Intro: SIG-Scheduling

Da (Klaus.) Ma (@k82cn, mada3@huawei.com)

kubernetes

Charter of SIG Scheduling

SIG Scheduling is responsible for the components that make Pod placement decisions. We build Kubernetes schedulers and scheduling features for Pods. We design and implement features that allows users to customize placement of Pods on the nodes of a cluster. These features include those that improve reliability of workloads, more efficient use of cluster resources, and/or enforces placement policies.

kubernetes

Overview of SIG Scheduling

Meetings

- 10AM PT Meeting: Thursdays at 17:00 UTC (biweekly starting Thursday June 7, 2018). Convert to your timezone.
- 5PM PT Meeting: Thursdays at 24:00 UTC (biweekly starting Thursday June 14, 2018). Convert to your timezone.

Leadership

- Bobby (Babak) Salamat (**@bsalamat**), Google
- Klaus Ma (**@k82cn**), Huawei

Contact

- Slack: https://kubernetes.slack.com/messages/sig-scheduling
- Mailing list: https://groups.google.com/forum/#!forum/kubernetes-sig-scheduling
- Open Community Issues/PRs: https://github.com/kubernetes/community/labels/sig/scheduling

kubernetes

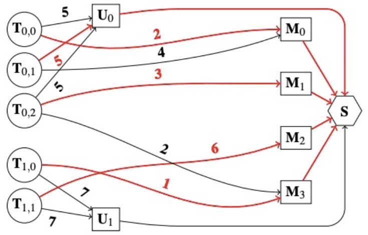
Sub-projects of SIG Scheduling

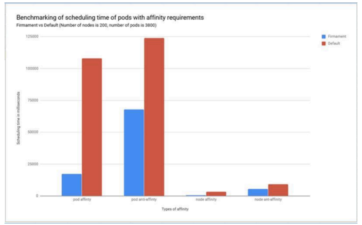
kube-batch

poseidon

kube-scheduler

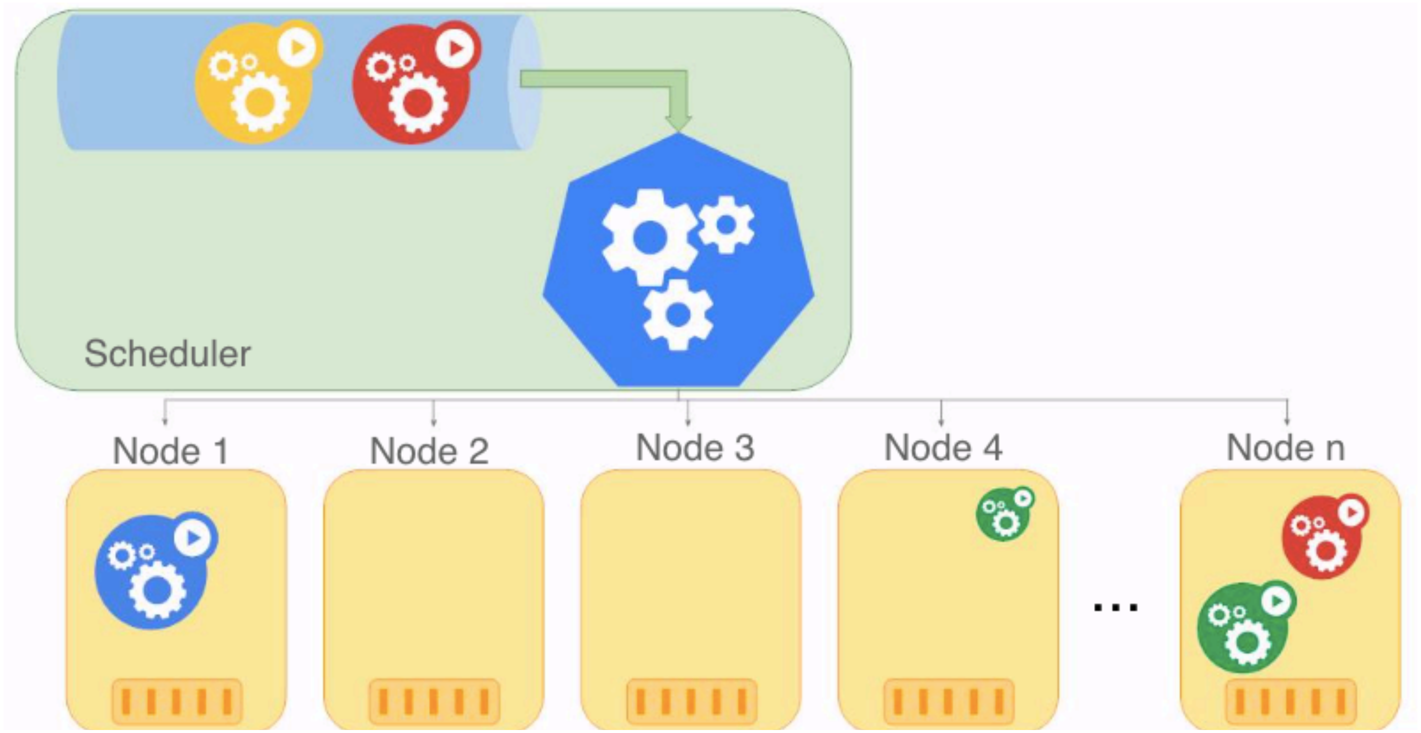
descheduler





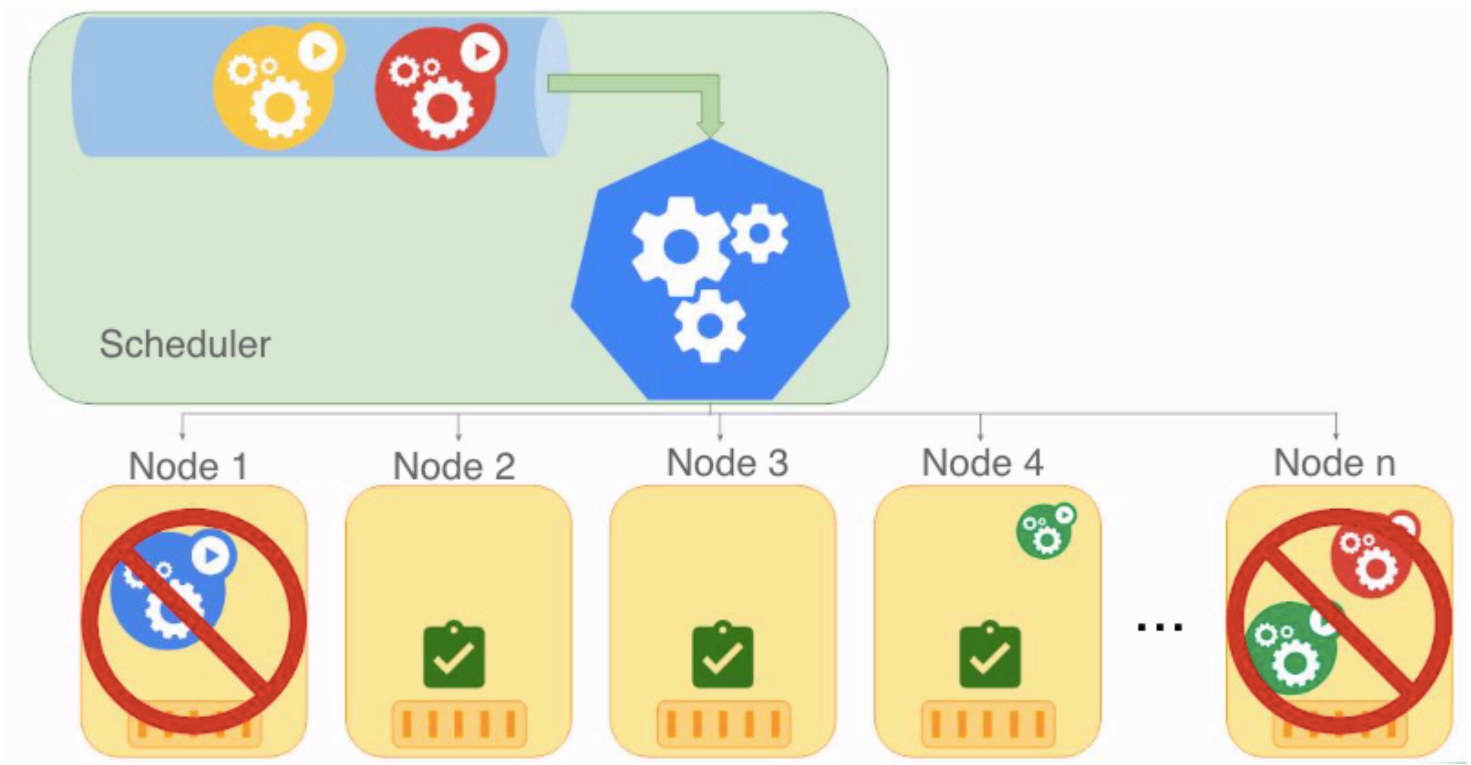
kubernetes

kube-scheduler schedules one Pod at a time



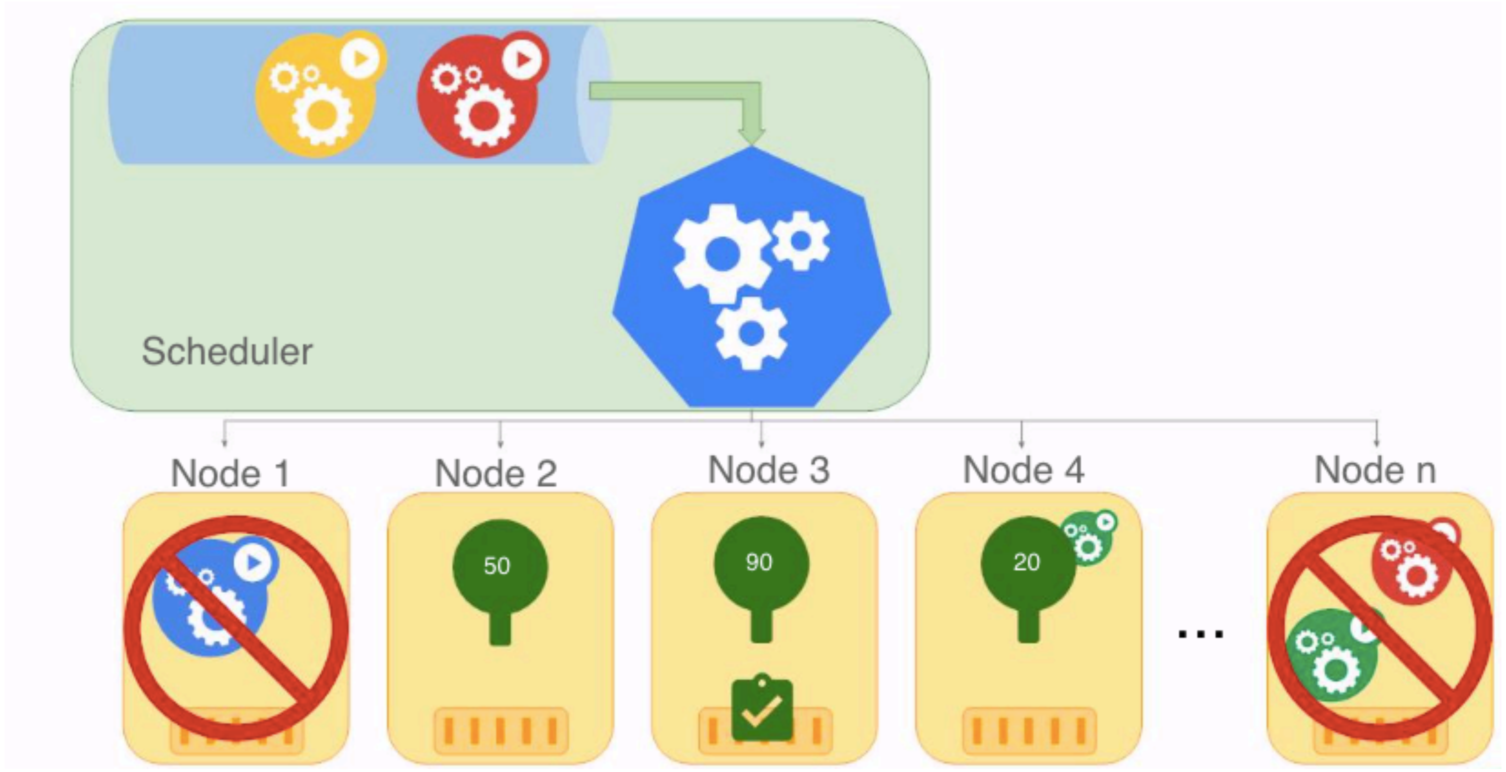
kubernetes

Predicate functions filter out Nodes



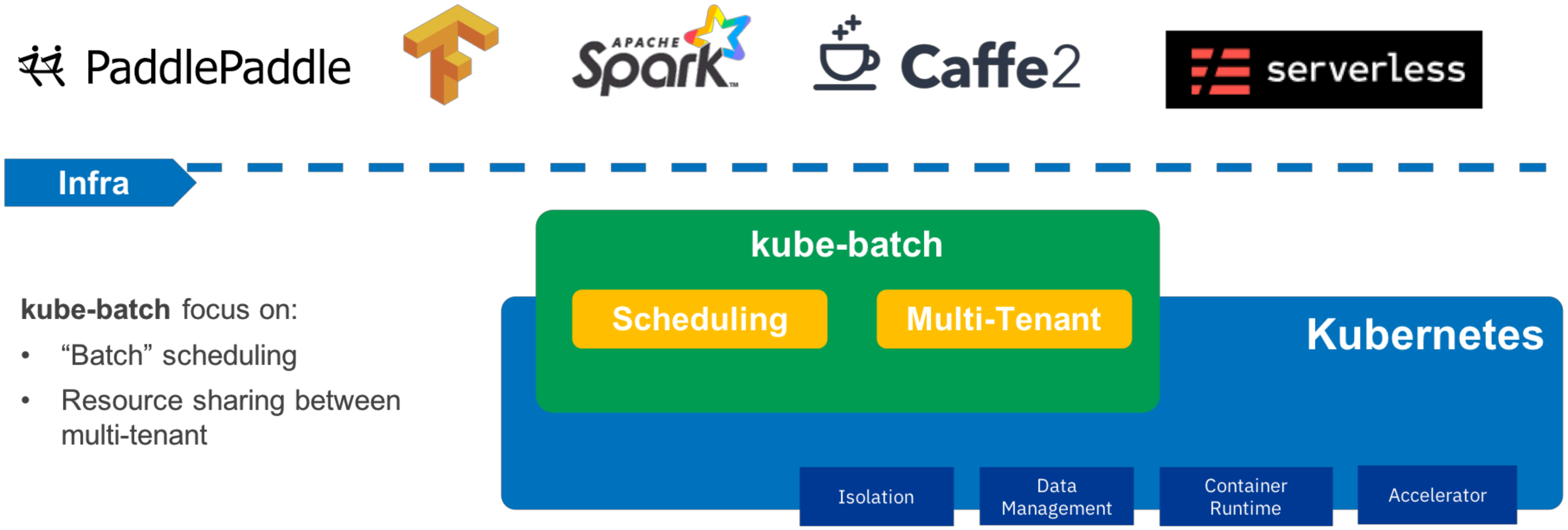
kubernetes

Priority functions rank the remaining Nodes



kubernetes

Overview of kube-batch



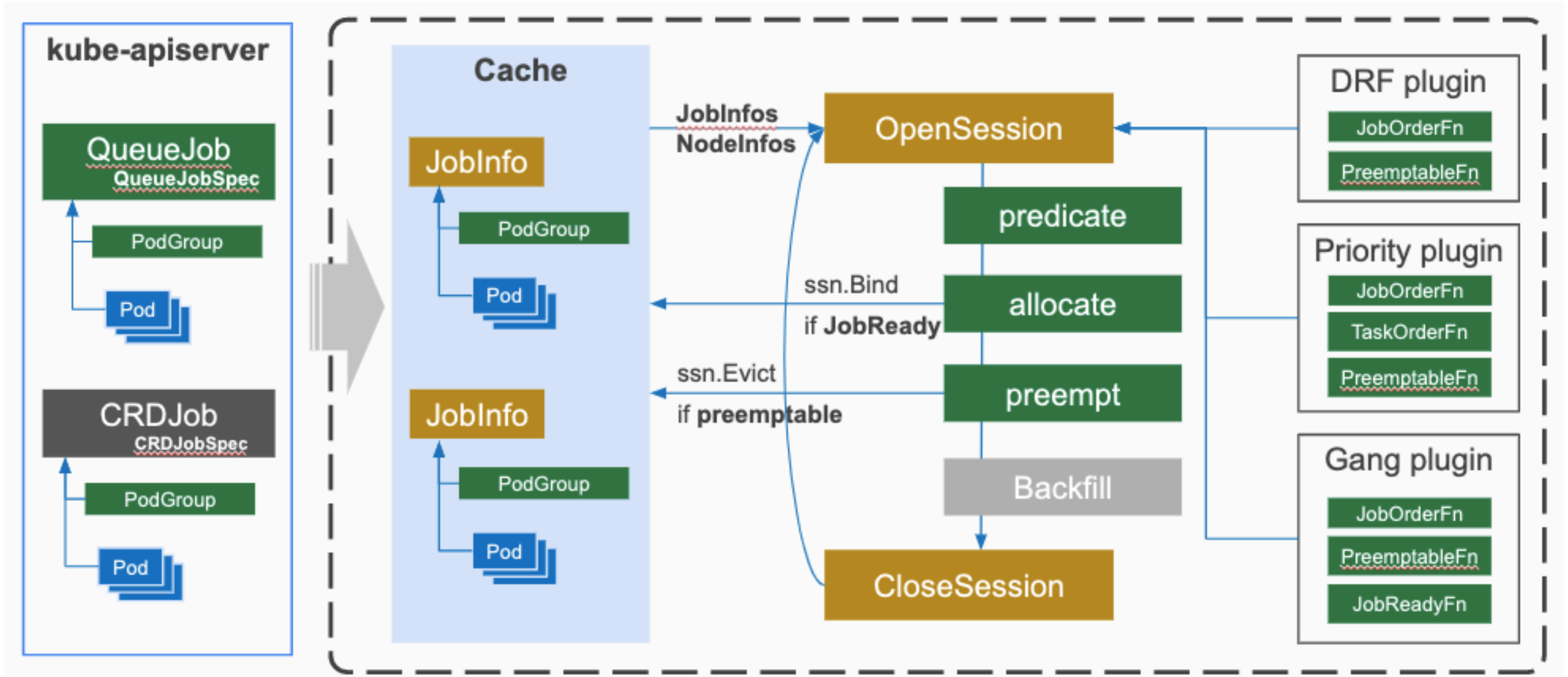
kube-batch focus on:

- "Batch" scheduling
- Resource sharing between multi-tenant

kube-batch **NOT** support:

- Data Management
- Accelerator (Kubelet), e.g. GPU
- Isolation for multi-tenant
- Job Management
- New container runtime, e.g. Singularity, CharlieCloud

Overview of kube-batch



Re-construct JobInfo in Cache by **PodGroup**

Predicate, *allocate*, *preempt* are **Actions**, and they're **pluggable**

Plugins on demand

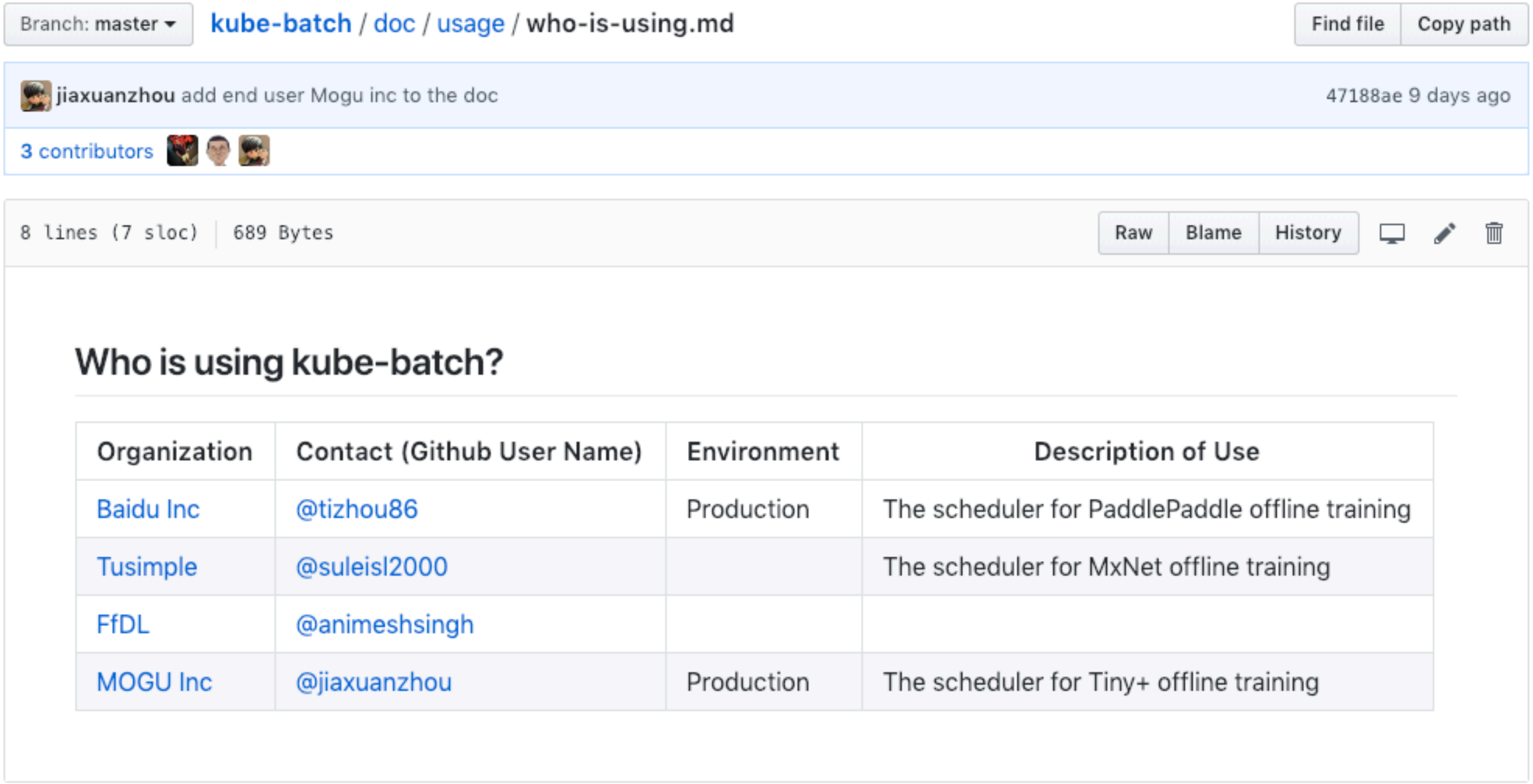
Features of kube-batch

- **Co-scheduling**
- “Fair-sharing” (job/queue)
- Preemption/Reclaim
- Task Priority within Job
- Predicates
- Queue
- Backfill (partially)
- Dynamic configuration

Bring Batch Capability into Kubernetes (#68357)

kubernetes

Users of kube-batch



Branch: master kube-batch / doc / usage / who-is-using.md Find file Copy path

jiaxuanzhou add end user Mogu inc to the doc 47188ae 9 days ago

3 contributors

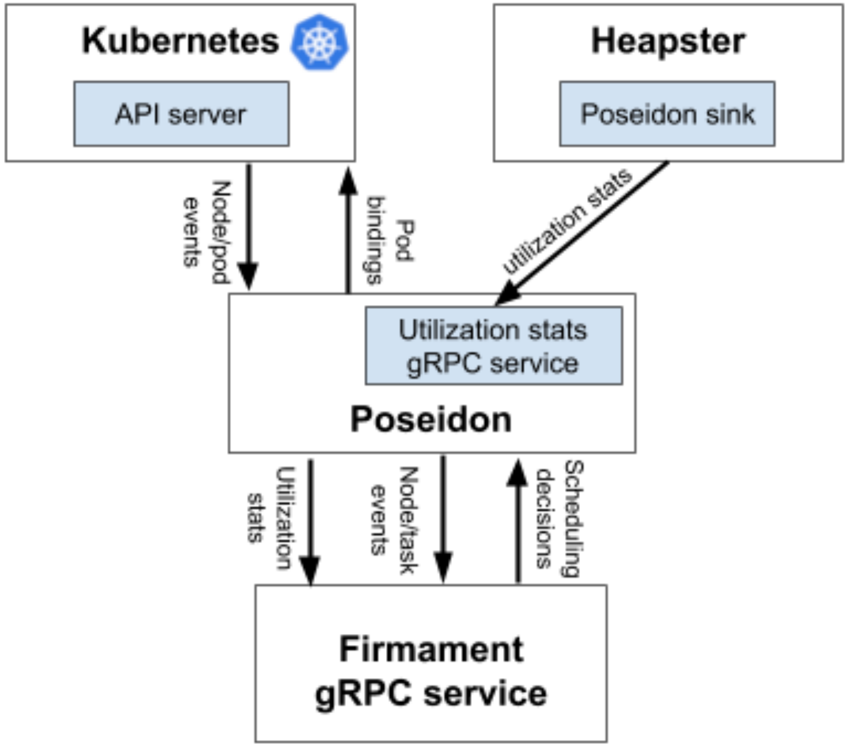
8 lines (7 sloc) 689 Bytes Raw Blame History

Who is using kube-batch?

Organization	Contact (Github User Name)	Environment	Description of Use
Baidu Inc	@tizhou86	Production	The scheduler for PaddlePaddle offline training
Tusimple	@suleisl2000		The scheduler for MxNet offline training
FfDL	@animeshsingh		
MOGU Inc	@jiaxuanzhou	Production	The scheduler for Tiny+ offline training

ietes

Poseidon



Poseidon/Firmament scheduler augments the current Kubernetes scheduling capabilities by incorporating a new novel flow network graph based scheduling capabilities alongside the default Kubernetes Scheduler.

Firmament models workloads on a cluster as flow networks and runs min-cost flow optimizations over these networks to make scheduling decisions.

kubernetes

Features of Poseidon

1. Node level Affinity and Anti-Affinity
2. Pod level Affinity and Anti-Affinity
3. Taints & Tolerations
4. Gang Scheduling

kubernetes

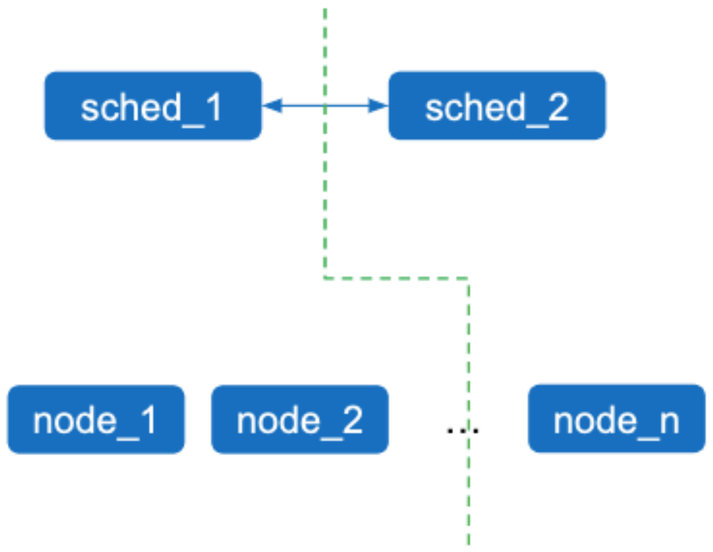
How those schedulers **work together ???**

kubernetes

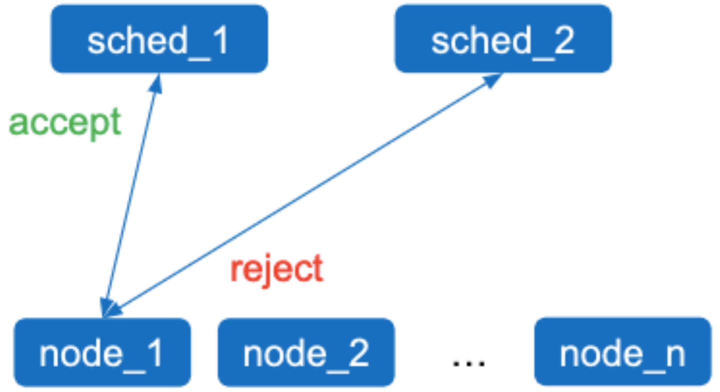
Sorry, I don-t know :(

kubernetes

Multi-Schedulers

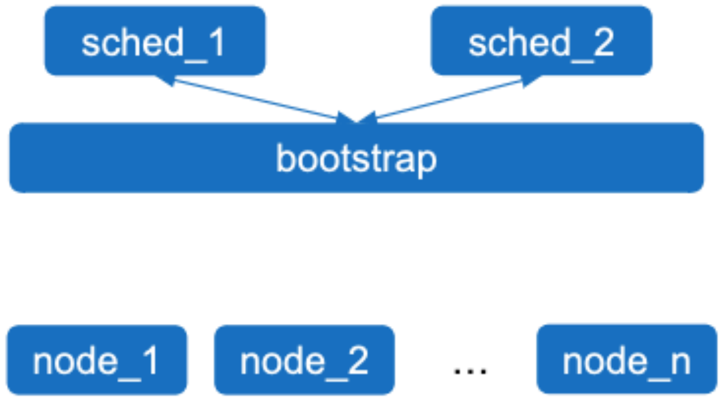


Option 1

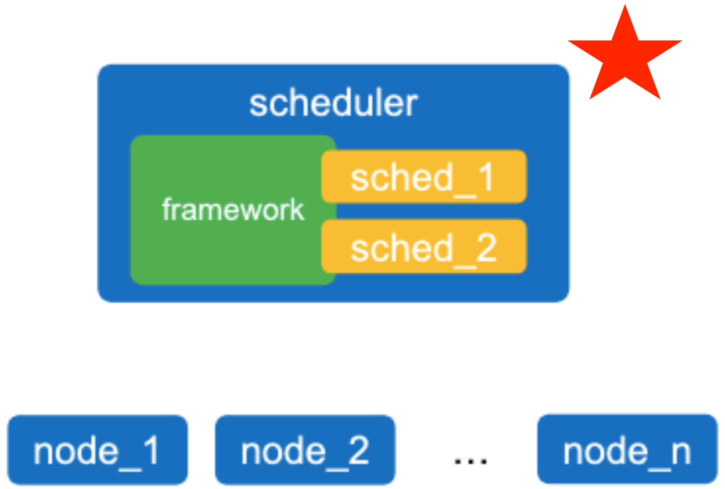


Option 2

Multi-Schedulers



Option 3



Option 4

Descheduler

Trigger Of Pod Movement/Migration

Eviction -> Creation -> Re-schedule

Scheduling in Kubernetes is the process of binding pending pods to nodes, and is performed by a component of Kubernetes called kube-scheduler. The scheduler's decisions, whether or where a pod can or can not be scheduled, are guided by its configurable policy which comprises of set of rules, called predicates and priorities. The scheduler's decisions are influenced by its view of a Kubernetes cluster at that point of time when a new pod appears first time for scheduling. As Kubernetes clusters are very dynamic and their state change over time, there may be desired to move already running pods to some other nodes for various reasons:

- Some nodes are under or over utilized.
- The original scheduling decision does not hold true any more, as taints or labels are added to or removed from nodes, pod/node affinity requirements are not satisfied any more.
- Some nodes failed and their pods moved to other nodes.
- New nodes are added to clusters.

Consequently, there might be several pods scheduled on less desired nodes in a cluster. Descheduler, based on its policy, finds pods that can be moved and evicts them. Please note, in current implementation, descheduler does not schedule replacement of evicted pods but relies on the default scheduler for that.

User Cases of Descheduler

- Some nodes are under or over utilized.
- The original scheduling decision does not hold true any more, as taints or labels are added to or removed from nodes, pod/node affinity requirements are not satisfied any more.
- Some nodes failed and their pods moved to other nodes.
- New nodes are added to clusters.

Policy & Strategy

- RemoveDuplicates
- LowNodeUtilization
- RemovePodsViolatingInterPodAntiAffinity
- RemovePodsViolatingNodeAffinity

kubernetes

Pod Eviction Restriction

- Critical pods (with annotations scheduler.alpha.kubernetes.io/critical-pod) are never evicted.
- Pods (static or mirrored pods or stand alone pods) not part of an RC, RS, Deployment or Jobs are never evicted because these pods won't be recreated.
- Pods associated with DaemonSets are never evicted.
- Pods with local storage are never evicted.
- Best efforts pods are evicted before Burstable and Guaranteed pods.
- Pod are never evicted If violates its PDB

kubernetes

Deep Dive: Scheduling SIG - Bobby (Babak) Salamat, Google

Thursday, December 13, 2018 4:30pm - 5:05pm ; 618-620

kubernetes

Thank You :)

kubernetes