实时计算在提升播放体验中的应用实践

爱奇艺 智能平台部 高级工程师
胡嘉伟

爱奇艺大事记

2010年4月22日
视频网站“奇艺”正式上线

2011年6月23日
推出“奇艺出品”战略
2011年11月26日
品牌战略升级为“爱奇艺”

2012年3月26日
爱奇艺发布“分甘同味”内容战略，开启网络视频的 iPPC 时代

2013年5月7日
爱奇艺与 PPS 合并，提供更优质服务

2014年7月17日
爱奇艺宣布成立影业公司，提出“爱 7.1 电影大计划”

2015年12月1日
爱奇艺 VIP 会员突破 1000 万

2016年5月6日
2016 爱奇艺世界大会圆满收官

2017年6月8日
发布全新企业愿景：做一家以科技创新为驱动的伟大娱乐公司

2018年3月29日
爱奇艺成功登陆美国纳斯达克，股票代码 IQ

2010 2011 2012 2013 2014 2015 2016 2017 2018

目录

- 用户播放体验
- 挑战与架构
- 大数据平台
- 数据挖掘与分析
- 总结 & 展望

用户播放体验

- 提供正版、高清、流畅的视频播放服务，始终是爱奇艺所追求的目标

- 以数据为驱动，立足于用户，爱奇艺高度重视用户播放体验

用户播放体验

- 什么是播放体验？
- 如何提升反映速度？
- 如何提升处理速度？
- 面临了什么样的难点？

挑战与架构

难点一

- 依赖服务多
- 子模块多
- 环境复杂

难点二

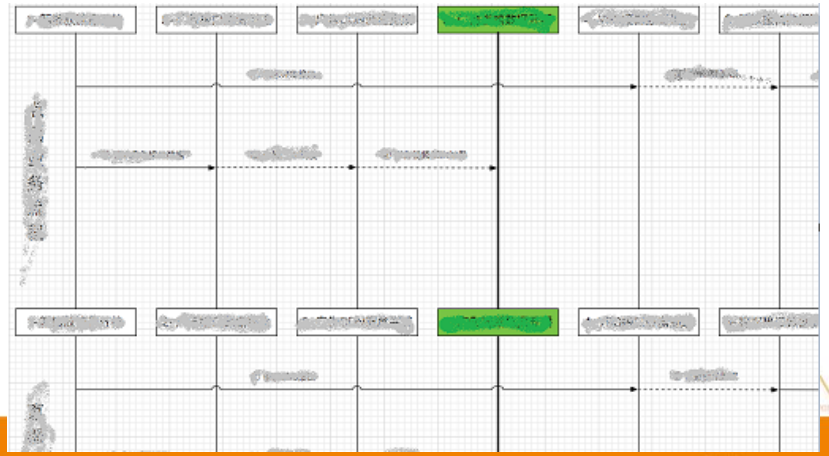
- 数据量大
- 维度多
- 查询复杂

难点三

- 时间变化
- 业务变化
- 指标变化

播放内核

视频数据模块 | 数据加载模块 | 广告模块 | 会员模块 | 解码模块 | 渲染模块



挑战与架构

难点一

- 依赖服务多
- 子模块多
- 环境复杂

难点二

- 数据量大
- 维度多
- 查询复杂

难点三

- 时间变化
- 业务变化
- 指标变化

顶层设计

大数据

数据挖掘

挑战与架构

- 播放数据整体架构

挑战与架构

- 技术选型
  - Kafka
    - 吞吐大，容错强，稳定高
  - Spark
    - 吞吐大，生态成熟
    - StreamingSQL（自研）
  - Flink
    - 实时性高，操作细腻
  - Impala
    - 与 Hadoop 生态兼容，支持 Join ，无缝支持 Kudu
  - Kudu
    - 即插即查，聚合性能优于 ES

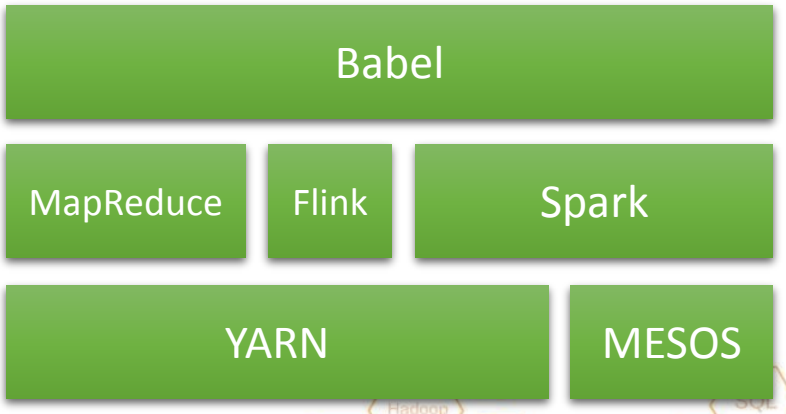
大数据平台

- 开发平台 - Babel
  - 批处理、实时计算
  - 开发、托管、运维
  - 元数据管理、权限管理

- 日志收集平台 - Venus
  - 收集拆分日志数据
  - 实时数据秒级延迟

大数据平台

- 开发平台 - Babel
  - 任务开发：
    - IDE模式：编写 SQL
    - Jar 模式：MR/Spark/Flink 程序
    - 工作流：批处理任务调度
  - 任务运维：
    - 批处理：作业调度，失败重试
    - 实时计算：状态监控，失败重提交
  - 元数据中心：
    - 数据源：注册，发布，检索
    - 数据权限，血缘依赖
  - 数据交换：
    - 各数据源互通



大数据平台

- StreamingSQL
  - 基于 Spark Streaming 与 Structured Streaming
  - 使用场景：编写 SQL 描述实时 ETL 与实时报表计算
  - 时间模式：处理时间、事件时间
  - 输出数据源：Hive，MySQL，Kudu
  - 语法定义：
    - 流表：定义输入 Kafka 数据源信息，以及如何解码数据
    - 维度表：定义静态表，可用于与流数据的 Join
    - 临时表：定义计算的中间状态
    - 结果表：定义输出数据源信息，Append 与 Upsert 两种输出模式
    - 自定义函数：用户通过拓展接口，能自定义函数

大数据平台

- StreamingSQL
  - 示例：打印数据延迟情况

```
CREATE STREAM TABLE t1 (`timestamp` long) WITH (
  type="kafka",
  brokers = "xxxxxxxxxx",
  topics = "xxxxxxxx",
  deserializer = "json");

CREATE TMP TABLE t2 AS
SELECT floor(`timestamp` / 1000) AS `timestamp` FROM t1;

CREATE RESULT TABLE r (`timestamp` timestamp, min_time timestamp, p99 timestamp,
  p50 timestamp, avg_time timestamp, max_time timestamp) WITH (type="console");

INSERT INTO r
SELECT
  current_timestamp() AS `timestamp`,
  min(`timestamp`) AS min_time,
  percentile(`timestamp`, 0.01) AS p99,
  percentile(`timestamp`, 0.5) AS p50,
  avg(`timestamp`) AS avg_time,
  max(`timestamp`) AS max_time
FROM t2;
```

大数据平台

- StreamingSQL
  - 线上实例

```
# mesos
ENV stream.platform="mesos";
ENV stream.cluster="Mesos-C1";

ENV stream.worker.num=60;
ENV stream.worker.cpu=8;
ENV stream.worker.mem=12;

ENV stream.queue="xxxxx";
ENV stream.principal="xxx";

ENV stream.interval=30;

ENV stream.log.level="ERROR";
ENV spark.hadoop.fs.hdfs.impl.disable.cache=true;
```

```
# 设置UDF

create udf pingback as 'StandardPingback' with ('v2', 6);
create udf parse_url as 'URLParser';
create udf bigVer as 'Ver2Int';

```

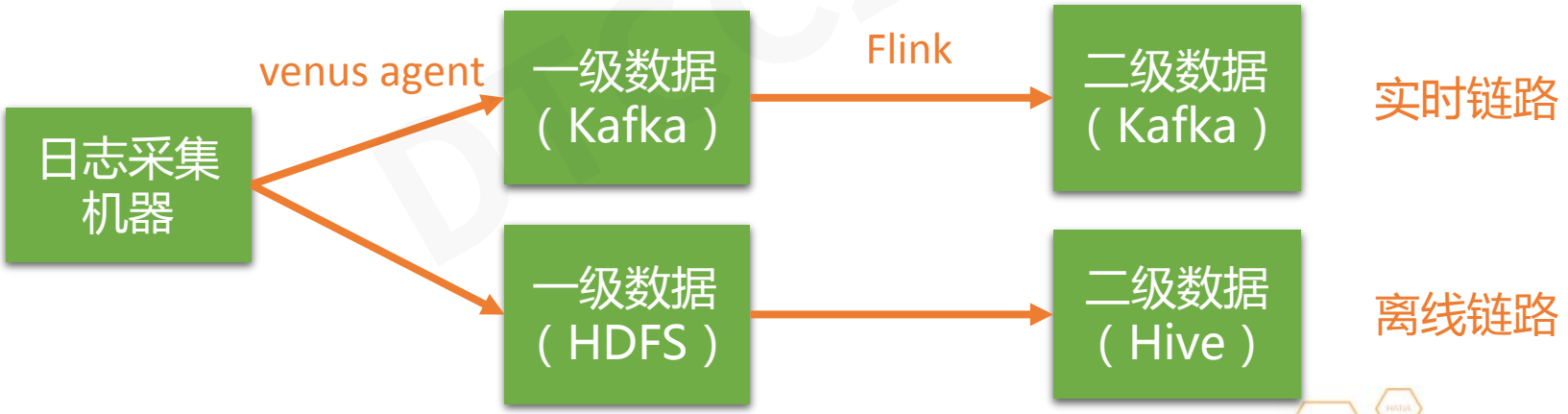
大数据平台

- StreamingSQL
  - 线上实例

```
# 设置SQL

CREATE STREAM TABLE t1 (rawMessage string, lv int) WITH (
  type="kafka", brokers = "h1:9092,h2:9092,h3:9092", topics = "PB-xxx",
  deserializer = "json", repartition=160);

create tmp table t2 as
   select unix_timestamp(m['datetime'], 'dd/MMM/yyyy:HH:mm:ss') as ts_tmp,
  parse_url(m['url']) as params from (select pingback(rawMessage) as m from t1 where lv % 100 = 3);

create tmp table t3 as
   select cast(ts_tmp div 300 * 300 as timestamp) as ts, params from t2
   where params['p1']='3' and  params['pv']='x-y' and params['dt'] is not null;

create tmp table t4 as
    select ts, params['v'] as v, params['w'] as w, params['ver'] as ver, params['sys'] as sys, params['dt'] as dt
    from t3 where params['v'] = 'd' or (params['v'] = 'e' and params['ot'] = '128_b') or (params['v'] = 'a' and params['dc'] = '33');

create tmp table t5 as
    select count(case when v='a' then 1 else null end) as a1, count(case when t='d' or t='e' then 1 else null end) as a2,
     bigVer(w) as bw, w, ver, sys, dt, ts from t4 where bigVer(w) >= 1105 group by bigVer(w), w, ver, sys, dt, ts;

create result table r (a1 long, a2 long, bw varchar(8), w varchar(50), ver varchar(256), sys varchar(256), dt varchar(256), ts timestamp)
with ( type="mysql", username="xxx", password="xxx", url="xxx", table="xxx");

insert into r select a1, a2, concat(bw div 100, '.', bw % 100) as bw, w, ver, sys, dt, ts from t5 where a1 > 20 or a2 > 60;
```

大数据平台

- 日志收集平台 - Venus
  - 采集日志数据，提供给下游计算使用
  - 分为实时链路与离线链路
  - 支持千万级 QPS

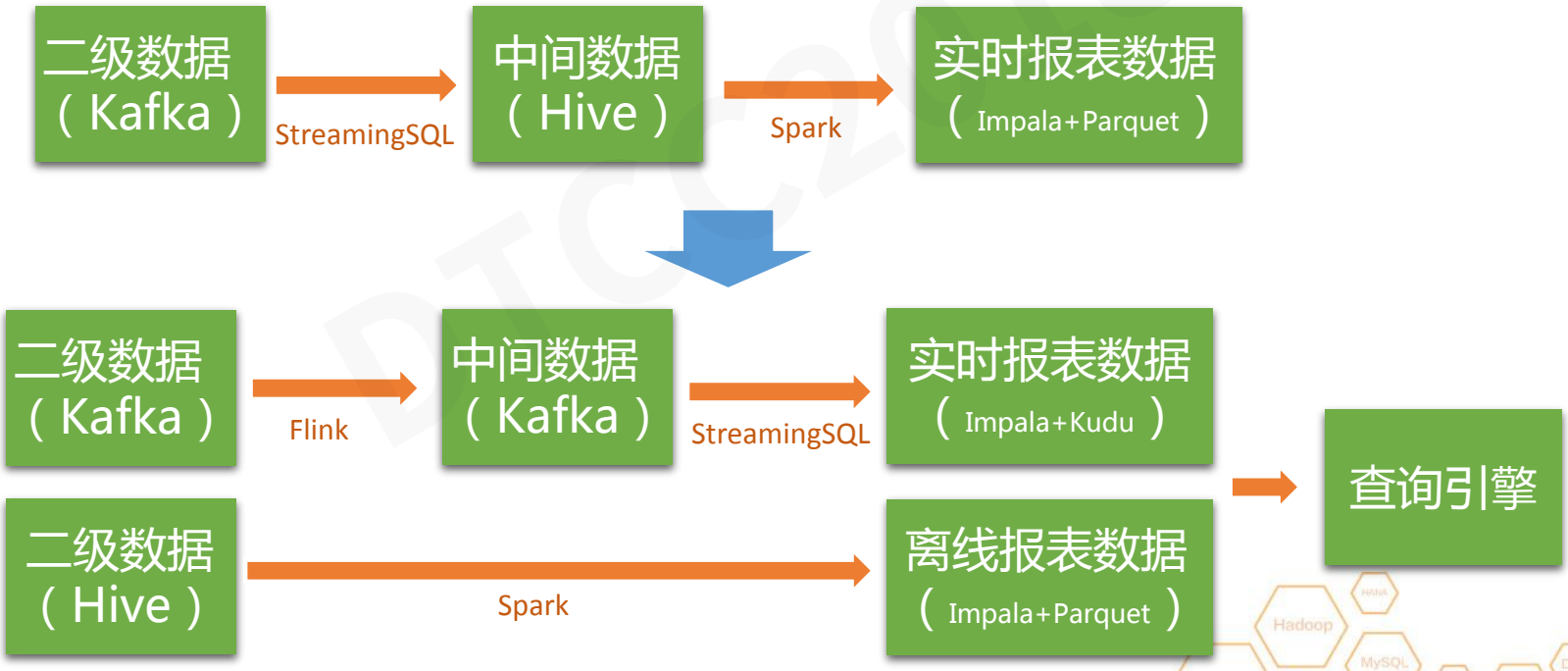


数据挖掘与分析

- 数据计算
  - 基础数据与指标计算
  - 数据质量监控
- 异常检测
  - 对细粒度指标监控疑似异常点
- 根因分析
  - 归因异常点
  - 详细维度补充
- Ad-hoc 查询
  - 细粒度数据查询

数据挖掘与分析

- 数据计算（实时部分）
  - 架构演变



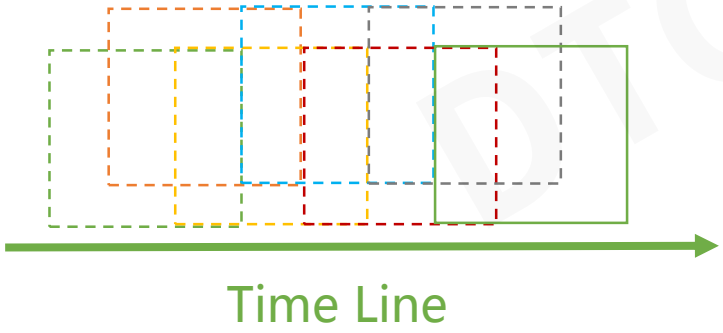
数据挖掘与分析

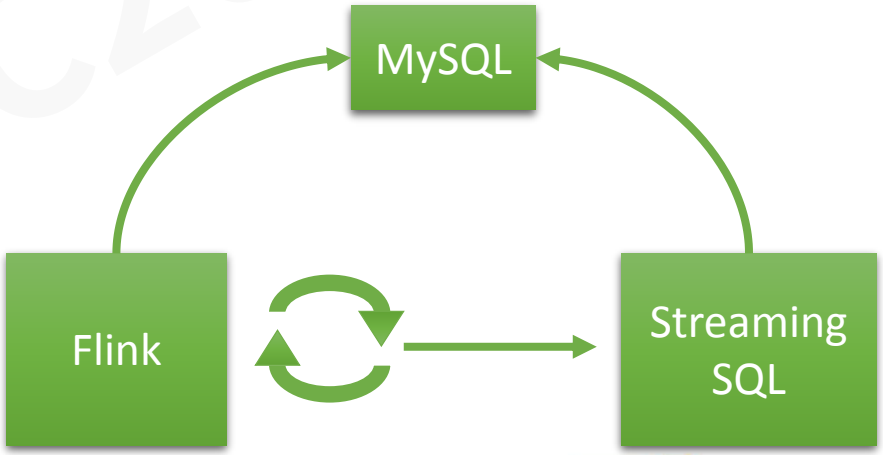
- 原方案
  - 架构简单
  - 节省计算资源
  - 有数据丢失风险
  - 计算延迟较高

- 现方案
  - Lambda架构
  - 数据最终一致
  - 计算延迟低
  - 计算链路复杂
  - 数据查询复杂

数据挖掘与分析

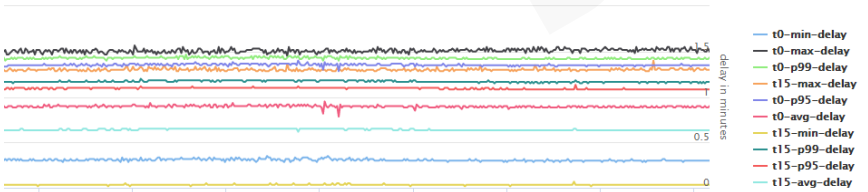
- 实时计算 – 反刷量
  - 刷量数据处理
    - iterate 运算符
    - sliding window 运算符
    - 制造数据链路时间差

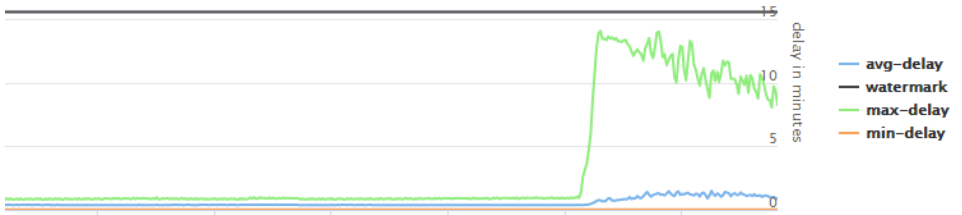




数据挖掘与分析

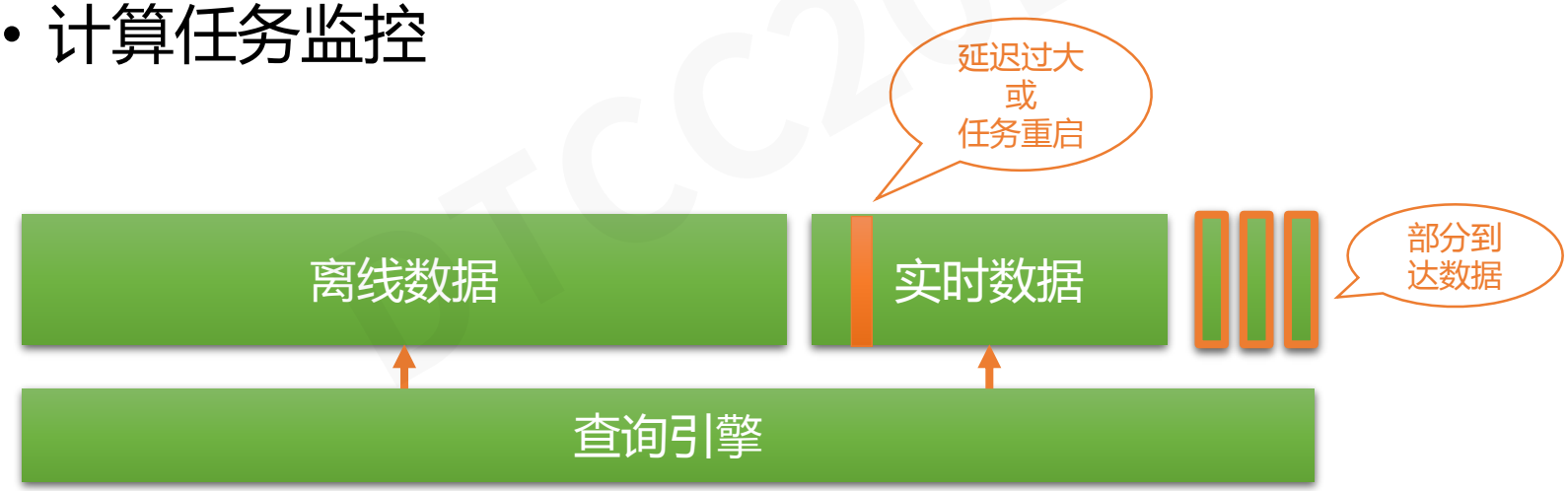
- 实时计算 – 事件时间
  - 数据到达情况判断
    - 单纯使用 watermark 判断数据到达情况不够灵活
      - watermark 设置过大，白白等待
      - watermark 设置过小，大量时段数据不可用
    - 每个微批次数据延迟情况 + 大 watermark
      - 数据延迟小，及时触发后续计算
      - 数据延迟大，最多等待至 watermark





数据挖掘与分析

- 实时计算 – 数据查询
  - 离线数据与实时数据融合
  - 数据到齐监控
  - 数据延迟监控
  - 计算任务监控

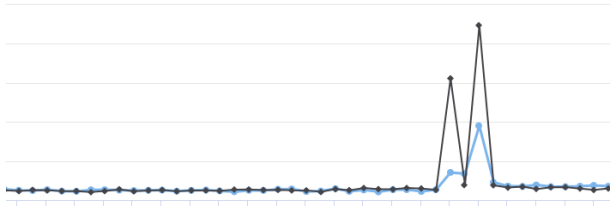


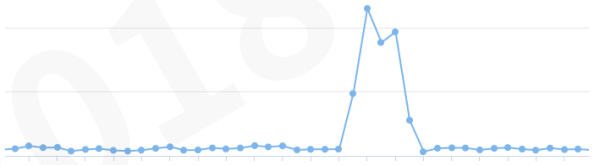
数据挖掘与分析

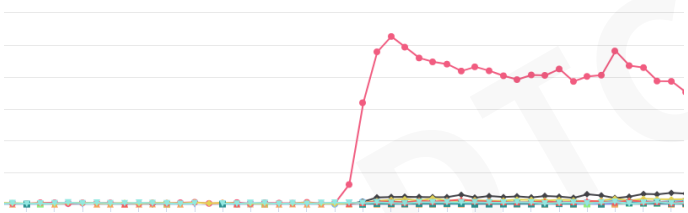
- 异常检测
  - 人工设定固定阈值
    - 针对测量值的 [绝对值/变化量] 设置 [同比/环比] 阈值
    - 需要不断微调阈值
    - 对约 100 指标进行监控
  - 基于机器学习
    - 时间序列预测或检测方法：
      - Median/MAD、p-Quantile、RPCA、GEN-ESD、Holt-Winters、ARIMA、TBATS、SSM、HTM
    - 针对我们的场景：
      - 时域的特征（周期/邻近时段） – Temporal Information
      - 曲线波动与偏移的特征 – Density Estimation、CUSUM
      - 消除异常数据影响 – Holt-Winters
      - 视觉上的特征 – Pattern Recognition
    - 对 3000+ 指标进行监控，整体延迟在 2 分钟

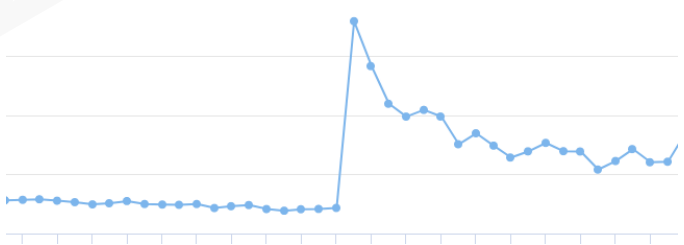
数据挖掘与分析

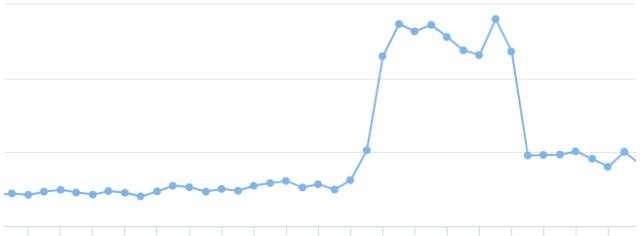
- 异常示例

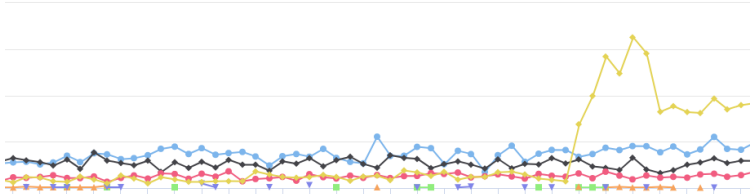












数据挖掘与分析

- 根因分析
  - 基于专家系统
    - 合并：根据疑似异常点的维度条件进行合并
      - 举例：针对运营商与地域合并
    - 归因：在合并后的维度条件基础上进一步寻找异常详细原因
      - 举例：查询增量最大的故障码
    - 补充：提供其他相关信息

- Ad-hoc查询
  - 报表数据（预聚合）的维度通常 ~10 维（维度膨胀）
  - 支持 20+ 维度任意组合的查询
  - 查询结果可视化，便于对查询结果进行分析对比

总结&展望

- 总结
  - 顶层设计非常重要
  - 易用、可靠的平台
  - Kafka + Flink + StreamingSQL + Impala（Kudu/Parquet）
  - 基于机器学习的异常检测
  - 基于专家系统的根因分析
  - 数据查询引擎（报表数据、Ad-hoc 查询）
- 未来工作
  - StreamingSQL 支持输出数据源 Kafka、ES
  - Kafka 1.x +
  - 增强数据质量监控
  - 考虑不同业务特点的异常检测
  - 关联查询

THANKS


HANA
Hadoop
SQL
MySQL
DB2
BigData
Spark
SQL Server
NewSQL
Oracle
MPP
NoSQL
OLAP

讲师申请
联系电话（微信号）：18612470168

关注“ITPUB”更多
技术干货等你来拿~

与百度外卖、京东、魅族等先后合作系列分享活动

微学堂

让学习更简单

微学堂是以ChinaUnix、ITPUB所组建的微信群为载体，定期邀请嘉宾对热点话题、技术难题、新产品发布等进行移动端的在线直播活动。

截至目前，累计举办活动期数60+，参与人次40000+。

ITPUB学院

ITPUB学院是盛拓传媒IT168企业事业部（ITPUB）旗下
企业级在线学习咨询平台
历经18年技术社区平台发展
汇聚5000万技术用户
紧随企业一线IT技术需求
打造全方式技术培训与技术咨询服务
提供包括企业应用方案培训咨询（包括企业内训）
个人实战技能培训（包括认证培训）
在内的全方位IT技术培训咨询服务

ITPUB学院讲师均来自于企业
一些工程师、架构师、技术经理和CTO
大会演讲专家1800+
社区版主和博客专家500+

培训特色

无限次免费播放
随时随地在线观看
碎片化时间集中学习
聚焦知识点详细解读
讲师在线答疑
强大的技术人脉圈

八大课程体系

基础架构设计与建设
大数据平台
应用架构设计与开发
系统运维与数据库
传统企业数字化转型
人工智能
区块链
移动开发与SEO

联系我们

联系人：黄老师
电　话：010-59127187
邮　箱：edu@itpub.net
网　址：edu.itpub.net
培训微信号：18500940168