知乎基于 Kubernetes 的 Kafka平台探索和实践

知乎 白瑜庆

自我介绍

知乎技术中台工程师

负责知乎存储相关平台

纲 要

Kafka 在知乎

为什么做基于 Kubernetes 的 Kafka 平台

如何实践基于 Kubernetes 的 Kafka 平台

Kafka 在知乎的应用

平台承载知乎业务日志、数据传输和消息队列服务

平台承载 Kafka 集群超过 40 个， 1000+ Topic，2000+ broker

知乎技术平台重要的组件

平台线上稳定运行 2 年

平台概览

- 多集群
- 高可用

为什么采用 Kubernetes

问题驱动

- Kafka 资源规划不合理
    - 单一集群造成系统单点
    - 不区分集群和 Topic 等级，影响重要业务
- 业务与 Kafka 深度耦合

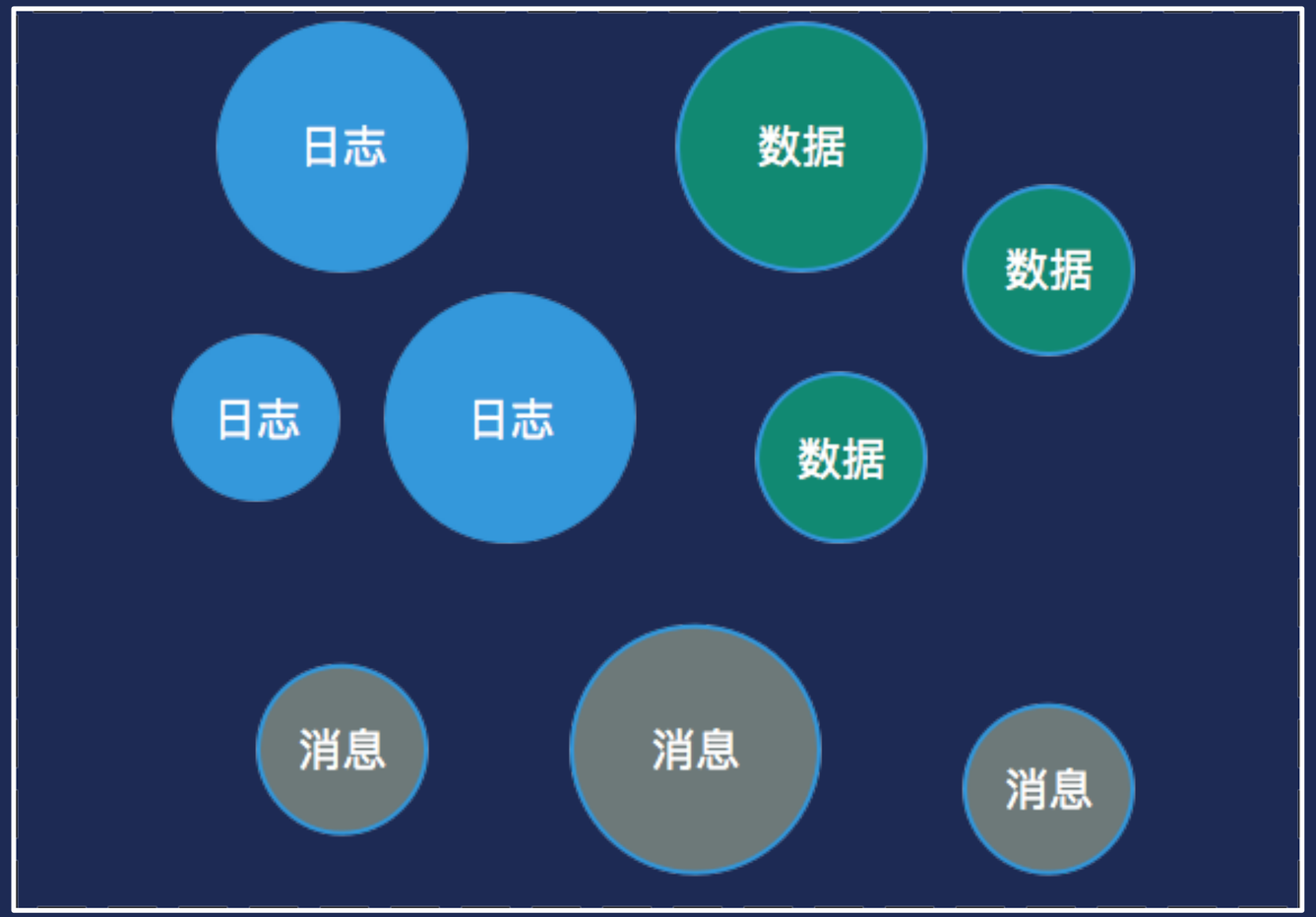
业务资源规划

多 Kafka 集群方式

根据 Topic 类型划分集群

同一类型 Topic 的集群细分

- Topic 服务等级、容量和规模划分



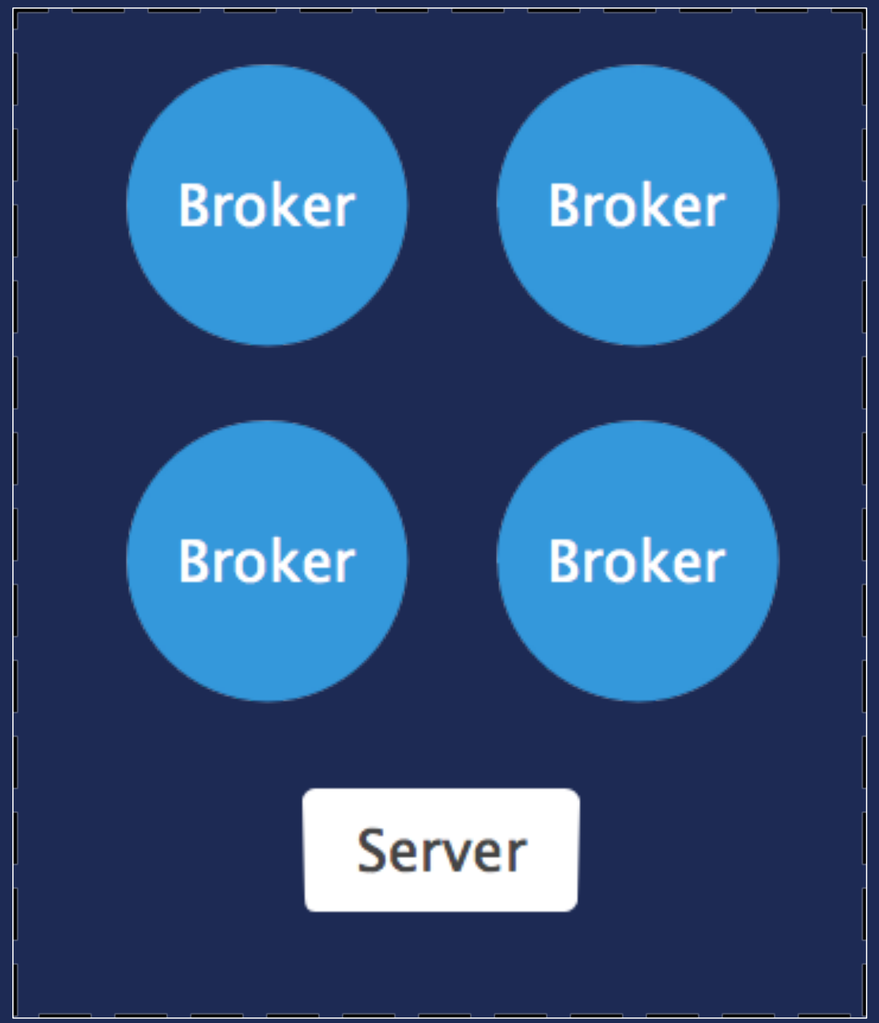
多集群问题

多变需求引发集群规模增长

- Broker, Topic 规模

服务器资源利用率

- 单机运行多 Broker 方式



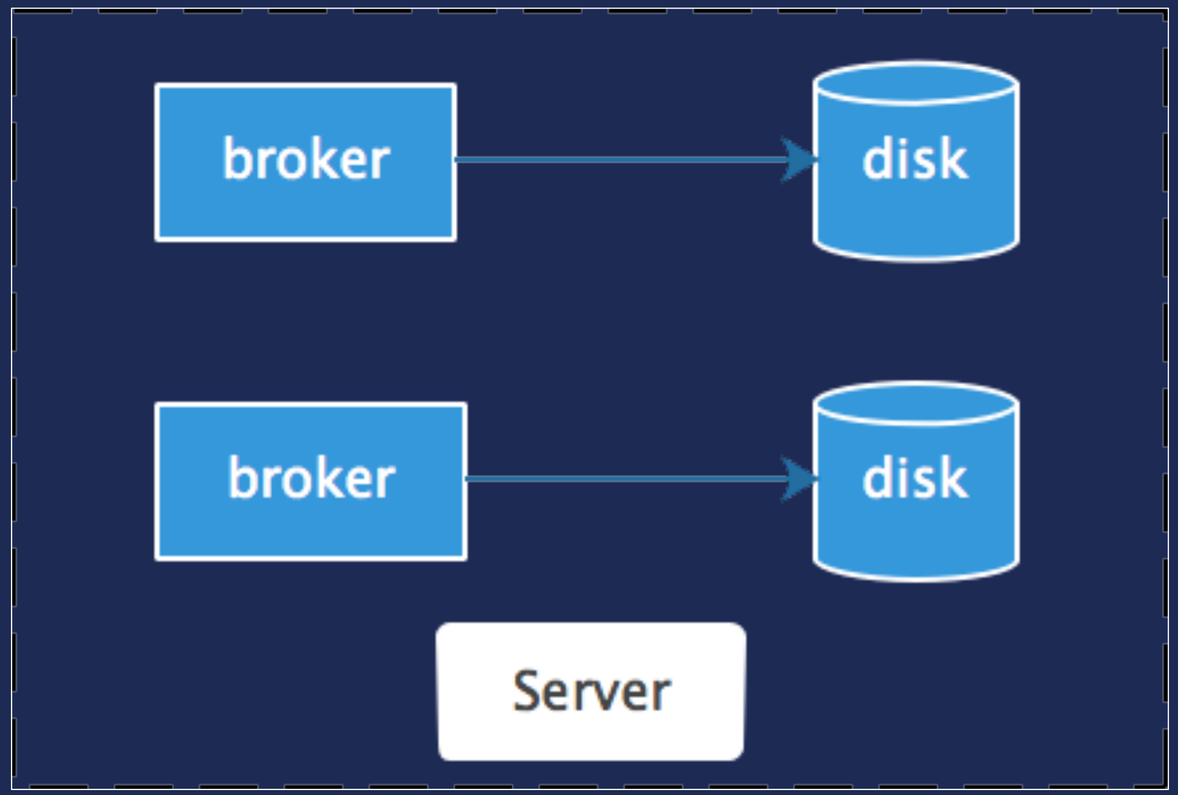
磁盘问题

磁盘因素

- 数据持久化
- 资源隔离

结论

- Broker 之间物理磁盘隔离



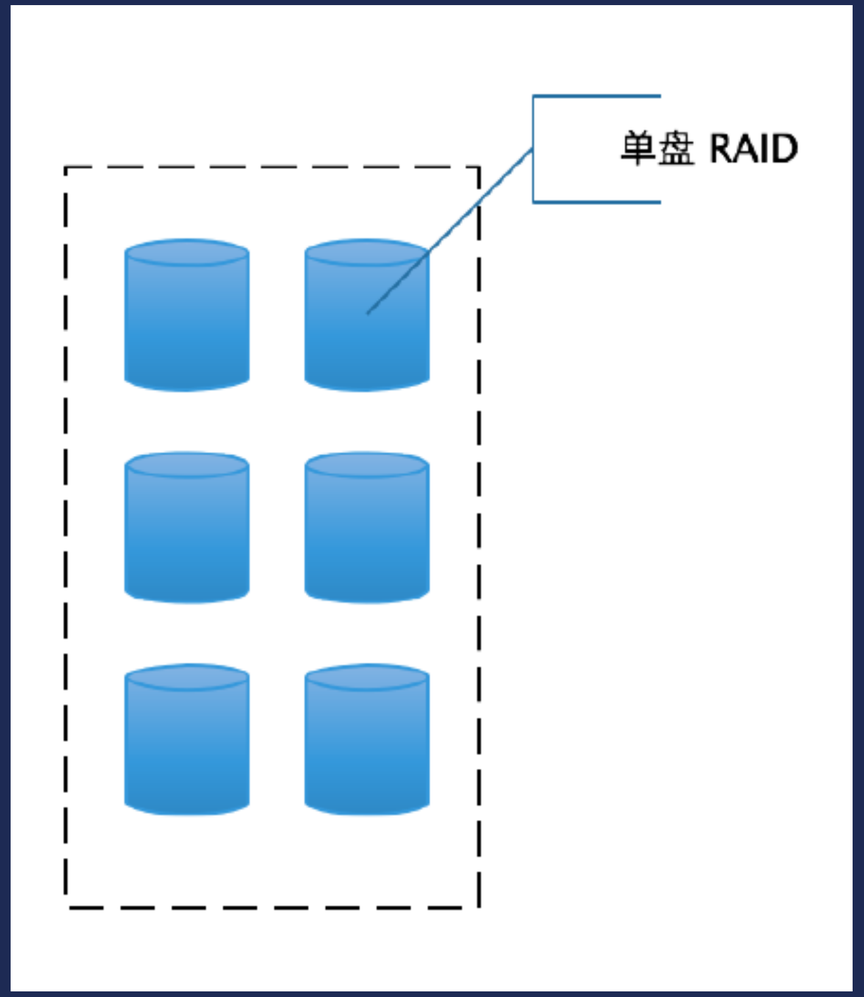
服务器选型

高密度存储服务器

- 多磁盘, 单盘 RAID 或无 RAID
- 服务器使用率高

黑石的 Kafka 高性能服务器

- 高性能磁盘 x 12
- 内存和 CPU



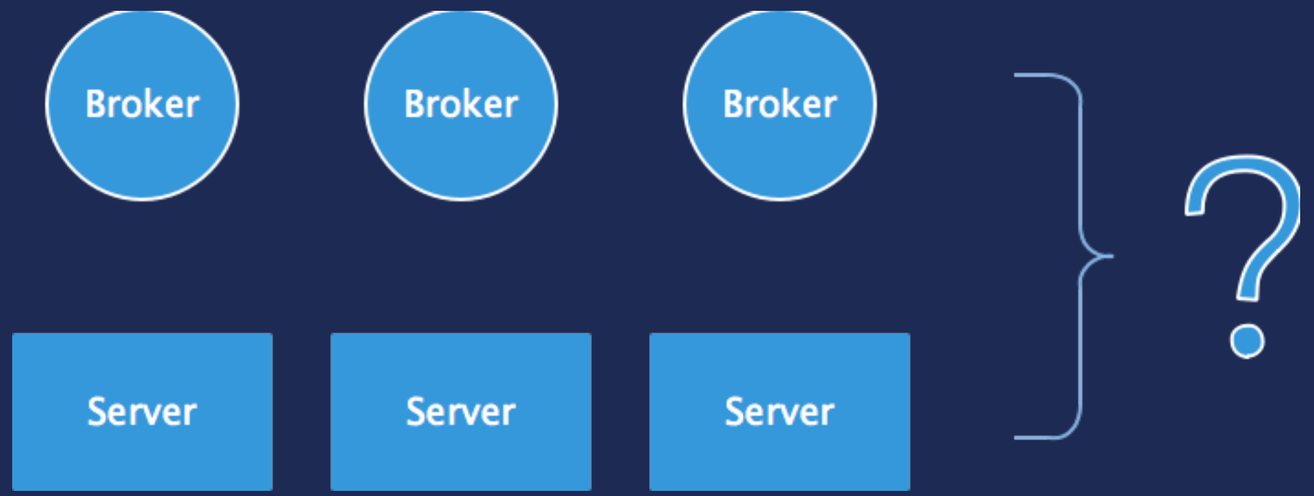
挑战

集群数增加，导致 Broker 扩张

- 如何调度它们
- 如何管理它们

服务器如何管理

调度关键因素是磁盘



Kubernetes

集群资源管理和调度

容器技术提供资源隔离

应用程序管理

Kafka on Kubernetes

设计 Kafka 容器

- 内存、CPU、网络和存储

调度 Kafka 容器

设计 Kafka 容器

内存 和 CPU

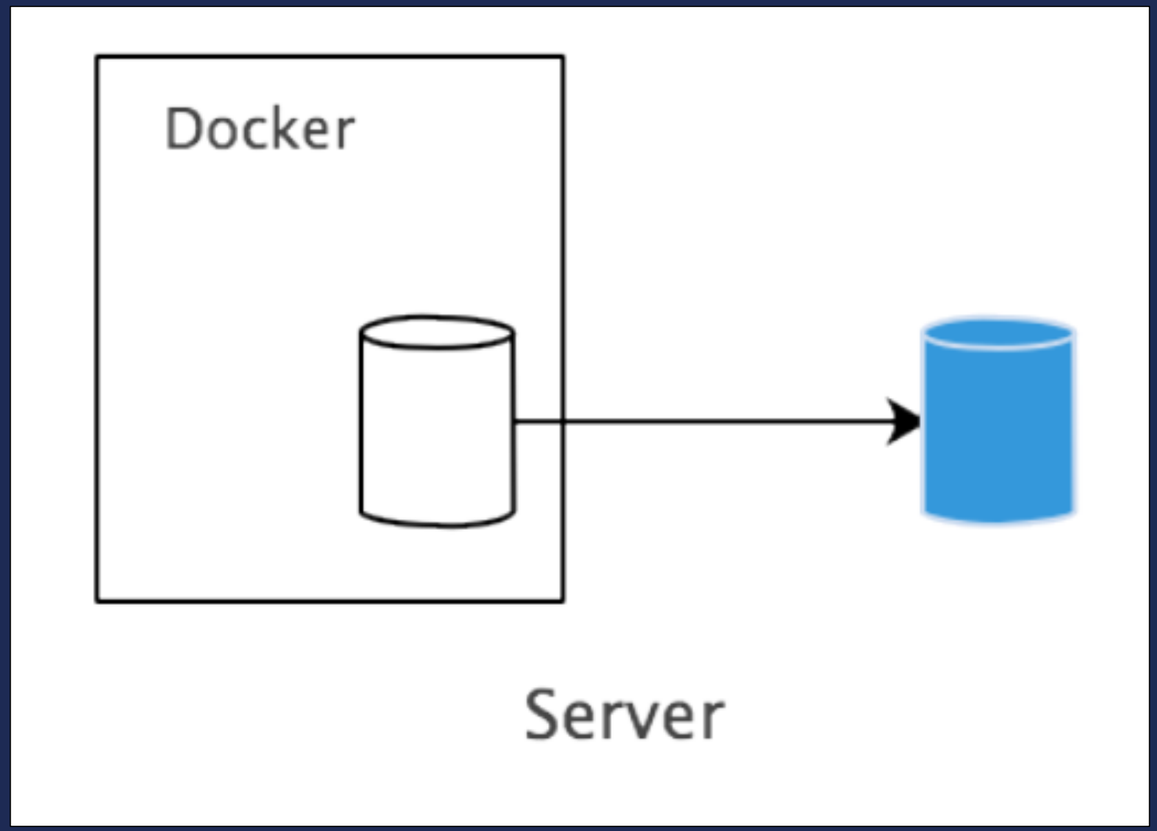
- 依照集群类型测试基准数据
- 消息、数据和日志类型 topic 不同特点
- 集群 broker 最多不超过 200 topic
- 依据服务器性能的基准数据

网络

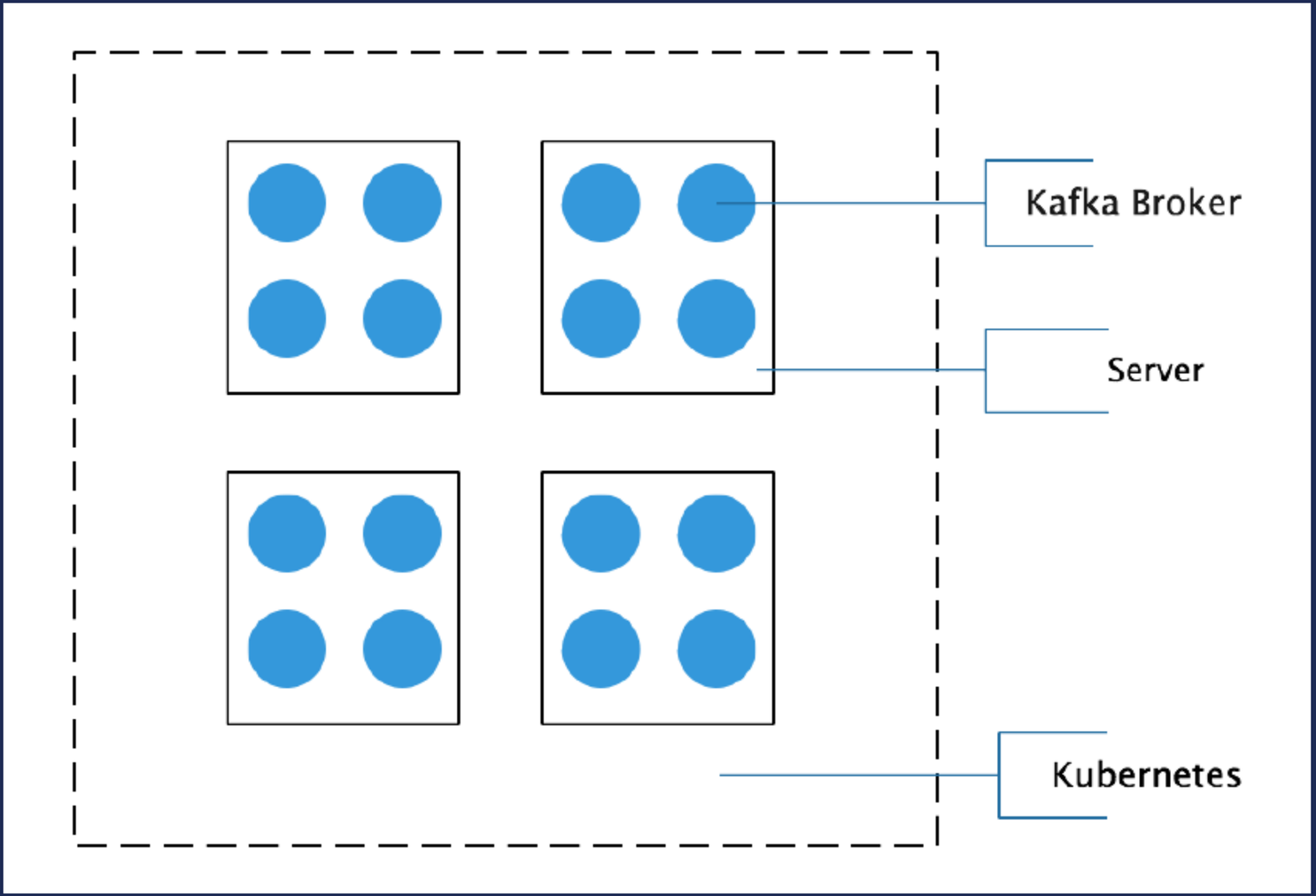
- 容器采用独立的内网 IP 方案
- 容器注册内网 DNS
- 黑石网络提供网络支持

容器挂载服务本地目录

- 高性能、日志持久化
- hostPath Volume



集群概览



如何调度 Kafka 容器器

磁盘是容器的调度单元

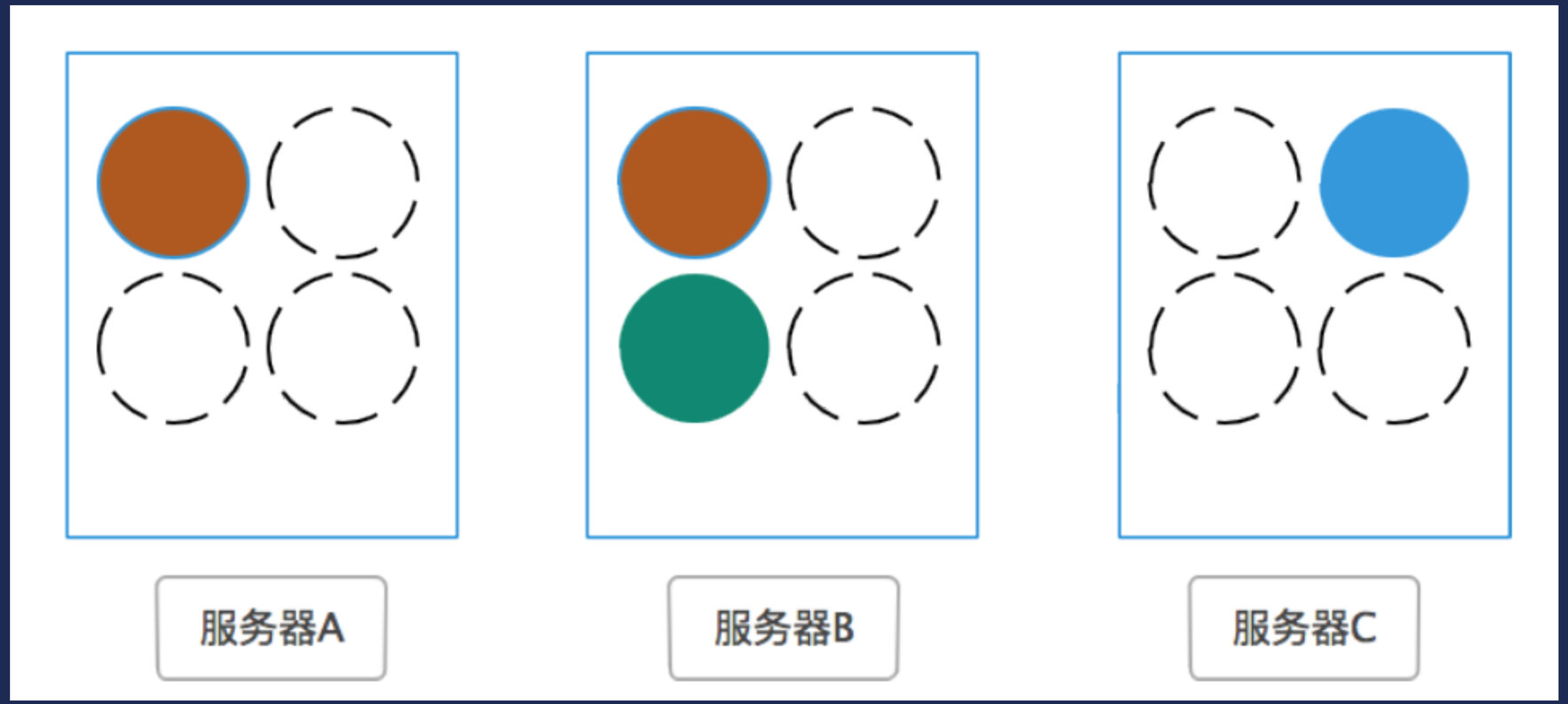
目标

- 单集群的 Broker 在节点分散
- 节点存储使用均匀

磁盘调度方法

根据服务器磁盘状态计算分数，分数高者被调度

- 集群 Broker 在节点部署情况
- 服务器可用磁盘情况



如果创建红色集群则服务器器 C 最优

如果创建蓝色集群则服务器器 A 最优

挑战

Kubernetes hostPath 的问题

- PersistentVolume 局限性
- 管理主机目录，增加、删除和所有者

期望

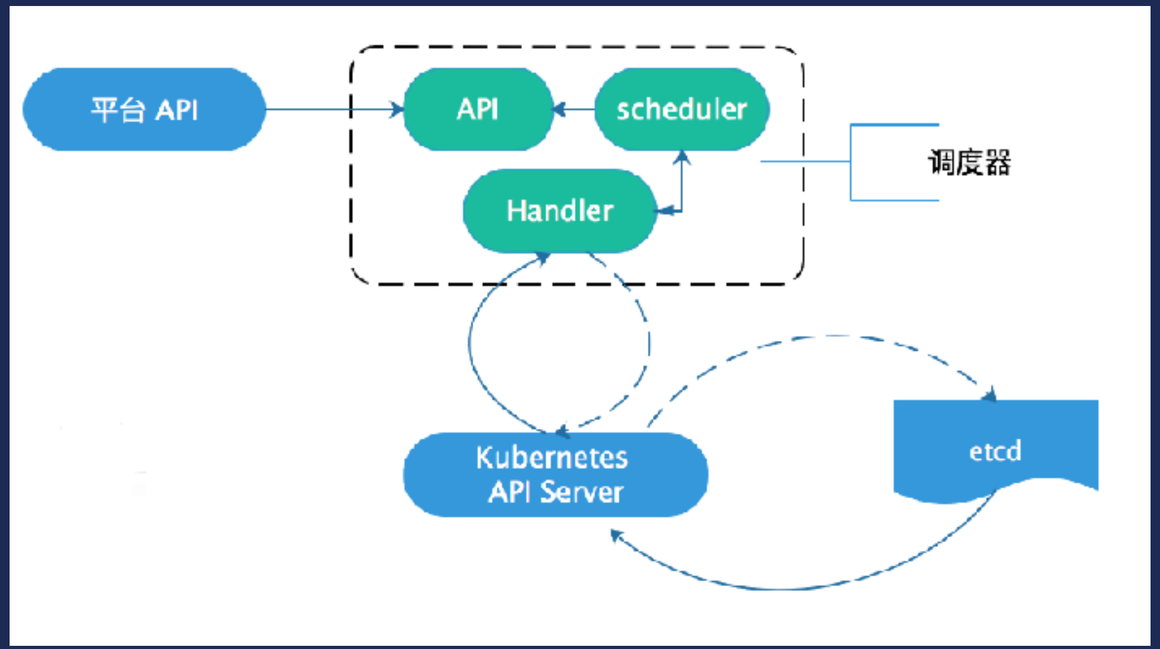
- 容器需要利用本地存储
- 满足调度方法（选出合理的节点）

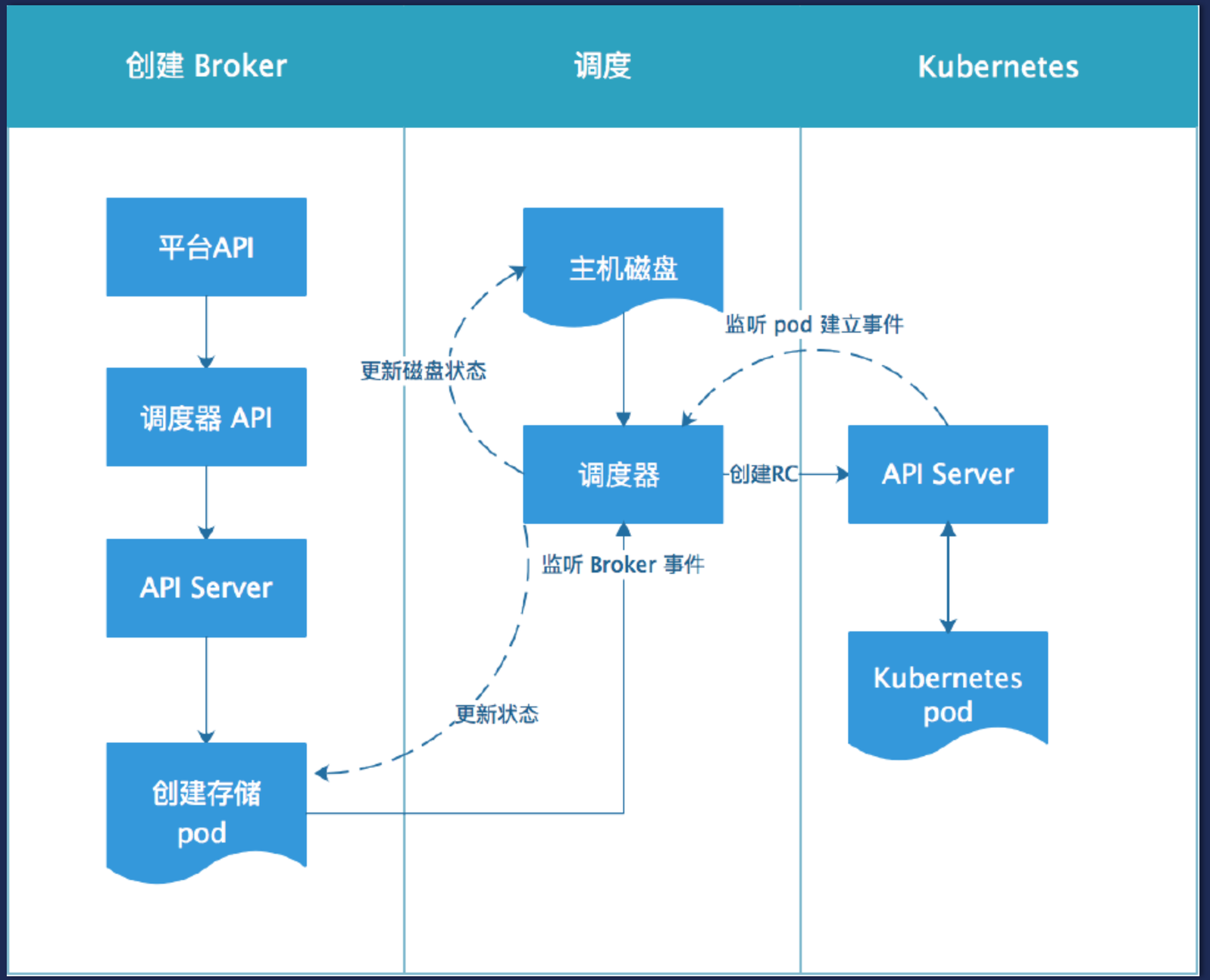
磁盘调度器

按照调度算法选择节点

创建 ReplicationController

监听 Kubernetes 状态更新磁盘信息

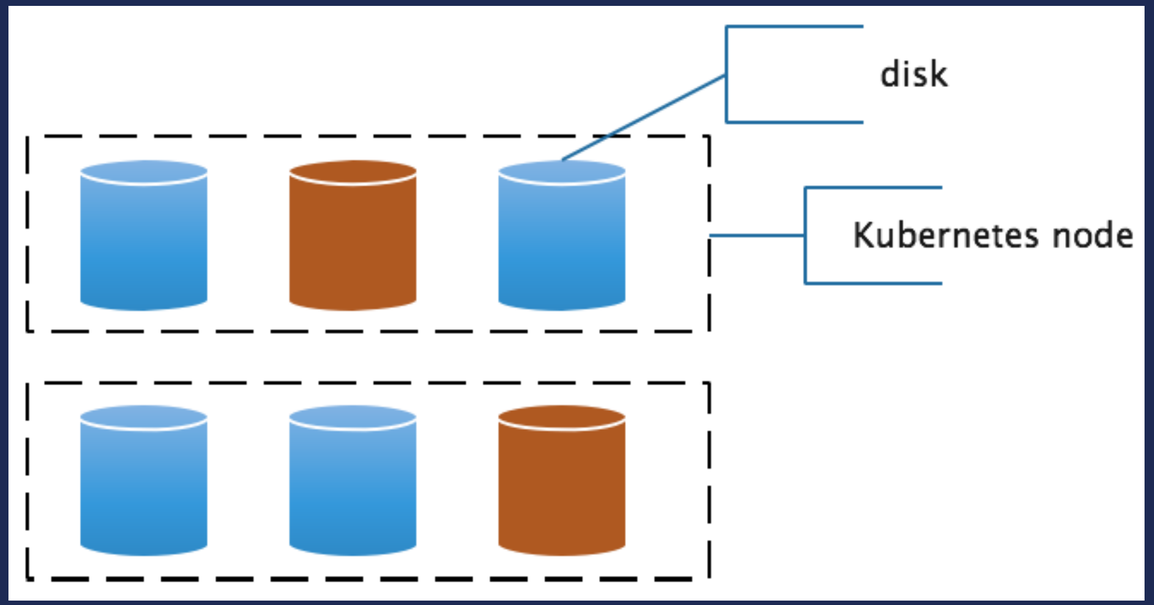



创建 Broker
调度
Kubernetes
平台API
调度器 API
API Server
创建存储 pod
主机磁盘
更新磁盘状态
监听 pod 建立事件
调度器
创建RC
API Server
监听 Broker 事件
更新状态
Kubernetes pod

服务器上线注册『磁盘』信息

etcd 保存的『磁盘』信息

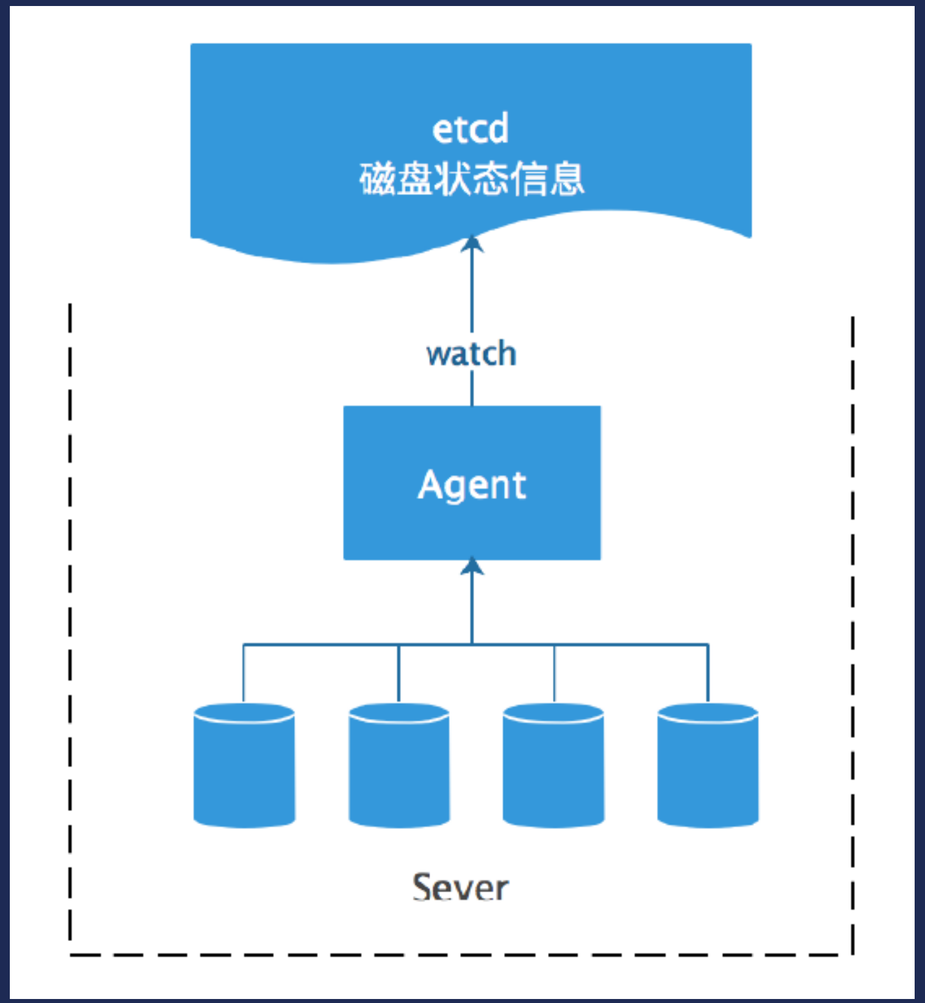
- 主机信息，比如 node
- 状态 : unused , used，cleaning
- 其他信息，例如集群信息



本地磁盘管理

Kubernetes 节点部署平台 Agent

- 监控服务器器磁盘状态
- 磁盘资源回收
- 磁盘故障处理

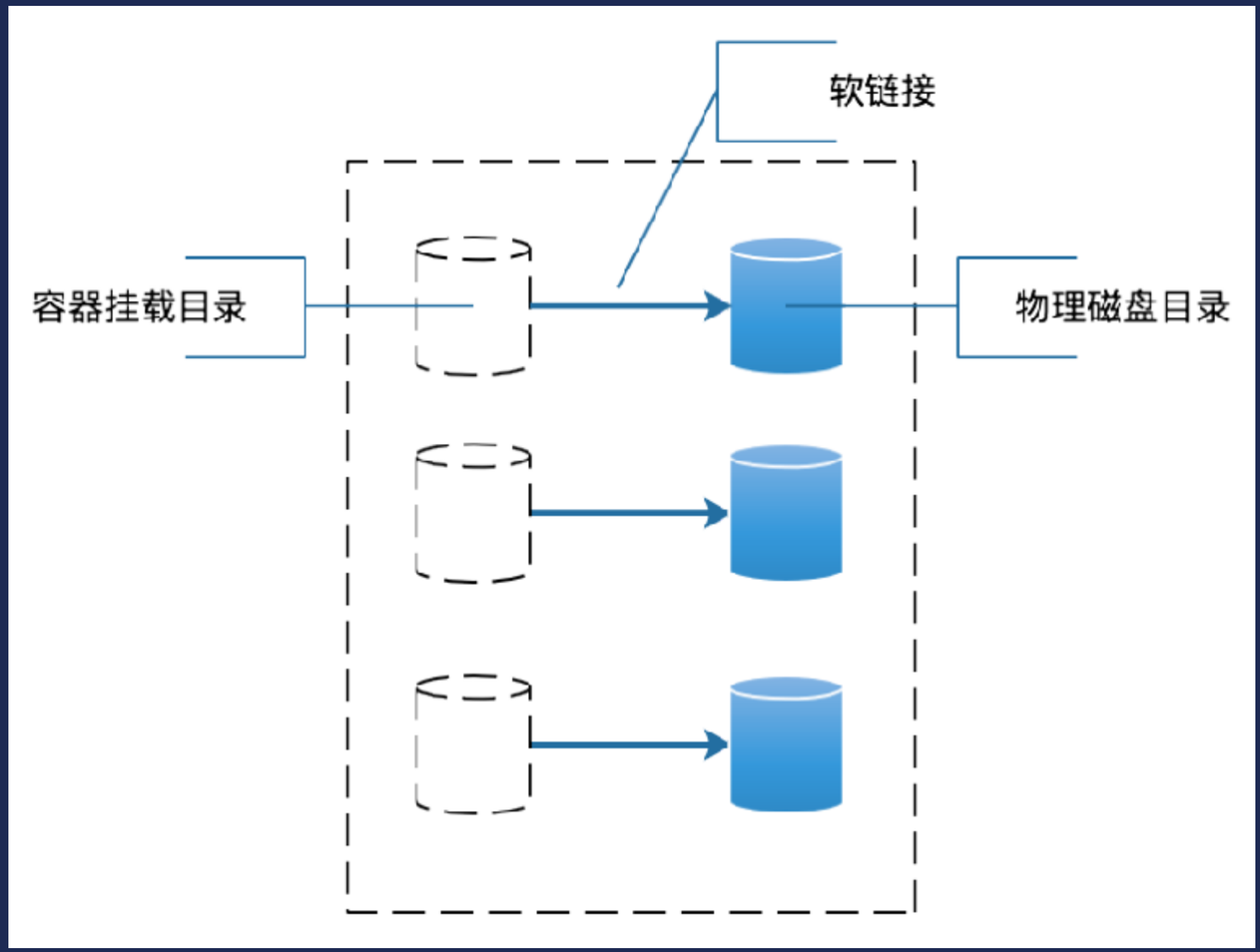


本地目录设计

容器挂载磁盘目录的软连接

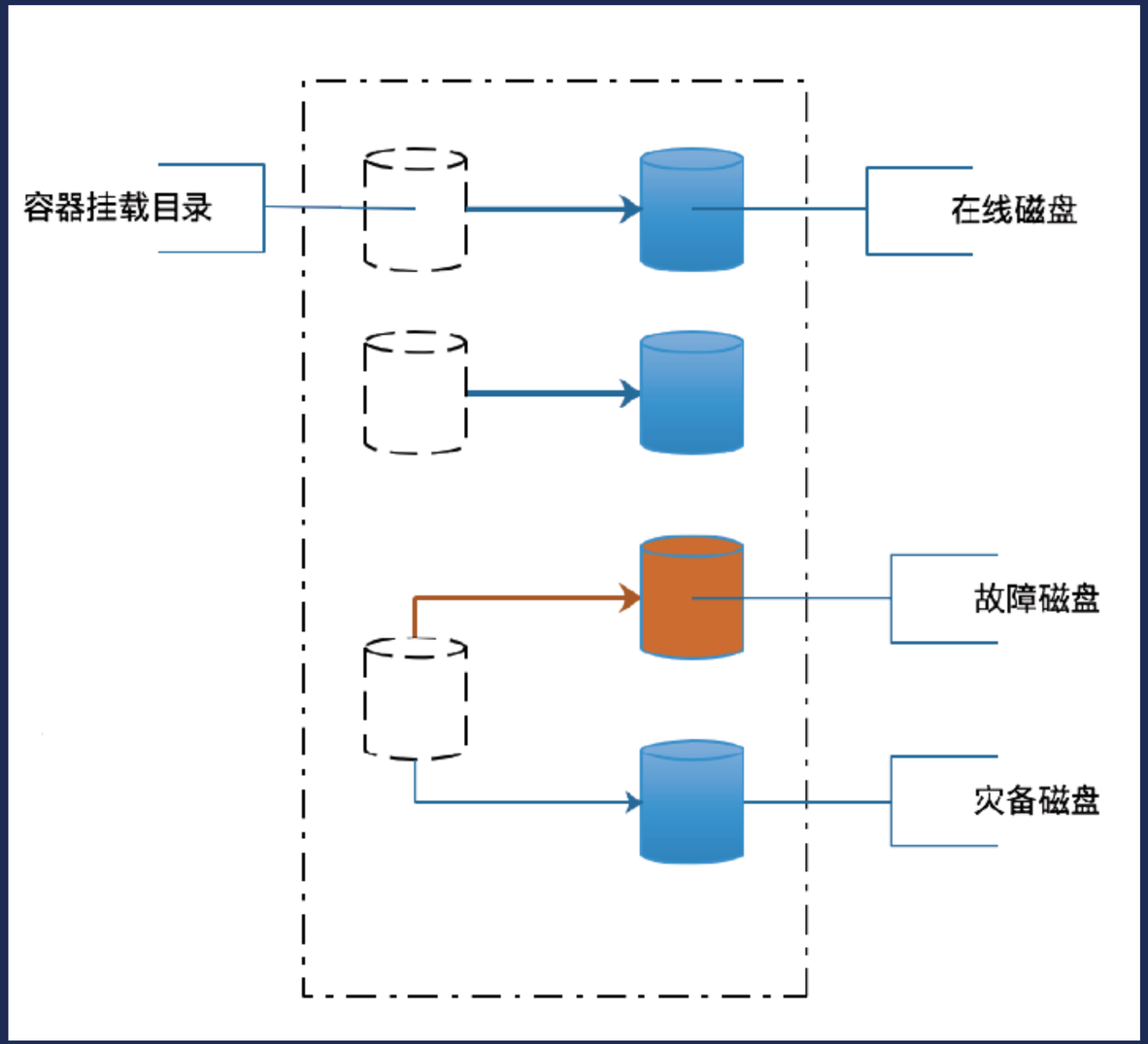
```
"spec" : {
   "volumes" : [
      {
         "name" : "storage",
         "hostPath" : {
            "path" : "/kafka-data5"
         }
      },
      {
         "name" : "hostname",
         "hostPath" : {
            "path" : "/etc"
         }
      }
   ],
   "nodeName" : "kafka01",
```

```
6 Oct 18  2016 kafka-data5 -> /data5
6 Oct 18  2016 kafka-data6 -> /data6
6 Oct 18  2016 kafka-data7 -> /data7
6 Oct 18  2016 kafka-data8 -> /data8
```



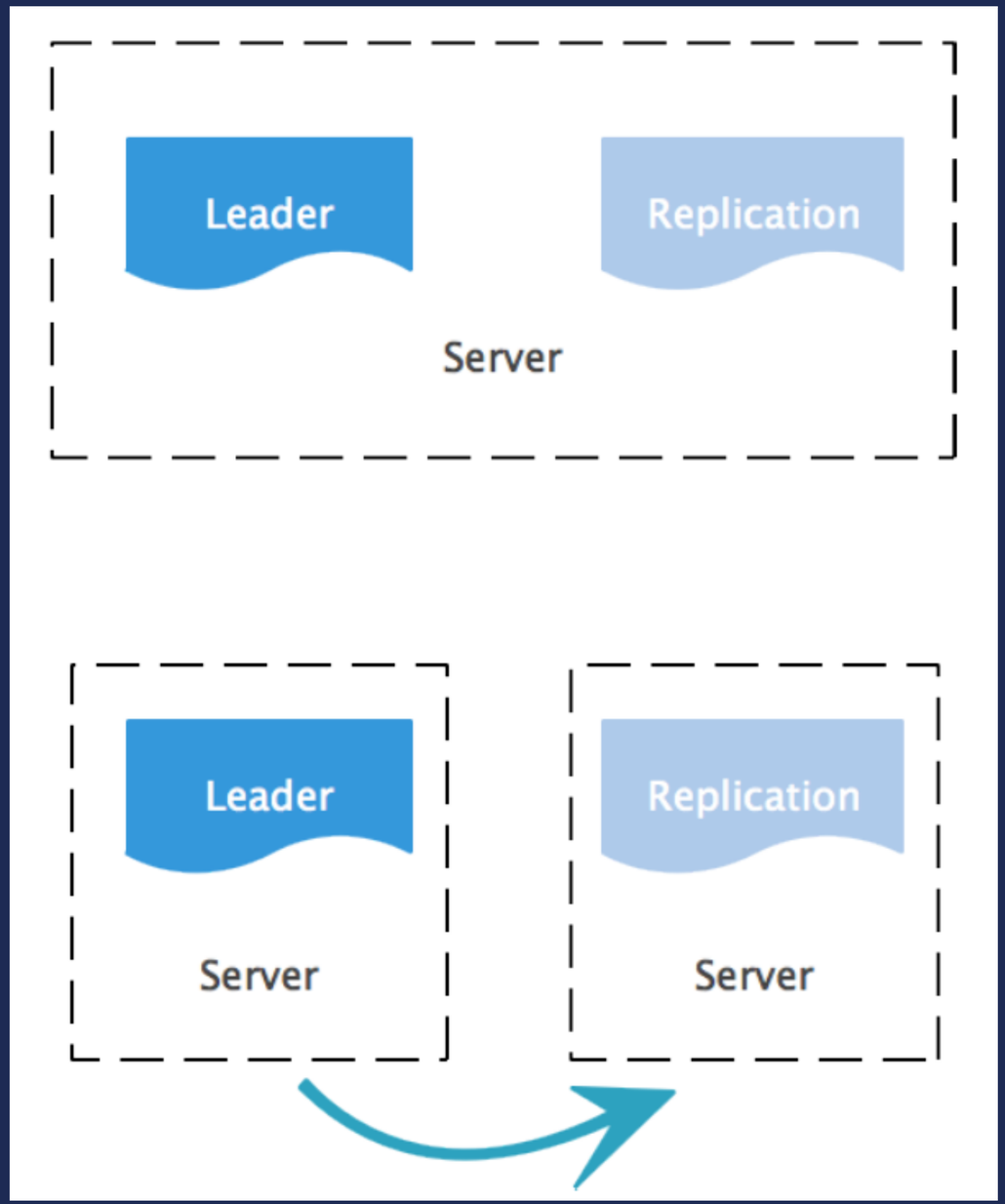
磁盘容错

- 磁盘故障不可避免
- 快速恢复，单盘故障启用备用盘



主机容错

- 磁盘调度算法
  - 运用 Kafka 机架感知特性
- 磁盘调度器重新选择主机



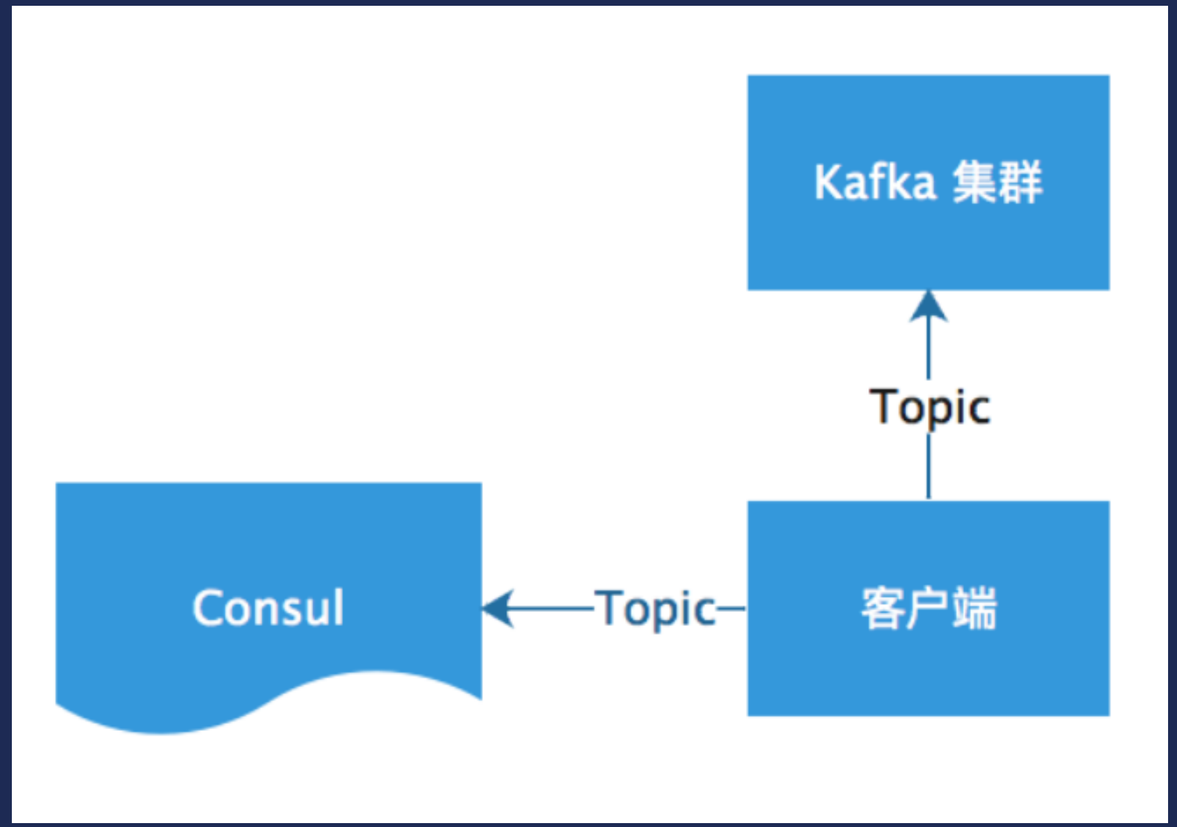
客户端

注册 Topic 的集群信息

- Broker, Zookeeper
- Status 是否启用

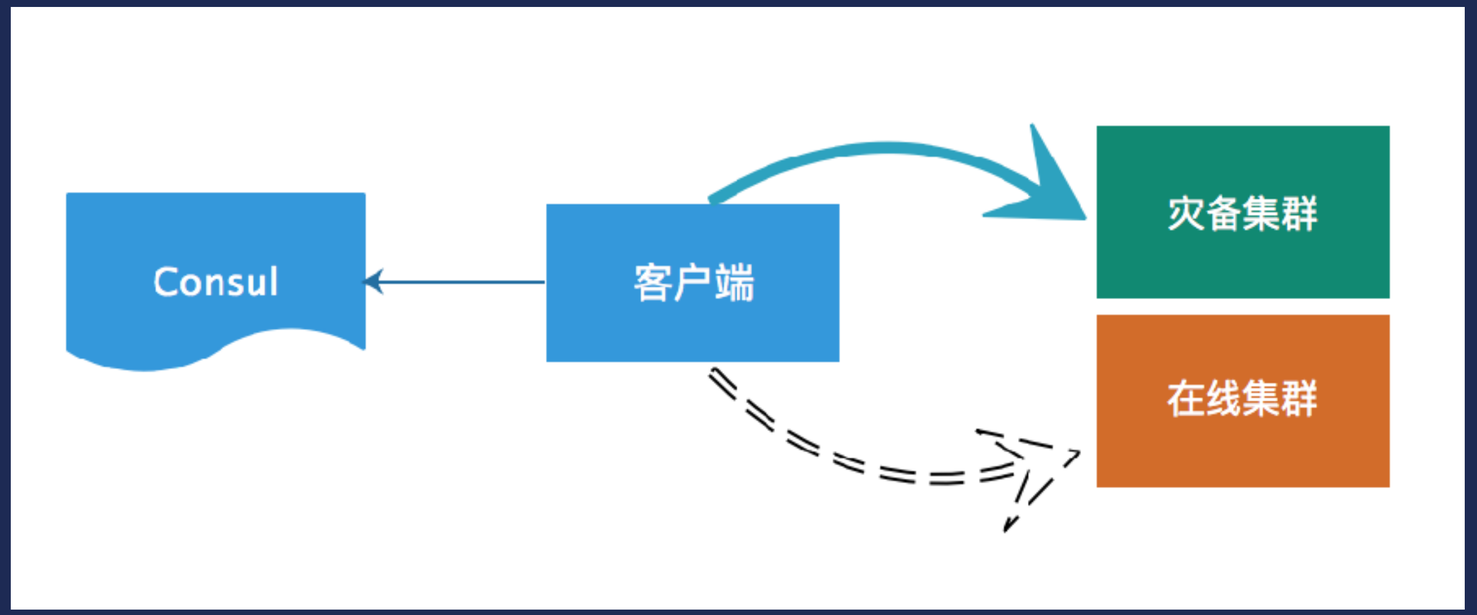
客户端

- 业务易用
- 标准客户端，降低集群风险



集群容错

- 灾备集群
  - 保证重要 Topic 高可用
- 客户端与服务器注册联动



监控

	指标维度	举例
Kubernetes	3	容器内存、CPU、运行状态
Broker	14	消息量，JVM， Leader分布，磁盘消耗
Topic	13	消息量，消费延迟
主机	4	内存、网络、CPU、磁盘
客户端	2	生产或消费 Topic 消息量

未来

平台自动化

提升资源使用率

Kubernetes 1.10 local-volume

Thanks!

联合主办方：腾讯云 | 开源中国 oschina.net | APACHE kafka A distributed streaming platform

直播支持：腾讯课堂 KE.QQ.COM 学习成就梦想