DTCC

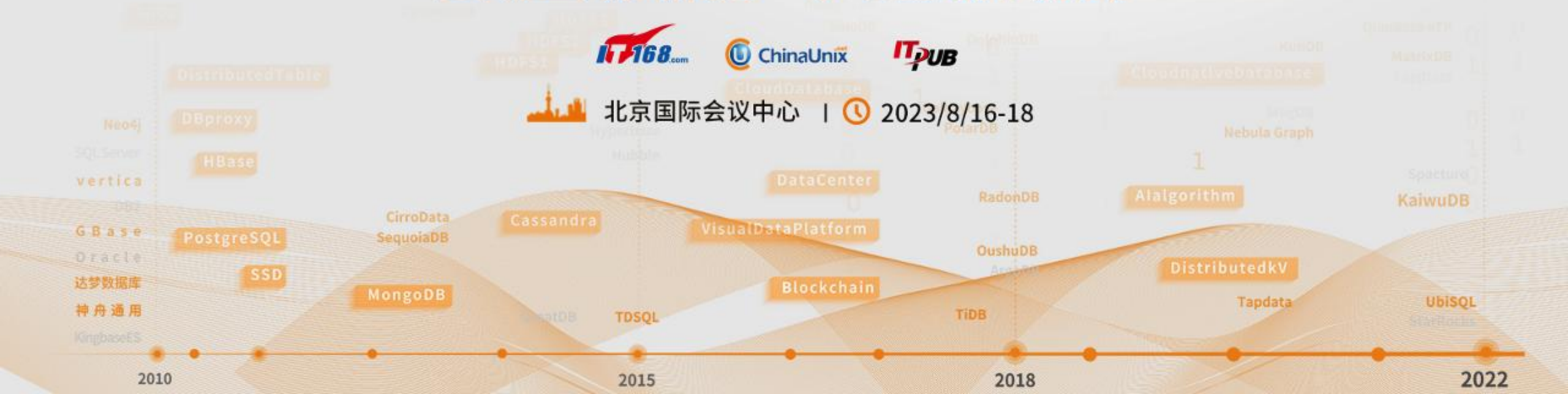
第十四届中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA

数智赋能 共筑未来

IT168.com ChinaUnix ITPUB

北京国际会议中心 | 2023/8/16-18



分布式数据库Blade云原生探索与实践

美团 数据库研发中心 陆宇

内容概要

- Blade简介
- 运维架构演进
- 云原生挑战和实践
- 现状及展望

Blade简介

Blade简介

背景

为什么美团需要Blade

- ✓ 业务需求驱动
- ✓ 设施成本优化

核心特性

- ✓ 高可用
- ✓ 易扩展
- ✓ 强一致
- ✓ 高度兼容MySQL协议但支持海量数据的分布式关系型数据库

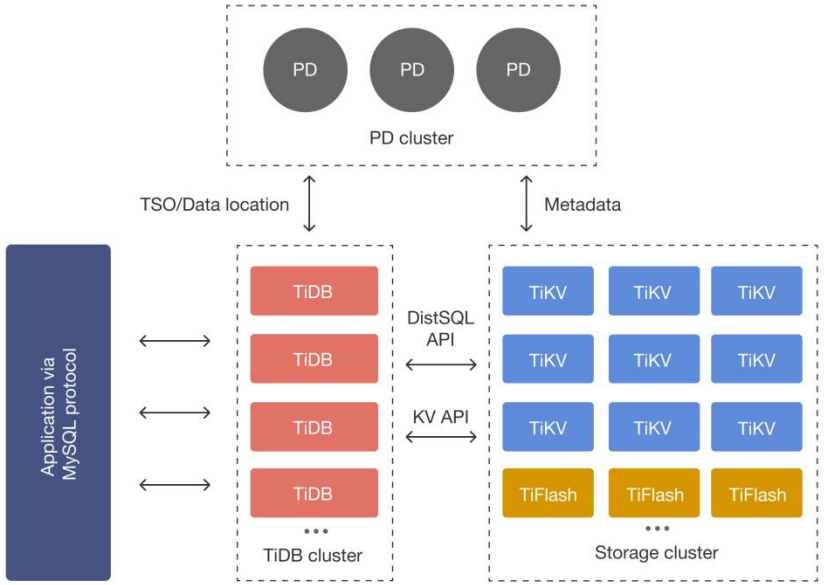
版本演进

- ➢ Blade 1.0（开源 + 虚拟化）
- ➢ Blade 2.0（自研 + 云原生）

Blade简介

Blade 1.0

- 基于开源TiDB构建
- 使用基于虚拟化的资源隔离
- 使用Ansible进行自动化脚本运维
- 搭建了管控运维平台来进行资源调度、集群变更等日常操作
- 解决了部分RDS场景下业务痛点
  - 分库分表业务耦合过深
  - 单机资源容量瓶颈
  - 主从复制延迟敏感
  - 归档成本高昂



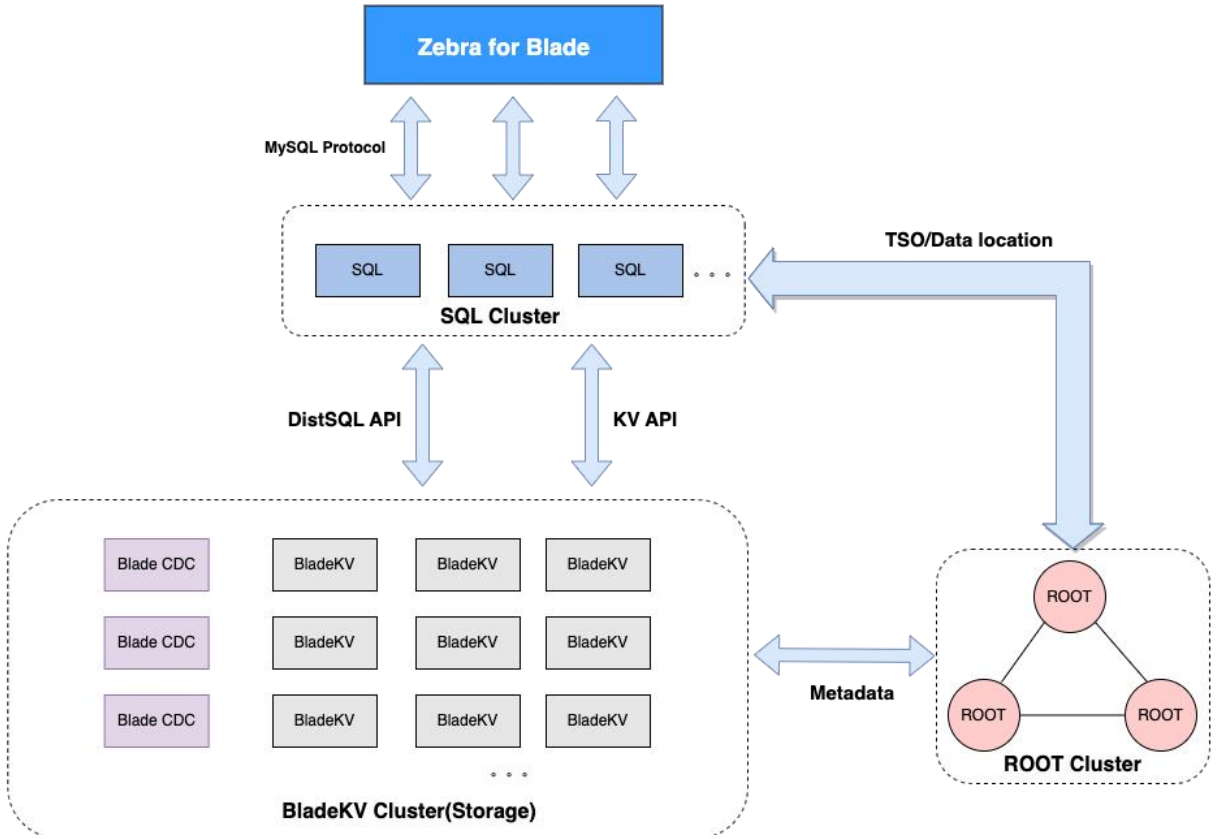
Blade简介

Blade 2.0

自研存储引擎，同时拥抱云原生

- 架构上继续兼容TiDB
  - 存储内核全新自研
  - 深度定制SQL计算层
- 针对美团内部场景进行专项优化
  - 提供与MySQL一致的RC事务隔离
  - 优化接口协议，降低事务关键链路的网络通信次数和IO次数，降低写入放大
  - 发挥大内存机型的容量优势，提升内存使用率
- 依托于云原生K8S资源调度实现
  - 存算分离调度
  - 容量水平扩缩
  - 跨机房、可用区的高可用容灾拓扑

等诸多特性



运维架构演进

运维架构演进

虚拟化运维架构

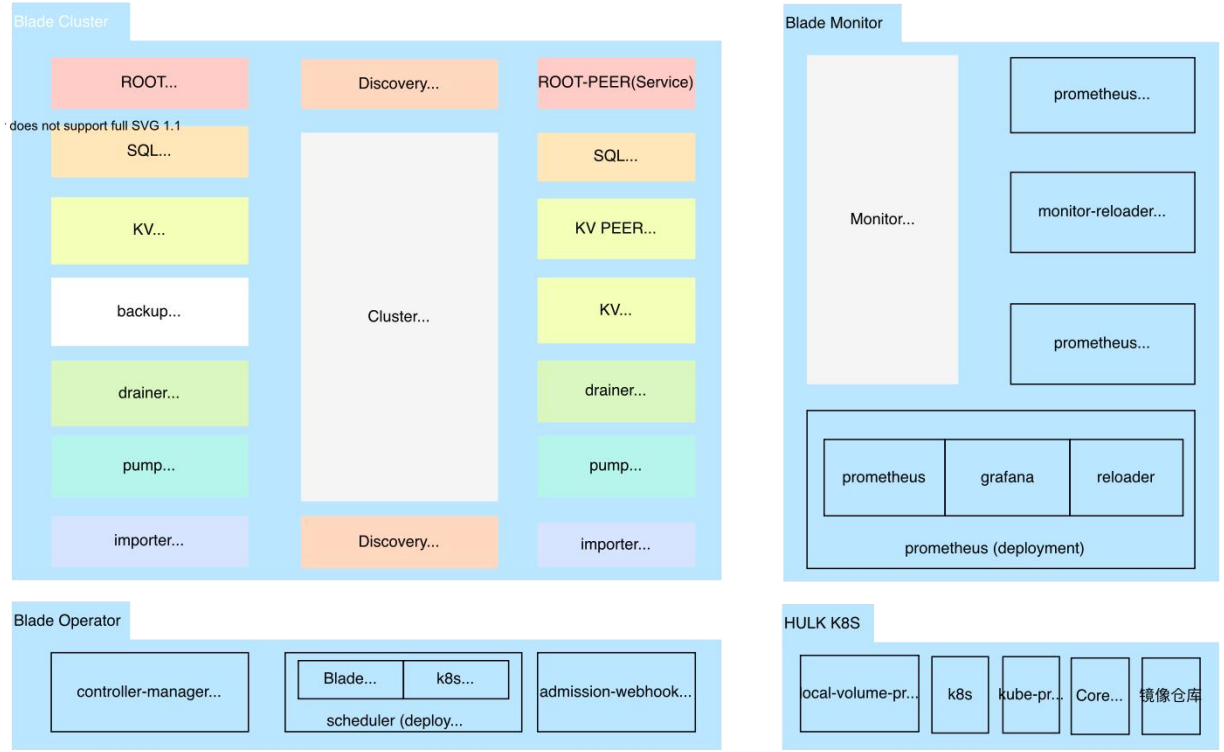
基于物理机资源的虚拟化节点部署

- 强依赖PaaS平台的资源管理流程
- 装机上线时间长
- 节点规格固定，无法按需灵活调整
- 无法快速完成故障节点摘除及集群自愈
- 中心管控节点基于SSH通道下发管理动作，安全维护成本较高

运维架构演进

云原生运维架构

- 底层基于美团自有K8S发行版（MKE）
  由系统调度团队提供基础资源安全保障
- 基础运维能力继承自开源tidb-opeartor
  - 得益于内核2.0架构与开源产品的高度兼容，避免了重复造轮子
  - 同时根据Blade自身实际需求进行深入定制开发
- 通过K8S丰富的扩展能力，实现多种运维逻辑的无侵入拓展，包括不限于：
  - 基于Webhook的资源回调及CMDB集成
  - 基于定制调度器的节点维护管理，及高可用定制

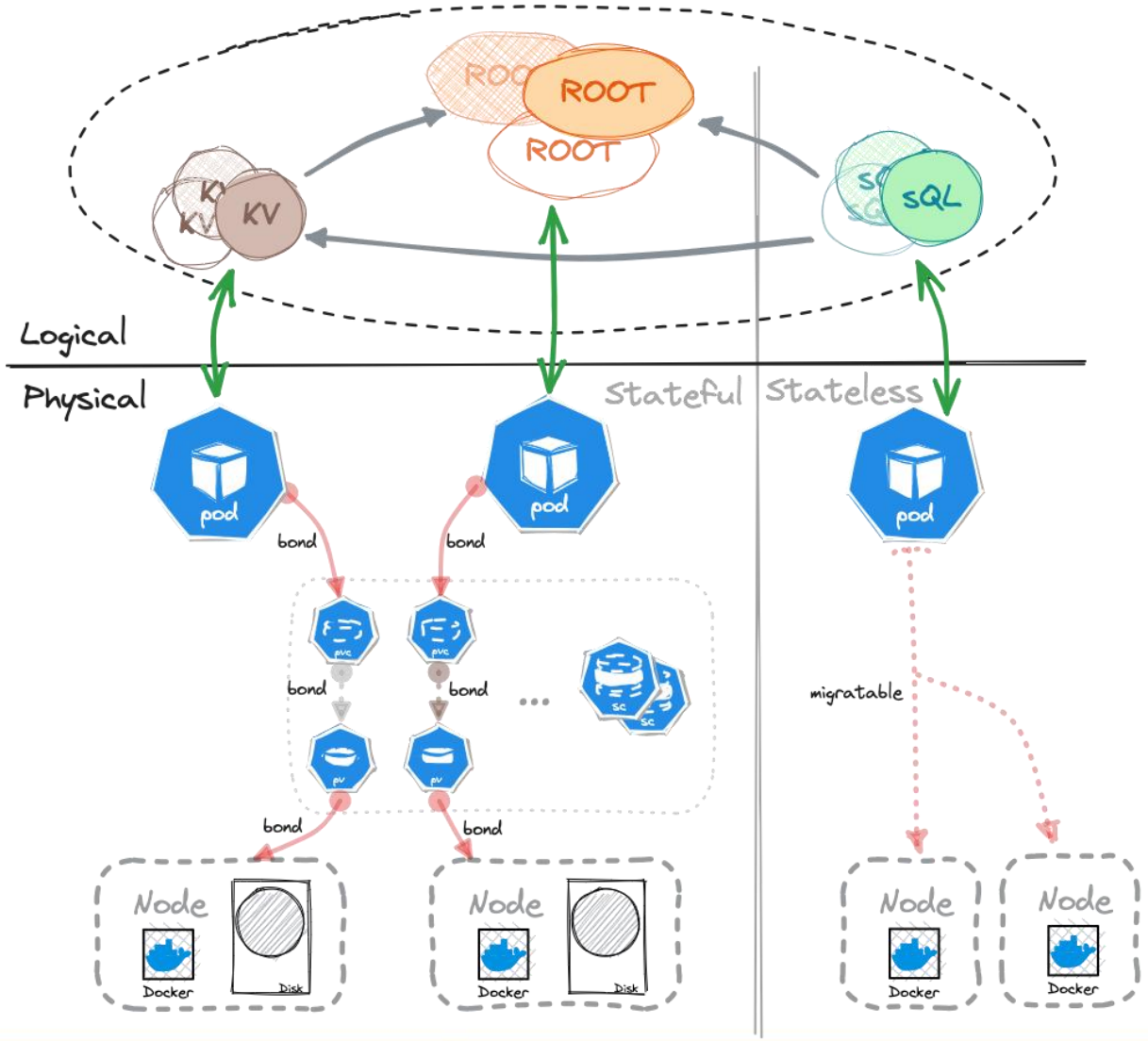


云原生挑战和实践

云原生挑战和实践

有状态的资源调度

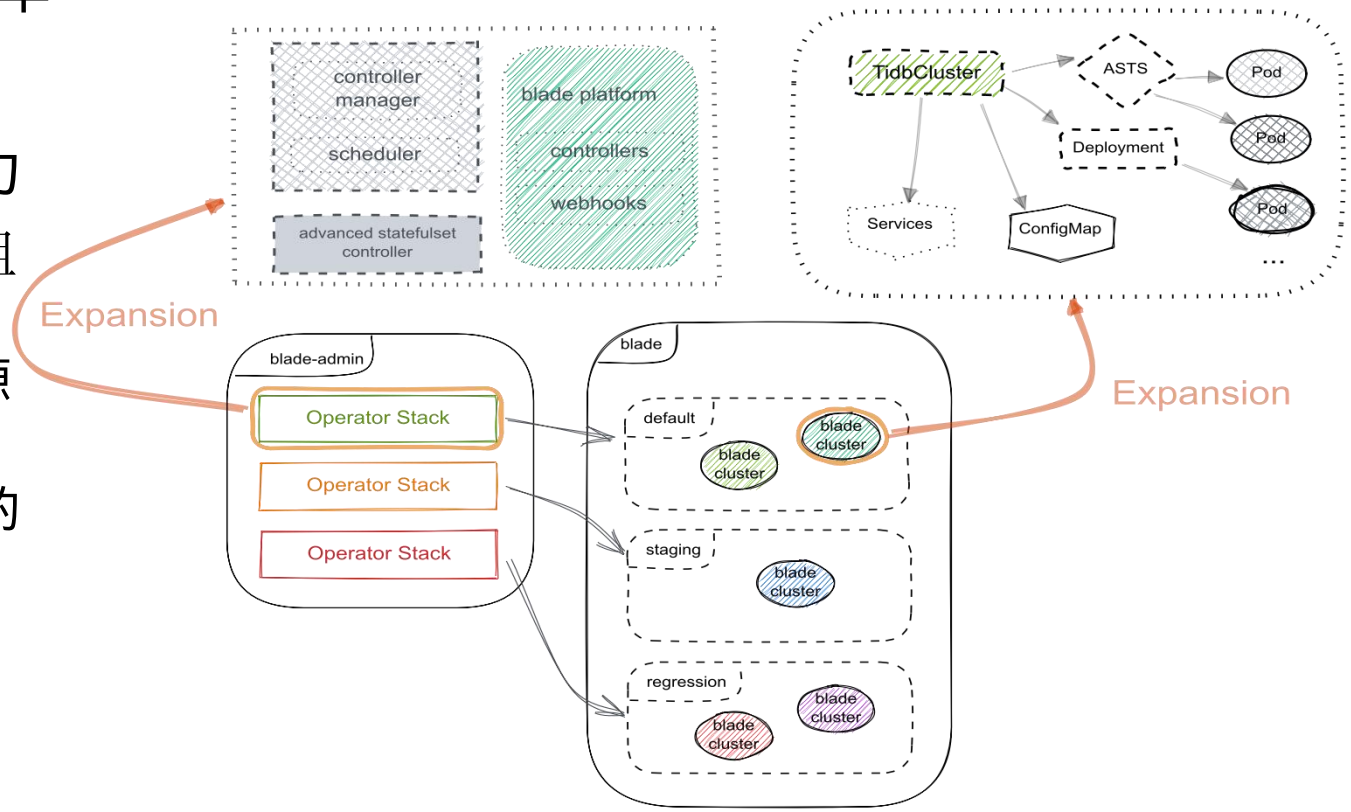
- 云原生天然更适合无状态服务
  - 设计上计算节点近乎无状态服务，可以更自由的横向扩展和转移
- 云原生的有状态服务维护存在诸多挑战
  - 存储状态
    - 网络远端存储性能无法满足大多数场景下的性能需求
    - 本地存储必然会导致Pod与宿主的绑定
  - 逻辑状态
    - 存储组件、调度组件都有逻辑选主的概念
    - 一致性系统中也需要优雅主动切主



云原生挑战和实践

大规模集群自动化运维

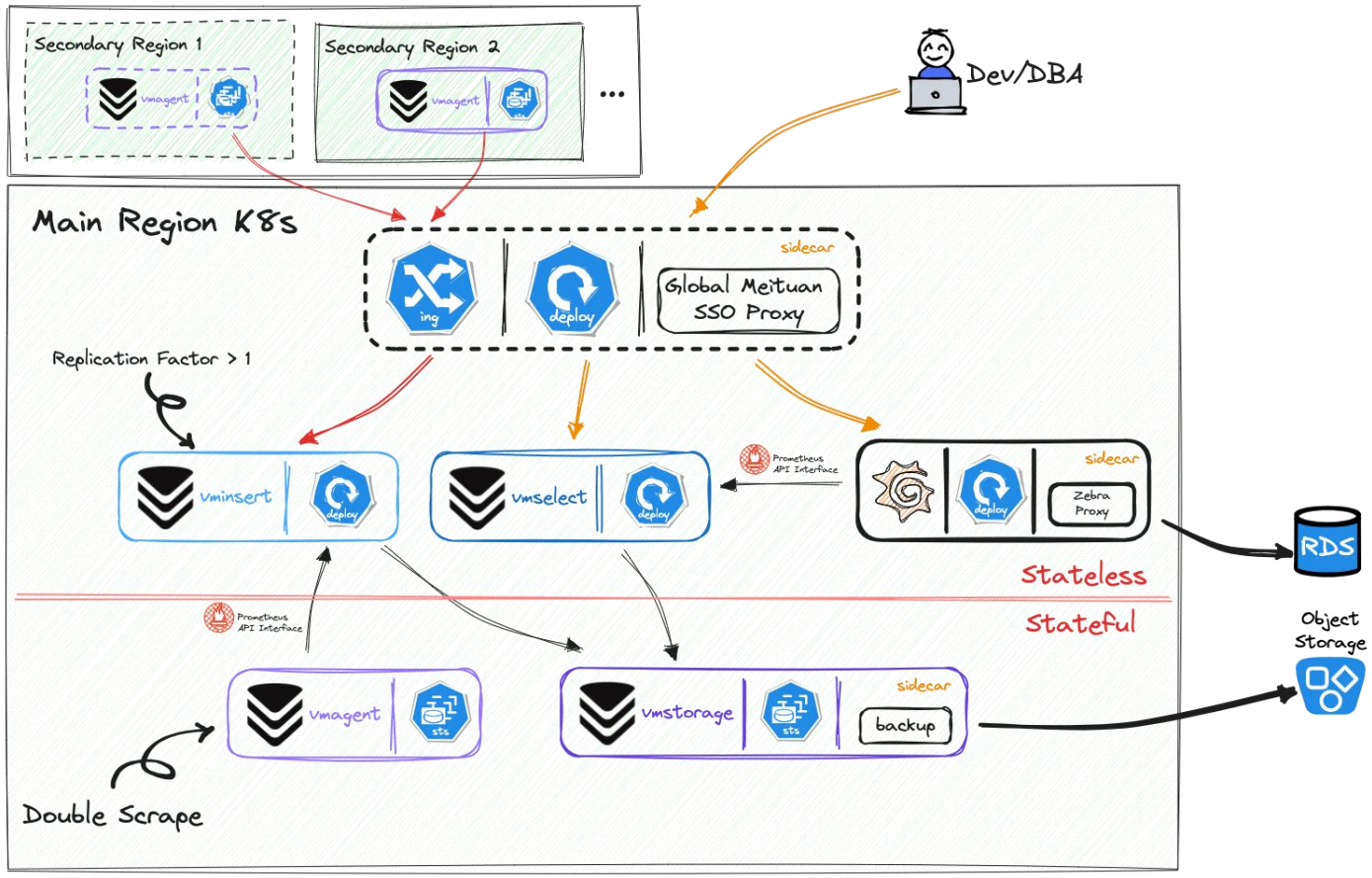
- 单一管控Operator实例无法满足效率及安全性要求
- Blade建设了逻辑运维实例隔离能力
  - 使用标签选择的方式为业务CRD分组
  - 基于OwnerReference机制在各Controller、Webhook中实现底层资源的动态逻辑分组隔离
  - 按照服务优先等级进行新功能特性的灰度分级发布



云原生挑战和实践

Blade服务感知体系

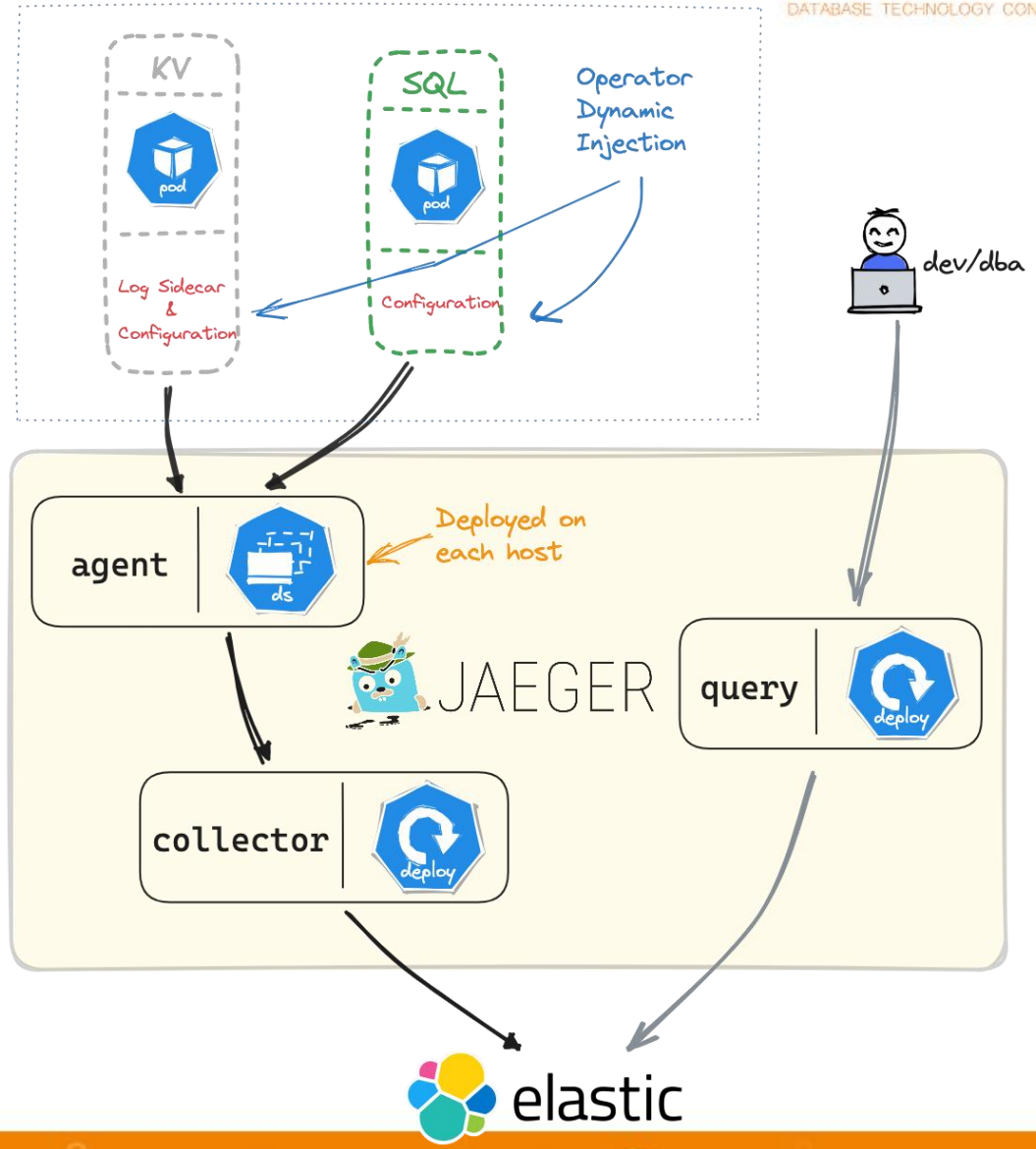
- 弃用TidbMonitor资源
  - 能够自动运维Prometheus及Grafana
  - 定制程度低，没有高可用方案
  - 独享TM资源消耗高
  - 共享TM维护难度大
- 基于开源VictoriaMetrics搭建分布式高可用监控体系
  - 各组件可横向扩展
  - 支持采集端、存储端双维度数据冗余
  - 支持监控数据备份
  - 兼容Prometheus及诸多其他系统API



云原生挑战和实践

基于边车的SQL追踪能力建设

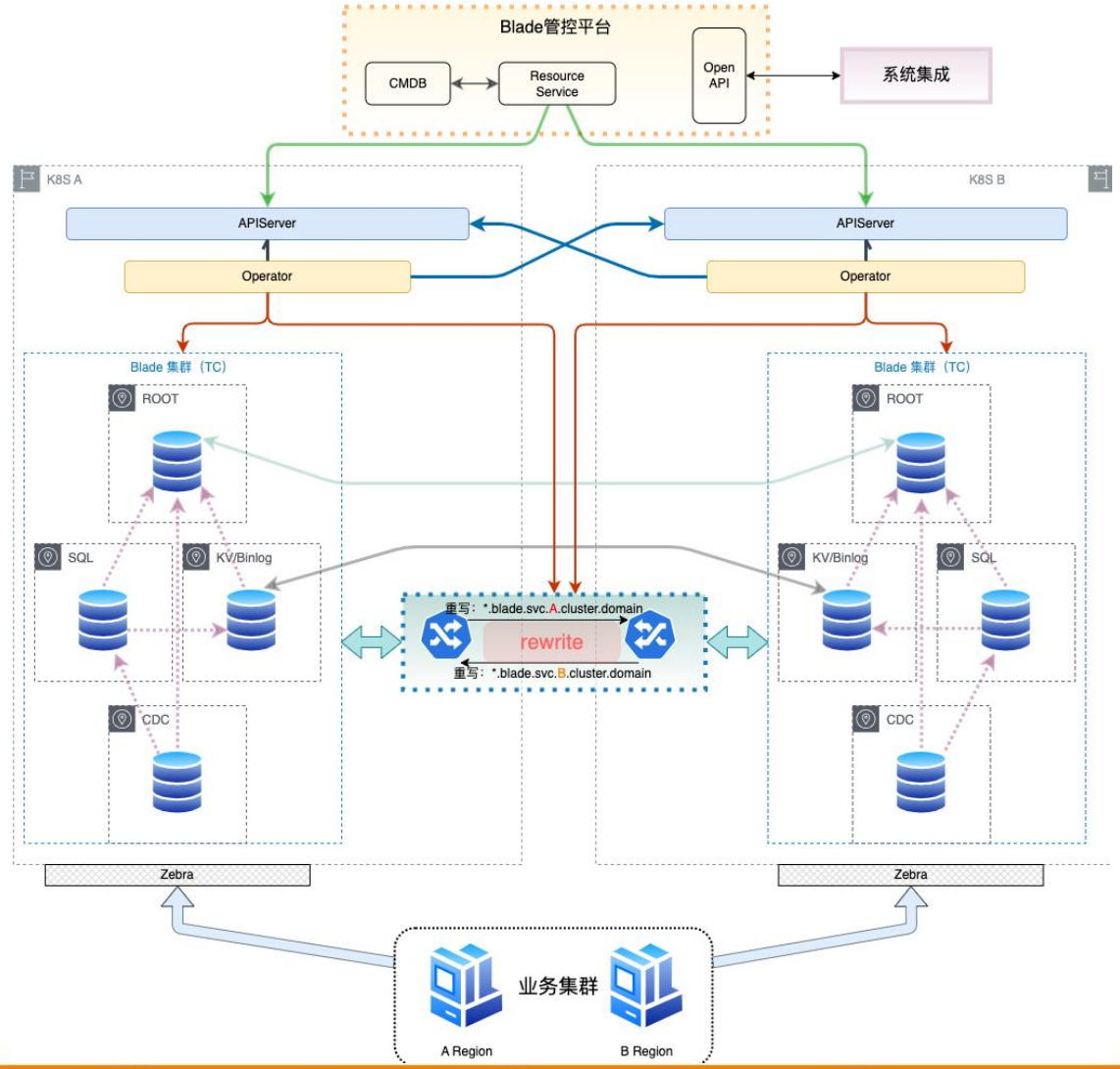
- 复用TiDB上游计算层基础追踪能力
- BladeKV存储内核实现基于日志的追踪数据采集及上报
- Operator能力建设支持自动在服务Pod中注入各类辅助应用，包括追踪日志分析采集程序、节点级代理通信转发程序等追踪基础设施边车
- 基于DaemonSet、Deployment对存储、采集以及查询各个组件进行云原生高可用部署



云原生挑战和实践

跨地域的集群迁移及容灾

- Blade内核
  - 基于Placement Rule实现了支持地域优先的调度和数据迁移能力
- 云原生架构设计
  - 得益于扁平的二层网络基础设施，不要求K8S基座跨地域部署
  - 定制Blade集群发现服务，实现基于地域标识的域名解析能力
  - 通过Webhook定制各集群组件Pod的服务发现底层配置，从而实现跨地域的地址解析与通信
  - 深入定制优化Operator对地域标识的兼容性，平滑变更集群的跨地域部署结构
- 业务接入
  - 在SDK层实现地域内流量闭环能力



现状及展望

现状及展望

- 目前Blade已经为美团多个业务线提供线上服务，数据总量达到PB级。
- 在宿主机器资源利用率超过45%的同时保障业务端到端可用率高于99.995%。
- 生产环境云原生已有五套K8S集群，跨三个主要地域，四个服务专区，支撑近万个容器化服务实例安全稳定运行，每天执行超过80万次自动化运维循环。

业务生产集群	物理宿主	业务请求
数百套	数千台	数百亿/天

现状及展望

未来规划

- 全面实现自研2.0替换开源1.0，完成所有业务集群的平滑无感升级
- 设计实现计算、存储资源隔离的多租户公共集群方案，降低业务运维复杂度和成本
- 探索基于网络存储的低成本小规模集群，进一步将本增效
- 应用智能网卡卸载内核网络中断，挖掘硬件潜力进一步提升吞吐降低延迟
- 进一步拥抱云原生，完成管控端服务的Serverless改造
- 深化云原生服务感知挖掘，进一步建设日志处理分析能力，提升服务可观测性
- 完成支持流量亲和的常态跨地域容灾集群部署架构

End