MP3 Report

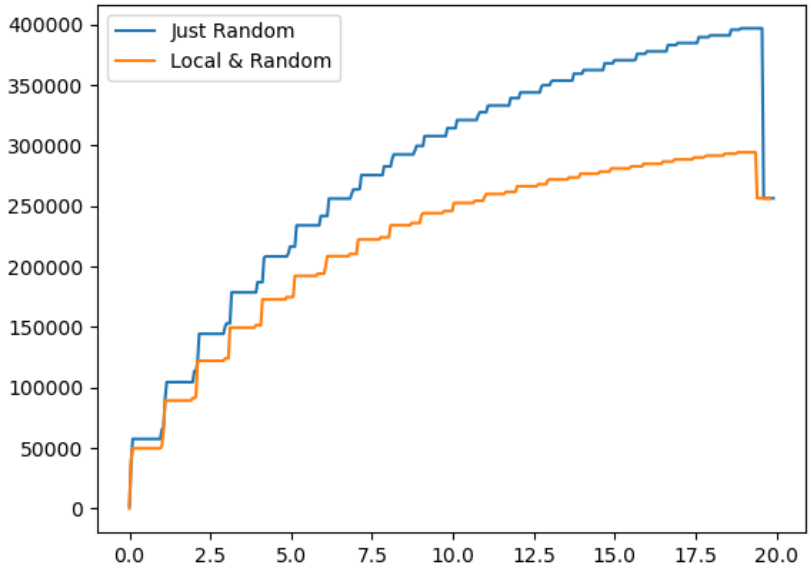
Dipayan Mukherjee (dipayan2)

Case Study 1:

In this study we have two types of memory access while maintaining the same workload. In one access the program has two jobs accessing memory randomly, while in the other one, we have smaller iteration of local access.

Here are the results :

Page Fault:



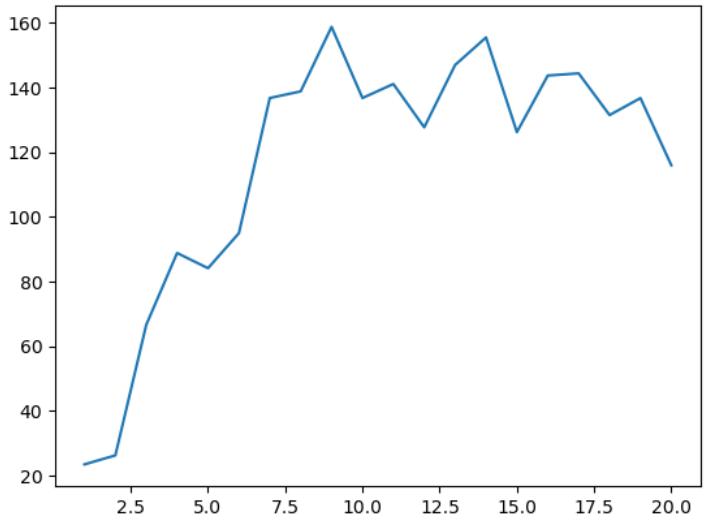
As evident the program with local access has less page faults, as there is temporal locality to the memory access. But still the number is high in both cases owing to the fact that, random access program had higher iteration count.

Time:

The execution time difference is negligible, I attribute it to them having a lower nice value, which makes the sleep most of the time. I am using both the user and system time combined.

Case Study 2 : Multiprogramming

This test checks for the CPU utilization as we add more processes:
I ran from 1-20 processes and plotted the max and the mean CPU utilization:
Max Util:
This shows that initially when we add processes the max utilization increases as we have more processes to run, but after a point, everyone gets into the I/O bottleneck and the utilization goes down. As evident from the mean utilization, we see that CPU/IO ratio is low and goes down drastically.



Mean Util:
Since the program sleeps most of the time due to high nice, we see low utilization, also as the number increases past 18, there is a drastic change, could be due the page utilization and thrashing.