

第2章-离线数据仓库开发

• 1.数仓背景与目标

- 1.1 早期面临的问题

- 95% 的表建在同一个数据库
 - xxx库3000+张表,表逻辑层次不清晰
- 表定义不规范, 标准不统一
 - 表命名形式多样化, 对应数据层级不清晰
 - 字段采用关键字命名: "from"、"date"
 - 时间分区形式不统一: yyyyMMdd、yyyy-MM-dd
 - 默认以textfile格式存储, 磁盘空间使用不合理
- 数据建模意识差, 缺乏对业务的高度抽象
 - 数据报表生产直接依赖底层表
 - 数据、主题域划分不清晰
 - 缺乏数据沉淀, 重复计算浪费资源
 - 数据RD在业务分工责任不明确, 信息入口不能收拢、统一
- 烟囱式开发, 产生数据孤岛
 - 数据调研投入不足: 业务、需求、数据库
 - 指标定义混乱, 重复开发、数据冗余
- 其他业务团队各自建数仓
 - 风格多样化
 - 管理维护成本高

- 1.2 目标期望

- 统一的数据仓库开发标准、规范
- 开放的数据存储、建模、计算能力
- 可落地、可扩展, 满足未来2年左右千万日活的业务体量

• 2. 数仓需求分析

● 痛点

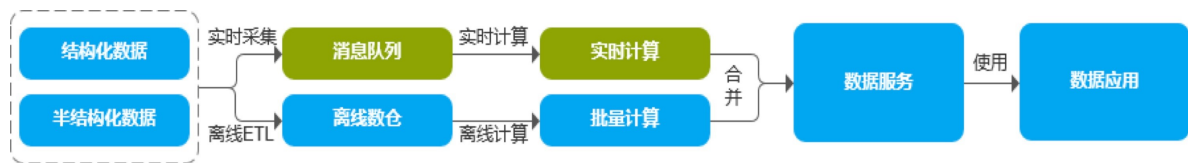
- 数据库、表命名管理乱
- 烟囱式开发，数据孤岛
- 找数难，用表难，不敢用
- 指标定义不统一，重复开发，数据不一致
- 敏感数据泄露风险，集群存储空间告警
- DB表全量同步效率低，影响线上业务
- 业务方自建数据仓库
- 数据开发新人如何上手

● 解决

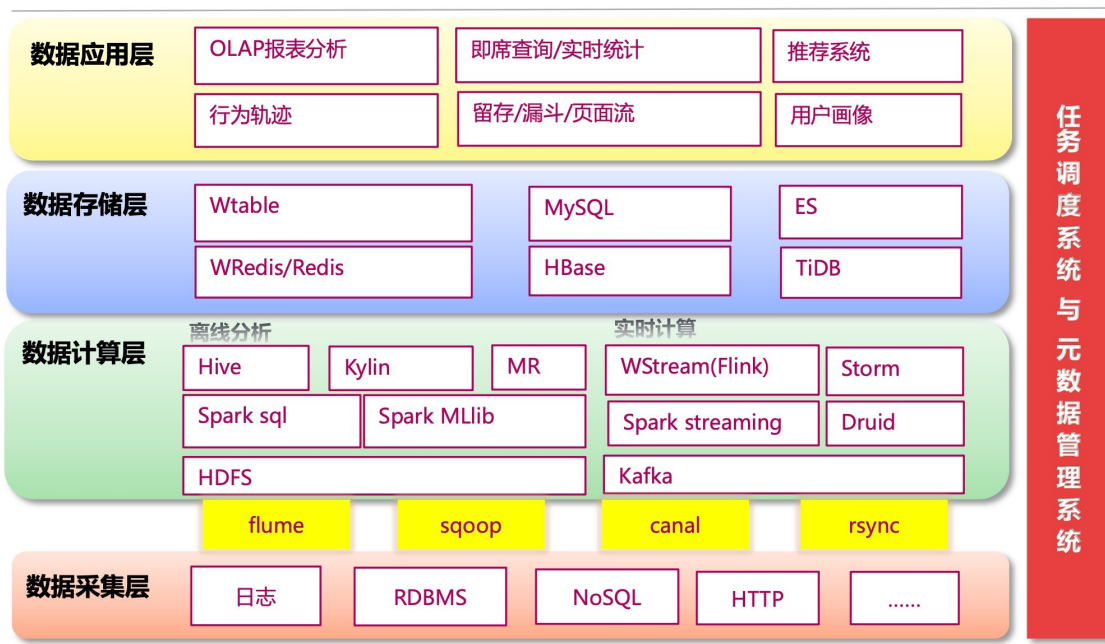
- 逻辑分层 制定规范 约束建表流程和命名，迁移老业务
- 维度建模 主题域划分 数据结构简单 分析灵活多样化
- 数据地图 收集表、任务元数据，追溯数据血缘关系
- 指标字典 [OneData](#)体系 命名规范管理 统一口径规则
- 数据治理 权限访问控制，数据生命周期管理
- 增量抽取 设计拉链表，后期考虑订阅binlog日志增量消费
- 开放共享 共享DW层 自建ADS层
- 培训宣贯 拉齐认知，新员工入职培训和数仓开发流程宣贯

● 3. PB级数仓构建

- 3.1 Lambda架构

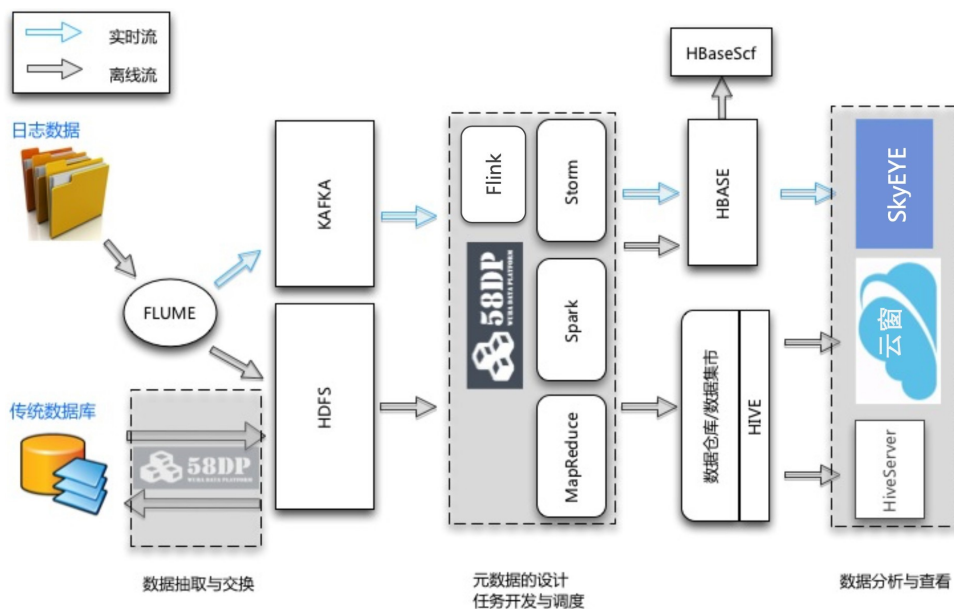


- 3.2 数据平台架构

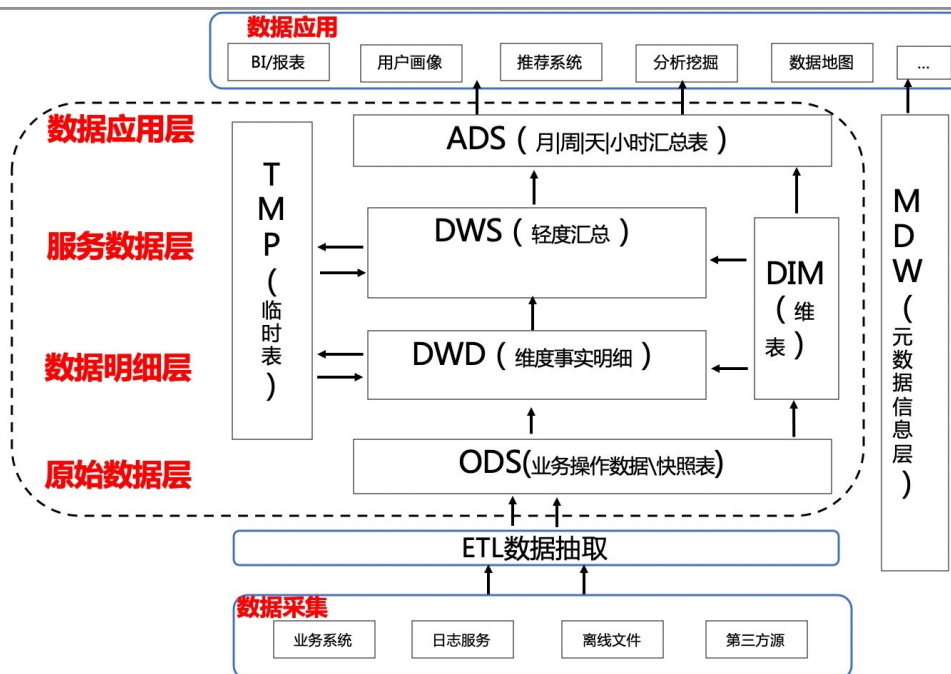


- 3.3 数据开发流程

- 未来大数据开发的趋势：
 - SQL + 拖拽开发
 - web
 - stream



3.4 数仓分层架构



3.4.1 为什么要分层

- 清晰数据结构：每一个数据分层都有它的作用域，这样我们在使用表的时候能更方便地定位和理解。
- 数据血缘追踪：简单来讲可以这样理解，我们最终给业务诚信的是一能直接使用的张业务表，但是它的来源有很多，如果有一张来源表出了问题，我们希望能够快速准确地定位到问题，并清楚它的危害范围。
- 减少重复开发：规范数据分层，开发一些通用的中间层数据，能够减少极大的重复计算。
- 把复杂问题简单化：将一个复杂的任务分解成多个步骤来完成，每一层只处理单一的步骤，比较简单和容易理解。而且便于维护数据的准确性，当数据出现问题之后，可以不用修复所有的数据，只需要从有问题的步骤开始修复。
- 屏蔽原始数据的异常：不论是数据的异常还是数据敏感性，使真实数据与统计数据解耦开
- 屏蔽业务的影响：不必改一次业务就需要重新接入数据。

3.4.2 分层定义

- ODS：Operational Data Store 原始数据层，主要包含业务数据库快照数据（rawdb）、运营埋点数据(rawdata)、其他业务等数据。
- TMP：临时层，数据处理的辅助处理层，服务于DWD、DWS层，主要是一些临时存储的数据,根据实际情况选择创建与不创建，包括：计算任务的中间结果数据。
- DIM：维度数据层，主要包含一些业务维度数据。实例：地区表，订单状态，支付方式，审批状态，商品分类，商品型号，渠道类型、终端类型、广告位、红包计划等
- DWD：data warehouse detail，存储经过标准规范化处理（即数据清洗）后的运营数据，是基础事实数据明细层。实例：行为事件明细表、mysql各业务数据经过ETL处理后的实体表。
- DWS：data warehouse service，数据服务主题层或者宽表层，按数据、业务专题进行划分，支持OLAP分析、数据分发等，其信息主要来源于DWD 或TMP层汇总数据。实例：新激活用户表、日活表、历史激活用户表
- ADS：application database service，应用数据层, 面向具体应用的表，要创建在这层，可导入hbase或mysql等使用。实例：按季、月、周、天、小时等粒度计算汇总的结果存入mysql、hbase的报表
- MDW：meta data warehouse元数据信息层，元数据机制主要支持以下五类系统管理功能：
 - 描述哪些数据在数据仓库中;
 - 定义要进入数据仓库中的数据和从数据仓库中产生的数据;
 - 记录根据业务事件发生而随之进行的数据抽取工作时间安排;
 - 记录并检测系统数据一致性的要求和执行情况;
 - 衡量数据质量。

3.4.3 数据库表命名规范

业务规范层	数据库名称	数据库释义	物理表命名规范	数据存储格式
ODS	hdp_bw_ods_global	从公司各业务mysql表同步的快照信息和运营埋点数据	埋点日志：ods_log {埋点说明} {更新方式} {时间粒度}运营数据库：ods {业务数据库表} {更新方式} {时间粒度}	Text

业务规范层	数据库名称	数据库释义	物理表命名规范	数据存储格式
DWD	hdp_bw_dwd_global	包含公司各业务经过ETL后的基础事实明细表	埋点日志: dwd_log 业务过程 更新方式 时间粒度运营数据库: dwd {数据库类型} (mysql hbase\wtable\redis) } {业务过程} {更新方式} {时间粒度}多数据源: dwd{业务过程}{更新方式} {时间粒度}	Parquet
DWS	hdp_bw_dws_global	按数据、业务专题进行划分的轻度汇总数据	dws {业务主题域} {业务过程} {更新方式} {时间粒度}	Parquet
DIM	hdp_bw_dim_global	业务维度相关的字典数据	dim {维度类型} (cate city channel group) } {更新方式}_ {时间粒度}	Text
TMP	hdp_bw_tmp_global	存放数据计算过程中的临时结果表	tmp {数据层类型}(dwd dws ads) {业务过程描述}	Parquet

业务规范层	数据库名称	数据库释义	物理表命名规范	数据存储格式
ADS	hdp_bw_ads_global	存放面向各业务应用分析的通用结果表	ads {数据主题域} {业务过程描述} {更新方式} {时间粒度}	Text\Parquet
ADS	hdp_bw_ads_{部门/产品线}	业务方个性化应用数据表	ads {数据主题域} {业务过程描述} {更新方式} {时间粒度}	Text\Parquet

3.4.4 业务主题域

- ODS层，按照业务主题域进行划分管理
- ADS层，按照数据主题进行划分管理

主题域	子域
用户：user	
商品：info	
	属性：param
交易：trade	
	订单：order
	优惠券：coupon
	回收：recycle
	租赁：lease
	维修：repair
流量：traffic	
	用户行为：ub
	归因：attribute
	直播：live
供应链：scm	
	物流：logistics
	采购：pur
	仓储：store
营销：market	
	内容：content
	活动：activity
服务：service	
	客服：csc
	售后：afs
	鉴定：auth
	质检：qc
商业广告：biz	
	线索：clue
	客户：customer

主题域	子域
	充值: recharge
	账务: account
	活动: activity
	推广: promote
	广告: ad
	增值产品: valueadd
	用户: user
渠道: channel	
地址: address	
财务: finance	
风控: spam	
竞品: compete	

3.4.5 数据主题域

用户增长: activity
 商品发布: addinfo
 交易链路: trade
 私信互动: social
 收入: income
 推送: push
 流量: traffic
 供应链: scm
 营销: market
 服务: service
 商业广告: biz
 渠道: channel
 地址: address
 财务: finance
 风控: spam
 竞品: compete

3.4.6 更新方式命名规范

增量: inc
 全量: full

3.4.7 表名时间粒度规范

分区表

小时(hour): 1h
天(day): 1d
周(week): 1w
月(month): 1m
季度(quarter): 1q
年(year): 1y
.....

分区字段: 日期分区统一命名为: dt, 格式: yyyy-MM-dd or yyyy-MM or yyyy

注意: 单张表的分区每日新增不宜超过200个, 分区太多导致文件分布太细, 对集群 namenode 压力过大

- 非分区表: 统一后缀加上: _0p

非分区表			
	增量表	xxx_inc_1h_0p xxx_inc_1d_0p xxx_inc_1w_0p	无dt字段
	全量表	xxx_full_1h_0p xxx_full_1d_0p xxx_full_1w_0p	无dt字段
分区表			
	增量&小时 更新	xxx_inc_1h	dt=yyyy-MM-dd-HH
	全量&小时 更新	xxx_full_1h	dt=yyyy-MM-dd
	增量&天 更新	xxx_inc_1d	dt=yyyy-MM-dd
	全量&天 更新	xxx_full_1d	dt=yyyy-MM-dd
	增量&周 更新	xxx_inc_1w	dt=yyyy-MM-dd
	全量&周 更新	xxx_full_1w	dt=yyyy-MM-dd
	增量&月 更新	xxx_inc_1m	dt=yyyy-MM
	全量&月 更新	xxx_full_1m	dt=yyyy-MM
	增量&年 更新	xxx_inc_1y	dt=yyyy
	全量&年 更新	xxx_full_1y	dt=yyyy

3.4.8 字段命名规范

日期字段: <业务主体>_date
时间字段: <业务主体>_time
属性字段: 属性自身英文单词; 如: status
id 字段: <标识主体>_id;
标识字段: is_<标识主体>; 如: is_true
指标字段: 时间周期+修饰词+原子指标 如: 近30天优品帮卖支付订单数 英: ypbm_payorder_1m
计次字段: <计数主体>_pv; 如: visit_pv
排重计数字段: <计数主体>_uv; 如: visit_uv
价格字段: <业务主体>_price; 如: pay_price
来源字段: <业务主体>_source; 如: order_source
比例字段: <业务主体>_rate; 如: gmv_yoy_rate

3.4.9 调度任务开发规范

- 1、任务命名规范

任务命名格式: {输出表名}-任务描述 或者 任务描述-{输出表名}

- 2、开发规范

- 1、表和列的注释是否有确定, 复杂计算逻辑是否有注释
- 2、任务是否支持多次重跑而输出不变, 不能有insert into语句insert overwrite into table partition(...)
- 3、分区表是否使用分区键过滤并且有有效剪裁
- 4、外链接的过滤条件是否正确, 例如在做链接的where语句是否存在右表的过滤条件
- 5、关联小表, 是否使用 /*+mapjoin(small_table)*/
- 6、不允许引用别的计算任务临时表
- 7、原则上不允许存在一个任务更新多个目标表
- 8、是否存在笛卡尔积
- 9、禁止在代码里使用drop table, create table, rename table, change column 等 DDL 语句
- 10、使用动态分区时, 有没有检查分区键值为 Null 的情况
- 11、DQC质量监测规则是否配置, 严禁裸奔
- 12、代码中有没有进行适当的规避数据倾斜的语句
- 13、Where条件中 is null 语句有没有进行 空字符串的处理 string(null "")

4. PB级数据采集平台架构设计

关于数仓采集方案:

1、数据源

小程序 + App + PC端 + Java后台系统 + 业务数据库

2、采集方式:

前端: 埋点

1、Google Analytics ga.js

2、淘宝数据收集规范 (流量跟踪体系设计: scm acm spm)

参照: <https://book.douban.com/subject/27074564/>

参数一本书：大数据之路
后端：JavaSDK

3、采集工具

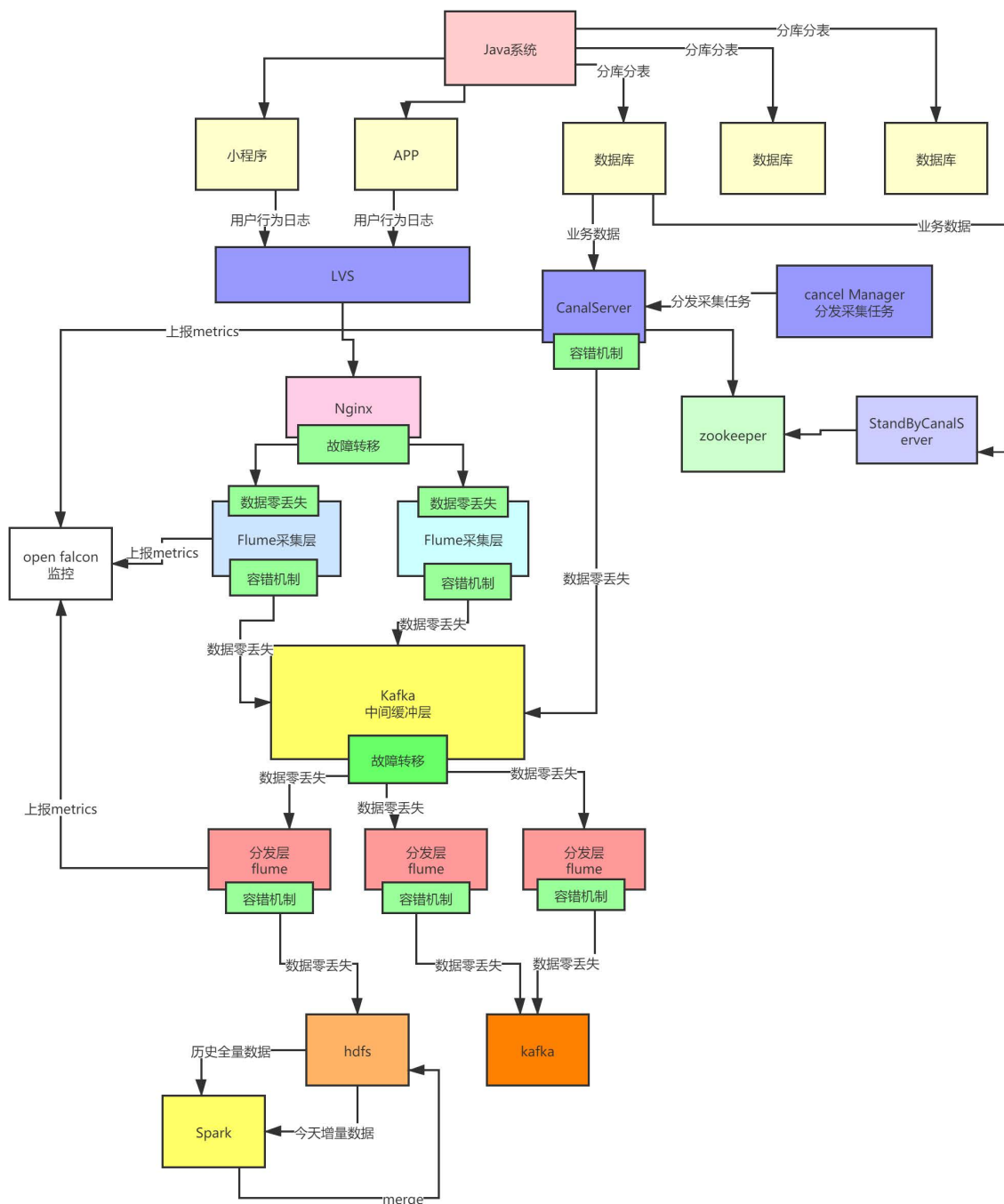
业务数据库：

Sqoop 全量抽取

Canal 针对 MySQL 实时同步

日志文件

rsync + Flume



• 5. PB级数仓平台资源评估

- 服务器配置

内存 (GB)	cpu	磁盘 (容量x个数)	网络带宽
128	8Core x2	SATA: 7TB x 12	万兆

- 数据规模(1副本)

每天增量 (TB)	每天记录数 (条)	累计数据量 (PB)	保留周期 (年)	hadoop节点	任务运行 数量
20+	400+亿	15+	>=3	700+	10000+/天

• 6. 总结

- 今天的重点内容:

- 1、数仓案例-用户行为分析
- 2、全域数仓背景和需求
- 3、全域数仓构建
- 4、PB级全域数仓构建
- 5、数据采集平台构建
- 6、资源评估

完全剖析，真实一二线互联网企业是如何构建全域数仓的核心思路。