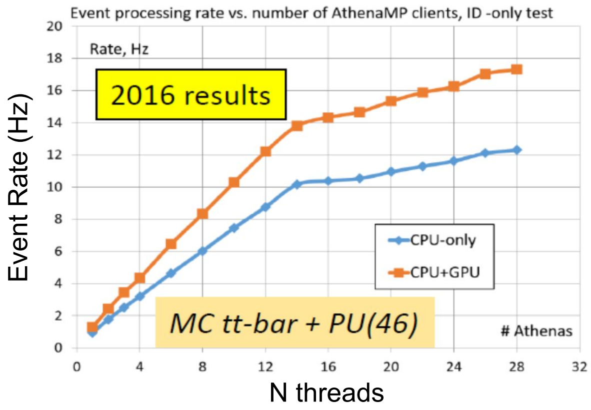
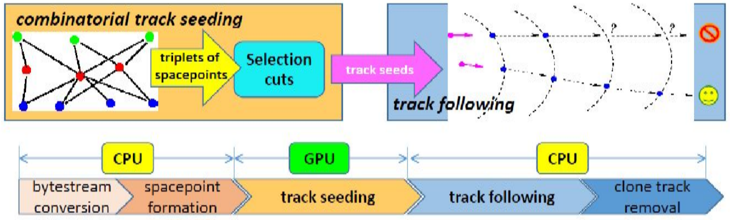
Exploring / Using Heterogeneous Architectures in ATLAS

Attila Krasznahorkay

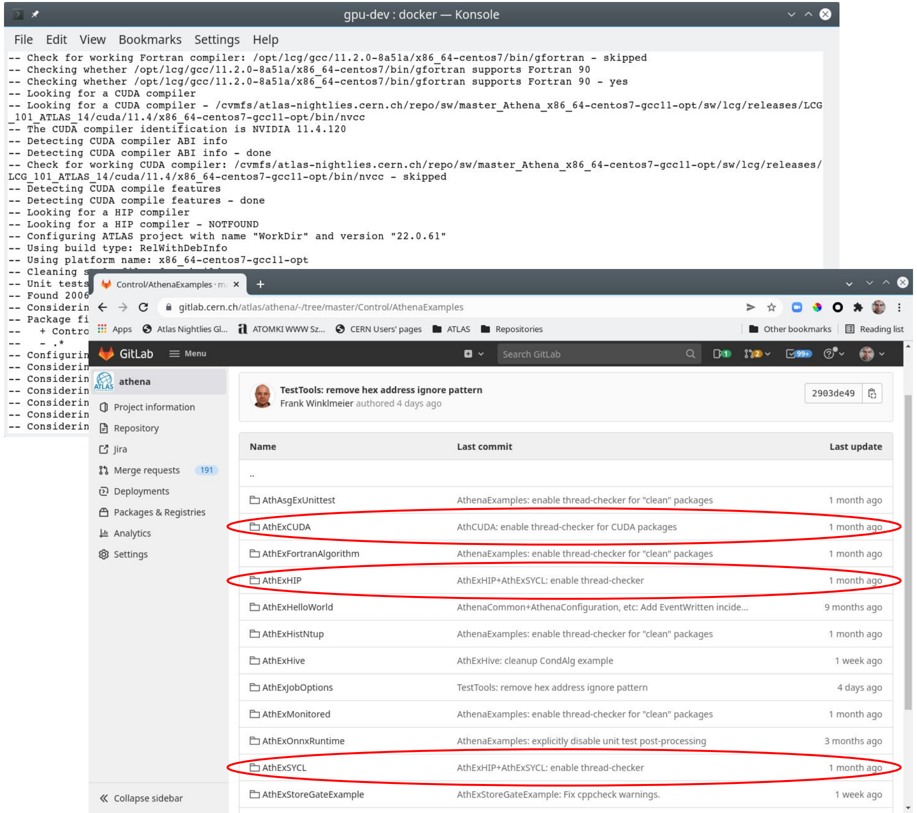
“Accelerator History” in ATLAS

- Exploring the usage of accelerators for the ATLAS HLT began a decade ago
    - 2012: ID Trigger prototype (ATL-DAQ-PROC-2012-006)
    - 2015: Trigger GPU Demonstrator (ATL-COM-DAQ-2019-059)
    - 2019: GPU ID pattern-matching prototype (ATL-COM-DAQ-2019-173)
    - 2020: GPU trigger algorithm integration in AthenaMT
- Was deemed not viable for Run-3 after all of the improvements with multi-threading





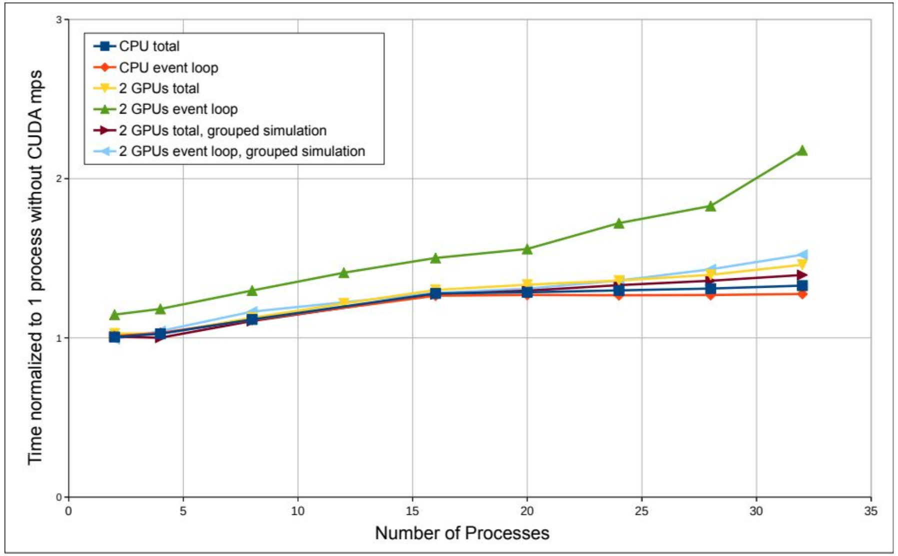
ATLAS's Offline Accelerator Support



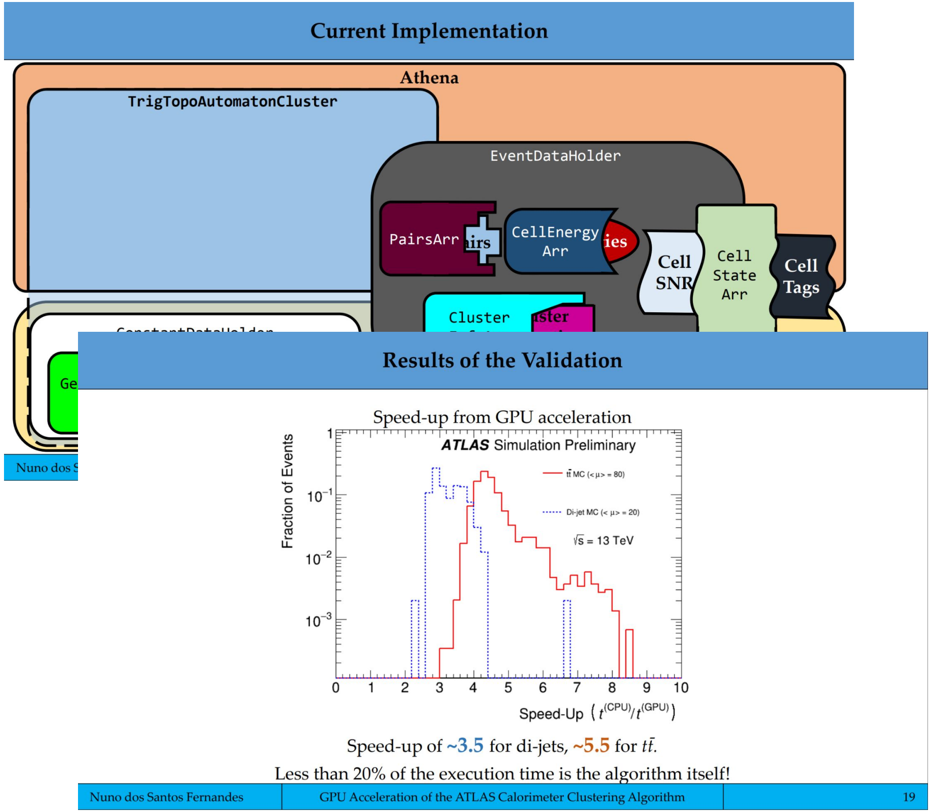
- In part due to the need of integrating previous GPU projects into the latest version of Athena, our Run-3 release does provide some GPU code support already
  - Though none of it is what I would consider production code, and we don't really do any asynchronous execution of GPU code in AthenaMT in practice
- The previous study on executing GPU kernels asynchronously in AthenaMT (https://doi.org/10.1051/epjconf/202024505006) will be revived in some form at one point

Calorimeter Simulation

- The GPU port of the ATLAS Fast Calorimeter Simulation is one of the most thoroughly tested for usage on different devices / architectures with different languages (ATL-COM-SOFT-2020-069)
  - Amongst others, the portability studies resulted in improvements to oneMKL (https://github.com/oneapi-src/oneMKL/pull/75)



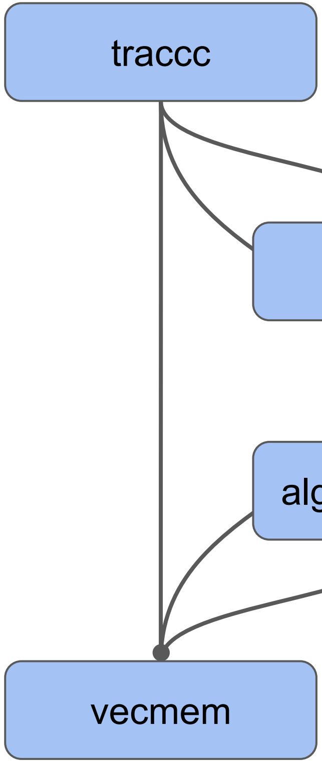
Calorimeter Reconstruction



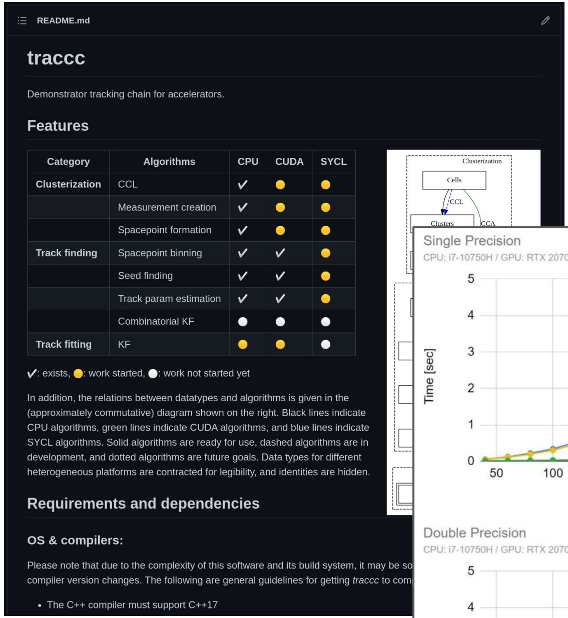
- One successful recent demonstrator was written for reconstructing clusters of energy deposits in the ATLAS calorimeters (https://indico.cern.ch/event/855454/contributions/4605031/)
  - Will need some additional work to integrate it with the latest version of Athena, and any (asynchronous) framework developments we may do later on

Track Reconstruction (1)

- A renewed R&D effort is happening at the moment to produce a demonstrator for reconstructing charged particle tracks in a “realistic setup”
  - The projects are on purpose staying away from “execution framework aspects” of the development (no MT testing for now)
- More details about the projects are available on:
  - https://indico.cern.ch/event/855454/contributions/4605054/
  - https://indico.cern.ch/event/855454/contributions/4605075/
  - https://indico.cern.ch/event/1073640/#3-parallelisation-in-acts



Track Reconstruction (2)



README.md

traccc

Demonstrator tracking chain for accelerators.

Features

Category	Algorithms	CPU	CUDA	SYCL
Clusterization	CCL	✓	🟡	🟡
	Measurement creation	✓	🟡	🟡
	Spacepoint formation	✓	🟡	🟡
Track finding	Spacepoint binning	✓	✓	🟡
	Seed finding	✓	✓	🟡
	Track param estimation	✓	✓	🟡
	Combinatorial KF	⚪	⚪	⚪
Track fitting	KF	🟡	🟡	⚪

✓: exists, 🟡: work started, ⚪: work not started yet

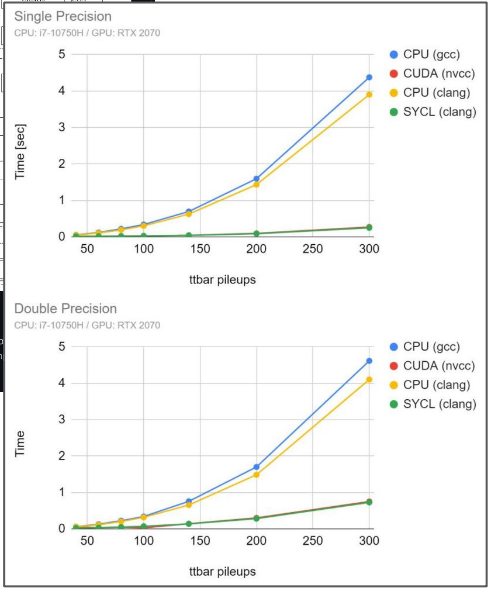
In addition, the relations between datatypes and algorithms is given in the (approximately commutative) diagram shown on the right. Black lines indicate CPU algorithms, green lines indicate CUDA algorithms, and blue lines indicate SYCL algorithms. Solid algorithms are ready for use, dashed algorithms are in development, and dotted algorithms are future goals. Data types for different heterogeneous platforms are contracted for legibility, and identities are hidden.

Requirements and dependencies

OS & compilers:

Please note that due to the complexity of this software and its build system, it may be so compiler version changes. The following are general guidelines for getting *traccc* to com

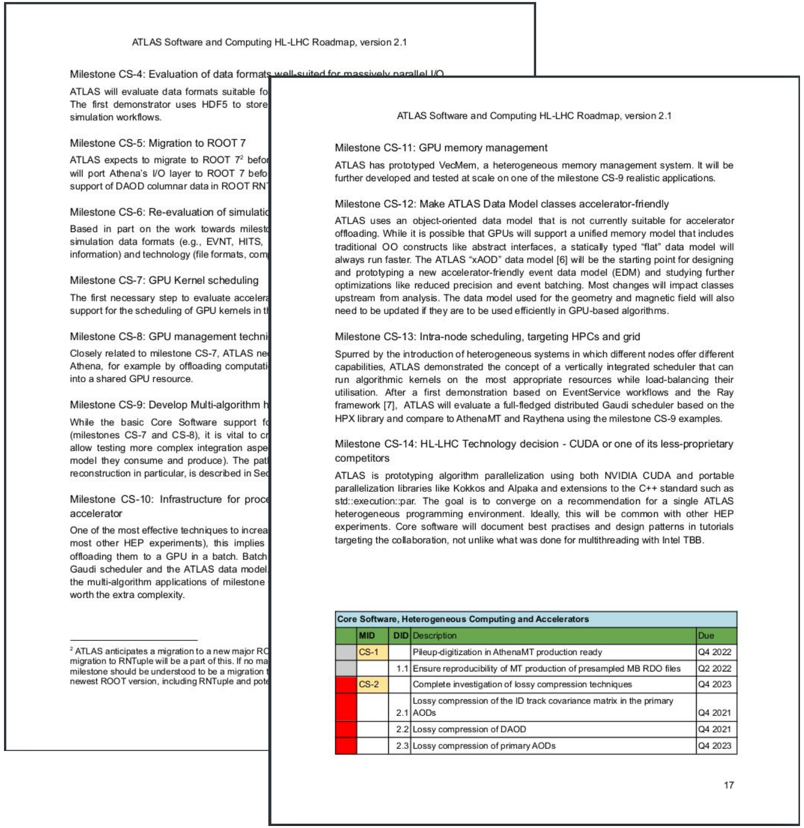
- The C++ compiler must support C++17



- Significant work for this development was provided by one summer student and one technical student funded by Intel through OpenLab
- Very much a work in progress, which we hope will succeed in demonstrating that we can reasonably accelerate track reconstruction on a single GPU in a multi-threaded application
    - Still a lot of work to do for that!

Summary

- GPUs and FPGAs are actively being evaluated for being used during HL-LHC in the ATLAS HLT and offline reconstruction
  - In order to fit the experiment’s computing budget, we will very likely need to use accelerators
- Very active R&D is going on for track reconstruction on GPUs as part of the Acts project
  - Whose results will largely influence how we will handle GPUs in the experiment’s offline software in the long term


ATLAS Software and Computing HL-LHC Roadmap, version 2.1

Milestone CS-4: Evaluation of data formats well-suited for massively parallel I/O

ATLAS will evaluate data formats suitable fo
The first demonstrator uses HDF5 to store
simulation workflows.

Milestone CS-5: Migration to ROOT 7

ATLAS expects to migrate to ROOT 7[2] befo
will port Athena’s I/O layer to ROOT 7 befo
support of DAOD columnar data in ROOT RN

Milestone CS-6: Re-evaluation of simulatio

Based in part on the work towards milesto
simulation data formats (e.g., EVNT, HITS,
information) and technology (file formats, com

Milestone CS-7: GPU Kernel scheduling

The first necessary step to evaluate accelera
support for the scheduling of GPU kernels in t

Milestone CS-8: GPU management techni

Closely related to milestone CS-7, ATLAS ne
Athena, for example by offloading computati
into a shared GPU resource.

Milestone CS-9: Develop Multi-algorithm h

While the basic Core Software support f
(milestones CS-7 and CS-8), it is vital to cr
allow testing more complex integration aspe
model they consume and produce). The pat
reconstruction in particular, is described in Se

Milestone CS-10: Infrastructure for proc
accelerator

One of the most effective techniques to increa
most other HEP experiments), this implies
offloading them to a GPU in a batch. Batch
Gaudi scheduler and the ATLAS data model
the multi-algorithm applications of milestone
worth the extra complexity.

[2] ATLAS anticipates a migration to a new major RO
migration to RNTuple will be a part of this. If no ma
milestone should be understood to be a migration t
newest ROOT version, including RNTuple and pote

ATLAS Software and Computing HL-LHC Roadmap, version 2.1

Milestone CS-11: GPU memory management

ATLAS has prototyped VecMem, a heterogeneous memory management system. It will be further developed and tested at scale on one of the milestone CS-9 realistic applications.

Milestone CS-12: Make ATLAS Data Model classes accelerator-friendly

ATLAS uses an object-oriented data model that is not currently suitable for accelerator offloading. While it is possible that GPUs will support a unified memory model that includes traditional OO constructs like abstract interfaces, a statically typed “flat” data model will always run faster. The ATLAS “xAOD” data model [6] will be the starting point for designing and prototyping a new accelerator-friendly event data model (EDM) and studying further optimizations like reduced precision and event batching. Most changes will impact classes upstream from analysis. The data model used for the geometry and magnetic field will also need to be updated if they are to be used efficiently in GPU-based algorithms.

Milestone CS-13: Intra-node scheduling, targeting HPCs and grid

Spurred by the introduction of heterogeneous systems in which different nodes offer different capabilities, ATLAS demonstrated the concept of a vertically integrated scheduler that can run algorithmic kernels on the most appropriate resources while load-balancing their utilisation. After a first demonstration based on EventService workflows and the Ray framework [7], ATLAS will evaluate a full-fledged distributed Gaudi scheduler based on the HPX library and compare to AthenaMT and Raythena using the milestone CS-9 examples.

Milestone CS-14: HL-LHC Technology decision - CUDA or one of its less-proprietary competitors

ATLAS is prototyping algorithm parallelization using both NVIDIA CUDA and portable parallelization libraries like Kokkos and Alpaka and extensions to the C++ standard such as std::execution::par. The goal is to converge on a recommendation for a single ATLAS heterogeneous programming environment. Ideally, this will be common with other HEP experiments. Core software will document best practises and design patterns in tutorials targeting the collaboration, not unlike what was done for multithreading with Intel TBB.

Core Software, Heterogeneous Computing and Accelerators				
	MID	DID	Description	Due
	CS-1		Pileup-digitization in AthenaMT production ready	Q4 2022
		1.1	Ensure reproducibility of MT production of presampled MB RDO files	Q2 2022
	CS-2		Complete investigation of lossy compression techniques	Q4 2023
		2.1	Lossy compression of the ID track covariance matrix in the primary AODs	Q4 2021
		2.2	Lossy compression of DAOD	Q4 2021
		2.3	Lossy compression of primary AODs	Q4 2023

17

http://home.cern