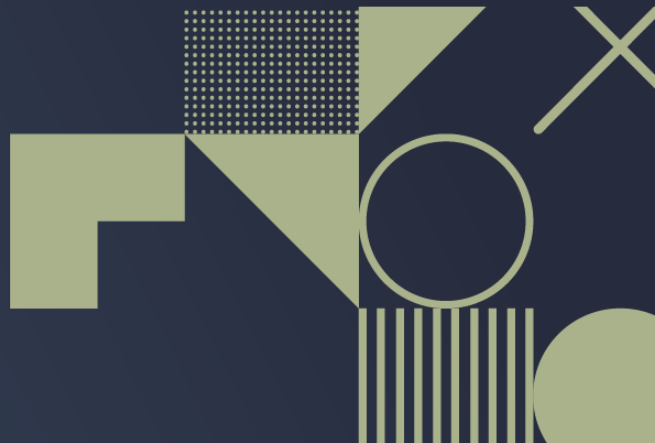
Analyzing GPU-accelerated Applications with HPCToolkit

John Mellor-Crummey

Department of Computer Science
Rice University

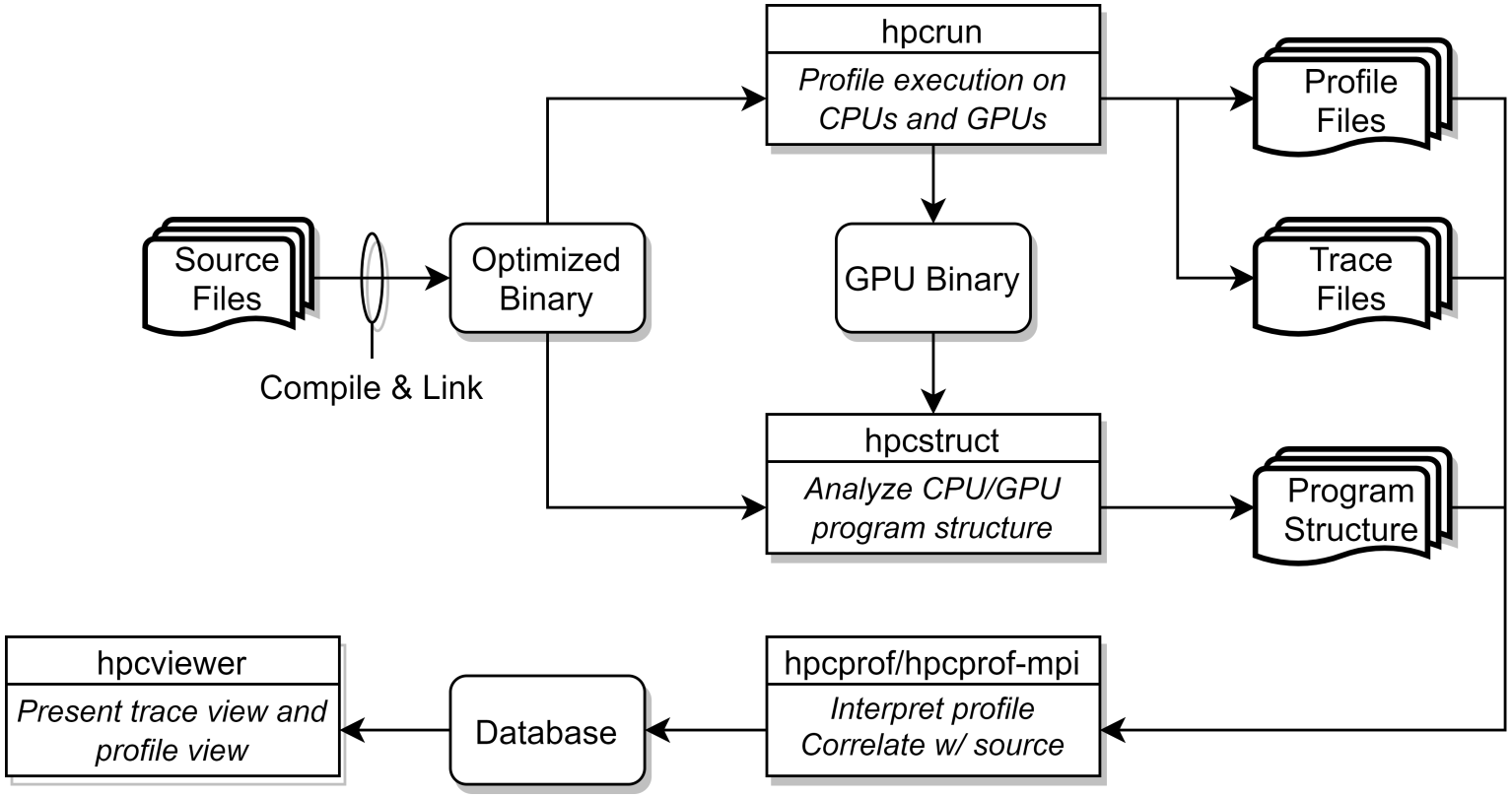
15 November 2021



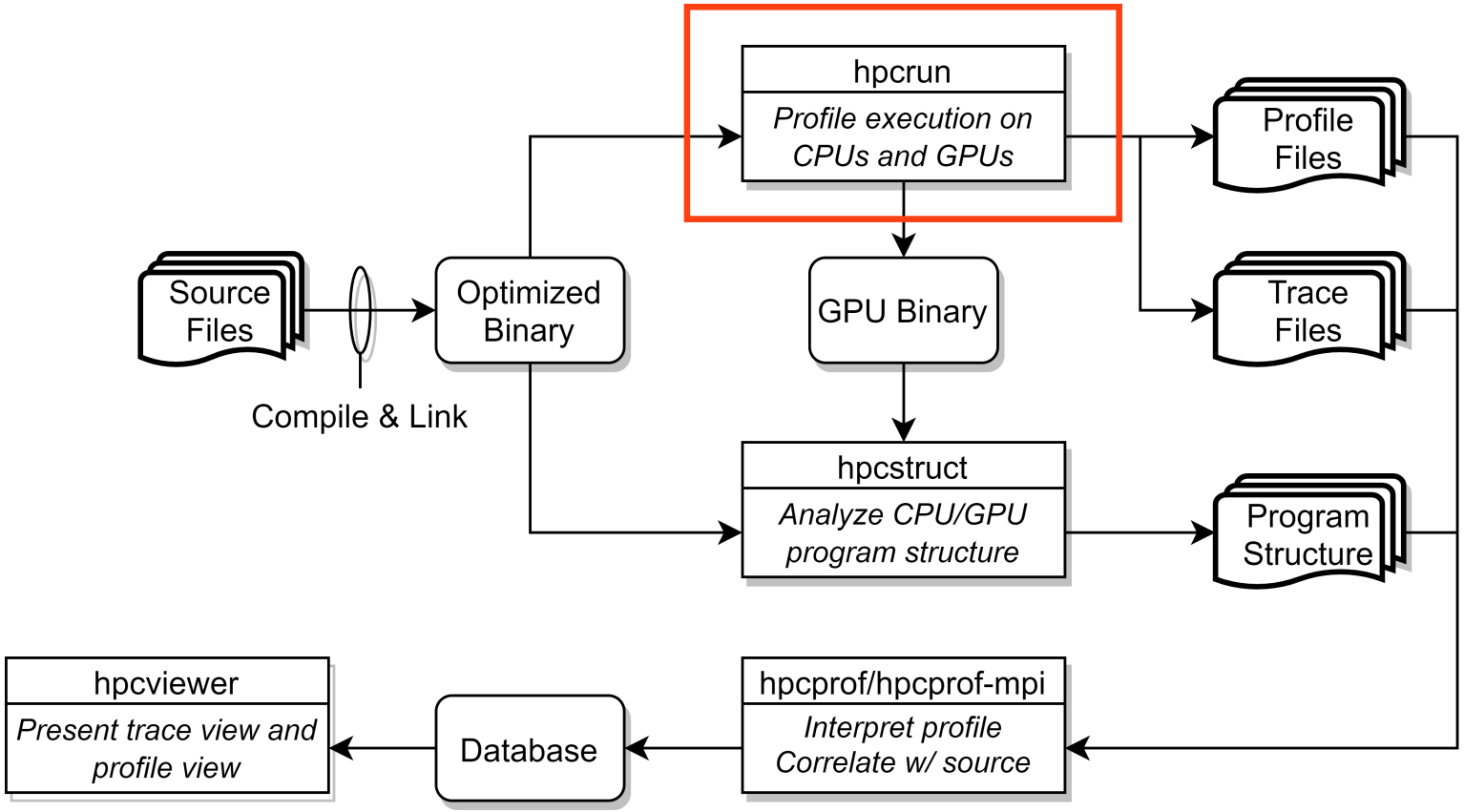
Outline

- **Describe use of HPCToolkit for GPU-accelerated applications**
- **Status for GPU vendors**
- **Demo on NVIDIA GPU**
  - operation-level monitoring
  - PC sampling
- **Ongoing work**

HPCToolkit's Workflow for GPU-accelerated Applications



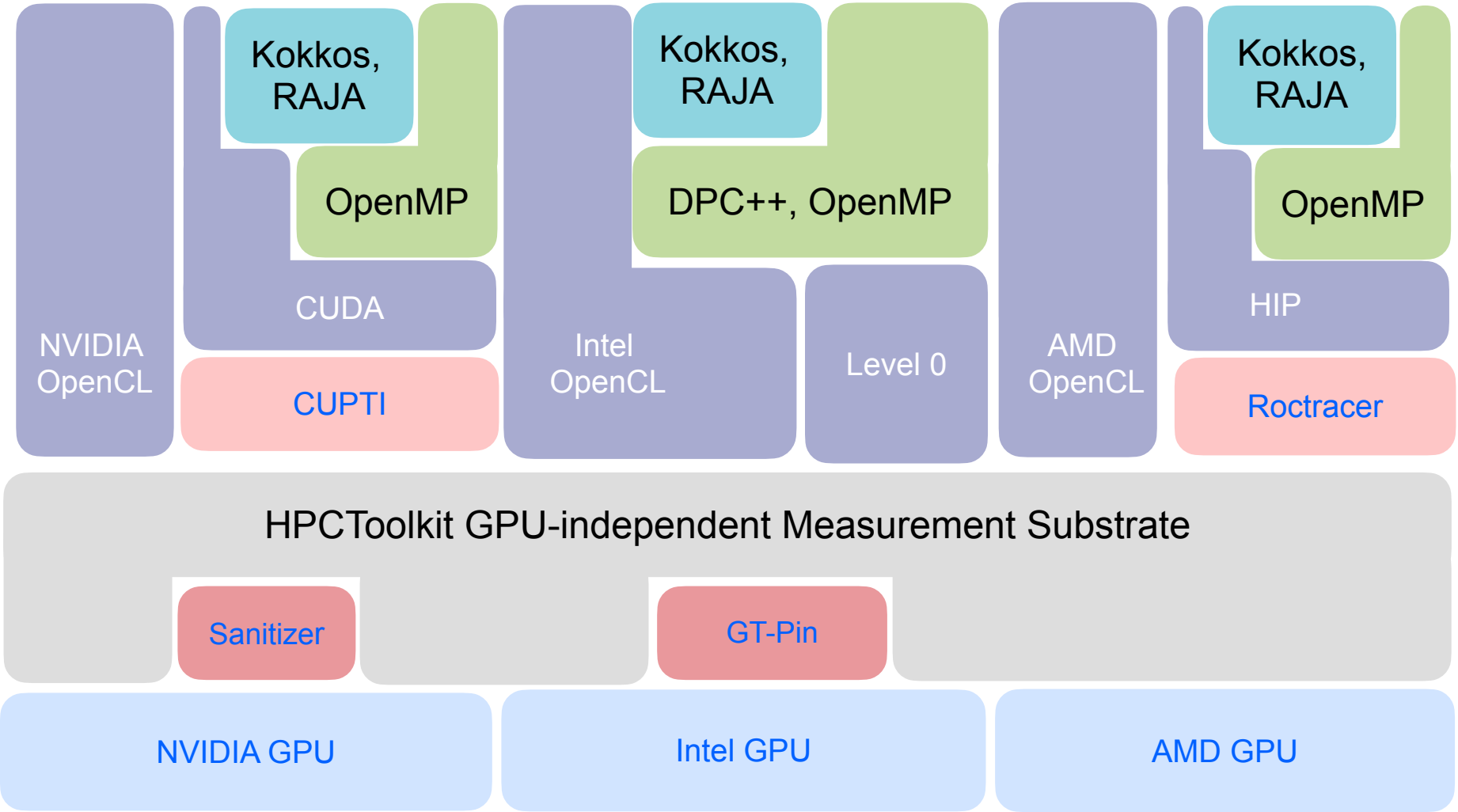
HPCToolkit's Workflow for GPU-accelerated Applications



hpcrun - Measure CPU and GPU execution

- GPU profiling
  - hpcrun -e gpu=xxx <app> …. // *xxx* ∈ *{nvidia,amd,opencl,level0}*
- GPU tracing (-t)
  - hpcrun -e gpu=yyy -t <app> // *yyy* ∈ *{nvidia,amd,opencl}*
- GPU PC sampling (NVIDIA only)
  - hpcrun -e gpu=nvidia,pc <app>
- CPU and GPU profiling and tracing
  - hpcrun -e REALTIME -e gpu=yyy -t <app>
- Use hpcrun with job launchers
  - jsrun -n 32 -g 1 -a 1 hpcrun -e gpu=*xxx* <app>
  - srun -n 1 -G 1 hpcrun -e gpu=*xxx* <app>
  - aprun -n 16 -N 8 -d 8 hpcrun -e gpu=*xxx* <app>

HPCToolkit and GPU Software Stacks



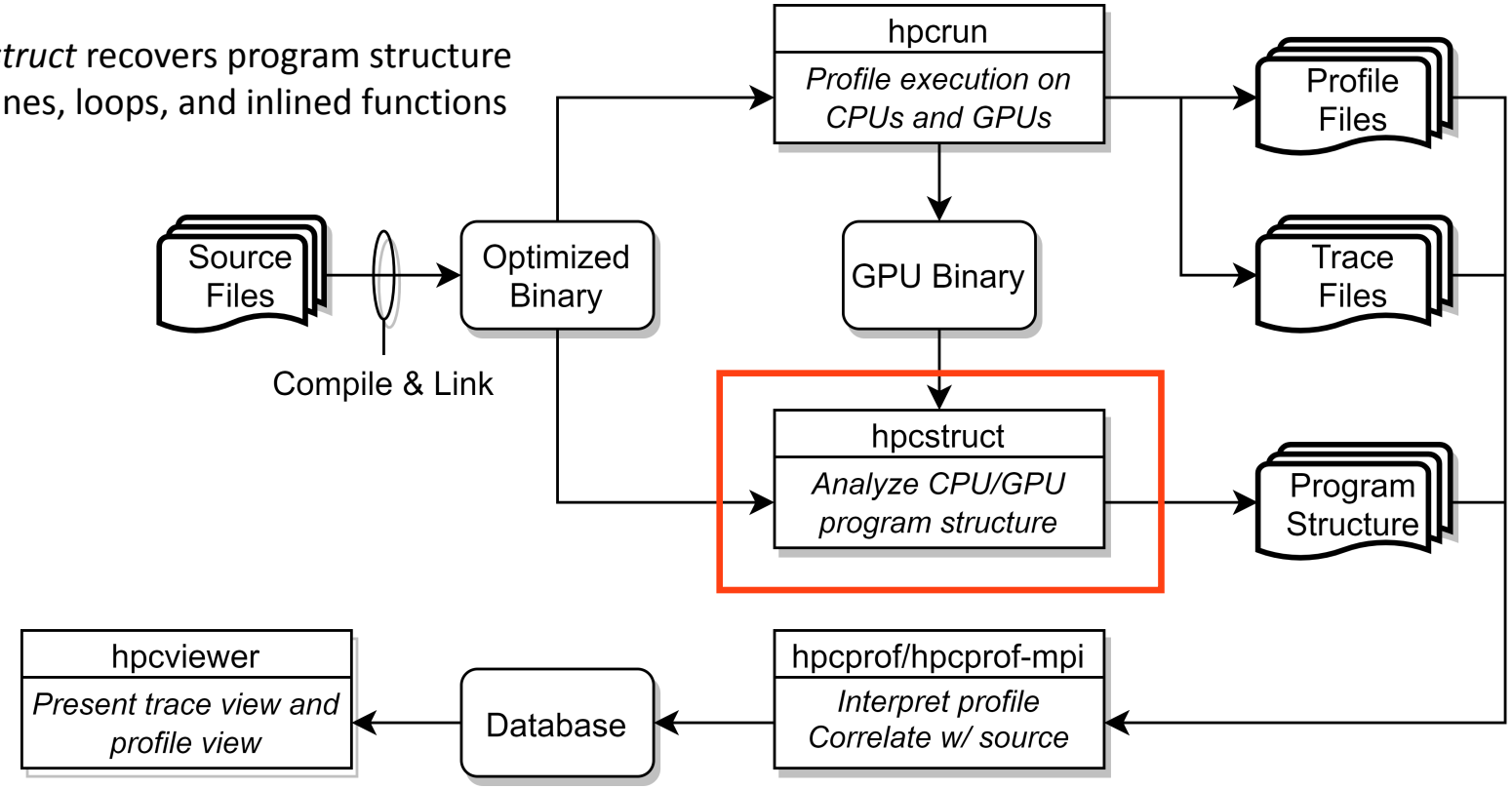
Measurement for GPU-accelerated Supercomputers

- **Measurement interfaces**
  - Hardware
    - CPU hardware performance monitoring unit
    - GPU hardware counters and PC sampling
  - Software
    - Glibc LD_AUDIT for tracking dynamic loading of shared libraries
    - Linux perf_events for kernel measurement
    - GPU monitoring and instrumentation libraries from vendors
- **Multiple measurement modalities and interfaces**
  - Sampling on the CPU
  - Callbacks when GPU operations are launched and (sometimes) completed
  - GPU event stream, including PC sampling measurements

HPCToolkit's Workflow for GPU-accelerated Applications

Step 3:

- *hpcstruct* recovers program structure about lines, loops, and inlined functions



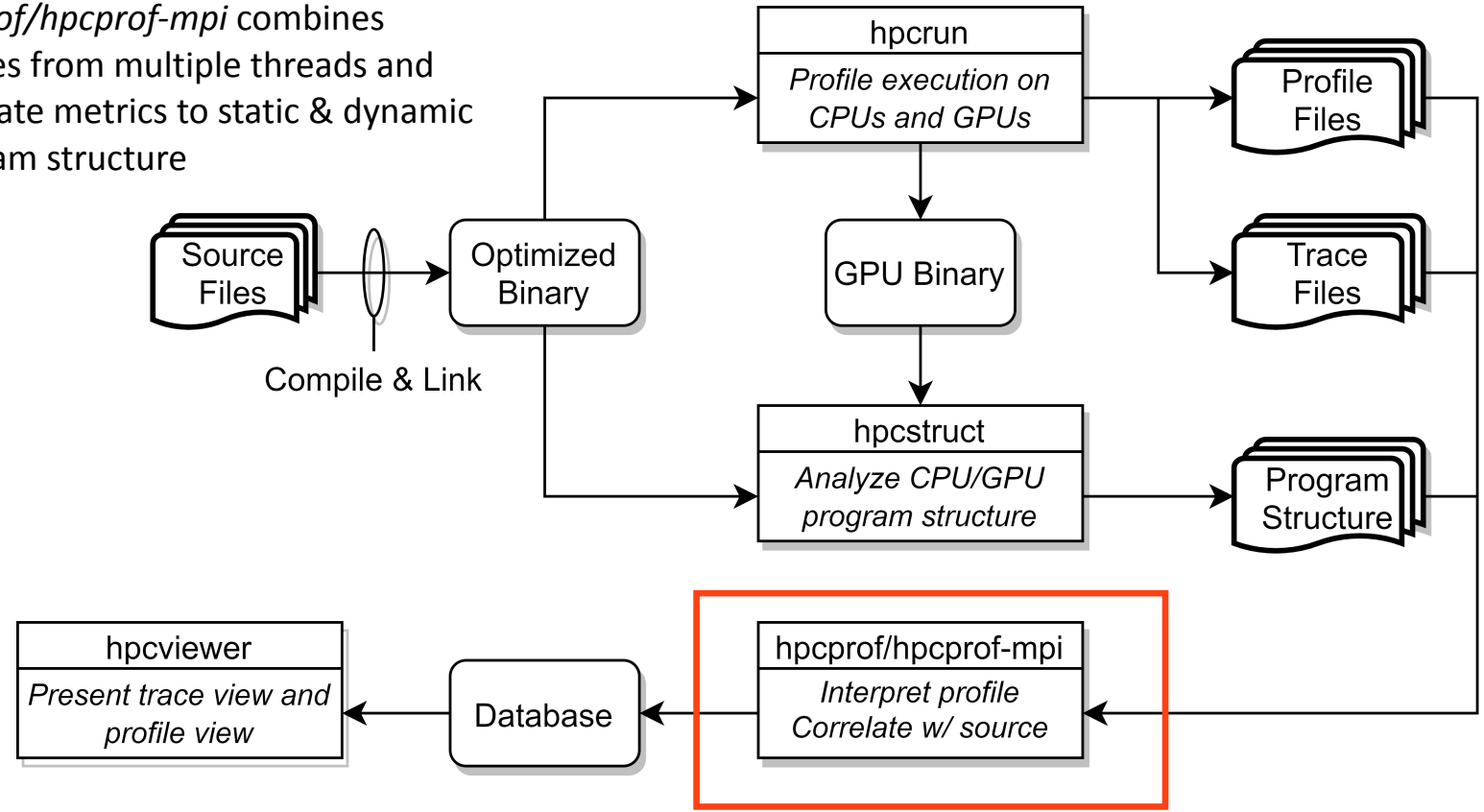
hpcstruct - Recover Structure for GPU-accelerated Programs

- Analyze all GPU binaries in <measurements-dir>
  - hpcstruct [--gpucfg yes] <measurements-dir>
    - “gpucfg yes” means recover GPU loop nests, calling context information
      - only useful on NVIDIA platforms at present: no fine-grain measurement from other vendors
      - use with care on large binaries: very costly because of NVIDIA’s lack of necessary APIs
  - adds a program structure file to the measurement directory for each GPU binary

HPCToolkit's Workflow for GPU-accelerated Applications

Step 4:

- *hpcprof/hpcprof-mpi* combines profiles from multiple threads and correlate metrics to static & dynamic program structure



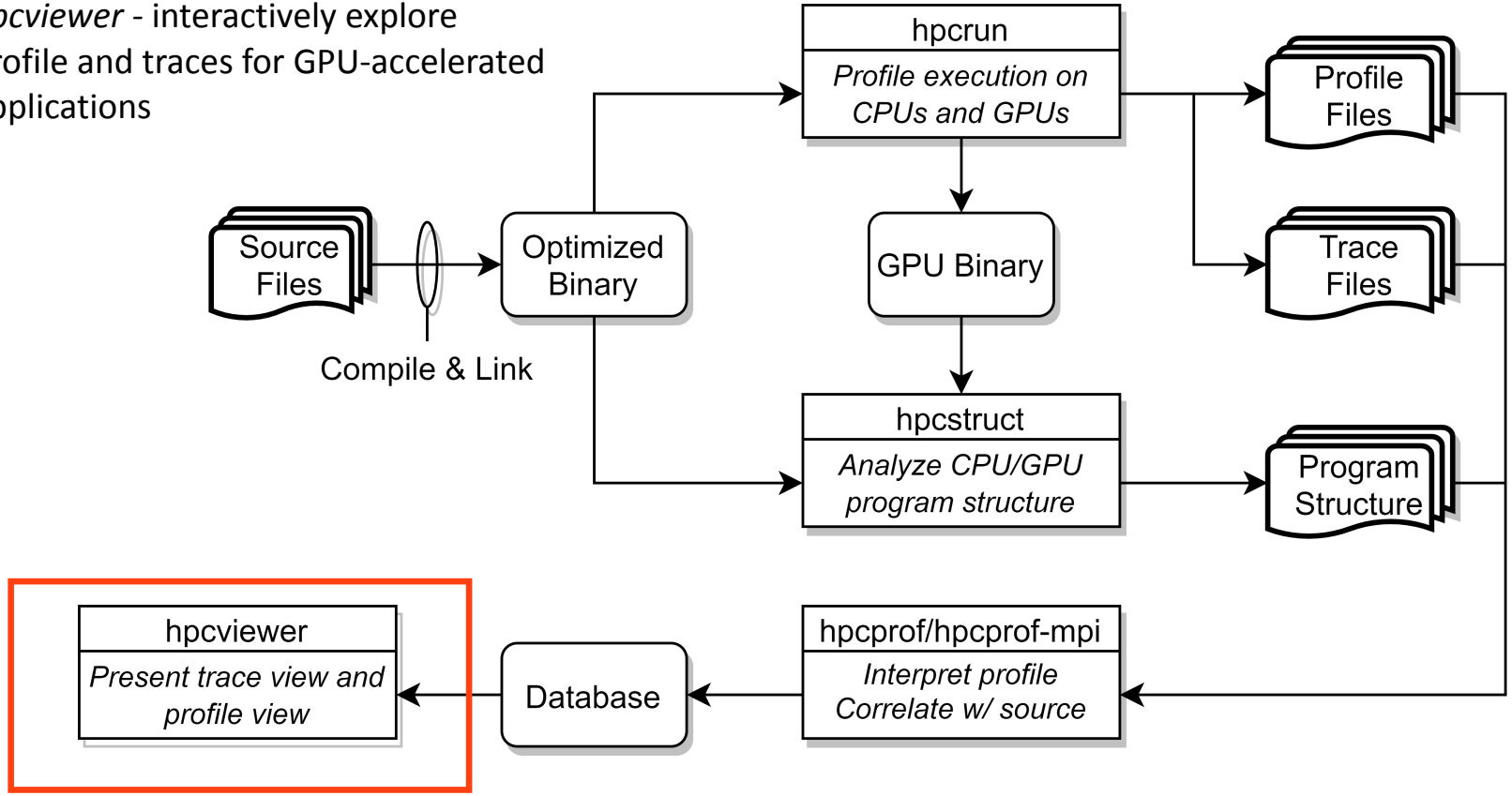
hpcprof/hpcprof-mpi - Correlate Measurements with Code

- Use a single process to combine performance data
  - hpcprof <measurements-dir>
- Use multiple processes to combine large-scale performance data
  - jsrun -n <np> hpcprof-mpi <measurements-dir>
  - srun -n <np> hpcprof-mpi <measurements-dir>

HPCToolkit's Workflow for GPU-accelerated Applications

Step 4:

- *hpcviewer* - interactively explore profile and traces for GPU-accelerated applications



GPU Monitoring Capabilities of HPCToolkit

Measurement Capability	NVIDIA	AMD	Intel
kernel launches, explicit memory copies, synchronization	callbacks + activity API	callbacks + Activity API	callbacks
instruction-level measurement and analysis	PC sampling of GPU code	Future*: PC sampling (as seen on Github) of GPU code	GTPin; Future*: instruction-level measurement of GPU code
kernel characteristics	Activity API	(available statically)	(unknown)

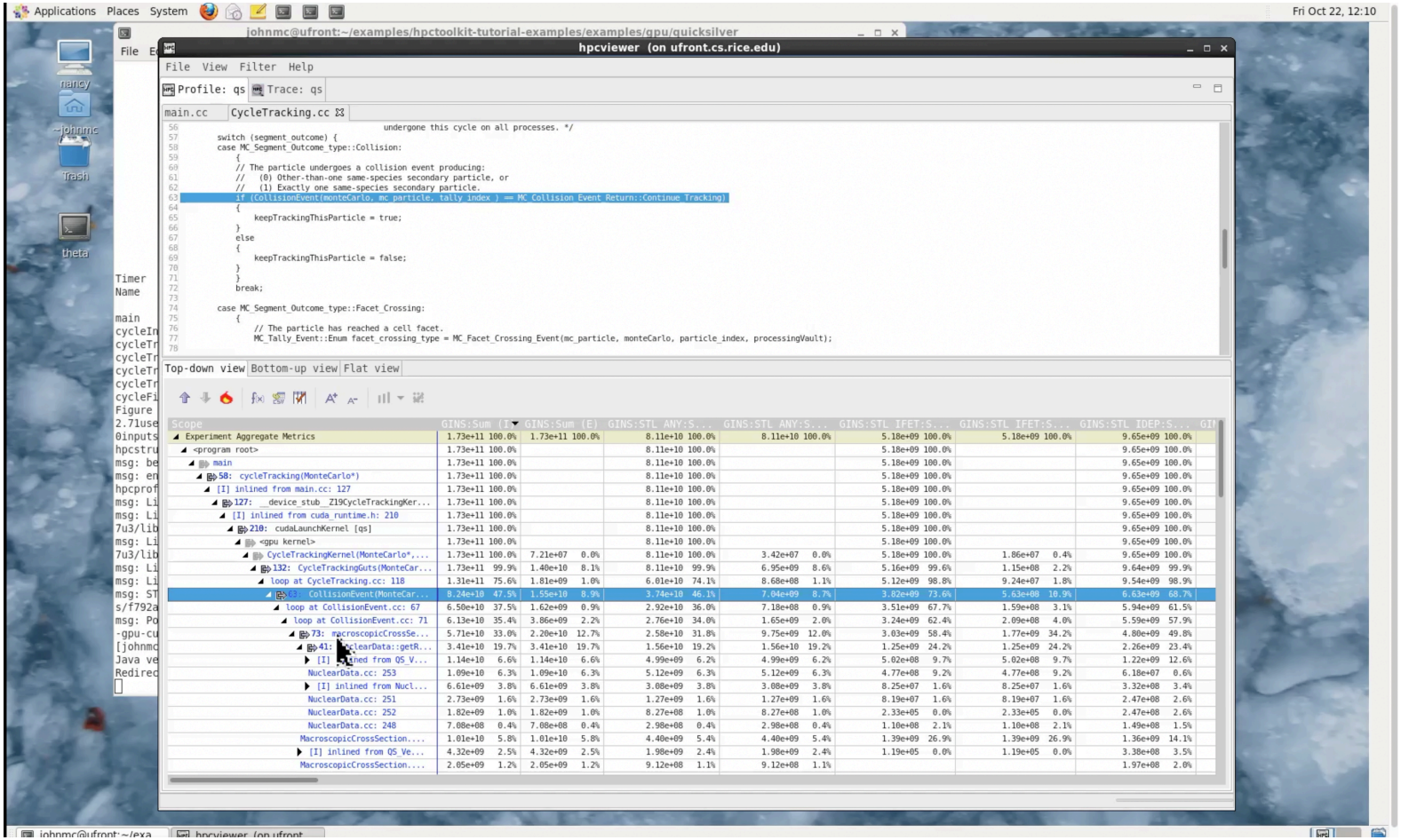
Significant support in master branch

Prototype support in master branch

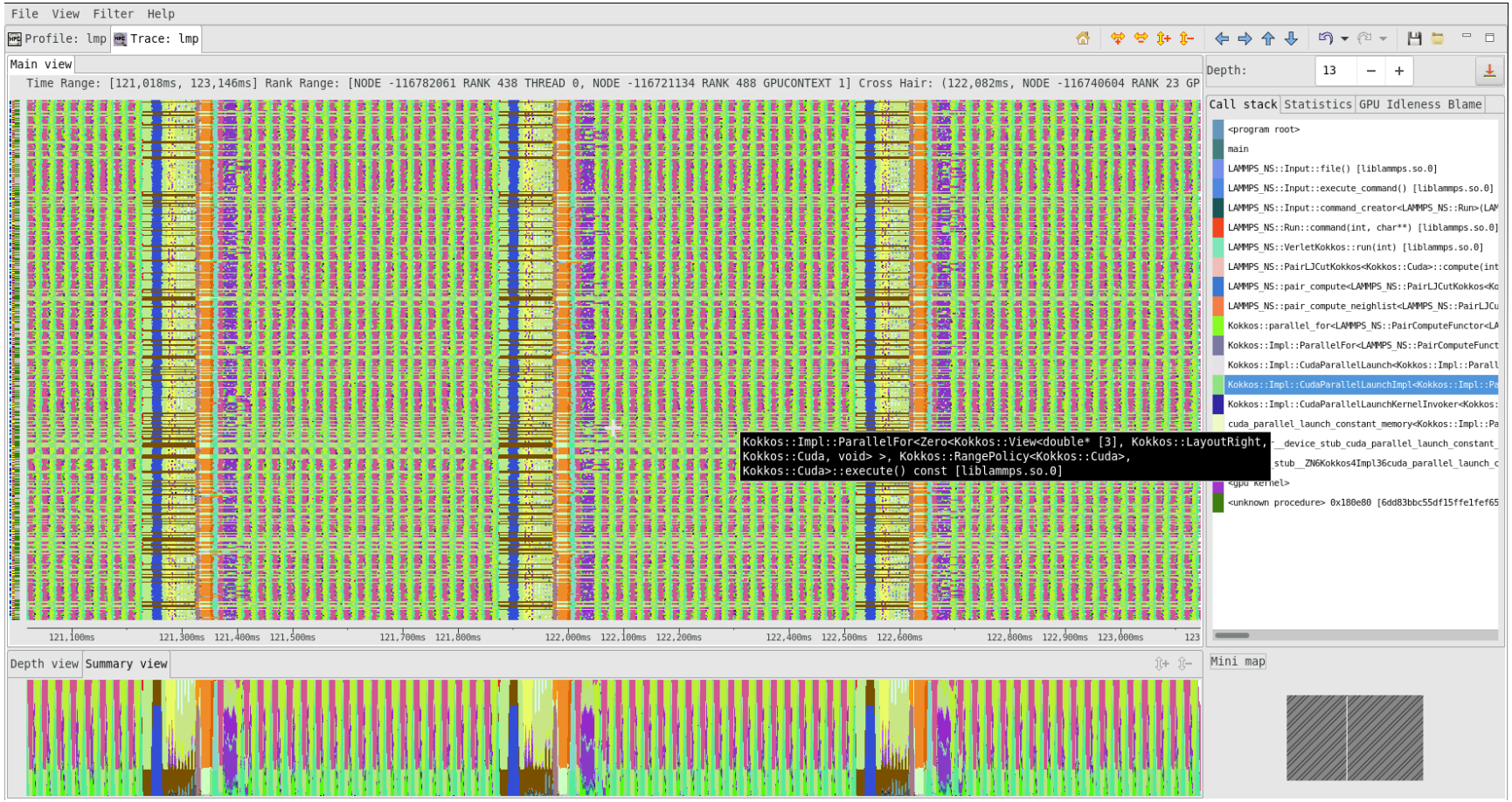
Prototype support in master branch

*Approved for public release

Demo: Measure and Analyze LLNL's Quicksilver (Video)



Emerging Work: Large-scale LAMMPS



128 nodes (AMD EPYC + 4 x NVIDIA A100), 512 MPI ranks, 512 GPU streams

Work in Progress

- **GPU Enhancements**
  - Intel GPUs
    - Measurement support for Intel GPUs using OpenCL and Level 0
    - Fine-grain measurement using GTPin
    - Fine-grain attribution using binary analysis
  - AMD GPUs
    - Binary analysis and instrumentation for fine-grain measurement and attribution
  - NVIDIA GPUs
    - New support for NVIDIA inlining info - distribute our patches to the community
    - Reduce fine-grain measurement overhead with low-overhead PC sampling (CUDA 11.3)
- **Scalability**
  - finalizing new version of hpcprof-mpi with massive threading and sparse formats
- **User interface**
  - overhaul metric view to enhance performance and scalability
  - associate trace lines with metadata (node, GPU, MPI rank, GPU stream …)
  - improve presentation of the many GPU metrics
- **Reliability and Completeness**