Opticks Status and UK National GPU Hackathon Report

SWIFT-HEP Workshop

Adam Davis[1], Sajan Easo[2], Keith Evans[1], Lucas Giardey[2,3],
Evelina Gersabeck[1], Marco Gersabeck[1], Yunlong Li[1], Raja Nandakumar[2]

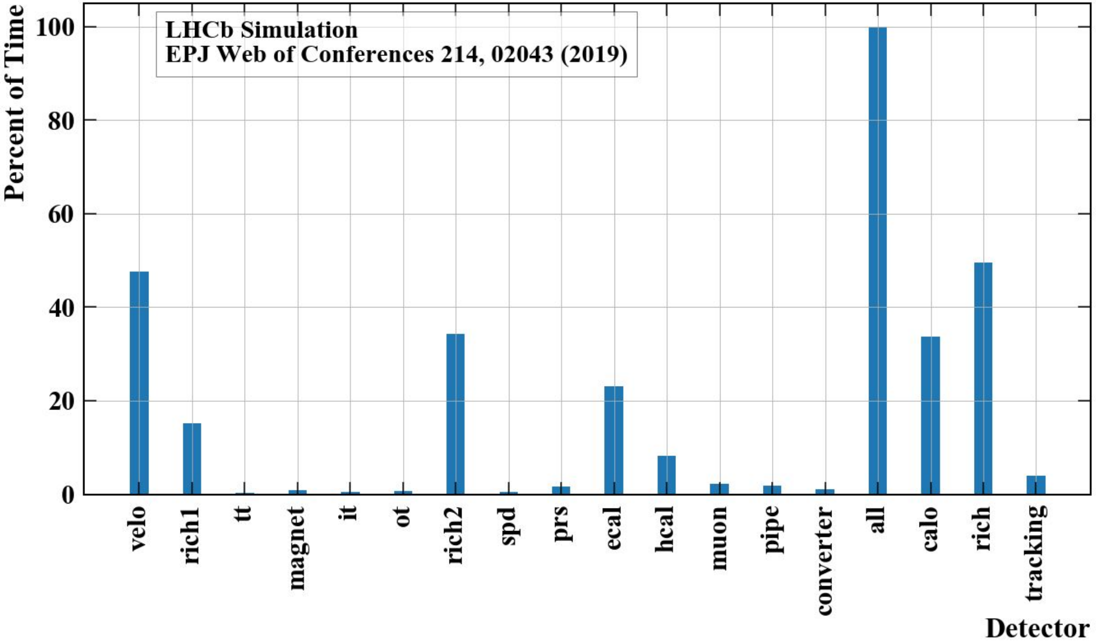
[1]The University of Manchester
[2]RAL
[3]Southampton University

24 March 2022

A Davis on behalf of many

Why do we care?

- Optical Photon simulation is an extremely time consuming process
- Just one example: LHCb

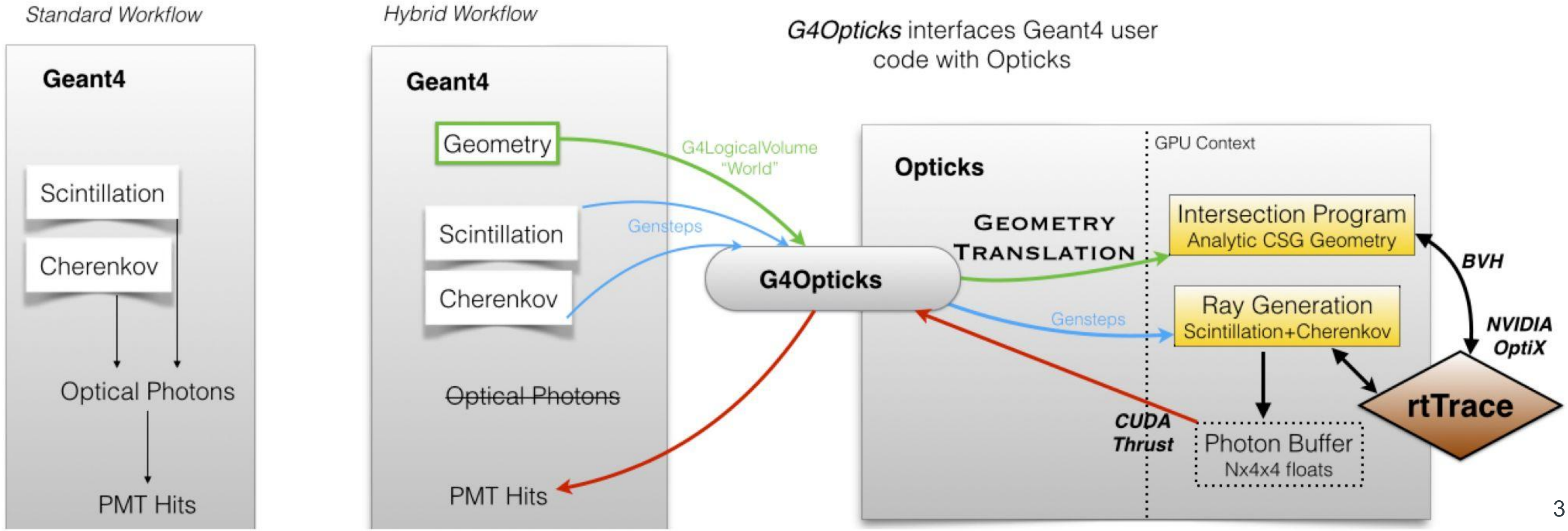


- Optical photon propagation is an embarrassingly parallel → ideally suited to GPU processing

Opticks in a nutshell

EPJ Web of Conferences 251, 03009 (2021)
EPJ Web of Conferences 245, 11003 (2020)
EPJ Web Conf. 214 02027 (2019)
J. Phys.: Conf. Ser. 898 042001

- Enter Opticks by Simon Blyth → Use NVIDIA RT Cores to perform this propagation

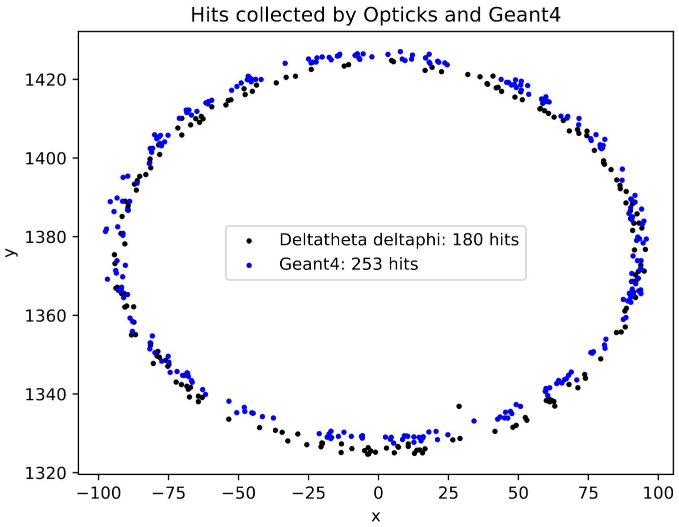


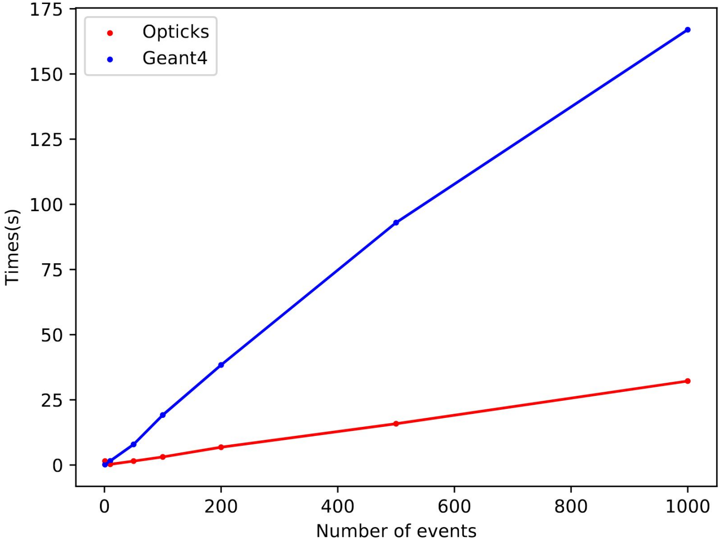
Since Last Time

- Driven largely by Hanz Wentzel of CaTS, Opticks incorporated as an advanced Geant4 Example in 11.0 release. Note the "node" from Geant4
- Very nice presentation at VCHEP from LZ on Opticks use
- Using additionally the CherenkovMinimal code from Sajan and Raja, we've implemented our own simplified RICH example using the LHCb RICH testbeam geometry
- From this, we've been able to uncover a few different issues:
  - Boolean subtraction of solids within Opticks was improved, and suggestions on possible geometry improvements provided by Simon Blyth
  - Ability to trace the exact hits available both in standalone example as well as in Opticks code
    - Fixed several geometry bugs within the simplified project
  - In preparation for GPU Hackathon, Simon ported the rendering to OptiX7

SimplifiedRich

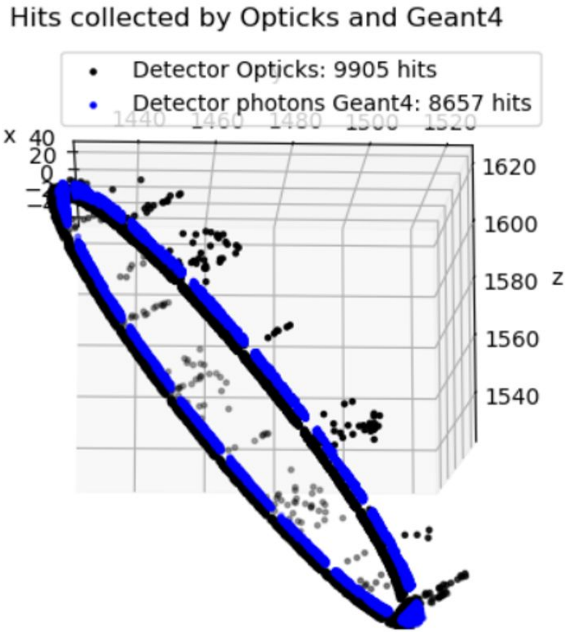
- Many features shown in the presentation at the last SWIFT-HEP workshop
- Simplified RICH test code created as a baseline: https://gitlab.cern.ch/liyu/opticks/
- Allows direct comparison of Geant4 and Opticks based hits with timing
  - Separate from implementation in Opticks
  - Basis for extendable CTest independent of geometry



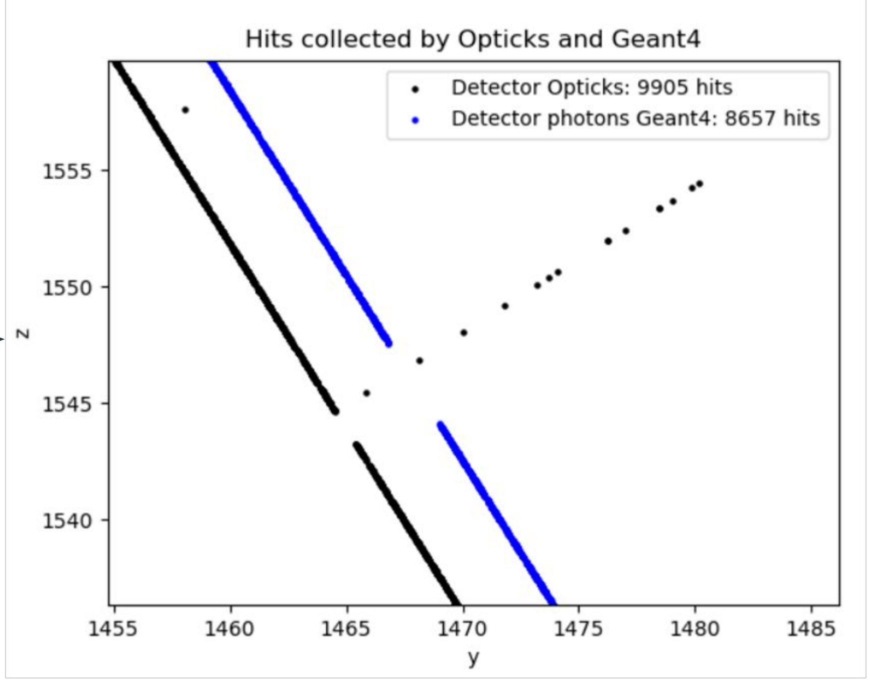


SimplifiedRich (2)

- Also has been extremely useful in debugging geometry issues

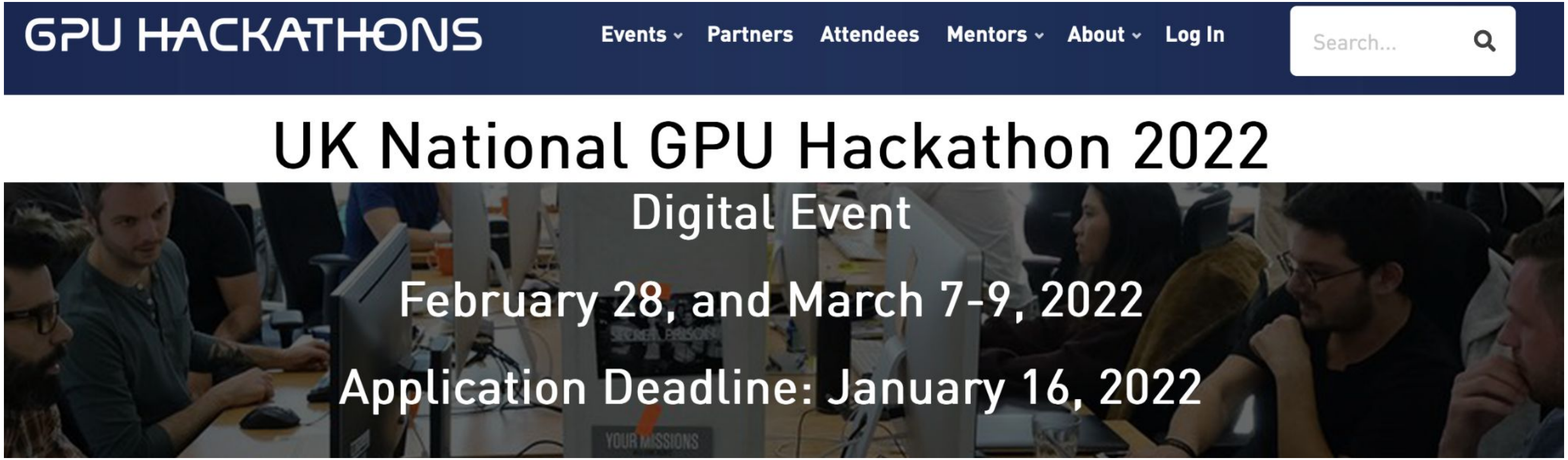


zoom



National GPU Hackathon

- UK National GPU Hackathon advertised before the end of the year



UK National GPU Hackathon 2022

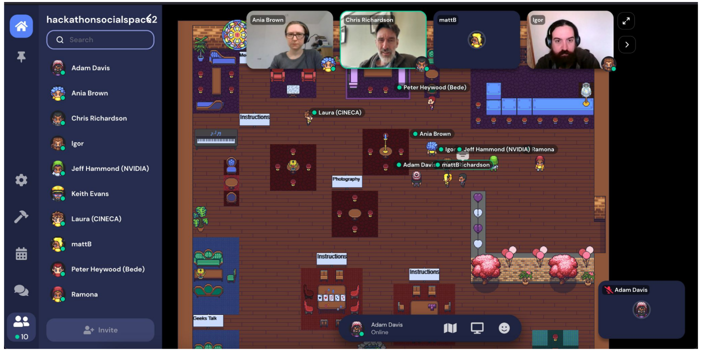
Digital Event

February 28, and March 7-9, 2022

Application Deadline: January 16, 2022

National GPU Hackathon: Logistics

- Took place remotely on the 28th of February and 7-9 of March with setup meeting on the 15th of February
- 5 projects chosen
  - HemeLB - Lattice Boltzmann method for 3D fluid dynamics in sparse geometries
  - Quantum Espresso - electronic structure calculations & material modeling at the nanoscale based on Density Functional Theory
  - x3div - Computational Fluid Dynamics solver to study turbulent flows
  - waveFEniCS - Solve time-dependent linear wave problems for high-intensity focused ultrasound to heat and destroy cancer cells
  - SWIFT-HEP Opticks
- Each project had assigned mentors
  - Ramona Hohl (NVIDIA) and Mateusz Malenta (UoM)
- Nearly all interactions happened via Slack and continuous Zoom breakout room with assigned mentors
- Complete with virtual meet-and greet



hackathonsocialspace2

A Davis on behalf of many

Available Compute Facilities

- **Baskerville** - Tier 2 HPC at University of Birmingham (4x A100 40GB GPGPUs per team)
- Bede - formerly NICE, Tier 2 HPC at Durham University (4x V100 GPUs per team)
- Raplab - dedicated private NVIDIA hackathon cluster (8x NVIDIA Volta V100 per node)
- Access to 4 node NVIDIA ARM HPC Developer Kit → We did not use this, but may be interesting for future works

- Also used HEP GPU cluster in Manchester (Tesla T4) as a failsafe
- Extra failsafe: Access to LHCb Online farm used for initial development by Raja and Sajan

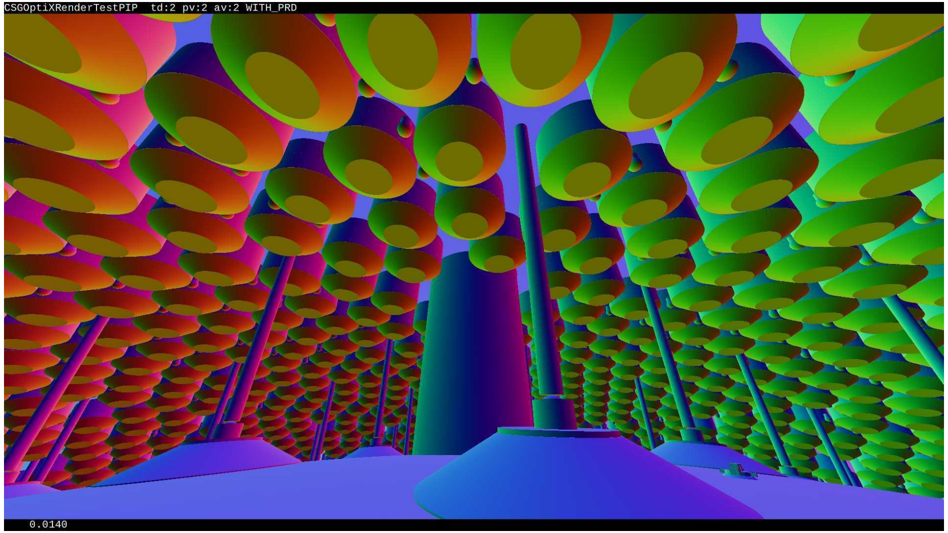
Goals of the Hackathon

- Understand from experts if there are clear areas for improvement on the geometry transfer from Geant4 to OptiX for GPU acceleration → Profiling
- Begin the definition of a unit test suite to automatically ensure capabilities with new releases of underlying software
  → Specifically with respect to different geometries and subtractions
  → add profiling
- Directly consult experts on design choices moving to OptiX 7

Thanks to the organizers, we were able to invite Simon Blyth to the hackathon who provided invaluable help

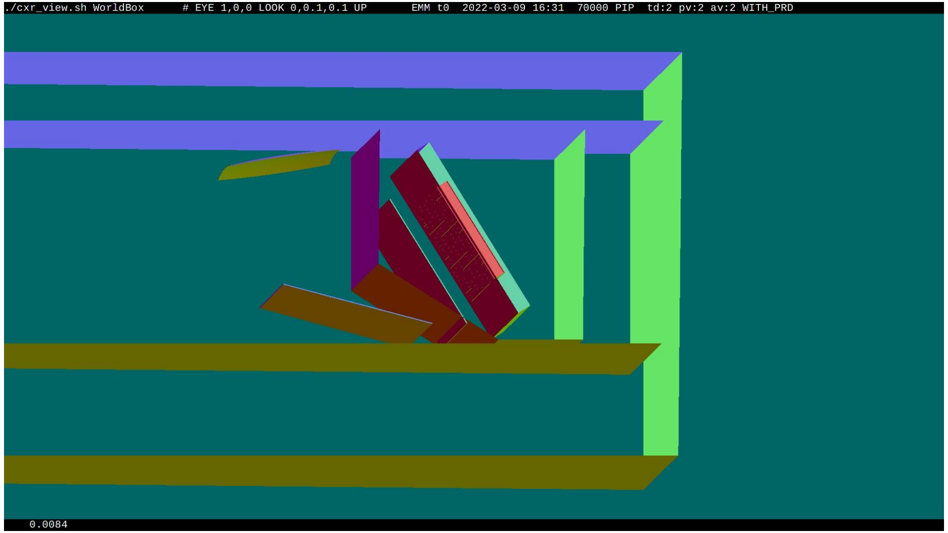
Geometries

- For Hackathon, use two different geometries for baseline testing
- JUNO geometry provided by Simon Blyth for OptiX 7
- Simplified RICH geometry



```
CSGOptiXRenderTestPIP  td:2 pv:2 av:2 WITH_PRD
0.0140
```

From Baskerville



```
./cxr_view.sh WorldBox      # EYE 1,0,0 LOOK 0,0.1,0.1 UP       EMM t0  2022-03-09 16:31  70000 PIP  td:2 pv:2 av:2 WITH_PRD
0.0084
```

From Manchester HEPGPU cluster

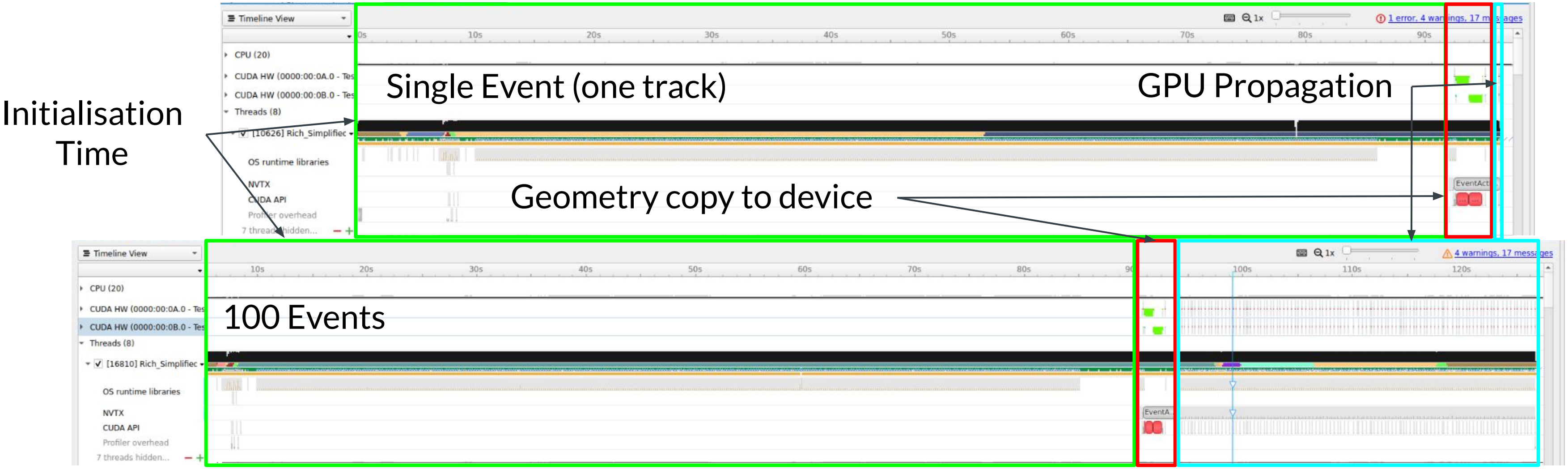
A Davis on behalf of many

Issues and Expert Discussions

- Compilation on an HPC environment was challenging to say the least
- At the end of the day, we did manage to get things working correctly, however, tried several backup options
  - Make a Docker container and ship off to HPC → Very similar to LZ approach from VCHEP. Did not manage to make this work
  - Use MicroMamba to install all dependencies - Worked on local machine tests, but due to complex dependencies on HPC environment, was not able to make work in the time allotted. However, one of the LHCb core software experts who also works on MicroMamba has expressed willingness to support Opticks when there is a more stable release version
- Used the time to instead have the expert discussions which were the other goals of the hackathon
  - Global payload trick example provided by Ramona to Simon, who began implementation
  - Discussed possibility of using VecGeom for geometry instead of pure Geant4
    - Put Ben M. in contact with Ramona, need to follow up

Profiling

- Profile the RICH Simplified geometry using NVIDIA Nsight profiling tools
  - To understand scaling, re-sample gensteps passed from Geant4 to bootstrap very large events and have control of number of photons used
- Also profile the rendering of the JUNO geometry with OptiX 7



Initialisation Time

Single Event (one track)

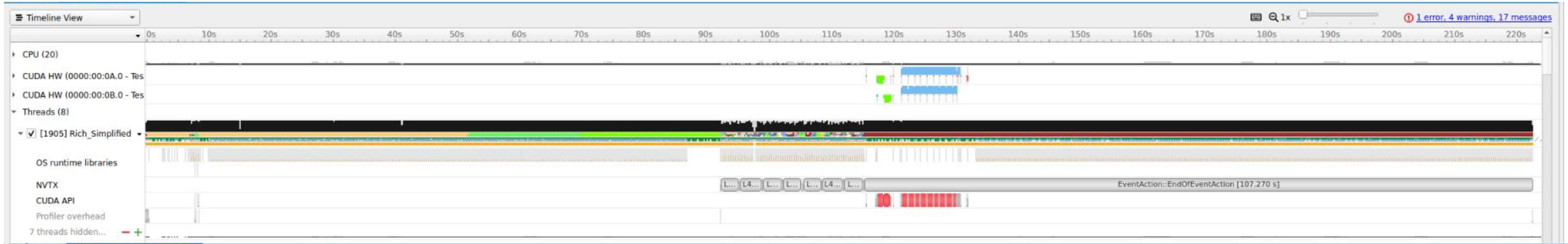
GPU Propagation

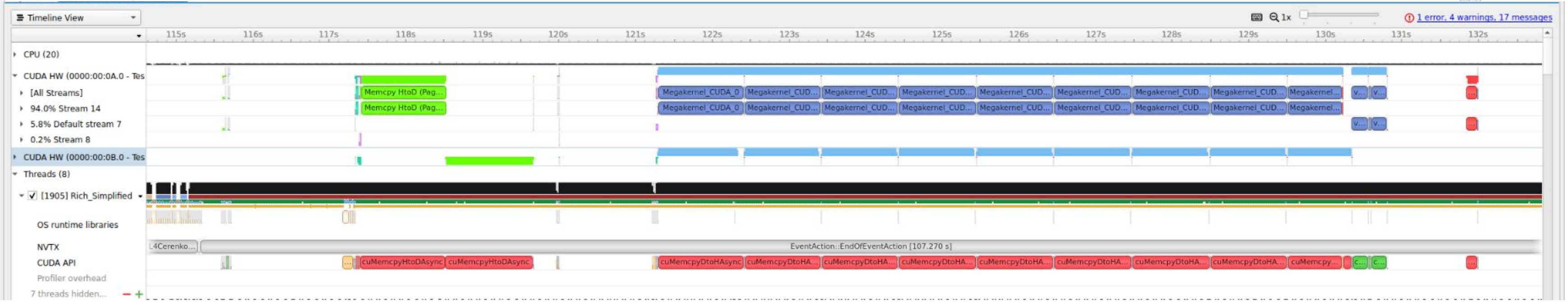
Geometry copy to device

100 Events

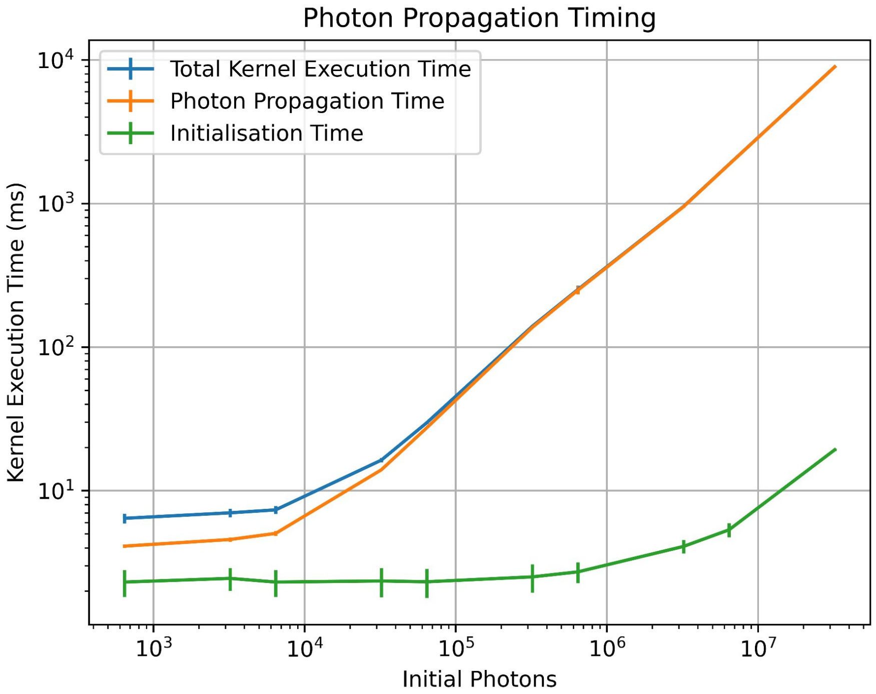
Profiling(2)

Single resampled event (50,000 samples) ~ 3.2 x 10^7 photons

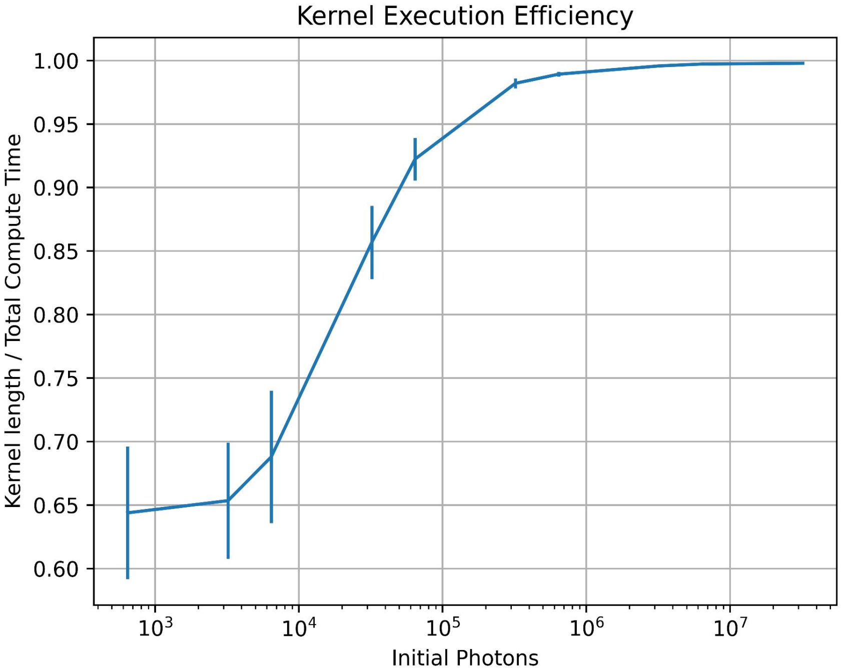




Scaling



Photon Propagation Timing
Total Kernel Execution Time
Photon Propagation Time
Initialisation Time
Kernel Execution Time (ms)
10^4
10^3
10^2
10^1
10^3
10^4
10^5
10^6
10^7
Initial Photons



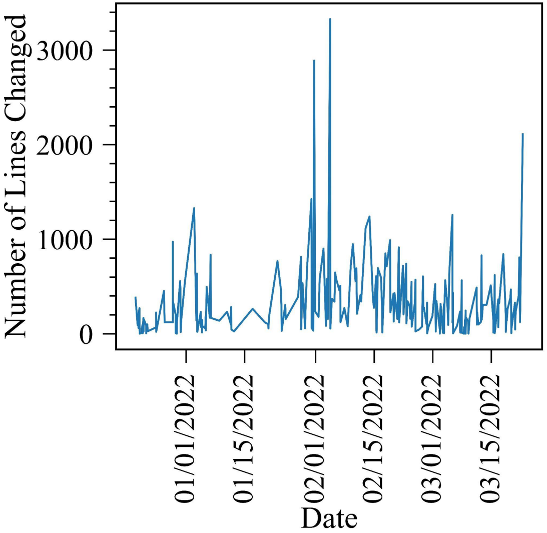
Kernel Execution Efficiency
Kernel length / Total Compute Time
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
10^3
10^4
10^5
10^6
10^7
Initial Photons

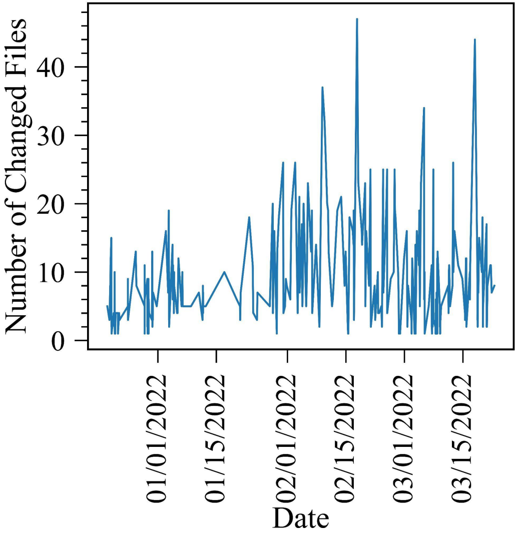
Discussion

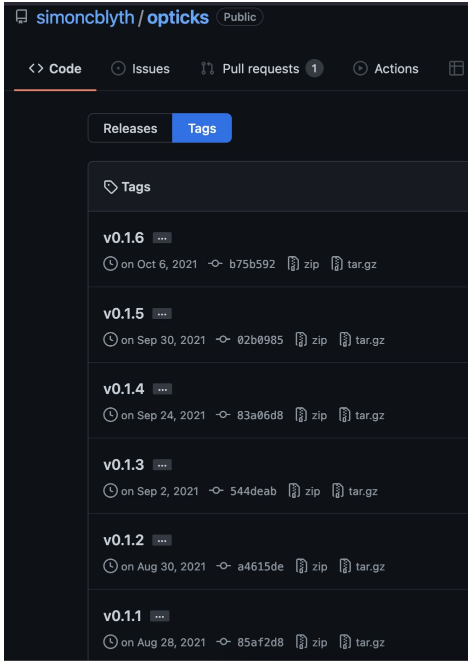
- CPU dominance depends on how many optical photons are to be propagated.
- Initialization of Opticks is a non-zero amount of time
- Only at greater than 10^5 optical photons do you get past 95% ratio of initialization to compute time
- This means there is a natural break even point per event - past this, one should think about multi-event parallelism to completely utilize the GPU resources available

Discussion (2)

- Opticks is a rapidly evolving, with tagged versions being released infrequently
- Fair to say that we are still in the development stage as opposed to deployment
- Active contributions from SWIFT-HEP can (and will) help to make this more stable







Tags

v0.1.6 ... on Oct 6, 2021 b75b592 zip tar.gz

v0.1.5 ... on Sep 30, 2021 02b0985 zip tar.gz

v0.1.4 ... on Sep 24, 2021 83a06d8 zip tar.gz

v0.1.3 ... on Sep 2, 2021 544deab zip tar.gz

v0.1.2 ... on Aug 30, 2021 a4615de zip tar.gz

v0.1.1 ... on Aug 28, 2021 85af2d8 zip tar.gz

Conclusion

Work on Opticks for general geometries is ongoing
Have found several bugs, and fixes have been implemented

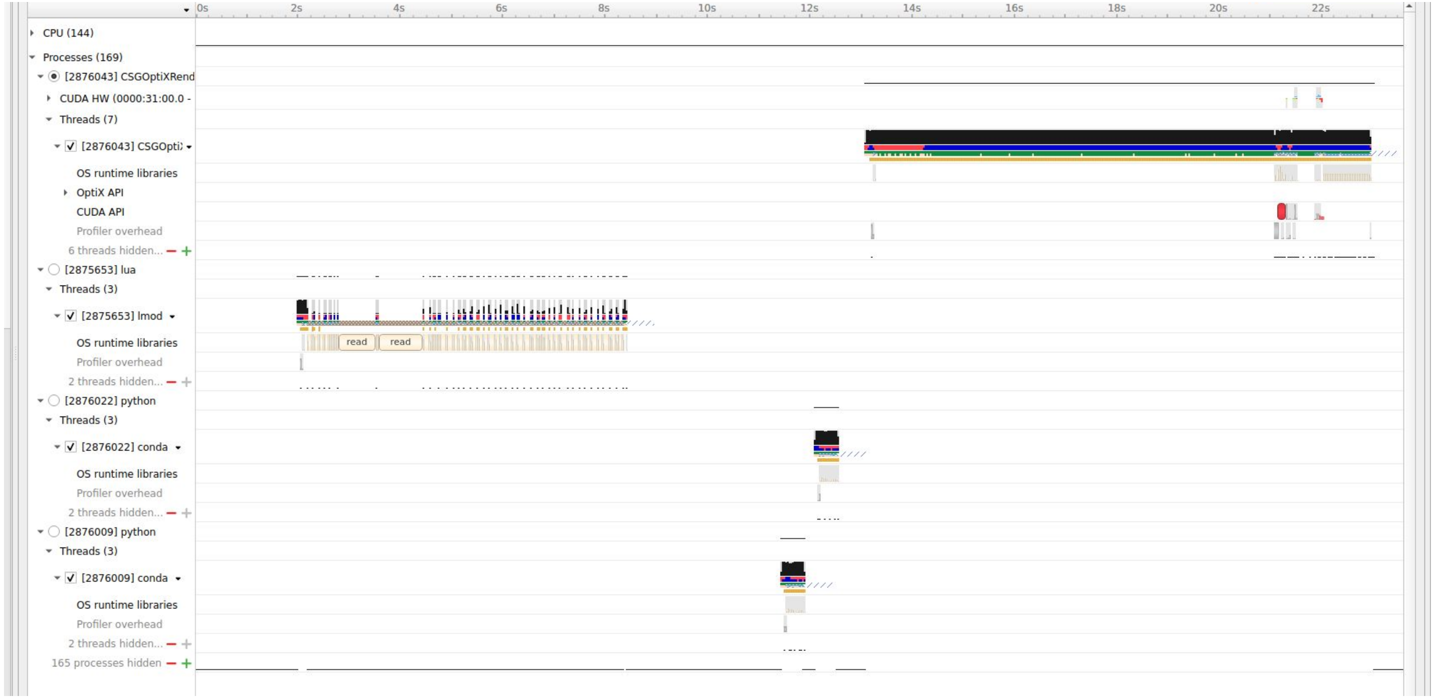
We participated in the National GPU Hackathon
Gained a lot of experience running NVIDIA profiling tools, and interpreting their output

Have a clear answer for “When is the right time to use Opticks”

Ongoing work to ensure unit tests for intersections are finalized and transition from “development mode” to “maintenance mode” is achieved

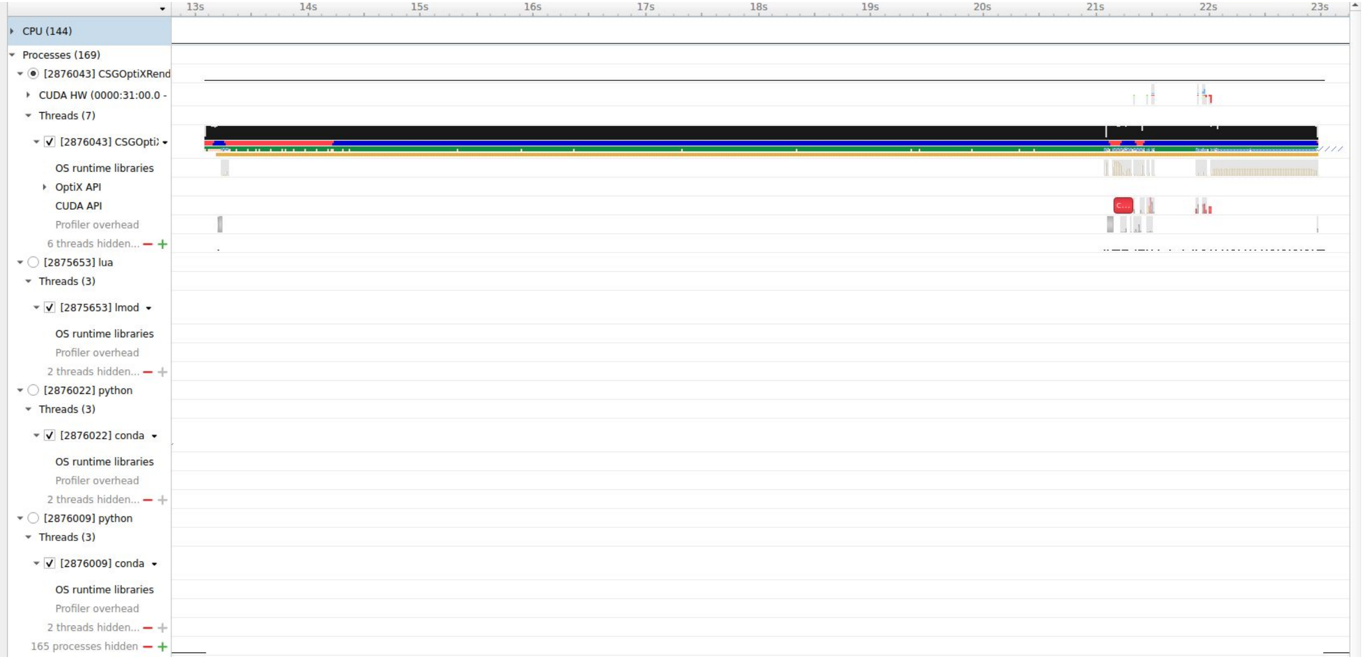
Fin

Profile of JUNO geometry on Baskerville with OptiX7

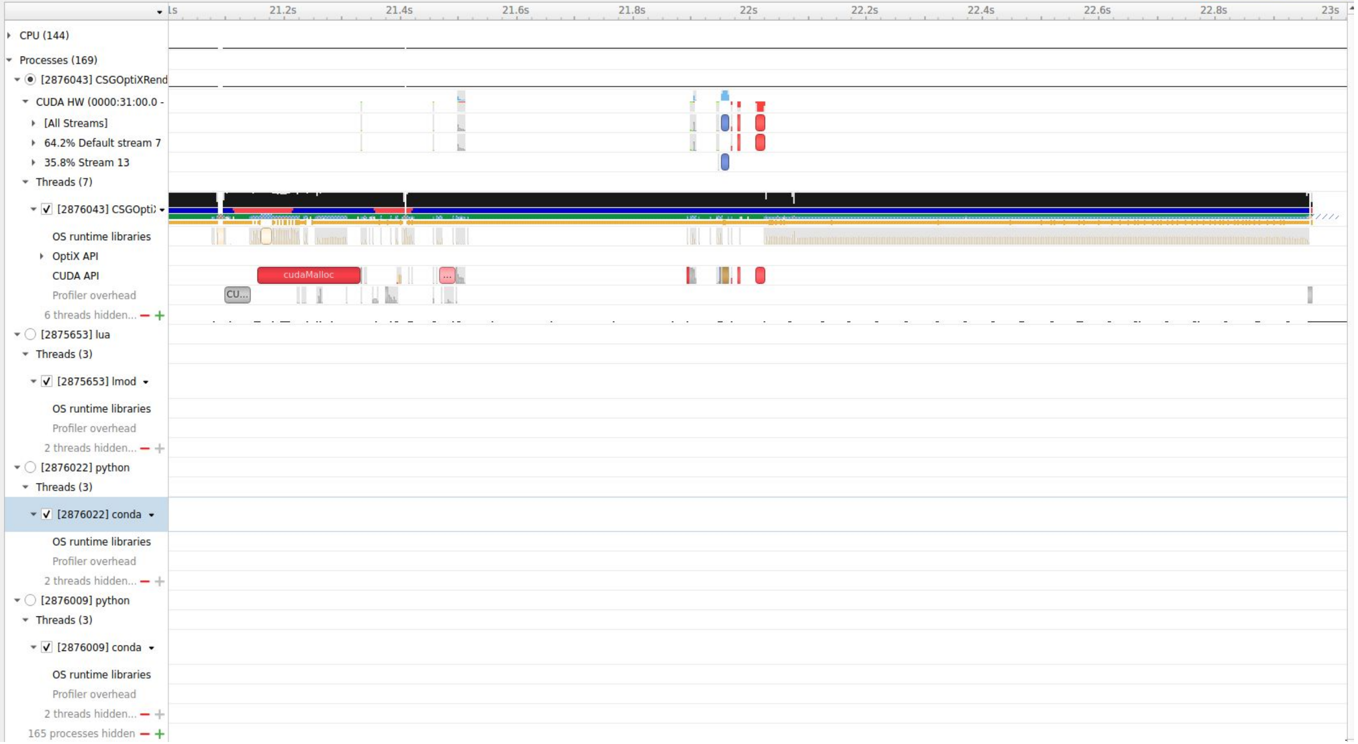


A Davis on behalf of many

Zoom


13s
14s
15s
16s
17s
18s
19s
20s
21s
22s
23s
CPU (144)
Processes (169)
[2876043] CSGOptiXRend
CUDA HW (0000:31:00.0 -
Threads (7)
[2876043] CSGOpti›
OS runtime libraries
OptiX API
CUDA API
Profiler overhead
6 threads hidden...
[2875653] lua
Threads (3)
[2875653] lmod
OS runtime libraries
Profiler overhead
2 threads hidden...
[2876022] python
Threads (3)
[2876022] conda
OS runtime libraries
Profiler overhead
2 threads hidden...
[2876009] python
Threads (3)
[2876009] conda
OS runtime libraries
Profiler overhead
2 threads hidden...
165 processes hidden

Zoom again


1s
21.2s
21.4s
21.6s
21.8s
22s
22.2s
22.4s
22.6s
22.8s
23s
CPU (144)
Processes (169)
[2876043] CSGOptiXRend
CUDA HW (0000:31:00.0 -
[All Streams]
64.2% Default stream 7
35.8% Stream 13
Threads (7)
[2876043] CSGOpti
OS runtime libraries
OptiX API
CUDA API
Profiler overhead
6 threads hidden...
cudaMalloc
CU...
[2875653] lua
Threads (3)
[2875653] lmod
OS runtime libraries
Profiler overhead
2 threads hidden...
[2876022] python
Threads (3)
[2876022] conda
OS runtime libraries
Profiler overhead
2 threads hidden...
[2876009] python
Threads (3)
[2876009] conda
OS runtime libraries
Profiler overhead
2 threads hidden...
165 processes hidden