An Interactive GPU Metric Dashboard for HPC clusters

Wentao Shi [⋆,†]
wshi6@lsu.edu

Brandon Cook [⋆]
BGCook@lbl.gov

Arghya Ronnie Chatterjee [⋆]
ronnie@lbl.gov

Johannes Blaschke [⋆]
jpblaschke@lbl.gov

[⋆]National Energy Research Scientific Computing Center (NERSC), Lawrence Berkeley National Laboratory
Berkeley, CA, USA
[†]Center for Computation & Technology, Louisiana State University
Baton Rouge, LA, USA

ABSTRACT

Scaling up programs to run at the scale of a modern high-performance computing (HPC) center can be a daunting task. One of the first questions developers ask is: "is my program using all the hardware available?" Many tools can extract detailed performance data on applications. But the level of detail that these tools deliver comes at a cost: significant resource and time must be invested in collecting and analyzing such performance data. But to answer a question like "am I using all 4 GPUs per node?", this level of detail is overkill. So, this project aims to provide developer a user-friendly application to have a peek in their program's GPU utilization.

INTRODUCTION

The goal of this project is to develop a web application that displays GPU utilization metrics. This application aims to provide users at NERSC a quick and convenient access to the GPU utilization metrics for performance analysis and trouble shooting. Existing monitoring tools such as Lightweight Distributed Metric Service (LDMS) [1] provide means to retrieve metrics data from HPC clusters. However, it is not trivial to extract and exploit LDMS data. Having users parse the LDMS output is an undue burden on new users. So, this work also serves as a template for other HPC centers to implement similar tools. As this tool is available all the time, non-invasive, and scalable to system-sized jobs, we envision that the data provided gives users insight into when more detailed profiling is needed. Furthermore, first-time users of HPC systems can be overwhelmed by the complexity of performance profiling tools (e.g., Nsight Systems). We develop a simple tool such as these as a "gentle on-ramp" to more sophisticated performance profilers.

DATA PROCESSING PIPELINE

The backbone of the application is a data processing pipeline that resides on the HPC clusters. In the initial state, it collects data from the compute nodes using the Lightweight Distributed Metric Service (LDMS) [1]. LDMS is integrated with the Nvidia Data Center GPU Manager (DCGM) to collect GPU metrics. The collected data are dumped to CSV files.Since the the CSV files are massive (at a scale of 100 GB per file), NERSC has developed a Python program to convert the CSV files into Apache Parquet [4] files. This program also generates an index of the Parquet files (using a SQL database), allowing for a near constant-time data retrieval. A Flask [3] web application along with another Python program exposes the data to users. Upon a user request, the Flask web application performs an SF API call to run this Python program to look up the data corresponding to a specific job in the Parquet files, and convert them into JSON strings. The JSON strings are stored in a database afterwards. The web application performs another SF API call to query the database for the JSON string and generate the plot using Plotly.js.

SYSTEM ARCHITECTURE

The system architecture consists of two parts. As illustrated in Figure 1, the web application retrieves data from the HPC cluster via the Superfacility API, and then plot the GPU utilization metrics to users.

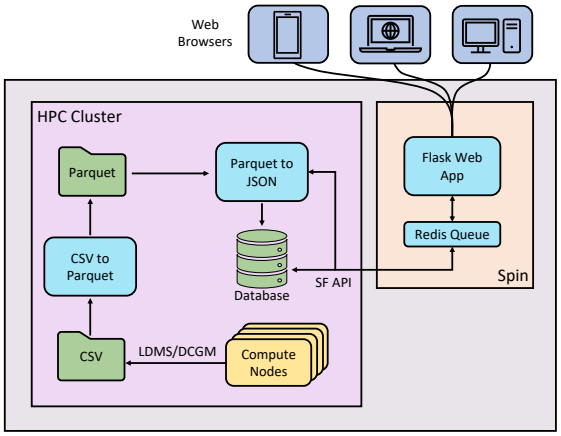

Figure 1. System architecture of NERSC GPU Dashboard.

Lightweight Distributed Metric Service (LDMS). LDMS is a scalable, lightweight, and system-wide monitoring tool for collecting system metrics. It collects high-fidelity data at low cost [1]. After being deployed the data collector instances of LDMS will be running on compute nodes to collect metrics, and data aggregator instances of LDMS will gather the

collected data then dump them to CSV files.

Data Center GPU Manager (DCGM). DCGM is a software suite developed by Nvidia for managing and monitoring GPUs in cluster environments. It is integrated with LDMS and the GPU metrics are stored as CSV files. Apache Parquet File Storage. The data are converted from CSV files to Parquet files for faster and more efficient access. Apache Parquet is a data format that leverages the columnar data storage technique, supporting efficient compressing and encoding schemes [4]. In this project, the file sizes are reduced approximately 10x to 100x depending on the property of data.

The GPU Dashboard Web App. In this project, the web application is built with the Flask [3] micro web framework written in Python. The webpages are rendered by the Jinja template engine that comes with Flask. The code that generates the interactive plots are based on Plotly.js.

The Super Facility (SF) API. The SF API is provided by NERSC to users for building applications that interact with the clusters [2]. In this project, our web app uses the "command" SF API to call the Python program that converts Parquet files to JSON. Then, the web app retrieves the JSON string according to the response by using the "task" SF API. SF APIs are available at https://api.nersc.gov/. Interactive Plots using Plotly.js. The interactive plot is implemented using Plotly.js. Users can select different sections of data to visualize, zoom in, and zoom out.

USER INTERFACE

Users only need to provide the Slurm job ID. After entering the job ID, a default plot will be loaded in the web browser. Users can specify which node to display; all the nodes involved in this job will be included. Users can select which metric to display; all the supported metrics are loaded, so users can quickly switch between different metrics. Users can specify which GPU(s) to display by checking the boxes on the left of the dashboard; this application supports displaying the metrics of multiple GPUs in a single plot. The plot is interactive, which means that the users can zoom in and zoom out.

REFERENCES

[1] Anthony Agelastos, Benjamin Allan, Jim Brandt, Paul Cassella, Jeremy Enos, Joshi Fullop, Ann Gentile, Steve Monk, Nichamon Naksinehaboon, Jeff Ogden, and others. 2014. The lightweight distributed metric service: a scalable infrastructure for continuous monitoring of large scale computing systems and applications. In *SC'14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 154–165.

[2] Bjoern Enders, Debbie Bard, Cory Snavely, Lisa Gerhardt, Jason Lee, Becci Totzke, Katie Antypas, Suren Byna, Ravi Cheema, Shreyas Cholia, and others. 2020. Cross-facility science with the Superfacility Project at LBNL. In *2020 IEEE/ACM 2nd Annual Workshop on Extreme-scale Experiment-in-the-Loop Computing (XLOOP)*. IEEE, 1–7.

[3] Miguel Grinberg. 2018. *Flask web development: developing web applications with python*. " O'Reilly Media, Inc.".

[4] Deepak Vohra. 2016. Apache parquet. In *Practical Hadoop Ecosystem*. Springer, 325–335.