Throughput Performance Benchmarking: Pre-Training Foundational Large Language Models on Kubernetes

Ronen Dar
March, 2024

Agenda

01
Intro

02
LLM training – a look under the hood

03
Benchmarking results

04
Demo

Ronen Dar

Co-Founder & CTO, Run:ai

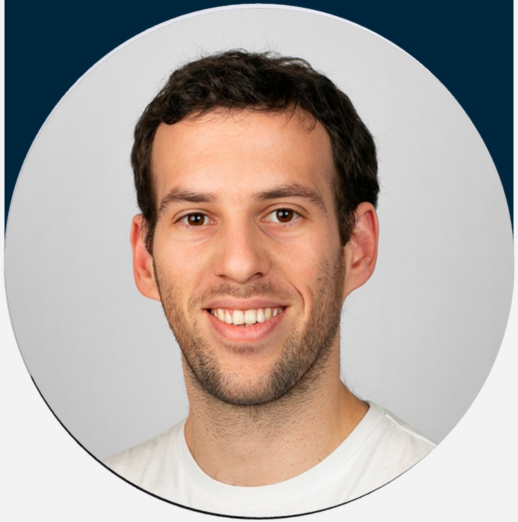
- Lives near Tel Aviv, in Israel
- PhD & Postdoc in Information Theory, background in Chip Startups
- Since 2018, Co-Founder & CTO at Run:ai
- **Run:ai** – AI Infrastructure Orchestration Platform

Certified for NVIDIA SuperPODs

Raz Rotenberg

Director of Engineering, Run:ai

- Lives near Tel Aviv, in Israel
- Engineering group responsible for advanced GPU provisioning capabilities in Kubernetes and LLM training and deployment
- Since 2018 with Run:ai



Blog

Introducing ChatGPT

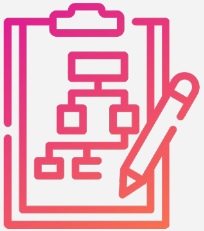
We’ve trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests.

Try ChatGPT ↗ Read about ChatGPT Plus

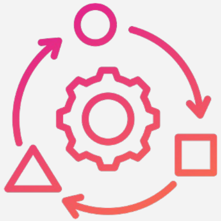
November 30, 2022

Authors
OpenAI ↓

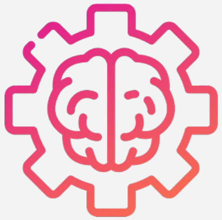
Types of AI Initiatives



Prompt Engineering
(+RAG)



Fine Tuning



Training From Scratch

Why organizations fine-tune or train models from scratch?

Control training datasets

Adjust the model to specific use cases or to proprietary data

Reduce costs

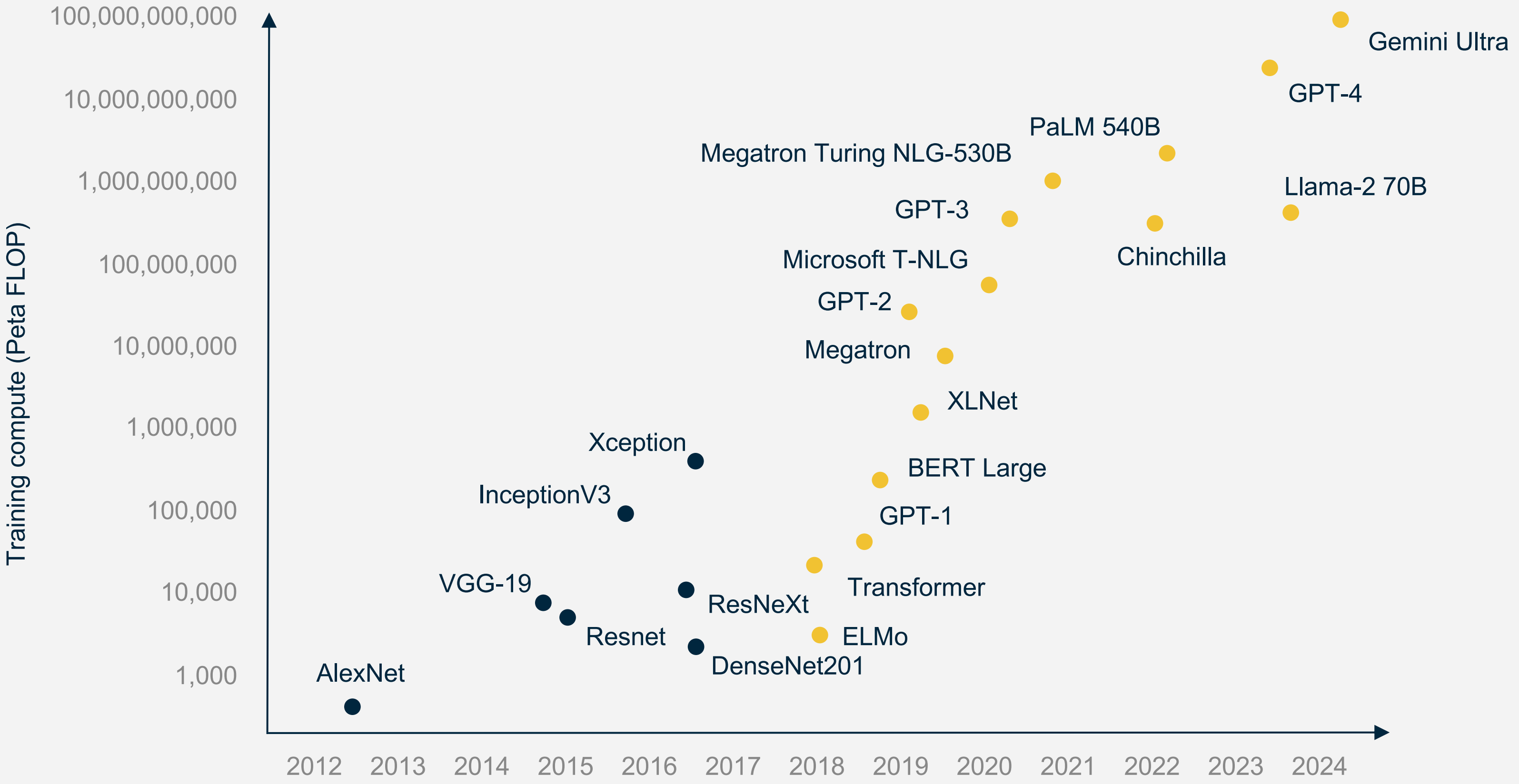
Control IP

Why organizations fine-tune or train models from scratch?

Bad News
Training complexity has increased significantly in the LLM era

Good News
New tools and frameworks simplify and abstract that complexity

Explosive growth in model size



LLMs don't fit into a single GPU

	Weights	Memory requirements for training	GPU requirement s for training
Llama-2 7b float32	**~28GB**	**~98GB**	**4 GPUs** (assuming 40GB GPU Memory)
Llama-2 13b float32	**~52GB**	**~192GB**	**8 GPUs** (assuming 40GB GPU Memory)
Llama-2 70b float32	**~280GB**	**~1TB**	**32 GPUs** (assuming 40GB GPU Memory)

* According to Hugging Face Memory Calculator, assuming Adam Optimizer

LLMs don’t fit into a single GPU

	Weights	Memory requirements for training	GPU requirement s for training
Llama-2 7b float16	~14GB	~49GB	2 GPUs (assuming 40GB GPU Memory)
Llama-2 13b float16	~26GB	~96GB	4 GPUs (assuming 40GB GPU Memory)
Llama-2 70b float16	~140GB	~512GB	16 GPUs (assuming 40GB GPU Memory)

* According to Hugging Face Memory Calculator, assuming Adam Optimizer

LLMs don’t fit into a single GPU

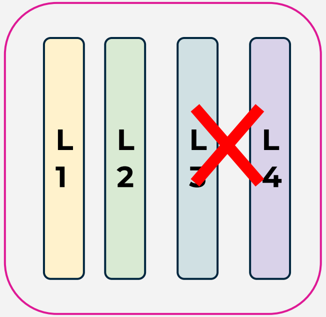
	Weights	Memory requirements for training	GPU requirement s for training
Llama-2 7b int8	**~7GB**	**~25GB**	**1 GPUs** (assuming 40GB GPU Memory)
Llama-2 13b int8	**~13GB**	**~48GB**	**2 GPUs** (assuming 40GB GPU Memory)
Llama-2 70b int8	**~70GB**	**~256GB**	**8 GPUs** (assuming 40GB GPU Memory)

* According to Hugging Face Memory Calculator, assuming Adam Optimizer

LLMs don’t fit into a single GPU

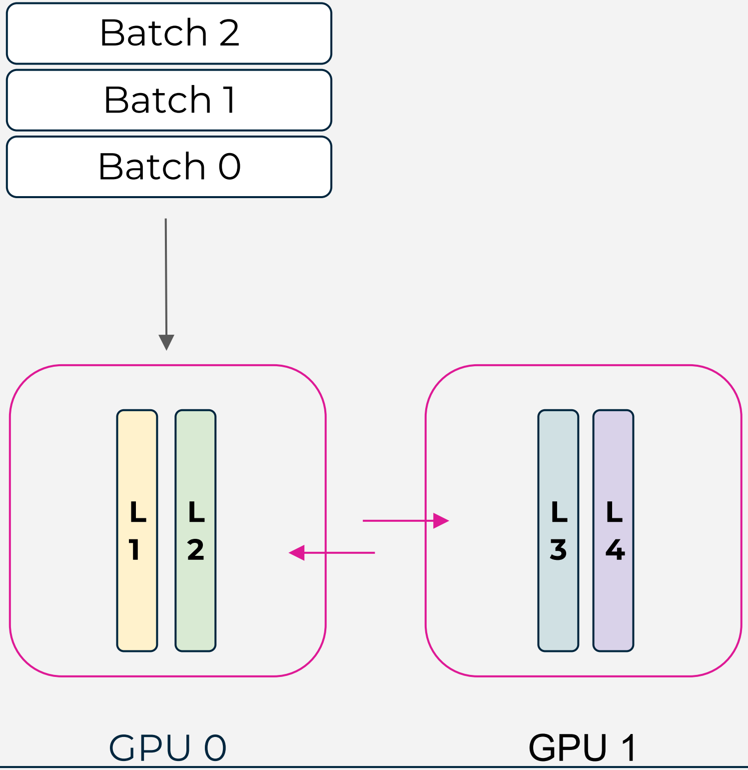
- Quantization usually comes with a significant accuracy degradation
- Training with mixed 16/32 bit precision can keep reasonable tradeoff between accuracy and memory reduction

Parallelism strategies

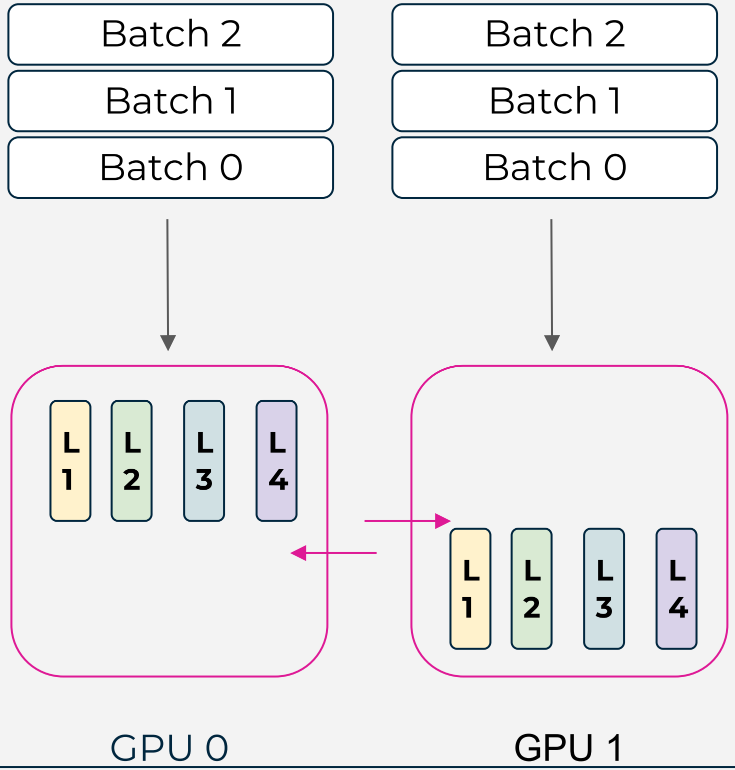


GPU 0

Parallelism strategies – Pipeline parallelism



Parallelism strategies – Tensor parallelism



More advanced parallelism strategies

Pipeline parallelism

Offloading memory to CPU

Model Parallelism

3D parallelism

Tensor + pipeline + data parallelism

Zero Redundancy Optimizer (ZeRO)

Fully Sharded Data Parallelism (FSDP)

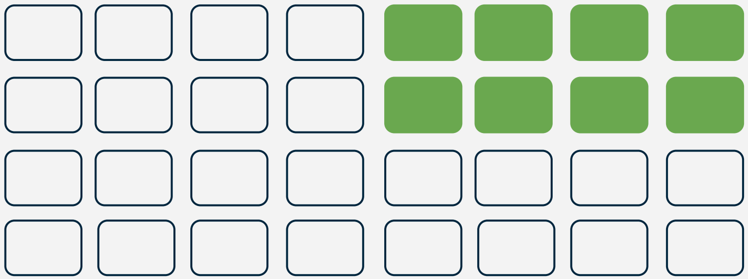
Parallelism Strategies for Distributed Training
https://www.run.ai/blog/parallelism-strategies-for-distributed-training

Good news – full software stack for large scale training

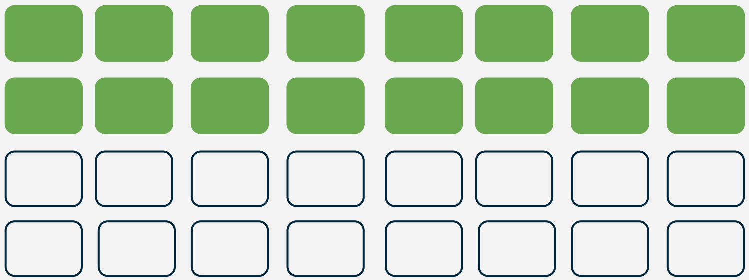
Model-Specific Libraries	NVIDIA NeMo Framework
High-level Interface	Pytorch Lightning
Parallelism Libraries	DeepSpeed, FairScale
Deep Learning Framework	Pytorch
Communication Backend	MPI, NCCL, Gloo

What it takes to train large models on shared AI clusters

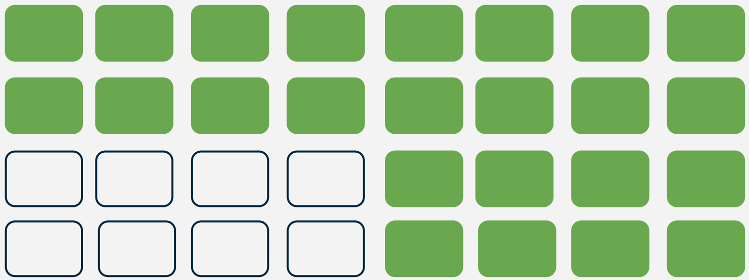
Scaling up a single training job



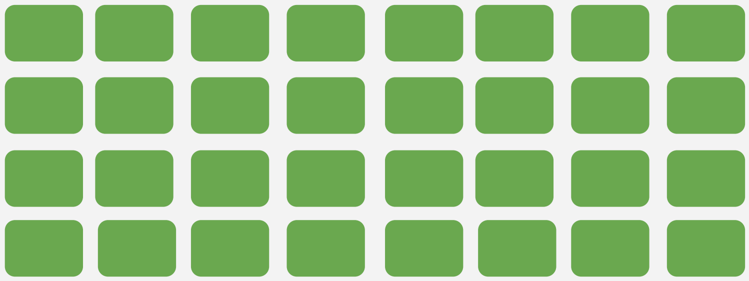
Scaling up a single training job



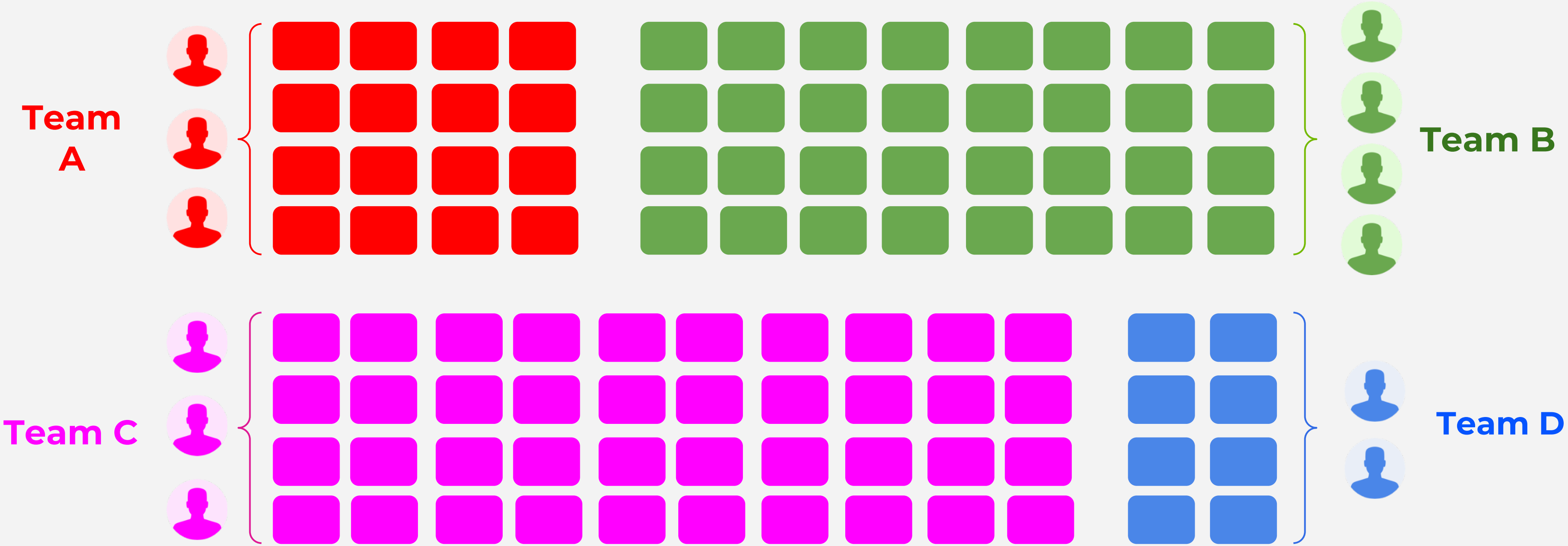
Scaling up a single training job



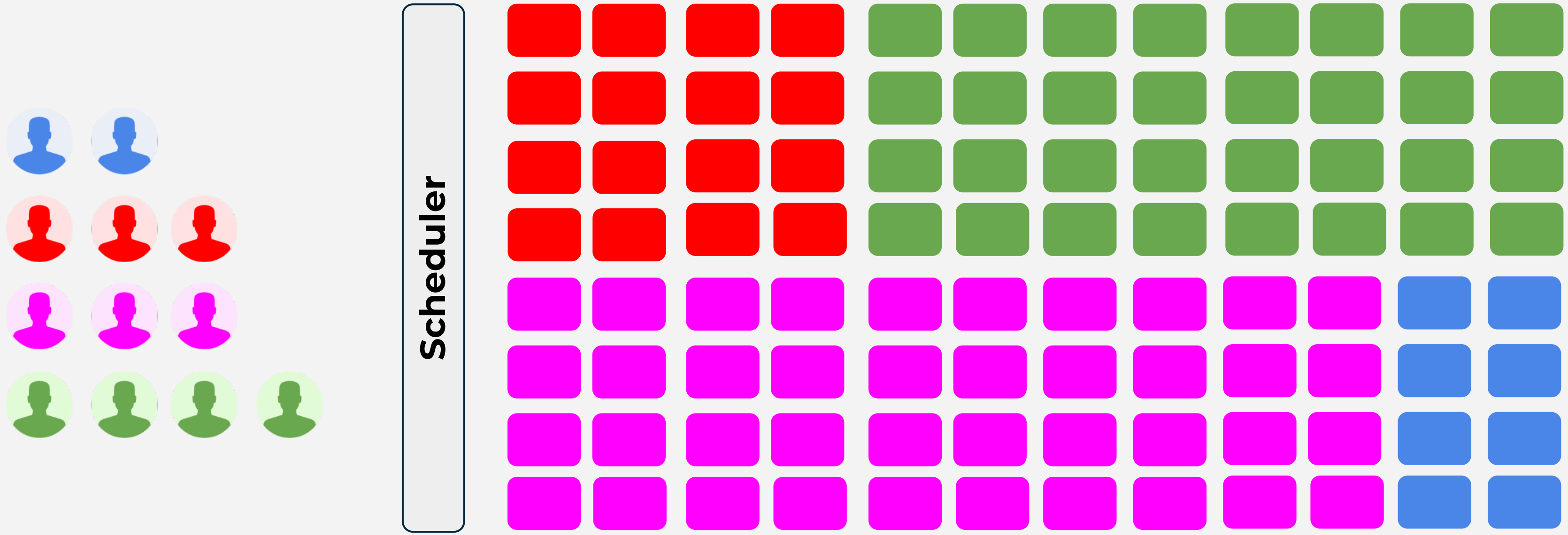
Scaling up a single training job



From an organizational point of view



GPU Pooling + Schedulers

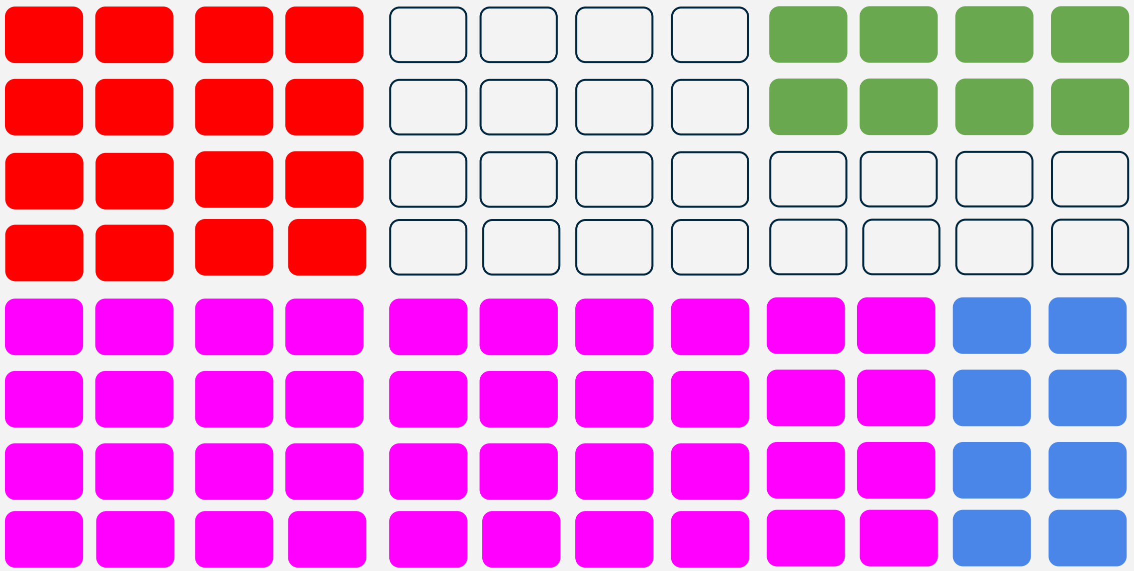


GPU Pooling – from siloed AI to collaborative efforts

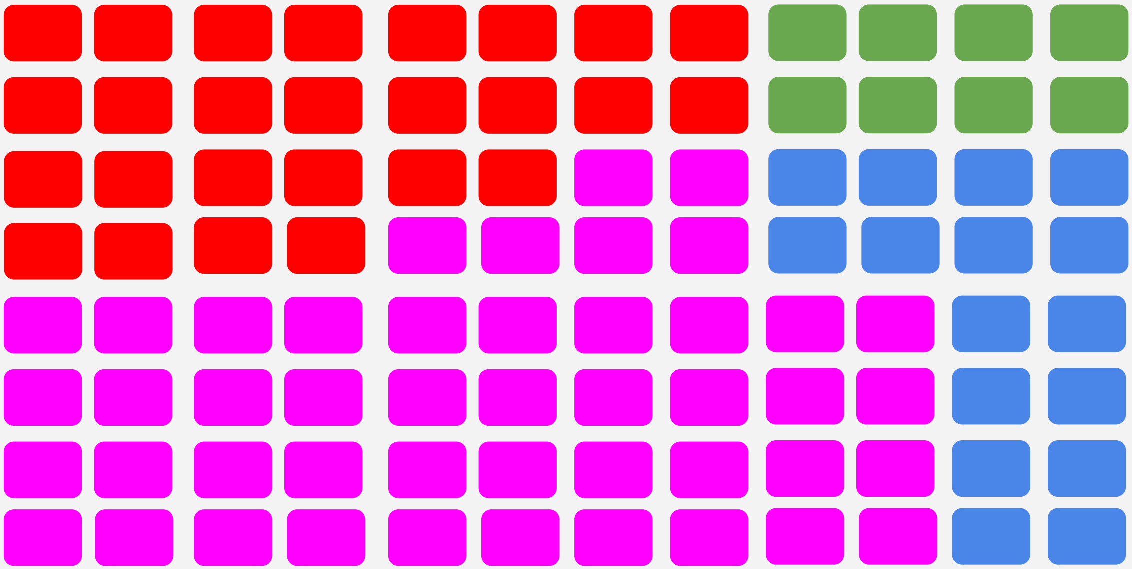
Siloed Infrastructure → **Shared Clusters**

On-Demand Compute → **Reserved Clusters**

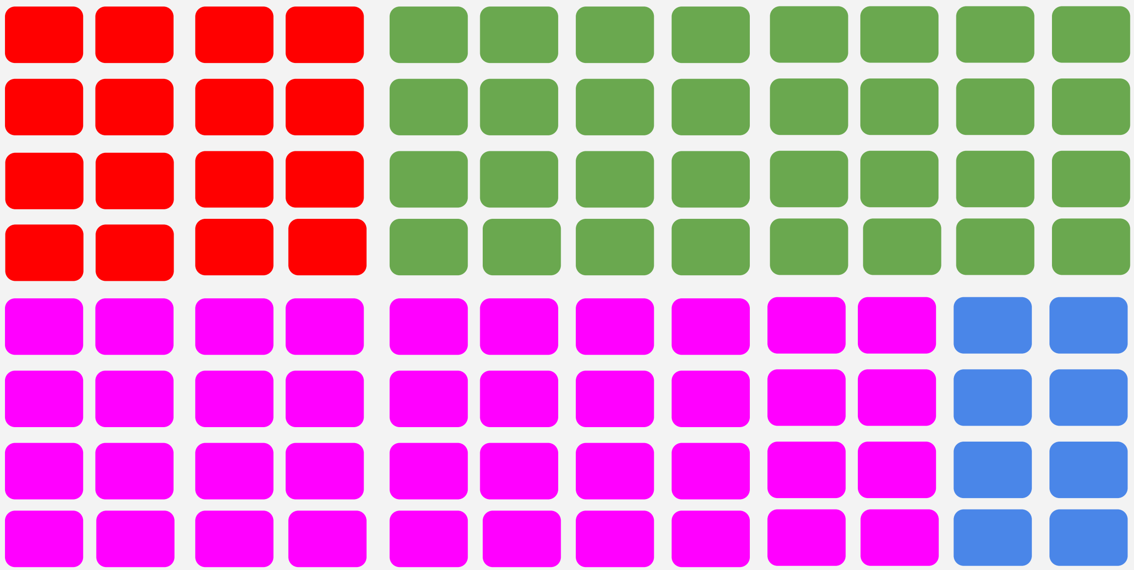
Repurposing resources between different **teams**



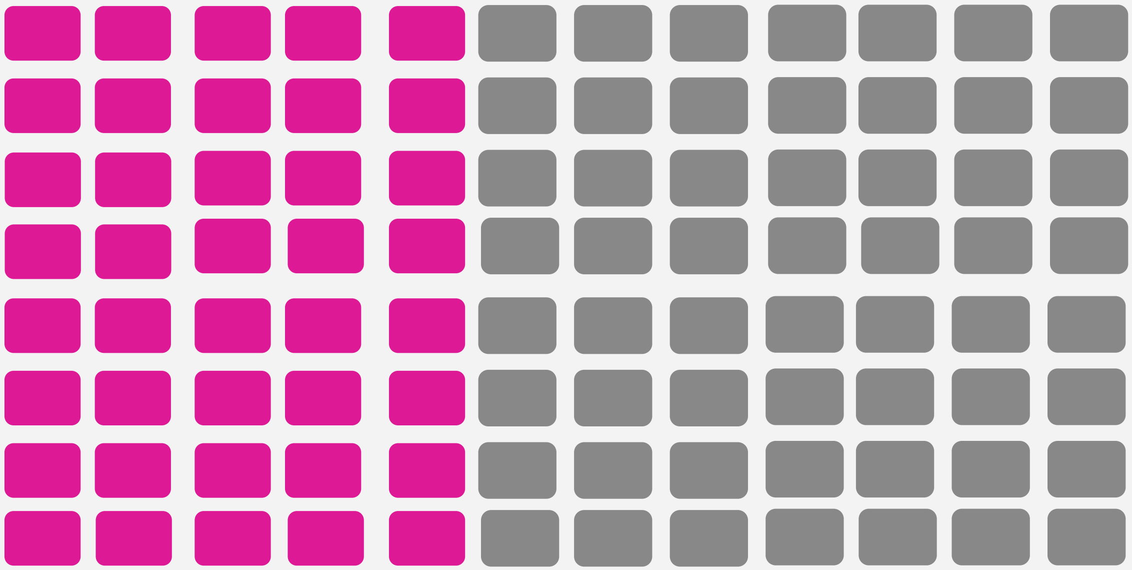
Repurposing resources between different **teams**



Repurposing resources between different **teams**



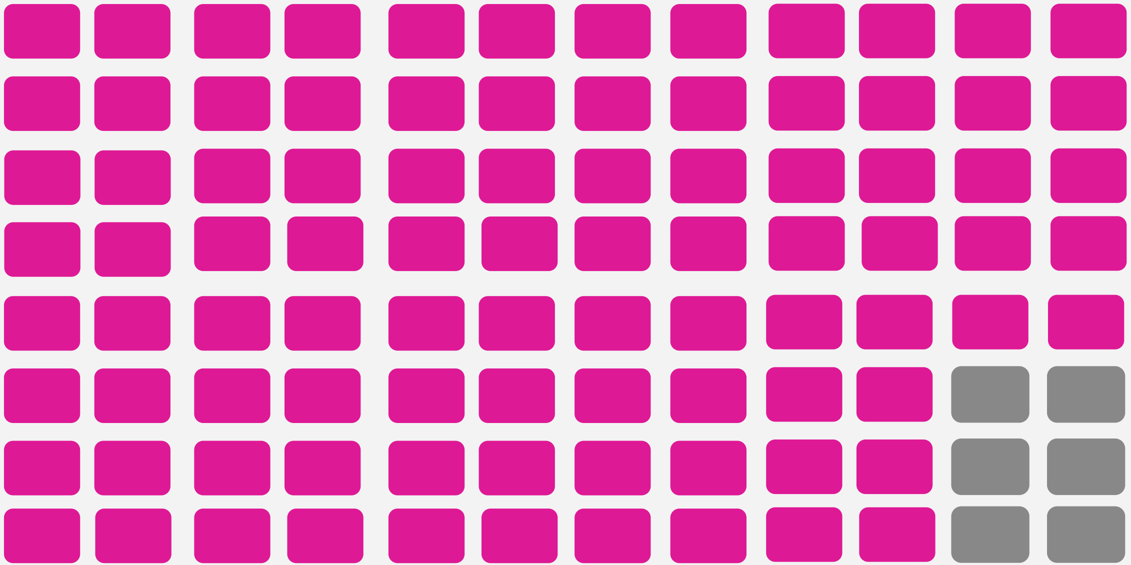
Repurposing resources between different **workloads**



Training

Inference @ day

Repurposing resources between different **workloads**



Training

Inference @ night

Benefits

Higher Efficiency

Through sharing and repurposing resources

More GPU Accessibility

Users become more productive with easier access to more GPUs

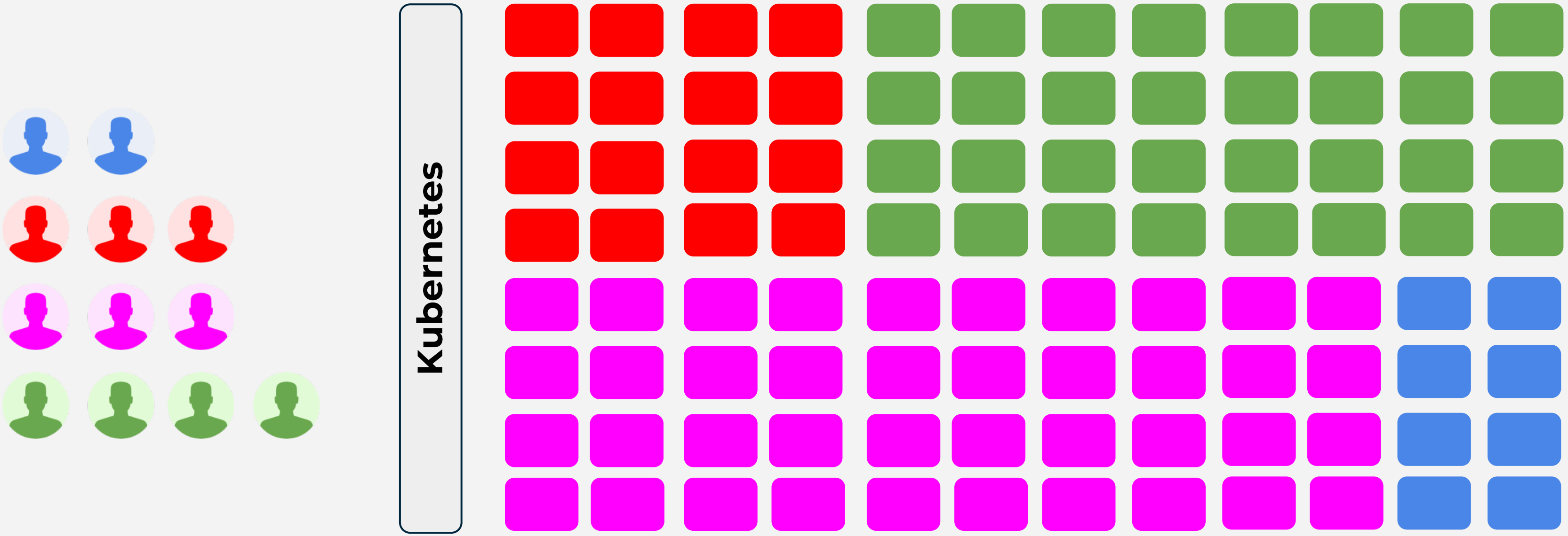
Controls & Governance

Ability to align resources with business goals

Centralized Visibility

Better planning and decision making

Kubernetes as the orchestration layer



Training benchmarking

Infrastructure setup

- 4 x NVIDIA DGX A100-80GB Nodes, with a total of 32 x NVIDIA A100 Tensor Core GPUs
- 8 x 200 Gb HDR NVIDIA InfiniBand connectivity per node

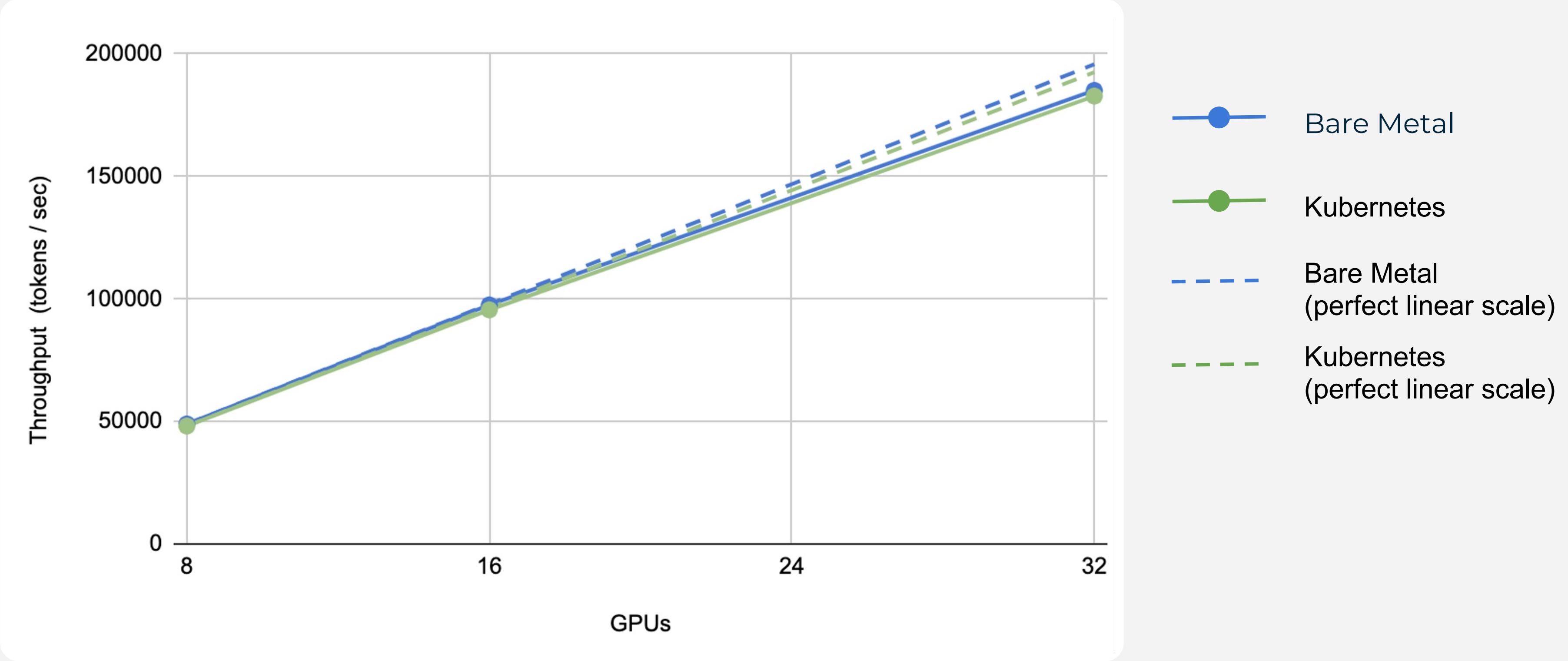
Kubernetes with the following components

- NVIDIA GPU Operator
- NVIDIA Network Operator
- Kubeflow Training Operator

Training benchmarking: GPT-3 / 5B parameters

	1 Node (8 GPUs)	**2 Nodes** (16 GPUs)	**4 Nodes** (32 GPUs)
Bare metal (tokens / Sec)	48941	97541	185090
Kubernetes (tokens / Sec)	48131	95545	182791
Diff.	**1.65%**	**2.05%**	**1.24%**

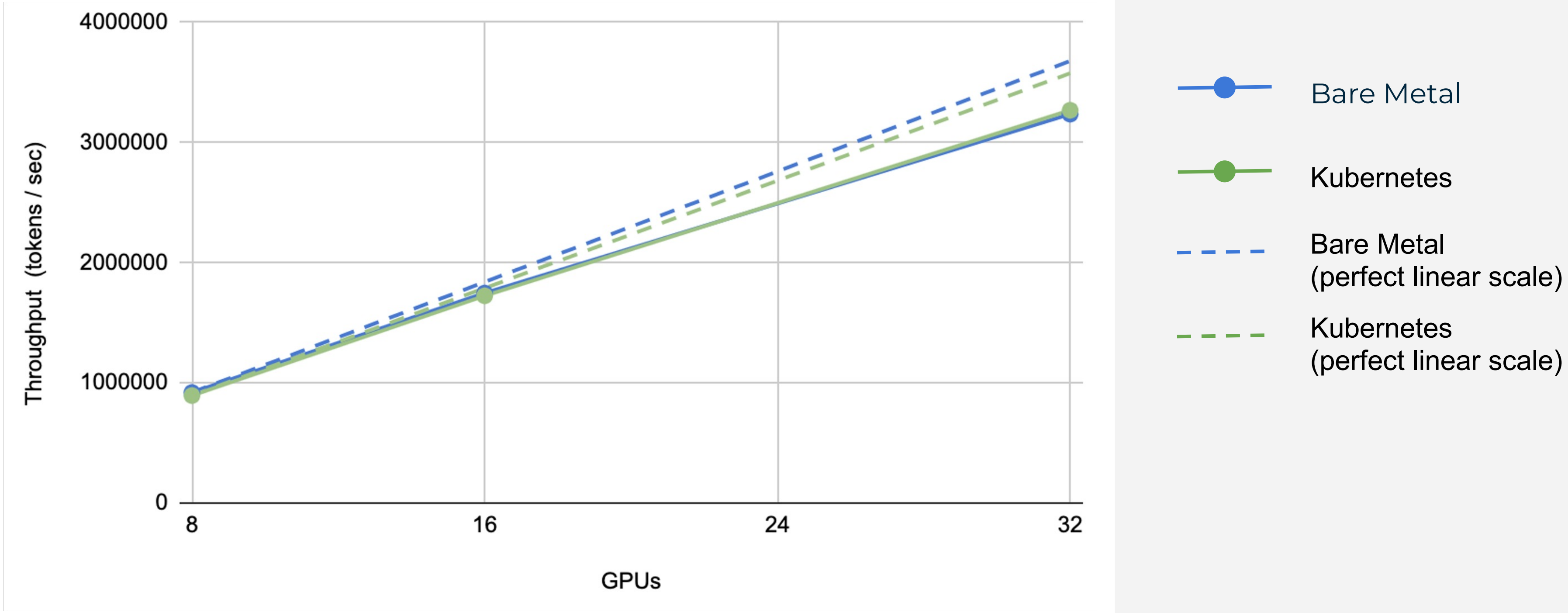
Training benchmarking: GPT-3 / 5B parameters



Training benchmarking: GPT-3 / 126M parameters

	1 Node (8 GPUs)	**2 Nodes** (16 GPUs)	**4 Nodes** (32 GPUs)
Bare metal (tokens / Sec)	919803	1747626	3236345
Kubernetes (tokens / Sec)	894208	1724417	3268835
Diff.	**2.78%**	**1.33%**	**1.00%**

Training benchmarking: GPT-3 / 126M parameters



Demo

Good news – the software stack for large scale training

Run on Clusters	NVIDIA NeMo Megatron Launcher
Model Collection	NVIDIA NeMo Megatron
Model-Specific Libraries	NVIDIA NeMo Framework
High-level Interface	Pytorch Lightning
Parallelism Libraries	DeepSpeed, FairScale
Deep Learning Framework	Pytorch
Communication Backend	MPI, NCCL, Gloo

Bad News
Training complexity has increased significantly in the LLM era

Good News
New tools and platforms simplify and abstract that complexity

Visit us at
Booth 1408

Parallelism Strategies for Distributed Training
https://www.run.ai/blog/parallelism-strategies-for-distributed-training

Monday 10am
Accelerating AI Workflows on AI Data Center Infrastructure
Omri G. & Ersin Y. from Adobe

Tuesday 3pm
Throughput Performance Benchmarking: Pre-Training Foundational Large Language Models on Kubernetes
Ronen D. & Raz R.

Wednesday 2pm
Accelerating AI Workflows on AI Data Center Infrastructure
Ronen D. & Guy S.

On-Demand
Expert Perspectives on the Evolution of AI Infrastructure
Panel

On-Demand
Considerations for Choosing LLM Serving Technologies
Ekin K.