Seminar 5- MultiThreading- OMP

Activity 1

Evaluated the performance of the three implementations using three different sizes. Thread

Number was kept at 16 threads for both OpenMP and PTThread. The results were-

For size of 1000:

* Sequential- 22ms
* OpenMP- 21,234ms
* PTThread- 1196ms

Sequential performing best when size is 1000.

For size of 100,000:

* Sequential- 1868ms
* OpenMP- 29,830ms
* PT Thread- 2751ms

Sequential is yet again performing best when size is 100,000.

For size of 1,000,000,000:

* Sequential- 22,990,641ms
* OpenMP- 36,301,493ms
* Pt Thread- 12,052,122ms

PT Thread is performing best for such a large size of 1,000,000,000.

From the results above, it can be concluded that Sequential works best with smaller sizes. But as size grows larger, PT thread becomes a better choice. Meanwhile, OpenMP is performing the worst in all the size categories.

Activity 2

1. Default(none) does not allow any variables to be shared among the threads. That is why we are getting an error.

Using shared(size): Gives compilation error. This is because we did not share total, v1,v2 and v3.

Using shared(v1): Gives compilation error. This is because we did not share total, size,v2 and v3.

Using shared(size,total,v1,v2,v3): This works as all the variables in the parallel program are shared now.

Text

Description automatically generated

Using private(total): Now the total variable from shared to private. This prevents the total value from getting incremented and I get 0 as the final output.

Conclusion: Only data shared between threads will be executed in the parallel program. Or else, we are bound to get different values.

1. Text

   Description automatically generated

USING ATOMIC DIRECTIVE

1. Text

   Description automatically generated

Úsing Reduction

1. Text

   Description automatically generated

Using Critical region

I get similar results which is close to exact for all the cases.

1. Only changing the scheduling technique to dynamic affected the execution time. For higher chunks, dynamic’s performance was similar to static or guided. But for lower chunks, the execution time significantly increased. Below is the example when I used size 2 chunk and size 100 chunk for dynamic. For size 2 chunk execution time=876160. For size 100 chunk execution time=216559.

Text

Description automatically generated

Text

Description automatically generated