Monitoring and visualization of metric traces on GPU accelerated applications

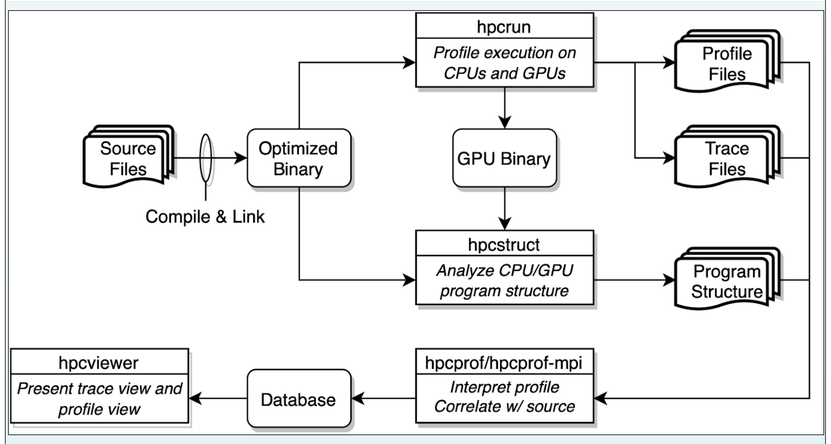
Dejan Grubisic[1], Mentor (Primary Investigator) Brian Austin[2]
[1]Rice University, [2]Lawrence Berkeley National Laboratory

Abstract

HPCToolkit supports measurement and analysis of large-scale GPU-accelerated applications. Besides collecting performance metrics about GPU kernels and data movement, it is important to monitor system wide data like power consumption, temperature of devices and their utilization. These metrics can suggest which kernels are the most power intensive, whether there are problems with cooling system, and how much of GPUs stays idle during the computation.

We added support for monitoring system level GPU metrics into HPCToolkit using the Nvidia NVML library. The profiling overhead was around 3.3 percent.

HPCToolkit Infrastructure



hpcrun:
•Collects profiling data and associate it to the source-code
•Creates:
- **Profile files**: metrics associated with calling context trees
- **Trace files**: timeline of CPU threads and GPU streams
- **Metric-trace files:** timeline of system-wide metrics

hpcstruct:
•Recovers **code structure** of CPU/GPU binaries with Dyninst (call sites, inlined functions, loop nests, source lines, etc.)

hpcprof and **hpcprof-mpi**:
•Reconstructs global calling context tree, and associates measurements back to CPU and GPU source code
•Create database suitable for visualization.

hpcviewer:
•Presents a heterogenous calling-context tree that spans both CPU and GPU contexts, annotated with metrics.
•Visualizes trace data and provides tools for statistical analysis

Metric Trace

We ran HPCToolkit with Laghos (LAGrangian High-Order Solver) on the Cori supercomputer on 8 GPUs (two nodes). We collected and summarized data power consumption and temperature data across all GPUs to produce a traceview of the execution.

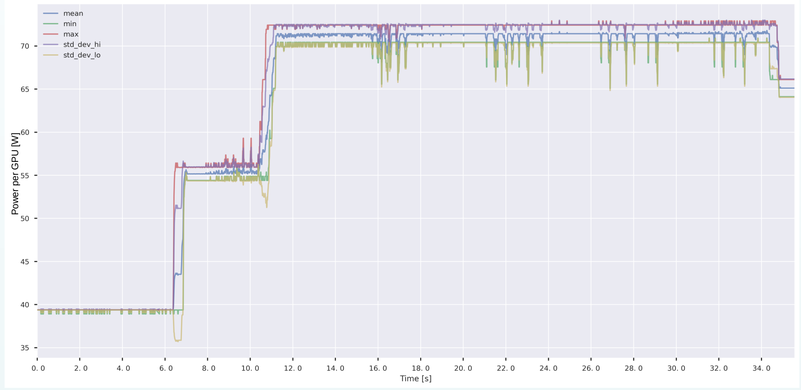


Figure 1: Power consumption for the Laghos application

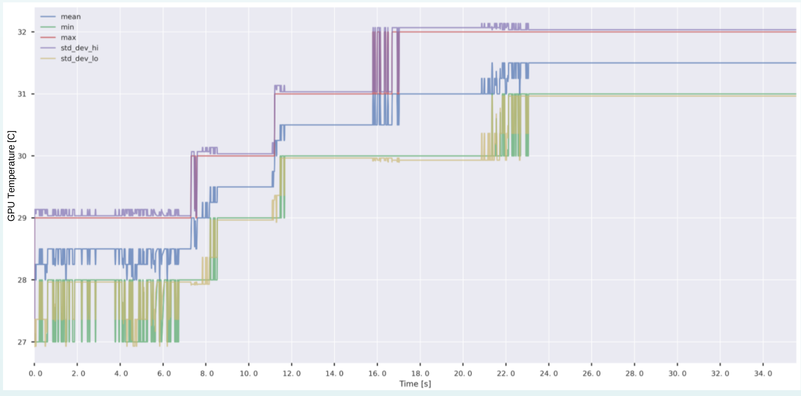


Figure 2: Temperature for the Laghos application

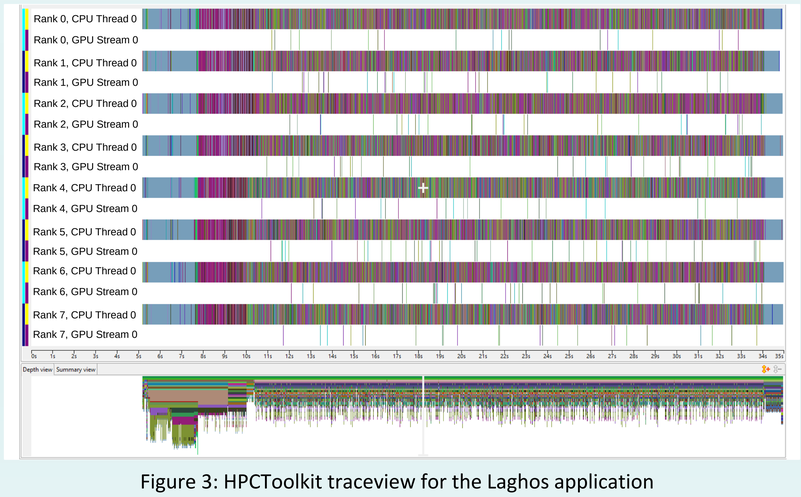


Figure 3: HPCToolkit traceview for the Laghos application

Findings:
- Power consumption and temperature are correlated to specific phases of the execution (transition points 5s, 10s)
- Some GPUs have sudden drops of power consumption

Use-case

- We run vector add application with 40 CPU threads and 40 GPU streams on 8 GPUs (two nodes), and we monitored the utilization of each GPU. Thanks to minimum value trace line, we noticed that one of them had a tiny drop of the utilization for 141ms.

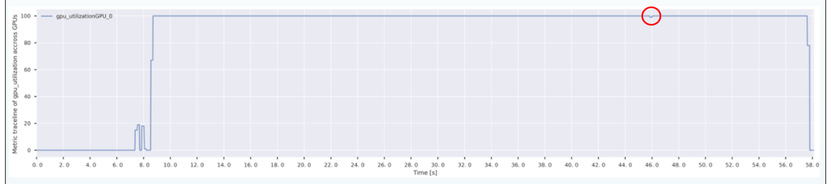


Figure 4: HPCToolkit utilization traceline of the problematic GPU

- By examining trace view for the problematic GPU, we saw that the reason for utilization drop was insufficient kernel offloading marked on the picture below

Figure 4: HPCToolkit traceline analysis of the problematic GPU

Conclusion

- ❖ HPCToolkit is able to monitor system wide metrics during the program execution.
- ❖ Visualization of metric traces from multiple devices through statistical data (min, max, mean, standard deviation) is an effective way of giving insight into the larger picture
- ❖ By measuring the GPU utilization and applying HPCToolkit traceview analysis we were able to locate and identify the cause of performance drop

Acknowledgements

Thanks to John Mellor-Crummey and Jonathon Anderson for guidance and support.
This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231.

Project repository: https://github.com/hpctoolkit/hpctoolkit Project web-page: http://hpctoolkit.org User-group mailing list: hpctoolkit-forum@rice.edu