阿里云 Serverless 技术公开课

- Serverless 容器从入门到精通: Serverless K8s -

第 04 讲

# 低成本运行 Spark 数据计算

## —阿里云弹性计算弹性容器实例（ECI）

关注“ Serverless ”公众号
获取第一手技术资料

柳密 阿里巴巴开发工程师

1

产品介绍（ASK & ECI）

2

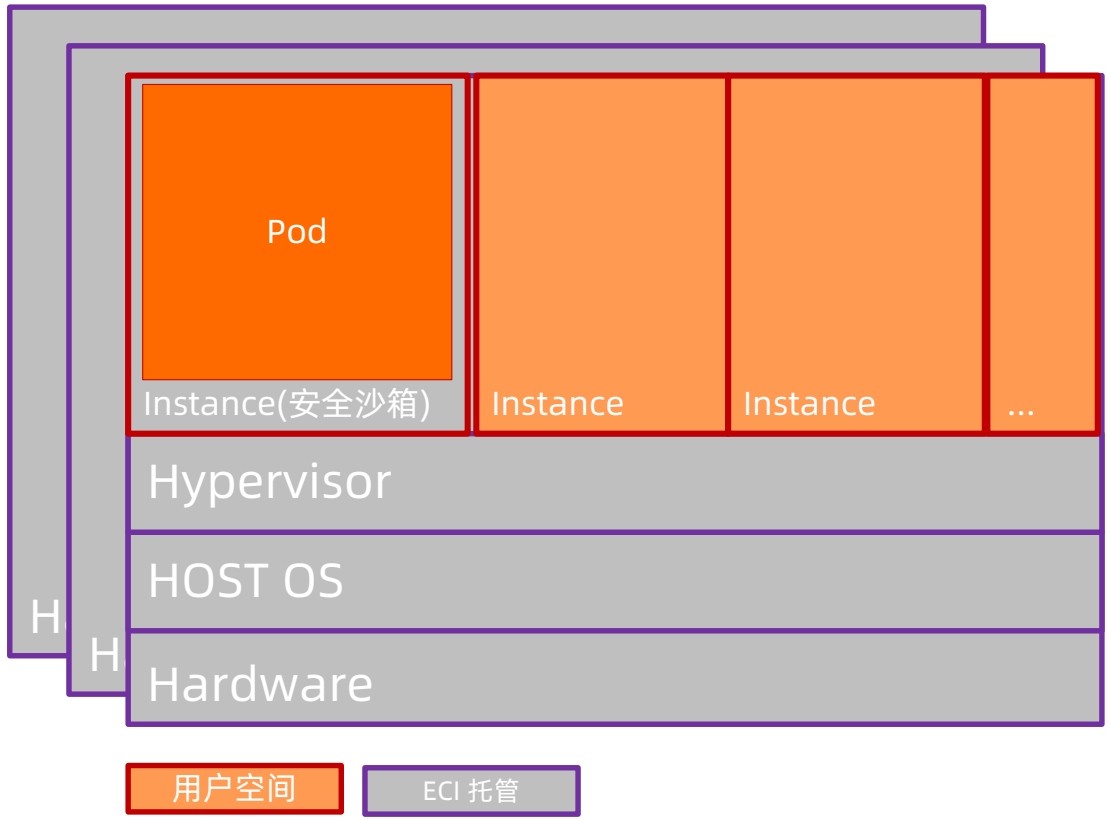
Spark on Kubernetes

3

演示

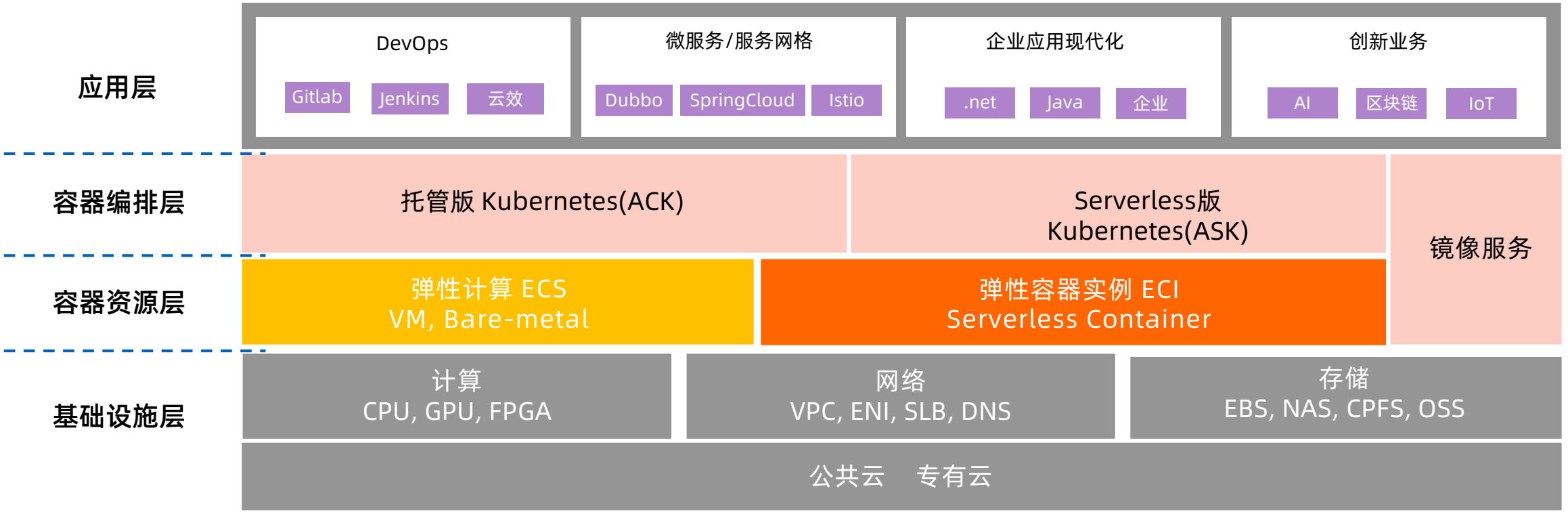
# 什么是弹性容器实例 ECI

阿里云弹性容器实例（Elastic Container Instance）提供安全的 Serverless 容器运行服务。无需管理底层服务器，只需要提供打包好的 Docker 镜像，即可运行容器，并仅为容器实际运行消耗的资源付费。



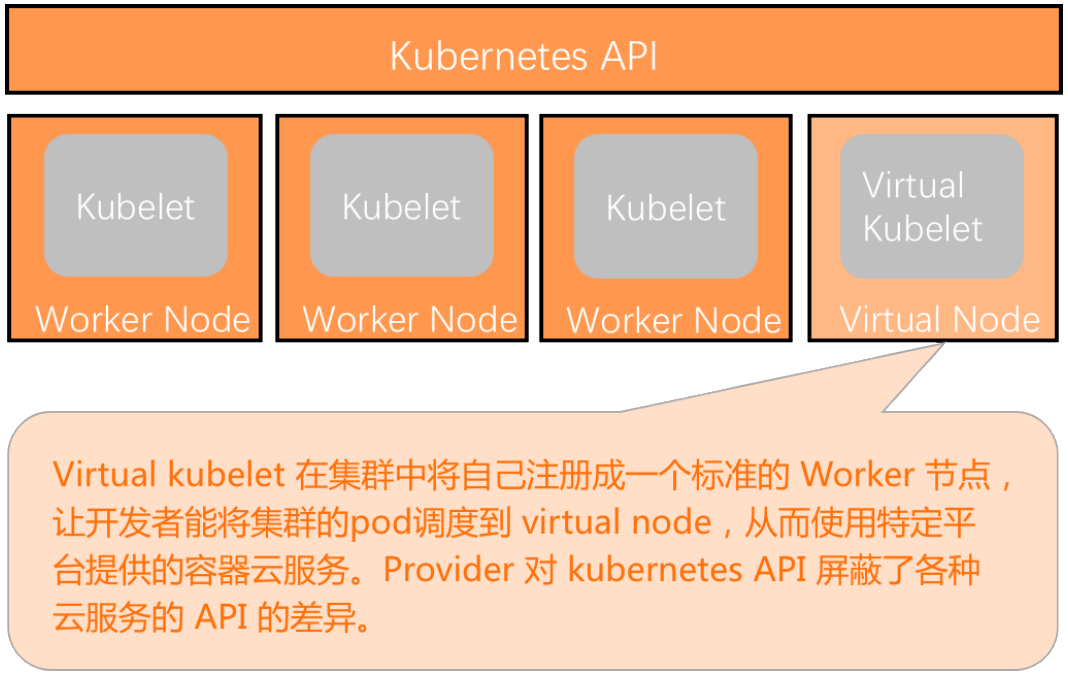

- 免运维的 IaaS 层服务，**用户不需要购买和管理 ECS，可以直接在阿里云上运行容器/Pod**；从购买 ECS，然后部署容器（ECS 模式），到直接部署容器（ECI 模式）。
- 基于 **Kata** 的**安全沙箱**容器，提供 vm 级别的安全和资源隔离，深度整合优化的轻量级虚拟化解决方案，启动更快，效率更高。
- 无缝对接容器服务 **Kubernetes**。

# 阿里云容器服务产品族


应用层
DevOps
Gitlab
Jenkins
云效
微服务/服务网格
Dubbo
SpringCloud
Istio
企业应用现代化
.net
Java
企业
创新业务
AI
区块链
IoT
容器编排层
托管版 Kubernetes(ACK)
Serverless版
Kubernetes(ASK)
镜像服务
容器资源层
弹性计算 ECS
VM, Bare-metal
弹性容器实例 ECI
Serverless Container
基础设施层
计算
CPU, GPU, FPGA
网络
VPC, ENI, SLB, DNS
存储
EBS, NAS, CPFS, OSS
公共云 专有云

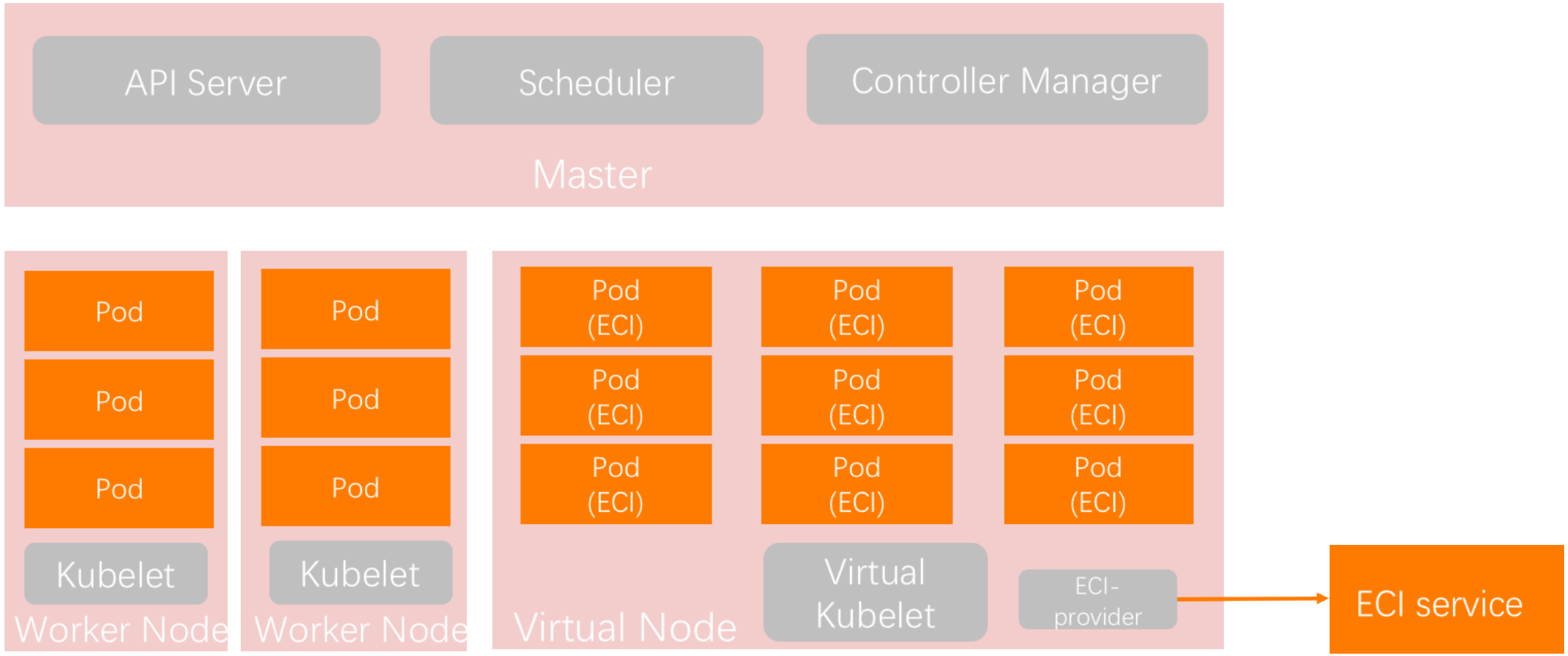
# 弹性容器实例 ECI—虚拟节点（Virtual Node）

Virtual Kubelet 是 Kubernetes kubele 的一个开源实现，它能将云提供商的容器服务作为一个 Kubernetes 集群的标准的节点，即 Virtual Node 。这给 Kubernetes 集群带来了灵活的扩展能力。它提供插件式的扩展方式，每个云厂商都可以实现自己的 provider，Alibaba Cloud ECI Provider 便是其中之一。集成了 ECI provider 的 virtual node 就成了 ECI 服务对接 Kubernetes 的桥梁。



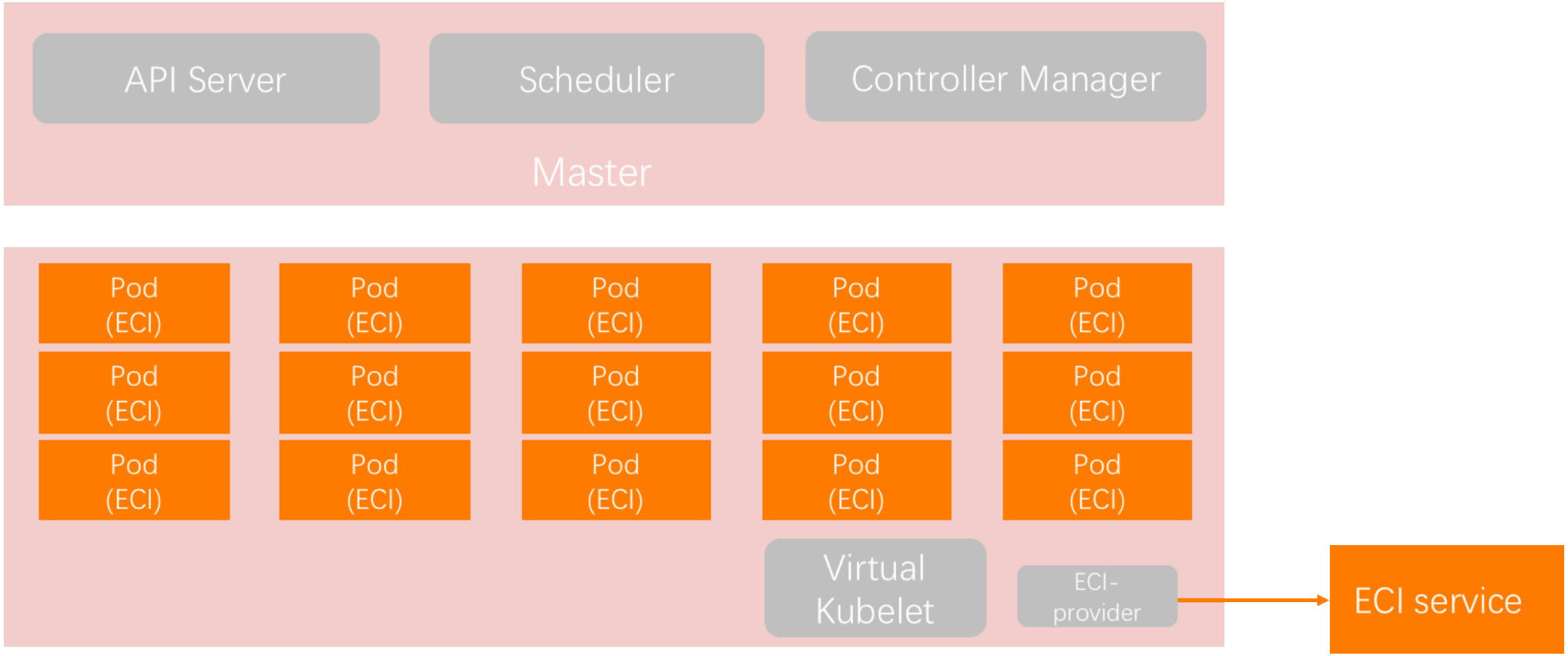
# Kubernetes + ECI

标准 Kubernetes 集群可以将 ECS 和虚拟节点混部，将 Virtual Node 作为应对突发流量的弹性资源池。



# ASK(Serverless Kubernetes) + ECI

Serverless 集群中没有任何 ECS worker 节点，也无需预留、规划资源，只有一个 **Virtual Node**

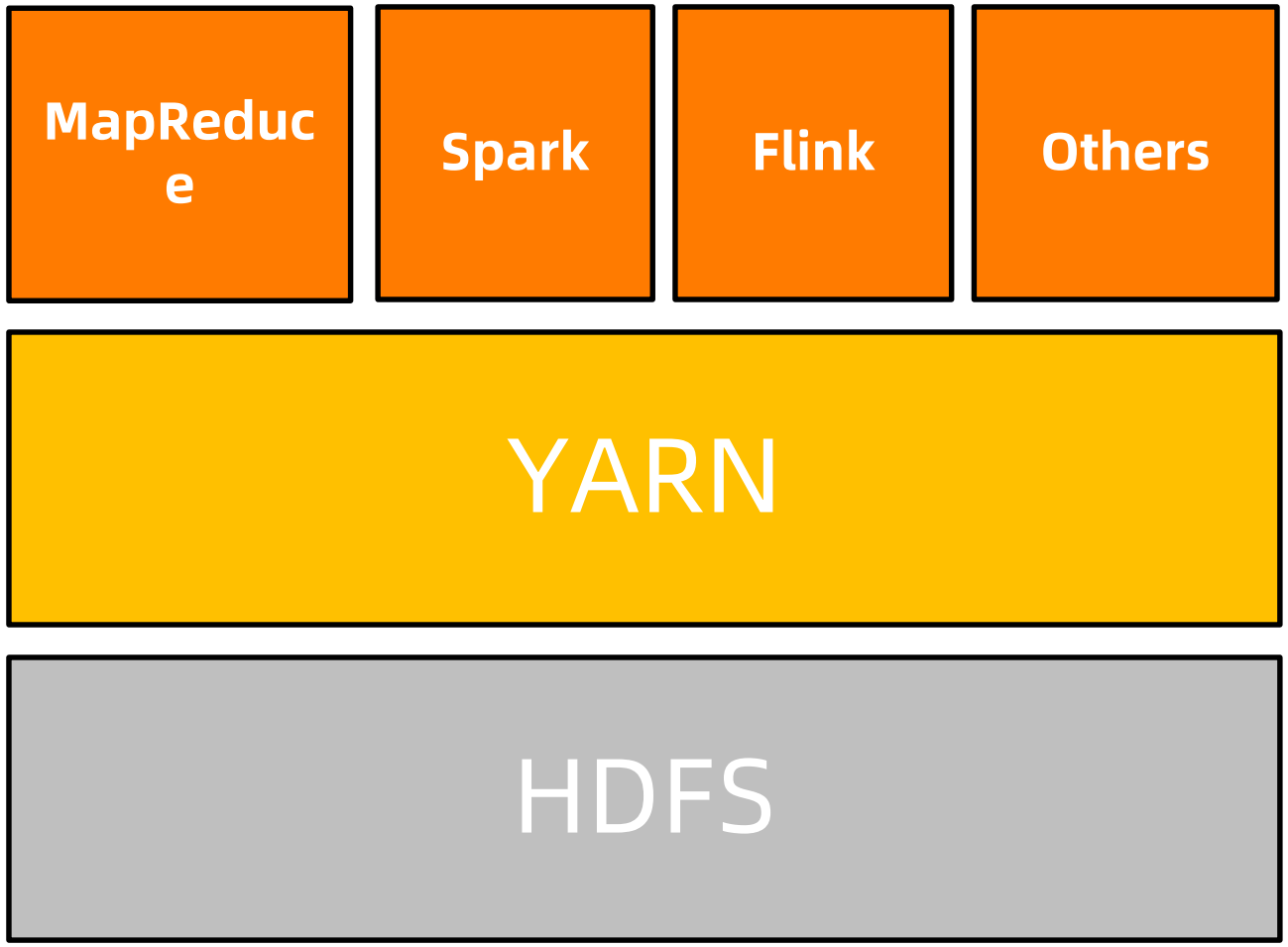


# YARN to Kubernetes

Spark 自 2.3.0 开始试验性支持 Standalone、on YARN 以及 on Mesos 之外的新的部署方式：Running Spark on Kubernetes，而且一直在不断持续完善。

Spark on kubernetes 相比于 on Yarn 等传统部署方式的优势：

- 统一的资源管理与调度。
- 计算与存储分离。
- 弹性的集群基础设施。
- 轻松实现复杂的分布式应用的资源隔离和限制。
- 容器化的优势。
- 大数据上云。

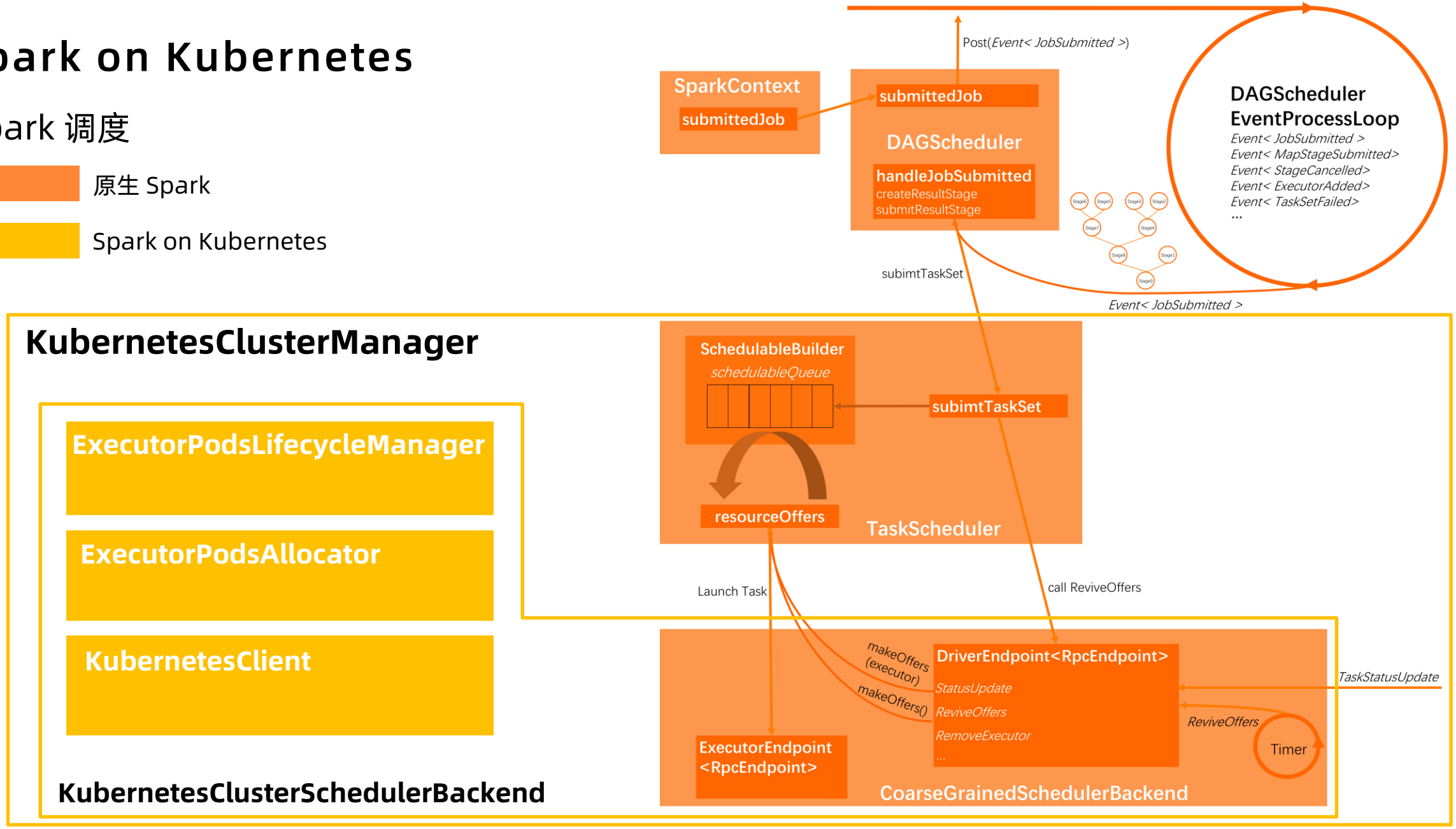


# Spark on Kubernetes

## Spark 调度

原生 Spark

Spark on Kubernetes



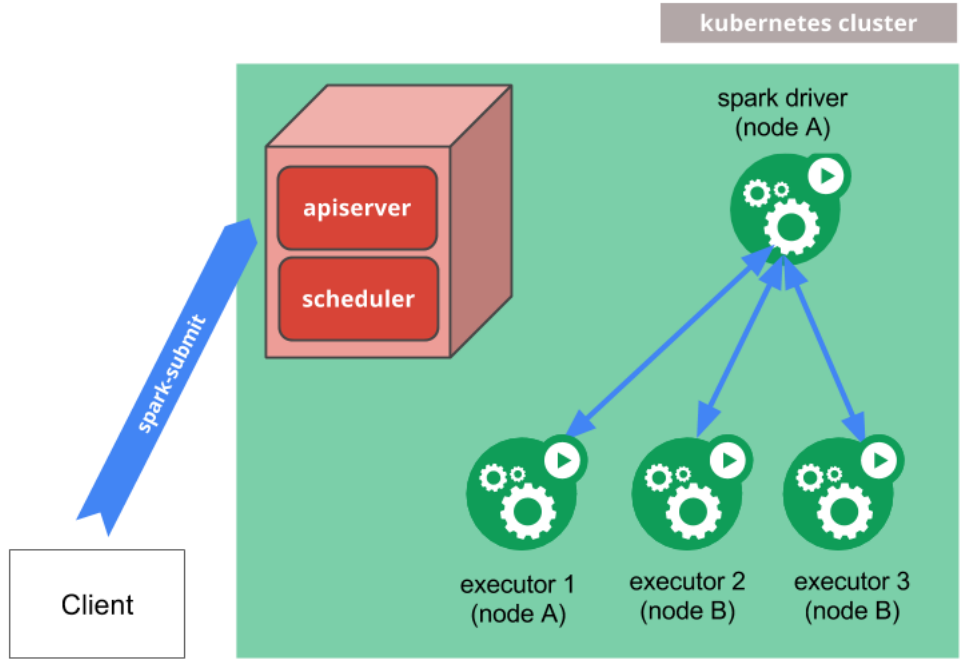
# Spark on Kubernetes

## Spark submit

在 Spark Operator 出现之前，在 Kubernetes 集群提交 spark 作业只能通过 spark submit 的方式。

创建好 Kubernetes 集群，在本地即可提交作业。

```
liumihustdeMacBook-Pro:spark-on-k8s liumihust$ ./spark-2.3.0-bin-hadoop2.6/bin/spark-submit
--master k8s://121.199.47.95:6443
--deploy-mode cluster
--name WordCount
--class com.aliyun.liumi.spark.example.WordCount
--conf spark.kubernetes.authenticate.driver.serviceAccountName=spark
--conf spark.executor.instances=2
--conf spark.kubernetes.container.image=registry.cn-beijing.aliyuncs.com/liumi/spark:2.3.0-hdfs-1.0
local:///opt/spark/jars/SparkExampleJava-1.0-SNAPSHOT.jar
```




| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| wordcount-f191cf97ed7b3645affc7ee44db2acbd-driver<br>spark:2.3.0-hdfs-7 | 运行中 | 0 | 172.22.0.36 | cn-hangzhou.192.168.3.142<br>192.168.3.142 | 2019-11-10 13:04:29 | | 0.008 | 511.176 Mi | 详情 \| 更多 |
| wordcount-f191cf97ed7b3645affc7ee44db2acbd-exec-1<br>spark:2.3.0-hdfs-7 | 运行中 | 0 | 172.22.0.170 | cn-hangzhou.192.168.3.141<br>192.168.3.141 | 2019-11-10 13:04:34 | | 0.907 | 721.035 Mi | 详情 \| 更多 |
| wordcount-f191cf97ed7b3645affc7ee44db2acbd-exec-2<br>spark:2.3.0-hdfs-7 | 运行中 | 0 | 172.22.0.169 | cn-hangzhou.192.168.3.141<br>192.168.3.141 | 2019-11-10 13:04:34 | | 0.91 | 728.285 Mi | 详情 \| 更多 |

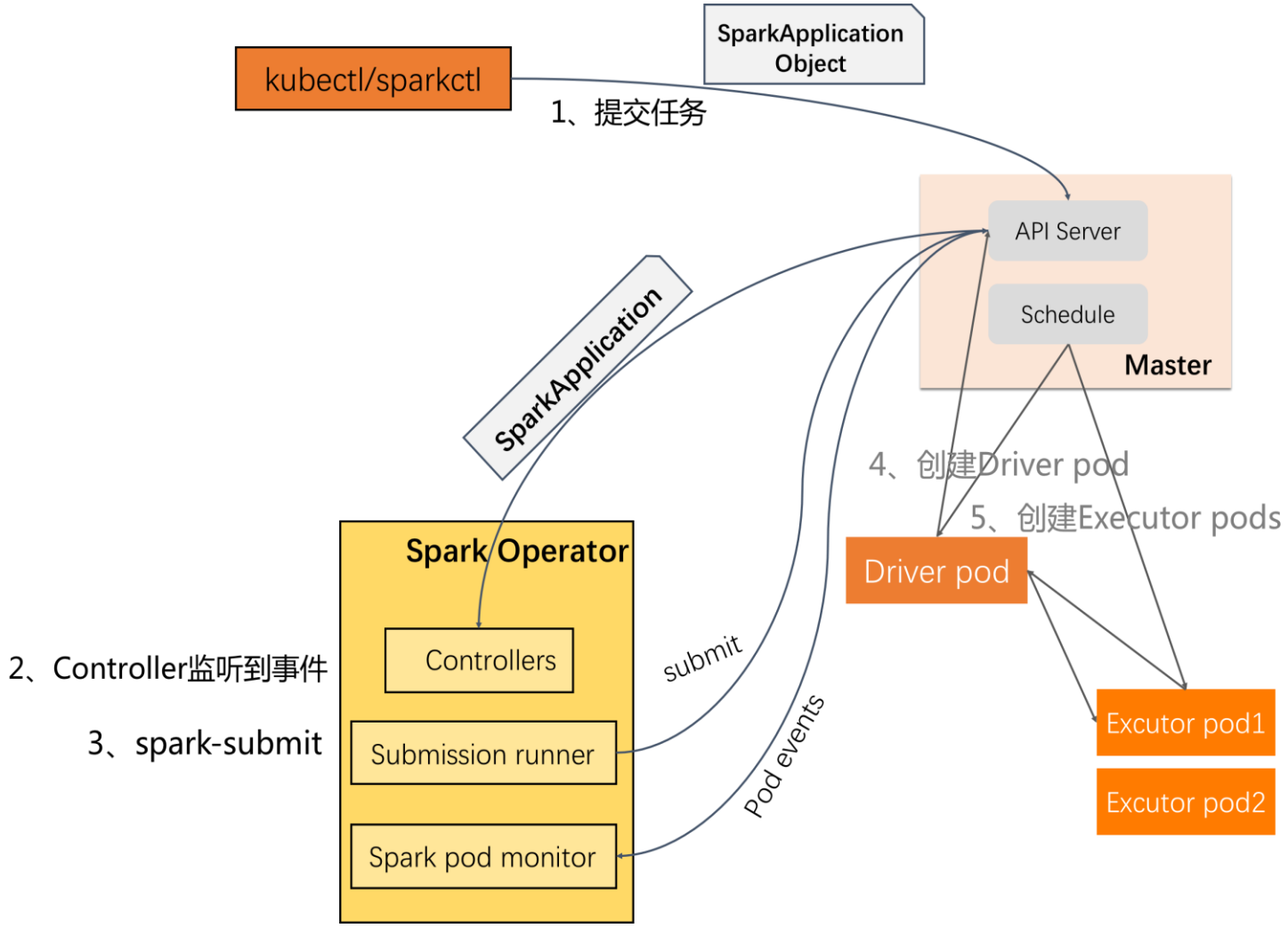
# Spark on Kubernetes

## Spark Operator

Spark Operator 就是为了解决在 Kubernetes 集群部署并维护 Spark 应用而开发的，Spark Operator 是经典的 CRD + Controller， 即 Kubernetes Operator 的实现。

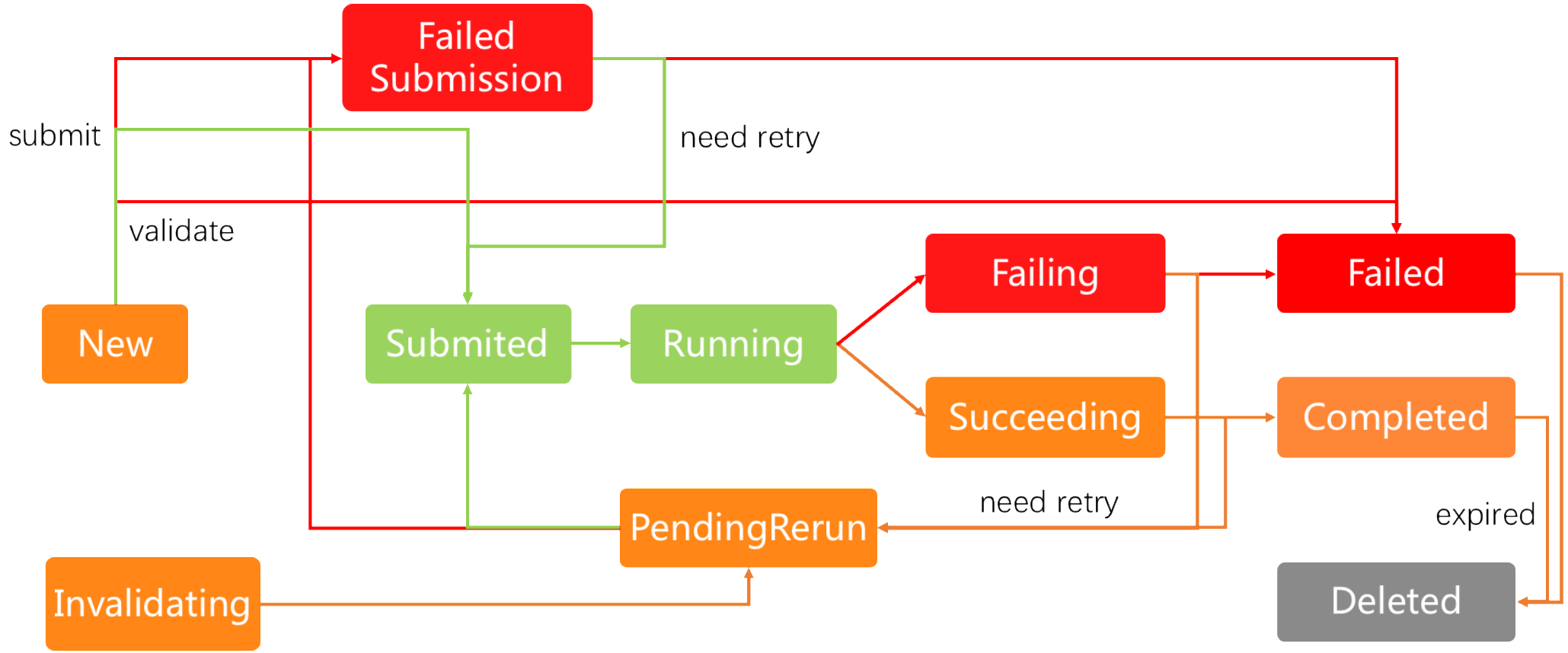
Spark Operator 几个主要的概念：

- SparkApplication：标准的 K8s CRD。Controller 负责监听 CRD 的创建、更新、以及删除等事件，并作出对应的 Action。
- Submission runner：对 Controller 发起的创建请求提交 spark-submit。
- Spark pod monitor：监听 Spark pods 的状态和事件更新并告知 Controller。



# Spark on Kubernetes

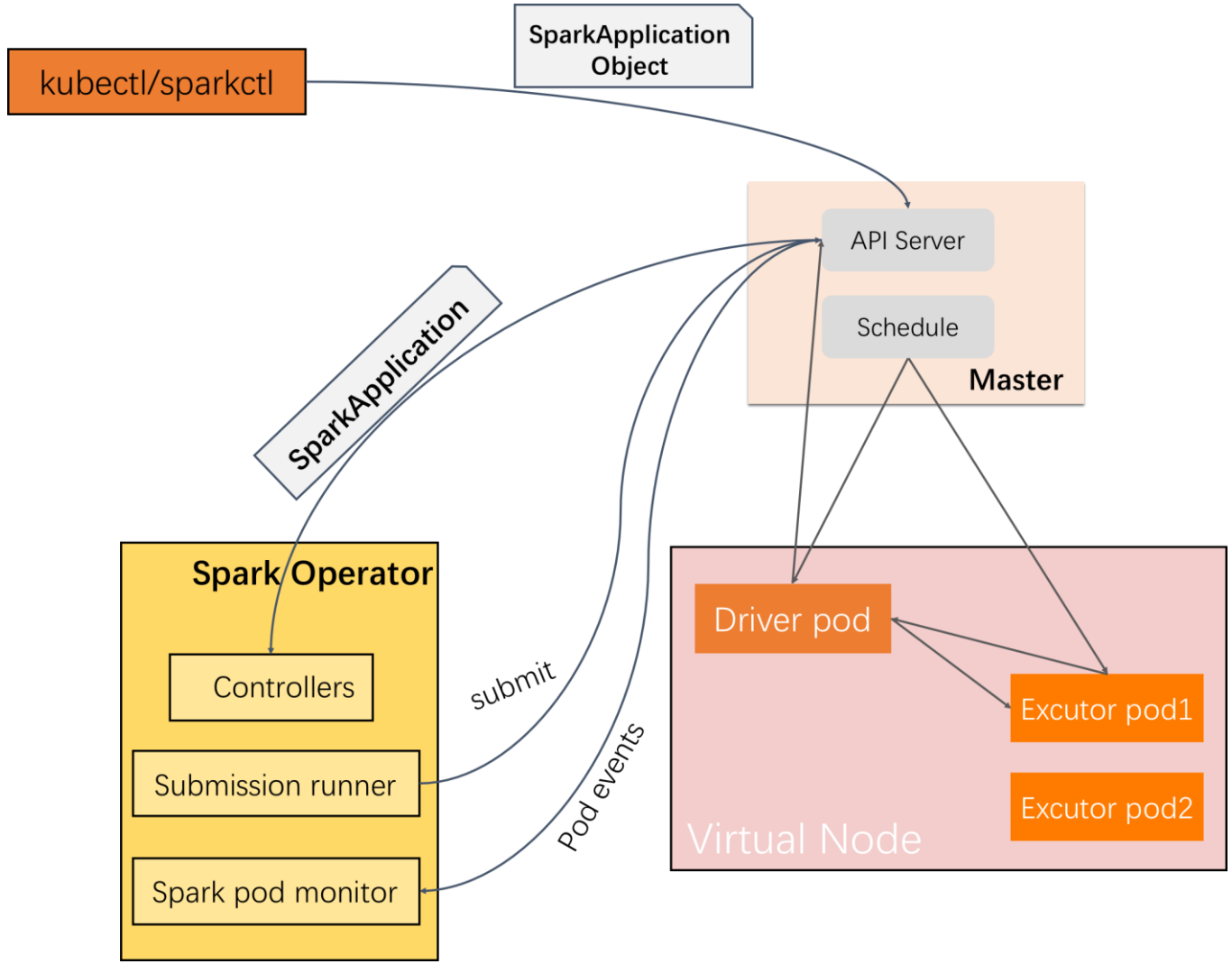
## SparkApplication 状态机



# Spark on Kubernetes

## Serverless kubernetes + ECI

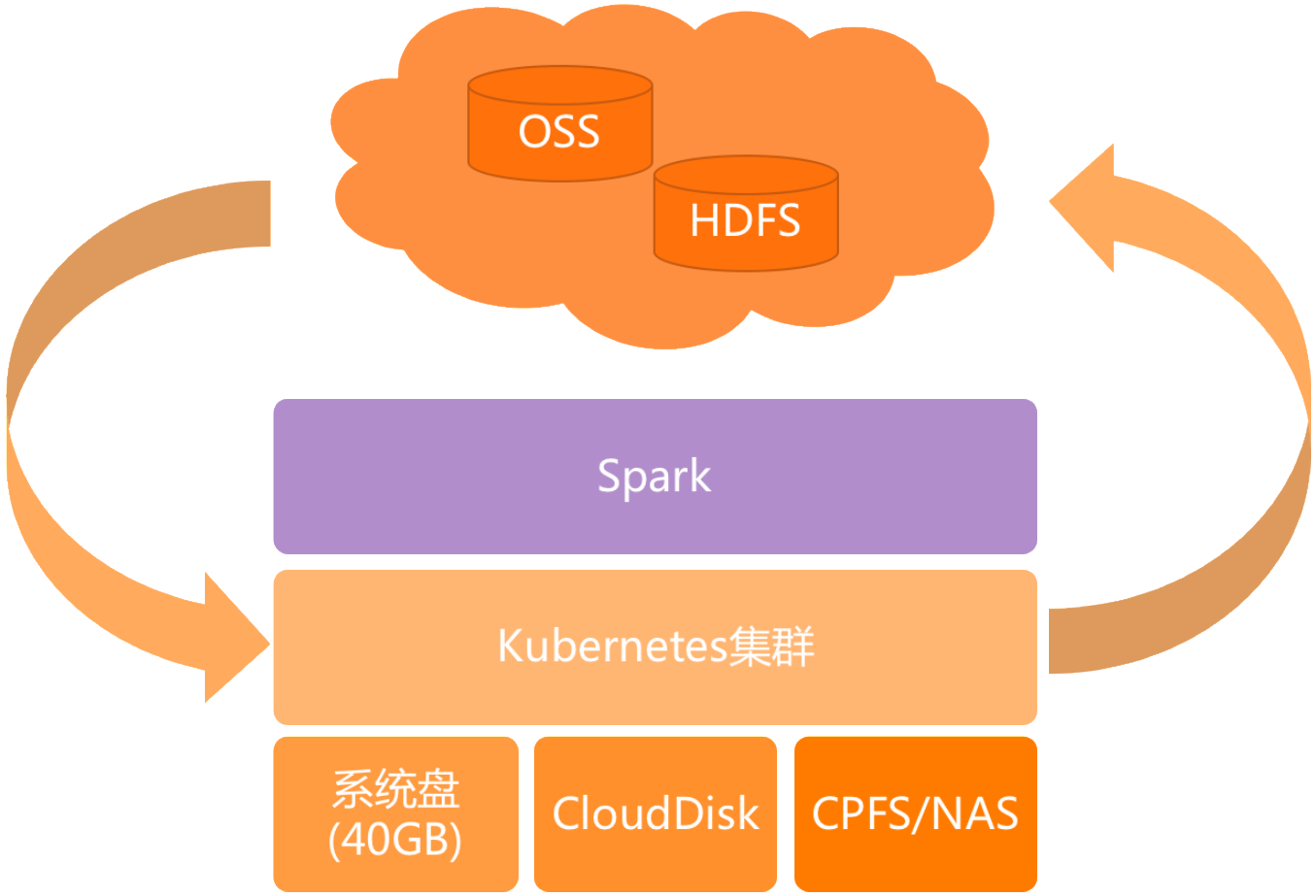
- 完全兼容 Spark/Kubernetes 生态
- 提交作业前无需提前预留任何资源
- 无需关心集群的扩缩容
- 所有资源随作业提交自动开始申请
- 作业执行结束后自动释放，支持自定义 Job 元数据清理时间
- 提供快速启动(10~20s)和大规模并发能力（单AZ，500个Pod/30s）



# Spark on Kubernetes

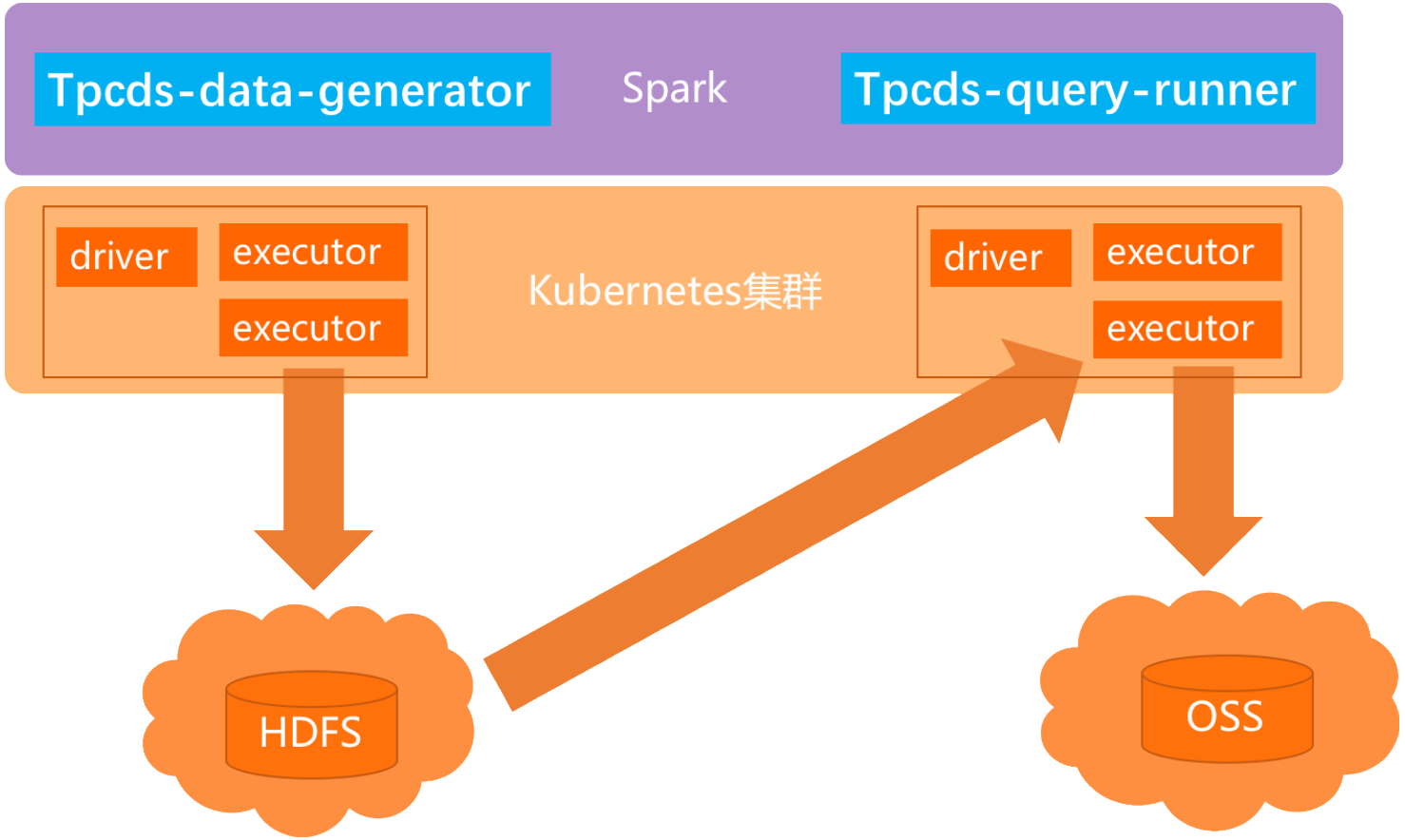
## 计算与存储分离

- 数据源
- Shuffle 数据



# 演示

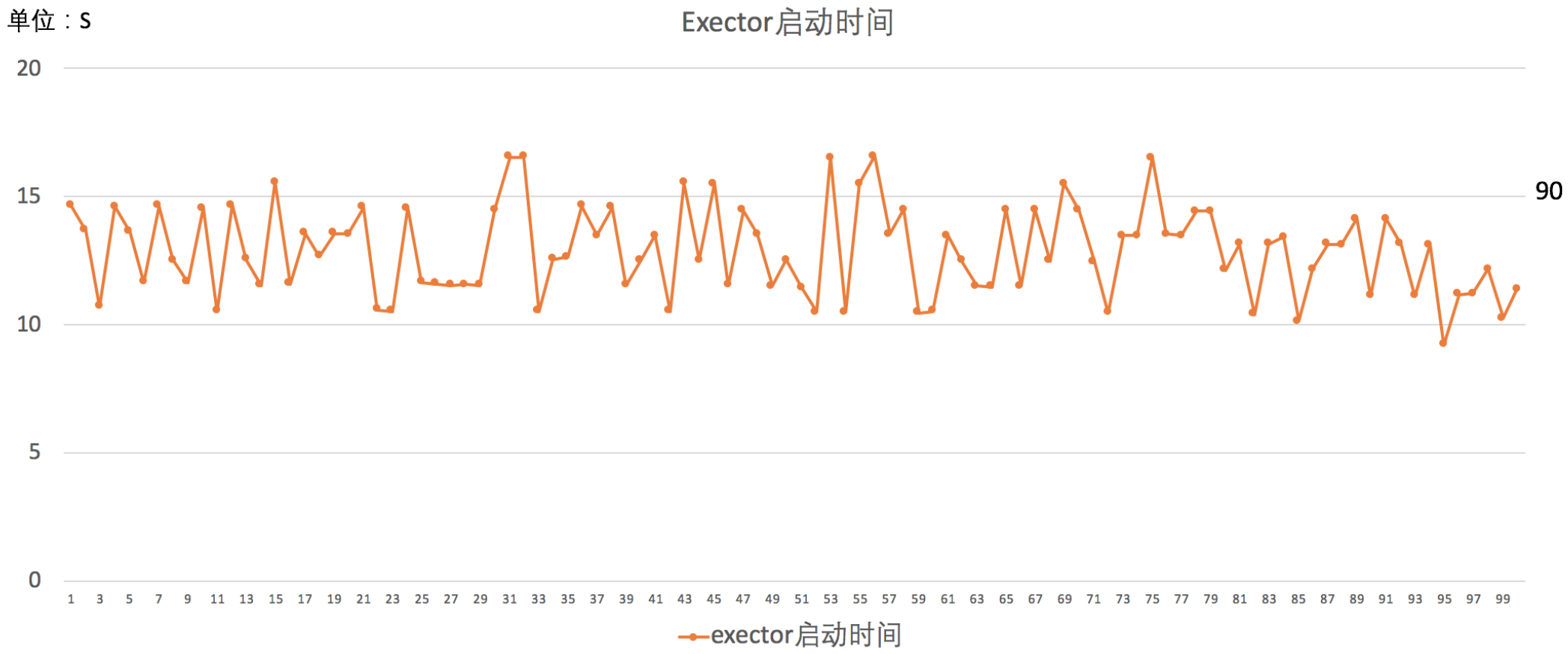
TPC-DS


Tpcds-data-generator
Spark
Tpcds-query-runner
driver
executor
executor
Kubernetes集群
driver
executor
executor
HDFS
OSS

# 演示

## WordCount

100 个 1C2G 的 Excutor 并发启动，应用的镜像大小约为 500 MB。
我们可以看到，100 个 excutor pod 全部在 job 提交后的 17s 内完成启动，其中 90% 的 excutor pod 能在 15s 内启动。



# 演示

## WordCount


100 个 1C2G 规格 的 ECI 处理 30G 的数据耗时为 87S（实际的计费时长约为 87-15S），我们可以计算出本次 Spark 作业的计算总费用为：

*¥0.00004288/s x (87-15)s x 100 = ¥0.308736*

而 ECI 最近新上线的抢占式实例，即 ECI Spot，非常适合于数据处理场景，更是将计算成本降低至目前的 **10%** 不到：

*¥0.00004288/s x (87-15)s x 100 x 10% = ¥ 0.0308736*

# 谢谢观看

## THANK YOU

关注“ Serverless ”公众号

获取第一手技术资料