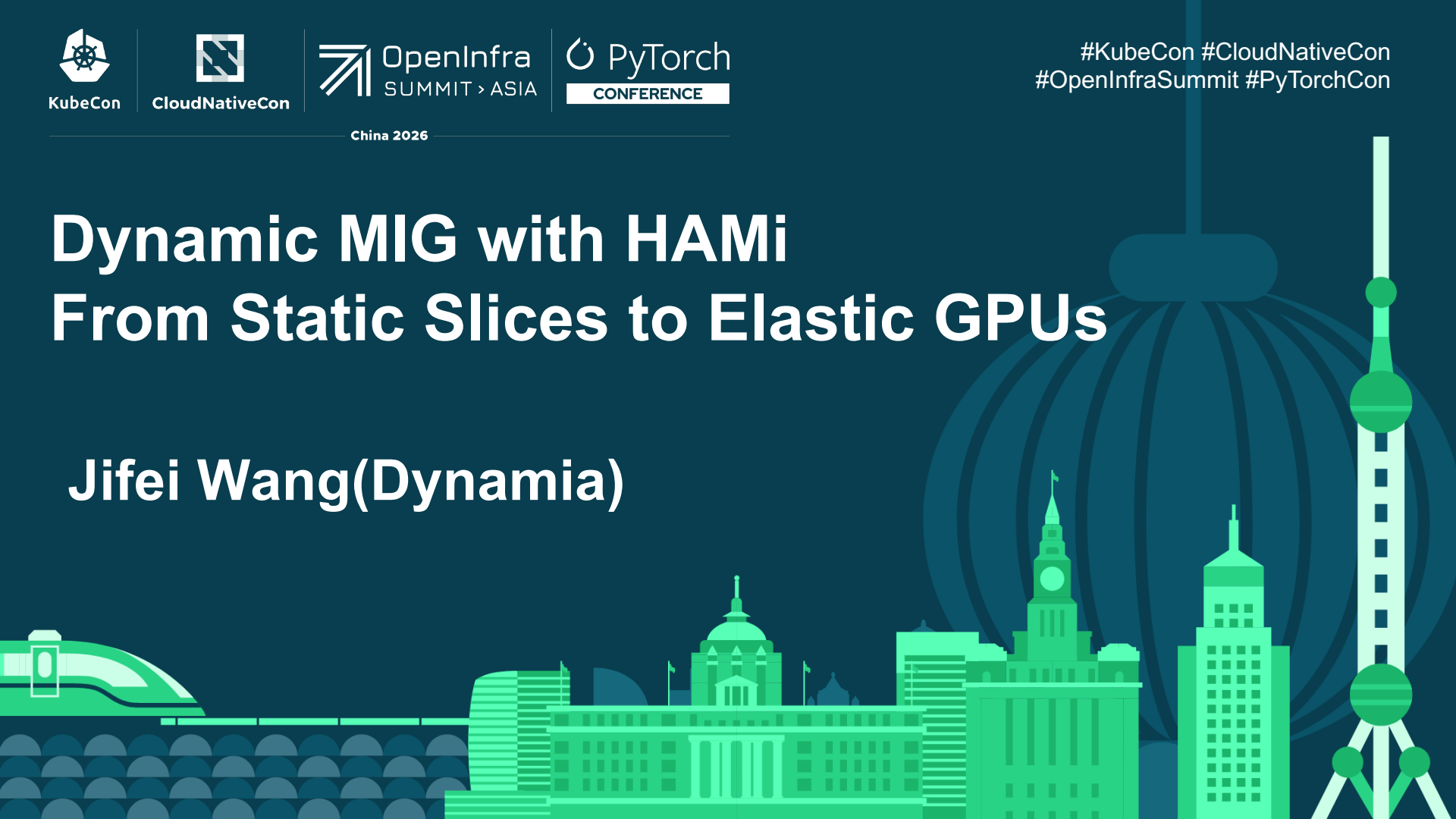
KubeCon CloudNativeCon OpenInfra SUMMIT › ASIA PyTorch CONFERENCE

China 2026

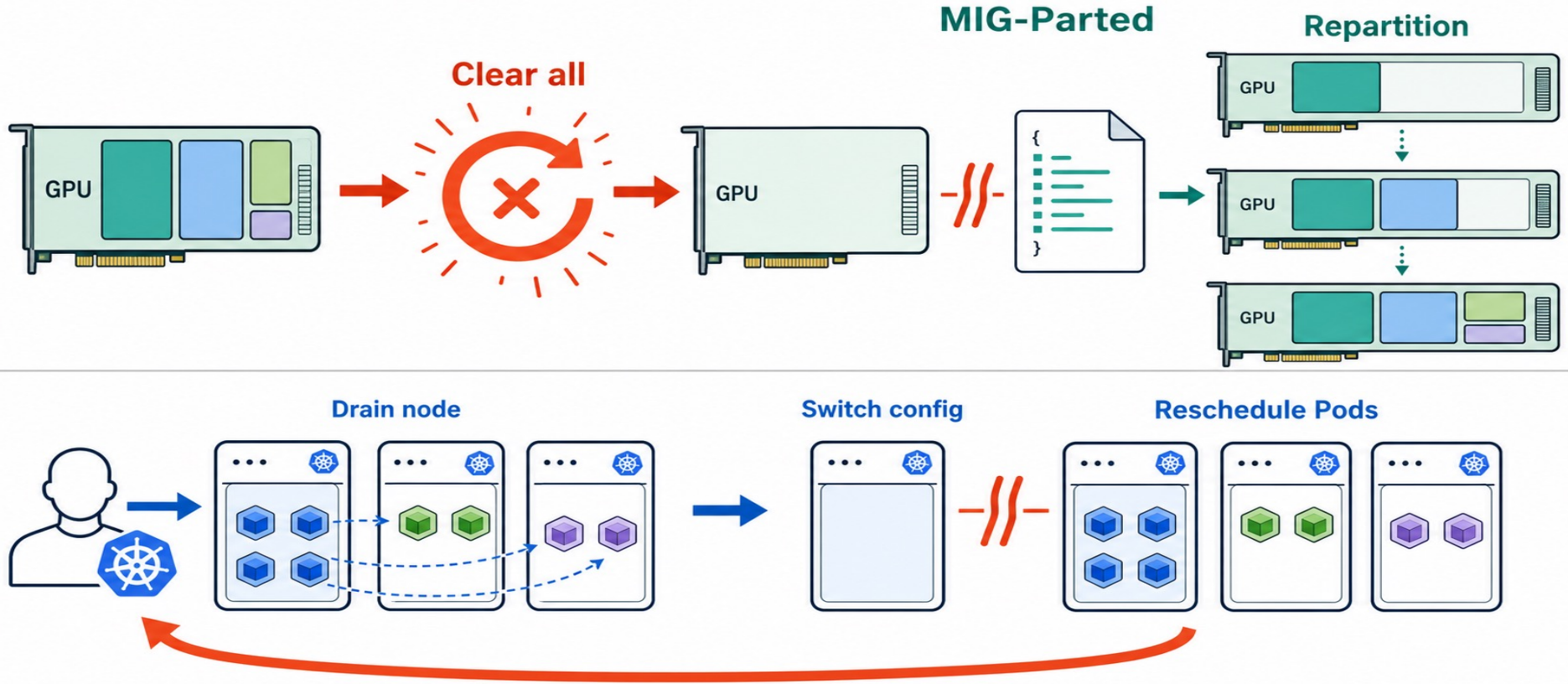
#KubeCon #CloudNativeCon
#OpenInfraSummit #PyTorchCon

Dynamic MIG with HAMi
From Static Slices to Elastic GPUs

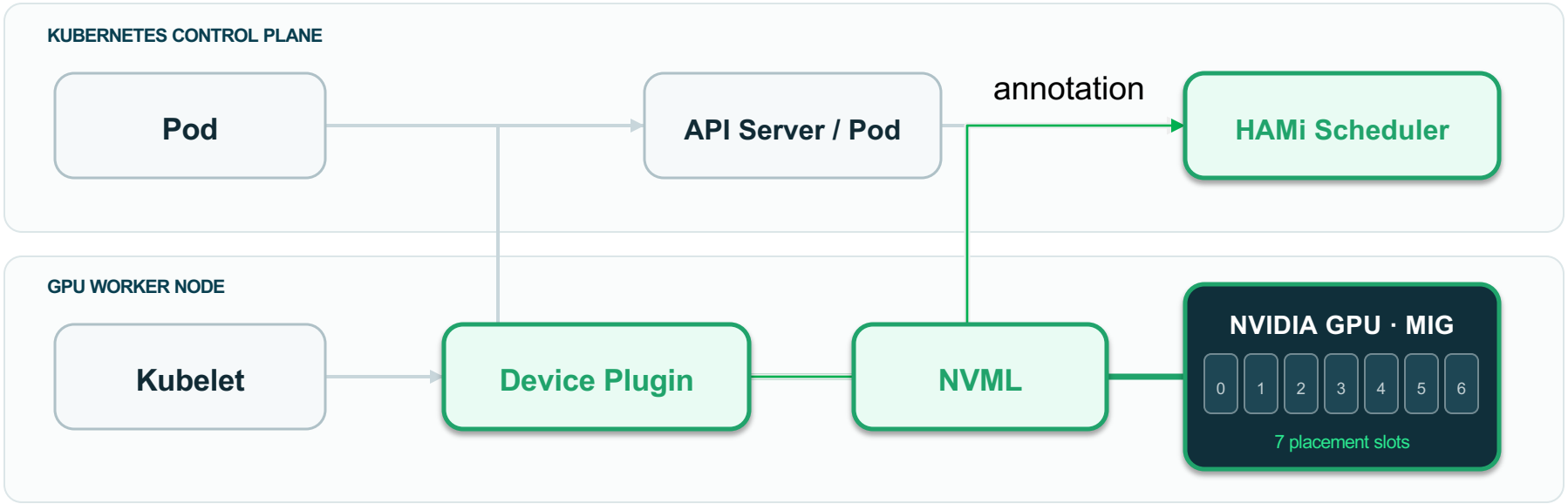
Jifei Wang(Dynamia)



Classic MIG manager: MIG partd

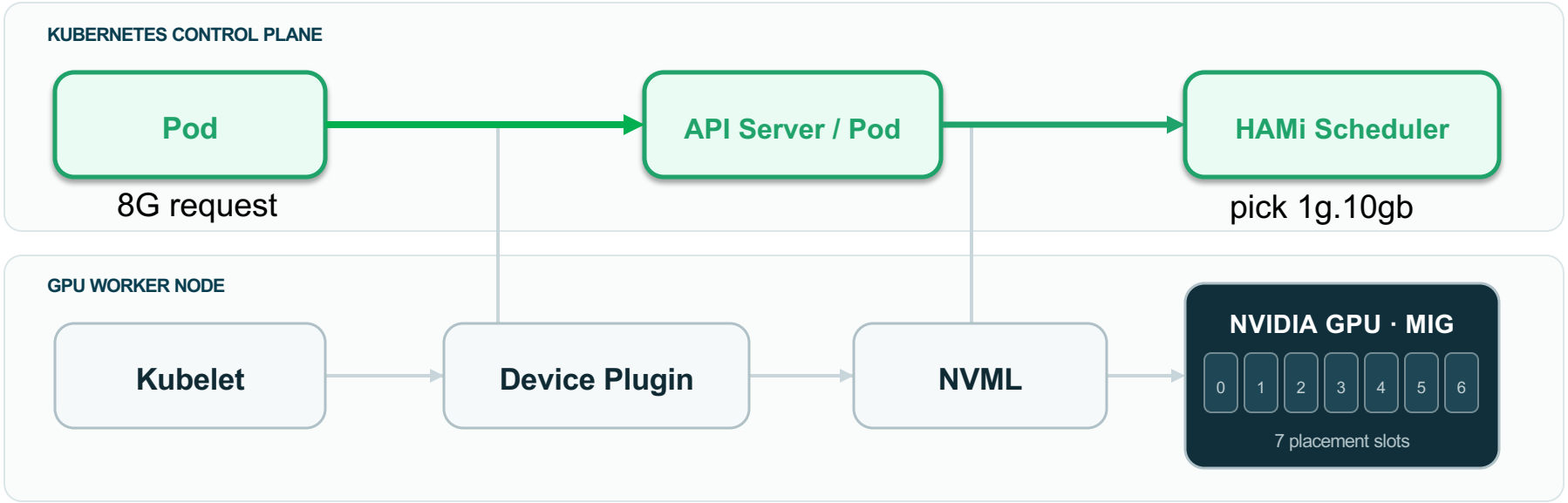


Dynamic MIG — Publish capacity



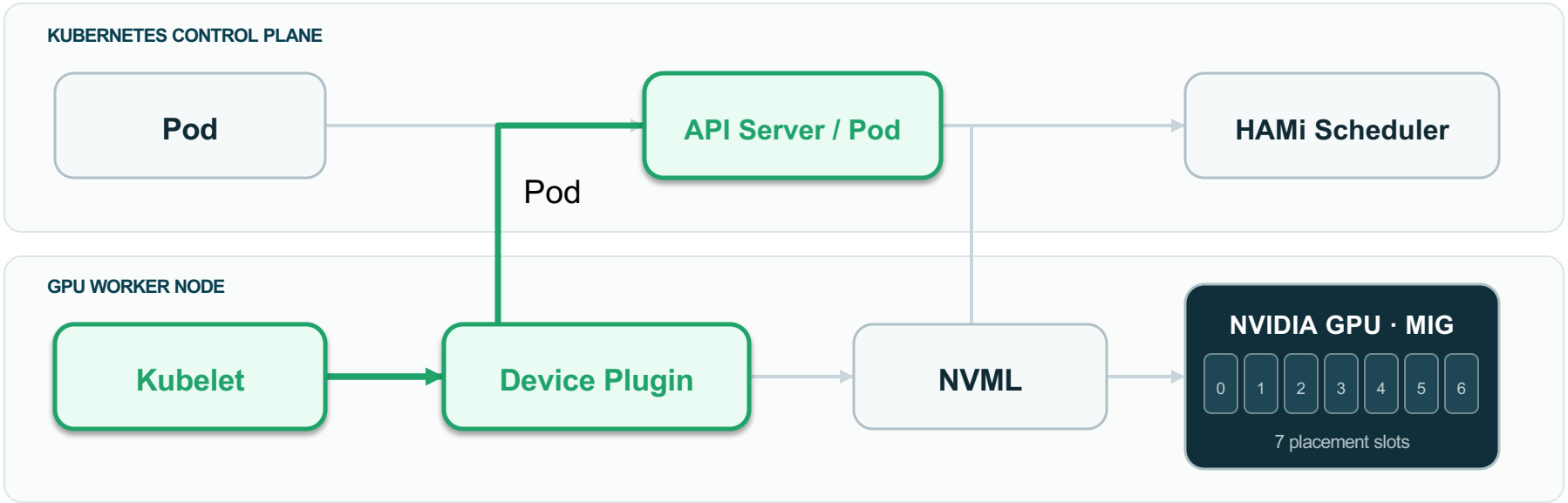
The Device Plugin call NVML to find out possible MIG profiles and placement and report them to the node annotation. The scheduler use it to discover devices.

Dynamic MIG — Reserve allocation



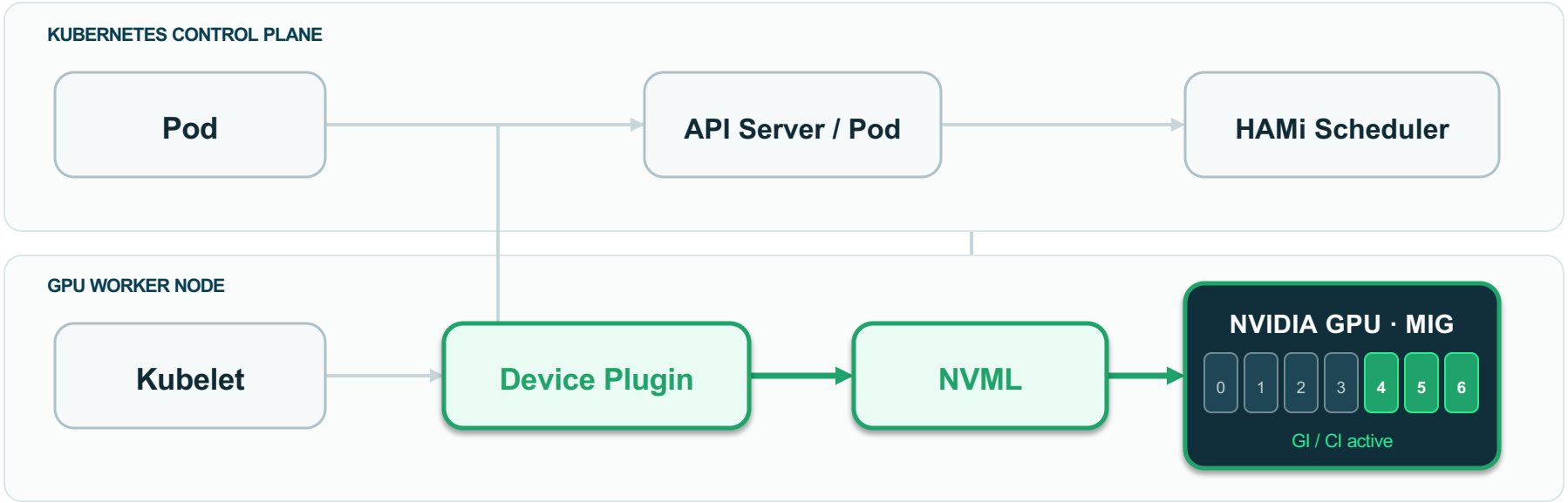
When we submit a pod request the MIG device, the scheduler find out the avaliable placement and annotate the result to the pod.

Dynamic MIG — Allocate on the node



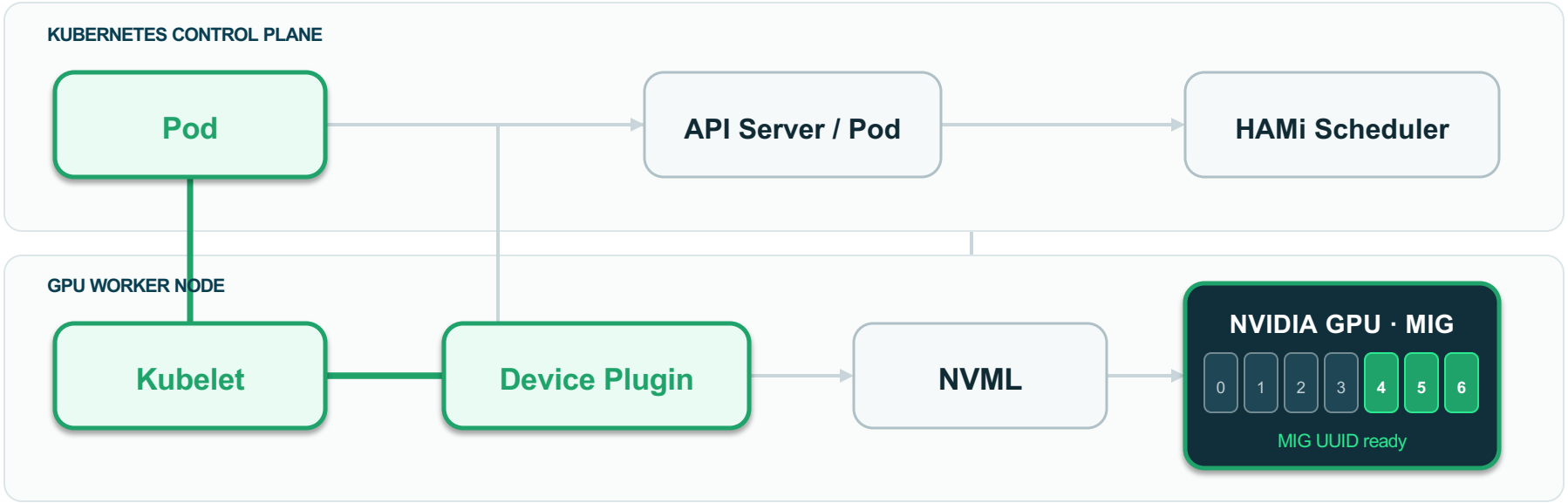
Kubelet call Device Plugin Allocate(), the Device plugin resolve Pod annotation instead of device id to build the MIG device request.

Dynamic MIG — Create MIG Instance



The Device Plugin call NVML to Create MIG instance with determined placement and profile which guarantee valid by the scheduler.

Dynamic MIG — Return runtime identity



After the driver return the MIG device UUID, we can finally mount the exact MIG device to the Pod using the UUID.

VS Code Settings

问题 输出 调试控制台 终端 端口

```
poff@a100-test ~> nvidia-smi -L
GPU 0: NVIDIA A100-SXM4-40GB (UUID: GPU-4cf5c6c5-b53a-a002-dd51-59c9f46638bb)
poff@a100-test ~> kubectl apply -f hami-dynamic-mig-1g.yaml
pod/hami-dynamic-mig-1g created
poff@a100-test ~> kubectl apply -f hami-dynamic-mig-2g.yaml
pod/hami-dynamic-mig-2g created
poff@a100-test ~> kubectl get pod
NAME                  READY   STATUS    RESTARTS   AGE
hami-dynamic-mig-1g   1/1     Running   0          15s
hami-dynamic-mig-2g   1/1     Running   0          7s
poff@a100-test ~> nvidia-smi -L
GPU 0: NVIDIA A100-SXM4-40GB (UUID: GPU-4cf5c6c5-b53a-a002-dd51-59c9f46638bb)
  MIG 2g.10gb     Device  0: (UUID: MIG-79b07c44-6473-5363-8f21-7b06a11531b8)
  MIG 1g.5gb      Device  1: (UUID: MIG-8fbf9680-7223-5bb3-8799-969bdfee7ac4)
poff@a100-test ~> kubectl get pod
```

⊗ 0 ⚠ 0

New 3g request cannot be placed

Both legal start positions are already occupied

Pod stays Pending — free slices exist, but no legal 3g placement fits.

Future enhancements

Config	Slice #0	Slice #1	Slice #2	Slice #3	Slice #4	Slice #5	Slice #6	OFA	NVDEC	NVJPEG	P2P	GPU Direct RDMA
1	7							1	7	7	No	Supported MemBW proportional to size of the instance
2	4				3			0	4+3	4+3	No	
3	4				2		1	0	4+2+1	4+2+1	No	
4	4				1	1	1	0	4+1+1+1	4+1+1+1	No	
5	3				3			0	3+3	3+3	No	
6	3					1		0	3+2+1	3+2+1	No	
7	3			1	1	1		0	3+1+1+1	3+1+1+1	No	
8	2		2		3			0	2+2+3	2+2+3	No	
9	2		1	1	3			0	2+1+1+3	2+1+1+3	No	
10	1	1	2		3			0	1+1+2+3	1+1+2+3	No	
11	1	1	1	1	3			0	1+1+1+1+3	1+1+1+1+3	No	
12	2		2		2		1	0	2+2+2+2+1	2+2+2+2+1	No	
13	2		1	1	2		1	0	2+1+1+2+1	2+1+1+2+1	No	
14	1	1	2		2		1	0	1+1+2+2+1	1+1+2+2+1	No	
15	2		1	1	1	1	1	0	2+1+1+1+1+1	2+1+1+1+1+1	No	
16	1	1	2		1	1	1	0	1+1+2+1+1+1	1+1+2+1+1+1	No	
17	1	1	1	1	2		1	0	1+1+1+1+2+1	1+1+1+1+2+1	No	
18	1	1	1	1	1	2		0	1+1+1+1+1+2	1+1+1+1+1+2	No	
19	1	1	1	1	1	1	1	0	1+1+1+1+1+1+1	1+1+1+1+1+1+1	No	

Well design scheduling to maximize utilization for NVIDIA A, H and B series.

On the shoulders of giants

Special thanks for dra-driver-nvidia-gpu for:

1. Prove feasibility of dynamic MIG management
2. Providing reference implementation of CDI
3. Pushing adaptation of ecosystem like dcgm-exporter

But it still have problems:

1. Unable to customize scheduling strategy
2. DRA requires high kubernetes version

DRA is the future, but we must bring everyone along.

KubeCon CloudNativeCon OpenInfra SUMMIT › ASIA PyTorch CONFERENCE

China 2026

Thank you