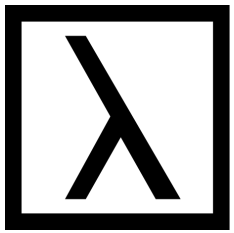


Lambda

ML Infrastructure Playbook

Stephen Balaban

Co-founder and CEO

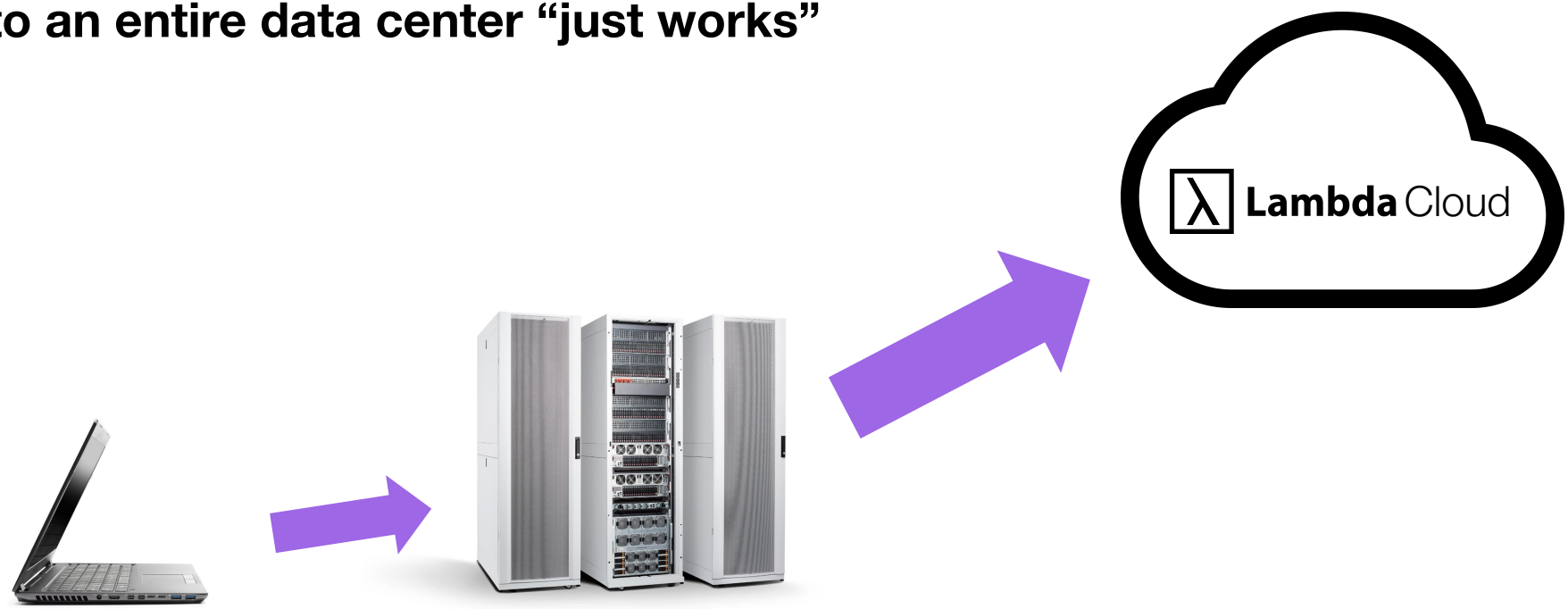
Lambda

Outline

1. Introduction to Lambda
2. Lambda AI Survey Overview
3. A playbook for getting started
    a. Cloud vs On-prem vs Hybrid
    b. Incremental scaling
4. A playbook for expansion
    a. Scaling from workstations to servers
        i. Shared resources
        ii. A quick hack for sharing GPUs
        iii. Set up jupyter notebook
        iv. Need more compute
        v. Need GPUs with more memory
        vi. Co-location services
    b. Considering a software stack at every scale
        i. Small
        ii. Medium
        iii. Large
    c. Scaling from servers to clusters
        i. Network
        ii. Power
        iii. Colo / on-prem
        iv. InfiniBand
5. Finishing Thoughts

A bit about Lambda

Lambda is building a future where scaling from a single GPU to an entire data center “just works”



Lambda provides the hardware & software to build ML infrastructure for your team

Tensorbook	Vector	Scalar	Echelon	Lambda Colo	Lambda Cloud
GPU Laptop	GPU Workstation	GPU Server	GPU Cluster		






Lambda Stack

A managed, always up-to-date, software stack for Deep Learning

In the beginning, there was a workstation.

"Our network takes between five and six days to train on two GTX 580 3GB GPUs. All of our experiments suggest that our results can be improved simply by waiting for faster GPUs and bigger datasets to become available."

- *Krizhevsky, Sutskever & Hinton 2012*

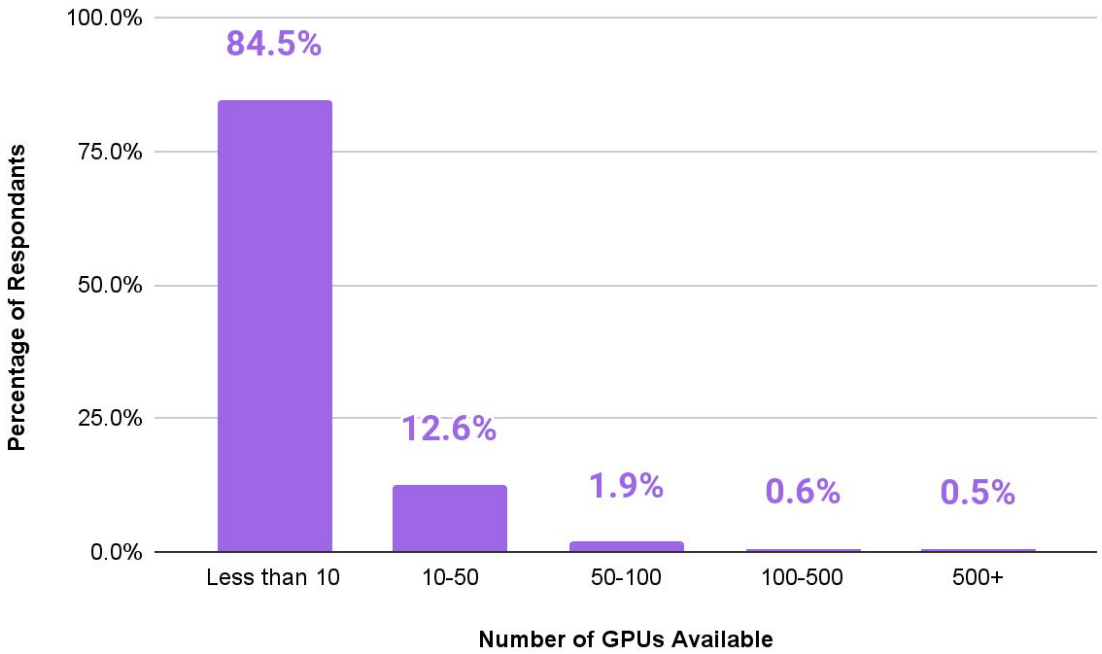
Fast forward to 2021. Massive models reign. But, most work still happens on a workstation.

Lambda Survey Results

Lambda

The distribution of compute, like wealth, follows a power law.

How many total GPUs does your lab have available for training Neural Networks?

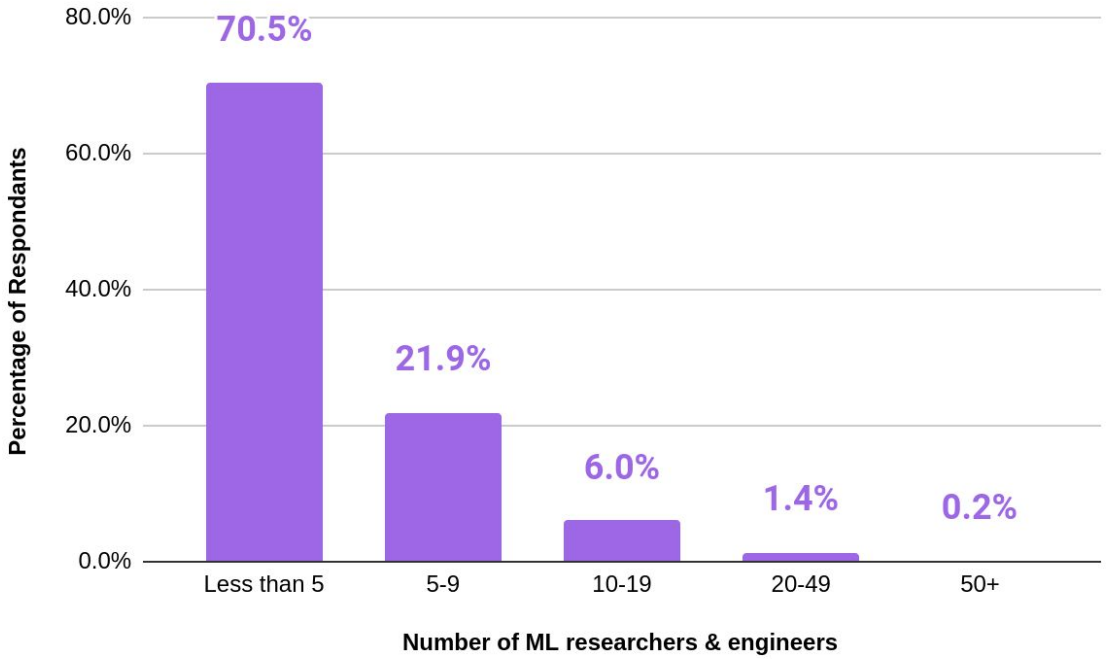


Source: Lambda AI Survey 2020-2021. N=647

Lambda

The distribution of R&D headcount also follows a power law.

How many ML researchers & engineers does your team have?

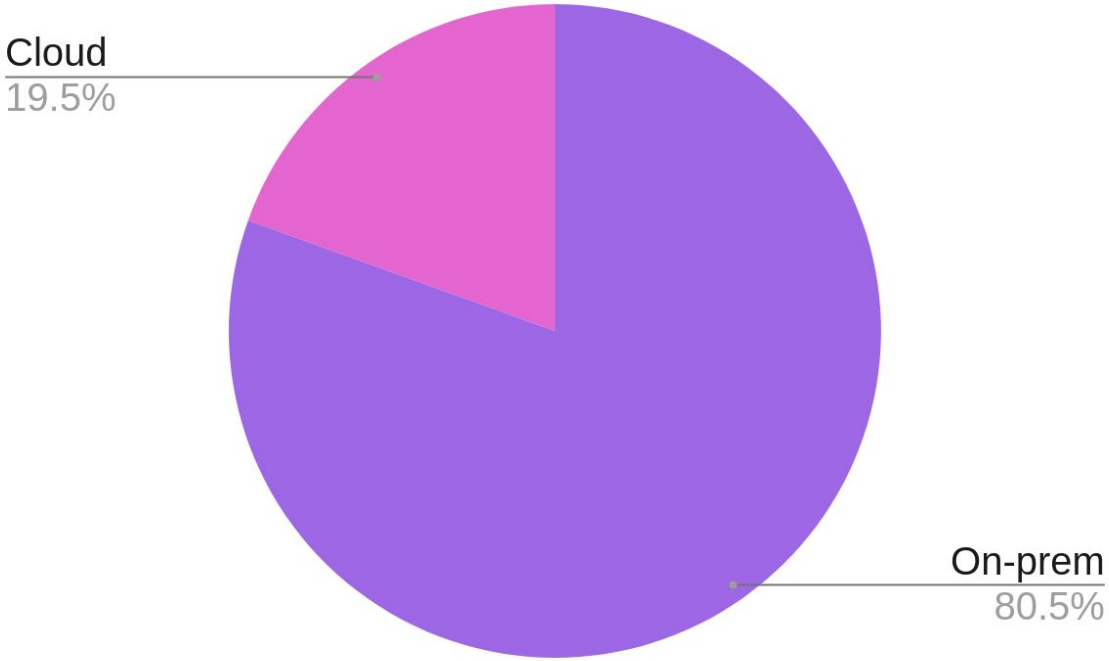


Source: Lambda AI Survey 2020-2021. N=645

Lambda

Can you guess what percentage of respondents train on-prem?

Do you train on-prem or in a public cloud?



Source: Lambda AI Survey 2020-2021. N=645

Lambda

A playbook for getting started

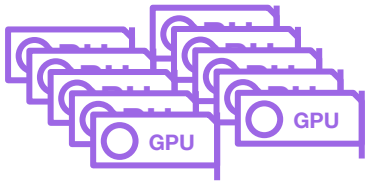
Lambda

Decide on

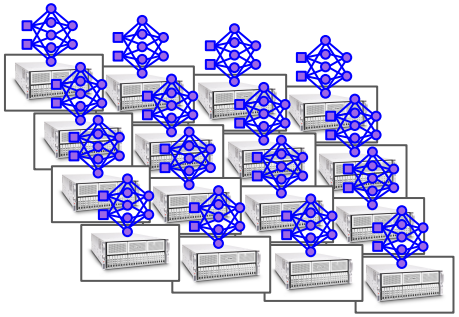
Cloud vs
On-prem vs
Hybrid

Lambda

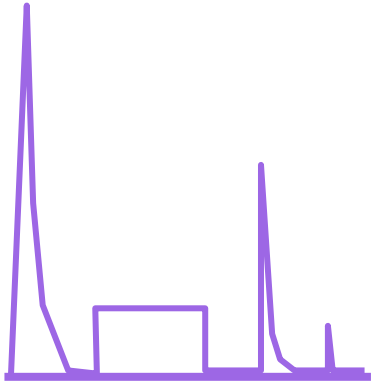
Good reasons to consider cloud



Need to use 100 GPUs now!



Doing production inference

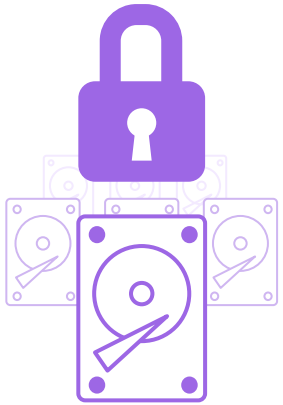


Spikey / inconsistent workload

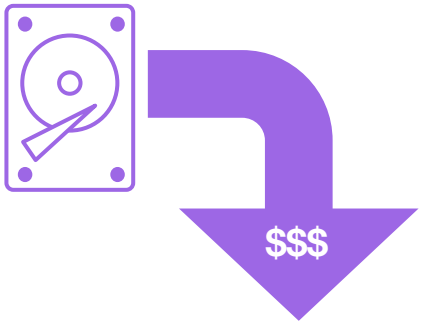
Lambda

Good reasons to consider on-prem

More compute for less money



Data sovereignty & security



Big data sets & expensive egress

Lambda

On-prem: start with a laptop or single workstation

This is what most researchers & engineers go to first and, unless you have a very good reason as to why you're special, you should be starting here too.

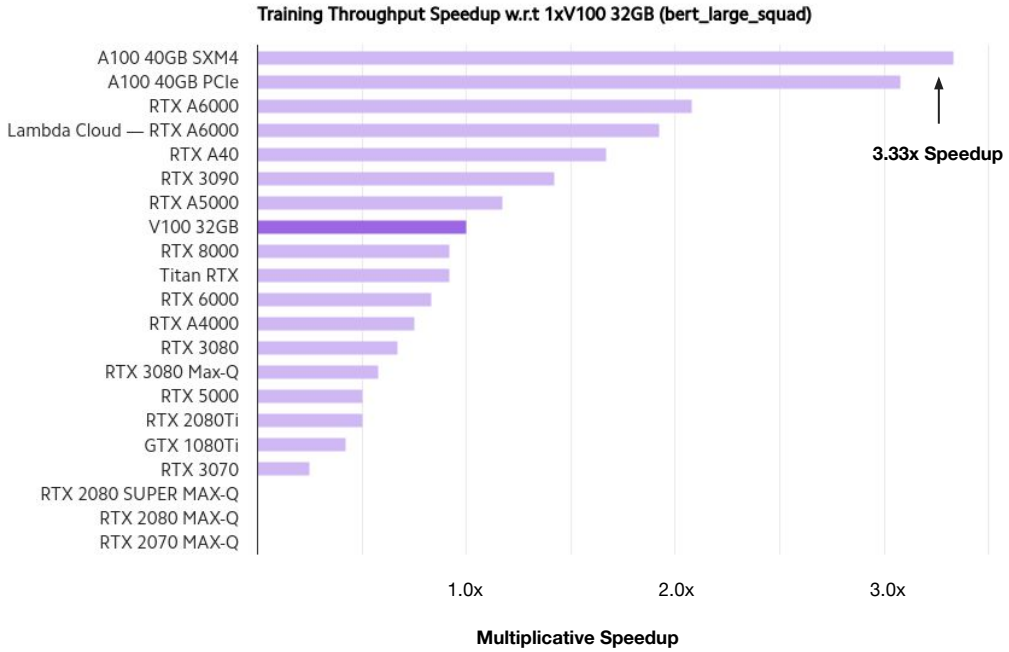
Lambda

Research the latest GPU benchmarks for the model you'll be training

https://lambdalabs.com/gpu-benchmarks

https://mlperf.org

Lambda GPU Benchmarks = throughput results for specific model/GPU pairs



Source: https://lambdalabs.com/gpu-benchmarks

Lambda

Incrementally add compute for each new hire

Lambda

A playbook for expansion

Lambda

From workstations to servers

When your team starts to **share resources**, that's usually a sign they should get a centralized server to provide compute to the whole organization.

Simple hack for resource sharing

CUDA_VISIBLE_DEVICES is your friend. Give everybody ssh access and use CUDA_VISIBLE_DEVICES to mask which GPU their process uses. For example, on an 8 GPU server:

```
Joe's SSH Terminal:

    $ CUDA_VISIBLE_DEVICES=0,1 python train.py
```

```
Francie's Local Terminal:

    $ CUDA_VISIBLE_DEVICES=3,4,5,6,7 python train.py
```

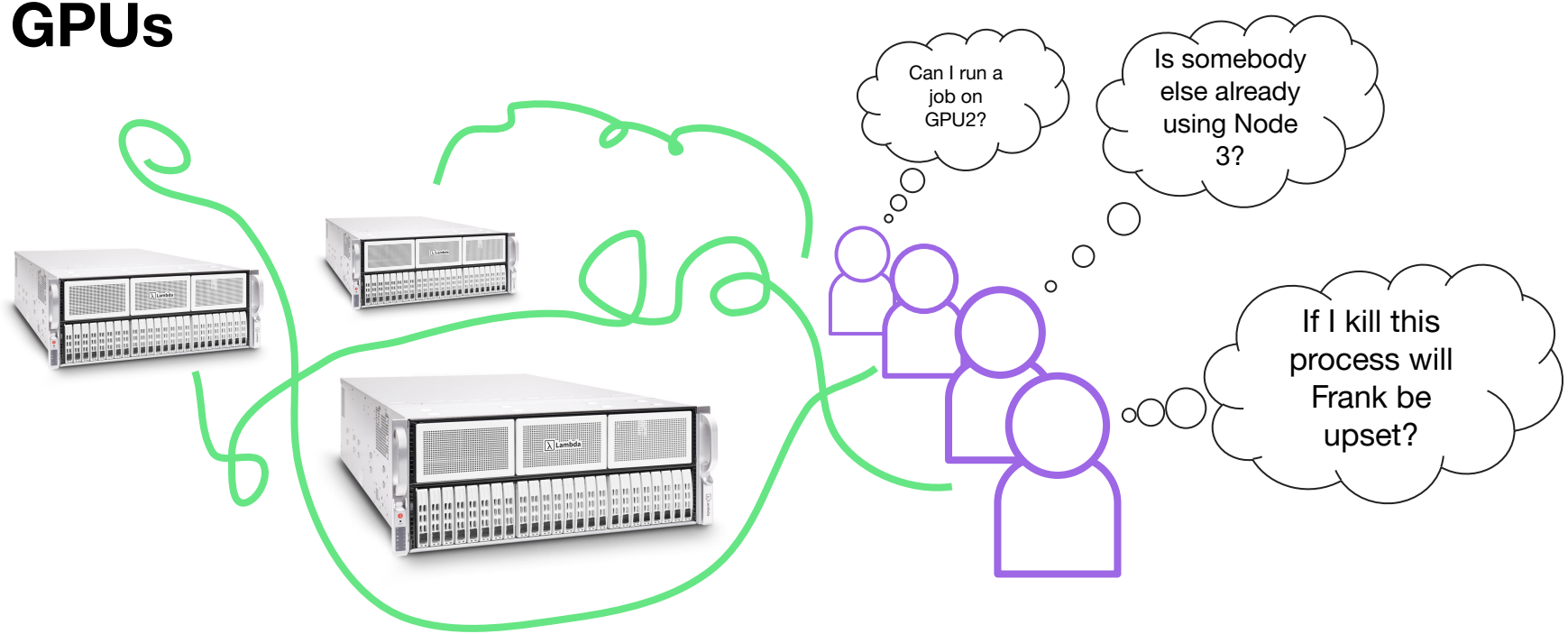
Simple hack for resource sharing

You can also put their allocation directly into their .bashrc

```
On Joe's account:

     $ cat >> ~/.bashrc <<EOF
export CUDA_VISIBLE_DEVICES=0
EOF
```

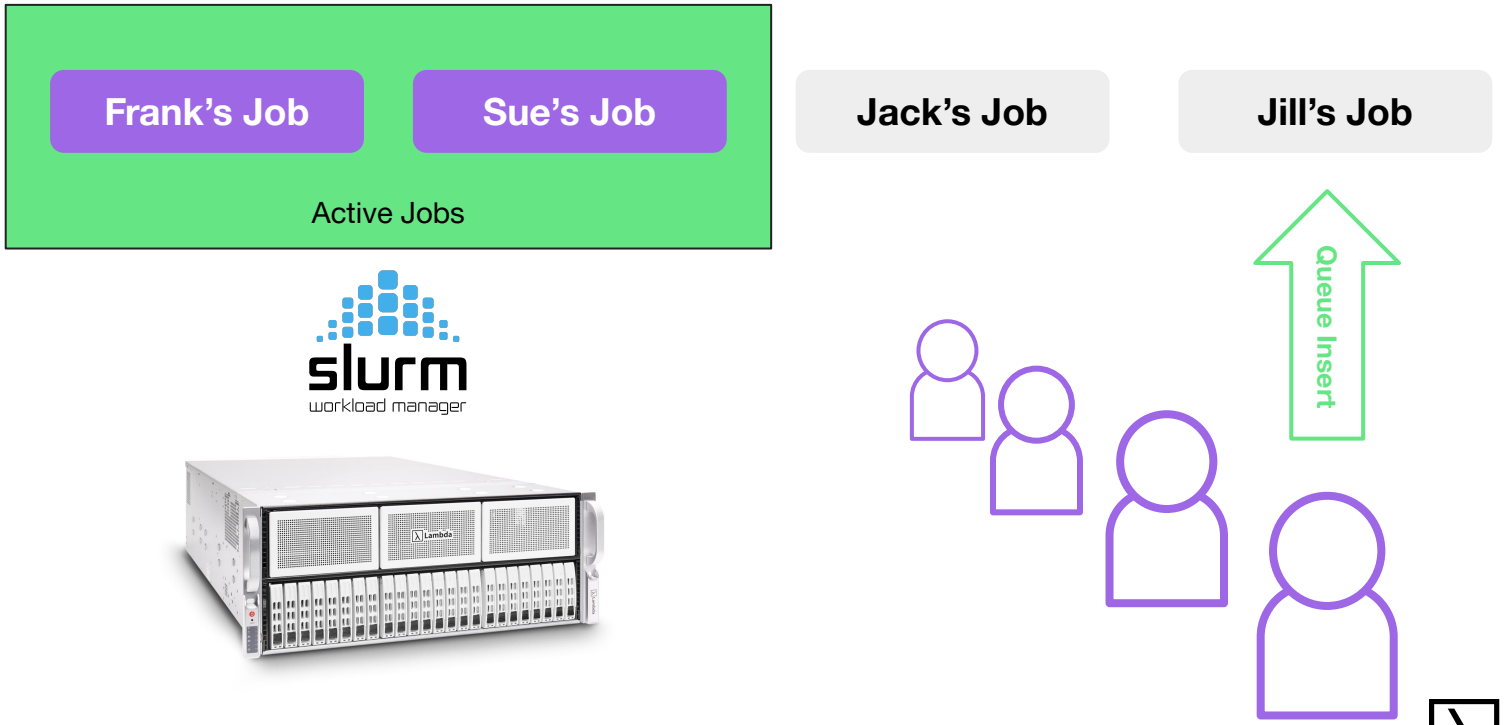
```
On Fancie's account:

     $ cat >> ~/.bashrc <<EOF
export CUDA_VISIBLE_DEVICES=1,2,3,4,5,6,7
EOF
```

Over time, this method will become unwieldy as you try and figure out who is running what jobs on which GPUs



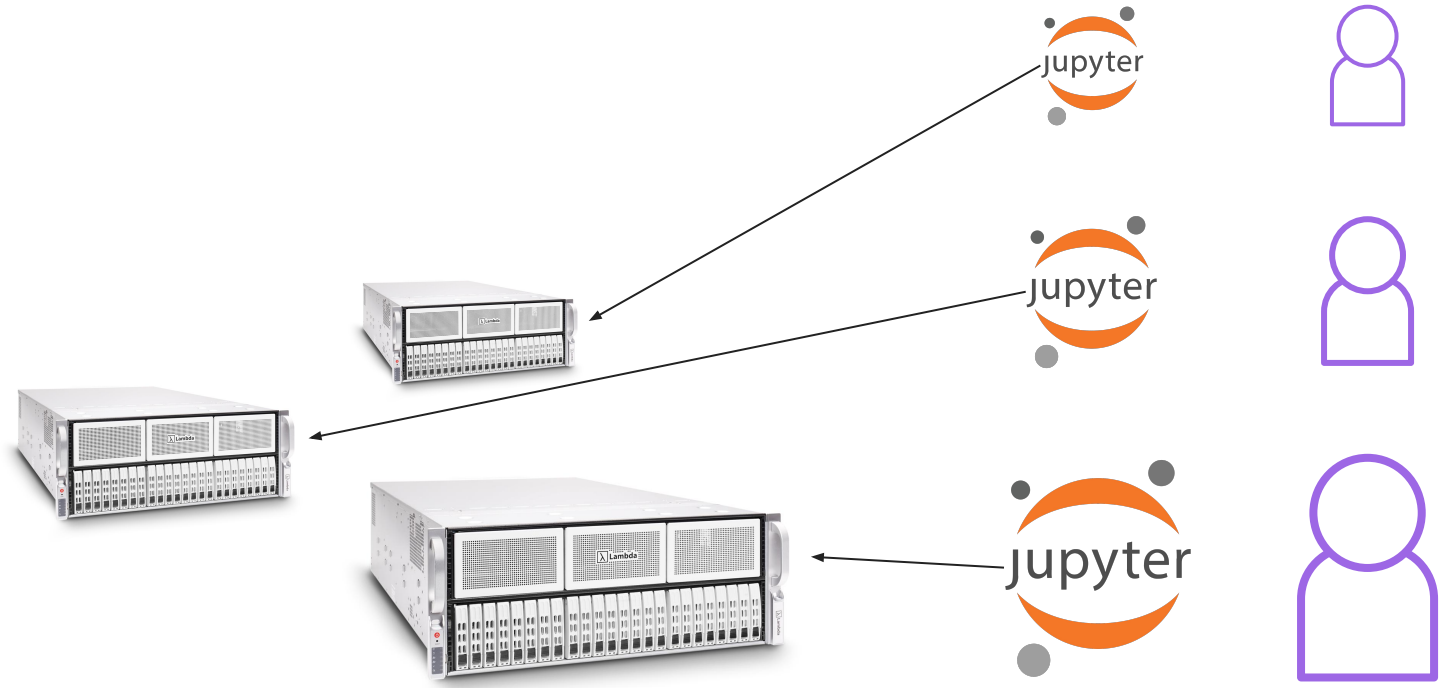
Lambda

When you get to that point, try using a job scheduler like SLURM or Kubeflow



Lambda

You can also set up separate notebook instances



Lambda

Considering a software stack at every scale

When you’re small, just use Lambda Stack

Lambda

```
LAMBDA_REPO=$(mktemp) && \
wget -O${LAMBDA_REPO} https://lambdalabs.com/static/misc/lambda-stack-repo.deb && \
sudo dpkg -i ${LAMBDA_REPO} && rm -f ${LAMBDA_REPO} && \
sudo apt-get update && sudo apt-get install -y lambda-stack-cuda
sudo reboot
```

https://lambdalabs.com/lambda-stack-deep-learning-software

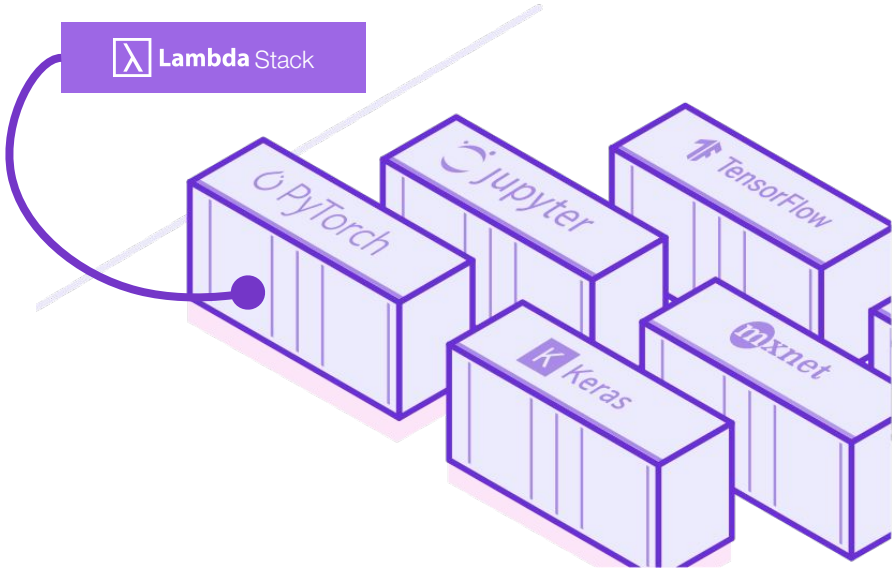
Lambda Stack provides the same environment everywhere

Lambda TensorBook	**Lambda** Vector	**Lambda** Blade	**Lambda** Echelon	**Lambda** GPU Cloud
				
Laptops	Workstations	Servers	Clusters	Cloud
Lambda Stack	**Lambda** Stack	**Lambda** Stack	**Lambda** Stack	**Lambda** Stack
TensorFlow PYTORCH ubuntu NVIDIA CUDA	TensorFlow PYTORCH ubuntu NVIDIA CUDA	TensorFlow PYTORCH ubuntu NVIDIA CUDA	TensorFlow PYTORCH ubuntu NVIDIA CUDA	TensorFlow PYTORCH ubuntu NVIDIA CUDA

Lambda

As you grow, consider using Lambda Stack + nvidia-container-toolkit

Compatible with all Docker and NGC containers




```
sudo apt-get install docker.io nvidia-container-toolkit
```

Source:
https://lambdalabs.com/blog/nvidia-ngc-tutorial-run-pytorch-docker-container-using-nvidia-container-toolkit-on-ubuntu/

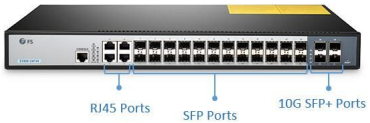
At a certain scale, use containers and consider an MLOps platform

Lots to choose from

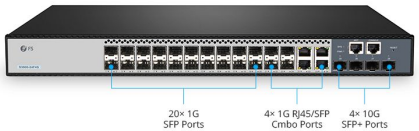
From servers to clusters

Network considerations

As you scale up your cluster, you'll want to go from 1 Gbps to 10 Gbps to 100 GbE and maybe even to 200 Gbps HDR InfiniBand. This depends on your team's need to do node-to-node communication and node-to-storage communication.



1 Gbps Ethernet



10 Gbps SFP+ Ethernet

200 Gbps InfiniBand

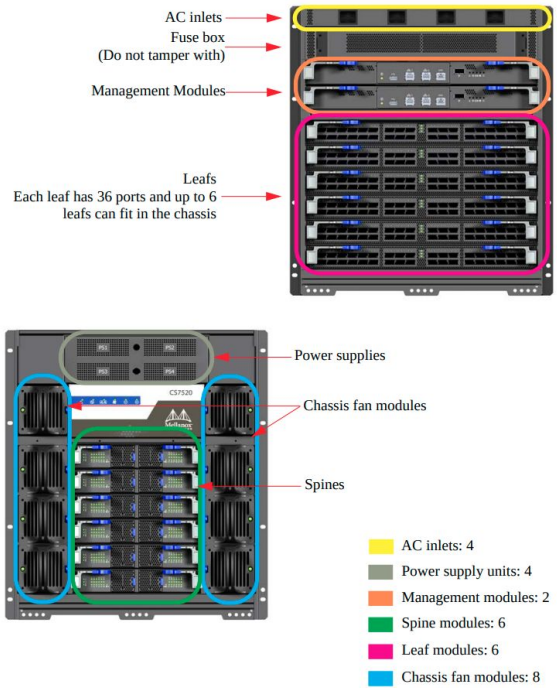
Lambda

Director switches: the final boss of the networking world

- Use a backplane instead of cables to build out the internal (spine <> leaf) network topology.
- To get 100% non-blocking bandwidth in the CS7520 you NEED to have all 6 spines installed.
- CS7520 has 216 southbound ports (and thus 216 northbound ports).
- Assuming each spine has only 36 ports, how many spines do you need to support 216 ports?
    - 36 ports / spine, 216 ports needed
    - 216 ports / 36 (ports / spine) = 6 spine switches

For more info see:
https://www.mellanox.com/related-docs/prod_ib_switch_systems/CS7520_Dismantling_Guide.pdf

Figure 1: Front and Rear View of the CS7520

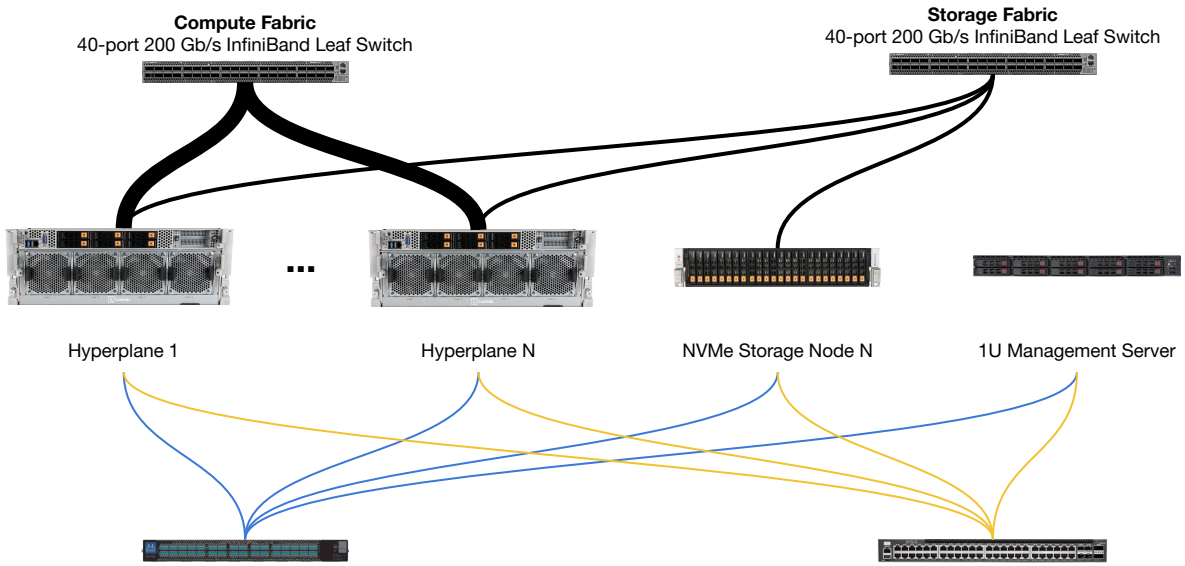


Source: How to build a GPU cluster for AI

Lambda

Lambda Echelon Network Topology

(Single Rack Configuration)



Source: Lambda Echelon Whitepaper v1.2

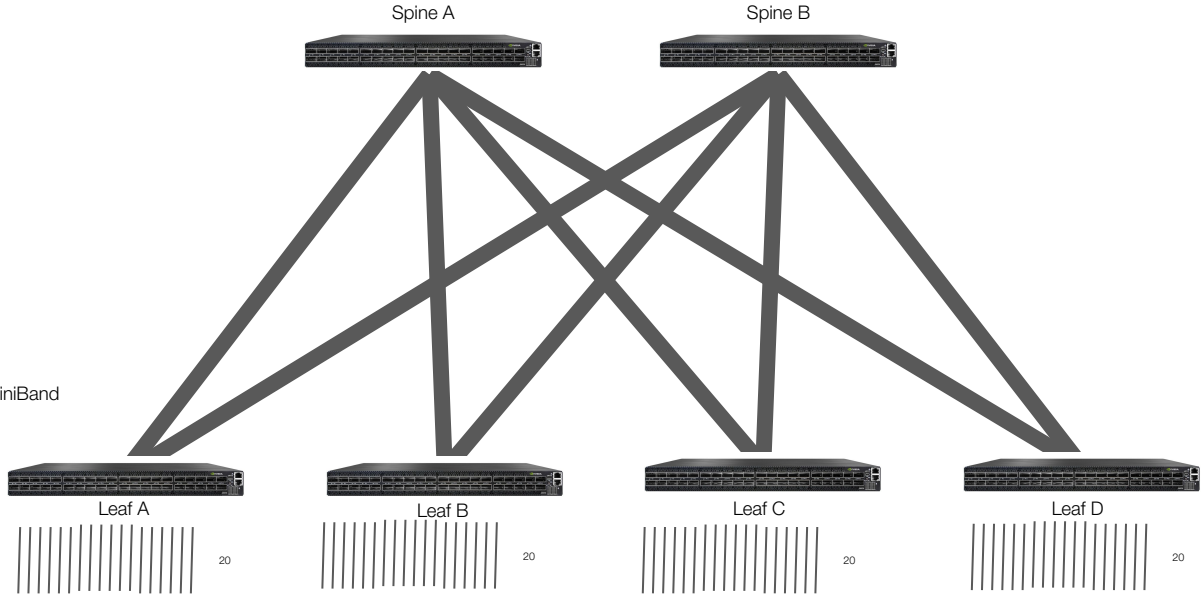
To InfiniBand and Beyond!

Legend

A 40-port 200Gb/s IB HDR Switch

Each thick grey line represents 10 InfiniBand cables.

Each thin black line represents 1 InfiniBand cable.



Lambda

Storage considerations

Open source storage options

Proprietary storage options

Source: Lambda Echelon Whitepaper v1.2

Power considerations

Single phase systems

P (watts) = V (volts) * I (amps)

We use *I* from the French, *intensité du courant.*
Because that's what André-Marie Ampère used.

3-phase systems

P = 3 * V / sqrt(3) * I

This simplifies to P = sqrt(3) * V * I
Because 3 / sqrt(3) = sqrt(3)

Real life 3-phase systems

P = sqrt(3) * V * I * 0.8

It's very common to see an 80% regulatory derating factor applied to PDUs.

How do PDU manufacturers calculate power capacity?

Example PDU:
APC 8966

From the APC8966 Data Sheet:

Input frequency
50/60 Hz

Number of Power Cords
1

Load Capacity
17300VA

Maximum Input Current
60A

Maximum Line Current
48A

Regulatory Derated Input Current (North America)
48A

Nominal Output Voltage
208V

Nominal Input Voltage
208V 3PH

Input Connections
IEC 60309 60A 3P + PE

P = sqrt(3) * V * I * 0.8

Plug in the numbers from the data sheet:

P = sqrt(3) * **208** * **60** * **0.8**
P = 17292.7953

P = 17.3kVA

*See how they derate the maximum input current of 60A to the "Regulatory Derated Input Current (North America)" 48A? That's 48A = 60A * 0.8.*

That's where the 0.8 came from in the previous slide.

Lambda

Plug types frequently seen in HPC

IEC60309 - 60A 3-phase plug - 208V
Blue means the system is between 200 and 250V.

IEC60309 - 60A 3-phase plug - 415V
Red means the system is above 400V.

NEMA L15-30P 30A 3-phase plug - 208V

Lambda

Receptacles & plugs, continued

Plug Photo	Name	Plugs into	Receptacle Photo	Max Amps
	IEC C13 Plug	IEC C14 Receptacle (on server)		**15A** (Max power ~2.5kW)
	IEC C14 Plug	IEC C13 Receptacle (on PDU)		**15A** (Max power ~2.5kW)
	IEC C19 Plug	IEC C20 Receptacle (on server)		**20A** (Max power ~3.2kW)
	IEC C20 Plug	IEC C19 Receptacle (on PDU)		**20A** (Max power ~3.2kW)

***IEC** stands for International Electrotechnical Commission, an international standards organization headquartered in Switzerland.*

Colo vs on-prem

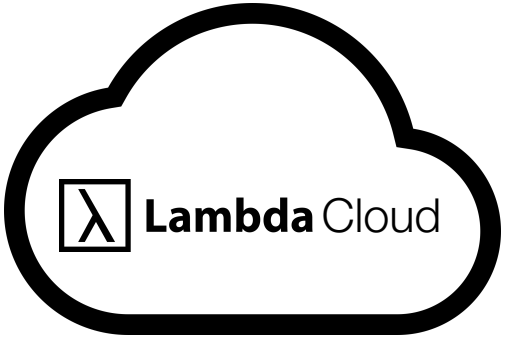
Lambda

For a comprehensive clustering guide

https://www.youtube.com/watch?v=rfu5FwncZ6s

Lambda

Finishing thoughts


Lambda Cloud

LAMBDALABS.COM

https://youtube.com/c/lambdalabs