Setup

Log in to Salomon

Update (or clone) the git repository
$ git clone https://code.it4i.cz/jansik/SummerOfHPC.git
$ git pull

Get interactive allocation on Salomon
$ qsub -A DD-17-18 -q R10752[89-94] -l select=1:ncpus=8,place=scatter -I

$ qstat –a
$ qstat –u username –n

Lab 1

OpenMP Hello world!

The goal is to write and run a simple Hello world program using OpenMP threads

OpenMP Hello world

File: helloworld.c

Tasks:

1. Modify helloworld.c so that it it runs with multiple OMP threads, writing out ***Hello world thread from thread XX of YY***
2. Change number of threads to 6
3. Use #pragma master to write from master thread only, find TID of master thread
4. Use #pragma single to write from a single thread
5. Write 3x ***Hello world*** line from 3 different sections, find TID of threads executing the sections

Ingredients:

- #pragma omp parallel
- omp_get_thread_num()
- omp_get_num_threads()
- clause private(var), shared(var)
- export OMP_NUM_THREADS=
- #pragma omp master
- #pragma omp single
- #pragma omp sections
- #pragma omp section

Compilation:

- Icc –qopenmp helloworld.c –o helloworld.x

Lab 2

OpenMP private / shared variables

The goal is understand behavior of private and shared variables

OpenMP private / shared variables

File: variables.c

Tasks:

1. Review variables.c, compile and run, rationalize the results.
2. Make var1 private
3. Make var1 shared
4. Make var1 firstprivate
5. Make var1 private, set value to TID from within the parallel region
6. Make var1 shared, set value to TID form within parallel region
7. Use var1 to sum up the TIDs, using reduction clause. Observe var1 values and pointers inside and outside of parallel region.
8. **Rationalize the results!**

Ingredients:

- #pragma omp parallel
- omp_get_thread_num()
- omp_get_num_threads()
- clause private(var), shared(var), firstprivate (var)
- clause reduction(+:var)

Compilation:

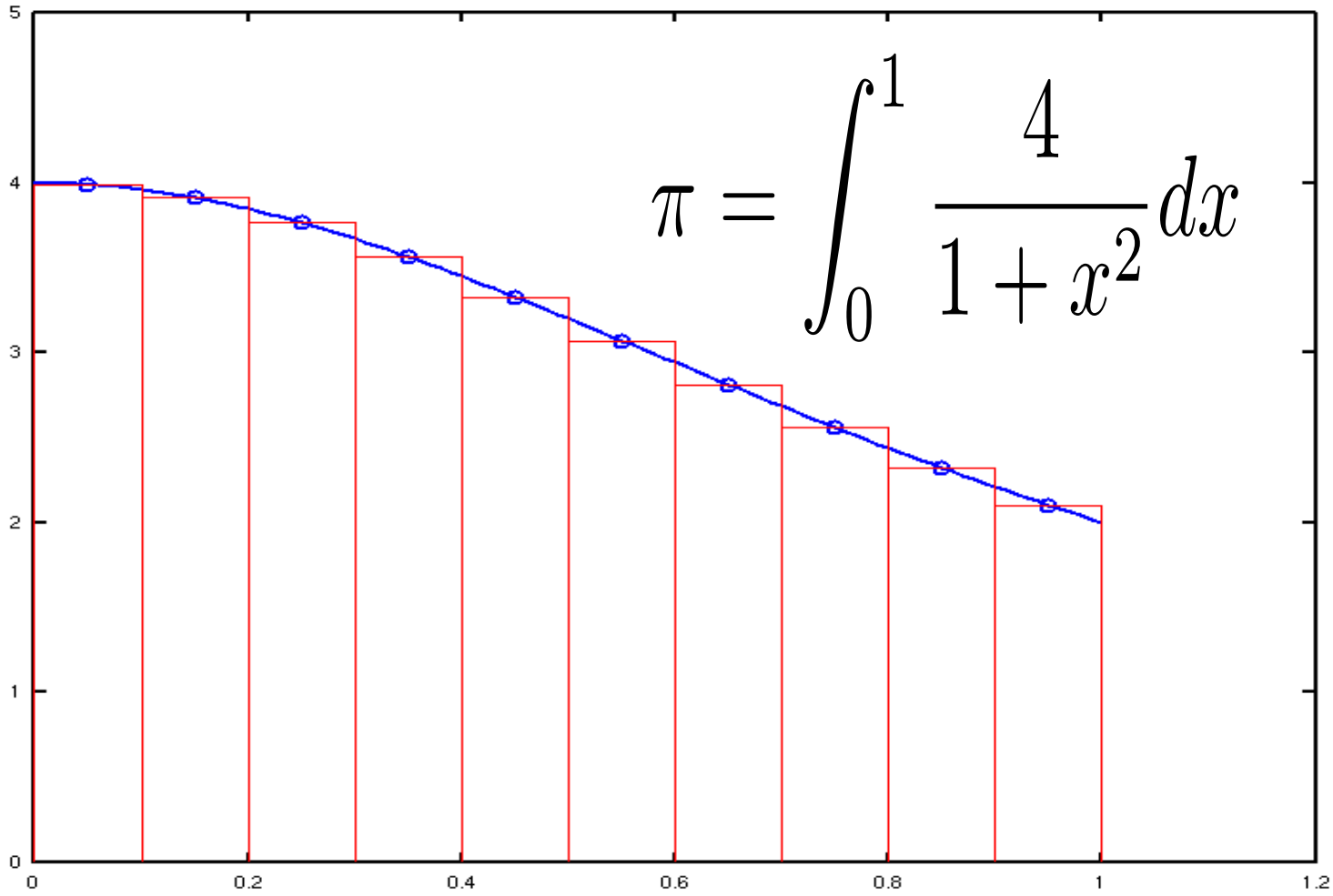
- Icc –qopenmp helloworld.c –o helloworld.x

Lab 3

Parallelize pi integration using OpenMP

The goal is to have OpenMP parallel program integrating value of pi.

OpenMP pi integration



OpenMP pi integration

File: pi3serial.c

Tasks:

1. Review pi3serial.c, compile and run, rationalize the results.
2. Parallelize using #pragma omp parallel for, use reduction clause
3. Find out which iterations are executed by which thread
4. Parallelize using #pragma **omp parallel only,** use reduction clause

Ingredients:

- #pragma omp parallel
- omp_get_thread_num()
- omp_get_num_threads()
- clause private(var), shared(var)
- clause reduction(+:var)
- #pragma omp parallel for

Compilation:

- Icc –qopenmp helloworld.c –o helloworld.x

Thank you!