Understanding GPU Utilization Using LDMS Data on Perlmutter

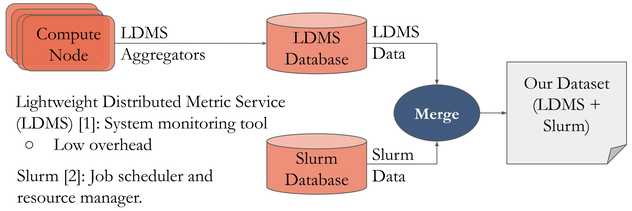
Onur Cankur[1], Brian Austin[2], Abhinav Bhatele[1]

[1]Department of Computer Science, University of Maryland [2]NERSC, Lawrence Berkeley National Laboratory

Abstract

GPGPU-based clusters and supercomputers have grown significantly in popularity over the past decade. While numerous GPGPU hardware counters are available to users, their potential for workload characterization remains underexplored. In this work, we analyze previously overlooked GPU hardware counters collected via the Lightweight Distributed Metric Service (LDMS) on Perlmutter. We examine spatial imbalance, defined as uneven GPU usage within the same job, and perform a temporal analysis of how counter values change during execution. Using temporal imbalance, we capture deviations from average usage over time. Our findings reveal inefficiencies and imbalances that can guide workload optimization and inform future HPC system design.

Data Gathering and Preparation



- Lightweight Distributed Metric Service (LDMS) [1]: System monitoring tool
  - Low overhead
- Slurm [2]: Job scheduler and resource manager.

Monitoring Data Used

- We retrieved performance counter measurements collected by using the Data Center GPU Manager (DCGM) [3], configured with a 10-second sampling rate.
- We retrieved ~4 months of data, spanning August 16 to December 13, 2023.
- Our dataset includes information about 345,154 jobs after cleaning and preprocessing.

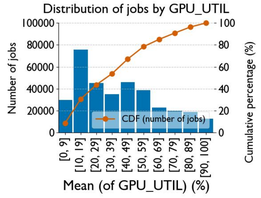
Counter Name	Short Name	Description
DCGM_FI_DEV_GPU_UTIL	GPU_UTIL	The fraction of time during which at least one kernel was executing on the GPU.
DCGM_FI_PROF_PIPE_FP{16/32/64/TENSR}_ACTIVE	FP{16/32/64/TENSR}_ACTV	The fraction of cycles the FP{16/32/64/Tensor} cores were active.

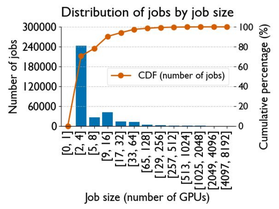
Overview of Data

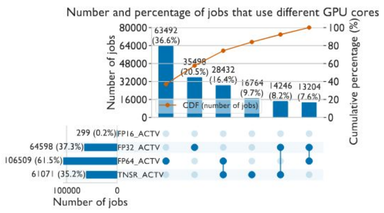
Question 1: How does overall GPU utilization and job count vary with job size (number of GPUs allocated)?

Question 2: Are there differences in how GPU jobs utilize FP16, FP32, FP64, and Tensor cores?

$$\textbf{Job-level Mean} = \frac{\text{Sum of all hardware counter values}}{\text{Number of GPUs} \times \text{Number of time points}}$$







43% of jobs run at a mean GPU_UTIL of less than 30%.

Most jobs use only a single GPU node.

Over 60% of jobs use FP64 exclusively. Tensor operations co-occur with both FP32 and FP64.

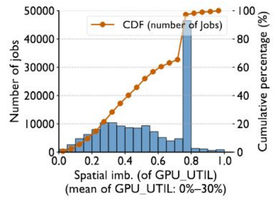
Analyzing the Spatial Behavior of Jobs

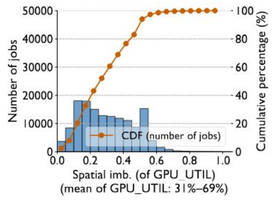
Question 3: Do jobs that allocate multiple GPUs use them evenly, or are some GPUs heavily loaded while others stay idle?

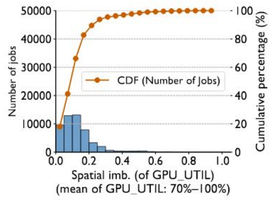
$$SI(job, window) = 1 - \frac{\text{Sum of counter values for all GPUs in the time window}}{\text{Highest per-GPU total in that window} \times \text{Number of GPUs}}$$

Measures how evenly a job spreads work across its GPUs.

Histogram and CDF of spatial imbalance of GPU_UTIL







The jobs that have higher overall usage have less spatial imbalance.

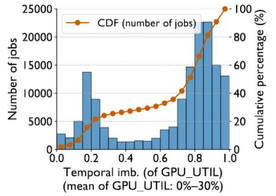
Analyzing the Temporal Behavior of Jobs

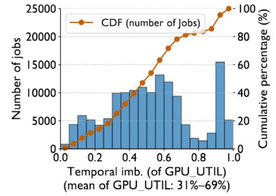
Question 4: Are GPUs used consistently over time within a single job, or do we see bursts of heavy usage and idle periods?

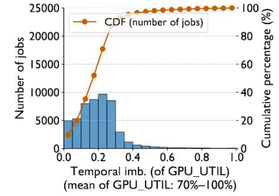
$$TI(job, gpu) = 1 - \frac{\text{Sum of counter values of the GPU over time}}{\text{Peak counter value} \times \text{Number of time points}}$$

Quantifies how evenly a hardware counter is used during a job's runtime.

Histogram and CDF of temporal imbalance of GPU_UTIL







Jobs with low mean of GPU_UTIL tend to be either consistently idle/underutilized (very low imbalance) or exhibit sporadic bursts (high imbalance).

Discussion and Conclusion

- Uneven workload distribution across GPUs suggests potential for requesting fewer GPUs or improving load balancing.
- Perlmutter shows high demand for double precision (scientific/engineering) and lower demand for ML/AIjobs.
- Job-level computation phases mainly drive inconsistent usage patterns over time.
- Jobs with high GPU hours but low mean GPU_UTIL are prime optimization targets, identified by our methods.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 2047120. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility, operated under Contract No. DE-AC02-05CH11231 using NERSC award DDR-ERCAP0034262.

References

[1] Agelastos, Anthony, et al. "The lightweight distributed metric service: a scalable infrastructure for continuous monitoring of large scale computing systems and applications." SC'14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. IEEE, 2014.
[2] 2020. Slurm Workload Manager. https://slurm.schedmd.com/documentation. html
[3] NVIDIA Corporation. 2023. NVIDIA Data Center GPU Manager (DCGM). https: //developer.nvidia.com/dcgm. Accessed: 2024-10-15.