Apache YuniKorn: preemption for multi-tenant Kubernetes clusters

Wilfred Spiegelenburg
Craig Condit
AGENDA

Introduction YuniKorn

Kubernetes priority and preemption

Preemption in YuniKorn

Architecture - Deep Dive

Demo

Q & A
APACHE YUNIKORN

Introduction

YuniKorn capabilities:

• Diverse workloads
  – Services (long running)
  – Batch (short running and or recurring)
• Fast scheduling decisions
• Multi-tenancy
• Multiple deployment modes
  – Standalone
  – Plugin for K8s default scheduler

Workload Queueing
AGENDA

Introduction YuniKorn

**Kubernetes priority and preemption**

Preemption in YuniKorn

Architecture - Deep Dive

Demo

Q & A
KUBERNETES PREEMPTION AND PRIORITY

Preemption and priority limitations

Priority definition and use:
- PriorityClass defines name and value
- Cluster-wide definitions
- Scheduling: priority sort only, one cluster wide queue
- Any rogue user can create a pod with the highest defined priority

Preemption:
- Scheduling opt-in: use preemption to make space for this pod?
- Priority ranking only
- Opt-out from getting preempted by scheduler is NOT possible
AGENDA

Introduction YuniKorn

Kubernetes priority and preemption

Preemption in YuniKorn

Architecture - Deep Dive

Demo

Q & A
BATCH WORKLOADS

Why is preemption mission critical?

Static Queues

- Fixed distribution of all available resources
- Guaranteed to always have the resources
- **Caveat**: under-utilization of resources over time

Elastic Queues

- Oversubscription of available resources
- "Borrow" resources when not used by other queues
- **Caveat**: could starve other queues

Preemption can rebalance resources across queues
BATCH WORKLOADS

Why is preemption mission critical?

No preemption

- Queue1
- Queue2
- Guaranteed

With preemption

- Queue1
- Queue2
- Guaranteed
High-level overview

Queue resource config:
- **Maximum**: never more than this hard limit
- **Guaranteed**: amount always available for this queue
- Set and enforced at each level in the hierarchy

Preemption
- Adjust queue usage towards **guaranteed** resources
- Hierarchy aware
- Application aware
AGENDA

Introduction YuniKorn

Kubernetes priority and preemption

Preemption in YuniKorn

Architecture - Deep Dive

Demo

Q & A
LAWS OF PREEMPTION
The lesson learned from YARN

Before design and implementation: describe behaviour

1. Preemption policies are **strong suggestions**, **NOT guarantees**
2. Preemption can never leave a queue lower than its guaranteed capacity
3. A task **cannot preempt other tasks** in the **same** application
4. A task **cannot trigger preemption** unless its queue is **under** its guaranteed capacity
5. A task **cannot be preempted** unless its queue is **over** its guaranteed capacity
6. A task can only **preempt a task** with **lower** or **equal** priority
7. A task **cannot preempt** tasks **outside** its preemption fence (one-way constraint)
PREEMPTION SUPPORT

Design

YuniKorn preemption design

- Leverage existing K8s priority class object
- Integrated with the queue hierarchy
- Guaranteed resources configurable per queue
- Extended features:
  - Application aware: originator pod
  - Allow pods to request not to be preempted
  - Preemption delay
  - Fencing of queues in hierarchy
  - Priority offset
PREEMPTION SUPPORT

K8s Configuration

Annotated K8s PriorityClass

- YuniKorn preemption policy
- Allow preemption
  - opt-in: true
  - opt-out: false

```
apiVersion: scheduling.k8s.io/v1
kind: PriorityClass
metadata:
  name: high-priority
annotations:
  yunikorn.apache.org/allow-preemption: "true"
value: 1000
globalDefault: false
```
PREEMPTION SUPPORT

Queue configuration

Queue properties

- **Guaranteed resources**
  - Threshold for preemption
  - Preemption goal
- **Preemption Fence**
  - Queue scoped
  - Limit preemption to child queues
- **Preemption Delay**
  - Delays preemption to allow normal scheduling to place the request
  - Default: 30 seconds

```
queues:
- name: sales-ops
  parent: false
  resources:
    guaranteed: {memory: 24Gi, vcore: 6}
    max: {memory: 32Gi, vcore: 8}
  properties:
    preemption.policy: fence
    preemption.delay: 30s
    priority.policy: fence
    priority.offset: "1000"
```
PREEMPTION SUPPORT

Workflow

root

org

sales

fence

dev

prod

fence

fence

prod

fence

marketing

dev

prod

preempted low priority

high priority workload

over guarantee

below guarantee

over guarantee

below guarantee

system

high priority workload

below guarantee

preempted low priority

org

dev

prod

sales

marketing

fence
AGENDA

Introduction YuniKorn

Kubernetes priority and preemption

Preemption in YuniKorn

Architecture - Deep Dive

Demo

Q & A
DEMO

Prepared cluster

- Showing two pieces of functionality
  - Quotas
  - Preemption

- Kind cluster (1.28.0)
  - Plugin version deployed
  - 3 nodes (control-plane + 2 workers)
  - Hierarchical queues:
    - Complex structure defined (multi layer)
    - Different guaranteed resources
    - Fencing at different levels
AGENDA

Introduction YuniKorn

Kubernetes priority and preemption

Preemption in YuniKorn

Architecture - Deep Dive

Demo

Q & A