KubeCon | CloudNativeCon

Europe 2022

WELCOME TO VALENCIA

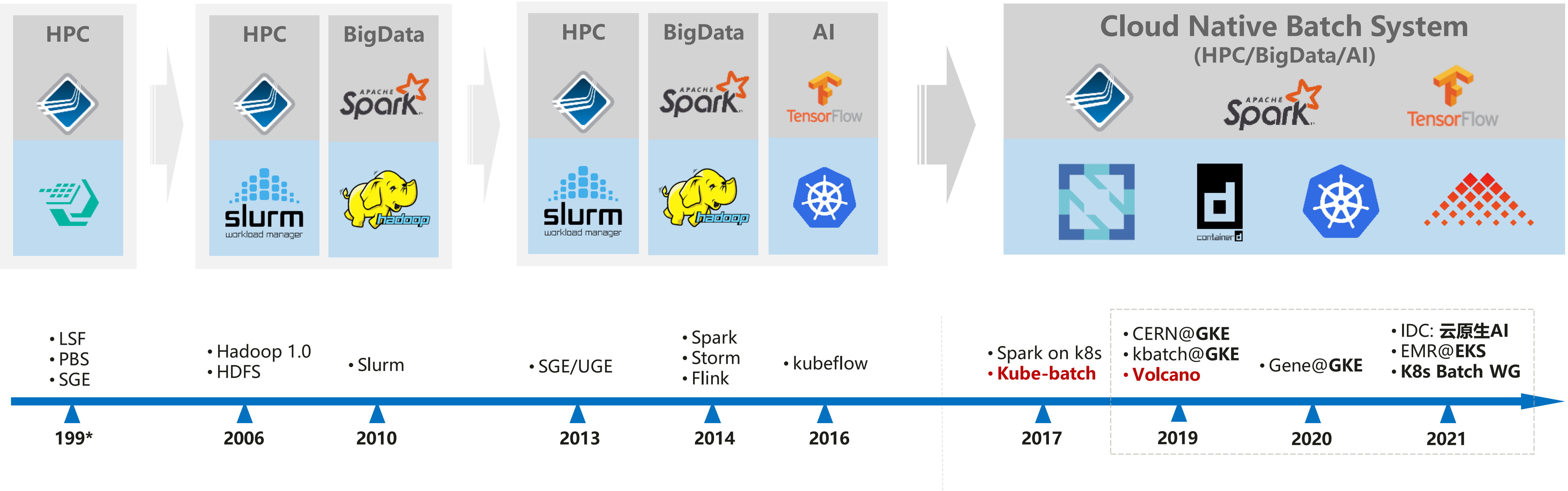
Intro & Deep dive

Volcano: A Cloud Native Batch System

Klaus Ma (@k82cn)

Cloud Native for Intelligent Workload

More and more organization are leveraging cloud native technology to avoid fragmental ecosystem, isolated stack, low resource utilization.



HPC

HPC | BigData

HPC | BigData | AI

Cloud Native Batch System
(HPC/BigData/AI)

199*
- LSF
- PBS
- SGE

2006
- Hadoop 1.0
- HDFS

2010
- Slurm

2013
- SGE/UGE

2014
- Spark
- Storm
- Flink

2016
- kubeflow

2017
- Spark on k8s
- **Kube-batch**

2019
- CERN@**GKE**
- kbatch@**GKE**
- **Volcano**

2020
- Gene@**GKE**

2021
- IDC: **云原生AI**
- EMR@**EKS**
- **K8s Batch WG**

Batch on K8s: Challenges

Job management

- Pod level scheduling, no awareness of upper-level applications.
- Lack of fine-grained lifecycle management.
- Lack of task dependencies, job dependencies.

Scheduling

- Lack of job based scheduling, e.g. job ordering, job priority, job preemption, job fair-share, job reservation.
- Not enough advanced scheduling algorithms, E.g. CPU topology, task-topology, IO-Awareness, backfill.

Multi-framework support

- Insufficient support for mainstream computing frameworks like MPI, Tensorflow, Mxnet, Pytorch.
- Complex deployment and O&M because each framework corresponding to a different operator.

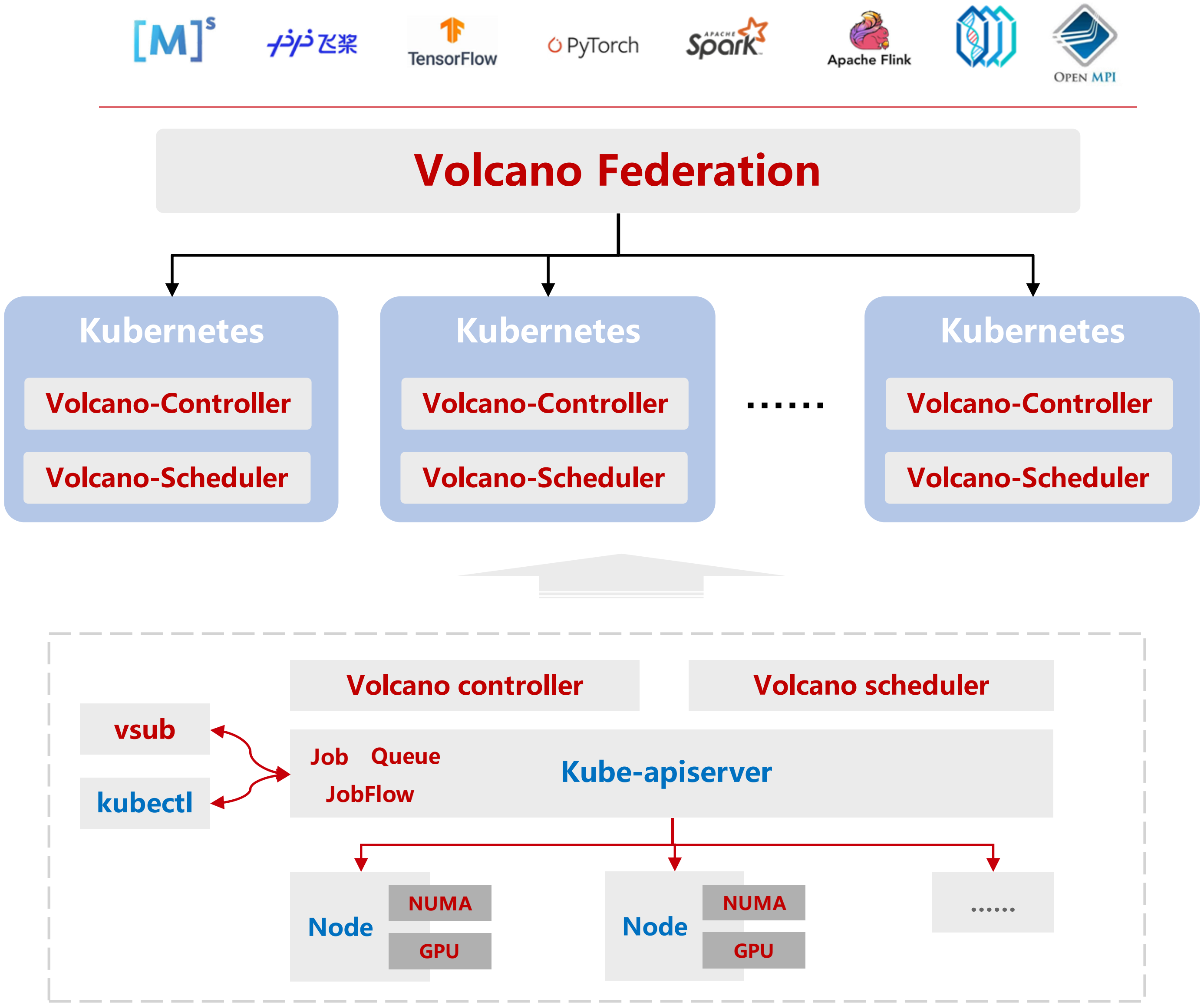
Resource planning, sharing, heterogeneous computing

- Lack of support to resource sharing mechanism between jobs, queues, namespaces.
- Lack of Deeper support on heterogeneous resources.

Performance

- Not enough throughput, roundtrip for batch workload.

Volcano Overview



- Created at March 2019; Sandbox at April 2020; Incubator at April 2022
- 2.3k star， 350+ contributors，latest version v1.5.1
- 50+ enterprises adopt Volcano in production environments.

HUAWEI | aws | Baidu 百度 | Tencent 腾讯
JD.COM 京东 | 滴滴出行 | 小红书 | iQIYI 爱奇艺

Key Concept

Job

- Multiple Pod Template
- Lifecycle management/Error handing

User/namespace/resource quota

- namespace is regarded as user
- resource quota is regarded as the upper limit resource that users in the namespace are able to use at most.

  Like the QPS in Kube-apiserver.

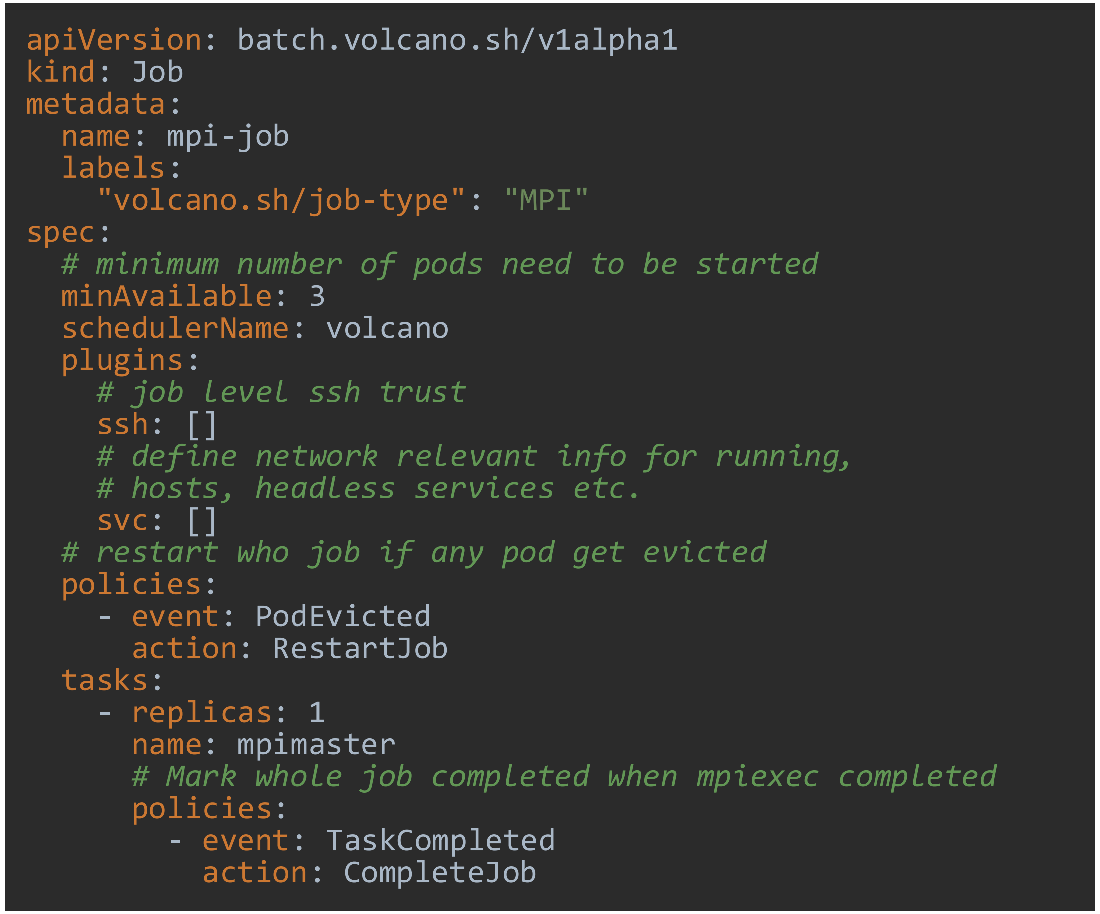
Resource share:

- Use Queue for resource sharing.
- Share resources between different "tenants" or resource pools.
- Support different scheduling policies or algorithms for different "tenants" or resource pools.

Job management

Volcano Job:

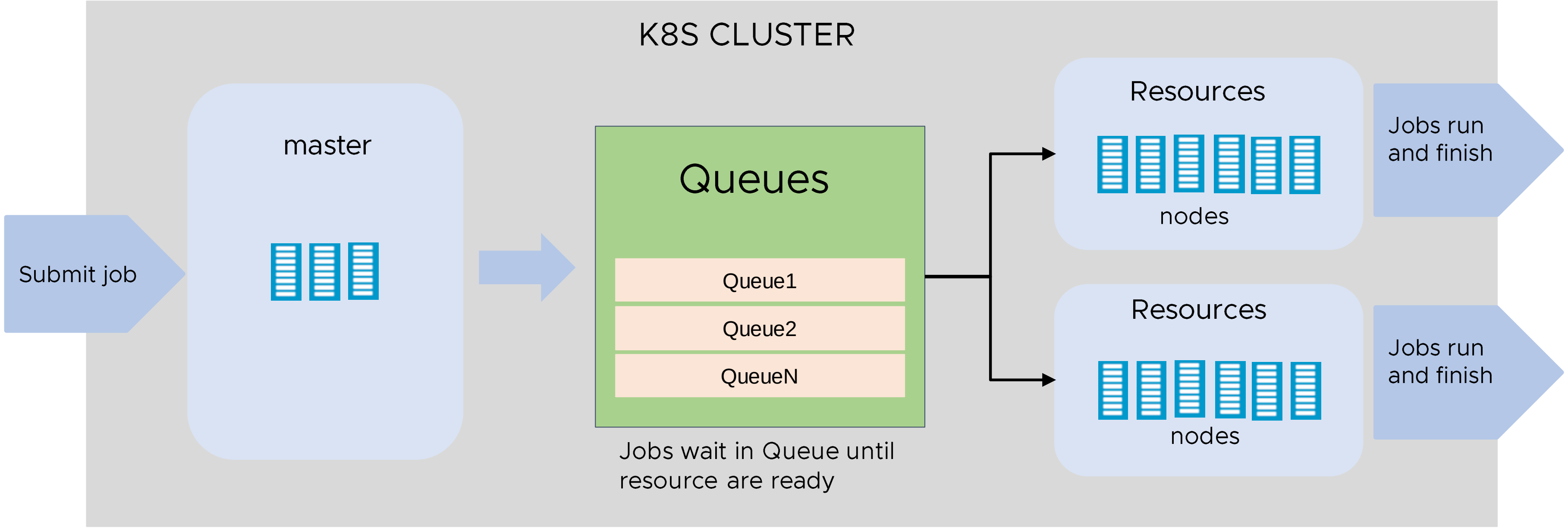
- Unified Job interface for most of batch job like mpi, pytorch, tensorflow, mxnet, etc.
- Fine-grained Job Lifecycle management
- Extendable job plugin
  - Env
  - Svc
  - Ssh
  - Tensorflow
- Coordinate with Scheduler
- Job dependency



```yaml
apiVersion: batch.volcano.sh/v1alpha1
kind: Job
metadata:
  name: mpi-job
  labels:
    "volcano.sh/job-type": "MPI"
spec:
  # minimum number of pods need to be started
  minAvailable: 3
  schedulerName: volcano
  plugins:
    # job level ssh trust
    ssh: []
    # define network relevant info for running,
    # hosts, headless services etc.
    svc: []
  # restart who job if any pod get evicted
  policies:
    - event: PodEvicted
      action: RestartJob
  tasks:
    - replicas: 1
      name: mpimaster
      # Mark whole job completed when mpiexec completed
      policies:
        - event: TaskCompleted
          action: CompleteJob
```

Resource management- Queue

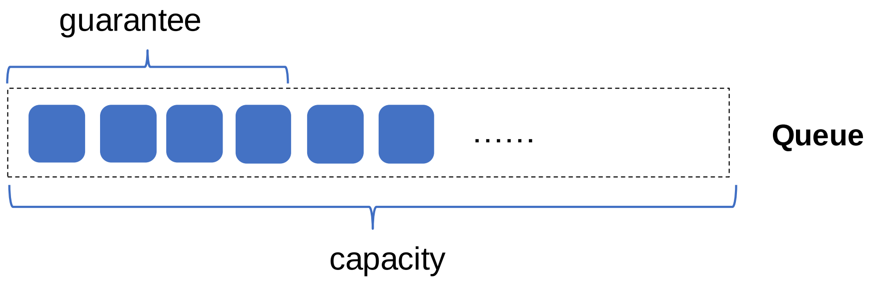
- Queue is cluster scoped, decoupled with user/namespace.
- Queue is used to share resources between “multi-tenants” or resource pool.
- Configure policy for each queue, e.g. FIFO, fair share, priority, SLA.

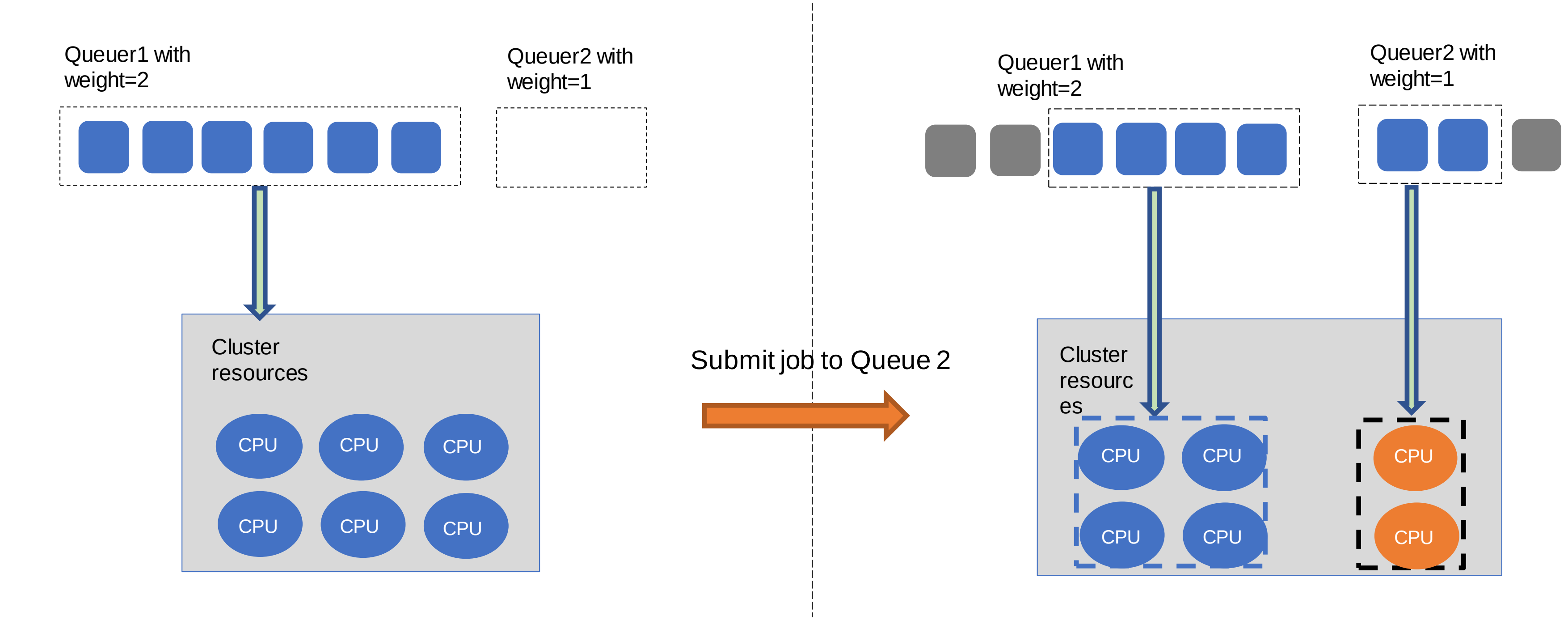


Jobs wait in Queue until resource are ready

Dynamic resource sharing between queues

- Queue Guarantee/Capacity
- Share resource between Queues proportionally by weight



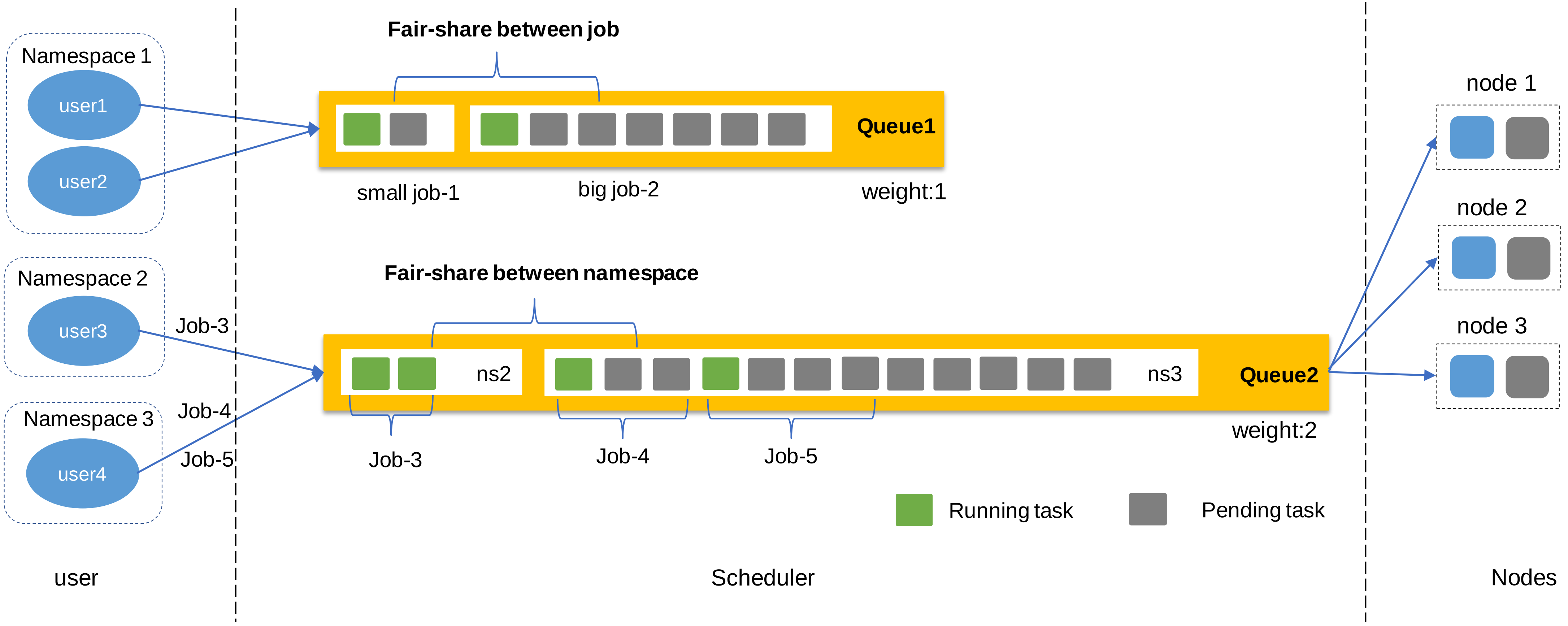


Queue2 is empty. Q1 can borrow resources from Queue2.

Queue2 has workload, it will reclaim resources from Queue1.

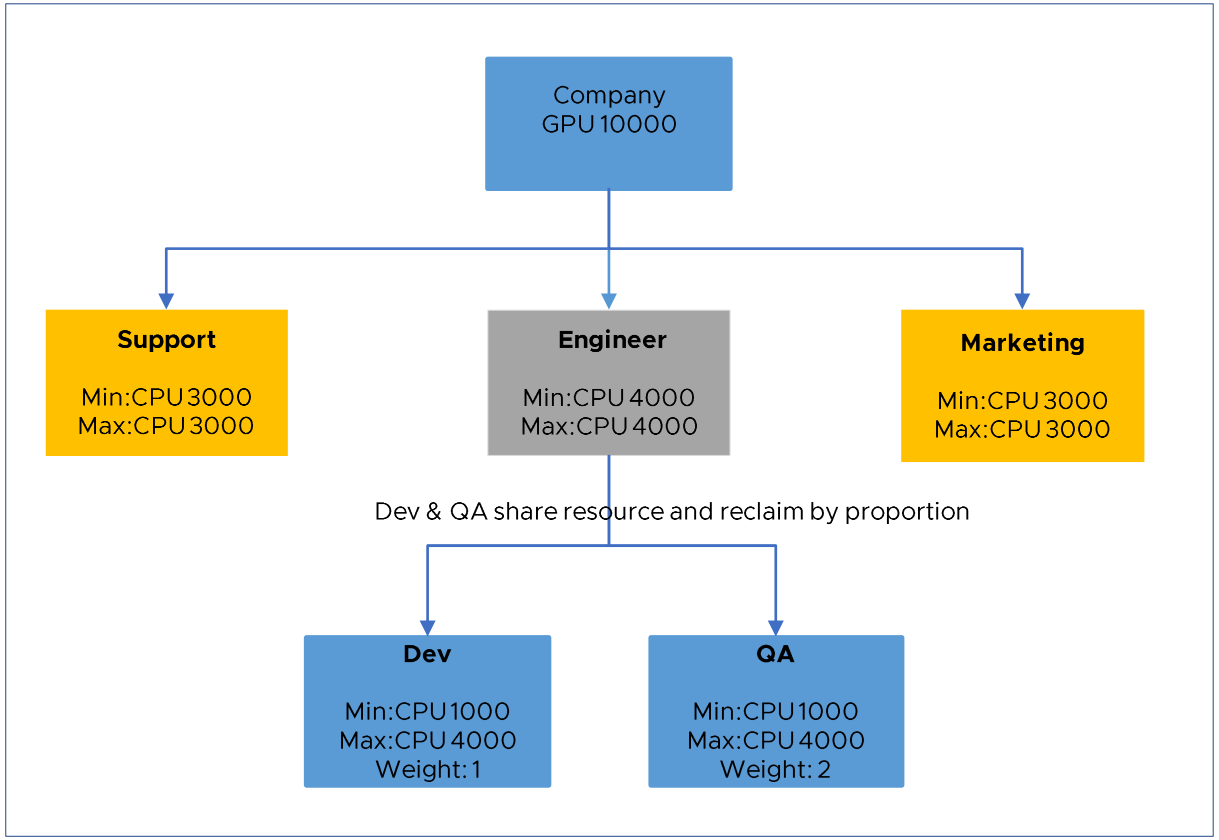
Fair share within Queue

- Sharing resource between jobs
- Sharing resource between namespaces
- Per-Queue policy (FIFO, Priority, Fair share , …)



Case: hierarchical queue

- How to share resource in a multi-level org more easily?
- Problem: flat queue can not meet complex resource share and isolation easily for big org



D0m021ng and qiaoshuangshuang from Baidu Company is contributing this feature.

Solution

- Multiple level queue constructs a tree which is mapped to the org.
- Each level queue has min, max, weight. Use max to isolate resource, use queue weight to balance resource between queues.
- Share resources between queues and reclaim by weight

Benefit：

- Flexible resource management, easy to map the organization
- fine-grained control resource share and isolation for a big multi-tenants organization
- The queue min capacity ensures guaranteed resource, the proportion by weight offers flexible sharing

Scheduling policy in Volcano

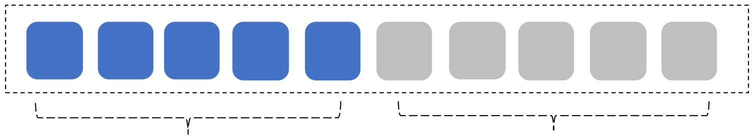
- Gang-Scheduling
- Job priority
- Job queue
- Job order
- Preemption
- backfill
- Job Fair-share
- Namespace fair-share
- Task-topology
- IO-Awareness
- Resource reservation

- SLA
- GPU sharing
- NUMA-Awareness
- HDRF
- Hierarchy Queue
- Co-location
- Elastic scheduling
- TDM
- Proportional scheduling
- ……

Scenario: Elastic scheduling

- **What is elastic job**

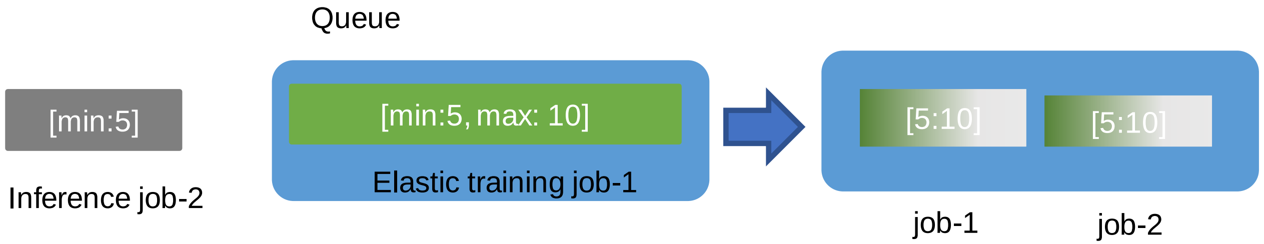
```
apiVersion: batch.volcano.sh/v1alpha1
kind: Job
metadata:
  name: test-job
spec:
  minAvailable: 5 #min
  tasks:
    - replicas: 10  #max
      template:
       spec:
        containers:
          - image: train_script
            resources:
                nvidia.com/gpu: 1
… …
```



Scenario: hybrid elastic job and parallel job in one cluster to improve utilization

Example:

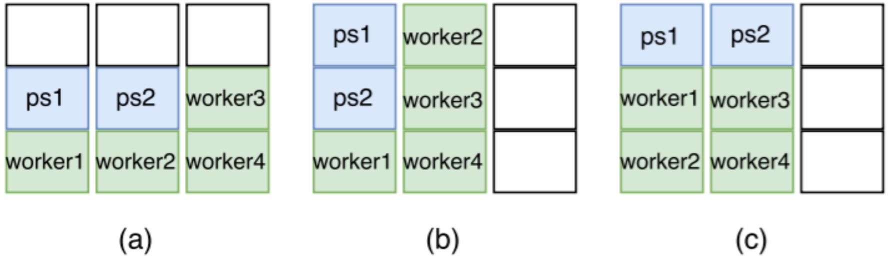
- inference job-2 (high priority) preempt elastic pod from training job-1 with low priority to ensure it's SLA.
- The elastic training job-1 get more Pod scheduled when there is free resource

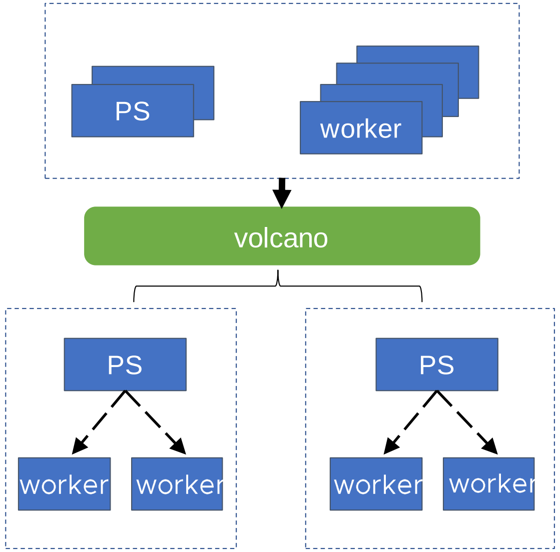


Qiankunlli (github id) from Ximalaya Company ontributed this feature.

Scenario: Topology awareness

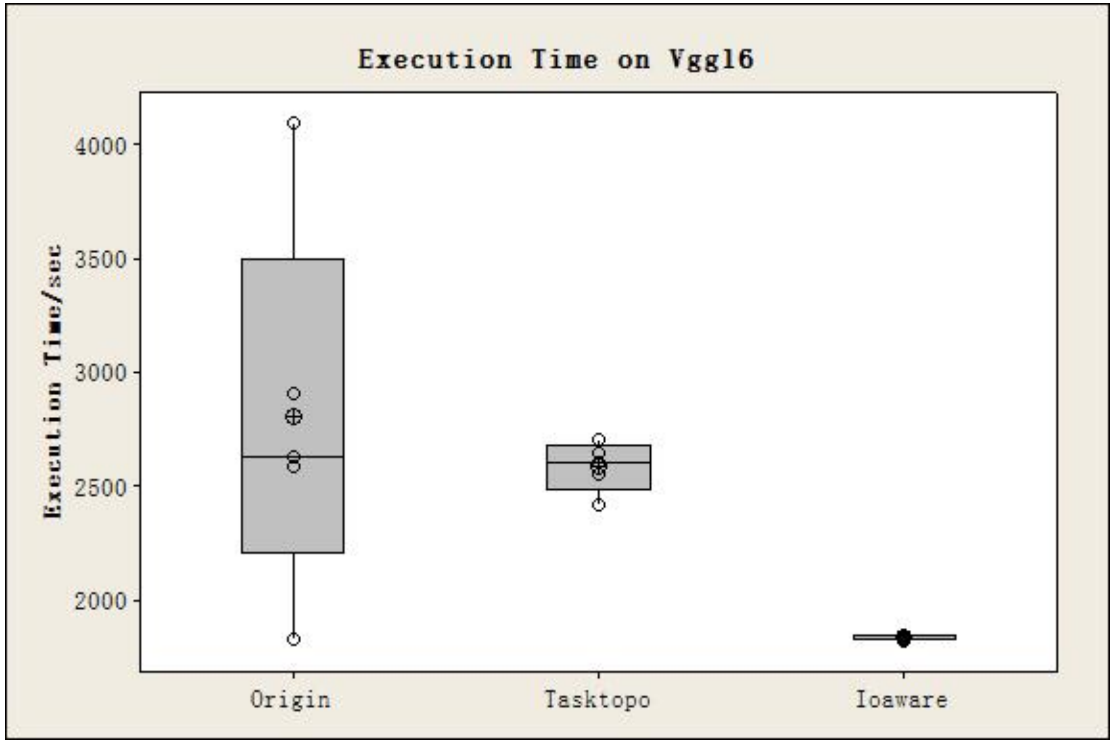
- The execution time of 3 jobs in total; 2ps + 4workers for each job in k8s





Task-topology scheduling

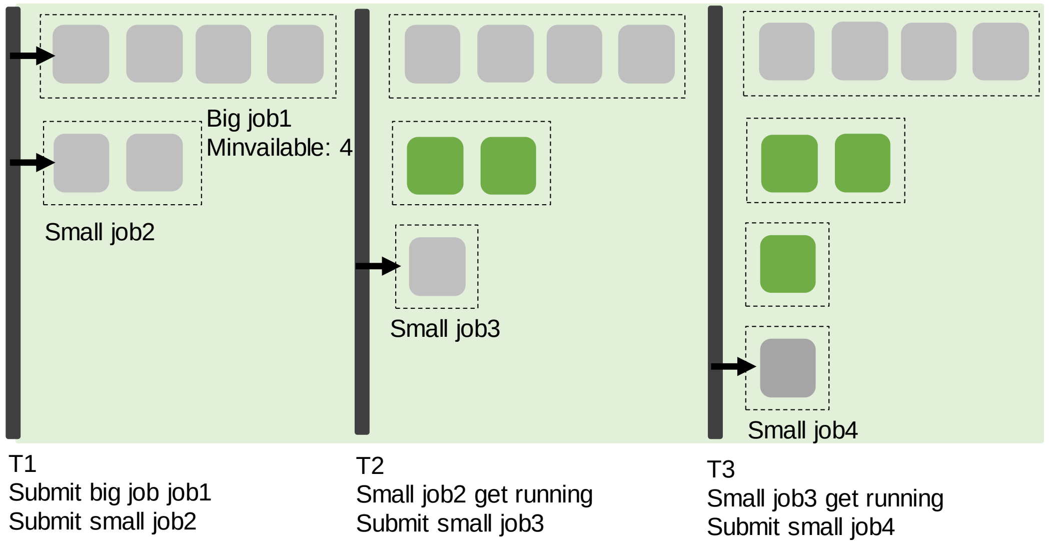
- IO-Awareness: minimized the maximum of communication cost between any two nodes.



Scenario: SLA scheduling

- Scenario: Big job get starving while co-existing with small job.

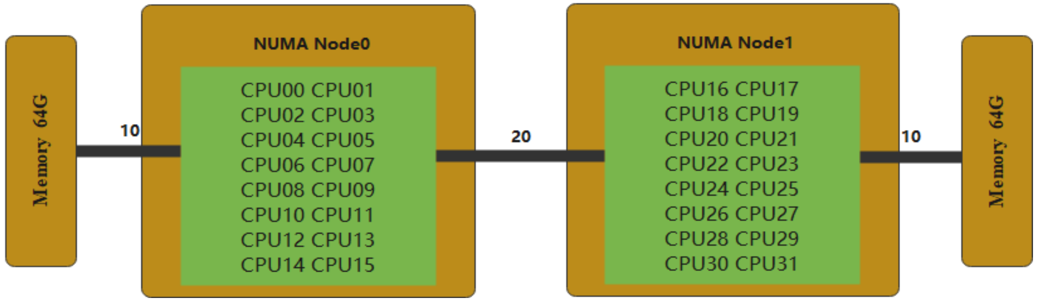
Job Stream

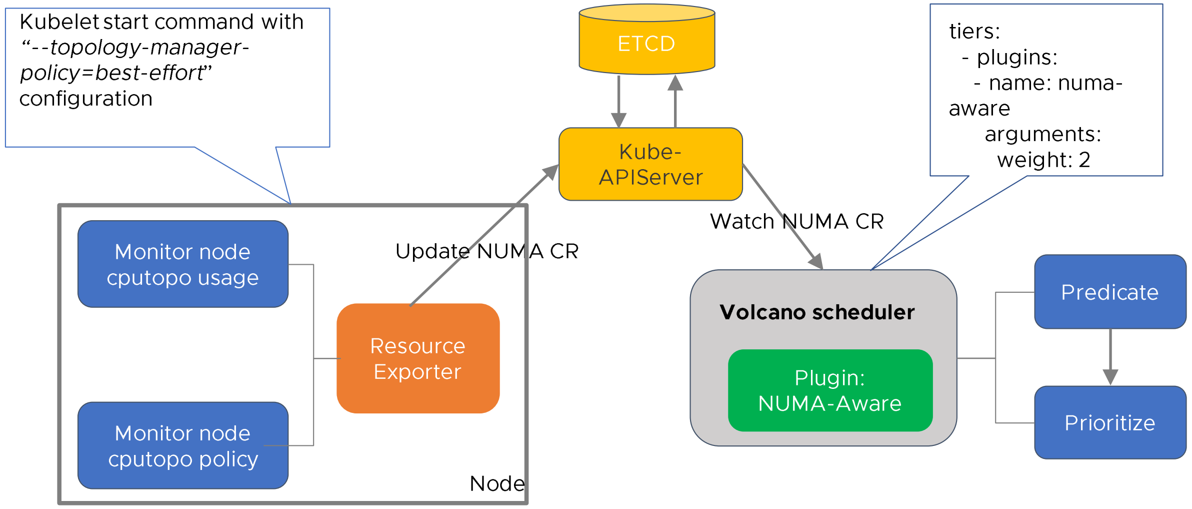


- SLA scheduling allow to configure the job so that it is completed on time and reduce the risk of missed deadlines.
- SLA support argument `sla-waiting-time` to realize job resource reservation: `sla-waiting-time` is the maximum time that one job should stay in pending. When `sla-waiting-time` reached, SLA plugin move the pending job to next state. And start to reserve resources for this job until the job’s request is satisfied.

```
apiVersion: batch.volcano.sh/v1alpha1
kind: Job
metadata:
  name: test-job
  annotations:
    sla-waiting-time:1h
spec:
  minAvailable: 5
  tasks:
    - replicas: 5
      template:
 ... ...
```

Scenario: CPU Topology awareness





Kubelet start command with *"--topology-manager-policy=best-effort"* configuration

```
tiers:
  - plugins:
    - name: numa-aware
      arguments:
        weight: 2
```

NUMA-Aware scheduling

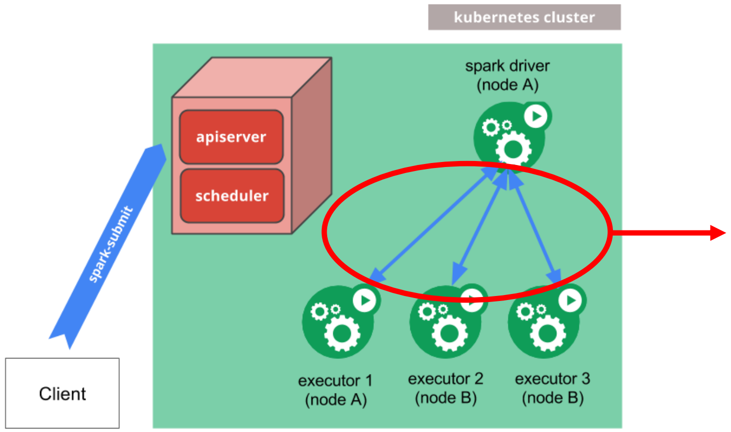
NumaAware

Volcano watch CPU topology and schedule pods to the nodes which NUMA topology.

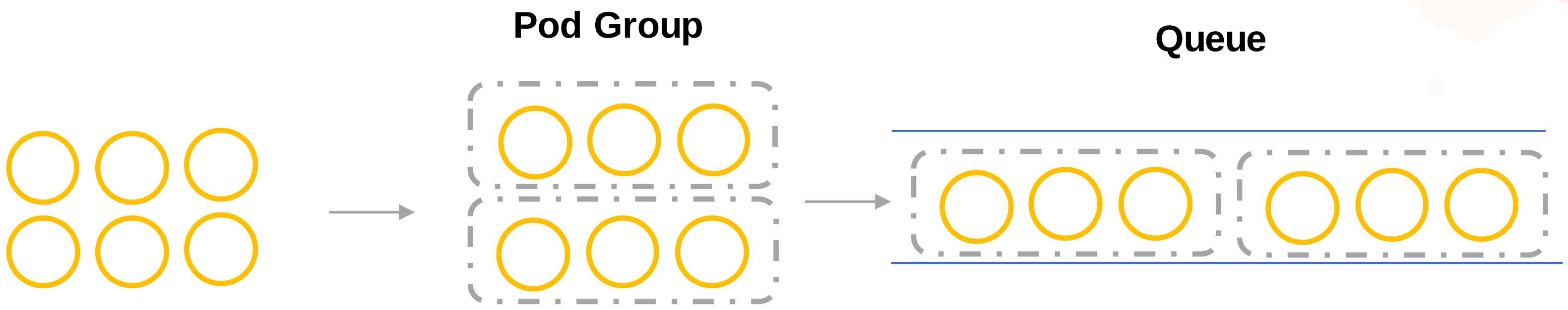
Scenario

Scientific calculation, video decoding, animation rendering, big data offline processing and other specific scenes which are computation-intensive jobs that are sensitive to CPU parameters, scheduling delays.

Scenario: batch scheduler for Spark

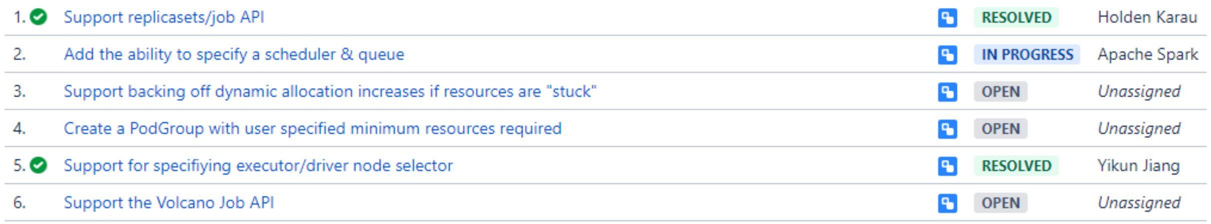


More k8s native resource support



- Schedule based on **Pod Group**
- Minimal Resource reservation (CPU/MEM)
- Job priority

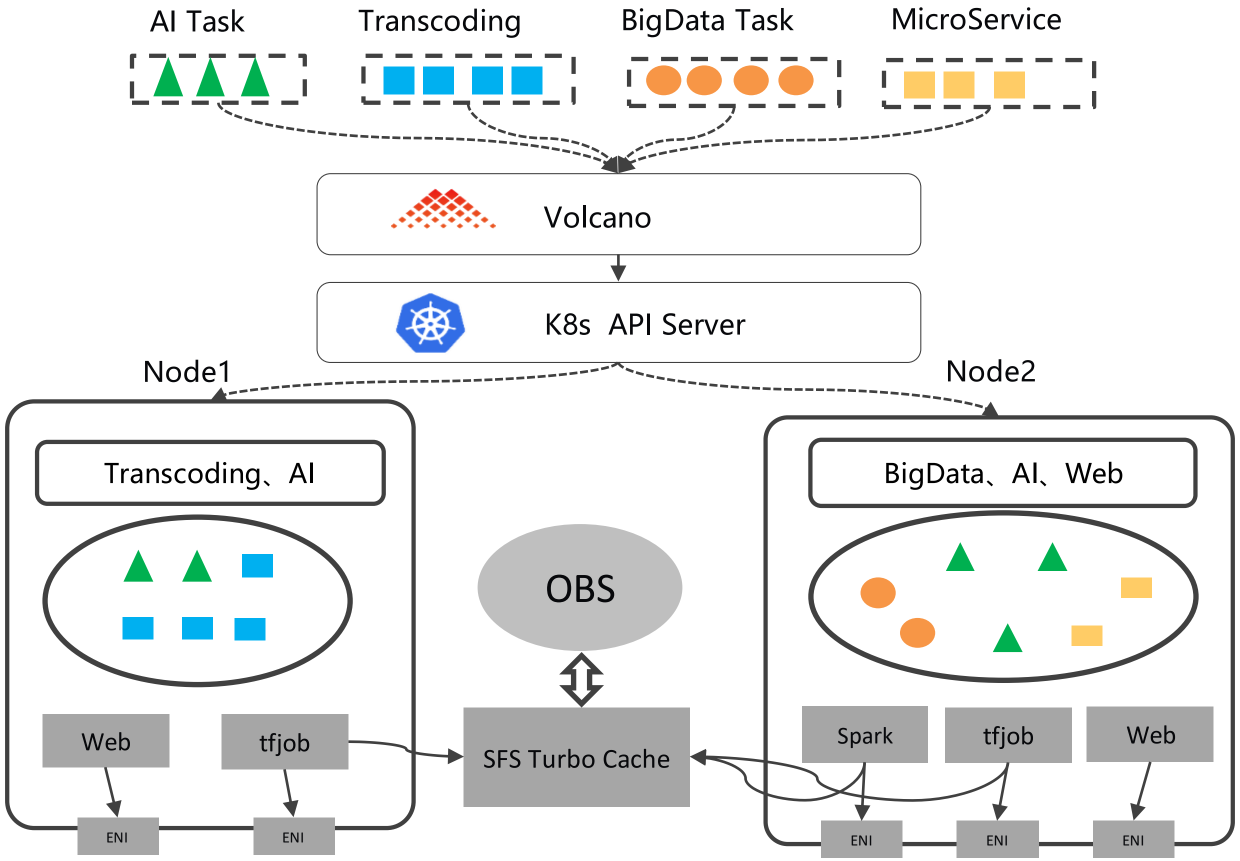
- Introduce Queue
- Multi-tenants
- Priority
- Fair-share
- Preempt
- ……



1. ✓	Support replicasets/job API	RESOLVED	Holden Karau
2.	Add the ability to specify a scheduler & queue	IN PROGRESS	Apache Spark
3.	Support backing off dynamic allocation increases if resources are "stuck"	OPEN	*Unassigned*
4.	Create a PodGroup with user specified minimum resources required	OPEN	*Unassigned*
5. ✓	Support for specifiying executor/driver node selector	RESOLVED	Yikun Jiang
6.	Support the Volcano Job API	OPEN	*Unassigned*

SPARK-36057: Support volcano/alternative schedulers

Scenario: Co-location and oversubscription



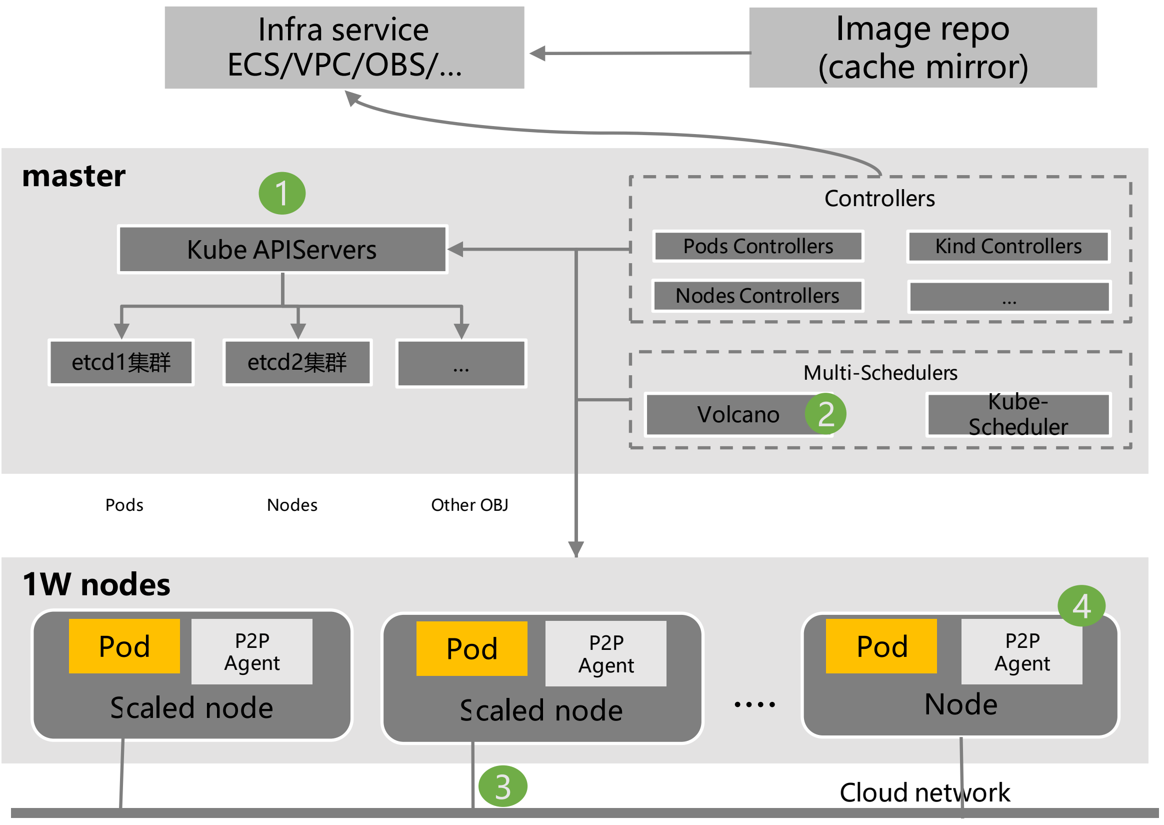
Scenario and requirement：

- There is peak and valley for online service，request is set based on the peak for system stability
- Some users do not know the resource usage of their service, request for excessive resources.
- High allocation rate, low usage，CPU average utilization less than 15%

Key technical

- qos-feature：Qos support in OS level
  - qos/CPU fast preemtion
  - Qos Memory reclaim with priority
  - qos/NET&IO monitor and isolocation
- kubelet：Update/Change Qos model，support resource report realtime, compute oversubscription resource, evict job.
- scheduler：Support multiple types of workload together.

Scenario: Big scale cluster: 10k nodes, 1 million Pod



Container ochistration:

- Dedicated ETCD, e.g. Event stored on separated DB，wal/snapshot mounted separately.
- multiple controller-manager instances
- Kube-apiserver scaling based on biz

Scheduler：

- improve throughput of single instance by EquivalenceCache, pruning algorithm, batch bind etc.
- Support 1.5k pod/s throughput
- multiple instances based on shared resource view.

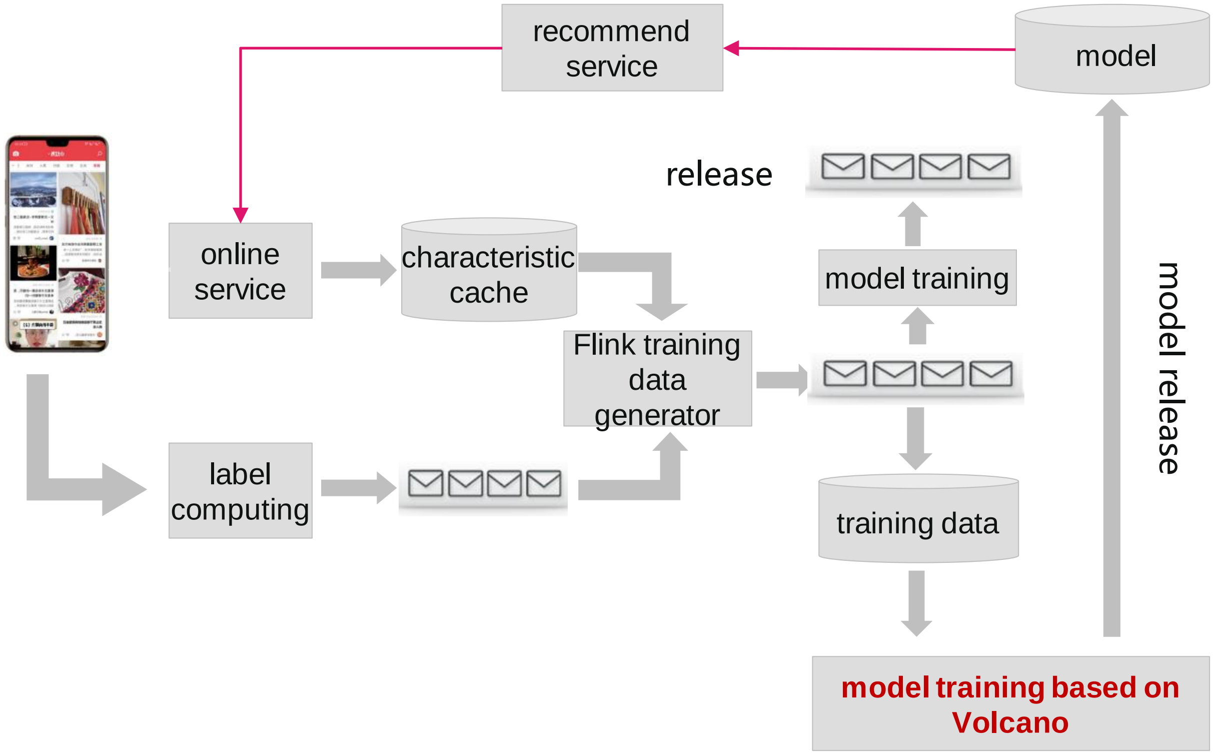
Network:

- increase density of container in each node and ENI capacity.
- improve dispatching efficiency of network interface by pre-apply for network interface for Warm Pool
- support big scale app based on eBPF/XDP, e.g. network policy

Container engine:

- optimization for containerd startup
- Lazy loading

Use Case: AI platform at Xiaohongshu



*Based on these pain points, we did some research and found Volcano, **which can completely solve our pain points**. Therefore, we also participated in the Kubernetes batch community and became a loyal user of volcano.*

-- Yi Guo(Tech-Lead at Xiaohongshu)

About Xiaohongshu

- Top social media and e-commerce platform with over 100 million active users per month
- Recommendation is one of core business. AI platform at Xiaohongshu consists of online and offline training system, which undertakes hundreds of thousands samples analysis and model training. The model generation has already be on the minute scale.

Challenge

- Training cluster with thousands of nodes
- Recommendation model with nearly 100 billion parameters
- A single training task contains hundreds of PS and workers
- Require best topology scheduling and performance

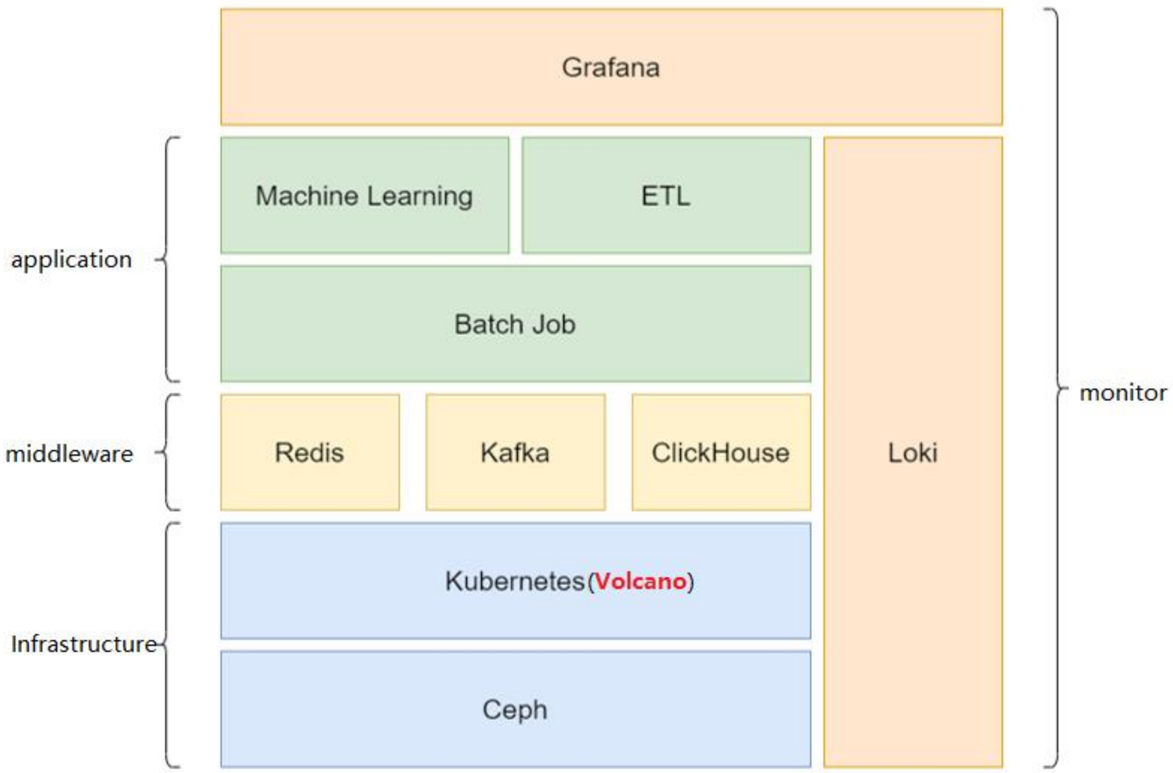
Solution

- Adoption of binpack and task-topology scheduling policy
- Adoption of gang-scheduling/SLA

Benefit

- 20% increase on AI training speed overall
- 20% increase on AI training throughput
- Starvation prevention on big jobs

Use Case: Batch at RuiTian investment



Background:

- Top financial investment company in China
- AI training and reasoning
- Big data ELT & batch offline tasks

Scenarios:

- Fair scheduling for resource sharing among multiple teams
- gang-scheduling to avoid dead lock
- support mixed scheduling for AI, Big data, batch tasks

Benefit:

- Volcano for all business online
- Queue and Fair scheduling policies satisfy all requirements
- More than 300k pods scheduled per day in production

“Volcano is based on Kubernetes, and its robust job scheduling and control policies meet all of our requirements.
*Its simple architecture was **a major reason** we decided to migrate our scheduling platform **from Yarn to Kubernetes**.”*

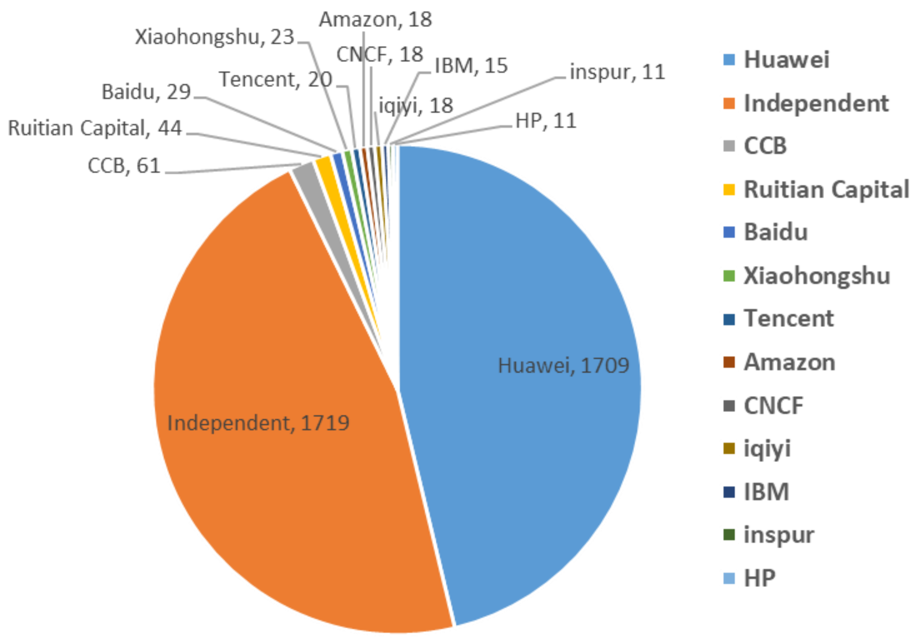
-- Yunzhi Cheng, Director of Engineering at Ruitian Investment

Ref: https://www.cncf.io/blog/2021/03/26/how-ruitian-used-volcano-to-run-large-scale-offline-hpc-jobs/

Community

HUAWEI
aws
Bai du 百度
百度一下 你就知道
Tencent 腾讯
JD.COM 京东
滴滴出行
滴滴一下 美好出行
小红书
全世界的好东西
iQIYI 爱奇艺
悦 享 品 质
唯品会
品 牌 特 卖
趣头条
让阅读更具价值
BOSS直聘
锐天
RUITIAN CAPITAL
SHAREit
建信金融科技
CCB Fintech
云知声
Unisound
vivo
BIBDR
北京大数据研究院
leinao 中科类脑
GrandOmics 希望组
kt NexR

Volcano code diversity in recent 1-year

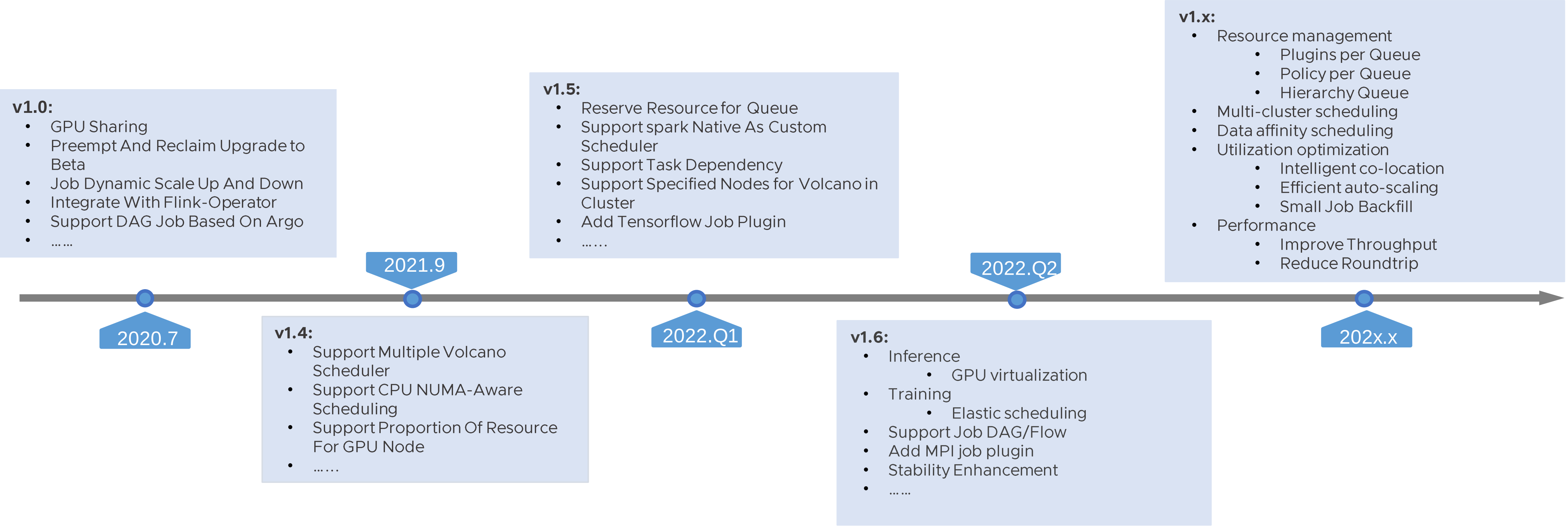


Data from https://volcano.devstats.cncf.io

CLOUD NATIVE
COMPUTING FOUNDATION

Release

Volcano: A Kubernetes Native Batch System



2020.7

v1.0:

- GPU Sharing
- Preempt And Reclaim Upgrade to Beta
- Job Dynamic Scale Up And Down
- Integrate With Flink-Operator
- Support DAG Job Based On Argo
- ……

2021.9

v1.4:

- Support Multiple Volcano Scheduler
- Support CPU NUMA-Aware Scheduling
- Support Proportion Of Resource For GPU Node
- ……

2022.Q1

v1.5:

- Reserve Resource for Queue
- Support spark Native As Custom Scheduler
- Support Task Dependency
- Support Specified Nodes for Volcano in Cluster
- Add Tensorflow Job Plugin
- ……

2022.Q2

v1.6:

- Inference
  - GPU virtualization
- Training
  - Elastic scheduling
- Support Job DAG/Flow
- Add MPI job plugin
- Stability Enhancement
- ……

202x.x

v1.x:

- Resource management
  - Plugins per Queue
  - Policy per Queue
  - Hierarchy Queue
- Multi-cluster scheduling
- Data affinity scheduling
- Utilization optimization
  - Intelligent co-location
  - Efficient auto-scaling
  - Small Job Backfill
- Performance
  - Improve Throughput
  - Reduce Roundtrip

Resource

- Website: https://volcano.sh/en/
- Github**:** https://github.com/volcano-sh
- Slack channel**:** https://volcano-sh.slack.com

Join our community and give us feedback!