Machine Learning-based GPU Energy Prediction for Workload Management in Datacenters

SECURE for Student Success

Brandon Ismalej, Matthew Smith, Dr. Xunfei Jiang (Faculty Advisor)

Department of Computer Science

CALIFORNIA STATE UNIVERSITY NORTHRIDGE

Introduction

Motivation: Data Centers (DCs) play a crucial role in today's digital economy, accounting for ~1% of global energy usage. As the demand for cloud computing and GPU hardware increases for high-performance computing (HPC), the need for energy efficient solutions has grown. Current research lacks integration of real-world workload traces in the prediction of GPU power consumption to advance the study of workload management methods.

Overview: In this work, a machine learning model was developed to predict GPU power consumption using synthetic data that mimics real-world workload traces. The model is designed for integration into a modified version of GPUCloudSim Plus to further the study of load-balancing methods for energy efficient computing.

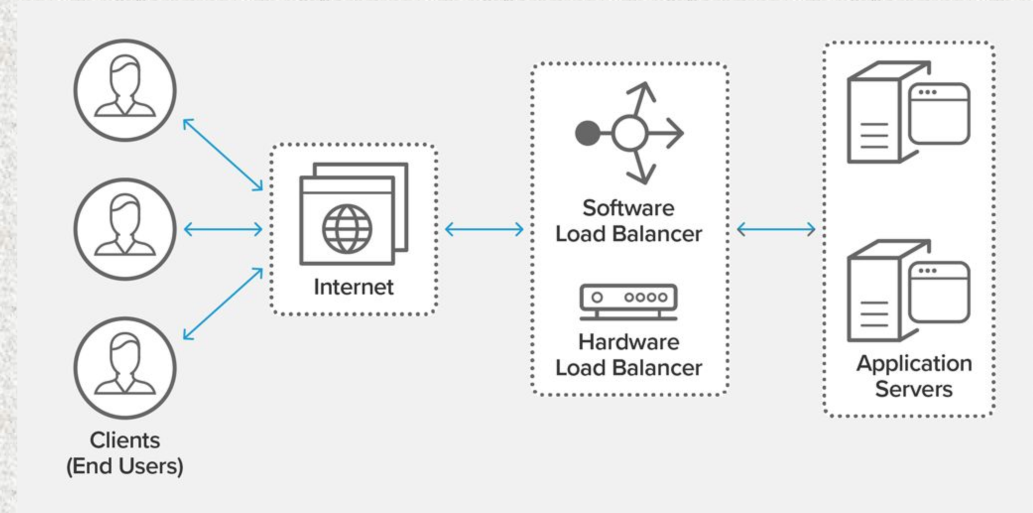


Figure 1: The process of load balancing from client to server [1].

Load Balancing

Load balancing is how a **network distributes tasks** amongst its servers. Software load balancers (**load balancing algorithms**) are adaptable to real-time data center info.

Important questions:

- Can we **reduce energy** in data centers by using **thermal-aware load balancers**?
- How would new load balancers compare to existing ones in research?
- How can we test our load balancers on a **real data center**?

GPU Power Modeling

Statistical Analysis of Real-World Workload Traces

- We train the GPU power machine learning (ML) model using data that mimics real-world operational environments
- Statistical analysis of real workload traces provides insights of real-world inter-task delay times.
- Heavily left-skewed data shows most inter-task delays times of 0 seconds, we illustrate the distribution up to the 95th percentile in Figure 2.

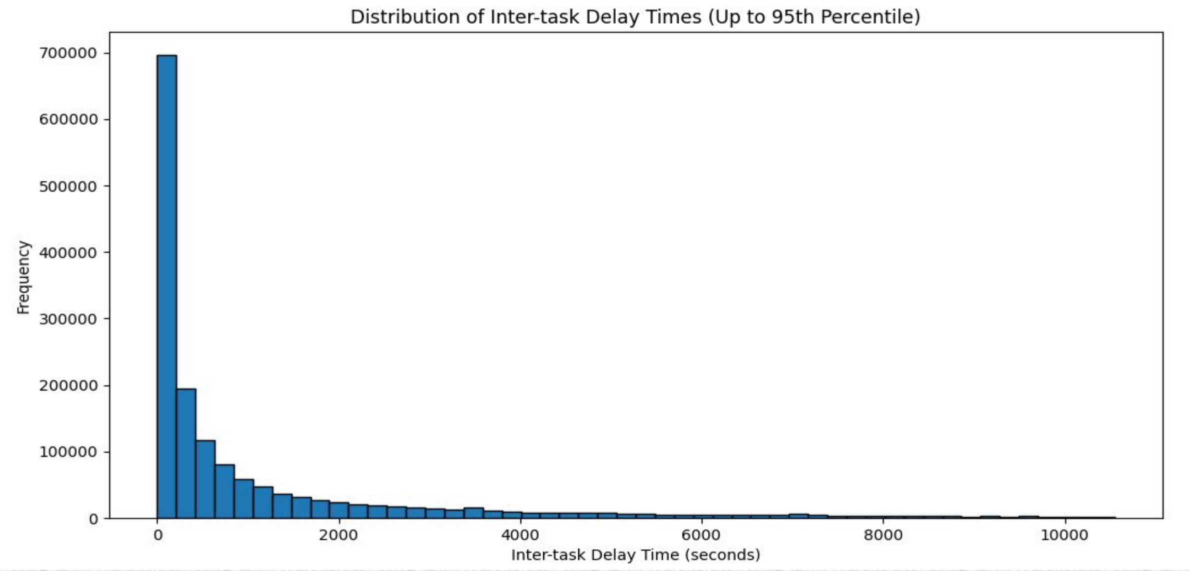


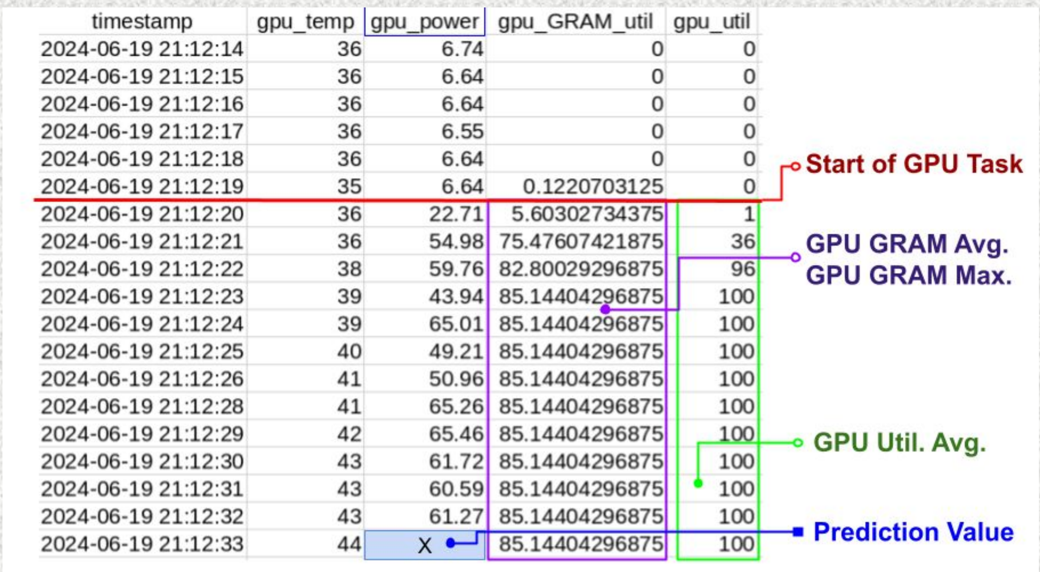
Figure 2: Positive Inter-Task Delay Times of Alibaba GPU Cluster Trace

Table 1: Experiment Configurations

	Inter-Task Delay Time (s)	Task Order
Exp 1	No delays	In order; Reverse order; Shuffled
Exp 2	0 - 10	Completely shuffled
Exp 3	1 - 20	In order; Reverse order; Shuffled
Exp 4	1 - 30	Completely shuffled
Exp 5	300 - 1000	Completely shuffled

Experimentation & Data Synthesis

- Utilizing analyzed inter-task delays, we designed 5 experiments (shown in Table I), synthesizing approximately 40 hours of GPU data.
- Simulation of GPU-intensive tasks in randomized order, with varied delay times, running from 0s to 1000s emulates a robust and varied real-world workload scenario
- GPU-intensive tasks:
  - Matrix Multiplication, Generative Adversarial Network, Natural Language Processing, Unsupervised Learning



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

Figure 3: Data Preprocessing - Detection of Task Execution to Calculate Average Metrics

Data Preprocessing

- The training data was preprocessed to match the features available in the Alibaba workload trace
- Throughout the duration of a GPU task, GPU GRAM [GiB] and GPU utilization [%] is averaged, and the maximum GPU GRAM [GiB] is extracted, which make up model training features.

$$Power_{avg} = f(Util_{avg}, GRAM_{avg}, GRAM_{max})$$

Equation 1: Input and Output Features of Model Training

Model Development

- GPU power prediction can be modeled by Eq. [1] for any given temp step.
- Average and maximum values of critical metrics ensures models do not learn simplistic patterns and encourages generalization of workloads.
- Training XGBoost, CatBoost, LightGBM, & LSTM models
- 60/20/20 data split for training/testing/validation.
- Performing hyperparameter tuning
- Best performance model: XGBoost ML model RMSE = 1.217.

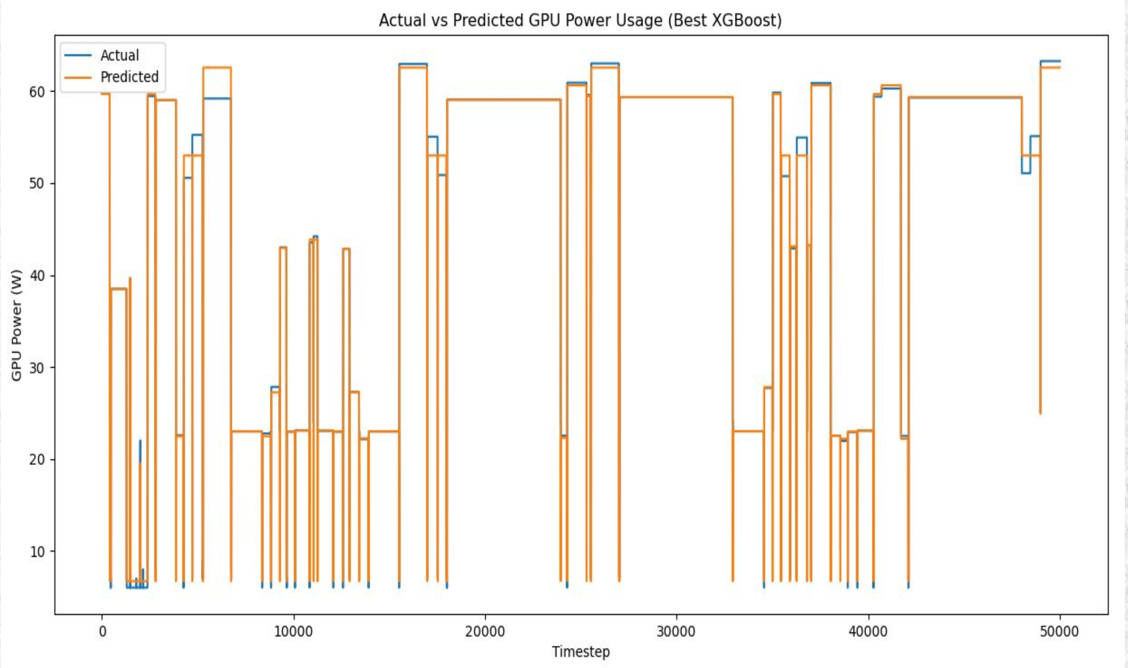


Figure 4: Best XGBoost Model - Predicted vs. Actual GPU Power Consumption

max_depth	4
learning_rate	0.1
n_estimators	1000
min_child_weight	1
subsample	0.6
colsample_bytree	1.0

Table 2: Optimal XGBoost Hyperparameters

Alibaba GPU Workload Trace

The Alibaba v2020 trace contains information about **jobs** to the **Alibaba GPU data center** over **two months**. We processed four data files from the trace:

- **Machine spec** - server configuration
- **Task** - VM allocated resources
- **Instance** - program execution: start time, end time, machine executed on
- **Sensor** - program execution: utilization metrics

We parsed **~350K instances** from the 1M instances on **T4 GPUs**.

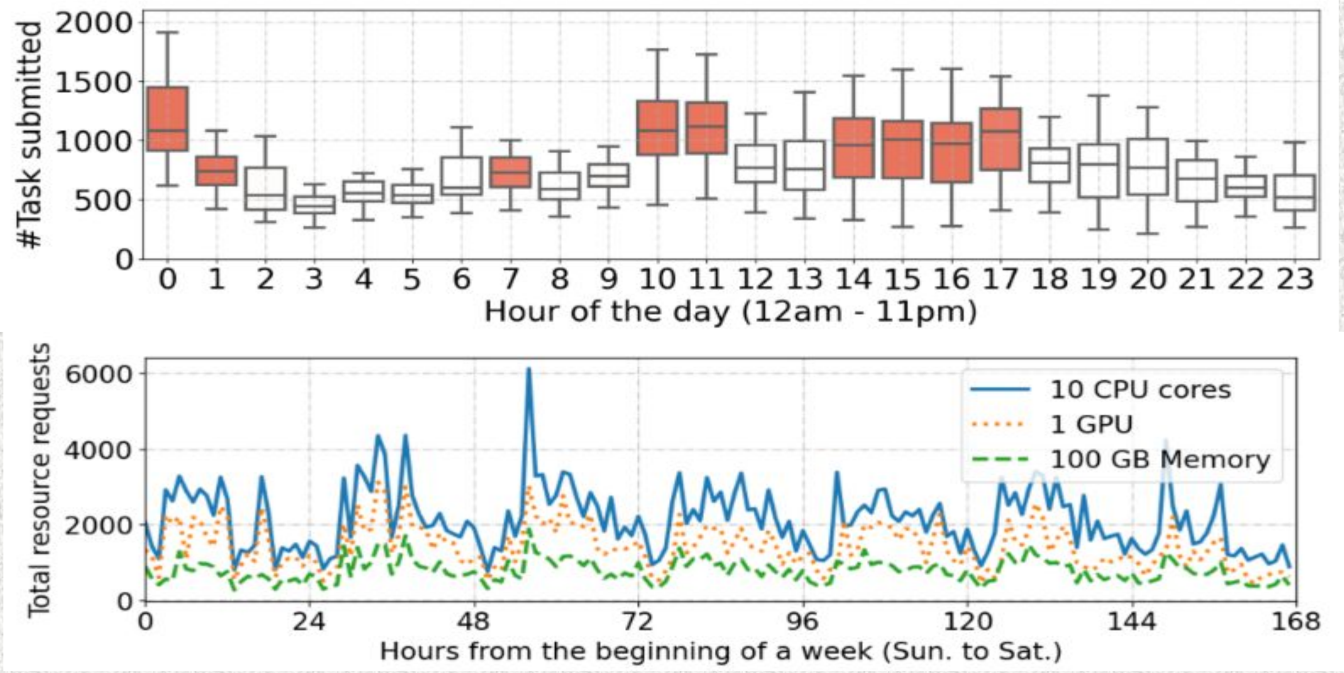


Figure 5: Number of tasks submitted in a day and resource requests over a week [2].

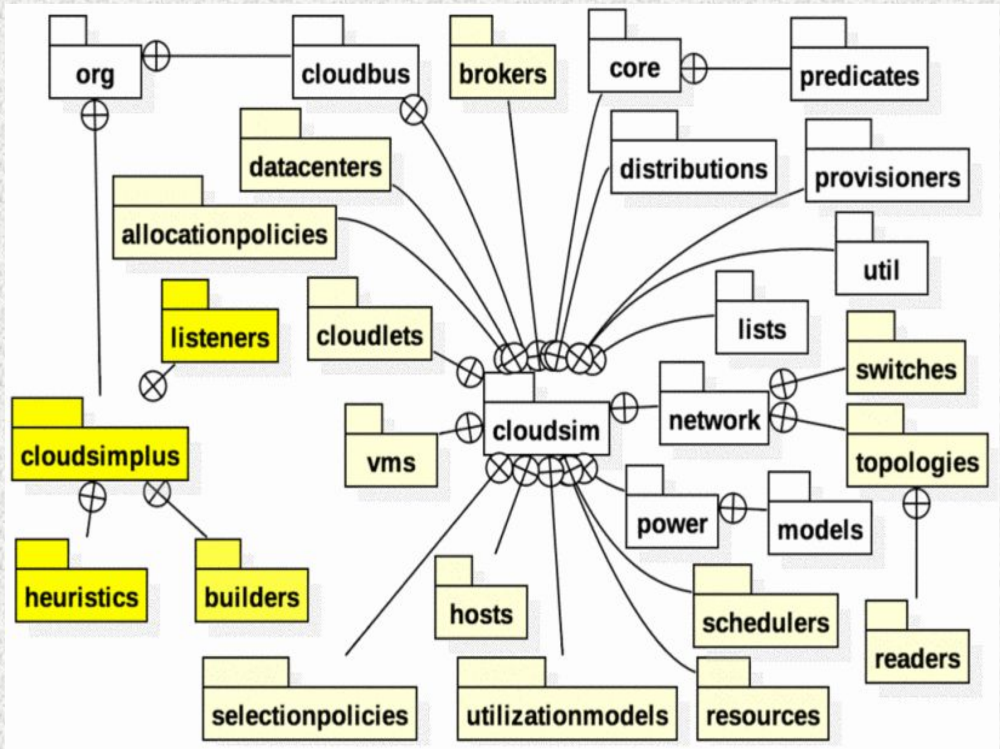


Figure 6: The package structure of (GPU)CloudSim Plus.

GPUCloudSim Plus

GPUCloudSim Plus is a **cloud computing simulator** that can model servers, virtual machines, and programs run in data centers on both the CPU and GPU. We use it to **simulate load balancing algorithms** and evaluate their impact on the energy consumption of data centers.

The most important classes for our work in GPUCloudSim Plus include:

- **Host –** machine in Alibaba trace
- **Vm** – task in Alibaba trace
- **Cloudlet –** instance and sensor in Alibaba trace
- **VmAllocationPolicy** – the load balancing algorithm

Adapting our Previous Research

- Configure the simulation to **use the Alibaba v2020 trace**
- Register VMs/cloudlets at appropriate simulation time
- Submit events to the simulator for resource utilization
- Adapting load balancers to run on VMs instead of cloudlets
- **Optimizing simulation** runtime with **ML batch predictions**
- Optimizing simulation runtime with **ML GPU acceleration**

Next Steps for Obtaining Results

- Utilize new GPU power model into simulations
- Support multi-CPU and multi-GPU server energy models
- Train ML GPU temp. model like new GPU power model
- Evaluate tasks that do not use both CPU and GPU:
  - **CPU only tasks**
  - **GPU only tasks**

References

[1] NGINX. *What Is Load Balancing?* URL: https://www.nginx.com/resources/glossary/load-balancing/

[2] Q. Weng et al., "{MLaaS} in the Wild: Workload Analysis and Scheduling in {Large-Scale} Heterogeneous {GPU} Clusters," presented at the 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22), 2022, pp. 945–960. Available: https://www.usenix.org/conference/nsdi22/presentation/weng

[3] Manoel C. Silva Filho et al. "CloudSim Plus: A cloud computing simulation framework pursuing software engineering principles for improved modularity, extensibility and correctness". In: 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM). 2017, pp. 400–406. DOI: 10.23919/INM.2017.7987304.

Acknowledgement

•Supported by the SfS[2] Program and funded by the United States Department of Education FY 2023 Title V, Part A, Developing Hispanic-Serving Institutions Program five-year grant, Award Number P31S0230232, CFDA Number 84.031S.

•However, the contents of this presentation do not necessarily represent the policy of the US Department of Education, and you should not assume endorsement by the Federal Government.