PULSAR Meetup | Pulsar Meetup 深圳 2024

Apache Pulsar 腾讯大数据场景替换 Kafka 实践

分享人：陈洪

目录

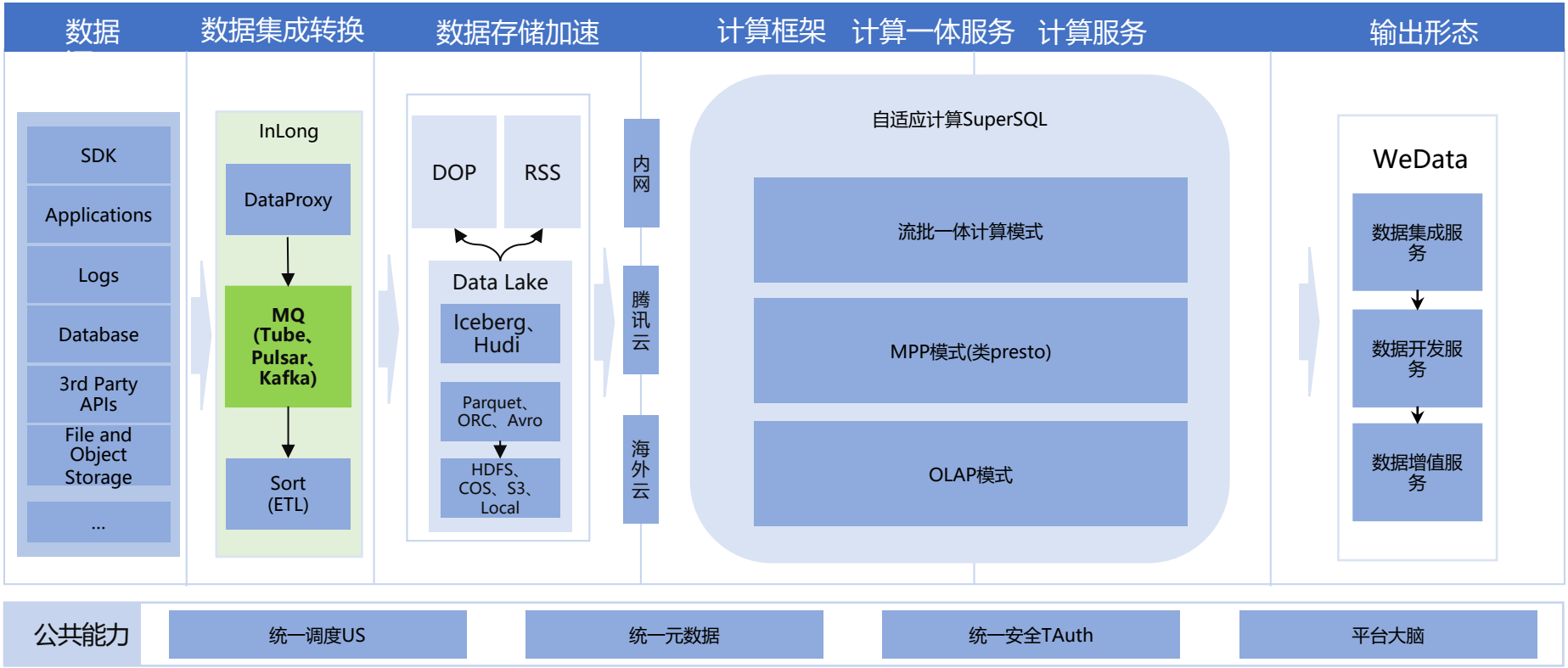
1 腾讯大数据 MQ 概况

2 使用 Kafka 遇到的痛点

3 Pulsar 带来的优势与不足

4 Pulsar 未来规划

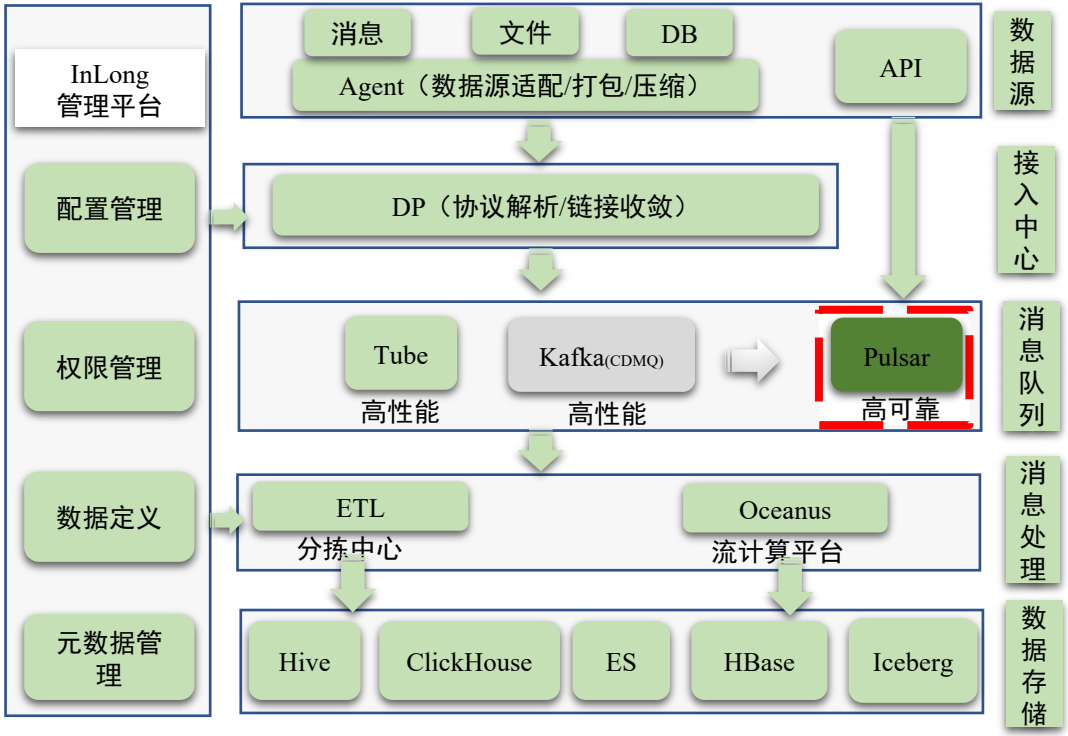
腾讯大数据平台


数据
数据集成转换
数据存储加速
计算框架
计算一体服务
计算服务
输出形态
SDK
Applications
Logs
Database
3rd Party APIs
File and Object Storage
...
InLong
DataProxy
MQ (Tube、Pulsar、Kafka)
Sort (ETL)
DOP
RSS
Data Lake
Iceberg、Hudi
Parquet、ORC、Avro
HDFS、COS、S3、Local
内网
腾讯云
海外云
自适应计算SuperSQL
流批一体计算模式
MPP模式(类presto)
OLAP模式
WeData
数据集成服务
数据开发服务
数据增值服务
公共能力
统一调度US
统一元数据
统一安全TAuth
平台大脑

腾讯大数据平台--InLong

Apache InLong（应龙）是一个一站式、全场景海量数据集成框架，同时支持数据接入、数据同步和数据订阅，提供自动、安全、可靠和高性能的数据传输能力，同时支持批和流，方便业务构建基于流式的数据分析、建模和应用。

- 使用 MQ 作为内部存储
  - ✓ Tube 单副本、极致性能
  - ✓ Kafka 高吞吐
  - ✓ Pulsar 高可靠、高性能



大数据 MQ 当前运营现状

十万亿/天 消息接入

- ✓ 大数据场景
- ✓ 中间件场景

MQ	条数/天	大小/天
Tube	百万亿	十 PB级
Kafka(CDMQ，停新增)	十万亿↓	PB 级
Pulsar	十万亿↑	PB 级

支撑多个核心业务

腾讯广告 Tencent Marketing Solution

财付通 TENPAY.COM 会支付 会生活

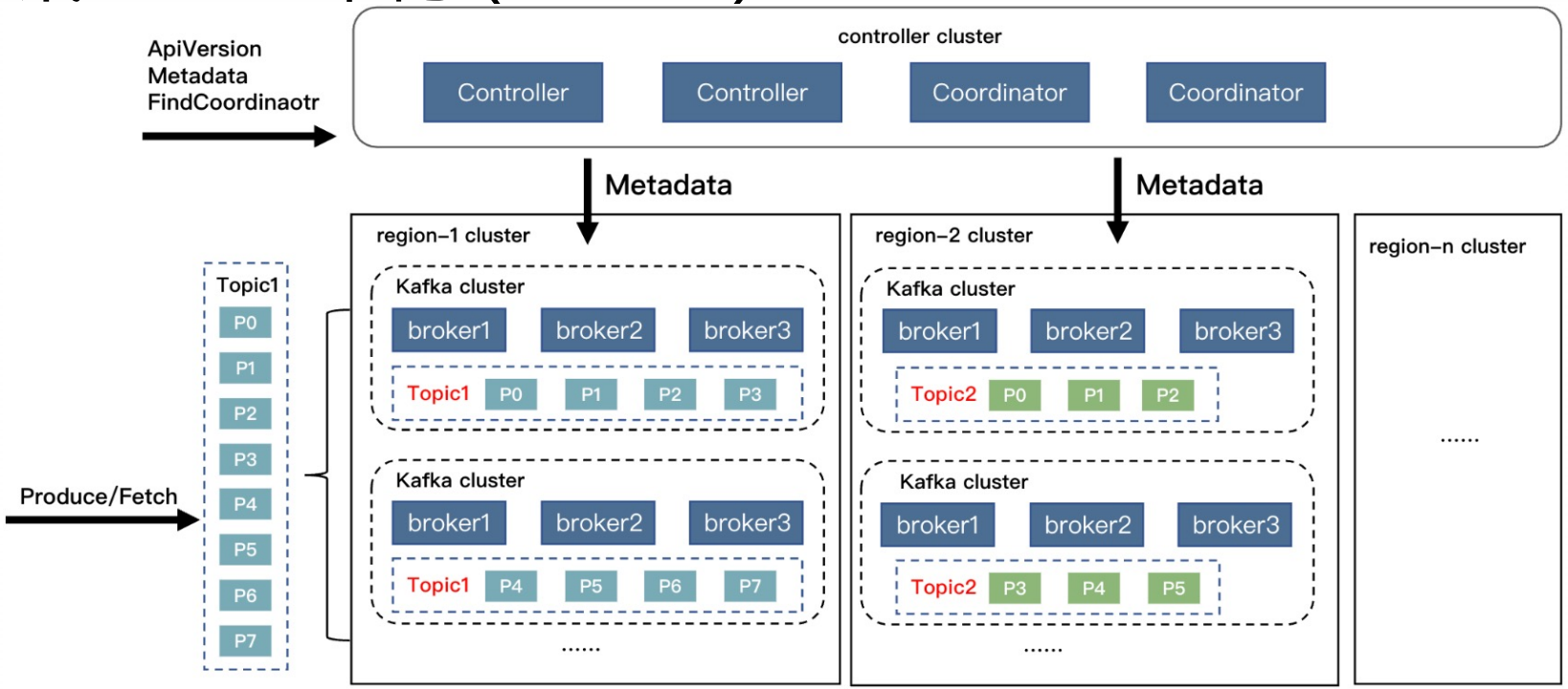
微信支付

腾讯视频 不负好时光

米大师 腾讯移动支付Midas

腾讯游戏 Tencent Games

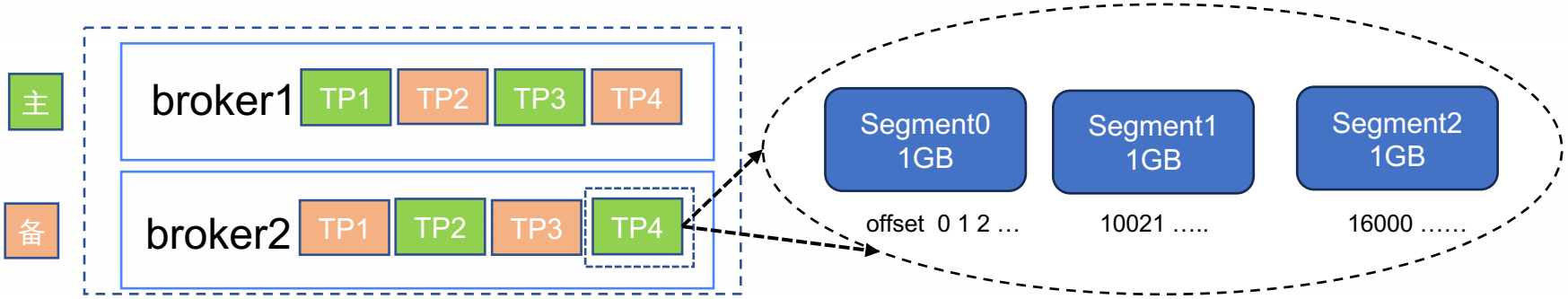
腾讯 Kafka 架构 (CDMQ)



- ✓ 联邦模式
- ✓ 子集群粒度资源化 & 标准化
- ✓ 分区级别禁读写
- ✓ 快速扩缩容
- ✓ 快速故障屏蔽

Kafka 痛点1-Topic 有上限

- 存算不分离，存储绑定副本节点
- 分区单独文件 顺序读写IO
- 按 Segement 维度过期数据



Partition 增多，append 写退化为随机写, 性能下降 (单磁盘上分区64→256,吞吐量下降60%)

InLong 使用Kafka 时需要合并数据流写入到相同Topic(多 topic 但量小)

Kafka 痛点2-存储压力

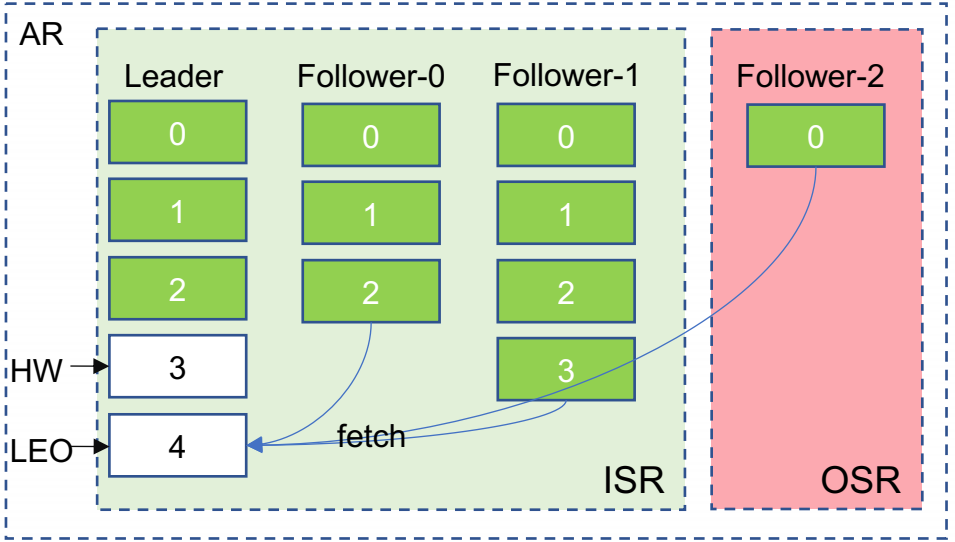
◘ 保证磁盘均匀

- 磁盘 **Raid** 化
- 分区数限制为节点数**倍数**(加剧分区消耗)

机型	Topic数	Partitions	Raid
TS60(12*SATA) *3	600	1800	RAID10
SH2(4*SSD) *3	2000	6000	RAID0

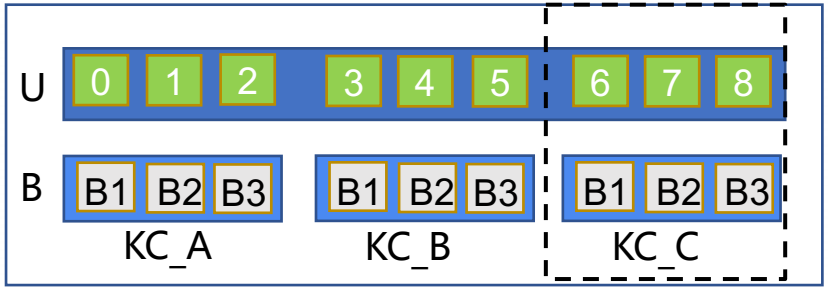
内部一致性协议需要3副本保证可靠性及可用性

acks = all
高可靠: min.insync.replicas > 1
高可用: min.insync.replicas < replication.factor

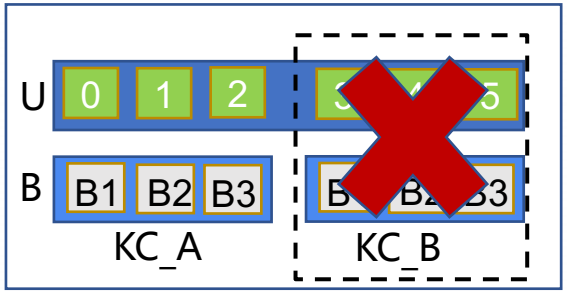


Kafka 痛点2-扩缩容-禁读写

扩容



缩容 / 禁读写



为什么不做 reassign？

- 数据迁移耗时长、不能救急
- 额外流量影响业务
- 分区增加低版本 SDK 不感知

缩容如何保证数据不丢？

- ✓ 生产消费元数据**解耦**，**读写状态**可单独设置
- ✓ 先**禁写**，数据**过期**后再禁读、回收分区
- ✓ 缩容无需做分区数据迁移，过程可控，稳定
- 禁写后 按 Key 写入模式会阻塞

禁读写、扩缩容会导致分区变化，无法做到全场景对用户透明

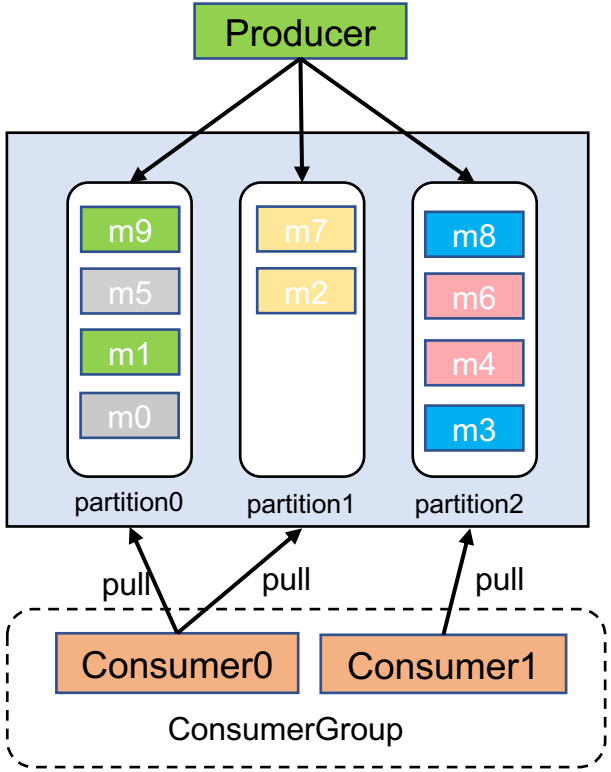
Kafka 消费-分区数限制消费

局部有序

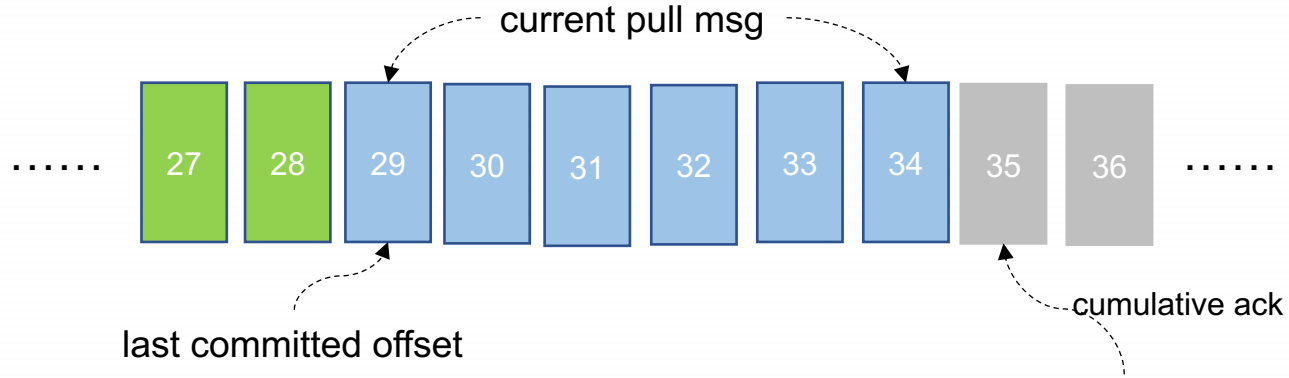
- 生产指定分区或者按Key Hash
- 消费单个分区只能分给单个消费者

分区数决定消费吞吐

- 扩容分区数增加消费吞吐量(加剧分区消耗，Kafka 单 topic 分区数超过1w)
- 分区数或者消费节点数变化需 **Rebalance** (异常消费也会导致)
- Pull 模式由客户端主导，服务端指标少，消费问题难排查



Kafka 消费-仅支持流模式消费

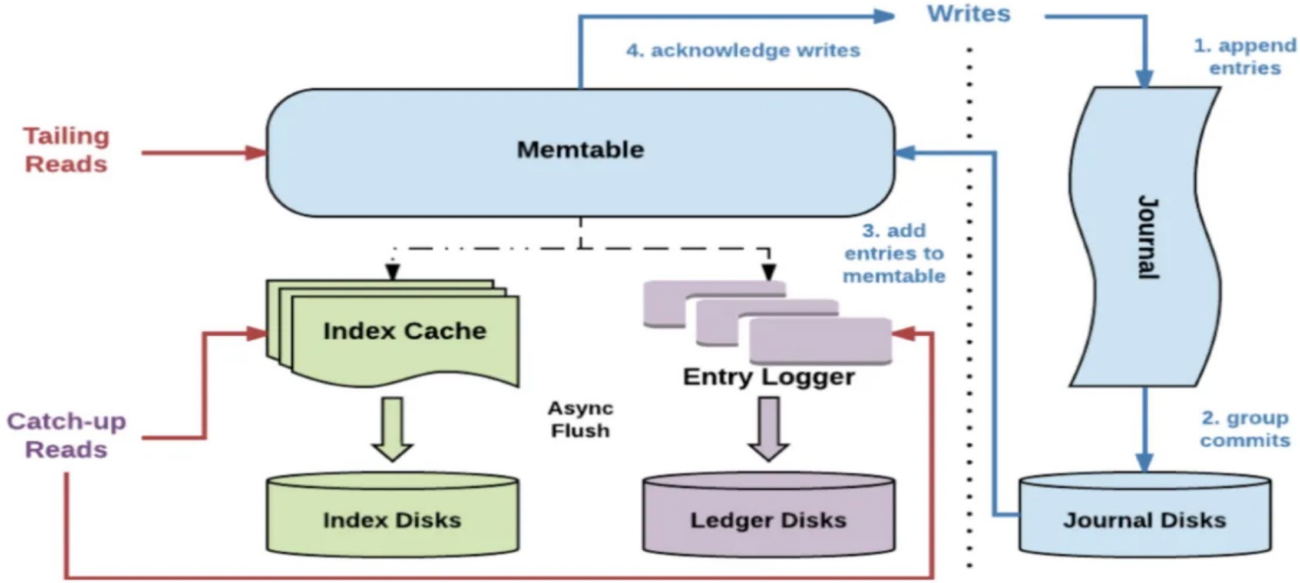


仅支持流模式消费

- 只能顺序消费(不能跳过)
- 只支持cumulative ack
- 不支持重试、死信队列
- InLong 内部异步处理消息并ack 最小连续 offset，节点迁移后重复率较高

spring-kafka 2.7.x 在客户端通过辅助 Topic 实现**重试、死信**队列

Pulsar 带来的优势-Topic 数量上限高



顺序写、随机读

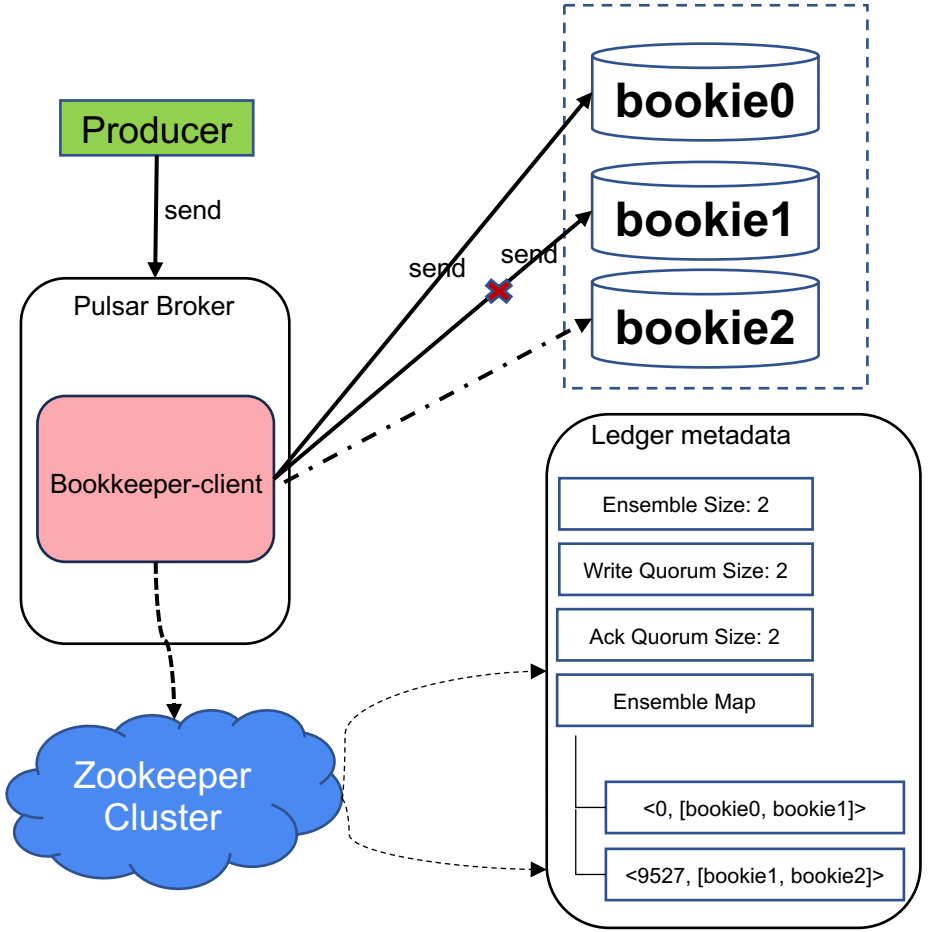
- ✓ 多个Topic 数据写入单个EntryLog
- ✓ 保证**顺序**写

- ✓ InLong 单个数据流独占 Topic
  优化掉 cmatser、pmaster 组件
- ➢ 随机读取
- ➢ 数据过期需要做 compaction

Pulsar 外部一致性协议

外部一致性协议

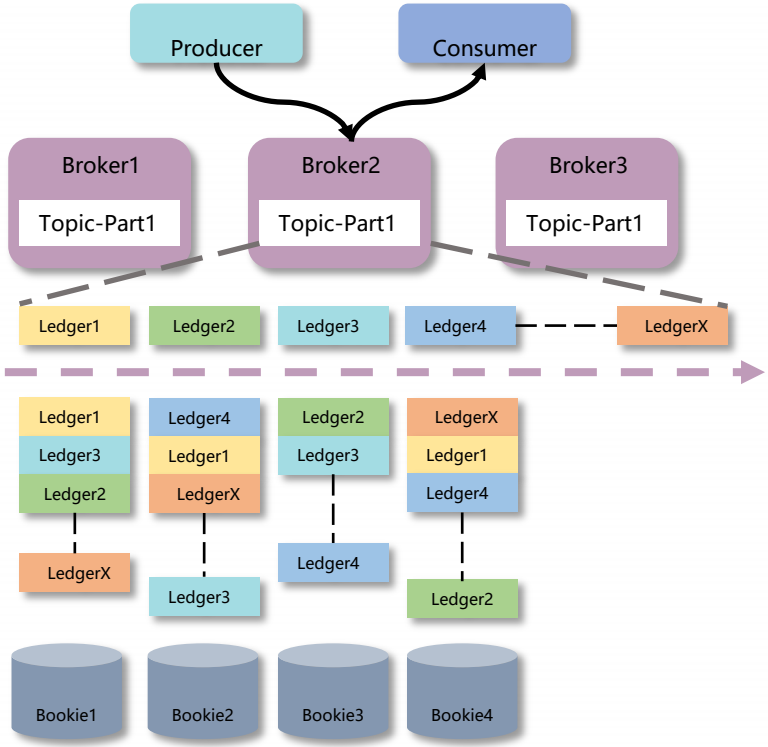
- ✓ **两副本**即可保证**高可用&高可靠**
- ✓ 失败触发 `EnsembleChange` 写入新节点
- ✓ 最终失败后会创建新 ledger 继续写入
- ✓ 写入总体耗时没有增加(2 RTT)
- ✓ Qw > Qa 可降低时延



Pulsar 存储天然均衡

存算分离(得益于外部一致性协议)

- ✓ **逻辑**分区、物理分片 (Ledger) 隔离
- ✓ 分区数可任意指定
- ✓ 单分区容量**无上限**，均衡使用到全部磁盘



Pulsar 快速扩缩容 & 高可用

由于使用外部一致性协议， Pulsar 与 BK 节点迁移和变更可以很轻量完成，大幅度降低运营成本。

快速扩容

- ✓ Broker – 无状态-leadership 切换轻量
- ✓ Bookie 扩容无需数据迁移-新节点立即使用上
- ✓ 扩容分区数无变化，对用户几乎透明

缩容 & 节点故障

- ✓ Broker – 无状态-leadership 切换轻量
- ✓ Bookie 设置 `readonly` `decommissionbookie`
- ✓ 缩容分区数无变化，对用户几乎透明

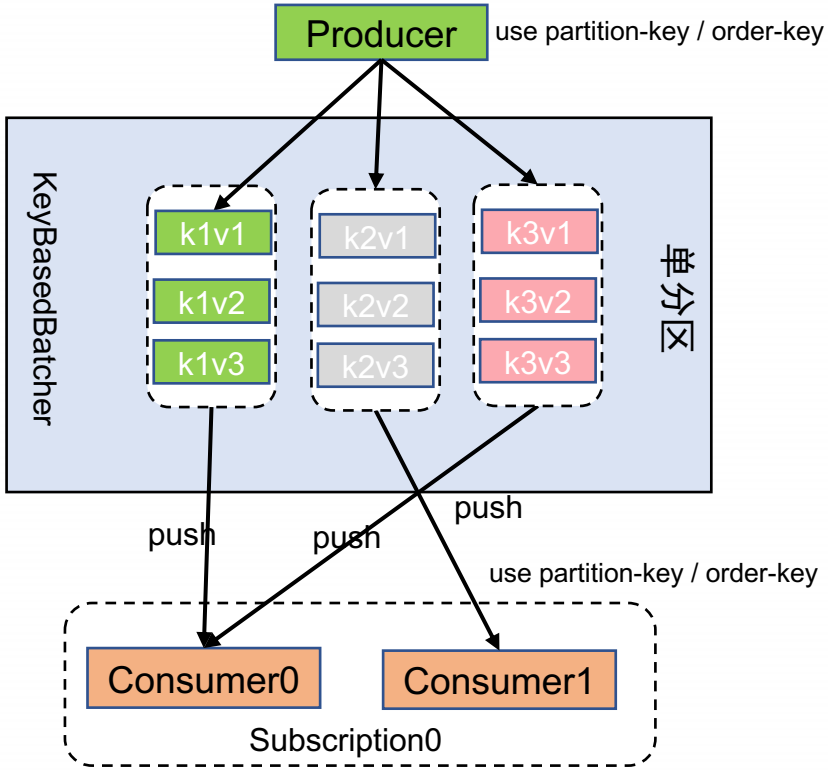
Pulsar 消费

细粒度消息顺序保障

- ✓ **单分区**可以被**多消费者**同时消费(解决消费性能受限于分区数)
- ✓ 消费可配置 **order-key** 实现定制化推送
- ✓ Push 模式服务端指标丰富，便于定位消费问题

支持队列模式

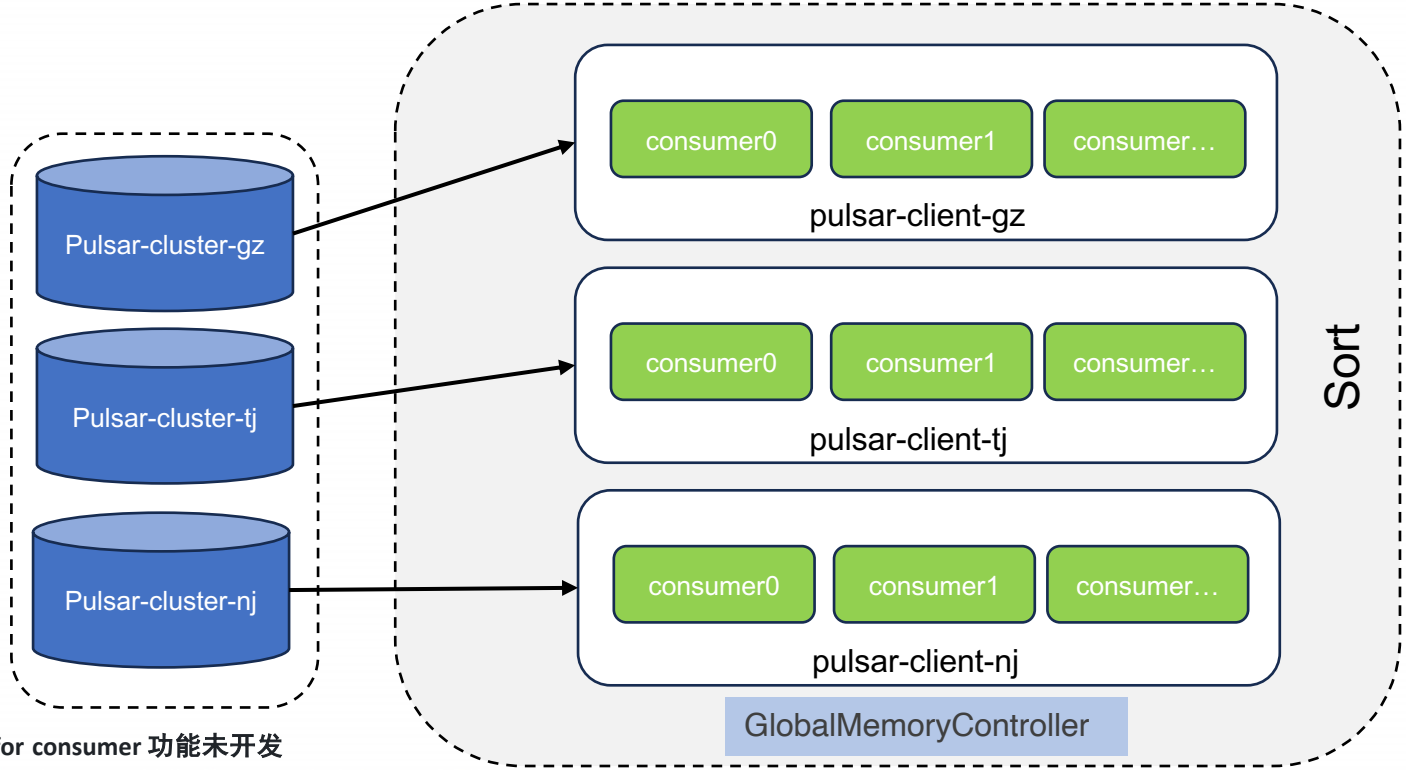
- ✓ individual ack / Cumulative ack
- ✓ 支持丰富的重试机制和死信队列



InLong Pulsar消费-进程级别内存限制

全局消费内存控制

- 内存使用与**分区数**正相关
  分区数不可控
- **内存**敏感
- 单使用` receiverQueueSize`
  达不到效果



当时 PIP 74: Pulsar client memory limits for consumer 功能未开发

简单总结

Pulsar

外部一致性协议

- ✓ 存算分离
- ✓ 磁盘天然均衡，不存在热点
- ✓ 运营成本低

支持队列/流模式

- ✓ individual ack / Cumulative ack
- ✓ 支持丰富的重试机制和死信队列

Kafka

内部一致性协议

- ➢ 存储强绑定计算
- ➢ 磁盘需谨慎使用才能保持均衡
- ➢ 单分区存储有上限
- ➢ 运营成本高

仅支持流模式

- ➢ 用户使用成本高

性能

- ✓ 顺序读写可达到极致吞吐量(分区数可控)
- ✓ Catch-up read 对集群影响较小

Pulsar 带来的新挑战

组件

- 强依赖Zookeeper，严格保障 Zookeeper 稳定

性能

- 大量Catch-up read 场景，影响生产(目前在试用 `directIO` 与 `BatchRead` 特性)

未来规划

- 支撑更多业务、更大数据量
- 优化Pulsar –Manger，提高易用性，降低使用和运维成本
- 继续优化滞后场景性能

Thanks