KubeCon | CloudNativeCon
North America 2021

RESILIENCE REALIZED

Capacity Scheduling for Elastic Resource Sharing in Kubernetes

Qingcan Wang *Alibaba*
Yuan Chen *Apple*

About us

Qingcan Wang
Software engineer, Alibaba Cloud
Github: denkensk
LinkedIn: qingcan-wang-24aa21b6
qingcan.wqc@alibaba-inc.com

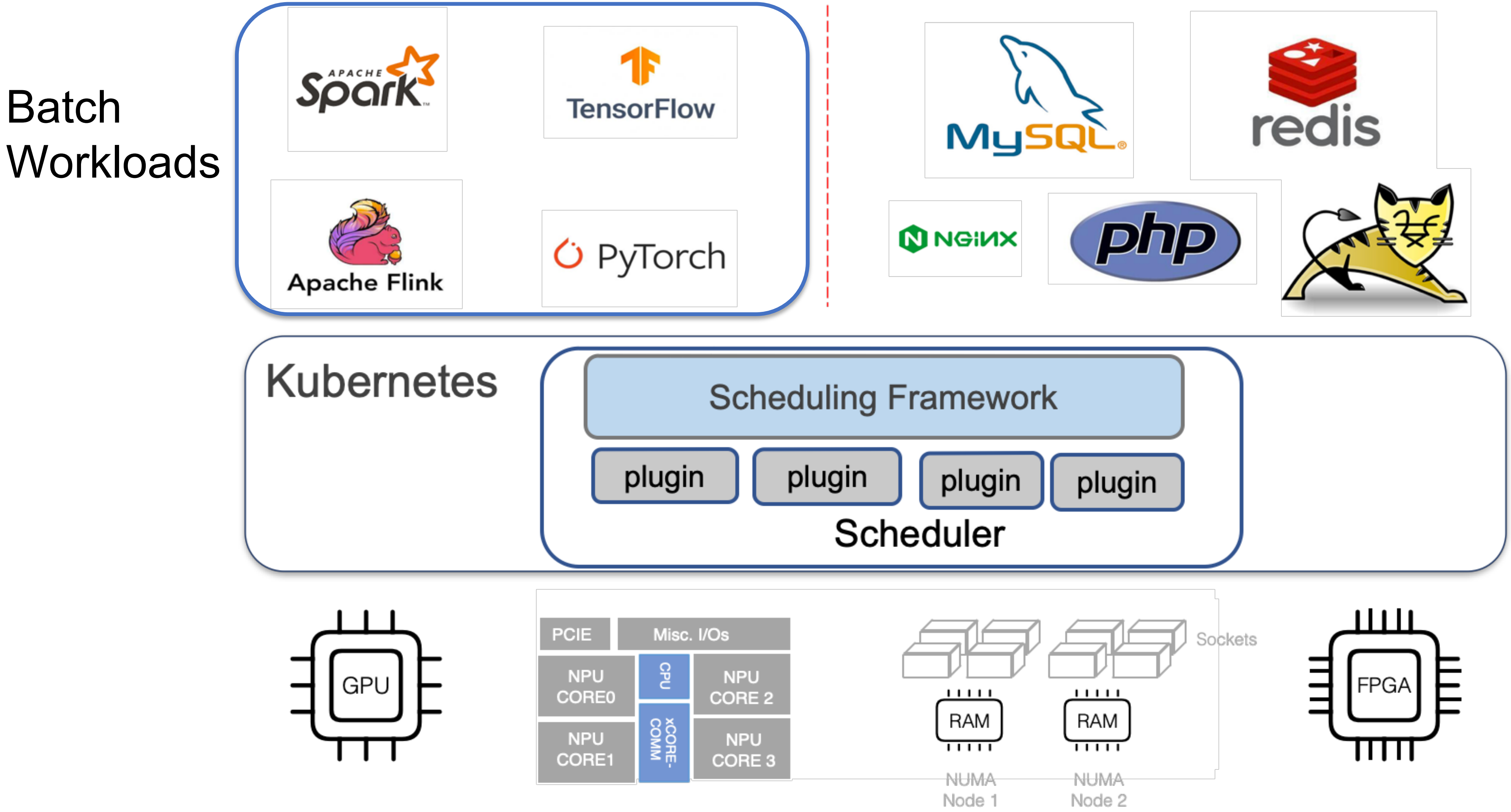
SPIDER-MAN
KubeCon CloudNativeCon North America 2018
KubeCon CloudNativeCon San Diego 2019
KubeCon CloudNativeCon China 2018
Kubernetes Summit 2020

Yuan Chen
Software engineer, Apple Cloud Services
Github: yuanchen8911
LinkedIn: yuanchen, Twitter: @baseloaded
yuanchen97@gmail.com

Agenda

- **Introduction**
- Capacity Scheduling
- Job Queue
- Demo
- Summary

The Diversity of Workloads in Kubernetes



Batch Workloads

Kubernetes

Scheduling Framework

plugin plugin plugin plugin

Scheduler

https://www.cncf.io/online-programs/how-alibaba-extends-k8s-scheduler-to-support-ai-and-big-data-workloads/

Resource Sharing in Kubernetes

Current status

- Quota is for capacity planning and admission control
- A single resource quota (request) value for a namespace
- Pod priority-based preemption

As a result

- Lack of flexible resource sharing between namespaces
- Low cluster utilization
- A roadblock to efficiently running batch workloads in k8s

Agenda

- Introduction
- **Capacity Scheduling**
- Job Queue Management
- Demo
- Summary

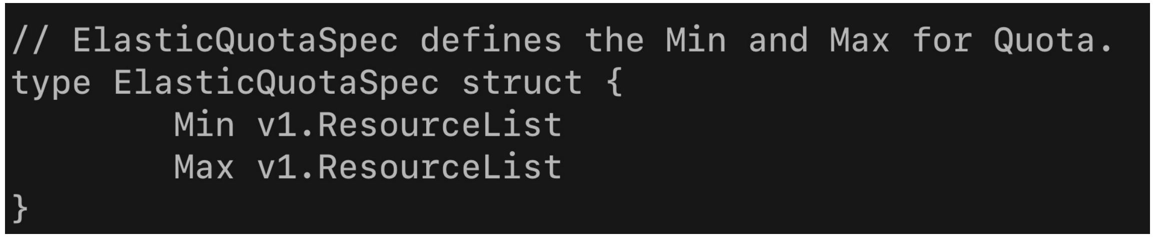
Elastic Quota and Capacity Scheduling

Dynamic resource sharing between namespaces

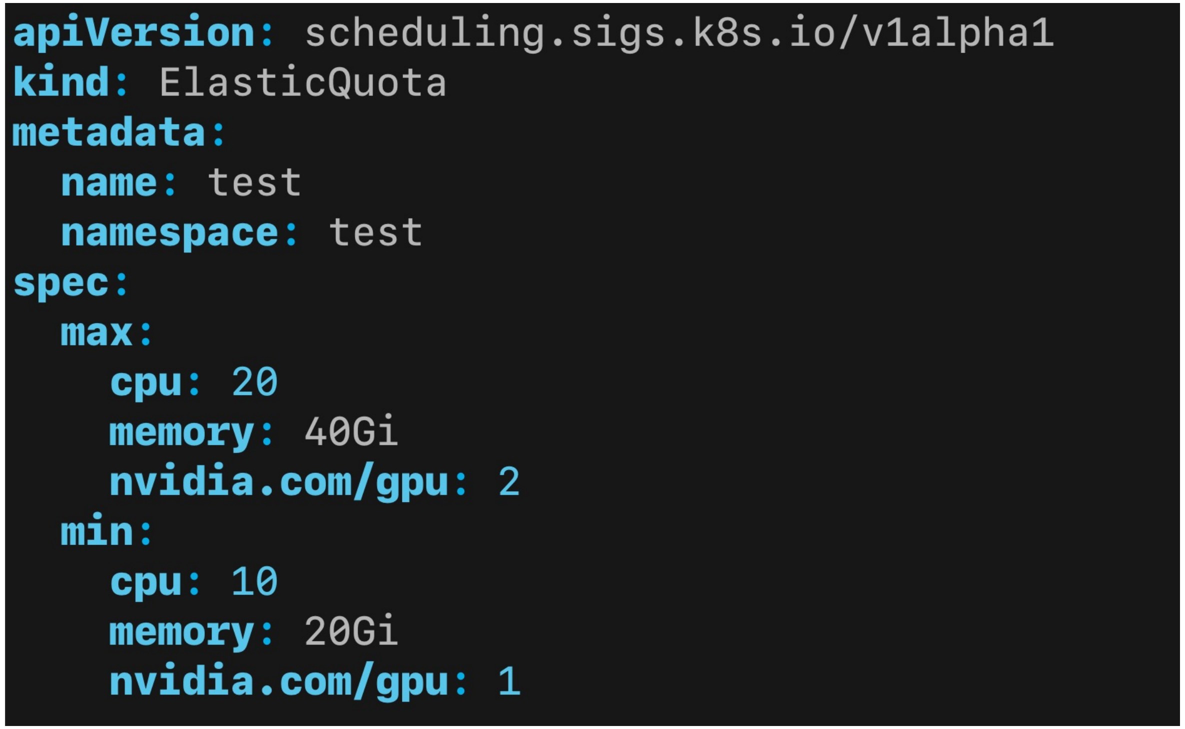
- Elastic quotas
- Fair sharing
- Hierarchical resource quotas

Elastic Quotas

- *ElasticQuota* CRD
- Min and Max resources
  - Min: guaranteed resource
  - Max: maximum resource
- Multi-resource types
  - CPU, memory, disk, GPU, extended resources
- Independent of existing *ResourceQuota*



```
// ElasticQuotaSpec defines the Min and Max for Quota.
type ElasticQuotaSpec struct {
        Min v1.ResourceList
        Max v1.ResourceList
}
```



```
apiVersion: scheduling.sigs.k8s.io/v1alpha1
kind: ElasticQuota
metadata:
  name: test
  namespace: test
spec:
  max:
    cpu: 20
    memory: 40Gi
    nvidia.com/gpu: 2
  min:
    cpu: 10
    memory: 20Gi
    nvidia.com/gpu: 1
```

Resource Guarantee and Fairness

When an ElasticQuota(namespace)'s min resource cannot be met

Resource.Request + Resource.Allocated < ElasticQuota.Min

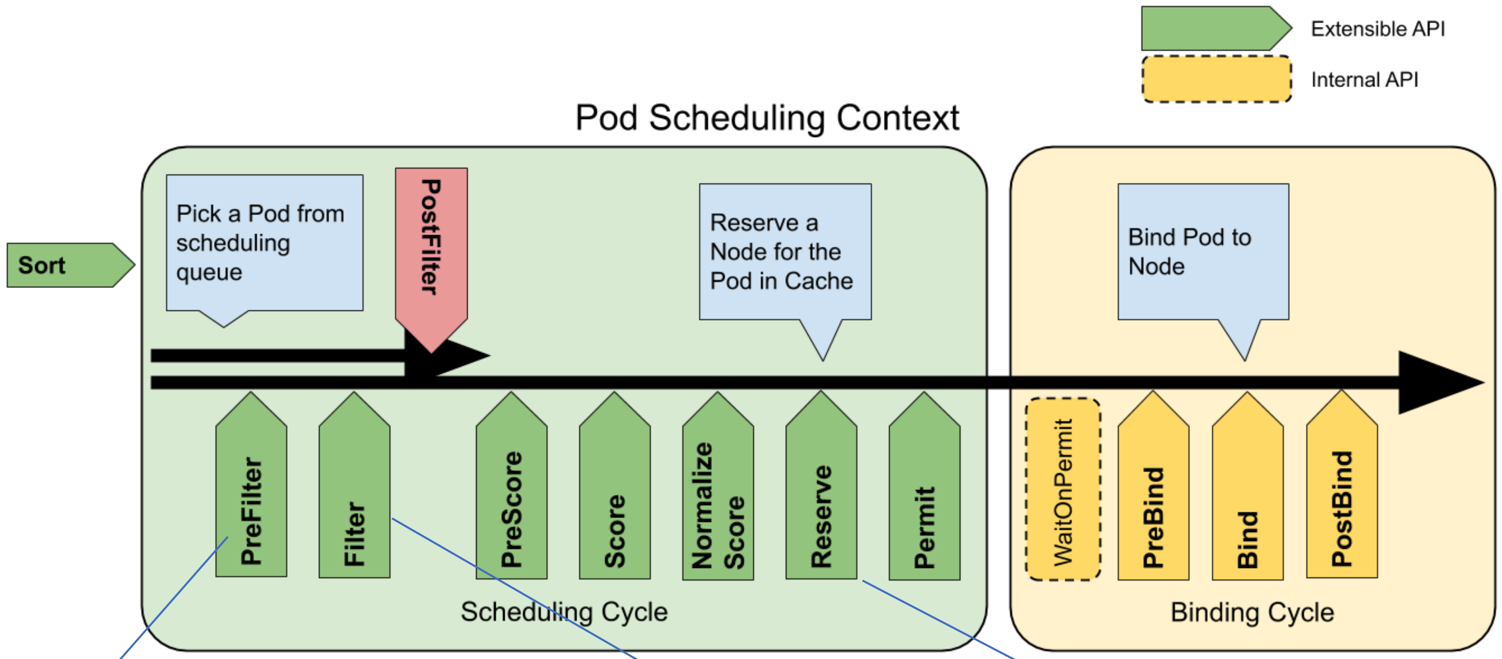
Preemption

- ElasticQuota/Namespace candidates

  Resource.Request + Resource.Allocated > ElasticQuota.Min

- Pod candidates: lower priority pods first, minimize the number of evicted pods

Capacity Scheduling Implementation



PreFilter: *Ensure that the used resources of every elastic quota doesn't exceed max*

PostFilter: *Custom preemption to ensure guaranteed resources*

Reserve:

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

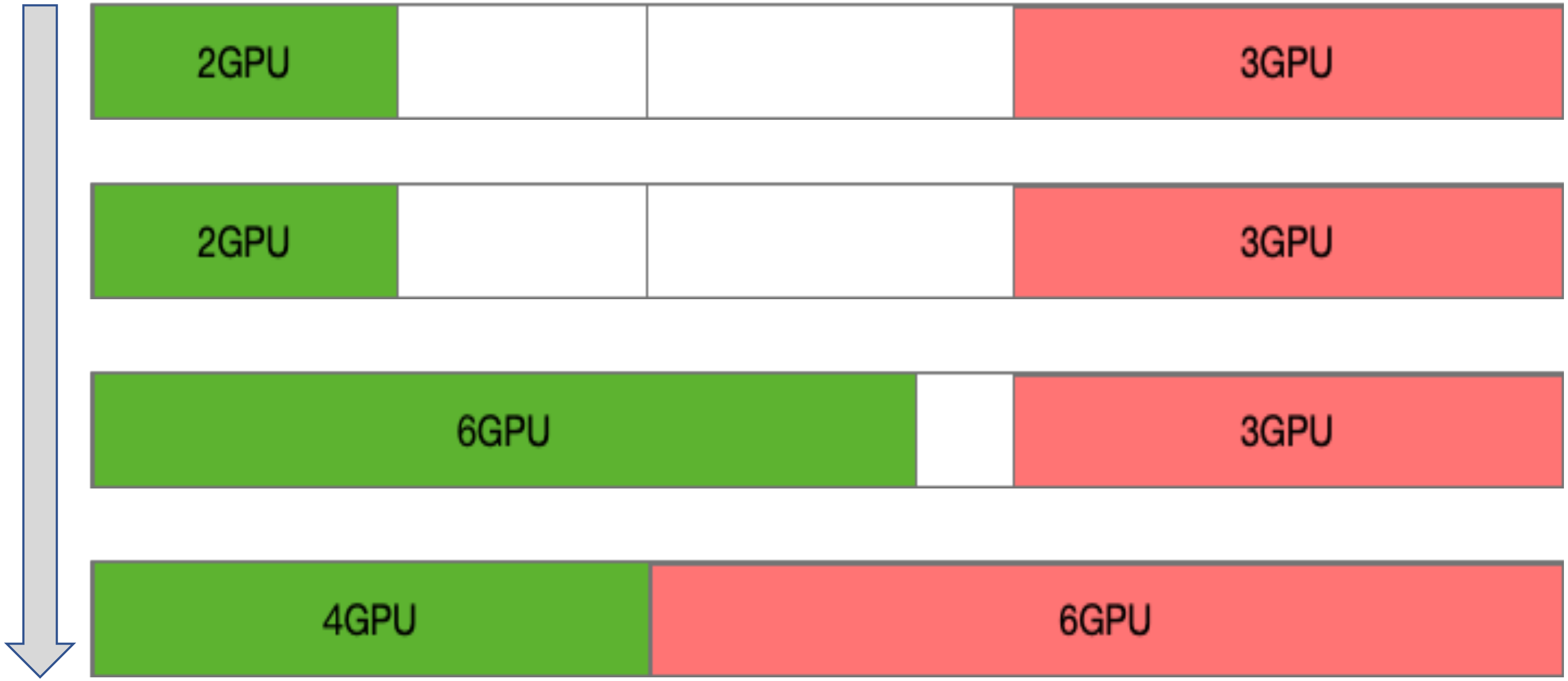
https://kubernetes.io/docs/concepts/scheduling-eviction/scheduling-framework/
https://github.com/kubernetes-sigs/scheduler-plugins

Elastic Quota Examples

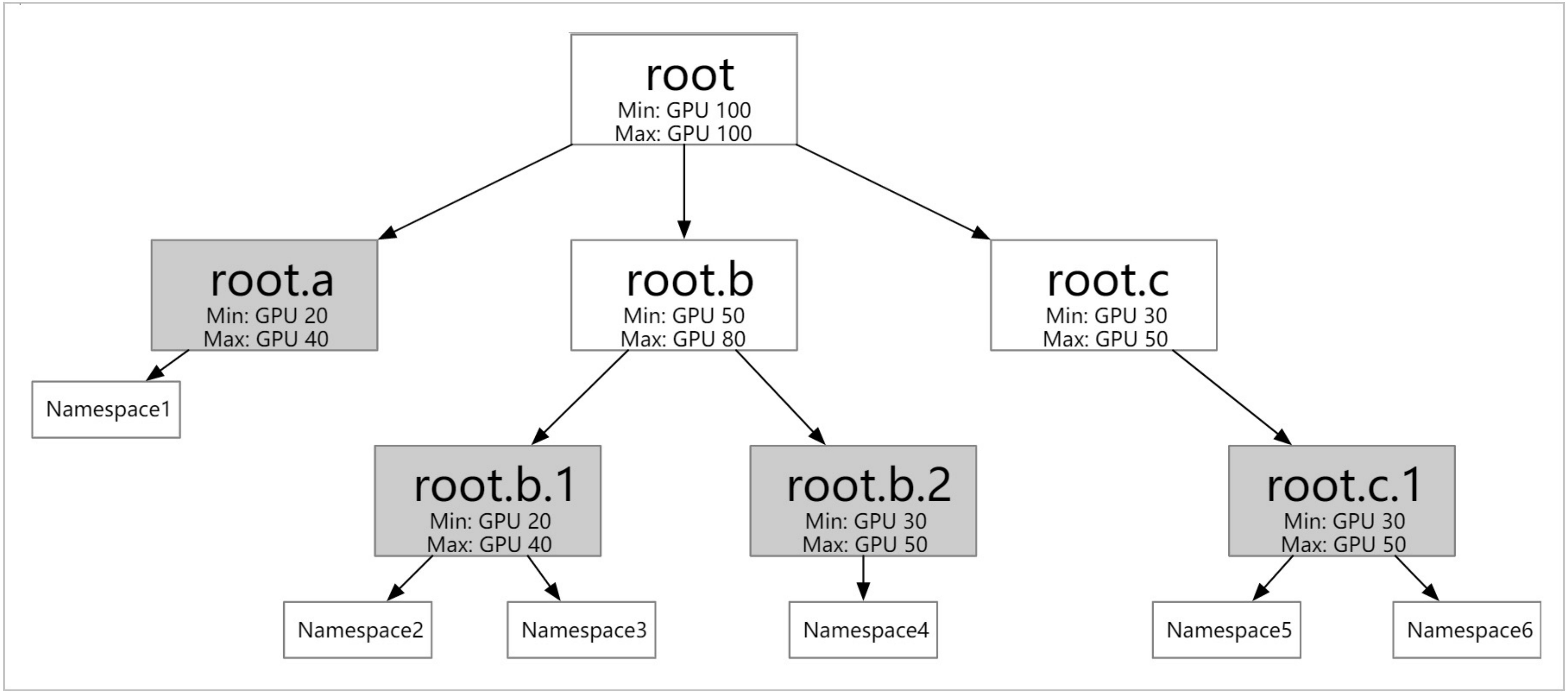
KubeCon | CloudNativeCon
North America 2021

Namespace 1: min:4, max:6

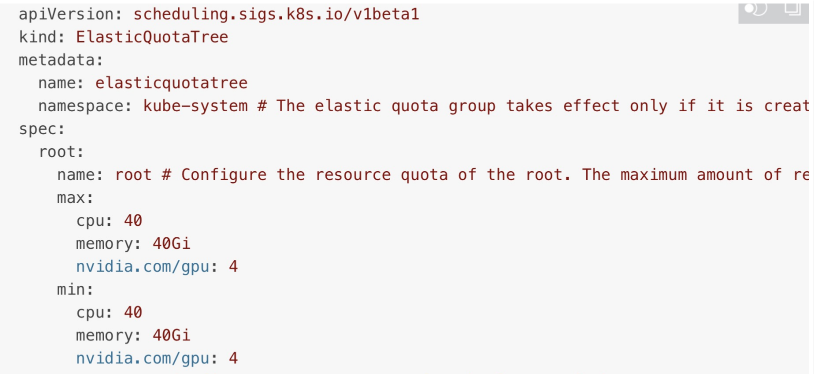
Namespace 2: min:6, max:8



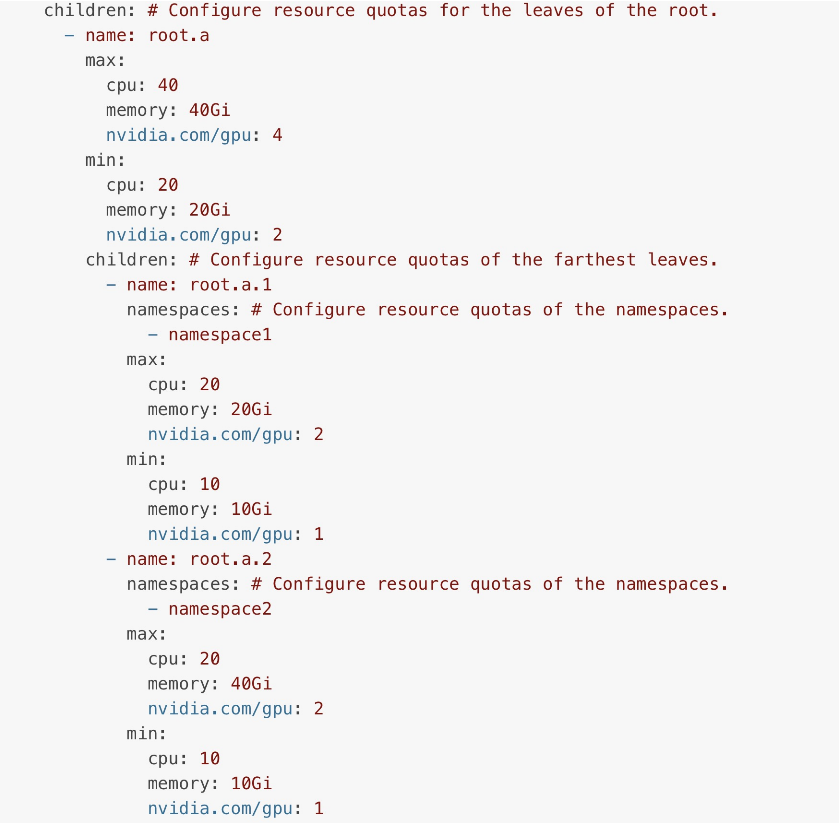
Hierarchical Quota



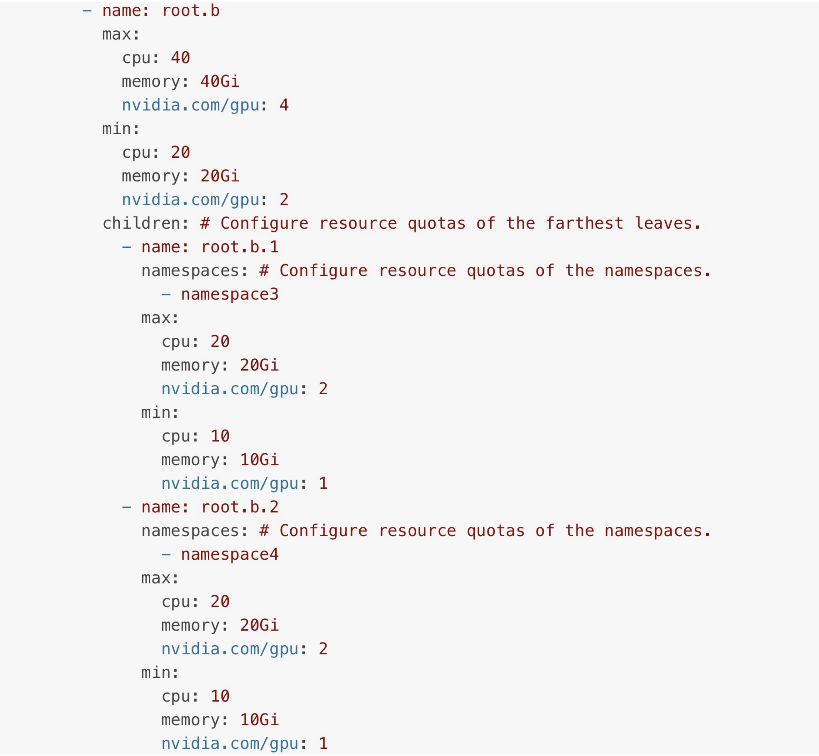
Hierarchical Elastic Quota Example



```yaml
apiVersion: scheduling.sigs.k8s.io/v1beta1
kind: ElasticQuotaTree
metadata:
  name: elasticquotatree
  namespace: kube-system # The elastic quota group takes effect only if it is creat
spec:
  root:
    name: root # Configure the resource quota of the root. The maximum amount of re
    max:
      cpu: 40
      memory: 40Gi
      nvidia.com/gpu: 4
    min:
      cpu: 40
      memory: 40Gi
      nvidia.com/gpu: 4
```



```yaml
    children: # Configure resource quotas for the leaves of the root.
      - name: root.a
        max:
          cpu: 40
          memory: 40Gi
          nvidia.com/gpu: 4
        min:
          cpu: 20
          memory: 20Gi
          nvidia.com/gpu: 2
        children: # Configure resource quotas of the farthest leaves.
          - name: root.a.1
            namespaces: # Configure resource quotas of the namespaces.
              - namespace1
            max:
              cpu: 20
              memory: 20Gi
              nvidia.com/gpu: 2
            min:
              cpu: 10
              memory: 10Gi
              nvidia.com/gpu: 1
          - name: root.a.2
            namespaces: # Configure resource quotas of the namespaces.
              - namespace2
            max:
              cpu: 20
              memory: 40Gi
              nvidia.com/gpu: 2
            min:
              cpu: 10
              memory: 10Gi
              nvidia.com/gpu: 1
```



```yaml
      - name: root.b
        max:
          cpu: 40
          memory: 40Gi
          nvidia.com/gpu: 4
        min:
          cpu: 20
          memory: 20Gi
          nvidia.com/gpu: 2
        children: # Configure resource quotas of the farthest leaves.
          - name: root.b.1
            namespaces: # Configure resource quotas of the namespaces.
              - namespace3
            max:
              cpu: 20
              memory: 20Gi
              nvidia.com/gpu: 2
            min:
              cpu: 10
              memory: 10Gi
              nvidia.com/gpu: 1
          - name: root.b.2
            namespaces: # Configure resource quotas of the namespaces.
              - namespace4
            max:
              cpu: 20
              memory: 20Gi
              nvidia.com/gpu: 2
            min:
              cpu: 10
              memory: 10Gi
              nvidia.com/gpu: 1
```

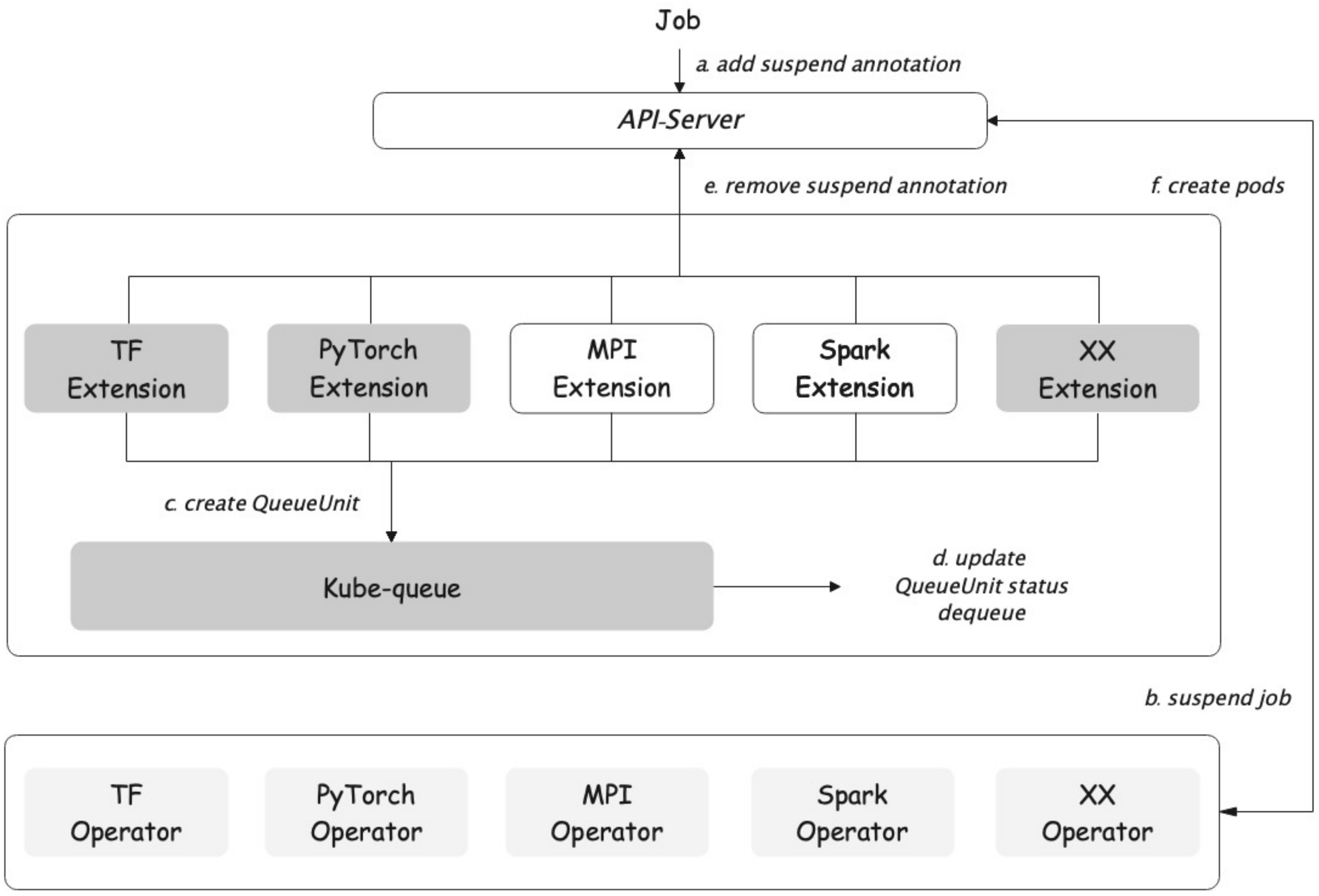
Agenda

- Introduction
- Capacity Scheduling
- **Job Queue**
- Demo
- Summary

Job Queue

- Manage workloads instead of pods
- Schedule workloads according to **priorities**, **creation time**, **quotas**: *ResourceQuota, ElasticQuota, Cluster Capacity*
- Provide fairness between different queues (under development)

Job Queue Management



https://github.com/kube-queue/kube-queue

Agenda

- Introduction
- Capacity Scheduling
- Job Queue
- **Demo**
- Summary

Demo

Elastic Quota Demo

Agenda

- Introduction
- Capacity Scheduling
- Job Queue
- Demo
- **Summary**

Current Status

Open source projects

- Elastic quota and capacity scheduling

  https://github.com/kubernetes-sigs/scheduler-plugins/tree/master/pkg/capacityscheduling
- Job queue

  https://github.com/kube-queue/kube-queue
- Hierarchical quota (next)

Early adoptions

- Alibaba Cloud: AI/ML, Spark on Kubernetes
- Apple: Spark on Kubernetes (ongoing)
- Baidu: Self-driving simulation (ongoing)

References

- *https://github.com/kubernetes-sigs/scheduler-plugins/tree/master/pkg/capacityscheduling*
- *https://github.com/kube-queue/kube-queue*
- *https://www.alibabacloud.com/help/doc-detail/213695.htm*
- *https://help.aliyun.com/document_detail/213695.htm* *(in Chinese)*

Acknowledgement

Many thanks to people who have contributed to the projects

(in alphabetical order)

- Abdullah Gharaibeh (Google)
- Aldo Culquicondor (Google)
- Chenkun Yao (Alibaba Cloud)
- Fei Guo (Alibaba Cloud)
- Jichuan Sun (SmartMore)
- Kai Zhang (Alibaba Cloud)
- Lei Yin (Alibaba Cloud)
- Wang Zhang (Tencent Cloud)
- Wei Huang (IBM)
- Xuan Gong (Salesforce)
- Yan Xu (Apple)

Thank you!

Qingcan Wang
Alibaba Cloud
Github: denkensk
LinkedIn: qingcan-wang-24aa21b6
qingcan.wqc@alibaba-inc.com

Yuan Chen
Apple Cloud Services
Github: yuanchen8911
LinkedIn: yuanchen
Twitter: @baseloaded
yuanchen97@gmail.com