WOTA

51CTO

World Of Tech 2017

全球架构与运维技术峰会

2017年4月14日-15日　北京富力万丽酒店

ARCHITECTURE

WOTA

51CTO

出品人及主持人：

吕　毅　链家网架构师
大数据平台团队负责人

大数据系统架构

WOTA

51CTO

今日头条大数据平台实践

WOTA

51CTO

王烨

今日头条

数据平台架构师

分享主题：

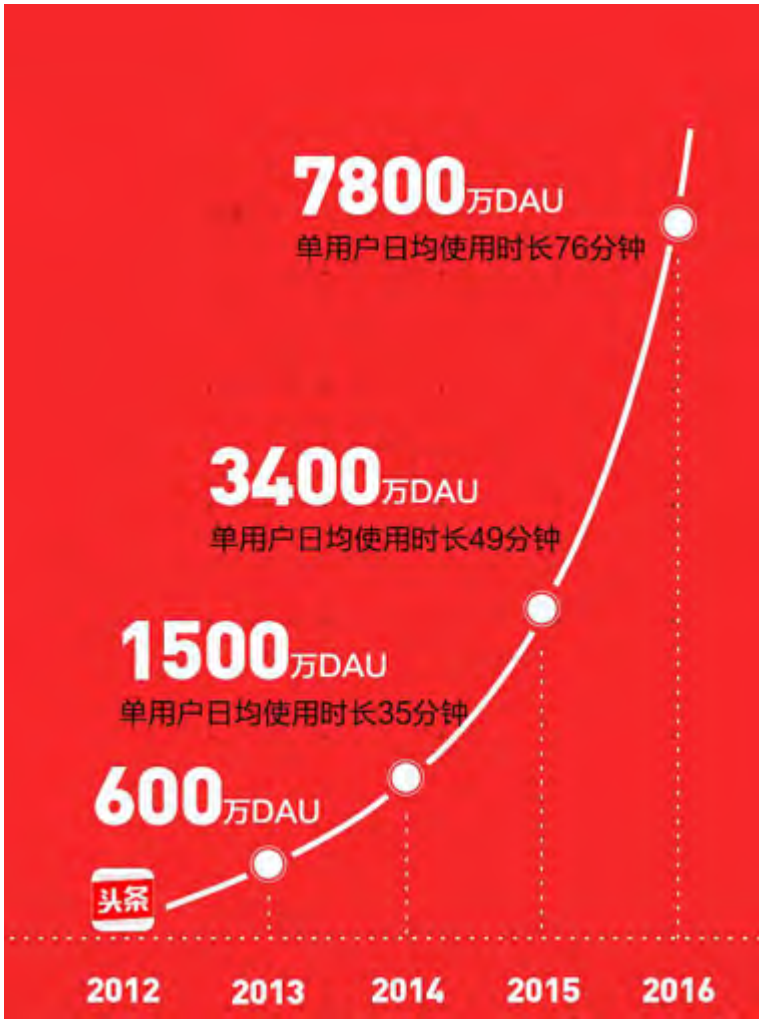
今日头条大数据平台的演进

关于我

- 今日头条数据平台团队架构师
- 2011 - 2014年 豆瓣
- 2014 至今 头条

头条简介

- 2012年创立
- 专注于信息分发
- 基于数据挖掘的推荐引擎
- 8500万DAU
- 单用户日均使用时长76分钟



数据规模

- 每日处理数据量7.8PB
- 训练样本量200亿条
- 服务器总量40000台
- Hadoop节点3000台

数据平台职责

- 源自推荐业务需求，搭建面向全公司的通用数据平台
- 维护用户行为数据流
- 维护平台基础数据仓库，协助维护业务部门数据仓库
- 维护面向RD、分析师的数据工具集
  - 查询、日志收集、入库、调度、依赖管理、元数据、报表
- 维护面向PM、运营的通用用户行为分析平台
- 维护底层查询引擎
  - Hive, Presto, Kylin等OLAP查询引擎，支撑上层数据平台和数据仓库
- 兄弟团队：inf团队，维护hadoop、流式计算等存储和计算基础设施

理念

- 头条从骨子里贯彻了数据驱动的思想，带来大量数据需求
- 数据生命周期：
  1. 生成
  2. 传输
  3. 入库
  4. 统计、分析、挖掘
- 数据生命周期每个环节的难度都会随着数据规模的扩大而上升
- 提供整体解决方案，降低使用门槛，方便接入各种业务数据
- 抽象常用互联网产品分析模式，覆盖大部分常用需求
- 支持业务部门自行解决数据需求，期望参与业务的人能更直接的掌握数据
- 解放业务部门工程师生产力，不至于被各种临时跑数需求困扰
- 支持专门的数据分析师的工作，提供更专业的工具支持

数据生成与采集

- C端业务公司以日志形式为主，如头条、百度等
- B端业务公司以关系数据为主，如美团、58等
- 头条数据以用户行为数据居多
- 采用事件模型来描述日志
- 以SDK形式接入，支持客户端、服务端埋点

埋点演进1

- 历史实现：各业务场景自定义日志格式
- 演进：统一到事件模型，标准化格式、基本维度
- 收益：
  - 保证信息的结构化、自描述
  - 降低接入成本，可以复用统一处理流程、数据仓库、分析平台

埋点演进2

- 历史实现：埋点通过文档、Wiki等管理
- 演进：实现埋点管理系统，覆盖埋点生命周期
- 收益：
  - 得到埋点元信息的结构化描述，后续应用在数据清洗、分析平台等场景
  - 标准化埋点上线流程
  - 支持客户端自动化测试

SDK演进1

- 历史实现：头条App按约定打数据
- 演进：实现通用SDK，包括客户端SDK和服务端SDK
- 收益：
  - 保证生成的日志符合埋点规范
  - 统一基本口径， 包括App启动、设备标识等
  - 减少新App适配成本

SDK演进2

- 历史实现：使用JSON描述数据
- 演进：实现Protobuf描述数据
- 收益：
  - 通过IDL实现强制约束，包括数据类型、字段命名等

关系数据采集

- 历史实现：定期用单机全量抓取Mysql数据表
- 演进1：用Spark实现类Sqoop的分布式抓取
- 收益：
  - 加快抓取速度，解决单机瓶颈
- 演进2：用Canal接收Mysql binlog，离线merge出全量表
- 收益：
  - 不再直接读Mysql，减少Mysql压力
  - 对大表(千万/亿级)的处理速度更快

经验

- 数据质量很重要，埋点规范趁早确立
- 脏数据是不可避免的，引入必要的约束、清洗等

数据传输

- 以Kafka作为数据总线
  所有实时和离线数据的接入都通过Kafka，包括日志、binlog等
- 自研Databus，作为单机Agent，封装Kafka写入，提供异步写入、buffer、统一配置等feature
- Kafka数据Dump到HDFS，供后续离线处理

HDFS dump演进

- 历史实现：单机上传类Flume实现
- 演进1：用Storm+Hadoop Cli实现分布式上传
- 收益：
  - 支持高吞吐，实际最高支持到2GB/s的单个topic
- 演进2：用Spark streaming实现DumpService
- 收益：
  - 实现exactly once，保证数据不丢不重
  - 支持按内容时间戳归档，应对上游延迟、HDFS故障等
  - 减少Cli调用开销
  - 服务化，方便整合到数据平台工具

经验

- 尽早引入消息队列，与业务系统解耦

流式计算

- 用户行为数据的解析、转换、计算、存储
- 统一计数服务

计算框架演进

- 历史实现：单机kafka consumer脚本
- 演进1：手工分布式脚本
- 收益：
  - 应对单机瓶颈，按partition分割
- 演进2：使用Storm+自研python处理框架
- 收益：
  - 声明式描述topology
  - 自动处理分布式
  - 自动failover

计算框架演进

- 演进3.1：JStorm
- 收益：
  - 引入资源管理
- 演进3.2：Spark Streaming+Yarn
- 收益：
  - 引入资源管理，Spark2支持动态资源分配
  - 吞吐相比Storm更大
  - 实现At Least Once等特性比Storm更简单

经验

- 选择合适的计算框架，权衡吞吐/响应时间
- 处理的副作用操作如果能实现成幂等的，failover会简单很多

数据仓库

- 杜绝直接基于业务数据库跑统计SQL
- 杜绝基于原始日志写脚本统计
- 防止重复劳动，尽量共享中间结果
- 保证数据一致性，对维度等统一建模
- 层次化建设

数据库 VS 数据仓库

- OLTP，供业务系统使用
- 存储当前状态，快速变化
- 为高速读写优化，支持数据量有限(如单表行数最多为千万级)
- 单机或简单sharding

- OLAP，供分析使用
- 存储历史状态，定期增加分区(如每个分区都为某一天的数据快照)
- 一般不支持随机更改数据，支持海量数据(如单分区可以到百亿级)。对大量数据扫描支持更好
- 分布式存储/计算

数据模型

- 星型模型
  - 划分出系列事实表与维度表
- 层次
  - ODS，原始操作数据
    - 基于HDFS Log实现成Hive外部表
    - 基于MySQL binlog滚动生成天级Snapshot表
  - DWD，数据仓库层，清洗后的明细数据，实现成Parquet格式Hive内部表
  - MID，数据集市层，计算后的指标数据

存储

- 数据表元信息存放在Hive metastore
- 数据以Parquet格式存放在HDFS
  - 列式存储，支持嵌套数据结构

ETL演进

- 历史实现：
  - MapReduce+Python+Hadoop Streaming
  - 结果数据生成JSON
- 演进：Spark直接生成Parquet
- 收益：
  - 减少复杂处理流程中间数据落地
  - 更丰富的处理原语，逻辑更简洁

计算引擎演进

- 历史实现：Hive only
- 演进1：Hive + Impala
- 收益：
  - 引入MPP计算引擎，更好的支持adhoc类查询
- 不足：头条数据规模下稳定性变差

计算引擎演进

- 演进2：Hive+Presto+Slider
- 收益：
  - 使用Yarn资源池
  - 稳定性相对Impala提高很多
- 不足：
  - 稳定性仍有不足

计算引擎演进

- 演进3.1：Hive+SparkSQL+自研QAP
- 收益：
  - QAP通过Query分析，选择合适的查询引擎
  - 进一步解决Presto的稳定性问题
- 演进3.2：Kylin
- 收益：
  - Cube模型很适合模式固定的查询
  - 可以作为Tableau的查询后端

经验

- MPP类查询引擎各有优劣， 不追求银弹

数据平台

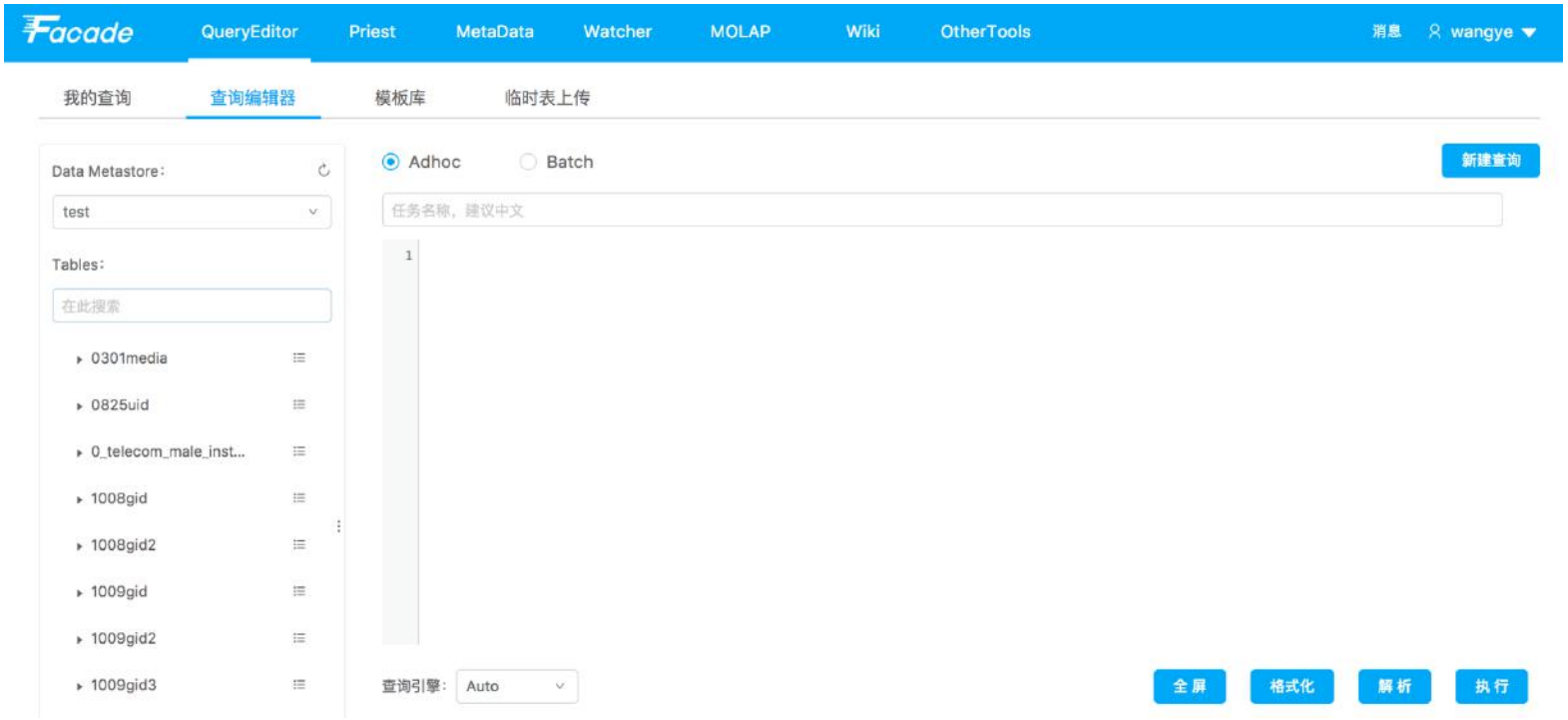

Facade
QueryEditor
Priest
MetaData
Watcher
MOLAP
Wiki
OtherTools
消息
wangye
Facade 是面向整个数据体系所提供的一站式自助平台，包含数据抽取、数据建设、元数据、多维分析、监控
几大模块。以平台化思维，向下封装底层基础设施，向上抽象用户需求场景。
「旨在降低数据的使用门槛，提高以数据为核心的迭代效率」
数据抽取
QueryEditor
数据建设
Priest
元数据
MetaData
数据监控
Watcher
多维分析
MOLAP
由 dataplatform@bytedance.com 强力驱动

数据抽取系统演进

- 历史实现：直接调用Hive Cli
- 演进1：使用开源系统Hue
- 收益：
  - 有了WebUI，门槛大幅降低
- 不足：
  - 用户体验一般，二次开发不便
  - Bug比较多

数据抽取演进

- 演进2：替换成自研的QueryEditor
- 收益：
  - 用户体验大幅提高



数据建设演进

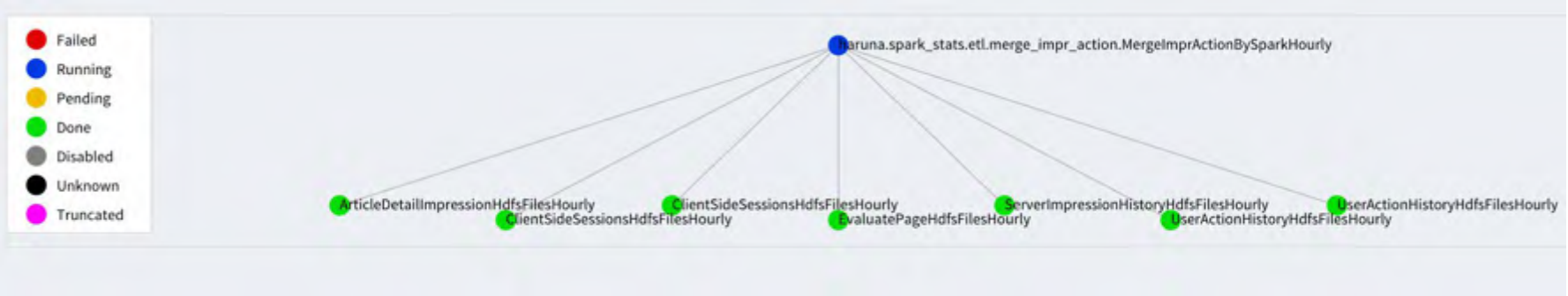
- 历史实现：业务方自行实现各种ETL
- 演进：替换成自研系统Priest，封装通用ETL过程
- 收益：
  - 统一托管平台
  - 声明式ETL定义

数据建设演进


Facade
QueryEditor
Priest
MetaData
Watcher
MOLAP
Wiki
OtherTools
消息
wangye
Hive -> Hive
Hive -> Tableau
Kafka -> HDFS
HDFS -> Hive
MySQL -> Hive
A/B test任务
基本信息
* 任务名称：
由字母，数字，下划线构成
* 优先级别：
普通优先级
* 任务周期：
天级
机房选择：
国内
美东
新加坡
任务描述：
请输入任务详细描述，方便任务管理

依赖管理演进

- 历史实现：时间调度，到点执行
- 演进：引入Luigi
- 收益：
  - 转变为Data trigger，数据ready时再执行，不依赖时间
  - 通过Python代码描述依赖条件，很灵活
- 不足：
  - 框架式设计，侵入性比较高



任务调度系统演进

- 历史实现：单机定时任务
- 演进：引入Chronos
- 收益：
  - 分布式执行，避免单点失败

报表平台演进

- 历史实现：通过后台或者邮件实现报表
- 演进1：引入Tableau
- 收益：
  - 非常丰富的报表和可视化功能
- 演进2：实现数据导入Service
- 收益：
  - 规整报表数据模型
  - 封装存储后端

总结

- 不同的阶段做不同的事情，先把需求搞定

We are hiring

- 大数据架构高级工程师
- 大数据查询引擎高级工程师
- 数据仓库与数据分析高级工程师
- 大数据产品研发工程师

请将简历投递至 hr@bytedance.com


WOTA
51CTO
Thank you !