



Using oneAPI in ATLAS

Attila Krasznahorkay
on behalf of ATLAS HCAF



oneAPI and ATLAS

- Heterogeneous software development is happening on a lot of fronts in ATLAS
 - While I **should** be aware of all of them, I am pretty sure that I am not 😞
- Given the short format, will focus mostly on some technical aspects in the talk
 - One big omission will be the lack of machine learning in this talk. However on that front we are anyway trying to use “systems” that allow us to train and evaluate networks on a wide array of backends, I believe we do not use oneAPI’s ML features specifically at the moment...
- I will be listing (some of) the areas where we are making use of oneAPI in software R&D

First, some technicalities...

Using oneAPI in Software Development



- There are a few technical difficulties off the bat with using any heterogeneous computing language in a software project
 - The OS / platform on which to use oneAPI
 - The integration of oneAPI into the project's build system

OS / Platform

- Using oneAPI is easiest on Ubuntu 18.04 / 20.04
 - Very understandable, but still proves a bit challenging given that ATLAS's production platform is CentOS 7
- To simplify using DPC++ on top of the ATLAS x86_64-centos7-gcc8-opt software releases, we provide a Docker image
 - <https://gitlab.cern.ch/akraszna/atlas-gpu-devel-env>
- “Standalone” projects like Acts just build on Ubuntu natively instead

ATLAS GPU Development Environment
Project ID: 87672

centos7 - atlas-gpu-devel-env

Added a "Full CUDA" image to the repository. ...
Attila Krasznahorkay authored 3 weeks ago

Name	Last commit	Last update
derived-images	Added a "Full CUDA" image to the repository.	3 weeks ago
docker	Switched to CUDA 10.2 and intel/mv 2020-11.	1 month ago
syclinfo	Added the syclinfo executable to the image.	6 months ago
.dockignore	Created a working configuration for CUDA code d...	1 year ago
.gitlab-ci.yml	Added a "Full CUDA" image to the repository.	3 weeks ago
Dockerfile	Revert "Added cuDNN to the image."	3 weeks ago

```
@cd06721c3625$ /home/kraszna
This is the ATLAS GPU code development environment.

The container starts with the Intel oneAPI, the NVidia CUDA and the AMD ROCm/HIP
environment set up. Along with an appropriate version of GCC, Cmake and NINJA.

In case you mounted CVMFS into the container, you can also go ahead and set
up one of the ATLAS nightly releases, simply like:

setupATLAS
setup Athena,master,latest

This will also leave the previously mentioned compilers set up for the build.

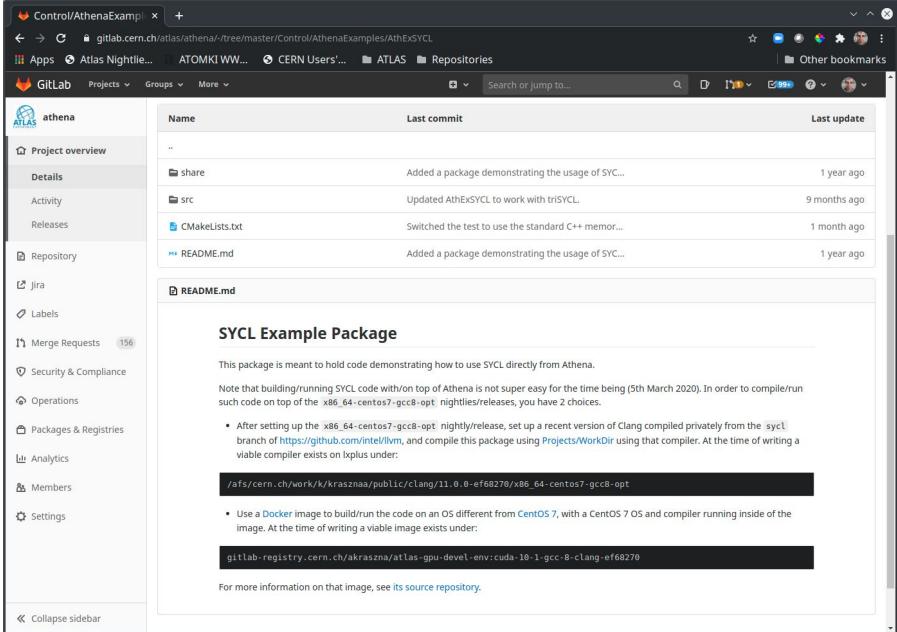
[kosh1@atlas1 kraszna] ~ syclinfo
***** Platform *****
Name : Intel(R) OpenCL HD Graphics
Vendor : Intel(R) Corporation
Version : OpenCL 2.0
Device :
Name : Intel(R) Graphics [0x3eab]
Vendor : Intel(R) Corporation
Version : 1.0.0
*****
```

Build System Integration

- Theoretically you just need to build all of your code with DPC++, and you should be good to go
 - In practice however things are a bit more complicated. Some code can just not be compiled by DPC++, and non-heterogeneous code compiles a bit slowly with DPC++...
- In ATLAS we use [CMake](#). For that I developed 2 different methods of using oneAPI so far.
 - [FindSYCL.cmake](#)
 - Provides a helper function to specify which library/executable needs to be compiled as a oneAPI/[SYCL](#) library/executable
 - Is currently in use in the ATLAS offline software nightlies, in combination with [triSYCL](#)
 - Code teaching CMake how to handle SYCL “as its own language”
 - <https://github.com/acts-project/vecmem/tree/main/cmake/sycl>
 - Very similar to [CMake’s CUDA integration](#). After calling `enable_language(SYCL)`, all files with the `.sycl` extension (in the “current scope”) are compiled using the oneAPI compiler “in the right way”.

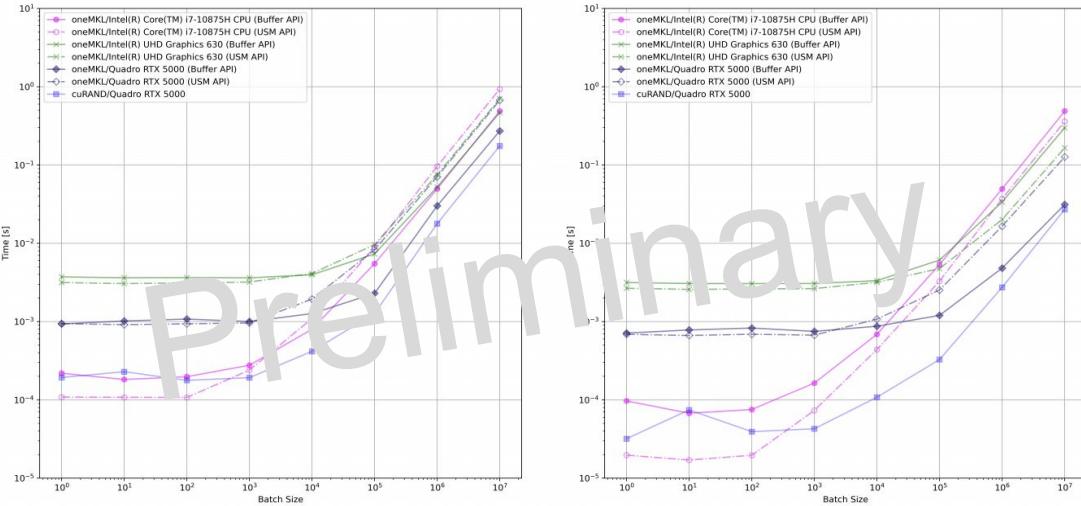
ATLAS oneAPI R&D projects

Offline Software Demonstrator



- A trivial demonstrator has existed in our software for using oneAPI/SYCL in an Athena algorithm, for some time
 - <https://gitlab.cern.ch/atlas/athena/-/tree/master/Control/AthenaExamples/AthExSYCL>
 - Using the “`FindSYCL.cmake` mechanism” outlined earlier
- However no meaningful oneAPI algorithm development is happening on top of our offline software at the moment

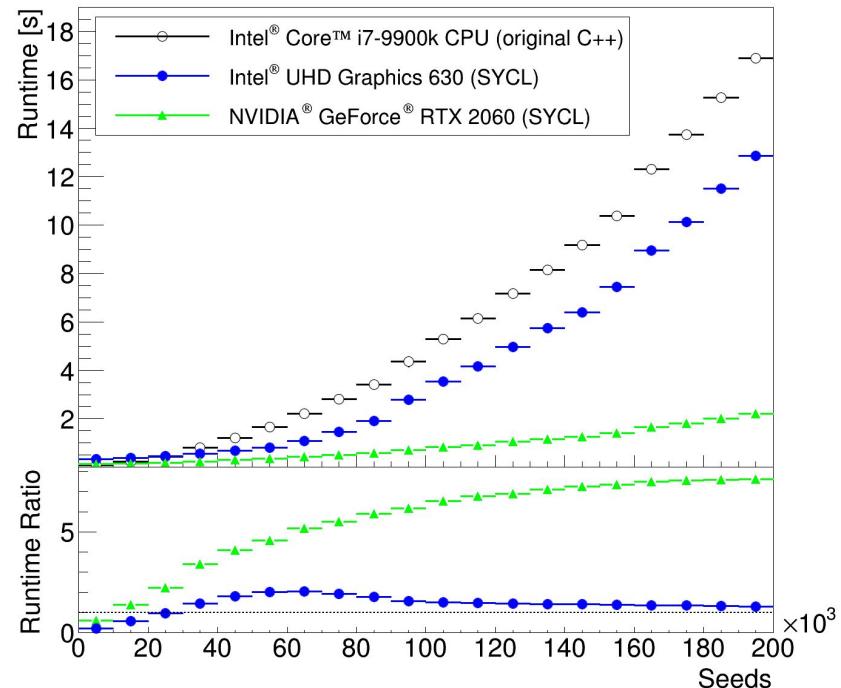
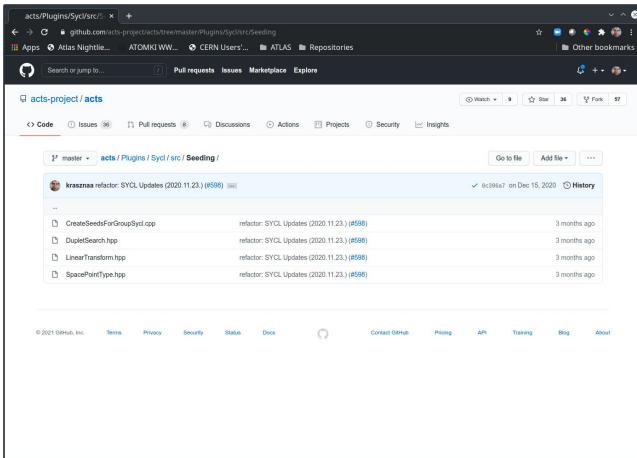
Fast Calorimeter Simulation



- Is used as a testbed for heterogeneous computing developments
- Most recently used for thorough tests with random number generation using oneMKL
 - Testing its interoperability and performance with/wrt. cuRAND

Acts Seed Finding

- Developed as part of last summer's Summer Student programme
- Not seeing active development right now, but will possibly be used as part of this year's "Acts full chain demonstrator"



The VecMem R&D Project



A screenshot of a GitHub repository page for 'acts-project/vecmem'. The page shows the repository's structure, commit history, and various metadata sections like About, Releases, Packages, and Contributors.

About: Vectorised data model base and helper classes.

Releases: No releases published. Create a new release.

Packages: No packages published. Publish your first package.

Contributors: 2

Languages: C++ 77.9%, CMake 20.8%, Cuda 1.1%, Dockerfile 0.2%

Commit History:

- krasznaa Merge pull request #36 from stephenswat/enhancement/allocator 3 hours ago
- 5feefaf2 3 hours ago 136 commits
- .devcontainer Added a DevContainer configuration for building all possible parts of... 25 days ago
- .github/workflows Make Github Actions workflows compatible with act 8 days ago
- .vscode Merge branch 'main' into syscl-make-config-main-20210212 25 days ago
- cmake Merge branch 'main' into HIPMemoryResource-main-20210304 4 days ago
- core Add support for a resource-aware allocator 19 hours ago
- cuda Merge branch 'main' into enhancement/rail_select_device 6 days ago
- hip Removed vecmem::hip::direct_memory_manager. 5 days ago
- syscl Removed vecmem::syscl::direct_memory_manager. 5 days ago
- tests Add tests for the new allocator class 19 hours ago
- .gitattributes Taught GitHub and VSCode about the .syscl file-extension. 25 days ago
- .gitignore Added a DevContainer configuration for building all possible parts of... 25 days ago
- CMakeLists.txt Stopped including the VecMem CMake modules with their absolute paths. 25 days ago
- LICENSE Adding the first commit, with a README and a LICENSE file. last month
- README.md Adding the first commit, with a README and a LICENSE file. last month

- As mentioned on the previous slide, we are working on putting a multi-algorithm track reconstruction workflow together with Acts
 - To manage the event and conditions data for that demonstrator in a way that would make it easy to offload calculations to accelerators, we are actively at work on a new R&D project
- Too early to say much about it yet, but the `.syscl` file extension “build style” is something that I experiment with here... 😊

Summary

- ATLAS is undertaking a number of R&D projects to evaluate the feasibility of heterogeneous computing in its data processing
 - Efforts are going on for the trigger system as well, which I did not have time to go into...
- Our goal is to implement our code in a way that would resemble the future C++ ISO standard the most
 - With our (my...) bet being that C++3x will look a lot like [oneAPI/SYCL](#)





<http://home.cern>