PPS: Status Report and Plans for 2023 and Beyond

Oli Gutsche, Charles Leggett, Meifeng Lin

HEP-CCE All Hands Meeting
Apr 11 2023

Current Status of Projects

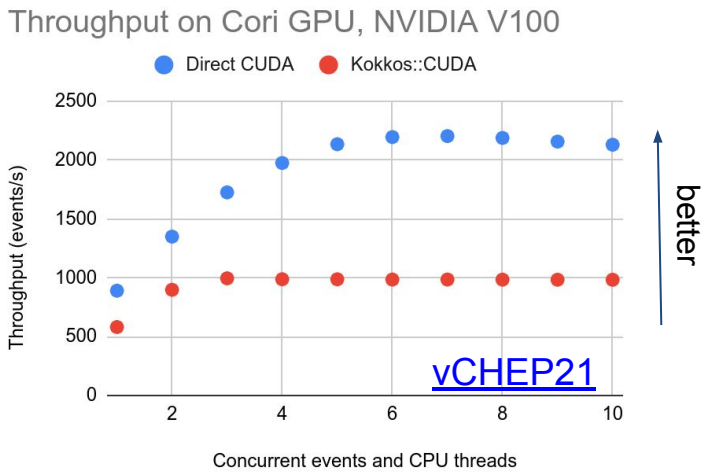
Tremendous work in the past 3 years!

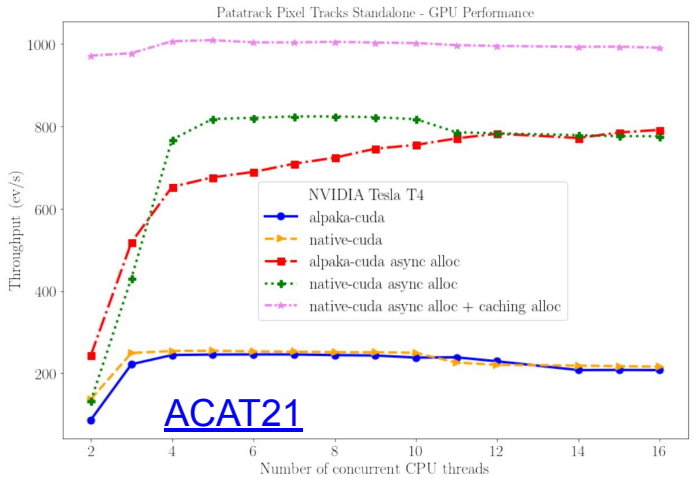
on top of rapidly shifting software and hardware

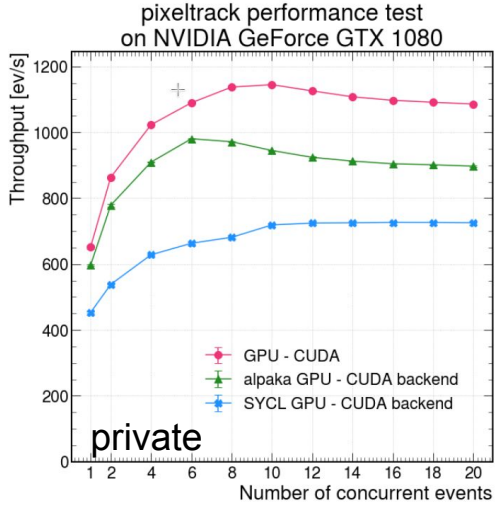
	Kokkos	SYCL	OpenMP	Alpaka	std::par
Patatrack	Done	Done	WIP	Done	Done compiler bugs
Wirecell	Done	Done	Done	**NO**	Done
FastCaloSim	regular: done group: done	regular: done group: **NO**	regular: done group: done	regular: done group: done	regular: done group: done
P2R	done	Done	OpenACC	Done	Done

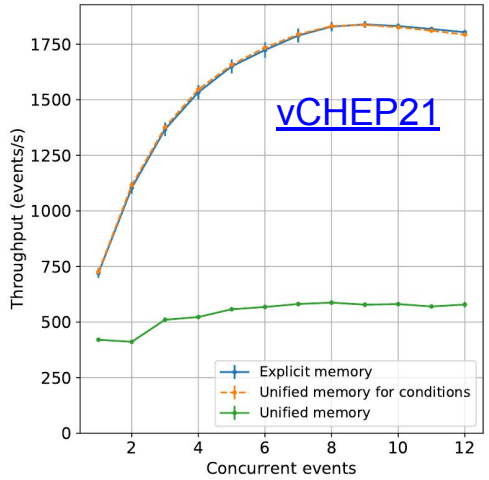
Patatrack

- CMS pixel reconstruction, developed originally in CUDA, extracted into a standalone application
  - Mimics relevant complexities of CMSSW framework and build system
  - An evolved version of the code used in CMS High Level Trigger since 2022
- Ported to
  - Kokkos, HIP, std::par, OpenMP (WIP); CUDA Unified memory by CCE
  - Alpaka by CERN group with some CCE involvement
  - SYCL by CERN group
- Working on consistent comparison between all for CHEP









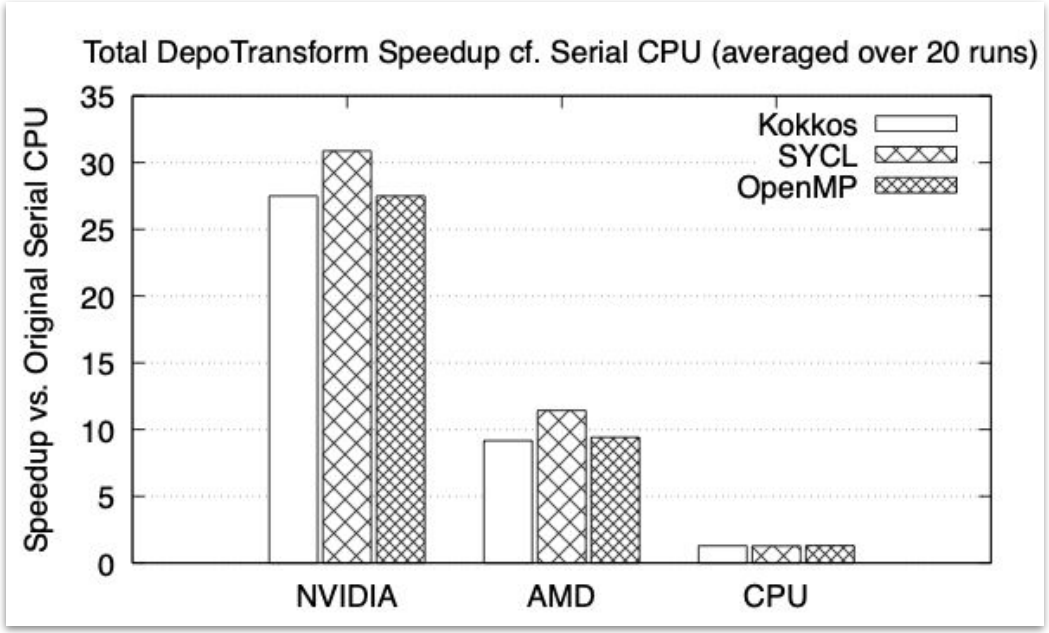
Wirecell Toolkit

Three major steps of LArTPC simulation

- **Rasterization:** depositions ⟶ patches (small 2D array, ~20×20)
  - # depo ~100k for cosmic ray event
- **Scatter adding:** patches ⟶ grid (large 2D array, ~10k×10k)
- **FFT:** convolution with detector response

Current Status:

- Restructured the code to expose more parallelism
- Wrappers to use optimized vendor libraries
- Ported to **CUDA (partial), Kokkos, SYCL, OpenMP** and **std::par** implementations
- Developed a stand-alone testing framework (without LArSoft dependence)
- Validated and benchmarked Kokkos, SYCL and OpenMP implementations
- std::par benchmarking ongoing



Speedup in DepoTransform compared to original CPU on NVIDIA V100, AMD Raedon Pro VII, and AMD Ryzen 24-core CPU with Kokkos, SYCL and OpenMP

Publications:

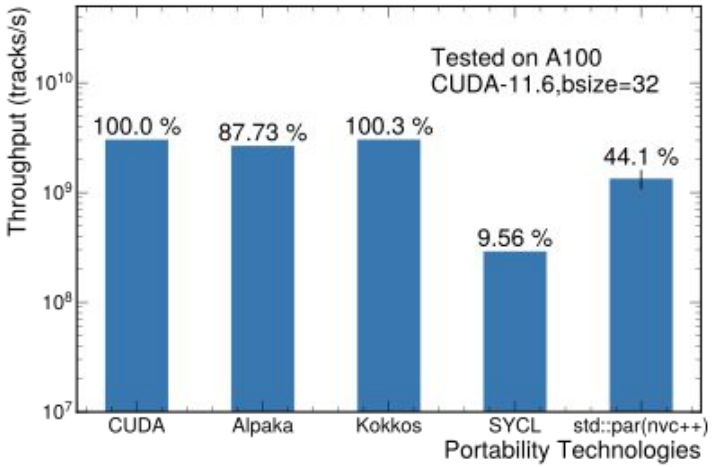
- Yu, Haiwang, et al. "Evaluation of Portable Acceleration Solutions for LArTPC Simulation Using Wire-Cell Toolkit." *EPJ Web of Conferences*. Vol. 251. EDP Sciences, 2021.
- Dong, Zhihua, et al. "Evaluation of Portable Programming Models to Accelerate LArTPC Detector Simulations." *Journal of Physics: Conference Series*. Vol. 2438. No. 1. IOP Publishing, 2023.
- Lin, Meifeng, et al. “Portable Programming Model Exploration for LArTPC Simulation in a Heterogeneous Computing Environment: OpenMP vs. SYCL," [arXiv:2304.01841 [hep-ex]]

p2r

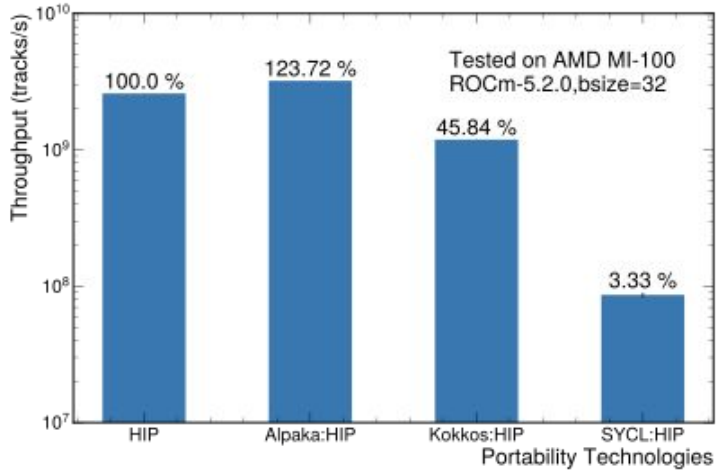
- Standalone track propagation+Kalman update kernels
  - Extracted from mkFit (vectorized-CPU full tracking application)
- Status:
  - Majority of porting done: CUDA, HIP, TBB (reference implementations) Alpaka, Kokkos, SYCL, std::par, OpenACC
  - Focus on benchmarking ***CPU*** results
  - GPU results submitted to ACAT proceedings: link
    - Present GPU+CPU results together with p2z in CHEP

NEW!!

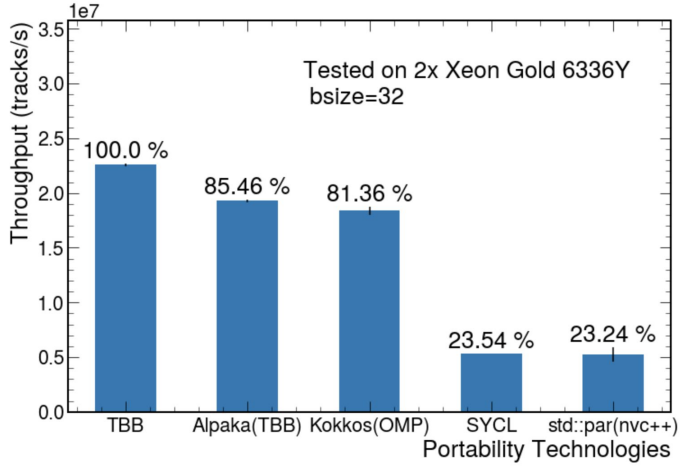
NVIDIA A100



MI-100



Xeon Gold 6336 CPU(48 core)



FastCaloSim

ATLAS Parametrized Calorimeter Simulation

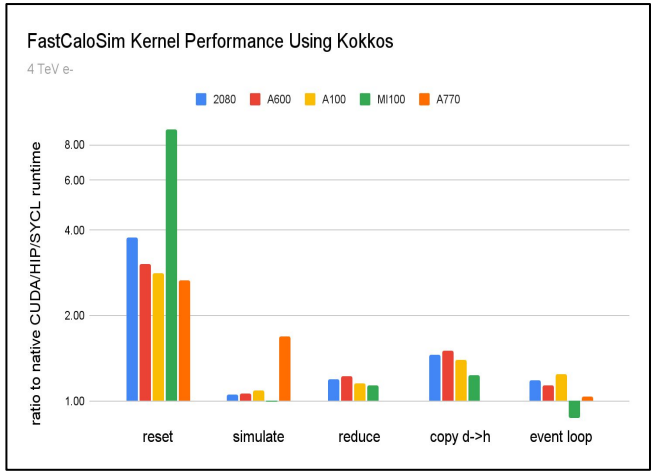
- "reset", "simulate", "reduce", "transfer"
- relatively uncomplicated kernels

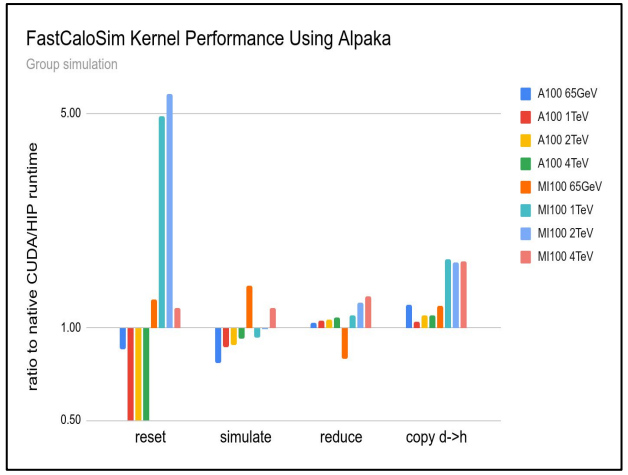
Publications:

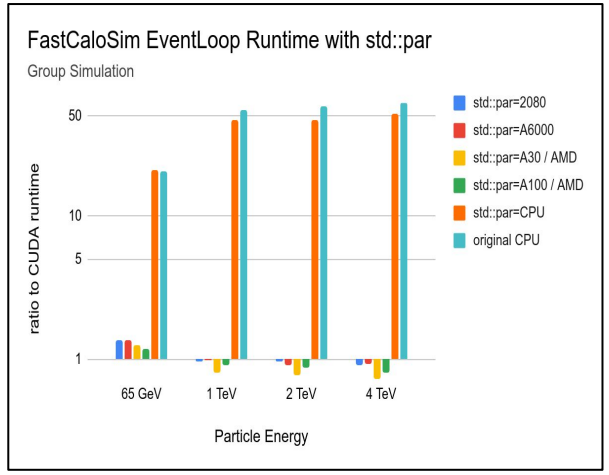
- Dong, Zhihua et al. "Porting HEP Parametrized Calorimeter Simulation Code to GPUs", *Frontiers in Big Data 4* (2021) 665783

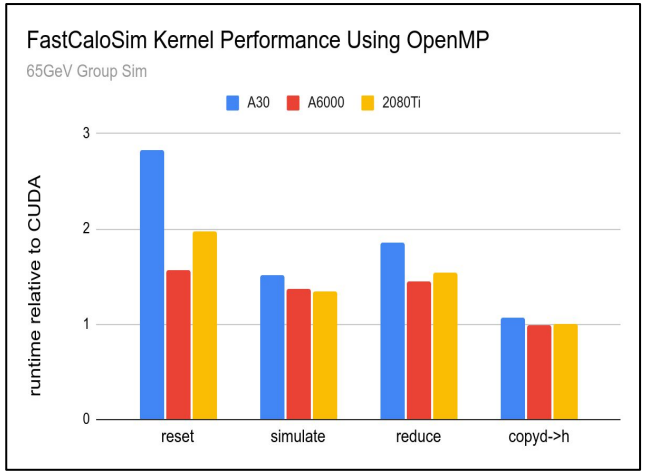
Current Status

- ported to Kokkos, SYCL (not group), std::par, OpenMP, alpaka
- benchmarking OpenMP ongoing
- waiting for NVIDIA to fix nvc++ to do multicore backend testing of std::par









ACTS

- We had high hopes that ACTS would have a full tracking chain enabled for GPU by now, but development has been slower than expected.
- Mainly due to "infrastructure" issues
  - Geometry description
  - Magnetic field maps
- ACTS has developed a robust data management system with both CUDA and SYCL backends
  - vecmem
- Some pieces (clusterization) have been ported to std::par

- Will be useful to monitor progress over the next year
  - question as to whether ATLAS will use GPUs for Run 4 and beyond is still in the air

Plans for FY23

End major code development by April 1 2023

- should still finish outstanding areas
- benchmark testbeds and platforms more consistently
- monitor evolving compilers and hardware (H100, Grace/Hopper, Aurora)

Allocate remaining time to write reports

- expand the metrics notes, make more formal guidelines
  - If you are doing *this*, then choose *that* or don't choose *that*
- common format to report/store results
  - including metadata such as hardware, compiler/library/driver versions, code versions, etc
- target conferences
  - CHEP
    - reporting current results
  - SC23
    - if the paper is accepted
    - can we do a BOF session?
    - not really the right audience for HEP
  - ACAT 24? (can't find a date)

Outreach

Report back to experiments

- present results in software meetings
  - ATLAS: June Software & Computing week
  - others?
- more focussed workshops
- get feedback from experiments - do we have enough time?
  - something to target beyond FY23
- what deadlines do experiments have for selecting the "next" language?

Report to community via other channels

- HSF
- IRIS-HEP
- when?

Both PPS and overview from all of CCE

Beyond FY23

Very important to continue monitoring portability layers

- hardware support, feature availability, performance
- especially *wrt* standards integration
  - std::par
  - senders / receivers
  - C++26

Integration with IOS

Other ideas?

Other Projects for FY24 - 29

HPC Friendly Data Models

- link

Distributed Scheduling

- link

Parallel Event Generators

- link

Simplify Statistical Analysis by using HPCs

- link

Data Reduction / Data Streaming

- link

LSST / DESC Workflows

- link