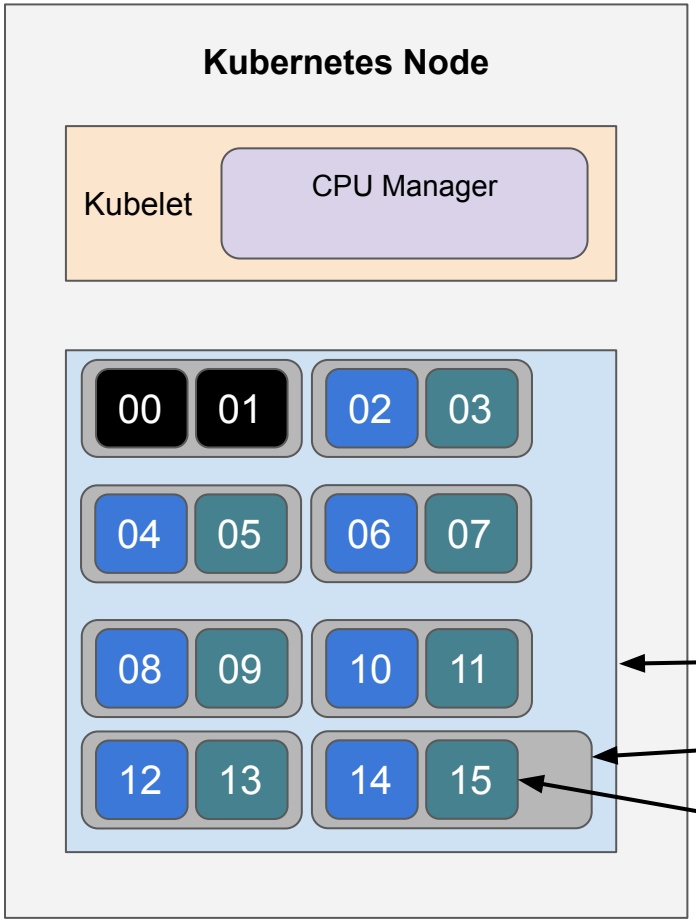
more SMT-awareness in the CPU Manager

@fromani (fromani@redhat.com) @swatisehgal (swsehgal@redhat.com)

Full session and KEP to be presented to sig-node meeting



For simplicity sake: let's consider a node, with 1 CPU, 2-way SMT capable, with 8 physical cores, and (8*2=)16 virtual cores.

We will use: "virtual core", "hw thread" as synonyms

Reserved (--reserved-cpus) unavailable for workloads

CPU Complex (package - TODO: check the term)

HW thread pair (physical core - 2-way SMT)

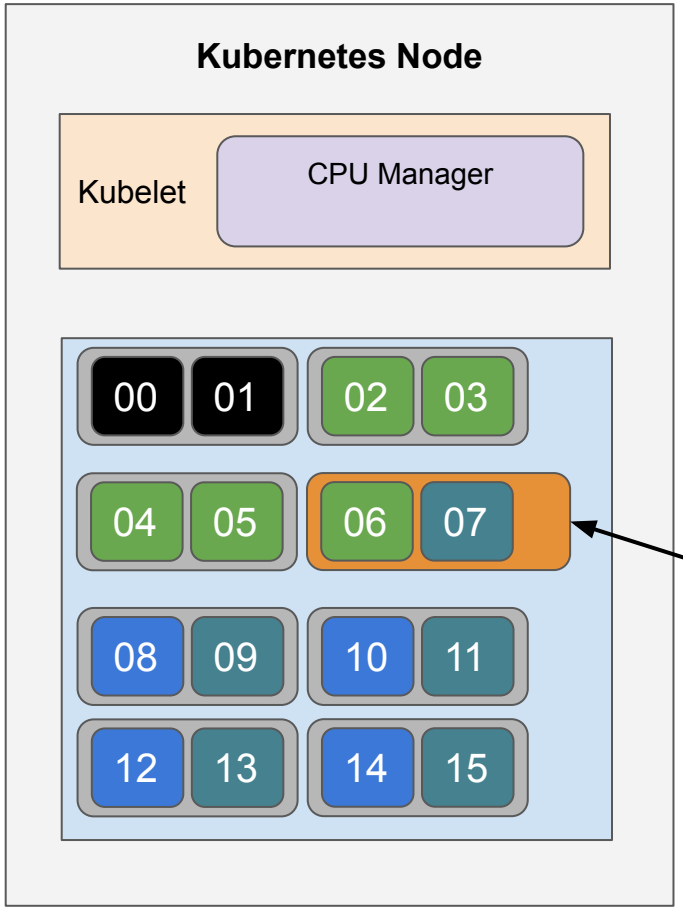
HW thread (virtual core)

Red Hat

Current Behaviour of CPU Manager with static policy

- Containers have access to exclusive CPUs on the node
- On SMT-enabled systems, this means virtual CPUs - aka HW threads
- Some applications require more isolation, at HW thread level
  - Latency-sensitive applications (DPDK, RT)
  - Mitigate cache-based side channel attacks
- Similar capabilities are already present in OpenStack

```
apiVersion: v1
kind: Pod1
Spec:
  containers:
  - name: nginx
image: nginx
resources:
        limits:
          memory: "256Mi"
          cpu: "5"
        requests:
          memory: "256Mi"
          cpu: "5"
```



Reserved core

Virtual Core allocated to Pod1

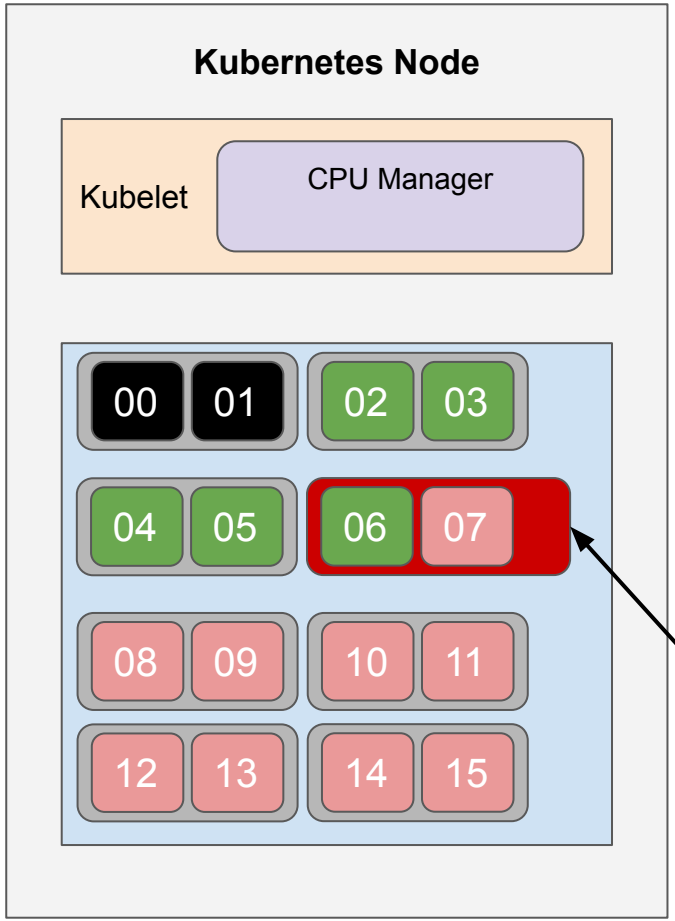
Potential for noisy neighbour!

When different containers run on the same physical cpu

Red Hat

```
apiVersion: v1
kind: Pod1
Spec:
  containers:
  - name: nginx
image: nginx
resources:
        limits:
          memory: "256Mi"
          cpu: "5"
        requests:
          memory: "256Mi"
          cpu: "5"
```

```
apiVersion: v1
kind: Pod2
Spec:
  containers:
  - name: nginx
image: nginx
resources:
        limits:
          memory: "256Mi"
          cpu: "9"
        requests:
          memory: "256Mi"
          cpu: "9"
```



Reserved core

Virtual Core allocated to Pod1

Virtual Core allocated to Pod2

Actual noisy neighbour!
Two container share the same physical core!

HW thread share some silicon (part of execution units, L2 cache...) - so even **non malicious** containers interfer to each other.

Red Hat

Proposal: add two new CPU Manager Policies to make it more SMT-aware

1. **smt-aware** - aiming for kubernetes 1.22 - new policy with minimal changes, to prevent noisy neighbours

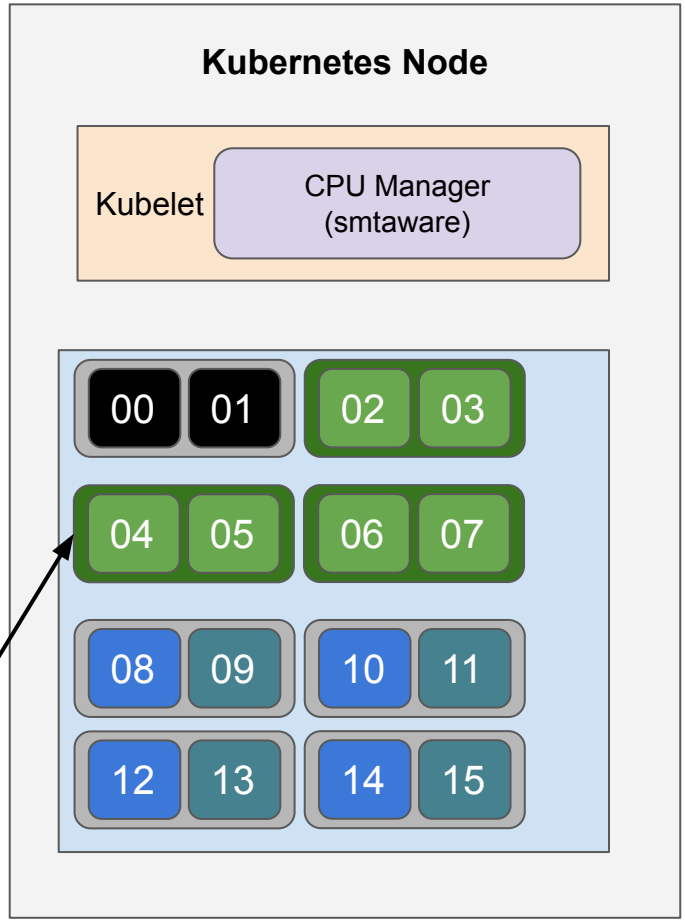
2. **smt-isolate** -aiming for kubernetes 1.23+ - new policy to emulate no-smt on smt-enabled machines

smt-aware policy

```
apiVersion: v1
kind: Pod1
Spec:
  containers:
  - name: nginx
image: nginx
resources:
        limits:
          memory: "256Mi"
          cpu: "5"
        requests:
          memory: "256Mi"
          cpu: "5"
```

Ask for 5 cores
Get 6 **virtual** cores

We need to reconcile the resource accounting (solution WIP)



Reserved core

Virtual Core allocated to Pod1

Physical Core allocated to Pod1

Always allocate full physical cores

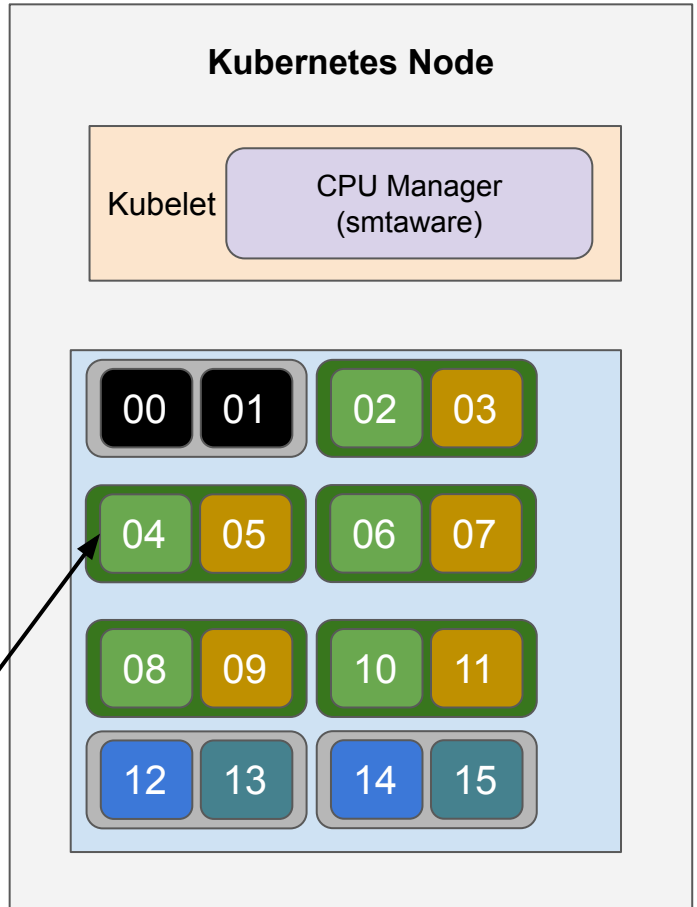
Round up allocated physical cores to prevent any possible noisy neighbours

smt-isolate policy

```
apiVersion: v1
kind: Pod1
Spec:
  containers:
  - name: nginx
image: nginx
resources:
        limits:
          memory: "256Mi"
          cpu: "5"
        requests:
          memory: "256Mi"
          cpu: "5"
```

Ask for 5 cores
Get 10 **virtual** cores

We need to reconcile the resource accounting (solution WIP)



Reserved core

Virtual Core allocated to Pod1

Physical Core allocated to Pod1

Virtual core accounted to Pod1, but not usable

Allocate the requested amount of physical (no longer virtual) cores

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make
Red Hat a trusted adviser to the Fortune 500.


linkedin.com/company/red-hat


youtube.com/user/RedHatVideos


facebook.com/redhatinc


twitter.com/RedHat

Red Hat