



# Integrating oneAPI/SYCL in the ATLAS Software

Attila Krasznahorkay on behalf of many people





- Computing challenge(s) for ATLAS
- Accelerated code development in ATLAS
- Using SYCL in the Acts Parallelization R&D projects
- Latest performance results

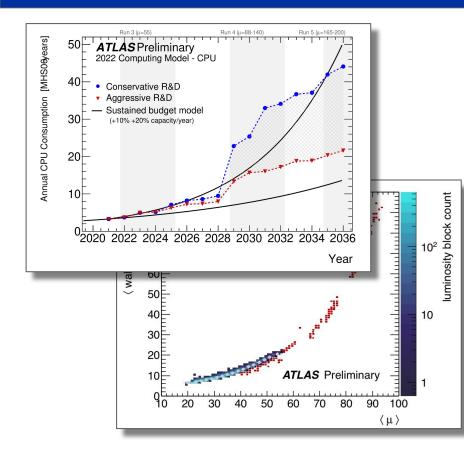




- Computing challenge(s) for ATLAS
- Accelerated code development in ATLAS
- Using SYCL in the Acts Parallelization R&D projects
- Latest performance results

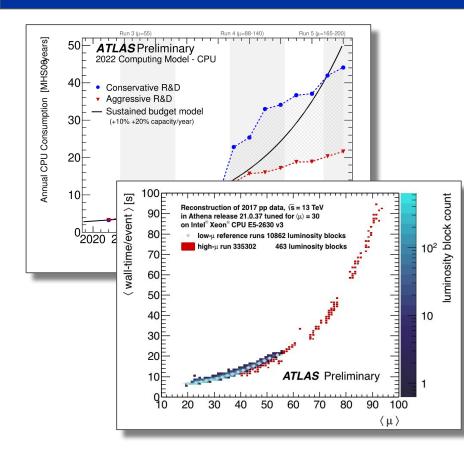
Way too much material to cover in ~12 minutes. Will mostly focus on performance.

# The HL-LHC Computing Challenge



- The problem is well known / widely advertised at this point
  - The "more complex" proton-proton collision events that we will record at the High-Luminosity LHC will require much more CPU power than we can afford
- This is in a large part due to the behaviour of charged particle tracking in ATLAS's reconstruction
  - Though the CPU code did become a lot better since we made these original plots...

# The HL-LHC Computing Challenge



- The problem is well known / widely advertised at this point
  - The "more complex" proton-proton collision events that we will record at the High-Luminosity LHC will require much more CPU power than we can afford
- This is in a large part due to the behaviour of charged particle tracking in ATLAS's reconstruction
  - Though the CPU code did become a lot better since we made these original plots...

# Past and Ongoing Work

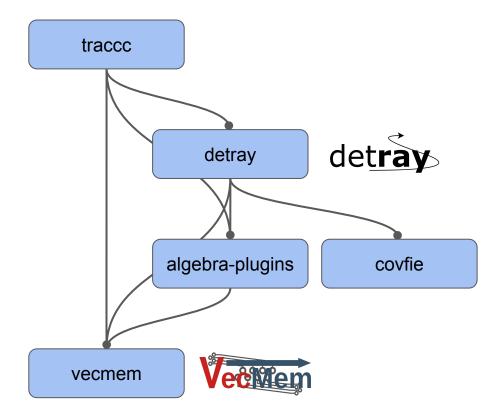


- Accelerator code development is happening in multiple areas in ATLAS, and since many years
  - I pointed to those last year already (<u>https://indico.cern.ch/event/1100904/timetable/?view=standard#13-exploring-heterogeneous-arc</u>)
- What I focus on today is just one of these areas
  - However one of the more significant ones...

# The ACTS Parallelization R&D

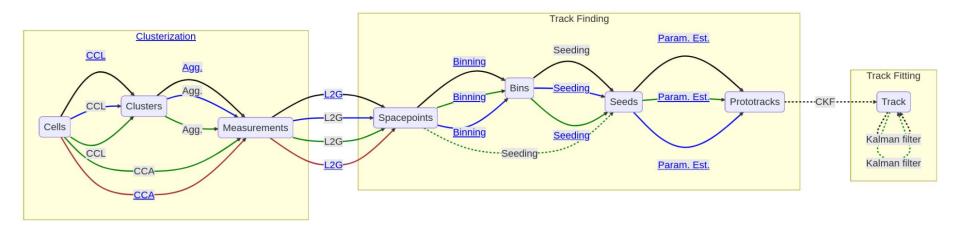


- A number of projects were brought to life in the R&D effort
  - It seemed to be a good idea to create functionally distinct units independently.
     Even if eventually they'll likely end up in a unified repository.
- Algorithmic development is happening in <u>traccc</u> and <u>detray</u>
  - The rest are providing "non-algorithmic" helper code for the project



# The traccc Algorithms





- Multiple algorithms were developed for for every step
  - Black: CPU/C++, Blue: SYCL, Green: CUDA, Brown: Futhark
- Sharing as much code between the GPU implementations (and even between the CPU and GPU implementations) as possible in the form of "common functions"

#### State of the Track Reconstruction Chain



Category	Algorithms	CPU	CUDA	SYCL	Futhark
Clusterization	CCL	$\checkmark$	V	$\checkmark$	$\checkmark$
	Measurement creation	$\checkmark$	$\checkmark$	V	V
	Spacepoint formation	$\checkmark$	V	$\checkmark$	
Track finding	Spacepoint binning	V	$\checkmark$	$\checkmark$	
	Seed finding	V	$\checkmark$	$\checkmark$	
	Track param estimation	$\checkmark$	$\checkmark$	V	
	Combinatorial KF	-			Also ex Kokk sign
Track fitting	KF		•	•	in et

### **Throughput Measurement Code**

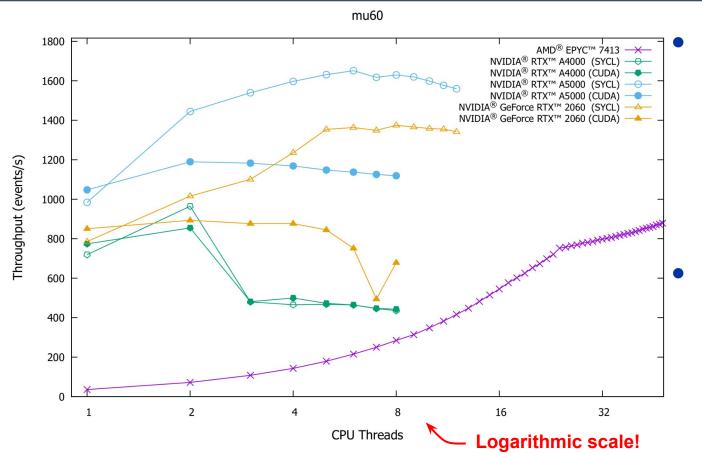
CERN

- Introduced multi-threaded executables that process events pre-loaded into host memory using TBB
  - One task per event
- For GPU devices multi-threading helps because:
  - Our algorithm chain still executes some operations on the CPU, even when a GPU is used
  - Our kernels often don't fully utilise GPUs, so multiple kernels can run in parallel

94	full_chain_algorithm::output_ty	<pre>rpe full_chain_algorithm::operator()(</pre>
	<pre>const alt_cell_collection_t</pre>	ypes::host& cells,
	<pre>const cell_module_collectic</pre>	n_types::host& modules)
97		
	<pre>// Create device copy of ir</pre>	put collections
	alt_cell_collection_types::	<pre>buffer cells_buffer(cells.size(),</pre>
		<pre>*m_cached_device_mr);</pre>
101	(*m_copy)(vecmem::get_data(	cells), cells_buffer);
102	cell_module_collection_type	es::buffer modules_buffer(modules.size(),
103		<pre>*m_cached_device_mr);</pre>
104	(*m_copy)(vecmem::get_data(	<pre>modules), modules_buffer);</pre>
	<pre>// Execute the algorithms.</pre>	
107	<pre>const clusterization_algori</pre>	.thm::output_type
	m_clusterization(cells_	buffer, modules_buffer);
	<pre>const track_params_estimati</pre>	.on::output_type track_params =
110	m_track_parameter_estim	nation(spacepoints.first,
111		<pre>m_seeding(spacepoints.first));</pre>
112		
113	// Get the final data back	to the host.
114	bound_track_parameters_coll	.ection_types::host result;
115	(*m_copy)(track_params, res	ult);
116		
117	// Return the host containe	er.
118	return result;	
119	}	

#### CUDA and SYCL





Both our CUDA and SYCL code is sub-optimal still (not taking full advantage of parallel / asynchronous execution) Current SYCL code performs better with MT, so only showing

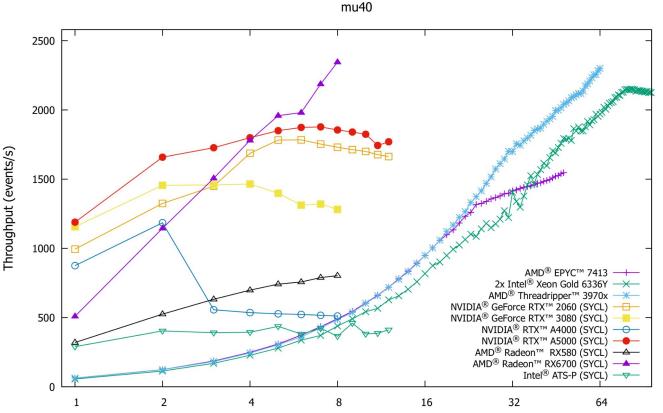
those results in the

rest of the plots

11

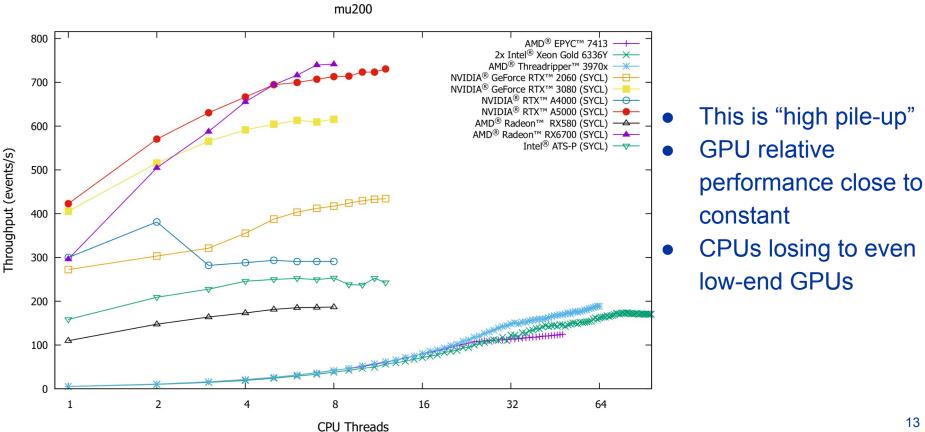
### **Throughput Measurements**



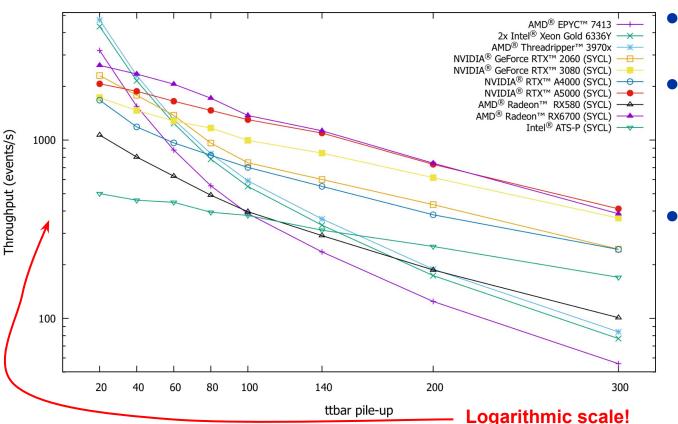


- This is "low pile-up"
- Only showing SYCL results
- High-end CPUs still beat GPUs
- The RX6700 card is doing something weird (take with huge grain of salt!)

#### **Throughput Measurements**



### **Throughput Measurements**



Peak performance across different hardware

- Putting it all
  together...
  GPUs (all) beat
  CPUs in throughput
  at HL-LHC event
- complexities Will use such results in our final decisions about ATLAS development strategies...

### **Integration Into Athena**

- All of this only lives in our R&D projects at the moment
- The plan is to start moving the code into the main Acts repository this year
  - And then to pick up the code in Athena from Acts
- Still some strategic questions will need to be decided for this about the way Acts and Athena would handle "event data"
  - No blocker issues on the horizon however

acts-project / acts	Public	Ç Notific	cations 🖞 Fork 116 🛱 Star 85 🗸							
> Code 🕢 Issues 58	17 Pull requests 27 Q Discussions 🕥 Actions	🗄 Projects 🕕 Security 🗠 Insigh	its							
រិ <sup>ស</sup> main 👻 រិ <sup>ស</sup> 96 brand	ches 🔯 127 tags	Go to file Code -	About							
CarloVarni fix: itk_see	ding script runs again (#1947)	49e2c9e 2 hours ago 🕥 6,824 commits	Experiment-independent toolkit for (charged) particle track reconstruction in (high energy) physics experiments							
💼 .github	ci: Add script checking for #pragma once to C	I (#1866) last month	implemented in modern C++							
Alignment	refactor: Remove LoggerWrapper in favor of Log	gger refs (#1812) last month	<i>∂</i> acts.readthedocs.io							
🖿 CI	atlas > 🔕 athena									
Core										
Examples	ATLAS Project ID: 53790 👸 Leave project	① → Hunstar 167 ¥ Fork 1970								
Fatras		831.8 MB Project Storage 💿 359 Releases								
- riagina -	The ATLAS Experiment's main offline software repo									
Tests ල cmake	DOI 10.5281/zenodo.2641997 Doxygen master									
docs										
hirdparty		Merge branch 'EMBEMECFrozenShowers_main' into 'master' 🚥								
clang-format	Edward Moyse authored 1 hour ago									
C .clang-tidy	master > athena / + >		Find file Web IDE							
gitattributes	■ README 한 LICENSE ● Add CHANGELOG	Add CONTRIBUTING     Enable Auto D	evOps 🗄 Set up CI/CD 🔞 Configure Integrations							
gitignore			erops [ [ ] secup circo ] [ [ ] coningue integrations ]							
🗅 .gitlab-ci.yml	Name	Last commit	Last update							
.gitmodules	devcontainer         flake8: cleanup error code selection									
	🖹 .vscode	flake8: cleanup error code selection	1 year ago							
	🗅 AsgExternal/Asg_Test	Updating PHYSLITE input files for CI tests	s 2 months ago							
	ComponentAccumulator-based configuration of Tile T									
	Activate Linear Extrapolation only for RUN3 (NTLASR      Build prepare_release_notes compatibility with 22.0-mc20									
	Calorimeter TestDefaults: rename RAW test file to RAW_RUN2									
	Commission CommissionUtils: delete unused package and job opti									
»	🗅 Control	Merge branch 'master-dq-art-enable-23.0	' into 'master' 1 day ago							





- Don't take any of these results as the end-all-be-all!
  - We're in the process of making our performance testing code even faster with both CUDA and SYCL
- We seem to be well on track to produce maintainable code for track reconstruction that would minimise the duplication between CPU and GPU codebases
- Performance numbers are very encouraging at the moment
  - But do take them, especially the AMD ones, with some amount of salt...



http://home.cern