Onto Petaflops with Kubernetes

Vishnu Kannan
Google Inc.
vishh@google.com
Key Takeaways

Kubernetes can manage hardware accelerators at Scale

Kubernetes provides a playground for ML

ML journey with Kubernetes is just getting started.
Kubernetes Overview

Kubernetes is a multi-workload platform
Supports CPU, Memory and Storage
Simplify packaging
Portable across clouds
Multiple managed solutions exist
Many success stories
Kubernetes Users

- The New York Times
- OpenAI
- Goldman Sachs
- SAP
- Samsung SAMSUNG SDS
- wepay
- SoundCloud
- Home Office
- CONCUR
- Amadeus
- Ancestry
- CCP
- LivePerson
- Monzo
- Box
- Pokémon GO
- Yahoo Japan
- Philips
- Buffer
- Comcast
- Wikimedia Foundation
- Pearson
- Zulily
- eBay
Kubernetes Overview

Master Node

Worker Node

Worker Node
GPUs in Kubernetes

Massively parallel workloads demand GPUs
Some of those workloads need scale and simplified packaging
Kubernetes now supports GPUs
Compute per node increases orders of magnitude
Portability

There are lots of GPUs
Add FPGAs, ASICs, NICs to the mix
Each with its own SW and HW requirements
Container portability is a challenge
Portability

Support multiple versions of
- Kernels
- Drivers
- User facing APIs
- Hardware versions

Not a trivial problem

Kubernetes provides primitives
Better left to Kubernetes vendors
Hardware Device Plugins

Hardware Device Plugins
Kubernetes workloads are expected to be portable
Kubernetes cannot support 100+ hardware natively
Solution for managing infra dependencies at scale
Package SW requirements as a container, manage via Kubernetes
One click HW integration with Kubernetes - Magic!
Hardware vendors author device plugins
User consumes Kubernetes APIs
Hardware Device Plugins

Step 1 - Hardware Vendor publishes device plugins for Kubernetes
Step 2 - Kubernetes cluster admins deploy device plugins

```
kubectl create -f http://vendor.com/device-plugin-daemonset.yaml
```

Step 3 - Device plugin registers with Kubelet (Node agent) and advertises devices
Step 4 - Applications can request the newly available device as a kubernetes resource
Challenges

- Many hardware configurations
- Performance guarantees
- Heterogenous nodes
- Bin packing vs Performance
- Monitoring
- Better Scheduling Primitives
ML Playground with Kubernetes
ML and Kubernetes

ML benefits from Kubernetes and Accelerators

Productionizing ML is a devops problem

Requires large amounts of compute and data

Many dependencies
  - Hardware, libraries, frameworks, language runtimes

Interactive, Streaming and Batch

Various tools and solutions

Difficult to deploy and manage lifecycle
Single Node - two steps to productivity

1. Install kubernetes (kubeadm/minikube)
2. Install kubernetes apps pre-configured with ML stack of choice

Works across distros, OSes
Extensible to other hardware configurations
Portable to the Cloud
Google Cloud Platform

Kubernetes

Operating system (Linux, Windows)

Quota Logging Monitoring Auth

NFS Ceph Cassandra MySQL

Spark ML Airflow Jupyter

Tensorflow Caffe TF-Serving Flask+Scikit

CPU Memory SSD Disk GPU FPGA ASIC NIC

GCP AWS Azure On-prem
Scale out ML Playground

Distributed training
Hyperparameter search
Batch preprocessing
Automate using Kubernetes APIs!
Compose with your tools of choice
Build common ML platforms
Kubernetes on GCP

Container Engine - managed Kubernetes
Auto scaling, Auto upgrades, Auto repairs
Backed by Kubernetes maintainers and Google SREs
Supports Nvidia K80 GPUs (Alpha)
  - More in the future
Tensor Processing Units
ASICS designed for Neural Nets
V1 - Announced in 2016
15 - 30X higher performance
Kubernetes on GCP

Tensor Processing Units
V2 - Announced in 2017
Cloud TPUs
180 TFlops each
Kubernetes on GCP

Tensor Processing Units
V2 - Announced in 2017
Cloud TPUs
180 teraflops each
Combined into Pods
  - 11.5 petaflops
You may not need that much!
That’s ok!
Kubernetes on GCP

- Container Engine - managed Kubernetes
- Nvidia K80s - more in the future
- Tensor Processing Units (TPUs)
- BigQuery - Analyze massively large data
  - fully managed
  - petabyte scale
  - low cost
Kubernetes on GCP

Container Engine - managed Kubernetes
Nvidia K80s - more in the future
Tensor Processing Units (TPUs)
  - ~25x faster for neural net
BigQuery - Analyze massively large data
Global load balancing
  - Global predictions
Kubernetes on GCP

Container Engine - managed Kubernetes
Nvidia K80s - more in the future
Tensor Processing Units (TPUs)
  - ~25x faster for neural net
BigQuery - Analyze massively large data
Global load balancing
Cloud AI - Image, Video, Text, Translation, MLaaS
Demo
Demo Overview

Goal is to show what Kubernetes can provide

1. Interactive ML with Jupyter and Tensorflow on Kubernetes
2. Analyze, train, package and deploy a model from Jupyter on Kubernetes
3. Focus on rapid iteration
4. Customizable to other workflows

`helm install https://github.com/vishh/helm-chart/raw/master/jupyterhub-v0.4.0.tgz`
Deploy Jupyterhub

- Master
- Node 1
- Node 2
Deploy Jupyterhub

Run proxy pod

Run hub pod

Node 1

Node 2
Google Cloud Platform

Master

Node 1

Proxy Service

Proxy

Login

Node 2

Hub Service

Hub

Service
A diagram illustrating a network architecture involving a Master, a Proxy Service, and a Hub Service. The diagram shows:

- A Login request from the Master to a Proxy Service on Node 1.
- The Proxy Service on Node 1 sends a request to the Hub Service.
- The Hub Service on Node 2 is connected to the Hub Service on Node 1.
Login

Proxy Service

Proxy

Node 1

Master

Create notebook pod

Hub Service

Hub

Node 2

Proxy request to hub

Proxy request to hub
Google Cloud Platform
Master
Node 1
Node 2
Proxy
Service
Proxy
Hub
Service
Proxy request to hub
Create TF notebook pod
Run TF notebook pod
Login
Proxy Service
TF notebook
GPU
Login
Hub
Service
Create TF notebook pod
Proxy request to hub
Node 1
Node 2
Login

Proxy Service

Proxy

TF notebook

GPU

Node 1

Proxy request to hub

Master

Hub Service

Hub

Node 2

Proxy request to hub

Redirect to notebook

TF notebook

Redirect to notebook

GPU
Google Cloud Platform

Master

Node 1

Proxy
Service

Proxy

TF
notebook

GPU

Node 2

Hub
Service

Hub
ML and Kubernetes Journey
Existing Solutions

1. Pachyderm
2. Tensorflow Serving - Inception Helm Chart
3. Pipeline AI - Tensorflow and Spark on Kubernetes
4. Clarifai - Webinar
5. Ebay Machine Learning
6. Baidu - Paddle Paddle on Kubernetes
7. Open AI - Tensorflow on Kubernetes
8. Distributed TF training
9. And more...
Accelerated Journey

Enabling Accelerated workloads on Kubernetes is a journey
Contributors across many organizations
Interested in GPUs, join the community
There are problems that needs to be solved across the stack
Be part of a thriving OSS community!
Summary

Kubernetes can manage hardware accelerators at Scale

Kubernetes provides a playground for ML

ML journey with Kubernetes is just getting started. Join the party!
Thank you

@vishh [github, slack]
@vishnukanan [Twitter]