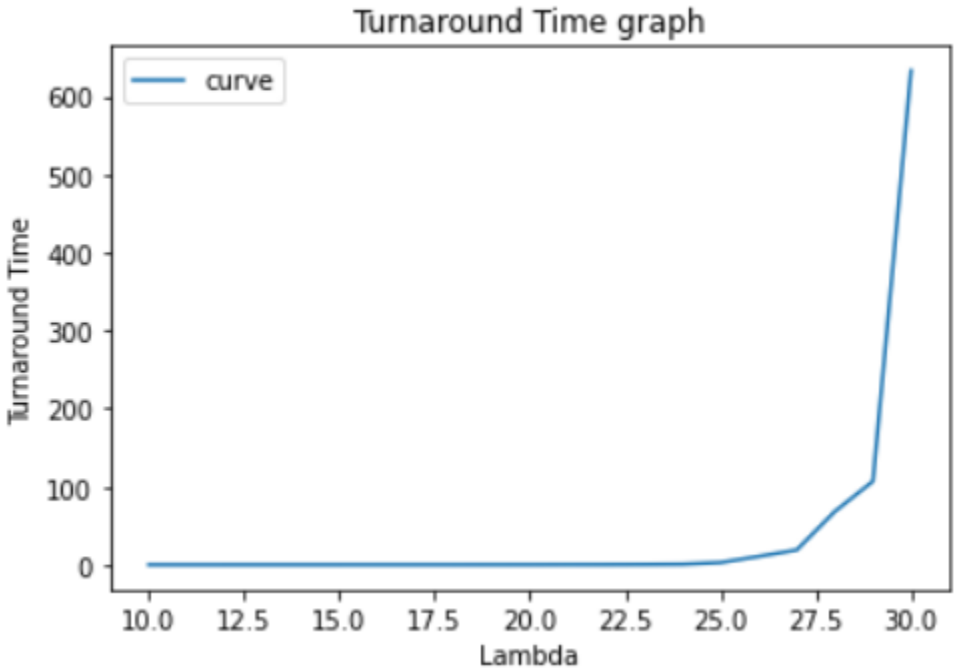
Ali Alsaedi

CS 3360 Project 1, Discrete-Time Event Simulation

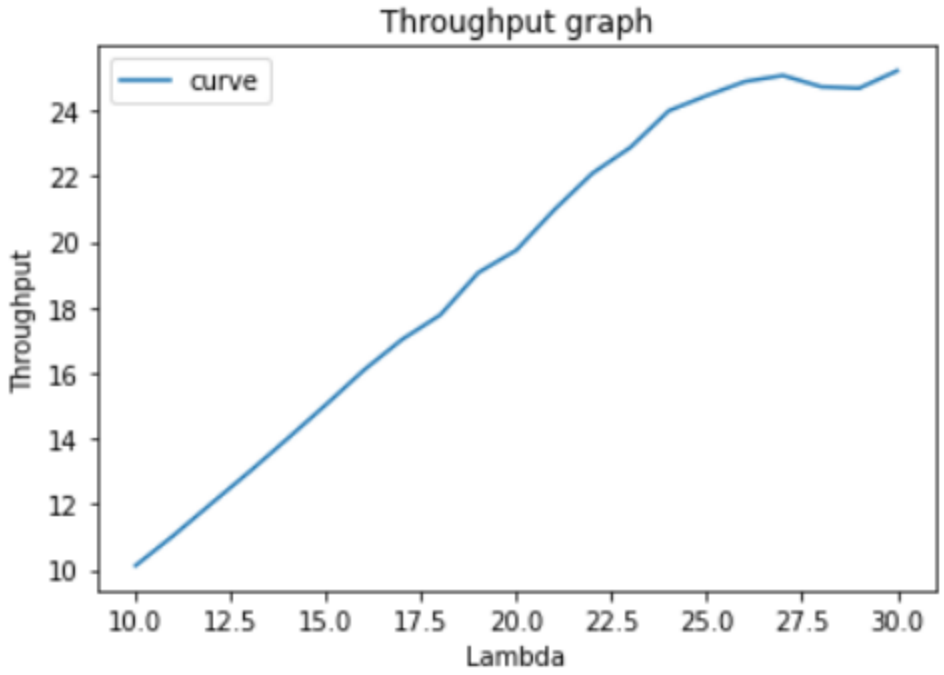
For this assignment I wrote a program that calculates the following:

- The average turnaround time of processes
- The total throughput (number of processes done per unit of time)
- The average CPU utilization
- The average number of processes in the Ready Queue of a simulated First-Come First-Served CPU scheduling algorithm.

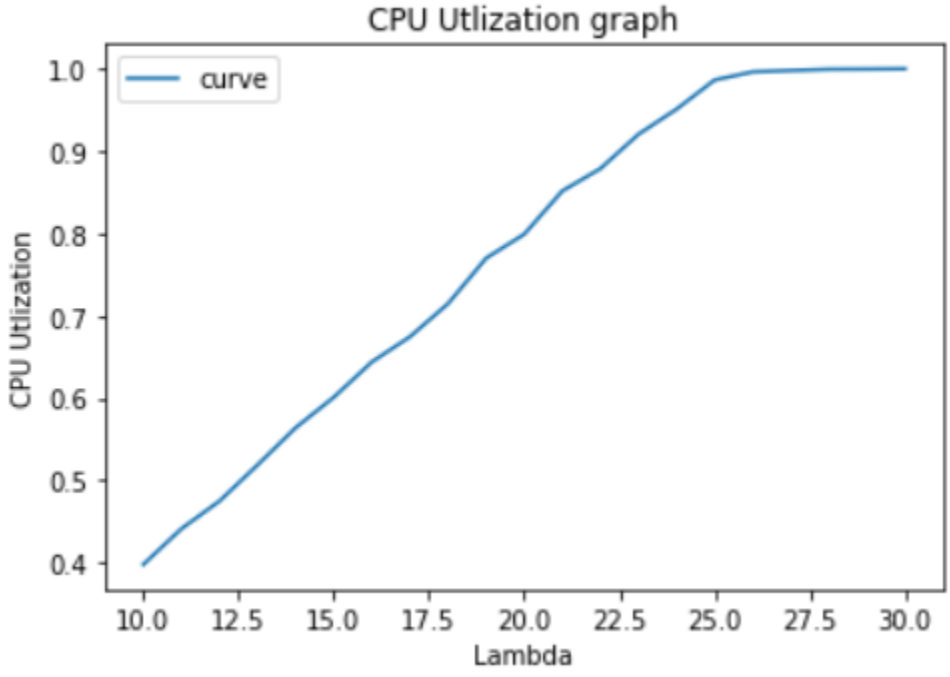
The program asks the user to enter an average arrival rate for processes entering the system and an average service time for those processes(commented out for the sake of the professor using a script to grade these) instead will run 20 runs with lambdas 10-30 with a TS of .04. Metrics are calculated after 10,000 processes have exited the system. Below are graphs of data obtained from 20 runs. The average arrival rate was set to 10 processes per second for the first run and was incremented by 1 for each run. The average service time was set to 0.04 seconds.



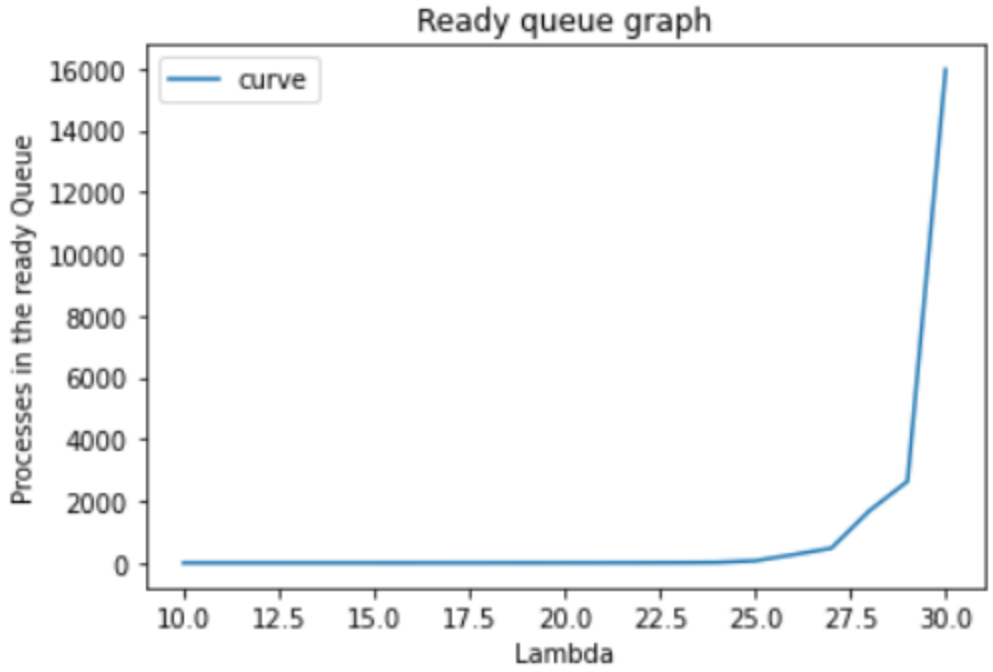
The average turnaround time increases dramatically when the average arrival rate is above 24 processes per second. This is expected because 1 second / 0.04 seconds per process = 25 processes After processes arrive at a rate of 25 processes per second, they spend significant time in the ready queue before being processed. Although I can tell the data is not perfect I think this is because of how the program runs on python, python is not the best language for counting through and simulating something low-level like this looking back I should have done this in CPP



The total throughput roughly matches the average arrival rate until the arrival rate reaches 25 processes per second. After 25 processes per second, the graph plateaus. This is expected because the CPU can only process 25 processes in one second.



The graph of the average utilization is similar to the graph of the total throughput. The CPU is busy roughly 40% of the time, with 10 processes entering the system in one second, with a service rate of 0.04 seconds. When the arrival rate hits 25, the CPU is close to max utilization. This continues for any arrival rates over 25 processes per second.



The average number of processes in the ready queue dramatically increases after the arrival rate hits 24 processes per second. Because the CPU can only process 25 processes in one second, the processes back up in the ready queue. Again similar to the turnaround time graph, this data is a little bit off I think this is in part due to working in python for low-level work.