Kubernetes native two-level resource management for AI workloads

Diana Arroyo and Alaa Youssef
IBM Research

October 2020
Agenda
Why the Kubernetes scheduler is not enough

Desired capabilities

MCAD - open source project

How it works

Demo

Call to action
Why the Kubernetes scheduler is not enough

Desired capabilities

MCAD - open source project

How it works

Demo

Call to action
Why the Kubernetes Scheduler Is Not Enough for Your AI Workloads

KubeCon + CloudNativeCon sponsor guest post from Alaa Youssef, manager of the Container Cloud Platform at IBM Research
Some Characteristics of AI Workloads

- Multiple concurrent learners or executors
- Co-location/affinity preferences
- Massively parallel; big number of short running tasks
- Resource hungry
- Elastic jobs
- Elapsed time may vary while total consumed resources remain the same
- Increasing trend of interactive use cases
Managing Jobs vs Tasks

• Tasks map to PODs
• Jobs are composed of many tasks
• At which level do you set...
  • Priorities
  • Classes of service
  • Quotas
• Task, job, user, org, service, ...?
• At which level do you queue, allocate resources, preempt, ...
• What happens to your scheduler when 1000s of PODs are pending? ...
  ... Overwhelming!!
Kubernetes Pod Based Scheduling

- pods
- scheduler
  - filter()
  - prioritize()
  - select()
- nodes
Application/Job Pod Scheduling

pods

scheduler

filter() prioritize() select()

Application Job1: 4 GPU Demand
Application Job2: 4 GPU Demand
Application Job3: 4 GPU Demand

Each Pods Requires 1 GPU
Application/Job Pod Scheduling Problem

- filter()
- prioritize()
- select()

Each Pod Requires 1 GPU

- Application Job1: 4 GPU Demand
- Application Job2: 4 GPU Demand
- Application Job3: 4 GPU Demand
Cluster Resources Are Limited

• Scaling up the cluster by adding nodes takes time
• These jobs are resource hungry and are willing to consume any resources you provide
• Practically, there is a limit on available resources, at every moment, even if not permanent

“The sky is the limit”, but this is only a cloud!
Multi-cluster, Hybrid Cloud, and Edge

- Organizations own tens of clusters
- Smaller clusters are easier to manage
- Static assignment of users or apps to clusters is not efficient
- Bursting from on-prem to public is needed more often and less predictable for AI workloads
- The rise of edge computing paradigm introduces more clusters to manage and choose from for running a job
Why the Kubernetes scheduler is not enough

Desired capabilities

MCAD - open source project

How it works

Demo

Call to action
Holistic Application/Job Queuing/Dispatching

Queue Controller

Applicaiton Job1: 4 GPU Demand
Each Pod Requires 1 GPU

Applicaiton Job2: 4 GPU Demand

Applicaiton Job3: 4 GPU Demand

scheduler

filter()
prioritize()
select()
More Desired Capabilities

- Multi-cluster dispatching
- Priorities and classes of service (Gold/Silver/Bronze).
- Hierarchical quota management – soft and hard quotas, multiple resources
- Preemption
- Unified view of jobs belonging to multiple AI/ML frameworks – AppWrapper
Why the Kubernetes scheduler is not enough

Desired capabilities

MCAD - open source project

How it works

Demo

Call to action
What is MCAD?

Multi-Cluster Application Dispatcher

OperatorHub.io

AI/ML Job

https://github.com/IBM/multi-cluster-app-dispatcher
Why the Kubernetes scheduler is not enough

Desired capabilities

MCAD - open source project

How it works

Demo

Call to action
MCAD Controller

• AppWrapper Custom Resource Definition
  • Wraps All Compute Resource Consuming Kubernetes Objects for a Job
    • E.g. Deployments, Pods, Statefulsets
  • Wraps All Non-Compute Resource Consuming Kubernetes Objects for a Job
    • E.g. Services, Namespaces

• MCAD Custom Resource Controller
  • Determines Runnable Job for All Kubernetes Objects within an AppWrapper Holistically
  • Unwraps and Dispatches All Kubernetes Objects within an AppWrapper when Job is Runnable
  • Queues Non-Runnable Jobs
  • Supports Preemption/Requeuing
Multi-Cluster Application Dispatcher

2 Runtime Configurations
• Standalone
• Dispatcher/Agent
Standalone: Multi-Cluster Application Dispatcher

Batch Jobs

MCAD Controller

Priority Queue

Available Compute Capacity

K8s POD

K8s Stateful-Set (leaners)

Status: queued, starting, running

AppWrapper

Priority Queue

K8s State

Kubernetes Infrastructure

klet klet klet klet klet

AW status, lifecycle mgmnt
Multi-Cluster: Dispatcher/Agent

K8s Replica-Set) – K8s Deployment – K8s Service – PVC

MCAD Dispatcher

Priority Queue

(AppWrapper status, lifecycle mgmt)

Available Capacities per Cluster

K8s State

Agent K8s Cluster

MCAD Agent

K8s State

Agent K8s Cluster

MCAD Agent
Roadmap: Current Work - Quota Management

Quota Container

Quota Management

ReST Server

Y/N + Preempt List

/allocate

Available Capacity

MCAD Controller

Queue

MCAD Pod

K8s Quota Tree

Custom Resources

K8s State

Context Quotas

Org 1

Context C

Context D

Context A

Context B

Quota Tree CR

NEW
AppWrapper Yaml

```yaml
kind: AppWrapper
metadata:
  name: job1-context2-1replica
labels:
  quota_context: Context-2

spec:
  priority: 1000
  resources:
    Items:
      - replicas: 1
        type: StatefulSet
    template:
      apiVersion: apps/v1
      kind: StatefulSet
      metadata:
        name: job1-context2-1replica
        labels:
          app: job1-context2-1replica
```
Why the Kubernetes scheduler is not enough

Desired capabilities

MCAD - open source project

How it works

Demo

Call to action
Why the Kubernetes scheduler is not enough

Desired capabilities

MCAD - open source project

How it works

Demo

Call to action
We Need Your Help!!

Try it out!
Give feedback!
Contribute!

https://github.com/IBM/multi-cluster-app-dispatcher