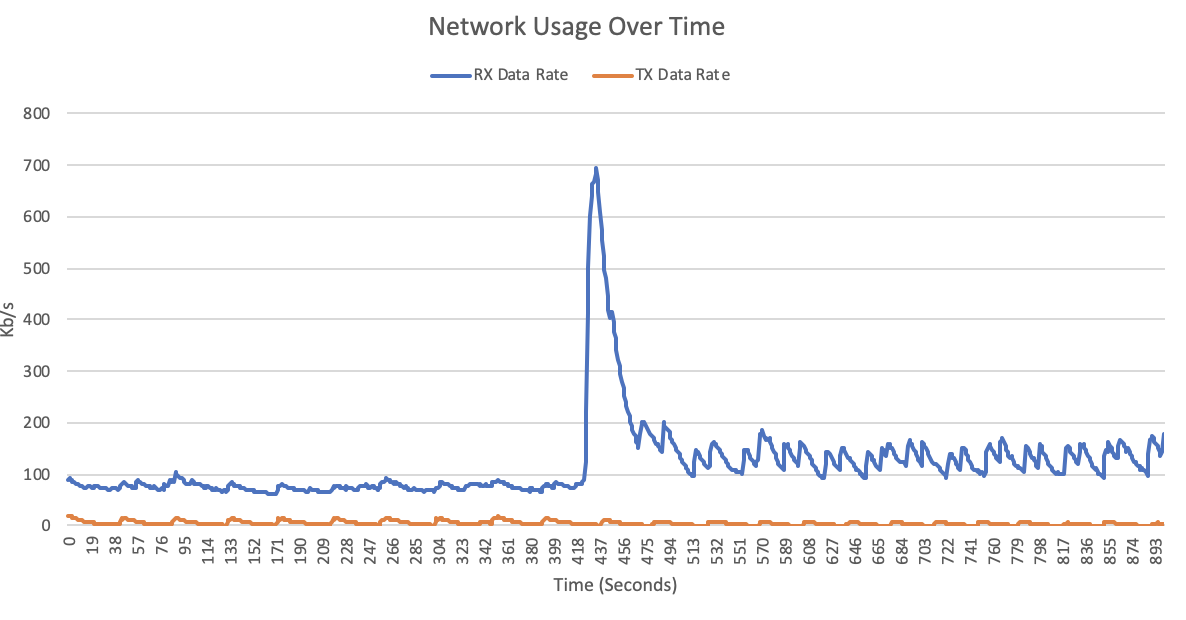


The CPU utilization plot above shows what percent of the CPU each executable for the project utilized while running. Only APM 1, APM 3, and APM 4 utilized the CPU in a capacity above zero percent. APM 1 and 3 both used a high CPU percentage on startup and then dipped a but in their utilization while running. APM 4 only utilized a small percentage of the CPU early in its life cycle. At the 690 seconds mark APM 4 started to steadily increase its CPU utilization.

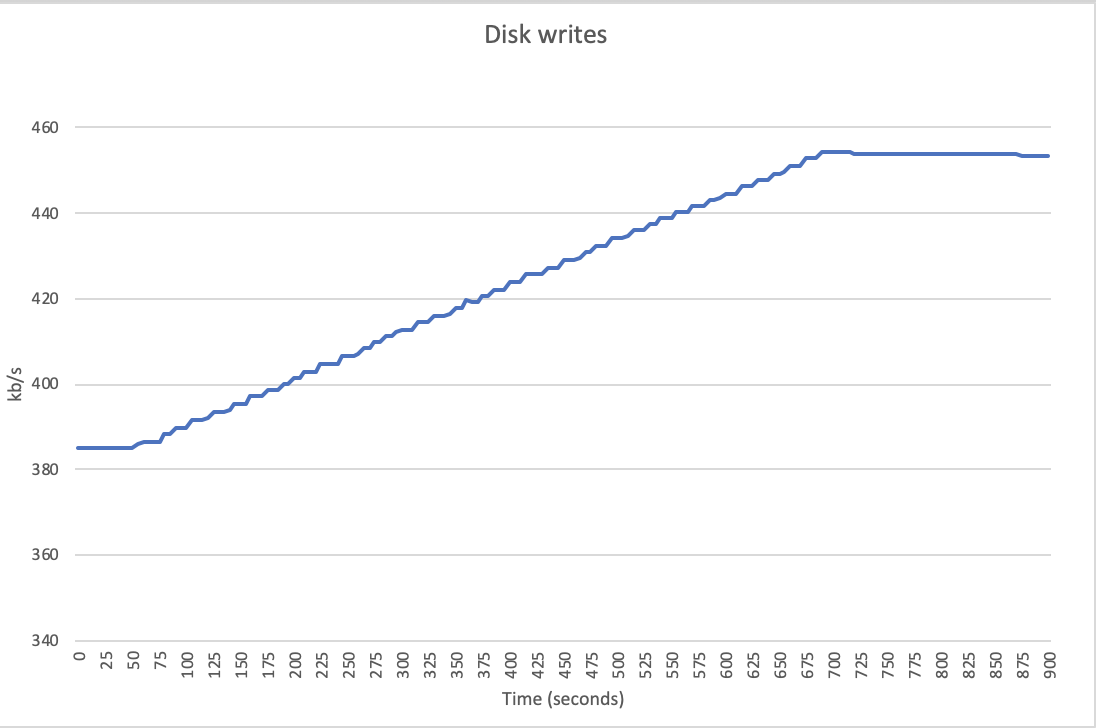


The memory utilization plot above shows which processes used a certain percentage of memory over the fifteen minute interval. Roughly every minute APM 1 spiked from zero percent memory utilization to a range of ~3.2% – ~4.8% memory utilization. APM 6 is a source of a memory leak, each time it utilizes memory it does not release it when it is finished using it so its memory utilization percentage creates a staircase pattern on the plot above and will continue to use more and more memory until the system crashes.

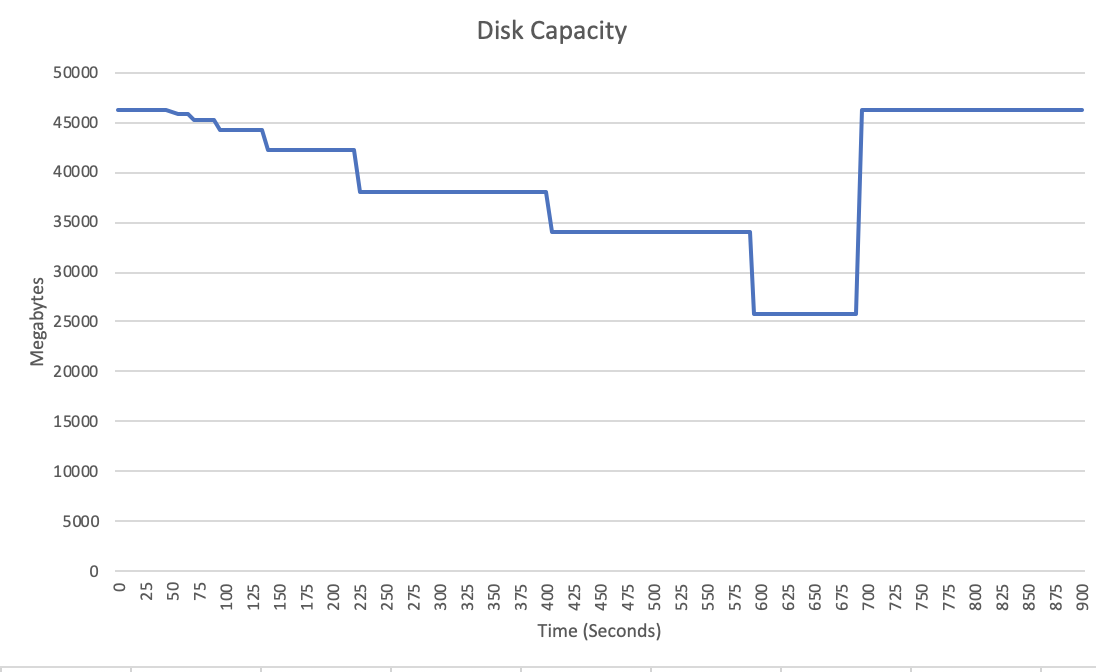
**System Level Metrics**



The transmit and receive data rates were different in size, but for the first half of the time they followed a similar pattern. Then there was a spike in the RX data rate and the pattern changed, while the TX rate and pattern stayed about the same.



The hard disk access rates plot above shows the measure of hard disk writes in kb/second to the primary hard drive of the system. As time progresses, the rate of kb/second increases starting around 385 kb/s and eventually reaching around 455 kb/s.



Over time the programs use up more disk space. The space goes down in a step like pattern, which makes sense as the programs would usalize chunks of space at once and not use the space as they ran. After this the disk got all of its space back as the chart shows it is back to its original capacity.

**Summary and Lessons Learned**

This VM had enough computing resources to handle the mic of application processes that were running. At the start we did see a large spike in CPU usage, but we never saw a 100% use of the VM’s resources and we did not experience any crashes. One lesson learned was how to get system data from a machine as it runs processes over time. Another lesson learned was how to identify any potential issues that may occur as a result of running the processes, like a memory leak. We also learned how to collaborate with others while working as a team to complete tasks together in an organized manner.