KubeCon | CloudNativeCon

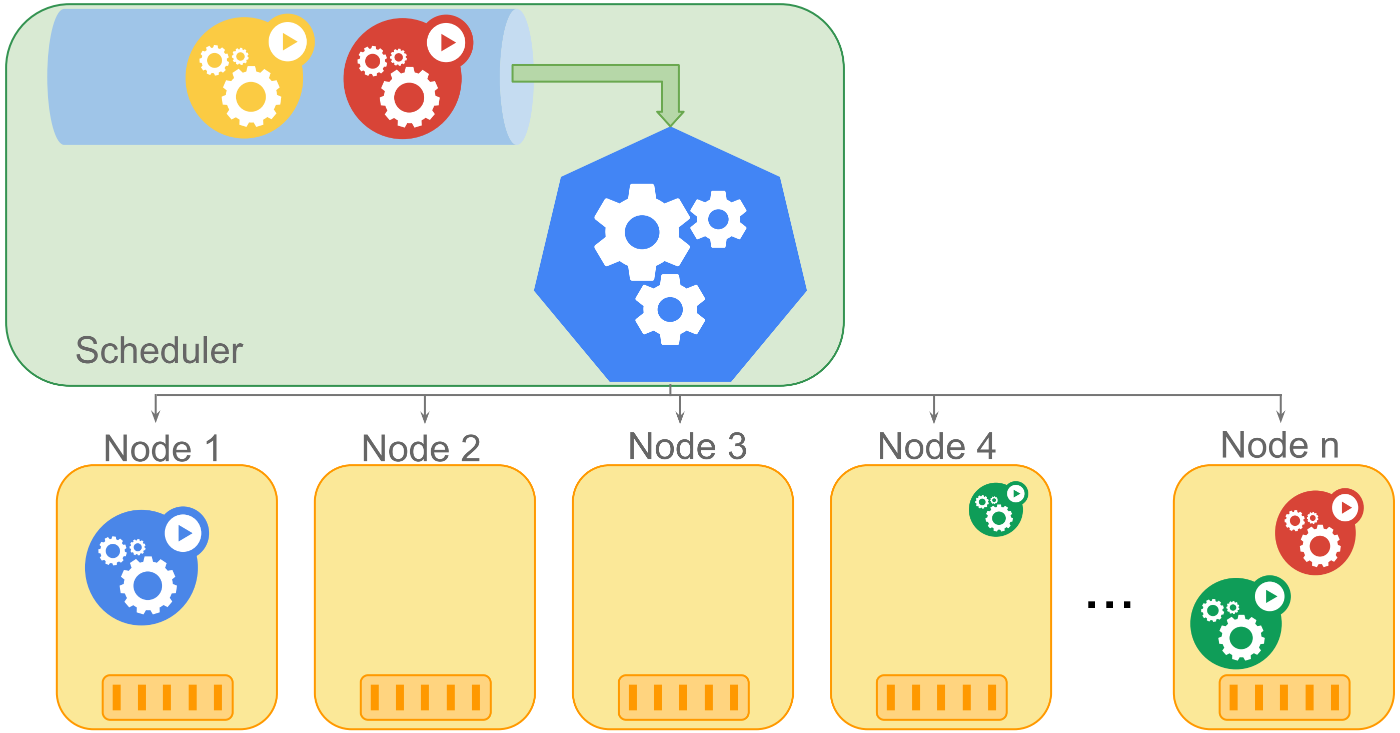
North America 2018

Deep Dive: SIG Scheduling

Babak "Bobby" Salamat, Google

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

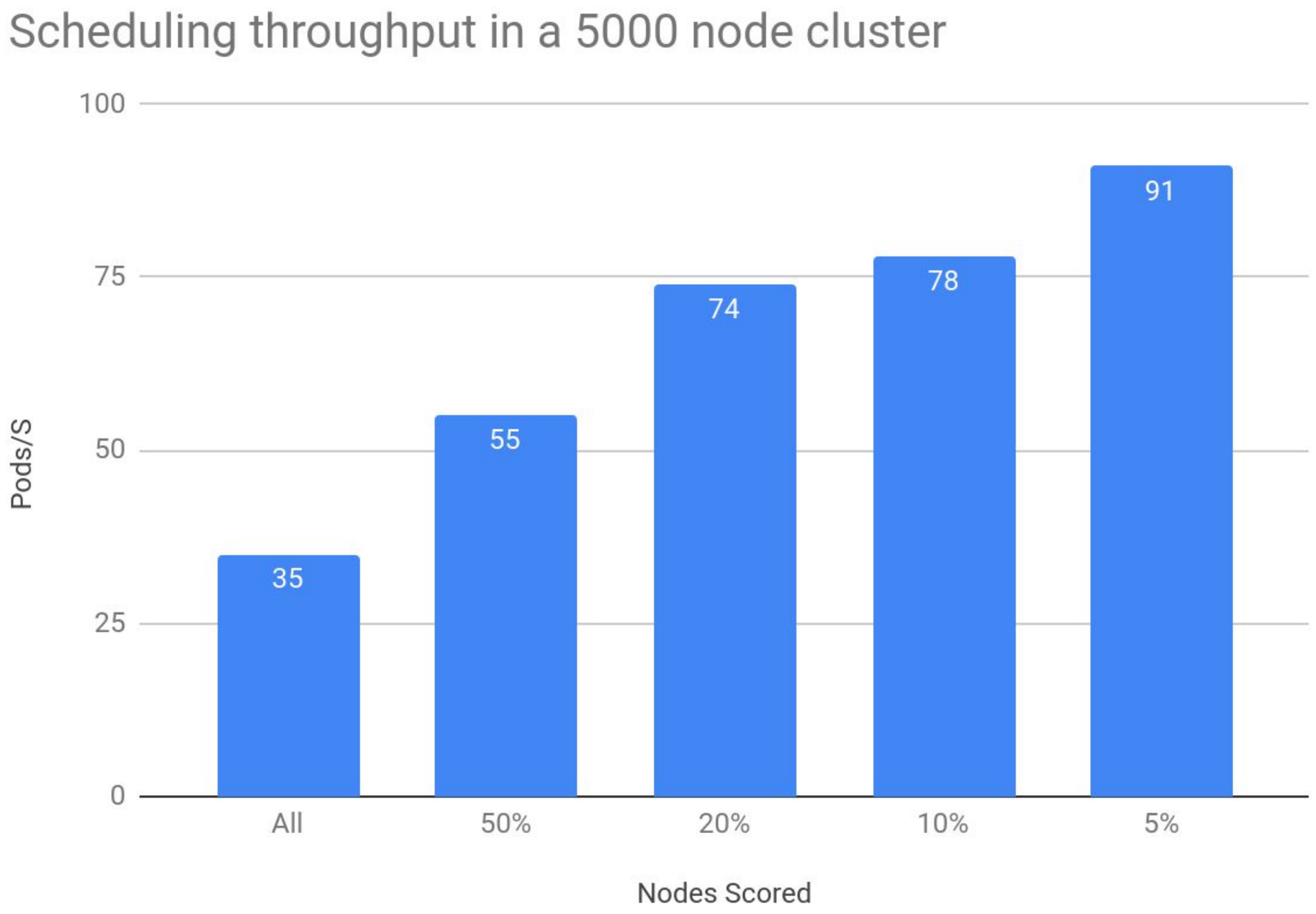
- Check node resources
- Spread Pods of a collection, such as a ReplicaSet, among nodes
- Support taints and tolerations
- Support node affinity
- Support inter-pod affinity
- Check node conditions, such as memory pressure, PID pressure, etc.
- Prefer nodes with lowest/highest levels of resource usage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

Recent Development

Recent Performance Improvements

Idea: stop scoring more nodes, once a certain percentage of nodes are found feasible

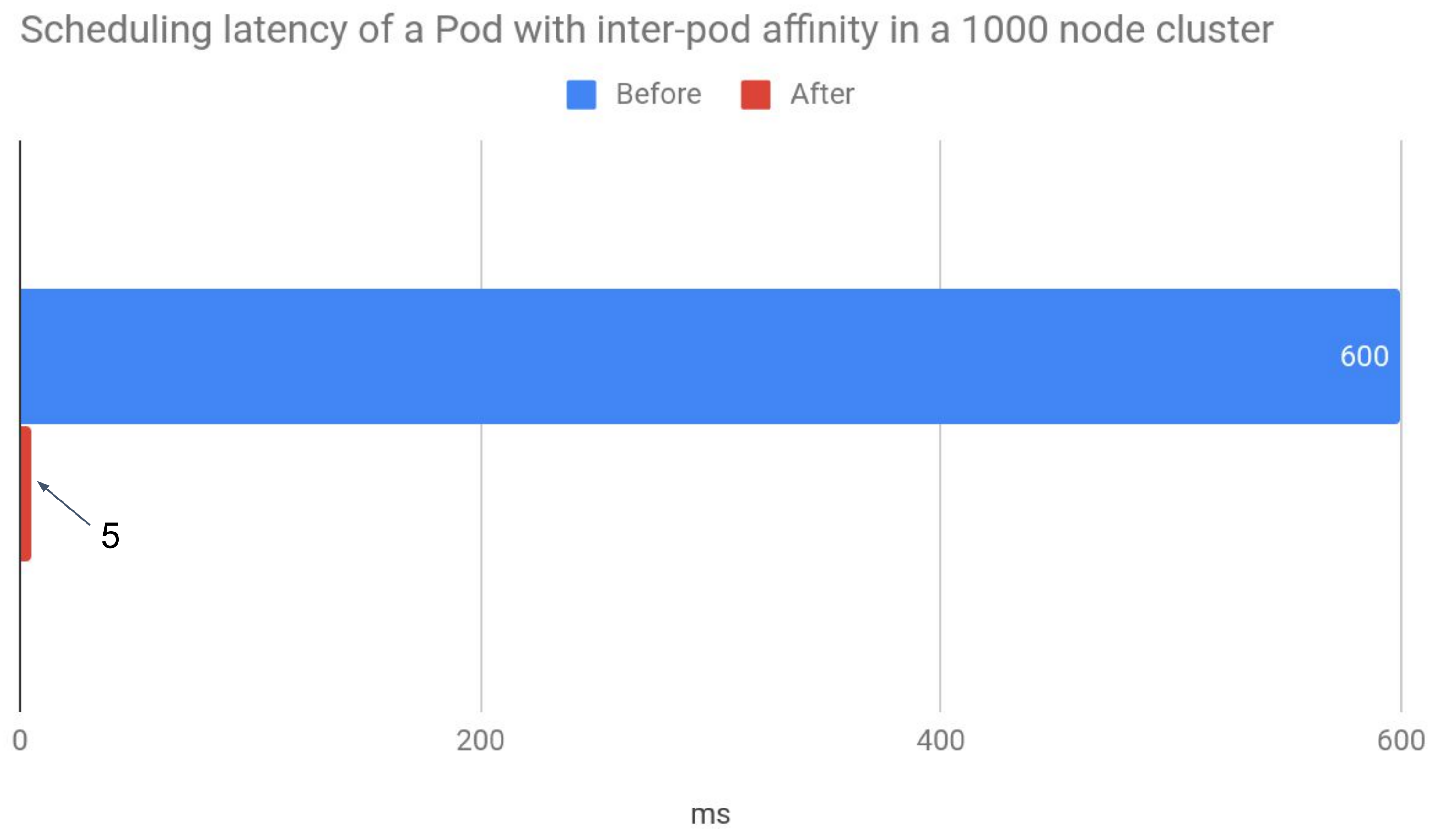
Achieves significant performance improvement in large clusters



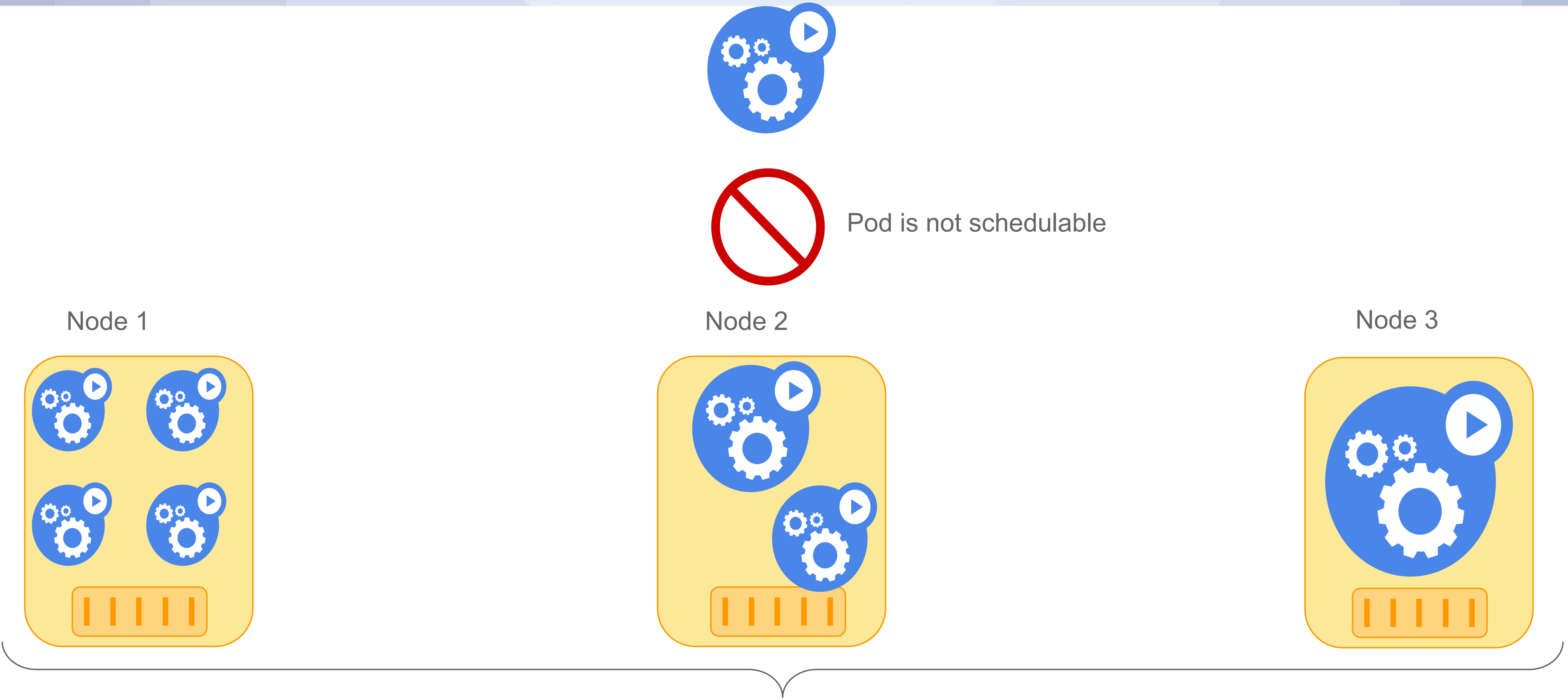
Recent Performance Improvements

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

We achieved 120X performance improvement by preprocessing and caching



Pod Priority and Preemption



Pod is not schedulable

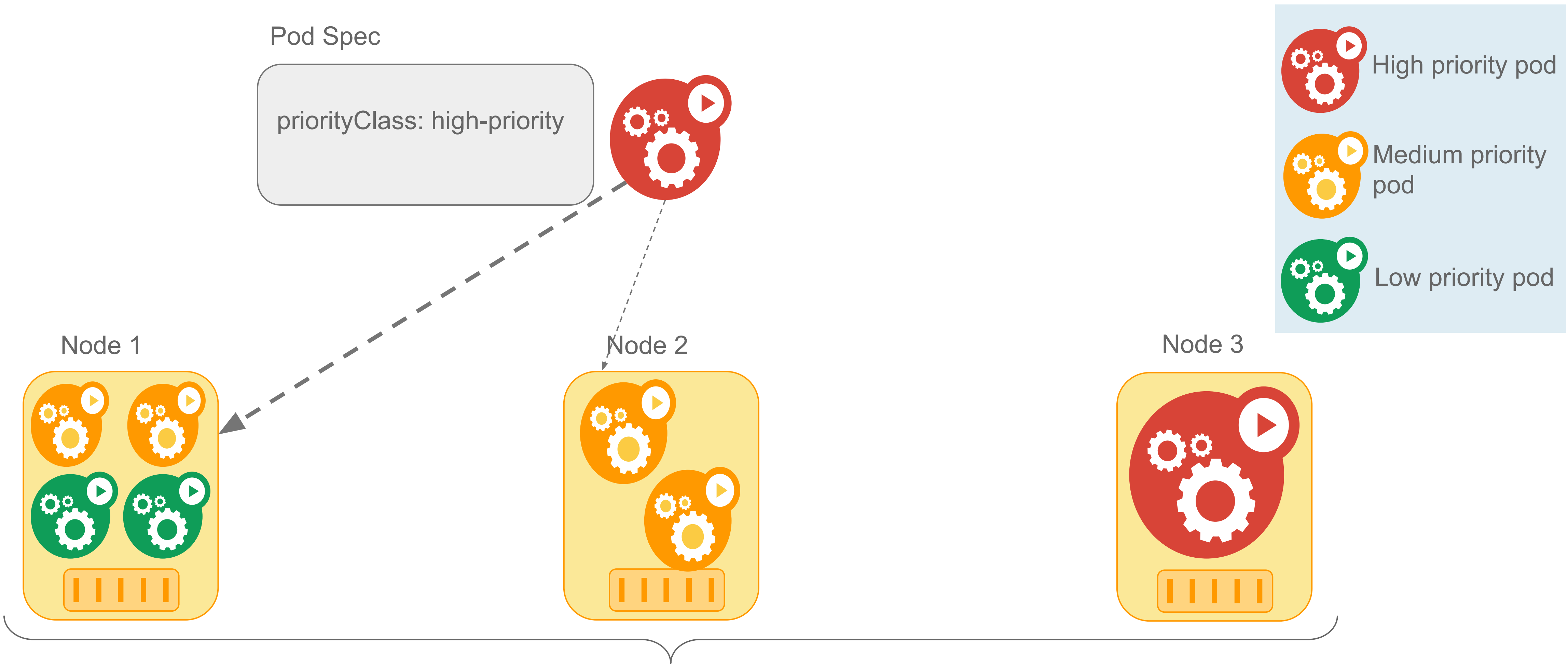
Node 1

Node 2

Node 3

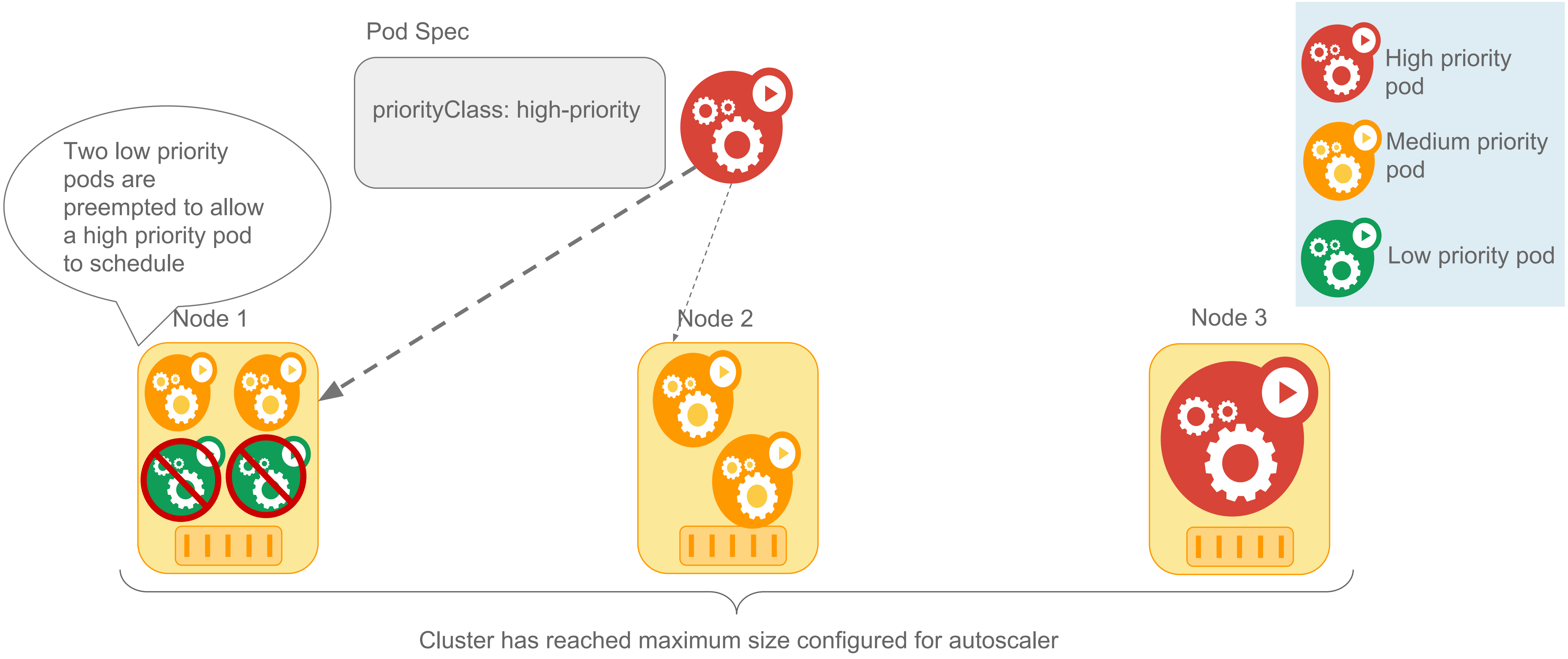
Cluster has reached maximum size configured for autoscaler

Pod Priority and Preemption



Cluster has reached maximum size configured for autoscaler

Pod Priority and Preemption



Pod Spec

priorityClass: high-priority

Two low priority pods are preempted to allow a high priority pod to schedule

High priority pod

Medium priority pod

Low priority pod

Cluster has reached maximum size configured for autoscaler

Planned Features

Gang Scheduling (Coscheduling)

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

ALL

or

NOTHING

- 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.

Pod Scheduling Policies

In a multi-tenant cluster, a user can add scheduling requirements that prevent other users from running their pods, or cause undesired placement of pods.

Pod Scheduling Policies

- Pod Scheduling Policies allow cluster admins to restrict certain namespaces.
- Policies can specify:
    - Allowed priority classes
    - Allowed tolerations
    - Allowed Pod anti-affinity
    - Required node selector/affinity
    - Required/allowed schedulers (in multi-scheduler clusters)

Scheduling Framework

- The scheduling framework provides a barebone of scheduling and almost all the features become plugins for the framework.
- Makes customizing the scheduler easy.
  - Customizations are contained in one or more plugins.
- A couple of extension points and the interface is already merged.
  - More to come in the next two releases.

imgur/funkblast1

Descheduler

- A cluster state changes as time passes 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.

Poseidon/Firmament Scheduler

- Poseidon is a Firmament based scheduler built for Kubernetes
- It achieves higher scheduling throughput than the default scheduler in certain scenarios.
- Targets batch and gang scheduling for starter
- It does not support all the Kubernetes features yet, but it supports most of them and is adding more.
- It is available in an incubator project

Questions and Comments