KubeCon | CloudNativeCon

OPEN SOURCE SUMMIT

China 2023

Practice of Building Large-Scale AI Training Cluster Based on Kubernetes and RoCEv2

Dekui WANG, IEI

KubeCon | CloudNativeCon
OPEN SOURCE SUMMIT
China 2023

Agenda

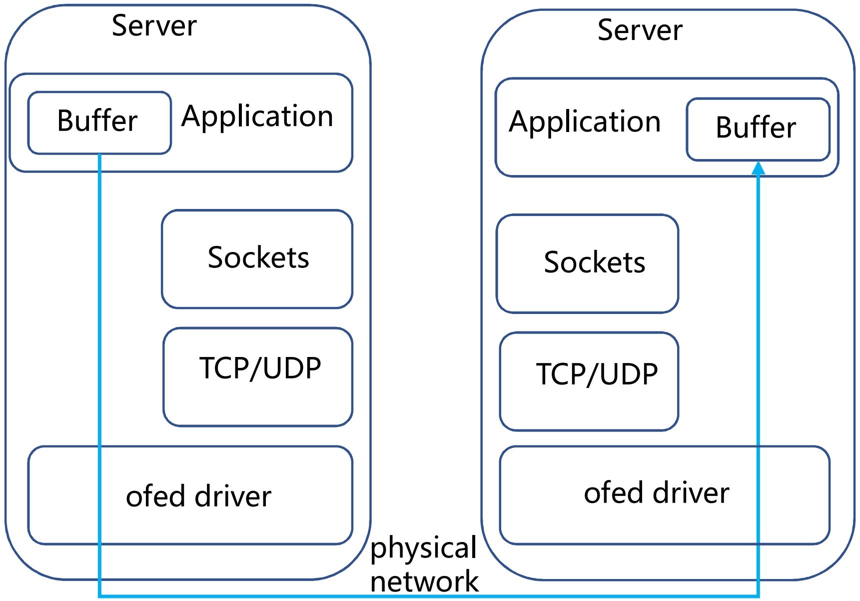
KubeCon CloudNativeCon OPEN SOURCE SUMMIT China 2023

1. Background and Challenges
2. RoCEv2 solution
3. Practice and Tests

Network issues in AI training infrastructure

1. large scale training jobs require multiple nodes
2. network congestion problems
3. the difference between Infiniband and RoCE
4. GPU node with multiple GPU cards and multiple rdma cards

RDMA（Remote Direct Memory Access）



rdma data flow

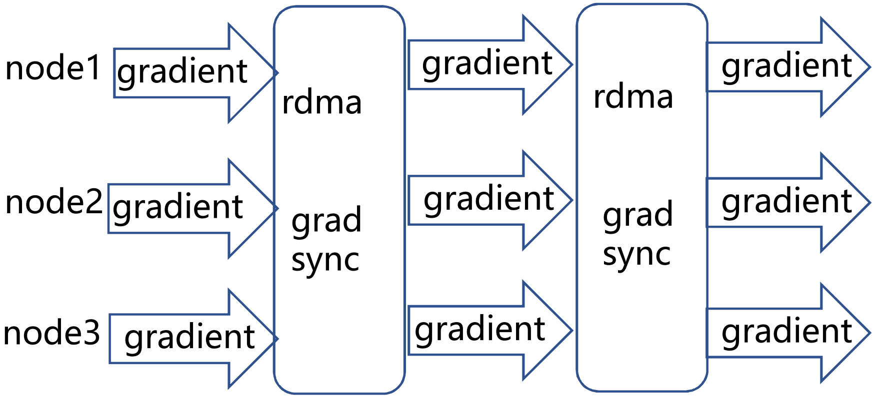
	Infiniband	RoCEv2
End-to-end delay	2us	5us
Flow Control Mechanism	Credit-based flow control mechanism	PFC/ECN，DCQN
Forwarding Mode	Forwarding based on Local ID	IP-based Forwarding
Load Balancing Mode	Packet-by-Packet Adaptive Routing	ECMP Routing
Recovery	Self-Healing Interconnect Enhancement for Intelligent Datacenters	Route Convergence
Network Configuration	Zero configuration through UFM	Manual Configuration

ref: https://www.naddod.com/blog/infiniband-vs-roce-v2-which-is-best-network-architecture-for-ai-computing-center

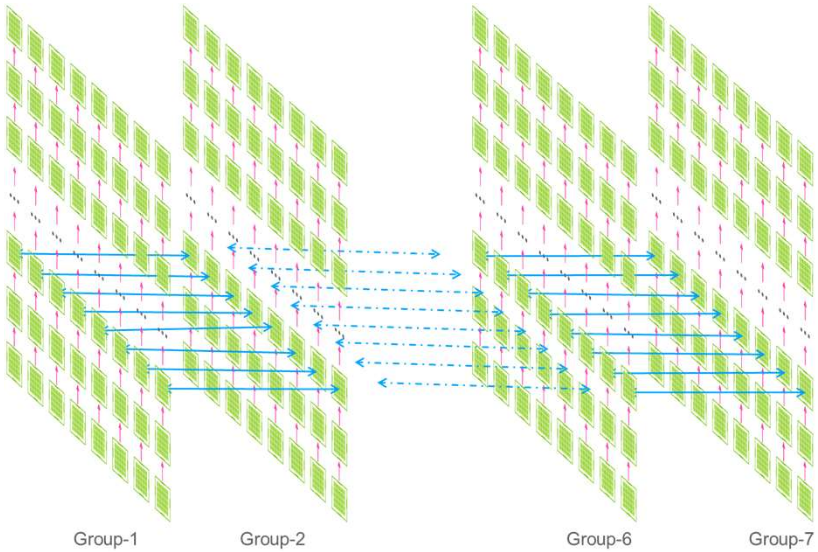
Infiniband VS RoCEv2

- Infiniband, not compatible with existing Ethernet devices, require all Infiniband devices
- RoCE, compatible with existing Ethernet devices. RoCEv2 is implemented based on udp
- iWARP, based on tcp, need many memory resources , implemented based on tcp

RDMA network requirements



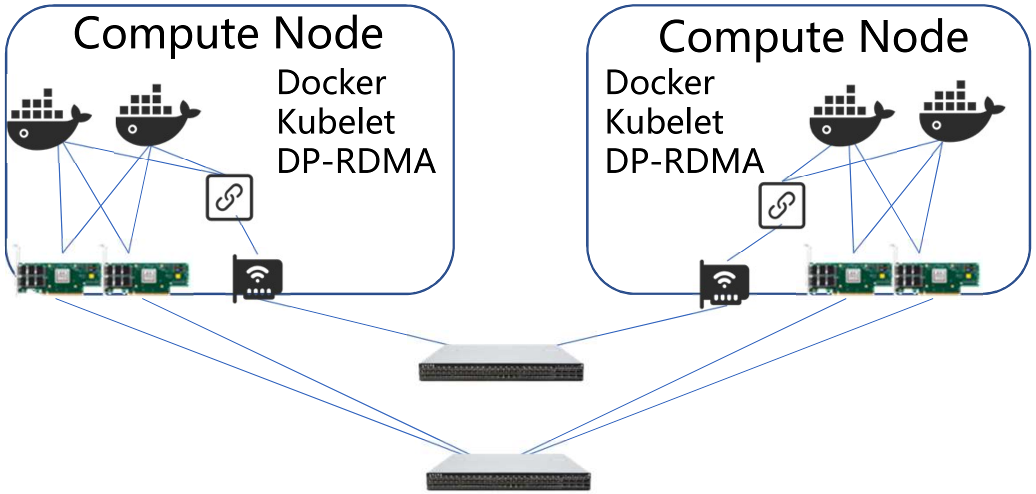
gradient sync based on rdma



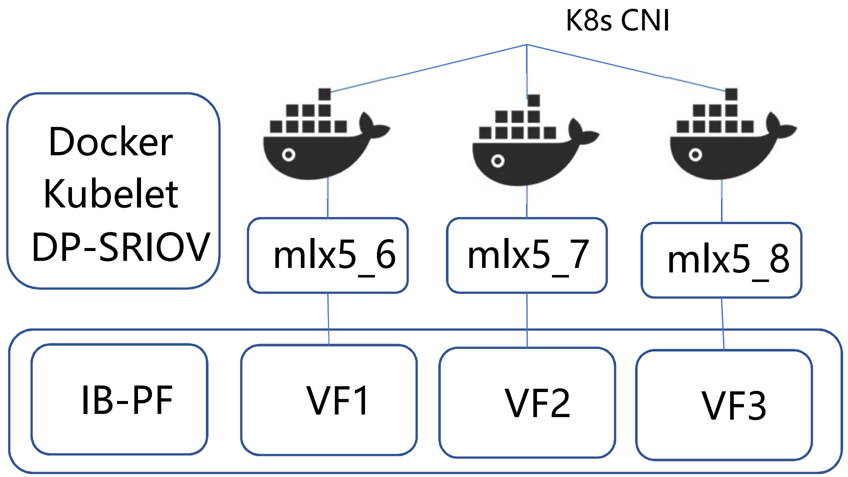
3D parallel computing based on rdma

- data parallelism: network between nodes should have high bandwidth and low latency
- model parallelism and pipeline parallelism require rdma network
- gpt-3 with 128 A100 nodes
  - pipeline parallelism bandwidth between nodes is 12GB/s, with 0.1GB data per communication,0.16s each time
  - data parallelism bandwidth between nodes is 27.4 GB/s, with 44GB data per communication,32s each time

AI cluster with IB



Solution: mount IB card to pod

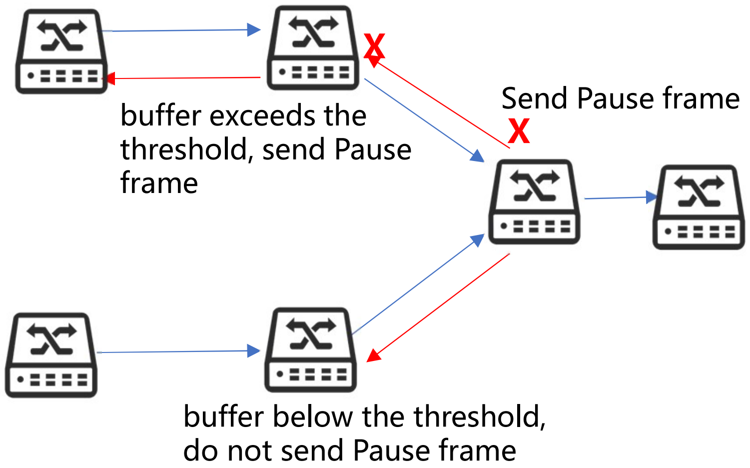


Solution: sriov

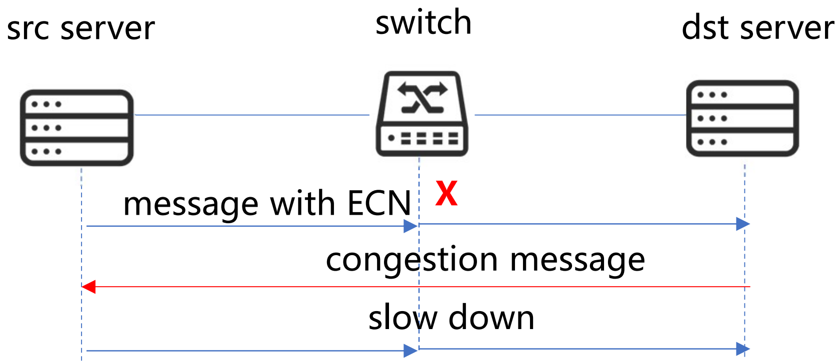
- the metadata exchange problem during RDMA communication
- IB communication between nodes is based on OpenSM,LID,UFM
- clusters with over 10000 nodes

ref: https://github.com/Mellanox/k8s-rdma-shared-dev-plugin.git
https://docs.nvidia.com/networking/pages/releaseview.action?pageId=18481842

PFC+ECN



Priority-based Flow Control



Explicit Congestion Notification

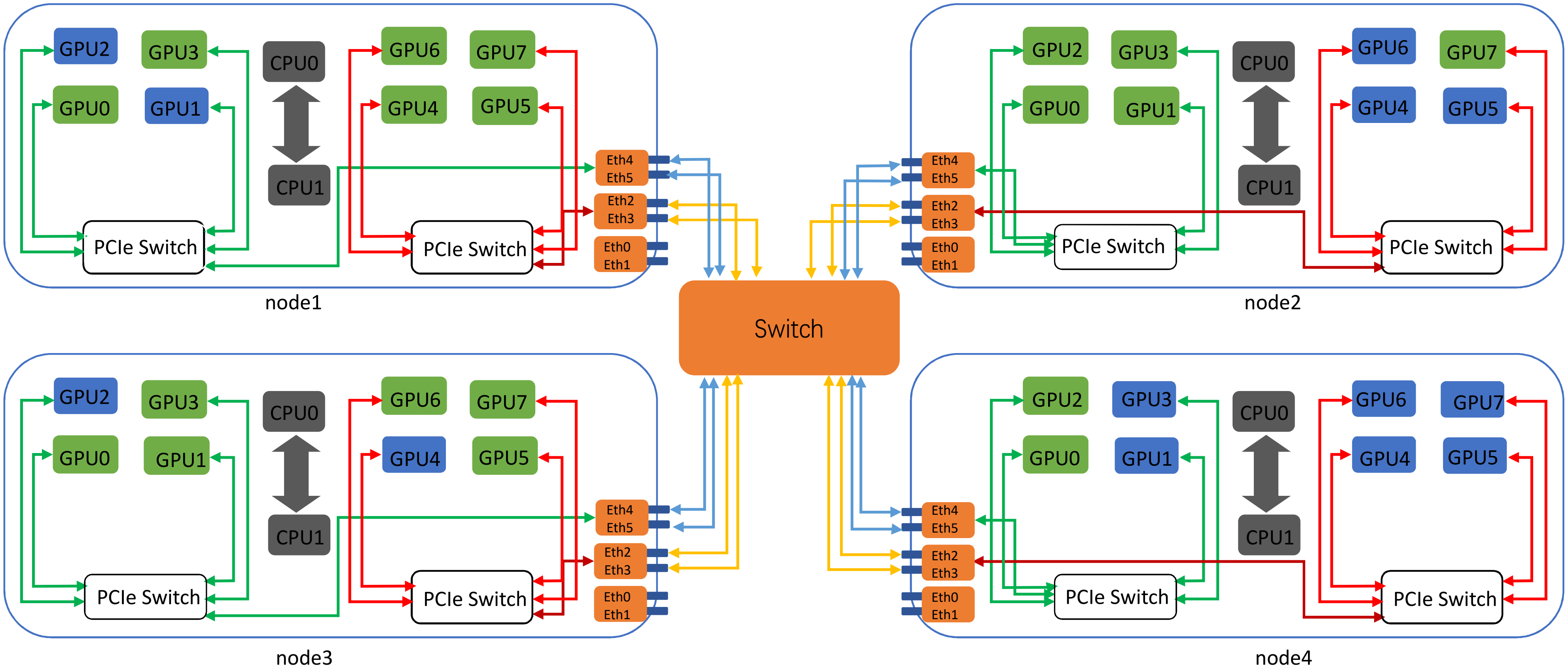
1.Switch Configuration

- PFC: data link layer, based on the packet priority and queue priority
- ECN: network layer, based on the identification bit in the data packet header

2. Host Configuration

- Linux、OFED Driver

GPU fragmentation



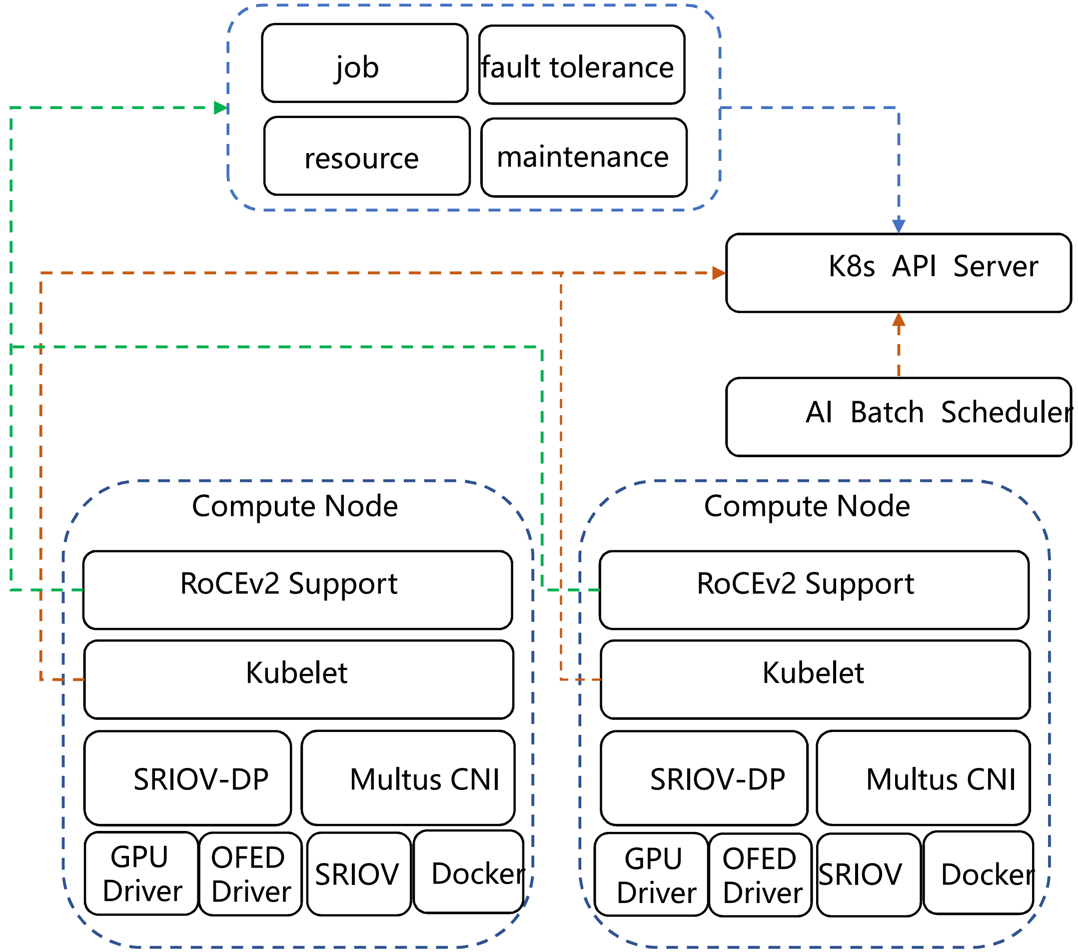
- cluster GPU resources are fragmented , idle GPU cards are disordered
- GPU fragmentation affects the used RoCE network card in multi-node training tasks

Agenda

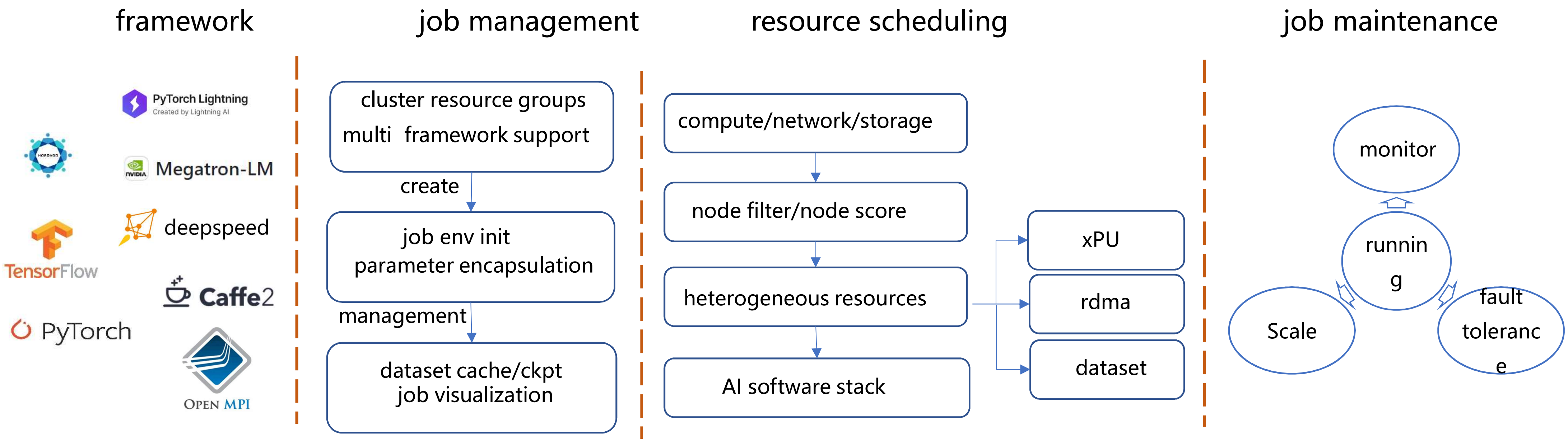
1. Background and Challenges
2. RoCEv2 solution
3. Practice and Tests

Software architecture for RoCEv2

- resource management
    - allocation and configuration for RoCE resources
    - PF/VF network traffic monitor, alarm, job fault tolerance
    - resource scheduling of computing nodes with different network types
- network management
    - business network based on Calico, multiple VF as the computing network
    - cross subnet management , route management
- components
    - sriov-dp and multus-cni , support multiple VF,
    - represent multiple VF network cards with only one K8s resources

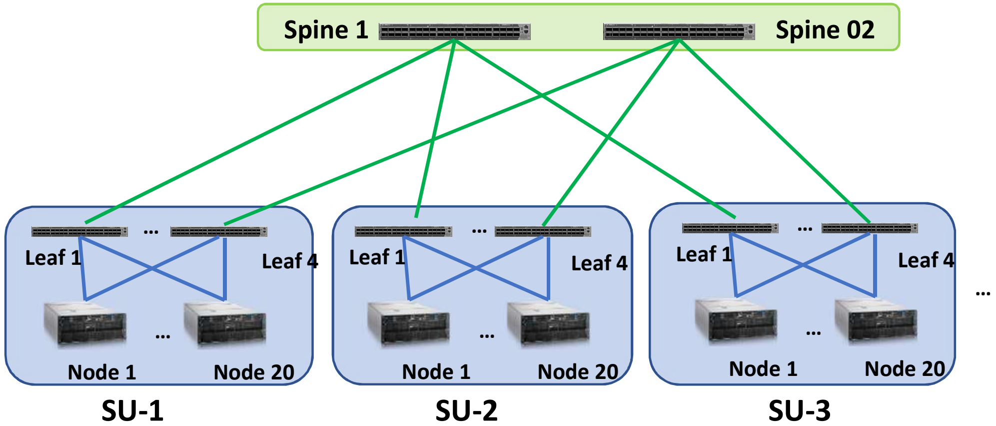


AI job management

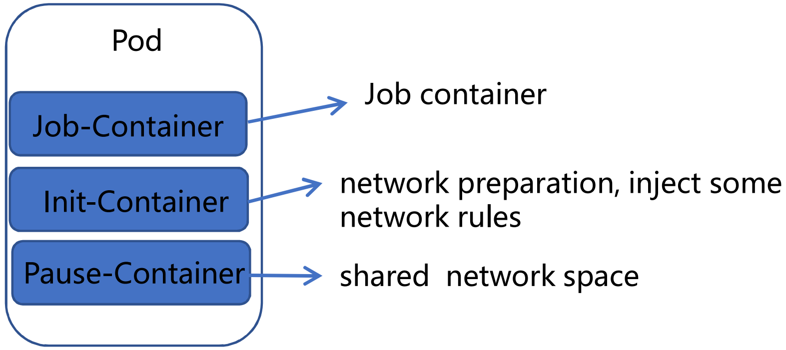


training task management on AI training platform

Physical architecture for RoCEv2



physical network topo



network preparation

- spine-leaf network, horizontal scaling support for spine switch and leaf switch
- different RoCE cards with different vlan
- configure subnet information at switch, VF using physical subnet and routing

Network simulation in container

container1
eth0:10.233.96.32
net1:192.168.10.2(roce)
net2:192.168.20.2(roce)

shell:
ping 192.168.10.254
ping 192.168.10.254

arp broadcast

Protocol	Address	Age	Hardware Addr	Interface
Internet	190.11.17.14	-	a302.98df.11b6	vlan20
Internet	190.11.27.30	-	a302.98df.23dk	eth-0-1
Internet	192.168.10.2	0	5678.5678.5678	vlan20

leaf1:show ip arp

route learning

B 192.168.10.2/32 [20/0] via 190.11.17.1,eth-0-1,00:00:24
B 192.168.30.2/32 [20/0] via 190.11.37.1,eth-0-2,00:00:24

spine: show ip route

container2
eth0:10.233.84.32
net1:192.168.30.2(roce)
net2:192.168.40.2(roce)

shell:
ping 192.168.30.254
ping 192.168.40.254

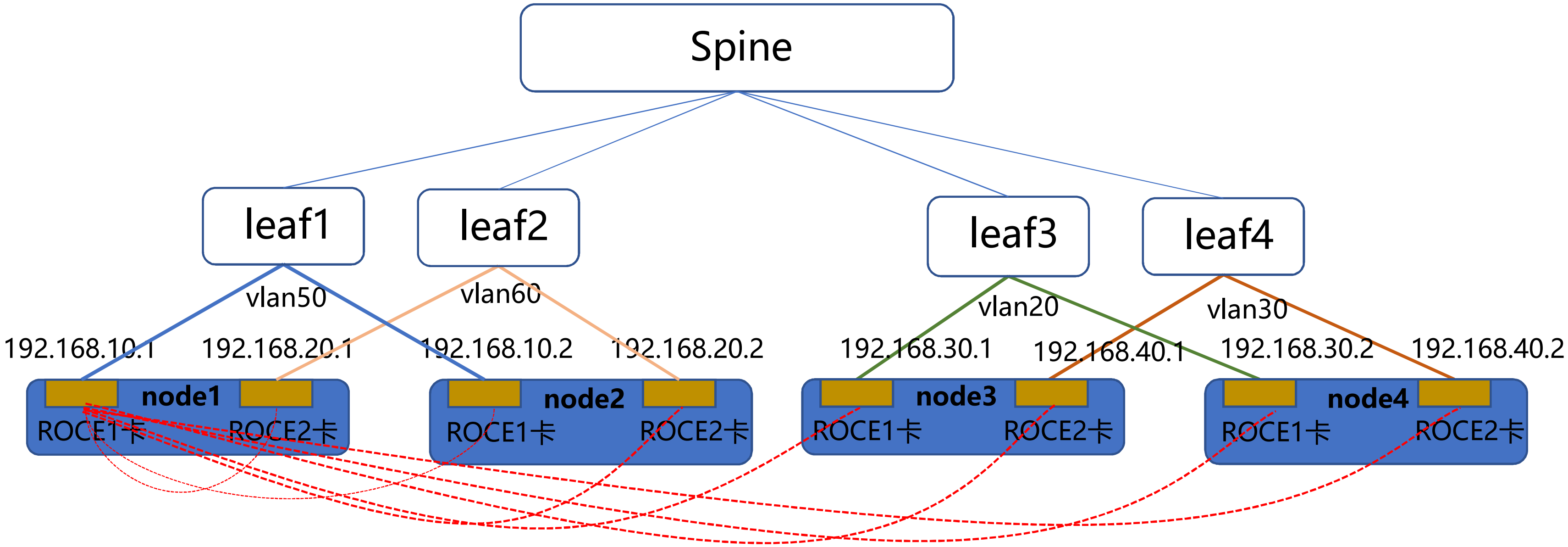
arp broadcast

Protocol	Address	Age	Hardware Addr	Interface
Internet	190.11.37.53	-	a301.98df.1ab5	vlan20
Internet	190.11.47.54	-	a301.98df.2440	eth-0-1
Internet	192.168.30.2	0	1234.1234.1234	vlan30

leaf2:show ip arp

- container starts quickly, switches can not update the arp table before training tasks start
- ip reused by another container, but the arp table of the switch was not refreshed immediately
- adjust the aging time of arp table
- adjust the arp table capacity of switch

RoCE nic with P2P communication



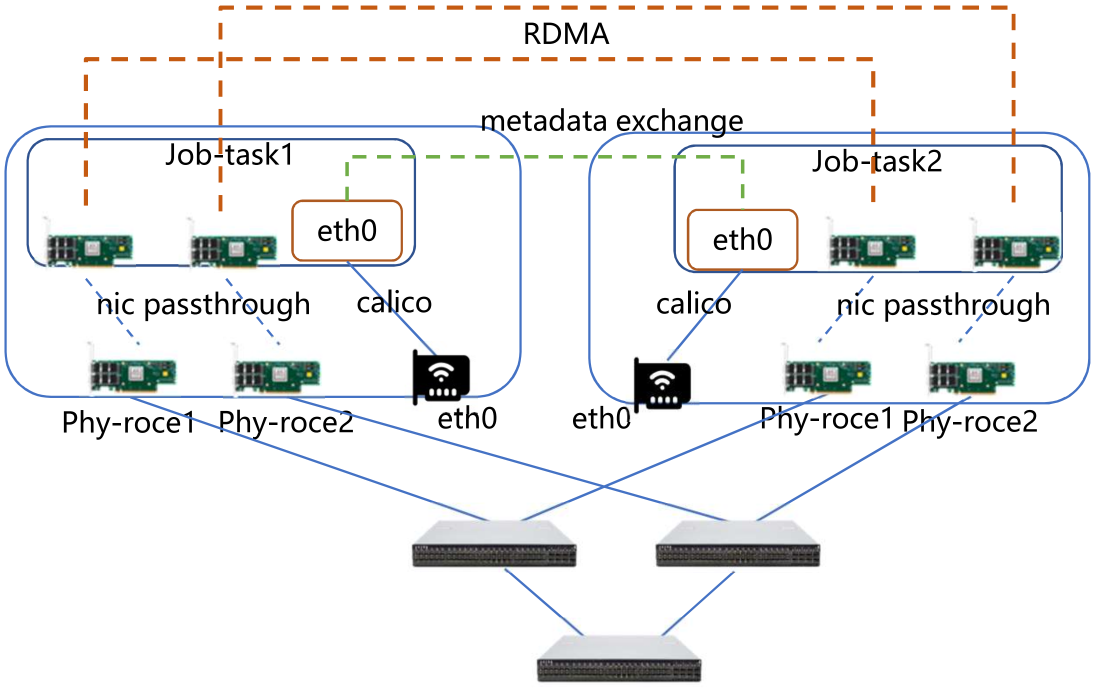
the VF of the RoCE1 network card at node1 can communicate with any VF of any node in the cluster

Other considerations

- GPU p2p exception when sriov virtualization enabled
- network traffic sharing problem between multiple vf
- roce gid index problem when using macvlan
- all vf and pf of the node can be recognized in container
- the maximum number of VFs for RoCE network cards

RoCEv2 for large model training

- large model training scenarios, all GPU of the node will be used by one pod
- calico network for metadata exchange , using physical RoCE nic in pod
- multus cni,sriov-dp supprot RoCE PF
- large model training jobs use the characteristics of nccl, such as pxn

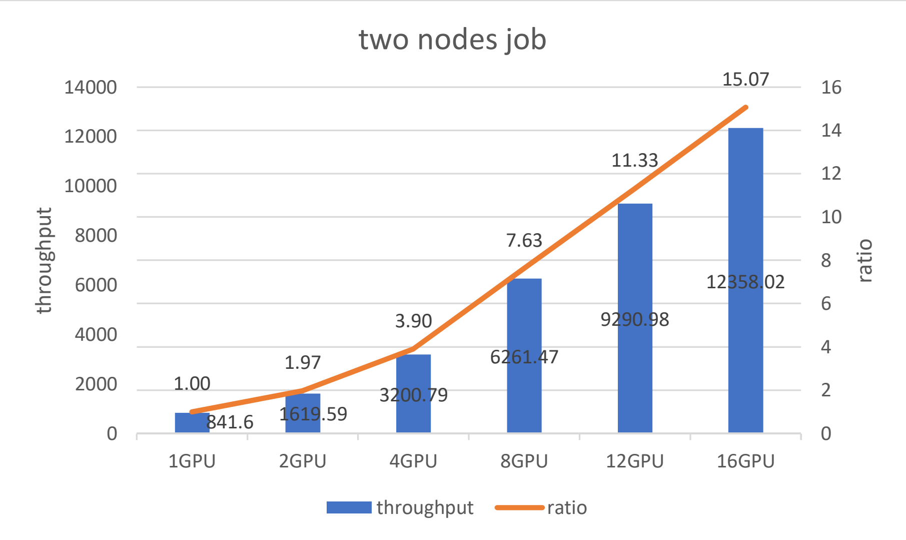


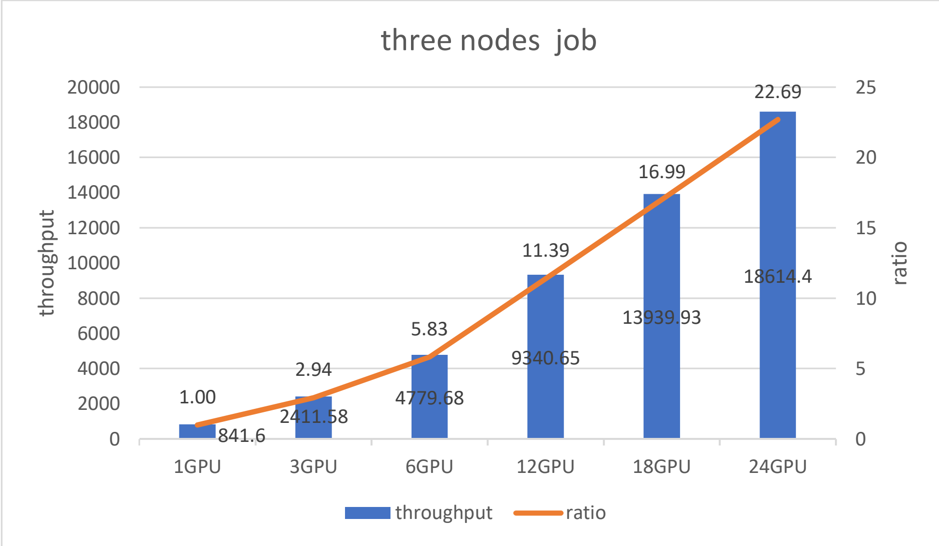
pod with calico and physical RoCE nic

Agenda

1. Background and Challenges
2. RoCEv2 solution
3. Practice and Tests

Job tests





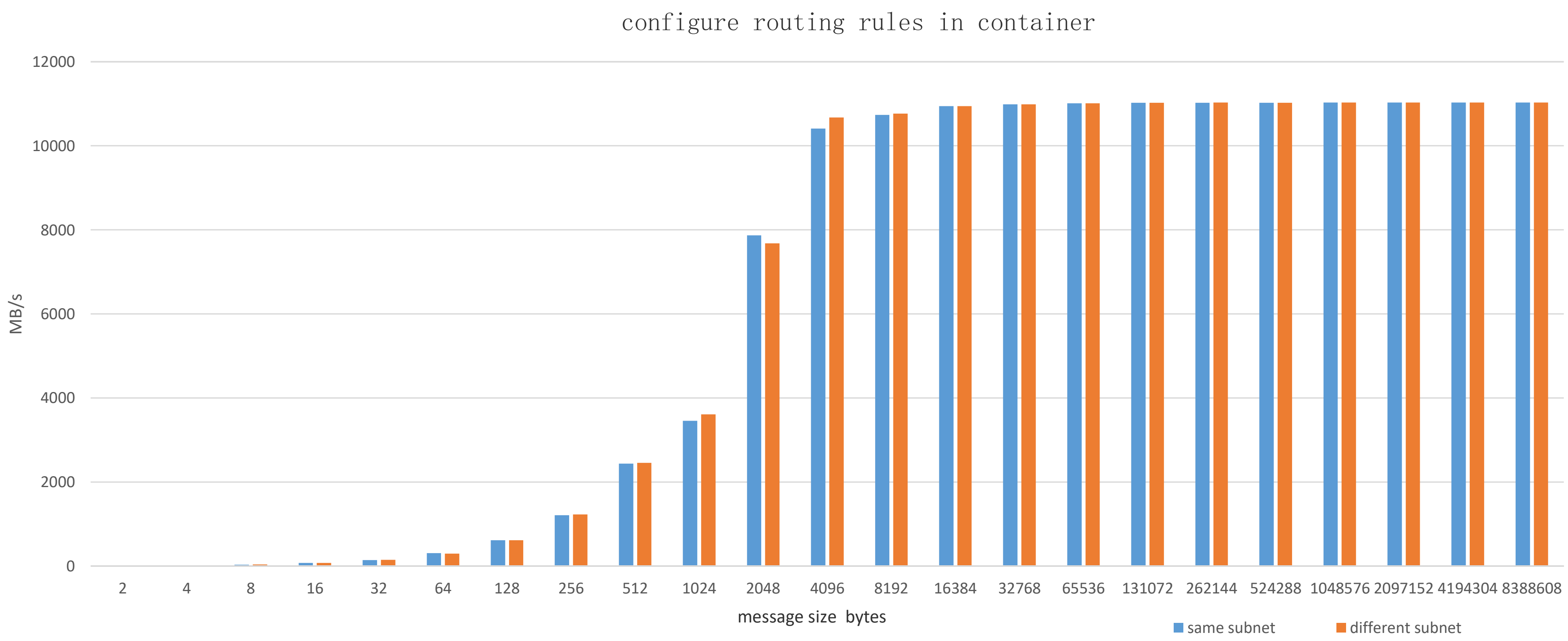
GPU server info：

- NF5468M5
- CPU：Intel(R) Xeon(R) Gold 6230R CPU @ 2.10GHz
- GPU：A100-PCIE-40GB
- IB：Mellanox Technologies MT27800 Family 100Gb
- GPU driver：450.102.04
- IB Driver：5.4-1.0.3.0

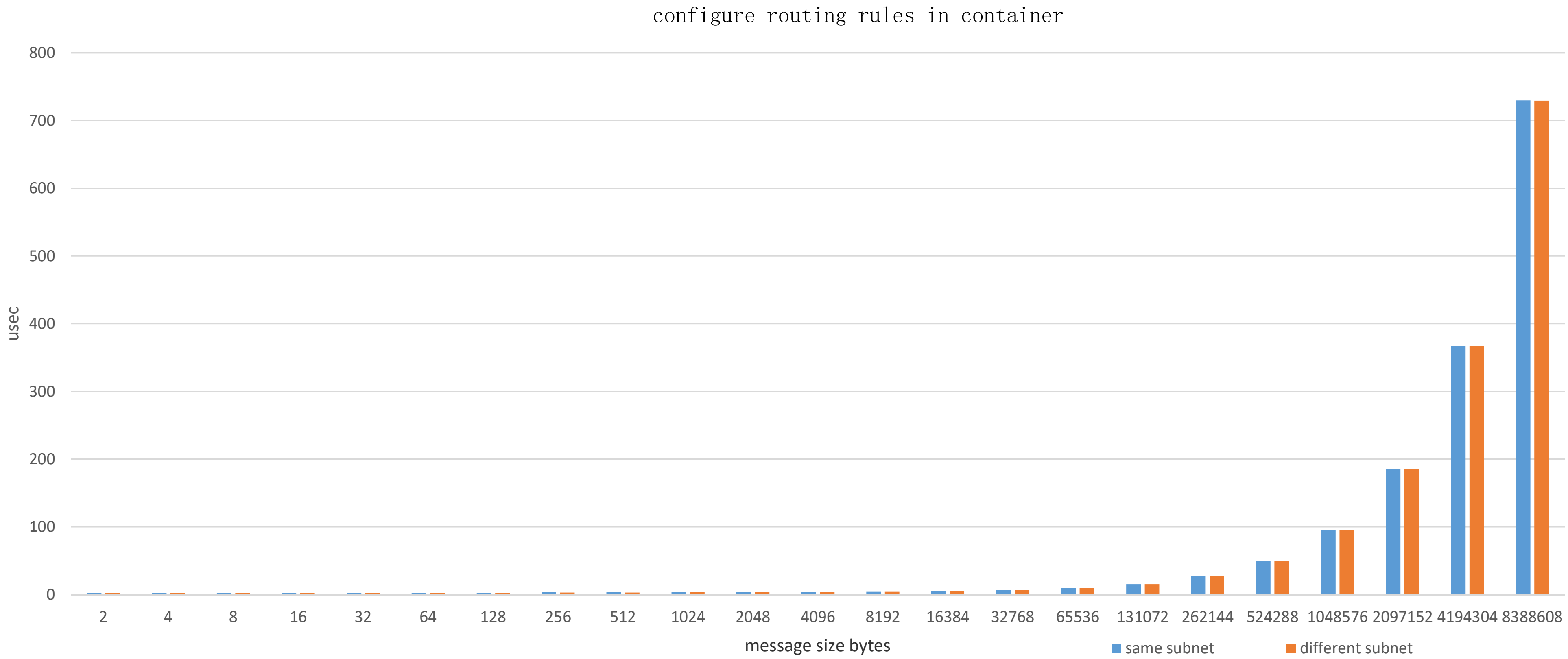
software info：

- CUDA：11.0
- NCCL：2.12.6
- Tensorflow：1.15.3+nv
- Tensorflow-cnn-benchmark,imagenet(synthetic),resnet50,bs=256,iter=500

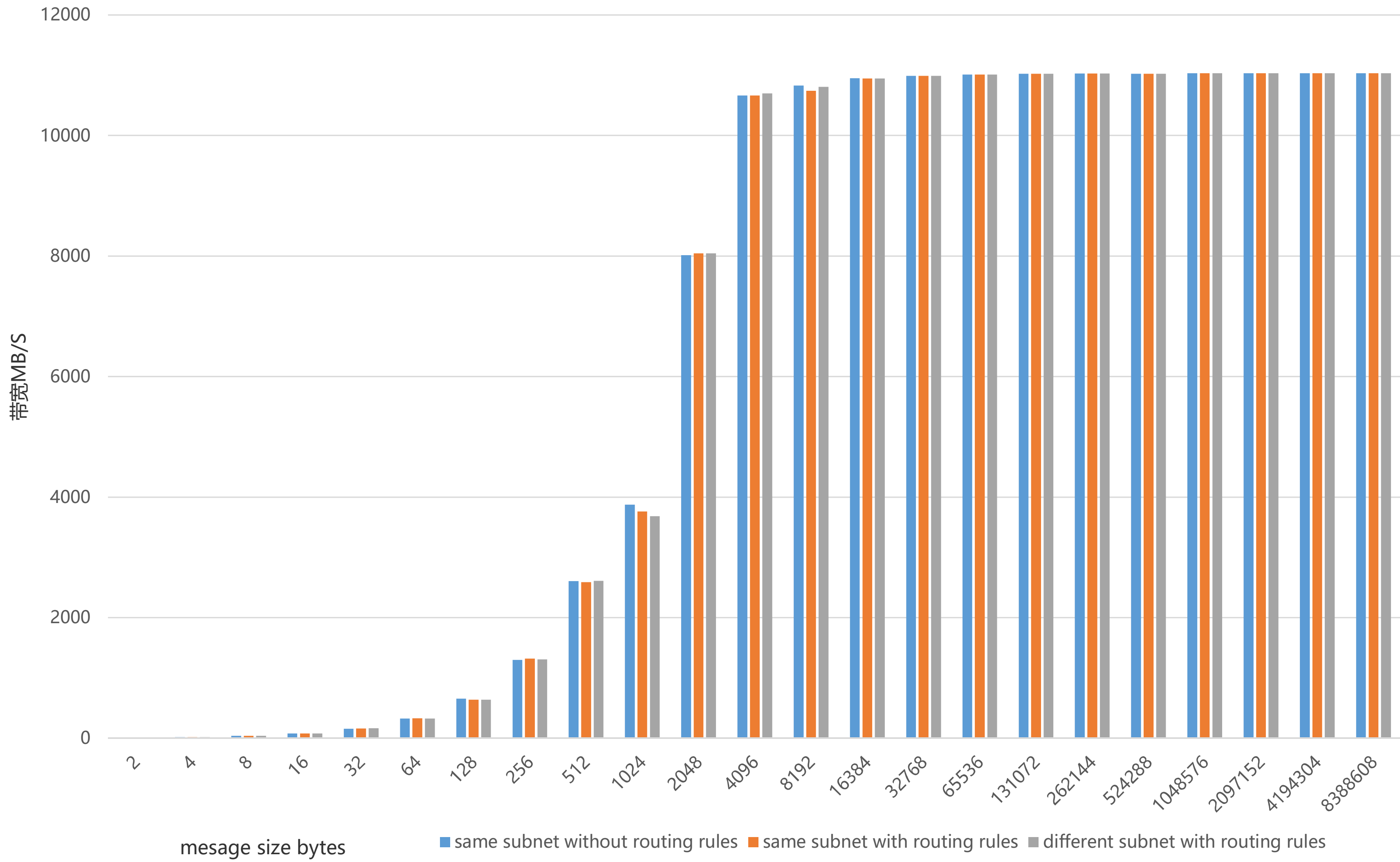
Bandwidth test between containers



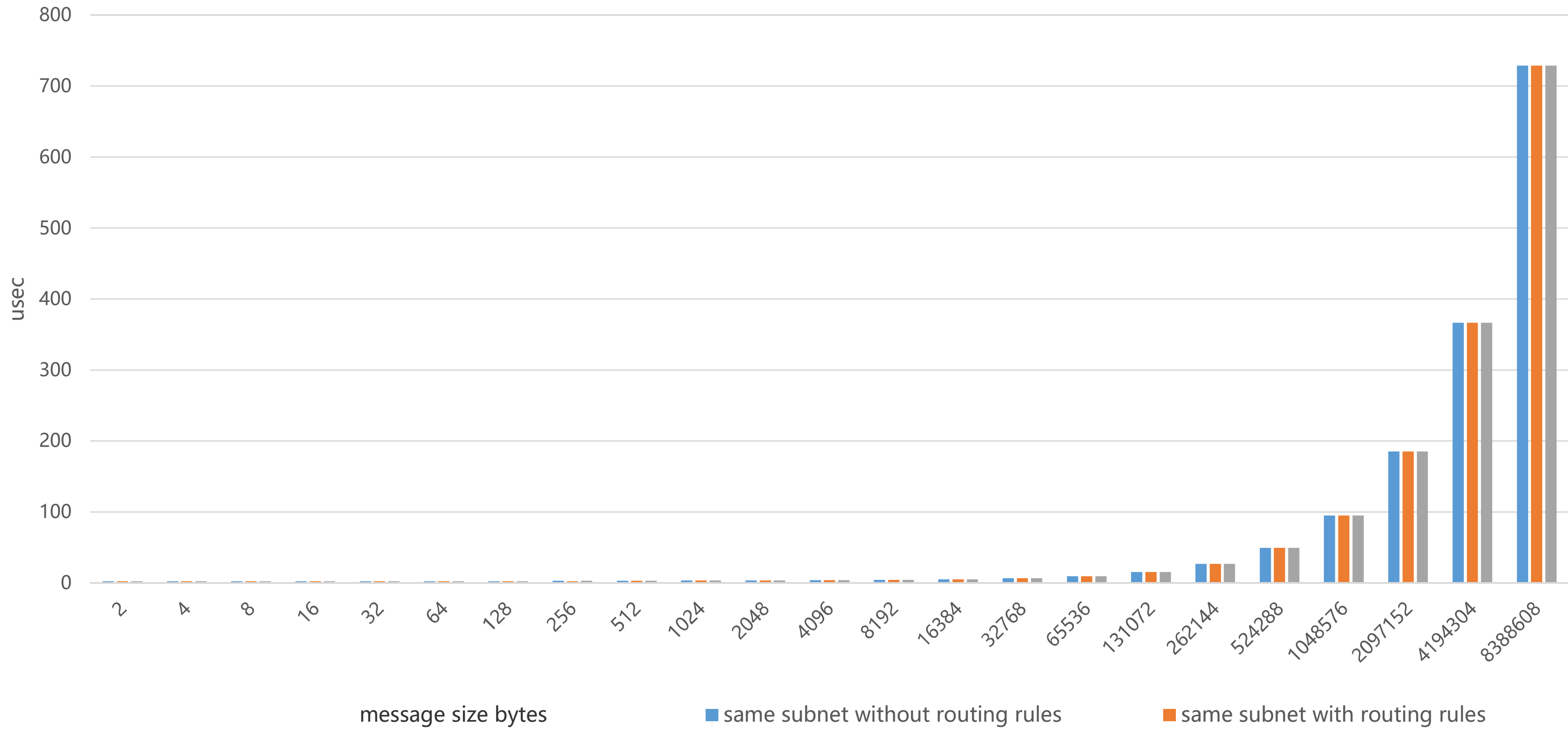
Latency test between containers



Bandwidth test between hosts



atency test between hosts



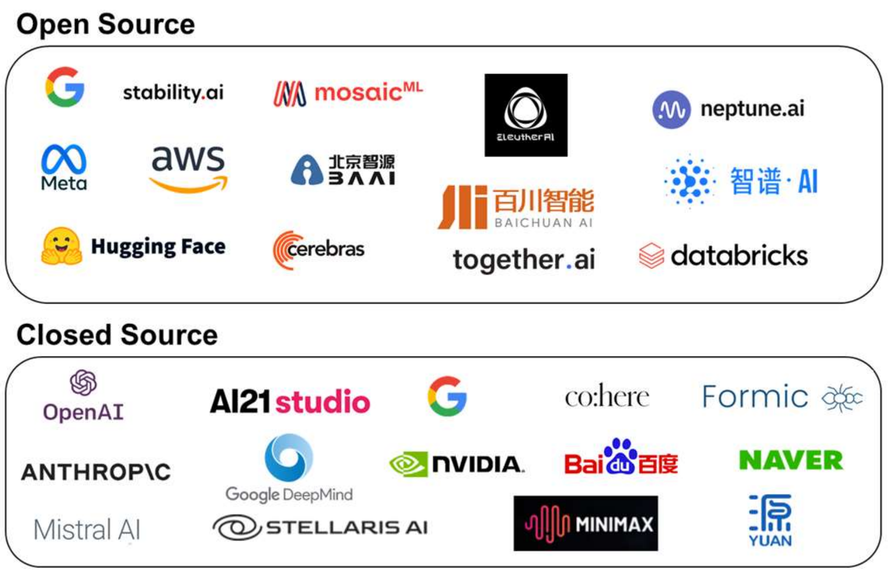
KubeCon | CloudNativeCon
OPEN SOURCE SUMMIT
China 2023

AIStation Platform

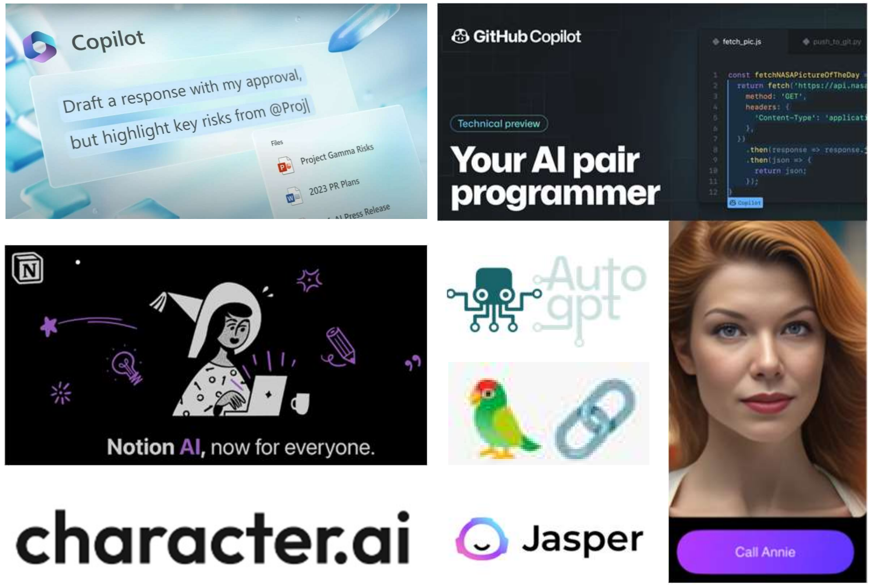
Chao WANG, IEI

Gen.AI Trend

LLM, Multimodal, ...



Gen.AI Applications



LLM Training Challenges

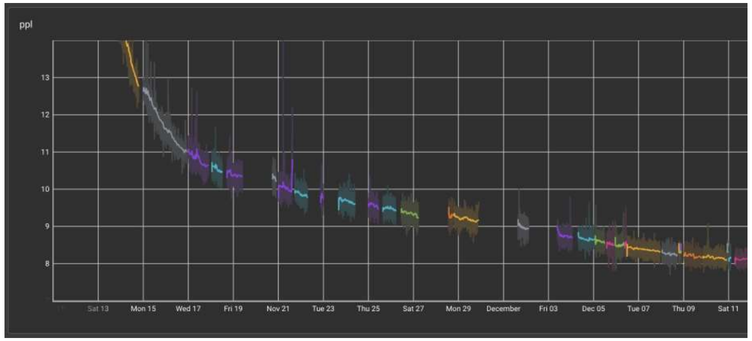
LLM Training

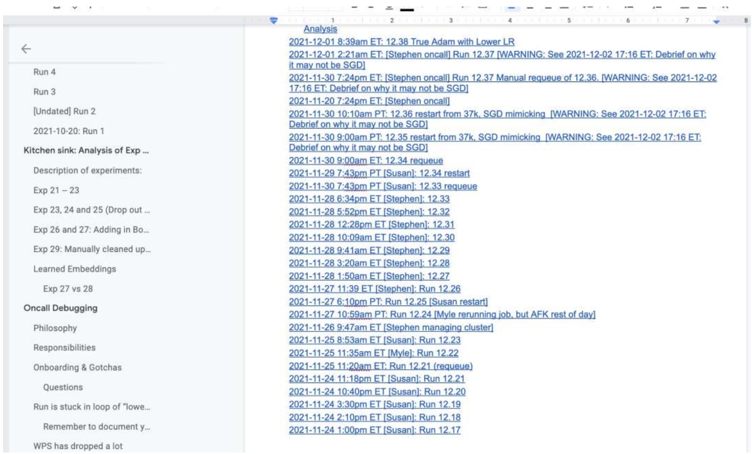
- 1000+ GPU Cluster
- PB data collecting, cleaning, etc.
- Training optimization
- GPU malfunction
- Unstable loss ……

Issues from customers

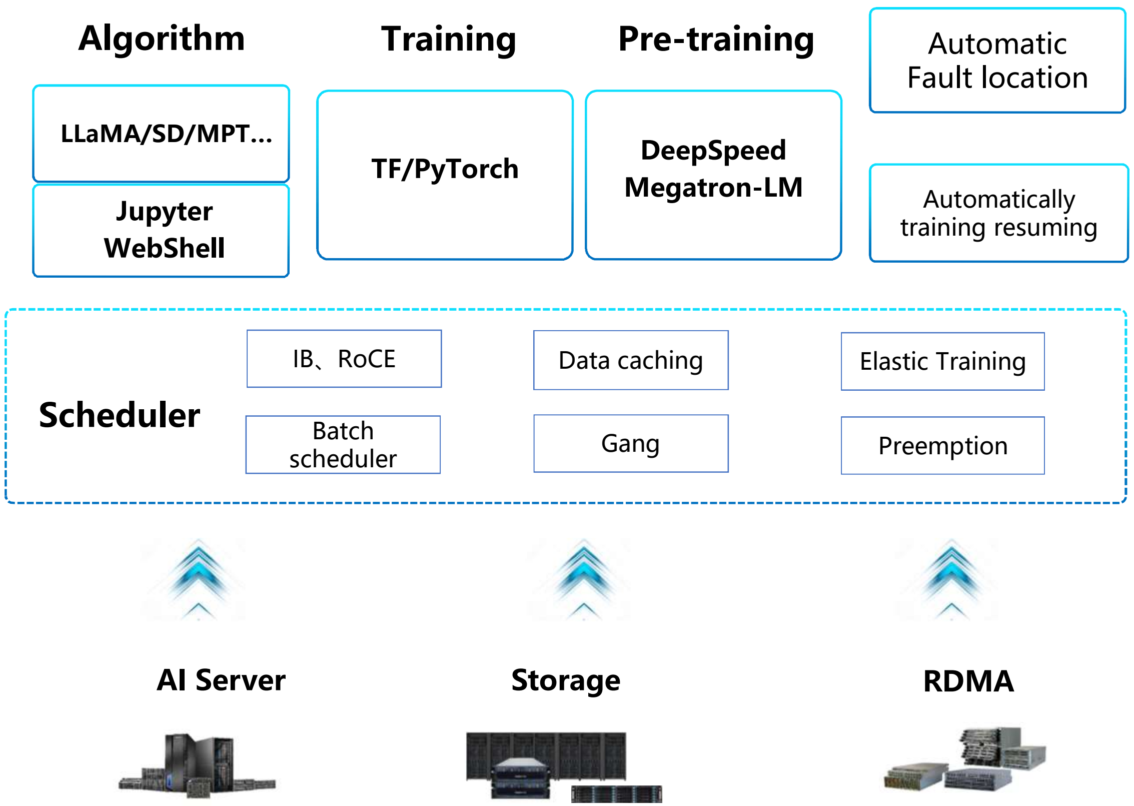
- CUDA initialization failure
- Poor NCCL performance
- GPU direct RDMA malfunction
- RoCE network
- Distributed training task
- GPU cluster performance optimization
- ……

- Meta OPT-175B top three record long runs of the experiment these past two weeks, lasting 1.5, 2.8, and 2 days each.





Distributed Training Adoption



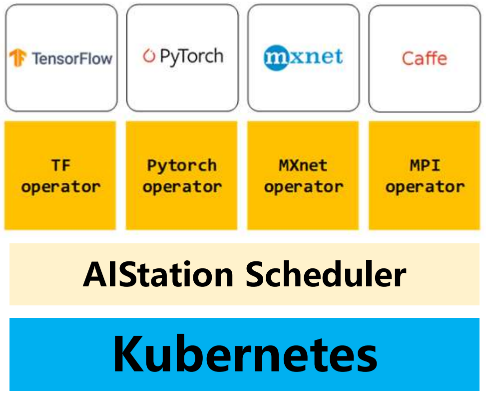
Fast training abnormal locating, automatic training resuming

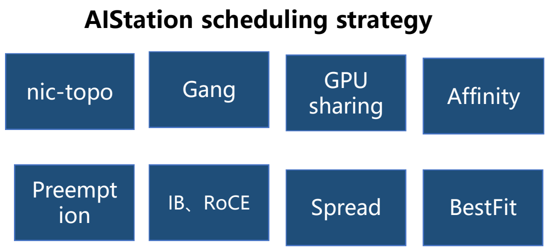
- ✓ Fast chip, network abnormal locating and fault pause process to hold global training.
- ✓ Calculating standby node and automatic elastic replacement.
- ✓ Health node CheckPoint reading and automatic training resuming.

Simplify network adaptation, flexible and efficient

- ✓ Compatible with IB, RoCE and other complex cluster networking environment.
- ✓ Flexible resource matching for large-scale training scenario.
- ✓ Automatic fault tolerance to ensure long-term model training efficient and stable.

Distributed Training Optimization





Tensorflow	Pytorch	Mxnet	Paddle
ParameterServer	Distributed Data Parallel Training (DDP)	Data Parallel (Server-worker-scheduler)	ParameterServer
Mirrored			
MultiWorker Mirrored	Collective Communication		Collective
CentralStorage			
MPI	MPI	MPI	MPI

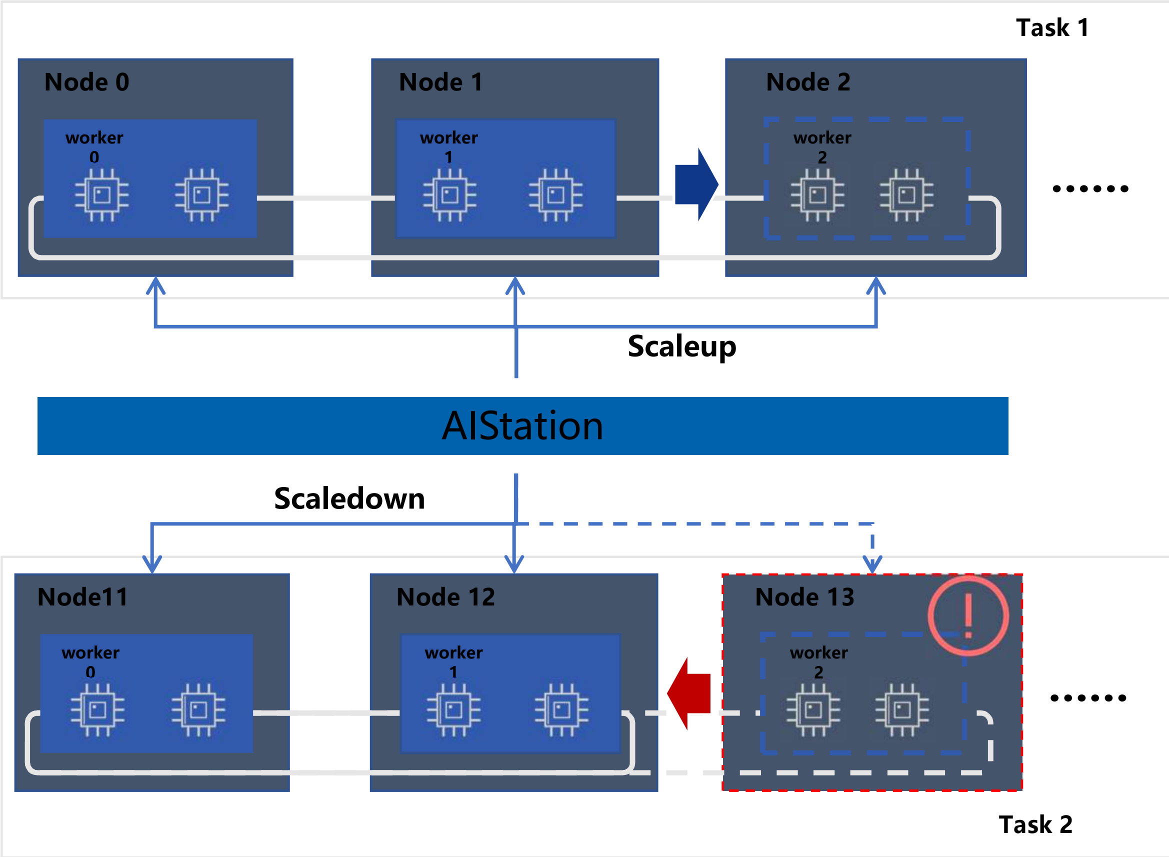
Convenient way

- ✓ Simple configuration, one-click launch distributed job.
- ✓ Large model training scenarios, quick start and support Megatron-LM, DeepSpeed, etc.

Professional optimization

- ✓ Operator optimization, support tensorflow, pytorch, mxnet, caffe, paddle native distributed training and MPI mode.
- ✓ Optimized distributed scheduling strategy to achieve a fast computing resources allocation, and automatic distributed training process launch.

Elastic and Dynamic Resource Usage



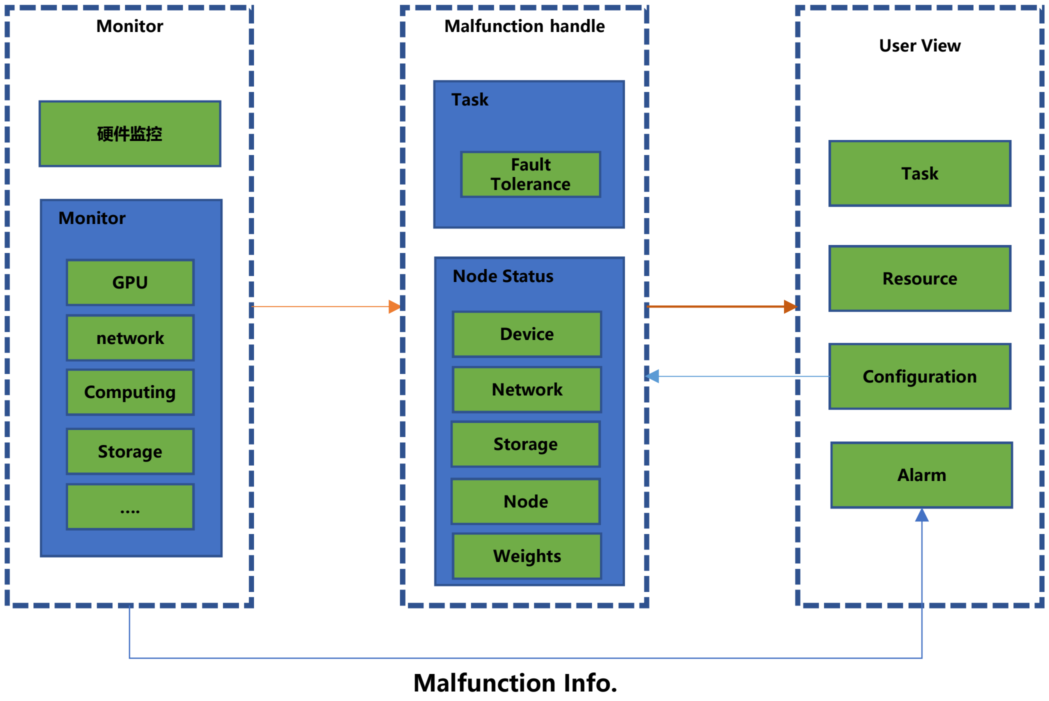
Elastic training

- ✓ Elastic training mode, dynamic resource allocation on demand.
- ✓ Maximizing the use of GPUs to achieve a high utilization.

Comprehensive guarantee for large-scale training

- ✓ Simplify the resources evaluation strategy, dynamic adjustment training resources.
- ✓ Timeliness and reliability of the huge scale of training.
- ✓ Training anomaly recovery with limited resources, automatic fault awareness and self-healing elastic resource usage.

Automatic Fault Tolerant Processing



Basic function

- When training mission abort, such as worker exit, master exit.
- Fault tolerance: GPU node displacement process, task to restart.

For elastic training task

- Master malfunction: resubmit training task
- Worker malfunction： handle by framework
- NIC fault tolerance: handle by framework
- GPU malfunction： replace the abnormal mode and restart the task

KubeCon CloudNativeCon
OPEN SOURCE SUMMIT
China 2023

Thanks