Building Machine Learning Stack on Kubernetes

Atin Sood - ML Tech Lead, IBM Watson
asood@us.ibm.com @soodatin

Sahdev Zala - K8s contributor and org member, Co-chair K8s IBMCloud SIG
spzala@us.ibm.com @sp_zala
Agenda

- AI and ML
- Kubernetes
  - What and Why
- Deep Learning
- Leveraging K8s for ML
  - KubeFlow
  - Fabric for Deep Learning (FfDL or Fiddle)
AI is everywhere
AI is everywhere
AI is everywhere
AI is everywhere
Gartner predicts: By 2020, 85% of CIOs will pilot AI programs

85%
What’s different about ML workloads?
ML Workload Characteristics

- CPU intensive
- Strength of HPC and GPUs
- High memory
- Need to scale workloads as the app grows
Perception of ML Apps

• Lot of code around ML and very little code around infrastructure management, scaling, monitoring, logging, configuration management etc.
• People think that these other things are somehow being taken care of
Reality of ML Apps

- Major burden is around running ML apps in production at scale around the things like
  - how do I deploy it
  - scale it
  - manage it
  - secure it
  - push continuous updates to it
Watson AI workloads on IKS

IBM Watson workloads:
Proven AI workload on IBM Cloud Kubernetes Service

12 Watson services/apps represented as 800+ Kubernetes services

“We no longer worry about managing the infrastructure because IBM Cloud Kubernetes Service takes care of that for us.” – Watson Project Team

One deployment example:
3000+ pods on 500+ nodes
Kubernetes Overview

- **Cloud Native Computing Foundation project**
- Enterprise level container orchestration
- Provision, manage, scale applications (containers) across a cluster
- Declarative model
  - Provide the "desired state" and Kubernetes will make it happen
- Github
  - [github.com/kubernetes/kubernetes](https://github.com/kubernetes/kubernetes)
Kubernetes Cluster

- A running Kubernetes cluster contains a cluster control plane (AKA *master*) and worker node(s), with cluster state backed by a distributed storage system (*etcd*). Cluster can be a single node to several nodes.

- Kubernetes can run on various platforms – Laptop, VMs, Rack of bare metal servers. The effort required to set up a cluster varies from running a single command to crafting your own customized cluster.
Constrain Cluster Resources / Scheduler Optimization

Requests and limits

```
apiVersion: v1
kind: Pod
metadata:
  name: frontend
spec:
  containers:
    - name: db
      image: mysql
      resources:
        requests:
          memory: "64Mi"
          cpu: "250m"
        limits:
          memory: "128Mi"
          cpu: "500m"
  nodeSelector:
    nvidia.com/gpu: 1
```

Node Affinity

```
spec:
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
      - matchExpressions:
          - key: kubernetes.io/e2e-az-name
            operator: In
            values:
              - e2e-az1
              - e2e-az2
        preferredDuringSchedulingIgnoredDuringExecution:
        - weight: 1
          preference:
            matchExpressions:
              - key: another-node-label-key
                operator: In
                values:
                - S1
              - key: another-node-label-value
        topologyKey: failure-domain.beta.kubernetes.io/zone
```

Pod Affinity

```
spec:
  affinity:
    podAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
      - labelSelector:
          matchExpressions:
            - key: security
              operator: In
              values:
              - S1
          topologyKey: failure-domain.beta.kubernetes.io/zone
      preferredDuringSchedulingIgnoredDuringExecution:
      - weight: 100
        podAffinityTerm:
          labelSelector:
            matchExpressions:
              - key: security
                operator: In
                values:
                - S2
          topologyKey: kubernetes.io/hostname
```

Node label

Schedule pod to a node per this GPU constraint

- 64 MiB
- 250 millicore
- millicpu
What is Deep Learning?

- python programs executed via:
  - notebooks
  - scripts
- tend to deal with a lot of data
- tend to run for long periods of time
- tend to require a lot of setup, esp. with gpus
Why ??

- Deep learning on containers
- Deep learning on Kubernetes
- Deep learning on cloud
How??

• Floydhub
• Kubeflow
• FfDL
Kubeflow

- Developed by Google and contributions from others
- Training and Inferencing
- Expects the user to have knowledge of kubernetes
- umbrella project for other DL related projects as well
Floydhub

- Focused on training
- Abstracts the notion of underlying infrastructure
- cli and jupyter lab support
FfDI

• Developed by IBM
• Abstracts the notion of underlying infrastructure
• cli UI and notebooks
• run a training
  – ffdl train <manifest file location> <model definition zip | model definition directory>

• manifest file

```
name: tf-convolutional_network_tutorial:
description: Convolutional network model using tensorflow
version: "1.0"
gpus: 0
cpus: 0.5
memory: 10G
learners: 1

data_stores:
  - id: sl-internal-os
type: mount_cos
training_data:
  container: tf_training_data
training_results:
  container: tf_trained_model
connection:
  auth_url: http://s3.default.svc.cluster.local
  user_name: test
  password: test

framework:
name: tensorflow
version: "1.5.0-py3"
command: >
  python3 convolutional_network.py --trainImagesFile $(DATA_DIR)/train-images-idx3-ubyte.gz
  --trainLabelsFile $(DATA_DIR)/train-labels-idx1-ubyte.gz --testImagesFile $(DATA_DIR)/t10k-images-idx3-ubyte.gz
  --testLabelsFile $(DATA_DIR)/t10k-labels-idx1-ubyte.gz --learningRate 0.001
  --trainingIters 2000
```
Behind the scenes

FfDL: Architecture
Lessons learnt

• What we did right

• What we did wrong

• Roadmap