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

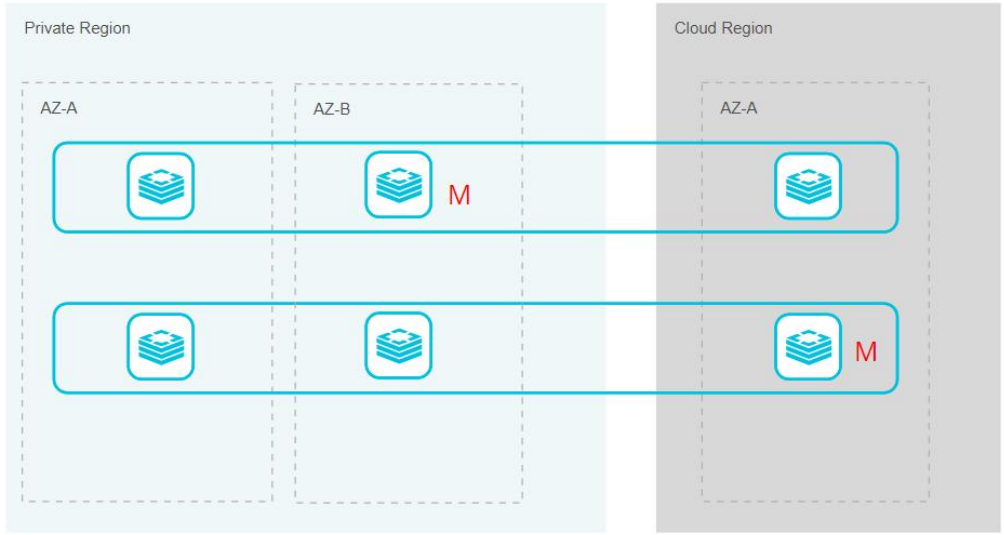
IT168.com ChinaUnix ITPUB

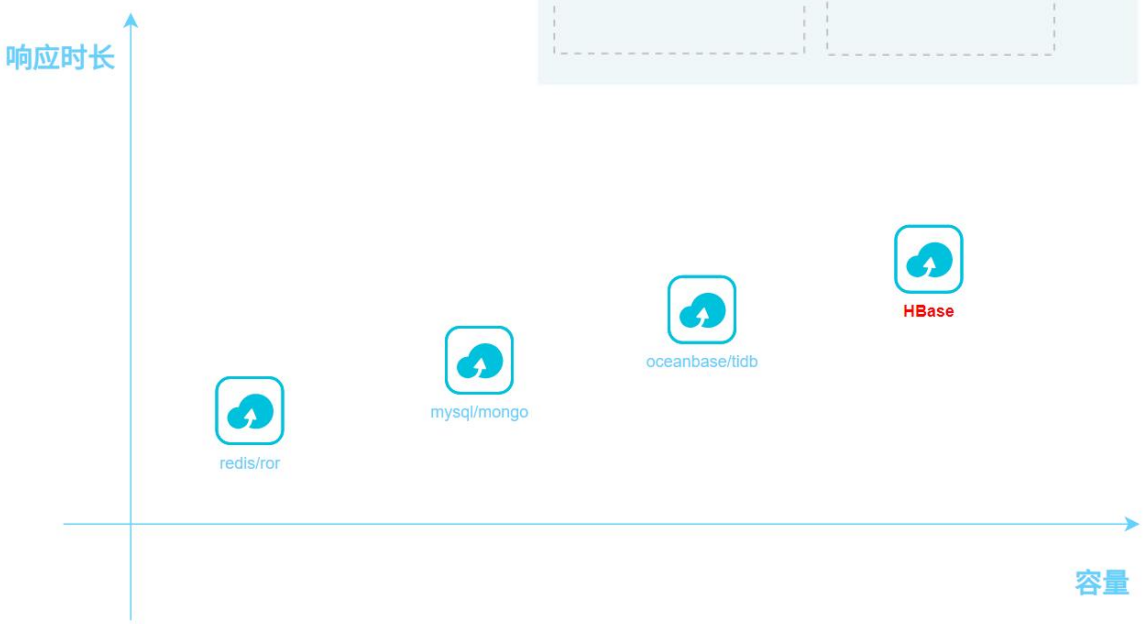
北京国际会议中心 | 2023/8/16-18

携程HBase混合云体系的建设与应用

携程数据库专家　吴宙旭

HBase@Trip





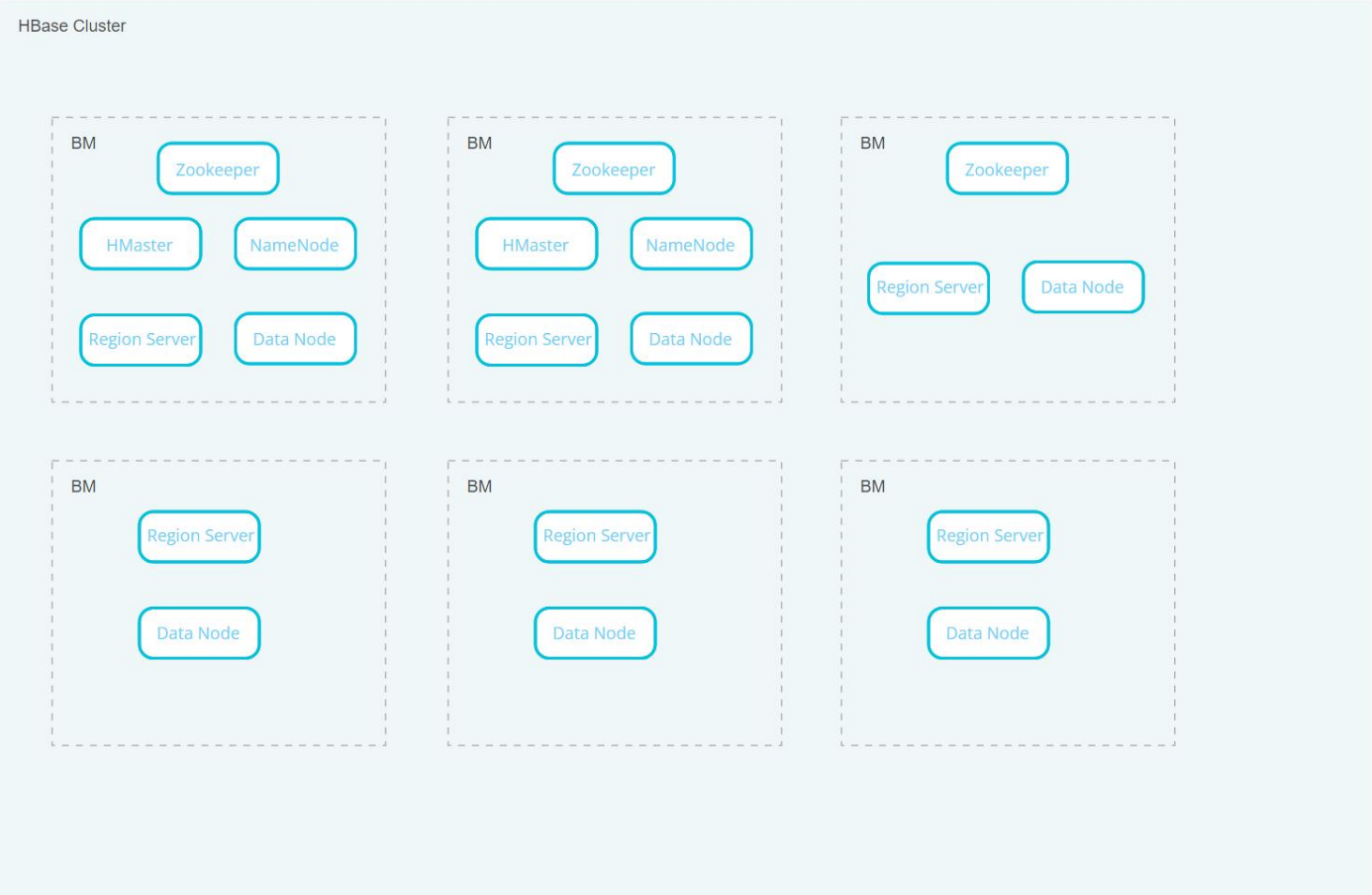
携程数据库架构：

- 多Region下多AZ：
  - ShangHai（IDC、ALi）
- 跨AZ级的高可用架构
- 数据库存储分级：
  - Redis、Redis on RocksDB
  - MySQL
  - OceanBase、TiDB
  - HBase

HBase定位：

- 响应时间略长
- 存储容量大
- 成本低廉
- 风控、酒店、机票、度假、营销等产品线

面临的一些挑战



挑战：

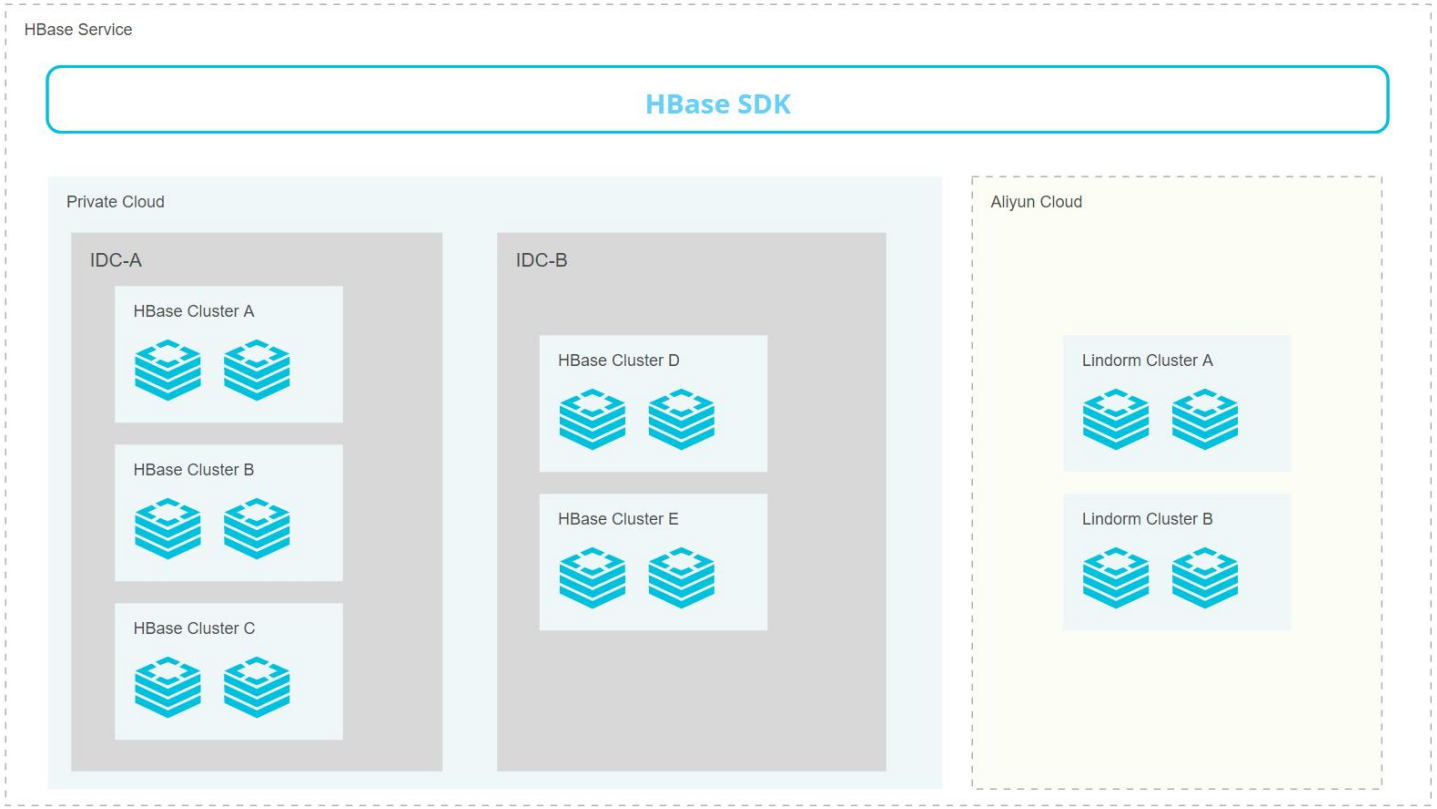
- **物理机带来计算、存储的利用率问题**
  - **单核管理的存储空间存在上限**
  - **计算密集型存储空间浪费**
- **业务季节性影响**
  - **应对五一、国庆、寒暑假等流量溢出场景**
- **服务的高可用性及Set化**

公有云选型：

- **阿里的Lindorm**
  - **开发生态，宽表对HBase支持较好**
  - **低成本兼顾性能（性能型、冷热、纯冷）**
  - **混合云网络延迟毫秒级别**

2022年引入混合云部署架构

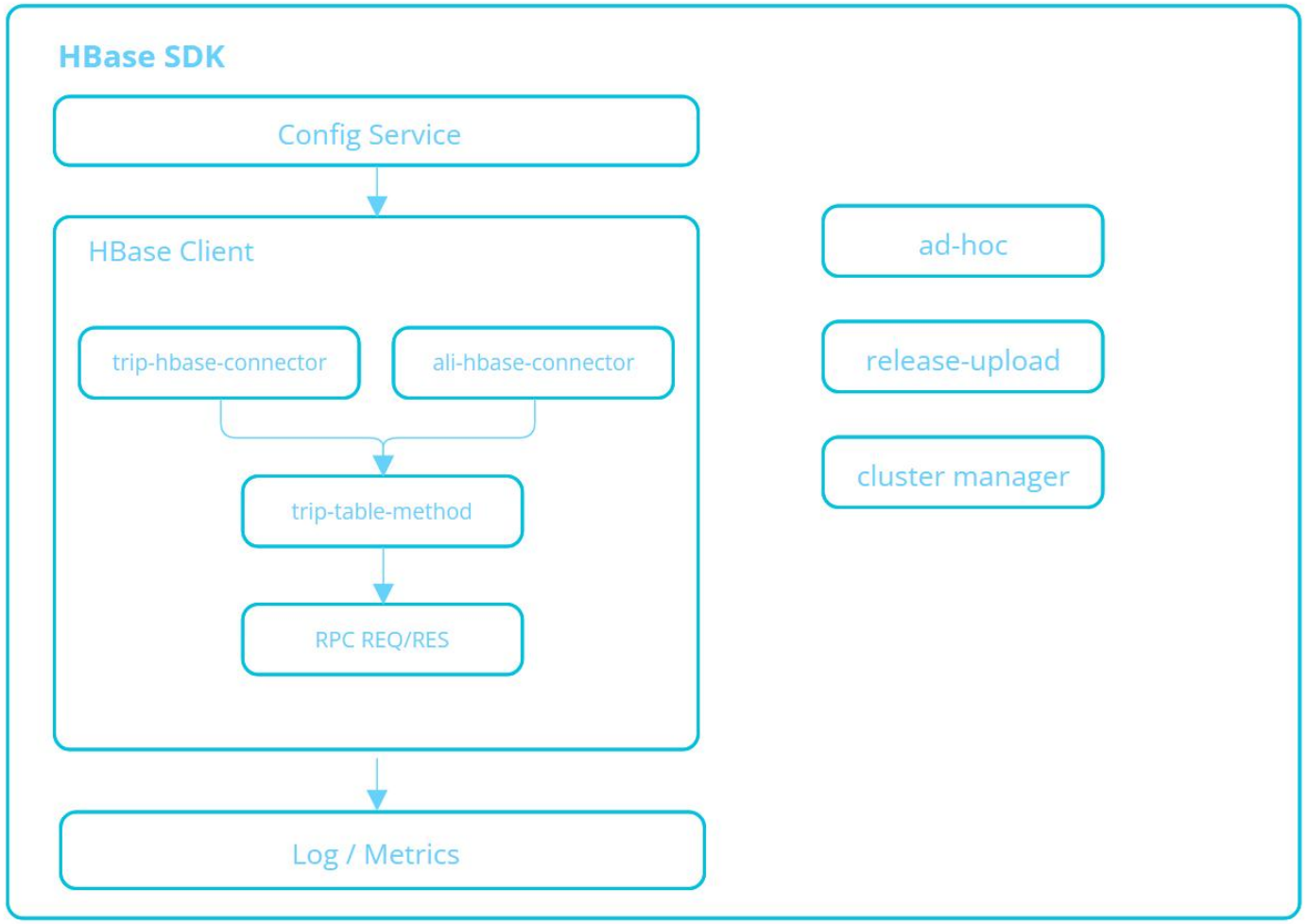
HBase 服务混合云部署推动Set化建设



混和云部署优势：

- **降低服务OPEX：**
  - 算、存利用率最大化（按需扩容计算节点和存储空间）
  - 冷、热存储分离（按数据读流量来确定冷热边界）
  - 应用改造成本低
- **提升服务可靠性：**
  - AZ级别故障（切换读写路由实现）
  - 变更
  - 故障处理更加从容
- **更加灵活的服务架构：**
  - 表级别的同步（表为单位同步数据）
  - 云溢出架构（多写单读）
  - 双读单写架构（单写多读）

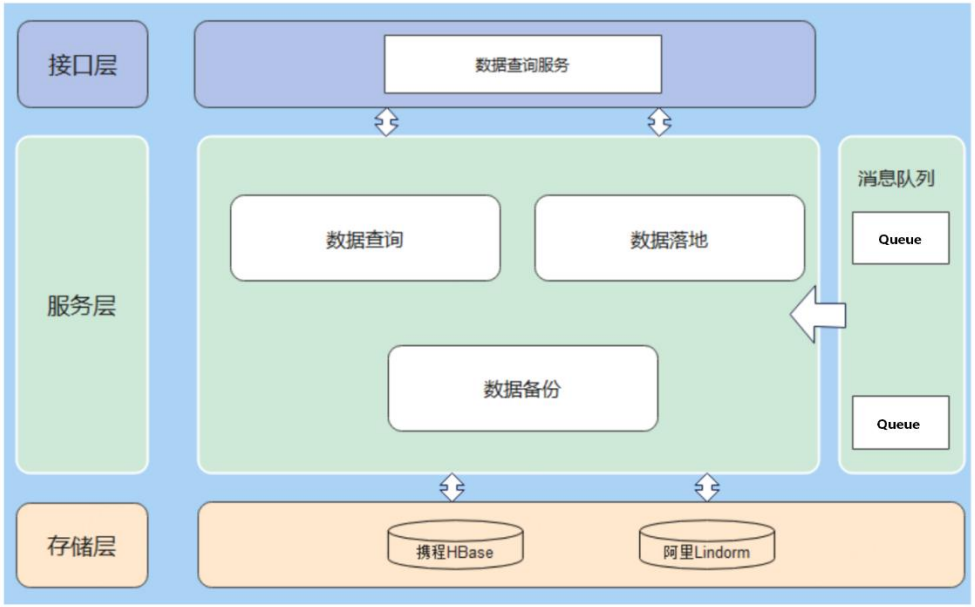
HBase 服务SDK

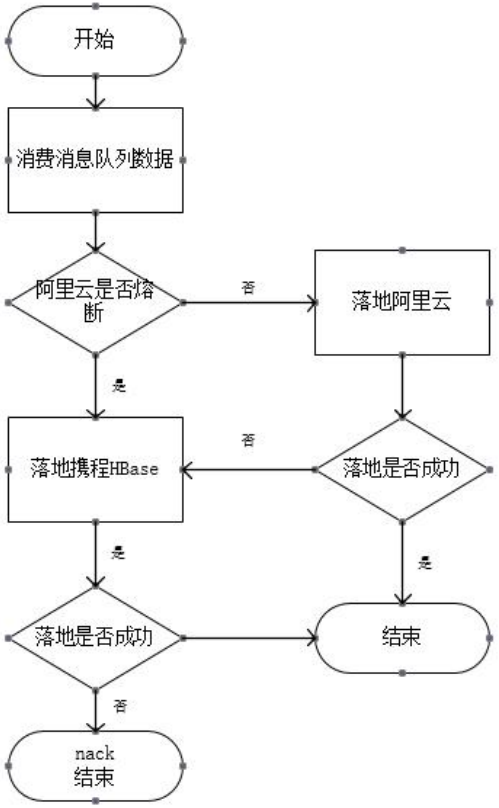


中间件支持：

- **ctrip-hbase-client：**
  - 公共配置组件可动态获取集群连接串
  - 具备单/多读写多个集群路由能力
  - 兼容社区及阿里端访问协议
  - 客户端Log、Metric埋点
  - 管理功能移除，下放到工具集完成
- **管理员工具：**
  - ad-hoc 即席查询，数据校验及抽取用
  - 上传发布系统（表的提交、审核、发布、修改）
  - 扩缩容工具（hbase、hadoop、lindorm）

双读单写架构





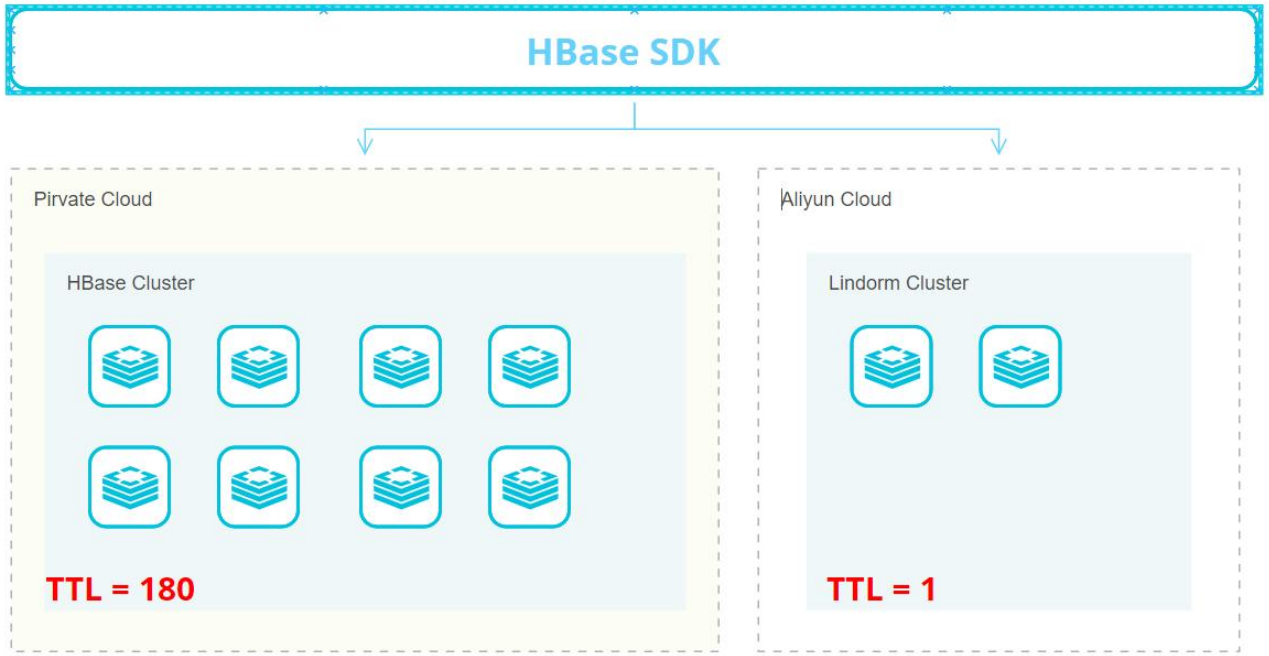
双读写架构：

- **数据读写：**
  - **主写阿里云Lindorm**
  - **Lindorm故障后自动切换写携程HBase**
  - **熔断：io.github.resilience4j.circuitbreaker**
  - **并发查询：云端和本地**
  - **结果汇总后返回请求（熔断时，断开故障节点）**

架构优势及局限：

- **多机房部署、服务高可用**
- **数据存储无冗余**
- **云端数据冷热分离，成本大幅降低**
- **熔断的实现有一定研发成本**

云溢出架构



云溢出架构：

- 数据读写：

  双写携程HBase+阿里Lindorm

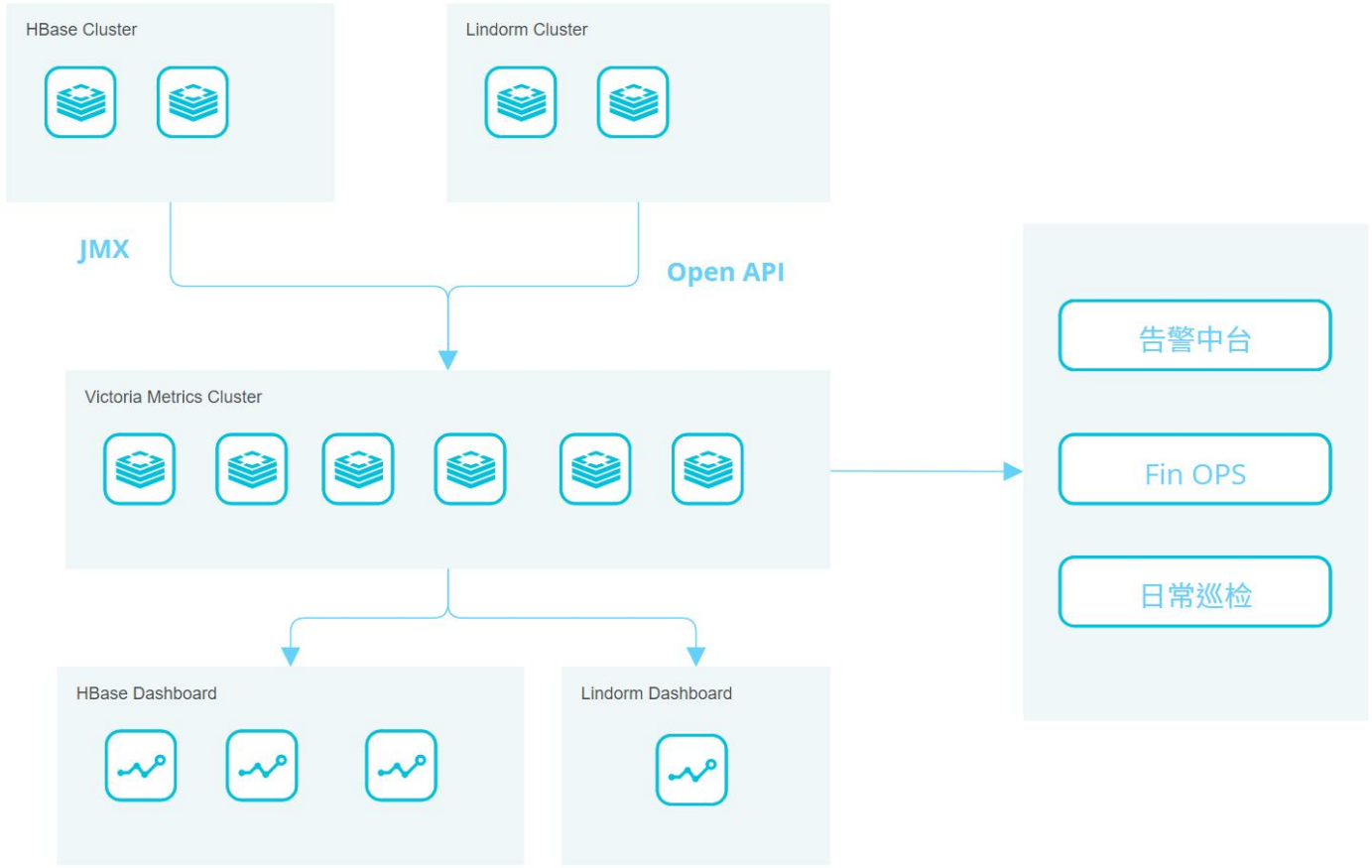
  主读携程

  单边故障后切换读写，故障端写数据缓存在消息队列中

架构优势及局限：

- 多机房部署，服务高可用
- 数据存储少量冗余
- 研发代码改造量少
- 用户95%以上查询集中在就近一天

HBase混合云服务感知体系



监控捕获：

- **服务端性能数据采集：**

  HBase通过JMX采集HBase及Hadoop（集群维度、region server维度、表维度）

  Lindorm采用OpenAPI，31个关键metric

  数据统一落地时序数据库VM

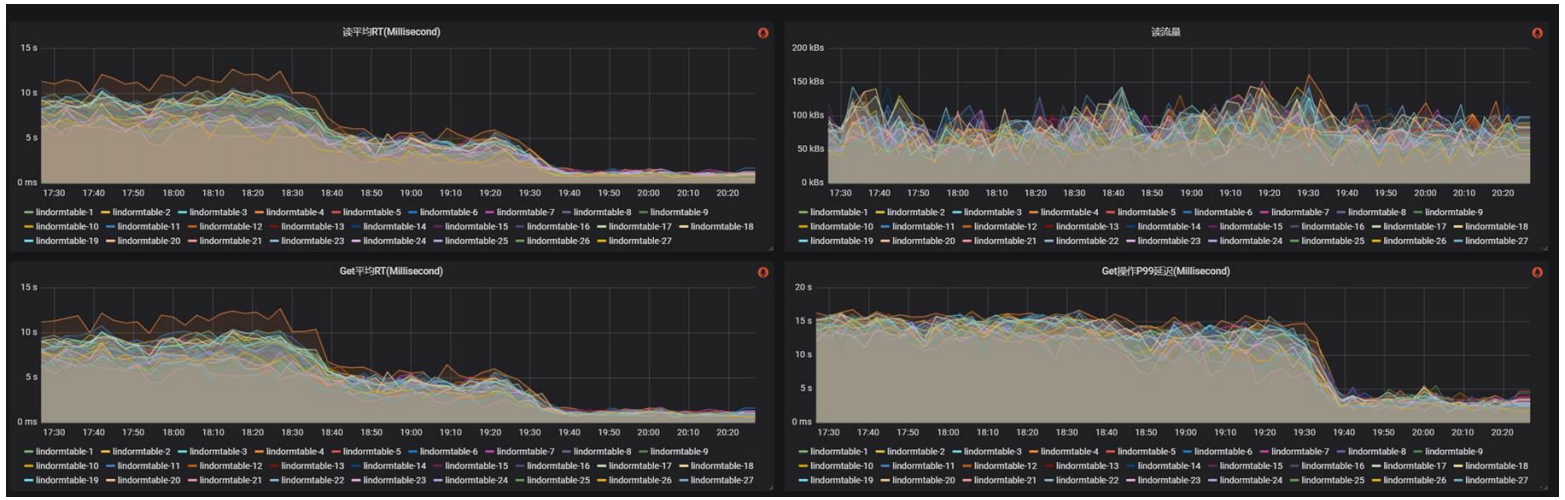
  自定义Grafana模板

  用户通过Dashboard查看

告警推送：

- **Critical：告警中台发送（mail、tts）**
  算力、存储：自动扩缩
  宕机、不可用：人工介入
- **Info、Warning：通过巡检发送**
- **Fin Ops：实时计算用量成本**

Hbase混合云优化案例



Get P99耗时: 10s - > 1s

Get优化：

- 冷热分离存储：

  Get请求的属性中设置hot_only标签

- 纯冷、纯热存储：

  Get请求中Key+timerange

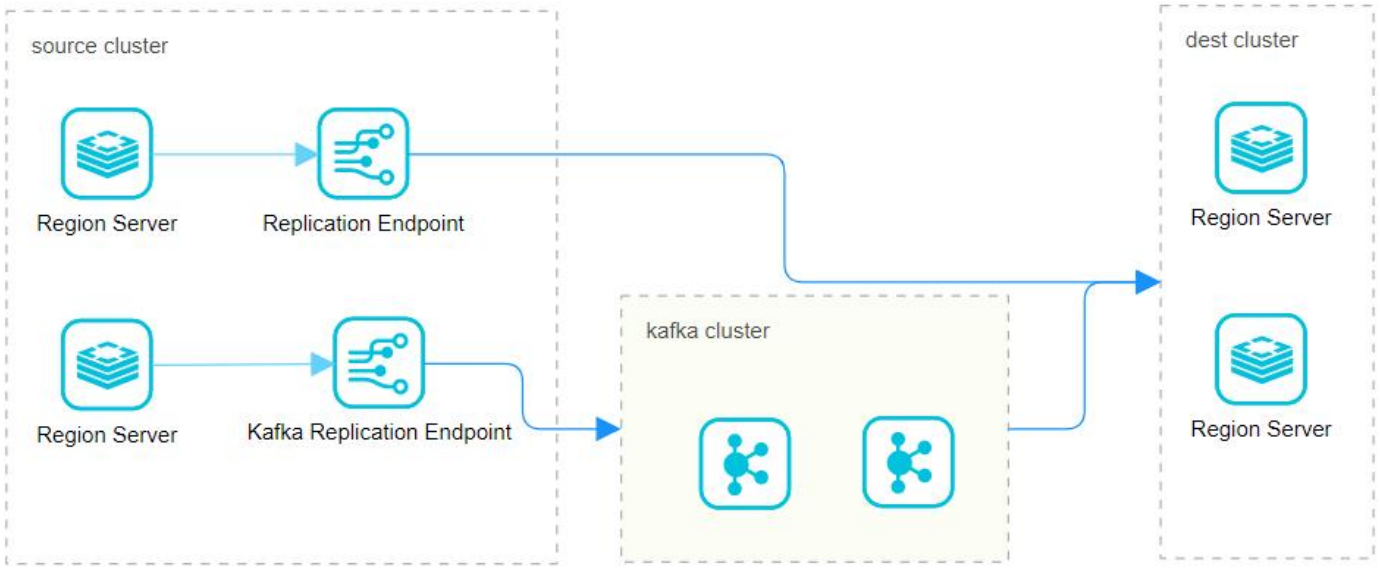
Scan优化：

- 大结果集的scan：

  高版本客户端上1.4+，scan.limit

  低版本客户端上，构造ResultScanner的迭代器，并且设置limit数据量后做迭代查询

规划-基于WAL的数据同步



基于Wal的数据同步

- HBase原生Replication
  - Export Snapshot + Replication
- 自定义Replication：
  - 改造基于Kafka的 Replcation Endpoint
  - WAL Log落地Kafka
  - 自定义消费到Dest Cluster
- LTS：
  - 构建回环链路

THANKS