Predictive Energy Modeling for GPU Workloads: A Machine Learning Approach to Sustainable Data Centers

Brandon Ismalej

California State University, Northridge

University of Idaho, Graduate Seminar Series

April 16, 2025

Foundation of Work

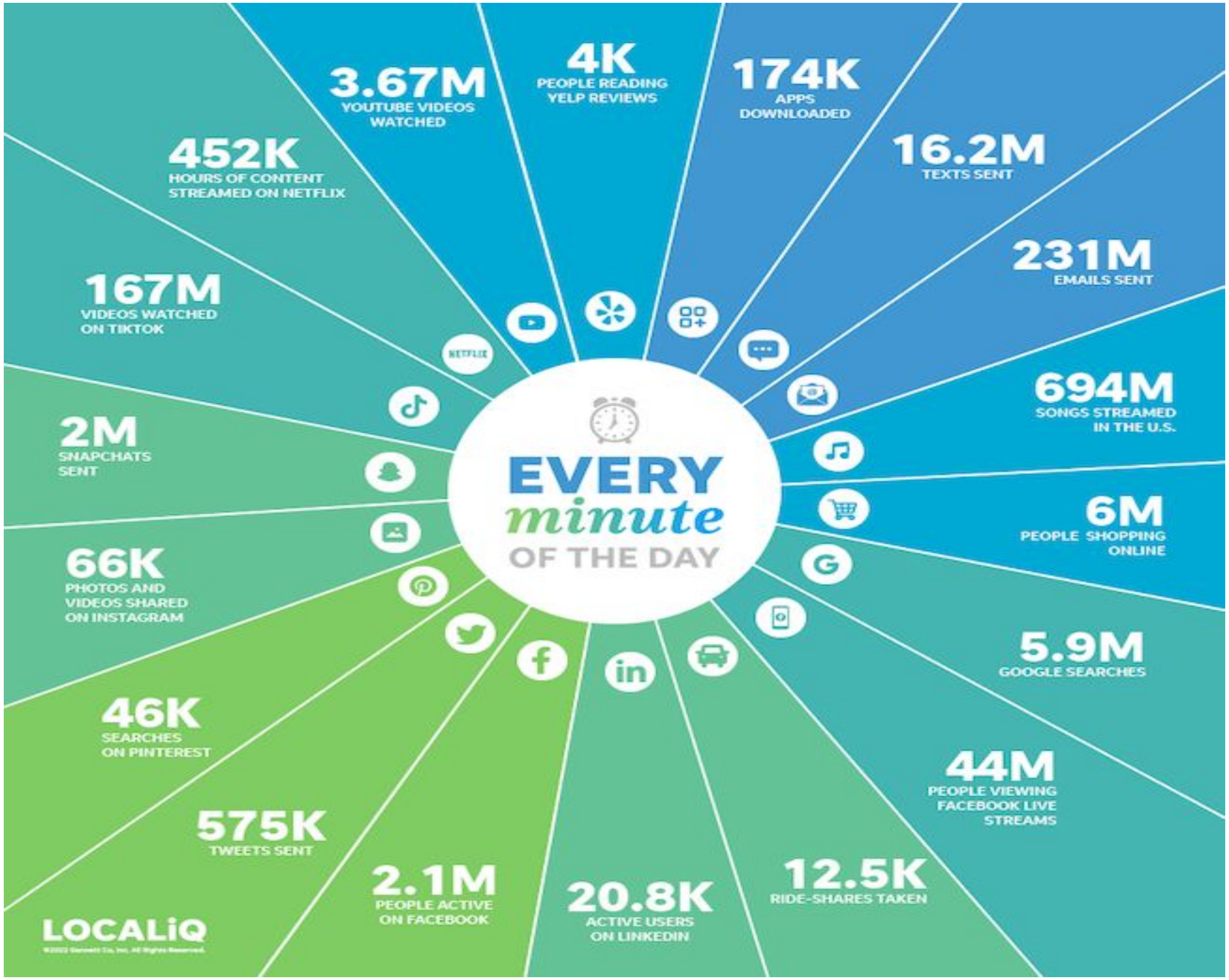
This work took place:

- CSUN's SfS2 Summer Program - Funded by US Dept. of Education
- With CSUN NSF REU Site: Applying Data Science on Energy-efficient Cluster Systems and Applications
- Advisor: Dr. Xunfei Jiang
  - REU PI
  - SfS2 Core Team
  - Co-director of Cloud Infrastructure and Server Architecture (CISA) Lab

Outline

- Introduction
- Background & Related Work
- Methodology
- Results
- Conclusion
- Q&A

Motivation & Problem



Motivation & Problem

Google Data Center in Berkeley County by Wade Spees wspees@postandcourier.com

Motivation & Problem

Motivation & Problem

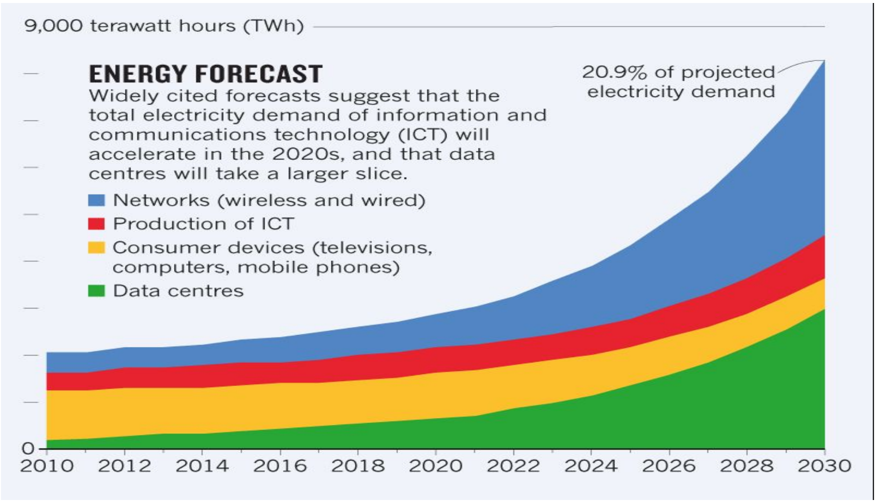
Facebook: 18 data centers
2.936 billion monthly active users (April 2022)
4 PB (4*1024 TB) of data generated per day
300PB in Hive (data warehouse)

Google: 23 data centers
Youtube: 1 billion hours of video watched per day (**1000 PB/day**)
621 hours of video uploaded every minute (**873 TB/day**)

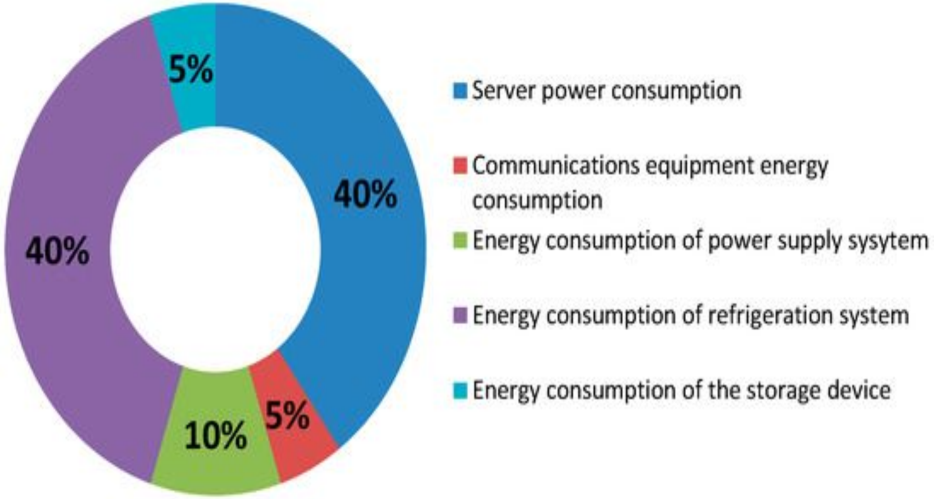
US Datacenters: 2-3% of all electricity generate
73 billion kWh in 2020 ($0.14*73 = $10.22 billion)
1% energy cost saving = $0.1 billion = $100 million

Background

Motivation:

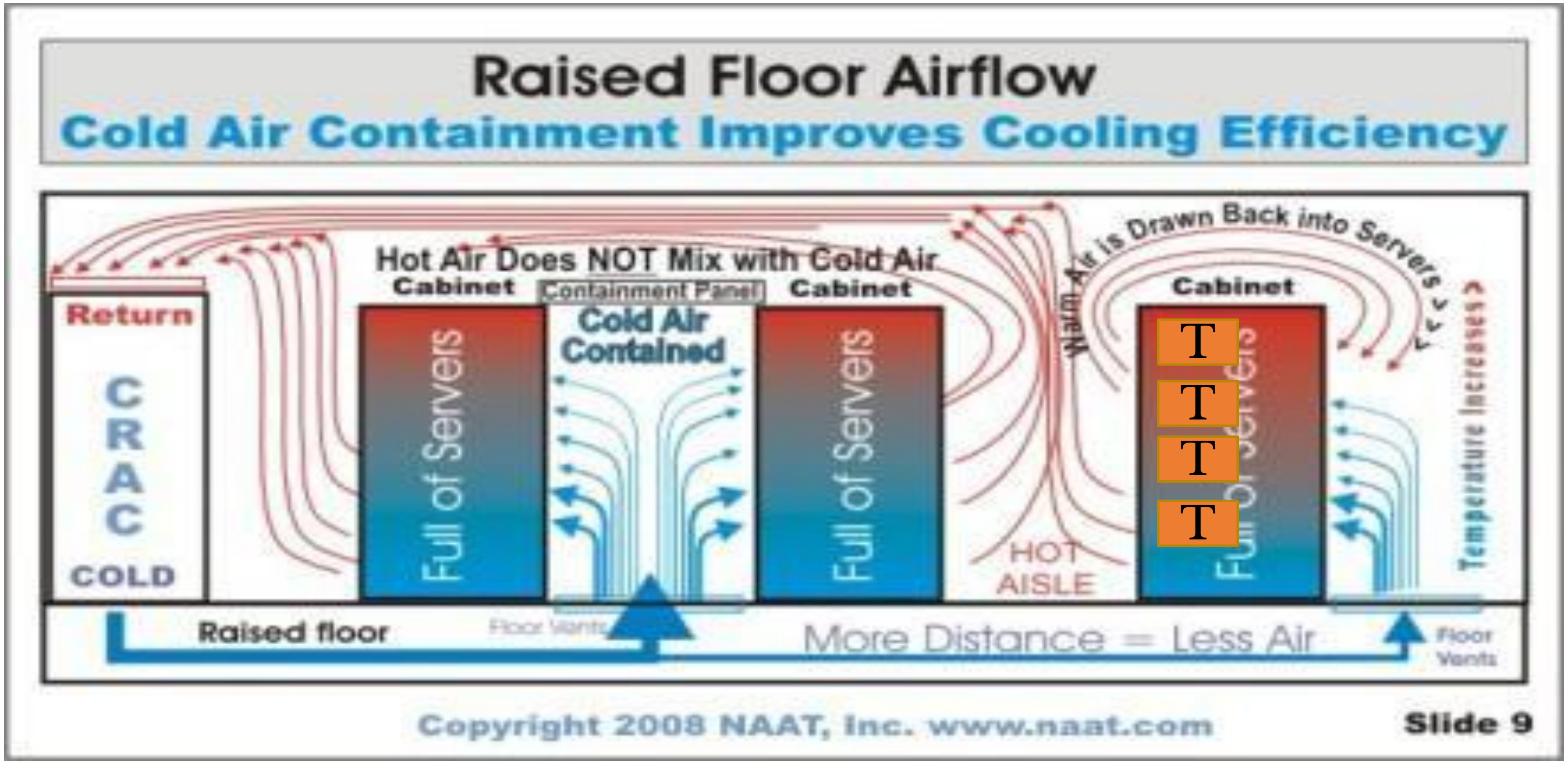


Electricity usage (TWh) of Data Centers 2010-2030 / Source: (Nature, 2018)



Summary of Energy Consumption Distribution in data centers / Source: (MDPI, 2017)

Thermal-aware energy-efficient workload scheduling



Raised Floor Air Cooling for Datacenters / Source: naat.com

T: Task **CRAC:** Computer Room Air Conditioner

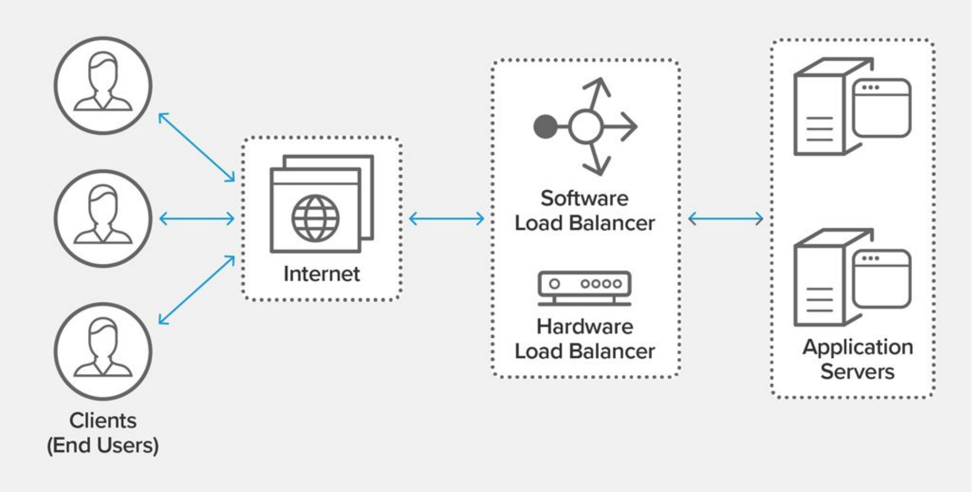
Thermal-aware energy-efficient workload scheduling

Load balancing:

- Network distributes traffic
  - Client → load balancer → server

Load balancing algorithms:

- Adaptable to changes in network traffic
- Thermal aware load balancing
- Energy efficient load balancing



Path from client to server using load balancing.

Modeling of Cluster Servers [CPU]

Previous Work: *I. Nisce, X. Jiang, S. P. Vishnu, 2023*

- **Purpose of the Paper:**
  - Using ML-based approach to predict temp. consumption of computer server based on CPU-intensive workload

- **Data Collection:**
  - Whetstone benchmark - to simulate CPU-intensive workload
  - Stream benchmark – used to collect main memory bandwidth
  - Postmark – used to simulate I/O-intensive workload by increasing the disk utilization

- **ML Algorithms for Thermal Prediction:**
  - XGBoost (eXtreme Gradient Boosting)
  - Light Gradient Boosting
  - Artificial Neural Networks

- **Results:**
  - XGBoost Model to predict CPU temperature

Modeling of Cluster Servers [GPU]

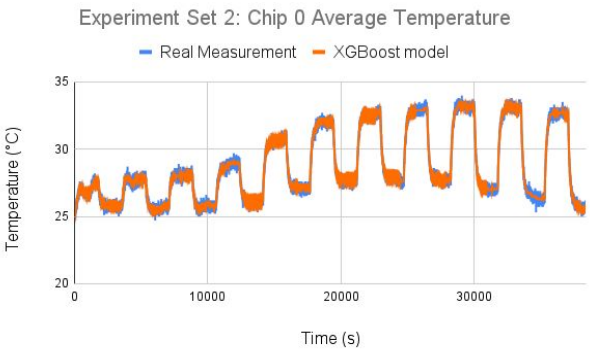
Previous works: *M. Smith, L. Zhao, J. Cordova, X.-F. Jiang, and M. Ebrahimi, 2023, 2024*

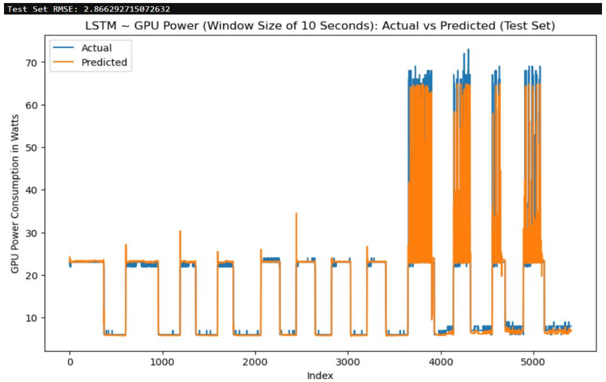
GPU-Intensive Work	Classification of Work
BERT	Large Language Model
DistilBERT	Large Language Model
Image Classification	Image Classification
Sepia Filtering	Image Filtering
Euclidean Distance	Math Heavy Operation
Blackscholes	Financial Algorithm
MonteCarlo	Statistical Inferences
Matrix Multiplication	Math Heavy Operation
K-means Clustering	Machine Learning Algorithm
Fourier Transformation	Decompose Audio Data
Spectrogram	Visualize Decomposed Audio Data

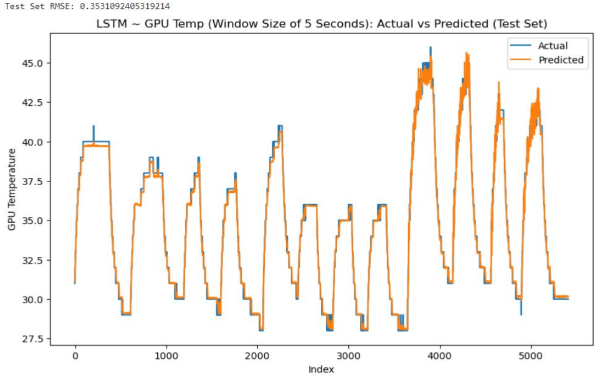
Trained Model Types
Linear Regression
Bayesian Ridge Regression
XGBoost*
LightGBM
Basic NN for Regression
Multi-Layer Perceptron
RNN with Attention Mechanism
LSTM*

* Most successful models that were tested.

Modeling of Cluster Servers







- The XGBoost model had better performance among all in CPU temperature predicting. The LSTM models had better performance predicting erratic behavior of GPU power and temperature. XGBoost models had similar levels of performance for GPU temperature.

- Will include more experiments with varied CPU/GPU utilization. Train different ML models, measure other components such as I/O, and measure server temperature and energy consumption with different workloads. Improving accuracy of GPU power model is important.

All aforementioned previous works.

Cluster System Configuration

- Located in **CSUN's Main Distribution Frame** (MDF)
- Concrete & brick, no windows
- Flat roof, 8" thick walls
- 2" interior insulation & gypsum board interior walls
- ~3,400 sq. ft. of floor space with basement for cable management
- Controlled physical access
- Availability of uninterruptible power supply
- Monitored power distribution
- Air conditioning (humidity, filtering, cooling)
- Fire suppression

CSUN MDF Server Racks

Cluster System Configuration

Our cluster
system configuration

CPU	2 * Intel(R) Xeon(R) CPU E5-2690 v2 @ 3.00GHz (10 cores)
Memory	98,304 MB (96 GB)
Disk	2 * Intel SSD DC S352- 480-GB
GPU	NVIDIA Tesla P4 8GB GDDR5
OS	Ubuntu 22.04.2 LTS

Machine Specifications of Cluster System Configuration

Energy-efficient Placement

Problem: we don’t have access to a large datacenter

- Even before we create a load balancer, how would we test it?

Cloud computing simulators

- Programs that simulate real-life datacenters
- Many different choices of simulators

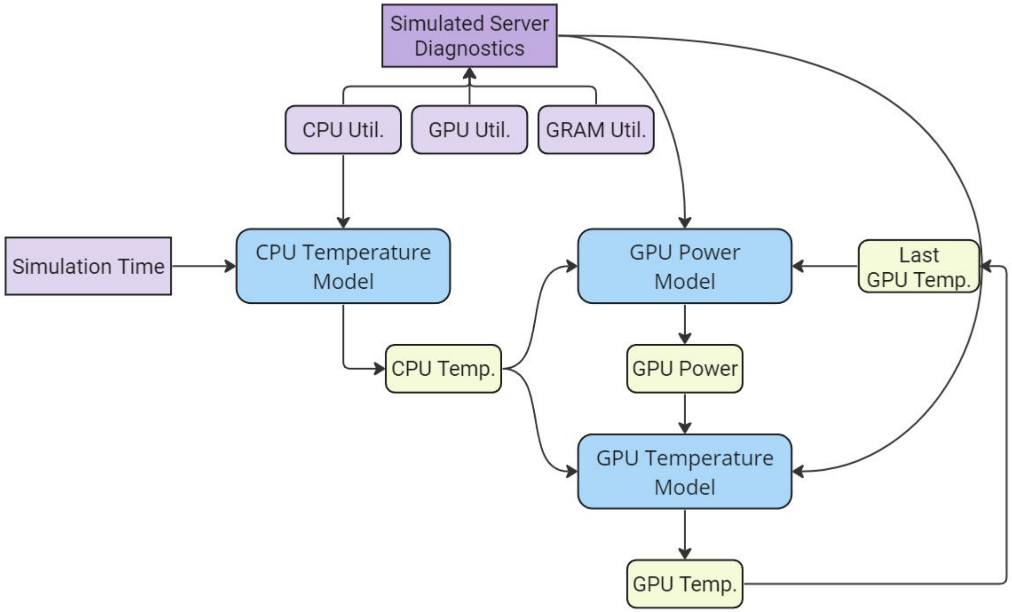
GPUCloudSimPlus

- Models both CPU and GPU behavior
- Easy to write a load balancer for

Energy-efficient Placement

Three ML models

- CPU models:
  - CPU Temperature
- GPU models:
  - GPU Power
  - GPU Temperature



The three machine learning models working together to predict the GPU temperature of each individual server.

Models work together

- Output: GPU temperature

Now we can design a load balancing algorithm

Energy-efficient Placement

ThermalAwareGpu Algorithm (TAGpu)

- Goal:
  - Limit info. needed by load balancer
  - Lower temperature of each server

- Sort hosts by GPU temp every 10 seconds

- General idea of the process:
  - Use first host in queue (lowest temperature)
  - Assign X tasks to VMs in that host
  - Assume this will make the host warmer
  - Move this host to end of queue

Works: *M. Smith, L. Zhao, J. Cordova, X.-F. Jiang, and M. Ebrahimi, 2023, 2024*

Algorithm 1 ThermalAwareGpu

```
procedure GETVMFORCLOUDLET(CLOUDLET)
    X = 0.5 × (numVMs / numHosts)
    if timeInSecsSinceLastSort >= 10 then
        Sort hostQueue by host GPU temperature
    end if

    count ← −1
    foundVm ← null
    while (!foundVm) && (++count < numHosts) do
        host ← hostQueue.peek()
        foundVm ← First VM with resources in host
        if foundVm ≠ null then
            Dequeue foundVm, enqueue to host
        else
            Dequeue host, enqueue to hostQueue
        end if
    end while

    if foundVm.hostId ≠ lastHostId then
        lastHostId ← foundVm.hostId
        host.numCloudlets ← 1
    else if ++host.numCloudlets >= X then
        Dequeue host, enqueue to hostQueue
        host.numCloudlets ← 0
    end if

    return foundVm
end procedure
```

Goals & Approach

Goal of current work:

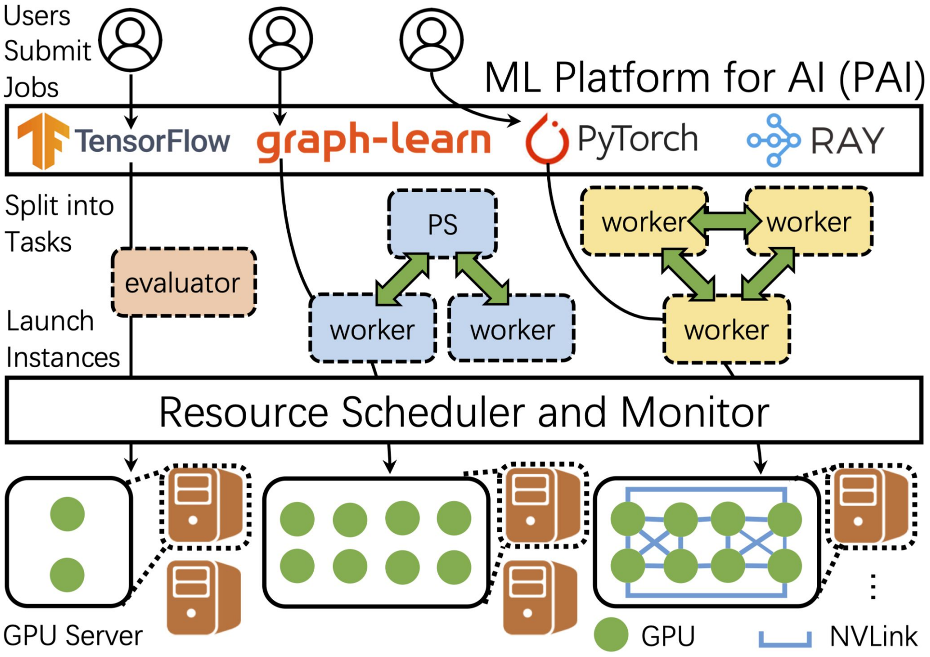
- Develop GPU power ML-model
  - Predict GPU power based on metrics gathered from each GPU task

- Training data:
  - Restricted to format of the Alibaba Real-World Cluster Trace
  - Would be data plugged into similar to enhance "aware" algorithms

Real-World Workload Trace

Alibaba 2020 Cluster Trace

- From the Alibaba Cloud Platform for AI (PAI)
  - enterprise-level ML and deep learning (DL) platform supporting a diverse range of AI-related services
- 6,500 GPUs, ~1,800 machines
- Users can submit ML jobs developed in a variety of frameworks
- Users specify needed resources upon submission (e.g. GPUs, CPUs, memory)
- Each task may have >= 1 instance, can run on multiple machines



Alibaba PAI Process Flow

Real-World Workload Trace

SenseTime Helios GPU Cluster Trace

- Helios: A private datacenter dedicated for DL research and production within SenseTime
- 8 independent GPU clusters
- > 12,000 GPUs total

The workload trace contains:

- 4 GPU clusters
- *Earth, Saturn, Uranus, & Venus*

	Venus	Earth	Saturn	Uranus	Total
CPU	Intel, 48 threads/node		Intel, 64 threads/node		-
RAM	376GB per node		256GB per node		-
Network	IB EDR		IB FDR		-
GPU Model	Volta	Volta	Pascal & Volta	Pascal	-
# of VC’s	27	25	28	25	105
# of Nodes	133	143	262	264	802
# of GPUs	1,064	1,144	2,096	2,112	6,416
# of Jobs	247k	873k	1,753k	490k	3.363k

Helios Datacenter Cluster Configurations

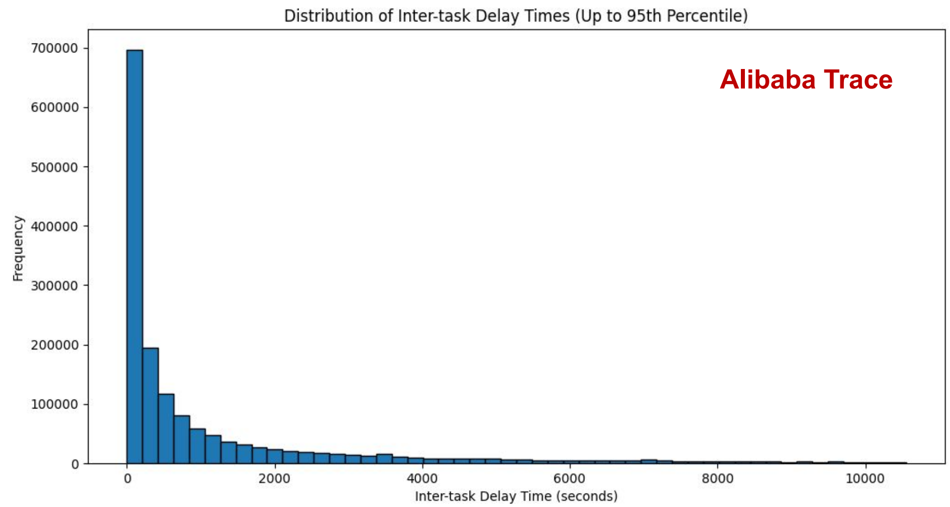
Workload Trace Analysis

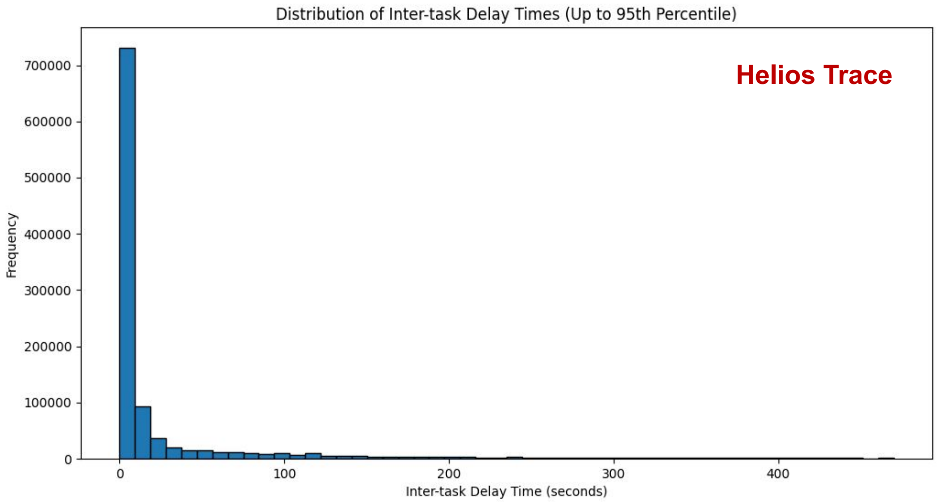
- Statistical analysis of Alibaba and Helios conducted to gain insights on task characteristics
- Overlapping tasks excluded
  - Our cluster server contains only 1 GPU
- Focus on 95th percentile
  - To gain more insights on non-zero delay times
  - Using 100% of data provided no valuable insights

Workload Trace Analysis

	Alibaba Trace	Helios Trace
Count	1,669,790	1,112,131
Q1	67	2
Median	348	4
Q3	1,624	21
Max	2,786,690	4,196,357

Statistical Summary of Workload Traces





Cluster System Configuration

Our cluster system configuration

CPU	2 * Intel(R) Xeon(R) CPU E5-2690 v2 @ 3.00GHz (10 cores)
Memory	98,304 MB (96 GB)
Disk	2 * Intel SSD DC S352- 480-GB
GPU	NVIDIA Tesla P4 8GB GDDR5
OS	Ubuntu 22.04.2 LTS

Machine Specifications of Cluster System Configuration

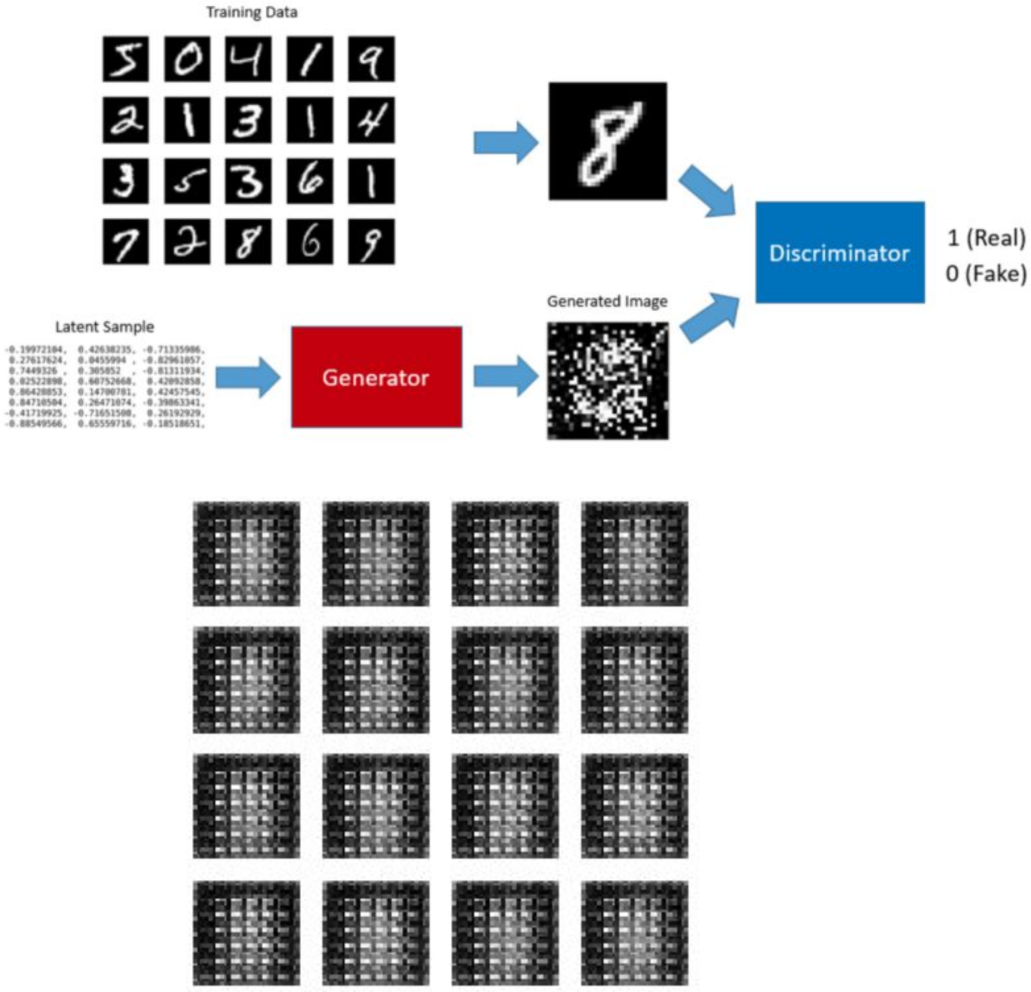
GPU Applications

GPU Application	Domain	Avg. Power [W]	Avg. GPU Util [%]	GRAM Avg. [GiB]
2D_CONVOLUTION	Image Processing	59.05	98.10	0.60
BERT_QA_PREDICTION	Natural Language Processing	60.73	95.05	1.29
BLACKSCHOLES	Computational Finance	22.86	8.83	0.16
DISTILBERT_TRAINING	Natural Language Processing	57.87	98.62	6.47
EUCLIDEAN_DISTANCE	Mathematical Computation	51.08	99.33	1.64
FFT	Signal Processing	55.19	99.45	4.61
GAN_MNIST_DIGITS	Generative Adversarial Network	60.77	97.96	0.79
IMAGE_CLASSIFIER	Image Classification	23.09	6.74	0.29
K_MEANS	Unsupervised Learning	22.86	17.46	0.41
MATMUL	Matrix Multiplication	27.85	13.06	1.36
MONTE_CARLO_PRICING	Financial Simulation	43.01	98.31	3.48
SEPIA_TRANSFORMATION	Image Processing	23.10	5.94	0.26
SPECTROGRAM_TRANSFORMATION	Audio Processing	23.65	5.18	0.18
SIMPLE_CNN	Image Recognition	42.30	97.21	2.02

GPU Applications & Avg. Statistics

GPU Applications

- Generative Adversarial Network (GAN) - A deep learning model used to generate synthetic images
- MNIST Digit Creation - The model learns to create realistic handwritten digits by training on the data
- High computational demand due to matrix operations for training deep neural networks



Process of MNIST GAN Training / Source: Kaggle.com

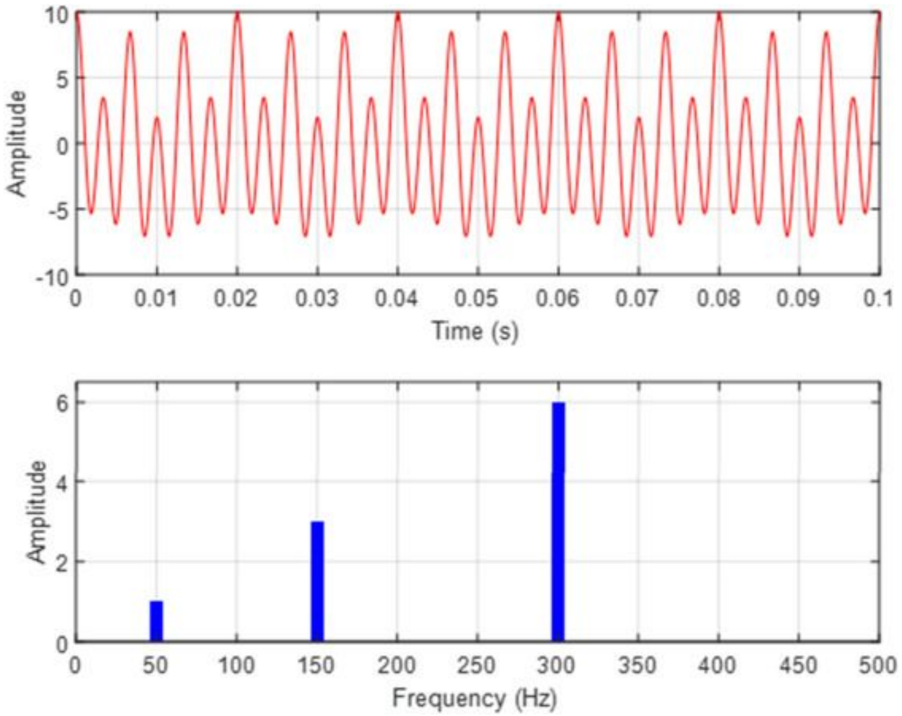
GPU Applications

GPU Application	Domain	Avg. Power [W]	Avg. GPU Util [%]	GRAM Avg. [GiB]
2D_CONVOLUTION	Image Processing	59.05	98.10	0.60
BERT_QA_PREDICTION	Natural Language Processing	60.73	95.05	1.29
BLACKSCHOLES	Computational Finance	22.86	8.83	0.16
DISTILBERT_TRAINING	Natural Language Processing	57.87	98.62	6.47
EUCLIDEAN_DISTANCE	Mathematical Computation	51.08	99.33	1.64
FFT	Signal Processing	55.19	99.45	4.61
GAN_MNIST_DIGITS	Generative Adversarial Network	60.77	97.96	0.79
IMAGE_CLASSIFIER	Image Classification	23.09	6.74	0.29
K_MEANS	Unsupervised Learning	22.86	17.46	0.41
MATMUL	Matrix Multiplication	27.85	13.06	1.36
MONTE_CARLO_PRICING	Financial Simulation	43.01	98.31	3.48
SEPIA_TRANSFORMATION	Image Processing	23.10	5.94	0.26
SPECTROGRAM_TRANSFORMATION	Audio Processing	23.65	5.18	0.18
SIMPLE_CNN	Image Recognition	42.30	97.21	2.02

GPU Applications & Avg. Statistics

GPU Applications

- Fast Fourier Transform (FFT) - Algorithm to convert time-domain signals into frequency components
- Applications in signal processing, audio analysis, image processing, & scientific computing
- PyTorch for GPU-based FFT computation
- High demand for parallel computation, leveraging GPU cores to speed up operations



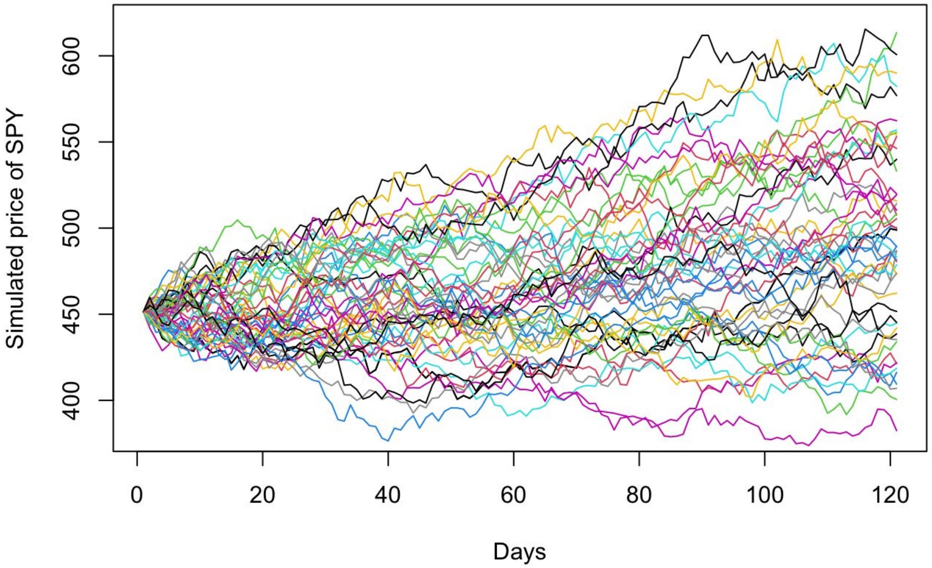
GPU Applications

GPU Application	Domain	Avg. Power [W]	Avg. GPU Util [%]	GRAM Avg. [GiB]
2D_CONVOLUTION	Image Processing	59.05	98.10	0.60
BERT_QA_PREDICTION	Natural Language Processing	60.73	95.05	1.29
BLACKSCHOLES	Computational Finance	22.86	8.83	0.16
DISTILBERT_TRAINING	Natural Language Processing	57.87	98.62	6.47
EUCLIDEAN_DISTANCE	Mathematical Computation	51.08	99.33	1.64
FFT	Signal Processing	55.19	99.45	4.61
GAN_MNIST_DIGITS	Generative Adversarial Network	60.77	97.96	0.79
IMAGE_CLASSIFIER	Image Classification	23.09	6.74	0.29
K_MEANS	Unsupervised Learning	22.86	17.46	0.41
MATMUL	Matrix Multiplication	27.85	13.06	1.36
MONTE_CARLO_PRICING	Financial Simulation	43.01	98.31	3.48
SEPIA_TRANSFORMATION	Image Processing	23.10	5.94	0.26
SPECTROGRAM_TRANSFORMATION	Audio Processing	23.65	5.18	0.18
SIMPLE_CNN	Image Recognition	42.30	97.21	2.02

GPU Applications & Avg. Statistics

GPU Applications

- GPU-Accelerated Monte Carlo Simulation for Financial Option Pricing
- Estimates the net present value of an option by simulating thousands of possible future asset price paths
- Massive parallelization of 100,000+ Monte Carlo samples across GPU cores
- Random number generation and cumulative product operations performed using CUDA-accelerated tensor operations in PyTorch



Source: Rpubs.com

GPU Applications

GPU Application	Domain	Avg. Power [W]	Avg. GPU Util [%]	GRAM Avg. [GiB]
2D_CONVOLUTION	Image Processing	59.05	98.10	0.60
BERT_QA_PREDICTION	Natural Language Processing	60.73	95.05	1.29
BLACKSCHOLES	Computational Finance	22.86	8.83	0.16
DISTILBERT_TRAINING	Natural Language Processing	57.87	98.62	6.47
EUCLIDEAN_DISTANCE	Mathematical Computation	51.08	99.33	1.64
FFT	Signal Processing	55.19	99.45	4.61
GAN_MNIST_DIGITS	Generative Adversarial Network	60.77	97.96	0.79
IMAGE_CLASSIFIER	Image Classification	23.09	6.74	0.29
K_MEANS	Unsupervised Learning	22.86	17.46	0.41
MATMUL	Matrix Multiplication	27.85	13.06	1.36
MONTE_CARLO_PRICING	Financial Simulation	43.01	98.31	3.48
SEPIA_TRANSFORMATION	Image Processing	23.10	5.94	0.26
SPECTROGRAM_TRANSFORMATION	Audio Processing	23.65	5.18	0.18
SIMPLE_CNN	Image Recognition	42.30	97.21	2.02

GPU Applications & Avg. Statistics

Experiment Setup

- Designed five distinct experiments to cover a wide range of inter-task delay times, using our application

- "No delays" simulates peak usage
- Short delays simulate high-frequency task switching
- Longer delays reflect idle periods, like downtime

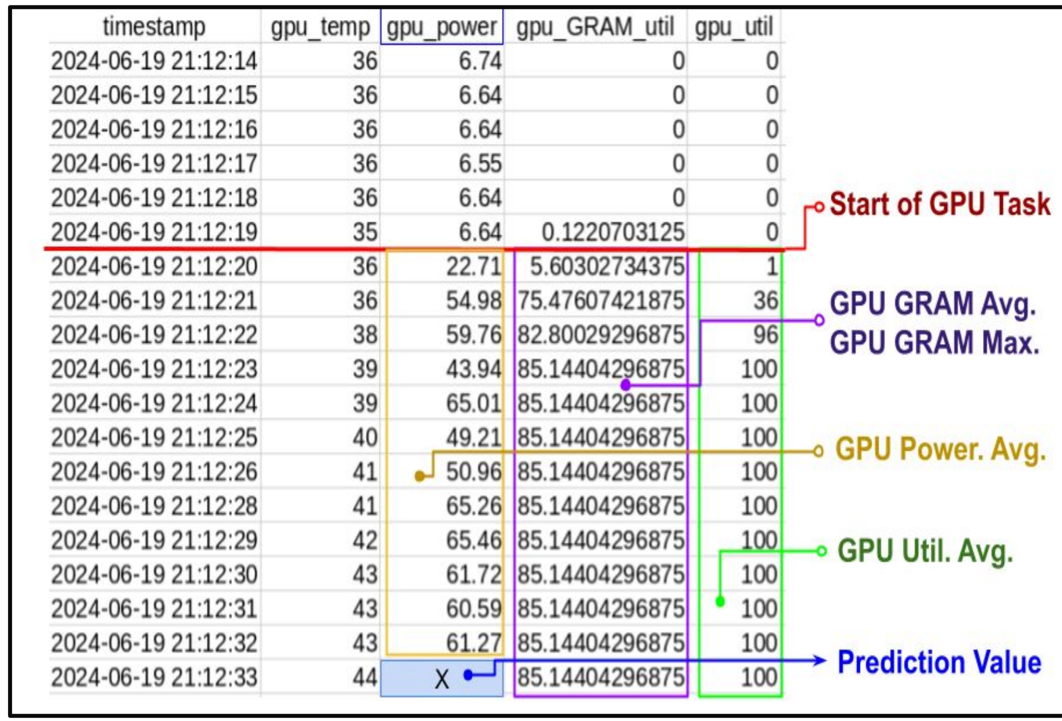
- Experiments yielded ~40 hours of second-to-second data
- Data collected with Bash scripts running in the background recording GPU and CPU metrics (e.g. temp., memory, power)

	Delay Times (seconds)	**Task Order**
Exp 1	No delays	Sequential, reverse, shuffled
Exp 2	0 - 10	shuffled
Exp 3	1 - 20	Sequential, reverse, shuffled
Exp 4	1 - 30	shuffled
Exp 5	300 - 1000	shuffled

Configurations of Experiments

Data Preprocessing

- Features extracted for duration of every task-active and idle period
  - To align with available features of real-world workloads that would be implemented in cloud simulator



timestamp	gpu_temp	gpu_power	gpu_GRAM_util	gpu_util
2024-06-19 21:12:14	36	6.74	0	0
2024-06-19 21:12:15	36	6.64	0	0
2024-06-19 21:12:16	36	6.64	0	0
2024-06-19 21:12:17	36	6.55	0	0
2024-06-19 21:12:18	36	6.64	0	0
2024-06-19 21:12:19	35	6.64	0.1220703125	0
2024-06-19 21:12:20	36	22.71	5.60302734375	1
2024-06-19 21:12:21	36	54.98	75.47607421875	36
2024-06-19 21:12:22	38	59.76	82.80029296875	96
2024-06-19 21:12:23	39	43.94	85.14404296875	100
2024-06-19 21:12:24	39	65.01	85.14404296875	100
2024-06-19 21:12:25	40	49.21	85.14404296875	100
2024-06-19 21:12:26	41	50.96	85.14404296875	100
2024-06-19 21:12:28	41	65.26	85.14404296875	100
2024-06-19 21:12:29	42	65.46	85.14404296875	100
2024-06-19 21:12:30	43	61.72	85.14404296875	100
2024-06-19 21:12:31	43	60.59	85.14404296875	100
2024-06-19 21:12:32	43	61.27	85.14404296875	100
2024-06-19 21:12:33	44	X	85.14404296875	100

Data Preprocessing Method on GPU Data

Machine Learning Algorithms

- XGBoost (eXtreme Gradient Boosting)
  - Tree boosting
  - Notable use in winning Kaggle competitions
  - High accuracy in predicting CPU temp. & GPU energy
- CatBoost (Categorical Boosting)
  - Gradient boosting
  - High accuracy in GPU energy prediction
- LightGBM (Light Gradient-Boosting Machine)
  - Gradient boosting decision tree

- Regression showed undesirable results (RMSE = ~5.0)

Machine Learning Algorithms

- Training Features:
  - GPU GRAM Avg. [GiB]
  - GPU GRAM Max. [GiB]
  - GPU Utilization Avg. [%]

	CatBoost	LightGBM	XGBoost
RMSE Value	1.218	1.224	1.284

Initial ML training results

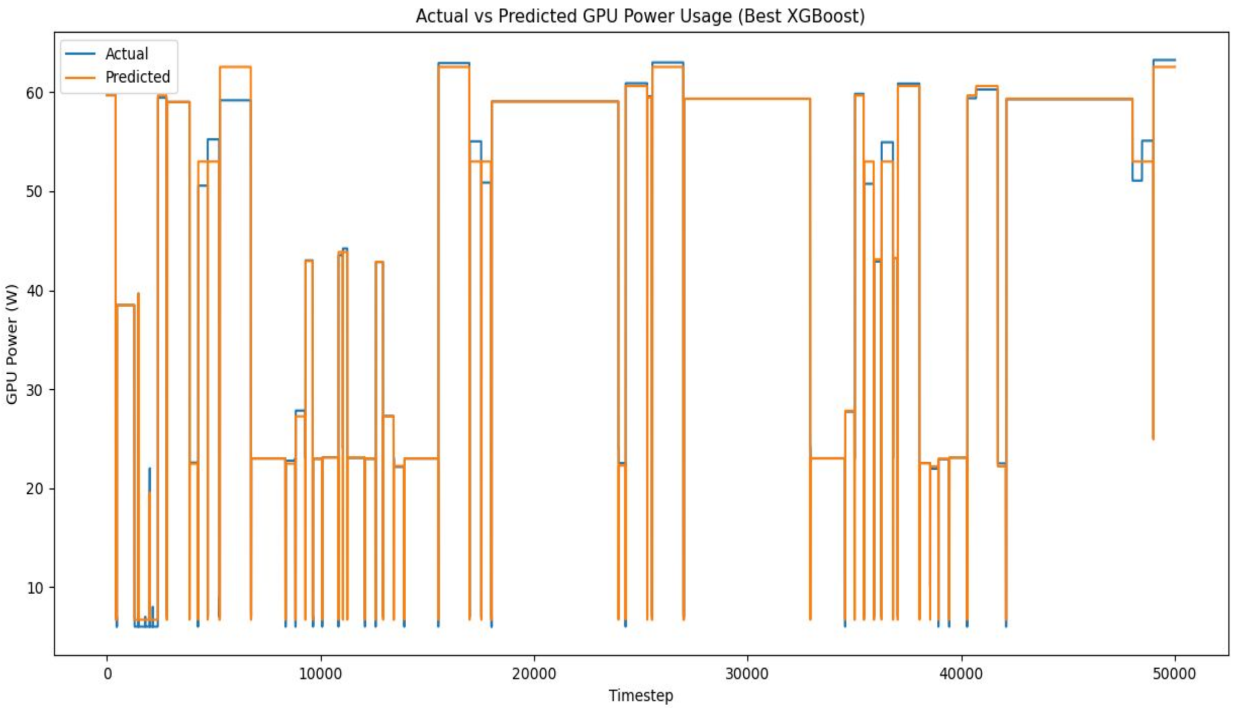
- Grid search hyperparameter tuning performed
  - Goal of improving RMSE

Model Performance

Best performing model - **XGBoost**

RMSE = 1.217

RMSE improved ~5.2% after hyperparameter tuning



Best XGBoost Model Predicted vs. Actual GPU Consumption

Next Steps

Alibaba v2020 (5 GBs)

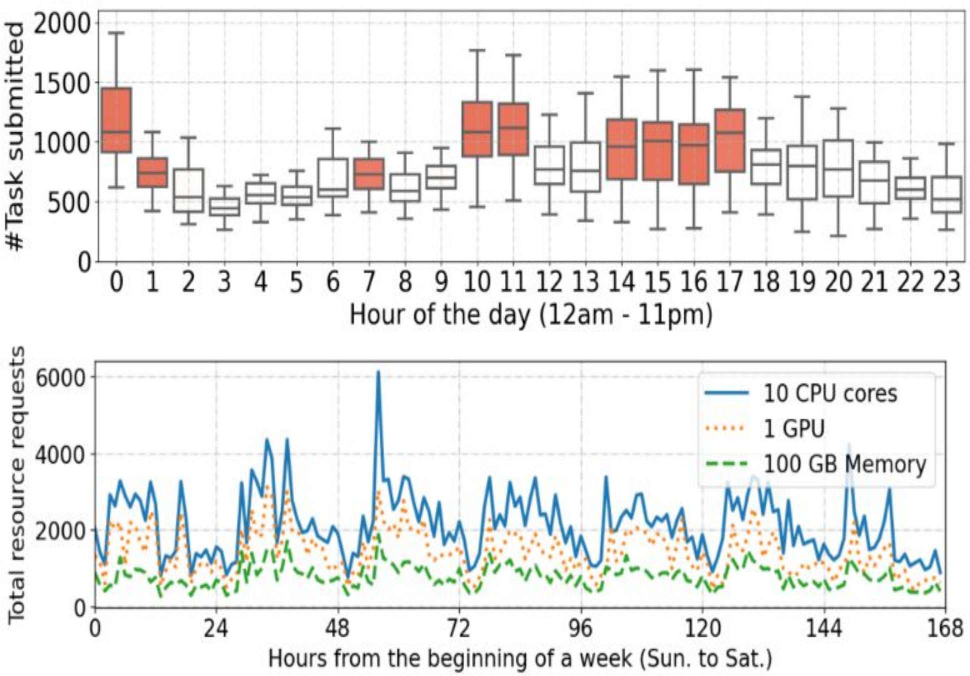
- We used 4 files: machine spec, task, instance, and sensor

Filtered out tasks missing:

- Planned VM resources
- Sensor, server information
- Task duration

Over 66 days

- 497 hosts, 325k VMs/tasks



Number of tasks submitted over a day and total resource requests over a week

Summary

- Developed a GPU power prediction to utilize task-average metrics collected from GPU
- Processed Alibaba v2020 cluster trace data to integrate into GPUCloudSimPlus
- To further our study of performance-aware load balancing algorithms
  - Aimed toward energy efficiency in datacenters

Future Work

- Integrate new GPU power model into modified GPUCloudSimPlus
- Model energy for servers using multi-CPUs and multi-GPUs
- Rewrite load balancing algorithms
- Run full duration simulations

Acknowledgement

I would like to thank:

- **Matthew Smith** for their collaboration and work on the GPUCloudSimPlus simulator & TAGpu
- **Rachel Finley** for their foundation work on ML tasks and development of GPU applications
- **Dr. Xunfei Jiang** for their guidance and mentorship throughout this work

- **SfS²** for their funding on this work

SfS²
SECURE for Student Success

Thank You

Questions?

brandon.ismalej.671@my.csun.edu

CSUN® | COMPUTER SCIENCE

To Main Paper:

References

- M. Smith, L. Zhao, J. Cordova, X.-F. Jiang, and M. Ebrahimi. “Machine Learning-based Energy-efficient Workload Management for Data Centers”. 2024 IEEE 21st Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 2024, pp. 799-802, doi: 10.1109/CCNC51664.2024.10454842.
- M. Smith, L. Zhao, J. Cordova, X.-F. Jiang, and M. Ebrahimi. "Energy-efficient GPU-intensive Workload Scheduling for Data Centers". 2023 Symposium for Undergraduate Research in Data Science, Systems, and Security (REU Symposium 2023), Special Session at the 22nd International Conference on Machine Learning and Applications (ICMLA 2023), Dec 15-17, 2023, Jacksonville, FL, USA, 2023, pp. 1735-1740, doi: 10.1109/ICMLA58977.2023.00263.
- I. Nisce, X. Jiang and S. P. Vishnu, "Machine Learning based Thermal Prediction for Energy-efficient Cloud Computing," 2023 IEEE 20th Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 2023, pp. 624-627, doi: 10.1109/CCNC51644.2023.10060079.

- Q. Weng, W. Xiao, Y. Yu, W. Wang, C. Wang, J. He, et al., "MLaaS in the wild: Workload analysis and scheduling in large-scale heterogeneous GPU clusters", *19th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 22)*, 2022.
- Q. Hu, P. Sun, S. Yan, Y. Wen, and T. Zhang, “Characterization and prediction of deep learning workloads in large-scale gpu datacenters,” in Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, ser. SC ’21. New York, NY, USA: Association for Computing Machinery, Nov. 2021, p. 1–15. [Online]. Available: https://dl.acm.org/doi/10.1145/3458817.3476223

References

- E. Masanet, A. Shehabi, N. Lei, S. Smith, and J. Koomey, “Recalibrating global data center energy-use estimates,” Science, vol. 367, no. 6481, p. 984–986, Feb. 2020.
- J. Ni, B. Jin, B. Zhang, and X. Wang, “Simulation of Thermal Distribution and Airflow for Efficient Energy Consumption in a Small Data Centers,” *Sustainability*, vol. 9, no. 4, p. 664, Apr. 2017, doi: https://doi.org/10.3390/su9040664.
- fazilbtopal, “Introduction to Generative Adversarial Networks,” *Kaggle.com*, Jun. 21, 2019. https://www.kaggle.com/code/fazilbtopal/introduction-to-generative-adversarial-networks (accessed Mar. 20, 2025).
- Sreenivas Sremath Tirumala, Seyed Reza Shahamiri, Abhimanyu Singh Garhwal, and R. Wang, “Speaker identification features extraction methods: A systematic review,” *Expert Systems with Applications*, vol. 90, pp. 250–271, Aug. 2017, doi: https://doi.org/10.1016/j.eswa.2017.08.015.