Kubernetes loves machine learning on on-premise

Hui Luo - VMware

About me

Software engineer at VMware cloud native application team.

Active contributor to upstream kubernetes in area like device plugin.

Contributor at vSphere cloud provider, cluster api vSphere.

Github: @figo

Machine learning on k8s landscape

Kubeflow

Kubernetes

CPU | Memory | GPU | Storage | Network

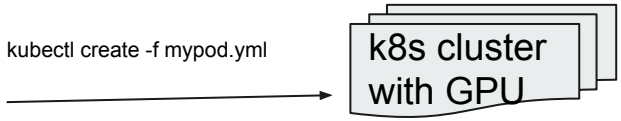
Google Cloud | amazon web services | Azure | vmware

Major aspects of GPU resource

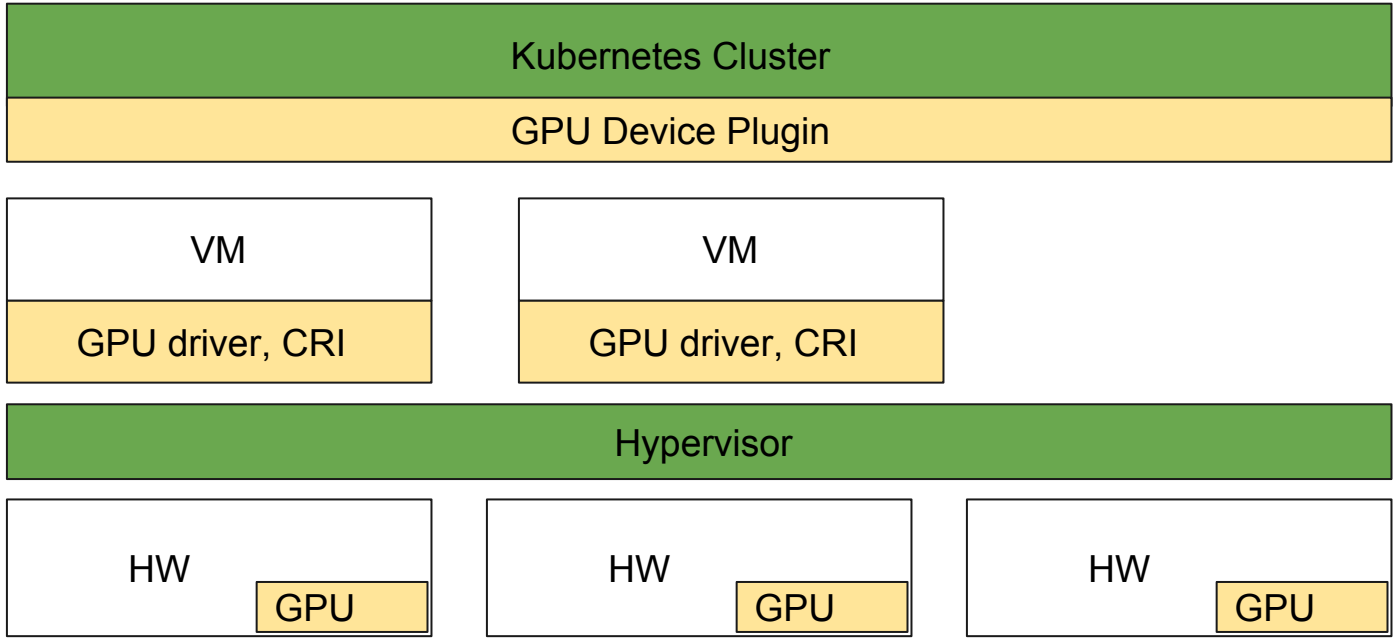
1. **Lifecycle management: setup, update, upgrade, auto-scaling**
2. Sharing and Isolation
3. Monitoring
4. Heterogeneous GPU types
5. Performance consistency

GPU resource in k8s

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: my-gpu-pod
spec:
  containers:
    - name: image-processor
      image: gcr.io/image-processor:latest
      resources:
        limits:
          nvidia.com/gpu: 1
```



Lifecycle management



Lifecycle management - Cont.

DIY solution

Use existing process and build automation solution by yourself.

VS

Vendor solution

Many choices exist

Sharing and isolation

Tips:

1) Use namespace and **GPU Quota**
2) Use Pod PriorityClass and Pod QoS

Note: unlike CPU, it does not support milicore

GPU resource monitoring

```
//AcceleratorStats contains stats of accelerators that attached to
container
type AcceleratorStats struct {
      Make string `json:"make"`
      Model string `json:"model"`
      ID string `json:"id"`
      MemoryTotal uint64 `json:"memoryTotal"`
      MemoryUsed uint64 `json:"memoryUsed"`
      DutyCycle uint64 `json:"dutyCycle"`
}
```

To make it extendable: [KEP] Compute device assignment

Homogeneous to heterogeneous

nvidia tesla k80 + p100?

Solutions:

1) [KEP] Resource api
2) Use labels

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: my-gpu-pod
spec:
  containers:
    - name: image-processor
      image: gcr.io/image-processor:latest
      resources:
        limits:
          gpu-gold: 1
```

Performance consistency

CPU manager, hugepage are supported

To further address NUMA and device locality requirement:

1) [KEP] NUMA manager
2) Hypervisor NUMA scheduler
3) Linux AutoNUMA

Join discussions at:
wg-machine learning
wg-resource management
sig-node

Contact me on github: @figo

References

1. [KEP] Compute device assignment https://github.com/kubernetes/community/pull/2454
2. [KEP] Resource api, kubernetes/community/keps/sig-node/00014-resource-api.md
3. [KEP] NUMA manager kubernetes/community/contributors/design-proposals/node/numa-manager.md
4. CPU manager https://kubernetes.io/blog/2018/07/24/feature-highlight-cpu-manager/
5. Hugepage https://kubernetes.io/docs/tasks/manage-hugepages/scheduling-hugepages/
6. Pod PriorityClass https://kubernetes.io/docs/concepts/configuration/pod-priority-preemption/

Thank you