Intel oneAPI DPC++ and SYCL

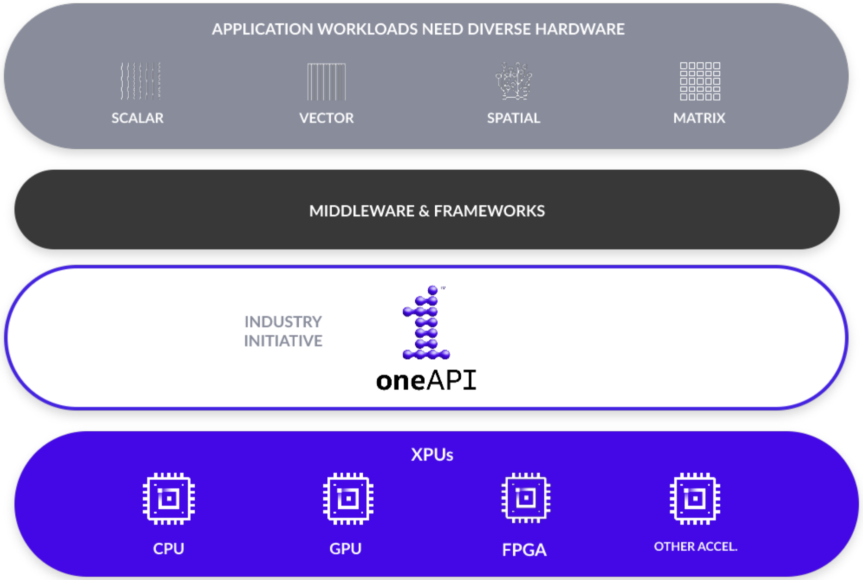
Tao Lin

<lintao@ihep.ac.cn>

2022.04.24

Motivation

- Heterogeneous computing
  - Hardware: CPU, GPU, FPGA…
  - Software: OpenCL, CUDA, ROCm …
- Intel oneAPI DPC++/SYCL
  - “oneAPI is an open, cross-industry, standards-based unified programming model that delivers a common developer experience across accelerator architectures”



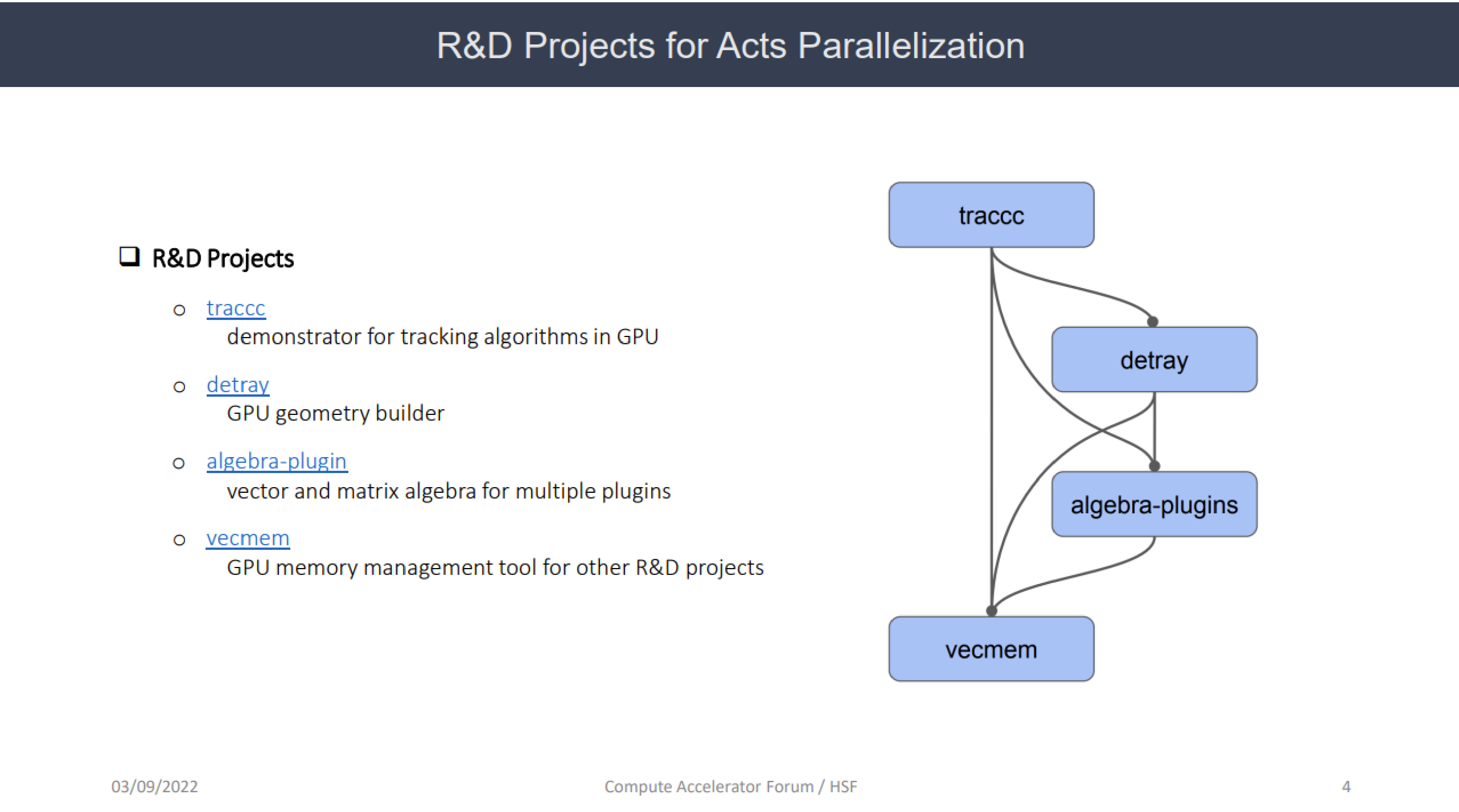
Applications of SYCL to HEP software

- ACTS and traccc (https://github.com/acts-project/traccc)
  - [1] Angéla Czirkos, https://indico.cern.ch/event/955133/contributions/4021301/
  - [2] Beomki Yeo, https://indico.cern.ch/event/1073640/#3-parallelisation-in-acts

Question: How to make PODIO/EDM4hep work with SYCL?

ACTS Parallelization (1)

From Beomki Yeo



R&D Projects for Acts Parallelization

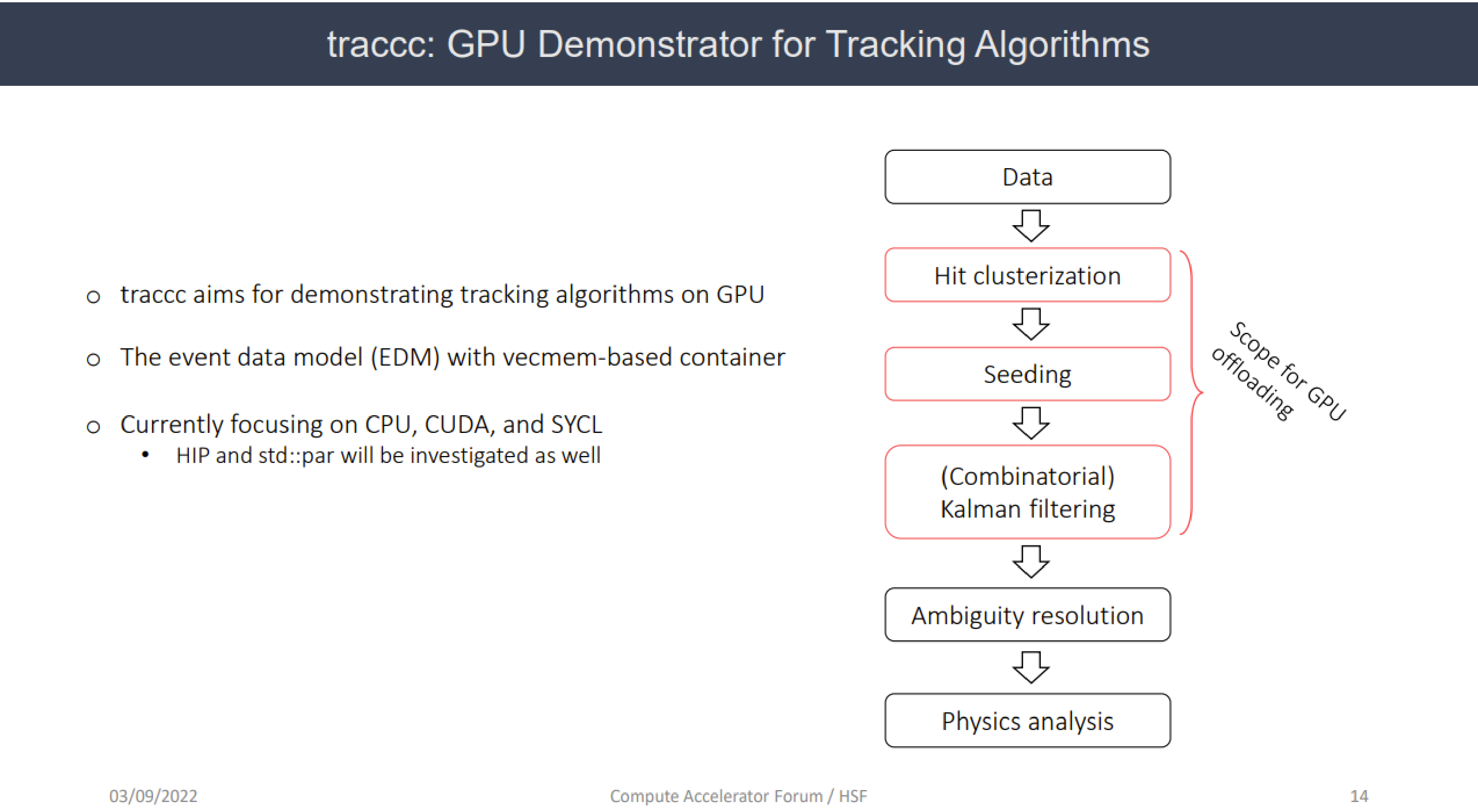
❑ R&D Projects

- traccc
  demonstrator for tracking algorithms in GPU
- detray
  GPU geometry builder
- algebra-plugin
  vector and matrix algebra for multiple plugins
- vecmem
  GPU memory management tool for other R&D projects

03/09/2022 Compute Accelerator Forum / HSF 4

ACTS Parallelization (2)

From Beomki Yeo



traccc: GPU Demonstrator for Tracking Algorithms

- traccc aims for demonstrating tracking algorithms on GPU
- The event data model (EDM) with vecmem-based container
- Currently focusing on CPU, CUDA, and SYCL
  - HIP and std::par will be investigated as well

Data

Hit clusterization

Seeding

(Combinatorial)
Kalman filtering

Ambiguity resolution

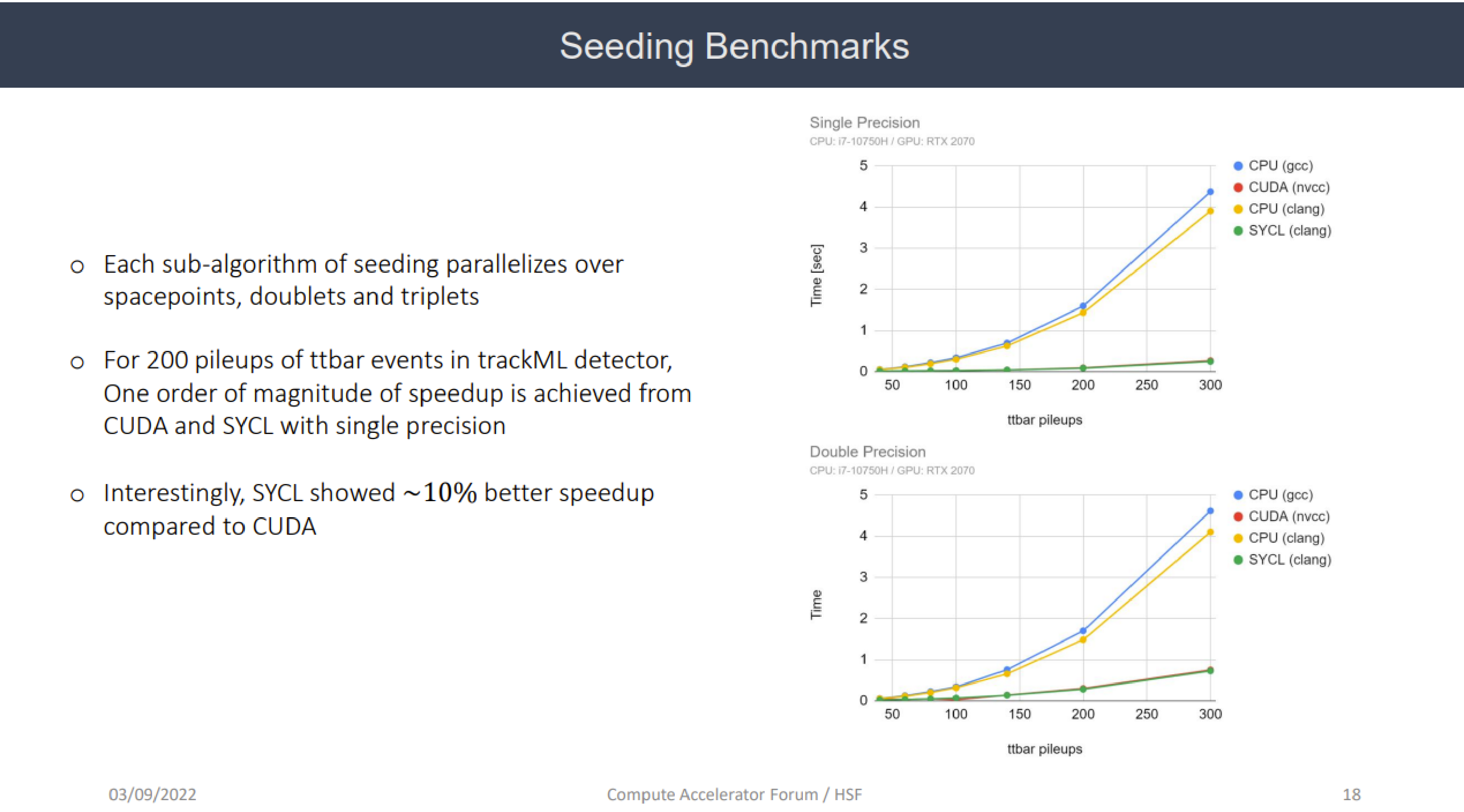
Physics analysis

Scope for GPU offloading

03/09/2022 Compute Accelerator Forum / HSF 14

ACTS Parallelization (3)

From Beomki Yeo



Seeding Benchmarks

- Each sub-algorithm of seeding parallelizes over spacepoints, doublets and triplets
- For 200 pileups of ttbar events in trackML detector, One order of magnitude of speedup is achieved from CUDA and SYCL with single precision
- Interestingly, SYCL showed ~10% better speedup compared to CUDA

Single Precision

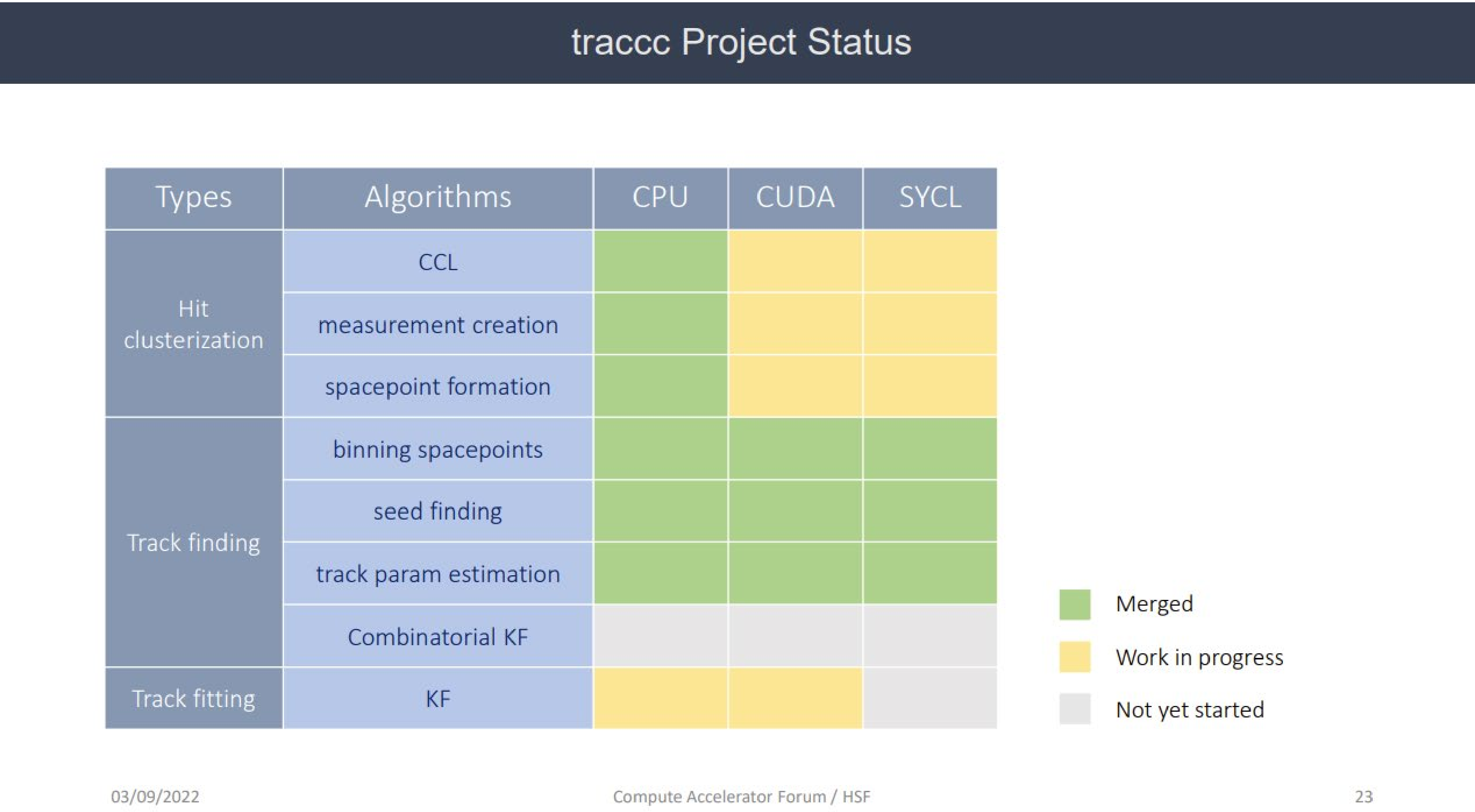
CPU: i7-10750H / GPU: RTX 2070

Double Precision

CPU: i7-10750H / GPU: RTX 2070

ACTS Parallelization (4)

From Beomki Yeo



traccc Project Status

Types	Algorithms	CPU	CUDA	SYCL
Hit clusterization	CCL	🟩	🟨	🟨
	measurement creation	🟩	🟨	🟨
	spacepoint formation	🟩	🟨	🟨
Track finding	binning spacepoints	🟩	🟩	🟩
	seed finding	🟩	🟩	🟩
	track param estimation	🟩	🟩	🟩
	Combinatorial KF	⬜	⬜	⬜
Track fitting	KF	🟨	🟨	⬜

🟩	Merged
🟨	Work in progress
⬜	Not yet started

03/09/2022 Compute Accelerator Forum / HSF 23

Building traccc + SYCL

- As part of traccc is already implemented with SYCL, it will be a good starting point for us.

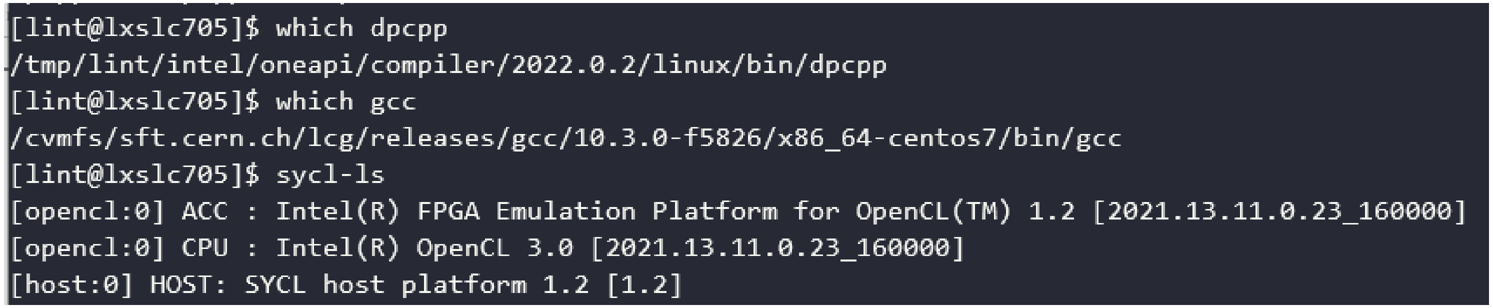
Config	Hardware	OS	Compiler	SYCL backend	Bulid traccc	Run traccc
1	Intel CPU (IHEP login node)	CentOS 7.8	LCG 101 (GCC 10.3 + clang 12) + oneAPI DPC++	CPU	OK	OK
2	Intel CPU (IHEP login node)	CentOS 7.8	LCG 101 (GCC 11.1) + intel/llvm (2021-12)	CPU	OK	?
3	Intel CPU (IHEP login node)	CentOS 7.8	LCG 101 (GCC 11.1) + intel/llvm (2021-12)	CUDA 11.0		
4	Intel CPU + NVIDIA RTX 8000 (workstation)	CentOS 7.9	LCG 101 (GCC 11.1) + intel/llvm (2021-12)	CUDA 11.2	OK	OK

Note:

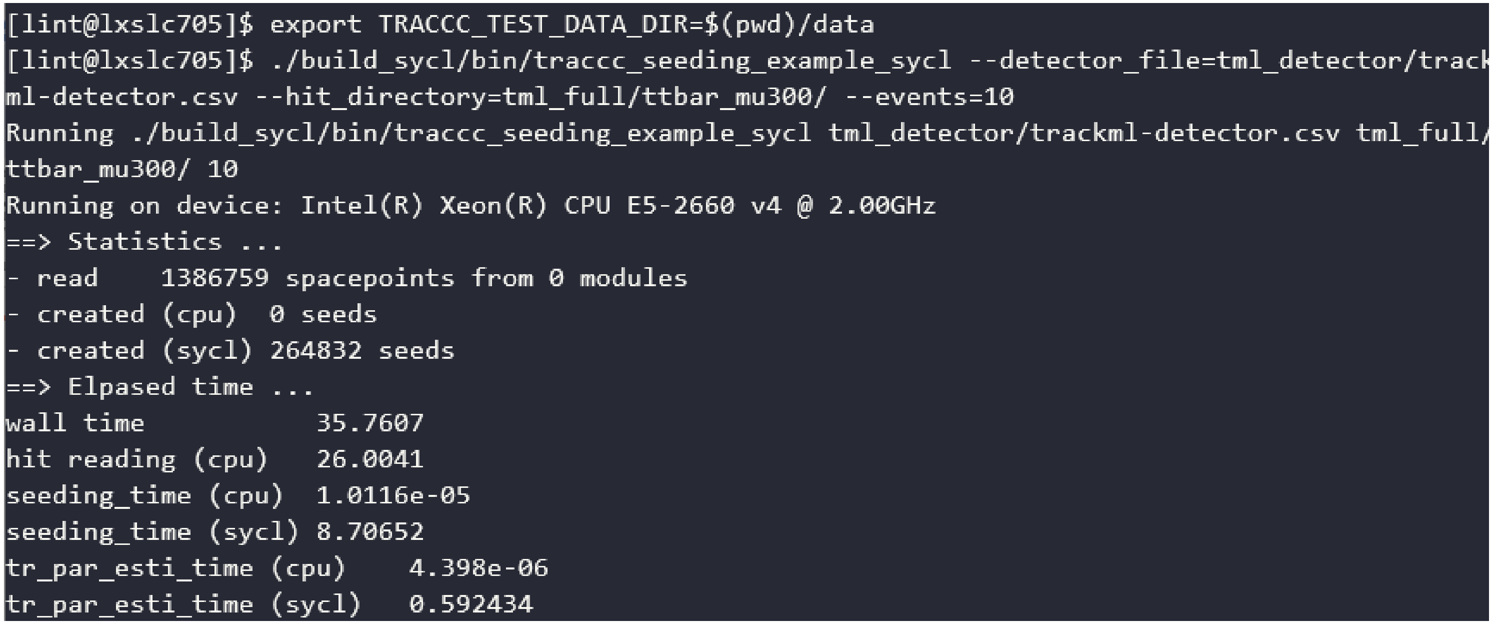
- Need to set envvar CXX to the compiler with SYCL support before building traccc.
  - `cmake -B build_sycl -S . -DTRACCC_BUILD_SYCL=ON`
- oneAPI DPC++ only provides CPU backend. No CUDA backend.
- In order to use the CUDA backend, need to build intel/llvm with customizations.
- When building intel/llvm, need to specify the GCC toolchain and CUDA.
  - `python $DPCPP_HOME/llvm-2021-12/buildbot/configure.py --cuda --cmake-opt="-DGCC_INSTALL_PREFIX=/cvmfs/sft.cern.ch/lcg/releases/gcc/11.1.0-e80bf/x86_64-centos7"`
- Building intel/llvm + CUDA at IHEP login node is fine, but there is runtime error in GPU nodes.

Intel DPC++ at IHEP login node

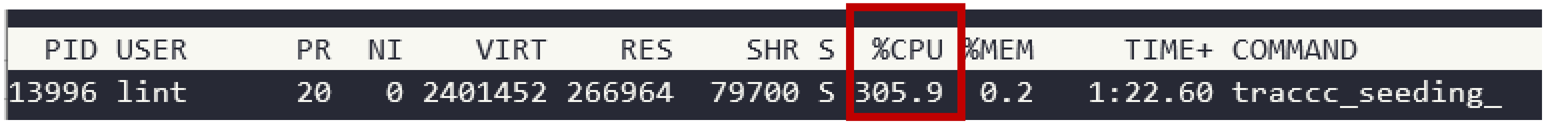
Intel oneapi provides CPU and FPGA backends



```
[lint@lxslc705]$ which dpcpp
/tmp/lint/intel/oneapi/compiler/2022.0.2/linux/bin/dpcpp
[lint@lxslc705]$ which gcc
/cvmfs/sft.cern.ch/lcg/releases/gcc/10.3.0-f5826/x86_64-centos7/bin/gcc
[lint@lxslc705]$ sycl-ls
[opencl:0] ACC : Intel(R) FPGA Emulation Platform for OpenCL(TM) 1.2 [2021.13.11.0.23_160000]
[opencl:0] CPU : Intel(R) OpenCL 3.0 [2021.13.11.0.23_160000]
[host:0] HOST: SYCL host platform 1.2 [1.2]
```

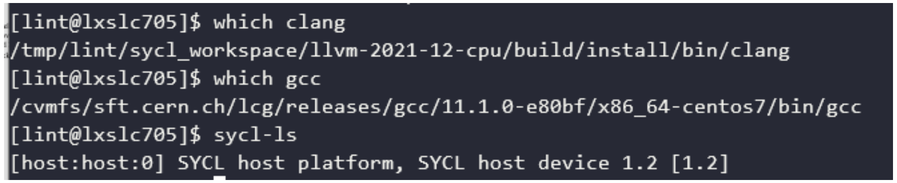


```
[lint@lxslc705]$ export TRACCC_TEST_DATA_DIR=$(pwd)/data
[lint@lxslc705]$ ./build_sycl/bin/traccc_seeding_example_sycl --detector_file=tml_detector/track
ml-detector.csv --hit_directory=tml_full/ttbar_mu300/ --events=10
Running ./build_sycl/bin/traccc_seeding_example_sycl tml_detector/trackml-detector.csv tml_full/
ttbar_mu300/ 10
Running on device: Intel(R) Xeon(R) CPU E5-2660 v4 @ 2.00GHz
==> Statistics ...
- read    1386759 spacepoints from 0 modules
- created (cpu)  0 seeds
- created (sycl) 264832 seeds
==> Elpased time ...
wall time           35.7607
hit reading (cpu)   26.0041
seeding_time (cpu)  1.0116e-05
seeding_time (sycl) 8.70652
tr_par_esti_time (cpu)    4.398e-06
tr_par_esti_time (sycl)   0.592434
```



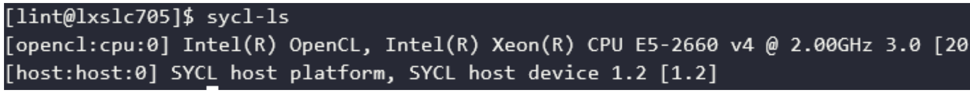
```
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
13996 lint      20   0 2401452 266964  79700 S 305.9  0.2   1:22.60 traccc_seeding_
```

Intel/llvm at IHEP login node



```
[lint@lxslc705]$ which clang
/tmp/lint/sycl_workspace/llvm-2021-12-cpu/build/install/bin/clang
[lint@lxslc705]$ which gcc
/cvmfs/sft.cern.ch/lcg/releases/gcc/11.1.0-e80bf/x86_64-centos7/bin/gcc
[lint@lxslc705]$ sycl-ls
[host:host:0] SYCL host platform, SYCL host device 1.2 [1.2]
```

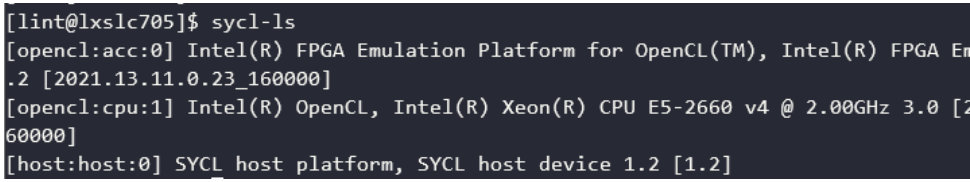
Before setting runtime, only SYCL host.



```
[lint@lxslc705]$ sycl-ls
[opencl:cpu:0] Intel(R) OpenCL, Intel(R) Xeon(R) CPU E5-2660 v4 @ 2.00GHz 3.0 [20
[host:host:0] SYCL host platform, SYCL host device 1.2 [1.2]
```

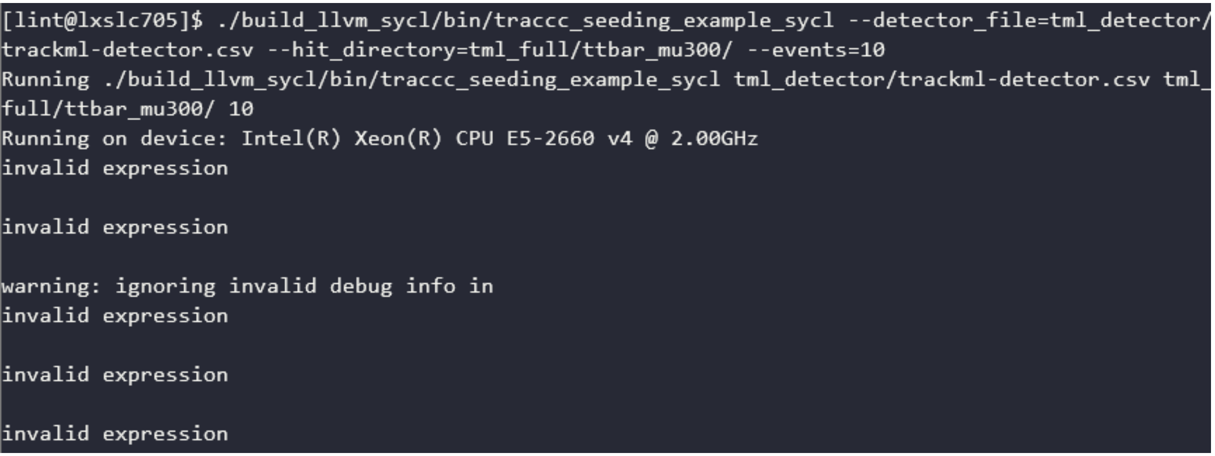
↑ Intel low level runtime

↓ Intel oneAPI

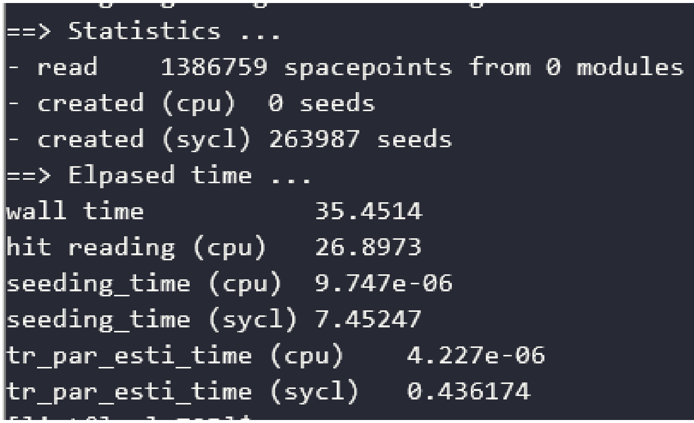


```
[lint@lxslc705]$ sycl-ls
[opencl:acc:0] Intel(R) FPGA Emulation Platform for OpenCL(TM), Intel(R) FPGA Em
.2 [2021.13.11.0.23_160000]
[opencl:cpu:1] Intel(R) OpenCL, Intel(R) Xeon(R) CPU E5-2660 v4 @ 2.00GHz 3.0 [2
60000]
[host:host:0] SYCL host platform, SYCL host device 1.2 [1.2]
```

After setting runtime, OpenCL could be found.



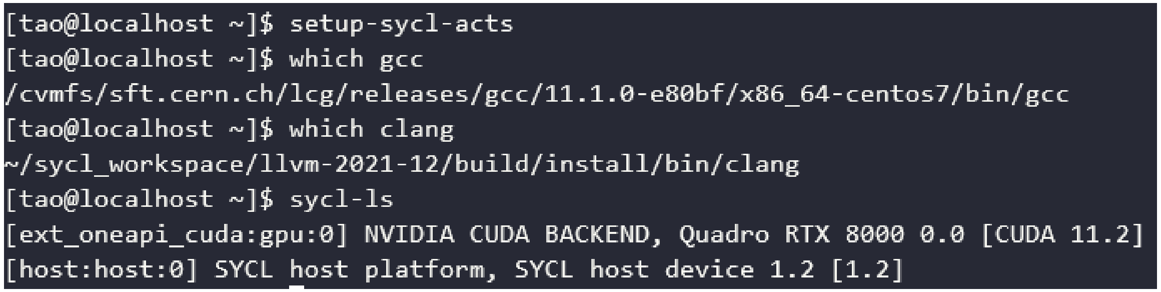
```
[lint@lxslc705]$ ./build_llvm_sycl/bin/traccc_seeding_example_sycl --detector_file=tml_detector/
trackml-detector.csv --hit_directory=tml_full/ttbar_mu300/ --events=10
Running ./build_llvm_sycl/bin/traccc_seeding_example_sycl tml_detector/trackml-detector.csv tml_
full/ttbar_mu300/ 10
Running on device: Intel(R) Xeon(R) CPU E5-2660 v4 @ 2.00GHz
invalid expression

invalid expression

warning: ignoring invalid debug info in
invalid expression

invalid expression

invalid expression
```



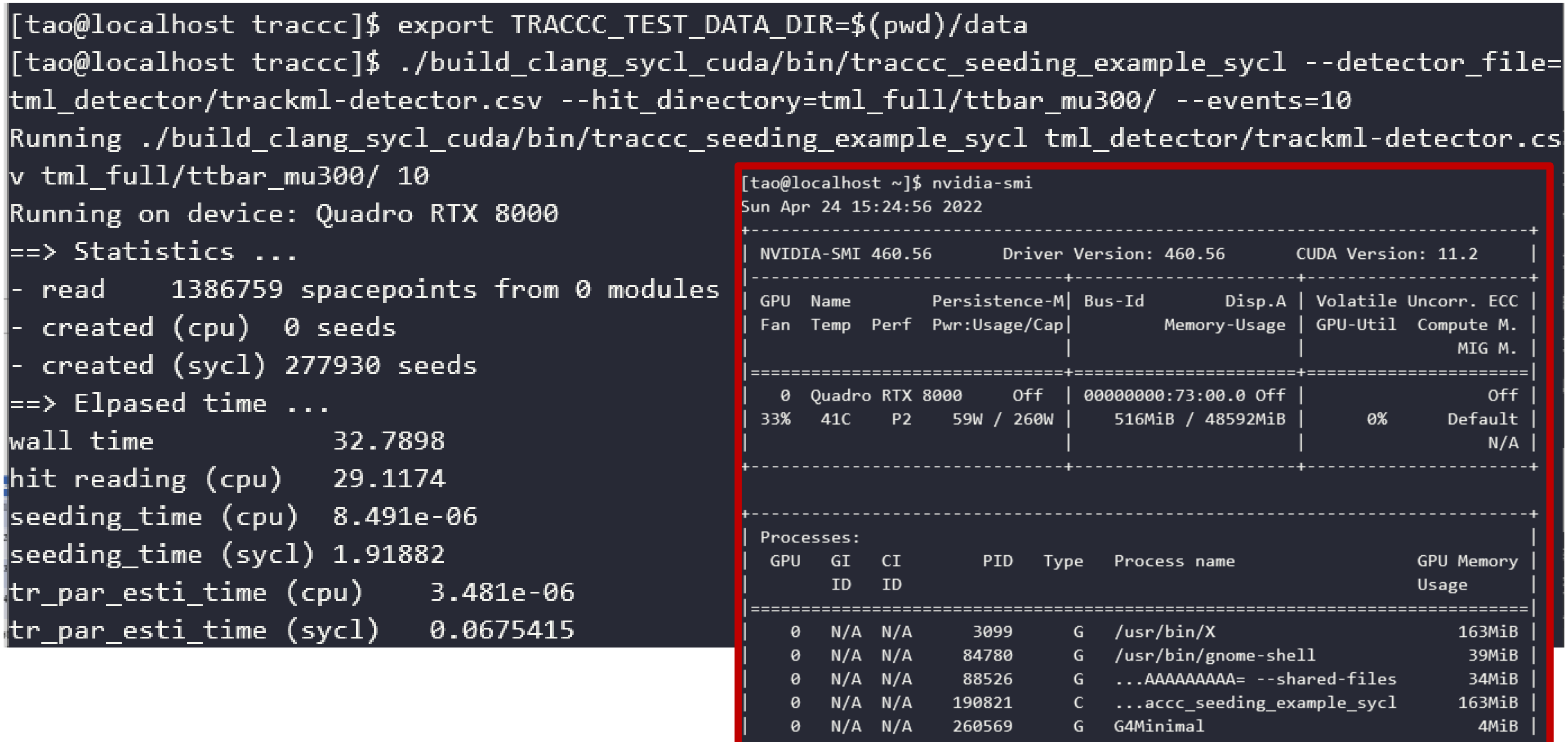
```
==> Statistics ...
- read    1386759 spacepoints from 0 modules
- created (cpu)  0 seeds
- created (sycl) 263987 seeds
==> Elpased time ...
wall time            35.4514
hit reading (cpu)    26.8973
seeding_time (cpu)   9.747e-06
seeding_time (sycl) 7.45247
tr_par_esti_time (cpu)    4.227e-06
tr_par_esti_time (sycl)   0.436174
```

But see errors “invalid expression”.

Intel/llvm + CUDA at workstation



```
[tao@localhost ~]$ setup-sycl-acts
[tao@localhost ~]$ which gcc
/cvmfs/sft.cern.ch/lcg/releases/gcc/11.1.0-e80bf/x86_64-centos7/bin/gcc
[tao@localhost ~]$ which clang
~/sycl_workspace/llvm-2021-12/build/install/bin/clang
[tao@localhost ~]$ sycl-ls
[ext_oneapi_cuda:gpu:0] NVIDIA CUDA BACKEND, Quadro RTX 8000 0.0 [CUDA 11.2]
[host:host:0] SYCL host platform, SYCL host device 1.2 [1.2]
```



```
[tao@localhost traccc]$ export TRACCC_TEST_DATA_DIR=$(pwd)/data
[tao@localhost traccc]$ ./build_clang_sycl_cuda/bin/traccc_seeding_example_sycl --detector_file=
tml_detector/trackml-detector.csv --hit_directory=tml_full/ttbar_mu300/ --events=10
Running ./build_clang_sycl_cuda/bin/traccc_seeding_example_sycl tml_detector/trackml-detector.cs
v tml_full/ttbar_mu300/ 10
Running on device: Quadro RTX 8000
==> Statistics ...
- read    1386759 spacepoints from 0 modules
- created (cpu)  0 seeds
- created (sycl) 277930 seeds
==> Elpased time ...
wall time           32.7898
hit reading (cpu)   29.1174
seeding_time (cpu)  8.491e-06
seeding_time (sycl) 1.91882
tr_par_esti_time (cpu)     3.481e-06
tr_par_esti_time (sycl)    0.0675415
```

```
[tao@localhost ~]$ nvidia-smi
Sun Apr 24 15:24:56 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.56       Driver Version: 460.56       CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Quadro RTX 8000     Off  | 00000000:73:00.0 Off |                  Off |
| 33%   41C    P2    59W / 260W |    516MiB / 48592MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    0   N/A  N/A      3099      G   /usr/bin/X                        163MiB |
|    0   N/A  N/A     84780      G   /usr/bin/gnome-shell               39MiB |
|    0   N/A  N/A     88526      G   ...AAAAAAAAA= --shared-files       34MiB |
|    0   N/A  N/A    190821      C   ...accc_seeding_example_sycl      163MiB |
|    0   N/A  N/A    260569      G   G4Minimal                           4MiB |
```

Other issues and notes

- Running traccc with intel DPC++ is crashed when ROOT is linking.
  - Maybe some conflicts between ROOT's internal TBB and oneAPI's internal TBB.
  - The code is modified a bit and ROOT related parts are removed.
  - My Git repo: git@github.com:mirguest/traccc.git
- The OpenCL runtime should be installed when using intel/llvm + CPU at IHEP login node.
  - Intel oneAPI or Intel low level Runtime (oclcpuexp and tbb) is setup when using intel/llvm.
  - Need to set envvar OPENCL_VENDOR_PATH to specify the OpenCL ICD file.
  - https://github.com/intel/llvm/blob/sycl/sycl/doc/GetStartedGuide.md#install-low-level-runtime
- The sycl-ls is failed when using intel/llvm + CUDA at IHEP login node.
  - This problem is not understood yet. Maybe some runtimes are missing.
  - As no administrator permission, can't upgrade any CUDA drivers or CUDA toolkits.
- At the workstation, I installed the intel DPC++ before building intel/llvm with CUDA, so some runtimes of DPC++ are already installed.

The next plans

- Learn basics of SYCL
    - James Reinders *et al.*, Data Parallel C++: Mastering DPC++ for Programming of Heterogeneous Systems using C++ and SYCL
    - https://link.springer.com/book/10.1007/978-1-4842-5574-2
- Understand the code in traccc
    - SYCL related parts
    - How to convert the input data and
- Integrate EDM4hep with traccc