ACCELERATING THE ADOPTION OF HARDWARE ACCELERATORS IN K8S

Software defined Data Centre solutions Group
SWATI SEHGAL (swati.sehgal@intel.com)
AGENDA

Device Plugin Overview
Intel® QuickAssist Technology Device Plugin
Flow of QuickAssist Device Plugin enablement in k8s
Demo
**DEVICE PLUGINS OVERVIEW**

**CHALLENGES**
- Device vendors have to write custom Kubernetes code in order to integrate their device with the ecosystem.
- Results in multiple vendors maintaining custom code making it difficult for a customer to consume.

**RESOLUTION**
- Device plugin framework provides a vendor independent solution for discovery, advertisement, allocation and health check of external devices.
- Registers with kubelet via Registration gRPC service.
- A plugin is a simple gRPC server that implements RPCs.
- Plugins can be easily deployed (manually/ as a Daemon set).
- Device requests are made via extended resource requests in the Pod.

**BENEFITS**
- A consistent, general and portable framework for users to consume hardware devices across k8s clusters.

**REFERENCE:** [https://kubernetes.io/docs/concepts/cluster-administration/device-plugins/](https://kubernetes.io/docs/concepts/cluster-administration/device-plugins/)

*Other names and brands may be claimed as the property of others.*
K8S DEVICE PLUGIN IMPLEMENTATION

REFERENCE: https://github.com/kubernetes/community/blob/master/contributors/design-proposals/resource-management/device-plugin.md

*Other names and brands may be claimed as the property of others.*
Intel® QuickAssist Technology
• Intel® QAT is an acceleration technology for cryptography and compression
• It benefits applications in Cloud, Networking, Big data and Storage

Intel® QAT support in K8s
• QAT support enabled through Device plugin framework
• QAT Device Plugin discovers QAT support on a node and the number of VFs configured, advertises this to the node and allocates VFs based on workload resource requests

REFERENCE: https://kubernetes.io/docs/concepts/cluster-administration/device-plugins

*Other names and brands may be claimed as the property of others.
Flow of QAT Device Enablement in K8S

Device Plugin gRPC Server Creation
Device Plugin Registration
Discovery QAT Devices
Device binding/unbinding
Advertisement of devices, Capacity Updated
Workload requesting QAT device

K8s NODE
qatdeviceplugin
Registration (resourceName, device socket, api)
QAT virtual functions: 32
kubelet.sock
KUBELET

Capacity: qat.intel.com/generic:32

K8s NODE
qatdeviceplugin
Registration (resourceName, device socket, api)
QAT virtual functions: 0
kubelet.sock
KUBELET

Capacity: qat.intel.com/generic:0

K8s NODE
qatdeviceplugin
Registration (resourceName, device socket, api)
qat.sock
kubelet.sock
KUBELET

Capacity: qat.intel.com/generic:0

K8s MASTER
ETCD
CONTROLLER MANAGER
API-SERVER
SCHEDULER

K8s MASTER

kind: Pod
Spec:
  Container requests:
    qat.intel.com/generic:1

*Other names and brands may be claimed as the property of others.
FLOW OF QAT DEVICE ENABLEMENT IN K8S

Device Plugin gRPC Server Creation
Device Plugin Registration
Discovery of QAT Devices
Device binding/unbinding
Advertisement of devices, Capacity Updated
Workload requesting QAT device
Device Allocation

Other names and brands may be claimed as the property of others.
CONCLUSION

For the code and more information please refer to the repository
https://github.com/intel/intel-device-plugins-for-kubernetes.git
Talk to us about your applications
THANK YOU