

数据来源：数据库产品上市商用时间



第十三届中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2022

数据智能 价值创新



线上直播 | 2022/12/14-16



云音乐实时数仓建设以及 任务治理实践

汪磊+网易云音乐+数据平台开发专家

- 云音乐实时相关业务现状和数据规模
- 分区流表技术介绍
- 数据任务治理实践
- 未来规划

音乐相关业务现状和规模



用户量

300+用户，覆盖数仓、数据产品、算法、分析师、QA、应用开发，服务音乐主站、心遇、直播等音乐所有业务线



业务类型

覆盖数据仓库建设、数据报表开发、线上排行榜积分统计、索引构建、实时特征/样本计算



任务量

1500+实时任务、80%+任务依赖实时数仓使用SQL开发

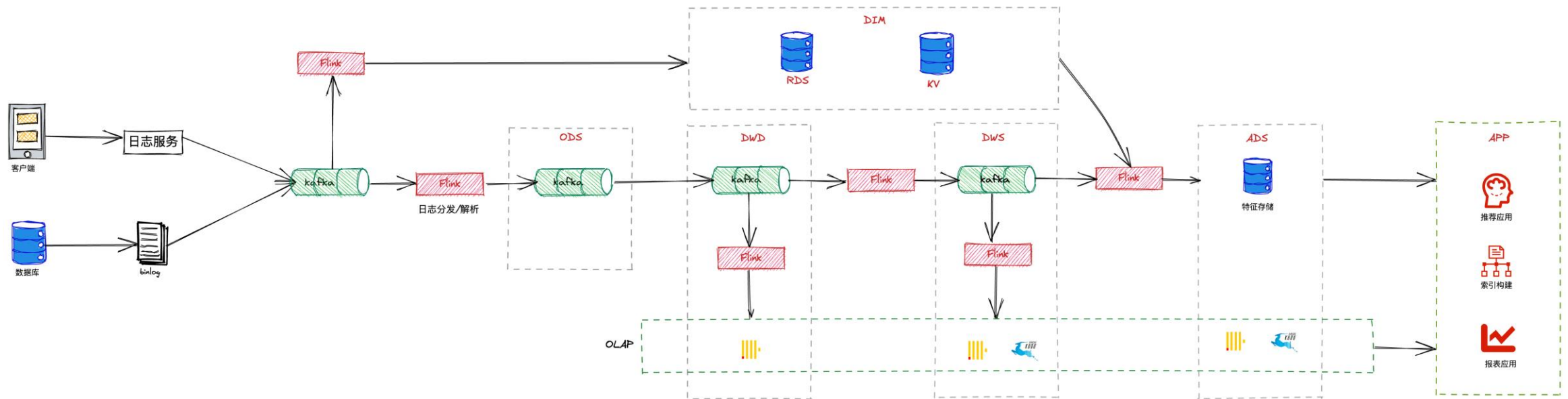


数据规模

实时集群：400+台物理机器；每日原始日志千亿级别

音乐相关业务现状和规模

整体架构:



音乐相关业务现状和规模

问题:

- 开发门槛高: FLINK本身复杂度问题、存储中间件众多、用户背景问题
- 模型设计问题: 按照离线数仓的构建数仓模型, 模型设计和任务性能需要权衡
- 任务运维问题:
 - 任务指标难以定制, 用户理解门槛高;
 - 流量波动问题、外部存储问题, 机器环境问题, 定位问题困难
 - 任务状态问题
- 任务治理问题:
 - 离职员工任务问题
 - 任务价值闭环问题
 - 资源配置和性能问题

分区流表技术建设

模型设计问题：按照离线数仓的思想构建数仓模型，模型设计和任务稳定性之间需要权衡

离线：

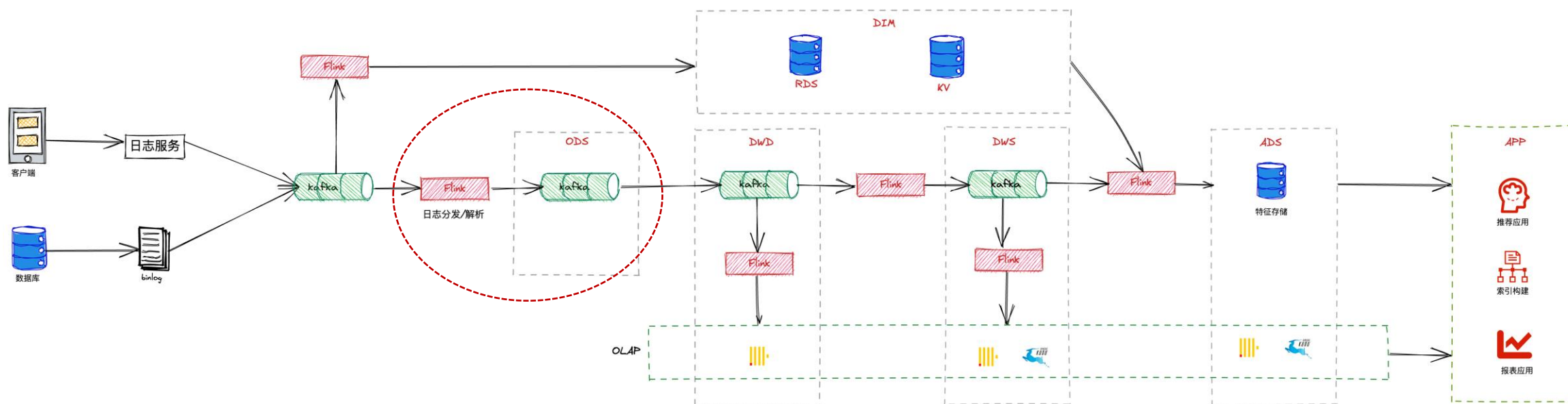
- 相比实时场景对数据量、资源消耗敏感度较低
- 优化手段多：分区、分桶、数据有序写入等可以高效的过滤掉无效的数据
- 存储稳定性：HDFS等落地存储，数据量的增长对齐稳定性的影响相对较小

实时：

- 对任务延迟非常敏感，数据量任务的稳定影响较大，从而导致不任务SLA达不到业务需求
- 优化手段少：以KAFKA为基础的实时流表几乎没有什么过滤数据的高效策略
- 存储稳定性：数据量的增长对KAFKA的稳定性影响较大，多一个消费，KAFKA性能/网络带宽就多一份压力

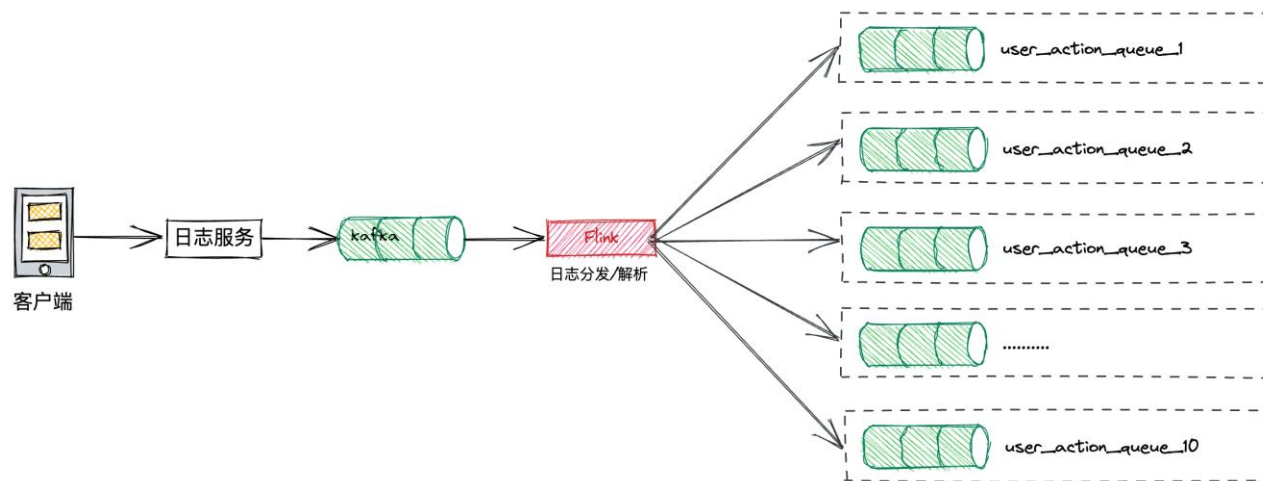
分区流表技术建设

整体架构:



分区流表技术建设

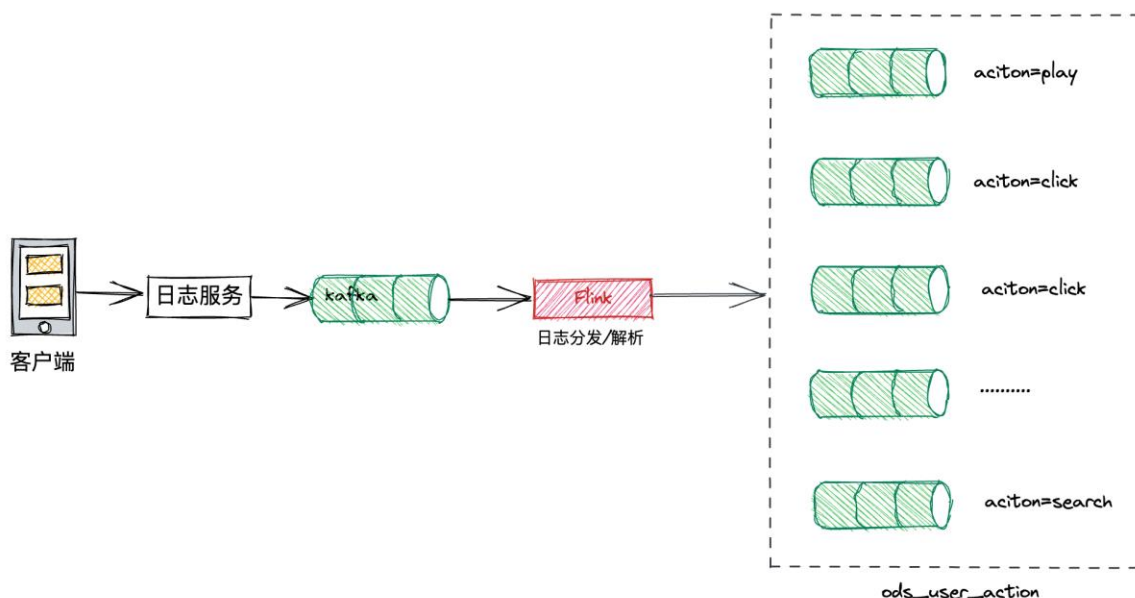
初期架构:



- 使用门槛高：用户需要记住每个表包含的日志内容
- 模型不统一：因为性能，放弃模型的统一
- 复用成本高：开发代价大，复用程度低
- 运维调整困难：分发策略调整代价大

分区流表技术建设

分区流表:



- 使用门槛低: 和HIVE分区表一样, 用户只需要关注业务分区字段即可
- 模型统一: 和离线模型保持一致, 使用分区策略, 优化流量和计算成本
- 复用成本低: 和HIVE分区表一样, 用户配置分区字段的元数据, 直接INSERT即可
- 运维难度低: 运行时动态新增或者建少分区策略

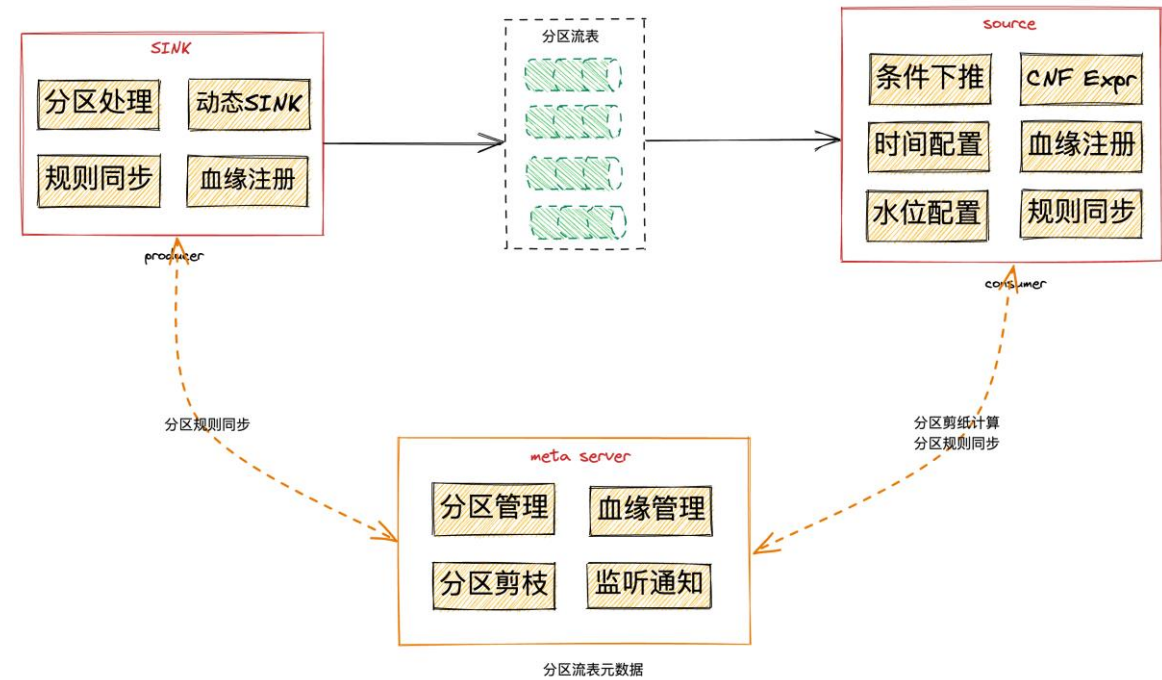
分区流表技术建设

产品实现:

列	类型	描述
p_action	string	按action字段内容分区
p_os	string	按os字段内容分区
p_sapp	string	按sapp字段内容分区
p_module	string	按选择其他字段内容分区

序号	p_action	p_os	p_sapp	p_module	topic	状态	分区数	创建时间	生效时间
1	impress	phone	music	recommend	UA_Paction_impress_Pos_phone_Ppage_recommend	生效	100	-	-
2	impress	other	music	recommend	UA_Paction_impress_Pos_pc_Ppage_recommend	生效	10	-	-
3	impress	phone	music	other	UA_Paction_impress_Pos_phone_Ppage_other	生效	100	-	-
4	impress	other	music	other	UA_Paction_impress_Pos_pc_Ppage_other	生效	20	-	-
5	impress	phone	music	search	UA_Paction_impress_Pos_phone_Ppage_search	生效	40	-	-
6	impress	other	music	search	UA_Paction_impress_Pos_pc_Ppage_search	生效	25	-	-
7	impress	phone	ksong	recommend	UA_Paction_default_Pos_default_Ppage_default_Psapp_ksong	生效	10	-	-
8	impress	other	ksong	recommend	UA_Paction_default_Pos_default_Ppage_default_Psapp_ksong	生效	10	-	-
9	impress	phone	ksong	other	UA_Paction_default_Pos_default_Ppage_default_Psapp_ksong	生效	10	-	-
10	impress	other	ksong	other	UA_Paction_default_Pos_default_Ppage_default_Psapp_ksong	生效	10	-	-

分区配置



技术方案

分区流表技术建设

效果:

- 易用性: 模型统一, 使用门槛大大降低, 覆盖音乐所有业务线
- 任务稳定性: 流量导致延迟问题得到彻底的解决
- 资源节省: 历史任务完成分区改造迁移总结节省成本 **700W+**, 单任务最高节省资源**90%**, 流量DOUBLE的情况下Kafka新增机器为 **0**

数据任务治理实践

现状

摸清现状，高效治理

清理

清理无用任务

优化

优化资源配置

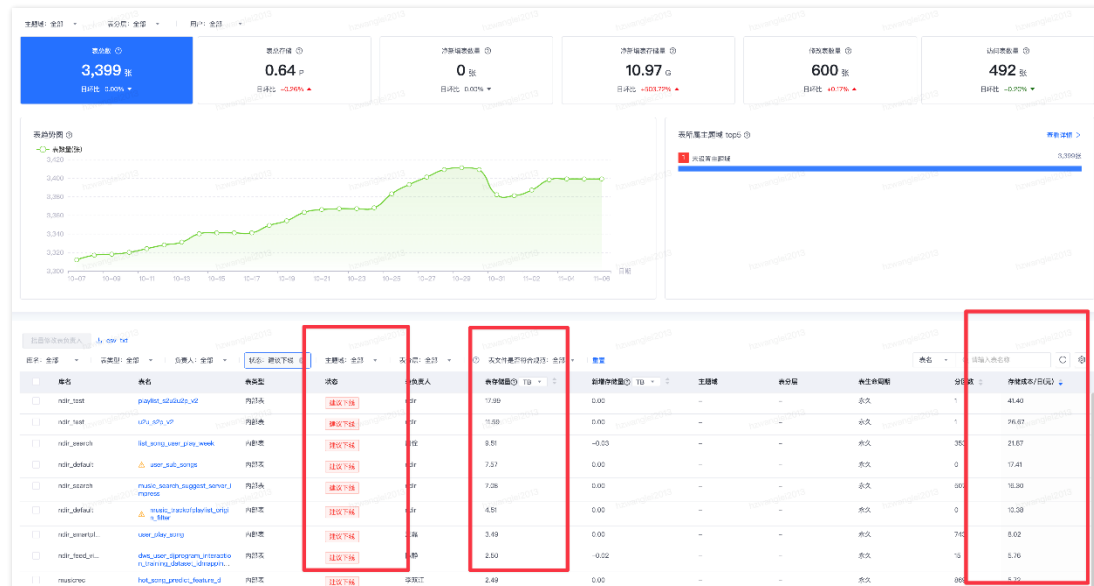
可持续

自动化，可持续

降本提效

数据任务治理实践

摸清现状：数据治理有的放矢



存储成本

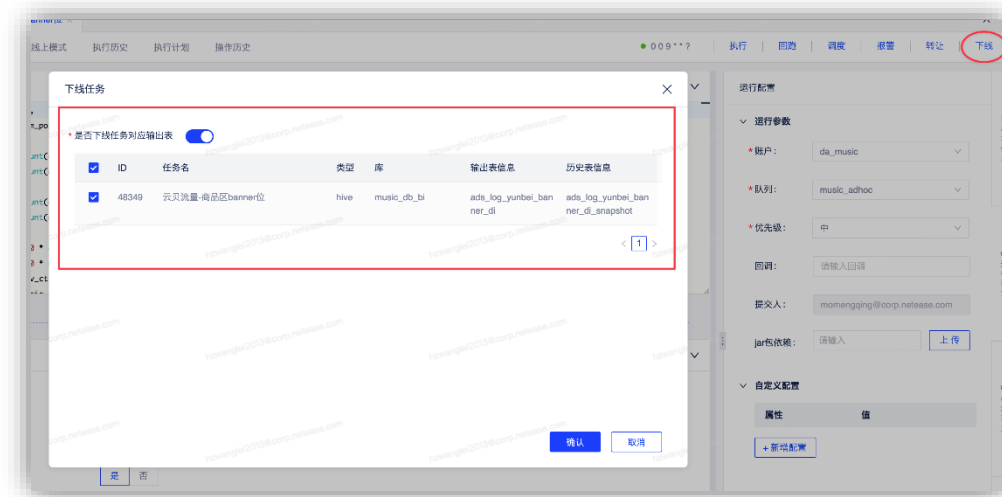
This table lists data assets with columns for '资产名称' (Asset Name), '资产类型' (Asset Type), '资产状态' (Asset Status), '资产负责人' (Asset Owner), '资产创建时间' (Asset Creation Time), '资产更新时间' (Asset Update Time), '资产使用频率' (Asset Usage Frequency), '资产使用范围' (Asset Usage Scope), '资产使用人数' (Asset Usage Count), and '资产使用时长' (Asset Usage Duration). A red box highlights the '资产使用频率' and '资产使用范围' columns.

This table lists data assets with columns for '资产名称' (Asset Name), '资产类型' (Asset Type), '资产状态' (Asset Status), '资产负责人' (Asset Owner), '资产创建时间' (Asset Creation Time), '资产更新时间' (Asset Update Time), '资产使用频率' (Asset Usage Frequency), '资产使用范围' (Asset Usage Scope), '资产使用人数' (Asset Usage Count), and '资产使用时长' (Asset Usage Duration). A red box highlights the '资产使用频率' and '资产使用范围' columns.

计算成本

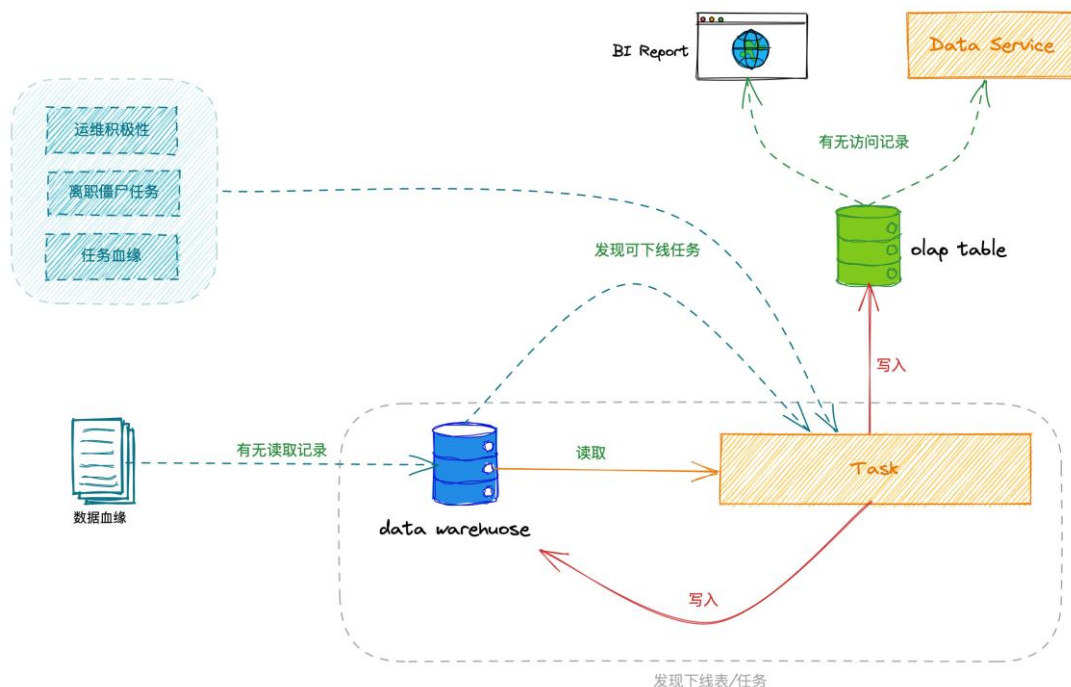
数据任务治理实践

下线：发现可下线任务，可下线表，执行下线操作



一键下线：表 + 任务

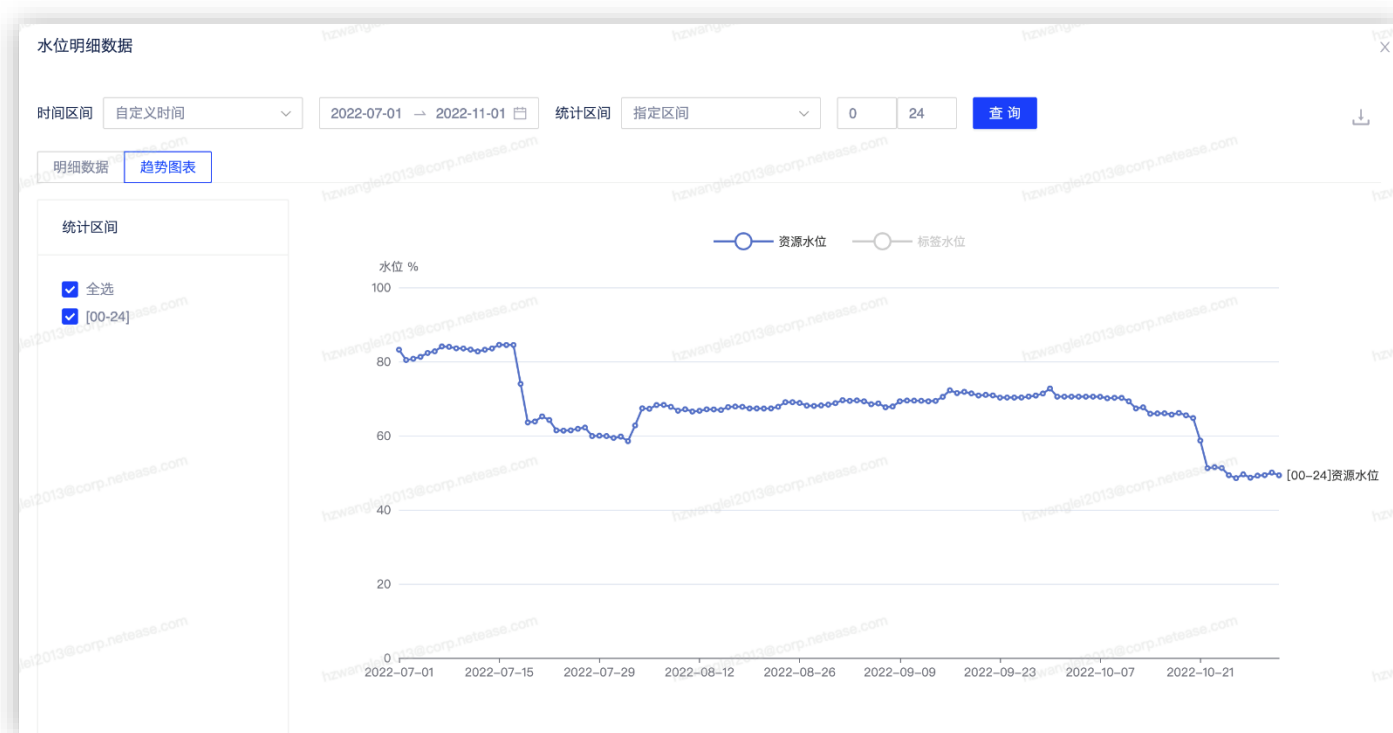
- 血缘：发现无用表 > 发现无用任务
- 运维积极性：报警处理积极性
- 僵尸任务：离职人员、转岗员工、长期不更新任务确认
- 任务血缘：活动下线、下游访问记录



数据任务治理实践

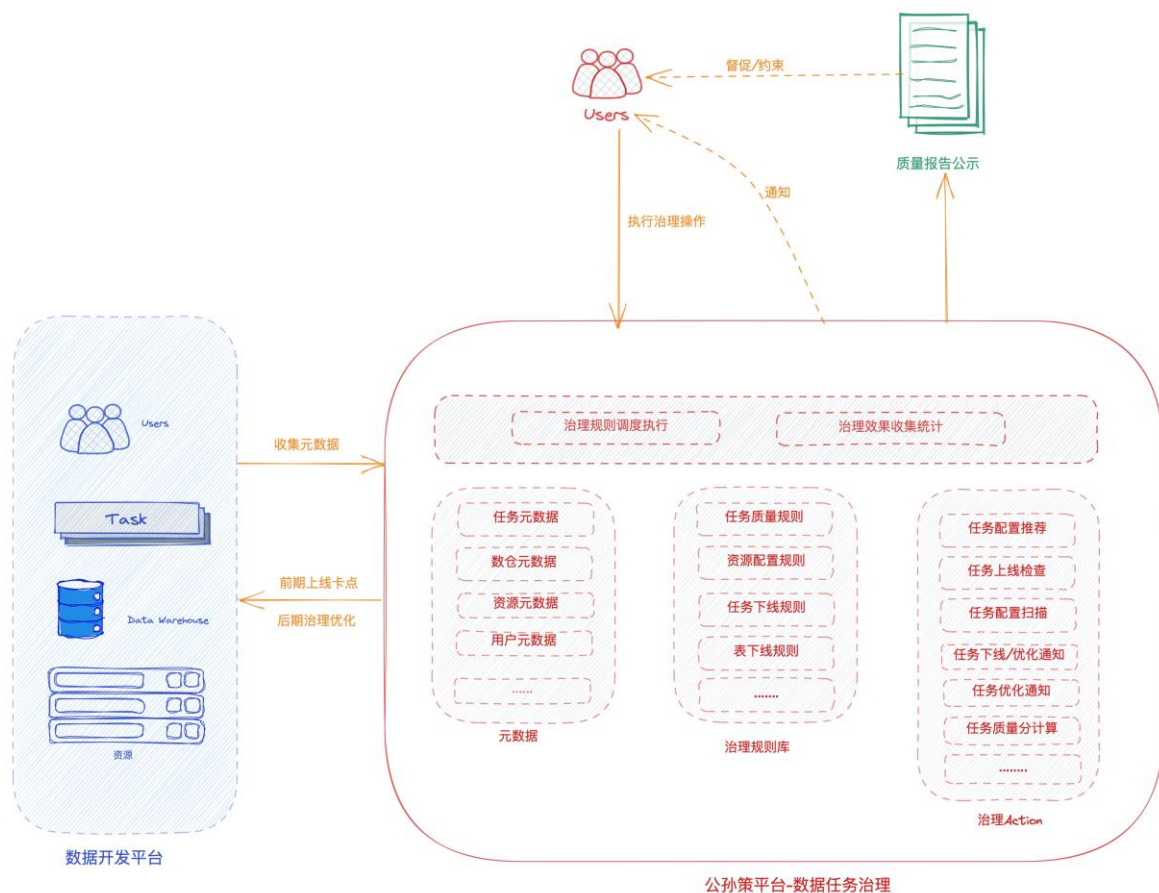
优化：技术升级、配置优化、资源优化

- 技术升级：历史任务升级到分区流表
- 配置优化：大资源使用任务优化
- 资源优化：模块化任务针对性优化，IO密集型任务、大状态任务定制资源队列配置



数据任务治理实践

可持续：告别大扫除式治理，数据任务治理常态化、平台化



- 收集任务、数仓、资源等元数据
- 沉淀前期治理经验，落地治理优化规则
- 整合公示开发质量报表，督促任务治理常态化
- 效果
 - 开发中：任务配置推荐、上线流程卡点
 - 运行中：任务质量扫描、数据/任务下线优化推进
 - 下线后：治理效果收集，治理规则持续优化

未来规划

未来规划

1

开发门槛问题

数据开发门槛高，配置优化困难，涉及中间件多

2

数据链路追踪

从数据收集到数据服务，出口多，难闭环

3

存储选择优化

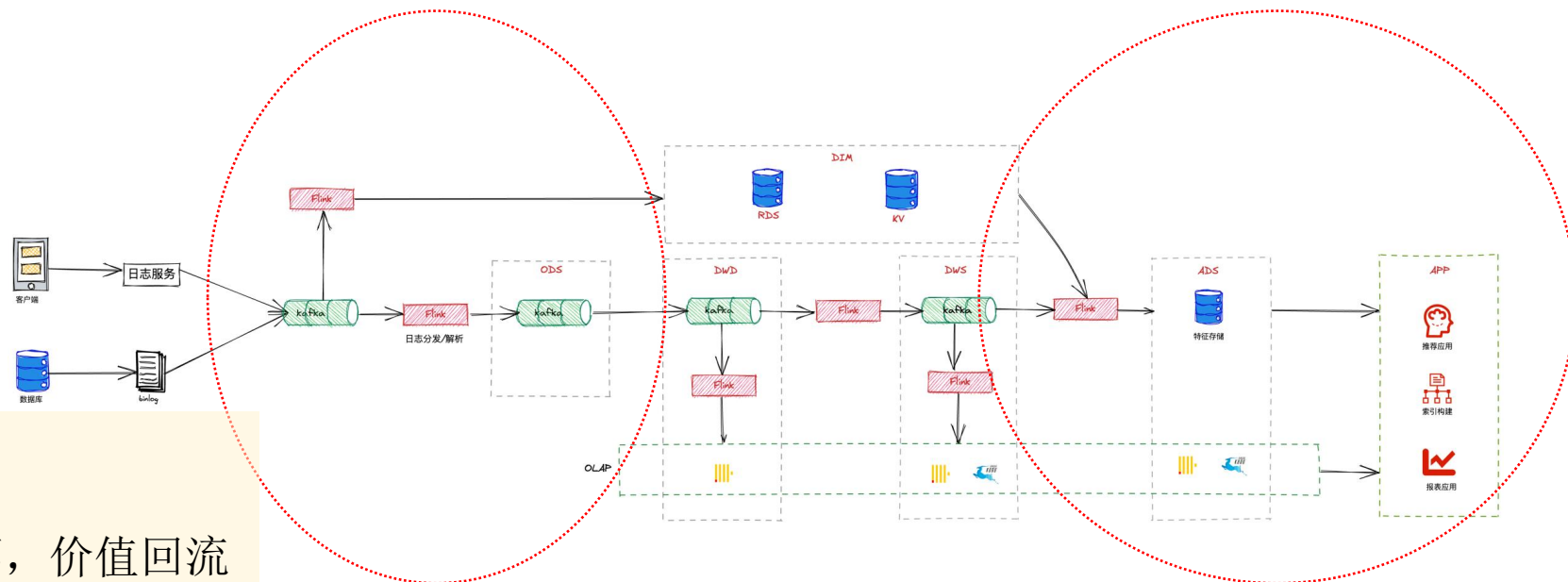
面对如此多的存储，如何选择，如何规范使用，如何优化性能

4

开发经验沉淀

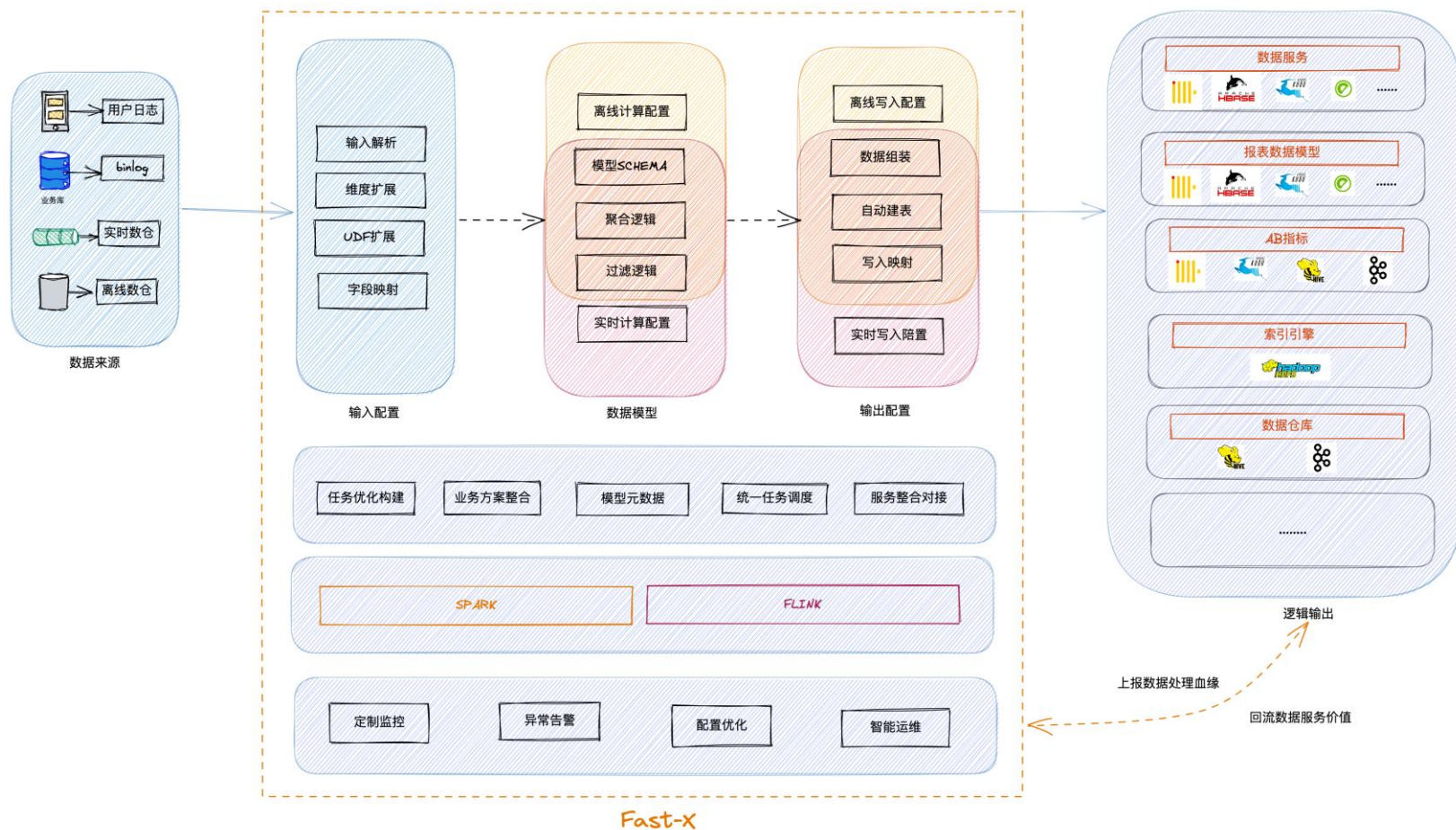
Lambda架构、存储优化、bitmap使用等

未来规划



- **底代码**：降低数据开发门槛
- **端到端**：从入仓到出仓数据闭环，价值回流
- **批流一体**：lambda架构平台化
- **场景化**：以业务场景为基础，和下游数据服务深度整合，沉淀数据架构+经验，平台化、工具化

未来规划



- 以数据模型为中心
- 配置化为主要开发手段
- 上下游服务、架构、业务逻辑深度整合
- 批流一体任务统计生成调度
- 解决80%通用的常见场景问题

THANKS

SQL Server
vertica
D B 2
G B a s e
O r a c l e
达梦数据库
神舟通用
KingbaseES

2010

2014

2018

openGauss
OceanBase
ArkDB
RASESQL
HotDB
StellarDB
QianBase xTP
GoldenDB
云树Shard
MatrixDB
DynamoDB
SinoDB
DolphinDB
FastData
Galaxybase
KunDB
GDB
GaussDB
PolarDB
KunDB
Spacture
SequoiaDB
OushuDB
ArgoDB
开务数据库
GreatDB
MongoDB
TDSQL
TiDB
Tapdata
StarRocks
UbiSQL