

Vrije Universiteit Amsterdam
<https://www.overleaf.com/read/mbzddrhpbvr#e96bec>



Honours Programme, Problem Statement

Precision Capacity Planning: Simulating the Cost-Energy Trade-offs of LLM Training on Homogeneous Clusters

Author: Thijs van den Heuvel (2849429)

1st supervisor: Prof. Dr. Ir. Cav. Alexandru Iosup
daily supervisor: MSc. Dante Niewenhuis

*A report submitted in fulfillment of the requirements for the Honours Programme,
which is an excellence annotation to the VU Bachelor of Science degree in
Computer Science/Artificial Intelligence/Information Sciences
version 1.0*

December 10, 2025

Abstract

The training of Large Language Models (LLMs) has evolved into a massive industrial operation, where single training runs are costing tens of millions of dollars and consuming gigawatt-hours of energy. As models scale from billions to trillions of parameters, the financial and environmental sustainability of AI is becoming a critical and global challenge. Datacenters training these kinds of massive LLMs rely currently on a "good enough" human judgment or massive over-provisioning to manage these workloads, leading to significant inefficiencies in capital and energy use. Simulation offers a way to optimize these systems without the cost of physical experimentation. However, state-of-the-art simulators like OpenDC currently lack the specialized workload models required to accurately capture the non-functional properties of modern LLM training loops, such as gradient synchronization latency and checkpointing overheads.

This report proposes to bridge this gap by developing a parametric LLM training workload model for OpenDC. We aim to simulate the cost and energy trade-offs of training on homogeneous clusters, enabling datacenter operators to identify the best possible configuration where performance is maximized before diminishing returns and excessive carbon emissions set in.

Keywords

Large Language Models, LLMs, Simulation, Datacenter, Cost-Energy

1 Introduction

Artificial Intelligence (AI) has shifted from a niche research field to a primary driver of global datacenter demand. This AI era has introduced a new class of workloads, like the training of Large Language Models (LLMs). These type of workloads are not short-lived inference tasks or traditional web services, but they are synchronous, monolithic, and resource-intensive training jobs that occupies thousands of GPUs for months at a time [3, 9]. Estimating the carbon footprint of models like BLOOM reveals that training alone can emit over 50 tonnes of CO₂[7]. Efficiency in these systems is not just a performance metric, but a necessity for both financial and environmental sustainability.

Training is estimated to account for 10-40% of the total energy consumed by an LLM in its lifetime[2]. This is substantial and optimizing this to reduce even a few percent would cut costs and emissions. Discrete-event simulators could get every bit of performance out of a system resulting into millions of dollars in operational savings and significant reduction in carbon emissions. These simulators Act as a critical lever in this optimization, enabling companies to identify these efficiency gains without the capital risk of physical experimentation.

Datacenter configurations where all compute nodes (GPUs) are identical, an example being 1000 NVIDIA H100s, is the current industrial standard for training Frontier Models to minimize the "straggler effect" and synchronization latency found in heterogeneous environments. Even in these heterogeneous clusters, there is still a lot of time lost on

synchronization. A simple background process can stall thousands of others during gradient synchronization[6]. Optimizing LLM training is notoriously difficult because physical experimentation is expensive. Operators often over-provision hardware to ensure stability, avoiding the risk of potential downtime but leading to massive resource waste[11]. Current simulators also fail to model the "system-level" components of training, specifically Checkpointing mechanisms (saving model states to disk in case of failure) and gradient synchronization (network communication between clusters). Recent surveys insidate that 30-40% of training time can be lost due to these overheads if not optimized, but standard simulators still treat them as negligible[12, 13].

Key prior work relies on the "Scaling Laws" of Neural Networks[4], which which predict the raw compute requirements for training models based on their size. To meet these demands, industry has adopted distributed training systems like Megatron-LM[9], which split the workload across thousands of GPUs. In the field of simulation, tools like OpenDC[8] and ASTRA-sim have successfully modeled general cloud and network behaviors. Still, a gap remains: recent surveys[1] and production traces[5] show that current simulators fail to account for the "system-level" friction of training, specifically the massive overheads of checkpointing and synchronization. This is a critical gap, as existing tools cannot accurately predict the true energy and financial impact of large-scale training workloads.

So this leads us to the main research question: **How can we utilize discrete-event simulation to determine the cost-optimal and energy-optimal homogeneous cluster configuration for training Large Language Models?** This will become the first open-source LLM Training Workload Model for OpenDC that specifically accounts for reliability overheads like checkpointing and gradient synchronization. Allowing researchers to study "Green AI" infrastructure without needing access to supercomputers to do these experiments. This project will contribute to the scientific community by providing a new tool for simulating LLM training, enabling more efficient and sustainable AI development. It will also help me develop my skills in simulation, distributed systems, and AI, preparing me for a career in this rapidly evolving field.

2 Background

2.1 The lifecycle of Large Language Models: Training vs Inference

It is crucial to distinguish between the two primary phases of an LLM's lifecycle: Training and Inference.

Inference is the process of generating a response from a trained model. It is stateless, highly parallelizable, and typically consumes a negligible amount of energy per request. Text classification for example consumes only 0.002Wh per request[2].

Training is the focus of this research, involving a stateful, monolithic workload that is iterating billions of parameters across thousands of GPUs. While inference accounts for the majority of accumulated energy usage over years of deployment, training is the most capital and resource intensive phase per unit of time. Training for GPT-4 is estimated to have an energy consumption of 51,772 to 62,318 MWh[2]. This is the equivalent to around 5,000 to 6,000 average American households' annual electricity consumption. Optimization of this phase is critical because unlike inference, which can be distributed across edge

devices, training requires a concentrated, high-performance datacenter environment where inefficiencies scale with cluster size.

2.2 Homogeneous Cluster and 3D Parallelism

To manage the computational scale of LLM training, the industry has standardized on 3D parallelism. This is a hybrid strategy that combines Data, Tensor and Pipeline parallelism to partition models across thousands of GPUs[9]. To support this synchronization, operators almost exclusively deploy Homogeneous Clusters, where all the compute nodes are identical. Homogeneity is enforced to minimize the "straggler effect", where slower nodes delay the entire training job due to faster nodes having to wait for slower nodes to finish[5]. However, even in these homogeneous environments, recent studies using production traces have shown that hardware homogeneity does not guarantee performance homogeneity. Software-level inefficiencies, such as uneven pipeline partitioning or pauses from garbage collection, can still cause a significant synchronization delay across nodes. These "software stragglers" can cause up to 45% of allocated GPU resources to be wasted, even in clusters with identical hardware specification[5].

2.3 Checkpointing and Fault Tolerance

Training runs take a long time, often spanning weeks or months. During this period, hardware failures are statistically inevitable. To ensure fault tolerance, training systems implement checkpointing: The process of periodically freezing computation to save the model's parameters and optimizer states to persistent storage (SSDs). While essential, this process introduces significant Checkpointing Overhead. Standard asynchronous checkpointing mechanisms can stall training over 1.3 seconds per save operation[13]. In a large-scale cluster where saving occurs frequently to mitigate failure risks, these stalls represent a massive efficiency loss. Techniques like "Gradient-Assisted Checkpointing" attempt to reduce this stall time to sub-second levels[13], but their impact on total energy consumption at the scale of 10,000+ GPUs has not yet been simulated.

2.4 Discrete-Event Simulation (OpenDC)

The problem with analyzing these inefficiencies on real hardware is the cost. One cannot simply experiment with different configurations on a 10,000 GPU cluster or simply "pause" a \$50 million training run to test a new scheduling hypothesis. Discrete-Event Simulation (DES) offers a solution by modeling the system as a sequence of events in time. OpenDC[8] is a state-of-the-art DES platform that has successfully modeled cloud datacenter workloads and serverless functions. However, current simulators often treat jobs as generic units of compute duration and assume that small-scale node behavior is still correct when scaled up[1]. They lack the specific workload models required to simulate the internal dynamics of LLM training, specifically the network bursts of 3D parallelism and the I/O stalls of checkpointing, rendering them insufficient for precise capacity planning in the context of LLM training.

3 Problem

We identified a critical gap in the current simulation tools: **There is no accessible instrument to accurately predict the cost-efficiency and energy impact of LLM training configurations on homogeneous clusters.** While there is a growing body of research that has focused on Model Optimization, reducing computational complexity by up to 35% through techniques like sparse training and quantization[10], far less attention has been given to System Optimization. Even with highly compressed models, the infrastructure overheads of fault tolerance and synchronization persist. Current simulators often focus on the former (compute time) while ignoring the latter (system-level overheads), creating a blind spot in energy estimation. Specifically, we identify the following issues:

- **The Fallacy of Idealized Execution:** Existing simulators typically model training jobs under the assumption of idealized execution, treating homogeneous clusters as perfectly synchronized straggler free systems. This ignores the non-deterministic delays of 3D parallelism. As revealed from production traces, even in homogeneous clusters designed for stability, "software stragglers" (caused by uneven pipeline partitioning or garbage collection pauses) can cause the system to waste up to 45% of its allocated GPU resources[5]. Current tools fail to capture these synchronization inefficiencies, leading to overly optimistic performance and energy estimates that mask real-world efficiency losses.
- **Ignored Reliability Costs:**

4 Related Work

5 Research Question(s)

This project will focus on **How can we utilize discrete-event simulation to determine the cost-optimal and energy-optimal homogeneous cluster configurations for training Large Language Models?** We focus specifically on the Training phase of LLMs, excluding inference, as training represents the most capital-intensive and critical bottleneck for infrastructure planning. We constrain our research to Homogeneous Clusters (identical GPUs and network topology), which is the industry standard for minimizing hardware-induced latency. Within this scope, we will focus on modelling the "System-level" non-functional properties, specifically Checkpointing Overhead and Gradient Synchronization, rather than low-level chip microarchitecture. We use the OpenDC simulator as our experimental platform.

5.1 RQ1: How can we model the non-functional properties of LLM training workloads, specifically gradient synchronization latency and checkpointing I/O bursts, for discrete-event simulation

This research question aims to first abstract the complex reality of a distributed job into a smaller mathematical model that defines its key performance characteristics. We need to do this because existing simulators treat jobs as generic units of compute duration

and assume that small-scale node behavior is still correct when scaled up. This will fail to capture the massive efficiency losses from synchronization delays and checkpointing stalls that can occur at scale. This part is challenging because it first requires a deep understanding of the mechanics of the training process, and then simplifying these complex structures into a simple model without losing the key dynamics. We must accurately model the "software-stragglers" that occur even in homogeneous clusters.

5.2 RQ2: How to implement this model within the OpenDC ecosystem to accurately measure power consumption and throughput on homogeneous GPU clusters?

A theoretical model is insufficient without a practical implementation. It must be validated within a simulation engine to run scenarios at a scale and collect accurate metrics. This will provide a reusable software tool for the scientific community, allowing researchers to study "Green AI" infrastructure without needing access to expensive supercomputers. OpenDC is primarily designed for cloud workloads (stateless requests). Extending it to support stateful, long-running training jobs requires a different logic for mechanisms for checkpointing and handling the complex synchronization patterns of training.

5.3 RQ3: What is the relationship between cluster size, checkpointing frequency, and training cost efficiency

To validate the model, we will run experiments verifying its accuracy and provide the "Precision Capacity Planning" from the title. We aim to identify the optimal configuration where adding more GPUs would yield diminishing returns due to increased synchronization overheads. This will answer the core societal and economic question of "Are we wasting money and energy by building clusters that are too big?" Operators currently lack the data to make these trade-off decisions, leading to over-provisioning and resource waste. It requires designing a good experimental setup that isolates the key variables to prove causality. We must also simulate the "What-if" scenarios of techniques like Gradient-Assisted Checkpointing to measure their potential benefits at larger scale (over 10,000 GPUs) that has never been tested before in an actual cluster.

6 Approach

7 Plan

8 Conclusion

References

- [1] E. Dryden et al. A survey of end-to-end modeling for distributed dnn training: Workloads, simulators, and tco. *arXiv preprint arXiv:2506.09275*, 2025.
- [2] Z. Ji and M. Jiang. A systematic review of electricity demand for large language models: evaluations, challenges, and solutions. *Renewable and Sustainable Energy Reviews*, 225:116159, Jan. 2026.
- [3] Z. Jiang, H. Lin, Y. Zhong, Q. Huang, Y. Chen, Z. Zhang, Y. Peng, X. Li, C. Xie, S. Nong, Y. Jia, S. He, H. Chen, Z. Bai, Q. Hou, S. Yan, D. Zhou, Y. Sheng, Z. Jiang, H. Xu, H. Wei, Z. Zhang, P. Nie, L. Zou, S. Zhao, L. Xiang, Z. Liu, Z. Li, X. Jia, J. Ye, X. Jin, and X. Liu. Megascale: Scaling large language model training to more than 10, 000 gpus, 2024.
- [4] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. Scaling laws for neural language models, 2020.
- [5] J. Lin, Z. Jiang, Z. Song, S. Zhao, M. Yu, Z. Wang, C. Wang, Z. Shi, X. Shi, W. Jia, et al. Understanding stragglers in large model training using what-if analysis. *arXiv preprint arXiv:2505.05713*, 2025.
- [6] J. Lin, Z. Jiang, Z. Song, S. Zhao, M. Yu, Z. Wang, C. Wang, Z. Shi, X. Shi, W. Jia, Z. Liu, S. Wang, H. Lin, X. Liu, A. Panda, and J. Li. Understanding stragglers in large model training using what-if analysis, 2025.
- [7] A. S. Luccioni, S. Viguier, and A.-L. Ligozat. Estimating the carbon footprint of bloom, a 176b parameter language model, 2022.
- [8] F. Mastenbroek, G. Andreadis, S. Jounaid, W. Lai, J. Burley, J. Bosch, E. van Eyk, L. Versluis, V. van Beek, and A. Iosup. Opendc 2.0: Convenient modeling and simulation of emerging technologies in cloud datacenters. In *2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, pages 455–464, 2021.
- [9] D. Narayanan, M. Shoeybi, J. Casper, P. LeGresley, M. Patwary, V. A. Korthikanti, D. Vainbrand, P. Kashinkunti, J. Bernauer, B. Catanzaro, A. Phanishayee, and M. Zaharia. Efficient large-scale language model training on gpu clusters using megatron-lm, 2021.
- [10] K. R. Narsepalle. Energy-efficient training and inference in large language models: Optimizing computational and energy costs. *International Journal of Computer Applications*, 187(14):1–13, Jun 2025.
- [11] D. Patterson, J. Gonzalez, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. So, M. Texier, and J. Dean. Carbon emissions and large neural network training, 2021.
- [12] W. Xu, X. Huang, S. Meng, W. Zhang, L. Guo, and K. Sato. An efficient checkpointing system for large machine learning model training. In *Workshops of the International Conference on High Performance Computing, Network, Storage, and Analysis (SC-W)*, pages 896–900, 11 2024.

- [13] K. Zhang, Y. Chen, Z. Hu, W. Lin, J. Xu, and W. Chen. Gockpt: Gradient-assisted multi-step overlapped checkpointing for efficient llm training, 2025.