WHAT'S IN THE BOX?
RESOURCE MANAGEMENT IN KUBERNETES

Louise Daly
Cloud Native Orchestration Engineer
louise.m.daly@intel.com

Ivan Coughlan
Cloud Native Orchestration Architect
ivan.Coughlan@intel.com

Special Mention
Balaji Subramaniam

NOTICES AND DISCLAIMERS

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.

All products, computer systems, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice.

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

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Intel processors of the same SKU may vary in frequency or power as a result of natural variability in the production process.

For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice Revision #20110804.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. **No computer system can be absolutely secure.**

Intel® Advanced Vector Extensions (Intel® AVX)* provides higher throughput to certain processor operations. Due to varying processor power characteristics, utilizing AVX instructions may cause a) some parts to operate at less than the rated frequency and b) some parts with Intel® Turbo Boost Technology 2.0 to not achieve any or maximum turbo frequencies. Performance varies depending on hardware, software, and system configuration and you can learn more at http://www.intel.com/go/turbo.

Intel® Hyper-Threading Technology available on select Intel® processors. Requires an Intel® HT Technology-enabled system. Your performance varies depending on the specific hardware and software you use. Learn more by visiting http://www.intel.com/info/hyperthreading.

All SKUs, frequencies, features and performance estimates are **PRELIMINARY** and can change without notice

Results have been estimated based on internal Intel analysis and are provided for informational purposes only. Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit http://www.intel.com/performance. Configurations: Based on Intel estimates.

AGENDA

CURRENT STATUS

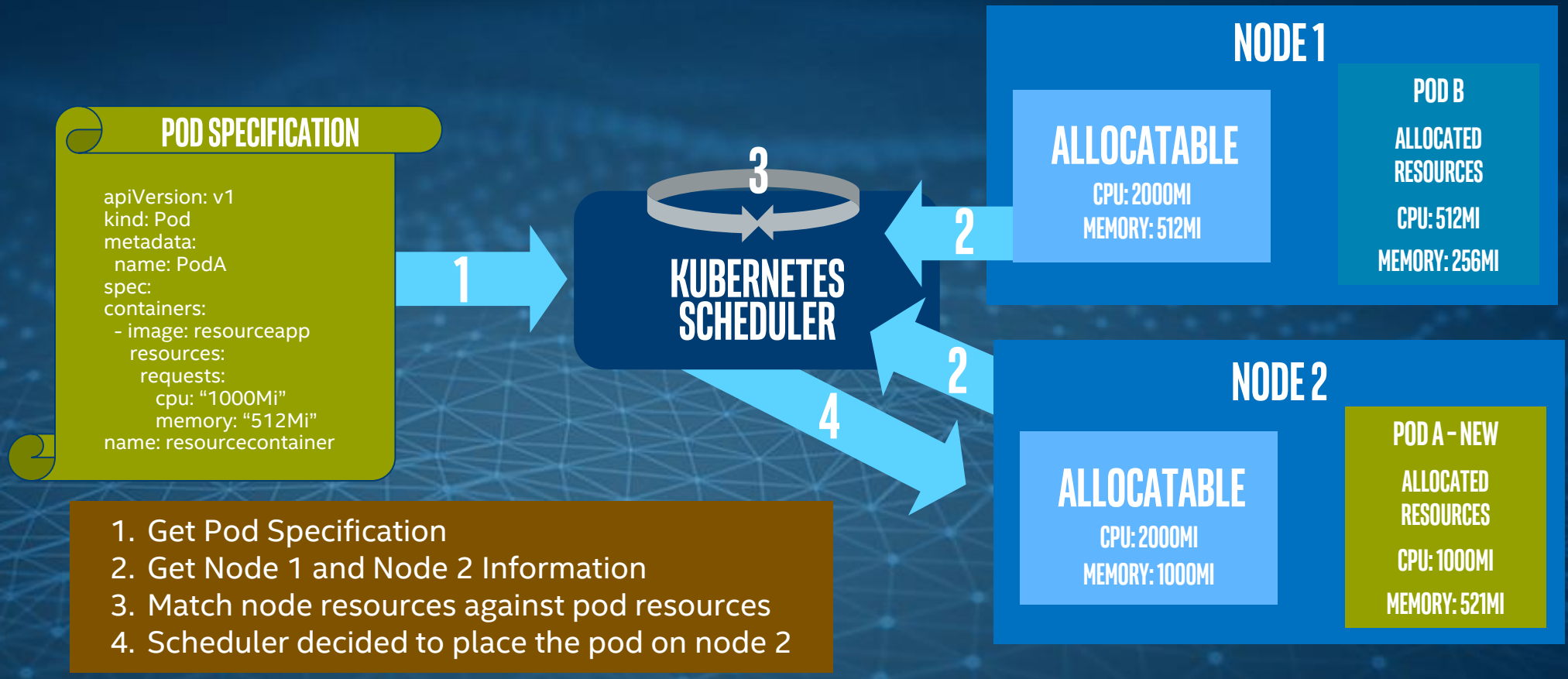
CHALLENGES

STEPS TAKEN

NODE FEATURE DISCOVERY | CPU PINNING | HUGE PAGES | DEVICE PLUGINS | NUMA

DEMO

KUBERNETES RESOURCE MANAGEMENT TODAY



1. Get Pod Specification
2. Get Node 1 and Node 2 Information
3. Match node resources against pod resources
4. Scheduler decided to place the pod on node 2

KUBERNETES RESOURCE MANAGEMENT

PROBLEM STATEMENT

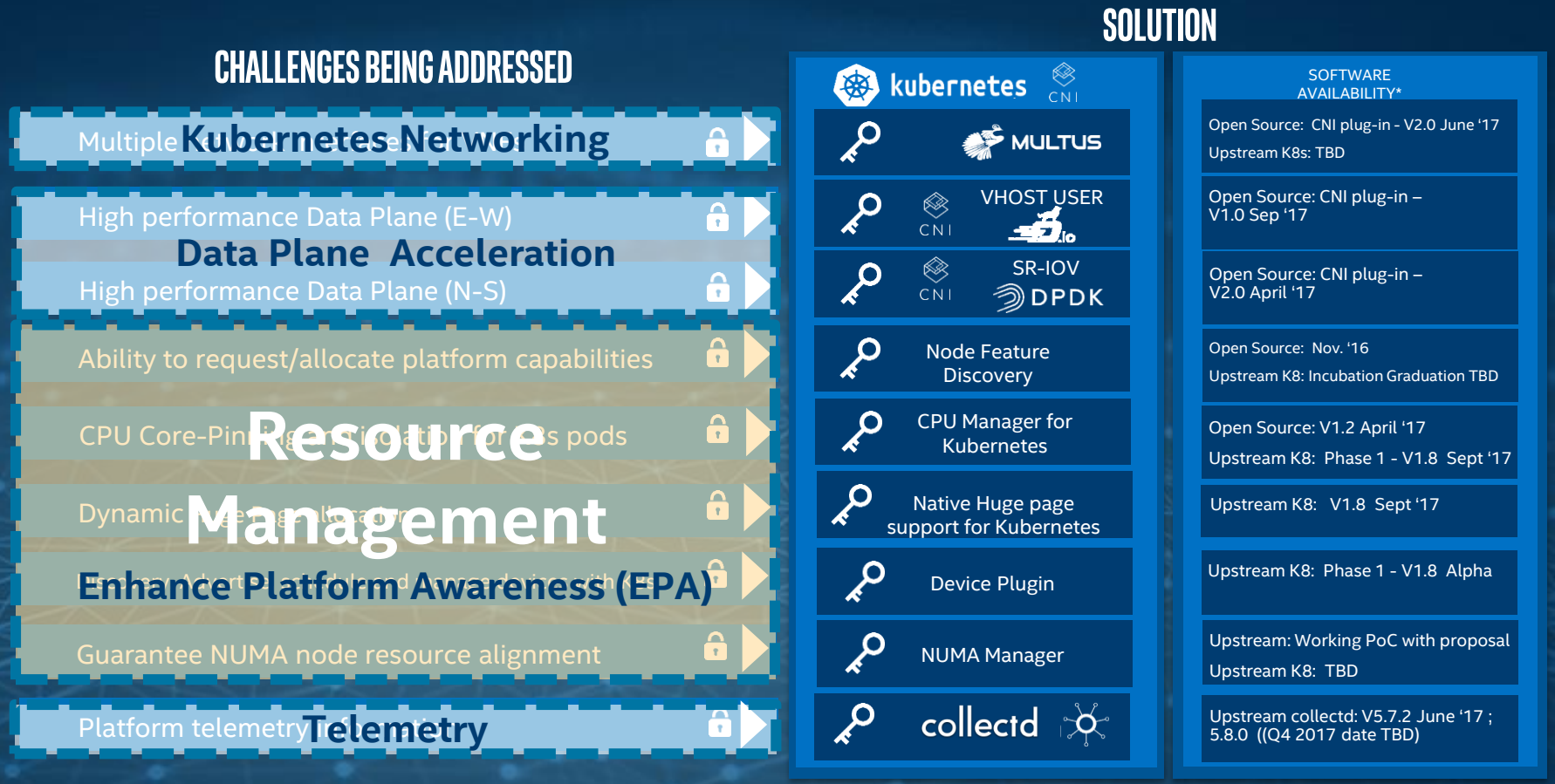
Kubernetes clusters are deployed on a wide array of heterogeneous environments with different hardware resources

Today, CPU and Memory are the core resources orchestrated by Kubernetes. Workloads have a wide variety of hardware resource requirements as well as CPU and Memory but Kubernetes is agnostic to these.

GOAL

Introduce a broader array of resources representing cluster abilities to cater for the wide range of workloads being deployed using Kubernetes

ADDRESS KEY CHALLENGES IN CONTAINERS BARE METAL



Open Source: Available on Intel github https://github.com/Intel-Corp | NFD at https://github.com/kubernetes-incubator/node-feature-discovery

NODE FEATURE DISCOVERY

PROBLEM

No way to identify hardware capabilities or configuration

Inability for workload to request certain hardware feature

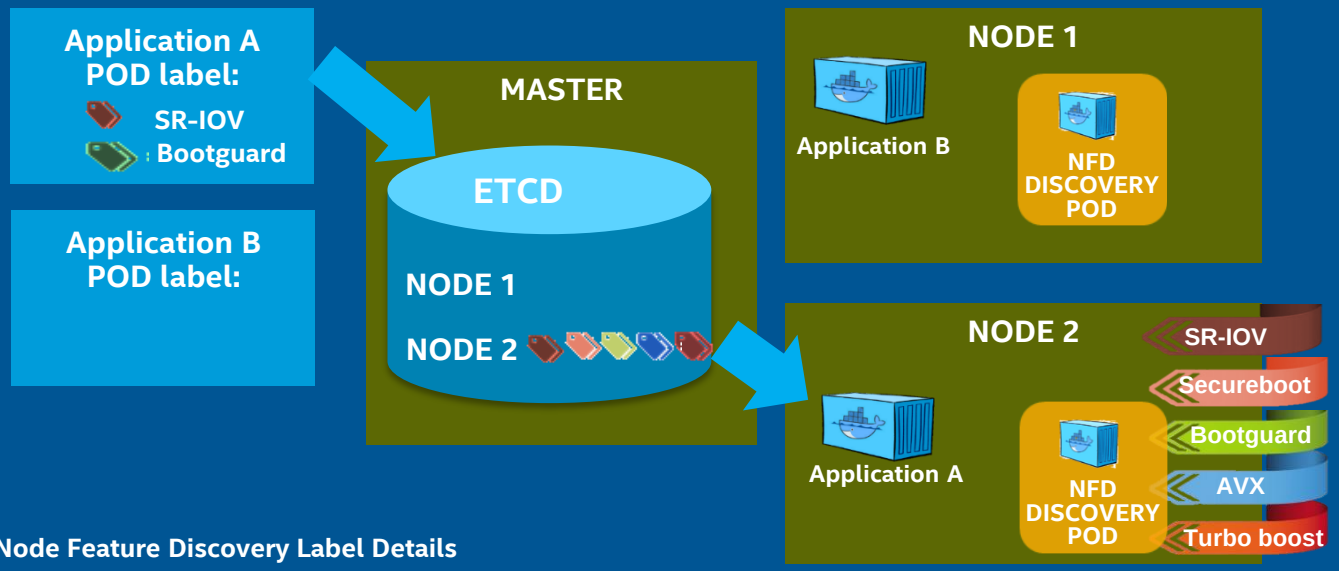
SOLUTION

Node Feature Discovery brings Enhanced Platform Awareness (EPA) in K8s

NFD detects resources on each node in a Kubernetes cluster and advertises those features

Allows matching of workload to platform capabilities

NODE FEATURE DISCOVERY IN K8s



Node Feature Discovery Label Details

SR-IOV	Network Features Single Root I/O Virtualization
BootGuard	A hardware-based boot integrity protection mechanism (New feature on Purley).
UEFI Secure Boot	Boot Firmware verification and authorization of OS Loader/Kernel components
AVX	CPUID Features: Intel® Advances Vector Extensions 512 (Intel® AVX 512)
Turbo Boost	Intel® Turbo Boost Processor accelerator

REFERENCE

github.com/kubernetes-incubator/node-feature-discovery

NFD SECUREBOOT USECASE

PROBLEM

The kernel does not allow IGB_UIO based DPDK applications on UEFI Secure Boot enabled systems

SOLUTION

Using node antiaffinty feature in kubernetes to prevent DPDK application requiring IGB_UIO driver support from landing on nodes with SecureBoot label created by Node feature Discovery

```
apiVersion: v1
kind: Pod
metadata:
  name: dpdkpodRequiringUIOSupport
spec:
 affinity:
    nodeAntiAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
      - labelSelector:
          matchExpressions:
          - key: “nfd-SecureBoot”
            operator: In
            values:
            - “true”
  containers:
  - image: dpdkapp
name: dpdkcontainer
```

CPU MANAGER FOR KUBERNETES – CPU PINNING AND ISOLATION

PROBLEM

Kubernetes has no mechanism to support core pinning and isolation

Results in high priority workloads not achieving SLAs

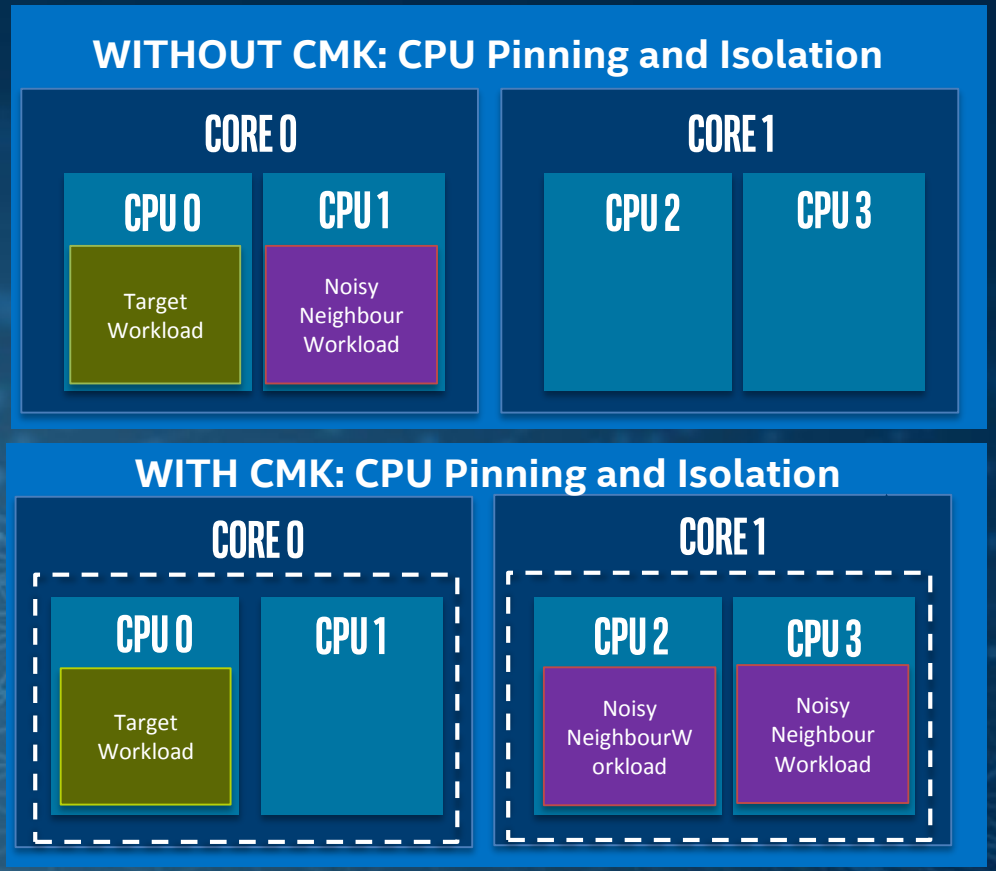
SOLUTION

CPU-Manager-For-Kubernetes introduces core pinning and isolation to K8s without requiring changes to the code base

Gives a performance boost to high priority applications

Negates the noisy neighbour* scenario

*** Noisy Neighbor Workload**: An application that effect causes other virtual applications that share the infrastructure to suffer from uneven performance



REFERENCE

https://github.com/Intel-Corp/CPU-Manager-for-Kubernetes

https://kubernetes.io/docs/tasks/administer-cluster/cpu-management-policies/

HUGE PAGE NATIVE SUPPORT IN KUBERNETES

PROBLEM

No resource management of Huge Pages in Kubernetes

Responsibility of the cluster operator to handle it manually

SOLUTION

Huge Pages introduced as first class resource in Kubernetes

Support for Huge Pages via hugetlbfs enabled through a memory backed volume plugin

Inherent accounting of Huge Pages

Automatic relinquishing of Huge Pages in case of unexpected process termination

REFERENCE

https://kubernetes.io/docs/tasks/manage-hugepages/scheduling-hugepages/

DEVICE PLUGINS OVERVIEW

WHY?

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

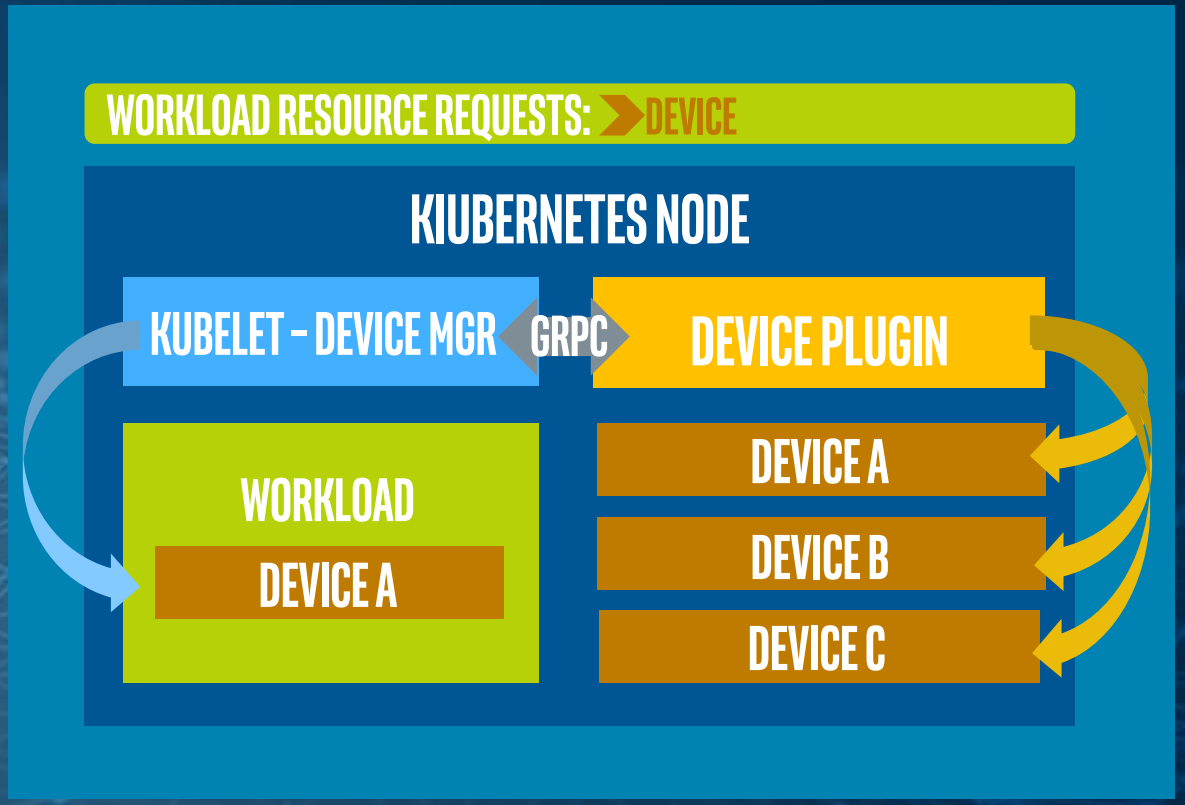
HOW?

Provide a device plugin framework which enables vendors to advertise, schedule and setup devices with native Kubernetes integration

Device Plugins are easily deployed and workload device requests are made via extended resource requests in the Pod Specification

BENEFITS

Enables effective resource utilization



REFERENCE

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

QAT SUPPORT IN KUBERNETES

PROBLEM

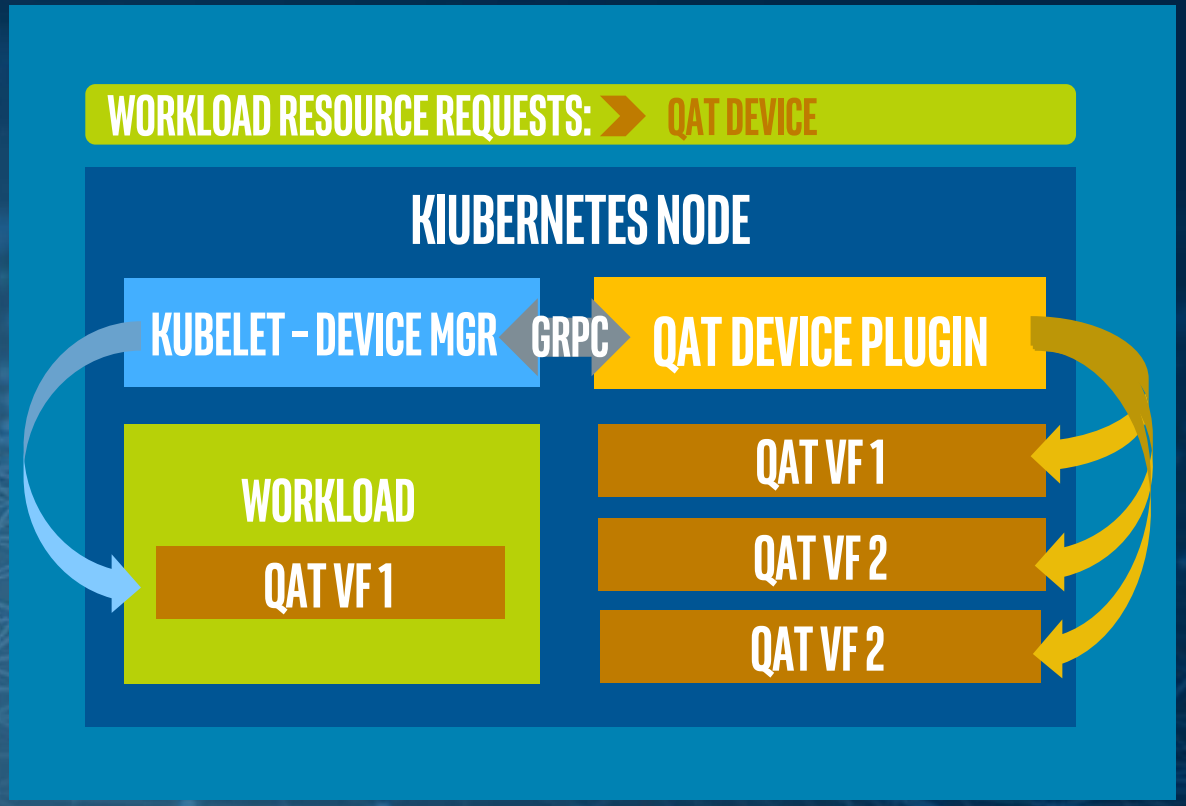
No way to identify QAT devices available in a Kubernetes cluster

Inability for a workload to request a QAT device along with other compute resources

SOLUTION

QAT support enabled through Device plugins

QAT Device Plugin discovers QAT cards 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

https://www.intel.com/content/www/us/en/architecture-and-technology/intel-quick-assist-technology-overview.html

NUMA MANAGER FOR KUBERNETES – NUMA ALIGNMENT OF RESOURCES

PROBLEM

Kubernetes has multiple independent components that handle resource allocation resulting in no alignment on Multi NUMA Node systems

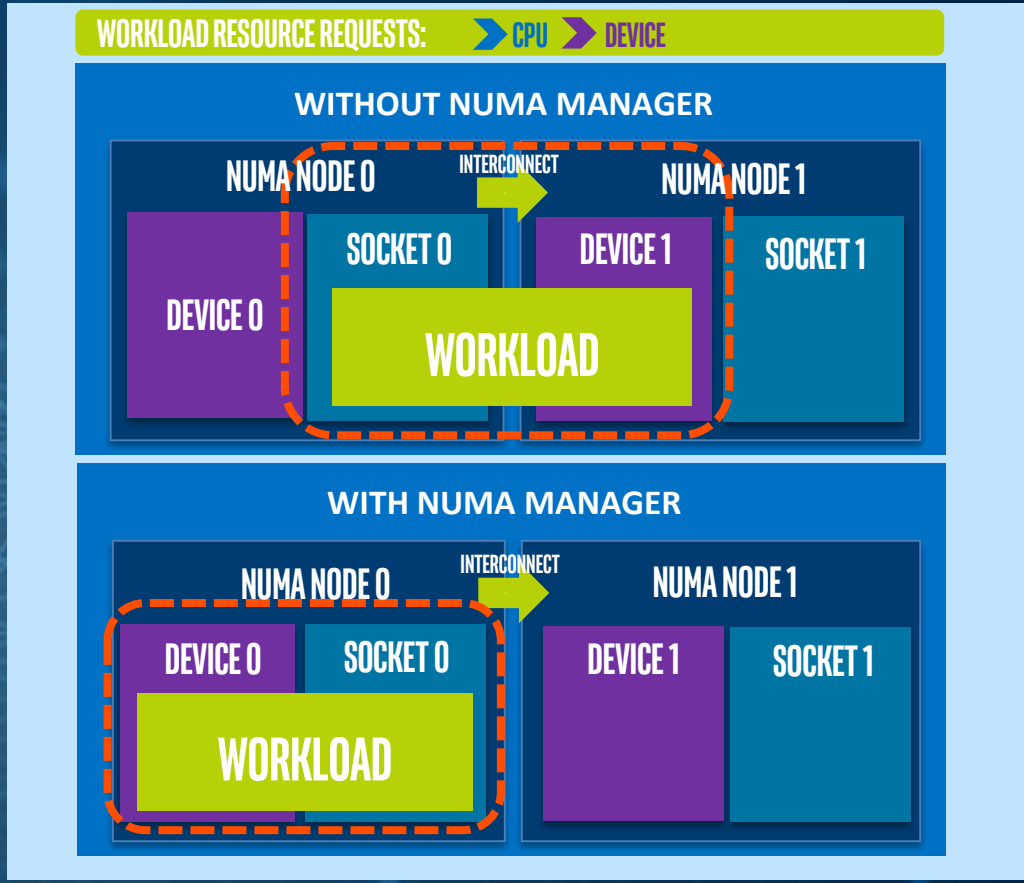
Results in workloads not achieving SLAs or increased resource utilization

SOLUTION

NUMA Manager provides a mechanism to guarantee NUMA Node Affinity of resources requested by a workload

NUMA Manager interfaces with components(eg. CPU Manager & Device Manager) that have NUMA awareness to enable NUMA aligned resource allocations

Gives a performance boost to priority applications as resources are NUMA Node aligned



REFERENCE

https://github.com/kubernetes/community/pull/1680

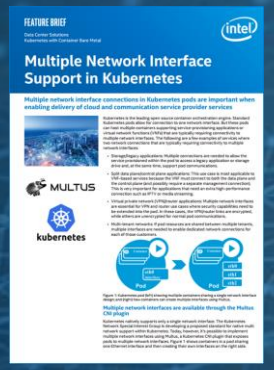
CONTAINER BARE METAL EXPERIENCE KITS

What it is?

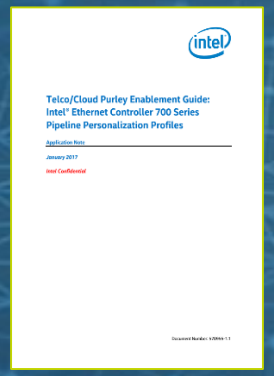
A library of best-practice development guidelines for Container bare metal orchestration

Shortens the time-to-expertise

Addresses challenges in performance, manageability, security and service assurance



FEATURE BRIEFS



FEATURE APPLICATION NOTES

SW SCRIPT

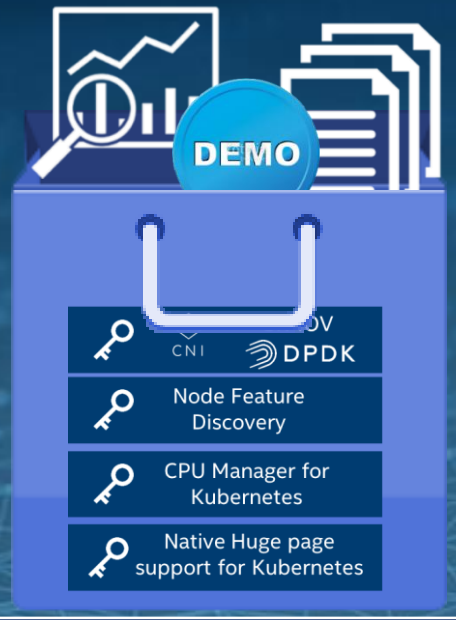
BENCHMARK REPORT

DEMOS

https://networkbuilders.intel.com/network-technologies/container-experience-kits

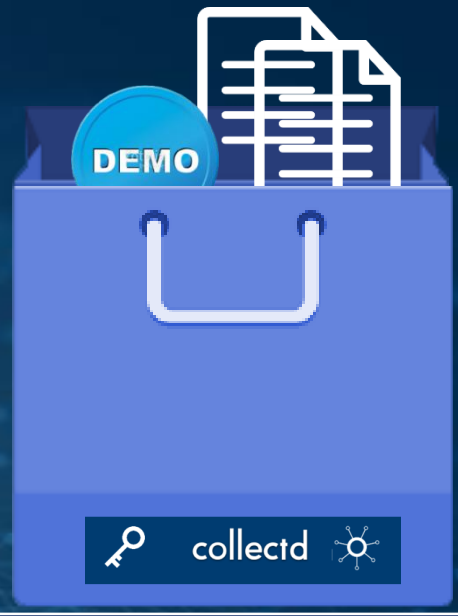
CONTAINER BARE METAL EXPERIENCE KITS

Reference Architecture
Reference Architectures Installation Scripts
Reference Architecture User Guide



Enhance Platform Awareness
Feature Brief
White Paper
Tech. Application Note
Benchmark Test Report
Demo

Kubernetes Networking
Feature Brief
Tech. Application Note
Demo



Platform Telemetry
Application Note
Feature Brief
Demo


DEMO

CALL TO ACTION

CHECKOUT THE CONTAINER BAREMENTAL EXPERIENCE KITS:

HTTPS://NETWORKBUILDERS.INTEL.COM/NETWORK-TECHNOLOGIES/CONTAINER-EXPERIENCE-KITS

USE CASES & FEEDBACK WELCOME ON:

NODE FEATURE DISCOVERY | CPU PINNING | HUGE PAGES | DEVICE PLUGINS | NUMA

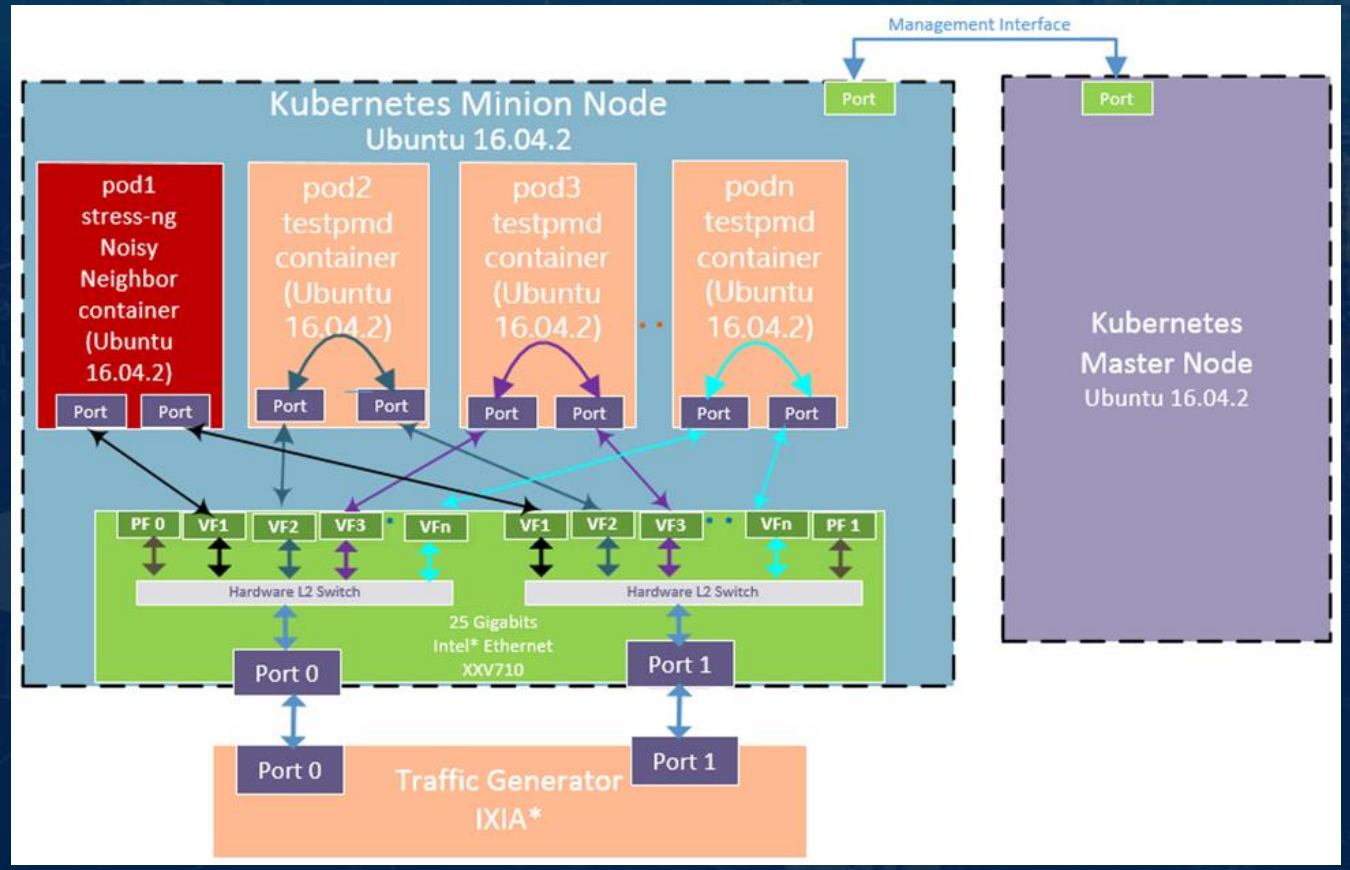
PARTICIPATION IN:

HTTPS://GITHUB.COM/KUBERNETES/COMMUNITY/TREE/MASTER/WG-RESOURCE-MANAGEMENT


intel®
experience
what's inside™

BACKUP

EXAMPLE: CPU MANAGER FOR KUBERNETES BENCHMARK TEST SETUP



Test configuration:

Master & Minion Nodes: {mother board: Intel Corporation; S2600WFQ; CPU: Intel® Xeon® Gold Processor 6138T; 2.0 Ghz; 2 socket; 20 cores; 27.5 MB; 125 W; Memory: Micron MTA36ASF2G72PZ; 1 DIMM/Channel, 6 Channel/Socket; BIOS: Intel Corporation SE5C620.86B.0X.01.0007.060920171037; NIC: Intel Corporation; Ethernet Controller XXV710 for 2x25GbE Firmware version 5.50; SW: Ubuntu 16.04.2 64bit; Kernel 4.4.0-62-generic x86_64; DPDK 17.05}

IXIA* - IxNetwork 8.10.1046.6 EA; Protocols: 8.10.1105.9, IxOS 8.10.1250.8 EA-Patch1

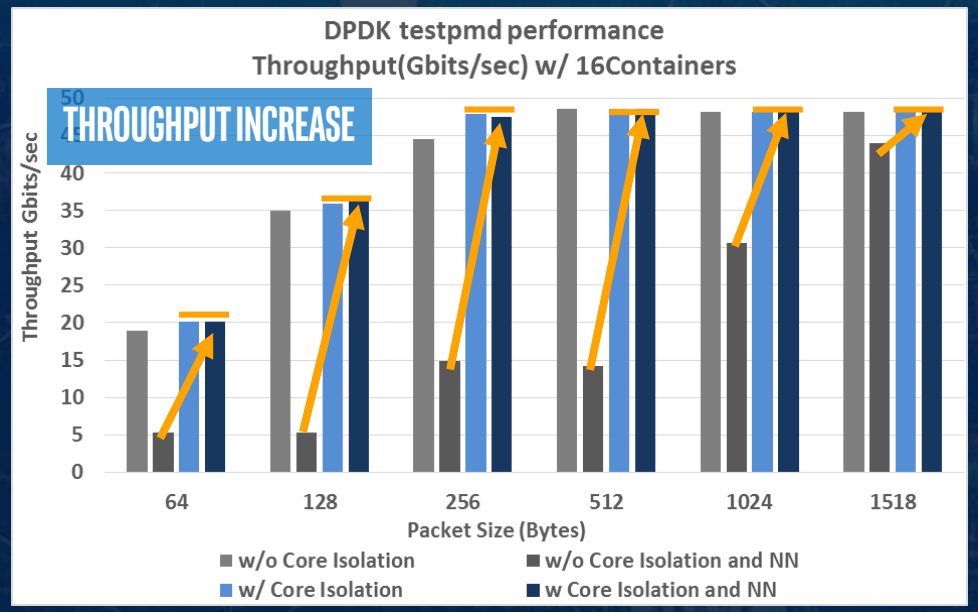
With core isolation	Performance is consistent with or without “noisy” application present
Without core isolation EPA feature, in presence of “noisy application”	• >70% Throughput drops for small packet sizes • > 10% Throughput drops for large packet sizes • > x10 Packet latency increased

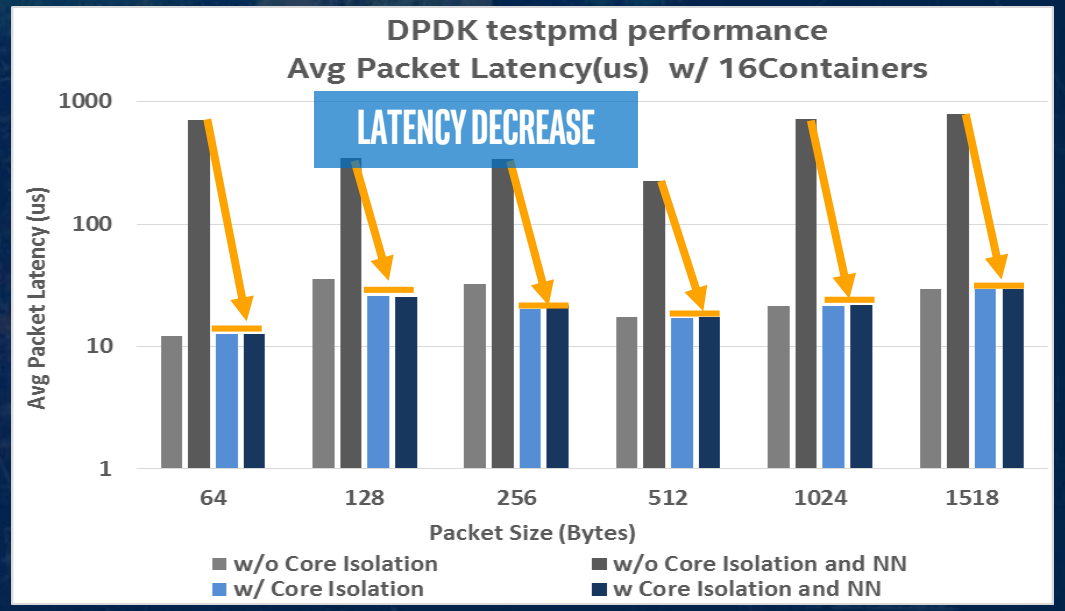
*For more complete information about performance and benchmark results, visit http://www.intel.com/performance. ; **Test configuration: Master & Minion Nodes:** {mother board: Intel Corporation; S2600WFQ; CPU: Intel® Xeon® Gold Processor 6138T; 2.0 Ghz; 2 socket; 20 cores; 27.5 MB; 125 W; Memory: Micron MTA36ASF2G72PZ; 1 DIMM/Channel, 6 Channel/Socket; BIOS: Intel Corporation SE5C620.86B.0X.01.0007.060920171037; NIC: Intel Corporation; Ethernet Controller XXV710 for 2x25GbE Firmware version 5.50; SW: Ubuntu 16.04.2 64bit; Kernel 4.4.0-62-generic x86_64; DPDK 17.05}; **IXIA*** - IxNetwork 8.10.1046.6 EA; Protocols: 8.10.1105.9, IxOS 8.10.1250.8 EA-Patch1

*Benchmark results were obtained prior to implementation of recent software patches and firmware updates intended to address exploits referred to as "Spectre" and "Meltdown". Implementation of these updates may make these results inapplicable to your device or system.

EXAMPLE: CPU MANAGER FOR KUBERNETES BENCHMARK TEST RESULTS

CORE ISOLATION LEADS TO PERFORMANCE CONSISTENCY SOLVING NOISY WORKLOADS PROBLEM





Core Isolation increase throughput of target-workload >200% for small packets in presence of Noisy Workload

Core Isolation decrease latency of target workload up >x13 in presence of Noisy Workload

For more complete information about performance and benchmark results, visit http://www.intel.com/performance. ; **Test configuration: Master & Minion Nodes:** {mother board: Intel Corporation; S2600WFQ; CPU: Intel® Xeon® Gold Processor 6138T; 2.0 Ghz; 2 socket; 20 cores; 27.5 MB; 125 W; Memory: Micron MTA36ASF2G72PZ; 1 DIMM/Channel, 6 Channel/Socket; BIOS Intel Corporation SE5C620.86B.0X.01.0007.060920171037; NIC: Intel Corporation; Ethernet Controller XXV710 for 2x25GbE Firmware version 5.50; SW: Ubuntu 16.04.2 64bit; Kernel 4.4.0-62-generic x86_64; DPDK 17.05}; **IXIA*** - IxNetwork 8.10.1046.6 EA; Protocols: 8.10.1105.9, IxOS 8.10.1250.8 EA-Patch1

*Benchmark results were obtained prior to implementation of recent software patches and firmware updates intended to address exploits referred to as "Spectre" and "Meltdown". Implementation of these updates may make these results inapplicable to your device or system.

Multiple Network Interfaces for VNFs

PROBLEM

Kubernetes support only one Network interface – “eth0”

In NFV use cases, it is required to provide multiple network interfaces to the virtualized operating environment of the VNF

USE CASES

Functional separation of control and data network planes

link aggregation/bonding for redundancy of the network

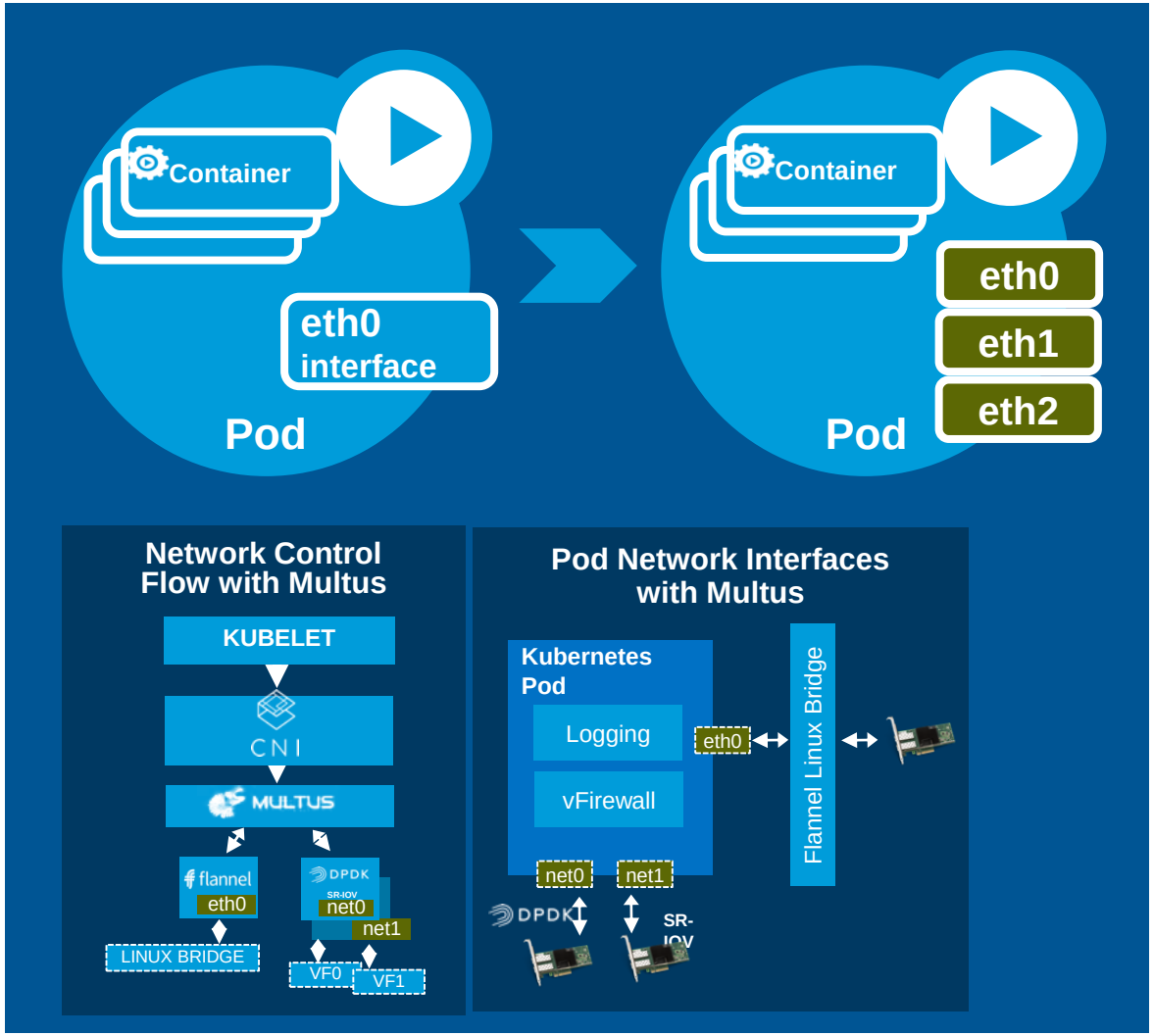
Support for implementation of different network SLAs

Network segregation and Security

REFERENCE

Multus CNI – https://github.com/Intel-Corp/multus-cni

Native Kubernetes - Mailing list with details on discussions :
https://groups.google.com/forum/#!forum/kubernetes-sig-network



Vhost User CNI Plugin

PROBLEM

No Container Networking with software acceleration
for NFV, particularly for East – West Traffic

SOLUTION

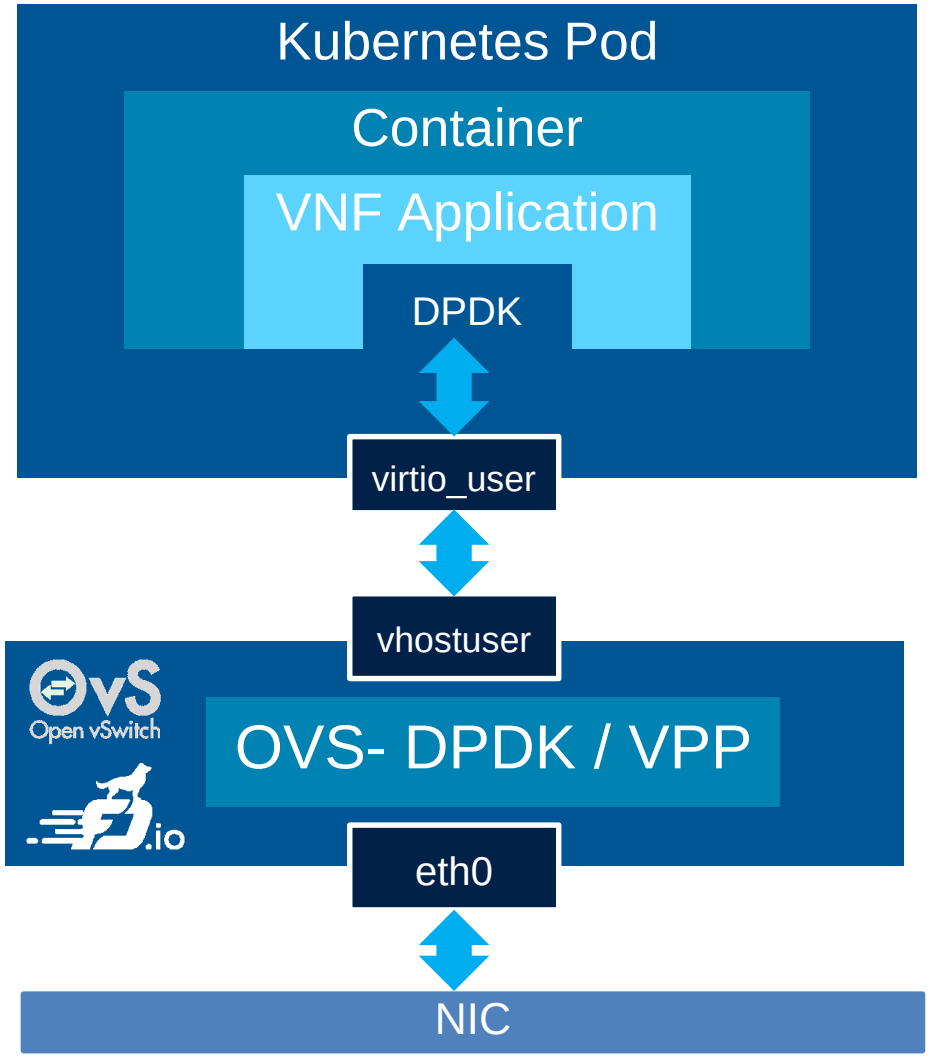
Virtio_user/ vhost_user performance better than VETH pairs

Supports VPP as well as DPDK OVS

Vhost_user CNI plugin enables K8s
to leverage data plane acceleration

REFERENCE

https://github.com/intel/vhost-user-net-plugin (V1.0 Sep ’17)



DPDK – SRIOV CNI Plugin

PROBLEM

Lack of support for physical platform resource isolation

No guaranteed network IO performance

No support for Data Plane Networking

SOLUTION

Allows SRIOV support in Kubernetes via a CNI plugin

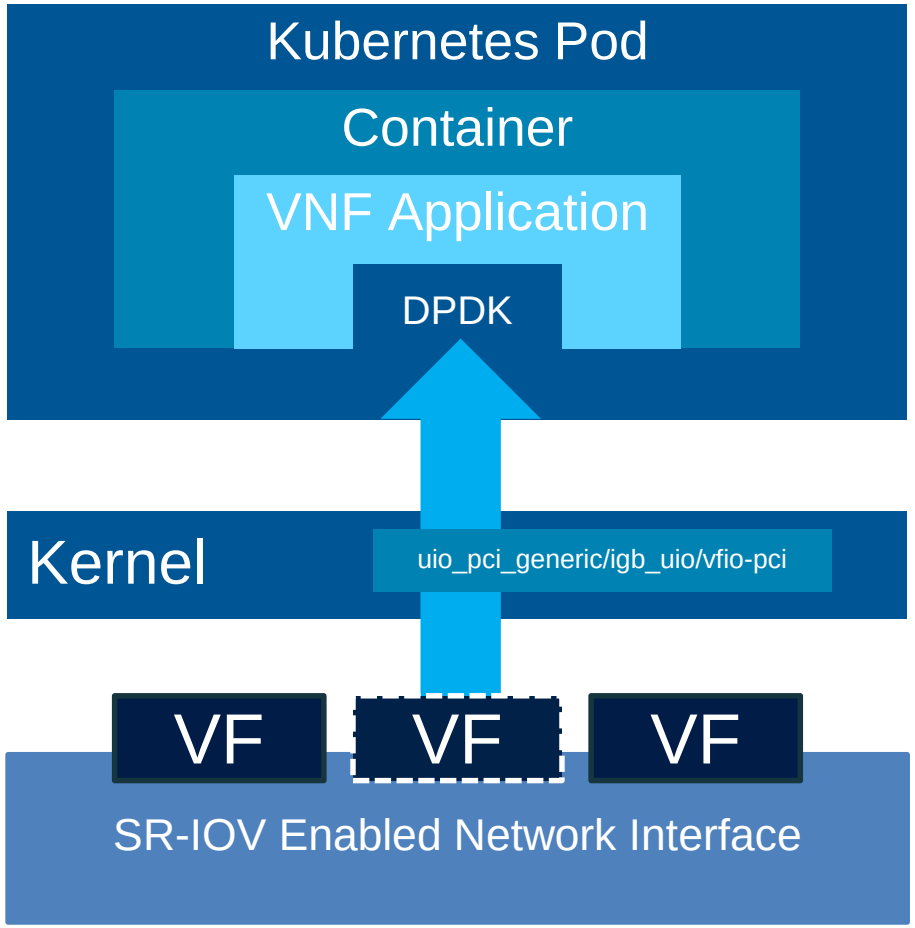
Supports two modes of operation:

SR-IOV: SR-IOV VFs are allocated to pod network namespace

DPDK: SR-IOV VFs are bounded to DPDK drivers in the userspace

REFERENCE

github.com/Intel-Corp/sriov-cni



Bonding CNI Plugin

PROBLEM

There is no redundancy of network link failure in container environment. This results in high-priority workloads not achieving expected high-availability. e.g., due to failure of NIC, network Switch or cable breakdowns.

SOLUTION

Bonding CNI provides a mechanism to aggregate multiple network interfaces into a single logical “bonded” interface in a Container environment. Thus providing a fail-over, high-availability network for containerized applications e.g., VNF.

REFERENCE

https://github.com/Intel-Corp/bond-cni