# I/O Patterns and Bottlenecks in Deep Learning Workloads

Pablo Alessandro Hugen[1]

[1]Institute of Informatics – UFRGS

Comp. Sys. Perf. Analysis 2025/2

# Content

# Context

- Recent growing interest in optimizations for Machine Learning/Deep Learning training and inference methods.
- Used in various fields: LLMs, Image reconition and classifications, and so on.
- Large models often need very large HPC infraestructures for processing the insane amount of training data.
- The performance of the storage and I/O subsystem of HPC systems is critical

# Context

- Traditional HPC workloads are characterized by large, sequential data access (PAUL; KARIMI; WANG, 2021).
  - Simulations which saves the state at the end or in checkpoints
- In contrast, ML workloads generate small, random reads across numerous files (PAUL; KARIMI; WANG, 2021).
- Large amounts of data (far greater than system memory) + random read pattern = lot of page faults and cache misses (VERY BAD)

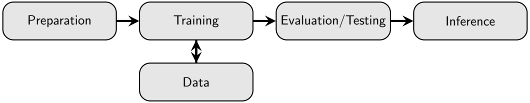
# Context

**Table 2.2** Example Time Scale of System Latencies

| Event | Latency | Scaled |
|---|---|---|
| 1 CPU cycle | 0.3 ns | 1 s |
| Level 1 cache access | 0.9 ns | 3 s |
| Level 2 cache access | 2.8 ns | 9 s |
| Level 3 cache access | 12.9 ns | 43 s |
| Main memory access (DRAM, from CPU) | 120 ns | 6 min |
| Solid-state disk I/O (flash memory) | 50–150 μs | 2–6 days |
| Rotational disk I/O | 1–10 ms | 1–12 months |
| Internet: San Francisco to New York | 40 ms | 4 years |
| Internet: San Francisco to United Kingdom | 81 ms | 8 years |
| Internet: San Francisco to Australia | 183 ms | 19 years |
| TCP packet retransmit | 1–3 s | 105–317 years |
| OS virtualization system reboot | 4 s | 423 years |
| SCSI command time-out | 30 s | 3 millennia |
| Hardware (HW) virtualization system reboot | 40 s | 4 millennia |
| Physical system reboot | 5 m | 32 millennia |

# Context

- At Large Scale Distributed DL Workloads, IO can take roughly 85% of the *training* time (DRYDEN et al., 2021).
- And training is often one of the most expensive parts of the pipeline (LEWIS; BEZ; BYNA, 2025).



# Objectives

General:
Understand **patterns in I/O operations and possible bottlenecks** in common Machine Learning workloads

Especifics:

- **Disk throughput**: Understand how disk throughput varies in training between epochs, checkpoints and when the number of training processes varies.
- **GPU usage**: Know how the GPU usage (%) behaves in those scenarios.

# Justification

- By understanding those patterns and possible bottlenecks I hope to **find some ideas and directions for further work**.

# Related Work

### Lewis, Bez e Byna (2025)

Surveys literature from 2019 to 2024 on the I/O challenges, patterns, and optimizations for machine learning applications on high-performance computing systems to identify gaps for future research.

### Párraga et al. (2021)

This paper presents a methodology for analyzing the input/output (I/O) patterns of deep learning applications on high-performance computing (HPC) systems, applying it to codes using TensorFlow2 and PyTorch with the MNIST and CIFAR-10 datasets to understand performance bottlenecks.

# Related Work

## Gainaru et al. (2022)

This paper discusses the complex I/O patterns and challenges of emerging machine learning workflows used for large-scale scientific data analysis, proposes methods to optimize data transfers, and demonstrates performance gains with a medical application case study.

## Paul, Karimi e Wang (2021)

This paper presents an in-depth I/O characterization of over 23,000 machine learning jobs from a one-year period on the Summit supercomputer, using the Darshan tracing tool to analyze how their behavior varies across different scientific domains and workload scales.

# Content

# Methodology (so far)

- Simulation: dlio_benchmark (DEVARAJAN et al., 2021).
- The experiments are available at ⟨https://github.com/HpcResearchLaboratory/perf_2025⟩.
  - Custom workloads
  - dlio_benchmark (python venv :(, will try to use nix shell )
  - Slurm script to perform all benchmarks

# Methodology (so far)

## System

Table: System Specifications

| **Component** | **Specification** |
|---|---|
| CPU | 4x ARM Neoverse-V2 (288 cores total) |
| Memory (RAM) | 857 GiB |
| GPU | 4x NVIDIA GH200 (120GB each) |
| Storage | 1.8 TB NVME |

# Methodology (so far)

## Workloads

Table: Workflow Configuration Summary

| Model Name | Framework Name | Epochs | Checkpoints Enabled |
|---|---|---|---|
| cosmoflow_h100_custom | tensorflow | 1 | No |
| default_custom | pytorch | 10 | No |
| dlrm_custom | pytorch | 3 | Yes (every 2 steps) |
| unet3d_h100_custom | pytorch | 5 | Yes (after epoch 5, then every 2) |

# Methodology (so far)

Experimental project

- **Input variables:** # of epochs, # of processes
- **Response variables:** Accelerator Usage (AU), I/O Throughput

# Methodology (so far)

Table: Experimental setup

| **Model** | **Epochs** | **Procs.** | **Run** |
|---|---|---|---|
| cosmoflow | 1 | 1 | N |
| cosmoflow | 1 | 2 | N |
| cosmoflow | 1 | 4 | N |
| cosmoflow | 1 | 6 | N |
| cosmoflow | 1 | 8 | N |
| default | 10 | 1 | Y |
| default | 10 | 2 | Y |
| default | 10 | 4 | Y |
| default | 10 | 6 | Y |
| default | 10 | 8 | Y |
| dlrm | 3 | 1 | Y |
| dlrm | 3 | 2 | Y |
| dlrm | 3 | 4 | Y |
| dlrm | 3 | 6 | N |
| dlrm | 3 | 8 | N |
| unet3d | 5 | 1 | Y |
| unet3d | 5 | 2 | Y |
| unet3d | 5 | 4 | Y |
| unet3d | 5 | 6 | Y |
| unet3d | 5 | 8 | Y |

# Content

# Accelerator Usage vs. Number of Processes

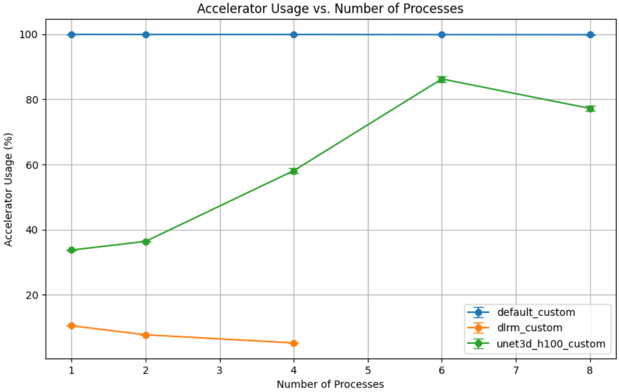



Figure: Accelerator usage for different models and number of processes.

# I/O Throughput vs. Number of Processes

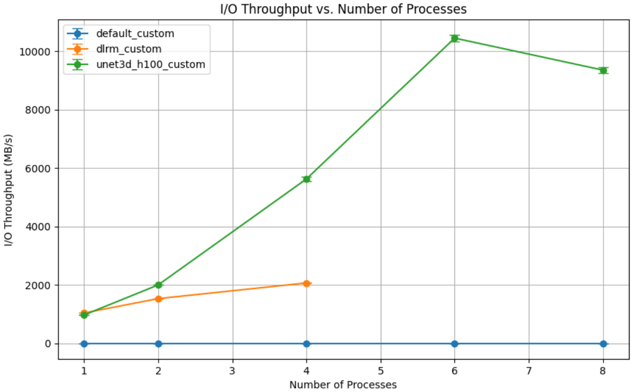



Figure: I/O throughput for different models and number of processes.

# Bibliography I


DEVARAJAN, H. et al. Dlio: A data-centric benchmark for scientific deep learning applications. In: *2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*. [S.l.: s.n.], 2021. p. 81–91.

DRYDEN, N. et al. Clairvoyant prefetching for distributed machine learning i/o. In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. New York, NY, USA: Association for Computing Machinery, 2021. (SC '21). ISBN 9781450384421. Disponível em: ⟨https://doi.org/10.1145/3458817.3476181⟩.

GAINARU, A. et al. Understanding and leveraging the i/o patterns of emerging machine learning analytics. In: NICHOLS, J. et al. (Ed.). *Driving Scientific and Engineering Discoveries Through the Integration of Experiment, Big Data, and Modeling and Simulation*. Cham: Springer International Publishing, 2022. p. 119–138. ISBN 978-3-030-96498-6.

# Bibliography II


LEWIS, N.; BEZ, J. L.; BYNA, S. I/o in machine learning applications on hpc systems: A 360-degree survey. *ACM Comput. Surv.*, Association for Computing Machinery, New York, NY, USA, v. 57, n. 10, maio 2025. ISSN 0360-0300. Disponível em: ⟨https://doi.org/10.1145/3722215⟩.

PÁRRAGA, E. et al. Analyzing the i/o patterns of deep learning applications. In: NAIOUF, M. et al. (Ed.). *Cloud Computing, Big Data & Emerging Topics*. Cham: Springer International Publishing, 2021. p. 3–16. ISBN 978-3-030-84825-5.

PAUL, A. K.; KARIMI, A. M.; WANG, F. Characterizing machine learning i/o workloads on leadership scale hpc systems. In: *2021 29th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*. [S.l.: s.n.], 2021. p. 1–8.