KubeCon | CloudNativeCon

Europe 2020

Virtual

Optimized Resource Allocation in Kubernetes? Topology Manager is Here

Conor Nolan, Intel & Victor Pickard, Red Hat

Summary

- Introduction and Motivation with Use Cases
- CPU Manager and Device Manager
- Topology Manager Overview
- Performance Results
- What’s next
- Contributing

The Need for NUMA Awareness

- Workloads in areas such as Telco 5G, scientific computing, machine learning, AI, financial services and data analytics often have NUMA alignment as a requirement
- DPDK based network applications may require dedicated CPUs, huge page memory, and SR-IOV VFs on the same NUMA node for optimal, low-latency execution.

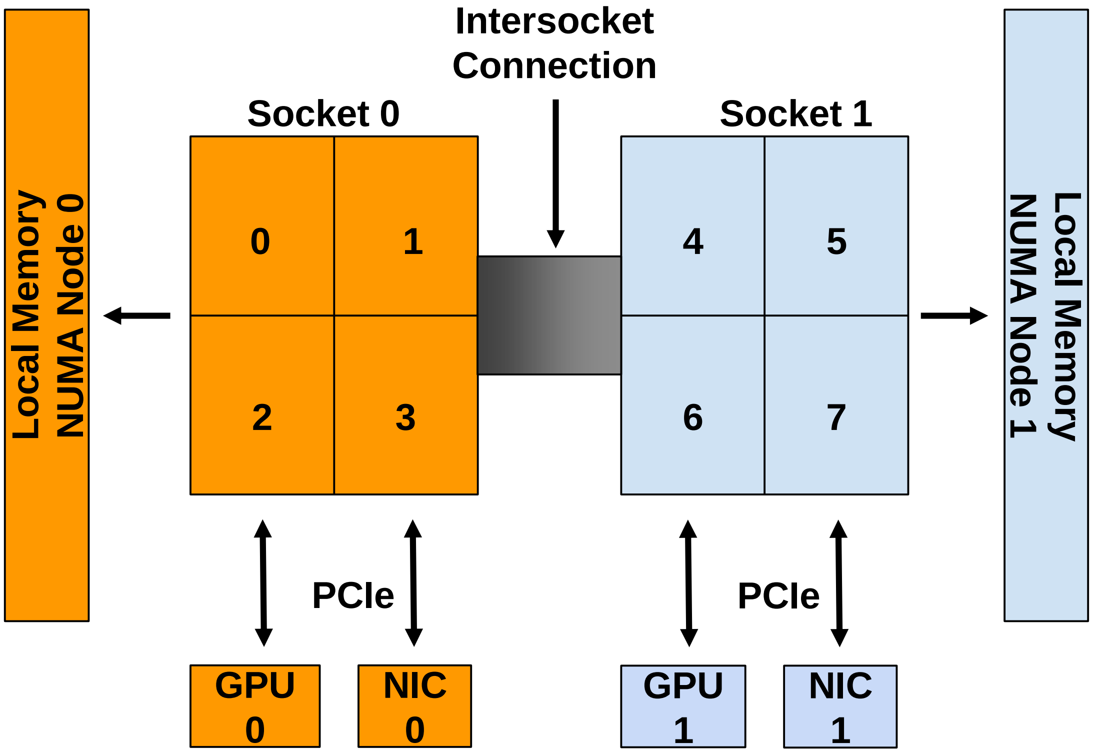
A Broader Context

- An increasing number of systems desire a combination of CPUs and hardware accelerators to support performance sensitive applications that desire low-latency and high-throughput
- Hardware resource allocations, such as CPUs and Devices (SR-IOV, GPUs), need to be coordinated to achieve optimal performance

What is NUMA

NUMA = Non-Uniform Memory Access

- On multi-CPU systems, all memory is visible and accessible from any CPU
- Local memory access is fastest
- Non-local memory access time is variable, depending on number of interconnects
- Peripheral devices also affected by local and non-local access
- For optimal performance, CPUs and devices should be on the same NUMA node



CPU Manager

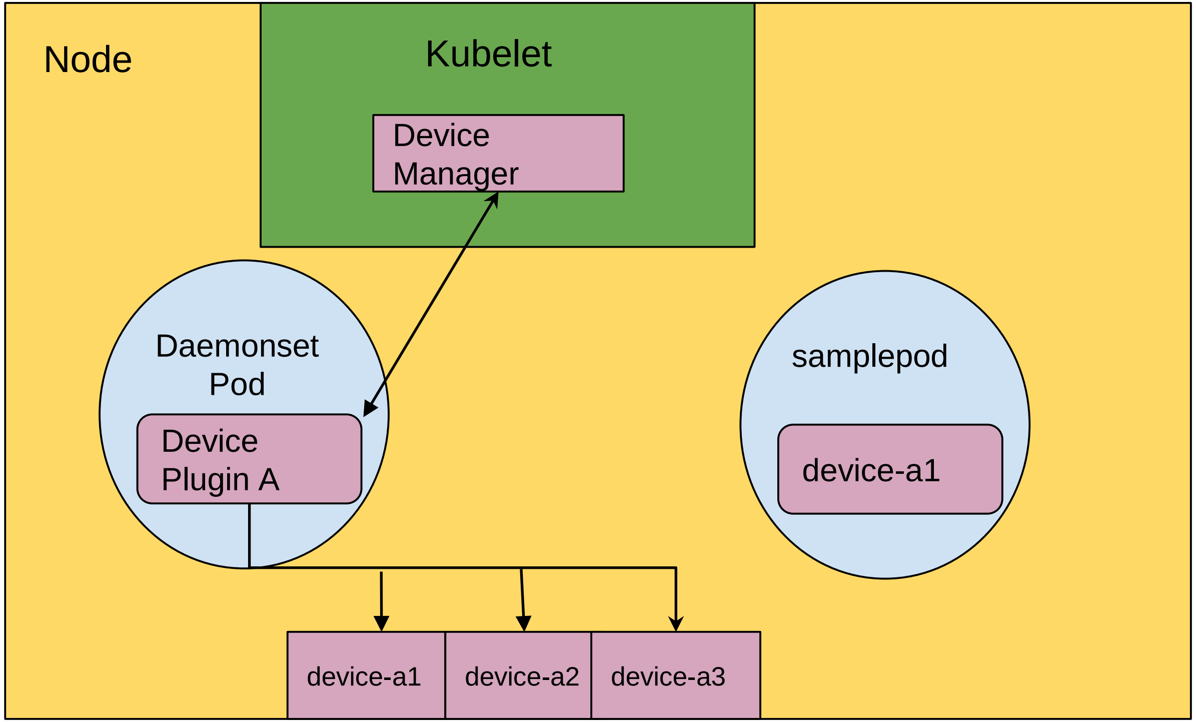
- CPU Manager - allocates exclusive CPUs to containers in Guaranteed Pods
- “Static” CPU Manager policy manages a shared pool of CPUs
- A container in a Guaranteed Pod with integer CPU request(s) is allocated CPUs that are assigned exclusively to the container

```
apiVersion: v1
Kind: Pod
spec:
  containers:
  - name: guaranteed-container
    image: nginx
    resources:
      requests:
        cpu: 2
        memory: 200Mi
        gpu-vendor.com/gpu: 1
        nic-vendor.com/nic: 1
      limits:
        cpu: 2
        memory: 200Mi
        gpu-vendor.com/gpu: 1
        nic-vendor.com/nic: 1
```

https://kubernetes.io/blog/2018/07/24/feature-highlight-cpu-manager/

Device Plugins

- Advertise system hardware resources to Device Manager in the Kubelet
- Enables vendor specific initialization and setup
- API for Device Plugins to communicate with Device Manager

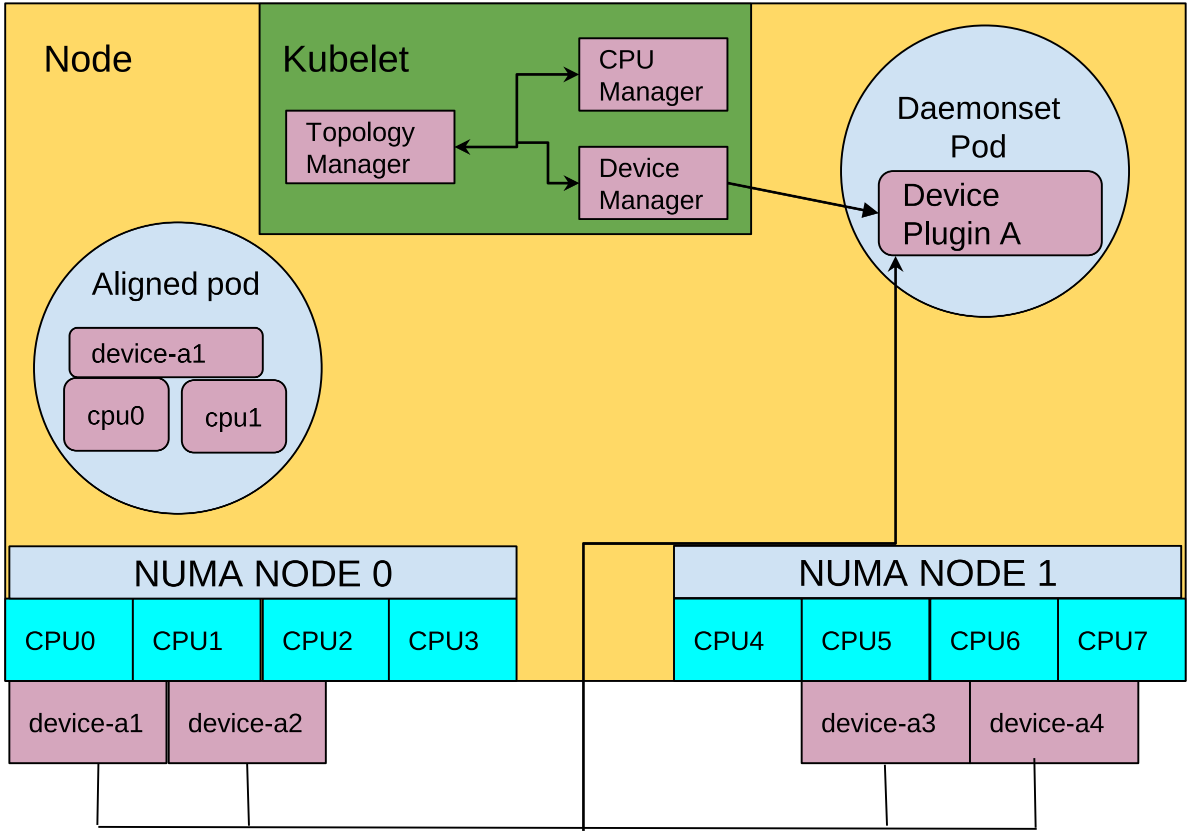


Introducing Topology Manager

- Beta as of Kubernetes 1.18
- CPU and Device Manager assign resources independently, which could result in sub-optimal allocation
- Topology Manager provides an interface to coordinate resource assignment on a Node level
- CPU and Device Manager implement the Topology Manager interface
- Ability to assign resources to Pod/Container from the same NUMA node
  - CPUs
  - SR-IOV VFs
  - GPUs

Topology Manager Continued

```
apiVersion: v1
Kind: Pod
spec:
  containers:
  - name: aligned-pod
    image: nginx
    resources:
      requests:
        cpu: 2
        memory: 200Mi
        vendor/device-a: 1
      limits:
        cpu: 2
        memory: 200Mi
        vendor/device-a: 1
```

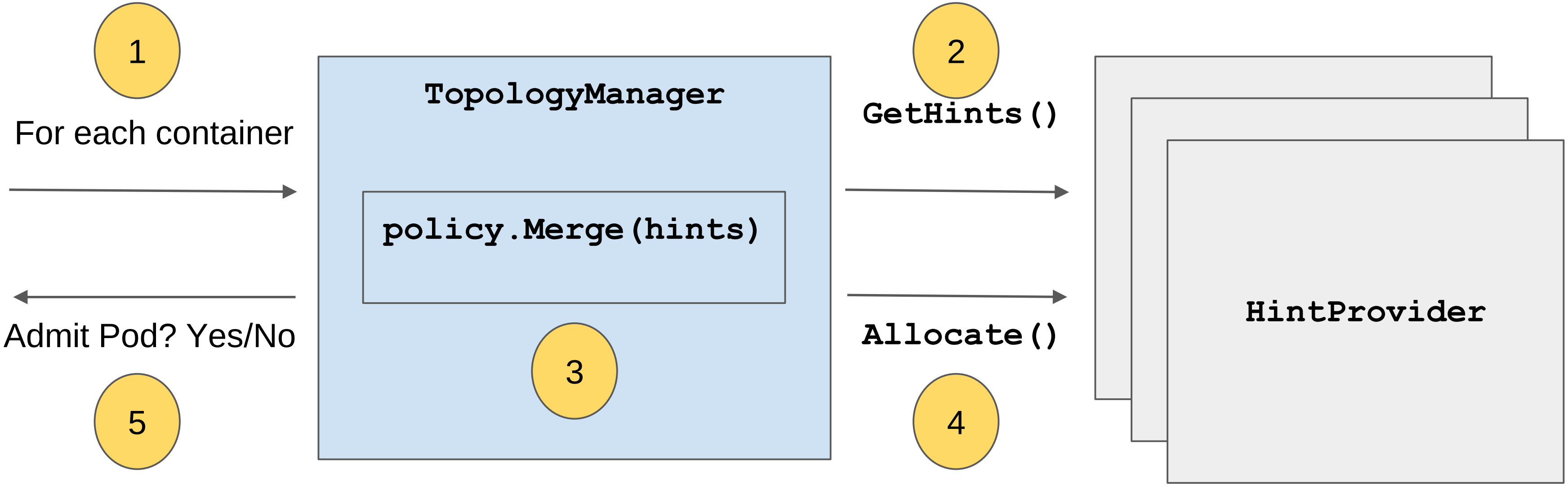


Topology Manager Policies

Node Level Policies:

- **`none`**: Default policy that does not perform any topology alignment
- **`best-effort`**: Attempts to align resources optimally on NUMA nodes
- **`restricted`**: Attempts to align resources optimally on NUMA nodes or pod admission fails
- **`single-numa-node`**: Attempts to align resources on a single NUMA node or pod admission fails

So How Does it Work?



1

For each container

TopologyManager

policy.Merge(hints)

3

2

GetHints()

Allocate()

4

HintProvider

Admit Pod? Yes/No

5

Topology Hints

A **`TopologyHint`** encodes a set of constraints from which a given resource request can be satisfied. At present, the only constraint we consider is NUMA alignment. It is defined as follows:

```
type TopologyHint struct {
    NUMANodeAffinity bitmask.BitMask
    Preferred bool
}
```

- The **`NUMANodeAffinity`** field contains a bitmask of NUMA nodes where a resource request can be satisfied.
- The **`Preferred`** field contains a boolean that encodes whether the given hint is "preferred" or not.

Example 1

Topology Hints for each resource:

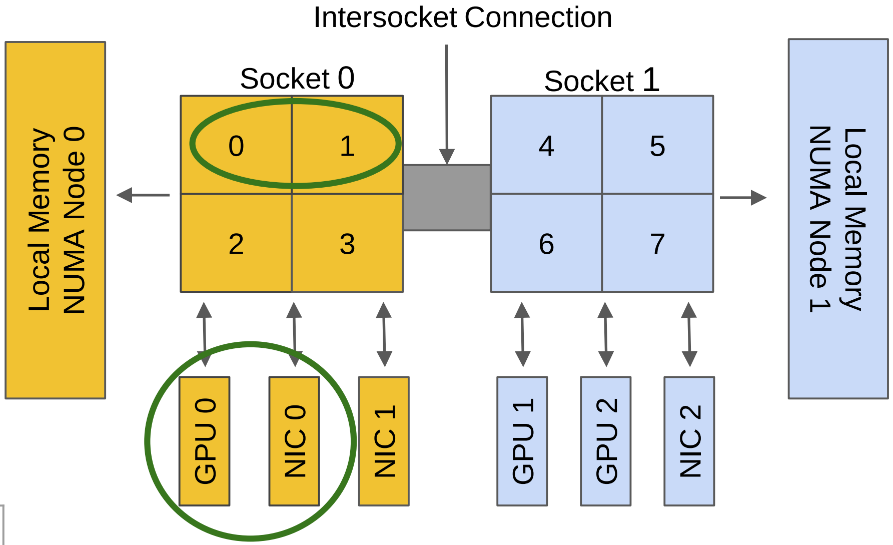
- CPU:
  `[{01 true} {10 true} {11 false}]`
- NIC:
  `[{01 true} {10 true} {11 false}]`
- GPU:
  `[{01 true} {10 true} {11 false}]`

Merged Hints:

Policy {01 true} {00 false}	Best Hint	Admit? Yes/No

{00 false}
Best Effort
{01 false}
	·{0·1·true}	Yes

Restricted {00 false}	{01 true}	Yes
{00 false} Single NUMA Node	{01 true}	Yes

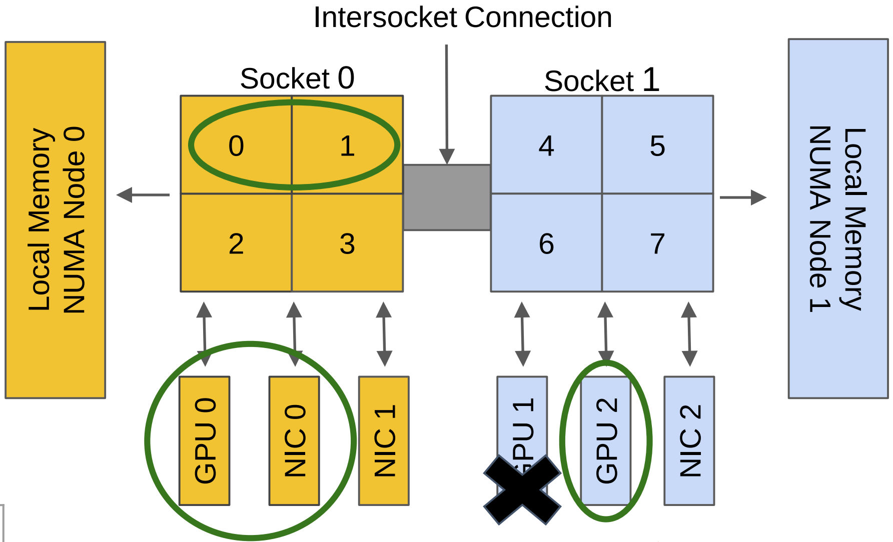


```
Kind: Pod
spec:
  containers:
    request:
      memory: 64Mi
      cpu: 2
      nic-vendor.com/nic: 1
      gpu-vendor.com/gpu: 1
```

Example 2

Topology Hints for each resource:

- CPU:
  `[{01 true} {10 true} {11 false}]`
- NIC:
  `[{01 true} {10 true} {11 false}]`
- GPU:
  `[{11 false}]`



Policy	Best Hint	Admit? Yes/No
Best Effort	{01 false}	Yes
Restricted	{01 false}	No
Single NUMA Node	{11 false}	No

```
Kind: Pod
spec:
  containers:
    request:
      memory: 64Mi
      cpu: 2
      nic-vendor.com/nic: 1
      gpu-vendor.com/gpu: 2
```

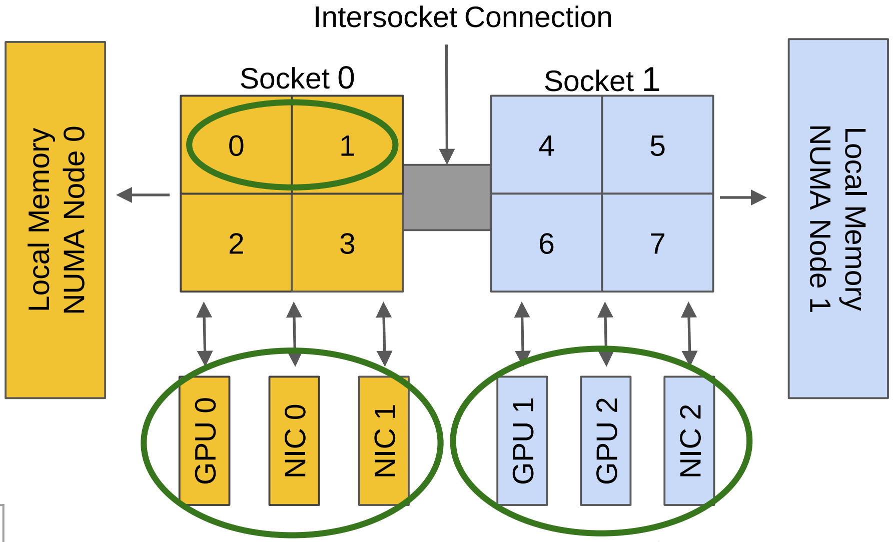
Topology Affinity Error

Example 3

Topology Hints for each resource:

- CPU:
  `[{01 true} {10 true} {11 false}]`
- NIC:
  `[{11 true}]`
- GPU:
  `[{11 true}]`

Policy	Best Hint	Admit? Yes/No
Best Effort	{01 true}	Yes
Restricted	{01 true}	Yes
Single NUMA Node	{11 false}	No



```
Kind: Pod
spec:
  containers:
    request:
      memory: 64Mi
      cpu: 2
      nic-vendor.com/nic: 3
      gpu-vendor.com/gpu: 3
```

Topology Affinity Error

Performance Improvement

Packet Size (B)	DPDK Throughput Without NUMA Alignment (GBPS)	DPDK Throughput With NUMA Alignment (GBPS)	**Performance Improvement**
64	27.97	58.81	**2.1x**
128	48.46	102.36	**2.1x**
256	86.59	190.60	**2.2x**
512	161.58	198.09	**1.2x**
1024	199.99	200.00	**0**

Reference:
https://builders.intel.com/docs/networkbuilders/topology-management-implementation-in-kubernetes-technology-guide.pdf

Future Enhancements

- ~~Support Device Specific Constraints~~
  - KEP: https://github.com/kubernetes/enhancements/pull/1121
- Support Pod Level Resource Alignment
  - KEP: https://github.com/kubernetes/enhancements/pull/1752
- NUMA Alignment for Hugepages
  - KEP: https://github.com/kubernetes/enhancements/pull/1203
- Topology Aware Scheduling
  - KEP: https://github.com/kubernetes/enhancements/pull/1870
  - KEP: https://github.com/kubernetes/enhancements/pull/1858
- Per-Pod Alignment Policy

Getting Involved

- Find out more about Topology Manager
  - Control Topology Manager Policies on a Node
  - Kubernetes Topology Manager Moves to Beta - Align Up!

- Interested in learning more or contributing? Join SIG-Node meetings
  - Every Tuesday at 10:00 PT https://zoom.us/j/4799874685

KubeCon | CloudNativeCon

Europe 2020

Virtual

KEEP CLOUD NATIVE CONNECTED