


KubeCon | CloudNativeCon

North America 2021

RESILIENCE

REALIZED

Large-Scale Practice of Persistent Memory in Alibaba Cloud

Junbao Kan (junbao.kjb@alibaba-inc.com)
Qingcan Wang (qingcan.wqc@alibaba-inc.com)

Who are we?

Junbao Kan
Senior engineer of
Alibaba Cloud
Github: fredkan

Qingcan Wang
Senior engineer of
Alibaba Cloud
Github: Denkensk

Agenda

- **1. Cloud Native PMEM Stack Introduction**
- 2. Application Practice on PMEM Stack
- 3. Demo
- 4. Other Related Works

What is Persistence Memory?

Persistence Memory(PMEM) allow programs to access data directly byte-addressable, while the contents are non-volatile.

- Data Persistence: as a non-volatile storage;
- High Performance: faster than disk storage;
- Large Capacity: larger capacity than DRAM;
- Low Price: low price to save your cost;

PMEM Requirements In Alibaba Cloud:

- In memory database cache need huge memory space;
- Large quantities of instances cost too much;

Why PMEM in Kubernetes?

PMEM is typically used as FileSystem(dax) volume to application, which can be provided as Persistence Volume with CSI.

- Containerized:
  - More applications are containerized, and use PMEM in container.
- Automatic:
  - Manages PMEM device automatic on Kubernetes.
- Capacity Limit:
  - PMEM resource is local resource and is shared in multi-tenant scenario.
  - The capacity used for each user should be limited.
- Capacity Aware:
  - PMEM is types of local resource and should be scheduled by capacity.

PMEM Stack Architecture

Node Resource Manager:

- Define and initialize PMEM to Namespace/Device;
- Create/Format file system(Dax) on namespace;
- Record the file system capacity;

CSI Plugin:

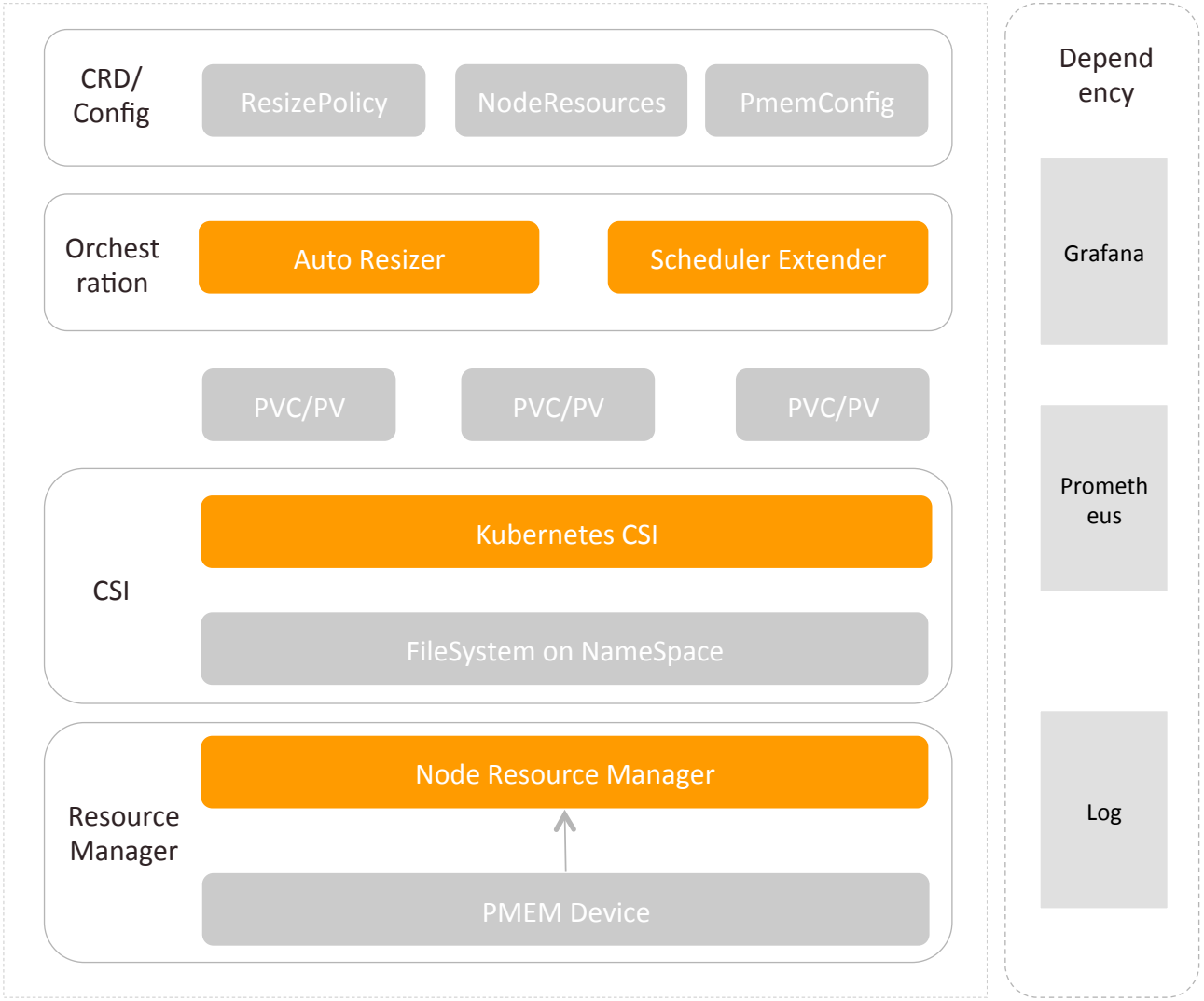
- Provision volume(PMEM) on FileSystem;
- Resize/Monitor volume for payload;

Scheduler Plugin:

- Schedule Pod with PMEM Capacity;
- Schedule Pod with PMEM and NUMA locality;

Auto Resizer：

- Automatically expand PMEM volume according to usage;



Node Resource Manager

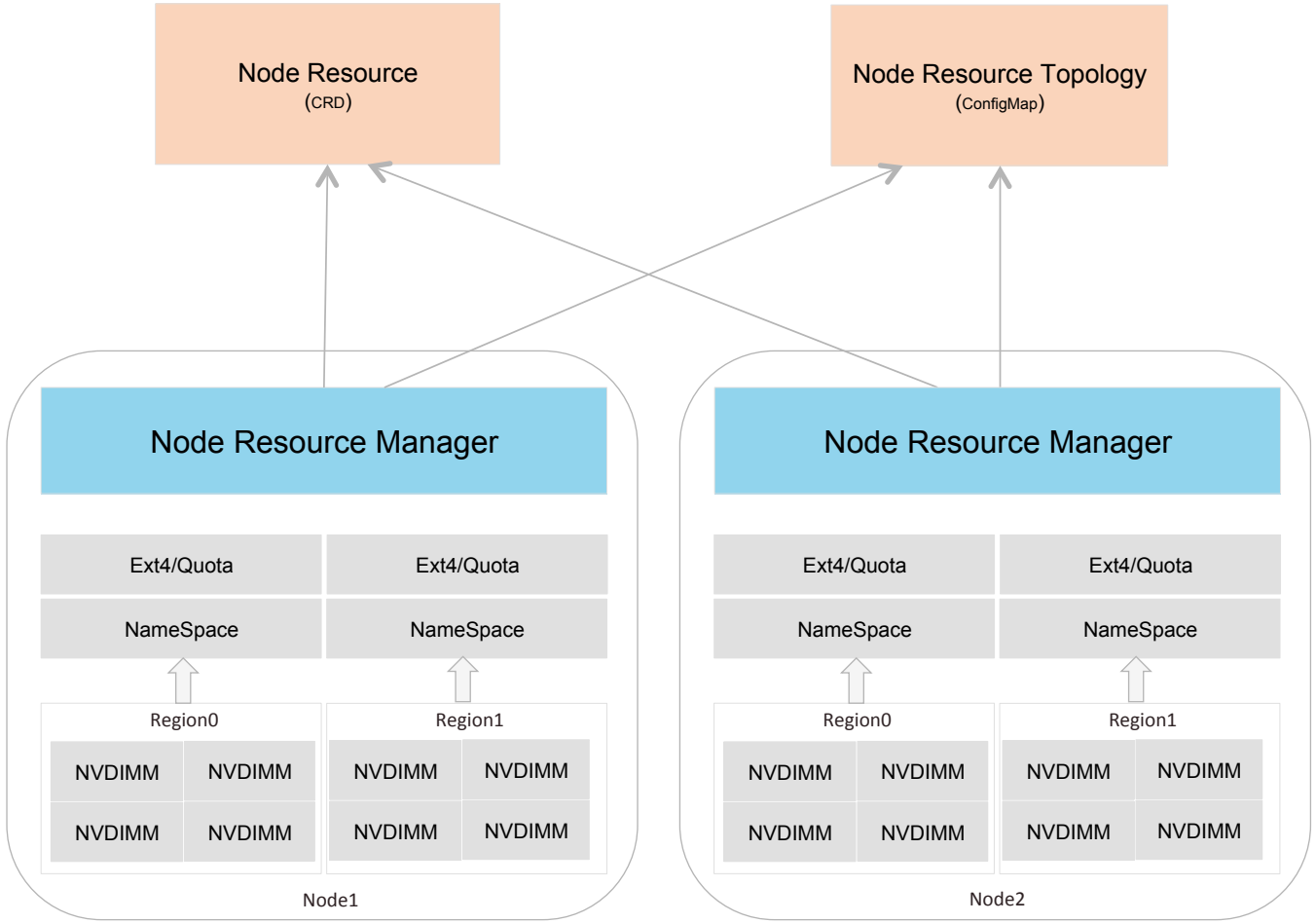
Node Resource(CRD):

- Record file system capacity and NUMA topology for each node;
- Create one CR per node;

Node Resource Topology:

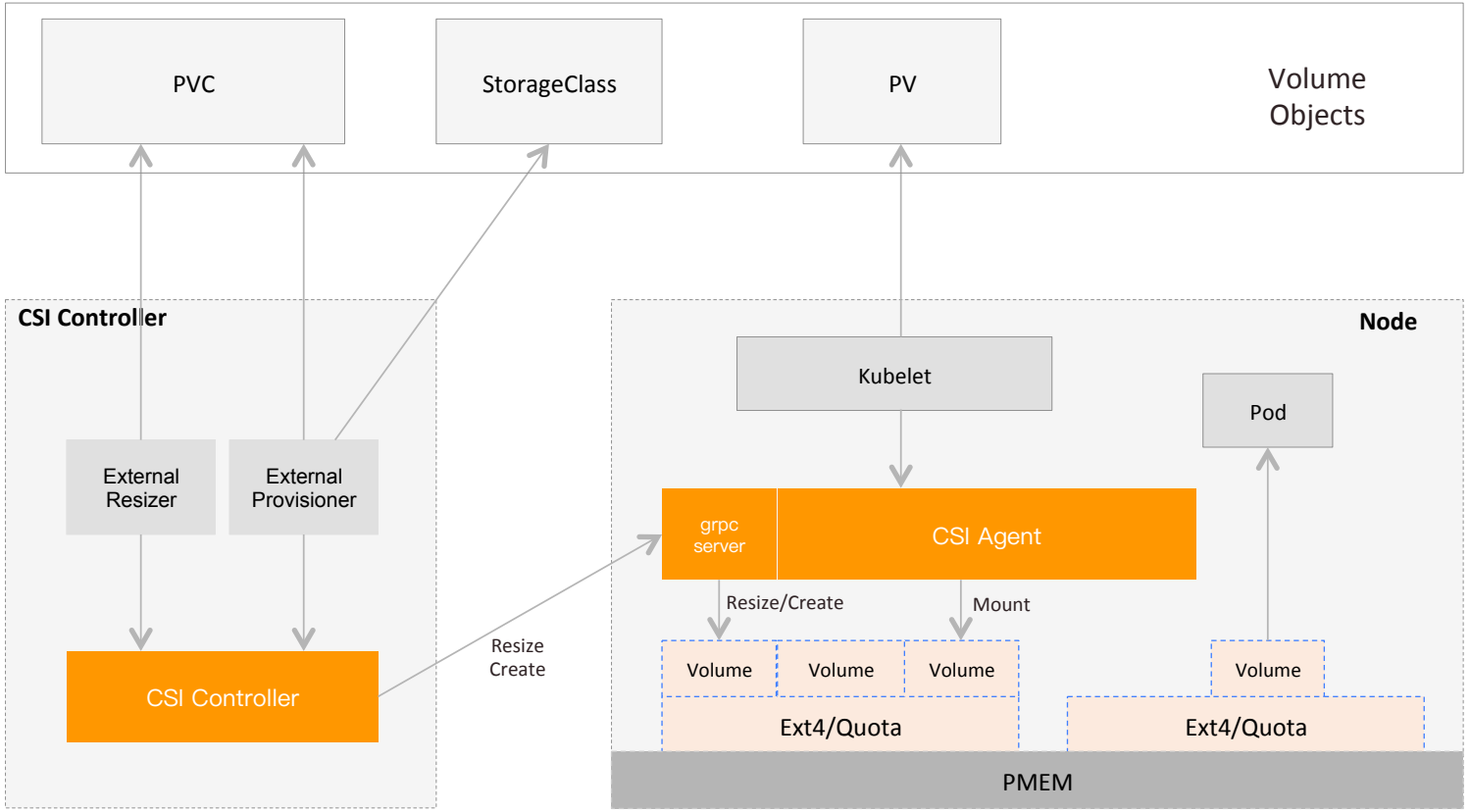
- Define the PMEM device topology for each node;
- Define the PMEM region and file system details;

```
- name: /mnt/path1
  key: kubernetes.io/hostname
  operator: In
  value: cn-beijing.192.168.3.36
  topology:
   type: pmem
   options: prjquota,shared
   fstype: ext4
   regions:
   - region0
```



https://github.com/openyurtio/node-resource-manager

PMEM CSI



Volume Manager:

- Create/Delete Volume through GRPC;
- Mount/Umount ops on node;

Automatic:

- Volume Dynamic Provision;
- Online Resize volume without app pause;

Monitor:

- Volume capacity/inode monitoring;
- PMEM capacity usage monitoring;

https://github.com/kubernetes-sigs/alibaba-cloud-csi-driver

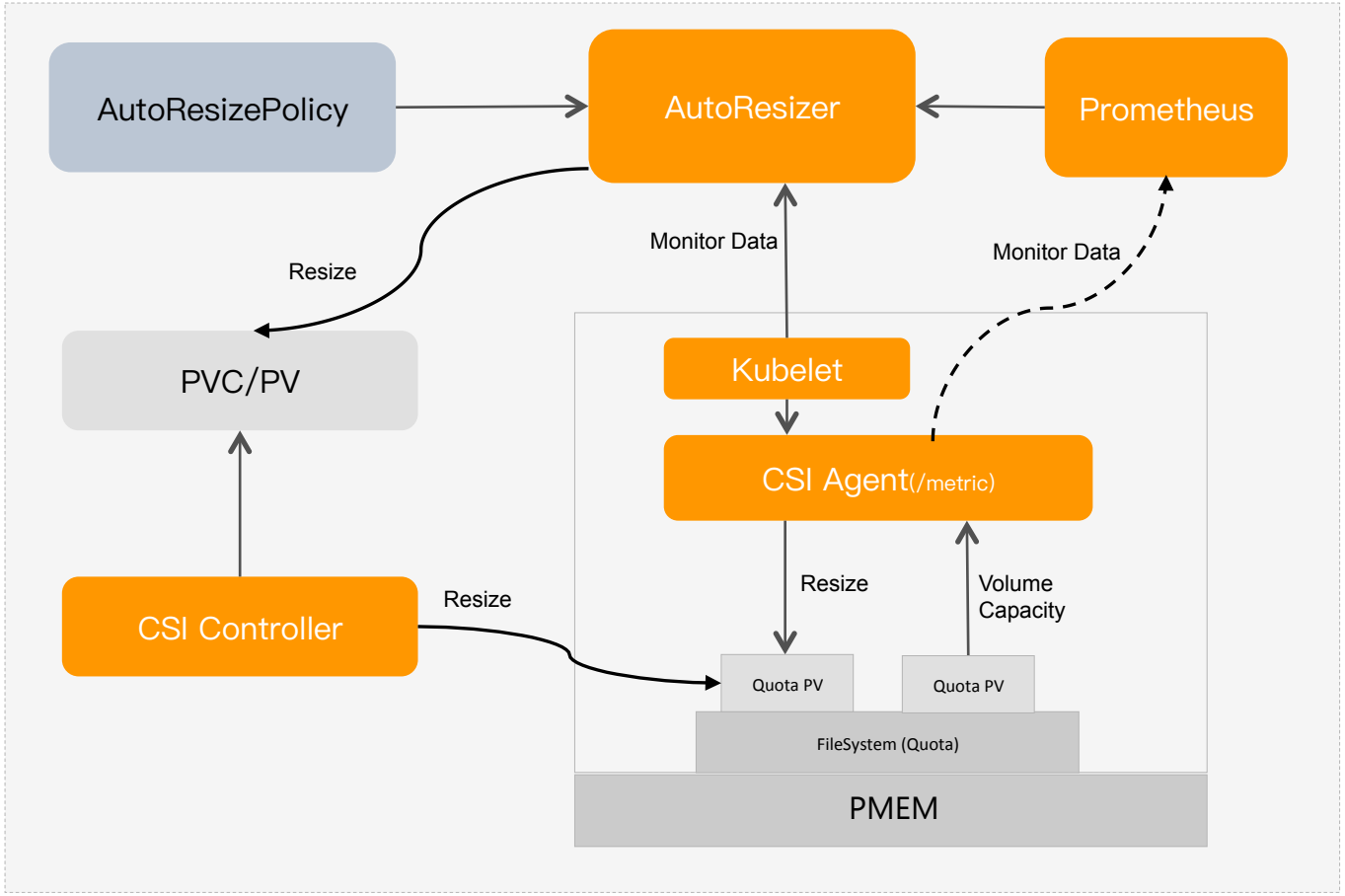
Volume Auto Resizer

AutoResizePolicy:

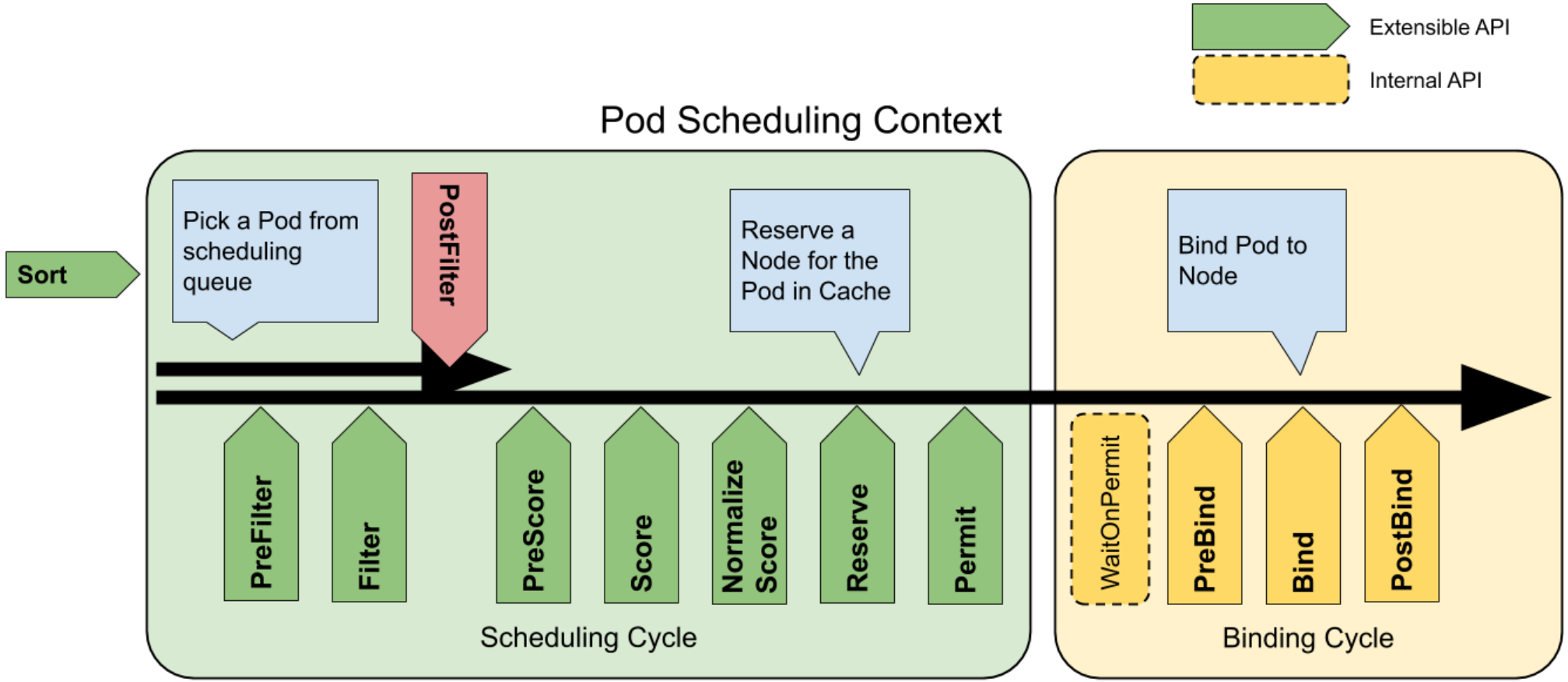
- Define resize policy for pvc;
- Trigger resize by volume percentage/size;
- Max volume size limit;

Volume Monitor:

- Monitor Data from Prometheus;
- Monitor Data from kubelet directly;



Scheduler Framework



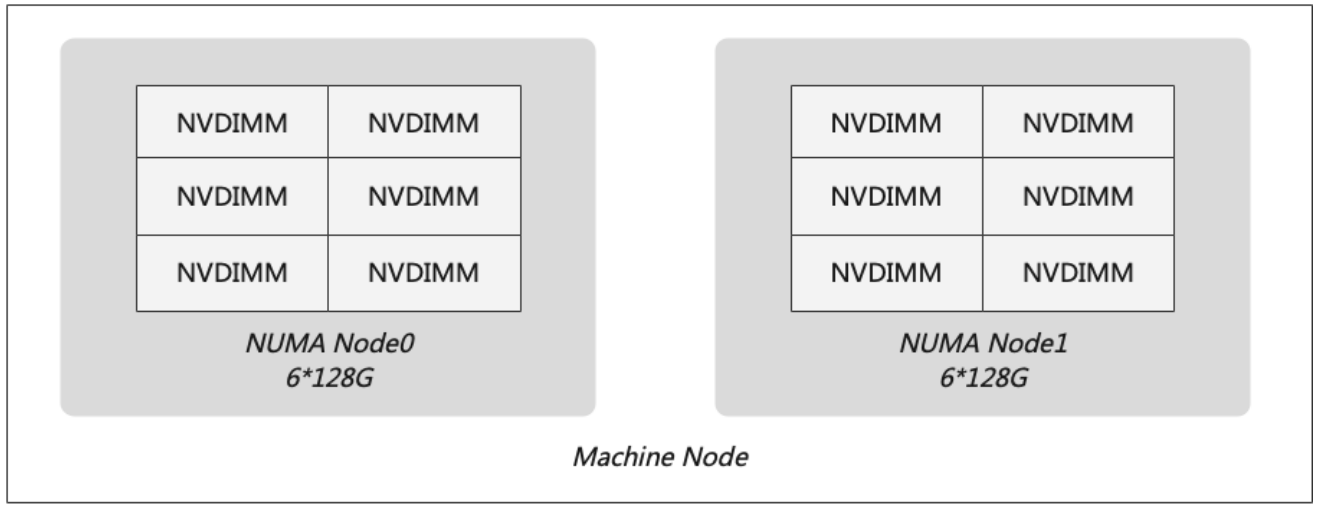
https://kubernetes.io/docs/concepts/scheduling-eviction/scheduling-framework/
https://github.com/kubernetes-sigs/scheduler-plugins

Plugins For Capacity Scheduling

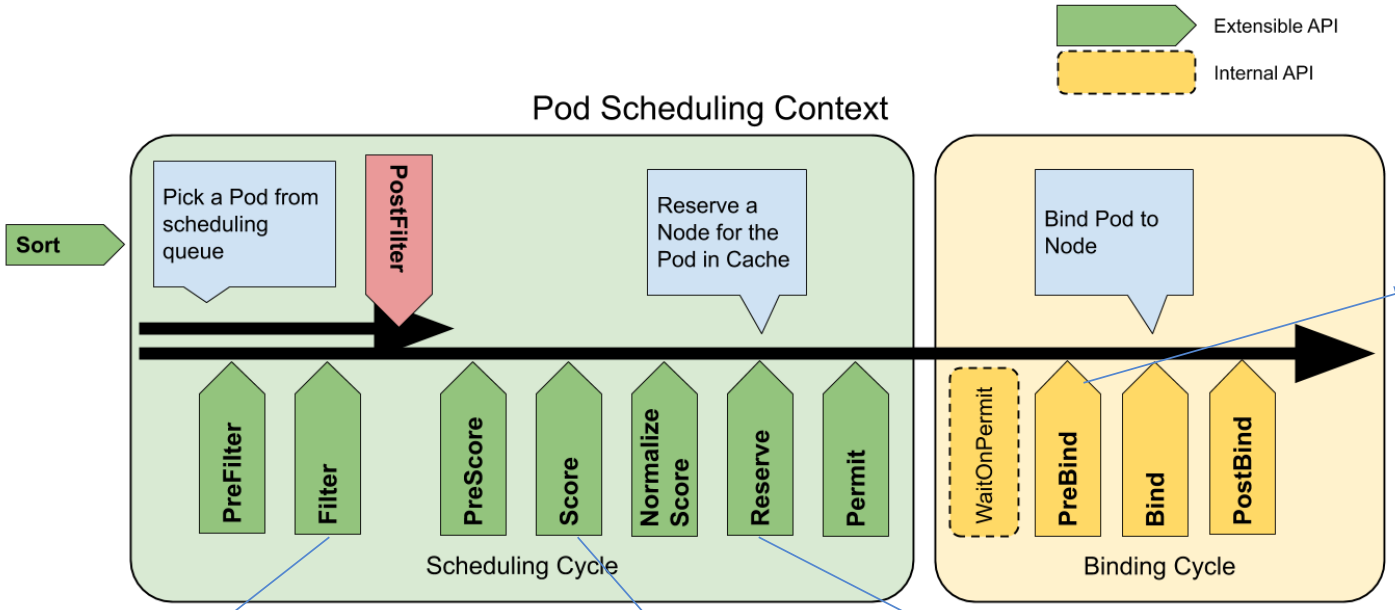
PMEM capacity selection policy

- ***Single NUMA node*** *: Allocate pod requests to the same NUMA node based on capacity*

- ***Binpack*** *: Allocate the NUMA node with the lowest remaining amount so that the pod with larger resource requests can be meet in the future*



Plugins For Capacity Scheduling



PreBind: *Update the PMEM scheduling result to the annotation of PVC*

Filter: *Filter out the nodes that can't meet the requirement of PMEM*

Score: *Select the optimal PMEM scheduling result by algorithm*

Reserve:

- *Reserve the optimal PMEM scheduling result to prevent reallocation to other pods*
- *Clean the PMEM scheduling result if failure occurs in the binding cycle*

Plugins For NUMA Aware Scheduling

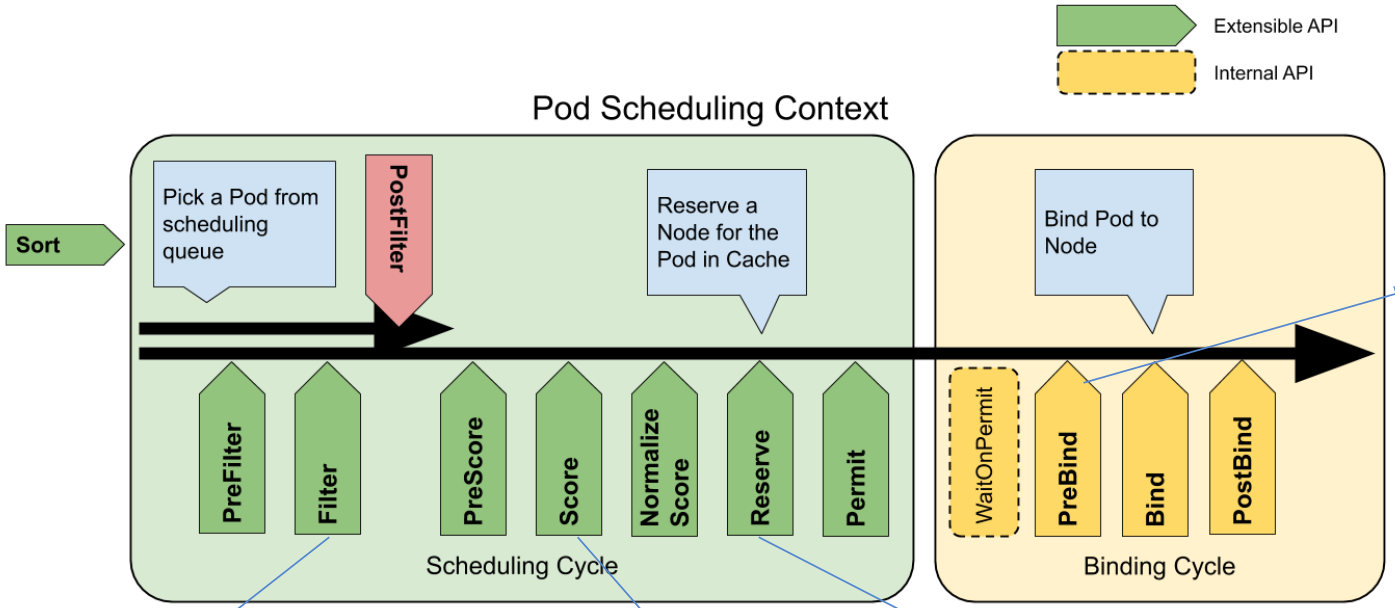
NUMA node selection policy

- ***Single NUMA node:*** *Allocate the CPU of the same NUMA node*
- ***Shortest Node Distances***: *Select the combination of PMEM and CPU with the shortest distance*

	Node 0 (DRAM)	Node 1 (DRAM)	Node 2 (PMEM)	Node 3 (PMEM)	Swap space
CPU 0 (Node 0)	10	21	17	28	
CPU 30 (Node 1)	21	10	28	17	

Node distances between NUMA nodes

Plugins For NUMA Aware Scheduling



PreBind: *Create Pod's Cgroup request to store the CPU scheduling results*

Filter: *Filter out the nodes that can't meet the requirement of the pod for CPU*

Score: *Select the combination of PMEM and CPU with the shortest distance*

Reserve:

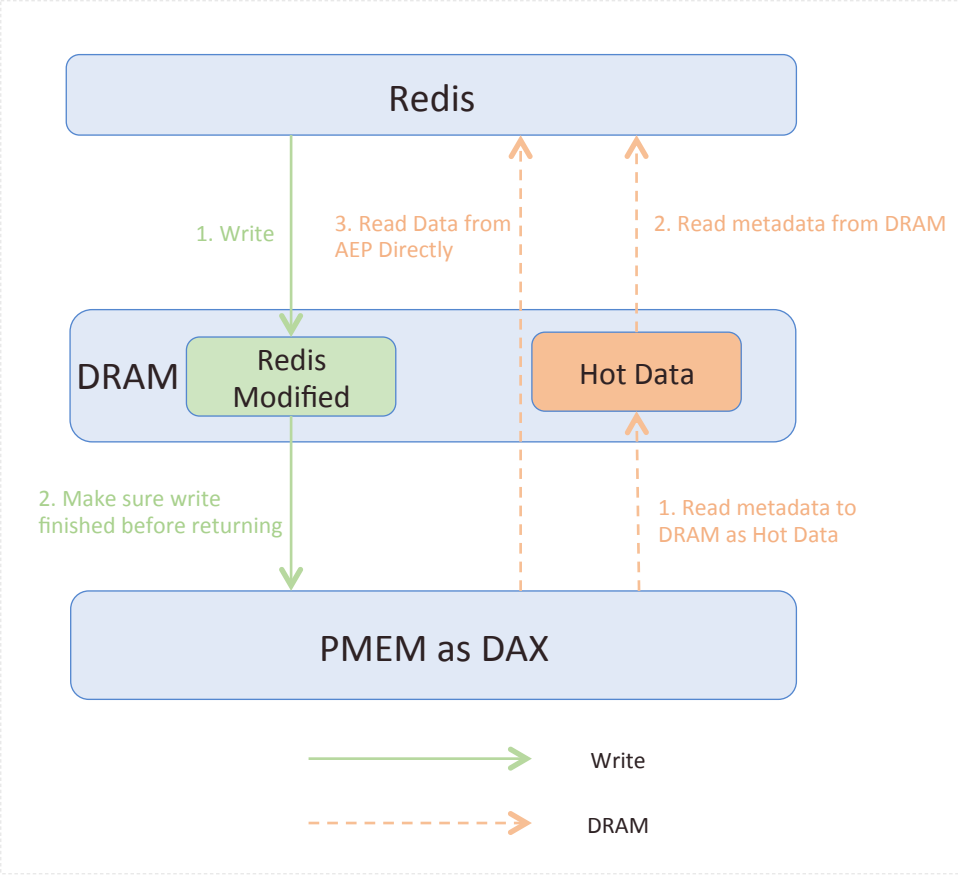
- *Reserve the optimal CPU scheduling result to prevent reallocation to other pods*
- *Clean the CPU scheduling result if failure occurs in the binding cycle*

https://static.sched.com/hosted_files/kccncna20/52/Practice%20of%20Fine-grained%20Cgroups%20Resources%20Scheduling%20in%20Kubernetes.pdf

Agenda

- 1. Cloud Native PMEM Stack Introduction
- **2. Application Practice on PMEM Stack**
- 3. Demo
- 4. Other Related Works

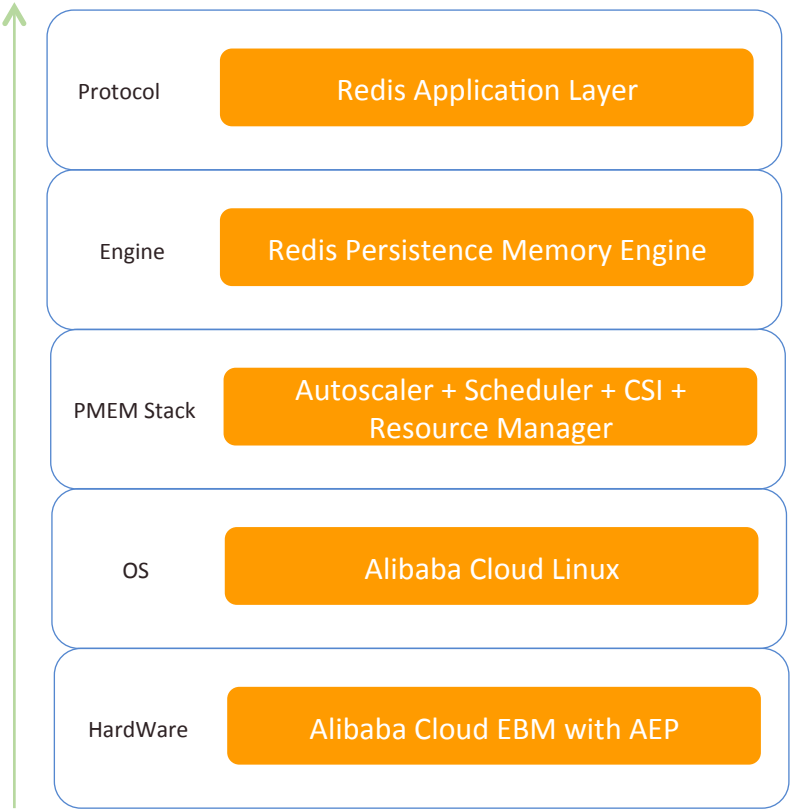
Redis on PMEM Stack



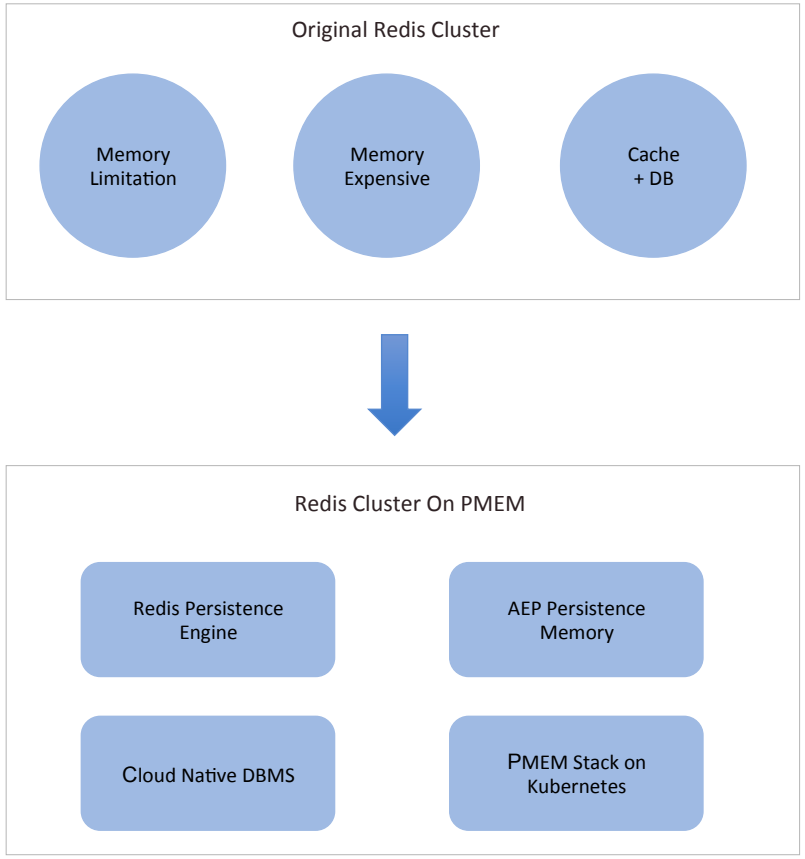
Customized Engine

Containerized Deploy

Automated Ops



Benefits

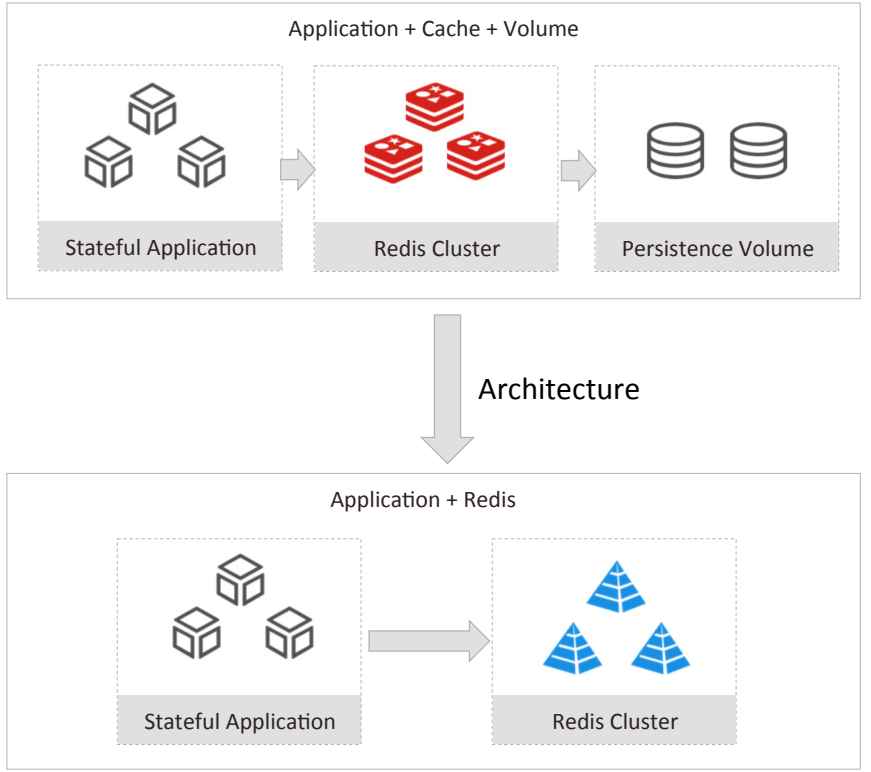


90% Performance

70% Cost

Persist for each Ops

Large Capacity



Agenda

- 1. Cloud Native PMEM Stack Introduction
- 2. Application Practice on PMEM Stack
- **3. Demo**
- 4. Other Related Works

Demo

The cluster has PMEM devices in one node, and we deploy an application using PMEM volume.

This demo will show the whole process of using PMEM volume in Kubernetes platform.

The Demo contains :

- PMEM devices management
- CSI volume management
- Online resize and auto resize volume

Agenda

- 1. Cloud Native PMEM Stack Introduction
- 2. Application Practice on PMEM Stack
- 3. Demo
- **4. Other Related Works**

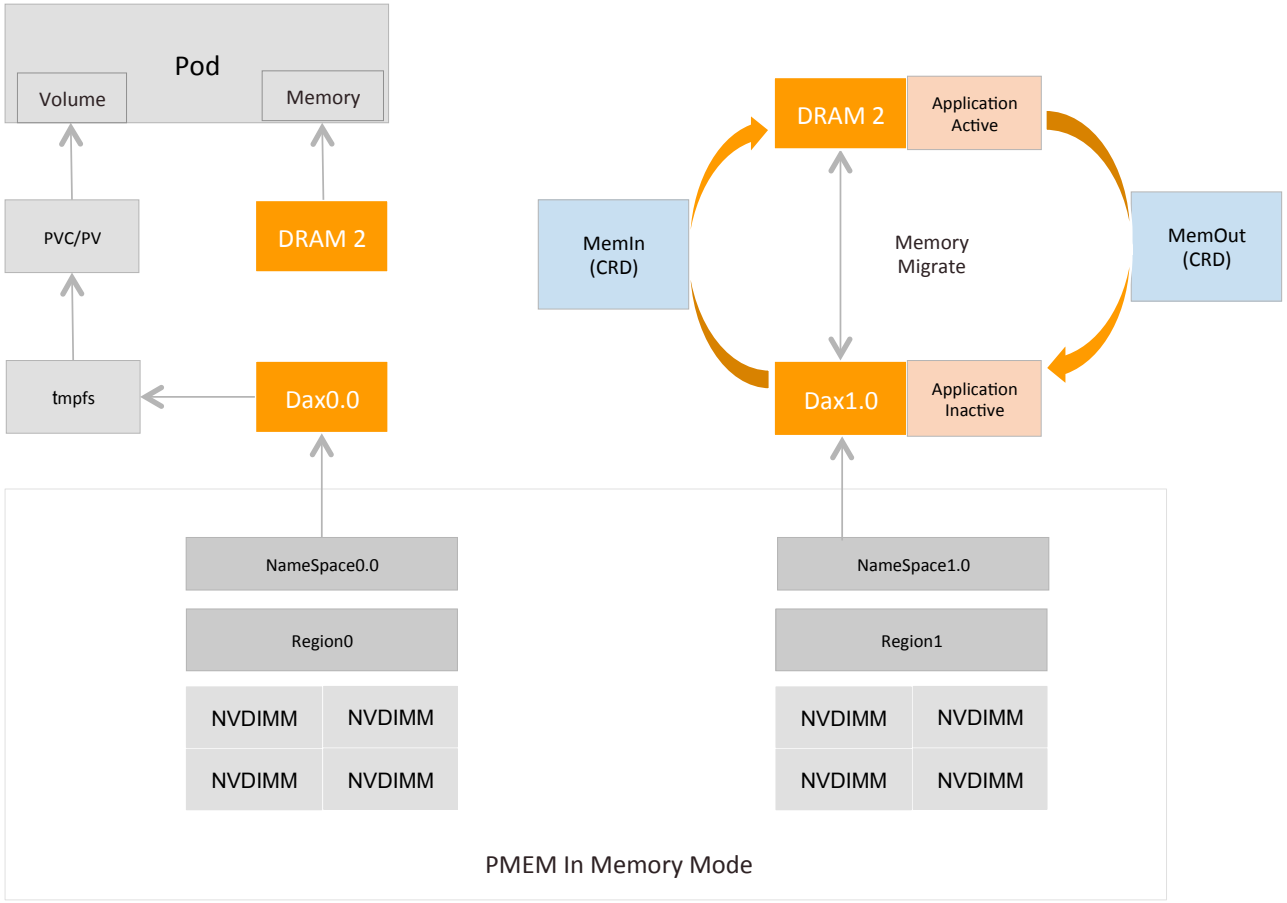
PMEM in Memory Mode

PMEM used as tmpfs:

- Format Namespace into devdax and configure to system-ram;
- Provision tmpfs volume from PMEM memory;
- Used as high performance storage in Pod;

PMEM as memory cache:

- DRAM is used to active app;
- PMEM is used to inactive app;
- Migrate pages from DRAM to PMEM for TaskSet of Pod/Container
  - https://github.com/AliyunContainerService/numactl



Future works

PMEM in Big Data:

- Scheduling
- Access data acceleration

Memory pool:

- Dynamic provision memory from pool
- Attach/detach memory
- Computing memory separation

RESILIENCE REALIZED