KubeCon | CloudNativeCon

Europe 2022

WELCOME TO VALENCIA

KubeCon CloudNativeCon Europe 2022

This is The Way!

A Crash Course on the Intricacies of Managing CPUs in K8s

Swati Sehgal
Principal Software Engineer
Red Hat

&

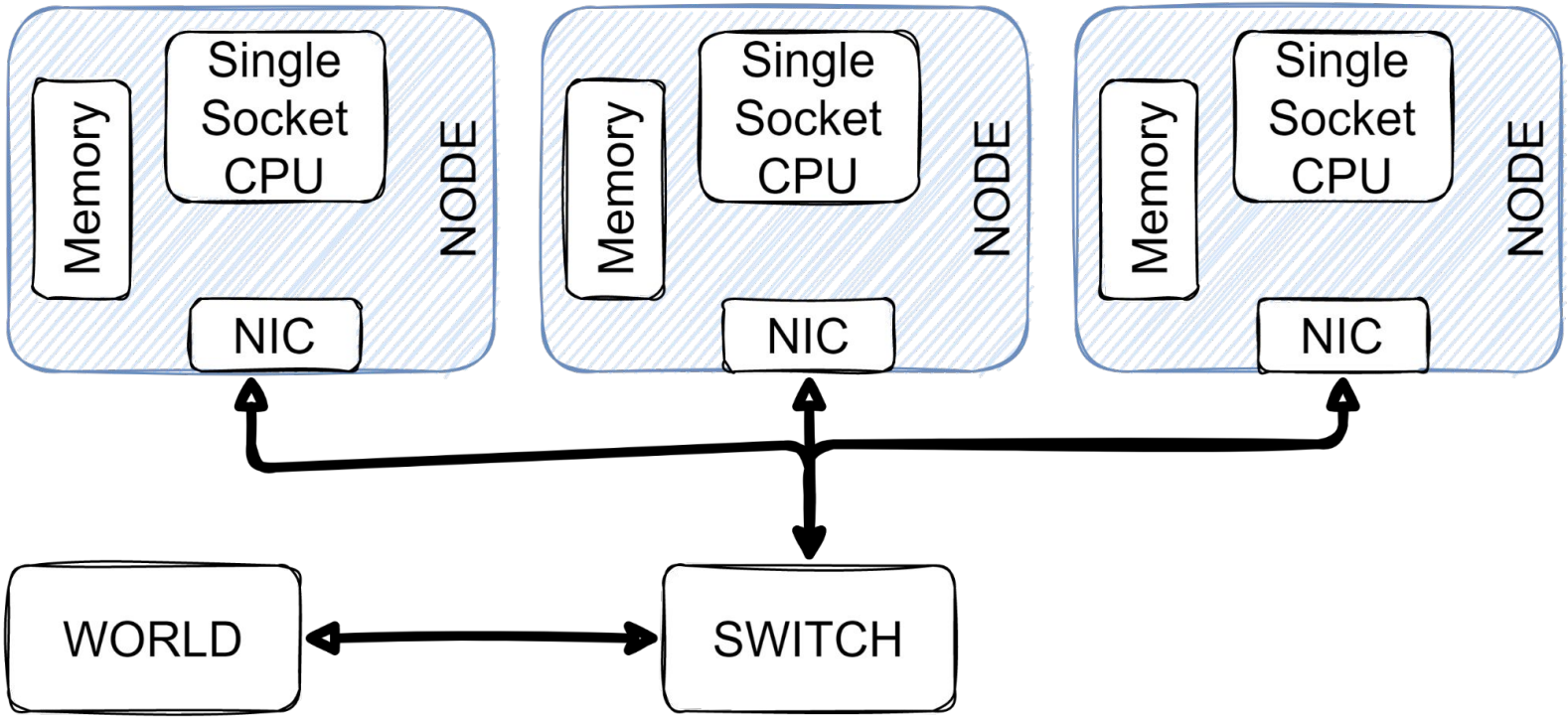
Marlow Weston
Cloud Native Architect
Intel

Scope:

We will cover CPU Management requirements Only, but also reference other projects.

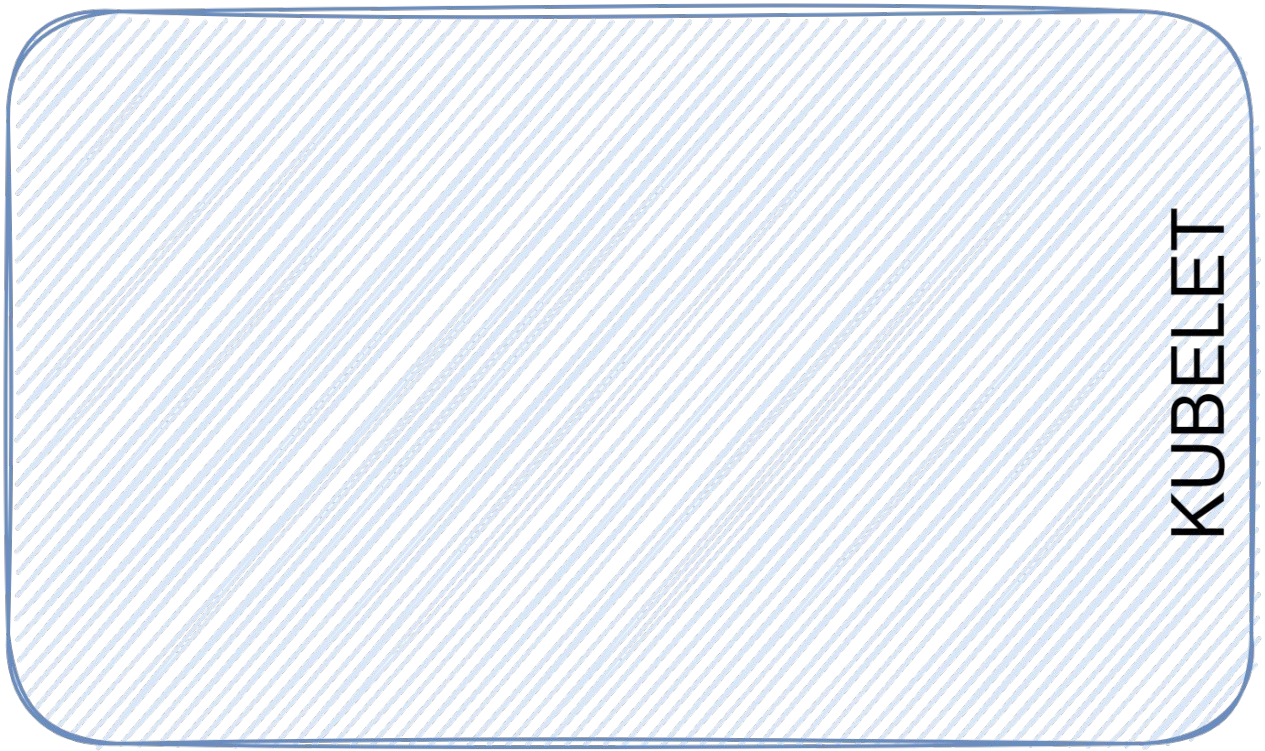
In the Beginning, Systems were Simple

Simple Systems



Kubelet was designed for Simple… at first

Early Kubelet



Resource Management in Kubelet

Early days(< Kubernetes v1.8- before 2017)


```yaml
apiVersion: v1
kind: Pod
metadata:
  name: frontend
spec:
  containers:
  - name: app
    image:
my-company.example/myapp
    resources:
      requests:
        memory: "64Mi"
        cpu: "250m"
      limits:
        memory: "128Mi"
        cpu: "500m"
```

Resources supported:

- CPU
- Memory

Requests: Ask for resources for your container

Limits: Limit the amount of resources consumed by the container

Resource Management in Kubelet

Kubernetes v1.8-v1.11 (2017-2018)

Resource Management Working Group in 2017

Pre-allocated hugepage support as a native resources:

Alpha support v1.8 (graduated to Beta in v1.11)

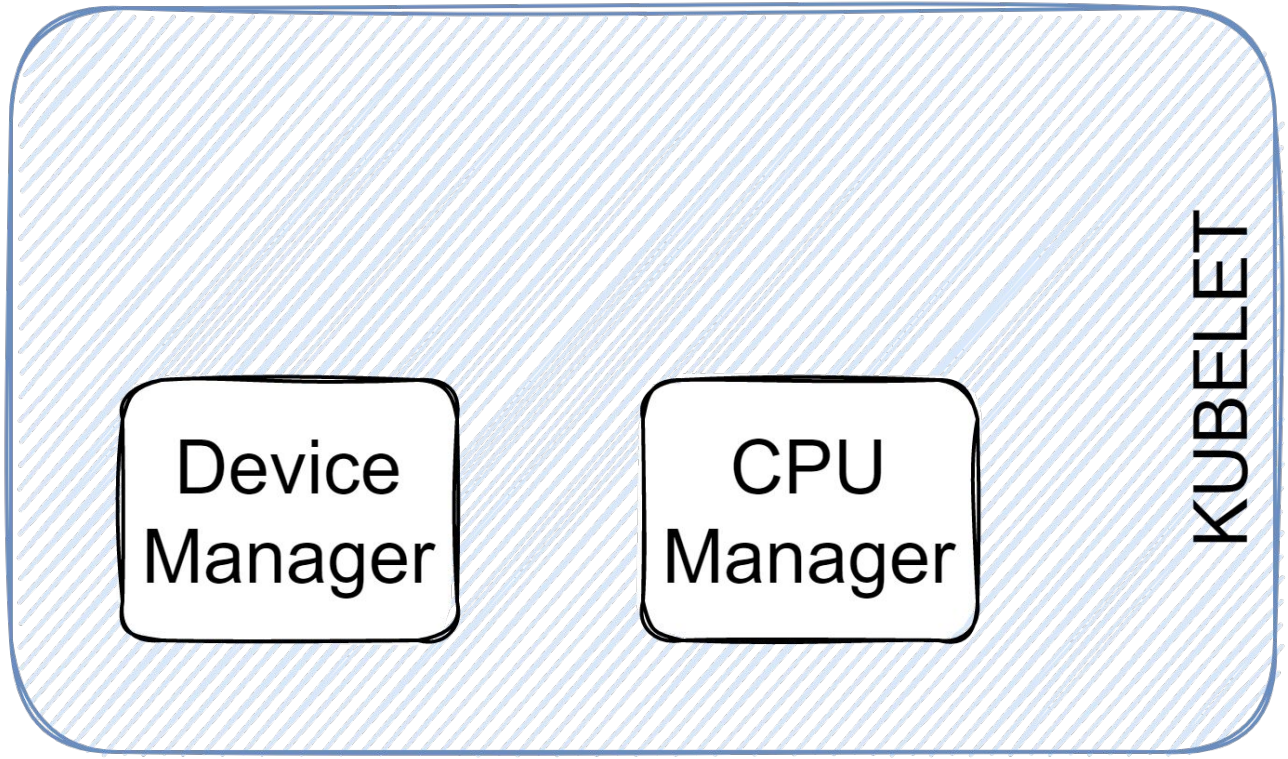
CPU Manager support to enable container level CPU affinity support

Alpha support v1.8 (graduated to Beta in v1.11)

Device Plugin Support to enable a consistent and portable solution for users to consume hardware devices across k8s clusters

Alpha support in v1.8 (Graduate to Beta in v1.10)

Early Kubelet

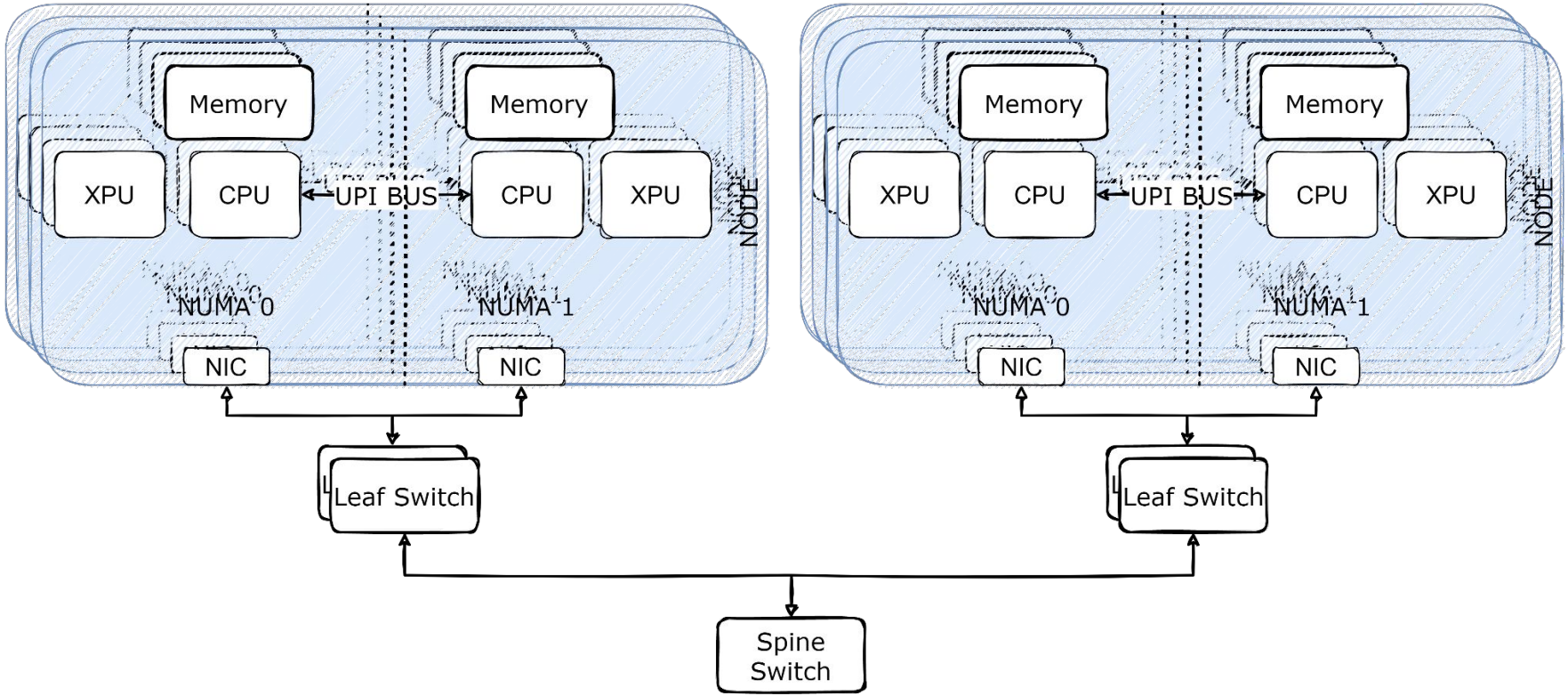


“If you give a mouse a cookie, he’s going to ask for a glass of milk.”
–Laura Numeroff

High Performance Use Cases

- Performance Sensitive Workloads

High Performance, AI/ML Clusters



CPU Manager - Pinned Cores

CPU Manager with static policy allocates CPUs exclusively for a container if

- pod QoS is Guaranteed
- has a positive integer CPU request
- does not change CPU assignments for exclusively pinned guaranteed containers after the main container process starts

CPU Manager Policies

--cpu-manager-policy kubelet flag used to specify the policy

None

- Default
- Provides no affinity beyond what the OS scheduler does automatically
- Can handle partial CPUs

Static

- allows containers access to exclusive CPUs on the node
- does not change CPU assignments for exclusively pinned guaranteed containers after the main container process starts
- Only uses whole CPUs, so increases perceived CPU utilization
- Only by container, not by pod

CPU Manager Policy Options (Introduced in v1.22, Beta in v1.23)

--cpu-manager-policy-options kubelet flag used to specify the policy option

full-pcpus-only	• Beta option, visible by default • the static policy will always allocate full physical cores, so guarantee same NUMA zone. • Fails with SMTAlignmentError for partial core allocation.
distribute-cpus-across-numa	• alpha, hidden by default • the static policy will evenly distribute CPUs across NUMA nodes • Still per container

Exclusive CPU Assignment to a pod

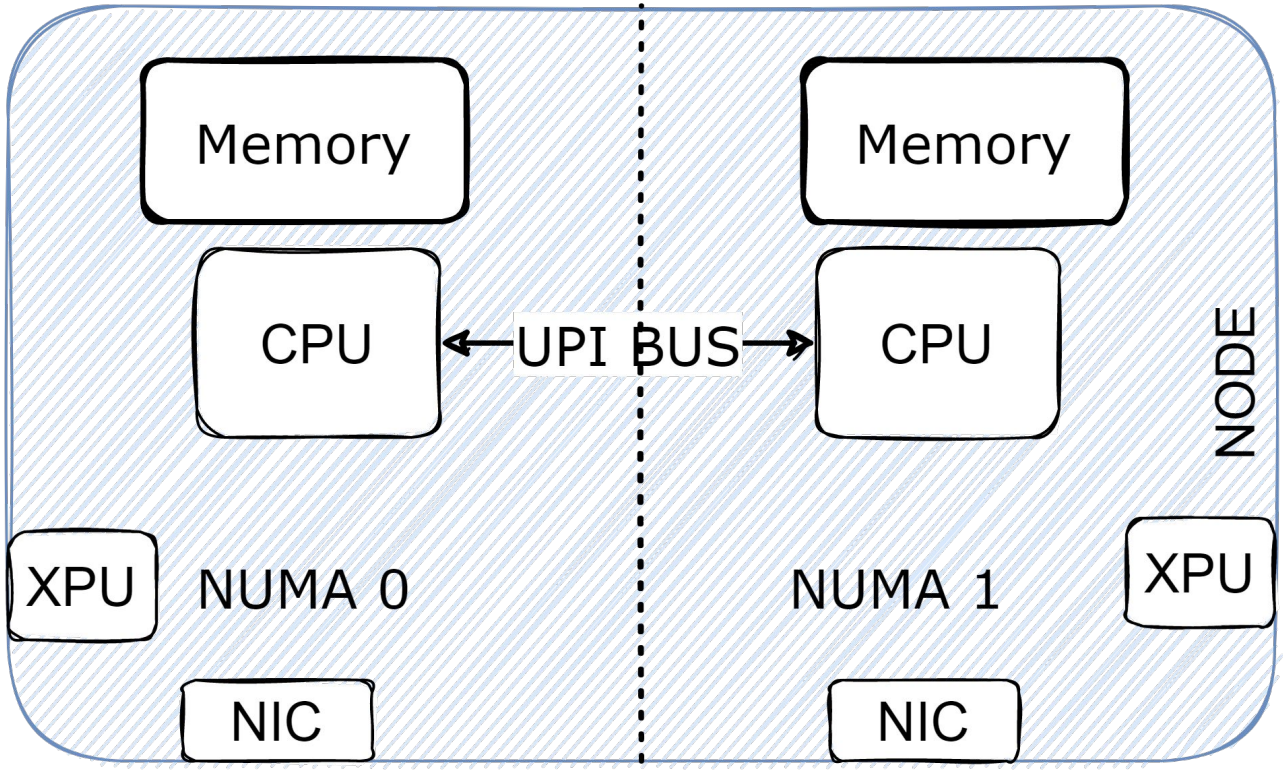

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: frontend
spec:
  containers:
  - name: app
    image: my-company.example/myapp
    resources:
     requests:
       memory: "128Mi"
       cpu: "5000m"
     limits:
       memory: "128Mi"
       cpu: "5000m"
```

If kubelet is configured with **--cpu-manager-policy=static**, this pod is allocated exclusive CPU!

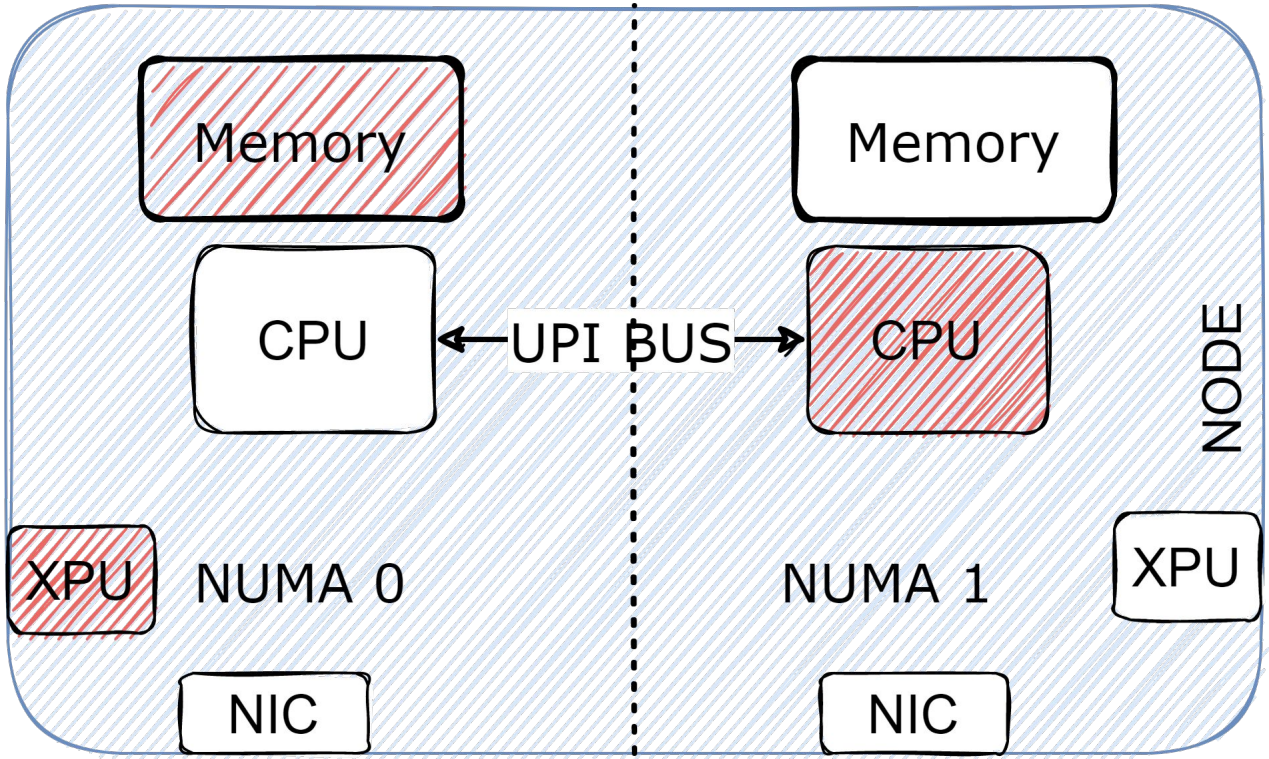
If kubelet is configured with **--cpu-manager-policy-options=full-pcpus-only** this pod will fail with SMTAlignmentError

(assuming system has 2 vCPUs per pCPU)

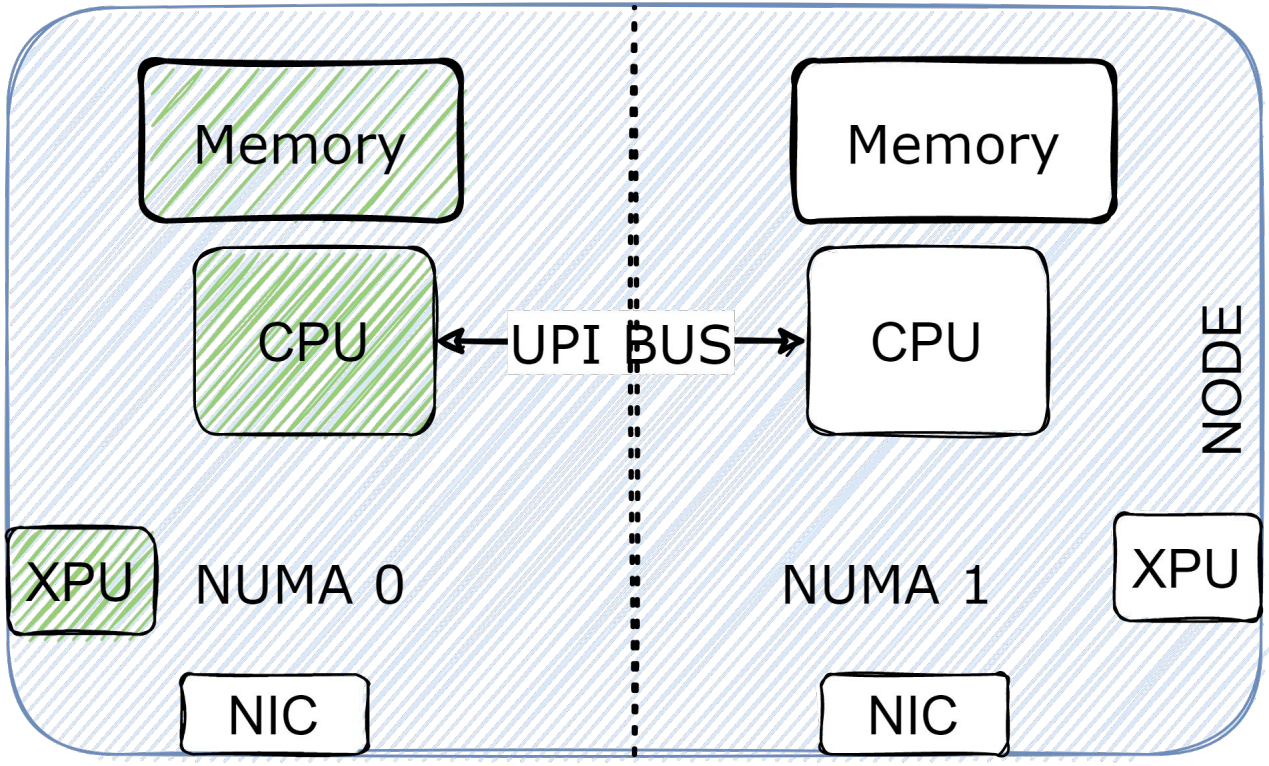
NUMA Zones: Not for the weak of heart



NUMA Zones: Not for the weak of heart

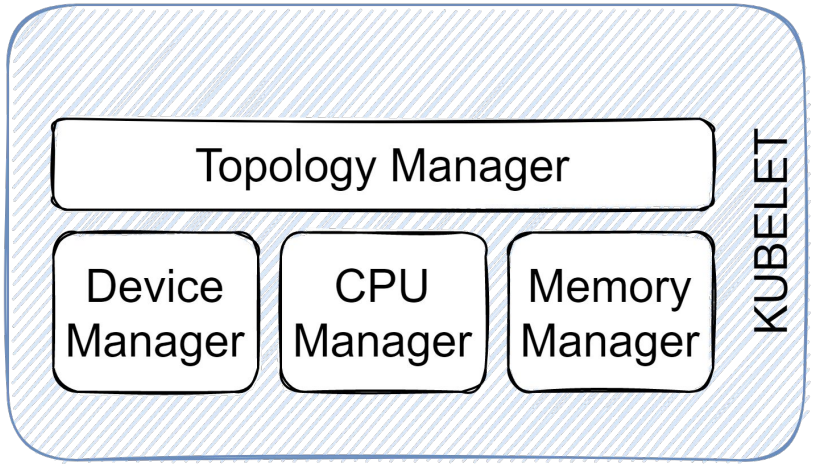


NUMA Zones: Not for the weak of heart



Along Comes Topology Management…

>Kubernetes v1.8 (2019 onwards)



Topology Manager to coordinate resource assignment to avoid cross NUMA assignments

Alpha support V1.16 (graduated to Beta in v1.18)

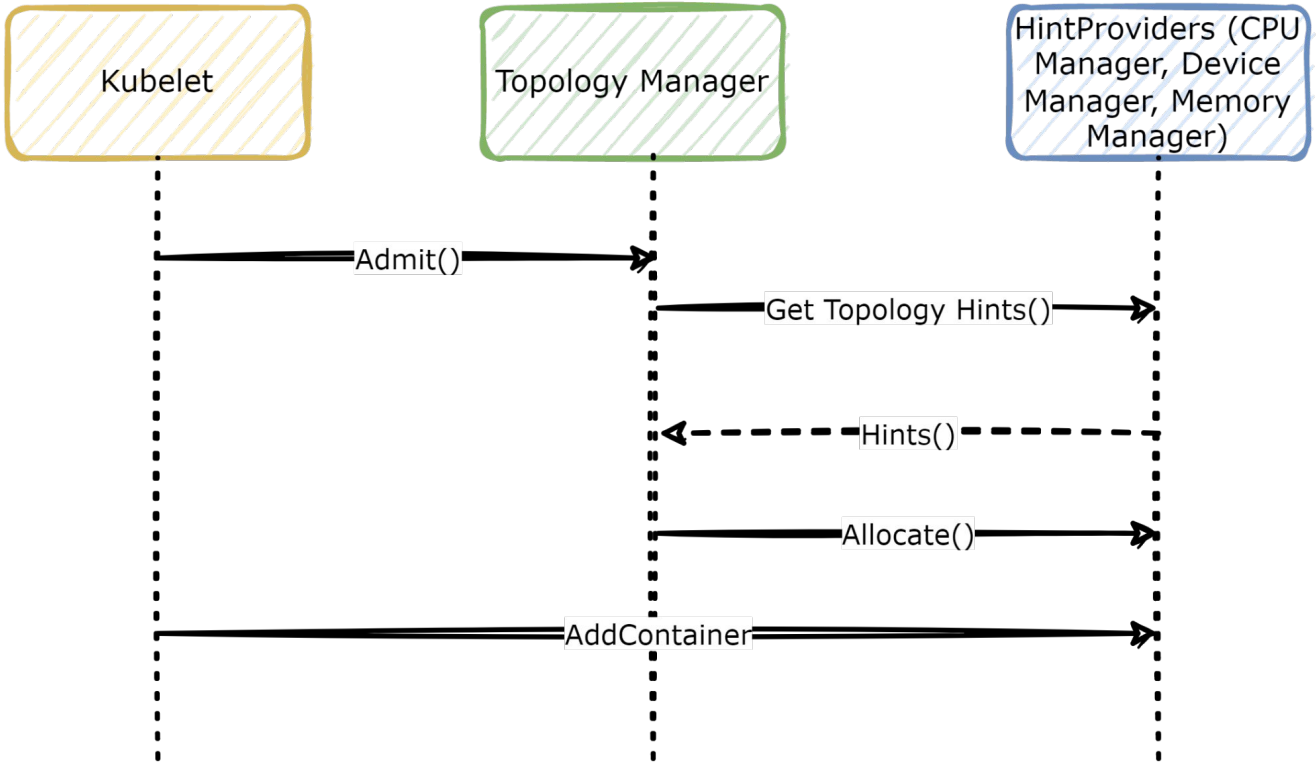
Memory Manager for guaranteed memory (and hugepages) allocation to pods

Alpha support V1.21 (graduated to Beta in v1.22)

Known Issue: Scheduler is not NUMA aware and pod can fail with **TopologyAffinityError** if kubelet is unable to align all the resources based on the Topology Manager policy.

Going with the Topology Flow

>Kubernetes v1.8



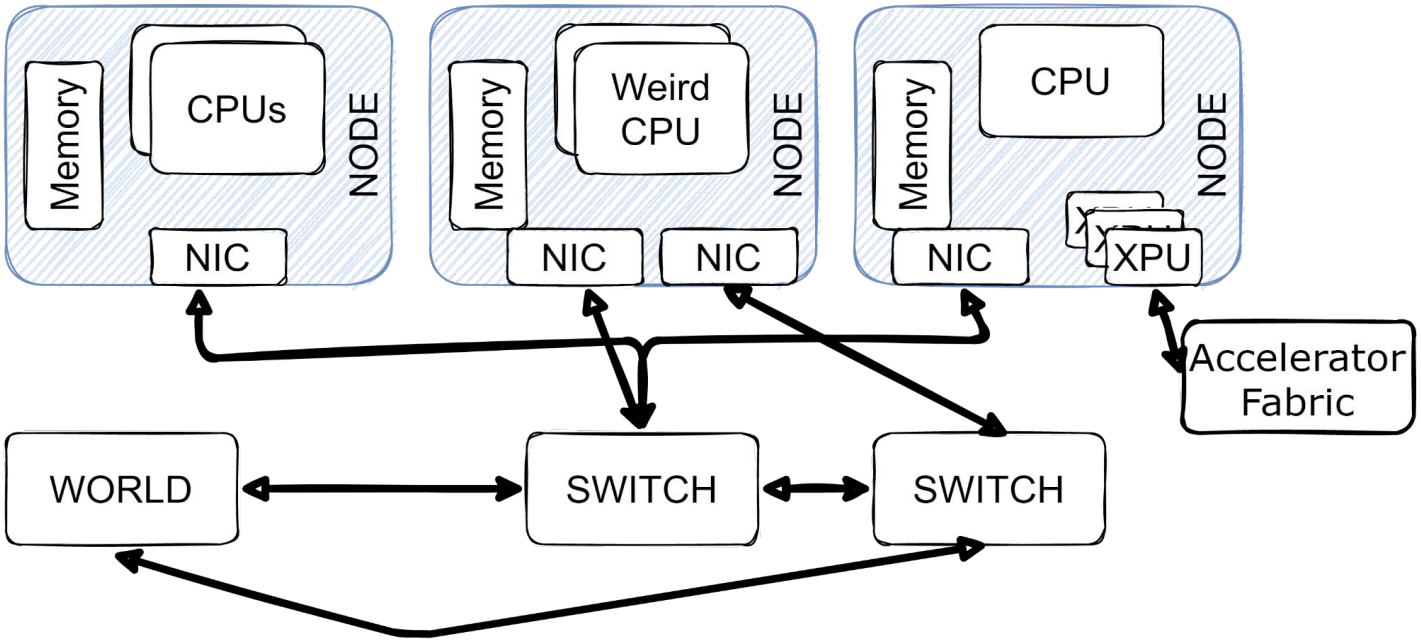
Current Gaps for High Performance Compute

- Affinity node-level only. CANNOT choose affinity of resources.
- Limits with larger numbers of NUMA zones
- Cores cannot be pinned by pod
- Can’t mix pinned with shared cores
- Cannot schedule one pod or container per NUMA zone

Heterogeneous Clusters

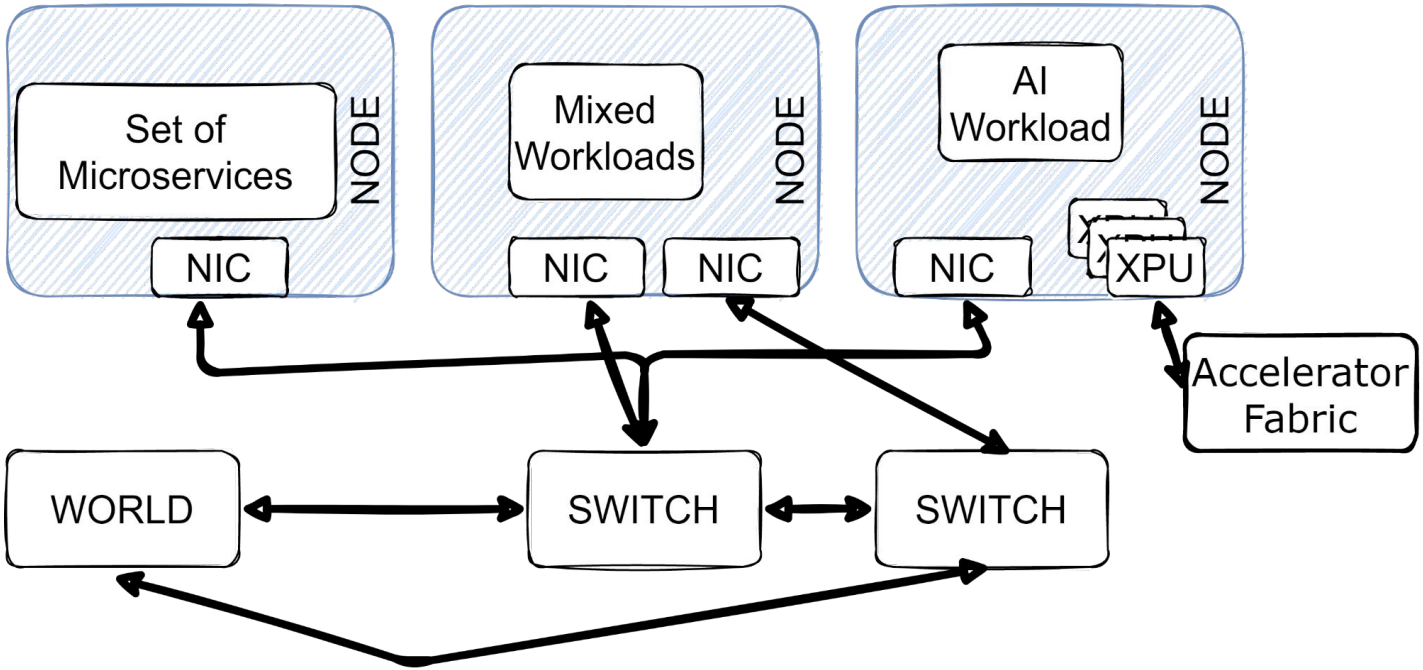
- Fun for the whole family

Hardware View



NOW WHAT???

Workload View

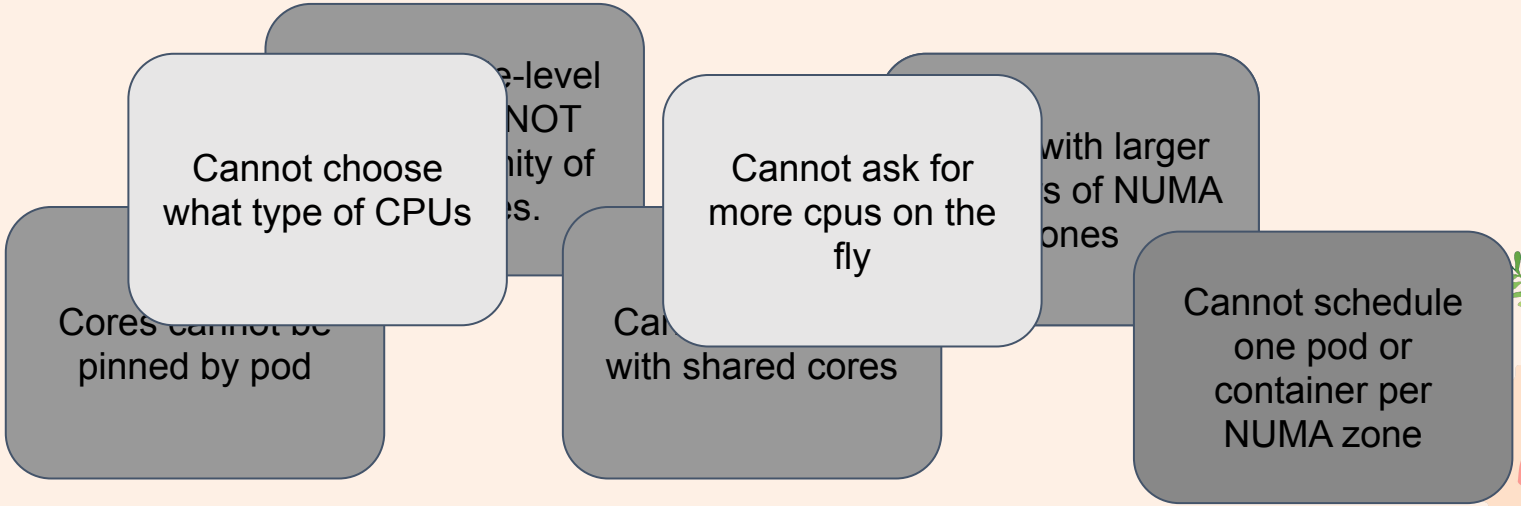


NOW WHAT???

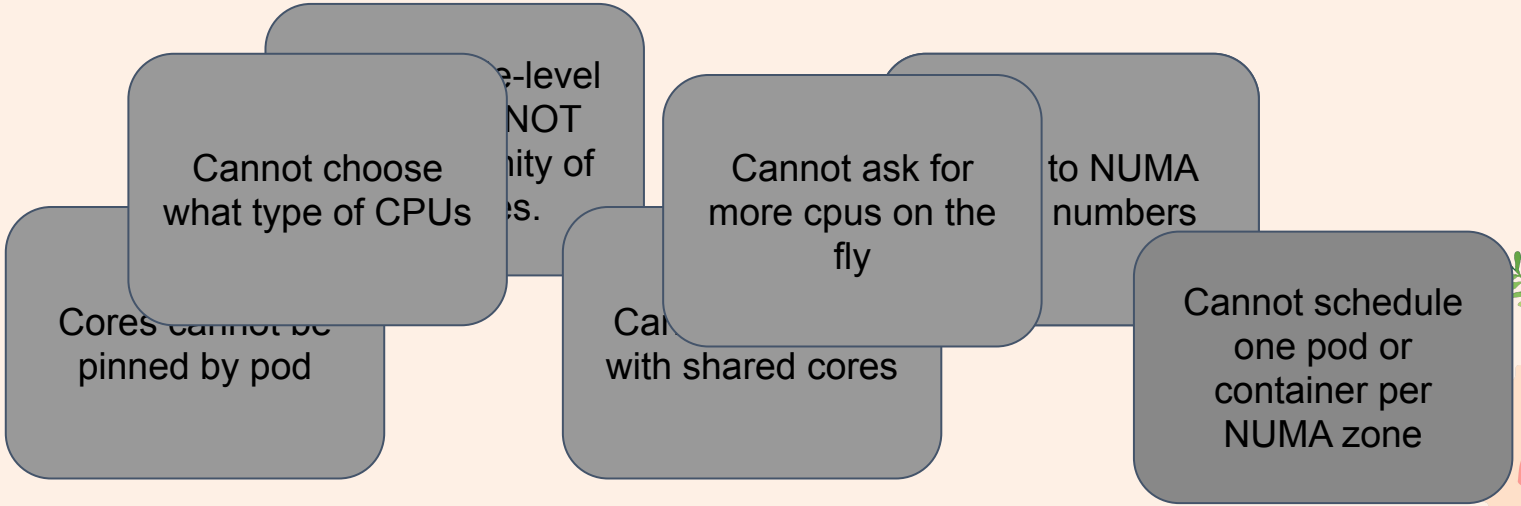
Current Gaps for Heterogeneous Clusters

- Affinity node-level only. CANNOT choose affinity of resources.
- Limits with larger numbers of NUMA zones
- Cores cannot be pinned by pod
- Can't mix pinned with shared cores
- Cannot schedule one pod or container per NUMA zone

Current Gaps for Heterogeneous Clusters



Job Security:
SO MUCH TO DO!!!



Other Options?

Out of Tree solutions for CPU Management

- All rely on turning off all resource management by the Kubelet - specifically cpu manager features

CRI Resource Manager (CRI-RM)

- a pluggable add-on for controlling resource assignment to containers
- plugs in between the Kubelet and the container runtime
- keeps track of the states of all containers on a node
- intercepts CRI protocol requests from the kubelet

CPU Pooler™

- A solution for Kubernetes to manage predefined, distinct CPU pools of Kubernetes Nodes
- physically separate the CPU resources of the containers connecting to the various pools
- A Device Plugin that exposes the CPU cores as consumable devices to Kubernetes.

CMK
(deprecated now)

- accomplished core isolation by controlling the logical CPUs each container may use for execution
- wrapped target application commands with the CMK command-line program for managing CPU pools and constraining workloads to specific CPUs

Out of Tree solutions for CPU Management

- All rely on turning off all resource management by the Kubelet

BUILD YOUR OWN

- Go around the Kubelet, manage resources on your own, at whatever level you would like.

How Can YOU Get Involved?

Community Discussion

- **CPU Management Kubelet Use Cases, unaddressed:** Initial document to gather current requirements not handled by the kubelet
  https://docs.google.com/document/d/1U4jjRR7kw18Rllh-xpAaNTBcPsK5jl48ZAVo7KRqkJk

- **Kubelet Resource Plugin RFC:** Suggestion to move to a Kubelet Resource Plugin model, splitting the kubelet into a control plane (node-level type management) that uses a plugin model, and a data plane (advertises to the scheduler resource availability)
  https://docs.google.com/document/d/1O5G4HMhfyC9AdaGai1eV5OJpCugV3vFIW19_FCuMOaY

- **NRI**: Node Resource Interface: CNI-type interface for managing resources on a node for Pods and Containers
  https://github.com/containerd/nri

- **Dynamic Resource Management**: https://github.com/kubernetes/enhancements/pull/3064
  - An alternative to the device plugin API
  - Primary idea is that resource allocations can be ephemeral or persistent
  - Ability to enable users to specify resources with resource specific parameters

Kubernetes SIGs

SIG Node: Resources directly on the Node

- Github: https://github.com/kubernetes/community/tree/master/sig-node
- Slack: SIG Node Slack: https://kubernetes.slack.com/channels/sig-node

SIG Scheduling: Which nodes the workloads get scheduled to

- Github: https://github.com/kubernetes/community/tree/master/sig-scheduling
- Slack: https://kubernetes.slack.com/channels/sig-scheduling

Working Groups

Topology aware Scheduling: Topology scheduling, part of scheduling

- Github: https://github.com/k8stopologyawareschedwg
- Slack: https://kubernetes.slack.com/channels/topology-aware-scheduling

CNCF TAG Runtime

- Github: https://github.com/cncf/tag-runtime
- Slack: https://cloud-native.slack.com/messages/CPBE97SMU

Batch Working group: Large-scale compute concerns, similar to HPC

- Github: https://github.com/kubernetes/community/tree/master/wg-batch
- Slack: https://kubernetes.slack.com/channels/wg-batch

Questions?

Marlow Weston: Marlow.weston@intel.com
Slack: @mweston
Swati Sehgal: swsehgal@redhat.com
Slack: @swsehgal