Observable resource usage for cloud workloads

Wilfred Spiegelenburg
Peter Bacsko
AGENDA

Introduction YuniKorn

- Quotas
- Tracking resource usage
- Historical resource usage
- Demo
- Q & A
Introduction

APACHE YUNIKORN

Workload Queueing

Gang Scheduling

Application Sorting
AGENDA

Introduction YuniKorn

Quotas

Tracking resource usage

Historical tracking

Demo

Q & A
QUEUETYPE HIERARCHY & QUOTAS

Hierarchical model

• Resources are managed via queues
  - Auto created queues
  - Static queues
• Resource located in leafs
  - Propagate up to the root
  - Requests (pending)
  - Allocation (assigned to node)
• Quota
  - Set per queue (also for users)
  - Enforced at each level in the hierarchy
QUEUE HIERARCHY & QUOTAS

Example

- **root**
  - Resource limit: 60 GB / 4 CPU
  - Usage: 50 GB / 5 CPU

- **development**
  - Resource limit: 20 GB / 2 CPU
  - Usage: 20 GB / 2 CPU

- **production**
  - Resource limit: 30 GB / 3 CPU
  - Usage: 30 GB / 3 CPU

- **team1**
  - 10 GB / 1 CPU
  - 10 GB / 1 CPU

- **team2**
  - 10 GB / 4 CPU

- **sales**
  - 30 GB / 3 CPU
AGENDA

Introduction YuniKorn

Quotas

**Tracking resource usage**

Historical resource usage

Demo

Q & A
TRACKING RESOURCE USAGE

How and what is tracked

- Allocation tracking as part of the scheduling cycle
- Asynchronous updates for finished Allocations
- Scheduler only cares about current state

Diagram:
- Queue
- Try
- Application
- Try
- Node
- Pass
- Current state
  - Quotas
  - Application
  - Node

Finished allocation

Try

Fail
TRACKING RESOURCE USAGE
Deep dive into tracking data...

Quotas:
• Usage tracked for quota enforcement:
  – Queue
  – User
  – Group
• Hierarchical: enforced during scheduling cycles

Application
• Usage tracking only

Node
• Usage must always fit in available resources
AGENDA

Introduction YuniKorn

Quotas

Tracking resource usage

Historical resource usage

Demo

Q & A
HISTORICAL RESOURCE USAGE
Additional detail needed

- YuniKorn is stateless
  - re-init current state on startup
  - guarantees same state as previous run
- Scheduler only cares about current state
- Reusable identifiers
  - YuniKorn instance identification
  - Node is known by name only
  - Application ID may be reused
- Configuration changes over time
  - Queues
  - Quotas
USAGE INFORMATION OVER TIME

Event System

Design guidelines
- Low memory consumption
- YuniKorn remains stateless
- No graphical user interface
- Existing events system

Additional details
- States (application)
- Requests (application)
- Configuration changes (queues)
- Cluster changes (nodes)
EVENT SYSTEM
Usage information over time

si.EventRecord {
  Type:
  ChangeType:
  ChangeDetail:
  TimestampNano:
  ObjectID:
  ReferenceID:
  Resource:
  Message:
}

eventRecordType, eventChangeType, eventChangeDetail, int64, String, String,

REQUEST_ALLOC = 101  // Request allocated
APP_ALLOC = 200  // Allocation changed
APP_REQUEST = 201  // Request changed
APP_REJECT = 202  // Application rejected on create
APP_RUNNING = 206  // State change to running

NODE_ALLOC = 303  // Allocation changed
NODE_CAPACITY = 304  // Capacity changed

QUEUE_CONFIG = 400  // Managed queue update or removal
QUEUE_MAX = 403  // Max resource changed

REQUEST = 1
APP = 2
NODE = 3
QUEUE = 4
NONE = 0
SET = 1
ADD = 2
REMOVE = 3

examples
EVENT SYSTEM
Integration in the scheduling cycle

- queue
  - application
    - node
      - current state

- configuration changes
- cluster changes
- requests & states

- REST API
  - quotas
  - application
  - node

- try
- try
- pass
- fail
- finished allocation
EVENT SYSTEM

Limitations and possible future developments

Limitation
- REST API: all or nothing
- No user or group events

Future
- Summarised usage
- Metrics collection replacement
- Streaming support
- Add on service
  - Storage
  - Filtered events (REST)
  - De-duplicate events from restart
AGENDA

Introduction YuniKorn

Quotas

Tracking resource usage

Historical resource usage

Demo

Q & A
DEMO

Prepared cluster

- Showing multiple pieces of functionality
  - Quotas
  - Preemption
  - EventSystem

- Kind cluster (1.28.0)
  - Plugin version deployed
  - 3 nodes (control-plane + 2 workers)
  - Hierarchical queues, leafs defined:
    - Same quota (maximum usage)
    - Different *guaranteed* resources
AGENDA

Introduction YuniKorn

Quotas

Tracking resource usage

Historical resource usage

Demo

Q & A
Website: https://yunikorn.apache.org
Email: dev@yunikorn.apache.org
Slack: YuniKorn Slack

THANK YOU