KubeCon | CloudNativeCon

Europe 2020

Virtual

SIG Scheduling Deep Dive

Aldo Culquicondor, Google
Mike Dame, Red Hat

Outline

- SIG Scheduling introduction
- What’s new in kube-scheduler
    - The scheduling framework
    - Topology-aware pod spreading
    - Multiple Profiles
    - Performance improvements
- What’s new in descheduler
    - Releases matching k/k, gcr.io images
    - New descheduling strategies
        - RemovePodsHavingTooManyRestarts
        - PodLifetime
        - TopologySpread
    - Switch from Travis to Prow
    - Helm chart

SIG Scheduling Introduction

SIG Scheduling is responsible for the components that make Pod placement decisions.

Leads:

- Wei Huang (@Huang-Wei), IBM
- Abdullah Gharaibeh (@ahg-g), Google

Projects:

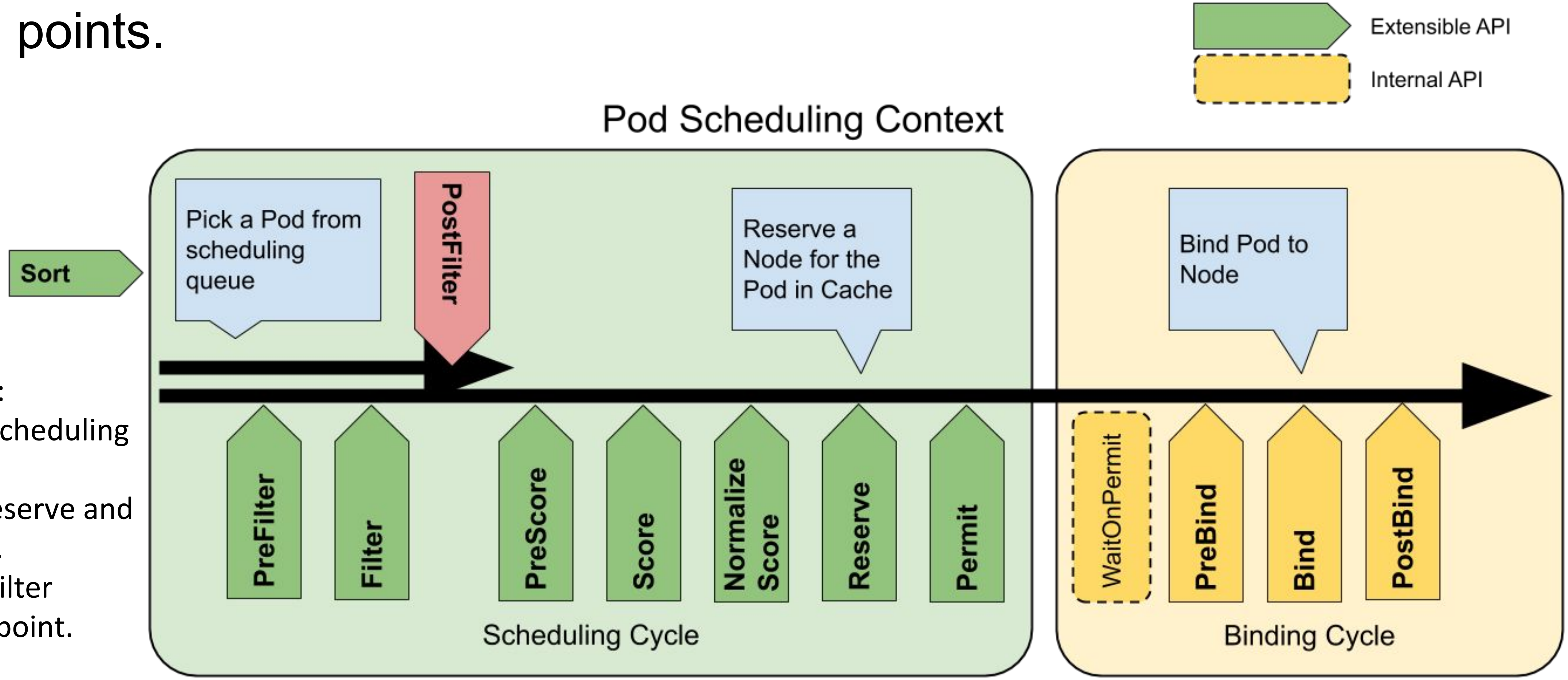
- kube-scheduler, part of kubernetes/kubernetes
- descheduler, a controller for rebalancing pods
- scheduler-plugins for incubation of scheduling plugins

What’s new in kube-scheduler

- The scheduling framework
- Scheduling profiles (Alpha in 1.18, Beta in 1.19)
- Topology-aware pod spreading (Beta in 1.18, GA in 1.19)
- Performance improvements

The scheduling framework

A refactoring of kube-scheduler that facilitates extensibility and building custom schedulers. Features are contained in plugins that implement the extension points.

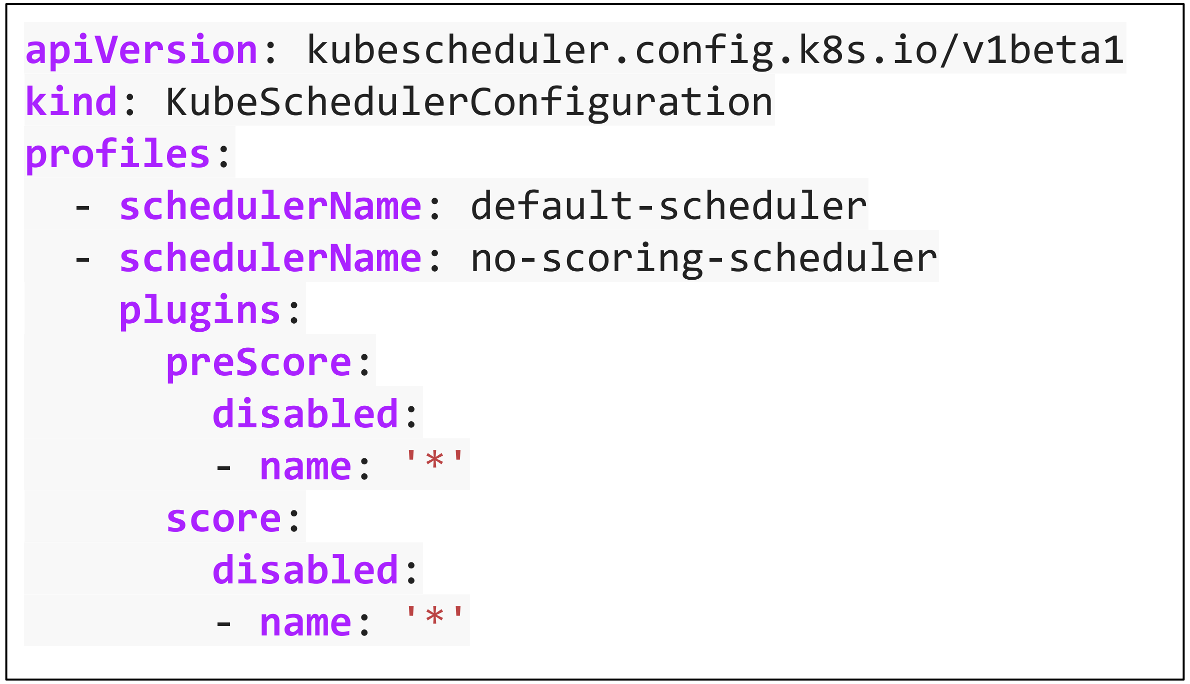


Recent changes:

- Permit in scheduling cycle.
- Merged Reserve and Unreserve.
- New PostFilter extension point.

Scheduling Profiles

- The cluster admin facing API for the scheduler framework.
- Users can disable, enable and reorder plugins.
- A **single** kube-scheduler can run **multiple** profiles. Pods can select the profile using `.spec.schedulerName`
- Beta in 1.19



```yaml
apiVersion: kubescheduler.config.k8s.io/v1beta1
kind: KubeSchedulerConfiguration
profiles:
  - schedulerName: default-scheduler
  - schedulerName: no-scoring-scheduler
    plugins:
      preScore:
        disabled:
        - name: '*'
      score:
        disabled:
        - name: '*'
```

Topology-aware pod spreading

- Control how Pods are spread across failure-domains such as zones, nodes or other user-defined topologies.
- The constraints can be:
  - hard: only schedule in nodes that satisfy the configured skew.
  - soft: nodes that satisfy the skew are scored higher.
- Cluster administrators can set default constraints that apply to Services and ReplicaSets
- GA in 1.19. What's new:
  - More influential scoring
  - maxSkew can be used to control scoring strength

```yaml
kind: Pod
apiVersion: v1
metadata:
  name: mypod
  labels:
    foo: bar
spec:
  topologySpreadConstraints:
  - maxSkew: 2
    topologyKey: zone
    whenUnsatisfiable: DoNotSchedule
    labelSelector:
      matchLabels:
        foo: bar
  containers:
  - name: pause
    image: k8s.gcr.io/pause:3.1
```

Performance improvements

Continuous work:

- In 1.17, we focused on vanilla workloads.
  - 2.5X latency improvement
  - We achieved 100 pod/s in clusters with 15k nodes.
  - Improved latency for Pod (Anti)Affinity: 24X faster for preferred and 7X for required.
- In 1.18 and 1.19, we focused on
  - Pod (Anti)Affinity (2x improvement)
  - Pod Topology Spreading (now comparable to legacy SelectorSpread plugin).
- In 1.20 and beyond, we will focus on preemption and the effect of unschedulable pods.

What’s new in Descheduler

KubeCon | CloudNativeCon Europe 2020 Virtual

- Releases matching k/k, gcr.io images
- New descheduling strategies
  - `RemovePodsHavingTooManyRestarts`
  - `PodLifetime`
  - `TopologySpread`
- Switch from Travis to Prow
- Helm chart
- Misc. improvements and refactors

Descheduler releases

- Release cycle now matches k8s
  - Tags (v0.19.0) and branches (release-1.19)

- Prod gcr.io images:
  - `asia.gcr.io/k8s-artifacts-prod/descheduler/descheduler:v0.18.0`
  - `eu.gcr.io/k8s-artifacts-prod/descheduler/descheduler:v0.18.0`
  - `us.gcr.io/k8s-artifacts-prod/descheduler/descheduler:v0.18.0`

New descheduling strategies

- **`RemovePodsHavingTooManyRestarts`**
  Used to evict crashlooping pods, or any pod constantly restarting
  - `podRestartThreshold`
    Number of restarts at which a pod should be evicted
  - `includingInitContainers`
    Bool to set whether to include InitContainer restarts in calculation

- **`PodLifetime`**
  Removes pods older than maxPodLifetimeSeconds
  - `maxPodLifetimeSeconds`
    Seconds after which a pod should be evicted

- **`TopologySpread`** (in progress)

We're always accepting new contributions!

Helm Chart & Misc. Changes

- Helm chart published automatically with release
- Pod & Namespace selectors
- Go 1.14.4
- GH Issue templates
- Eviction reasons/events
- Improved logging, code refactors...

KubeCon | CloudNativeCon

Europe 2020

Virtual

KEEP CLOUD NATIVE

CONNECTED