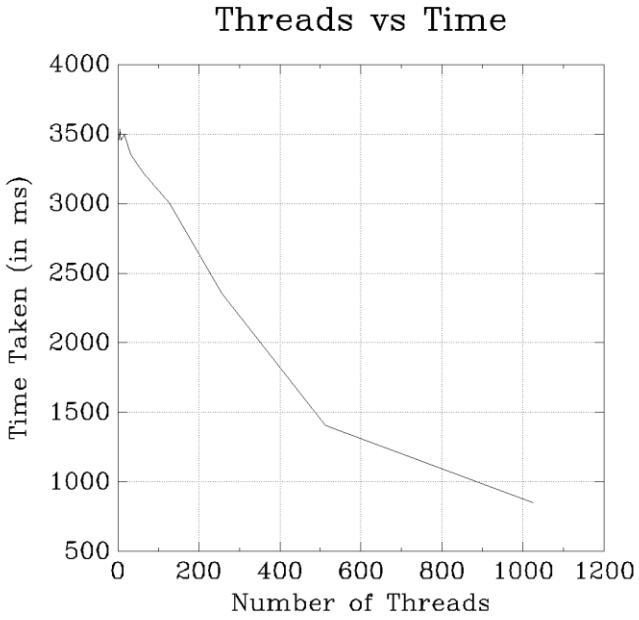
CS333 Operating Systems Lab-7

Virendra Kabra (200050157)

0. processes.c – Considering Copy-on-Write in play, when x is incremented by 5 in the child process, the parent and child no longer have a shared memory for x. Thus, value of x in parent is not modified. Also, PIDs of parent and child are different.
threads.c – Since the threads belong to the same process, they have the same PID (but different thread IDs, of course). t2 sleeps for 2 seconds (bar()), and t1 executes x+=5 by then. Moreover, since x is globally declared, this memory is same for all threads. Thus, x=7 is seen by both threads.

1. Without mutexes, account balance can be less than 10 million, as the critical section is not protected.
We use a mutex (lock-unlock) around the critical section (i.e., modification of account_balance).



As observed from the plot, increasing the number of threads decreases the time taken. We can also see that the performance benefit *decreases* after ~ 500 threads (slope can be seen to flatten out). This is due to the increased overheads compared to the actual work. For example, for 1024 threads, the work that each thread performs is just 2 increments. But there are added costs of thread creation, join, and mutex locking-unlocking.