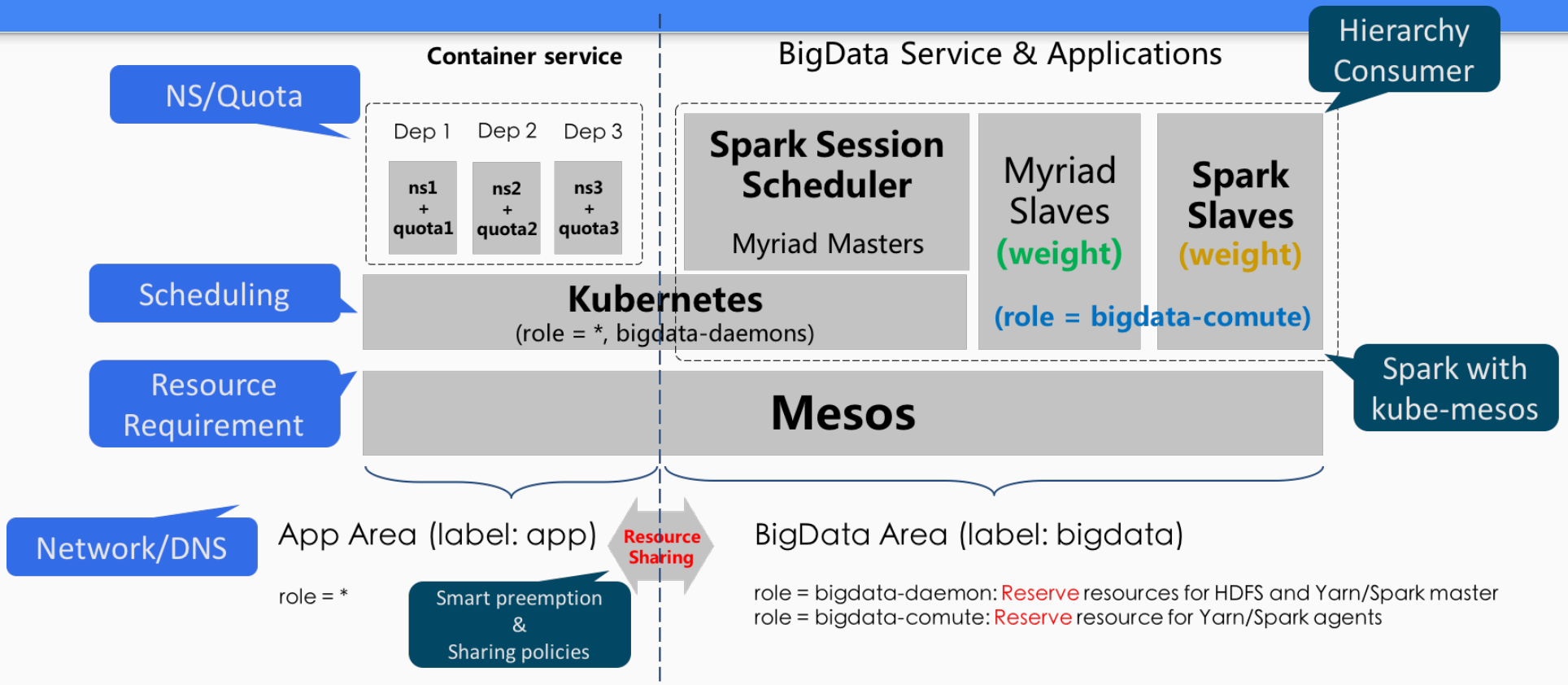
Kuberentes#269: Batch Job Admission and flexible resource allocation (基于策略的资源共享)

@k82cn, 马达.IBM

User Cases: Run multiple type of workloads in DataCenter



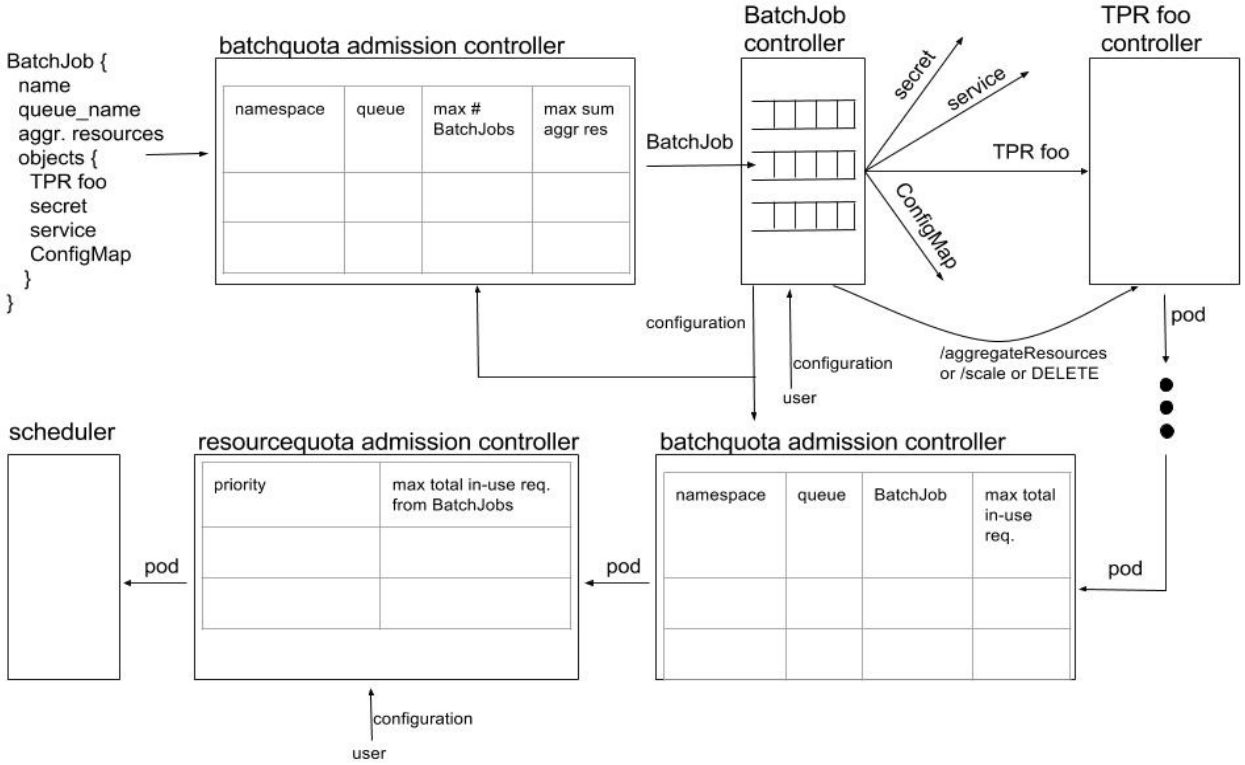
User Cases: Run multiple type of workloads in DataCenter

- Long running service (app area) & bigdata (bigdata area) can share resources:
  - Support define resource usage of each area, e.g. 50% resources to app area, 50% to bigdata area.
  - Support **borrow/lending** protocol: if the resources is idle in one area, it can be lend out and be preempted back when launch more tasks
- Run multiple cluster in bigdata area, e.g. Hadoop & Spark:
  - Support define resources usage of each cluster within bigdata area
  - Support sharing resources between those clusters

Kubernetes features & gaps

- Admission Controller + Quota: static plan / allocation
- Multiple Scheduler: No QoS
- Auto-Scaling & Node-level QoS: no cluster-level QoS
- Re-scheduling & Preemption/Eviction
- Workload-specific controller & ThridPartyResources

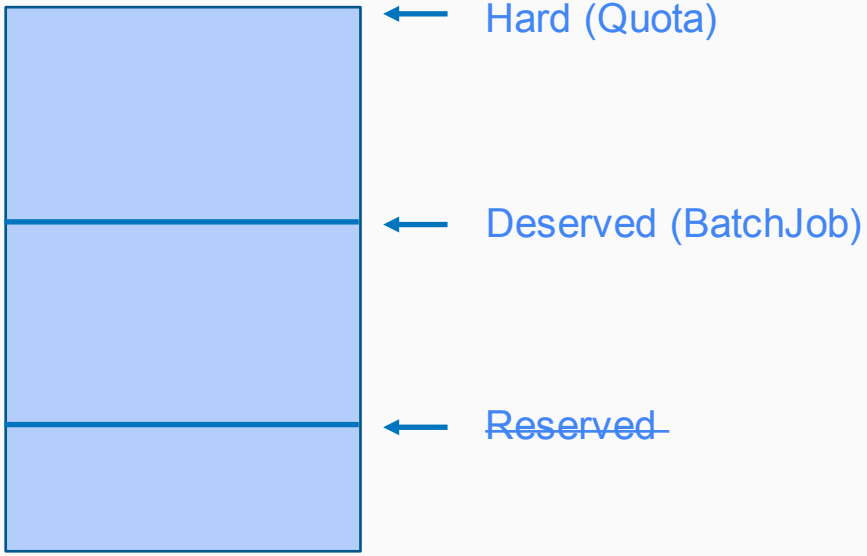
Architect Overview



1. BatchJobController will calculate **deserved** resource based on Scheduler's configuration, arbitrator's policy, e.g. DRF, and namespace's request
2. BatchJobController will evict Pods of overused namespace
3. Scheduler dispatch tasks based on **Quota** (# of deserved), **Pods** and **Nodes attributes**
4. BatchJob Admission make sure "namespace will not be overused"

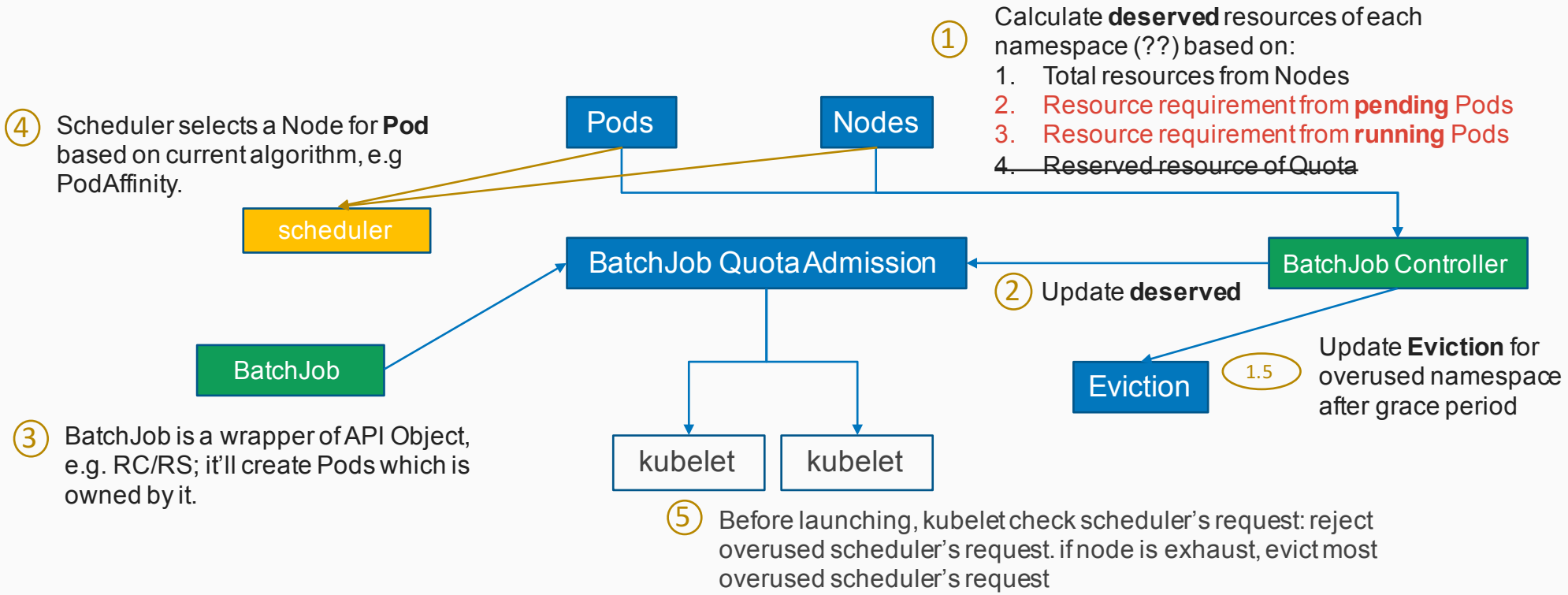
Architect Overview

Quota & BatcJobQuota (??)

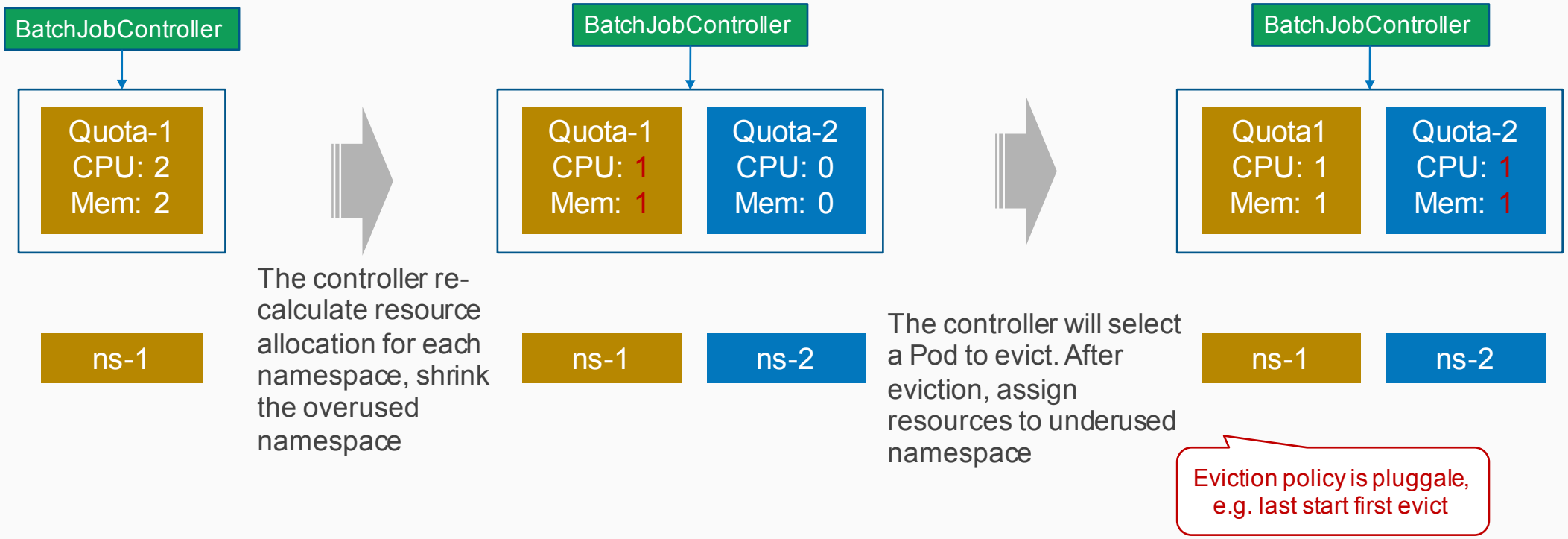


1. The **reserved** section defines the resources that reserved for the namespace. The total reserved resources can not exceed cluster resources
2. The **deserved** is updated by BatchJobController instead of user, it defines the total resources that allocated to a namespace; the deserved resources can not exceed **Quota.hard** and can not less than ~~**Reserved**~~ (exception excluded, e.g. Node failed)

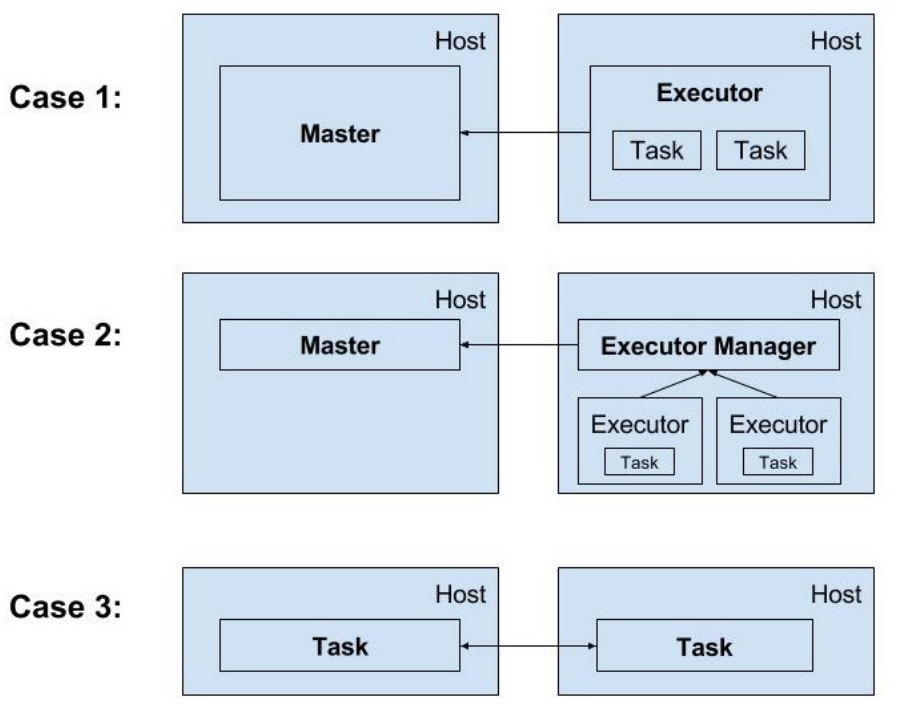
Architect Overview



Pre-emption & Reclaim in BatchJob Controller



Resource Requirements of BatchJob



- **Case 1:** The resource requirement is sum of each executor (the number of total tasks is unknown)
- **Case 2:** The resource requirements is sum of non-terminal tasks (pending + running), but the total number task is unknown
- **Case 3**: The resource requirement is sum of total tasks

Backlog

- BatchJob controller policy & configuration (e.g. DRF ??, weight of namespace)
- HA of BatchJob Controller
- Hierarchical namespace
- BatchJob controller policy for limits (Resource QoS)
- Hostname in deserved ??

Reference

- Design Doc: https://docs.google.com/a/google.com/document/d/1-H2hnZap7gQivcSU-9j4ZrJ8wE_WwcfOkTeAGjzUyLA/edit?usp=sharing
- Related Issues:
  - Manage multiple applications in Kubernetes (#36716)
  - Peemption/priority schema (#22212)
  - Rescheduler

Thank You !