KubeCon | CloudNativeCon

China 2018

Deep Customized Kubernetes for Machine Learning in Tencent

Shengbo Song, thomassong@tencent.com

Agenda

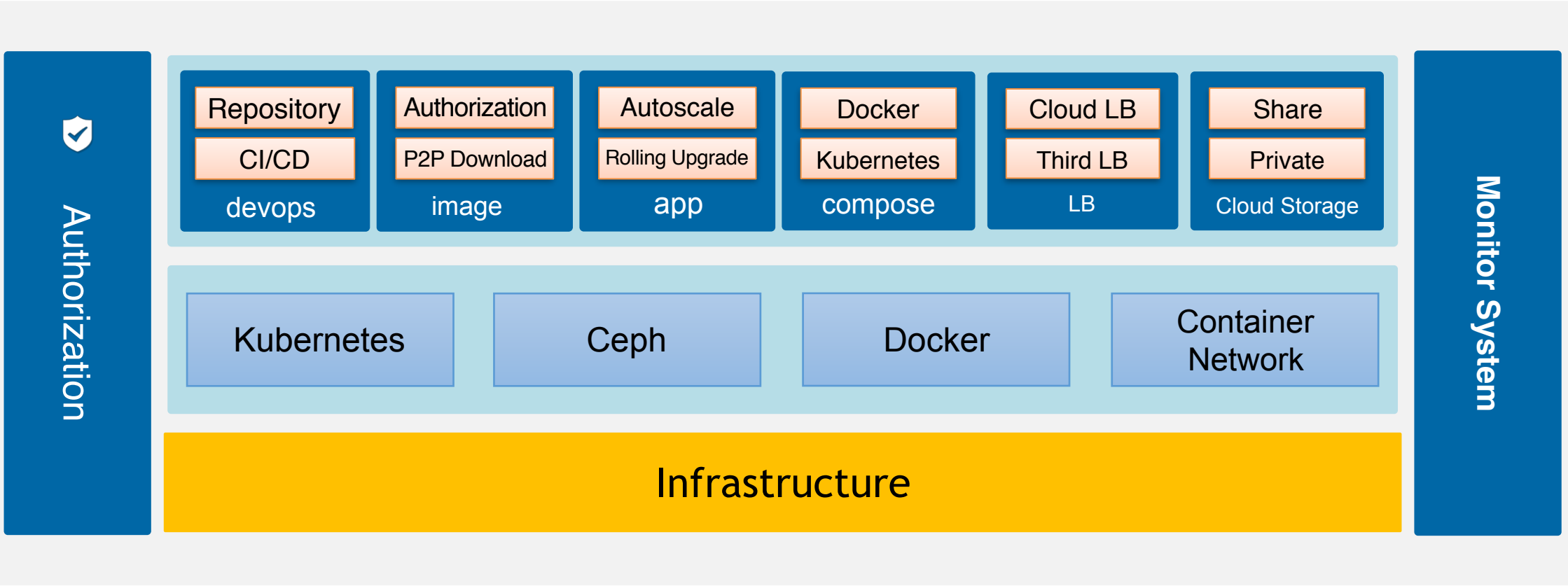
1. GaiaStack Introduction
2. Why need a custom Kubernetes
3. Highlights of GaiaStack

KubeCon | CloudNativeCon

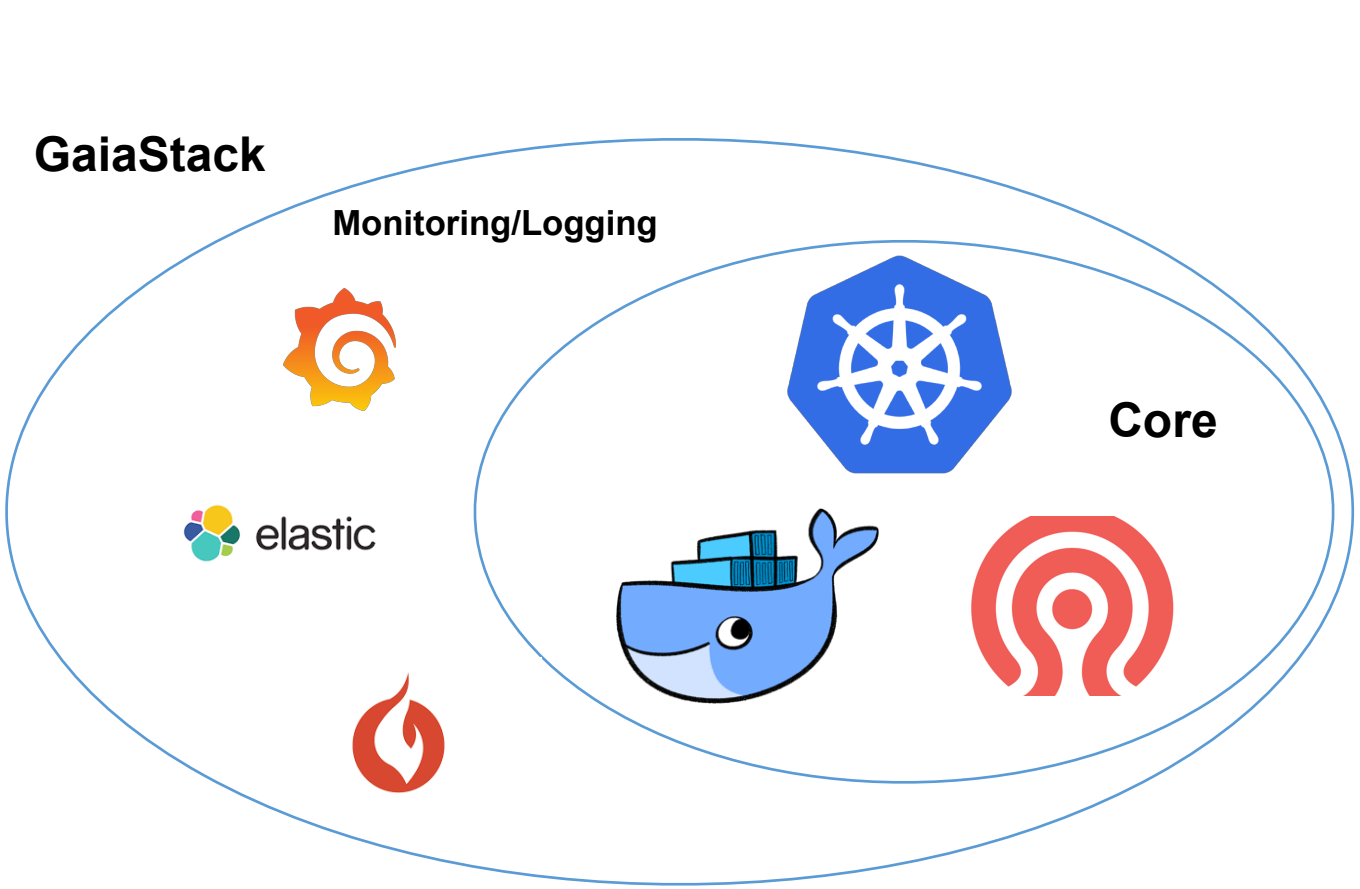
China 2018

Intro to GaiaStack

GaiaStack Overview



GaiaStack ecosystem



Apache Spark

Caffe2

PYTORCH

...

KubeCon | CloudNativeCon

China 2018

Why need a custom Kubernetes

Why need a custom Kubernetes

- Kubernetes official release is not designed for ML
  - GPU Topology awareness
  - Host devices for some apps
  - GPU Resource management
  - Deployments, Statefulset, Job, CronJob isn't suitable for ML
  - ...

- Need some custom design and optimization for ML apps

KubeCon | CloudNativeCon

China 2018

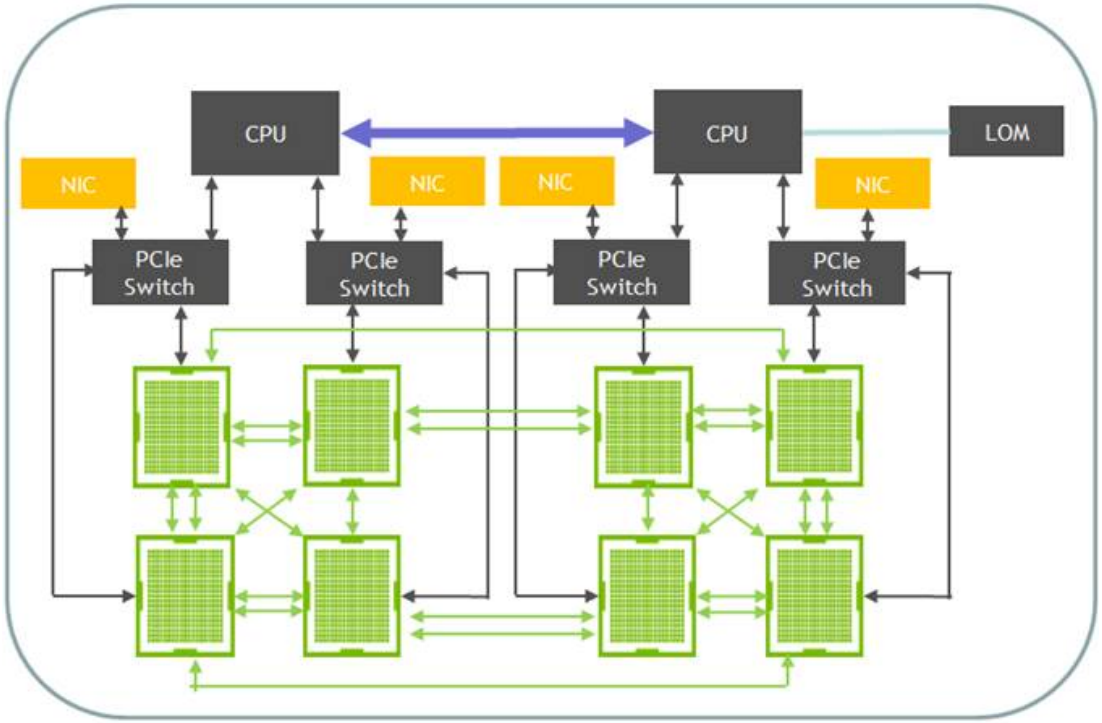
Highlights on GaiaStack

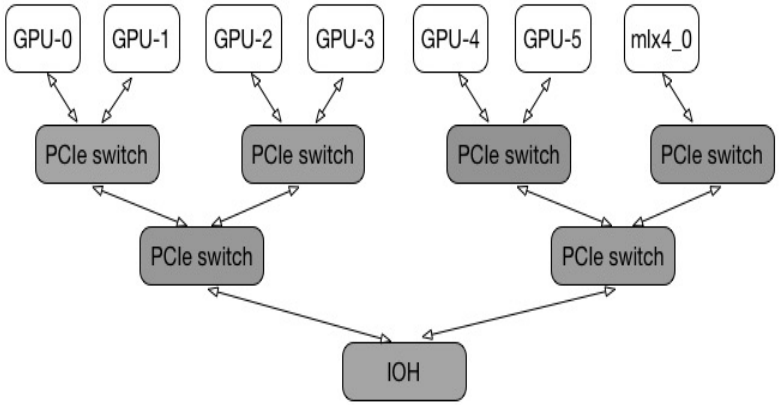
Highlights on GaiaStack

- GPU topology awareness scheduler
- GPU resource management
- Tapp - a CRD of Kubernetes
- Galaxy - a powerful CNI plugin

GPU Topology scheduler

- **NVIDIA GPU topology**





Different combination of GPU for an app causes big difference performance results.

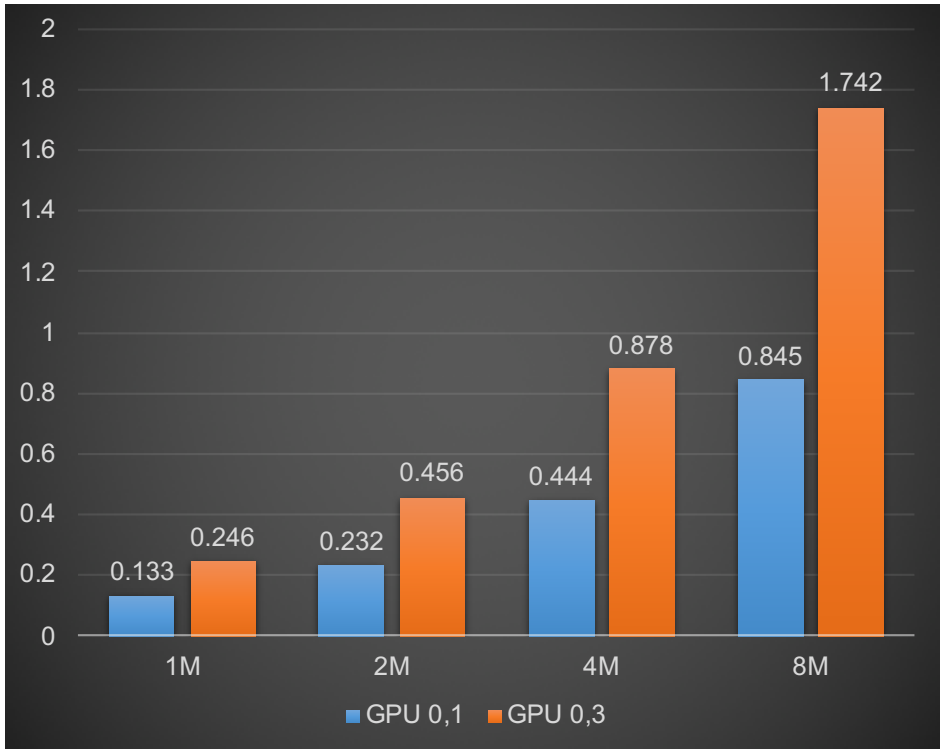
For example, an app runs on GPU-0 and GPU-1 has shorter execution time than the one with GPU-0 and GPU-3

GPU Topology scheduler

- **Gaia GPU scheduler**
  a scheduler can be aware of the topology of GPUs

	GPU0	GPU1	GPU2	GPU3
GPU0	X	PIX	SYS	SYS
GPU1	PIX	X	SYS	SYS
GPU2	SYS	SYS	X	SYS
GPU3	SYS	SYS	SYS	X

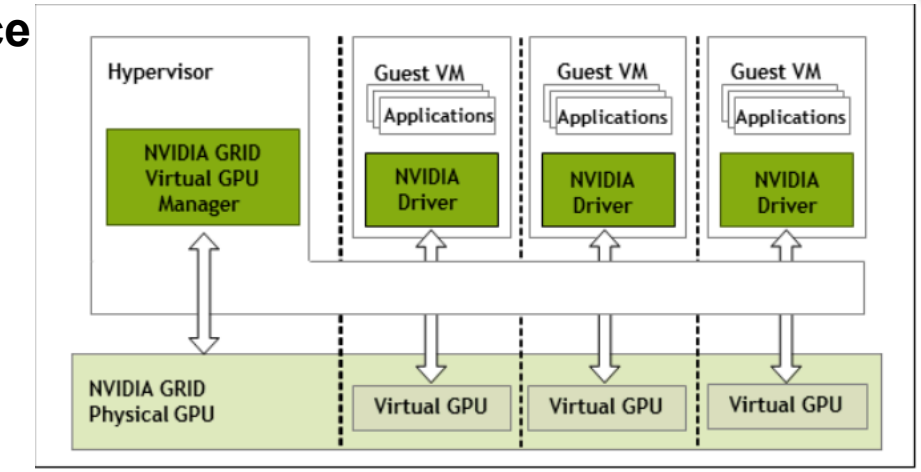
The chosen combination with Gaia scheduler (GPU0,1) has 200% data transmit speed than the default scheduler does (GPU0,3)

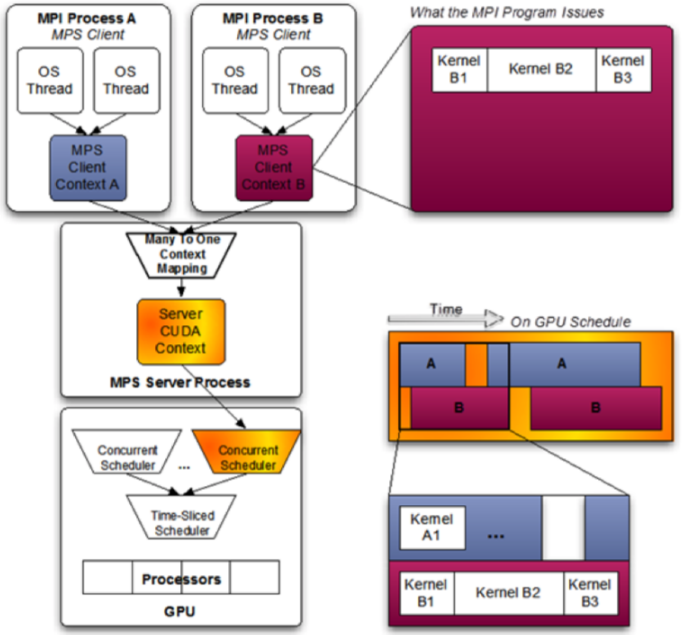


GPU Share and Limit

NVIDIA supports sharing single GPU by:

- **VMs use NVIDIA GRID**
- **Processes use MPS Service**





Pros:

- **Official support**
- **Easy to use**

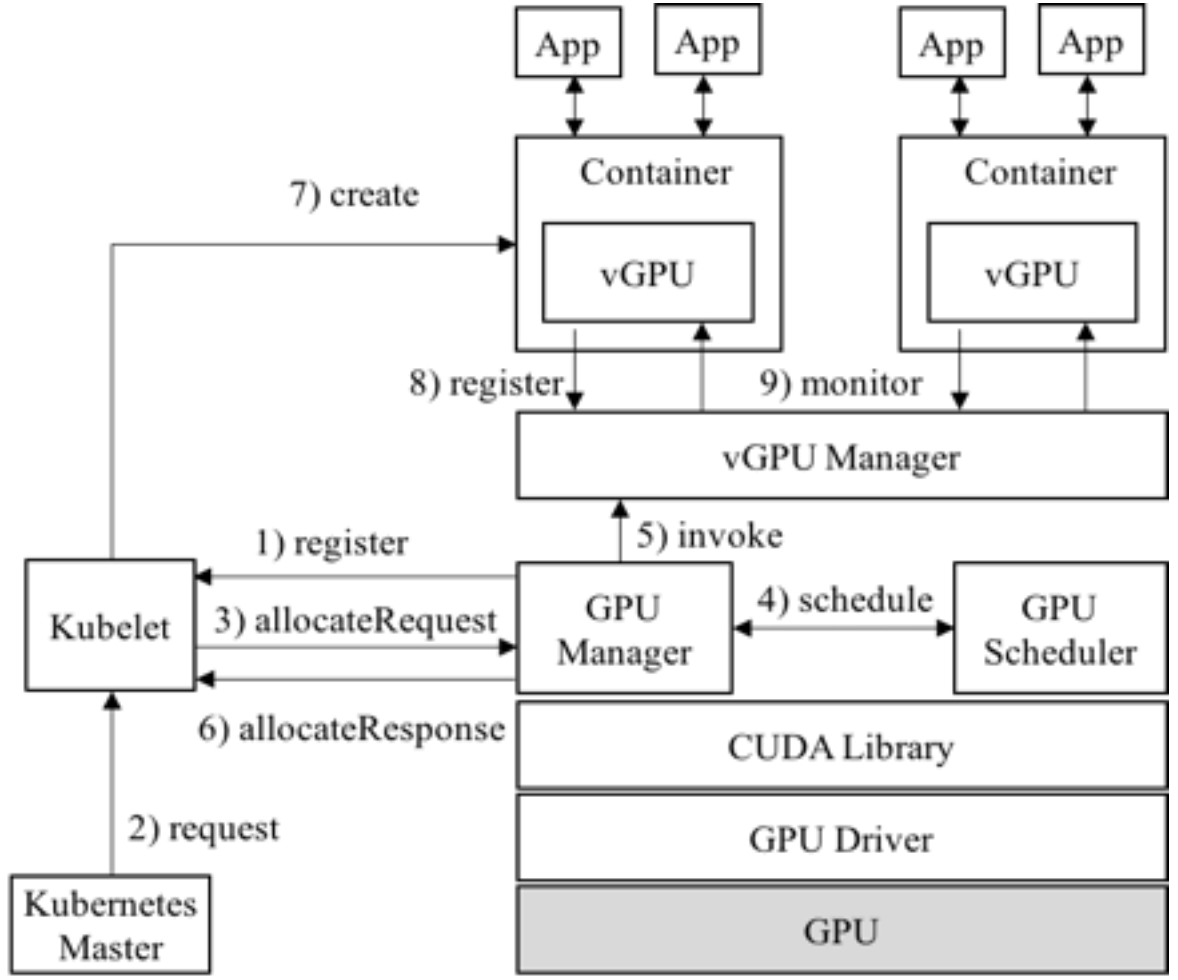
Cons:

- **NVIDIA GRID not suitable for Kubernetes which runc is default container runtime**
- **Single MPS client can affect other MPS client and MPS server**
- **Hard resource limits, not changed after process is running**
- **Time-slice is not based on requests of containers**

GPU Share and Limit

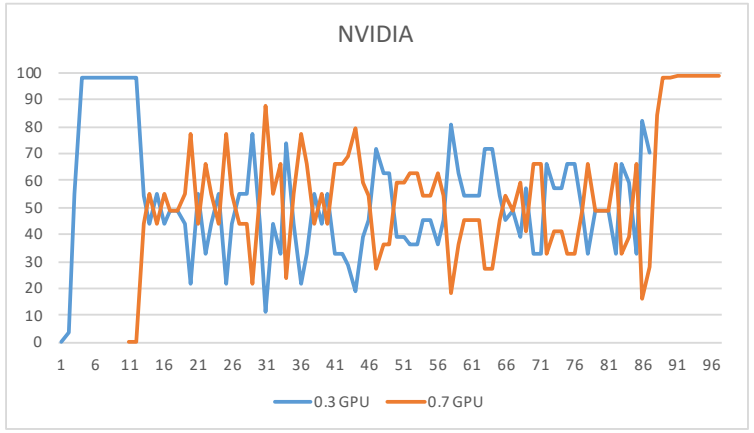
Our solutions:

- **Use a lightweight server**
- **Time-slice is based on request of share**
- **Clients are not affected each other**
- **The limits can be changed at any time**
- **Zero-injection to user applications**



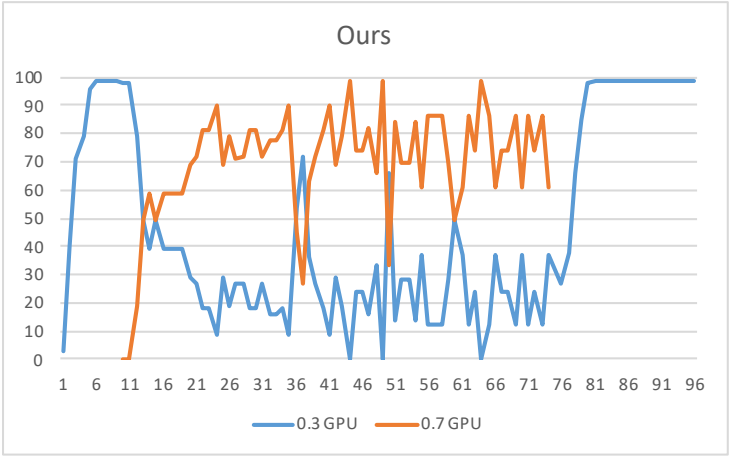
GPU Share and Limit

AlexNet on Tesla P4				
	NVIDIA		Ours	
	Execution Time		Utilization	
0.3 GPU	87.3s	97.79s	57.16%	46.59%
0.7 GPU	87.86s	66.91s	47.98%	69.62%
Total Time	97.86s	97.79s		



The default strategy of NVIDIA is divided the available thread equally to each clients and first run first finish.

But our strategy is divided the available thread based on request of share, large share first finish.



Tapp - a CRD of Kubernetes

Tapp is a CRD of Kubernetes which is similar with Statefulset but more powerful than it.

Similarity:

- Pod has a increased and unique id
- Support Persistent Volumes
- ...

Difference:

- Repeated operations of any Pod, including deletion, stop, reboot, rolling upgrade, downgrade
- Different versions of Pod existed in the same Tapp
- Change image version of a Pod, even add a new container

Galaxy - a powerful CNI plugin

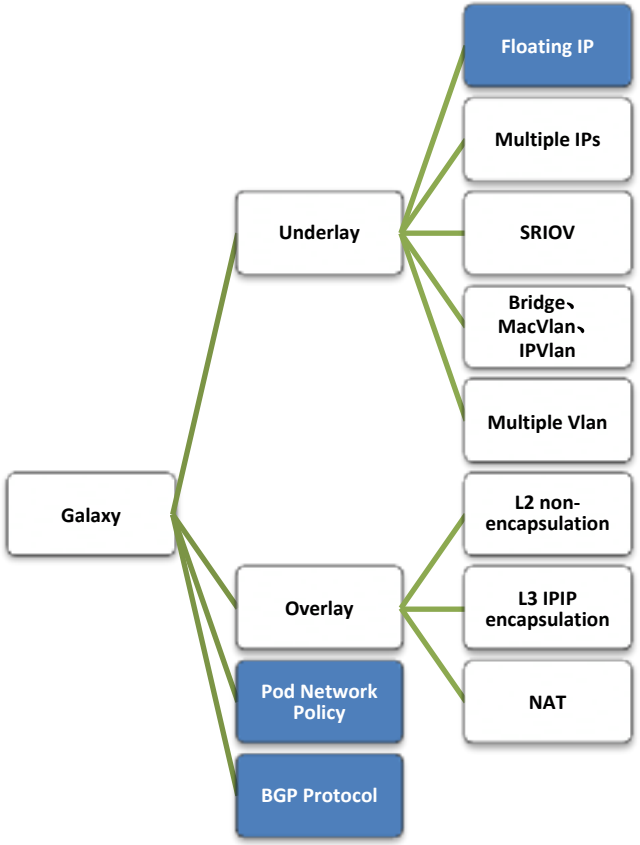
For various scenario, we design a CNI plugin called Galaxy

Pros:

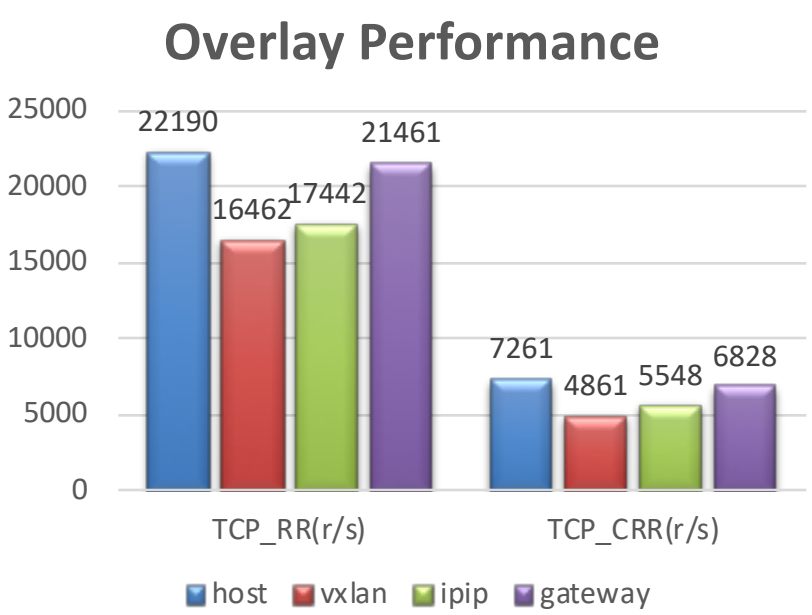
- **Underlay + Overlay**
- **Adapt to any network scenario**
- **Good Performance**
- **Zero-injection to client network**

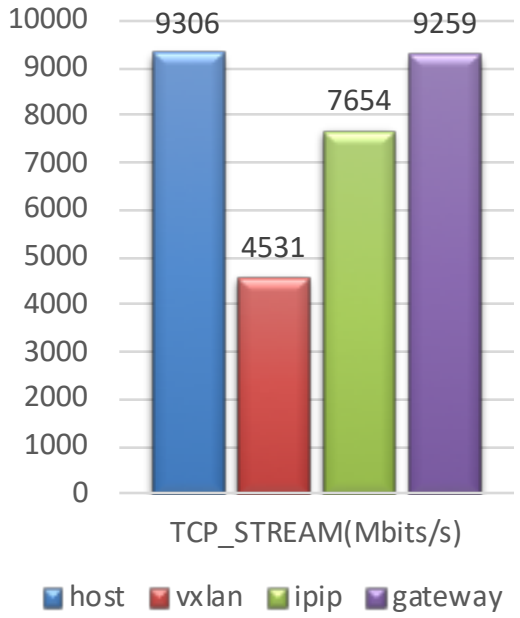
For application:

- **Different apps can choose different network mode**
- **Pods on the same host can have different network mode**



Galaxy - a powerful CNI plugin





Overlay solution on GaiaStack is IPIP + Host Gateway.

The performance is +14%-+40% than Vxlan(flannel), and our solution is accepted by flannel community.

KubeCon

CloudNativeCon

China 2018

Thank you !