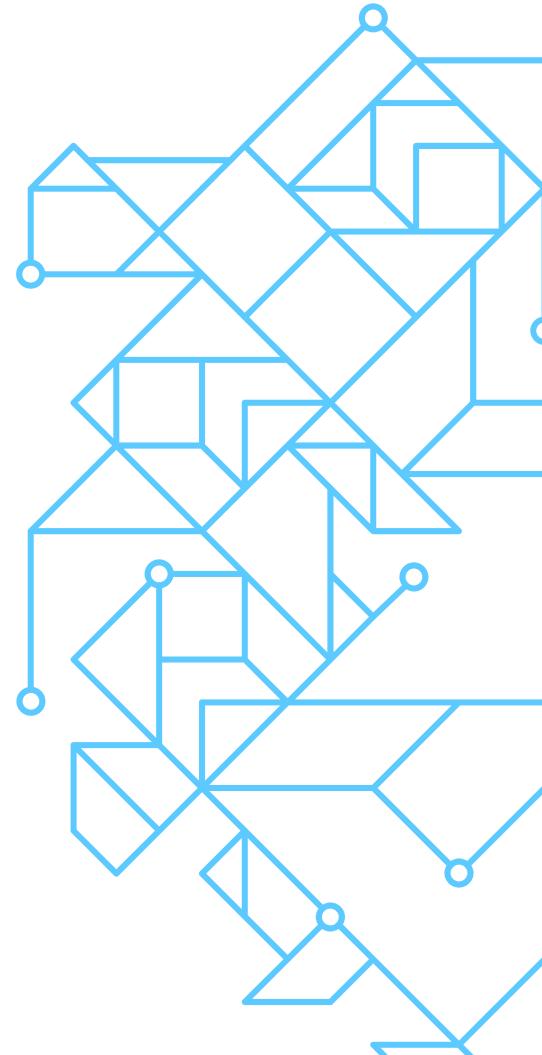


# Welcome!

We're happy to have you here.





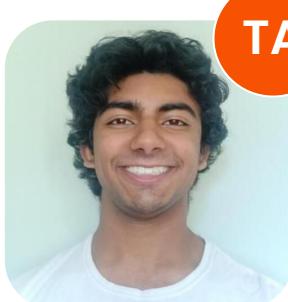
# Meet the team!



Kamil



Emmy



Balaji



Xiaowei

TA



Artur

TA

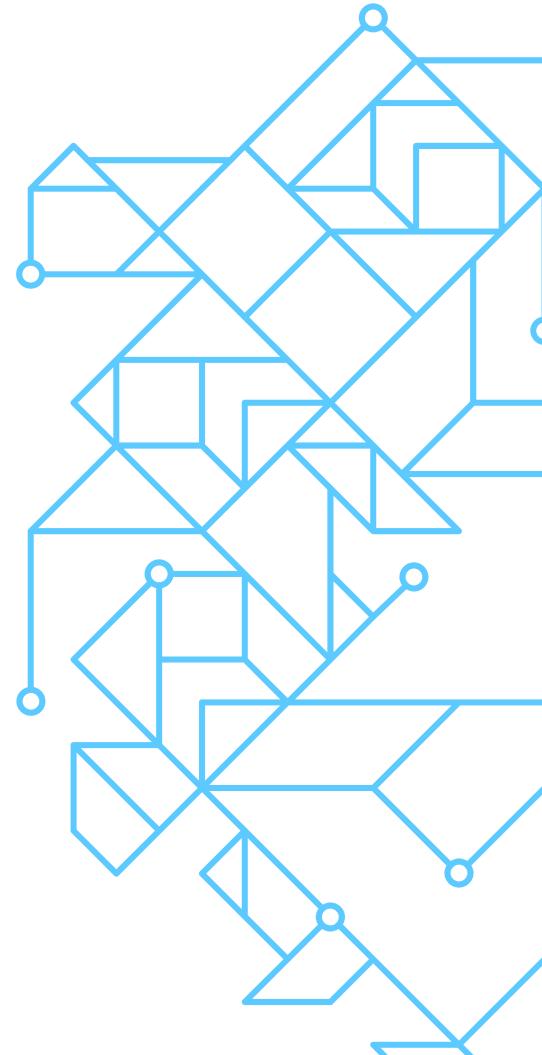


Yunxuan

TA

# The Plan

Here's what to expect today.



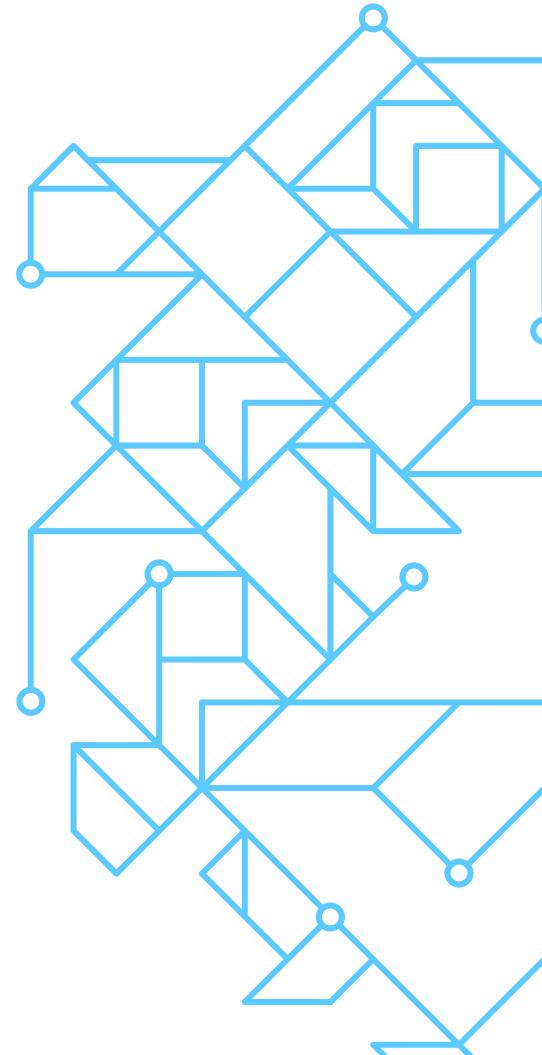


# Today's agenda.

Time	Description
10 min	Welcome + Accessing Your Cluster
15 min	Introduction to Ray and Anyscale
20 min	Anyscale Clusters + Workspaces
45 min	Hugging Face Fine-Tuning, Hyperparameter Tuning, and Batch Inference w/ Ray AIR
10 min	Anyscale Jobs
5 min	Summary + Survey

# Course Logistics

**How to access your cluster and other resources.**





# Tech check.



## Accessing Anyscale clusters.

- All work will be in Anyscale provisioned clusters.
- Our GitHub repo will be mounted automatically.
- Access begins now.
  - Check your email for login information.
  - Step-by-step instructions to follow.



# Anyscale Login

Link to Anyscale cluster: [console.anyscale.com](https://console.anyscale.com)

The screenshot shows a web browser window with the URL `console.anyscale.com` in the address bar. The page has a dark blue header with the Anyscale logo and the text "Scale your application from your laptop to the cloud". Below the header, there's a "Get started" section with a "Work email" input field containing `john@acme.com` and a "Next" button.

Check your **email** for your username and password!

select  
“Clusters”

The screenshot shows the Anyscale web interface. On the left, a sidebar menu is displayed under the heading "anyscale". The menu items are: Home, Projects, Workspaces, Interactive sessions, Jobs, Services, Clusters (which is highlighted with an orange box and has a blue arrow pointing to it from the left), and Configurations. The main content area is titled "Clusters" and contains a table of clusters. The table has columns for Name, Status, and Active resources. One cluster, "ray-acm-emmy", is listed with a status of "Active (auto-terminates in 117 minutes)" and 0 CPU resources. A blue arrow points from the text "click on your cluster" to the "ray-acm-emmy" row. The top right of the interface features a toolbar with buttons for Create, Start, Terminate, and Archive, along with search and filter options.

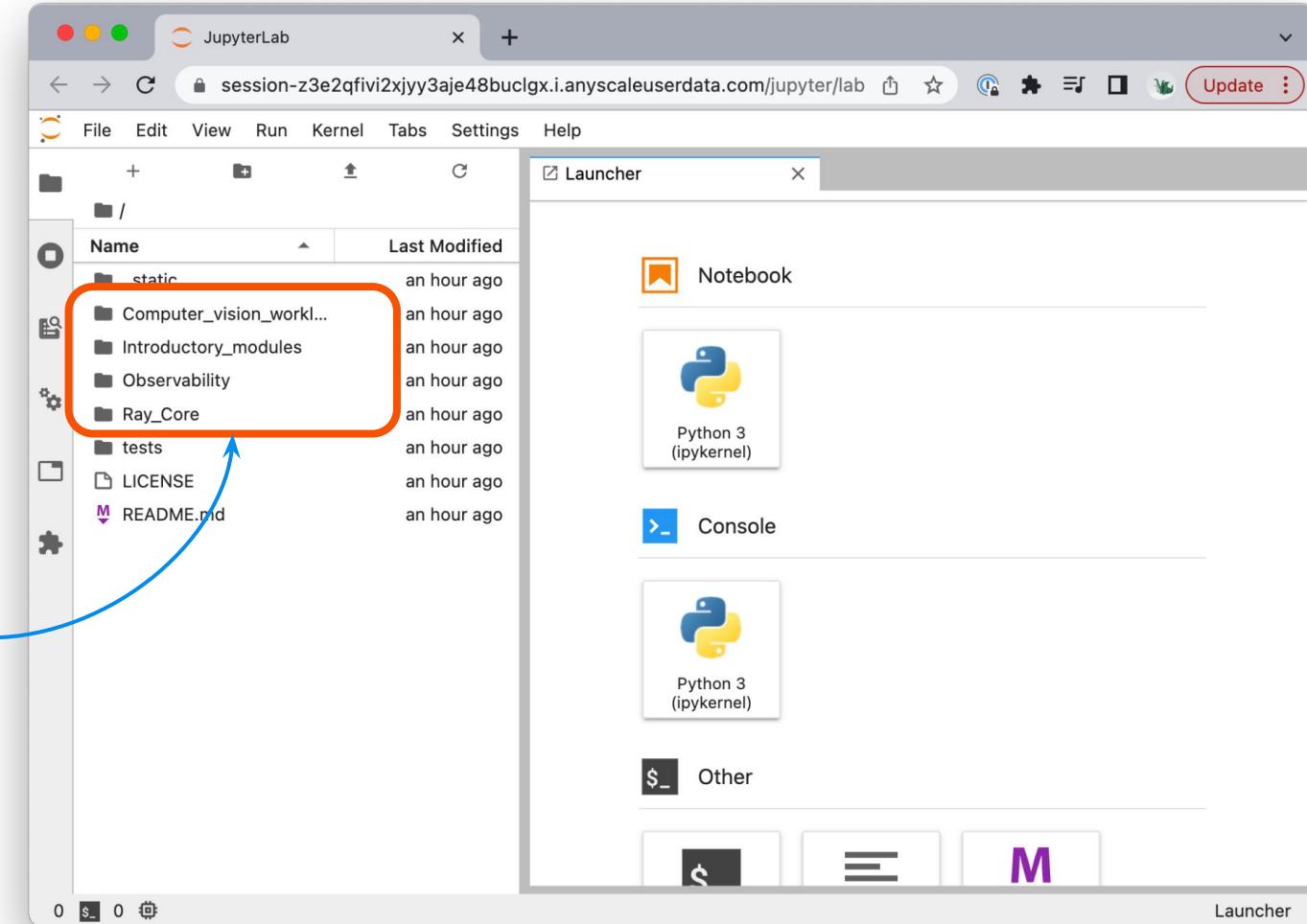
<input type="checkbox"/>	Name	Status ↓	Active resources
<input type="checkbox"/>	ray-acm-emmy	Active (auto-terminates in 117 minutes)	0 cpu

click on your  
cluster

“Start” the cluster,  
then  
select  
“Jupyter”

The screenshot shows the Anyscale console interface. On the left, a sidebar menu includes Home, Projects (selected), Workspaces, Interactive sessions, Jobs, Services (highlighted with a blue line), Clusters, Configurations, Emmy Li, Help, Feedback, and Collapse. The main content area shows a breadcrumb path: ray... > emmy-ra... > Jupyter. The Jupyter button is highlighted with a red box and an arrow pointing to it from the 'Services' highlight. Below the breadcrumb is a section titled 'About this cluster' with details: Status (Active), ID (ses\_z5yfnmzamcpxc4uk95mezhr4), Created by (emmy@anyscale.com), Created at (Dec 8, 2022 at 10:01:54 AM), Access (Everyone in your organization can view a...), and Project (ray-saturday). A section titled 'Resource usage' shows CPU, Object store memory, and GPU all at 0%. Below that is a section titled 'Configuration' with fields: Cluster environment (ray-sat-v1:10), Compute config (kk-rs-config-for-emmy), and Cloud (anyscale\_default\_cloud (aws, us-west-2)). At the bottom is a 'Terminal' section.

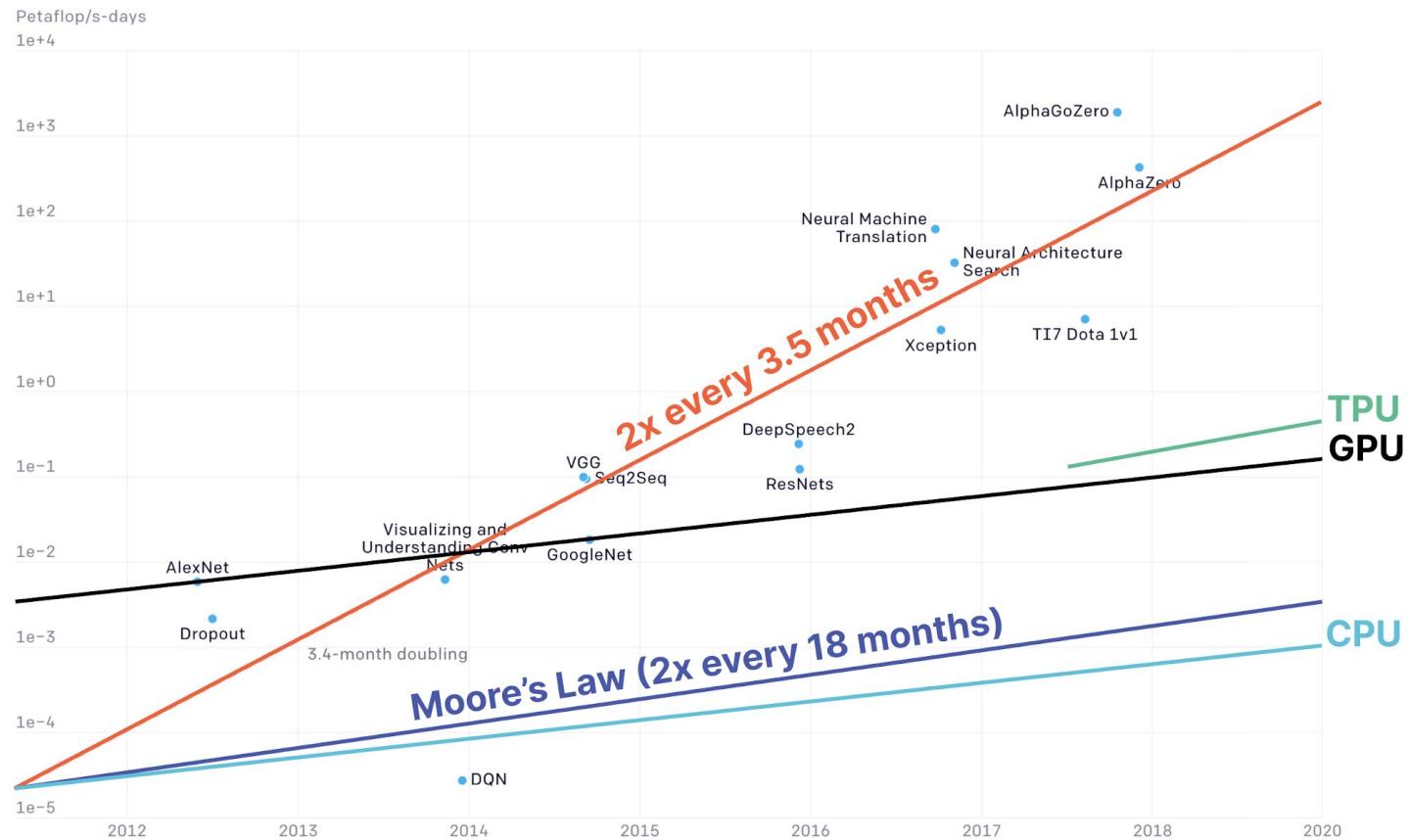
View  
modules  
here



# The State of Ray



# Distributed computing: a bit of context







Ray 2.0 is production ready

Introducing Ray

Ray's core high availability

avalanche

**800+ Contributors**

# Ray: A global community



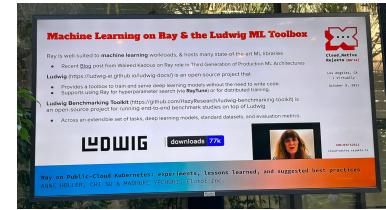
Colombia

## Ray Serve

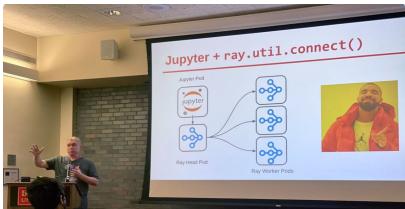
MLモデルを簡単にスケーラブルにデプロイできるモデルサービング

1. Framework依存なし: PyTorch, Tensorflow, keras, scikit-learnなどどんなFrameworkでもOK
2. Pythonファースト: ConfigurationをPythonで書ける

Japan



Los Angeles



Boston

## Veloce: 基于Ray的异构训练低代码工具库

—— 360 机器学习平台在推荐场景下的探索

翟晓宇

360

China



Toronto

... and many more!

# Ray: Fastest Growing Scalable Compute Framework



McKinsey  
& Company



ERICSSON



cruise

Morgan Stanley



J.P.Morgan



Uber



RICARDO



verizon<sup>✓</sup>



**25,000+**

GitHub  
stars

**800+**

Community  
Contributors

**5,000+**

Repositories  
Depend on Ray

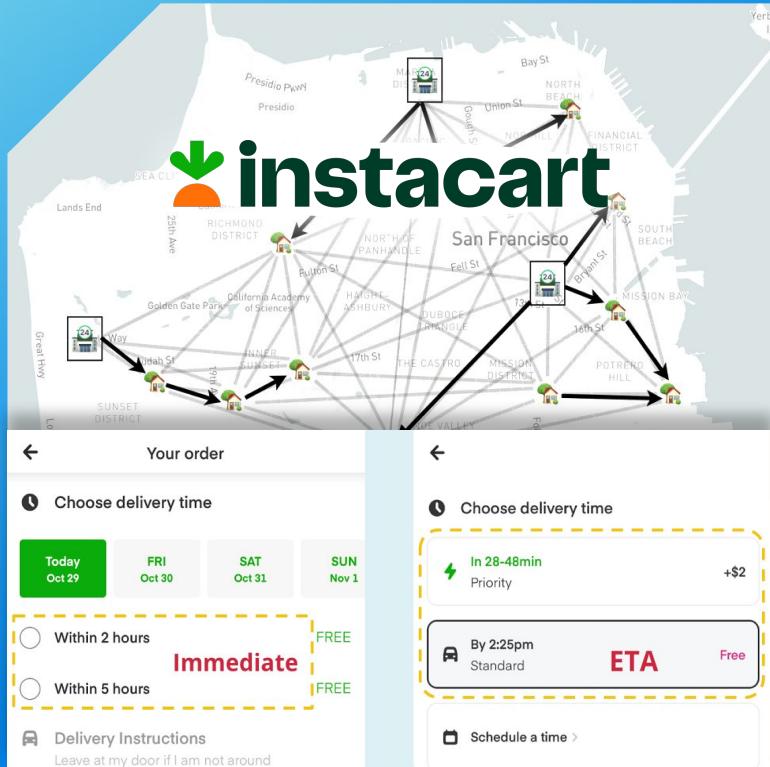
**1,000+**

Organizations  
Using Ray

ETA prediction, safety,  
maps, eats, marketplace

Deep learning,  
classical ML,  
hyperparameter  
tuning, data ingest





Order fulfillment,  
scheduling, delivery ETAs  
**Reduce training times  
for 12,000 models  
from days to hours**

Designing sailboats in  
simulation, building them  
in the real world

They won the  
America's Cup

McKinsey  
& Company



# OpenAI on Ray

---

*"We use Ray to train our largest models including ChatGPT."*

**Greg Brockman, Co-founder and President, OpenAI**



## LLM Pain Points

- Large-scale: need to run on many **expensive** GPUs 
- Existing training & inference solutions **don't work** (can't naively scale) 
- Technology is emerging - important to be able to **iterate quickly at scale** 

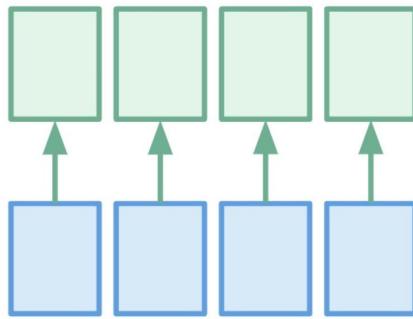
## Ray: fitting solution

- **Spot instance** support for training.
- **Flexible scheduling** and resource allocation enables new scaling patterns.
- Very **fast development** model – re-run your Python script across a cluster.

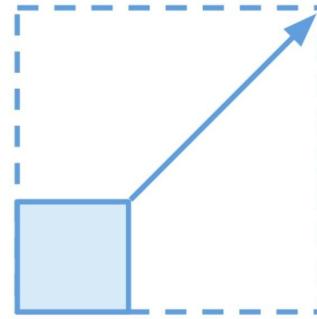
# Key Ray characteristics



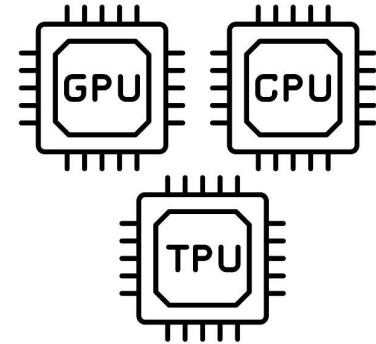
Python first approach



Simple and flexible API

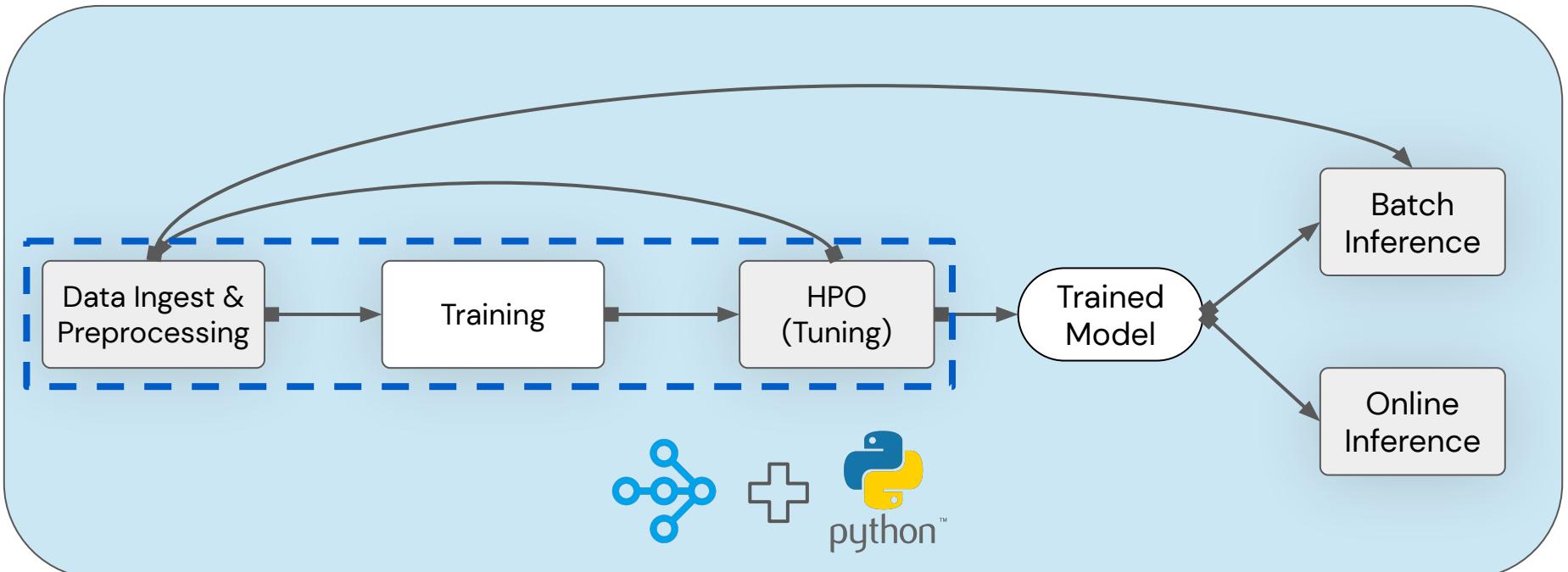


Scalability



Support for heterogeneous  
hardware

# ML Application Lifecycle



**Ray enables building end-to-end ML applications, all in Python**

# Ray AIR and integrations



# About Anyscale

---

## Founded by the creators of Ray

- Ion Stoica, Chairman & President (Co-Founder Databricks, Lead UC Berkeley RISELab and AMPLab, Founder Conviva)
- Robert Nishihara, CEO (Ray co-creator, Ph.D. Berkeley)
- Philipp Moritz, CTO (Ray co-creator, Ph.D. Berkeley)
- Michael Jordan (UC Berkeley RISELab, AMPLab)

## Team

- Experts in Distributed Computing and AI
- Ex-Uber, Stripe, AWS, Databricks, Google Brain, Facebook

## Investors & Board

- Investors: Andreessen Horowitz, NEA, Addition, Foundation, Intel Capital
- Board: Ben Horowitz, Pete Sonsini
- Raised \$250M+



# Three Trends

---

- 1 AI computational demands are exploding
- 2 The end of Moore's Law
- 3 AI is permeating every industry

# Ray | Unified framework for scalable ML

Ray

	Unified	Scalable	Open
	<ul style="list-style-type: none"><li>• Common infrastructure for all distributed workloads</li><li>• Libraries for training, serving, data loading</li><li>• Extensible to future workloads</li></ul>	<ul style="list-style-type: none"><li>• Laptop to cloud with zero-code changes</li><li>• Elastic scaling in production</li><li>• Scale to support the most demanding workloads</li></ul>	<ul style="list-style-type: none"><li>• Open source</li><li>• Integrates with the entire ML and Python ecosystem</li><li>• Run anywhere, including on any cloud</li></ul>



# Anyscale | The best way to run Ray

## SCALABLE COMPUTE PLATFORM

Anyscale

Unified	Scalable	Open
<ul style="list-style-type: none"><li>Common infrastructure for all distributed workloads</li><li>Libraries for training, serving, data loading</li><li>Extensible to future workloads</li></ul>	<ul style="list-style-type: none"><li>Laptop to cloud with zero-code changes</li><li>Elastic scaling in production</li><li>Scale to support the most demanding workloads</li></ul>	<ul style="list-style-type: none"><li>Run anywhere, including on any cloud</li><li>Integrates with the entire ML and Python ecosystem</li><li>Open source</li></ul>

### Accelerate time to market with Anyscale

- Managed service
- Enterprise ready
- Dev-to-prod experience
- Built-in observability & operational tools



# Conclusion

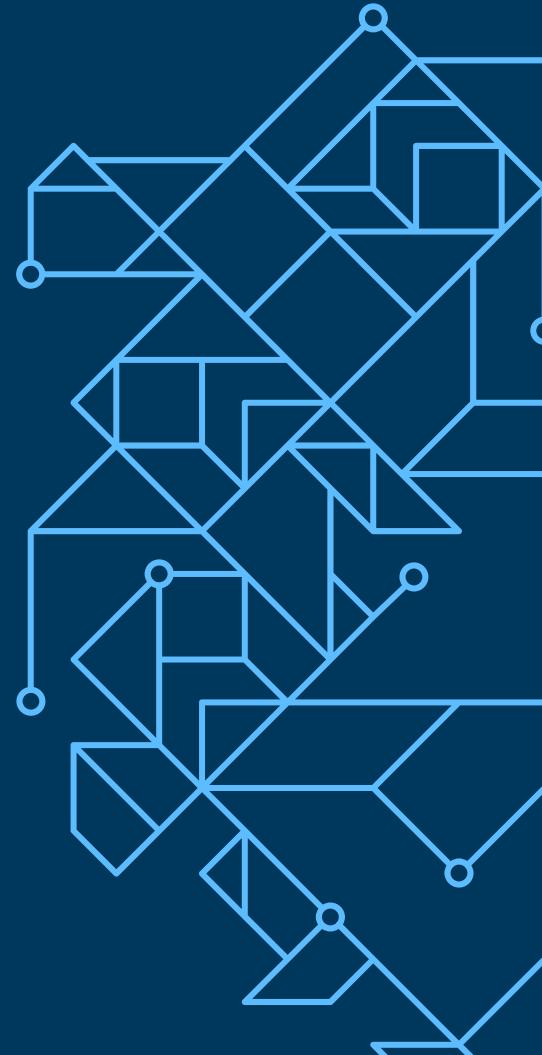
More applications need to **scale** (driven by AI).

**Ray:** unified framework for scalable computing.

**Anyscale:** the best platform for Ray.

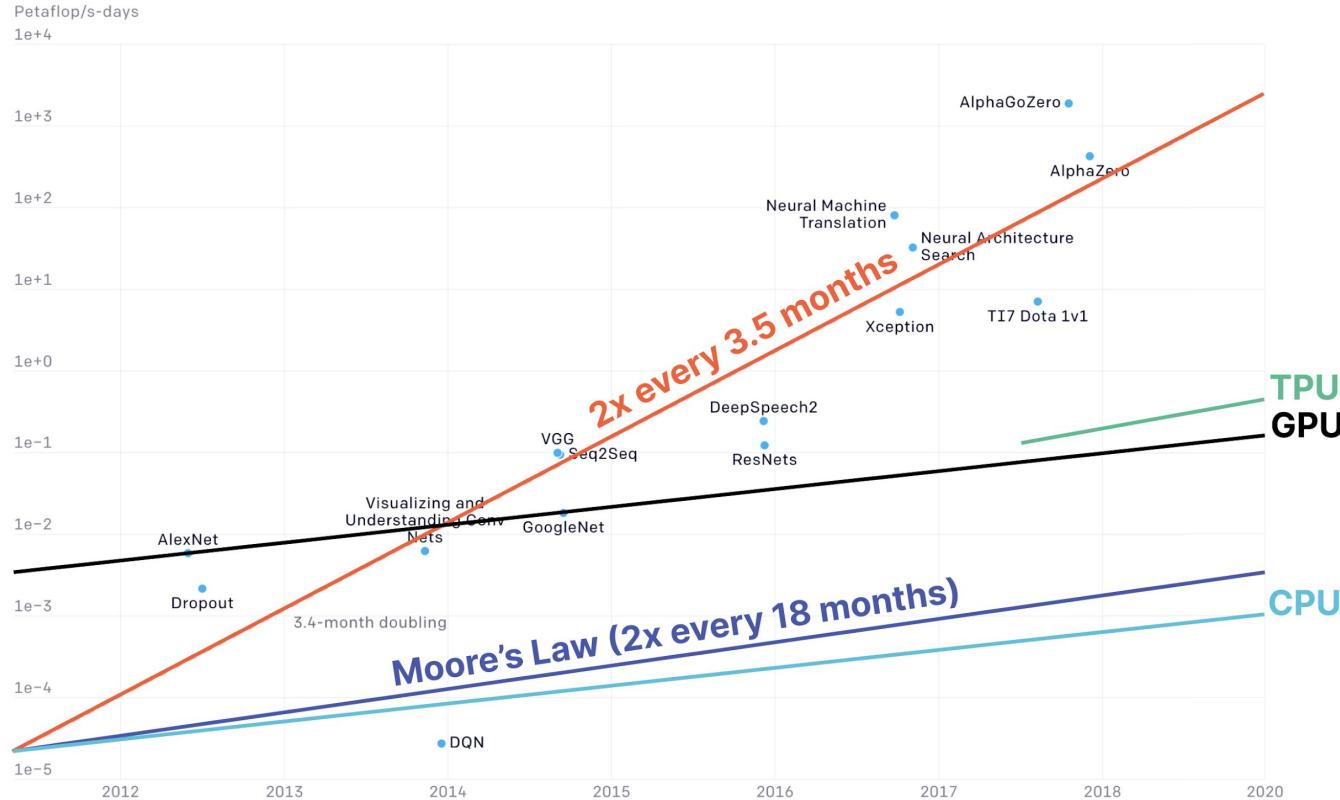
# Distributed Model Fine-Tuning

Data parallelism across  
multiple machines.



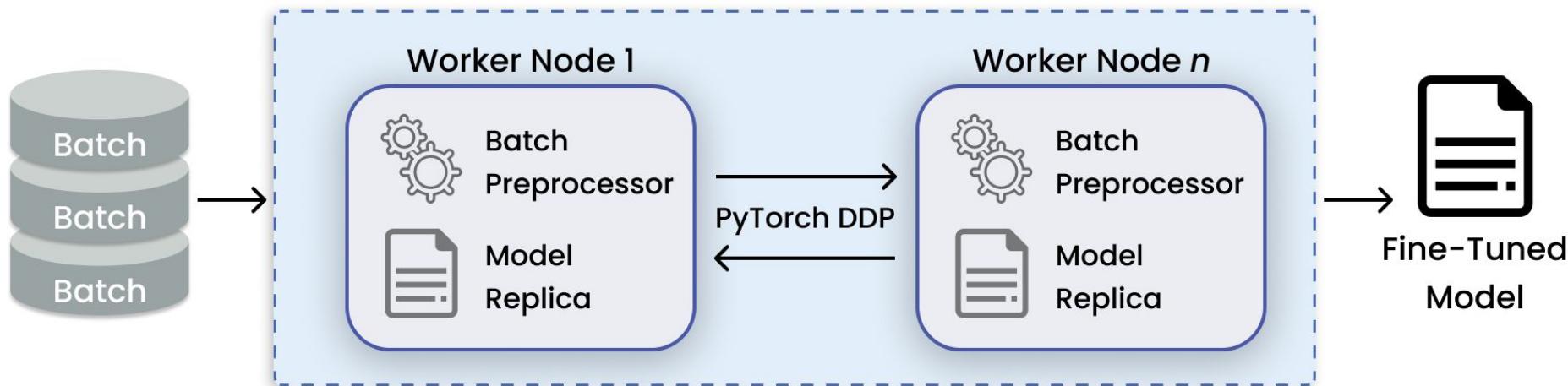


# Why distribute training?





# Data parallelism





# The task.

## Instructions and demonstrations

- **instruction** - A prompt and/or question.
- **input** - Additional context information.
- **output** - The generated response.

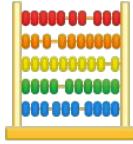


# The data.

## Alpaca

- 52k instructions and demonstrations.
- Generated by Open AI's  
**text-davinci-003**

<b>instruction (string)</b>	<b>input (string)</b>	<b>output (string)</b>
"Identify the odd one out."	"Twitter, Instagram, Telegram"	"Telegram"



# The model.

Instruction finetuning

Please answer the following question.

What is the boiling point of Nitrogen?

Chain-of-thought finetuning

Answer the following question by reasoning step-by-step.

The cafeteria had 23 apples. If they used 20 for lunch and bought 6 more, how many apples do they have?

Multi-task instruction finetuning (1.8K tasks)

Inference: generalization to unseen tasks

Q: Can Geoffrey Hinton have a conversation with George Washington?  
Give the rationale before answering.

Language model

-320.4F

The cafeteria had 23 apples originally. They used 20 to make lunch. So they had  $23 - 20 = 3$ . They bought 6 more apples, so they have  $3 + 6 = 9$ .

FLAN-T5

Geoffrey Hinton is a British-Canadian computer scientist born in 1947. George Washington died in 1799. Thus, they could not have had a conversation together. So the answer is "no".



# The goals.



## Fine-Tuning

*Train FLAN-T5 on batches of Alpaca data.*



## Hyperparameter Tuning

*Run multiple trial experiments with different hyperparameters.*



## Batch Inference

*Generate predictions on batches of data.*

Meet me  
here!

Model\_finetu... - JupyterLab

File Edit View Run Kernel Tabs Settings Help

/ NLP\_workloads / Text\_generation /

Name Last Modified

- download ... 5 hours ago
- Model\_fin... 5 hours ago
- utils.py 5 hours ago

Model\_finetuning\_and\_batch.ipynb

Python 3 (ipykernel)

## Model Fine-Tuning and Batch Inference

RAY

```
graph LR; A[Data Preprocessing] --> B[Fine-Tuning]; B --> C[Hyperparameter Tuning]; C --> D[Batch Inference]
```

Welcome to this tutorial notebook, where you'll explore how to leverage Ray AI Runtime (AIR) to perform distributed data preprocessing, fine-tuning, hyperparameter tuning, and batch inference using the FLAN-T5 model applied to the Alpaca dataset.

FLAN-T5 is transformer-based language model based on Google's T5 architecture and fine-tuned on instruction data. You will be further training this model on Alpaca, a set of 52k instructions and demonstrations. Through Ray AIR's integration with the Hugging

Simple 0 \$ 1 Python 3 (ipykernel) ... Mode: Comm... ↗ Ln 1, C... Model\_finetuning\_and\_batch\_inferenc... 1 🔔



# Fill out the survey.



Go to [bit.ly/anyscale-ray-feedback](https://bit.ly/anyscale-ray-feedback)



Ray Summit 2023 tickets  
*A random draw from survey submissions.*

