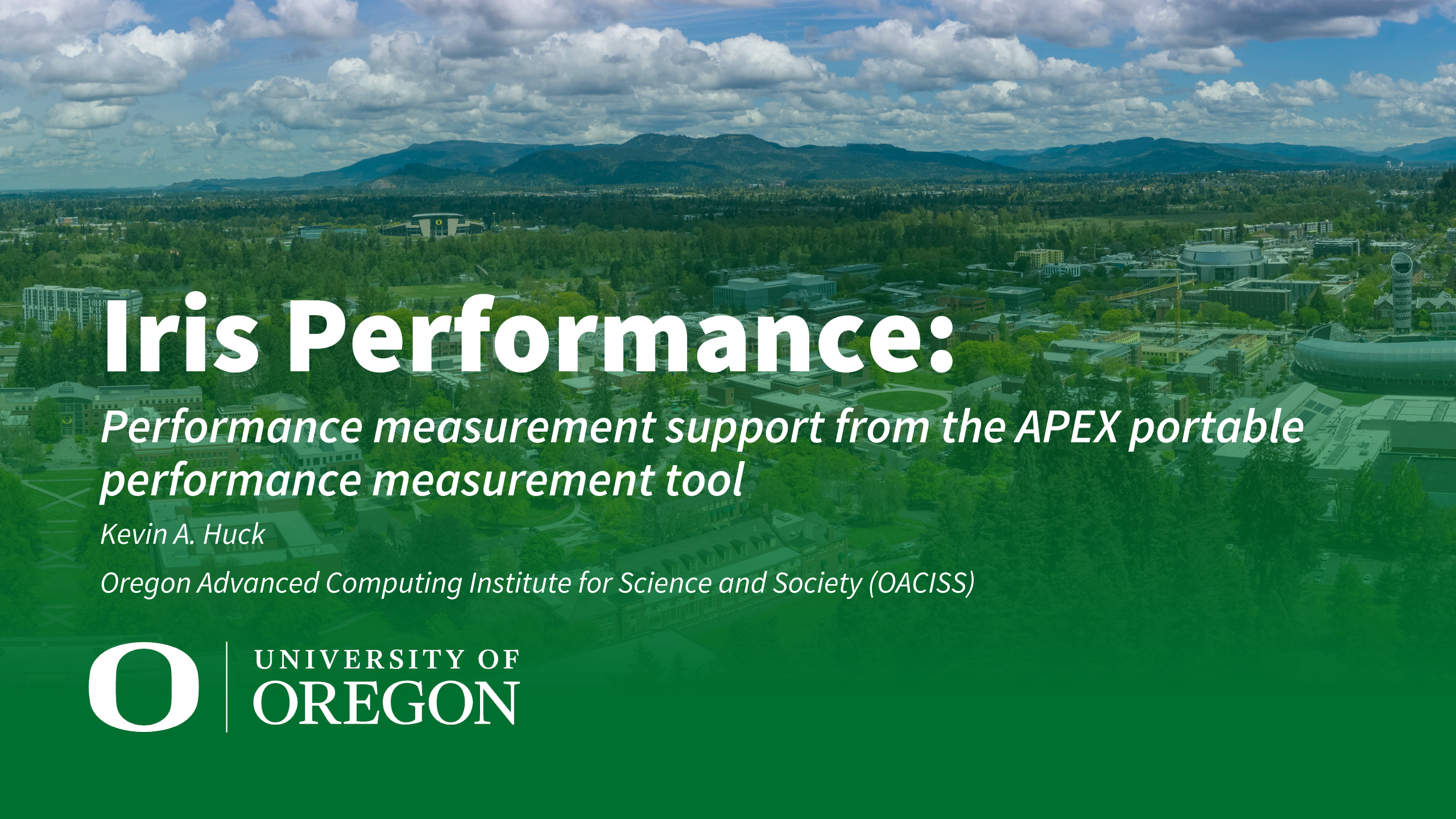


Iris Performance:

Performance measurement support from the APEX portable performance measurement tool

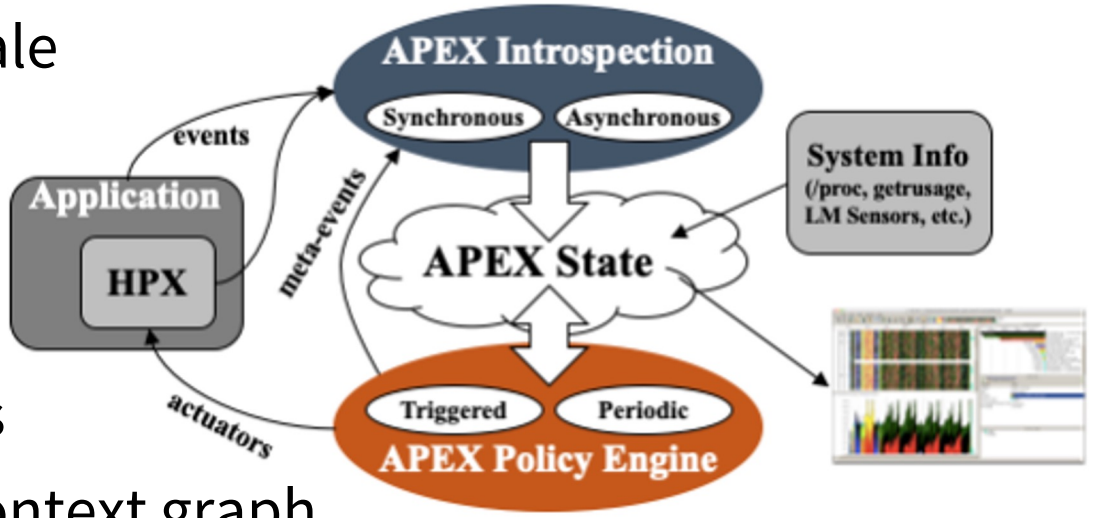
Kevin A. Huck

Oregon Advanced Computing Institute for Science and Society (OACISS)

UNIVERSITY OF OREGON

Autonomic Performance Environment for Exascale (APEX)

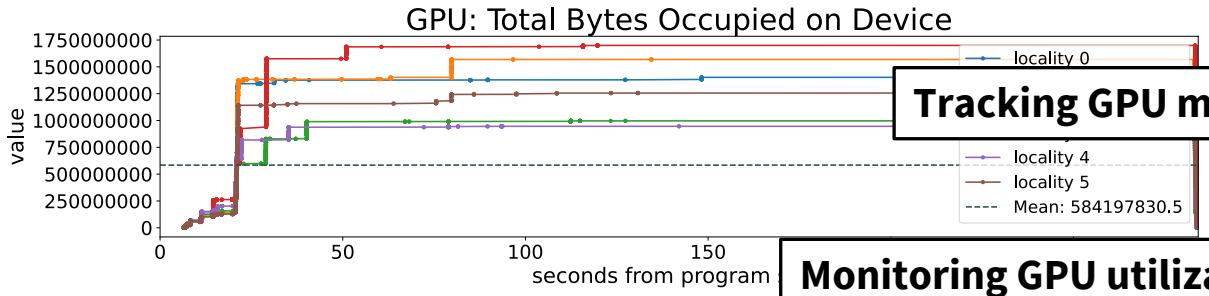
- Autonomic Performance Environment for eXascale
- **Performance Measurement**
- **Runtime Adaptation**
- Designed for AMT runtimes (HPX)
  - but works with conventional parallel models



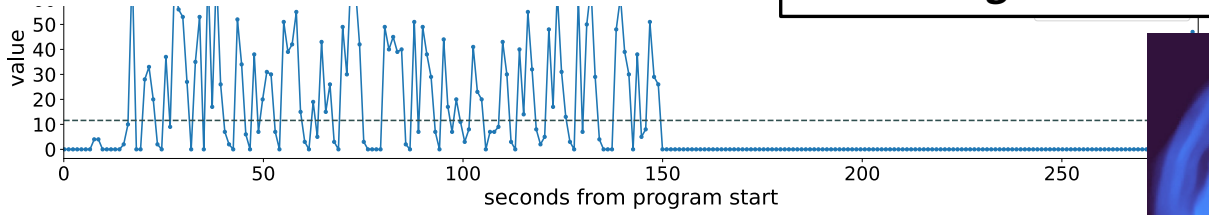
- Focus on **task dependency graph**, not calling context graph
- Supports HPX, C/C++ threads, OpenMP, OpenACC, Kokkos, Raja, CUDA, HIP, SYCL, StarPU... Working on PARSEc, YAKL, **Iris**
- https://github.com/UO-OACISS/apex and https://github.com/khuck/apex-tutorial
- Active Harmony* (Nelder Mead), simulated annealing, genetic search, hill climbing for parametric search methods, as well as exhaustive and random

https://doi.org/10.1109/ESPM256814.2022.00008 : "Broad Performance Measurement Support for Asynchronous Multi-Tasking with APEX", Huck, ESPM, 2022

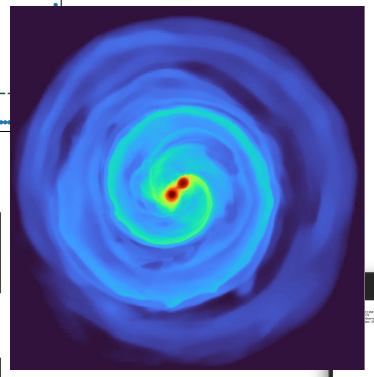
APEX example – Octo-Tiger (Octree astrophysics in HPX, Kokkos)

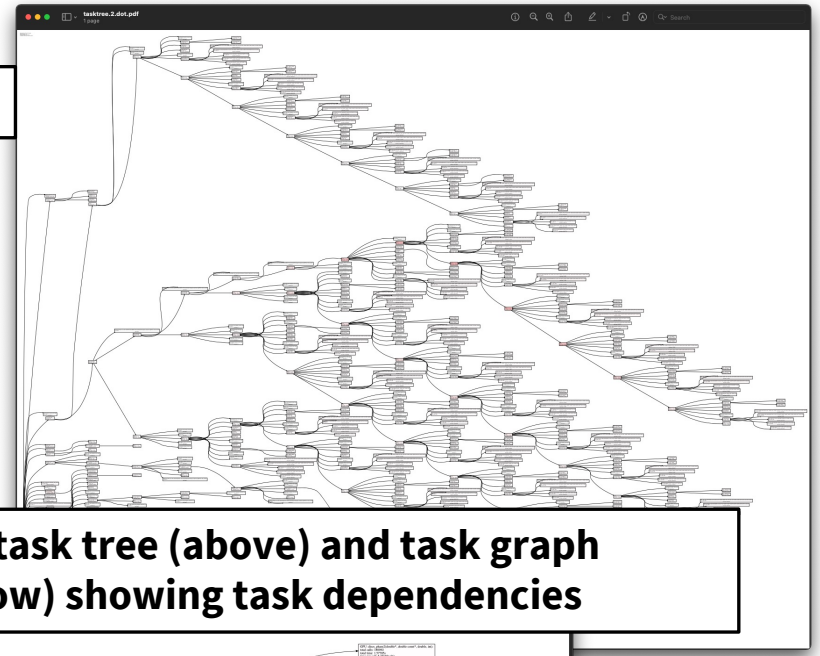


Tracking GPU memory usage with CUPTI

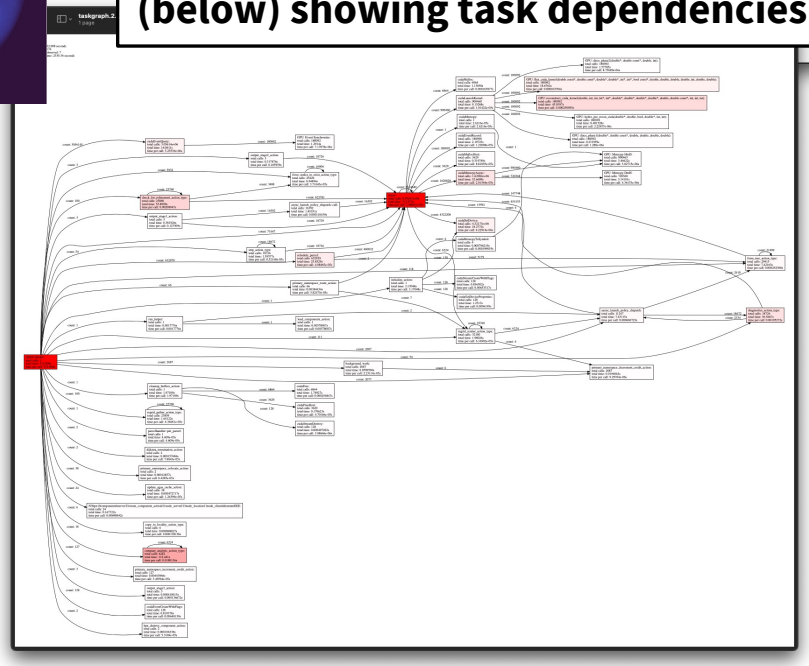


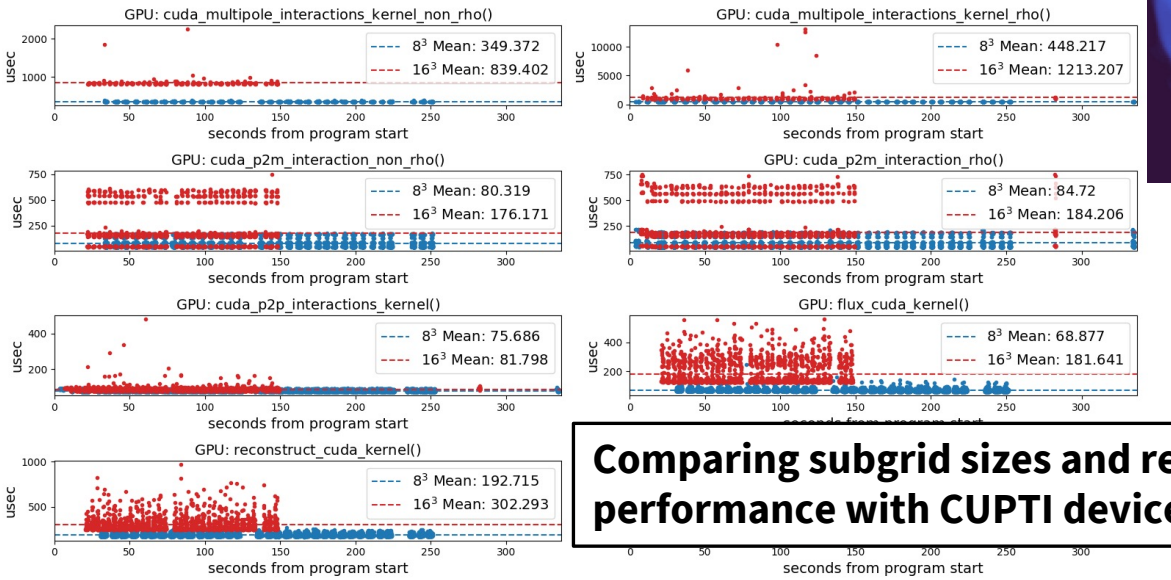
Monitoring GPU utilization with NVML library





Full task tree (above) and task graph (below) showing task dependencies



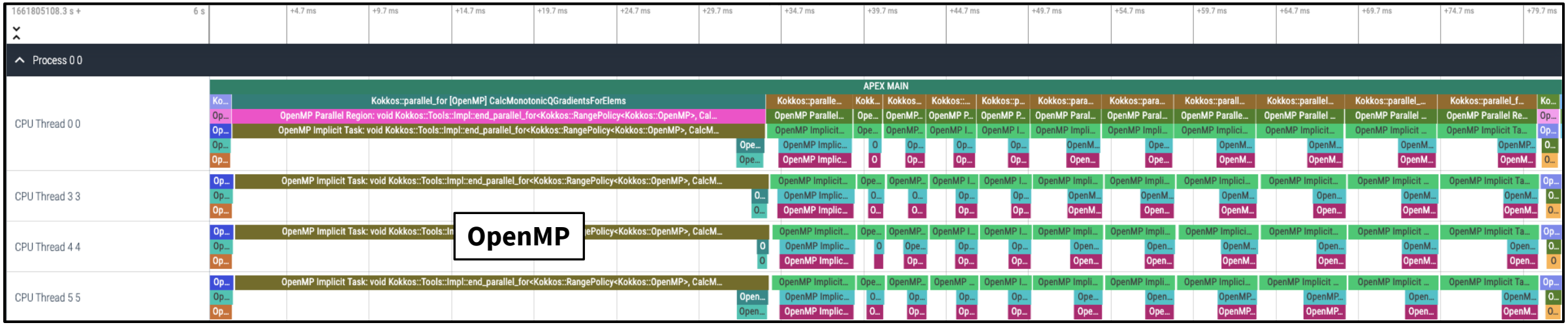


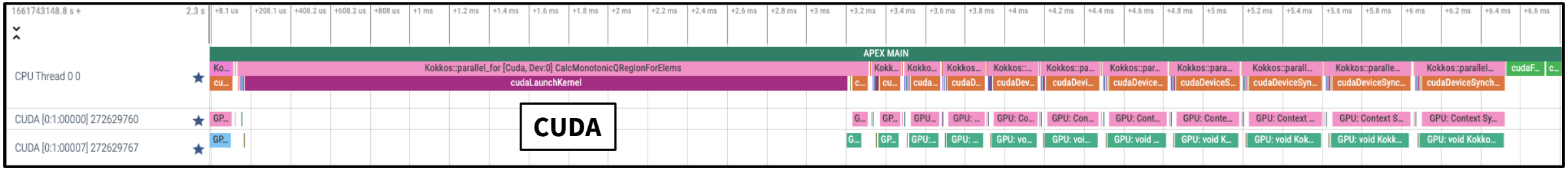
Comparing subgrid sizes and relative kernel performance with CUPTI device activity

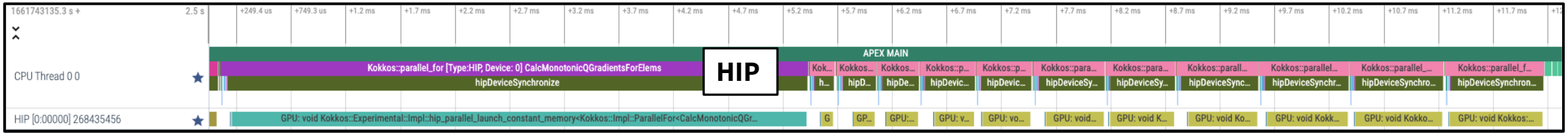
UNIVERSITY OF OREGON

https://github.com/STEllAR-GROUP/octotiger
https://github.com/STEllAR-GROUP/hpx

Kokkos Lulesh and APEX Tracing – OpenMP, CUDA, HIP back ends







UNIVERSITY OF OREGON

Kokkos Lulesh: https://github.com/kokkos/kokkos-miniapps

Trace visualization in Perfetto https://ui.perfetto.dev/

Kokkos Runtime Tuning Support in APEX

- APEX implements the same profiling API that TAU does, *and…*
- APEX provides auto-tuning (search) support
- Kokkos provides the ability to auto-tune with (at Cmake configuration time):

  `-DKokkos_ENABLE_TUNING=ON`

- Automatically provides input and context variables for parallel_for, parallel_reduce, parallel_scan, parallel_copy.
  - TeamPolicy: team size and vector length
  - MDRangePolicy: X, Y tile sizes (Z stays constant)
  - RangePolicy*: block size, occupancy

Provides an alternative to "heuristic" parameter choices

- Custom tuning also available – see https://github.com/khuck/apex-kokkos-tuning for examples like mm2d_tiling.cpp and idk_jmm.cpp

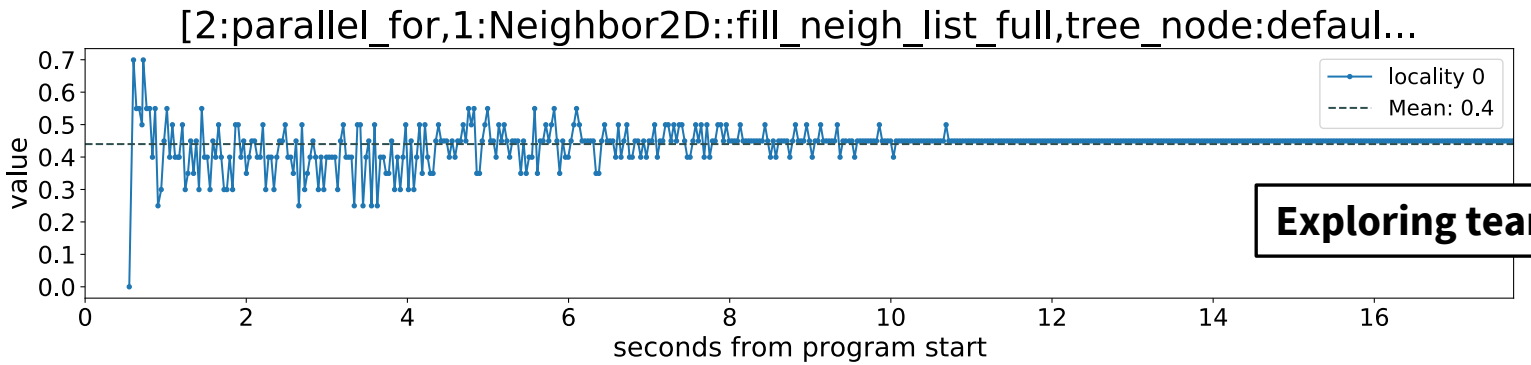
*(in a long-dormant development fork/branch, a new, updated PR for occupancy tuning has passed all tests and will be merged soon…would be nice to have because many kernels use RangePolicy)

APEX Autotuning of ExaMiniMD Neighbor2D::fill_neigh_list_full kernel

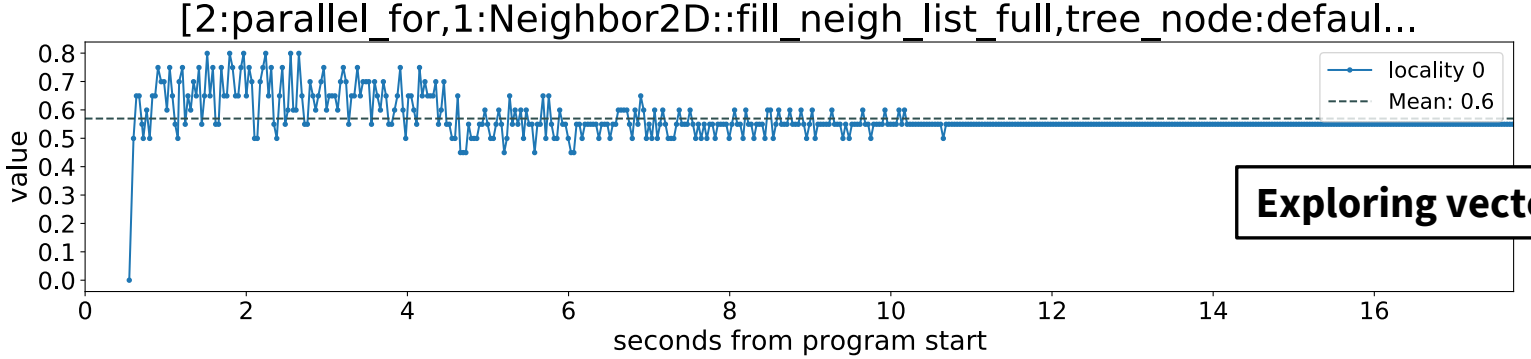
Much more promising results in the apex-kokkos-tuning repository

Baseline	17.4972s
Tuning	17.7294s
Tuned	17.3790s

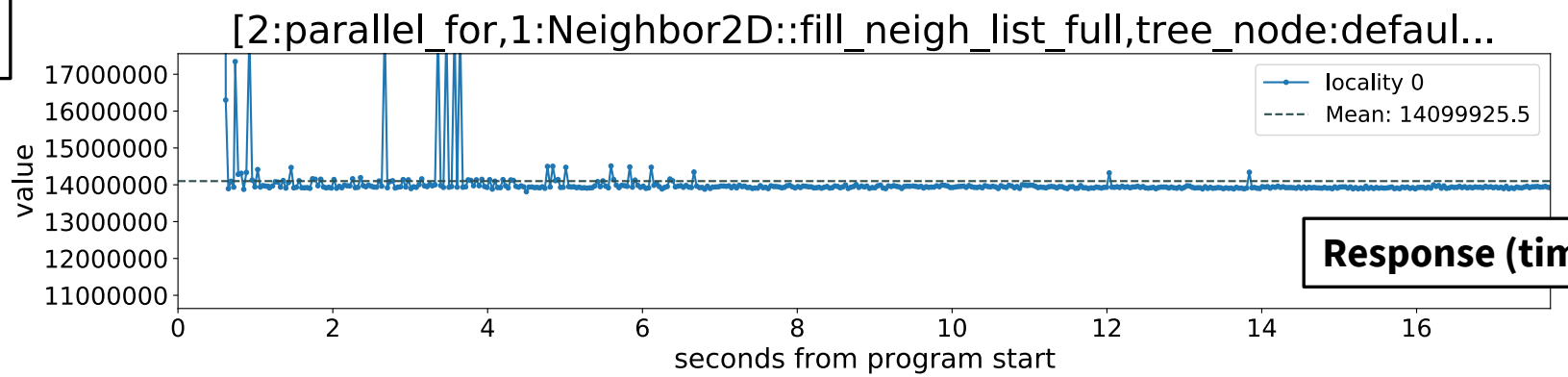
Only one kernel is TeamPolicy in ExaMiniMD – all of the rest of the kernels are Range policies…



Exploring team sizes



Exploring vector lengths



Response (time)

UNIVERSITY OF OREGON

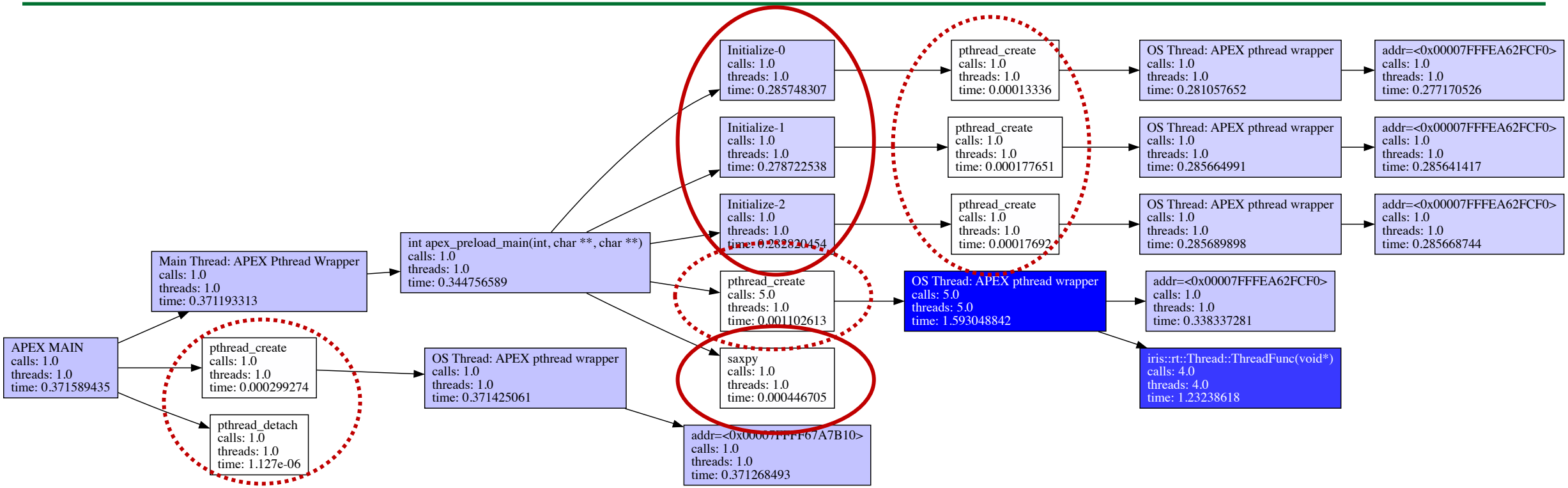
TaskStubs

- TaskStubs is a “frictionless” instrumentation library for tasking systems
  - https://github.com/khuck/taskStubs
  - One source file, three headers
  - Provides a plugin interface for performance tools
  - Can be compiled away if desired
  - Related to PerfStubs: https://github.com/UO-OACISS/perfstubs
- Integrating into several libraries as a git submodule
  - Iris
  - PARSEc
  - StarPU
- Provides runtime integration with APEX

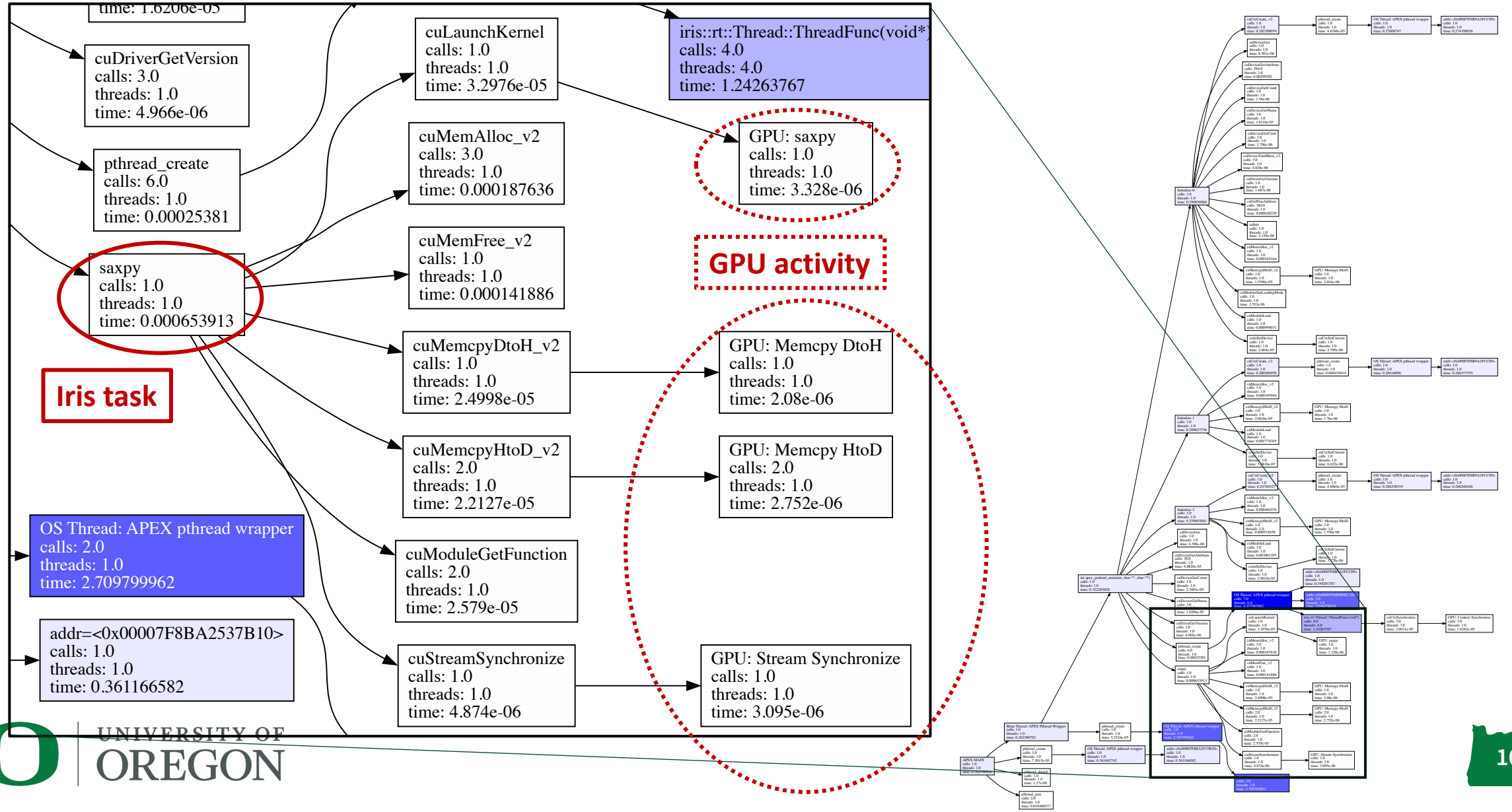
Integration into Iris

- ProfilerTaskStubs class, derived from iris::rt::Profiler
- Task lifecycle:
  - Construct
  - Schedule
  - Start
  - Yield
  - Resume
  - Stop
  - Destroy
  - Add children
  - Add parents

Iris example: saxpy with APEX, pthread support enabled



Iris example: saxpy with APEX, pthread and CUDA support enabled



Current status

- Created ProfilerTaskStubs class
- Integrated into Platform, Task, Worker, Scheduler classes
- Next steps:
  - Add the other (currently) stubbed arguments
  - Handle task dependencies across resources correctly
  - Additional testing with MatRIS
  - Work with Monil on all these items
- Questions:
  - Have I integrated into the right places? (probably not)
  - Are there auto-tuning opportunities in Iris? (probably)

Acknowledgements

Parts of this research was supported by the Exascale Computing Project (17-SC-20-SC), a joint project of the U.S. Department of Energy's Office of Science and National Nuclear Security Administration, responsible for delivering a capable exascale ecosystem, including software, applications, and hardware technology, to support the nation's exascale computing imperative.

This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

UNIVERSITY OF OREGON