

ASSIGNMENT-4 REPORT

PRUDHVI

CS16BTECH11016

In parallel computing, a barrier is a type of synchronization method. A barrier for a group of threads or processes in the source code means any thread/process must stop at this point and cannot proceed until all other threads/processes reach this barrier.

In the given assignment barrier is implemented using semaphores, which plays a vital role in synchronization process. And using inbuilt functions in pthread library `pthread_barrier_wait()`, which waits for all the threads to reach the specific points and unblock them as the final threads reaches it.

Barrier using semaphores:

In this two semaphores are used to implement the barrier. One semaphore is used to a common variable which is accessible by all threads i.e count variable. It counts all threads one by one to have a count of number of threads are about to enter the barrier point.

Another semaphore blocks all threads until the count reaches the total number of threads, when it becomes equal to n then it calls unblocking call to all the threads so that every thread releases from barrier.

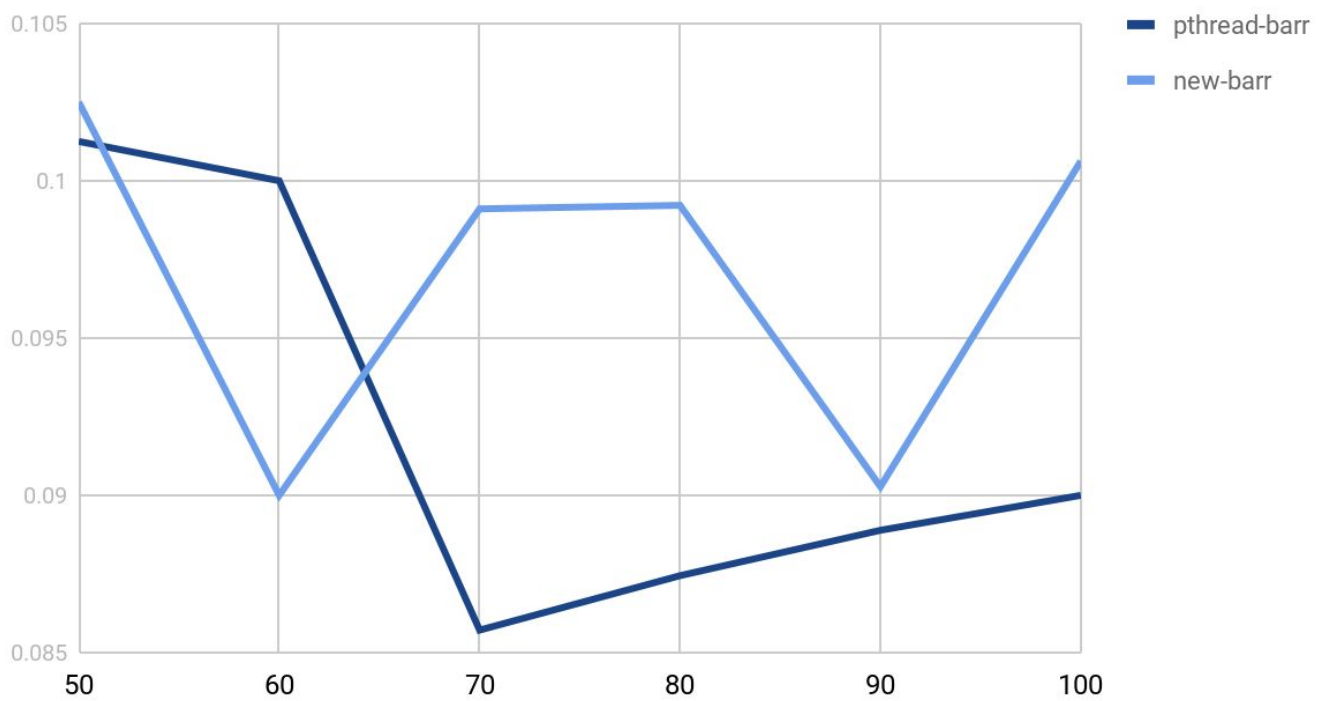
And this process repeats for k times.

Then average waiting time near barrier and waiting time of each thread are calculated and printed in a file.

GRAPH-1:

Here is the graph plotted against k (on X-axis) and average barrier waiting time (on Y-axis) keeping the number of threads = 16.

Average barr waiting time



GRAPH-2:

Below is the graph plotted against number of threads (on X-axis) and average waiting time of thread (on Y-axis) by keeping the value of $k=70$ (constant throughout all cases):

Average waiting time of thread

