OpenMP: Hands-on

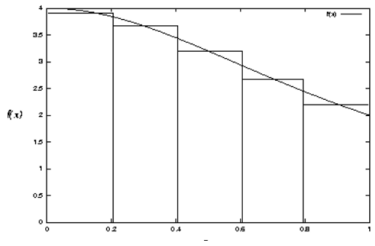
March 28, 2019

Exercise 1: Approximate π

π can be approximated using the following formula:

$$\pi = 4 \int_0^1 \frac{1}{1+x^2} \mathrm{d}x \approx 4 \sum_{i=0}^{N-1} h f(x_{i+\frac{1}{2}})$$

where $h = \frac{1}{N}$ and $f(x) = \frac{1}{1+x^2}$. The integral is approximated by slices of $h \cdot f(x)$ at midpoints. Use $N = 100000000$ (or bigger).



1. Implement the approximation in serial and then parallelize the code using OpenMP
2. Measure the time it takes to compute the sum using `omp_get_wtime()` and output the time to solution.
3. Create a job script which runs the program with 1, 2, 4, 8, 16, and 20 threads. Submit the script using sbatch. Make a plot of the time scaling curve obtained by registering the time of simulation at increasing the number of threads.
4. Try out using a critical section, atomic and reduction to protect the summation variable. Observe the scalability of each variant.

Exercise 2: OpenMP Loop Schedules

In this exercise you will create a visualization of two different OpenMP schedules using different chunks. Collect the various outputs so that can be visualised altogether. Briefly comment the results obtained.

1. First we create an array of size 250 and iterate over all elements in a loop. In the loop we save the current thread index in the array. Here is the serial version:

```
const int N = 250;
int a[N];
int thread_id = 0;
int nthreads = 1;

for(int i = 0; i < N; ++i) {
    a[i] = thread_id;
}
```

2. Use the `print_usage(a, N, nthreads)` function (provided in the example folder) to visualize the result. It takes the array a as first parameter, its size N as second parameter and the total number of threads as *nthreads*. When running only on one thread, this is the output:

Listing 1: Example Output with 1 thread

```
serial:

0: ************************************************************************
```

3. Parallelize the serial program using different schedules and use `print_usage()` to visualize how work is being distributed:

   - static
   - static, with chunk size 1
   - static, with chunk size 10
   - dynamic
   - dynamic, with chunk size 1
   - dynamic, with chunk size 10

   Write this program with only one **`omp parallel`** region. Only one thread must be printing the output!

4. Write a job script and run your code on the Ulysses cluster using 10 threads.

Listing 2: Example Output with 4 threads and static schedule

```
schedule(static):

0: ******************
1:                    ******************
2:                                       ******************
3:                                                          ******************
```