Naveen Tumkur Ramesh Babu
Computer Science and Engineering, OSU

LAB REPORT FOR PROGRAM-2

disposable, persistent is output file
0.1 is Epsilon
0.01 is affect-rate
testgrid_400_12206 is input file (Path /class/cse5441)
Number of threads would be specified in command line

Persistent Threads:-

1)**For testgrid_400_12206 with 2 threads.**

```
[naveentumkurrameshbabu.1@sl5 cse5441_lab2]$ time ./persistent 0.01 0.1 2 <
/class/cse5441/testgrid_400_12206

 ************************************************

 dissipation converged in 751978 iterations, with max DSV = 0.086671 and min DSV = 0.078004
 affectrate =0.010000 ;   epsilon=0.100000
 elapsed convergence loop time  (clock): 319.820007 sec

 Elapsed converged loop-time (time) : 189.000000

 Elapsed converged loop-time (chrono) : 189.000000
 ************************************************

real    3m9.306s
user    5m7.613s
sys     0m12.224s
```

2)**For testgrid_400_12206 with 8 threads.**

```
[naveentumkurrameshbabu.1@sl5 cse5441_lab2]$ time ./persistent 0.01 0.1 8 <
/class/cse5441/testgrid_400_12206

 ***********************************************

 dissipation converged in 751978 iterations, with max DSV = 0.086671 and min DSV = 0.078004
 affectrate =0.010000 ;   epsilon=0.100000
 elapsed convergence loop time  (clock): 483.679993 sec

 Elapsed converged loop-time (time) : 266.000000

 Elapsed converged loop-time (chrono) : 266.000000
 ***********************************************

real    4m26.137s
user    6m54.949s
sys     1m8.756s
[naveentumkurrameshbabu.1@sl5 cse5441_lab2]$
```

3)For testgrid_400_12206 with 16 threads.

```
[naveentumkurrameshbabu.1@sl5 cse5441_lab2]$ time ./persistent 0.01 0.1 16 <
/class/cse5441/testgrid_400_12206
 ***********************************************

 dissipation converged in 751978 iterations, with max DSV = 0.086671 and min DSV = 0.078004
 affectrate =0.010000 ;   epsilon=0.100000
 elapsed convergence loop time  (clock): 567.989990 sec

 Elapsed converged loop-time (time) : 430.000000

 Elapsed converged loop-time (chrono) : 430.000000
 ***********************************************
```

real 7m9.830s

user 7m38.199s

sys 1m49.812s

4)**For testgrid_400_12206 with 32 threads.**

[naveentumkurrameshbabu.1@sl5 cse5441_lab2]$

[naveentumkurrameshbabu.1@sl5 cse5441_lab2]$ time ./persistent 0.01 0.1 32 <
/class/cse5441/testgrid_400_12206

****

dissipation converged in 751978 iterations, with max DSV = 0.086671 and min DSV = 0.078004

affectrate =0.010000 ; epsilon=0.100000

elapsed convergence loop time (clock): 665.390015 sec

Elapsed converged loop-time (time) : 552.000000

Elapsed converged loop-time (chrono) : 552.000000

****

real 9m12.475s

user 8m26.294s

sys 2m39.118s

[naveentumkurrameshbabu.1@sl5 cse5441_lab2]$

Disposable Threads:-

1)**For testgrid_400_12206 with 2 threads.**

```
[naveentumkurrameshbabu.1@sl5 cse5441_lab2]$ time ./disposable 0.01 0.1 2 <
/class/cse5441/testgrid_400_12206

 ***********************************************

 dissipation converged in 751978 iterations, with max DSV = 0.086671 and min DSV = 0.078004
 affectrate =0.010000 ;   epsilon=0.100000
 elapsed convergence loop time  (clock): 357.510010 sec

 Elapsed converged loop-time (time) : 232.000000

 Elapsed converged loop-time (chrono) : 235.000000
 ***********************************************

real    3m52.373s
user    5m14.672s
sys     0m42.862s
[naveentumkurrameshbabu.1@sl5 cse5441_lab2]$
```

2)For testgrid_400_12206 with 8 threads.

```
[naveentumkurrameshbabu.1@sl5 cse5441_lab2]$ time ./disposable 0.01 0.1 8 <
/class/cse5441/testgrid_400_12206

 ***********************************************

 dissipation converged in 751978 iterations, with max DSV = 0.086671 and min DSV = 0.078004
 affectrate =0.010000 ;   epsilon=0.100000
 elapsed convergence loop time  (clock): 540.830017 sec

 Elapsed converged loop-time (time) : 455.000000

 Elapsed converged loop-time (chrono) : 458.000000
```

```
 ********************************************

real    7m35.426s
user    6m48.142s
sys     2m12.721s
[naveentumkurrameshbabu.1@sl5 cse5441_lab2]$
```

3)For testgrid_400_12206 with 16 threads.

```
[naveentumkurrameshbabu.1@sl5 cse5441_lab2]$ time ./disposable 0.01 0.1 16 <
/class/cse5441/testgrid_400_12206

 ********************************************

 dissipation converged in 751978 iterations, with max DSV = 0.086671 and min DSV = 0.078004
 affectrate =0.010000 ;   epsilon=0.100000
 elapsed convergence loop time  (clock): 1036.049927 sec

 Elapsed converged loop-time (time) : 1013.000000

 Elapsed converged loop-time (chrono) : 1012.000000
 ********************************************

real    16m52.865s
user    9m20.571s
sys     7m55.538s
[naveentumkurrameshbabu.1@sl5 cse5441_lab2]$
```

4)For testgrid_400_12206 with 32 threads.

[naveentumkurrameshbabu.1@sl5 cse5441_lab2]$ time ./disposable 0.01 0.1 32 <
/class/cse5441/testgrid_400_12206

****

dissipation converged in 751978 iterations, with max DSV = 0.086671 and min DSV = 0.078004
affectrate =0.010000 ; epsilon=0.100000
elapsed convergence loop time (clock): 1758.220093 sec

Elapsed converged loop-time (time) : 2214.000000

Elapsed converged loop-time (chrono) : 2208.000000
****

real 36m53.477s
user 12m4.806s
sys 17m13.498s

Serial Execution from previous Program:-

[naveentumkurrameshbabu.1@sl5 cse5441_lab1]$ time ./5441 0.01 0.1 <
/class/cse5441/testgrid_400_12206

****

dissipation converged in 751978 iterations, with max DSV = 0.086671 and min DSV = 0.078004
affectrate =0.010000 ; epsilon=0.100000

Elapsed converged loop-time (clock) : 280950000

Elapsed converged loop-time (time) : 282

Elapsed converged loop-time (chrono) : 282000000

****

real 4m41.961s

user 4m40.879s

sys 0m0.098s

Tabular Documentation of result

	Serial	Disposable	persistant
Previous Code	(clock): 280.95 sec (time): 282 sec (chrono): 282 sec real 4m41.961s		
2 threads		(clock): 357.510010 sec (time): 232.0000 sec (chrono): 235.0000 sec real: 3m52.373s	(clock): 319.820007 sec (time): 189.000000 sec (chrono): 189.000000 sec real: 3m9.306s
8 threads		(clock) : 540.830017 sec time: 455.000000 sec (chrono) : 458.000000 sec real : 7m35.426s	(clock) : 483.679993 sec time: 266.000000 sec (chrono) : 266.000000 sec real : 4m26.137s
16 threads		(clock): 1036.049927 sec time: 1013.0000 sec (chrono):1012.0000 sec	(clock): 567.989990 sec time: 430.000000 sec (chrono) : 430.000000 sec

		(real):16m52.865s	(real): 7m9.830s
32 threads		(clock): 1758.220093 sec time: 2214.0000 sec (chrono) : 2208.0000 s (real):36m53.477s	(clock): 665.390015 sec time: 552.000000 sec (chrono) : 552.0000sec real : 9m12.475s

Observations from the result:-

1) The program performed better for parallel version of program.
2) Execution with 2 threads was the most effective one. We got better results compared to serial version and other versions of the parallel code.
3) Persistent version of parallel programs were better than disposable threads because in case of disposable threads there would be an overhead of creating, deleting the threads for each iteration of convergence loop.
4) My results conflicted with my expectation. I expected when we use more threads we would get better results and less execution time. NOTE: May be the resources were busy when I used like core, CPU utilization etc. If it was a dedicated individual system then I would be the only user and would get better results with say 8 threads. Even with this if there are more threads and less cores, the core utilization would be more and give a bad result.
5) Unexpected anomalies: When you have less cores and try to increase number of threads, hoping for better improvement, the cores are limited and there would be an overhead of resource usage. You may not get the timing performance improvement you expected.
6) I noticed that loop-time, Chrono and real time are almost same. Real time would give a better idea for our results because it seems to be actual time required for program to complete execution. This time matches almost equal to calculated Chrono and loop-time.

* My parallel code for 8,16 and 32 threads are strange. May be the system was really busy when I tried to execute. May be the resources were busy when I used like core, CPU utilization etc. If more users are using the system, everyone would be competing for the hardware and hence we would get a poor result.