McXtrace

McStas

Heading toward the GPU's….

Peter Willendrup (pkwi@fysik.dtu.dk)

McCode on GPU?

1st prototype, “null”-instrument with only one component.

Based on NVIDIA compiler technology, PGCC and OpenACC pragmas

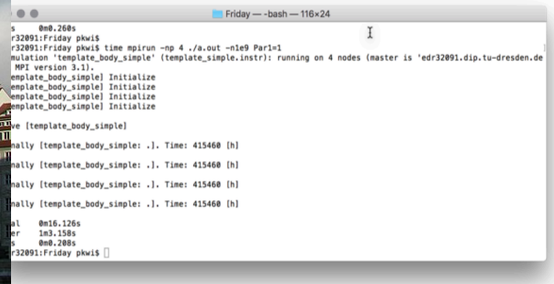
More to come in 2019!

McStas / McXtrace instrument

Port heavy part of component to GPU (Shows some potential)

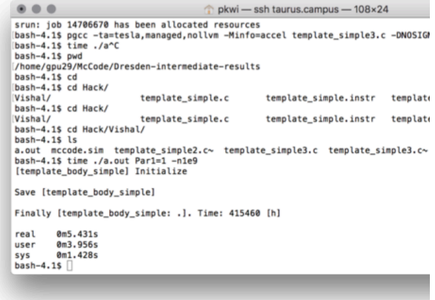
Port neutron loop to GPU
- Lots of code
- Code generation
- Compiles! Speed up today

CPU MPI 4 cores (95 % usage)



16.12 s (Single core 56.0 s)

GPU 5% usage



5.43 s

McStas + McXtrace heading for the GPU… October 2019

Rewritten code-generation with automated additions of OpenACC pragmas.

Quite transparent wrt. CPU vs. GPU

First simulations with meaningful output

GPU Hackathon

Share Tweet Like 0

Introduction

CSC is in collaboration with Nvidia and the E-CAM European HPC center of Excellence arranging a 3-day GPU hackathon. The GPU hackathon is a coding event in which teams of developers port their applications or kernels to run on GPUs, or optimize their applications that already run on GPUs. In particular the hackathon focuses on applications that can scale up to multiple GPU nodes.

We are looking for teams of 3-4 developers. Collectively the team should know the application intimately. Please keep in mind that we are looking for teams with plans to develop GPU code – not to just run their code on GPUs. During the hackathon each team is supported by one mentor with in-depth GPU programming expertise.

At CSC the new Puhti-AI partition provides 80 nodes with 4 NVidia Volta GPUs each. This system provides in total more than 2 petaflops of performance. This system is available during the course and accepted teams will also have access to the system beforehand to do some initial porting of the applications to Puhti.

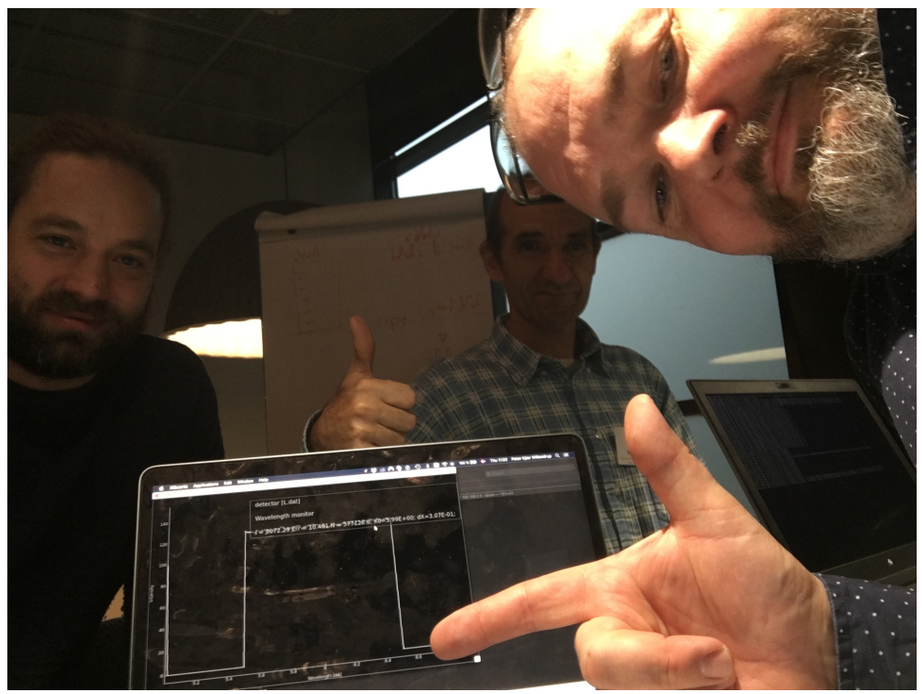
Date: 16.10.2019 9:00 - 18.10.2019 17:00

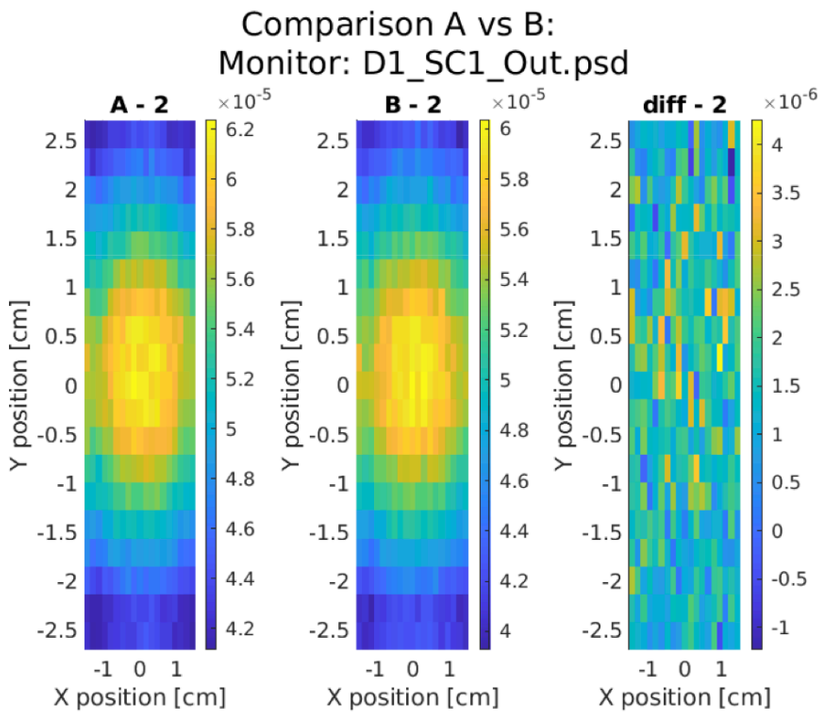
Location details: The event is organised at the CSC Training Facilities located in the premises of CSC at Keilaranta 14, Espoo, Finland. The best way to reach us is by public transportation; more detailed **travel tips** are available.

Language: English

Price:
- Free for Finnish universities, universities of applied sciences and governmental research institutes.
- Free for others.

The fee covers all materials, lunches as well as

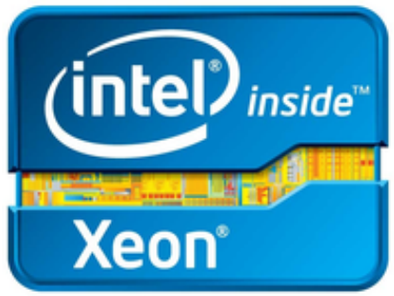


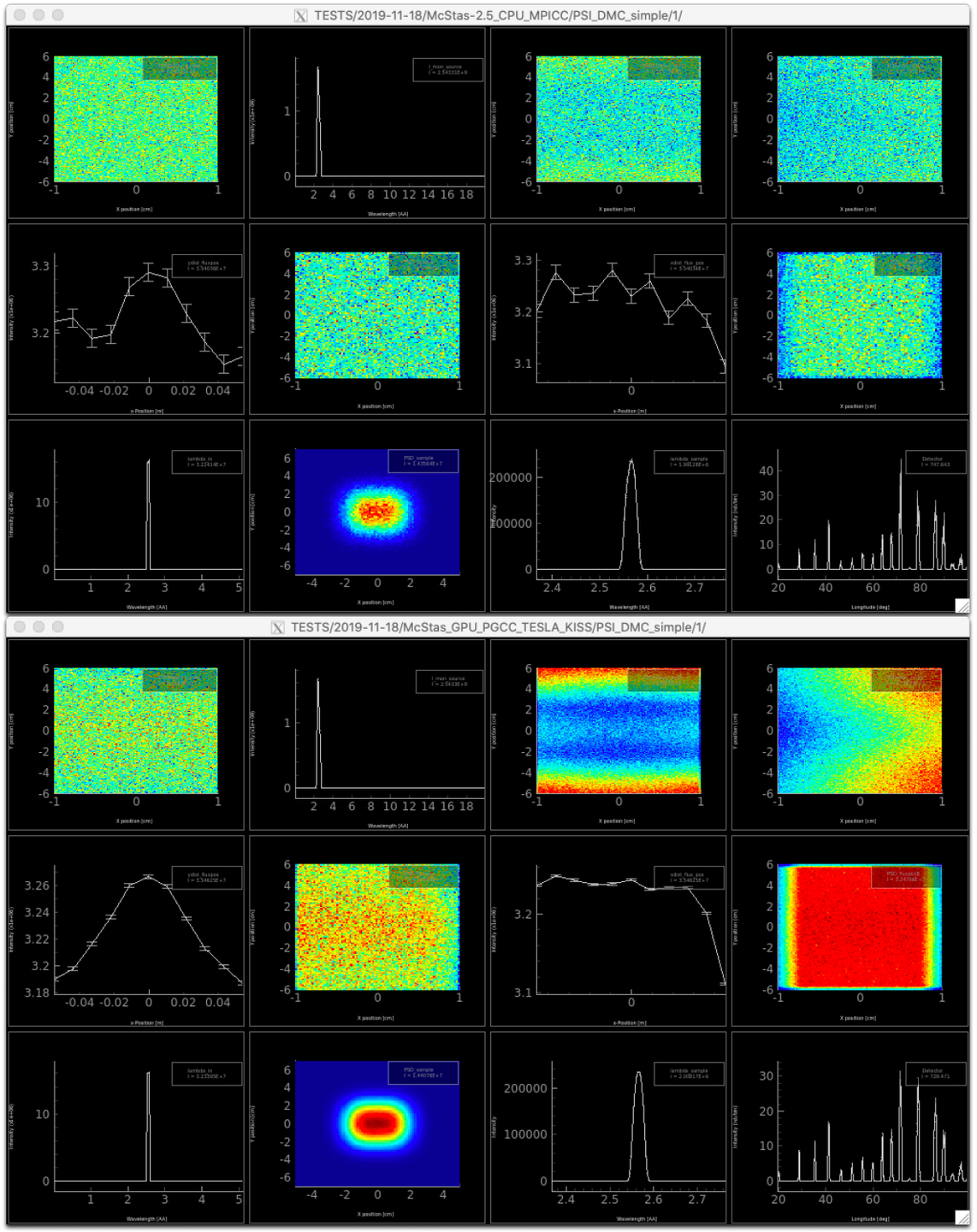


McStas + McXtrace heading for the GPU… Now

9 instruments fully ported, also realistic ones like PSI_DMC

10-core MPI run,
1e7 in 2 secs





~ 2 orders of magnitude wrt. a
single, modern CPU core

Tesla V100 run,
1e9 in 22 secs

Figure 1

File Edit View Insert Tools Desktop Window Help

Performance in the PSI_source case - execution time

PGCC Quadro T2000 doubles -O0
PGCC GeForce GTX 1080 -O0
PGCC Quadro T2000 floats -O0
PGCC GeForce GTX 1080 floats -O0
PGCC Volta V100 -O0
PGCC multicore -O2, 8 cores
PGCC -O2 cpu serial
gcc -O2 cpu serial
PGCC -O2 mpi, 8 cores
extrapolation gcc cpu
extrapolation,PGCC cpu

Wall-clock time / s

10^{3} 10^{2} 10^{1} 10^{0} 10^{-1} 10^{-2}

10^{4} 10^{5} 10^{6} 10^{7} 10^{8} 10^{9} 10^{10} 10^{11} 10^{12}

Problem size / neutrons

CPU

MPI

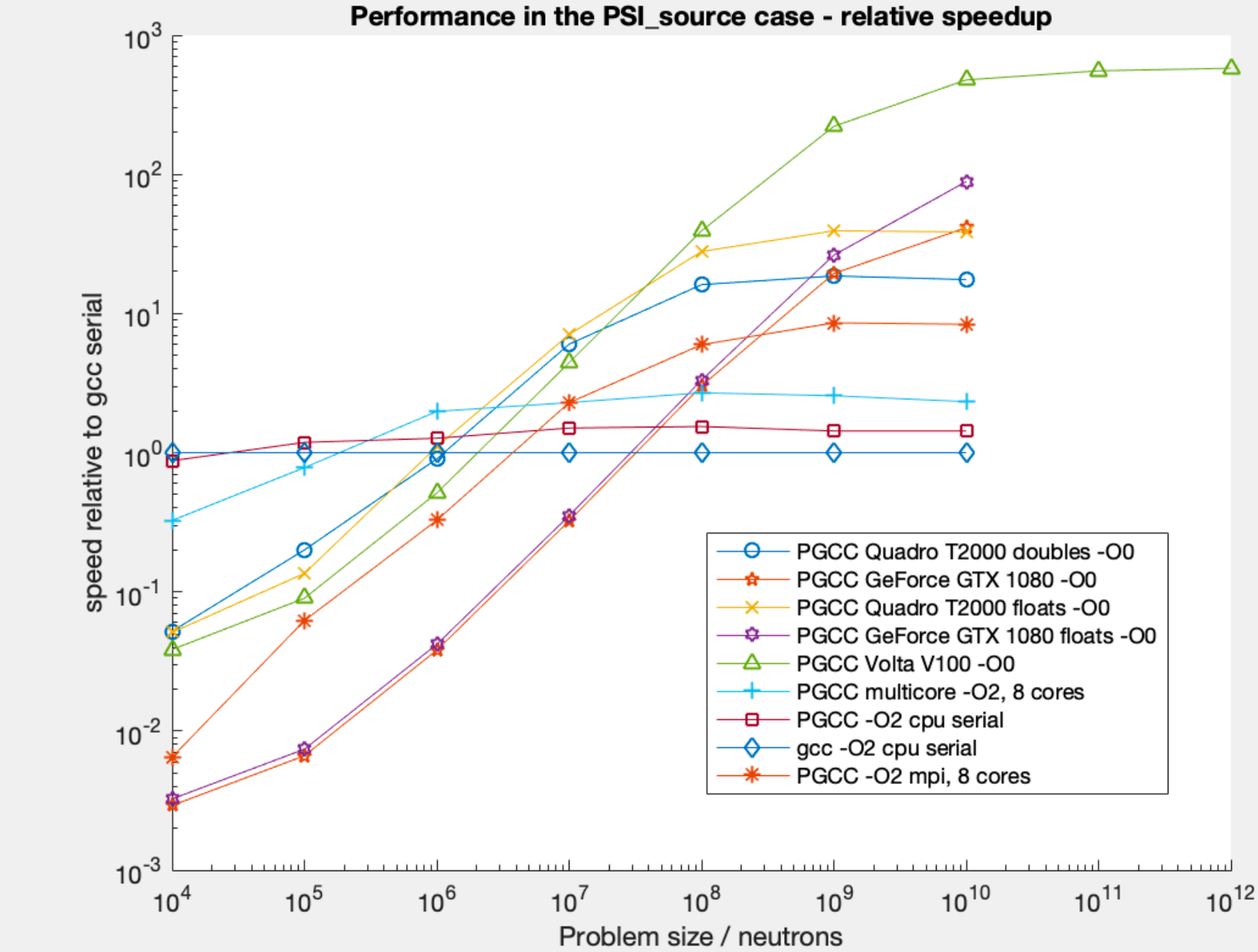
V100 (datacenter)

Quadro T2000 (laptop)

Geforce GTX 1080 (desktop)

Figure 2

File Edit View Insert Tools Desktop Window Help



First step: McStas 3.0 - next generation code generator

```
GPU port instrument statistics:
- 21 out of 142 enabled (184 total) in mcstas-3.0 branch - these work:
        PSI_DMC_simple.out
        template.out
        PSI_source.out
        micro.out
        h8_test_legacy.out
        Samples_vanadium.out
        Samples_Incoherent_off.out
        ISIS_TS2_Brilliance.out
        BNL_H8_simple.out
        linup-1.out
        template_simple.out
        FZJ_KWS2_Lens.out
        linup-3.out
        Tomography.out
        nano.out
        Vout_test.out
        templateLaue.out
        ISIS_TS1_Brilliance.out
        mini.out
        linup-2.out
        HZB_FLEX.out
GPU port component statistics:
       2 Al_window
      45 Arm
       1 Beamstop
       1 Bender
       2 Brilliance_monitor
       6 Collimator_linear
       1 Divergence_monitor
       7 E_monitor
       9 Guide
       1 Guide_channeled
       2 Guide_curved
       8 Guide_simple
       1 Guide_tapering
       1 Hdiv_monitor
       2 ISIS_moderator
       2 Incoherent
       3 L_monitor
       1 Lens_simple
       4 Monitor
       3 Monitor_nD
       1 Monochromator_2foc
       1 Monochromator_curved
      28 Monochromator_flat
      33 PSD_monitor
       2 PSD_monitor_4PI
       3 PSDlin_monitor
       1 PowderN
      12 Progress_bar
       1 Selector
       1 Single_crystal
      46 Slit
       2 Source_Maxwell_3
       1 Source_div
       2 Source_gen
      11 Source_simple
       3 V_sample
       1 V_selector
- i.e. 37 out of 207 total in mcstas-3.0 branch
```

- Limited-functionality “beta” release to be made public soon (jan-feb) after **2.6 (december)**
  - Expect bugs!
  - Only a subset of components / instruments
  - Event interchange with 2.6 possible via MCPL
- Main purpose: **get this working in ‘the wild’**
  - Your instruments will likely **require (limited) rewriting**
    - E.g. the declare section **cannot include assignments**
  - Your own components will **likely require rewriting**
    - E.g. the declare section **cannot include assignments**
    - Arrays must be declared/initialized using a new set of functions (i.e. not double PSD_I[nx][ny] with definition parms)
- Hence some backward compatibility is lost and we need **to increment major release #**

McStas McXtrace