Aum Sri Sai Ram

MTCS 103(P): Practical Parallel Processing

Lab Work II

Opening date: 14th Sept 2021

Due date: 16th Sept 2021

Follow Academic Integrity and Honour Code.

Execute the sequential and parallel programs for evaluating Pi.
Sketch of the programs given below.
Record your observations and results obtained for various No. of threads used and the various size of grid(No. of rectangles) used to evaluate.

Sequential Code:

```
static long num_steps= 100000;
double step;
int main ()
{
  int i;
  double x, pi, sum = 0.0;
  step = 1.0/(double) num_steps;

  for (i=0;i< num_steps; i++)
  {
        x = (i+0.5)*step;
     sum = sum + 4.0/(1.0+x*x);

  }
  pi = step * sum;
}
```

Parallel Code:

```
#include <omp.h>
#include <stdio.h>
#include <stdlib.h>
```

```
#define NUM_THREADS 8
static long steps = 1000000000; double step;
int main(int argc, const char* argv[])
{
   double pi = 0.0; int nthreads;
   step = 1.0 / (double)steps;
   double sum[NUM_THREADS];
   start = omp_get_wtime();
   omp_set_num_threads(NUM_THREADS);
#pragma omp parallel
   {
      double x; int id, i, nthrds;
      id = omp_get_thread_num();
      nthrds = omp_get_num_threads();
      if (id == 0) nthreads = nthrds;
      for (i = id, sum[id] = 0.0; i < steps; i = i + nthreads) {
         x = (i + 0.5) * step;
         sum[id] += 4.0 / (1.0 + x * x);
      }
   }
   for (int i = 0; i < nthreads; i++)
      pi += sum[i] * step;
   delta = omp_get_wtime() - start;
   printf("PI = %.16g computed in %.4g seconds with %d threads.", pi, delta,
NUM_THREADS);
}
```