Autodep: Low-Overhead Distributed Model Deployment

Matt Nappo

Model Deployment

- Once a model has been trained/tested/calibrated/etc…
  - How do we provide reliable access to the model?
  - How do we effectively integrate the model into a larger software system?

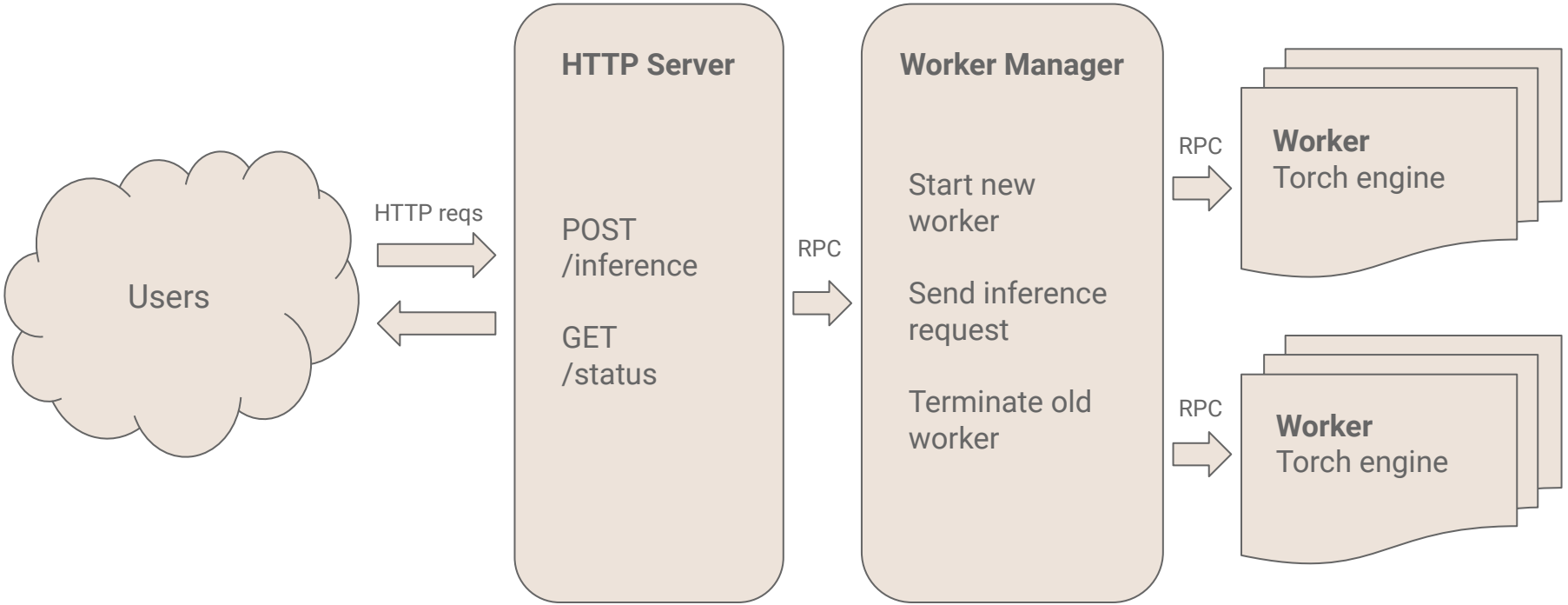
The Problem

- Deploying a model is a very tedious and complicated task
- Naive/simple solutions have many drawbacks
  - Cost, maintenance, bad scalability

Autodep

- A tool that automatically deploys PyTorch models in a scalable way
- Simply specify a TorchScript file, and Autodep automatically spins up an HTTP server providing inference
- Supports:
  - Image Classification models
  - Image-to-Image (seq-to-seq) models
- Entirely written in Rust

System Architecture

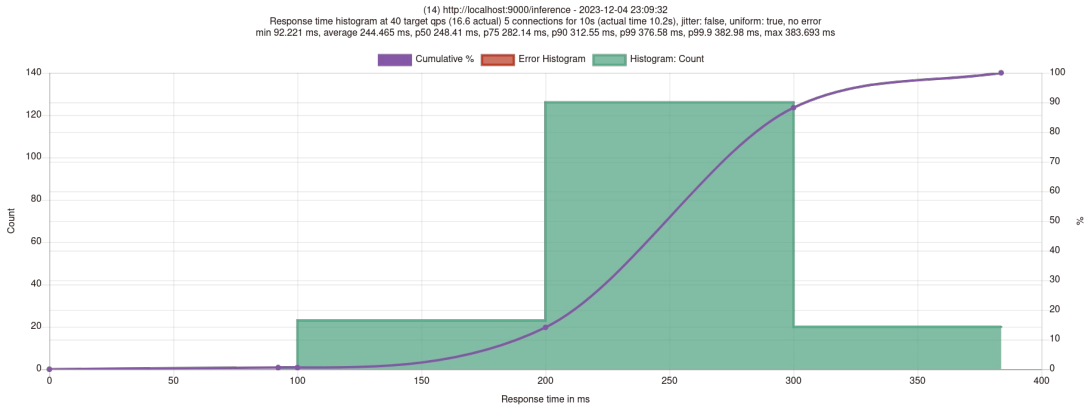


Features

- **Automatic setup** – just supply the TorchScript file
- **Distributed** – runs inferences across a cluster of nodes
  - Communicates over gRPC over HTTP/2
- **Dynamic scaling** – will spin up new workers to meet request demand
- **Fast, memory-safe, asynchronous** – written in Rust, powered by Tokio

Testing

- Image Classification
  - ResNet18
  - ResNet50
- Image segmentation:
  - DeepLab v3



- Measured average latency and benchmarked request throughput

Live Demo

- Image segmentation demo
- Autoscaling demo

Thanks!

Code

github.com/mattnappo/autodep