Porting McStas to GPU via code-generation and OpenACC pragmas
Main events on timeline of current developments

2017: E. Farhi
- Initial cogen modernisation

Fall 2018 onwards: J. Garde
- Further cogen modernisation and restructuring

March 2018:
- Participation at Dresden Hackathon.
- 1st “null” instrument prototype runs.
- Mentor: Vishal Metha

October 2019:
- Participation at Espoo Hackathon.
- First meaningful data extracted.
- Work on cogen and realising we need another RNG.
- Mentor: Christian Hundt

October 2019 onwards:
- J. Garde & P. Willendrup:
  - New RNG, test system, multiple functional instruments.

November-December 2019:
- First good look at benchmarks and overview of what needs doing for first release with limited GPU support.
McStas heading for the GPU… March 2018

McCode on GPU?

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

Based on NVIDIA compiler technology, PGCC and OpenACC pragmas

bootstrapping: 5 McStas/McXtrace developers @ 2018 GPU hackathon in Dresden

More to come in 2019!
Rewritten code-generation with automated additions of OpenACC pragmas.

Quite transparent wrt. CPU vs. GPU

First simulations with meaningful output

Speed on DELL with Quadro-card ~ on par with running on CPU with MPI
McStas heading for the GPU…
November 2019

9 instruments fully ported, also realistic ones like PSI_DMC*

10-core MPI run, $1 \times 10^7$ in 2 secs

Tesla V100 run, $1 \times 10^9$ in 22 secs

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

*Guide component without reflection-file support, SPLIT disabled, OFF geometry disabled
McStas heading for the GPU… November 2019 - first good look at performance. Wallclocks:

Idealised instrument with source and monitor only - i.e. without any use of the ABSORB macro.

(Likely a good indication of maximal speedup achievable.)
McStas heading for the GPU… November 2019 - first good look at performance. Speedup:

Idealised instrument with source and monitor only - i.e. without any use of the ABSORB macro.

(Likely a good indication of maximal speedup achievable.)

Looks like a factor of ~600

Renormalised to wall-clock of single-core gcc standard simulation
McStas heading for the GPU... December 2019 - today’s compilation status:

Numerical output with graphics: 
http://new-nightly.mccode.org/2019-12-06/2019-12-06_output.html

Statistics:
http://new-nightly.mccode.org/2019-12-06/stats.txt

(38 of 142 instruments, 62 of 207 components)
McStas 3.0 - next generation code generator - release plans

• Limited-functionality “beta” release to be made public soon (jan-mar) after 2.6 (January)
  • 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 #