HPCToolkit: Experiences at Exascale

Jonathon Anderson

Rice University

STW 2023

June 19, 2023

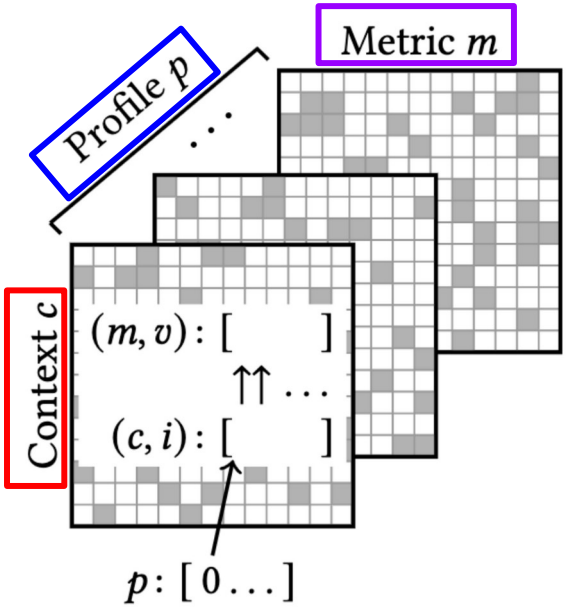
The Team

- PI: John Mellor-Crummey
- Staff
  - Mark Krentel, Laksono Adhianto, Marty Itzkowitz, Wil Phan, Matt Barnet
- Students
  - Jonathon Anderson, Yumeng Liu, Dejan Grubisic, Dragana Grbic

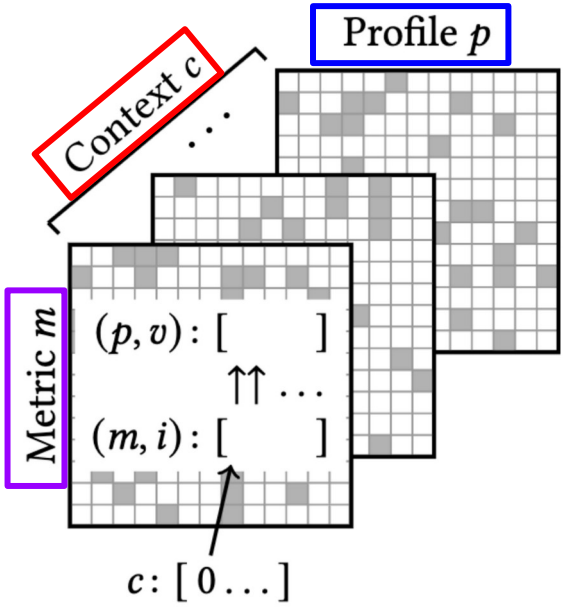
Recap: Preparing HPCToolkit for Exascale

- No other tool provides **detailed** performance analysis **at scale**!
- The problem: performance data at scale is **huge** and **slow** to process!
- Two-pronged solution, next slides

Recap: Efficient Sparse Formats for Performance Data

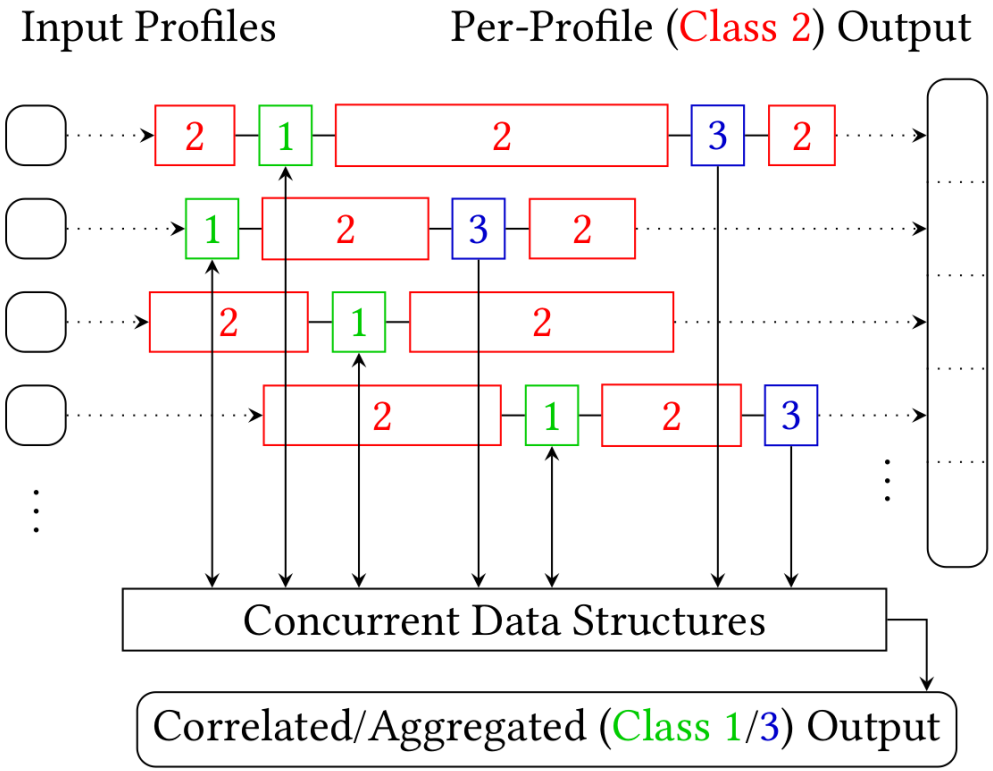


Profile-Major-Sparse (PMS):
data for each profile is contiguous



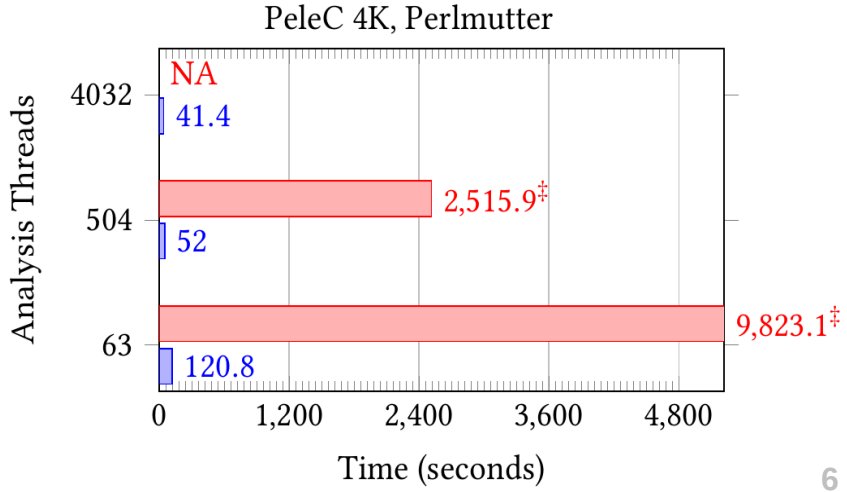
Context-Major-Sparse (CMS):
data for each context is contiguous

Recap: Highly-parallel Multithreading for Performance Analysis



Recap: Preparing HPCToolkit for Exascale

- End result: **Stellar improvements for GPU-accelerated codes!**
  - Running PeleC on 512 nodes of Perlmutter, ~17.1 PFLOPs
  - Size reduction from sparsity: 14.3 TB → **11.4 GB (1254x)**
  - Min. node reduction from threading: 16 nodes → **1 node**
  - Overall time reduction: 2.7 hr → **2 min (81.3x)**
- Also significant improvements for CPU-only cases
  - For details, see our paper in ICS'22



Last year, the only question asked was

Have you used this at Exascale?

Today, the answer to that question is

Yes, we have.

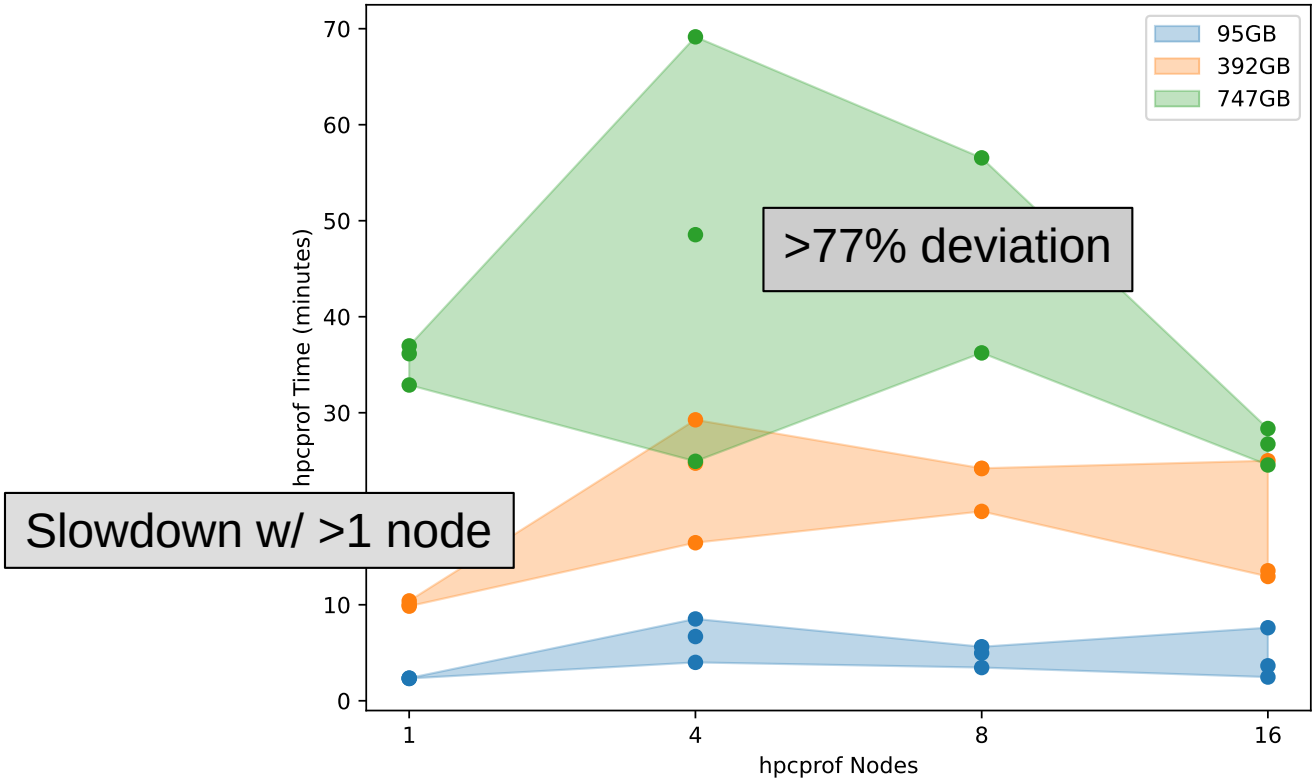
HPCToolkit Improvements for Frontier I

- HPCToolkit **built and (mostly) ran with no change**, but we made improvements for Frontier
- **hpcrun**
  - Bug fix: corrected support for Cray OpenMP
  - Improved how CPU threads are mapped to MPI ranks
    - now using job launcher variables: independent of any MPI version
  - Added support to boost resolution of CPU traces
    - collect CPU callstack when offloading GPU operations
- **hpcstruct**
  - Use OpenMP to inspect profiles to identify CPU and GPU binaries involved in the execution
    - Reduces preparation for binary analysis of an hpctoolkit database to seconds

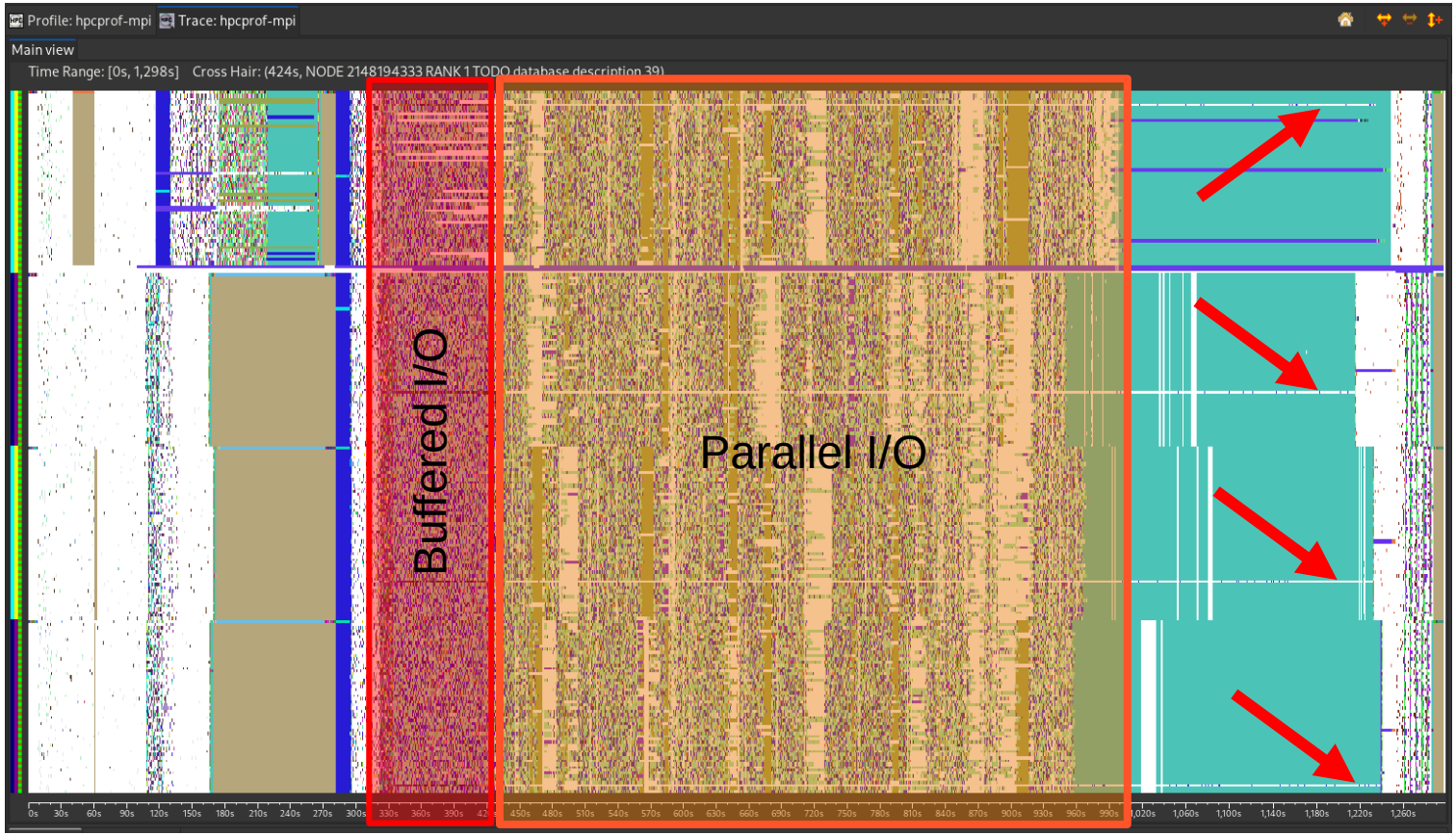
HPCToolkit Improvements for Frontier II

- **hpcprof**
  - Bug fix: handle truncated reads on network file systems
- **hpcviewer**
  - Reduced memory footprint when viewing large profiles and traces
  - Improved performance and usability of graphing metrics for many execution contexts
  - Corrected handling of corner cases at trace start and end

Remaining Issue: Poor Multi-node Performance of hpcprof



Remaining Issue: Serialized I/O



First Taste of Exascale –

First Taste of Exascale – False Start

- Large-scale jobs on Frontier **regularly fail**, and **fail immediately**
  - MPI bootstrap barrier times out after 2 minutes and fails the job
  - Unrelated to node crashes (which happen less immediately)
- OLCF Help: “Orion can’t serve libraries + executable for many nodes under 2 minutes”
- Recommended solution: copy all libraries to node-local NVMe first
  - User Guide “example” batch script is 60 lines long
  - Uses LD_LIBRARY_PATH, conflicts with Spack’s RPATH
- Spark discussion for alternative solutions. Spindle?
- Data on following slides collected in April, failure occurs **much more frequently** since

First Taste of Exascale – Leviathan

- Strong-scaling LAMMPS, **large** problem, **short** simulation
  - Up to 8192 nodes, ~1.6 EFLOPs (theoretical peak)
  - Lennard-Jones, up to 1.77 trillion atoms (216M atoms/node)
  - Runs for 7900 timesteps, 18.4 minutes (under measurement)
- Analysis
  - Final size: **0.793 TB**
  - Analysis time, single node: **36 minutes**

First Taste of Exascale – Smaug

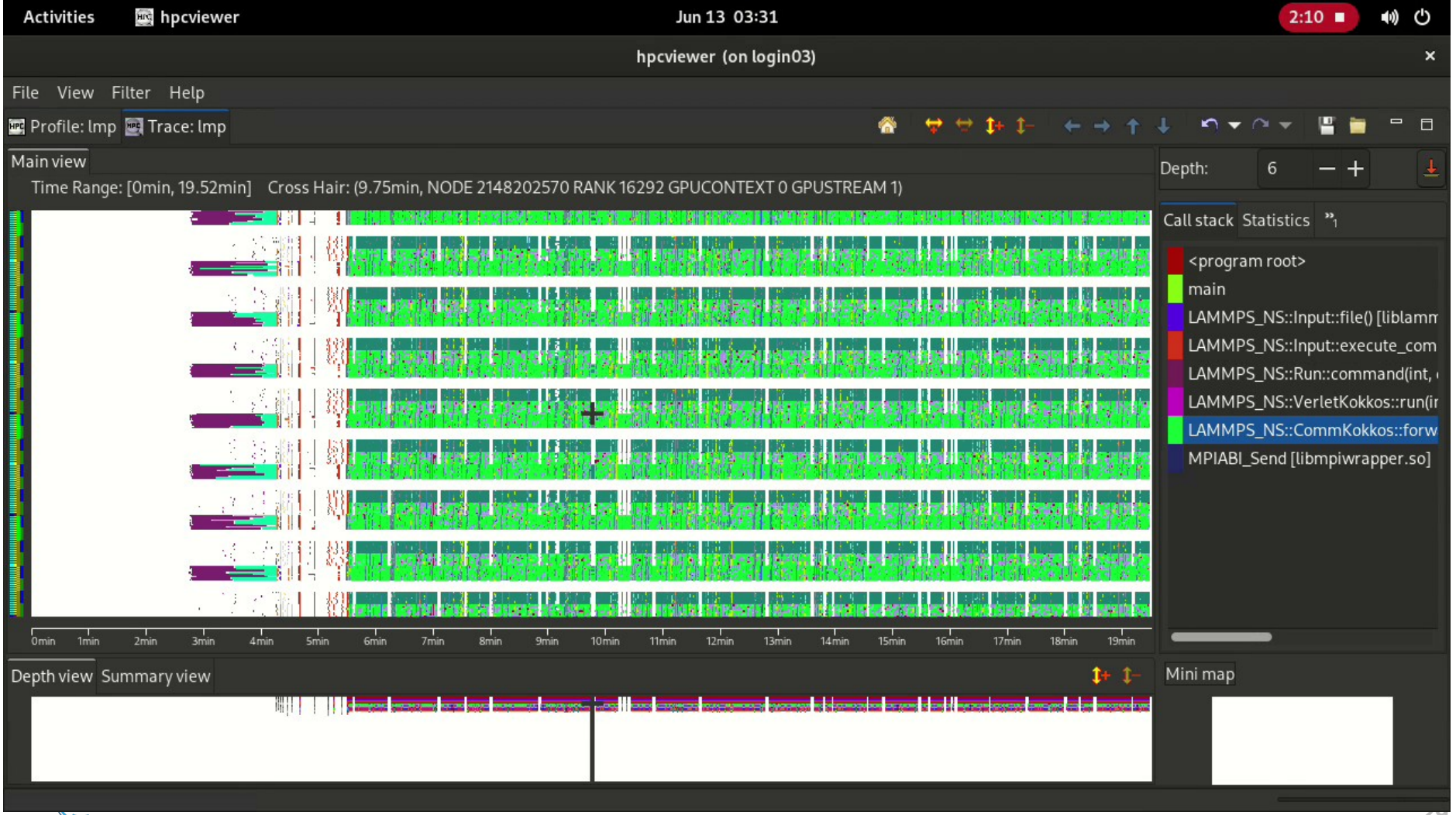
- Strong-scaling LAMMPS, **small** problem, **long** simulation
  - Up to 8192 nodes, ~1.6 EFLOPs (theoretical peak)
  - Lennard-Jones, up to 34 billion atoms (4.1M atoms/node)
  - Runs for 110 thousand timesteps, 19.7 minutes (under measurement)
  - Boosted CPU trace resolution, approximates Leviathan for 4.3 hours
- Analysis
  - Final size: **5.68 TB**
  - Analysis time, single node: **1 hour, 40 minutes**

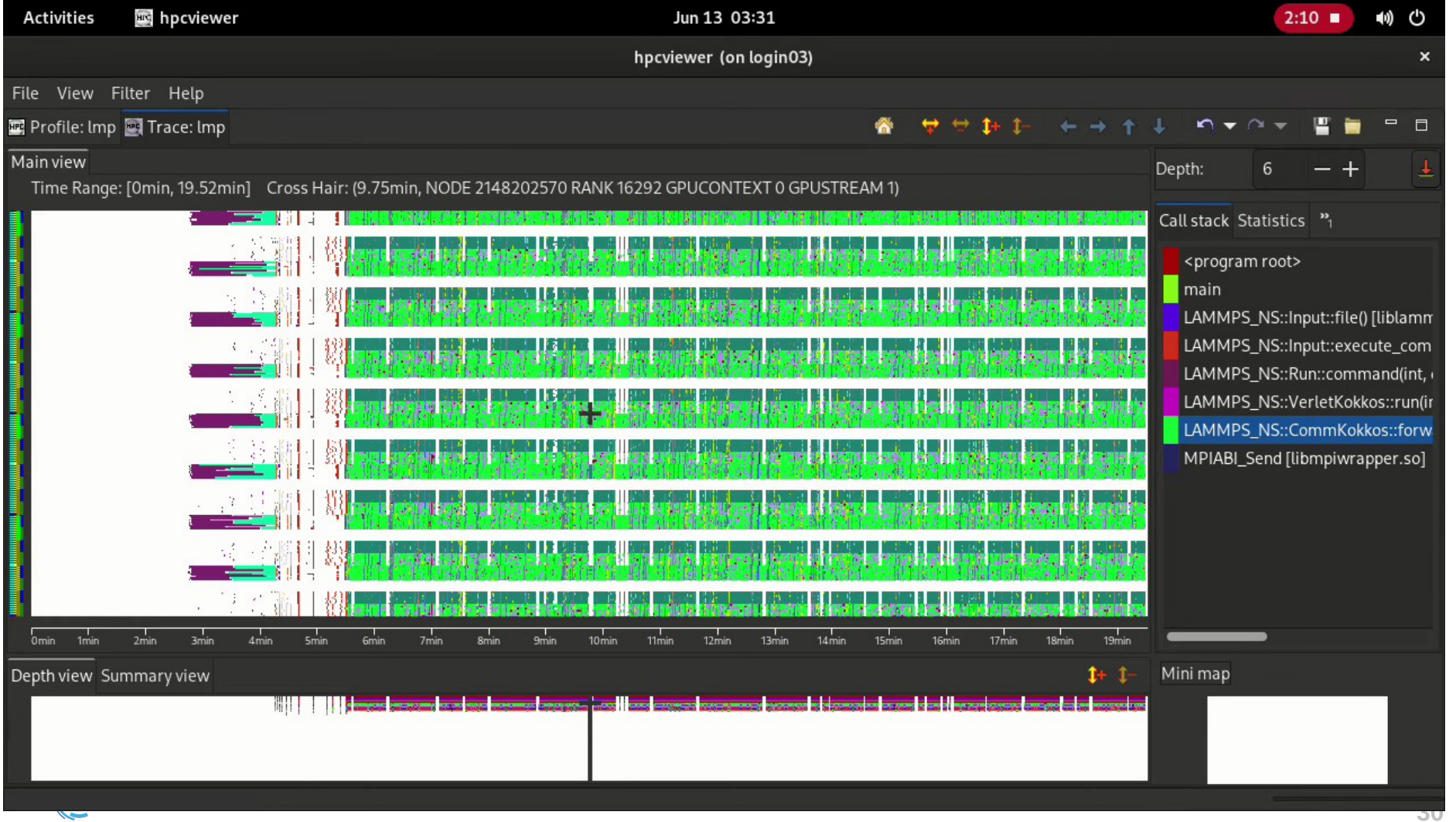
Activities hpcviewer Jun 13 03:29 0:00

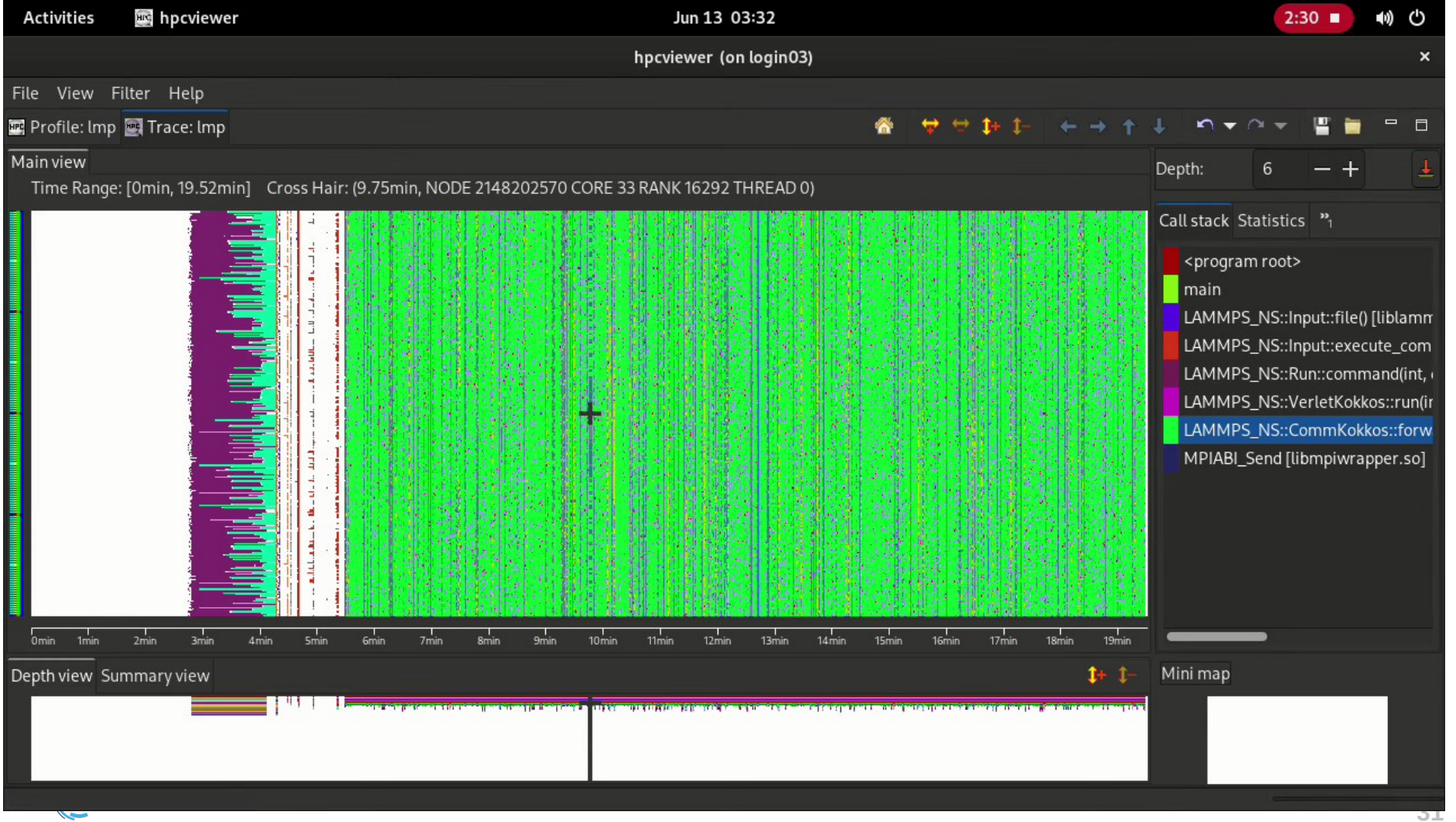
hpcviewer (on login03)

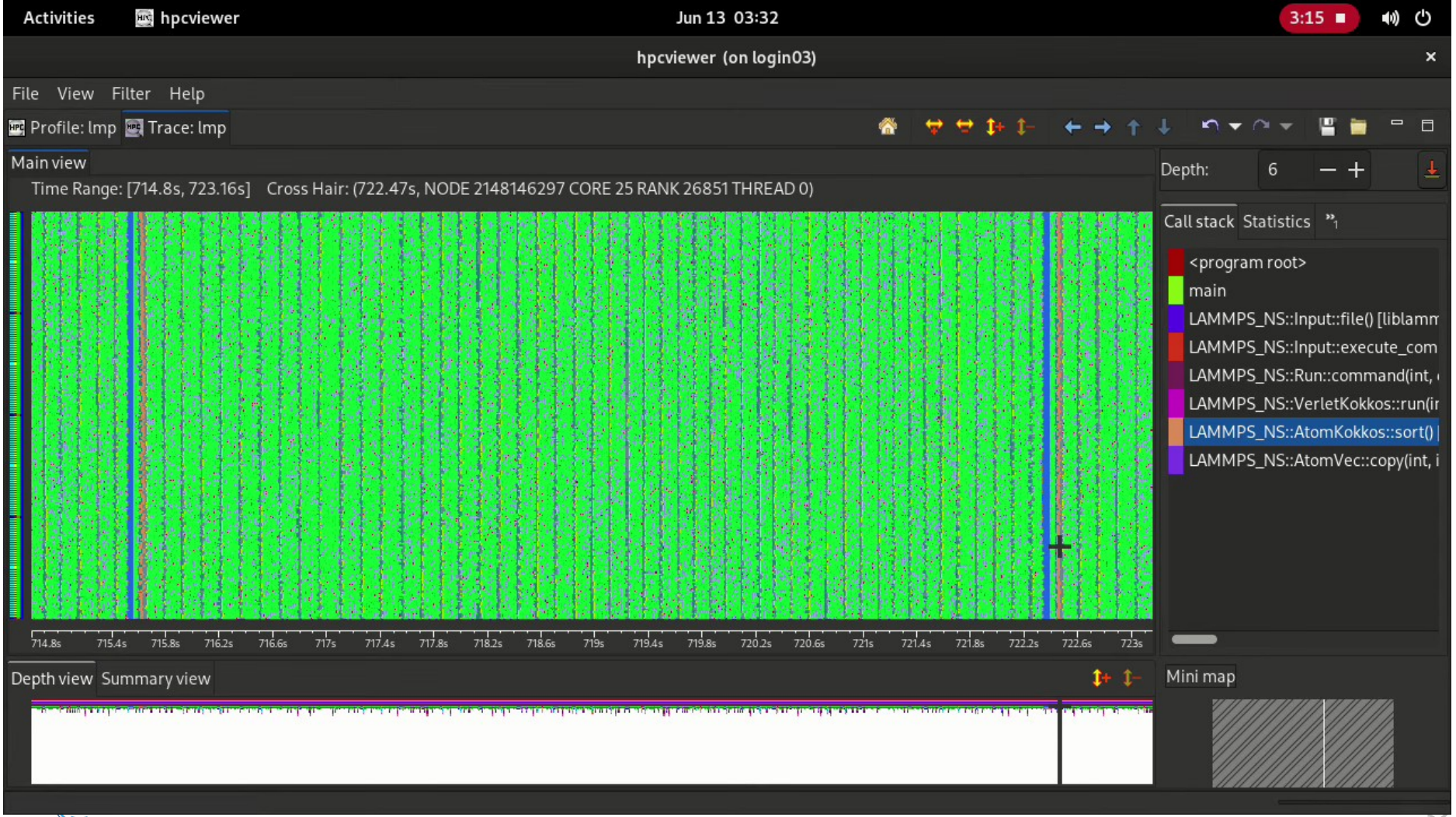
File View Filter Help

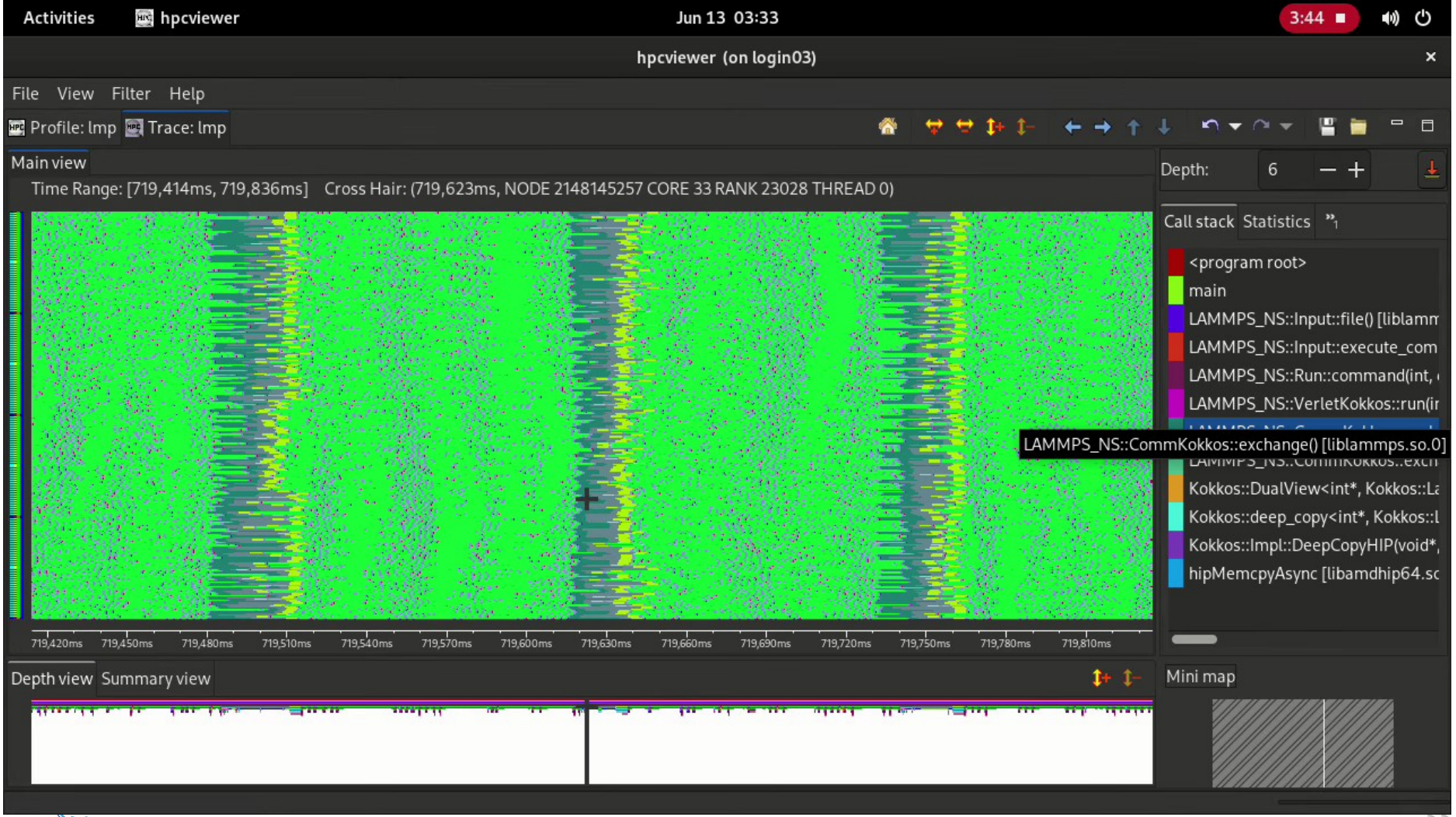
- Smaug configuration, 4096 nodes
  - Later run, after hpcrun improvements
  - Total size: 2.85 TB
- X11 over SSH tunnel from Frontier
  - ssh -Y frontier.olcf.ornl.gov
- Rendering on a virtual desktop at Rice
- Hpcviewer 2023.05

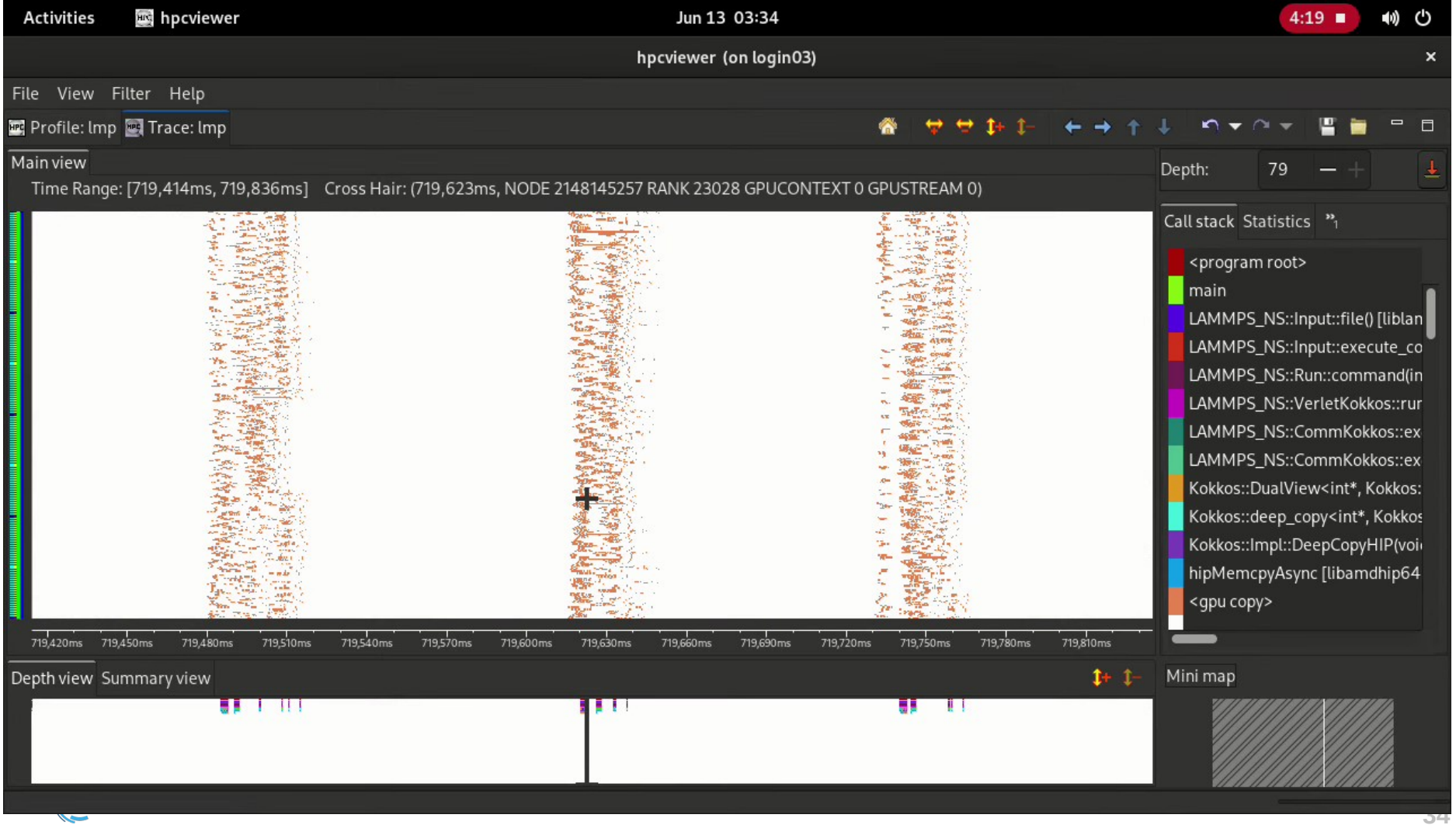

Activities
hpcviewer
Jun 13 03:31
2:10
hpcviewer (on login03)
File View Filter Help
Profile: lmp
Trace: lmp
Main view
Time Range: [0min, 19.52min] Cross Hair: (9.75min, NODE 2148202570 RANK 16292 GPUCONTEXT 0 GPUSTREAM 1)
0min 1min 2min 3min 4min 5min 6min 7min 8min 9min 10min 11min 12min 13min 14min 15min 16min 17min 18min 19min
Depth view
Summary view
Depth: 6
Call stack
Statistics
<program root>
main
LAMMPS_NS::Input::file() [liblamm
LAMMPS_NS::Input::execute_com
LAMMPS_NS::Run::command(int,
LAMMPS_NS::VerletKokkos::run(ir
LAMMPS_NS::CommKokkos::forw
MPIABI_Send [libmpiwrapper.so]
Mini map

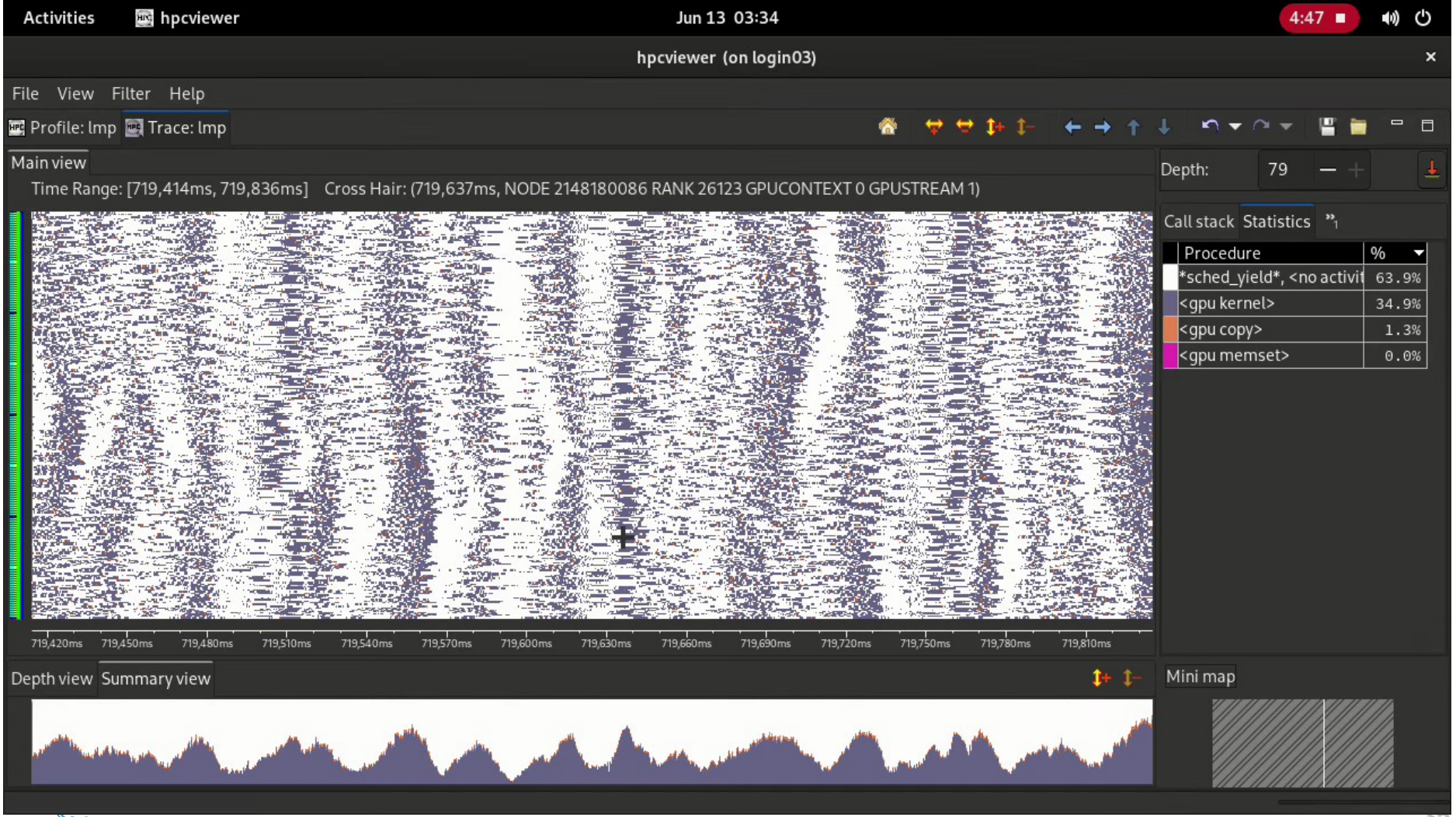

Activities
hpcviewer
Jun 13 03:31
2:10
hpcviewer (on login03)
File View Filter Help
Profile: lmp
Trace: lmp
Main view
Time Range: [0min, 19.52min] Cross Hair: (9.75min, NODE 2148202570 RANK 16292 GPUCONTEXT 0 GPUSTREAM 1)
0min 1min 2min 3min 4min 5min 6min 7min 8min 9min 10min 11min 12min 13min 14min 15min 16min 17min 18min 19min
Depth: 6
Call stack
Statistics
<program root>
main
LAMMPS_NS::Input::file() [liblamm
LAMMPS_NS::Input::execute_com
LAMMPS_NS::Run::command(int,
LAMMPS_NS::VerletKokkos::run(ir
LAMMPS_NS::CommKokkos::forw
MPIABI_Send [libmpiwrapper.so]
Depth view
Summary view
Mini map


Activities
hpcviewer
Jun 13 03:32
2:30
hpcviewer (on login03)
File View Filter Help
Profile: lmp
Trace: lmp
Main view
Time Range: [0min, 19.52min] Cross Hair: (9.75min, NODE 2148202570 CORE 33 RANK 16292 THREAD 0)
0min 1min 2min 3min 4min 5min 6min 7min 8min 9min 10min 11min 12min 13min 14min 15min 16min 17min 18min 19min
Depth: 6
Call stack
Statistics
<program root>
main
LAMMPS_NS::Input::file() [liblamm
LAMMPS_NS::Input::execute_com
LAMMPS_NS::Run::command(int,
LAMMPS_NS::VerletKokkos::run(ir
LAMMPS_NS::CommKokkos::forw
MPIABI_Send [libmpiwrapper.so]
Depth view
Summary view
Mini map


Activities
hpcviewer
Jun 13 03:32
3:15
hpcviewer (on login03)
File View Filter Help
Profile: lmp
Trace: lmp
Main view
Time Range: [714.8s, 723.16s] Cross Hair: (722.47s, NODE 2148146297 CORE 25 RANK 26851 THREAD 0)
714.8s 715.4s 715.8s 716.2s 716.6s 717s 717.4s 717.8s 718.2s 718.6s 719s 719.4s 719.8s 720.2s 720.6s 721s 721.4s 721.8s 722.2s 722.6s 723s
Depth: 6
Call stack
Statistics
<program root>
main
LAMMPS_NS::Input::file() [liblamm
LAMMPS_NS::Input::execute_com
LAMMPS_NS::Run::command(int,
LAMMPS_NS::VerletKokkos::run(ir
LAMMPS_NS::AtomKokkos::sort()
LAMMPS_NS::AtomVec::copy(int, i
Depth view
Summary view
Mini map


Activities
hpcviewer
Jun 13 03:33
3:44
hpcviewer (on login03)
File View Filter Help
Profile: lmp
Trace: lmp
Main view
Time Range: [719,414ms, 719,836ms] Cross Hair: (719,623ms, NODE 2148145257 CORE 33 RANK 23028 THREAD 0)
719,420ms 719,450ms 719,480ms 719,510ms 719,540ms 719,570ms 719,600ms 719,630ms 719,660ms 719,690ms 719,720ms 719,750ms 719,780ms 719,810ms
Depth view
Summary view
Depth: 6
Call stack
Statistics
<program root>
main
LAMMPS_NS::Input::file() [liblamm
LAMMPS_NS::Input::execute_com
LAMMPS_NS::Run::command(int,
LAMMPS_NS::VerletKokkos::run(ir
LAMMPS_NS::CommKokkos::exchange() [liblammps.so.0]
Kokkos::DualView<int*, Kokkos::La
Kokkos::deep_copy<int*, Kokkos::l
Kokkos::Impl::DeepCopyHIP(void*,
hipMemcpyAsync [libamdhip64.so
Mini map


Activities
hpcviewer
Jun 13 03:34
4:19
hpcviewer (on login03)
File View Filter Help
Profile: lmp
Trace: lmp
Main view
Time Range: [719,414ms, 719,836ms] Cross Hair: (719,623ms, NODE 2148145257 RANK 23028 GPUCONTEXT 0 GPUSTREAM 0)
719,420ms 719,450ms 719,480ms 719,510ms 719,540ms 719,570ms 719,600ms 719,630ms 719,660ms 719,690ms 719,720ms 719,750ms 719,780ms 719,810ms
Depth: 79
Call stack
Statistics
<program root>
main
LAMMPS_NS::Input::file() [liblan
LAMMPS_NS::Input::execute_co
LAMMPS_NS::Run::command(in
LAMMPS_NS::VerletKokkos::rur
LAMMPS_NS::CommKokkos::ex
LAMMPS_NS::CommKokkos::ex
Kokkos::DualView<int*, Kokkos:
Kokkos::deep_copy<int*, Kokkos
Kokkos::Impl::DeepCopyHIP(voi
hipMemcpyAsync [libamdhip64
<gpu copy>
Depth view
Summary view
Mini map


Activities
hpcviewer
Jun 13 03:34
4:47
hpcviewer (on login03)
File View Filter Help
Profile: lmp
Trace: lmp
Main view
Time Range: [719,414ms, 719,836ms] Cross Hair: (719,637ms, NODE 2148180086 RANK 26123 GPUCONTEXT 0 GPUSTREAM 1)
719,420ms 719,450ms 719,480ms 719,510ms 719,540ms 719,570ms 719,600ms 719,630ms 719,660ms 719,690ms 719,720ms 719,750ms 719,780ms 719,810ms
Depth: 79
Call stack
Statistics
Procedure
%
sched_yield, <no activit
63.9%
<gpu kernel>
34.9%
<gpu copy>
1.3%
<gpu memset>
0.0%
Depth view
Summary view
Mini map

Work in Progress

- Eliminating the I/O bottleneck in hpcprof
- Building a new remote hpcserver, companion to hpcviewer
  - Will load up the entire database in the memory of several compute nodes
  - Serves from memory, directly to an hpcviewer client on your laptop
- Collaborating with AMD on a new ROCm tools interface
  - Awaiting support for PC samples for non-root users
- Developing a Python API and library for performance data analysis
  - Supports automated and exploratory analysis of large-scale performance data
  - To be used for regression testing and validation
- Testing and tuning HPCToolkit on Sunspot (TDS for Aurora)