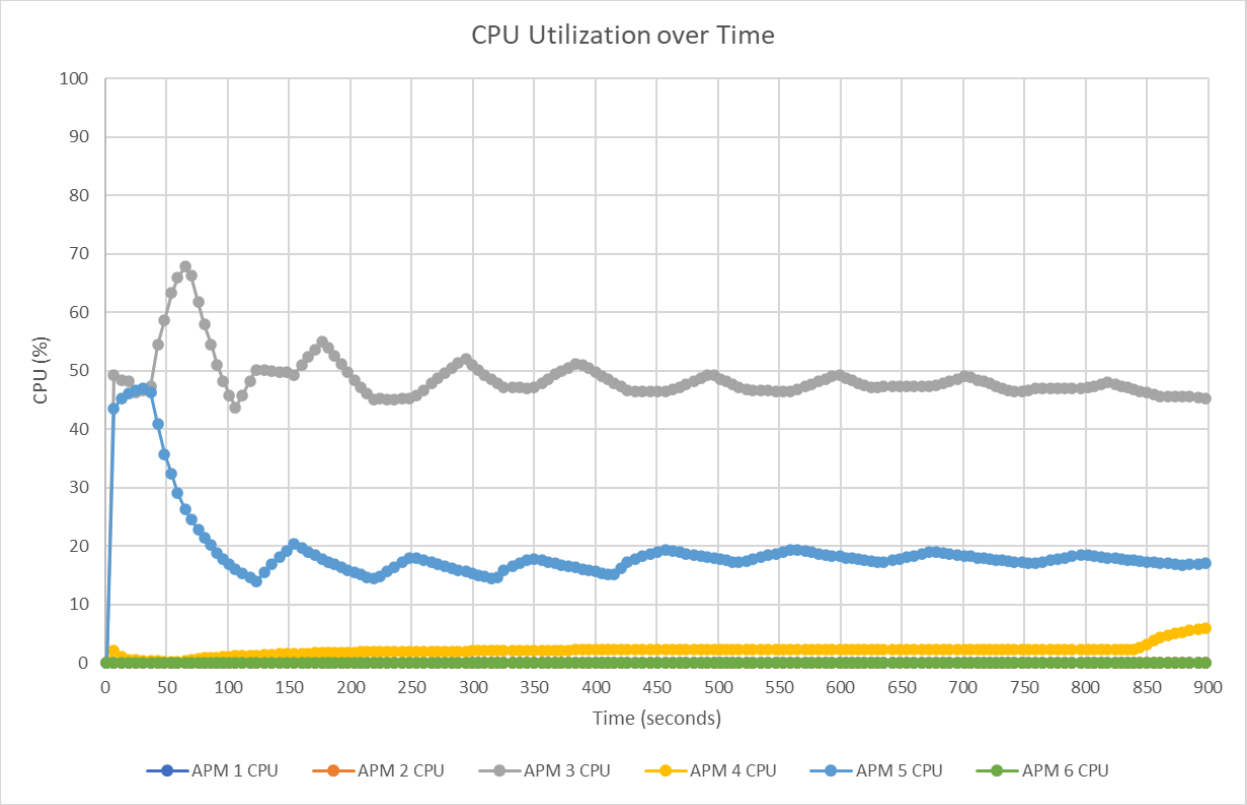
NSSA-220 Project 1: Application Performance Monitoring

Gary Shadders, Kevin Hlavaty, Aisha Bhakta, Damian Stevenson

Introduction

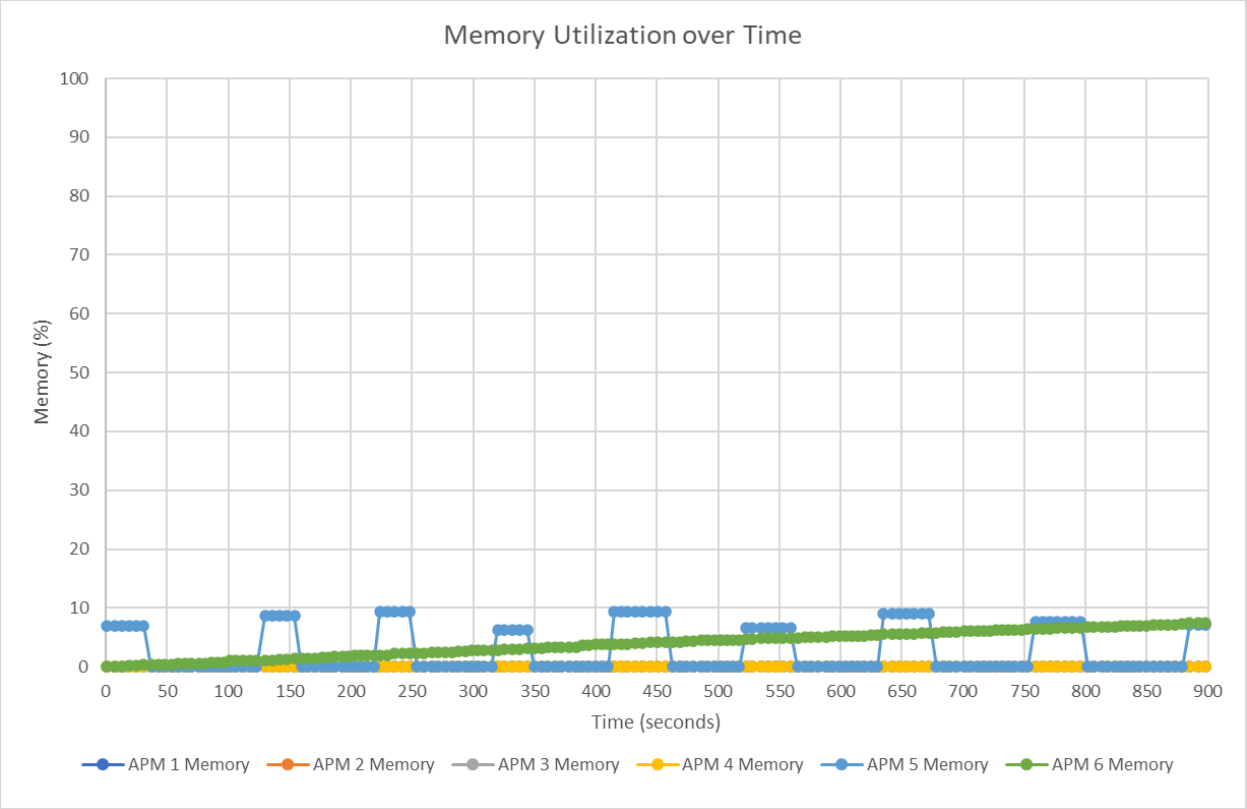
This project is about creating an Application Performance Monitoring script in Bash that is able to run and analyze metrics of 6 executable files written in C. The script will analyze the system level metrics, including network bandwidth utilization, hard disk access rates, and hard disk utilization. The APM script will also analyze the process level metrics including the amount of CPU and memory usage for each of the applications to discern how efficient or costly these applications are on the hardware. The APM also includes a clean up process at the end which kills all processes that are started at the beginning of the program and exports this information to csv files for easier analysis. Below are excel graphs that are there to represent the system and process level data.

Process Level Metrics



Programs 3 and 5 clearly show the highest rates of CPU utilization and are the two most costly programs to run on the VM with the CPU reaching its maximum resource consumption at 67.8 percent 65 seconds into program 3 running. Program 5 reaches maximum utilization at 47 percent 31 seconds into running. Both programs also repeat this wave like pattern where there are peaks and valleys of CPU utilization meaning that there are some repetitive processes

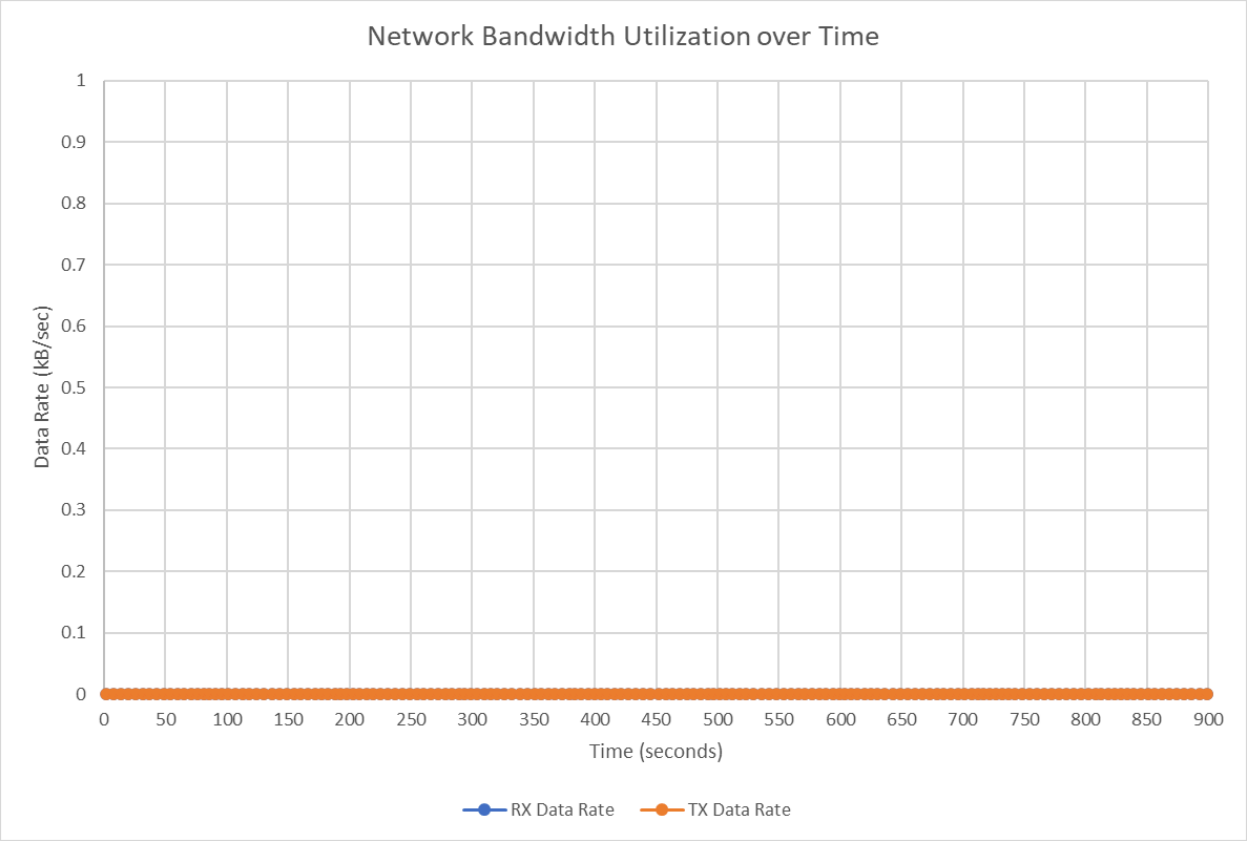
happening in both programs that are computationally demanding. Program 4 is relatively inexpensive in terms of CPU utilization, staying at around 2 percent for most of the program and rising up slightly to 6 percent towards the end of the program at around 850 seconds in, meaning it does something very minimally computationally taxing then begins to do slightly more work towards the end. Programs 2 and 6 consume 0 percent of the CPU the entire time.



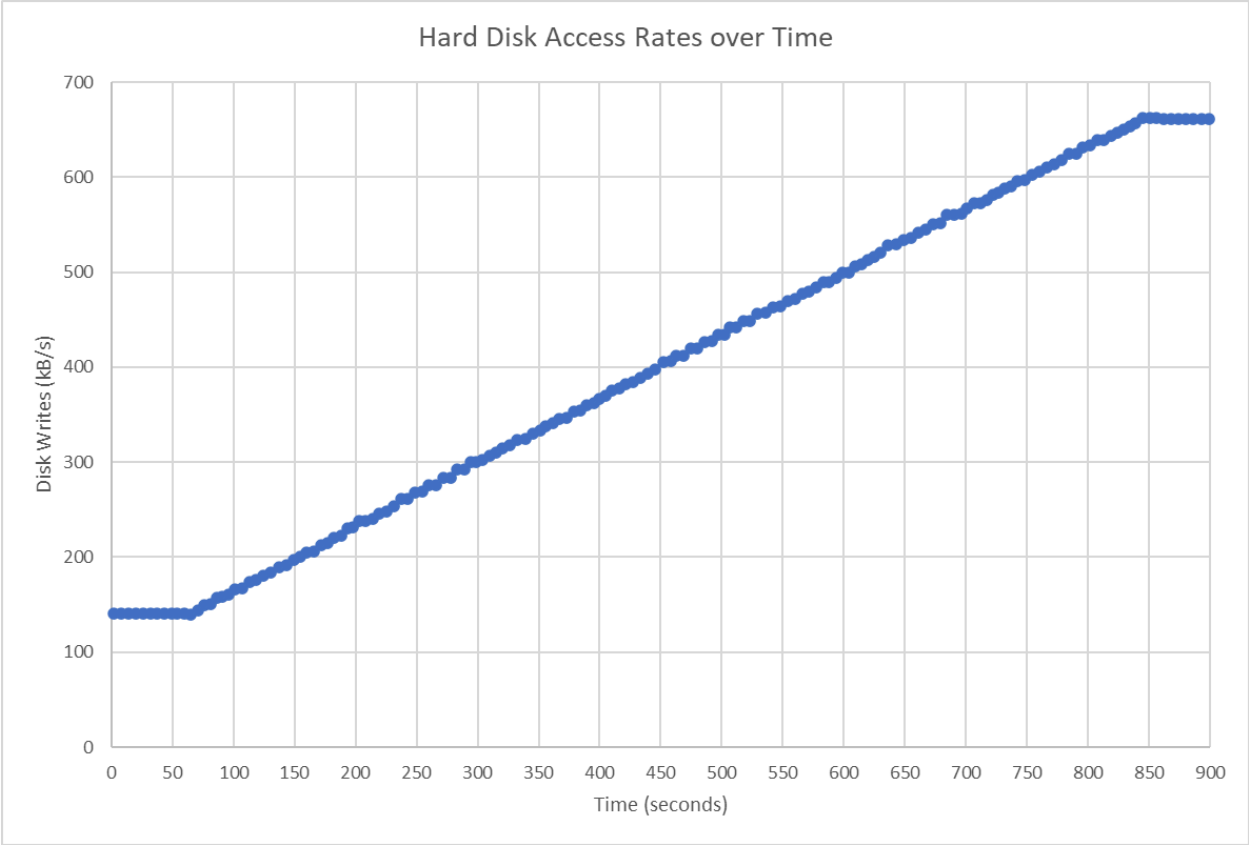
Programs 1, 2, 3, and 4 all consume 0 percent of the total memory, however program 6 steadily increases in the amount of memory percentage that it uses from 0 to 7.5 % over 15 minutes indicating that it could be a memory leak where the program allocates memory but then does not release that amount of memory when it is no longer needed. Program 5 has this rising and falling repetitive pattern where there are intervals such as 6.9 percent usage for the first 31 seconds then it drops to 0 %then 8.6% usage for between 130-154 seconds in, then 9.3% between 224-248 seconds in, and so on. This rising and falling pattern may indicate that this program has resource demanding tasks that it must perform and creates a lot of data to be stored but then falls back down to 0 using some sort of garbage collection process when it no longer needs the storage capacity.

\

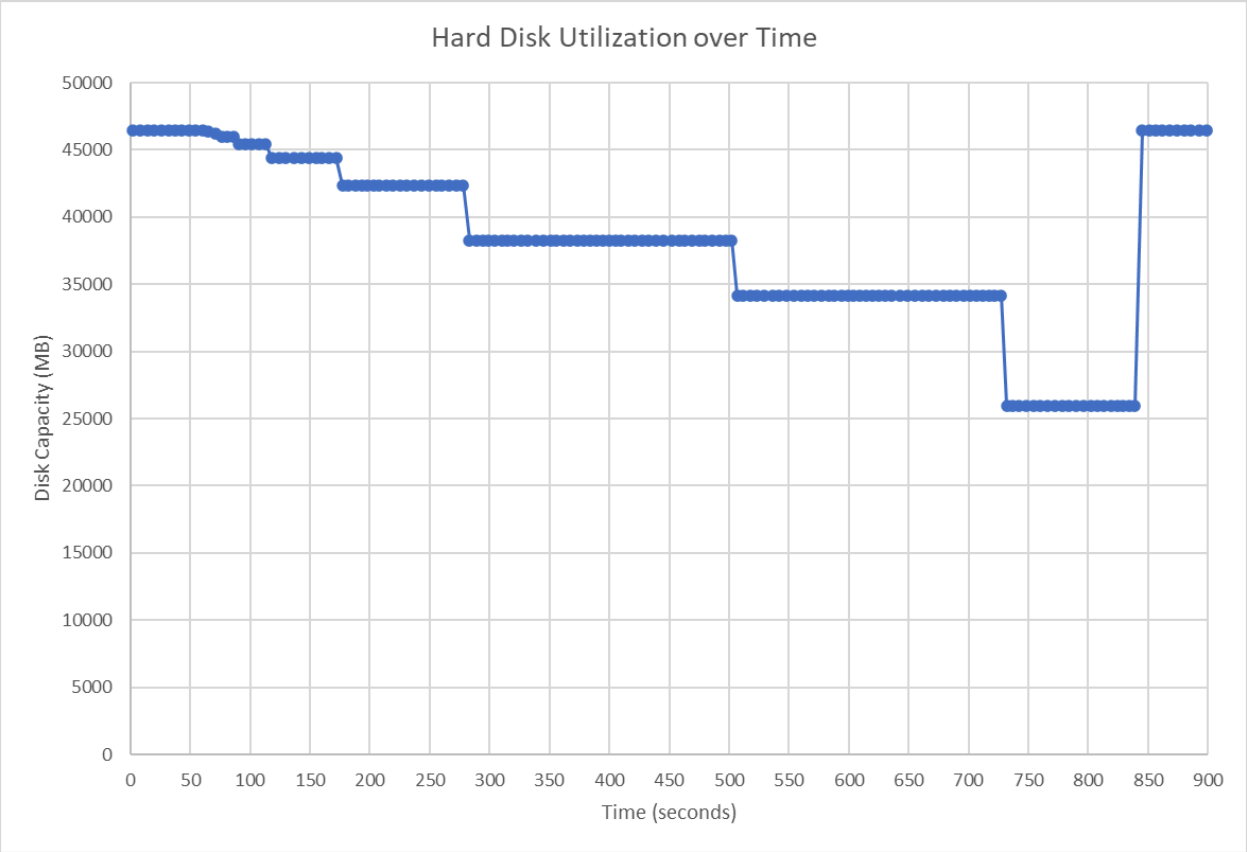
System Level Metrics



This plot shows that the ifstat command on the CentOS VM was not working as we expected. Some of the executables were definitely supposed to be sending or receiving data, but this plot supposedly shows that none of them used the network at all. When our bash script is run on any other machine, the network results contain plenty of non-zero values, indicating that this issue is specific to the RLES CentOS VM, and not representative of what the executables were actually doing.



This shows the hard disk access rates steadily and linearly increasing between around 70 seconds and 850 seconds. From 0-70 and 850-900, the hard disk access rates stay at a flat amount. This steady increase of Disk writes in kB/second indicates that there is an increase in the number of I/O operations and more data is being processed and stored.



This plot shows the hard disk capacity continuously decreasing throughout the duration that was being monitored, until the capacity sharply increases after around 850 seconds to the level it was when the bash script was started. The decrease in capacity all happens sharply in steps, with the flat sections becoming increasingly longer, which is potentially caused by how either the C library or Linux buffers outputs before writing them to disk.

Summary and Lessons Learned

From the metrics gathered, it can only be determined that the VM had enough RAM and disk capacity to handle the applications, since neither of those graphs show those resources becoming concerningly scarce. The total CPU usage tended to remain not far from 70%, which, despite never reaching 100%, could suggest that some of the programs were bottlenecked by the CPU, but that can't be confirmed. The disk access rate plot on its own cannot indicate much because there is no way to know how quickly the processes were attempting to access the disk. The network plot indicates nothing of value because we could not get ifstat working as expected on the VM.