KubeCon | CloudNativeCon

OPEN SOURCE SUMMIT China 2023

How Can Pod Start-up Be Accelerated on Nodes in Large Clusters?

Paco Xu, DaoCloud
Byron Wang, Birentech

Intro us

Paco Xu

DaoCloud Shanghai.

Mainly worked on kubeadm & sig-node

Github: pacoxu

Twitter: xu_paco

❤️ → ⚽ & PUBG

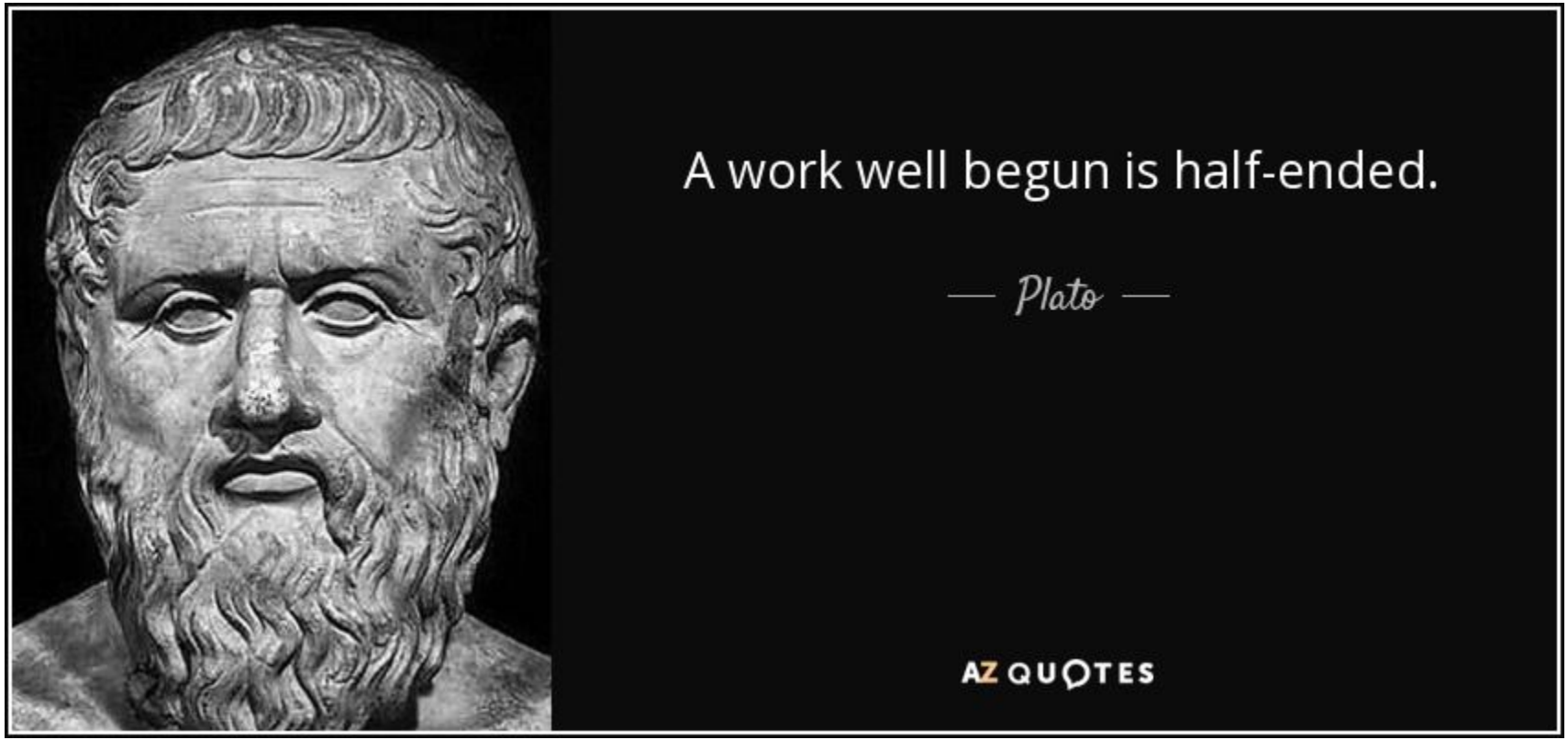
Byron Wang

壁仞科技 BIREN TECHNOLOGY Shanghai.

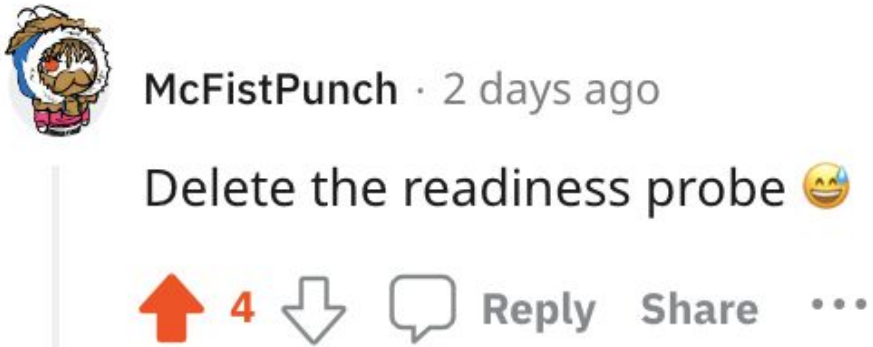
Mainly worked on scheduler & AIOps

Github: 41tair

Why is pod startup speed important?



How？



Can we edit the pod start-up time by kubectl？

Can we edit my salary with kubectl? 😄

How long should a pod startup cost?

- ❑ <50ms
- ❑ ~100ms
- ❑ ~seconds
- ❑ 15s ~ 1m
- ❑ <5m

The startup time for a pod can vary significantly depending on several factors, including the complexity of the application, the size of the container image, the resources allocated to the pod, and the underlying infrastructure. Here's a rough breakdown of the pod startup times you provided:

- <50ms: Extremely fast pod startup time, typically achievable for very lightweight applications or small-scale deployments with minimal dependencies.
- ~100ms: Still very fast startup time, suitable for simple applications or when using optimized and cached container images.
- ~seconds: A broader range that can encompass startup times in the range of a few seconds, which is common for many applications.
- 15s ~ 1m: A reasonable startup time for most applications. Many pods fall within this range, especially when dealing with larger container images or more complex setups.
- <5m: Startup times up to 5 minutes may occur for more resource-intensive applications or when pulling large container images over slower networks.

Content
目录

0. Pod Creation

APIServer

When the pod is created by a controller/operator, the pod creation speed matters.

If API Server is slow, why?

- Slow etcd
- High load
- Rate limit is hit
- Slow webhooks

If pod failed to create or waiting for some conditions,

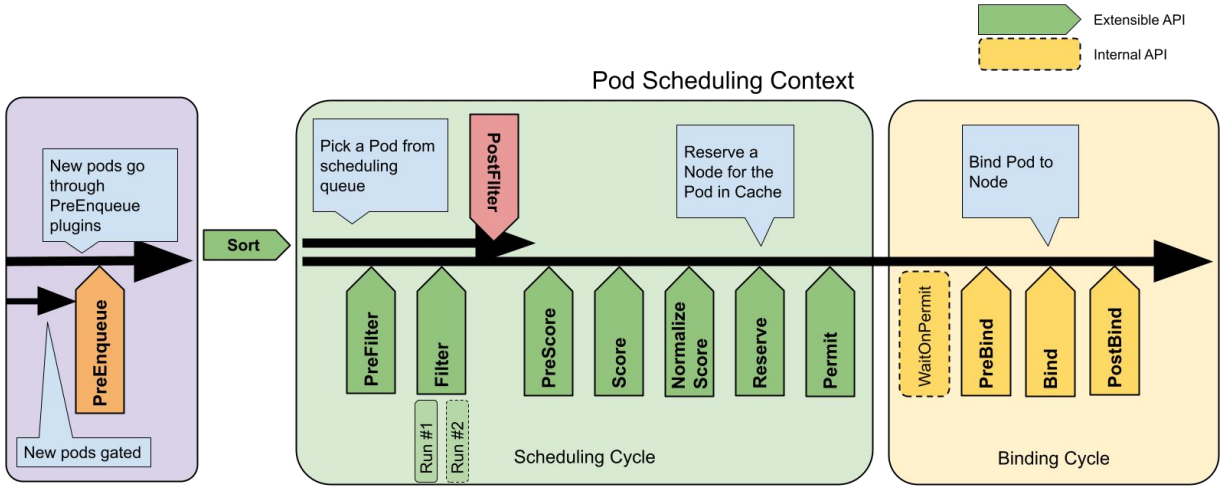
- Validation failure
- Webhooks: Pod Security Admissions
- Cluster resource is full(for instance, waiting for job completions)
  - Pod Priority and Preemption
- Namespace resource quota for pod count

1. Pod Scheduling

Default Scheduler

If pod.spec.Node is not specified, scheduler will find a node.

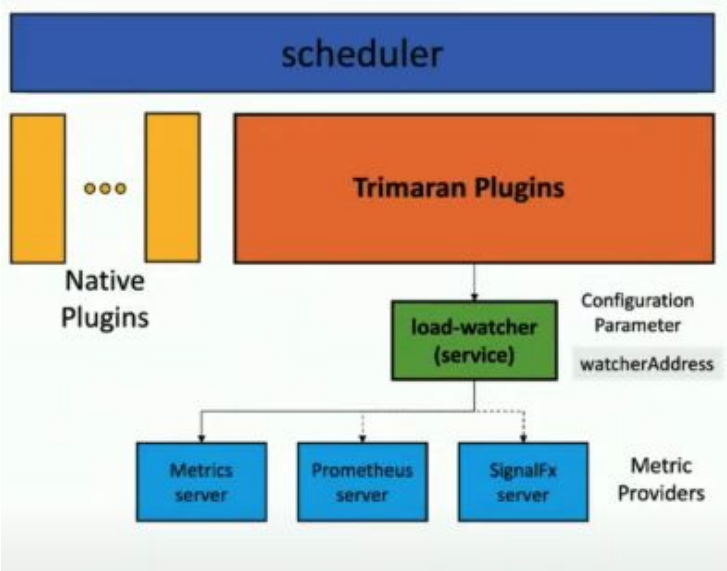
- Failures(retry): Is there a failure for the pod scheduling?
- Scale: How many nodes? A large cluster
- Conditions: Any filters like node affinity?
- Load: How many pods are scheduled at the same time?



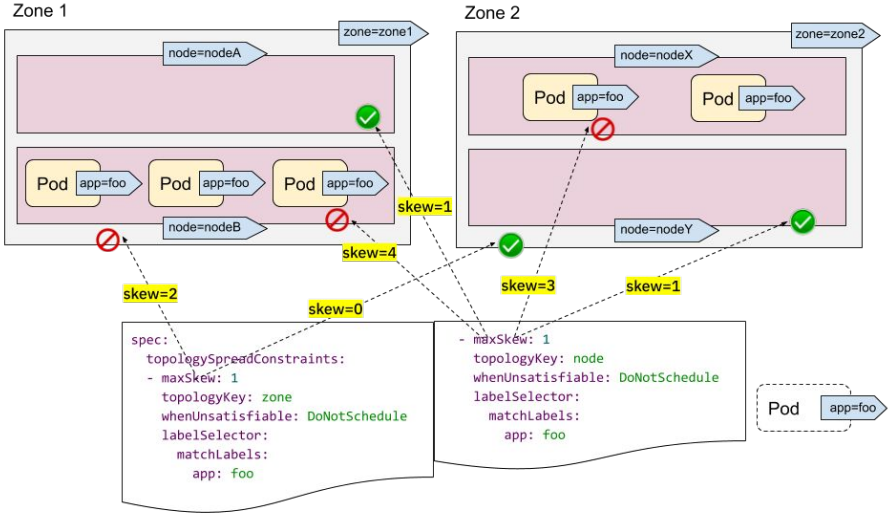
What's more?

`Load-aware`: node aware scheduling, Trimaran

- Pods will start on the least loaded node.



Spread Strategy: reduce the mutual influence of the overall pod startup as Pods are started on different nodes.

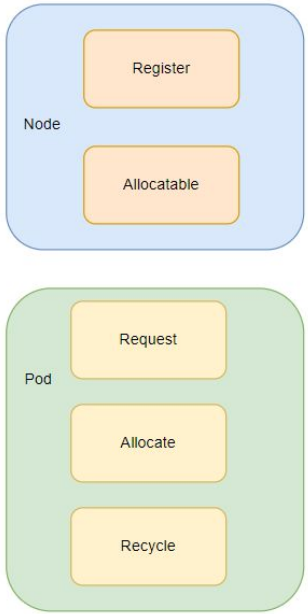



KubeCon
CloudNativeCon
OPEN SOURCE SUMMIT
China 2023
When it comes to AI/GPU

GPU Register and Allocate


```yaml
apiVersion: v1
kind: Pod
metadata:
  name: pod1
spec:
  restartPolicy: OnFailure
  containers:
  - image: ubuntu
    name: pod1-ctr
    command: ["sleep"]
    args: ["100000"]
    resources:
      limits:
        birentech.com/gpu: 1
```

Custom Resource:
birentech.com/gpu

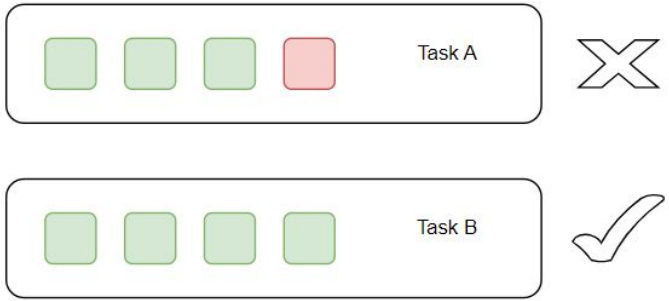


Problems

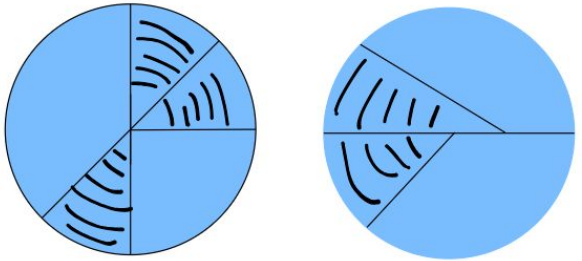
1. Device number changed or property changed
2. Different device types from single vendor
3. Multiple vendor’s device in one cluster
4. Allocate a device to one more pod or container
5. Allocate a selected device

GPU Scheduler Algorithm

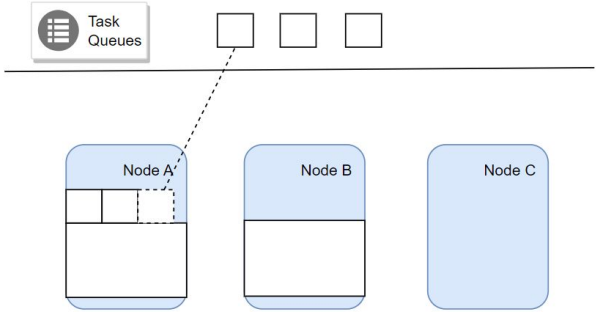
Gang



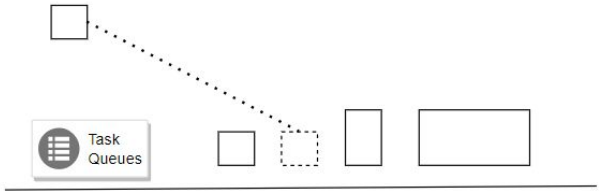
SKU



Binpack

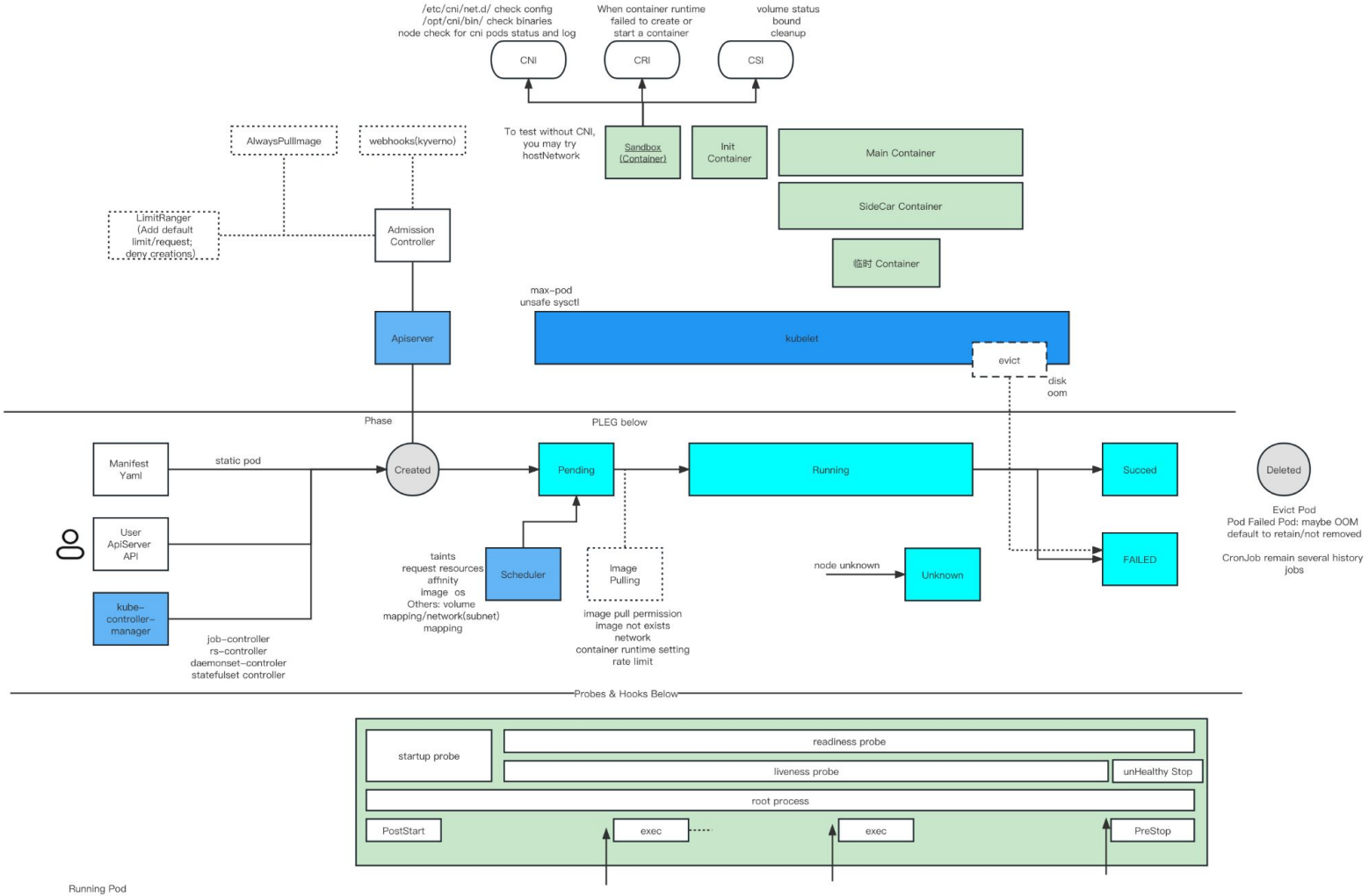


DRF



KubeCon | CloudNativeCon
OPEN SOURCE SUMMIT
China 2023

2. Pod Startup on Node


/etc/cni/net.d/ check config
/opt/cni/bin/ check binaries
node check for cni pods status and log
When container runtime failed to create or start a container
volume status bound cleanup
CNI
CRI
CSI
To test without CNI, you may try hostNetwork
Sandbox (Container)
Init Container
Main Container
SideCar Container
临时 Container
AlwaysPullImage
webhooks(kyverno)
LimitRanger (Add default limit/request; deny creations)
Admission Controller
max-pod
unsafe sysctl
Apiserver
kubelet
evict
disk
oom
Phase
PLEG below
Manifest Yaml
static pod
User ApiServer API
kube-controller-manager
job-controller
rs-controller
daemonset-controller
statefulset controller
Created
Pending
Running
Succed
FAILED
Deleted
Evict Pod
Pod Failed Pod: maybe OOM
default to retain/not removed
CronJob remain several history jobs
taints
request resources
affinity
image os
Others: volume mapping/network(subnet) mapping
Scheduler
Image Pulling
image pull permission
image not exists
network
container runtime setting
rate limit
node unknown
Unknown
Probes & Hooks Below
startup probe
readiness probe
liveness probe
unHealthy Stop
root process
PostStart
exec
exec
PreStop
Running Pod

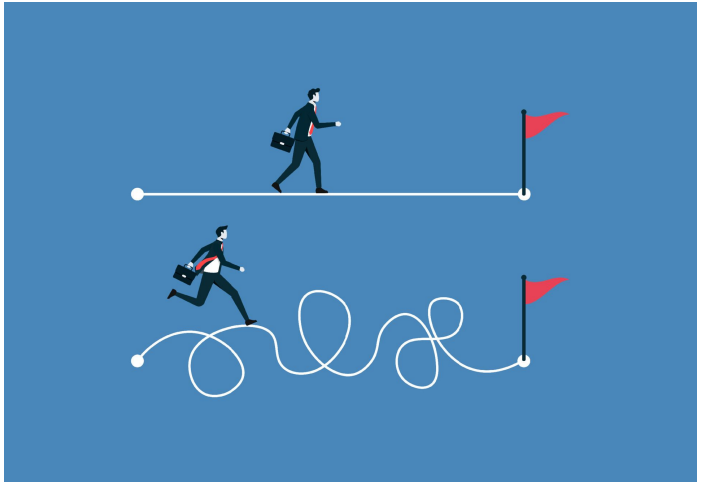
Pod Startup Steps: the simple way

A simple pod startup steps:

1. Pull Image
2. Sandbox Creation
   a. CNI
3. Volume mount or Secret/Configmap
4. Container main process starts

To make things simpler:

1. Pull Image in advance
2. Using `hostNetwork:true`
3. No volume mount
4. With no CPU/memory limitations



Big Image: make it smaller

Some tips

1. Add `.dockerignore` to exclude unnecessary files or dirs
2. Choose a smaller base image: alpine, scratch
3. Multistage builds to exclude files for building it
4. Less layers: merged layer or a squash may save some spaces
5. Don’t install debug tools: try to use `kubectl debug` by ephemeral containers

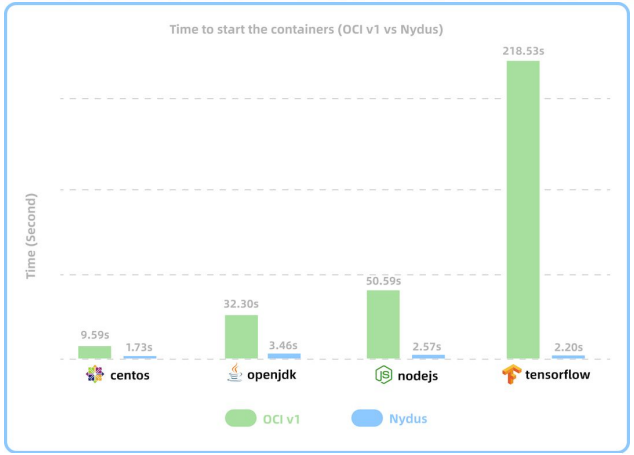
Extended reading:

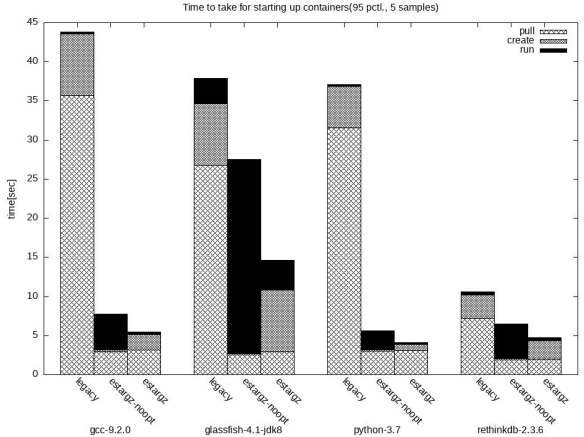
- wagoodman/dive is a tool for exploring each layer in a docker image

Big Image: p2p & lazy pulling

- Dragonfly is an open source P2P-based file distribution and image acceleration system.
  - Nydus: Dragonfly Container Image Service
- Spegel: a stateless cluster local OCI registry mirror. (lightweight)
- uber/kraken: P2P Docker registry capable of distributing TBs of data in seconds

- Containerd:
  - containerd/stargz-snapshotter
  - Lazy-pulling using Nydus Snapshotter





Parallel Image Pulling

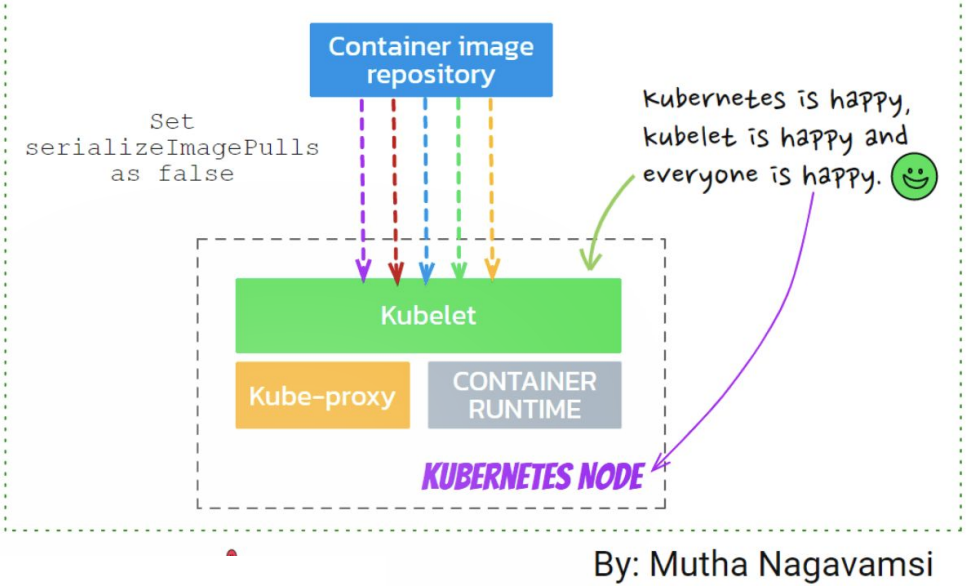
Kubelet will send only one image pull request at a time by default.

1. Set `serializeImagePulls: false`
2. Set `maxParallelImagePulls=n` to avoid heavy disk. (since v1.27)

Why `serializeImagePulls` is by default false?

1. old issue with aufs before docker v1.9
2. Performance issue before contained 1.6.3 & 1.7.0

Any feedback are welcome in #108405.



Init Containers

Do necessary initialization only. Do them in parallel.

Use `PostStart` hook if possible.

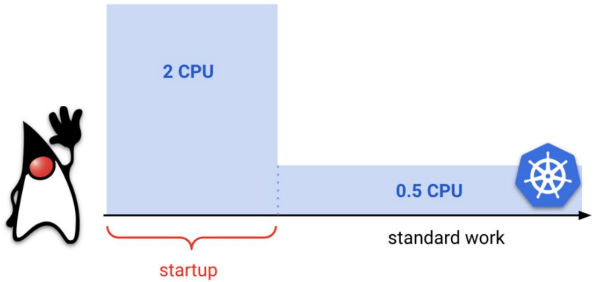
Make preparations as early as possible:

- Can this be done during image building?
- Can this be done with a DaemonSet/Pod on node?

Use a larger CPU limit at the beginning

Avoid CPU throttling.

Better use VPA since v1.27:
`--feature-gates=InPlacePodVerticalScaling=true`



Resize CPU Limit To Speed Up Java Startup on Kubernetes

By piotr.minkowski ● August 22, 2023 ● 8

Static CPU Policy

Change node CPU manager policy Steps

1. Drain the node.
2. Stop kubelet.
3. Remove the old CPU manager state file. The path to this file is `/var/lib/kubelet/cpu_manager_state` by default. This clears the state maintained by the CPUManager so that the cpu-sets set up by the new policy won't conflict with it.
4. Edit the kubelet configuration to change the CPU manager policy to the desired value.
5. Start kubelet.

⚠️ limitation:
https://github.com/kubernetes/kubernetes/issues/116086

- If your pod has sidecar containers, the sidecar CPU limit/request has to be **guaranteed**.

```
apiVersion: v1
kind: Pod
metadata:
  name: multi-container
spec:
  restartPolicy: Always
  hostNetwork: true
  containers:
  - name:  one-core
    image: m.daocloud.io/docker.io/library/ubuntu
    command:
    - sleep
    - '1100000'
    resources:
      limits:
        memory: "400Mi"
        cpu: "1"
      requests:
        memory: "400Mi"
        cpu: "1"
  - name: no-integer-cpu
    image: m.daocloud.io/docker.io/library/ubuntu
    command:
    - sleep
    - '1100000'
    resources:
      limits:
        memory: "400Mi"
        cpu: "100m"
      requests:
        memory: "400Mi"
        cpu: "100m"
```

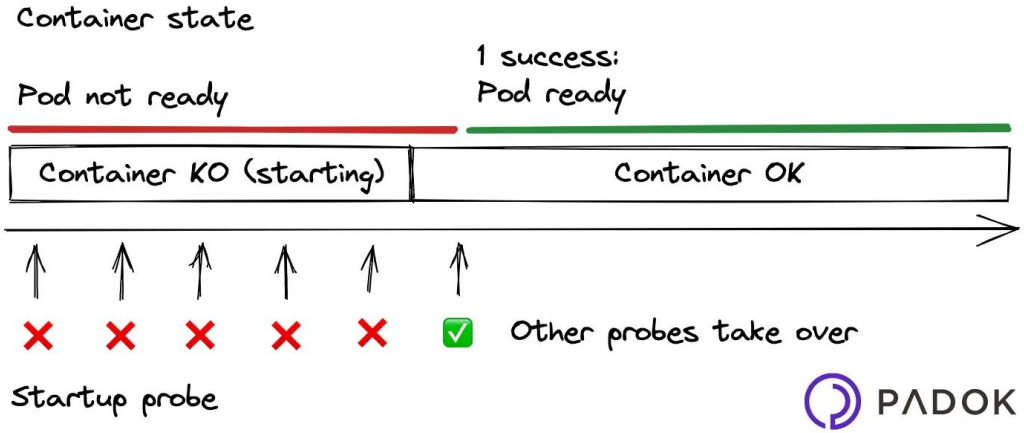
Probe

StartupProbe:

- If set, decrease the readiness probe's `initialDelaySeconds`.

ReadinessProbe:

- Set `initialDelaySeconds` if no startup probe is set.



Images from: https://www.padok.fr/en/blog/kubernetes-probes

Roadmap:

- Sub-second / More granular probes: make it possible to set millisecond for probes

Forensic container checkpointing

Slow Pod Startup Challenge:

- Loading extensive data into memory
- Time-consuming initialization processes

Solution: Forensic container checkpointing(alpha in v1.25)

- Create a snapshot of the fully initialized pod
- Store it for rapid use

Benefits:

- Instantaneous response to user requests
- Efficient resource utilization
- Consistent performance

General factors

For single pod startup:

- Container runtime: Docker vs Containerd?
- CNI: most ipam on node is very fast
- The SELinux Relabeling with Mount Options(beta in v1.27) mounts volumes with the correct SELinux label instead of changing each file on the volumes recursively.

For multi pods startup at the same time:

- Since v1.27, kubelet default API QPS limits bump 10 times.
- Event triggered updates to container status(Evented PLEG is beta since v1.27).

```
- kubeAPIBurst: 10
- kubeAPIQPS: 5
+ kubeAPIBurst: 100
+ kubeAPIQPS: 50
```


KubeCon
CloudNativeCon
OPEN SOURCE SUMMIT
China 2023
When it comes to AI/GPU

DRA(Dynamic Resource Allocator)

KubeCon CloudNativeCon OPEN SOURCE SUMMIT China 2023

Can do

1. Multiple resources args
2. Network attached resources
3. Init and clean strategy
4. User friendly api
5. Allow resource management cluster add-ons
6. More complicated allocation rules

Caveats & Risk

1. Slower pod scheduling
2. Additional complexity when describing pod requirements

KEP-3063: Dynamic resource allocation

GPU Management

1. A new container runtime

nvidia-container-runtime

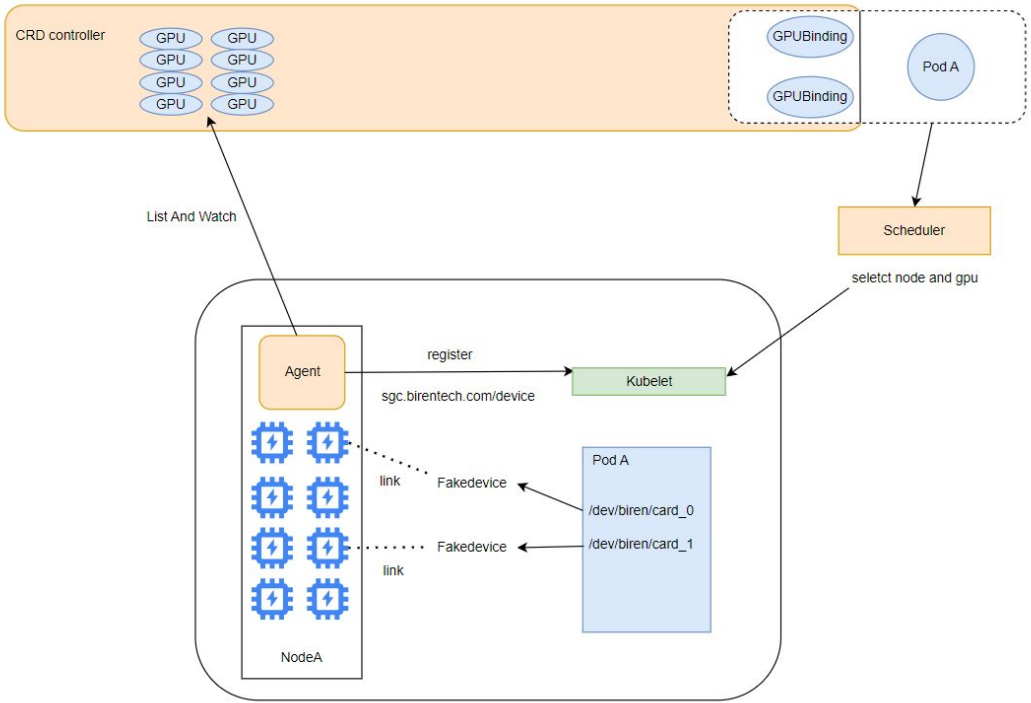
2. Use CRI proxy

koordinator

3. Modify kubelet core code

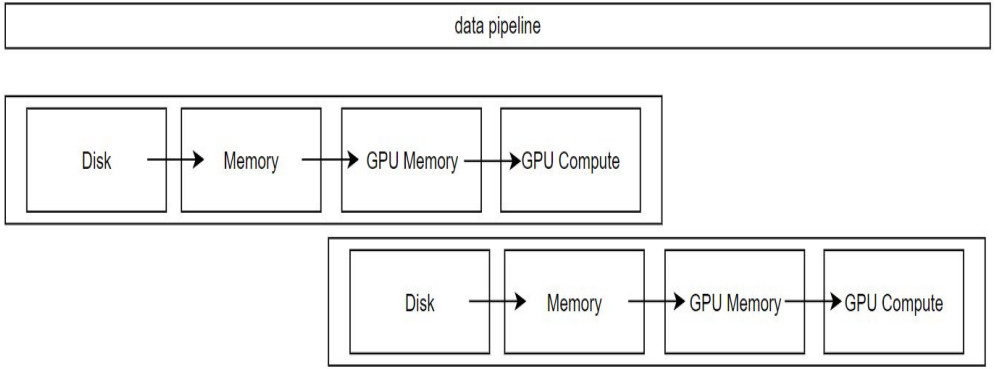
4. DRA and CDI

Fake device and Custom Scheduler

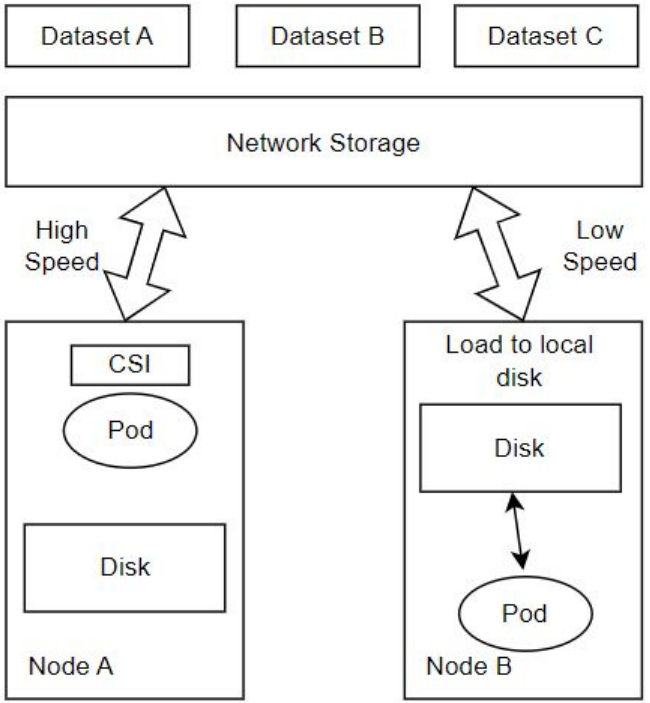



```yaml
apiVersion: batch.sgc.birentech.com/v1
kind: GPUBinding
metadata:
  creationTimestamp: "2023-08-22T09:27:18Z"
  finalizers:
  - batch.sgc.birentech.com/finalizer
  generateName: pytorch-5c53ab43-0d12-400f-8322-5583d80e3e84-worker-0-gpubinding-0
  generation: 1
  labels:
    podName: pytorch-5c53ab43-0d12-400f-8322-5583d80e3e84-worker-0
    sucloud.birentech.com/managedBy: resource-engine
    taskname: 5c53ab43-0d12-400f-8322-5583d80e3e84
  name: pytorch-5c53ab43-0d12-400f-8322-5583d80e3e84-worker-0-gpubnfsgq
  namespace: e4aeec3c-7761-4aca-87cd-c1736b97b461
  resourceVersion: "298630354"
  uid: a6cf8394-3afa-464d-b015-a453d4e46fad
spec:
  bindOptions: Only
  containerName: pytorch
  gpu: ""
  podName: pytorch-5c53ab43-0d12-400f-8322-5583d80e3e84-worker-0
  resourceName: biren/br104
status:
  status: Pending
```

Speed Up Data Loading

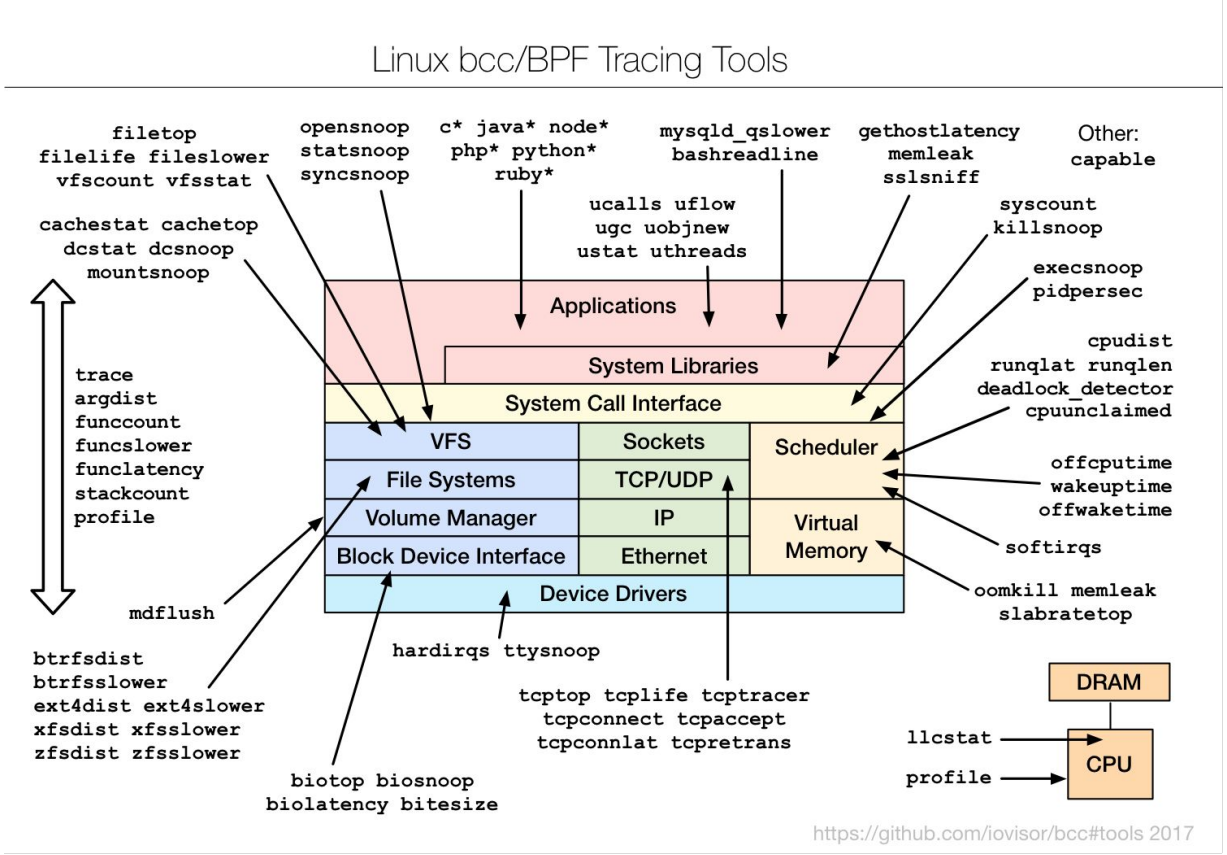


1. Prepare
2. Copy to memory
3. Copy to GPU memory
4. GPU Compute




KubeCon
CloudNativeCon
OPEN SOURCE SUMMIT
China 2023
Observability

Use Linux Tools



Log & Metrics

pod_start_sli_duration_seconds

```
"Observed pod startup duration"
pod="kube-system/konnectivity-agent-gnc9k"
podStartSLOduration=-9.223372029479458e+09
pod.CreationTimestamp="2022-12-30 15:33:06"
firstStartedPulling="2022-12-30 15:33:09"
lastFinishedPulling="0001-01-01 00:00:00"
observedRunningTime="2022-12-30 15:33:13"
watchObservedRunningTime="2022-12-30 15:33:13"
```

This log and metrics was added in v1.26.

You can make an alert ⚠️ based on this metrics or log.

```
# TYPE kubelet_pod_start_sli_duration_seconds histogram
kubelet_pod_start_sli_duration_seconds_bucket{le="0.5"} 2
kubelet_pod_start_sli_duration_seconds_bucket{le="1"} 2
kubelet_pod_start_sli_duration_seconds_bucket{le="2"} 4
kubelet_pod_start_sli_duration_seconds_bucket{le="3"} 4
kubelet_pod_start_sli_duration_seconds_bucket{le="4"} 4
kubelet_pod_start_sli_duration_seconds_bucket{le="5"} 5
kubelet_pod_start_sli_duration_seconds_bucket{le="6"} 5
kubelet_pod_start_sli_duration_seconds_bucket{le="8"} 7
kubelet_pod_start_sli_duration_seconds_bucket{le="10"} 8
kubelet_pod_start_sli_duration_seconds_bucket{le="20"} 13
kubelet_pod_start_sli_duration_seconds_bucket{le="30"} 14
kubelet_pod_start_sli_duration_seconds_bucket{le="45"} 15
kubelet_pod_start_sli_duration_seconds_bucket{le="60"} 15
kubelet_pod_start_sli_duration_seconds_bucket{le="120"} 15
kubelet_pod_start_sli_duration_seconds_bucket{le="180"} 15
kubelet_pod_start_sli_duration_seconds_bucket{le="240"} 15
kubelet_pod_start_sli_duration_seconds_bucket{le="300"} 15
kubelet_pod_start_sli_duration_seconds_bucket{le="360"} 15
kubelet_pod_start_sli_duration_seconds_bucket{le="480"} 15
kubelet_pod_start_sli_duration_seconds_bucket{le="600"} 15
kubelet_pod_start_sli_duration_seconds_bucket{le="900"} 15
kubelet_pod_start_sli_duration_seconds_bucket{le="1200"} 15
kubelet_pod_start_sli_duration_seconds_bucket{le="1800"} 15
kubelet_pod_start_sli_duration_seconds_bucket{le="2700"} 15
kubelet_pod_start_sli_duration_seconds_bucket{le="3600"} 15
kubelet_pod_start_sli_duration_seconds_bucket{le="+Inf"} 15
kubelet_pod_start_sli_duration_seconds_sum 150.7267392
kubelet_pod_start_sli_duration_seconds_count 15
[root@daocloud ~]#
```


KubeCon
CloudNativeCon
OPEN SOURCE SUMMIT
China 2023
Thanks!
Any Questions?