High Performance Computing (Master) in WS19

Exercise 3: shared memory parallelization using OpenMP

Deadline: Jan 7, 2020, 16:00

Status: Tuesday 3rd December, 2019, 14:31

Supervisor: Prof.Dr. G. Haase, `gundolf.haase@uni-graz.at`

We propose to read the OpenMP summary[1] as well as the OpenMP tutorial from LLNL[2]. See also this guide[3] by Joel Yliluoma and slides[4] by Annika Hagemeier.

1. Download[5] the template for the inner product of vectors (example II-A). (4 Pkt.)
   - Compile and run it.
     Use `GCC_` or `ICC_`, the `clang++-9` has problems with *mylib.h:38*.
   - Try several schedule types and junk sizes in *mylib.cpp:13*, see §4.1 and §2.7.1 in the OpenMP specifications.
   - Calculate the speedup for different number of cores (incl. hyperthreading)
     Use function `omp_set_num_threads(tn)` in your main function or call `export OMP_NUM_THREADS=tn` from the shell in order to run the code on `tn` parallel threads.
   - Try `omp_get_wtime()`, `omp_get_num_procs()` and `omp_in_parallel()`.
   - Write a second function `scalar` using a parallel environment `#pragma omp parallel` without `for` .
   - Write a function similar to function `reduction_vec(int n)` that appends the private vectors instead of adding them, see p. 74ff in slides by A. Hagemeier.
2. Parallelize task (B) (Data-IO; means and max/min of vector elements) from Exercise 1. (4 Pkt.)
3. Parallelize example (F) (Goldbach: count [, pairs]) from Exercise 1. (4 Pkt.)
4. Parallelize examples (B)-(D) from Exercise 2. (4 Pkt.)
   Take care that you use the OMP timing routine `omp_get_wtime()` !
5. Copy your sequential Code for example (E) (code[6], docu[7]) or (F) (code[8], docu[9]) from Exercise 2 and parallelize it. (8 Pkt.)

This document will be extended by further advices, links, etc.

Tuesday 3rd December, 2019

[1] http://www.openmp.org/wp-content/uploads/OpenMP-4.0-C.pdf

[2] https://computing.llnl.gov/tutorials/openMP/

[3] https://bisqwit.iki.fi/story/howto/openmp/#IntroductionToOpenmpInC

[4] https://www.fz-juelich.de/ias/jsc/EN/AboutUs/Staff/Hagemeier_A/docs-parallel-programming/OpenMP-Slides.pdf?__blob=publicationFile

[5] http://imsc.uni-graz.at/haasegu/Lectures/Math2CPP/Codes/shm/demo_skalar.tar

[6] http://imsc.uni-graz.at/haasegu/Lectures/Math2CPP/Codes/shm/jacobi_oo_stl.tar

[7] http://imsc.uni-graz.at/haasegu/Lectures/Math2CPP/Codes/shm/jacobi_oo_stl/html

[8] http://imsc.uni-graz.at/haasegu/Progs/gh_hack.tar

[9] http://imsc.uni-graz.at/haasegu/Progs/gh_hack/html