RAY

Scale AI and Python applications with Ray, a unified framework for scalable computing

Scaling AI and Python Workloads Effortlessly with Ray

Distributed Computing Is Inevitable

Modern AI and Python workloads require scale

Compute demands for machine learning (ML) training have grown 10x every 18 months since 2010. Over the same time period, the compute capabilities of AI accelerators such as GPUs and TPUs have less than doubled. This means that every year and a half organizations need 5x more AI accelerators/nodes to train the latest ML models and leverage cutting edge ML capabilities. Distributed computing is the only way to meet these requirements.

Distributed Computing Is Challenging

While solutions such as AWS SageMaker and GCP Vertex AI have emerged to help organizations deal with scaling AI workloads, these solutions put significant constraints on how applications are developed and which libraries they can use. This makes it difficult to keep up with the latest models and algorithms, and freely integrate with the rapidly evolving open ML ecosystem.

Select organizations with massive amounts of resources have been designing, developing, and managing their own distributed systems for ML. However, even these organizations struggle to achieve their goals cost effectively, and their efforts to build scalable compute infrastructures has resulted in being late to market.

This is because AI distributed systems are far more complex than other distributed systems. A recommendation system, for example, must include data ingestion, preprocessing, training, hyperparameter tuning, serving, and business logic. Companies often end up stitching together many different distributed systems (e.g. Spark for data preprocessing, TensorFlow for training, and so on) — each with its own API, characteristics, and data formats. This exacerbates the complexity in development, deployment, and management.

RAY

Scale AI and Python applications with Ray, a unified framework for scalable computing

Scaling AI and Python Workloads Effortlessly with Ray

Ray Simplifies Distributed Computing

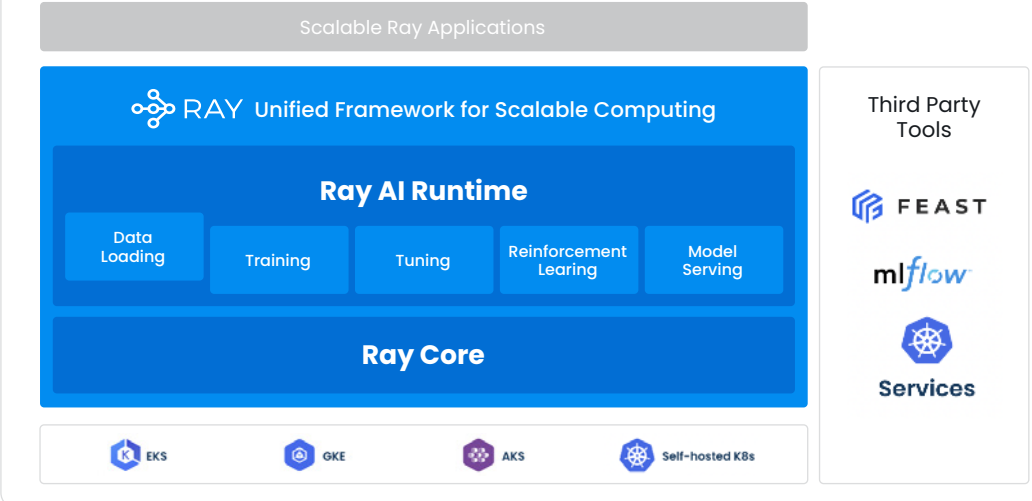
Ray, a unified compute framework, addresses these challenges head on by allowing ML engineers and developers to scale their workloads effortlessly from their laptops to the cloud without the need to build complex compute infrastructures.

Ray is an open-source, unified compute framework that makes it easy to scale AI and Python workloads. It's a flexible Python-native distributed computing framework with primitives to easily parallelize existing AI and Python applications on a laptop and to scale to a cluster on the cloud or on-premises with no code changes.

Ray includes the Ray AI Runtime (AIR), a native set of best-in-class scalable ML libraries. These libraries makes it easy to scale the most compute-intensive ML workloads, such as:

- ML data pre-processing tasks via **Ray Data**
- Training large models via **Ray Train**
- Hyperparameter tuning via **Ray Tune**
- Reinforcement learning via **Ray RLlib**
- Batch inference via **Ray Batch Predictor**
- Real-time inference via **Ray Serve**

Additionally, Ray and its libraries seamlessly integrate with the rest of the Python and ML ecosystem. With these libraries, non-experts can easily leverage distributed computing using simple Python APIs and their favorite ML and Python tools. Ray handles all aspects of distributed execution — from scheduling tasks to auto-scaling to fault tolerance and more — so that engineers and researchers can focus on developing application logic instead of learning and operating the internals of a distributed system.



RAY

Scale AI and Python applications with Ray, a unified framework for scalable computing

Scaling AI and Python Workloads Effortlessly with Ray

Advantages and Unique Capabilities

Advantages

Organizations globally are fast moving their AI/ML and Python workloads to Ray with proven success. Across all these organizations, the consistent realization of the power of Ray include:

1. **Effortless scaling.** The ability to scale every workload and unify these workloads on top of a single system makes it easier than ever to scale the most complex AI and Python applications.
2. **Flexibility.** By being open-source and Python-native, Ray makes it easy to integrate every ML library and framework in your applications.
3. **Improved productivity.** Ray allows developers to run the same code on their laptops and at scale on clusters with thousands of machines. This dramatically increases the speed of iterations for developers.

Unique Capabilities

- **Scalable**
  - Ray's parallel and distributed execution primitives and libraries allow developers to scale their existing AI and Python applications with just a couple lines of code.
  - Ray's Pythonic APIs allow developers to iterate quickly on their laptops and transparently scale out on a cluster without refactoring any code.
- **Unified**
  - Ray AIR provides a unified and scalable toolkit for common ML workloads such as distributed training, hyperparameter tuning, inference, and real-time serving.
  - Ray maintains dozens of integrations with popular ML projects such as Scikit-Learn, FastAPI, XGBoost, PyTorch, and TensorFlow, across the entire ML lifecycle making it possible to pick and choose your favorite library to accelerate and scale.
  - Ray provides support for heterogeneous hardware. It supports not only CPUs but also hardware accelerators such as GPUs and TPUs, and gives developers the ability to explicitly request to share CPUs and GPUs for different components of their Ray application as needed.

RAY

Scale AI and Python applications with Ray, a unified framework for scalable computing

Scaling AI and Python Workloads Effortlessly with Ray

Advantages and Unique Capabilities

Unique Capabilities

- **Open**
  - Ray is open source, with an active and thriving community. Community participants include major cloud organizations from Google, Amazon, and Microsoft; ML platform teams from top technology organizations such as Uber and Shopify; up and coming startups such as Cohere, Predibase, and Dendra; and tens of thousands of other research engineers, developers, and data scientists. Thousands of organizations already rely on Ray for scaling.
  - Ray integrates natively with existing ML and data ecosystems. Ray is Python native and is agnostic to ML frameworks, data sources, and other data tools. Users can then load data from Snowflake, Databricks, or Amazon S3. Lastly, they can also track experiments with Weights & Biases, MLflow and other MLOps tools. Ray enables users to leverage their favorite tools at scale.
  - Ray is portable, enabling users to easily run Ray applications anywhere. It provides Kubernetes support for on-premise deployments, and supports native VM-level deployments on all major public clouds, AWS, GCP, and Azure.

RAY

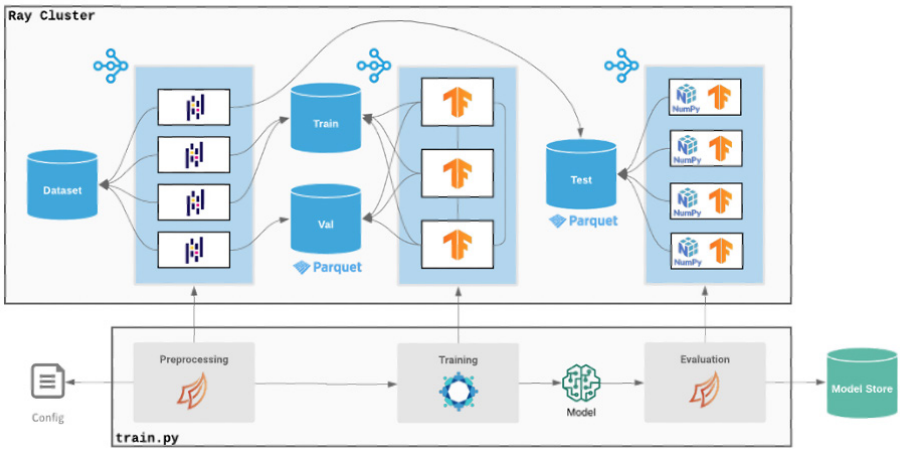
Scale AI and Python applications with Ray, a unified framework for scalable computing

Scaling AI and Python Workloads Effortlessly with Ray

How Major Organizations Are Using Ray

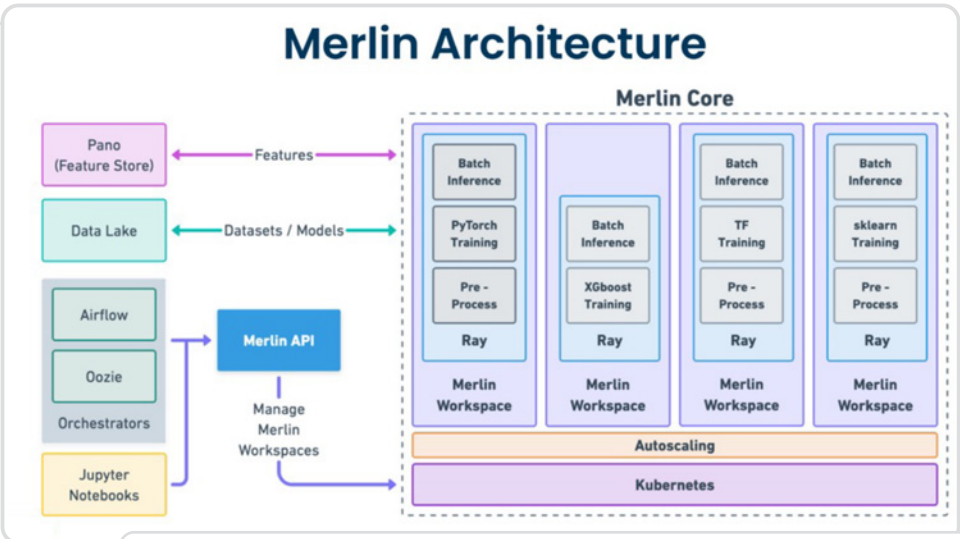
Uber

"By leveraging Ray, we can combine the preprocessing, distributed training and hyperparameter search all within a single job running a single training script."



"At Shopify, our data and compute demands are growing exponentially each year, and our previous tools were struggling to keep up. Our bet on Ray to power our machine learning platform is proving instrumental in our ability to accelerate and scale our entire ML lifecycle. Ray's simple, Pythonic APIs and rich library ecosystem, coupled with its open and extensible design is making it simpler and faster for our engineers and data scientists to deliver value to our 1.7 million+ merchants around the world."

shopify



Merlin, the platform Shopify built on Ray, enables:

- Scalability - ability to scale up machine learning workflows easily
- Fast Iterations - minimizes the cycle between prototyping and production
- Flexibility - ability to leverage any libraries or packages for ML models

RAY

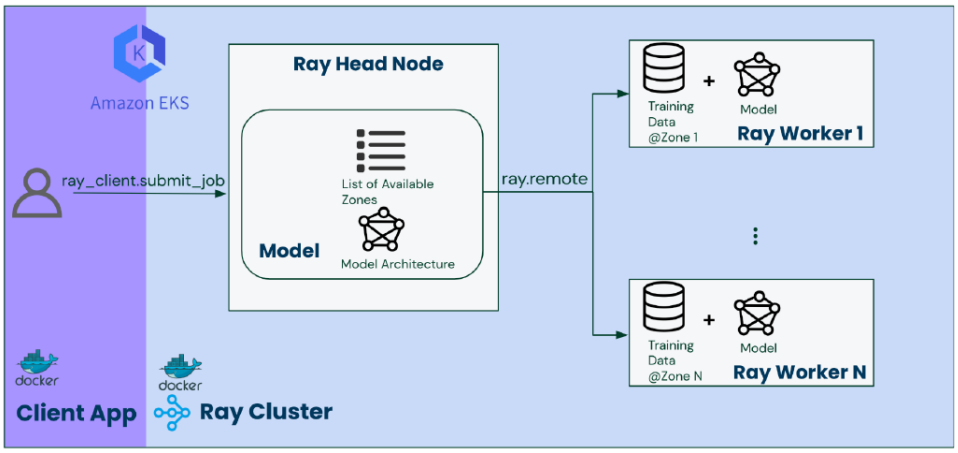
Scale AI and Python applications with Ray, a unified framework for scalable computing

Scaling AI and Python Workloads Effortlessly with Ray

How Major Organizations Are Using Ray

instacart

"We want our machine-learning engineers to focus on the core algorithmic work. Ray enables them to run their models on very large data sets without having to get involved in the details of how to run a model on a large number of machines."



"Using AWS Batch and Celery, It took ~4 hours to train 1,500 training jobs using 10 m6a.4xlarge instances. With the same compute resources we were able to complete the same workload in 20 minutes using Ray."

OpenAI

"At OpenAI, we are tackling some of the world's most complex and demanding computational problems. Ray powers our solutions to the thorniest of these problems and allows us to iterate at scale much faster than we could before."

Greg Brockman, CTO and cofounder, OpenAI

Want to get started with Ray?

Contact us at **info@anyscale.com** or join our community at **ray.io** to start your journey today!