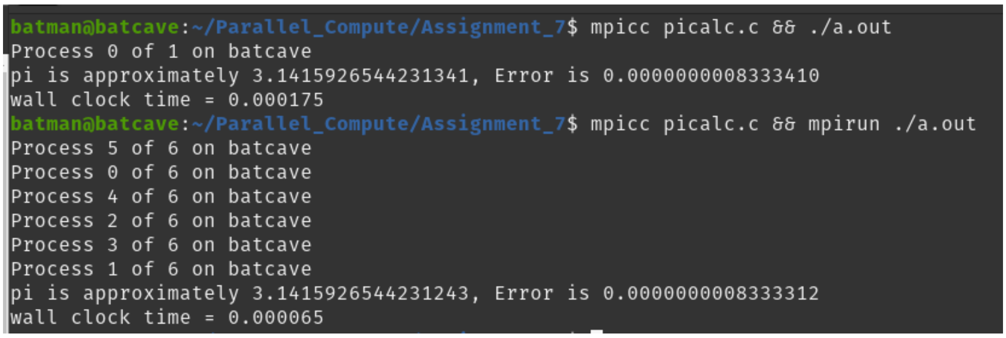
Assignment 7 : Lab 6

Date	@November 4, 2022
Course	Parallel Computing
Finished?	□
Tags	Lab

Programming Project : Working with MPI

Part 1 - Executing MPI enabled starter code

- Compiling and Running the `pical.c` file using the MPI and without MPI renders the following output for the local system.



```
batman@batcave:~/Parallel_Compute/Assignment_7$ mpicc picalc.c && ./a.out
Process 0 of 1 on batcave
pi is approximately 3.1415926544231341, Error is 0.0000000008333410
wall clock time = 0.000175
batman@batcave:~/Parallel_Compute/Assignment_7$ mpicc picalc.c && mpirun ./a.out
Process 5 of 6 on batcave
Process 0 of 6 on batcave
Process 4 of 6 on batcave
Process 2 of 6 on batcave
Process 3 of 6 on batcave
Process 1 of 6 on batcave
pi is approximately 3.1415926544231243, Error is 0.0000000008333312
wall clock time = 0.000065
```

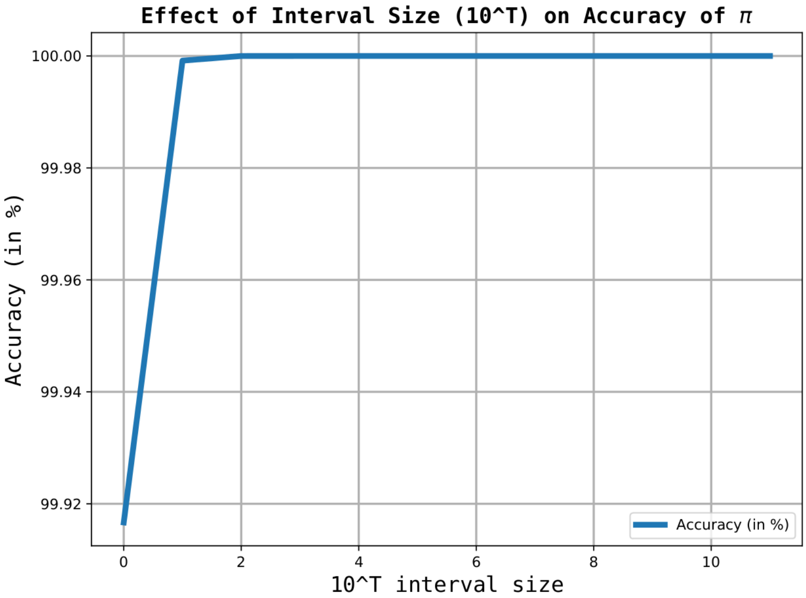
MPI Execution time on 6 processors vs using only 1 process

- Running the MPI enabled program on a 6 processes, it takes total wall time of 0.000065 seconds.
- While running the program to approximate the value of π using only 1 process takes about 0.000175 seconds. This way of execution runs 2.69x slower than that of using MPI.

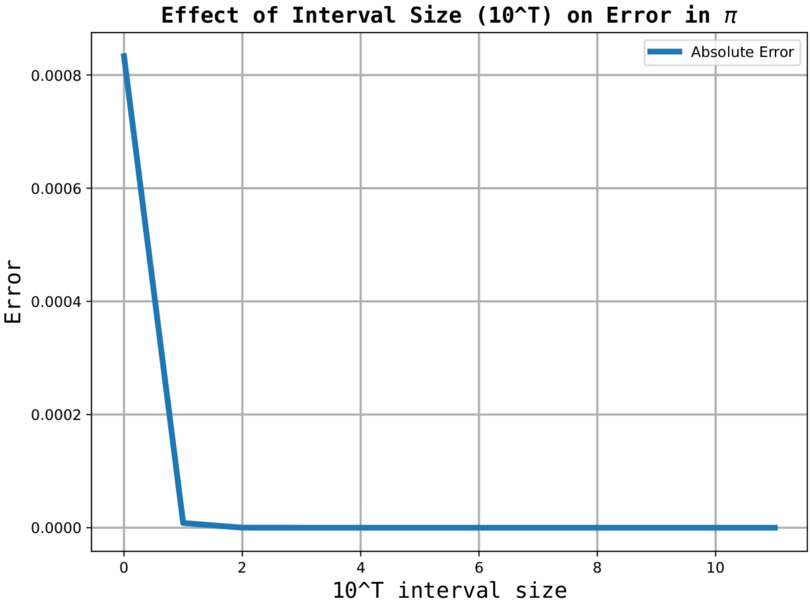
Part 2 - Effect of intervals on Accuracy

- Running the code on various intervals → from 10 to 10^12 and obtaining the results for visualization

Plot : Accuracy vs Interval Size



Plot : Error vs Interval Size



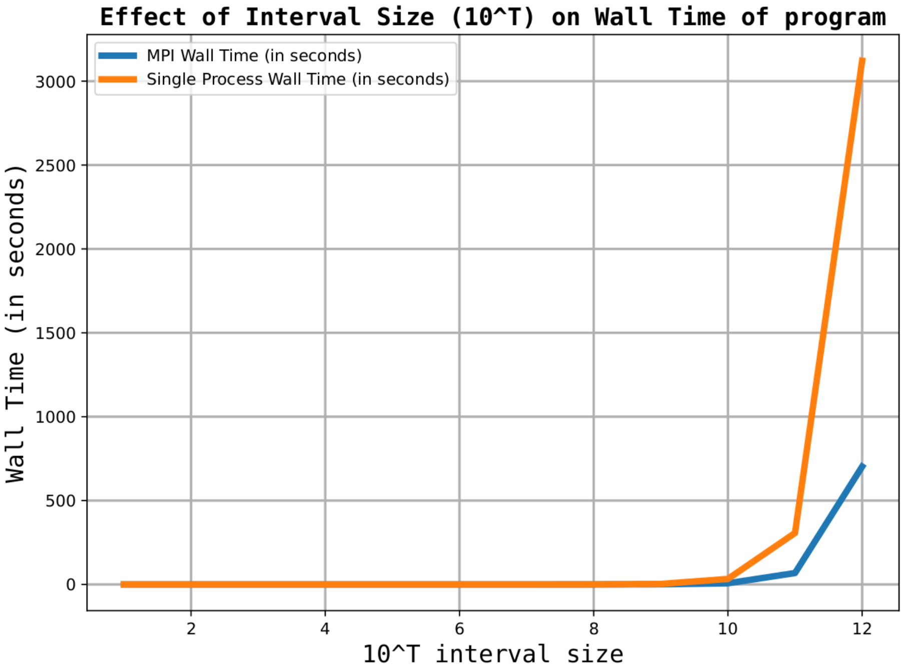
Plot : Wall time comparison vs Interval Size

→ As we can see from the plots below, executing the program on a single process is only faster than MPI approach till 10^4 computations.

→ As soon as the grid size grows more than 10^5, the MPI approach is significantly (almost one order of magnitude) faster than single processor approach.

→ The reason MPI is slower for less grid points can be attributed to the process being input-output bound. It takes more time just to pass the data from process to process and collect the effective sum back to node 0.

→ But as the problem size grows, and overall execution becomes computationally bound, MPI approach outperforms single process execution by a huge margin.



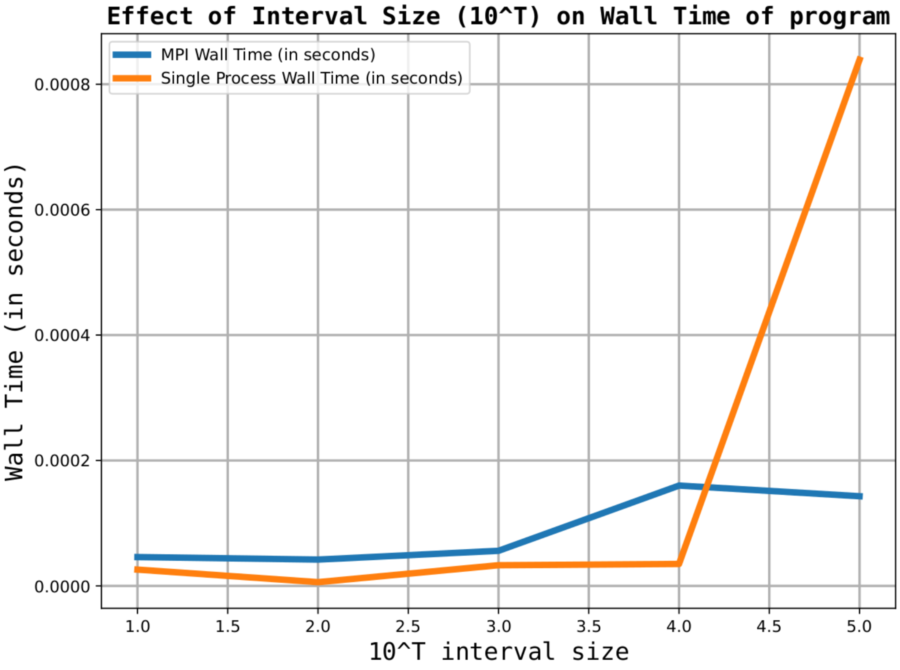


Table : Interval Size | Accuracy | Error | Wall Time Comparison

Index	Power of 10	Error (MPI)	Wall Time (MPI)	Wall Time (Single Process)
1	1	0.0008333314113056	0.000046	0.000026
2	2	0.0000083333333309	0.000042	0.000006
3	3	0.0000000833333331	0.000056	0.000033
4	4	0.0000000008333312	0.000160	0.000035
5	5	0.0000000000083413	0.000143	0.000839
6	6	0.0000000000001048	0.001248	0.003061
7	7	0.0000000000000226	0.010694	0.056281
8	8	0.0000000000001208	0.111374	0.306743
9	9	0.0000000000000884	0.735355	3.078677
10	10	0.0000000013865686	6.882624	32.805832
11	11	0.0000000017674591	67.974126	307.079831
12	12	0.0000000002392655	702.340895	3121.972127