降本增效的利器：TiDB+PikiwiDB(Pika) 双剑合璧

赵新

360集团技术中台基础架构团队

目录

个人简介

网名：于雨

- **PikiwiDB(Pika) 项目负责人**
- **apache/dubbo-go 项目创始人**
- **前蚂蚁集团 Seata 开源负责人**

- **2021 年阿里开源先锋人物、阿里开源大使**
- **2022 开放原子开源基金会 年度开源贡献之星**
- **2022 信通院 OSCAR 尖峰开源人物**
- **dubbo-go: 中国科学技术协会 2021年度优秀开源产品**

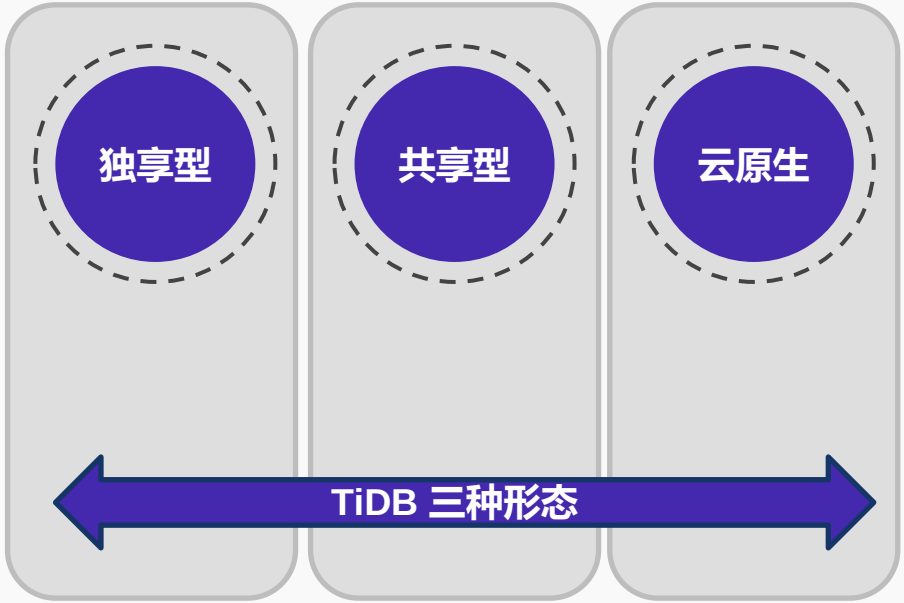
01 降本增效之 TiDB

产品形态

面临问题

- 独享型：大业务独占，自动扩缩
- 共享型：小业务共享，资源隔离
- 云原生：K8s + TiDB-Operator

为业务提供一个高可用、海量存储、强一致、易维护、分析能力强的类 MySQL 数据库，解决业务分库分表带来的巨大技术架构调整，满足业务对分析型数据仓库的迫切需求，助力业务突破大数据量单服务器存储无法支撑的限制，克服数据库上云本地盘无法高可用的风险。

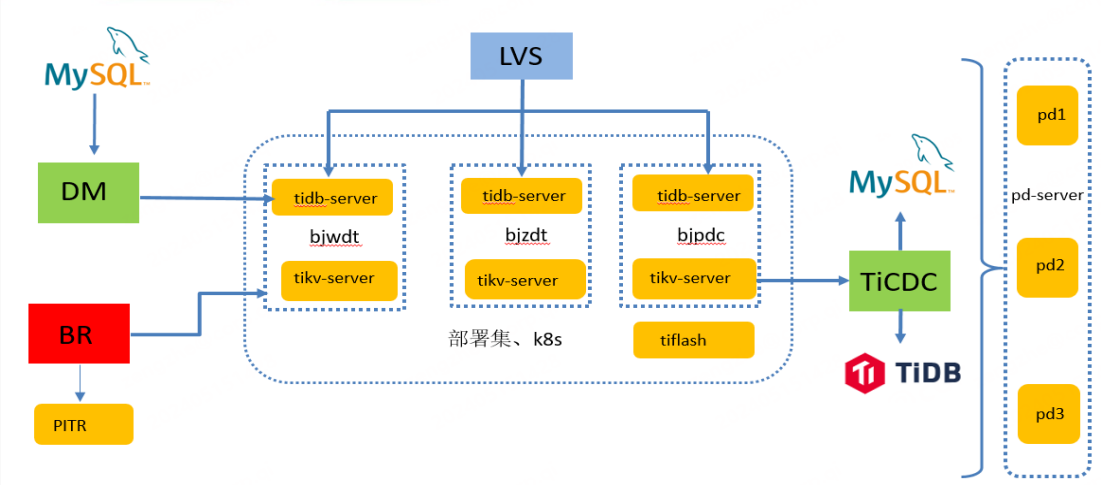


独享型

组件	说明
load balancer	负载均衡控制器
ProxySQL-cluster	SQL 路由、鉴权、审计
TiDB-server	Tidb集群计算节点
TiKV-server	Tidb集群存储节点
Pd-server	Tidb集群管理节点
Ti-Flash	Tidb集群分析节点
DM	数据迁移工具
TiCDC	数据同步工具
BR	数据备份工具
S3	数据备份介质
V-Metrics	时序数据库
Grafana	监控图形化展示
部署集	独享服务集群
K8S	独享型集群服务器
物理机	共享性/托管行服务器

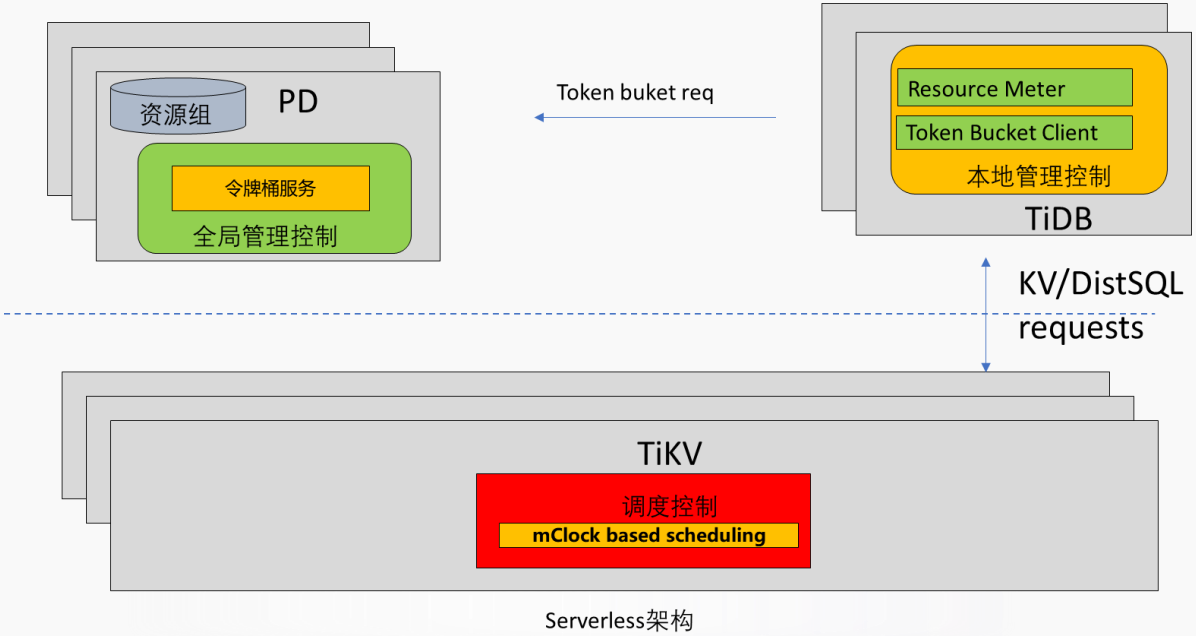
慢 SQL 处置:

建立 TiDB 表健康度巡检和处理机制，对健康度低于 95% 以下表进行 analyze 处理，防止查询耗时不稳定，波动较大，甚至 OOM 被 kill 。同时控制 analyze 并发线程数，保证线上服务稳定。



共享型

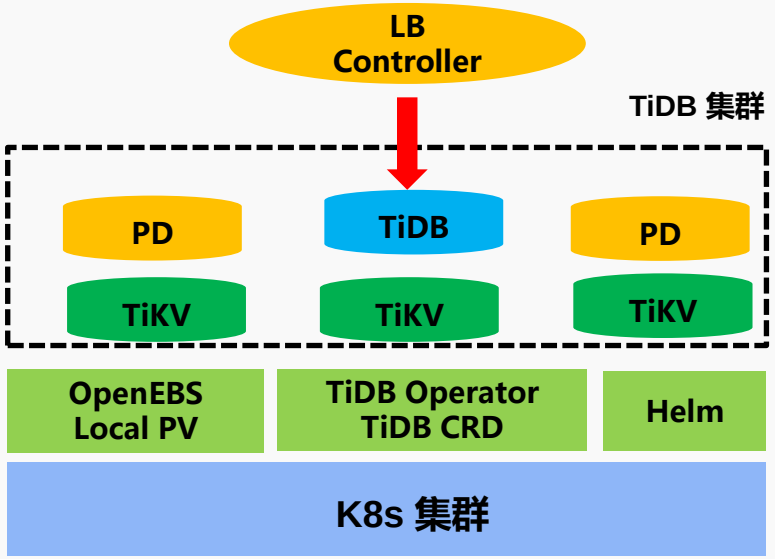
共享型主要基于资源管控，让多个小业务用户无需申请单独的 TiDB 集群，又有资源独占的体验。



TiDB 云原生集群规格

共享型主要基于资源管控，让多个小业务用户无需申请单独的 TiDB 集群，又有资源独占的体验。

组件	说明
Kubernetes	v1.24+
OpenEBS	Storage-class
TiDB-Operator	Tidb集群计算节点
Helm	v3.0.0+
CLASS STORAGE	Tidb集群管理节点
LB Conrtroller	负载均衡，实现外部服务访问集群内部 IP



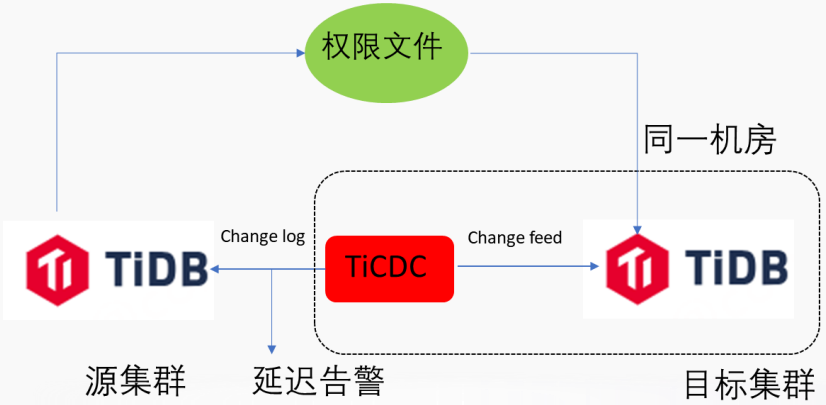
TiDB PV

1. 存储数据、日志
2. 网络 PV 存在网络延迟，且性能不可靠
3. 官方提供的 local-volume-provisioner，但是其为静态 PV，需要各种手动创建，复杂繁琐
4. 阿里的 Open-Local 相对于磁盘分区模式性能损耗 5%~10%
5. 采用支持动态分配存储的 OpenEBS 作为 local pv，既不需要提前创建大量静态存 storage_class，也无需担心磁盘是否独占（考虑 db 混布，充分利用资源成本），且配置简单

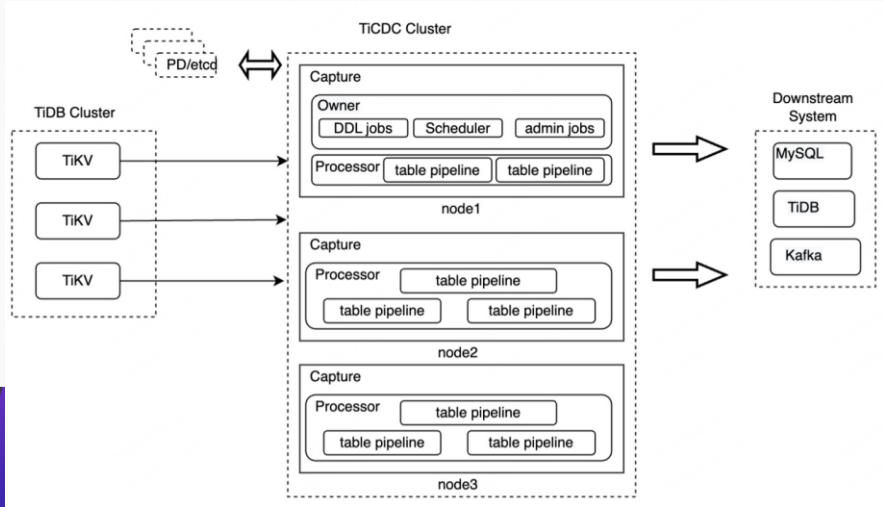
数据备份

延迟优化

问题：主备集群存在延迟较大的情况，虽然能控制 RTO 在分钟级别，但是 RPO 风险较大



解决方案：控制上游大事务，扩充 TiCDC 规模，调大 per-table-memory-quota，开启单表跨节点同步，TiCDC 和下游集群部署一起。

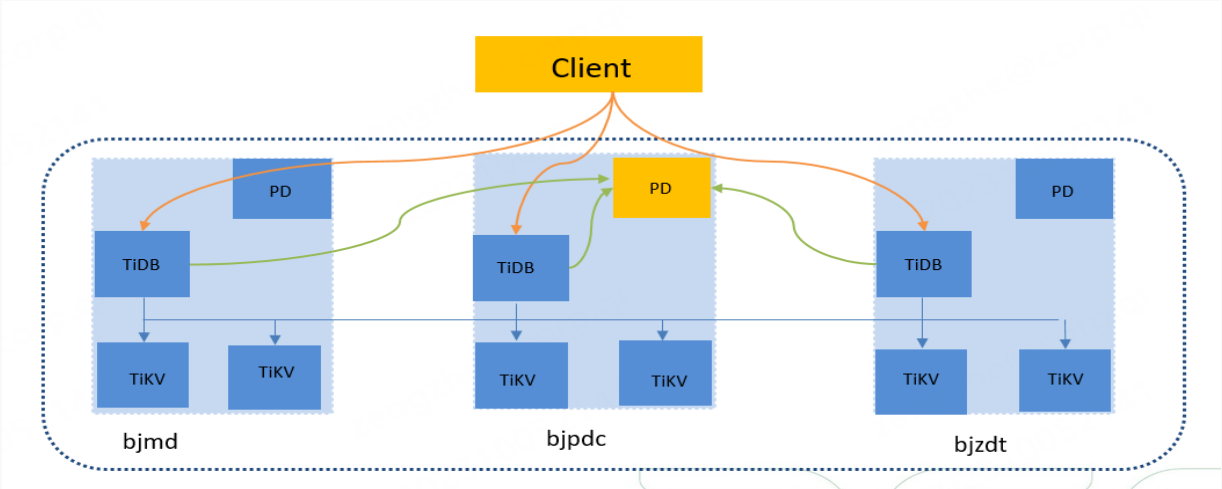


跨机房多活

延迟优化

问题：为保证同区域集群高可用，会进行同区域跨机房部署，但是会带来查询耗时增加的问题。

解决方案：节点配置以 [zone, host] 配置标签，配置 region 分布策略，对业务表进行策略加持，将 SQL 耗时降低85%以上，QPS 提高700%左右。

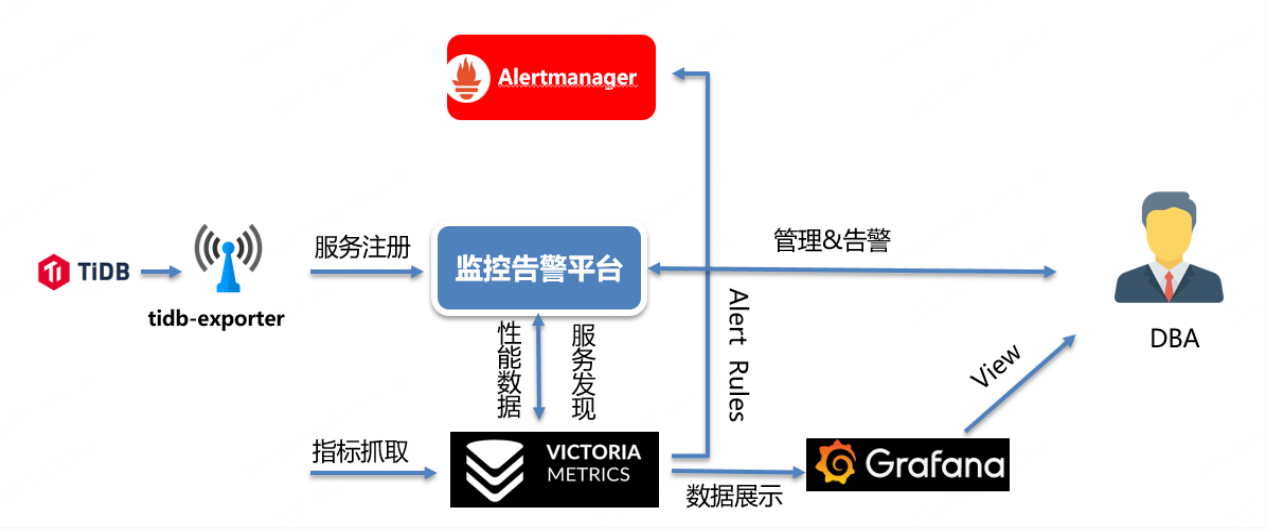


监控告警

统一监控

问题：TiDB自带的监控告警体系虽然很完整，但是自成体系，维护复杂，每一个都是监控孤岛，如何统一管理并融合到DBA自己的监控告警体系当中。

解决方案：通过废除TiDB自带监控告警，纳入到DBA监控告警平台统一管理，既解决了资源浪费问题，又可以做到监控告警统一管理，极大的提高运维效率。



总结

业务收益

TiDB 项目上线之后解决了业务分库分表的难题，极大提升业务数据分析能力，也解决了单块盘无法支撑业务数据规模的问题。

三种形态

TiDB 共享性化为小规模业务提供了使用平台；TiDB 云原生可以完美契合ToB 业务，提高一线同学交付效率。

上线规模

TiDB 项目目前已经全面覆盖集团业务线, 20 套集群,135 T+ 数据规模,最大单表 4.68 亿，稳定高效运行。

02 降本增效之 PikiwiDB(Pika)

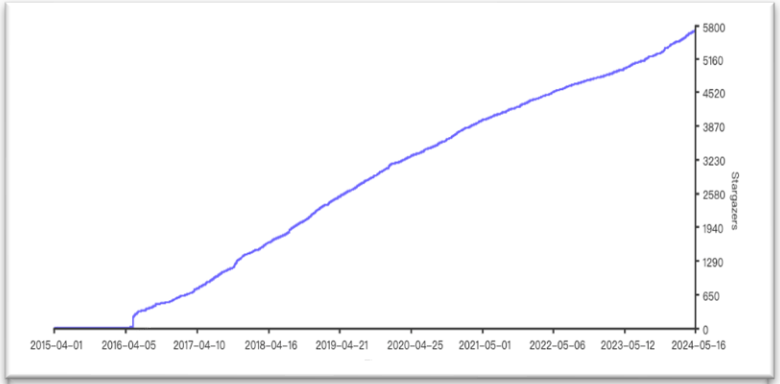
发展历史

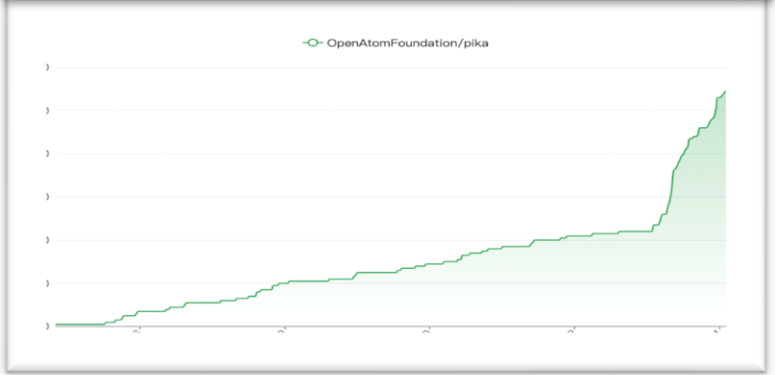
品牌升级

Pi-kiwi-DB：

1. "Pi" 音同 "π"
2. "Pik" 恰好保留了 "Pika" 的前三个字母
3. "kiwi" 音同 "KV"，形同几维鸟
4. 寓意极大容量、极致性能

PikiwiDB





2015.04 项目启动 → 2015.11 发布 1.0 → 2016.02 开源 → 2016.04 发布 2.0 → 2018.08 发布 3.0 → 2020.08 申请加入 OpenAtom → 2021.03 孵化运营

项目定位

产品特性

- **协议兼容：**完全兼容 Redis 协议，且极力追求高性能、大容量、低成本、大规模
- **数据结构：**支持 Redis 的常用数据结构 String/Hash、List、Zset、Set、Geo、Hyperloglog、Pubsub、Bitmap、Stream、ACL etc
- **冷热数据：**对热数据做缓存，全量数据持久化存储到 RocksDB，并且实现冷热分级存储
- **极大容量：**支持百 GB 的数据量级，减少服务器资源占用部署方式：单机主从模式 (slaveof) 和 Codis 集群模式

PikiwiDB(Pika) 是一款兼容 Redis 协议的开源高性能持久化的 NoSQL 产品。在业务数据大规模，对系统数据可靠性要求较高的高速请求场景下，是一个综合了速度与可靠性的解决方案且实现成本相对低廉。

PikiwiDB(Pika)力求在完全兼容 Redis 协议、继承 Redis 便捷运维设计的前提下，通过持久化存储的方式解决 Redis在大容量场景下的问题，如

应用场景

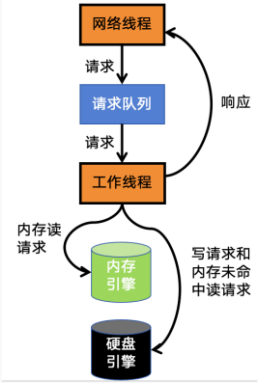
头部用户

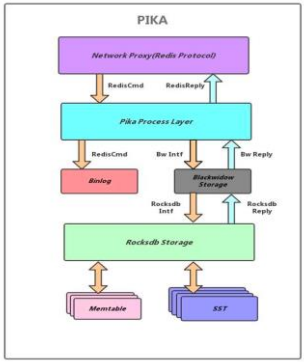
- 360 内部部署使用规模 10000+ 实例，每天访问量 2000 亿次，单实例数据量 1.8TB；
- 微博公司内部部署实例 10000+ 实例，每天访问量 240 亿次；
- 喜马拉雅(X Cache)实例数量 6000+，数据量 120TB+，每天访问量 400 亿次；

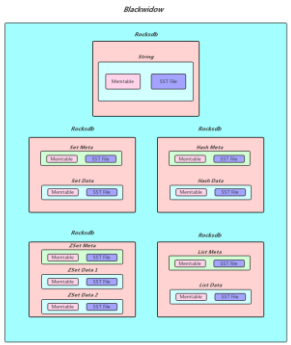
单体结构

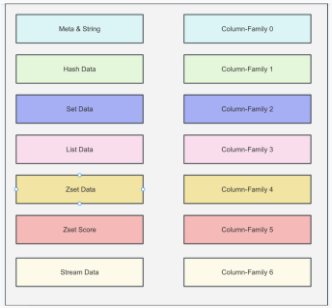
存储引擎 Floyd

- 支持多平台 centos、ubuntu、macOS
- 多线程模型
- 基于 RocksDB 的存储引擎
- 多粒度数据缓存模型







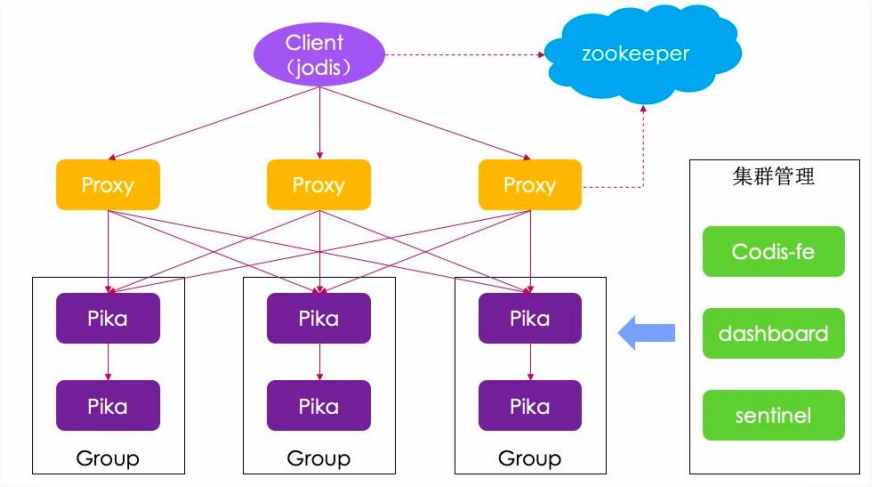


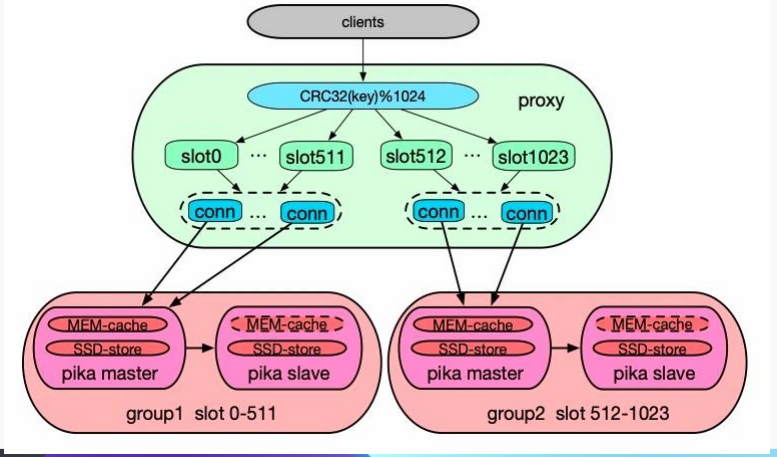
- 所有类型的 meta/data 都存到一个 RocksDB 中
- 解决旧版 Blackwidow 不同类型存在相同 key 的问题
- RocksDB 不再固定为 5 个，可任意多个但也不宜太多
- 关闭每个 RocksDB 的 WAL，提升写性能
- 同一个类型的数据可以分散存到多个 RocksDB 中，多个 RocksDB 矮化了 LSM Tree 高度
- 同样数据量的前提下，相比于 Blackwidow 单RocksDB，大大减轻了每个 RocksDB 的空间放大和读放大

集群架构

路由模式

- 采用 Codis 架构，支持多 Group
- 单 Group 内是一个主从集
- 以 Group 为单位进行弹性伸缩
- 以 slot 为单位进行预分配，默认 1024

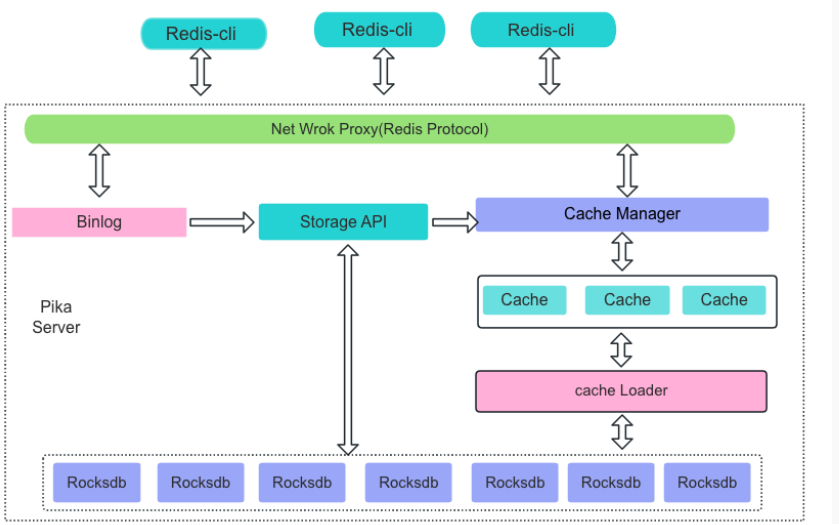


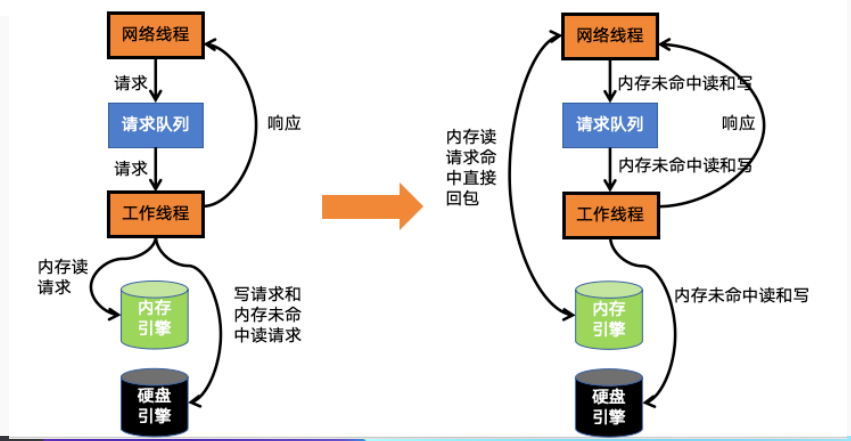


混合存储

性能参数

```
+--------------------+
| Uncompressed Cache |   (<10us, 10GB)
|     (In-mem)       |
+--------------------+
         |
         |
+--------------------+
|    Hybrid Cache    |   (~100us, 100GB)
|   (Block-level)    |
+--------------------+
         |
         |
+--------------------+
|     Disk Cache     |   (~1ms, 1TB)
|    (File-level)    |
+--------------------+
```



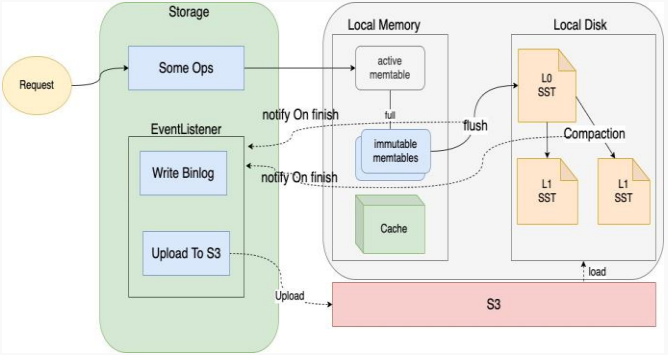


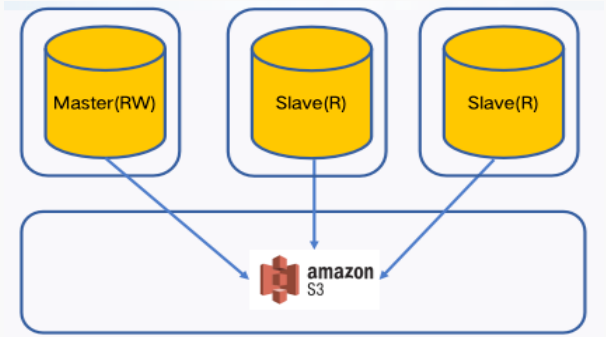
云原生

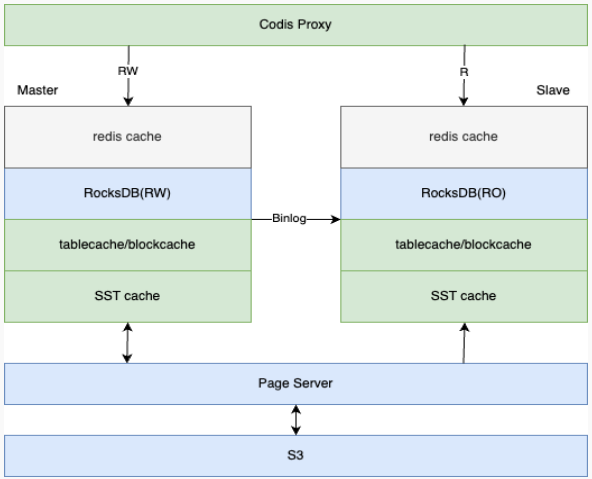
存算分离

解决的问题：

- 受限于单机磁盘容量
- 主从同步耗时长
- 不利于云环境进行弹性扩缩容

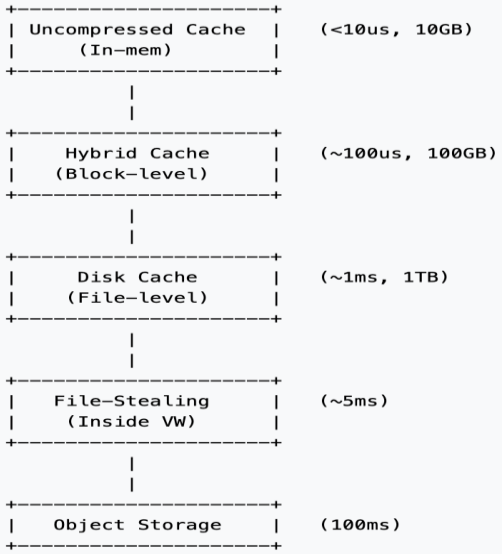






目标：

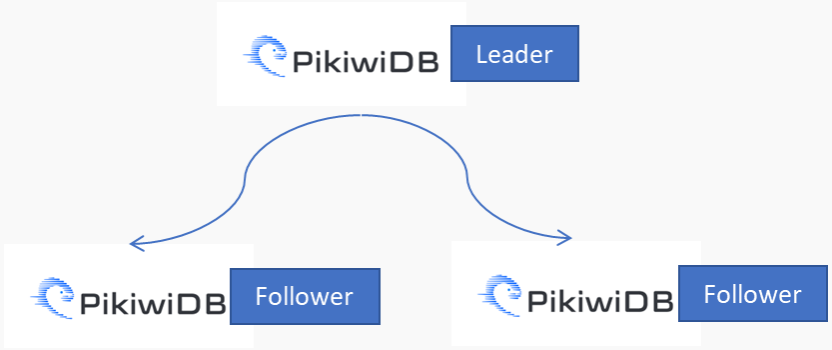
1. 并行写 WAL 提升写入速度
2. 将全量数据存储在S3
3. 主从节点使用同一份 S3 数据
4. 计算节点利用内存和本地磁盘实现两级 cache 减少用户请求耗时



强一致性

项目定位

Raft-Disk-Redis：该项目基于 Raft 协议构建，兼容 Redis 协议，是一款适用于大规模强一致性的键值存储数据库，特别适合存储高达 10TiB 级别的元数据场景



Raft Leader-Follower 强同步复制版本

Raft Leader-Learner 异步复制版本

repo: https://github.com/OpenAtomFoundation/pikiwidb

产品形态

四种产品

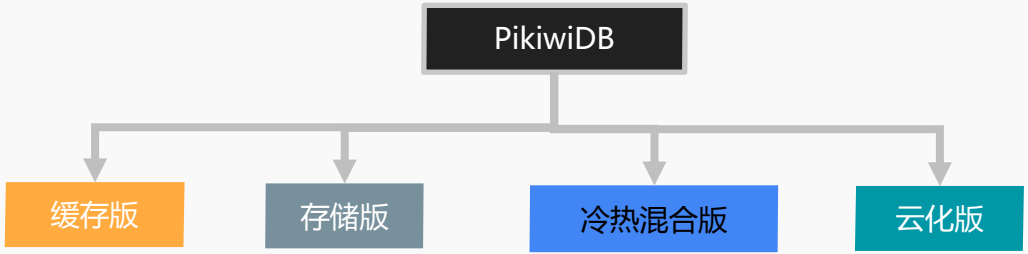
缓存版：大部分数据放置在内存，性能高，延迟低

存储版：使用 NVMe SSD 存储 PB 量级数据，降本增效

混合版： 缓存+存储混合架构：自动缓存热数据，降低访问延迟；

全量数据固化在磁盘，保障数据可靠性；平衡性能和成本

云化版：数据放置在低速介质或对象存储中，用于 Serverless 场景



社区运营

2023-1208　Oschina “2023 年度优秀开源技术团队”

2023-1213　被艾瑞咨询研究院列为 2023 年 “中国基础软件开源产业主要参与者”

2023-1229　PikiwiDB (Pika) 第一次以 PikiwiDB 的身份亮相 Oschina 2023 年《中国开源开发者报告》

2024-0201　“CSDN 2023 中国开发者影响力年度评选” 颁发的 “创新产品与解决方案” 奖

THANK YOU.