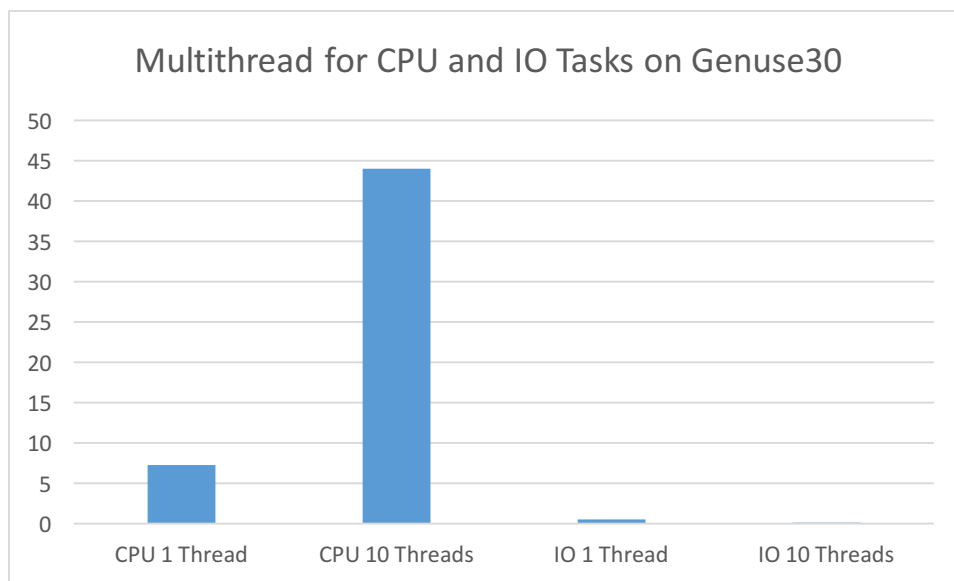
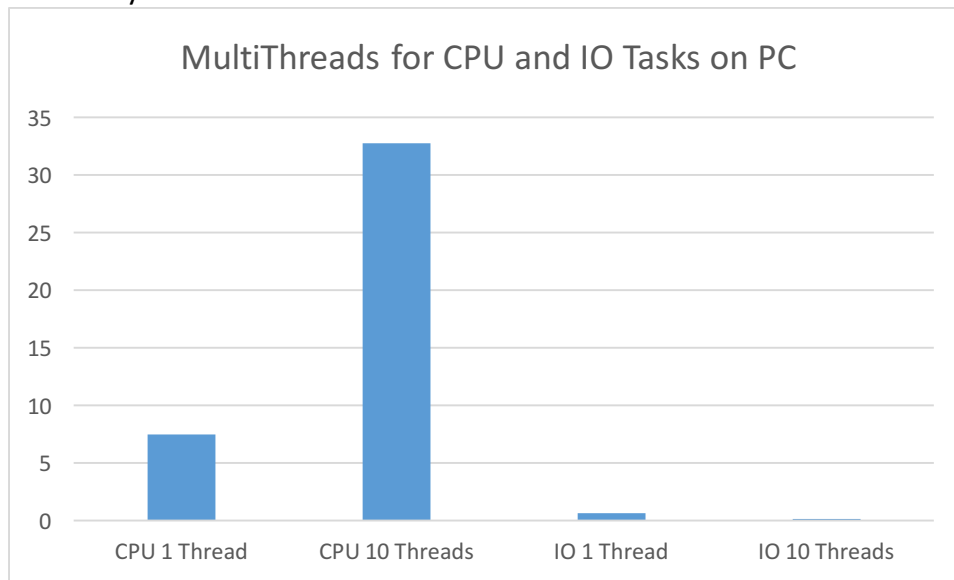
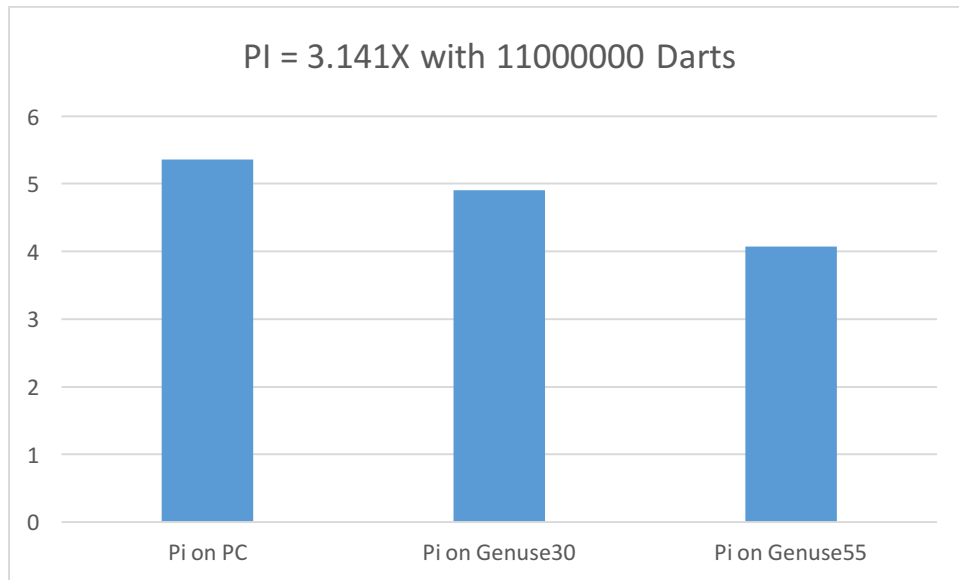


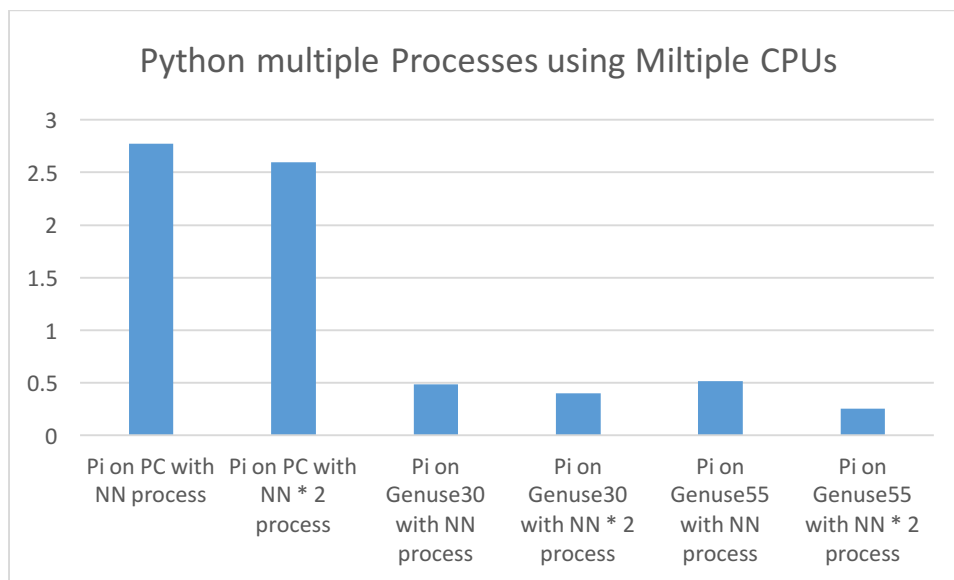
Note: all y axis times in seconds



IO tasks consistently take less time than CPU intensive tasks for multithreading. CPU times for single threads take much less time than multithreads. This is because to do a CPU intensive tasks takes the CPU, which when multithreading must be shared between each thread. However, with a single thread, that thread can complete the task and move on. IO tasks are more efficient when multithreading because IO take a large amount of time relative to other tasks in software. So while the IO is slowly working away, the CPU can be given to another task to complete.



This is also a CPU intensive task. The more CPU processing power a computer has the faster it will be able to complete this task. So it makes sense that Genuse55 with its fast ghz speed is able to complete the task faster than Genuse30 and my own PC. Likewise Genuse30 has more power than my computer and is able to complete it faster as well.



By dividing this task up among a computer's CPU, we are able to rapidly do CPU intensive tasks while sharing data without fear of race conditions. Rather than pile on our task onto one cpu and run through until it is stopped, we divide the task among multiple CPUs and simply combined the results in the end. This leads to faster times all together. By dividing our task further, we are able to make smaller even faster accomplished tasks thus reducing the time even further as shown by our graph.