High Performance Computing (Master) in WS22

Exercise 3: shared memory parallelization using OpenMP

Deadline: Jan 3, 2024, 16:00

Status: Wednesday 4th October, 2023, 11:39

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, p.25[4] by Annika Hagemeier.

Take care that to use the C++ system_clock[5], see example thread_17, or the OMP timing routine `omp_get_wtime()` !

1. Download[6] the template for the inner product of vectors (example II-A). (4 Pkt.)
   - Compile and run it.
   - 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. Compare the run time of the OpenMP approach with the execution policies[7] in C++17. (4 Pkt.)
3. Parallelize example (F) (Goldbach: count [, pairs]) from Exercise 1. (4 Pkt.)
4. Parallelize examples (B)-(D) from Exercise 2. (4 Pkt.)
   - Write in (A) also a parallel function that realizes the summation $s = \sum_{k=0}^{n-1} x_k$.
   - Compare the speedup of the sum as well as the inner product for various $n = 10^k$, $k \in [3, 8]$.
5. Copy your sequential Code for example (E) (code[8], docu[9]) or (F) (code[10], docu[11]) from Exercise 2 and parallelize it. (8 Pkt.)

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

Wednesday 4th October, 2023

---

[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://docplayer.org/19676777-Einfuehrung-in-openmp.html`
[5] `https://en.cppreference.com/w/cpp/chrono/system_clock`
[6] `http://imsc.uni-graz.at/haasegu/Lectures/Math2CPP/Codes/shm/demo_skalar.zip`
[7] `https://www.bfilipek.com/2018/06/parstl-tests.html`
[8] `http://imsc.uni-graz.at/haasegu/Lectures/Math2CPP/Codes/shm/../seq/jacobi_oo_stl.zip`
[9] `http://imsc.uni-graz.at/haasegu/Lectures/Math2CPP/Codes/shm/../seq/jacobi_oo_stl/html`
[10] `http://imsc.uni-graz.at/haasegu/Progs/gh_hack.zip`
[11] `http://imsc.uni-graz.at/haasegu/Progs/gh_hack/html`