CSE 5441: Program 2 Report

Harsh Gupta

Contents

Objective

The objective of this lab is to use threads and analyse how serial execution of the code differs from the parallel version of it. Two different versions of parallel code were implemented and analysed. They are as follows

a. Disposable
b. Persistent

This assignment was a good learning exercise for me as I learnt a lot about multi-threaded programs and it complexities. One big take away from this exercise was how to debug a program with many threads as bugs are difficult to detect.

Parallelizing the code

The DSV's of a group of boxes was calculated in parallel by different threads.

Disposable threads - Threads are created and destroyed in each iteration of convergence loop. This causes extra overhead and slows down the program as compares to persistent thread.

Persistent threads - Threads are created before calling convergence loop. These threads are destroyed only once when convergence is achieved. This saves a lot of thread creation and destruction overhead as only one set of thread are used throughout the convergence loop.

Optimizations done in the code - Boxes are divided almost equally between threads to achieve maximum performance. If number of boxes are not divisible by number of threads, then one extra box is assigned to the threads 0 to thread (extra value - 1).

Example: If there are 23 boxes (0 to 22) and 5 threads (0 to 4)
average box per thread = 4
extra boxes = (23%5) = 3
so, thread_0, thread_1, thread_2 gets 5 boxes each and thread_3 and thread_4 gets 4 boxes

Calculating the starting point for each thread
start point = average box per thread * thread ID + minimum (thread ID, extra boxes)
thread_0 will start from 4 * 0 + 0 =0, so thread_0 will compute box 0 to 4
thread_1 will start from 4 * 1 + 1 =5, so thread_1 will compute box 5 to 9
thread_2 will start from 4 * 2 + 2 =10, so thread_2 will compute box 10 to 14
thread_3 will start from 4 * 3 + 3 =15, so thread_3 will compute box 15 to 18
thread_4 will start from 4 * 4 + 3 = 19, so thread_4 will compute box 19 to 22

Summary of execution time

Persistent and disposable programs were executed for test grid testgrid_400_12206 using different number of threads. Same values for Max DSV, Min DSV and number of iterations was obtained.

Iterations	**Max DSV**	**Min DSV**	**Affect Rate**	**Epsilon**	**File**
75197	**0.086671**	**0.078004**	**0.1**	**0.1**	**Testgrid_400_12206**

Program	Number of threads	Clock	time	Chrono
Sequential	1	37270000	37	37288225.0
Disposable	1	48890000	50	49291955.0
Disposable	2	50450000	34	33913786.0
Disposable	4	55310000	34	34172576
Disposable	**8**	**67320000**	**31**	**31343012.0**
Disposable	16	116120000	66	65808348.0
Disposable	32	191750000	135	134279204.0
Persistent	1	39700000	40	39870045.0
Persistent	2	45410000	29	28726694
Persistent	**4**	**49220000**	**25**	**24826671**
Persistent	8	55030000	30	30238986
Persistent	16	61720000	41	41497096
Persistent	32	62290000	53	52423610

Analysis and Observations

Did this program perform better sequentially or in parallel?

The persistent as well as disposable program performed better than the sequential program when number of threads used were not very large.
When using greater than 8 threads, the performance of the multi-threaded programs started dropping and it was essentially because of the overheads caused by creation of threads and a lot of context switches happening in the background.

Which number of threads was most effective?
Persistent threads – The best performance came when the program ran with **4** threads in execution.

Disposable threads – The best performance was recorded when the code was executed with **8** threads.

Which parallel version was most effective?
Persistent thread program clearly comes out as a winner. Its faster than the disposable program in all cases. This was even expected because in case of persistent program, there was no time wasted to create and destroy the threads in each iteration, thereby saving a lot of time.
Major difference can be observed when 32 threads are used.
Persistent time: 53 seconds
Disposable time: 135 seconds

How did your results match or conflict with your expectations?
The results were generally on expected lines, I was however expecting a little more gain in terms of performance i.e. lesser execution time for threads between 2 to 4, but it seems that the gain obtained by parallelising the thread got offset by the thread creation overhead. For larger number of threads since there was a lot of context switch, so performance degradation was on expected lines.

Were there any unexpected anomalies in the timing information collected?

- My expectations were high in terms of numbers. I expected the code to run much faster with more threads but I didn't realize the overall overhead to create threads and context switches could be so high.
- I even expected that more number of threads will lead to more faster execution of code (at least for 50 threads) but it turned out that overhead was substantial and more than my expectation.
- As expected persistent program performed better than the disposable thread program.

Which timing methods seem best for parallel programs? How does this compare with your expectations?

After an analysis of various times, what I discovered was very surprising to me. Clock time showed a constant increase with increasing threads. This was something which I didn't expect. Later after some research I found out that clock time is the total processor time utilized by the program.

Since we are running multiple threads the processor time increases rapidly. Clock seems important after this assignment as it add timings for each thread. Both clock and Chrono are equally important and provide different timings.

Results

time ./gupta_harsh_**disposable** 0.1 0.1 2 < /class/cse5441/testgrid_400_12206

****
Dissipation converged in 75197 iterations
with max DSV = 0.086671 and min DSV = 0.078004
Number of threads: 2, Affect rate = 0.100000 , epsilon = 0.100000
elapsed convergence loop time(clock) : 50450000
elapsed convergence loop time(time) : 34
elapsed convergence loop time (chrono): 33913786.000000
****

real 0m33.936s
user 0m47.120s
sys 0m3.358s

time ./gupta_harsh_disposable 0.1 0.1 4 < /class/cse5441/testgrid_400_12206
****
Dissipation converged in 75197 iterations
with max DSV = 0.086671 and min DSV = 0.078004
Number of threads: 4, Affect rate = 0.100000 , epsilon = 0.100000
elapsed convergence loop time(clock) : 55310000
elapsed convergence loop time(time) : 34
elapsed convergence loop time (chrono): 34172576.000000
****

real 0m34.198s
user 0m48.443s
sys 0m6.884s

time ./gupta_harsh_disposable 0.1 0.1 8 < /class/cse5441/testgrid_400_12206
****
Dissipation converged in 75197 iterations
with max DSV = 0.086671 and min DSV = 0.078004
Number of threads: 8, Affect rate = 0.100000 , epsilon = 0.100000
elapsed convergence loop time(clock) : 67320000
elapsed convergence loop time(time) : 31
elapsed convergence loop time (chrono): 31343012.000000
****

real 0m31.365s
user 0m51.921s
sys 0m15.417s

time ./gupta_harsh_disposable 0.1 0.1 16 < /class/cse5441/testgrid_400_12206
****

```
Dissipation converged in 75197 iterations
with max DSV = 0.086671 and min DSV = 0.078004
Number of threads: 16, Affect rate = 0.100000 , epsilon = 0.100000
elapsed convergence loop time(clock) : 116120000
elapsed convergence loop time(time) : 66
elapsed convergence loop time (chrono): 65808348.000000
**********************************************************************

real    1m5.829s
user    1m1.171s
sys     0m54.964s

time ./gupta_harsh_disposable 0.1 0.1 32 < /class/cse5441/testgrid_400_12206
**********************************************************************
Dissipation converged in 75197 iterations
with max DSV = 0.086671 and min DSV = 0.078004
Number of threads: 32, Affect rate = 0.100000 , epsilon = 0.100000
elapsed convergence loop time(clock) : 191750000
elapsed convergence loop time(time) : 135
elapsed convergence loop time (chrono): 134279204.000000
**********************************************************************

real    2m14.302s
user    1m12.487s
sys     1m59.288s

time ./gupta_harsh_persistent 0.1 0.1 2 < /class/cse5441/testgrid_400_12206
**********************************************************************
Dissipation converged in 75197 iterations
with max DSV = 0.086671 and min DSV = 0.078004
Number of threads: 2, Affect rate = 0.100000 , epsilon = 0.100000
elapsed convergence loop time(clock) : 45410000
elapsed convergence loop time(time) : 29
elapsed convergence loop time (chrono): 28726694.000000
**********************************************************************

real    0m28.748s
user    0m44.314s
sys     0m1.112s
time ./gupta_harsh_persistent 0.1 0.1 4 < /class/cse5441/testgrid_400_12206
**********************************************************************
Dissipation converged in 75197 iterations
with max DSV = 0.086671 and min DSV = 0.078004
Number of threads: 4, Affect rate = 0.100000 , epsilon = 0.100000
elapsed convergence loop time(clock) : 49220000
elapsed convergence loop time(time) : 25
elapsed convergence loop time (chrono): 24826671.000000
**********************************************************************
```

```
real    0m24.847s
user    0m46.144s
sys     0m3.099s

time ./gupta_harsh_persistent 0.1 0.1 8 < /class/cse5441/testgrid_400_12206

**********************************************************************
Dissipation converged in 75197 iterations
with max DSV = 0.086671 and min DSV = 0.078004
Number of threads: 8, Affect rate = 0.100000 , epsilon = 0.100000
elapsed convergence loop time(clock) : 55030000
elapsed convergence loop time(time) : 30
elapsed convergence loop time (chrono): 30238986.000000
**********************************************************************

real    0m30.261s
user    0m47.898s
sys     0m7.157s

time ./gupta_harsh_persistent 0.1 0.1 16 < /class/cse5441/testgrid_400_12206

**********************************************************************
Dissipation converged in 75197 iterations
with max DSV = 0.086671 and min DSV = 0.078004
Number of threads: 16, Affect rate = 0.100000 , epsilon = 0.100000
elapsed convergence loop time(clock) : 61720000
elapsed convergence loop time(time) : 41
elapsed convergence loop time (chrono): 41497096.000000
**********************************************************************

real    0m41.518s
user    0m49.378s
sys     0m12.360s

time ./gupta_harsh_persistent 0.1 0.1 32 < /class/cse5441/testgrid_400_12206
**********************************************************************
Dissipation converged in 75197 iterations
with max DSV = 0.086671 and min DSV = 0.078004
Number of threads: 32, Affect rate = 0.100000 , epsilon = 0.100000
elapsed convergence loop time(clock) : 62290000
elapsed convergence loop time(time) : 53
elapsed convergence loop time (chrono): 52423610.000000
**********************************************************************

real    0m52.446s
user    0m47.577s
sys     0m14.735s
```