
KubeCon
CloudNativeCon
Europe 2023
TiKV

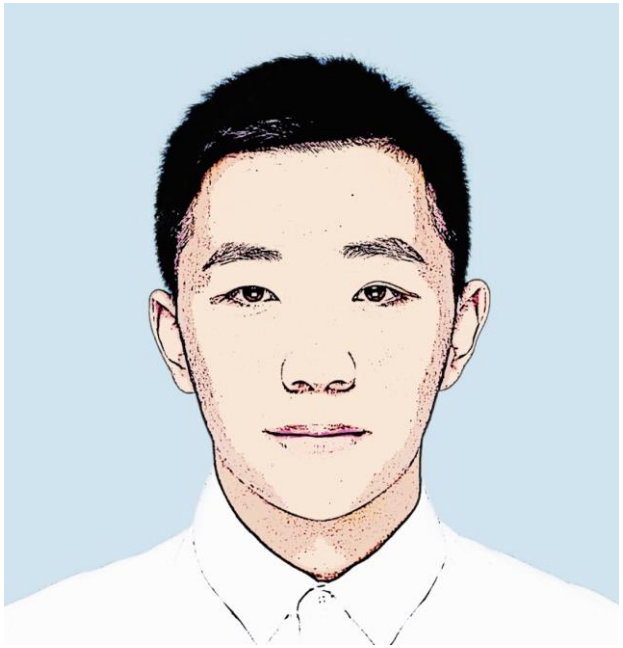
Best Practices for Accelerated Image Distribution Using Dragonfly

Wenbo Qi – Ant Group
Yiyang Huang - ByteDance

Speakers

Wenbo Qi

Software Engineer, *Ant Group*



Yiyang Huang

Software Engineer, *ByteDance*

Introduction

Introduction:

Dragonfly is an ***open source P2P-based file distribution and image acceleration system***. It is hosted by the Cloud Native Computing Foundation(CNCF) as an Incubating Level Project.

Milestone:

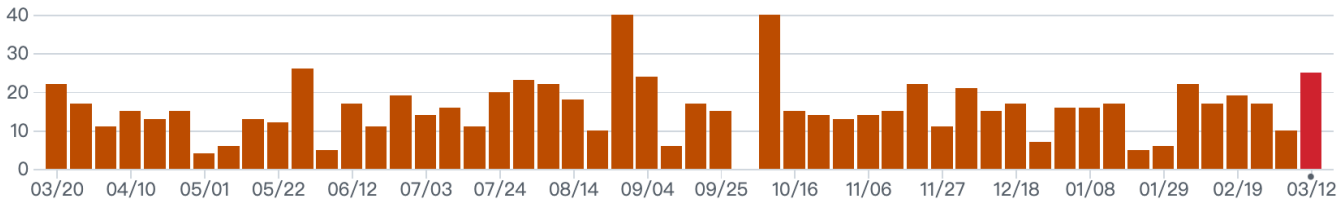
1. Dragonfly was accepted to CNCF on ***11/15/2018*** and it is a CNCF ***Incubating*** project.
2. Dragonfly ***1.X*** has been upgraded to ***2.0*** on ***9/9/2021***.
3. Dragonfly has released ***170+*** releases on ***16/3/2023***.

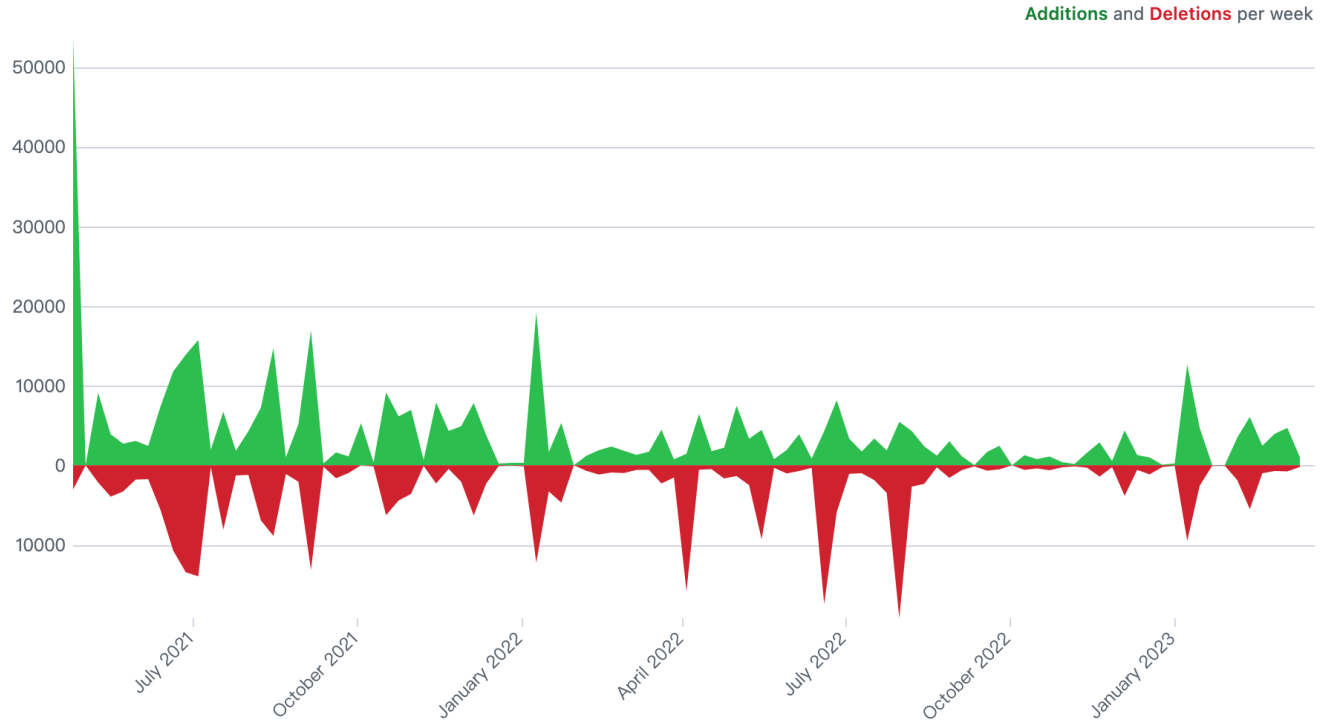
Maintainers:

From Ant Group, Alibaba, ByteDance, Baidu Group, GitLab, etc.

Discussion Group:

Slack Channel: ***#dragonfly*** on CNCF Slack
Twitter: ***dragonfly_oss***
Discussion Group: ***dragonfly-discuss@googlegroups.com***
Developer Group: ***dragonfly-developers@googlegroups.com***





Architecture

Manager:

Maintain the ***relationship*** between each P2P cluster, dynamic configuration management and RBAC. It also includes a ***front-end console***, which is convenient for users to visually operate the cluster.

Scheduler:

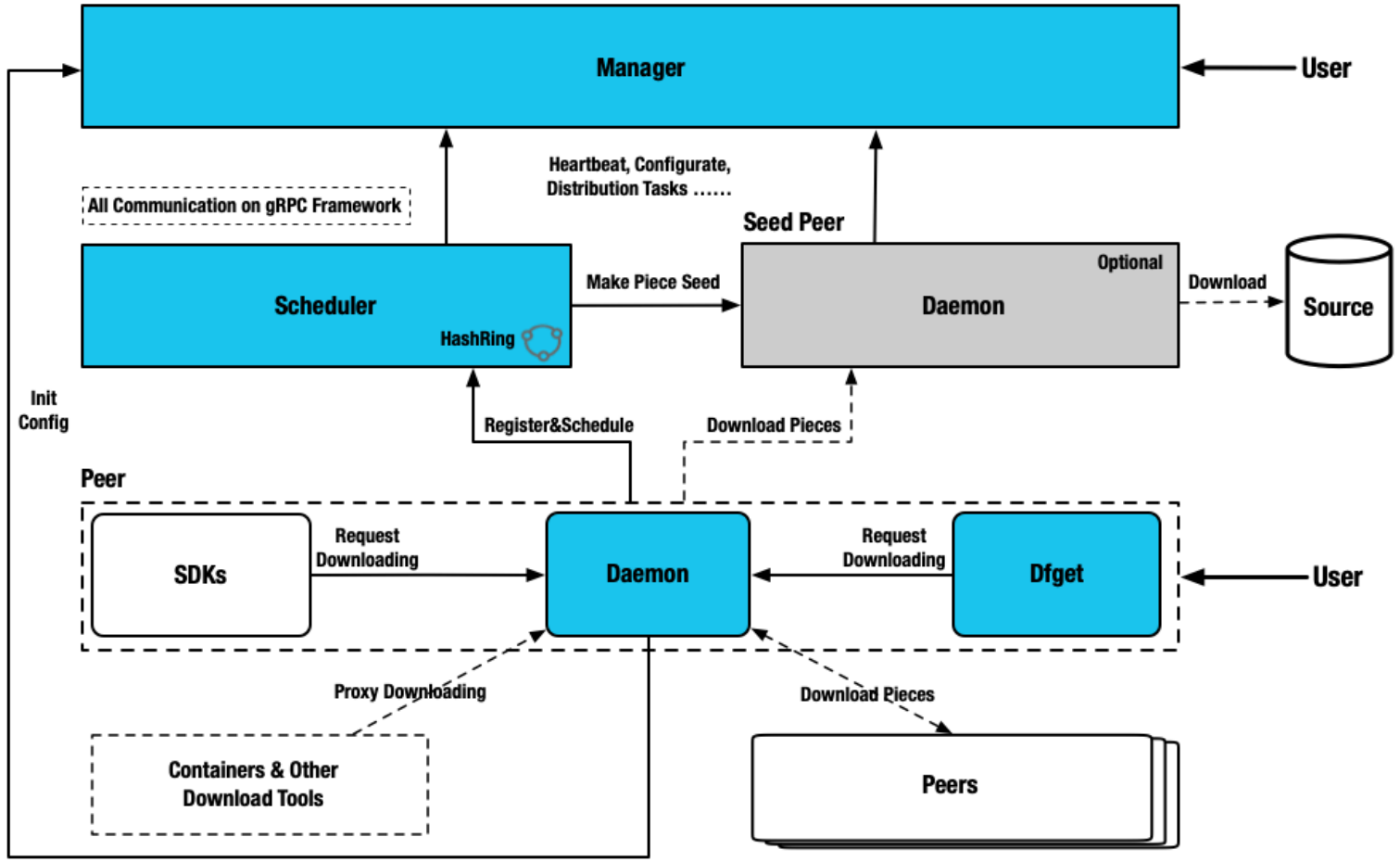
Select the ***optimal download parent*** peer for the download peer. Exceptions control peer back-to-source.

Seed Peer:

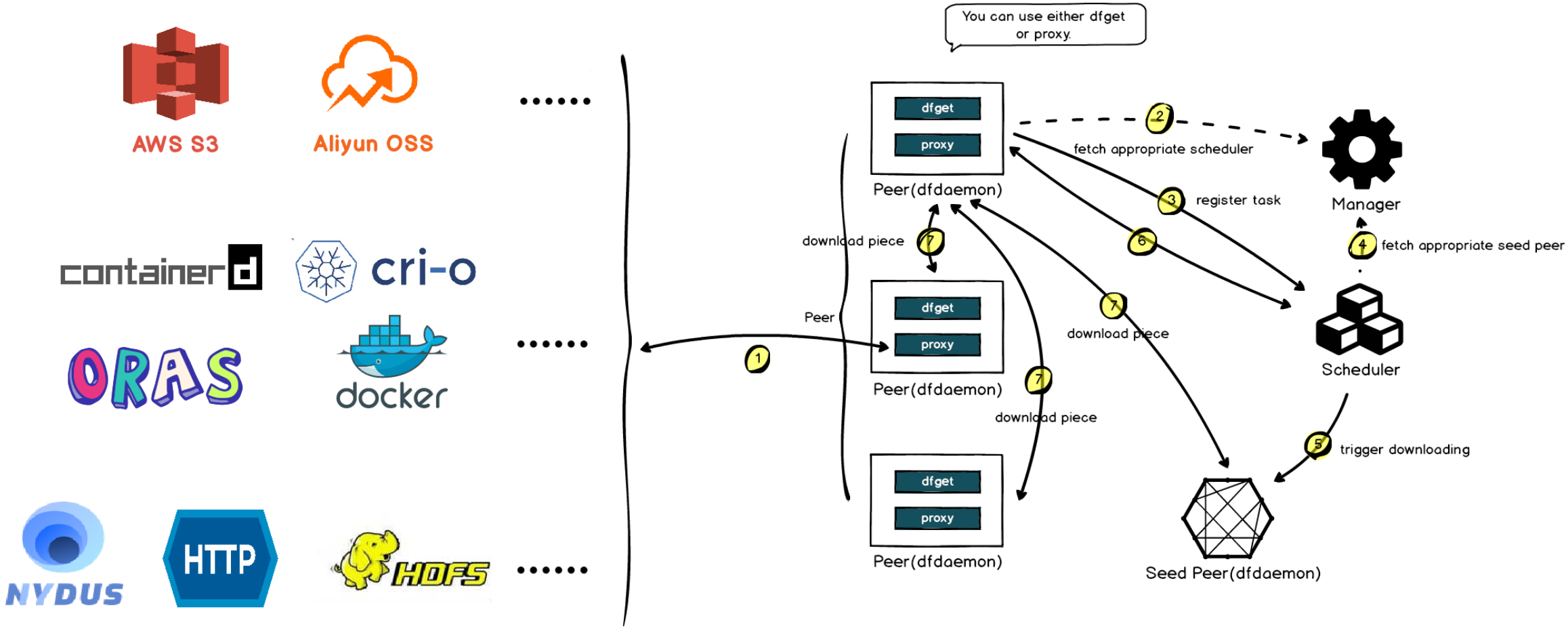
Dfdaemon turns on the Seed Peer mode can be used as a ***back-to-source download peer*** in a P2P cluster, which is the **root peer** for download in the entire cluster.

Peer:

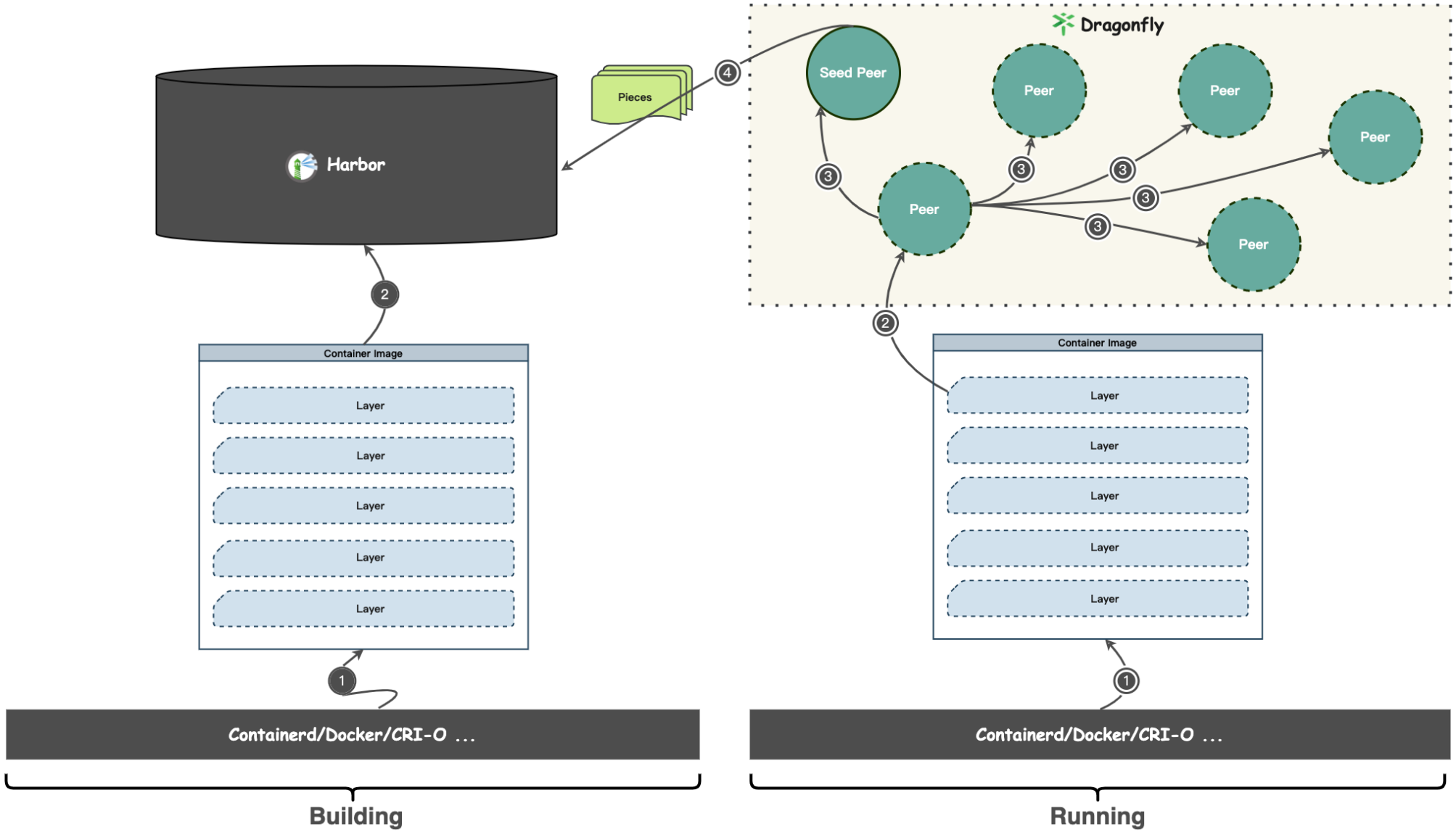
Deploy with dfdaemon, based on the C/S architecture, it provides the ***dfget*** command download tool, and the ***dfget daemon*** running daemon to provide task download capabilities.



File Distribution & Image Acceleration



Acceleration Framework For Image



Dragonfly Nydus

KubeCon CloudNativeCon Europe 2023

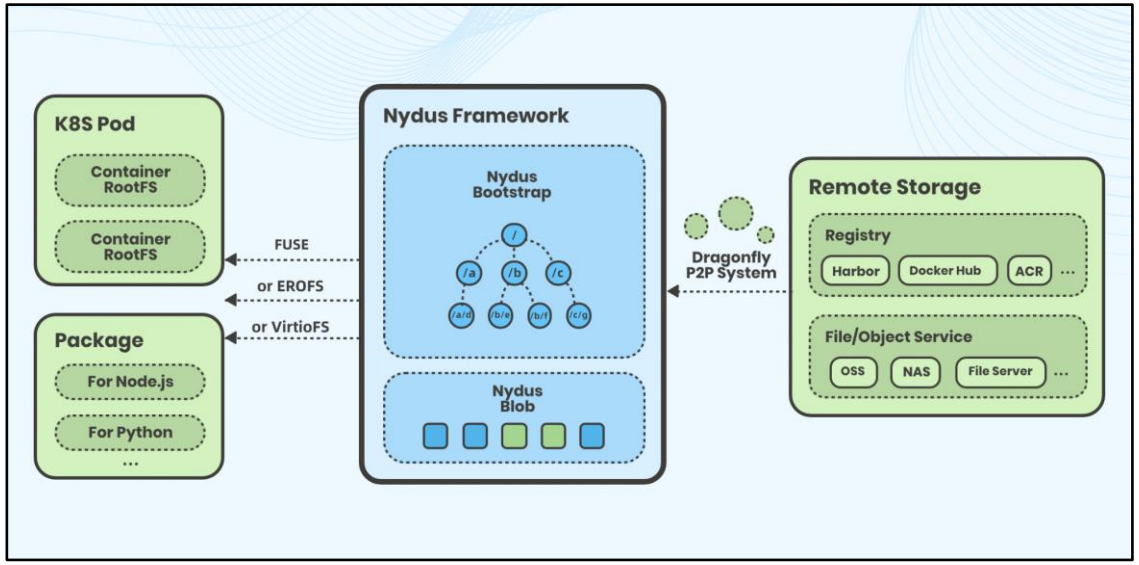
Introduction:

Nydus Acceleration Framework implements a content-addressable filesystem that can accelerate container image startup by lazy loading. It has supported the creation of millions of accelerated image containers daily. Currently, it primarily maintained by Ant Group, Alibaba Cloud, and ByteDance etc.

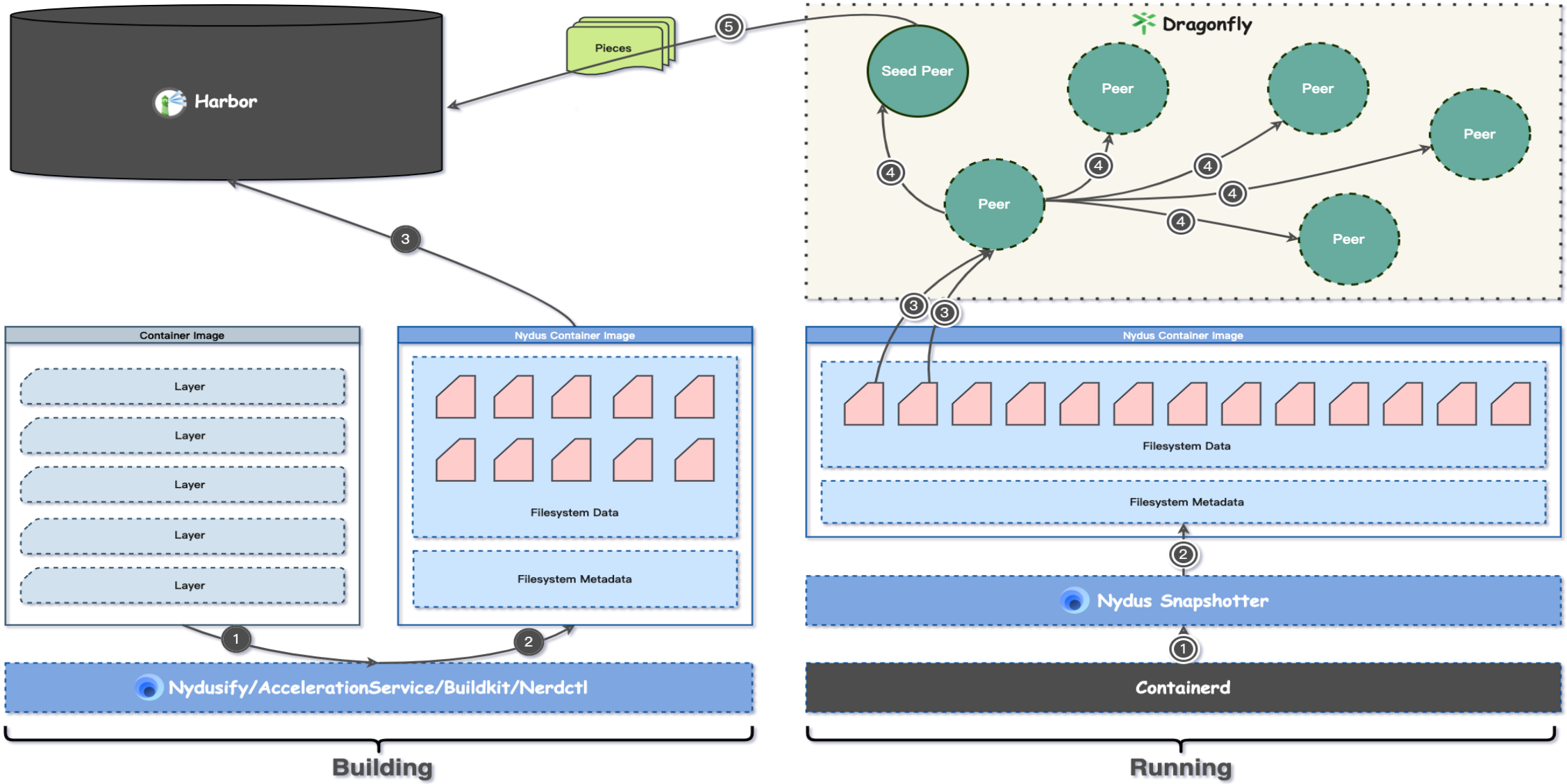
NYDUS

Features:

- Image can be fetched on demand in chunks for ***lazy pulling*** to boost container startup.
- Deeply integrated with the Linux kernel's ***erofs and fscache***, enabling ***in-kernel support*** for image acceleration.
- From build-time to run-time, nydus ***covers container ecosystems*** such as docker, containerd, podman, buildkit, nerdctl.
- ***Faster build speed*** than OCI v1 gzip/zstd format, as well as the ability to ***directly accelerate OCI v1 image.***
- The ability to deduplicate data at the chunk level between local and server-side (registry) ***reduces storage waste of over 50%+*** caused by incremental builds.



Acceleration Framework For Image



Performance Testing

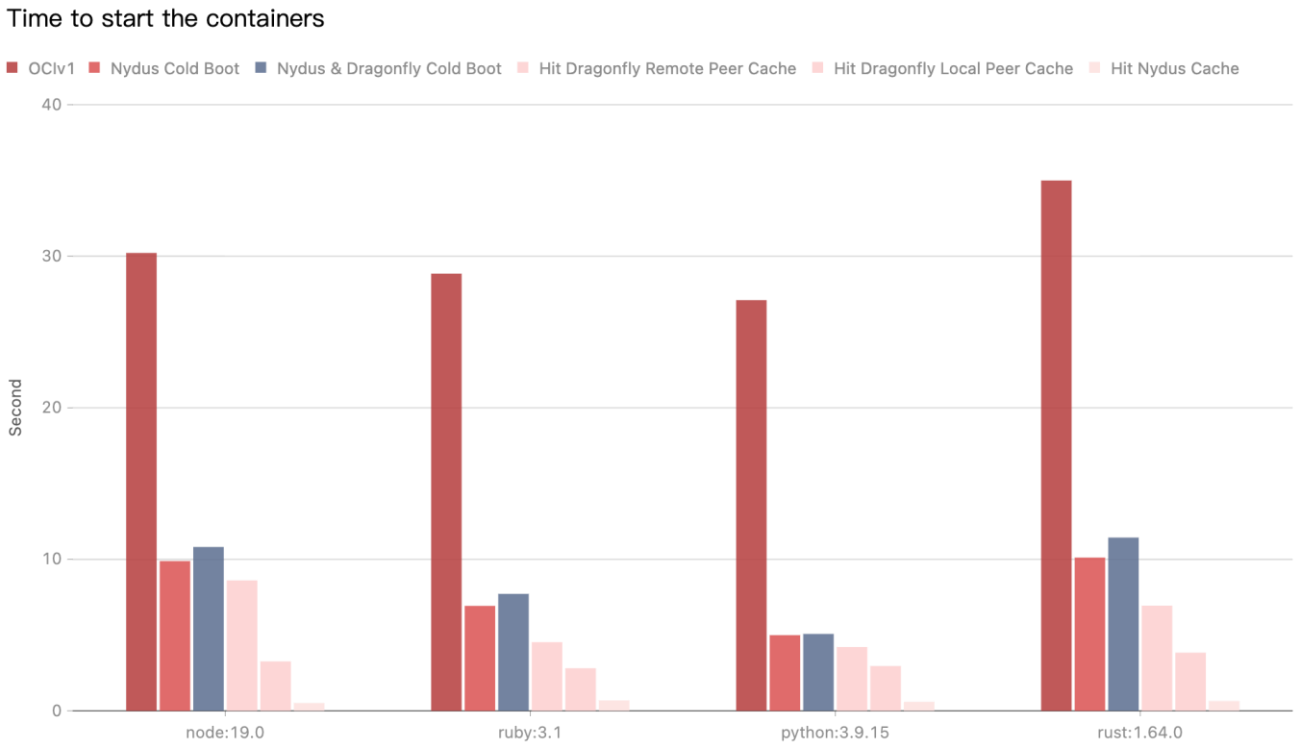
KubeCon CloudNativeCon Europe 2023

Background:

Test the performance of single-machine image download after the integration of ***nydus mirror mode and dragonfly P2P***. Test running version commands using images in different languages. For example, the startup command used to run a python image is ***python -V***.

Scenarios:

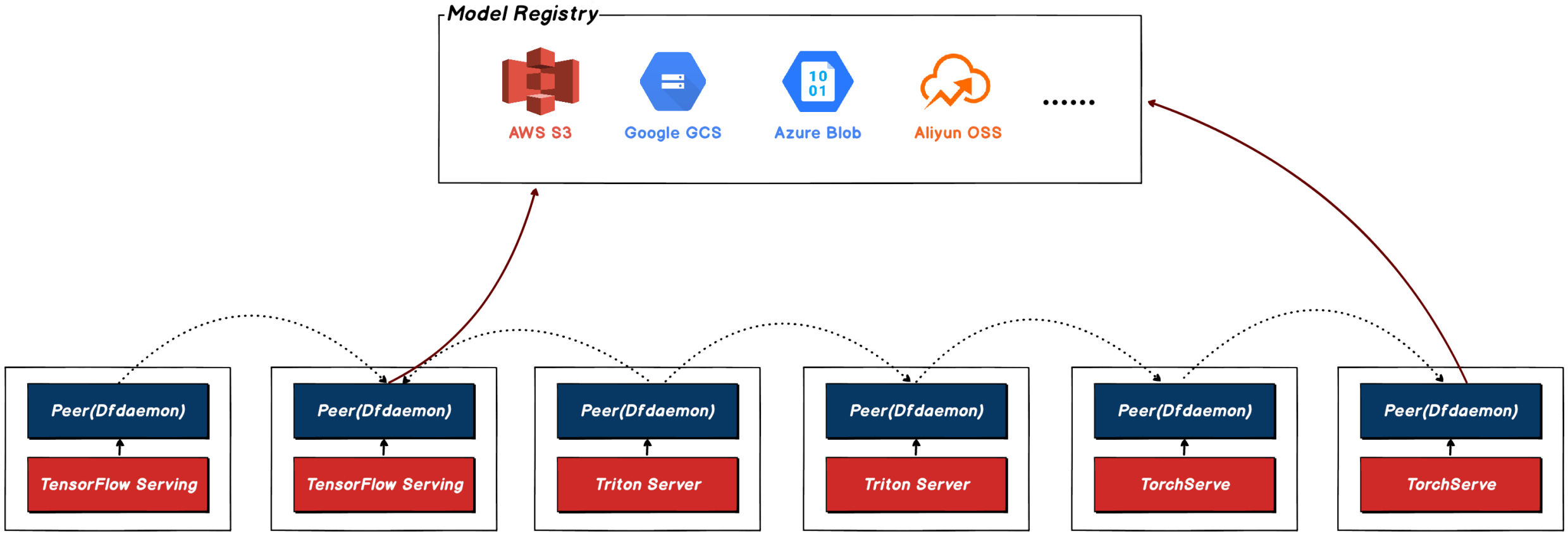
OCIv1: Use containerd to pull image directly.
Nydus Cold Boot: Use containerd to pull image via nydus-snapshotter and doesn't hit any cache.
Nydus & Dragonfly Cold Boot: Use containerd to pull image via nydus-snapshotter. Transfer the traffic to dragonfly P2P based on nydus mirror mode and no cache hits.
Hit Dragonfly Remote Peer Cache: Use containerd to pull image via nydus-snapshotter. Transfer the traffic to dragonfly P2P based on nydus mirror mode and hit the remote peer cache.
Hit Dragonfly Local Peer Cache: Use containerd to pull image via nydus-snapshotter. Transfer the traffic to dragonfly P2P based on nydus mirror mode and hit the local peer cache.
Hit Nydus Cache: Use containerd to pull image via nydus-snapshotter. Transfer the traffic to dragonfly P2P based on nydus mirror mode and hit the nydus local cache.



Machine Learning Model

KubeCon CloudNativeCon Europe 2023

P2P-based Downloading Models



Product Introduction

KubeCon CloudNativeCon Europe 2023

Container Registry, CR

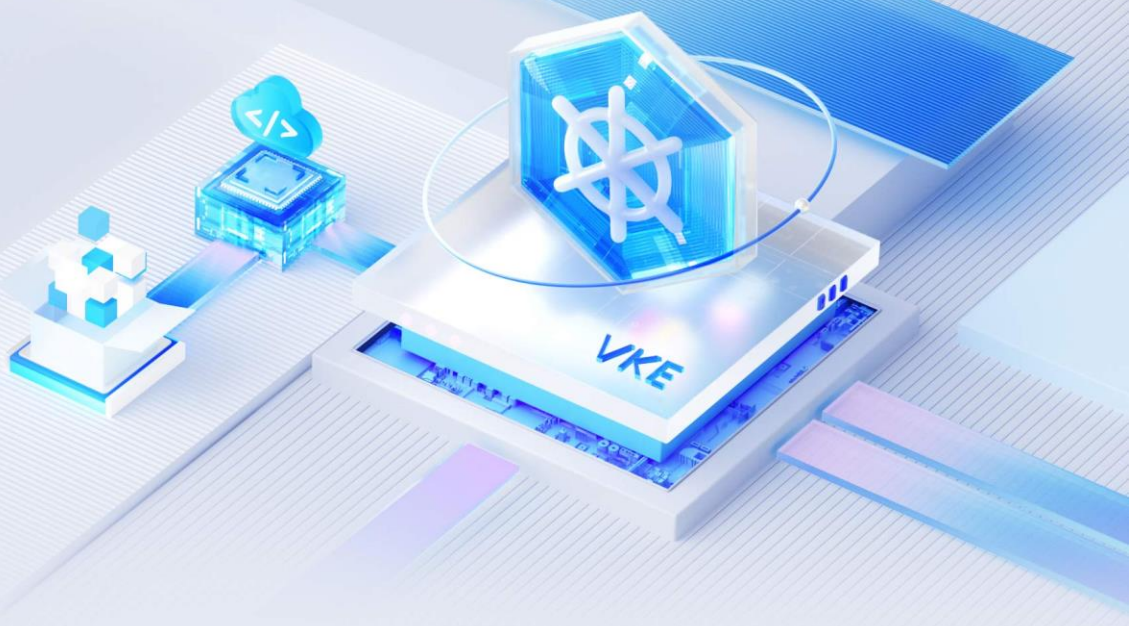
CR is a product of ByteDance's Cloud Service Volcano Engine, which provides secure and highly available container image hosting services to facilitate users to manage the entire lifecycle of container images.

Product Introduction

KubeCon CloudNativeCon Europe 2023

Volcengine Kubernetes Engine, VKE

VKE provides high-performance Kubernetes container cluster management services centered on containers through deep integration of next-generation cloud-native technologies, helping users quickly build containerized applications.



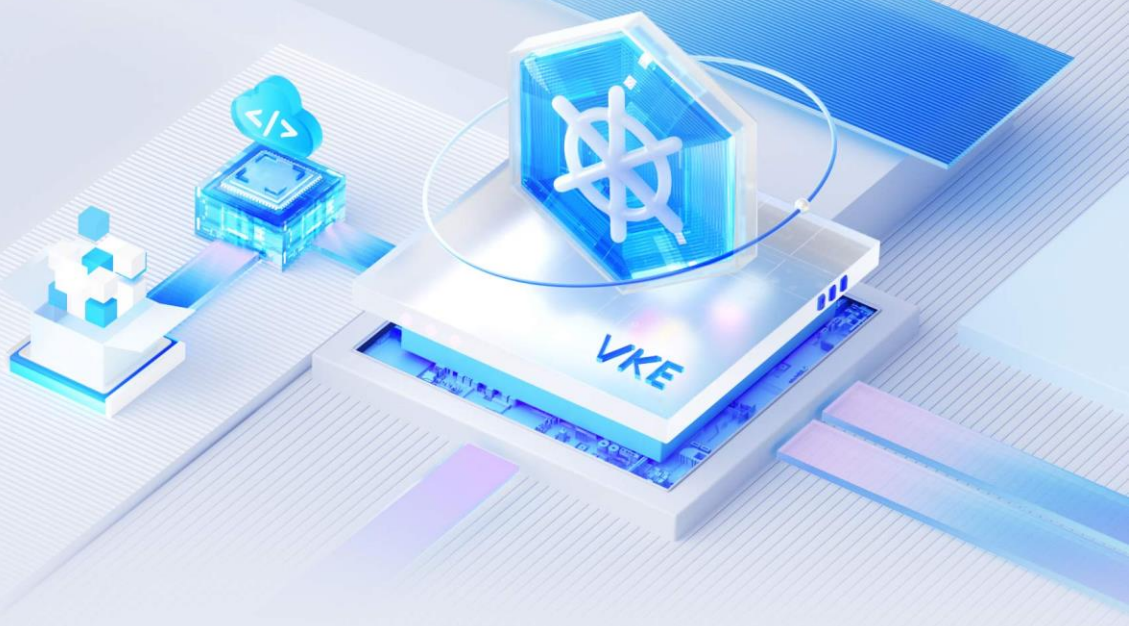
Product Introduction

KubeCon CloudNativeCon Europe 2023

Volcengine Container Instance, VCI

VCI is a serverless and containerized computing service. Currently, VCI can seamlessly integrate the container service VKE to provide Kubernetes orchestration capabilities.

With VCI, users can focus on building the application itself without purchasing and managing infrastructure such as underlying cloud servers, and only pay for the resources consumed by the actual running of the container.



Background: Why we need P2P

1. Bandwidth limit

2. QPS limit

P2P project

Dragonfly VS Kraken

	Dragonfly	Kraken
high availability	Scheduler HA by hashring	Tracker HA by hashring
containerd support	yes	yes
HTTPS artifact registry support	yes	yes
community active	yes	no
users	more	less
production available	yes	yes
Nydus compitable	yes	no
architectural complexity	less	more

Dragonfly is a better option.

Proposal

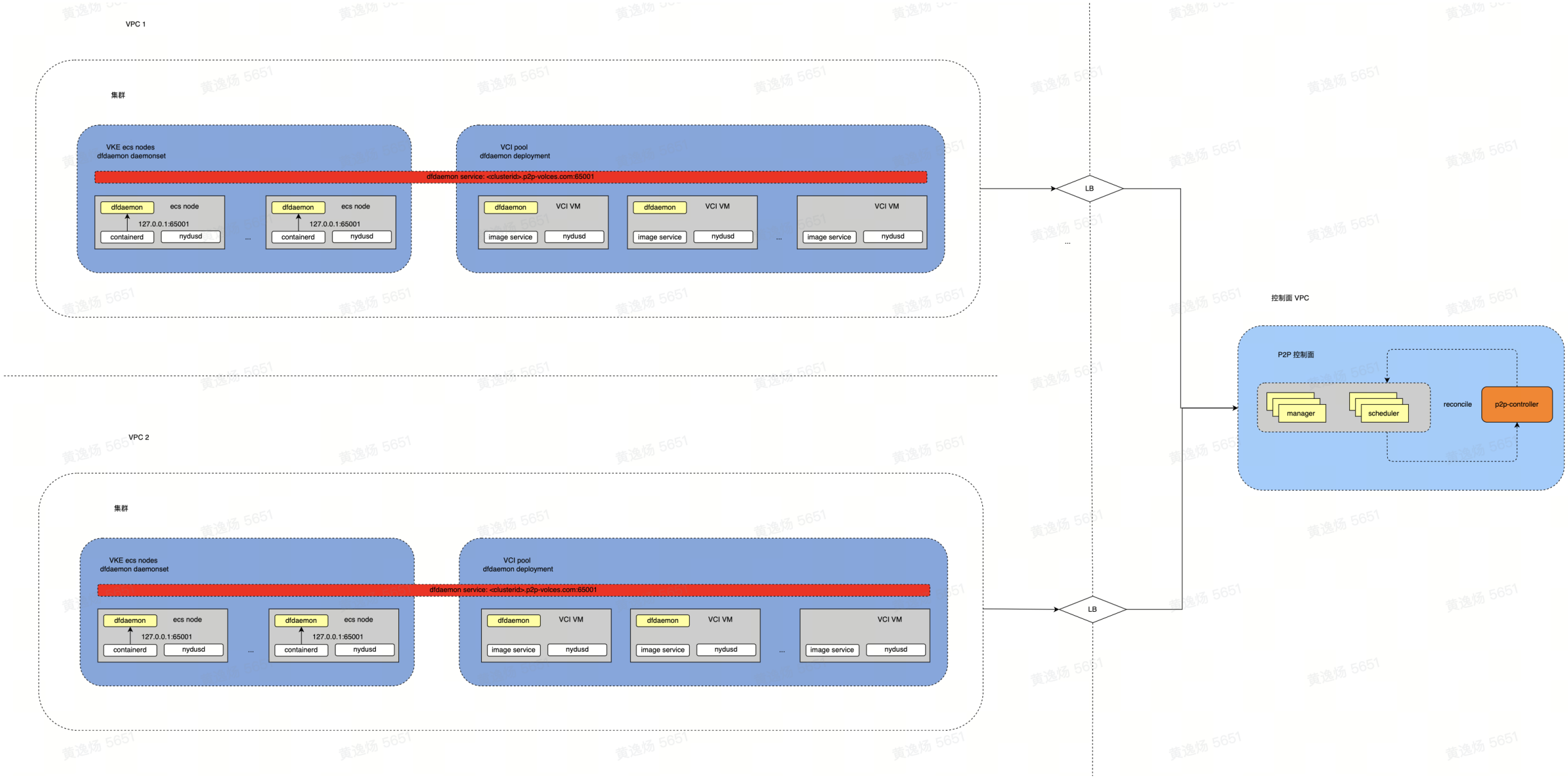
Dragonfly/Nydus in VKE/VCI

1. Dragonfly deployed in VKE

2. Dragonfly Served for VCI

3. Dragonfly Served for Nydus

Architecture



Benchmark Data

KubeCon CloudNativeCon Europe 2023

Environment

CR：bandwith 10Gbit/s
ECS: 4C8G， with local ssd, bandwith 6Gbit/s

Image

Nginx(500M)
TensorFlow(3G)

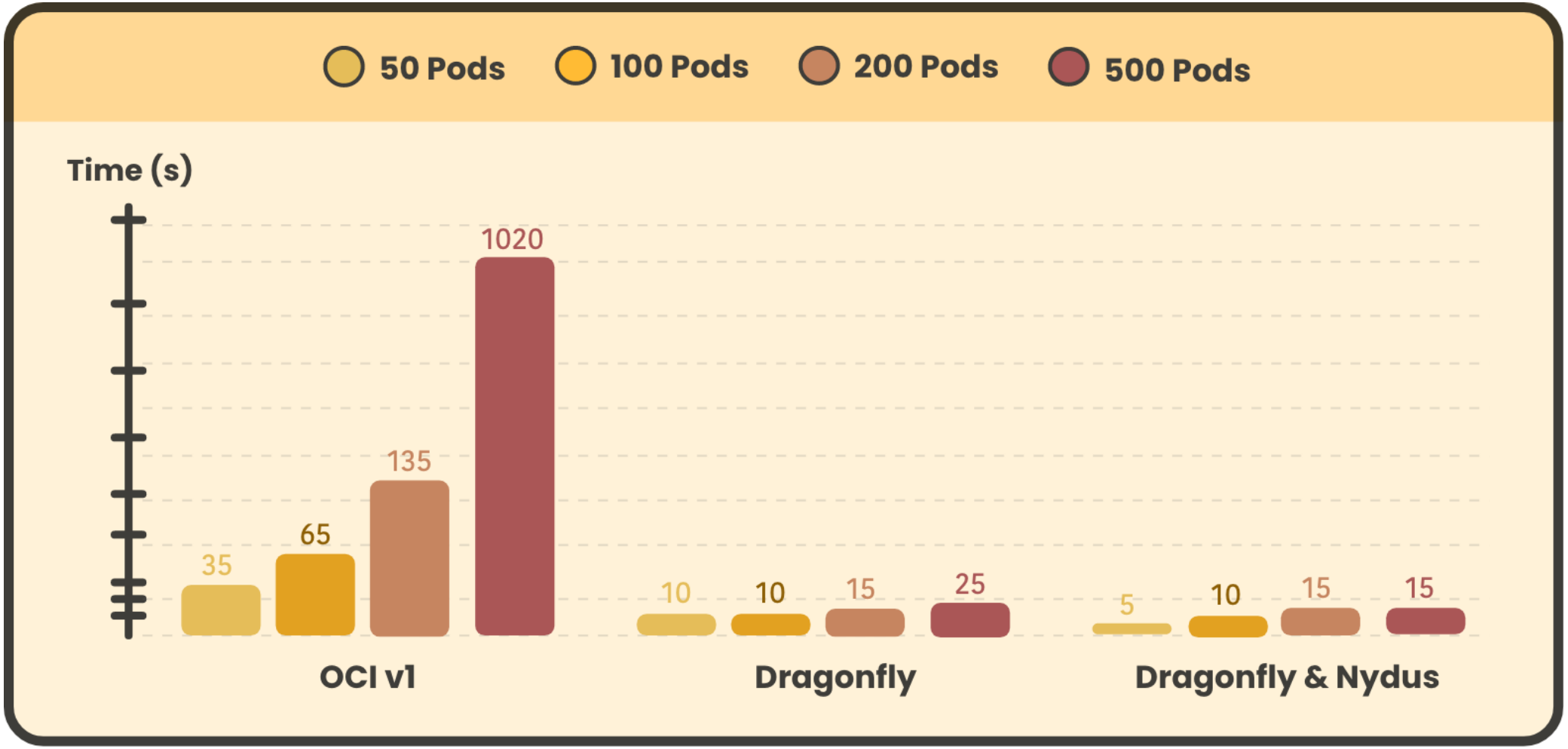
Version

Dragonfly v2.0.8

Quota

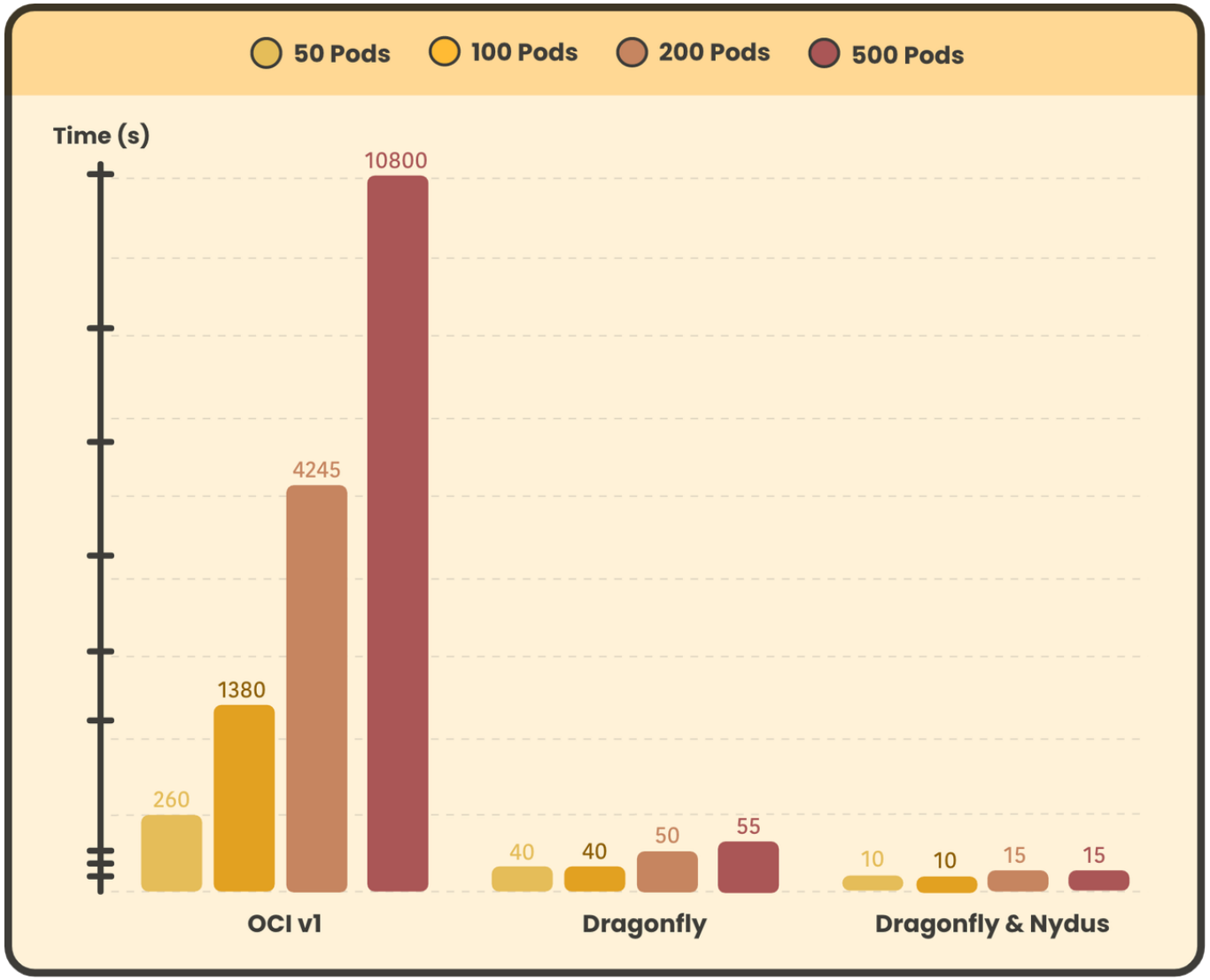
Dfdaemon: Limit 2C6G
Scheduler: 2 Replicas， Request 1C2G， Limit 4C8G
Manager: 2 Replicas， Request 1C2G， Limit 4C8G

Benchmark Data



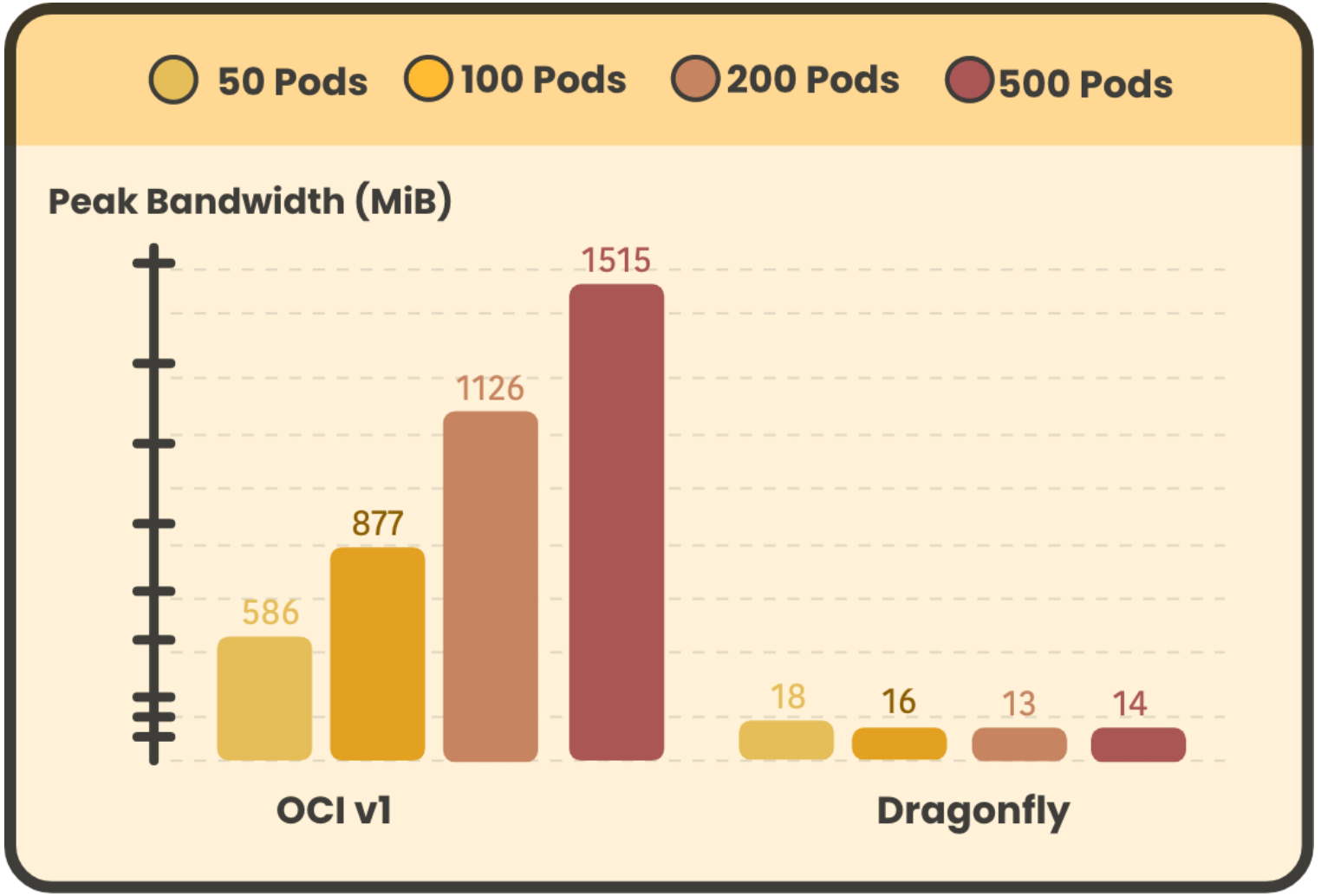
Nginx Pod Creation to Container Start

Benchmark Data



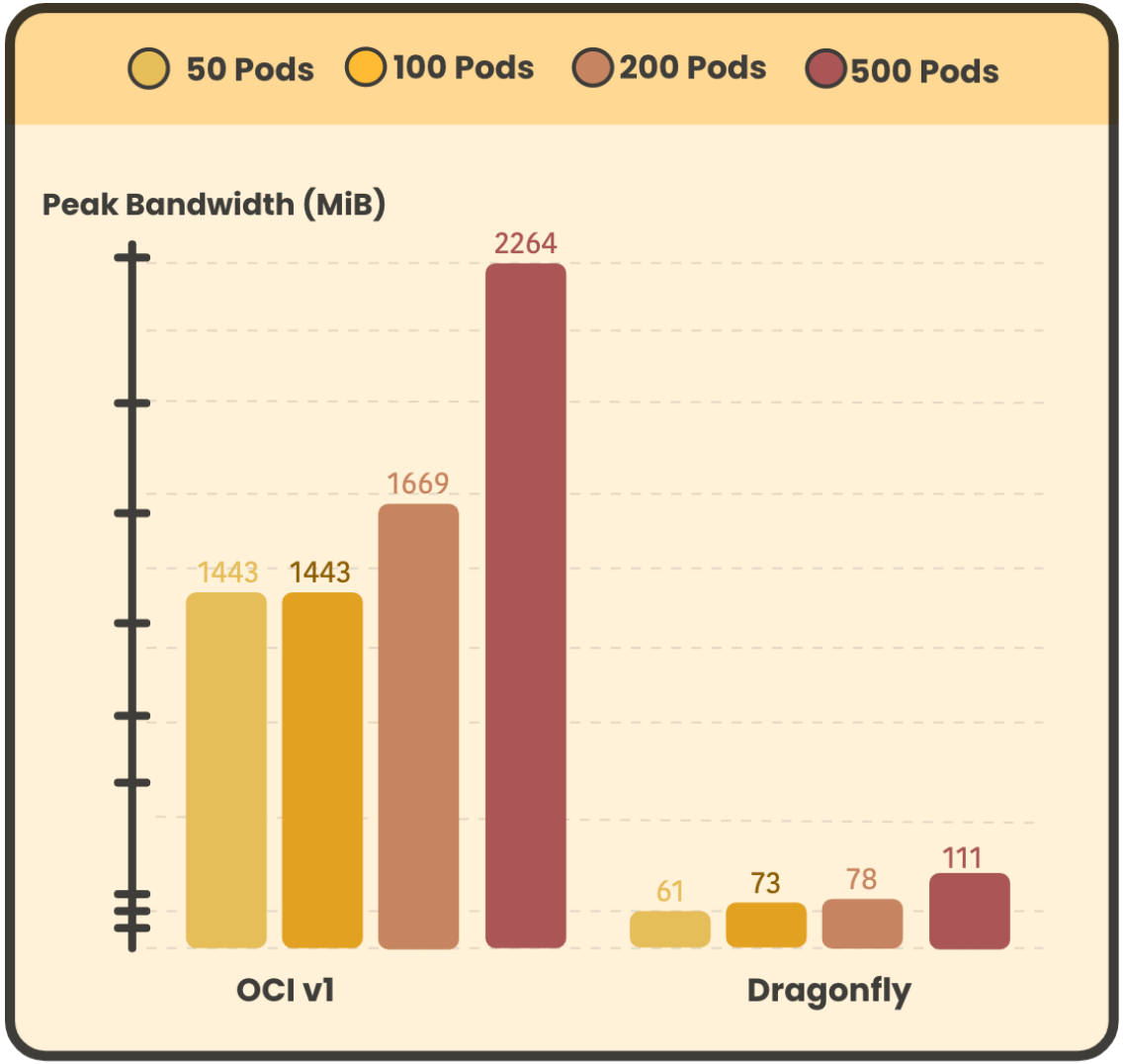
TensorFlow Pod Creation to Container Start

Benchmark Data



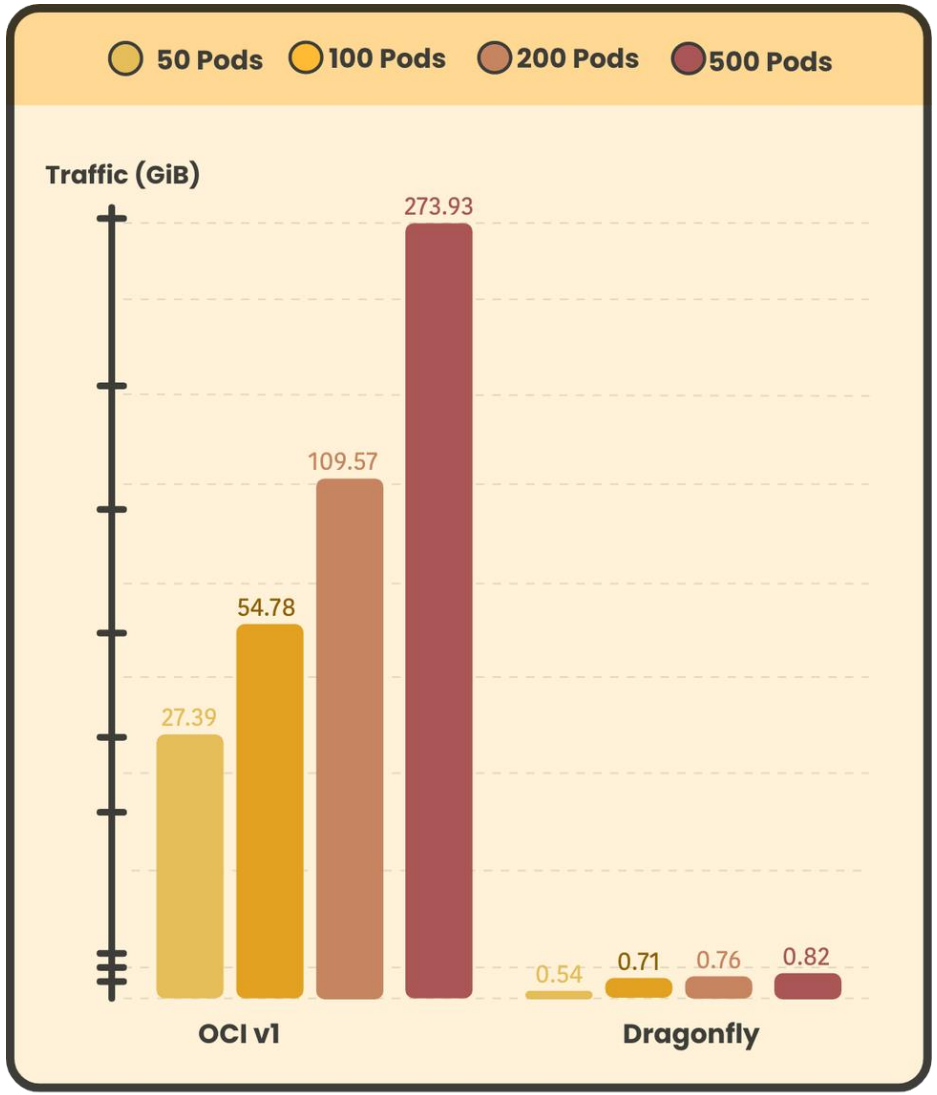
Impact of Nginx on Container Registry

Benchmark Data



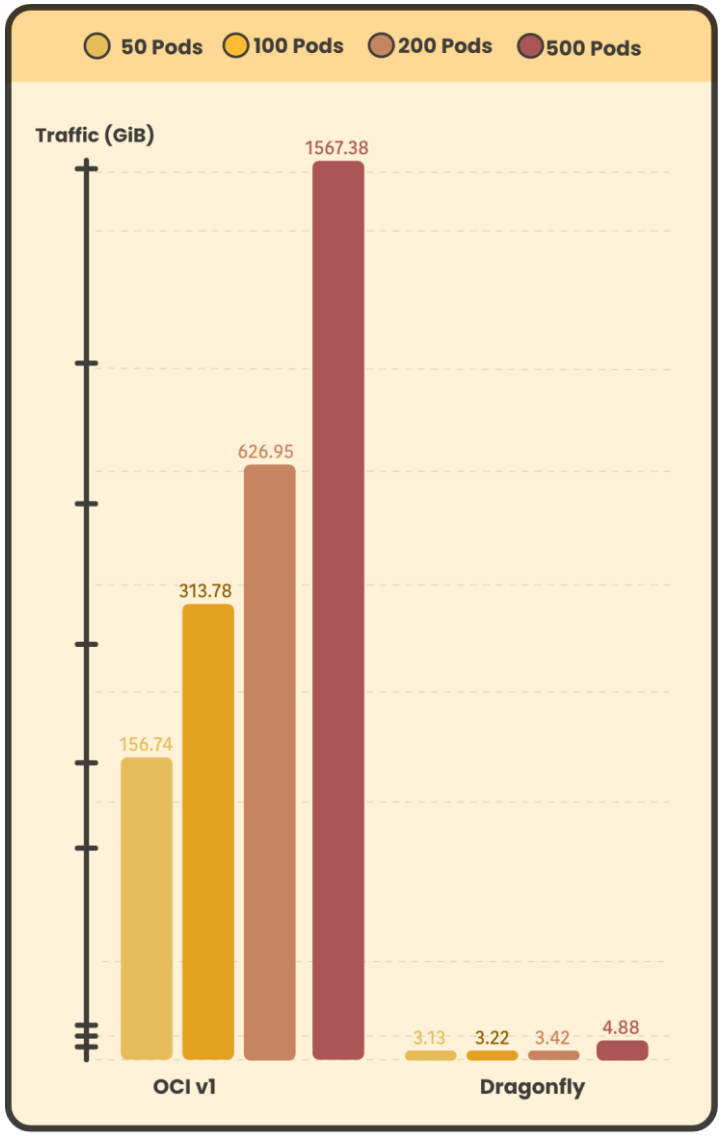
Impact of TensorFlow on Container Registry

Benchmark Data



Impact of Nginx on Container Registry

Benchmark Data



Impact of TensorFlow on Container Registry

THANK YOU!

Dragonfly Website

Dragonfly Github

Dragonfly Twitter

Dragonfly Slack

Nydus Website

Nydus Github

Nydus Slack