

OBSERVATION REPORT

Details of code:

The aim of this project is understanding effects of threads in execution time. To synchronize the operations, we are using mutexes and condition variables. In Task_queue function and Terminate function, we are using queue mutex. Because task queue is shared thing for enqueue and dequeue operation. For instance, the one of the problem here is two different threads can reach the task queue and they can try to dequeue same node. This is big problem. So we are separating them.

Threads are created and they are waiting a signal in Terminate function. If we don't use condition variable, one of the thread can reach Thread_work function before enqueue a task. So we enqueue a task in a task queue and we are sending a signal like : "Okay you can do an operation .".

We are checking some condition for broadcast in Task_queue function. There are threads which are not busy and there are task in task queue to do. So we are saying to threads: "Go on operations."

Another thing is using mutexes in Thread_work function. We are talking about the shared queue and we have to manage them. They mustn't reach to queue at the same time.

We are using thread count check to determine the thread terminated its job or not. (Does it have a job ?)

Observation:

The important thing is ratio of task over thread number. We observe that if a lot of task to do, multithreading cannot be beneficial for us. Because we know that there is a little time between threads while they are joining. Sequential version is advantage for us for this case. Our codes behaves like mostly sequential ; because threads are waiting other threads. Just one thread reach list to insert, delete or search operation ; so our time is more than sequential version. For 1000 task, we realize that more threads takes more time.

Conclusion, parallelized program is always faster than sequential one is wrong. Ratio is very important. Execution time depends on conditions. (task number, thread number etc.)

Examples:

```
alcatraz@alcatraz: ~/Desktop
File Edit View Search Terminal Help
alcatraz@alcatraz:~/Desktop$ ./taskqueue.o 2 20
Thread 0: task 0: 33 cannot be deleted
Thread 1: task 1: 27 cannot be deleted
Thread 1: task 3: 36 is inserted
Thread 1: task 4: 49 cannot be deleted
Thread 0: task 2: 43 cannot be deleted
Thread 0: task 6: 40 cannot be deleted
Thread 0: task 7: 13 cannot be deleted
Thread 1: task 5: 12 cannot be deleted
Thread 1: task 9: 22 cannot be deleted
Thread 1: task 10: 11 is not in the list
Thread 1: task 11: 17 is inserted
Thread 0: task 8: 40 is inserted
Thread 0: task 13: 12 cannot be deleted
Thread 1: task 12: 32 is not in the list
Thread 1: task 15: 29 is inserted
Thread 0: task 14: 17 is deleted
Thread 1: task 16: 22 is inserted
Thread 1: task 18: 43 cannot be deleted
Thread 0: task 17: 19 is inserted
Thread 1: task 19: 11 cannot be deleted
Th 0 > Queue empty
Th 1 > Queue empty
main: Final list:
    19 22 29 36 40

0.001879
alcatraz@alcatraz:~/Desktop$
```

```
alcatraz@alcatraz: ~/Desktop
File Edit View Search Terminal Help
alcatraz@alcatraz:~$ cd Desktop/
alcatraz@alcatraz:~/Desktop$ ./taskqueue.o 4 8
Thread 2: task 0: 33 cannot be deleted
Thread 3: task 1: 27 cannot be deleted
Thread 1: task 2: 43 cannot be deleted
Thread 1: task 3: 36 is inserted
Thread 1: task 4: 49 cannot be deleted
Thread 2: task 5: 12 cannot be deleted
Thread 1: task 6: 40 cannot be deleted
Thread 0: task 7: 13 cannot be deleted
Th 3 > Queue empty
Th 2 > Queue empty
Th 1 > Queue empty
Th 0 > Queue empty
main: Final list:
    36

0.002681
alcatraz@alcatraz:~/Desktop$
```

Graphs:





