OMEC in a Kubernetes Orchestrated Environment

Saikrishna Edupuganti

Disclaimer

This presentation contains the general insights and opinions of Intel Corporation (“Intel”). The information in this presentation is provided for information only and is not to be relied upon for any other purpose than educational. Use at your own risk! Intel makes no representations or warranties regarding the accuracy or completeness of the information in this presentation. Intel accepts no duty to update this presentation based on more current information. Intel is not liable for any damages, direct or indirect, consequential or otherwise, that may arise, directly or indirectly, from the use or misuse of the information in this presentation.

Intel technologies’ features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel, the Intel logo and Xeon are trademarks of Intel Corporation in the United States and other countries.

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

Agenda

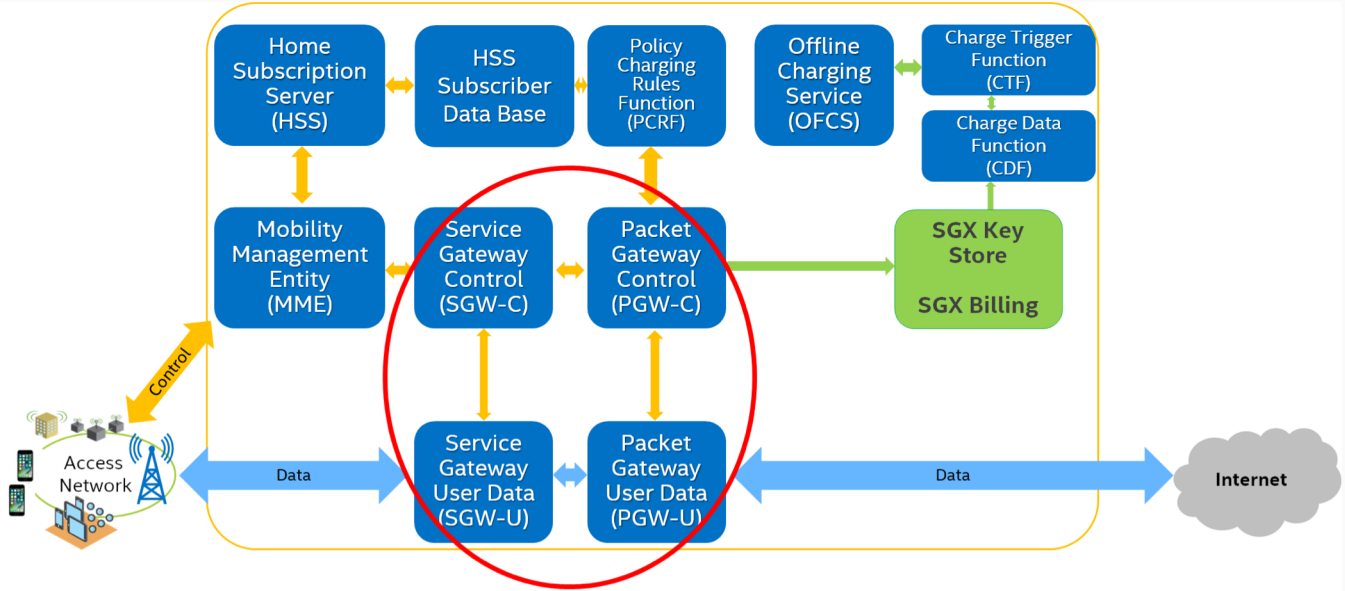
Enabling OMEC Network Functions

Current status

Operator

Enabling OMEC Network Functions

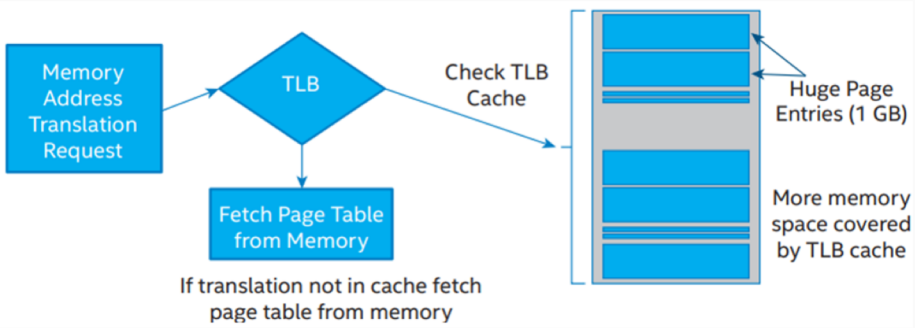
OMEC



- **Disaggregated Control & Data Plane**
- **I/O intensive, multi-interface data plane**

Enabling OMEC Network Functions

Hugepages




```
apiVersion: v1
kind: Pod
metadata:
  name: hugepages-example
spec:
  containers:
  - image: fedora
    command:
    - sleep
    - inf
    name: example
    volumeMounts:
    - mountPath: /hugepages
      name: hugepage
    resources:
      limits:
        hugepages-2Mi: 100Mi
        memory: 200Mi
      requests:
        memory: 100Mi
  volumes:
  - name: hugepage
    emptyDir:
      medium: HugePages
```

Enabling OMEC Network Functions

CPU Isolation/Affinity


```
apiVersion: kubelet.config.k8s.io/v1beta1
kind: KubeletConfiguration
# Allowing for CPU pinning and isolation in case of guaranteed QoS class
cpuManagerPolicy: static
```

```
apiVersion: v1
kind: Pod
metadata:
  name: grtd-qos-example
spec:
  containers:
  - image: fedora
    command:
    - sleep
    - inf
    name: example
    resources:
      limits:
        memory: 200Mi
        cpu: 2
      requests:
        memory: 200Mi
        cpu: 2
```

Enabling OMEC Network Functions

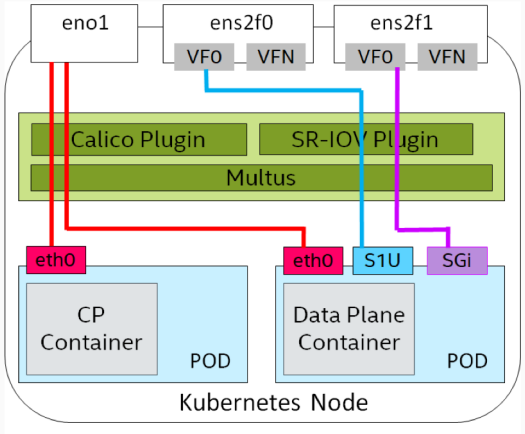
Multiple Network Interfaces

Multus: Intel, RedHat driven effort

Allows Pod to have multiple interfaces

Calls other CNIs but reports back to k8s/CRI result of primary CNI

```
apiVersion: v1
kind: Pod
metadata:
  name: multi-net-example
  annotations:
      k8s.v1.cni.cncf.io/networks: '[
            { "name": "s1u-net", "interface": "s1u" },
            { "name": "sgi-net", "interface": "sgi" }
    ]'
spec:
  containers:
  - name: busy
    image: busybox
    command: [ "top" ]
```



Enabling OMEC Network Functions

Multiple Network Interfaces

Bypasses k8s, so BYO service discovery, load-balancing, network policy

Alternatives

- nokia/danm
- networkservicemesh (cisco)

```
---
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: s1u-net
  annotations:
    k8s.v1.cni.cncf.io/resourceName: intel.com/sriov_vfio
spec:
  config: '{
    "type": "vfioveth",
    "name": "s1u-net",
    "ipam": {
        "type": "host-local",
        "subnet": "11.1.1.0/24",
        "rangeStart": "11.1.1.2",
        "rangeEnd": "11.1.1.63",
        "gateway": "11.1.1.1"
    }
}'
---
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: sgi-net
  annotations:
    k8s.v1.cni.cncf.io/resourceName: intel.com/sriov_vfio
spec:
  config: '{
    "type": "vfioveth",
    "name": "sgi-net",
    "ipam": {
        "type": "host-local",
        "subnet":"13.1.1.0/24",
        "rangeStart": "13.1.1.2",
        "rangeEnd": "13.1.1.63",
        "gateway": "13.1.1.10"
    }
}'
```

Enabling OMEC Network Functions

High-speed Network IO

Currently SR-IOV the only choice: Linux/DPDK

Device plugin helps with scheduling

Need Multus or enlightened CNI, to connect device plugin allocated with netdev to move into netns

Used only as secondary interface to the pod, so same limitations - bypasses k8s, BYO ...

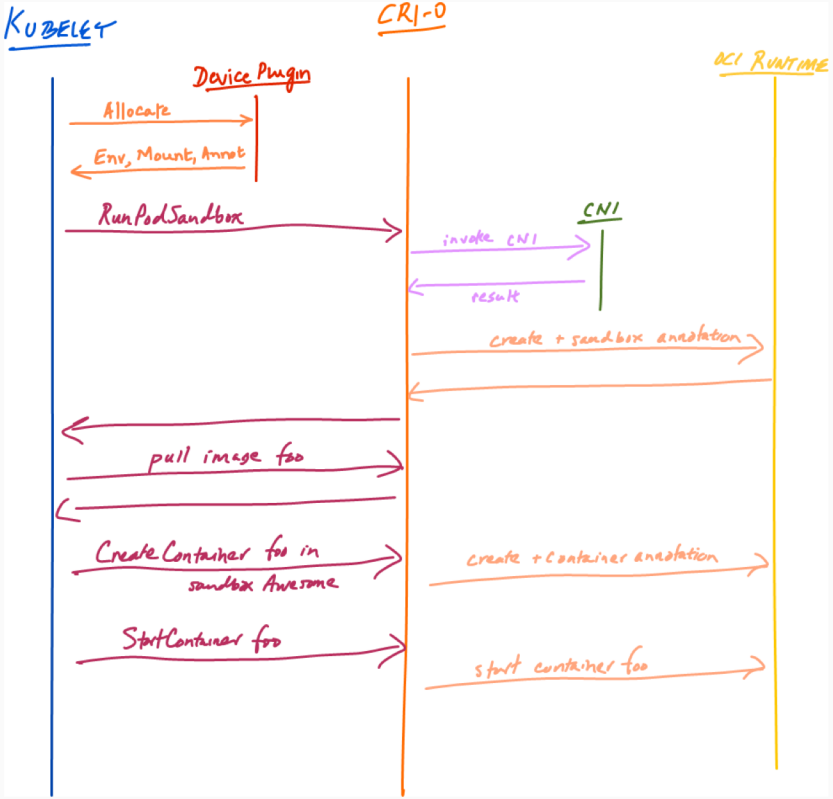
```
---
apiVersion: v1
kind: Pod
metadata:
  name: test-sriov
  annotations:
      k8s.v1.cni.cncf.io/networks: sriov-net
spec:
  containers:
  - name: busy
    image: busybox
    command: [ "top" ]
    resources:
      limits:
        intel.com/sriov_netdevice: '1'
---
apiVersion: v1
kind: Pod
metadata:
  name: test-sriov-dpdk
  annotations:
      k8s.v1.cni.cncf.io/networks: sriov-net-dpdk
spec:
  containers:
  - name: busy
    image: busybox
    command: [ "top" ]
    resources:
      limits:
        intel.com/sriov_vfio: '1'
```

Enabling OMEC Network Functions

High-speed Network IO

Device plugin helps with scheduling

Need Multus or enlightened CNI, to move device plugin allocated netdev into netns



Enabling OMEC Network Functions

High-speed Network IO

Implemented **vfioveth** CNI for DPDK mode

Create representor veth pair inside netns

Set MAC, IP & routes on one end

DPDK app uses the other end to send/receive control plane packets to/from the kernel

```
---
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: sriov-net
  annotations:
    k8s.v1.cni.cncf.io/resourceName: intel.com/sriov_netdevice
spec:
  config: '{
    "type": "sriov",
    "name": "sriov-net",
    "ipam": {
        "type": "host-local",
        "subnet": "198.19.0.0/24",
        "rangeStart": "198.19.0.100",
        "rangeEnd": "198.19.0.200",
        "gateway": "198.19.0.1"
    }
}'
---
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: sriov-net-dpdk
  annotations:
    k8s.v1.cni.cncf.io/resourceName: intel.com/sriov_vfio
spec:
  config: '{
    "type": "vfioveth",
    "name": "sriov-net",
    "ipam": {
        "type": "host-local",
        "subnet": "198.19.0.0/24",
        "rangeStart": "198.19.0.100",
        "rangeEnd": "198.19.0.200",
        "gateway": "198.19.0.1"
    }
}'
```

Enabling OMEC Network Functions

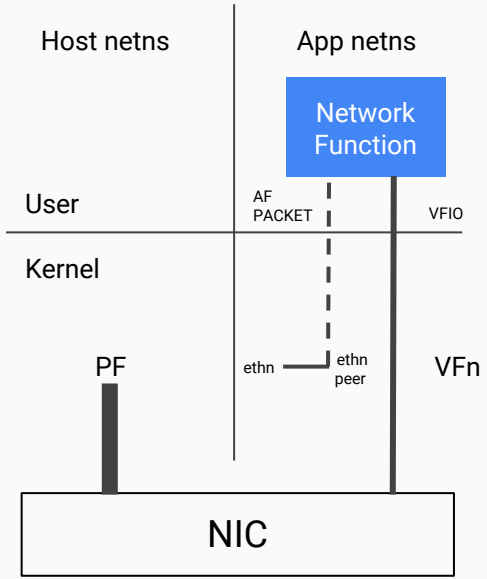
High-speed Network IO

Implemented **vfioveth** CNI for DPDK mode

Create representor veth pair inside netns

Set MAC, IP & routes on one end

DPDK app uses the other end to send/receive control plane packets to/from the kernel



Enabling OMEC Network Functions

Manual/bare-metal vs Automated/k8s

Test	Usr Sp Drv	Pinning	Huge	Pkts/sec*	*(w/noise)*
Native	yes	yes	yes	**1,550K**	*(1,100K)*
Kubernetes	yes	yes	yes	**1,450K**	*(1,150K)*
Kubernetes	**no**	yes	yes	**750K**	*(650K)*
Kubernetes	yes	**no**	yes	**1,450K**	***400K***
Kubernetes	yes	yes	**no**	**1,200K**	*(1,100K)*

NUMA impact untested. Expected to land in k8s 1.16

Agenda

Enabling OMEC Network Functions

Current status

Operator

Current status

Dockerfiles

- SPGW-C, SPGW-U, HSS, HSS-DB, OpenMME

k8s YAMLs

- SPGW-C, SPGW-U in tree omec-project/ngic-rtc/deploy/k8s
- HSS and HSS-DB are in PR omec-project/c3po/pull/21
- OpenMME not present

COMAC-in-a-box

Agenda

Enabling OMEC Network Functions

Current status

Operator

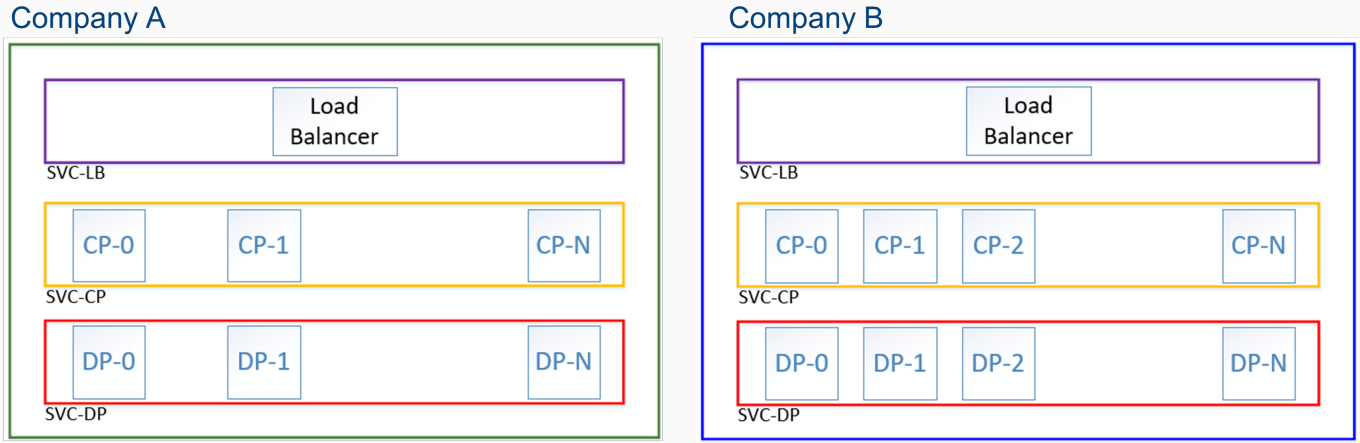
Operator

Custom Resource Definition + Controller

```yaml
apiVersion: ngic.intel.com/v1alpha1
kind: NGIC
metadata:
  name: ngic-att
spec:
  replicas: 2
  name: us-west-1
  mme: 13.1.1.110
  apn: apn1
  teid:
    s11Steps: 12
    s1uSteps: 12
  pool:
    ip: 16.0.0.0
    steps: 12
  ipam:
    s1uNet:
      ip: 2.2.2.0
      mask: 255.255.255.0
      gateway: 2.2.2.92 #RTR_S1U_IP
    sgiNet:
      ip: 3.3.3.0
      mask: 255.255.255.0
      gateway: 3.3.3.105 #RTR_SGI_IP
  dpdk: True
```

Operator

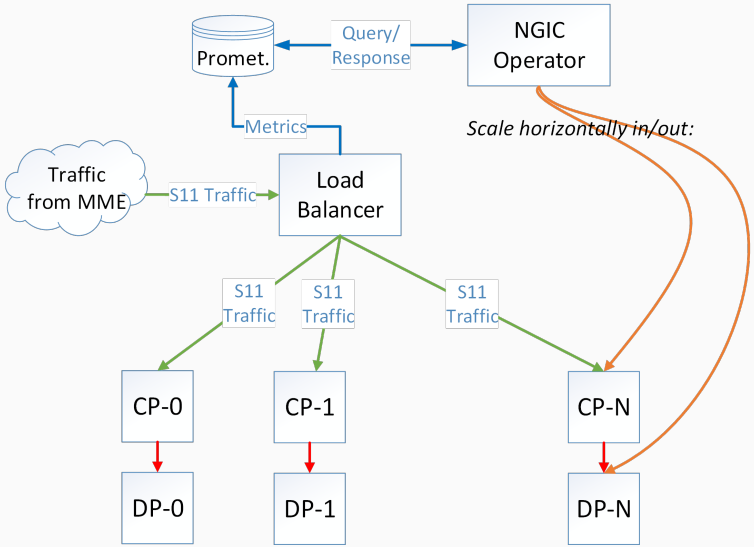
Custom Resource Definition + Controller



Operator

Custom Resource Definition + Controller

- Currently internal
- Instantiate EPC as k8s custom resource, controller will bring up individual components
- Scale SPGW-C and SPGW-U 1:1 based on active session Prometheus metrics
- Implemented S11 TEID load-balancer to pick backend SPGW-C instance



Operator

Custom Resource Definition + Controller