Kubernetes 运行大数据工作负载的探索和实践

Leibo Wang(wang.platform@Hotmail.com)
Huawei CloudBU Principal Engineer

架构师成长路径指南

扫码查看

持续提升 I 初级	技术进阶 I 中级	能力拓展 I 高级
邱岳的产品手记	许式伟的架构课	微服务架构实战 160 讲
微服务架构核心 20 讲	从 0 开始学微服务	Linux 性能优化实战
MySQL 实战 45 讲	技术管理实战 36 讲	左耳听风
从 0 开始学架构	Elasticsearch 核心技术与实战	Spring Boot 与 Kubernetes 云原生微服务实践

SPEAKER

INTRODUCE

王雷博　Principal Software Engineer

- Huawei(Now) – Cloud Native batch system (Volcano) development
- IBM spectrum computing – Cluster resource and workload scheduling platform development

Agenda

- Why Spark on Kubernetes
- Gaps for Spark
- Volcano solution for Spark
- Future works

Why Spark on Kubernetes

Kubernetes extends beyond container orchestration, it has been expanded to support for data-intensive and stateful apps.

Benefit:

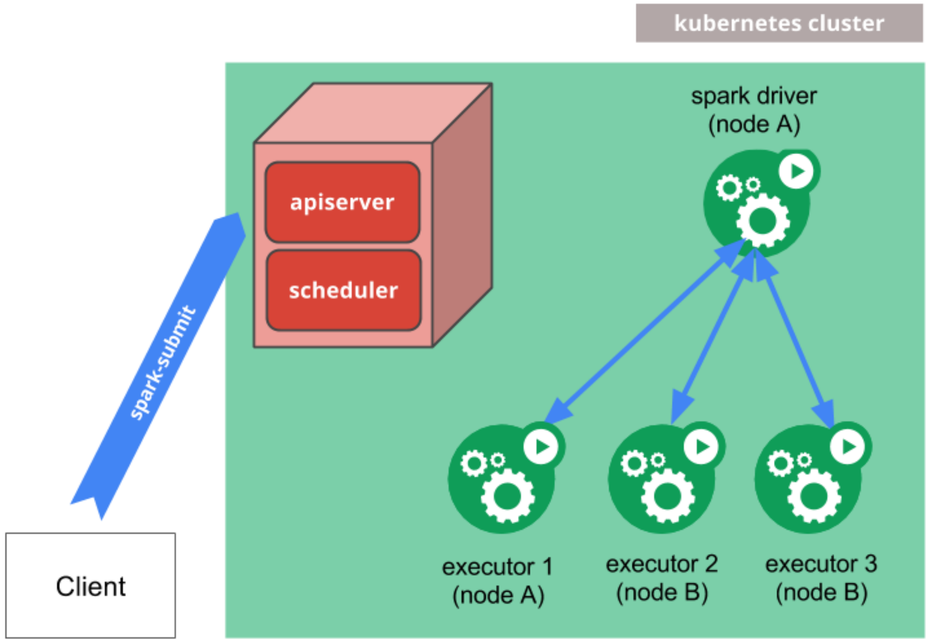
- Autoscaling in Cloud
- Consolidate online service and offline analysis
- Ecosystem(Monitor, logging etc)
- Fine grained resource isolation
- ……

About Spark on Kubernetes

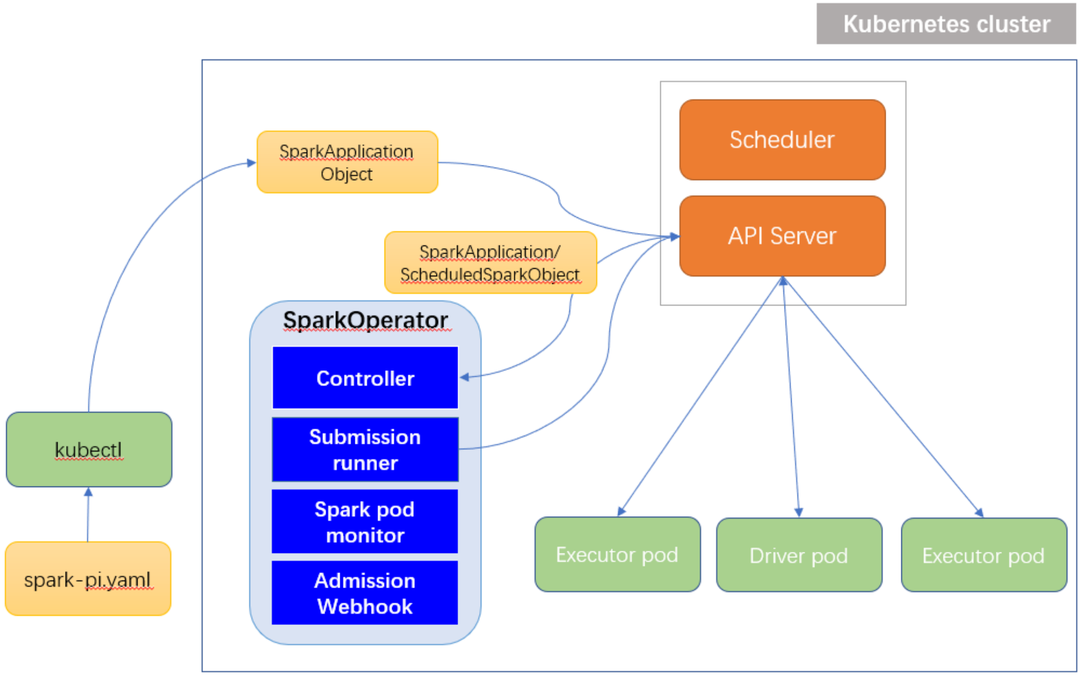
- https://github.com/apache-spark-on-k8s/spark
- The goal is to bring native support for Spark to use Kubernetes as a cluster manager like YARN, or Mesos.
- Spark 2.3 added native support for Kubernetes.
- Spark 2.4 added support for client mode, R, python etc.
- Spark 3.0 will add support for dynamic resource allocation, external shuffle service, Kerberos etc.

How it works

Spark on Kubernetes



Spark-operator



Gaps for Spark

- **Resource Management:**
  - Queue
  - Hierarchical queue
  - Fair-share
  - Preempt/Reclaim
  - …
- **Scheduler**
  - Job preemption
  - Fair-share scheduling
  - Queue scheduling
  - Resource reservation
  - Binpack
  - Task topology
  - Zone aware scheduling
  - …

- **Dynamic Resource Allocation**
- **Spark external shuffle service**
- **Performance**
- **Security**
  - Kerberos support
- **…**

Volcano: A Kubernetes native batch system

Website: https://volcano.sh

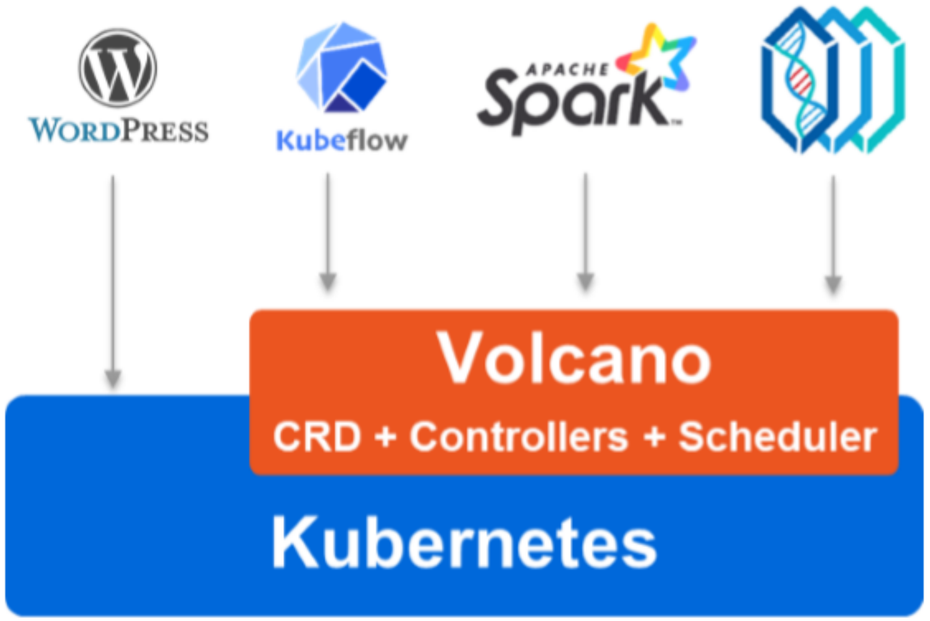
Github: http://github.com/volcano-sh/volcano

Twitter: https://twitter.com/volcano_sh

Slack: http://volcano-sh.slack.com

Email: volcano-sh@googlegroups.com

Architecture



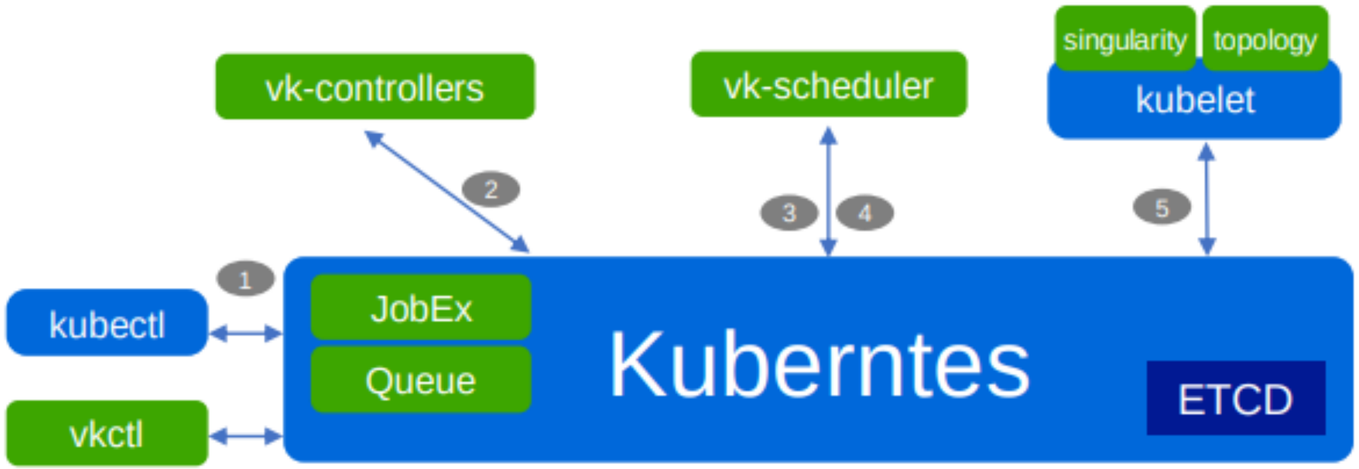
Domain frameworks:

- Deployment/Installation of framework in k8s
- Map framework's terms/concepts into common concept, e.g. Job, Queue
- Enable related features for frameworks, e.g. gang-scheduling for TensorFlow training

Common Service for AI, BigData, Gene, etc:

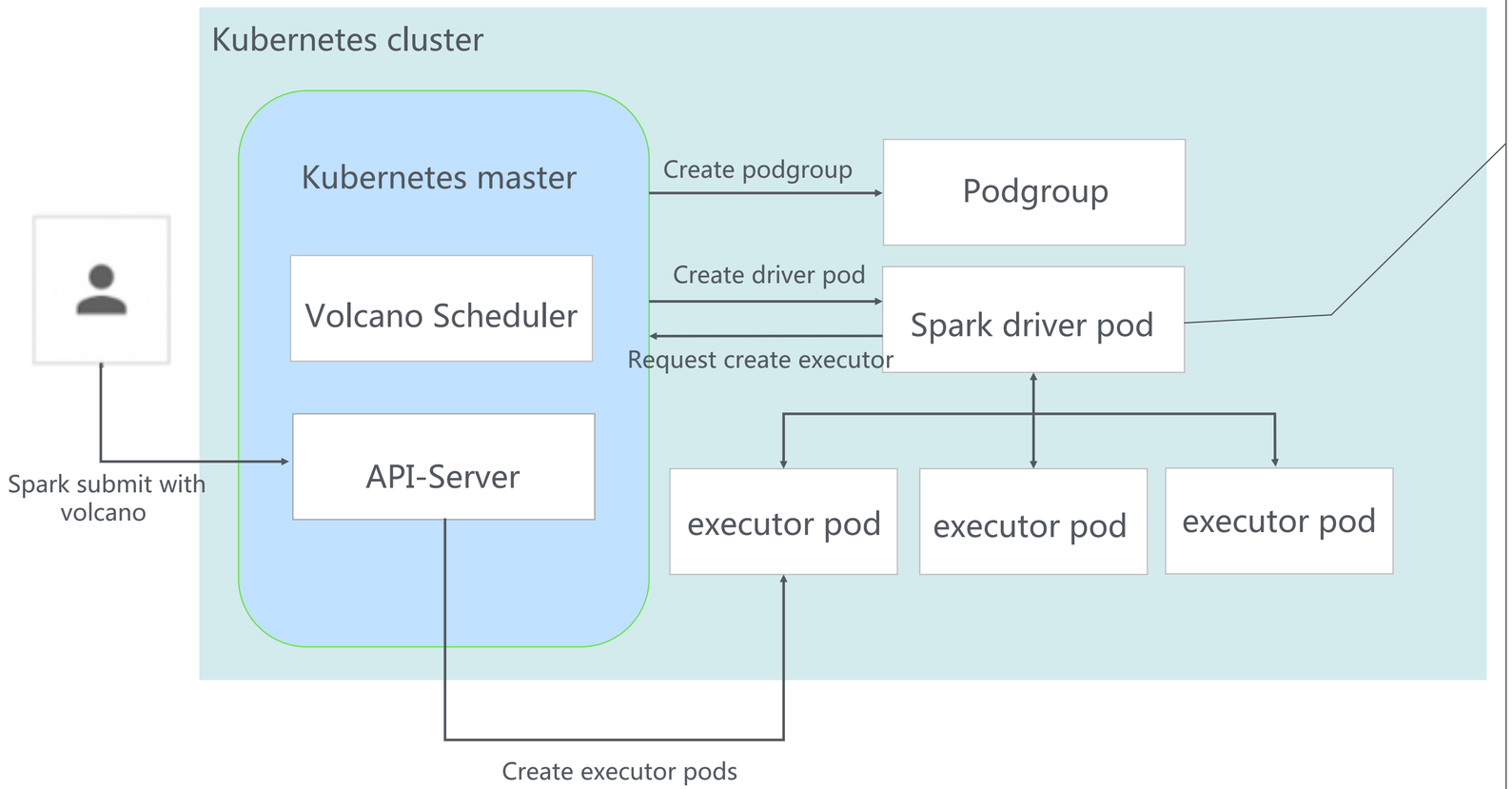
- Batch scheduling, e.g. fair-share, gang-scheduling
- Enhanced job management, e.g. multiple pod template, error handling
- Accelerator, e.g. GPU, FPGA
- kubectl plugins, e.g. show Job/Queue information

Architecture



1. KubectI creates a JobEx object in apiserver if all admission passed.
2. JobExController create Pods based on its replicas and templates.
3. vk-scheduler get the “notification” of Pod from apiserver.
4. vk-scheduler chooses one host for the Pod of JobEx based on its policy.
5. kubelet gets the notification of Pod from apiserver and then start the container.

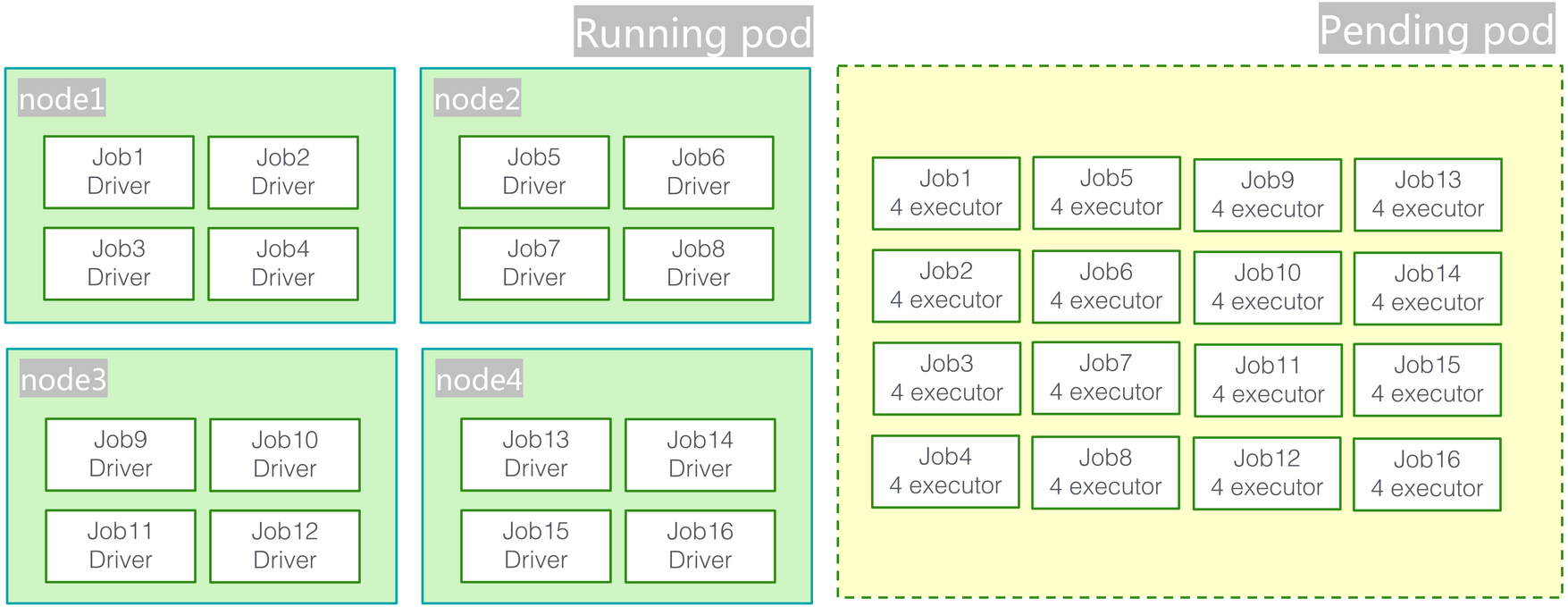
Spark on Kubernetes with volcano




```
apiVersion：v1
kind:  Pod
metadata:
    annotations:
      scheduling.k8s.io/group-name: job-1574739729783-podgroup
      volcano.sh/task-spec: spark-driver
    createTimestamp:  ”2019-11-27T09:33:19Z”
    labels:
     spark-app-selector:spark-6cc54577d7254b2d84924500375112f7
      spark-role: driver
    name: job-1574739729783-driver
    namespace: default
    resourceVersion:  “12093805”
    selfLink: /api/v1/namespaces/default/pods/job-1574739729783-driver
    uid: f26a81f3-10f8-11ea-938f-fa163eddd2ce
Spec:
   containers:
…
```

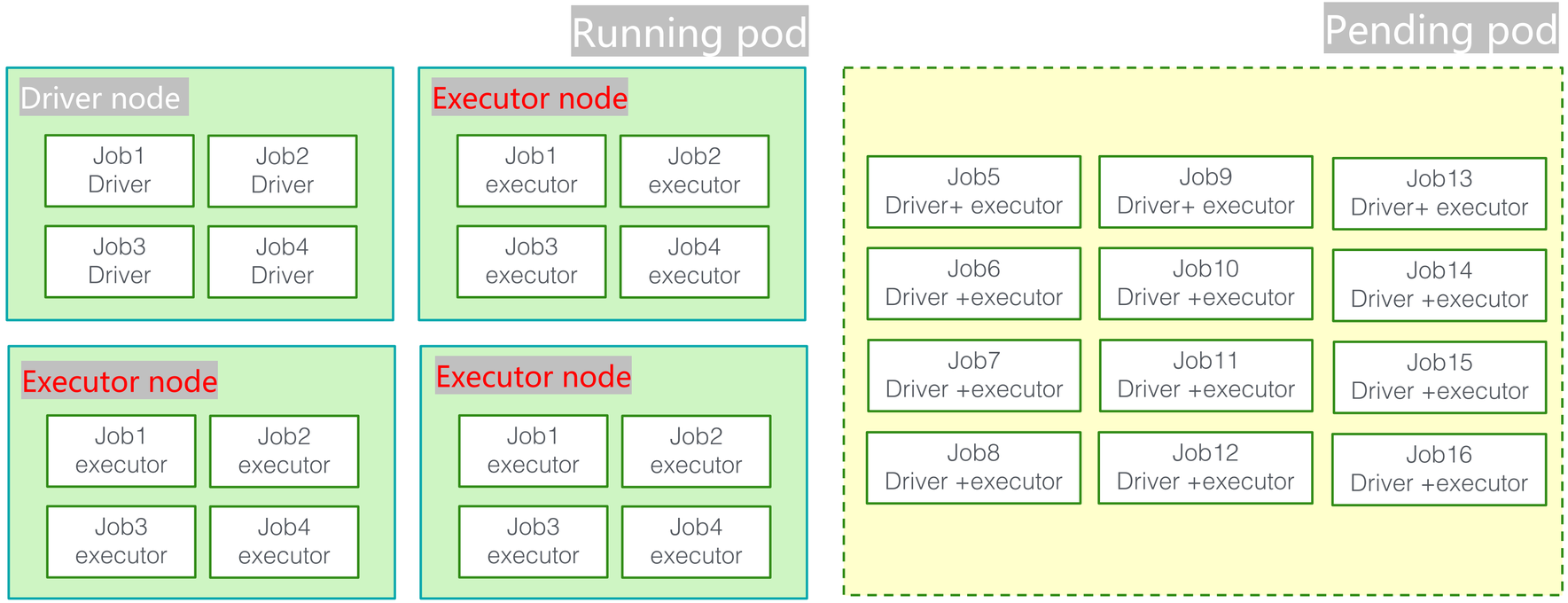
Scenario : High concurrent job submission

Submit 16 spark jobs (1 driver +4 executor) in cluster (16core)



Scenario : High concurrency job submission

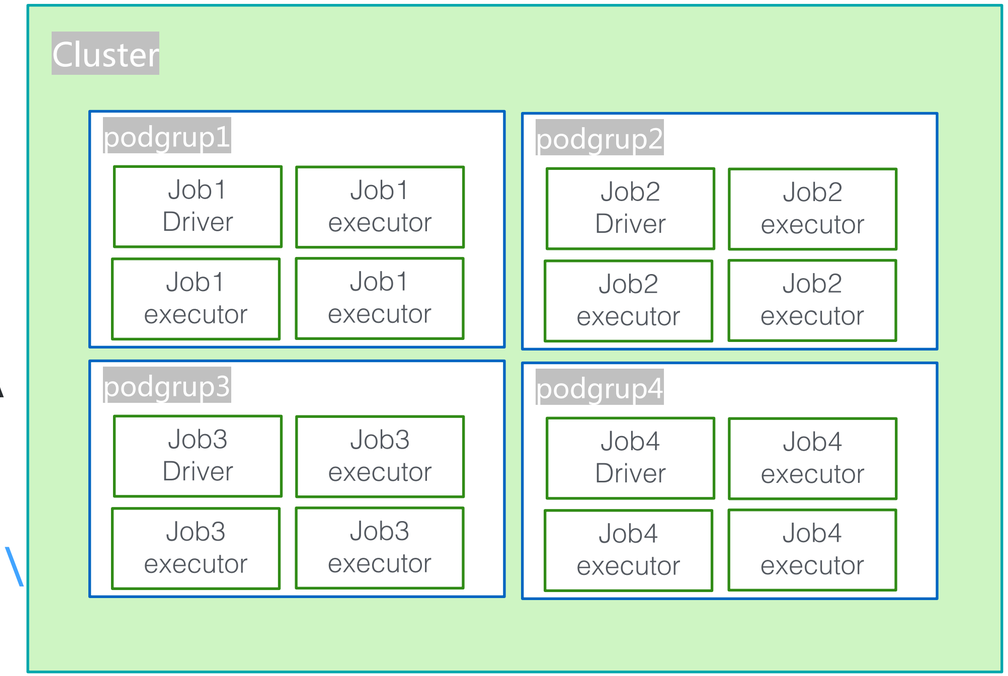
Separate cluster into driver and executor pools and submit 16 Jobs (1 driver + 3 executor)



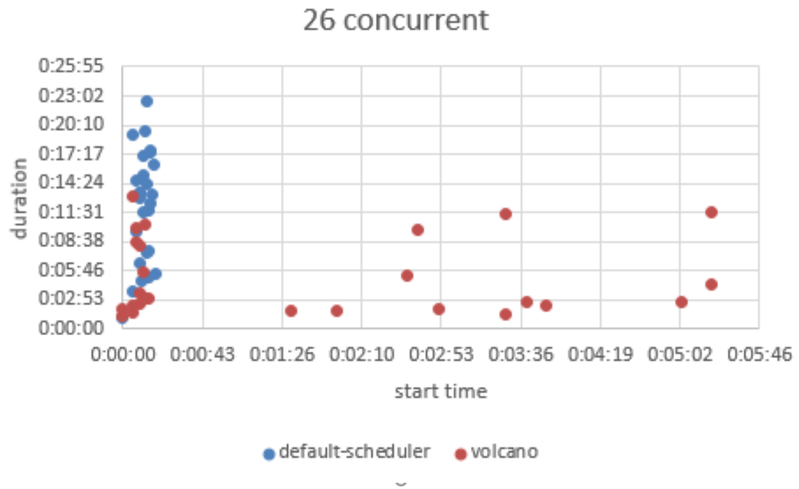
Solution: Pod delay creation

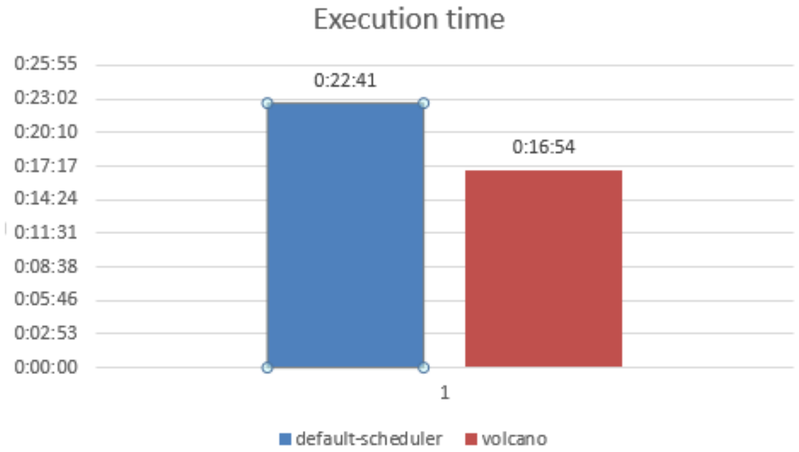
Submit via spark-Submit

```
spark-submit \
--master k8s://https://192.168.45.93:5443 \
--deploy-mode cluster \
--name query \
--class com.databricks.spark.sql.perf.BenchmarkQuery \
--conf spark.kubernetes.volcano.enable=true \
--conf spark.kubernetes.volcano.podgroup.cpu=5 \
--conf spark.kubernetes.volcano.podgroup.memory=10g \
...
```

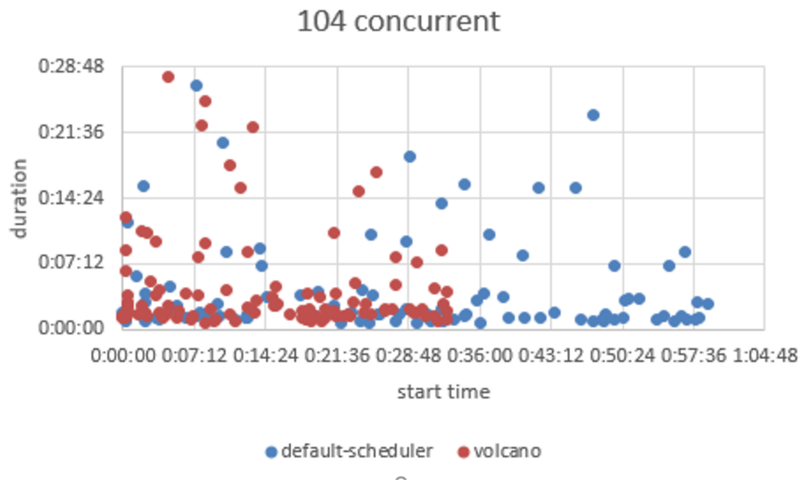


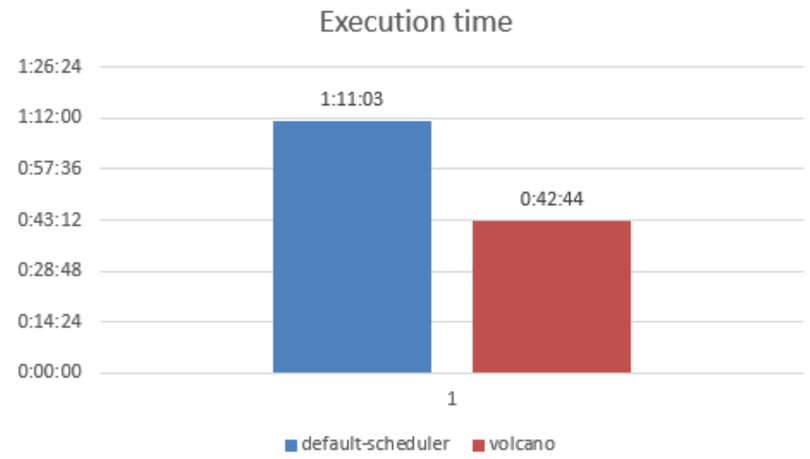
Solution: Pod delay creation





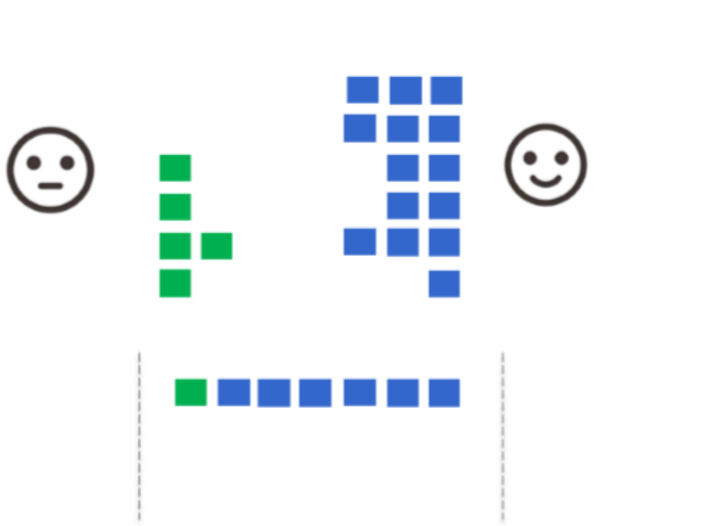
- Spark-sql-perf(TPC-DS, master)
- 26 concurrent
- 4 nodes(8core, 64G, HDD)
- Kubernetes 1.13



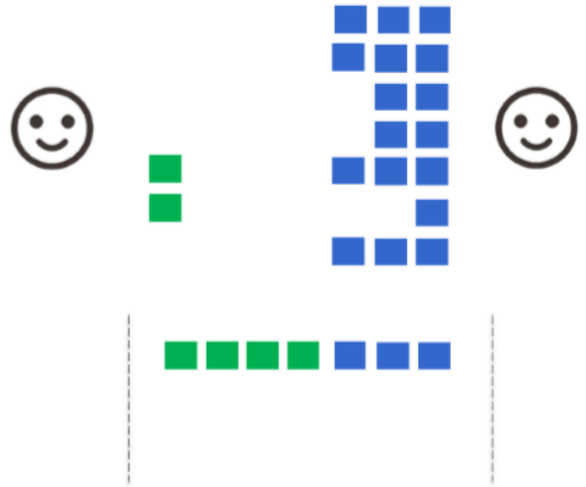


- Spark-sql-perf(TPC-DS, master)
- 104 concurrent
- 4 nodes(8core, 64G, HDD)
- Kubernetes 1.13
- 1 driver Node, 3 executor node for default scheduler

Scenario: resource fair-share



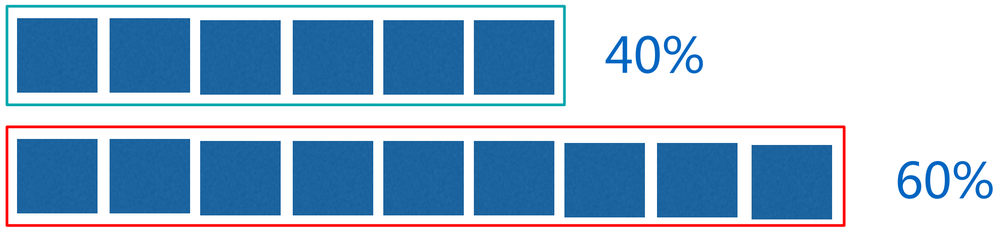
The more workload, the more resources???



Share resources by weight !!!

Solution: resource fair-share

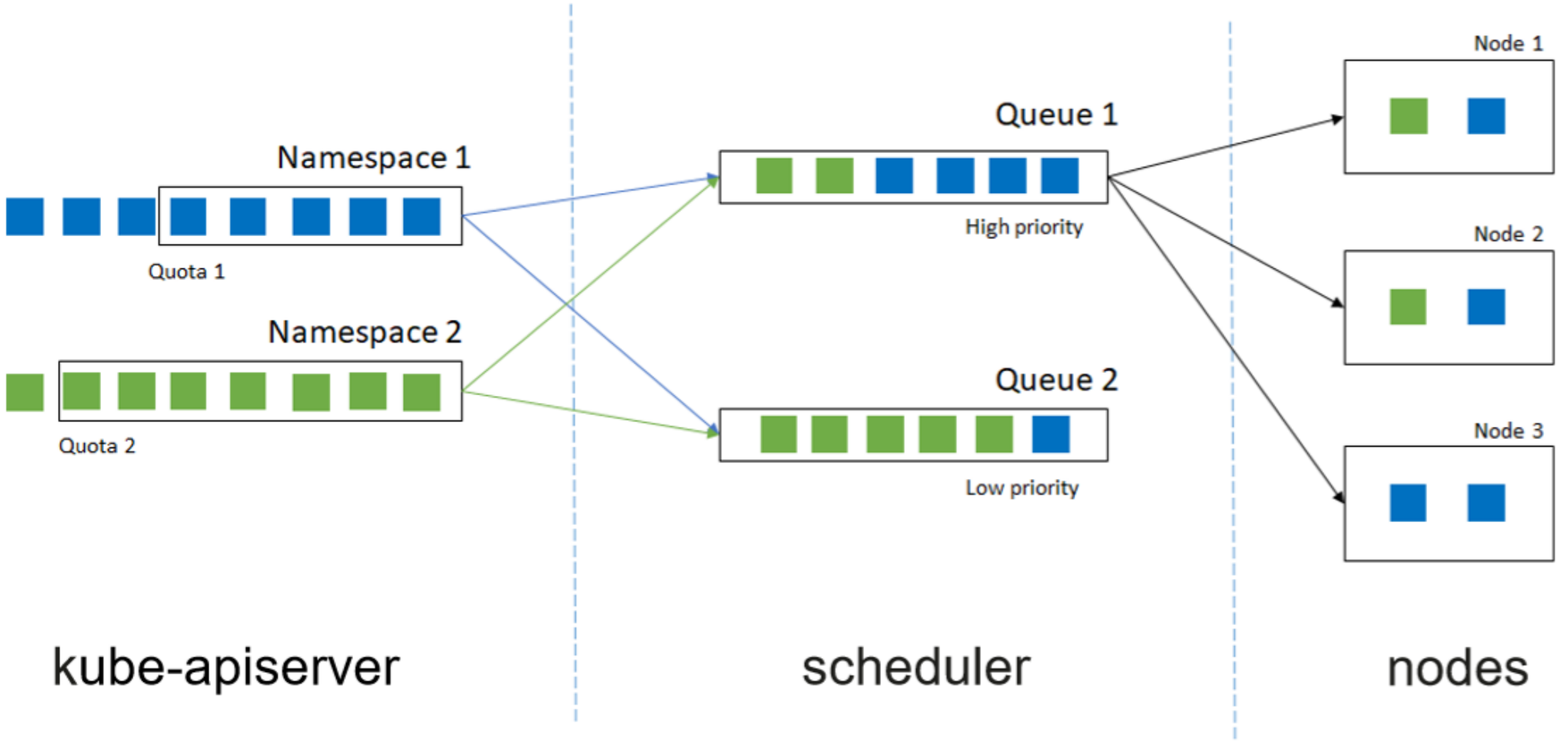
- Queue fair-share
- Job fair-share
- Namespace fair-share



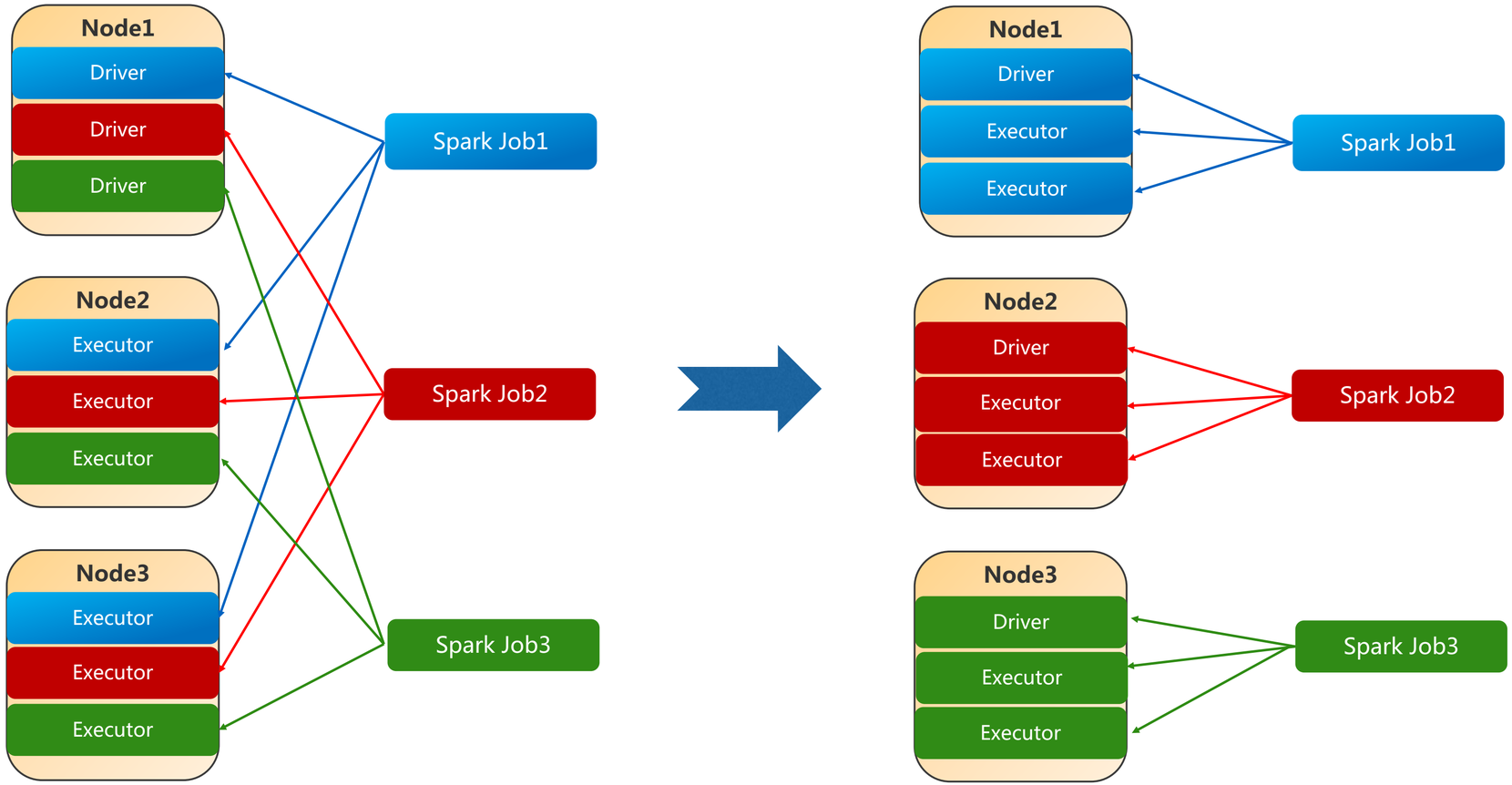
Queue job fair share via DRF

Namespace faire share

Solution: resource fair-share



Solution: Task-topology + Binpack



Summary

- How spark on Kubernetes works
- Volcano batch system
- Use delay pod creation feature to deal with high concurrent job submission
- Use queue proportion/namespace fair-share, job fair-share to share resource
- Use task-topology to improve the spark workload efficiency.

Future works for Spark

- Queue priority
- Queue reclaim
- Queue plugin
- Hierarchical queue
- Dynamic resource allocation
- External shuffle service
- Resource reservation
- Job preemption
- …

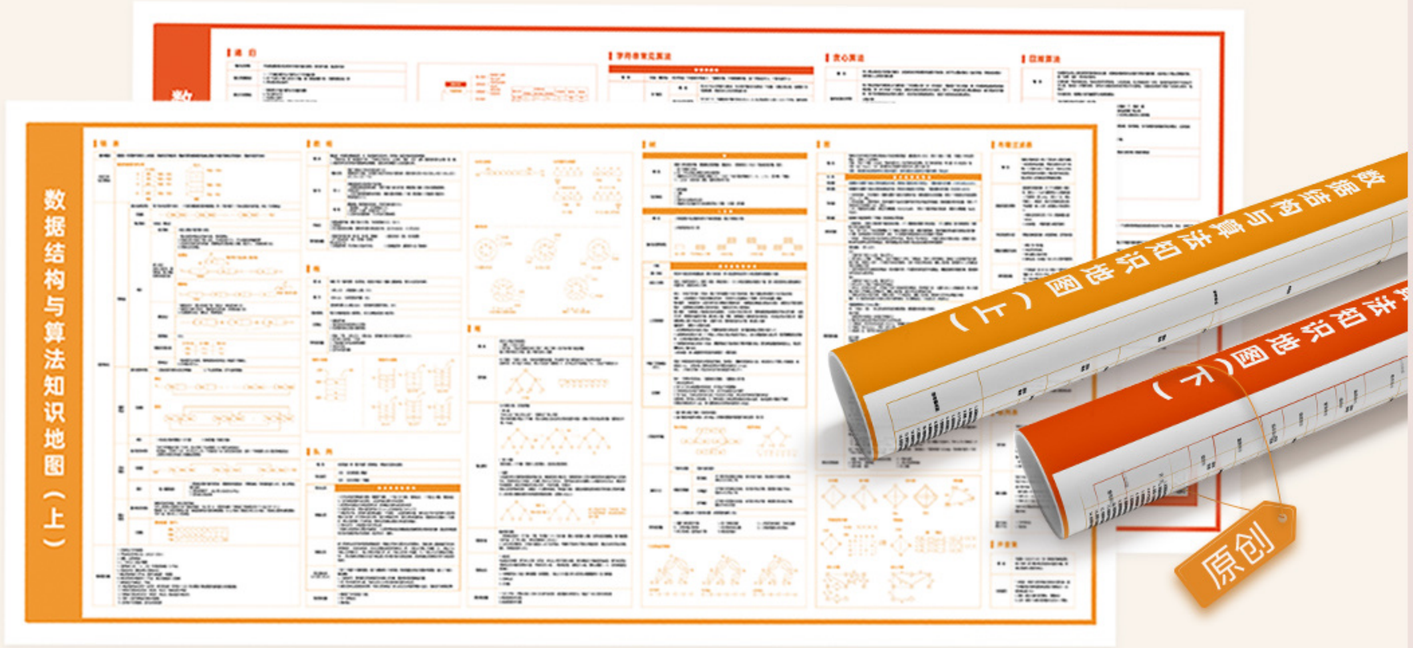
Q & A

极客时间

一起领【敲代码神器】

数据结构与算法
知识地图（上、下）

22个模块，70+面试考点，15000+字归纳



~~原价129元~~

扫码免费领取↑↑↑

仅限 200 份，先到先得

CHINA · BEIJING

THANKS

Global
Architect Summit