

Resource Usage and Power Profiling for High-Energy Physics Workflows

Background and Introduction

The Issue of Power and Memory Consumption

- HEP workflows are becoming more and more energy and memory taxing
 - More data volume and simulation complexity
 - E.g. DUNE
- Resource efficiency awareness will eventually become a requirement to maintain sustainable HEP experimental computations

How Can We Tackle This?

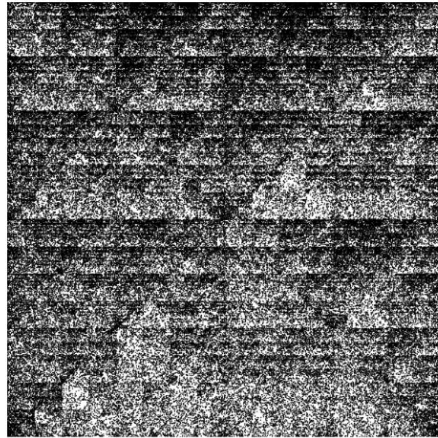
- Resource usage patterns in any HPC application are complex
 - Need to profile resource usage
 - Profiling can assist scheduling, programming, etc.
- Attempts have been made to profile resource usage generally
 - Karimi et al.³, Köhler et al.⁵
- Resource usage complexity → easier to dial into one application first

TIME-SERIES ANALYSES METHODS

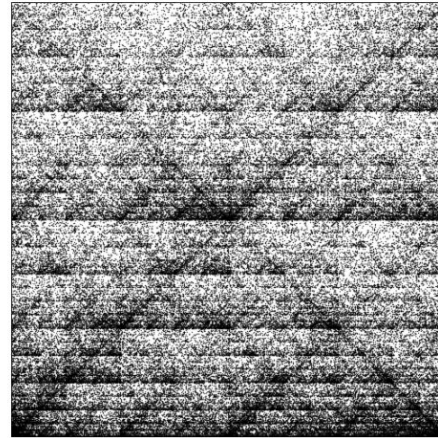
- Time-series analysis prevalent in biology, physics, etc.
 - Weather data
 - ECG (heart) data
 - EEG (brain) data
- Many techniques to filter and characterize data
 - Data preprocessing and smoothing
 - Statistical methods (ARIMA, Autocorrelation, etc.)
 - AI/ML methods

Chaos Game Representation (CGR)

- CGR transforms discrete sequential data into fractal-like geometric patterns¹
- It has most notably been used to for representing genome sequencing data²



(a) DNA of Human herpesvirus 5 strain TR, complete genome—235681 bp

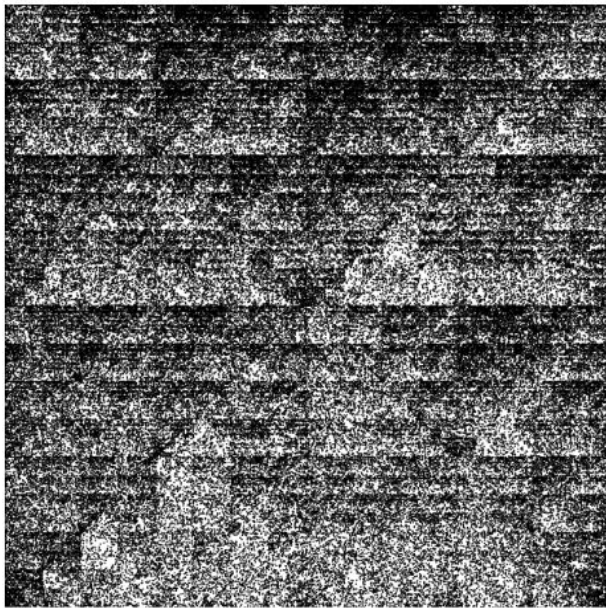


(b) DNA of *Chenopodium quinoa* cultivar Real Blanca chloroplast, complete sequence, whole genome shotgun sequence—152282 bp

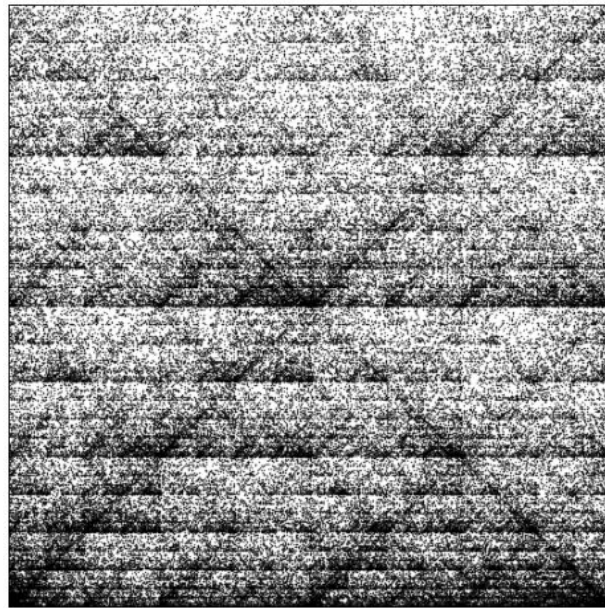
How Does It Work?

- Given n possible inputs
 - E.g. $n=4$ for genomes (A,T,C,G)
- Make an equilateral polygon with n vertices
 - Each input maps to a vertex
- Take a random point and mark the point halfway between the initial point and the vertex for the first input in our sequence
- Do this again using our new point and the next input in the sequence

Our Approach: Chaos Game Representation (CGR)



(a) DNA of Human herpesvirus 5 strain TR, complete genome—235681 bp



(b) DNA of Chenopodium quinoa cultivar Real Blanca chloroplast, complete sequence, whole genome shotgun sequence—152282 bp

***NOTE: we show $n=4$ because it is a simple, but different n values are fine

How Do We Adapt Power Usage Data

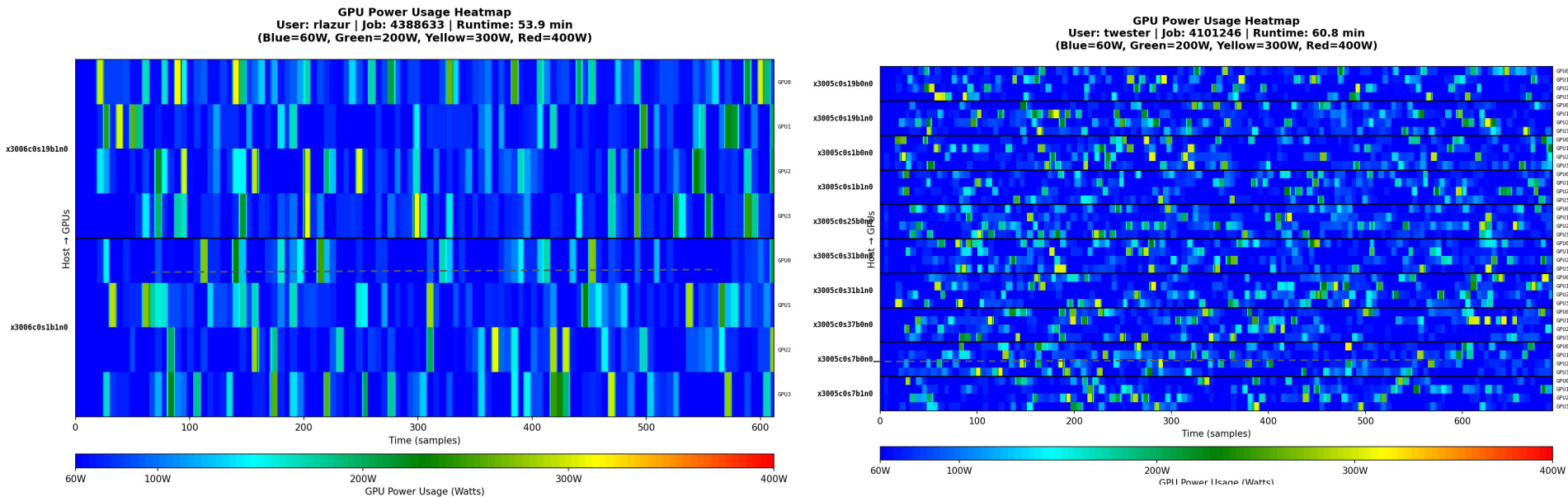
- We can “bin” power usage data to discretize the power usage domain⁴
- Naive approach: Make n evenly spaced bins
- Would be better to inspect the data to see if we can be more insightful with how we choose our bins
- We decided to analyze two users' resource usage (both from FNAL) and expand from there

Analysis of Power and Memory Usage Data on Polaris

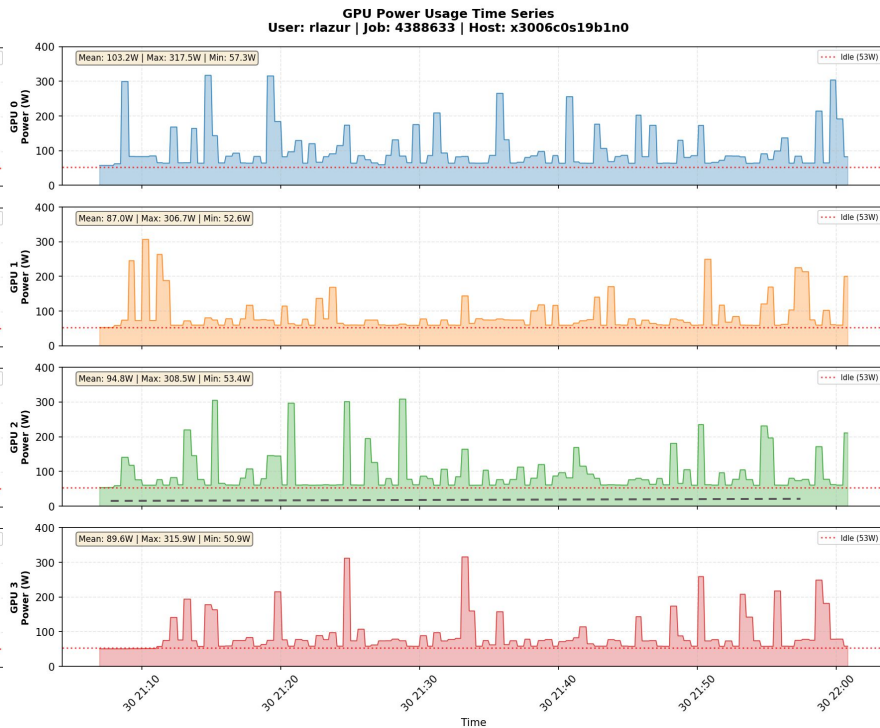
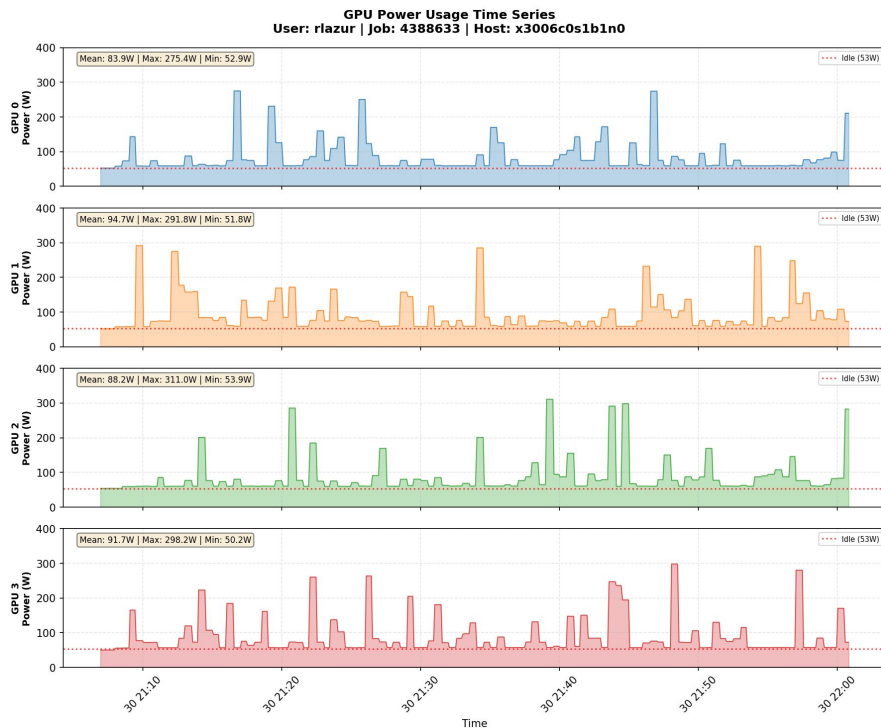
What Is The Raw Data (Time Series)

- Load (%)
 - Percent of time the GPU is used
- Power (Watts)
 - Average node power usage in a job
- Mem (%)
 - Percent of time reading or writing
- Mem_Alloc (bytes)
 - Amount of reserved and allocated memory
- Others
 - start/end time, number of nodes, etc.

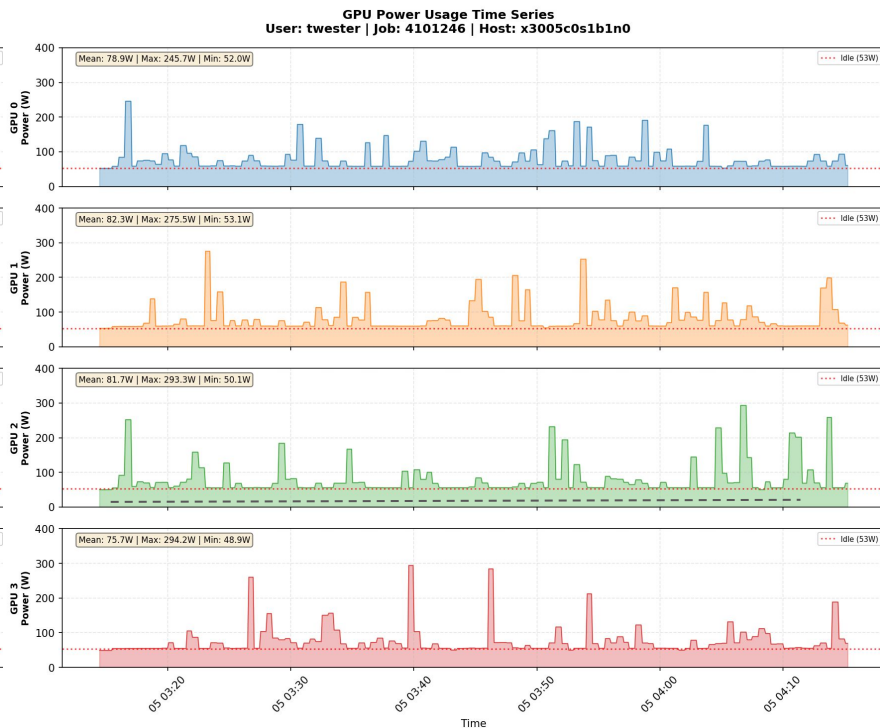
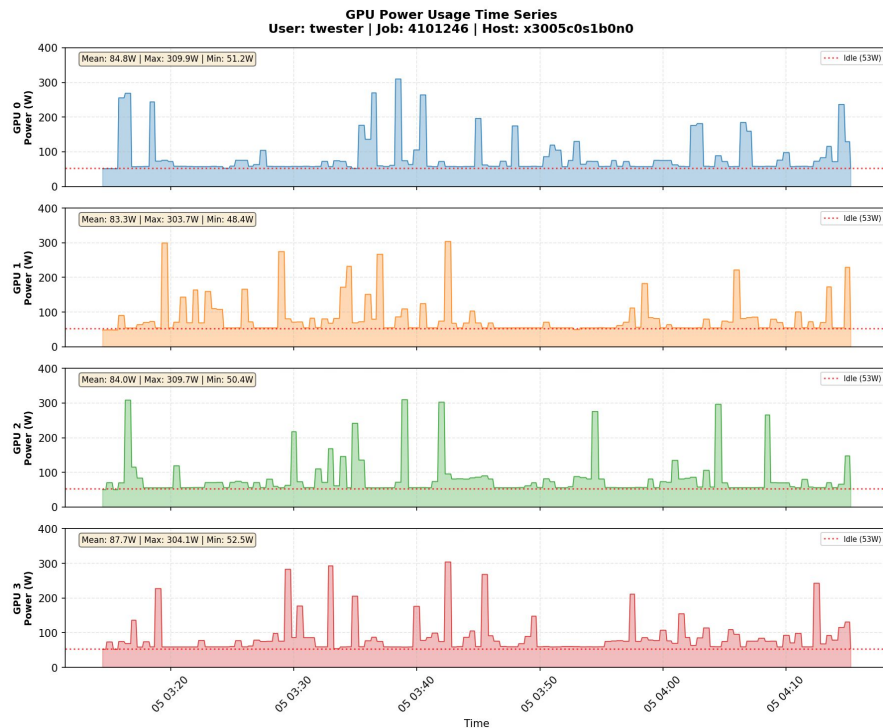
What Is The Raw Data: Heat Maps



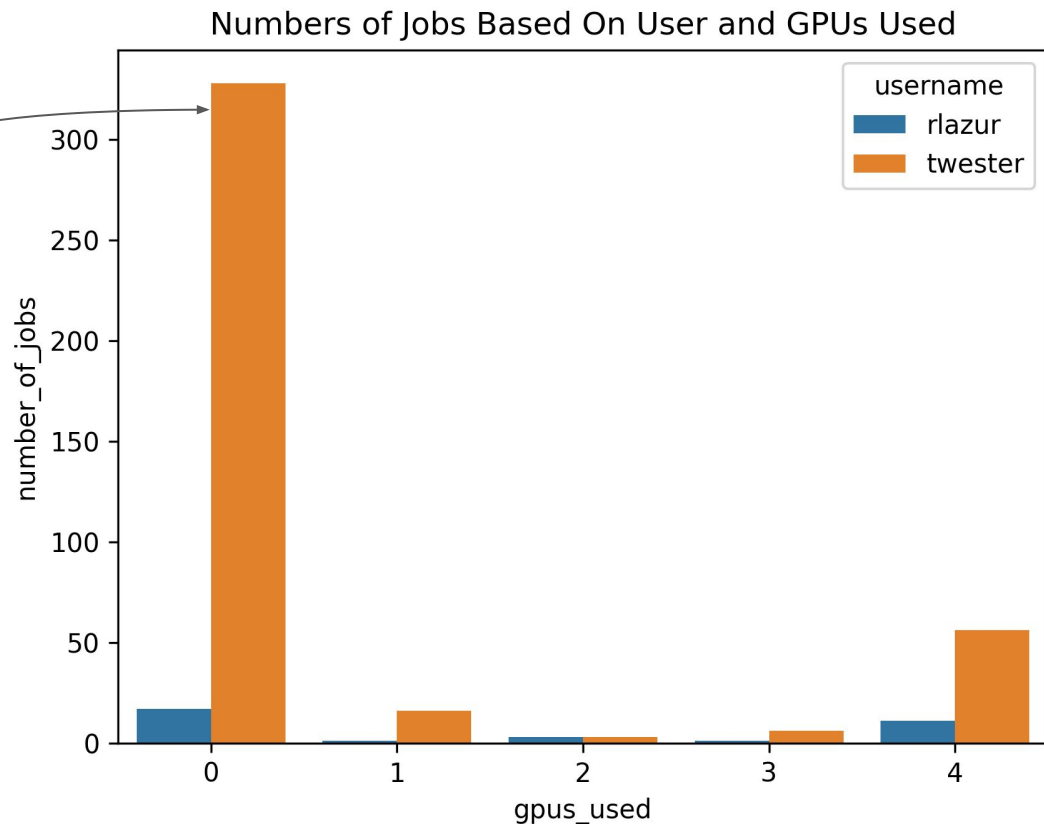
What Is The Raw Data: Time Series (rlazur)



What Is The Raw Data: Time Series (twester)



GPUs Used Statistic



```
10832160 0
10832161 0
10832162 0
10832163 0
10832164 0
Name: GPU_load_1, Length: 30880, dtype: int64
10832186 0
10832187 0
10832188 0
10832189 0
10832190 0
10832191 0
10832192 0
10832193 0
10832194 0
10832195 0
10832196 0
10832197 0
10832198 0
10832199 0
10832200 0
10832201 0
Name: GPU_load_1, dtype: int64
10832202 0
10832203 0
10832204 0
10832205 0
10832206 0
10832468 0
10832469 0
10832470 0
10832471 0
10832472 0
Name: GPU_load_1, Length: 271, dtype: int64
10836848 0
10836849 0
10836850 0
10836851 0
10836852 0
10836935 0
10836936 0
10836937 0
10836938 0
10836939 0
Name: GPU_load_1, Length: 92, dtype: int64
10836940 0
10836941 0
10836942 0
10836943 0
10836944 0
10837007 0
10837008 0
10837009 0
10837010 0
10837011 0
Name: GPU_load_1, Length: 72, dtype: int64
10837012 0
10837013 0
10837014 0
10837015 0
10837016 0
10837017 0
10837018 0
```

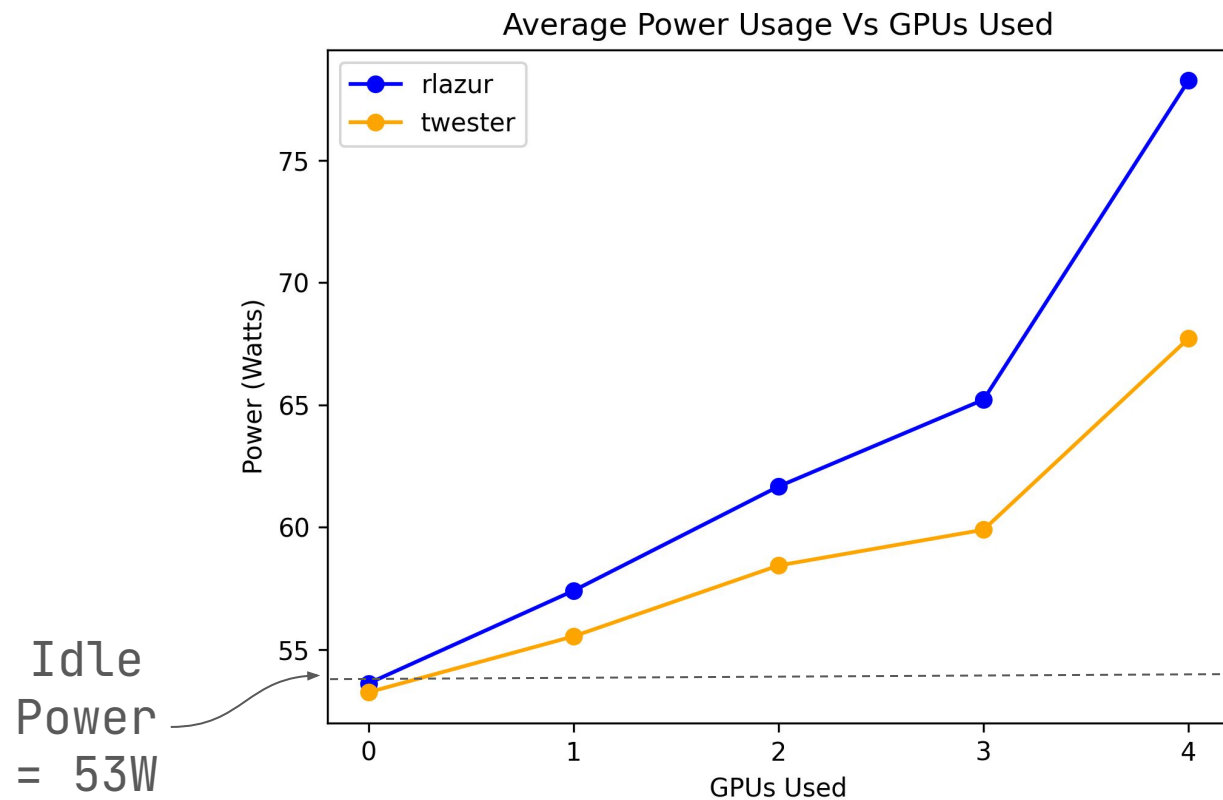
Data Aggregation

- Summarize data by job
 - load, power_usage, mem, mem_alloc
 - Average, standard deviation, etc.
 - GPUs Used
- Generate graphs
 - Line plots
 - Violin plots

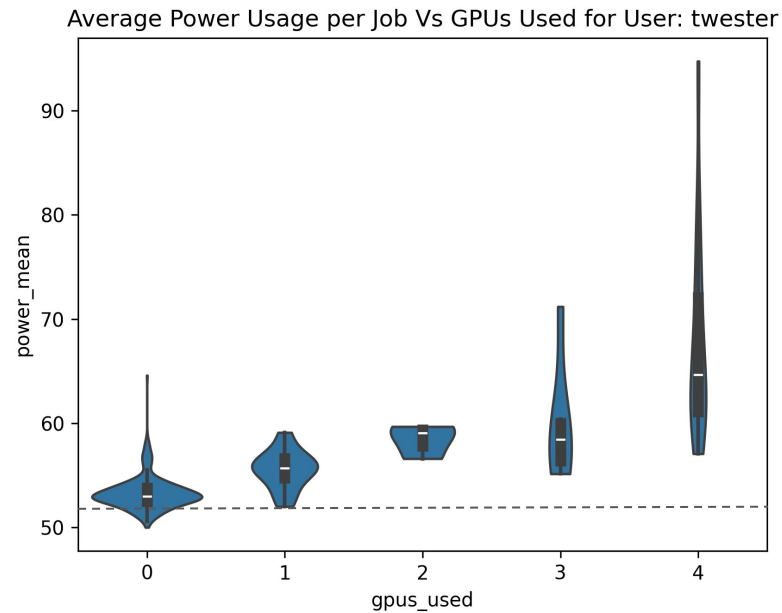
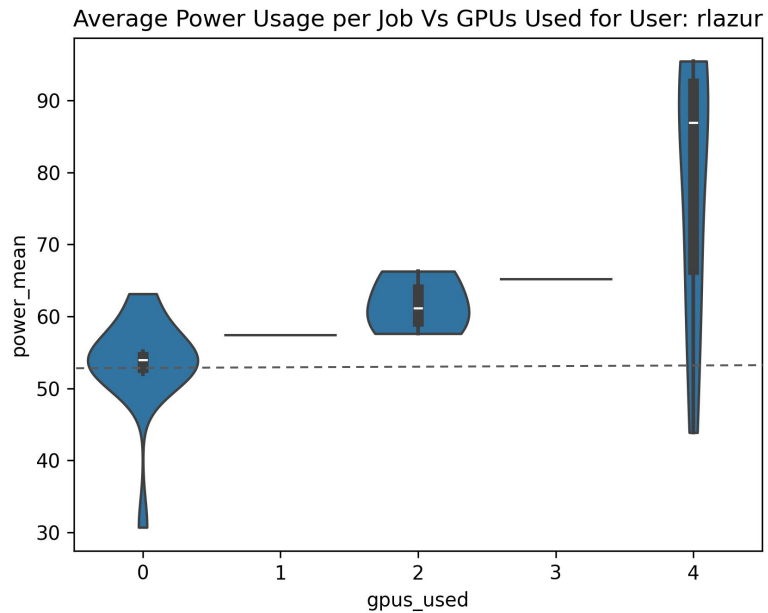
GPUs Used Statistic

- Load per GPU
 - 0/1 values based off if ANY of the GPU is being used
 - Aggregate these to tell what percentage of time the GPU was used
 - Must set a threshold to say a GPU is used (we used 2%)
- GPUs Used = Amount of GPUs with more than 2% load percentage

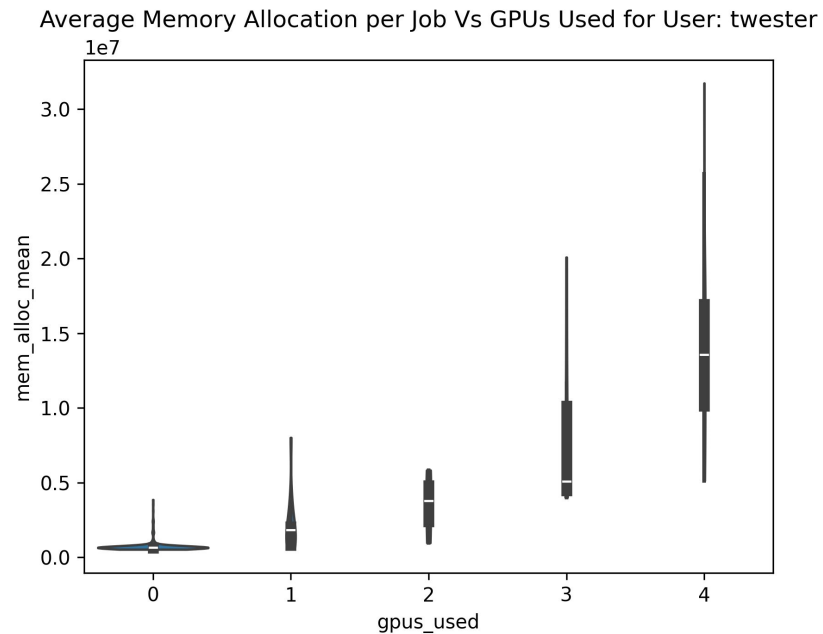
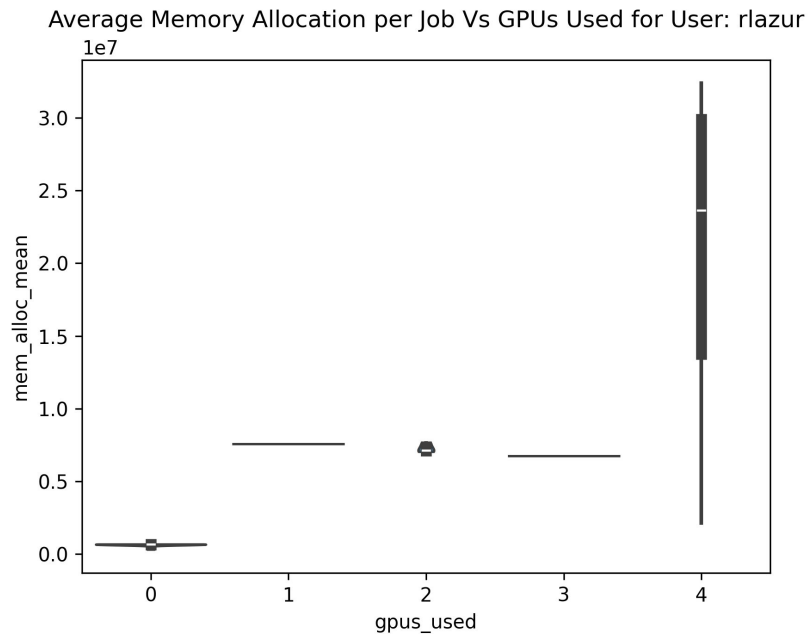
Power Usage vs GPUs Used



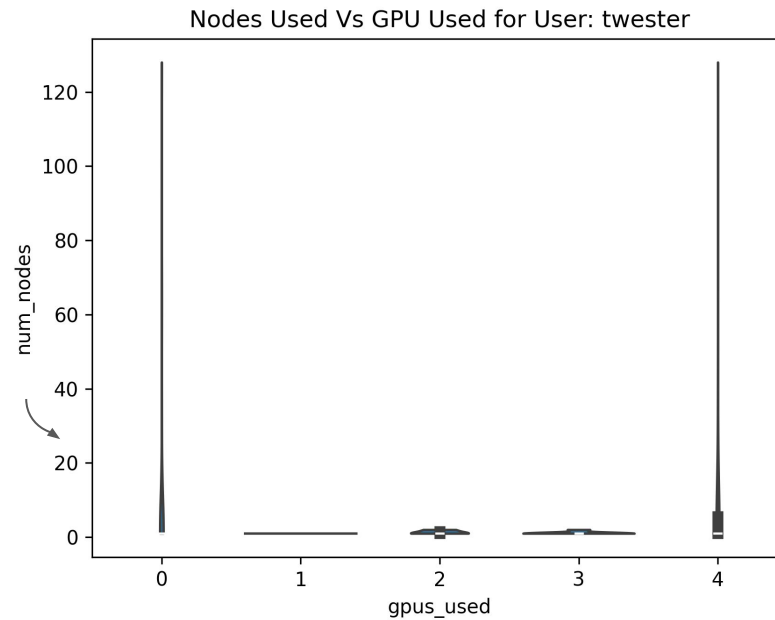
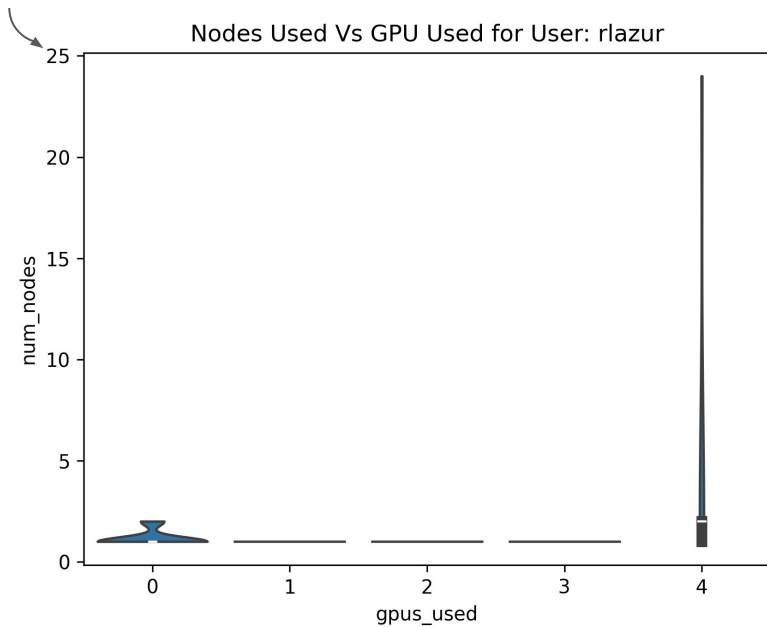
GPU Power Usage Vs GPUs Used



GPU Memory Allocation Vs GPUs Used



Nodes Used Vs GPUs Used



Issues with CGR

- Data scarcity
- Data uniformity
- Multiple data streams (one per GPU)
- Binning is unintuitive and application-specific

Possible Next Steps So Far

- Recently got access to Perlmutter data from the Tachyon project
 - See how other experiments' data looks like
 - Is it similar? different? howso?
- Explore other methods of profiling
 - CGR does not look that possible
 - What other methods are out there?

Source

1. Eunice Y. S. Chan and Robert M. Corless. 2020. Chaos Game Representation. arXiv preprint. ArXiv:2012.09638 [math].
2. H.Joel Jeffrey. 1990. Chaos game representation of gene structure. Nucleic Acids Research, 18(8):2163-2170.
3. Ahmad Maroof Karimi, Naw Safrin Sattar, Woong Shin, and Feiyi Wang. 2024. Power profile monitoring and tracking evolution of system-wide hpc workloads. In 2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS), pages 93-104.
4. Nitin Kumar, Venkata Nishanth Lolla, Eamonn Keogh, Stefano Lonardi, Chotirat Ann Ratanamahatana, and Li Wei. 2005. Time-series Bitmaps: a Practical Visualization Tool for Working with Large Time Series Databases. In Proceedings of the 2005 SIAM International Conference on Data Mining, pages 531-535. Society for Industrial and Applied Mathematics.
5. Sven Köhler, Lukas Wenzel, Max Plauth, Pawel Böning, Philipp Gampe, Leonard Geier, and Andreas Polze. 2021. Recognizing hpc workloads based on power draw signatures. In 2021 Ninth International Symposium on Computing and Networking Workshops (CANDARW), pages 278-284.

**Thank
You ! ! !**