Deep Dive: SIG Scheduling

Babak “Bobby” Salamat, Google

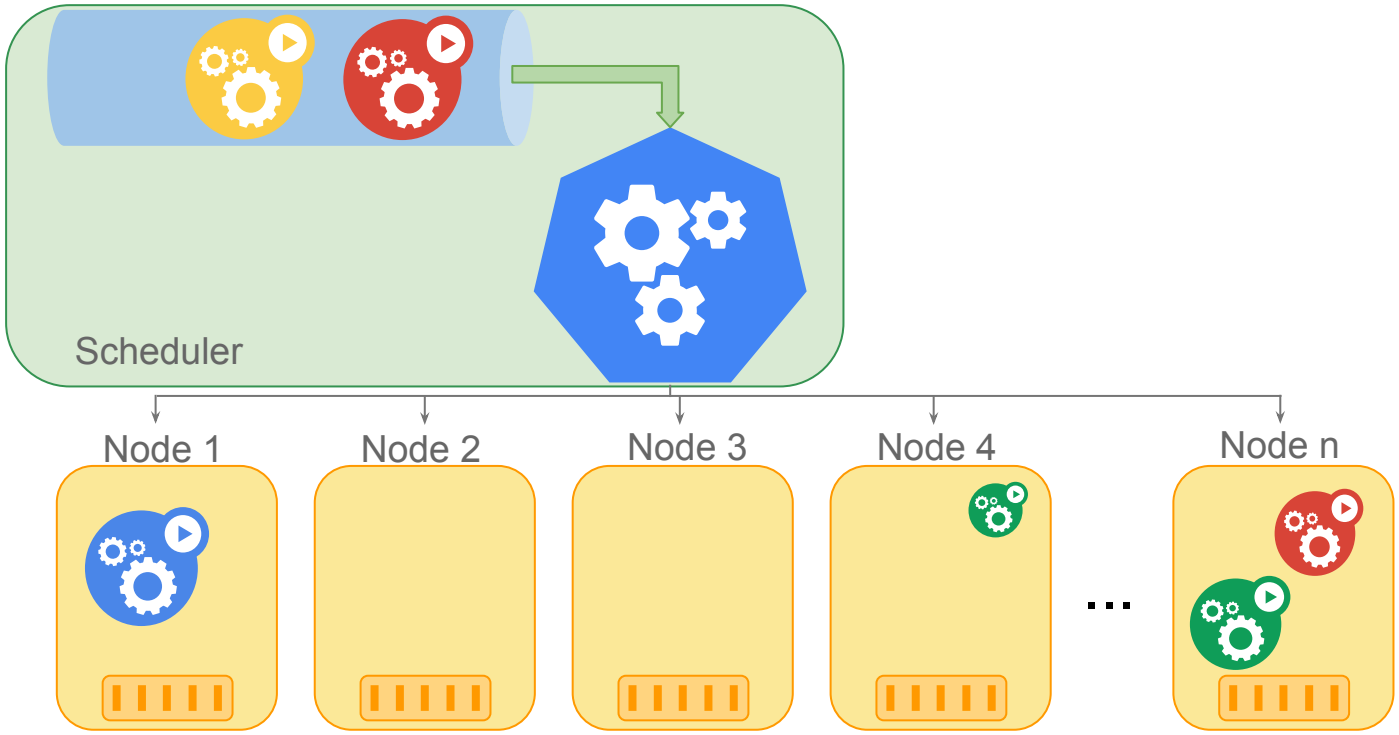
KubeCon 2019, Barcelona

Disclaimer

We go fast to give more time for questions.

Introduction

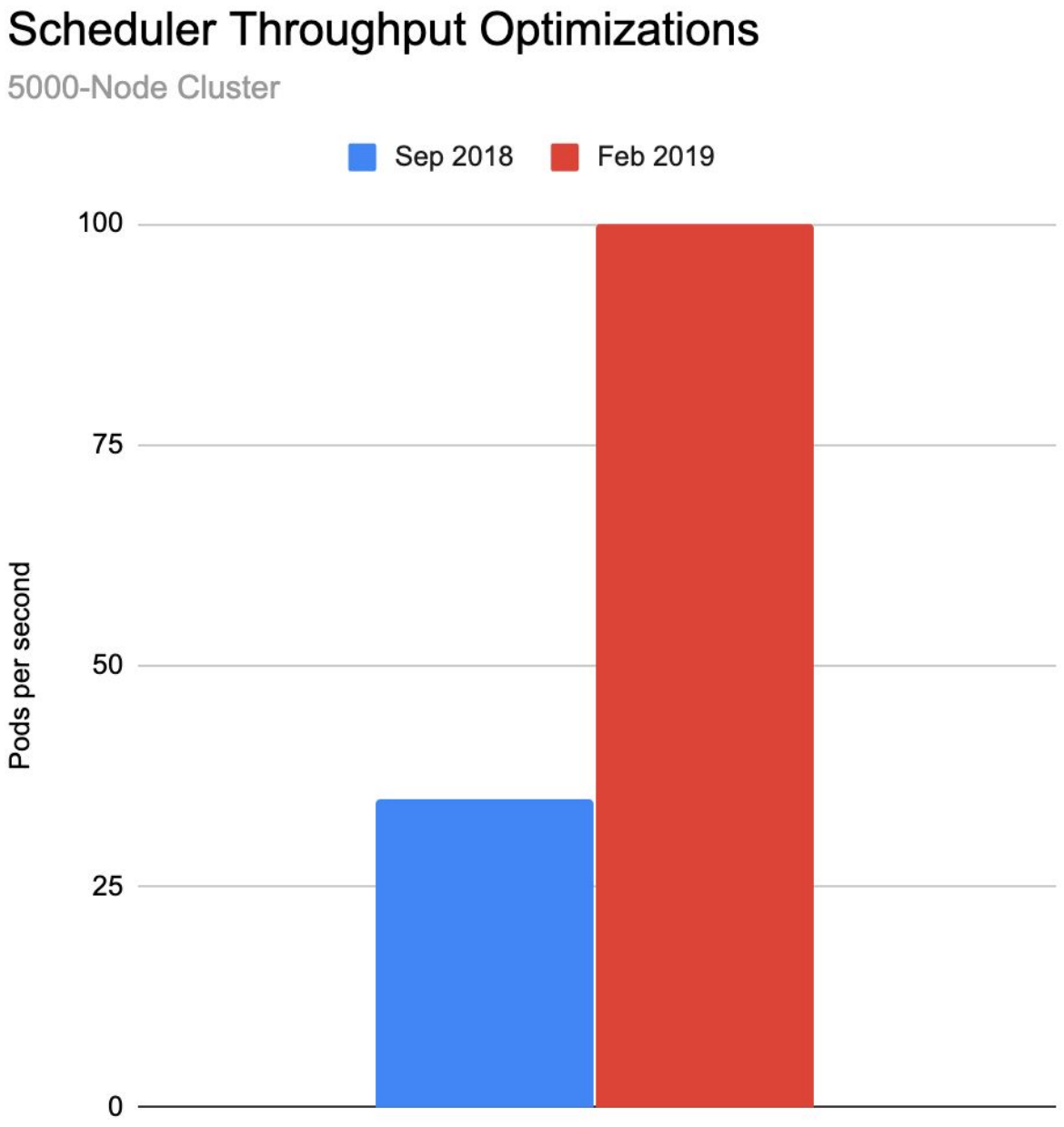
- Kubernetes Scheduler is responsible for finding appropriate nodes that can run Pods.
- The scheduler is not responsible for managing life cycle of Pods.



Notable features

- Check node resources
- Spread Pods of a collection, such as a ReplicaSet, among nodes
- Support taints and tolerations
- Support node affinity
- Support inter-pod affinity
- Check node conditions, such as memory pressure, PID pressure, etc.
- Prefer nodes with lowest/highest levels of resource usage
- Prefer nodes which already have images needed for the Pod

Recent Developments



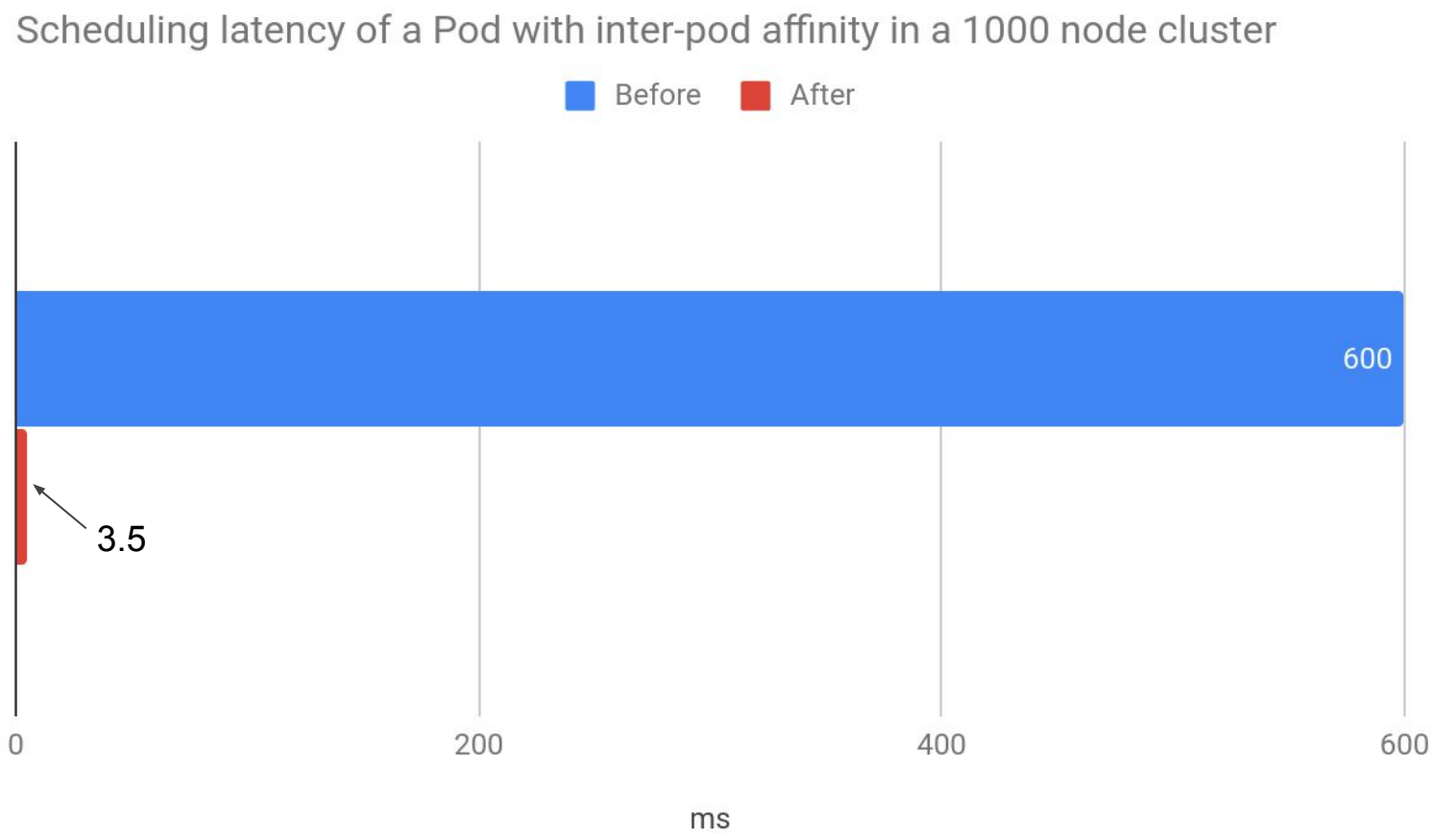
Recent Performance Improvements

3X throughput increase
in 6 months

Inter-Pod Affinity/Anti-affinity

Inter-pod affinity used to be ~1000 times slower than other scheduler features

Over 170X performance improvement by preprocessing and caching



Pod Priority and Preemption

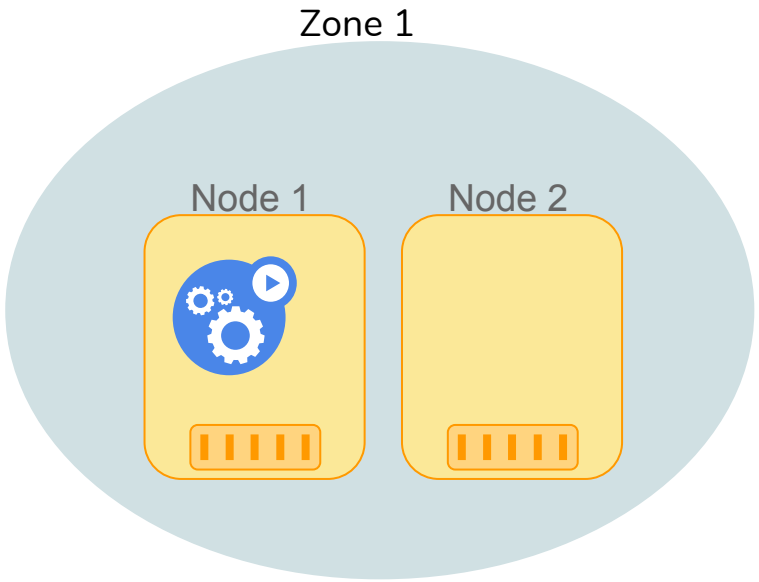
Graduated to stable (GA) in K8s 1.14

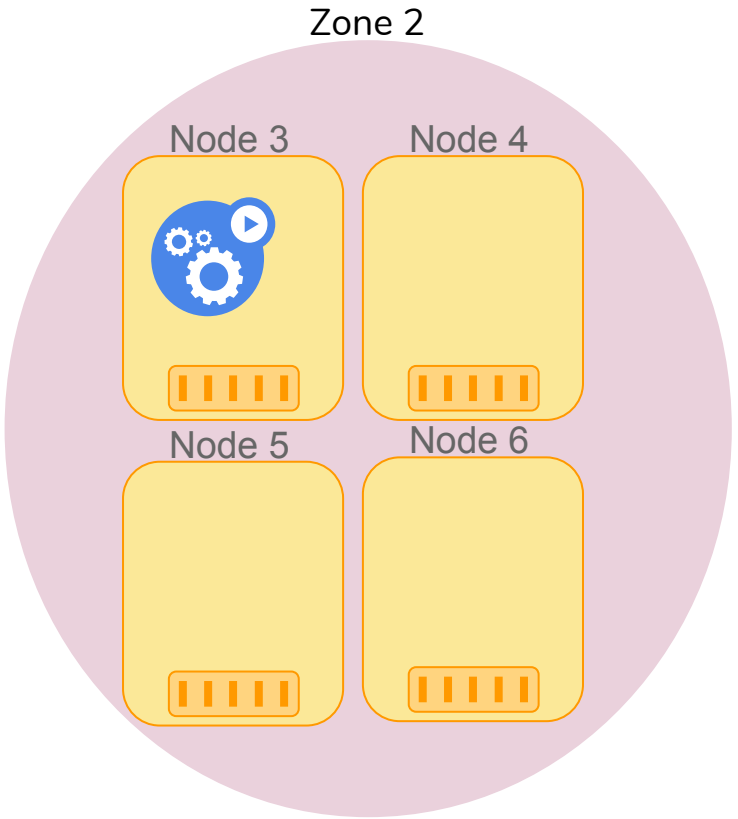
pexels/graduation

Planned Features

Even Pod Spreading

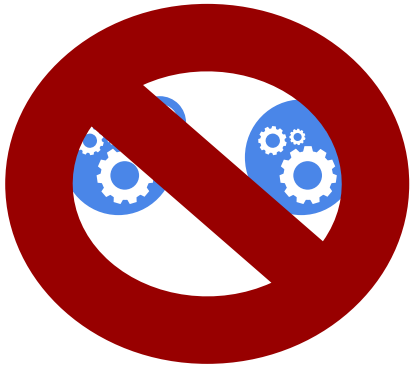
Hard pod anti-affinity didn’t allow more than one pod per failure domain

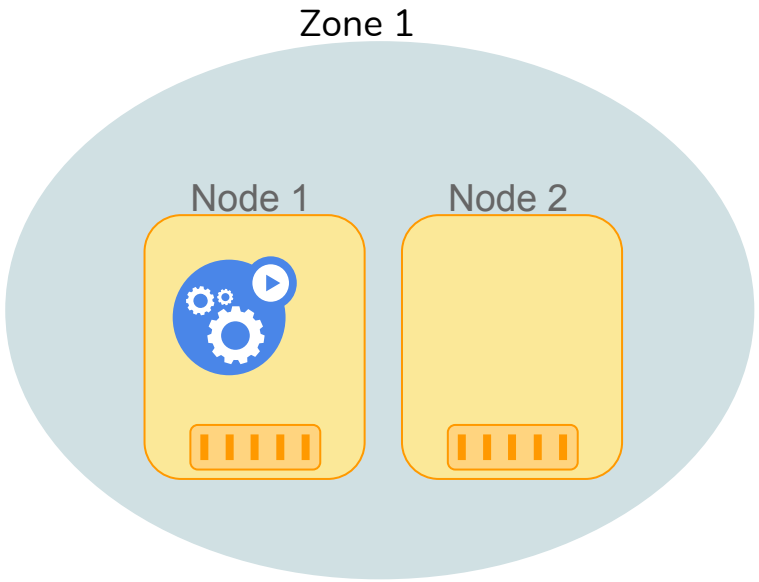


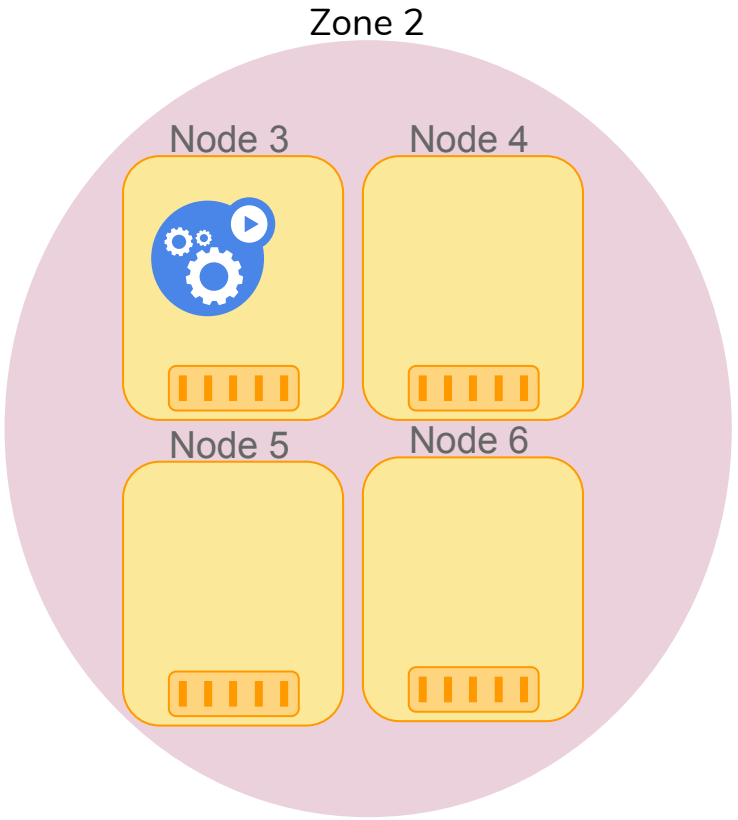


Even Pod Spreading

Hard pod anti-affinity didn’t allow more than one pod per failure domain

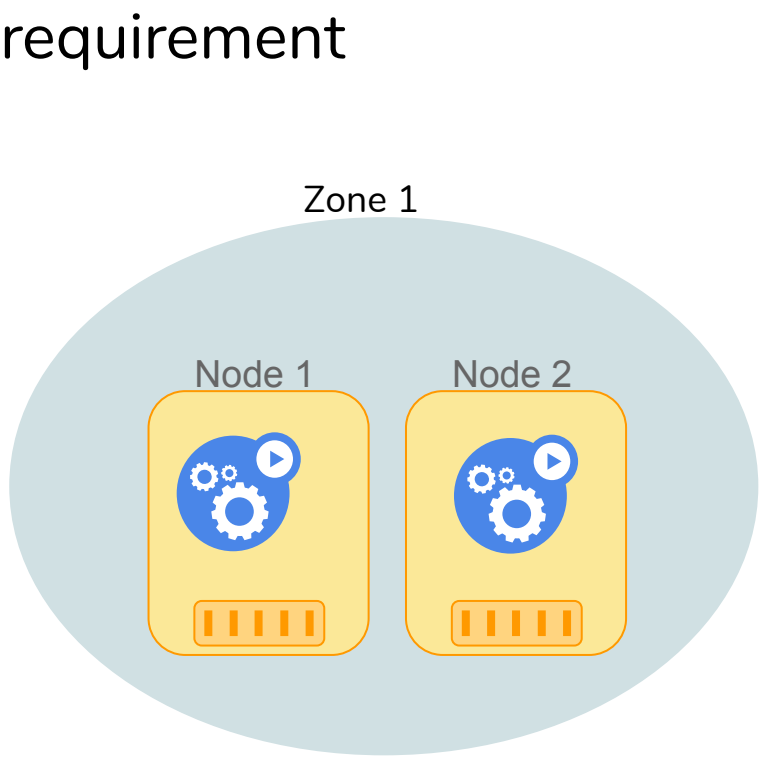


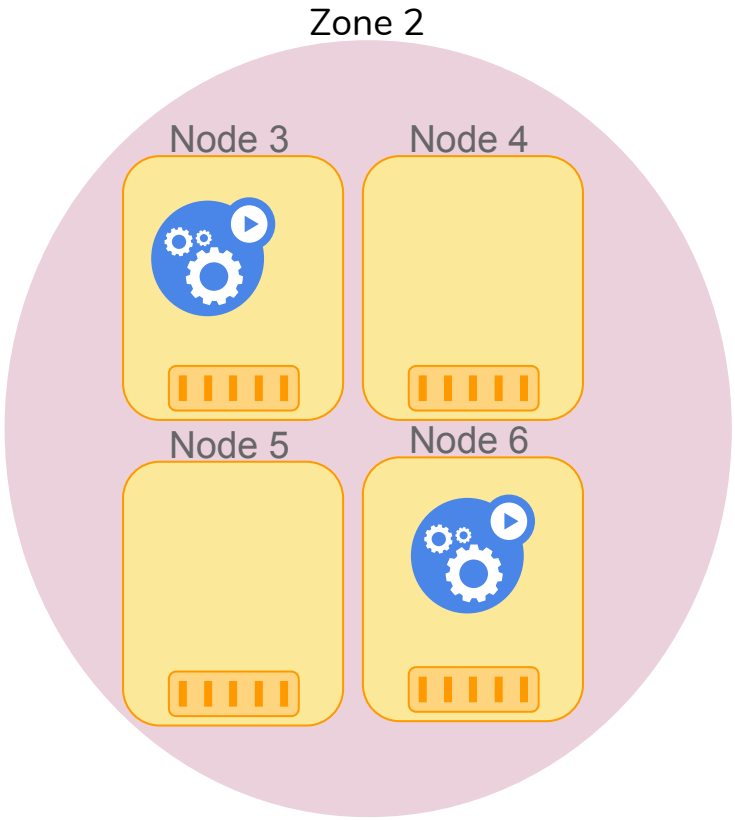




Even Pod Spreading

- Allows to spread pods in arbitrary topology domains, for example, zones, or nodes.
- Can be a hard or soft requirement





Scheduling Framework

- Highly customizable
- All scheduling features are converted into plugins
- Maintaining custom schedulers becomes easy
- Alpha version is planned for 1.15

imgur/funkblast1

Gang Scheduling (Coscheduling)

- Gang scheduling: schedule all members of a pod group or don’t schedule any of them
- Used extensively in batch processing.

- If a gang is partially scheduled none of the pods will progress. They will only waste processing resources.
- **Kube-batch** is an incubator project that has a proof of concept implementation
- We plan to make Gang Scheduling a standard feature.

Batch Scheduling

- Longer term projects
- Life-cycle management of batch workloads
- Supports dependency management among batch workloads
- Queue-jobs

Descheduler

- A cluster state changes over time and the scheduling decisions made in the past may no longer be optimal.
- Helps:
  - Rebalance node resources
  - Distribute pods of collections (ReplicaSet, Deployment, ...)
  - Apply inter-pod anti-affinity
  - Apply node affinity
- Is available in an incubator project.

Questions and Comments

Where to find us

- Chairs
  - @bsalamat
  - @k82cn
- Home page: https://github.com/kubernetes/community/tree/master/sig-scheduling
- Slack channel: https://kubernetes.slack.com/messages/sig-scheduling
- List: https://groups.google.com/forum/#!forum/kubernetes-sig-scheduling