

开发者社区 ×

飞天大数据平台
APSARA BIG DATA PLATFORM

飞天AI平台
APSARA AI PLATFORM

领军行业大数据 及AI实战

解锁九大行业领军企业
云上大数据及AI实践

BIG DATA



深度剖析大数据在直播、多媒体、新零售、物联网、金融科技、
社交、家居服务、互联网、泛娱乐9个行业实践场景
TOP企业案例真实讲解 助你速懂企业大数据实践

阿里云



阿里技术

扫一扫二维码图案，关注我吧



阿里云实时计算



MaxCompute 开发者交流钉钉群



DataWorks 开发者交流钉钉群



扫码关注阿里技术

目录

直播行业	混合云模式下 MaxCompute + Hadoop 混搭大数据架构实践	4
多媒体行业	基于 MaxCompute 的媒体大数据开放平台建设	10
	基于实时计算 (flink) 打造舆情分析平台——新华智云	16
新零售行业	MaxCompute 助力衣二三构建智能化运营工具	22
物联网行业	塑云科技：性能突破，基于 KafKa+OTS+MaxCompute 完成了一次物联网系统技术重构	29
金融科技行业	交易数据清算从 8 小时缩至 1.5 小时，飞天大数据平台 MaxCompute 解决余额宝算力难题	33
社交行业	小打卡：基于 MaxCompute+PAI 的推荐算法实践	37
	小打卡：快速部署大数据计算平台实践	42
	基于 MaxCompute 搭建社交好友推荐系统	49
	发光的二次元——克拉克拉上云实践	58
家居服务行业	万师傅大数据体系实践案例	68
互联网行业	基于实时计算 (Flink) 的商场实时客流分析系统——上海鸥新	74
泛娱乐行业	基于 MaxCompute 构建 Noxmobi 全球化精准营销系统	78

混合云模式下 MaxCompute + Hadoop 混搭 大数据架构实践

张龙 斗鱼 大数据高级专家

简介: 2019 杭州云栖大会大数据企业级服务专场，由斗鱼大数据高级专家张龙带来以“混合云模式下 MaxCompute+Hadoop 混搭大数据架构实践”为题的演讲。本文讲述了从 Apache Hadoop 阶段到 Cloudera CDH 阶段斗鱼大数据架构的发展历程。提出了上云过程中斗鱼遇到的问题和挑战，包括数据安全、数据同步以及迁移任务。概括了混合云模式给斗鱼带来资源效率更高和资源成本更低的变化。

斗鱼大数据架构发展历程



在 2014 年中期，斗鱼就开始使用大数据，最开始使用的是简单的 HBase 和 Hadoop。在 2015 年，开始使用 CDH 运维大数据集群，主要针对可视化运维。在 2017 年的下半年，斗鱼开始接触阿里云大数据的一些产品，并且与其他产品做了对比。最终选择了阿里云的 MaxCompute。

Apache Hadoop 阶段

由于业务场景比较简单，组件较少，并且使用的人也少，但可以灵活的操作，同

时集群规模较小，运维要求低，可以自由的利用开源，培养了许多人才。但在发展过程中也遇到了一些阻碍，例如：组件增多，运维成本高，业务增长快，集群扩容操作繁琐，人员增加，数据安全要求高，物理机操作，环境安全难保障。

Cloudera CDH 阶段

斗鱼为何选择 Cloudera CDH？原因主要有：首先，它能满足业务发展需要，多组件运维成本低，集群扩容操作简单，数据安全及环境安全有保障。其次，CDH 在国内被广泛使用。最主要的一点是斗鱼的团队内部有 CDH 人才。

Cloudera CDH 给斗鱼带来了许多便利，包括支持丰富的组件，不用考虑兼容性，可以通过 CM 统一管理，进行 Web 化管理，同时支持中文。另外，支持安全管理，以及对 Kerberos 安全认证。

自建集群遇到了发展瓶颈，涉及到资源效率问题和资源成本问题。资源效率问题包括资源预算审批慢、机器采购周期长以及机房部署效率低。资源成本问题包括机器资源成本高、机房成本高还不稳定以及闲时资源空置较多。



大数据上云的挑战

上云面临的挑战主要是如何保证数据安全，因为数据是企业核心的资源，安全性

是非常关键的。其次是如何保持数据同步，是因为云上云下存在着海量数据。最后，因为云下存在大量的历史业务，那该如何将业务安全迁移到云上也是一个问题。

• 如何保证数据安全？

对于数据丢失的问题，阿里使用原始数据进行备份，这是很关键的。对于核心数据泄露问题，几率是很小的，因为泄露数据之后所要承担的风险远大于打败竞争对手所提供的收益。对于云环境面向外网，如何保证安全访问的问题，可以增加账号访问 IP 白名单及审计，设置公司内部才可访问。

• 如何保持数据同步？

由于每天会产生 PB 级历史数据和 TB 级数据增量。如何快速准确同步数据问题，可以使用数据同步工具，主要是基于 DataX 的改造。同时提高网络专线能力，增加多根专线，自动地进行异常切换，与云上平台业务进行隔离。利用数据校验工具，校验数据同步任务以及数据量。

• 如何安全迁移业务？

业务的安全迁移需要做到三个要求：1. 不能引起故障，保证迁移可行性验证。2. 迁移成本不能太高，业务侧尽量少改动。3. 能上云也要能下云，尽量保证云上云下操作一致性。

为了做到不引起故障，要做到三个需要：需要做业务场景测试，保证业务场景全部覆盖到，并且能够识别能够迁移的业务场景。需要数据质量检验，确保相同业务云上云下产出数据的一致性。需要数据效率验证，确保云上任务数据产出时间，同时不影响业务。

• 如何保证较低的迁移成本？

斗鱼在 IDC 中运行的任务主要分两部分，第一部分是 Java 任务，占比很小，特点是基于封装的 HiveClient 工具进行查询计算。第二部分是 XML 配置化任务，特点是基于自定义 XML 文件，支持 HiveSQL 统计后导入其他存储。针对这些任务的特点，斗鱼也做了相应的改造。针对封装 OdpsClient，可以将 HiveClient 改成 Odp-

sClient, 并且改 Hive URL 为云环境。针对加模板改 URL, 可以引入 MaxCompute 参数模型, 改 Hive URL 为云环境。

为了保证能上云也能下云, 第一, 需要数据能上能下, 就是前面提到的数据同步中心。第二, 需要完善的配套工具, 云上云下环境尽量透明化使用。第三, 多使用通用功能, 通过 SQL+UDF 能覆盖大部分场景。

混合云模式带来的变化

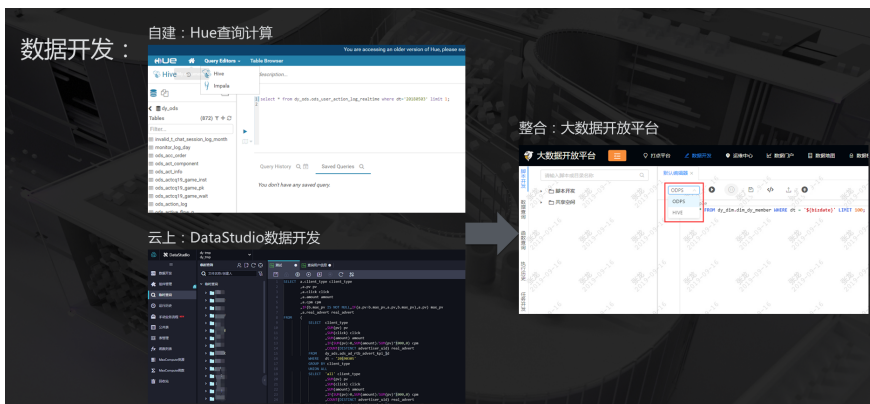


混合云模式带来的变化主要针对资源效率低, 难以跟上业务发展, 以及资源成本高, 企业财务压力大两方面。在资源效率方面, 从自建集群到 MaxCompute 有

一些变化，包括提前半年或一年提预算变成按量付费，采购耗时 1 到 3 个月变成资源可以无限使用，机房上架 1 周以上变为无机房概念。相比于 IDC 自建集群，MaxCompute 每年大概节约 1000w 成本，保障集群零故障。同时也有一些附加的收益，包括阿里云的专业服务，当遇到技术问题时可以请教阿里的专家来帮助解决，以及计算资源可以量化，可以知道钱花在哪些业务了，以及与阿里专家交流，帮助解决业务难题。



在自建机房时，斗鱼也做了一些开发，下图所示为数据开发，包括基于 Hue 的查询计算和云上的 DataStudio 数据开发，然后将 Hue 的 API 和 DataStudio 的 API 集中起来形成斗鱼的大数据开放平台，作用是可以提供给数据部门的人使用，也可以提供给业务部门的分析人员使用。



此外，斗鱼也做了一些实践，称为多活数据中心，如下图所示。斗鱼通过确立自建机房的数据和阿里云数据在这两个数据中心的角色，保证可以在多活数据中心的状态下支撑更多的业务。



混合云带来的变化总结起来，资源成本和资源效率是最大的两个变化，还有可量化的成本、增值服务、额外的专业服务，不仅可以给我们自己部门人员用，还可以给其他业务部门的人来用，并且他们对使用成本也是直接可见的。以上就是我今天的分享，谢谢大家。



基于 MaxCompute 的媒体大数据开放平台建设

李金波 新华智云 首席数据官

摘要：随着自媒体的发展，传统媒体面临着巨大的压力和挑战，新华智云运用大数据和人工智能技术，致力于为媒体行业赋能。通过媒体大数据开放平台，将媒体行业全网数据汇总起来，借助平台数据处理能力和算法能力，将有价值数据内容和能力开放给用户。本文主要从新华智云数芯平台，媒体行业数据特征，批流处理数据架构，以及通用的媒体大数据平台能力等几个方面介绍了如何基于 MaxCompute 做媒体大数据开放平台建设。

本次的分享主要围绕以下五个方面：

- 一、关于新华智云
- 二、数芯 - 媒体大数据开放平台
- 三、媒体大数据平台能力
- 四、Project 依赖
- 五、几点小提示

一、关于新华智云

新华智云是新华社与阿里巴巴合资成立的一家公司，主要面向运用大数据和人工智能技术，致力于为媒体行业赋能。随着自媒体的发展，自媒体公司非常的火热，如今日头条，抖音等。对传统媒体来说，面临着巨大的压力和挑战。传统媒体急切的希望互联网的技术帮助他们赋能。

媒体大脑 - 数芯

媒体大脑是新华智云底层的产品品牌，数芯是媒体大脑基础的数据平台。数芯定

位为媒体大数据开放平台，平台中包含了媒体行业所依赖的各种各样的数据。数芯希望通过平台本身的数据处理能力和算法加工，将挖掘的有价值数据内容和能力开放给上层用户。目前，数芯包含了各种各样的数据，涵盖了400多万互联网站点，主要以中文站点为主，日增7千万文章，包含微信公众号，微博账号，图片以及多媒体来源等。将不同来源存储在平台中，之后对接数据，进行更多的加工和运用。“数据已经不再是一种成本，而是成为一种生产资料”，这句话在媒体行业更能够有所感受。一篇新闻除了是新闻之外，还是一条数据，同时新闻本身又会衍化产生新闻。很多媒体公司利用互联网技术获得大量的新闻内容之后，辅助他们新闻的生成过程。数芯会将媒体数据从不同的源头，不同供应商，以不同内容格式存储到平台中，进行数据清洗，结构化，加工等等一系列的操作之后，生成有价值的内容。如，这段时间之内媒体关注的热点，互联网上发生的事件，同时识别新闻和机构之间的关系，基于实体将各个渠道上的数据整合起来，为用户提供统一的视图。用户可以看到信息的发酵过程，以及信息在什么地方传播。

二、数芯－媒体大数据开放平台

数芯是一个开放的大数据平台，开放的点主要有三个方面。第一，数据的开放。任何数据进到数芯之后，都会在整个大数据处理的链条中做计算，内容结构化，加入标签。同时基于用户感兴趣的数据范围，做标签的特征过滤，筛选用户想要的数。数芯帮助用户了解互联网上的信息，了解互联网上与自身相关的事件。第二，数芯提供智能能力的开放。用户不管怎么努力，不可能获取到互联网上所有的数据；而且不可能每个公司都会自建内容大数据平台，做数据分析，也不可能都配备算法工程师；而且公司的数据处理能力也不是足够强；媒体大数据平台可以帮助用户将与他们有关的数据加工好，通过算法能力获得用户关心的信息。如用户提供一个文章，数芯可以反馈这篇文章与谁相关，相同的文章在什么地方传播，文章由谁编写，发在什么地方等信息。通过开放算法能力，帮助用户做数据能力和算法能力的应用，提供文本反垃圾服务，互联网内容结构化服务，文本内容实体识别服务，文本去重判定服务，图像人物识别服务和图像标签化服务等等。第三，产品能力的开发，包含舆情监测能力，

版权检测，传播分析等能力。

媒体数据特征

媒体大数据有三大特征，第一，媒体数据是非结构化性非常强。与传统行业数仓最大的区别是媒体行业 90% 以上数据都是非结构化的，如文本，图片，视频等等。第二，数据来源多样性。媒体行业数据有多种外部的数据源，它们有不同的数据提供的方式，提供不同的数据能力。所以必须要有强大的数据汇总能力才能将所有数据汇总在一起并服务好下游客户。另外，数据效性要求非常高。媒体行业天然追逐新闻热点，假如知道一个事件是潜在的热点，媒体希望第一时间对热点进行追踪，报道和解读。媒体机构希望通过数芯平台，能够最快的发现互联网上发生的事情，发现热点之后以最快的方式，将结构化后的数据给到用户。

批流处理结合

基于媒体行业数据的特征，数芯平台采取批，流结合的方式解决目前客户和业务场景对数据的要求。批指的是大量的计算，基于平台特征完成复杂的模型，算法训练，长周期计算，文本实体识别，文本挖掘，借助批量的能力解决数据更深度，更大规模的加工。流指的是流式的计算，完成数据的清洗，结构化，轻计算和实时统计。出来一篇新闻，在整个新闻流传的过程中实时的将信息处理好。目前数芯整体的流计算大概需要 300 毫秒，即一篇新闻从源头过来，存储到平台，服务化出去中间需要经过 300 毫秒的时间。

批流结合的数据架构

数据从不同数据源过来，如 API，OTS，ROS，log，file 等，一方面数据要在实时计算当中做 URL 去重，正文结构化，标记来源，垃圾识别，实体识别和轻度的实时统计。另外基于 MaxCompute，利用 DataHub 将数据存储到批处理平台中，由于流计算本身不做持久化存储，所以所有数据都将存储在 MaxCompute 上。数据存储之后做主题建设，关系挖掘，知识图谱计算，算法训练。通过批流结合的处理

方式能够满足客户对数据本身能力的需要。之后，给用户提供搜索能力，大屏能力和BI能力等。

三、媒体大数据平台能力

内容结构化

人们在网页当中看到一篇新闻，而在数据库当中新闻是按字段存储的。比如，分为新闻标题，发布网站，时间，新闻来源，情感等等。平台需要将新闻信息进行结构化，成为后续计算过程需要依赖的数据结构化字段。

主题构建

媒体行业会分主题建设数据。平台会获取到不同的数据源，不同类型的数据。这些数据不可能完全的结合在一起，数芯平台会将所有数据分门别类的分成不同主题，按不同主题建设，存放和加工。媒体是一个非常复杂的行业，对各个行业的数据都有所诉求，媒体需要挖掘很多不同行业的数据支撑新闻生产和报道。目前，数芯集中在媒体内在的数据，体育，金融，气象等几个主题。数芯一方面将不同的数据源汇总到平台当中，另一方面数据进来之后，挖掘潜在的新闻点，生成选题策划，帮助用户做选题等工作。

实体识别

实体识别对于媒体大数据来说是最基础的能力。数芯目前积累的实体围绕人，机构，地点等三类数据。新闻行业，媒体行业会围绕某一个实体，关注与实体相关的数据能力。比如，很多公司会关注和自己相关的咨询，关心这些新闻是正面还是负面，哪些机构会发正面的信息，哪些机构会发负面的信息等。只有当将大批量的数据汇集起来之后，才能做相关内容的分析。实体识别场景在媒体大数据领域是非常基础的能力，首先，建立实体库。同时，当有一篇新闻产生，数芯需要实时的识别新闻与哪些人，机构和地点相关。另外，数芯需要汇集实体之间关系，做实体关系图谱。比如，

很多品牌会瞄准自己的竞争对手，调整品牌战略，实体关系图谱对很多公司品牌运营的推广有很大的帮助。

情感分析

情感分析也是媒体大数据平台中比较通用的能力，当一篇新闻出来之后，用户需要知道是情感上褒义还是贬义。信息数量少的信息可以人工判断，但是如果每天有上千篇，上完篇的内容就无法通过人工来判断。媒体行业的情感分析与学术上的情感分析有差异。目前，自媒体出来之后短文本的内容越来越多。短文本的情感分析和长文本的情感分析不同，以往都是采用同一个算法实现情感分析，但发现效果并不好。现在，数芯将情感分析场景分开，微博短文本的情感分析用 Word2vec+LSTM，新闻类的长文本的情感分析用 Word2vec+CNN+RNN。分开之后发现每一类情感分析的效果都有所提升。

内容去重

内容去重是媒体大数据平台非常重要的一部分。去重能力是对于常见的新闻摘录，编辑，删减有准确的判重能力。一篇新闻并不是只由一个人写，它会被很多机构和渠道所转发。如何知道一篇新闻在哪些渠道被转发，其实就是通过去重的方式实现的。通过从大量的渠道汇集数据后，平台需要比对一篇新闻与之前哪篇新闻相似，通过相似度的比对得到结果。最早的时候，去重是基于关键词进行比对，数芯采取关键词和语义，两种方式比对，去重效果明显提升。内容去重可以用于新闻的热度计算，新闻数据顾虑清洗，文章版权追踪等业务场景。

内容标签化

搜索引擎可以用来搜索新闻，通过关键词和文章的匹配度来决定这篇内容是否推荐给用户。但纯搜索的方式已经满足不了用户家的需求。今日头条之所以成功，是因为它基于新闻和用户的习惯推荐内容。内容标签化就是通过机器的方式理解新闻，理解新闻与哪些信息相关，基于文本挖掘的手段，实现对于全网采集的内容数据进行分类打标。

四、Project 依赖

很多时候，使用大数据平台要不要分项目都是一个很难的抉择。不分项目的好处是开发人员都在同一个平台工作，互相之间不需要太多的授权，整体工作效率会比较高。分项目的好处是利用不同平台做不同的业务会更清晰，更具条理性。数芯在开始使用 MaxCompute 时，便采取分项目方式，其考虑的原因有以下三点。首先，分项目可以区分业务优先级，避免低优先级任务影响高优先级的数据产出。另外，可以区分资源消耗型，避免出现高资源消耗任务整体影响数据产出。还有区分内外服务，避免内部服务互相交叉影响。总体上，分项目可以为数据产出的稳定性提供很好的保障。

五、几点小提示

首先，由于媒体行业大部分数据都是非结构化的数据，会造成对单字段的容量要求比较大的问题。而且不同的平台和传输工具对于数据的字段大小的限制不同。在不同平台做数据传输时，尤其需要关注这个问题。

第二，能用 UDF 解决的问题，不要使用 MR。使用 UDF 可以提高开发和运维的效率。即尽量用简单的表达式处理逻辑，这对整体数据产出稳定性有好处。

第三，对查询效率要求不高的数据报表可以直连 MaxCompute，减少中间环节。如此可以大大减少数据转换和数据维护成本。

第四，Datahub 一方面可以接数据源，另外还可以较好的串联批，流之间的计算流程，保持数据一致性和形成依赖。

第五，合理设计批，流的数据处理分工，减少重复计算。

第六，媒体大数据经常需要运用不同的算法，PAI 可以帮助解决很多算法问题，减少开发的工作量，提高数据处理的效率。

事件跟踪

美国能源信息署周三公布陈述显现,上星期美国原油库存降幅超过预期

首发时间 2019-01-14 11:21

取消跟踪

48

当前热度指数

2

平均热度指数

21

情感指数

48

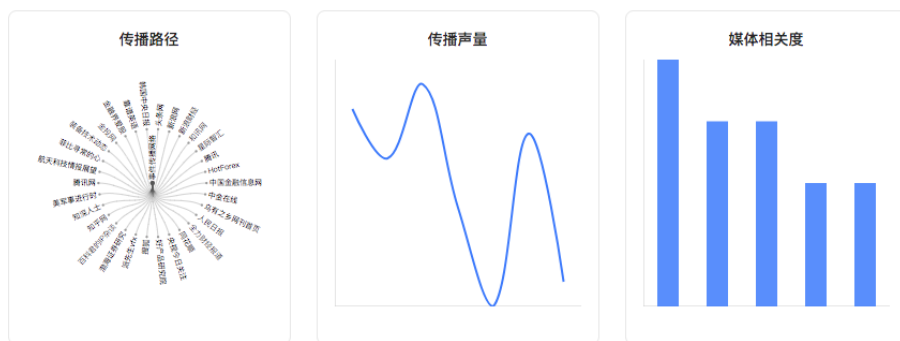
相关文章数

2

今日文章数

热点分析

热点新闻



系统设计

网络舆情是社会舆论的一种表现形式，通俗的说是人们通过网络表达对某些社会事件的看法和态度。网络舆情以事件为载体，以事件为核心，是广大网民情感、态度、观点的表达、传播、互动以及后续影响力的总和。

一个舆情分析系统，主要解决的问题包括：发现事件、跟踪事件、发掘观点、评估影响力等。

系统挑战

- 舆情分析系统需要对接多个上下游。

- 数据上游：爬虫数据和采购数据。爬虫采集不同数据结构类型的平台数据。
- 数据下游：写入不同的存储系统。如统计结果写入 RDS，清洗后数据写到 OTS 中，文章存储到 ES 中做索引等等。

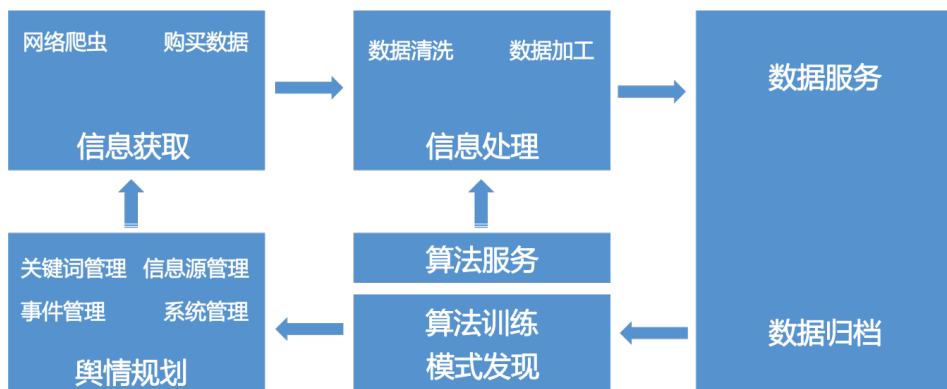
如果设计不合理，多上下游将会极大影响系统复杂度。

- 数据处理过程包括数据清洗和数据统计
 - 清洗过程既有基于规则的信息抽取，又有基于算法的实体识别。
 - 统计过程需要先把数据结构化拼装，然后按照各种维度进行灵活的统计。
- 具有自我迭代的能力

设计良好的舆情分析系统应该具有自我迭代的能力，能够根据历史数据进行优化，不断提供系统效能。

系统架构

大数据舆情分析系统架构图如下。



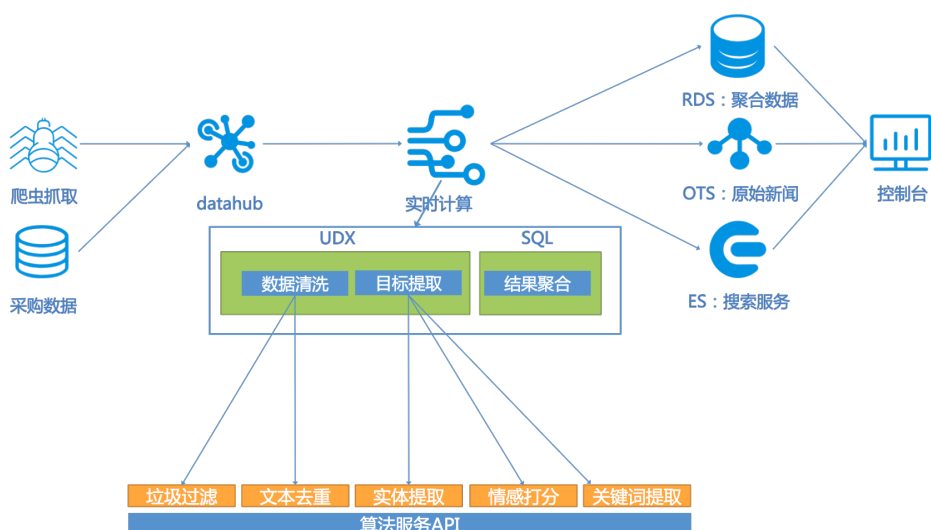
- 舆情规划是整个系统的控制器，控制着舆情发现的目标、信息源等；
- 信息获取是舆情分析的基础，能够快速、准确的获取足量信息是系统成功的前提；
- 信息处理与算法部分是舆情分析的关键，担负着把原始数据加工成信息与知识

的重任，并且对归档后的数据进一步分析能够发现系统问题、发现新的事件和新的模型，能够进一步提升系统效果；

- 最终的信息要服务的形式的暴露出来，为人所用；

信息处理

数芯平台的信息处理部分如下图所示。



- 数据来源（注：数据采集成功后统一发布到 DataHub 中）
 - 爬虫：实时的去各大网站、微博、微信等内容平台抓取数据。
 - 从其他机构采集数据。
- 数据处理

实时计算订阅 DataHub 然后进行数据处理。数据处理包括两个重要的部分：实时数据流和算法服务 API。
- 实时数据流

通过实时计算串起整个实时数据流，总体包括对数据进行清洗、提取目标，整理成结构化的数据，然后按照所需维度对结构化的数据进行聚合。并且要把原

始数据、结构化的数据、汇总的结果都发布到下游存储系统中。

- 数据清洗：对爬虫抓取的数据进行清洗，比如垃圾过滤、文本去重等。这一步通过实时计算提供的 UDX（自定义函数）来调用算法服务 API 来完成；
- 目标提取：从已经完成清洗的数据中抽取感兴趣的目标，比如抽取实体（人物、地点、事件等），对内容的情感评分，抽取文章关键词等，这一步的目标是从非结构化的文本中抽取结构化的信息；
- 结果聚合：对已经结构化的数据按维度聚合，比如某事件的按天计数，某新闻的传播热度等。最终把这些聚合信息联合原始信息分别写到下游存储中去。

- 算法服务 API

预先训练好模型，把垃圾过滤、文本去重、实体提取、情感打分、关键词提取等暴露成 API 服务，供实时计算调用。

- 数据存储与服务

RDS 保存聚合数据，OTS 保存原始新闻，ES 对新闻建立索引，提供搜索服务，这三个数据存储直接为最终产品服务。

总结

数芯之前数据处理部分使用的是自建的 spark，需要自行运维和对接各种上下游系统，迁移到了阿里云实时计算平台，整体收益包括：

- 运维成本：免运维，阿里云提供高保障。
- 对接上下游：直接注册，免开发。
- 开发成本：SQL 开发，效率高，门槛低。
- 数据流：一个产品串起整个数据流，ETL 用 UDX，统计用 SQL。

从更高的维度上看，这个案例属于实时 ETL 场景，实时 ETL 的目标把数据从

a 投递到 b，中间进行清洗、格式转化、信息抽取等。如果对吞吐、实时性有一定要求，可以在方案阶段优先考虑实时计算产品。

注：本文部分内容来自新华智云工程师杨丛聿的分享，特此感谢。

如果您有需求，欢迎联系付空。

MaxCompute 助力衣二三构建智能化运营工具

程异丁 衣二三 CTO

摘要：本文由衣二三 CTO 程异丁为大家讲解了如何基于 MaxCompute 构建智能化运营工具。

衣二三作为亚洲最大的共享时装平台，MaxCompute 是如何帮助它解决数据提取速度慢、数据口径差异等问题呢？程异丁通过衣二三数据体系架构，从用户运营应用、商品运营应用以及算法推荐系统三方面给大家剖析了 MaxCompute 是如何助力衣二三构建智能化运营工具的。

衣二三是谁？

衣二三是亚洲最大的共享时装平台。衣二三提供女性服饰包月租赁制服务，会员们在 APP 上挑选喜爱的衣服，可以在平台上用固定的月费在一整月内不断地换穿衣服，衣二三通过快递将衣服送到客户手中，客户只要会员期有效，就可以一直穿这件衣服，当会员不想穿的时候，衣二三还会通过快递按照约定的时间地点将衣服收回，这样会员就可以继续下新的订单，可以不断地换穿各种衣服。我们的客户从没有毕业的学生到工作多年的白领，多种类型的客户都能从 APP 上找到合适自己的衣服。



如今，电子商务发展非常成熟，线下逛街也是大家喜欢的生活方式，那为什么在这种情况下还要租衣服呢？衣二三为客户解决了以下几个痛点：

1. 打开衣橱总是找不到今天想穿的衣服。使用衣二三，用户可以灵活的换装，不断地尝试不同风格；
2. 费用高。衣二三标准的月费只有 499 元，相当于买一件衣服的价格，用户可以用买一件衣服的价格换穿一整月的不同服装。
3. 大城市的房屋空间有限，拥有一个衣橱对于很多用户来说更是奢望。衣二三可以回收衣服，帮助用户节约空间，同时还负责清洗衣服，正如宣传语所说的那样——“你负责貌美如花，我负责清洗收发”；
4. 卫生问题。衣二三联合福奈特、天天洗衣等共同打造了智能化现代化的洗衣工厂，每一件衣服需要经过多达 16 道清洗消毒流程，比自己家里洗衣服还要干净；
5. 购买和退换问题。衣二三通过体验式的方式帮助用户从款式、尺码等方面找到真正适合自己的衣服，并且允许用户买下来。

为什么使用 MaxCompute？

衣二三的前后端运营比较复杂，前端需要让客户在尽可能短的时间认识我们，了解我们，才能形成转化；后端也是特别复杂，与一般的电商相比，衣二三大有明显的不同，常规电商的客户在前端下好单后，会通过订单管理系统，将订单派发到仓储中心，仓储中心会对订单进行拣货、配货、包装、物流等，最终将商品送到客户手中，也可能伴随着一小部分的退货或返修，对于一般电商来讲，整个流程到这里就基本结束了，但是对于衣二三来说，该流程只进行了一半，我们需要对所有的衣服进行回收、清洗和质检，才能够再次上架让客户挑选，如此复杂的运营自然离不开大数据的支持，MaxCompute 作为整个数据体系的核心，帮助了衣二三很多。

那么，在未使用 MaxCompute 前我们遇到了哪些问题呢？主要包括以下几方面：

- 数据提取速度慢。随着系统不断地变复杂，SQL 越跑越慢。
- 数据口径差异。每个工程师将逻辑都写在 SQL 里，不同工程师得到的数据结果可能不一样，这就是因为我们没有集中化标准化的数据仓库对每一个数据维

度和业务的指标有相应明确的定义，造成所有的逻辑都在 SQL 中。

- Python 脚本实现基础 ETL。作业调度与依赖于维护与迭代，运维也是比较令人头疼的，修复时间长。
- 人工数据提取及报表制作。工程师忙于提供各个业务部门的数据支持。

这些问题促使我们想要做出改变，起初我们想做一套完整的 hadoop 全家桶进行替代，但评估后发现其运维代价和资源消耗对于我们的数据团队和运维团队来说，都是比较大的负担，而且很难快速体现价值。后来，我们很幸运的遇到了 MaxCompute，从试用开始，我们逐渐把数据仓库和数据体系建构在 MaxCompute 上。

MaxCompute 有哪些好处呢？主要从以下几个维度来分析：

1. MaxCompute 是基于云端的大数据仓库，无需复杂作业运维工作
2. 数据吞吐量大，查询性能好，支持 UDF
3. 可视化任务编辑界面，易于上手
4. 使用成本低，适合初创公司
5. 与阿里云大数据产品生态融为一体，比如 RDS，应用层的 Quick BI，Blink 以及日志服务。

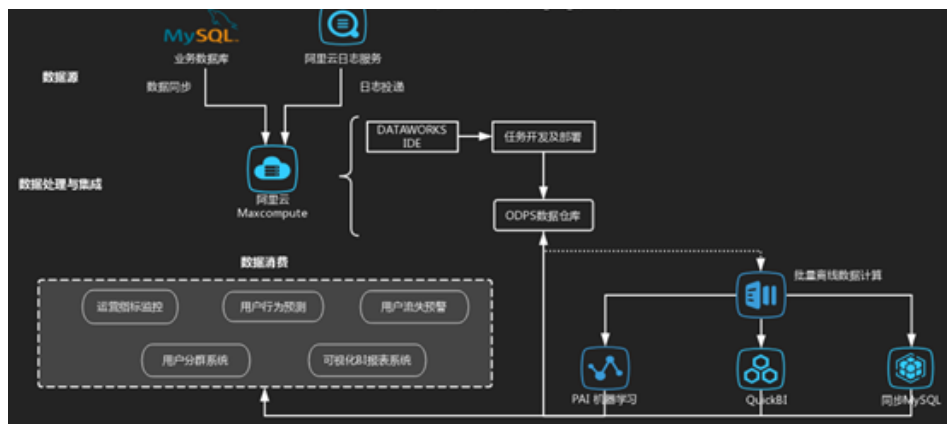
如何应用 MaxCompute？



衣二三数据体系架构如图所示，最底层是数据源，包括 RDS 生产数据库和日志服务等，通过日志服务、DataHub 等投递到数据计算层；数据计算层以 Max-Compute 为核心，同时配合我们自己做的脚本和 UDF 进行数据存储和计算，生成的结果反馈到前端数据应用层；数据应用层是数据分析展示工具，包括 YConsole、QuickBI 以及 DataV 等，我们使用 DataV 做展示业务全局的看板，让我们快速了解全国各地谁在下单、仓储中心的储备情况等。

我们还有包括用户画像、商品画像、爆款识别、衣二三指数和实时人效监控等。

用户运营应用

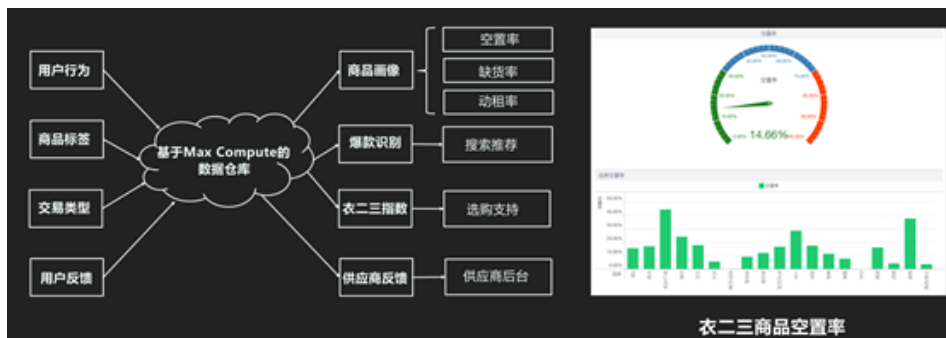


用户运营对于互联网公司来讲，都是极其重要的，它有两个业务指标，一是转化，一是留存。转化是指尽快让客户了解我们的价值，同时形成付费和转化；留存是指让已经成为我们用户的用户在平台上得到好的体验，让用户喜欢这种生活方式并留下来。

我们结合了 MySQL 数据和用户日志放到 MaxCompute 中，经过 PAI 平台，我们自己做了数据分析，包括使用数据挖掘算法和随机森林等的的数据维度分析，对我们整个用户运营做了很多量化指标，我们通过邮件日报对运营指标监控，基于这些指标我们开发了自有用户分群体系推进用户精细化运营；基于阿里云机器学习计算平台

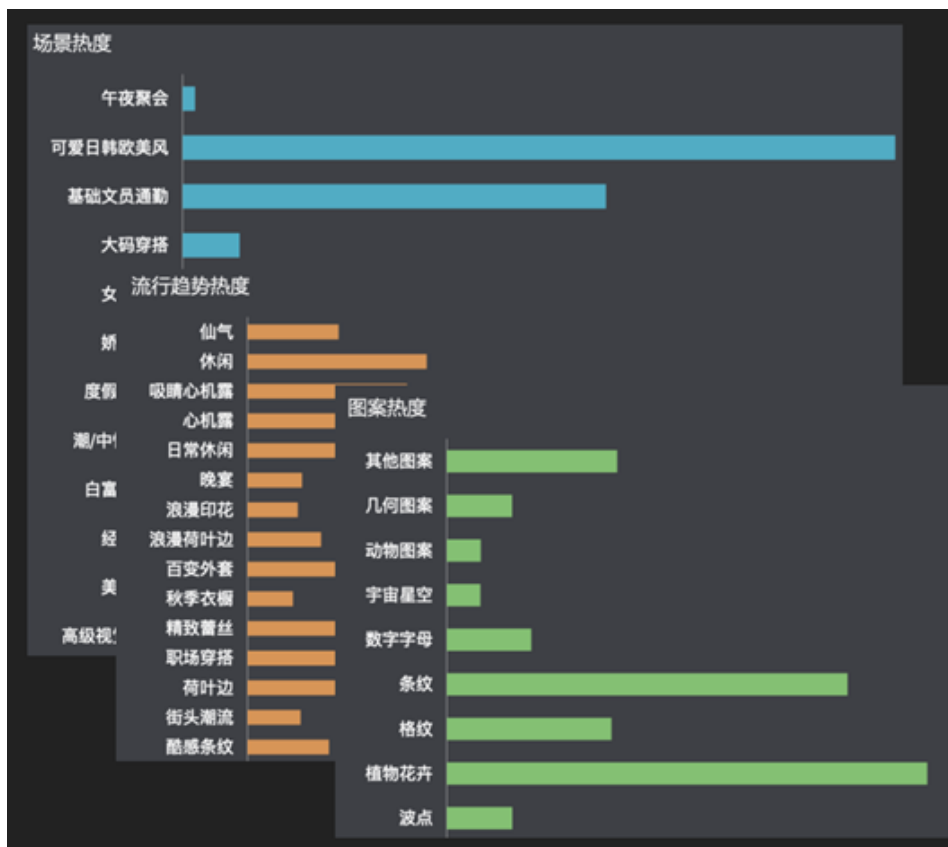
PAI，我们建立了预测模型，预测用户的转化率等；引入流失预警计算体系后，运营根据用户的流失指数进行精准挽留营销，将会员流失降低了超过 50%。

商品运营应用



衣二三的每一个商品都有结构化的数据，我们专业的买手会给商品打上 20+ 维度的标签。此外，我们会总结用户行为，用户与商品之间的互动都放在 MaxCompute 中做相关性分析，形成商品的一系列指标，比如爆款识别，通过爆款识别去预测满足哪些属性和维度的衣服会形成爆款，我们的买手会带着数据工具一起进行采购，极大的提高了运营效率。

我们还结合商品各个表现维度，包括库存深度、出租情况等计算出衣二三指数，它可以对商品进行排名。除了给合作伙伴应有的租赁收益外，我们还会提供数据工具，最大化业务模式的价值。



商品的灵魂就是标签，只要将标签做的足够细，才能从结构化上去理解它，去做一些预测性的指标。我们通过商品不同的标签及风格场景热度指数进行备货指导，通过最终商品空置率进行商品调拨及库存关联，通过衣二三指数反映商品热度排行，根据用户行为促进租售转化，完成商品流动闭环。

算法推荐系统

如何让用户在很短的时间内找到适合一个月穿的衣服呢？这就需要推荐算法的支持。



衣二三的推荐算法也是基于 MaxCompute，把用户对商品的行为通过日志收集到 MaxCompute 中，对用户进行用户画像，对用户画像进行模型训练，最终落地到商品列表展示给用户。使用推荐系统给我们的业务带来很大提升，选衣页推荐的点击率增加了 70%，人均点击数增加了 50%；相关单品推荐的点击率增加了 150%，人均点击率增加了 110%。

总结下来，大数据的魅力大家已经认识到，MaxCompute 这样的产品对于初创公司来讲将大数据门槛降的非常低，让大数据能够面向各种各样的公司来贡献力量和发挥价值。谢谢在场各位的倾听，以上就是我的分享。

塑云科技：性能突破，基于 Kafka+OTS+MaxCompute 完成了一次物联网系统技术重构

武良军 塑云科技 CTO

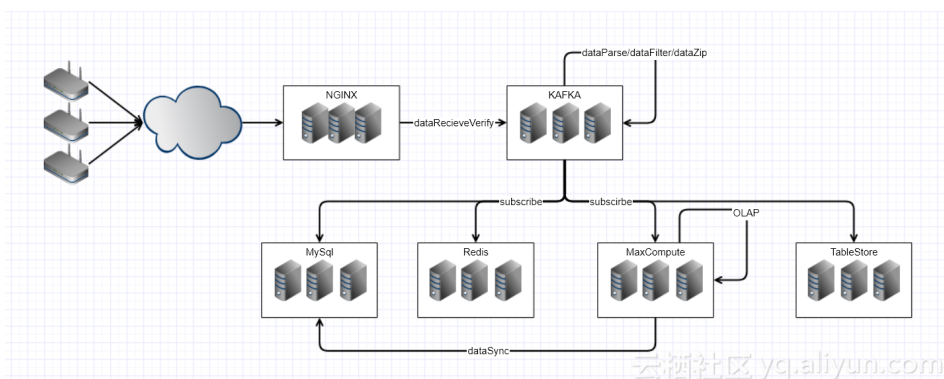
背景：创业团队，专注于氢能燃料电池生态链的运营支撑，当前主要的业务组成为新能源车整车实时运营监控分析，加氢站实时运营监控分析，车辆安全运营支撑。

系统面临的主要挑战：高频数据的实时解析、存储、分析。拿整车实时运营监控分析来讲，每辆车以每秒 1K 的原始报文上报，要求做到秒级延迟的解析应答以及入库。同时需要针对解析后的每车每秒 33K 的报文进行快速查询以及后继的分析。考虑到未来车辆接入的量，需要在考虑性能的基础上以最经济的方式进行系统设计。按照每车每秒 33K 的解析后报文，每车每月预计生成 30G 的报文数据（车辆按照每天运行 10 小时计算）。

原有系统存在的问题如下（罗列部分）：

1. 系统架构中未对 OLAP 和 OLTP 系统的范围进行清晰界定，使用 JAVA 程序对 OTS 的表定时进行任务统计，代码复杂并且性能极差并且影响到服务器上其他 OLTP 系统的正常运行。
2. 存储的解析后的报文数据，未针对 OTS 的计价规则进行针对性优化，一个大 JSON 串中冗余的 KEY 过多，KEY 的长度超长（平均 30 个字符串）。
3. OTS（阿里云 tablestore）按照公司进行分表设计，存在单个实例下表数量超过 OTS 限制（64 表）的风险。
4. OTS 以车月作为分区键，单个分区（30G）过大，超过 OTS 建议的 1G 推荐大小。
5. OTS 单车的分区连续分布未做散列，不能在物理机器层面最优并发性能。
6. 没有针对最核心的读取场景（按天按车查询报文）进行编码层面的优化。

在做系统优化之前，首先要做的就是架构层面的梳理，对产品中需要使用到的中间件产品的适用范围进行了明确的界定。数据在各个环节的流转进行明确的定义如下：



这里主要的改进：

一、引入 KAFKA 作为多个环节异步解耦的基础支撑，提升对终端的报文快速回复。

二、引入 MaxCompute 作为 OLAP 系统的基础支撑。将复杂的业务分析转交给 MaxCompute 来处理。

三、针对 OTS 的计价原则，对 OTS 的模型进行了重构（此文暂不讨论）

MaxCompute 作为阿里云强大的数据分析利器，因为之前的经历相对比较熟悉。所以在这次的改造中特别针对性能、成本、可运维等方面做了较多的思考。

这里首先讲一讲基于成本的考虑。首先根据数据的使用频度将数据切分为在线、离线、归档三类。车辆终端上报的报文数据作为归档数据选择 OSS 的归档存储。在线数据设定 N 月的生命周期，主要包括报文解析之后需要实时查询的数据，离线数据主要包括基于解析的报文数据进行离线分析统计之后形成的各类中间结果、报表数据。

针对数据的使用场景界定数据类型之后，这里主要考虑离线数据使用 OSS 还是 MaxCompute (ODPS) 或者是 OTS 来存储的问题。根据三类产品的存储计算成本我做了一个粗略对比如下：

车辆数	存储月份	车每月数据 (G)	存储类型	热数据月份	存储费用	计算费用
5000	36	10	OTS	36	7,159,500	216,000
5000	36	10	OTS+ODPS	1	1,417,968	待定
5000	36	10	OTS+OSS	1	1,460,925	待定

这里已经考虑通过压缩的方式存储 OTS 减少计价存储的情况。当然 MaxCompute 的计价是按照实际压缩存储之后的容量计算。MaxCompute 官方文档介绍的是 5:1 的压缩比，而我们的数据因为本身的特点，实测可以到 7~8 : 1 的压缩比，所以最后数据方案反倒是 MaxCompute 直接存储离线数据性价比最高。同时也符合数据靠近计算的原则。

经过测试使用 OTS 外部表作为数据载体的计算性能一般（当前 MaxCompute 对 OTS 的外部表的 Map Reduce 计算直觉觉得是基于 OTS 的分片，并且缺少分区概念，每次都是基于全表扫描，这点可以从 MaxCompute 的任务详情可以观测出来）。

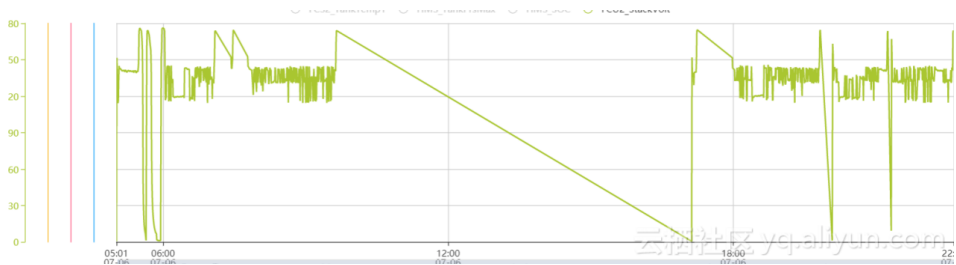
技术选型确定以后，剩下的是如何利用 MaxCompute 为业务提供可靠、稳定数据服务。这里特别需要强调的是数仓的建模、数据集成、工作运维的使用。

数据集成主要这方面主要体现 MYSQL 跟 MaxCompute 的双向同步，这个不需要特别讲，主要是设计上需要考虑到数据的重复同步的设计即可。关于工作运维则是更多地体现在对任务的运行状况的监控以及重跑的支持。

数仓的建模主要考虑的还是成本和模型的复用。首先针对海量、质量不高的底层数据进行分层建模。保证上层的业务模型只依赖中间结果。这里带来的直接效益就是计算成本的大幅下降（每每看到有些开发同事动不动就对着一个上百 G 的原始表做各种查询的时候，心是痛的…）。其次是中间模型为系统补数带来更快的性能，毕竟因为一些业务或者数据的原因需要重跑部分报表，这个时候如果需要重新扫描原始数据

的时候，首先就是费钱，非常费钱。其次就是耗时，非常耗时。

在离线统计分析的重构完成之后，系统充分利用 MaxCompute 的并行计算能力，并且借助其强大的函数尤其是窗口函数的支持，我们实现比较不错的分析能力，客户的一个核心部件的数据统计分析，之前一个专业的工作人员分析一个部分需要耗时一天，还容易出错。借助平台的分析能力，可以在 10 分钟内计算完将近 1000 个部件的数据分析工作。类似下面的曲线图分析每次数据波动期间的均值，之前几乎无法人工计算，即便是 JAVA 编码也是一个非常复杂的编码工作，通过平台的支持，系统处理得游刃有余。



一次计流水账式的总结，且当做一次经验的沉淀。

交易数据清算从 8 小时缩至 1.5 小时，飞天大数据平台 MaxCompute 解决余额宝算力难题

郑林贵 天弘基金 大数据技术总监

天弘基金作为国内总规模最大的公募基金，阿里云 MaxCompute 为我们构建了企业级一站式大数据解决方案。MaxCompute 对于海量数据的存储、运维、计算能力强大且安全稳定，MaxCompute 服务将原本需要清算 8 小时的用户交易数据缩短至清算 1 个半小时，同时减少了本地服务器部署压力，在显著提升我们工作效率的同时减少了大量开发成本和人力成本，使我们能更专注于业务发展，为用户提供高品质、高价值的金融服务。

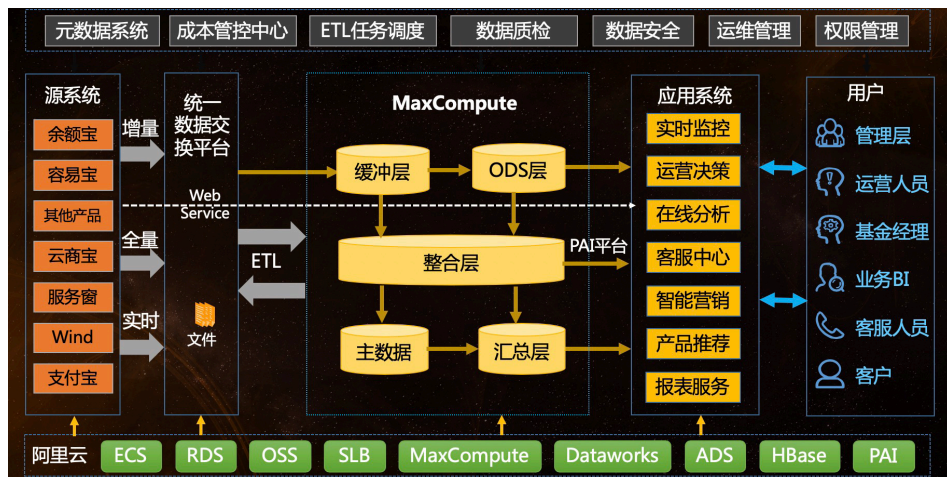
背景：

随着余额宝用户数持续呈指数级增长，数据量也成倍增长。在这种情况下，已经无法通过简单的 hadoop 集群进行数据的管理工作，而业务端面临需要通过数据了解用户、分析行为进而对业务决策和用户行为进行精准预测。基于这些业务的需求驱动需要一个大数据平台来承载，我们在对稳定性、成本、自身能力和复杂度等进行综合考量后，决定采用当前最流行和最成熟的云平台·阿里云 MaxCompute。

目标：

搭建大数据平台从技术指标的角度是数据存储和数据计算两大目标，而从各个环节的角度看是数据采集、数据清洗、在线 / 离线分析与预测、实时 / 非实时查询。而业务目标是为了能够快速响应业务需求，能够为业务分析提供稳定的开发和建模平台，为业务提供逻辑清晰和灵活便捷的可视化平台。从而实现从数据支持业务到数据驱动业务的逐步升级。

解决方案及架构：



整个架构都是搭建在阿里云上的，该架构是成熟的三层架构：采集层 + 整合层 + 应用层。

采集层：

采集层对接了我们几乎所有的业务，采集数据的频率有实时的、分钟级、小时级、日级、月级，支持不同的采集频率，而且这些都是灵活可配置的。将采集的数据通过企业级的数据交换平台进行存储和交换，该平台使用 OSS 实现。通过 OSS 可以实现数据的中转、分发和备份存储。

整合层：

在 MaxCompute 整个整合层包含了五大区：缓冲区、ODS 区、整合区、主数据和汇总区。不同的区域为了实现不同的功能，缓冲区是为了在正式进入数仓应用数据模块之前进行数据质检，满足质检后方可进行真正的加工处理，避免因数据错误污染整个数仓的数据；ODS 区是为了保留源系统格式的数据模块，一方面能够在有问题时追根溯源，另一方面能够满足部分业务的需要；整合区是数据仓库的核心区

域，通过主题建模的方式进行数据的模型化处理，使得数据的解释口径具有统一性；主数据则是与业务结合比较紧密的主题数据，这样更方便业务方的使用；汇总区则是提前将需要预加工统计的数据进行统计计算，避免多次开发计算带来的时间成本、开发成本和计算成本等。

应用层：

应用层主要是通过监控、管理看板、报表等可视化系统给业务提供直观的数据呈现，从而为业务的决策提供更加有力的数据支撑。在应用层通过 RDS、ADS、HBase 等不同的产品满足了不同的需求。

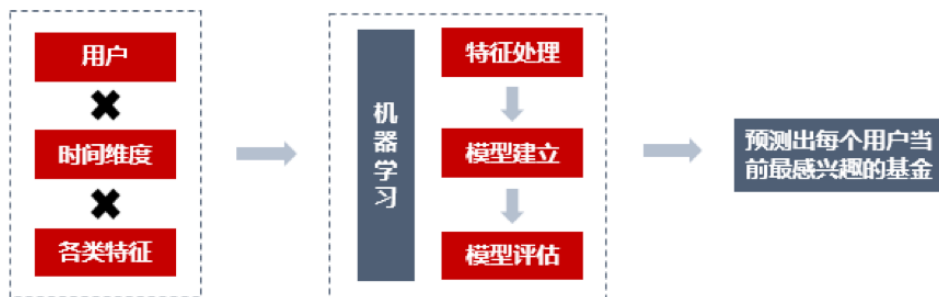
对于数据仓库来说是一个比较复杂的系统，需要很多配套的系统辅助才能做好这样的项目。而其中很多系统在 MaxCompute、DataWorks 中都已经产品化，大大的简化了大数据平台的搭建和运维，提供了一站式的解决方案，而且通过阿里云 MaxCompute、Dataworks 能够实现敏捷开发、快速响应、轻量化运维、低成本的实现大数据平台架构。其中包括最核心的调度系统、权限管理、元数据管控、数据安全保护伞等等一系列功能。而在使用中，数据分析师能够快速上手完成数据的加工和分析。

业务价值案例：

收益王者：

收益王者产品帮助用户追踪头部用户的交易行为，使用用户自身数据来影响用户心智，满足了用户的窥私欲、攀比欲。该产品为用户提供了投顾化的数据服务，为公司提升了用户粘性及交易转化，在 2018 年实现销量数亿元。在开发过程中，MaxCompute 帮助我们快速、精准地处理海量用户交易数据，为该产品数据的准确性、稳定性、及时性提供了有力的保障。

产品 AI 推荐：



我们根据用户自身属性、交易行为、资产属性以及与他类似的用户的产关注
和交易行为，预测每位用户当前最感兴趣的基金。产品 AI 推荐与传统的仅从市场
出发的产品推荐不同，我们从用户的角度，根据用户的行为数据，做出千人千面的
产品推荐，提升了用户体验，并提升了交易转化率。在特征加工、模型开发、预测
结果投入使用过程中，Dataworks 为我们提供了整套技术架构，包括算力强大的
MaxCompute、组件丰富的 PAI 机器学习平台以及 ADS、RDS 等产品，满足了
我们各方面的需求。

小打卡：基于 MaxCompute+PAI 的推荐算法实践

高清 小打卡 算法负责人

简介：小打卡是国内最大的兴趣社群平台，每天能够产生上百万条新的内容。依托于阿里云 MaxCompute，小打卡已经完成了 TB 级数据仓库方案。在此基础之上，结合机器学习 PAI，实现了千人千面的推荐算法。

前言：

小打卡是国内最大的兴趣社群平台，每天能够产生上百万条新的内容。在这样超大的内容生产背景下，平台也面临着千人千面、内容分发上的巨大挑战。依托于阿里云 MaxCompute，小打卡已经完成了 TB 级数据仓库方案。在此基础之上，结合机器学习 PAI，实现了千人千面的推荐算法。本文将从技术选型、推荐架构、开源算法结合三个方面，讲述小打卡在 MaxCompute 上的一些实战经验。

一、为什么选择 MaxCompute

MaxCompute 是阿里云完全自研的一种快速、完全托管的 TB/PB 级的数据仓库解决方案，并且上层提供了 DataWorks 以实现 workflow 可视化开发、调度运维托管的一站式海量数据离线加工分析平台。除此之外，MaxCompute 还与阿里云服务的多个产品集成，比如：

- 数据集成

完成 MaxCompute 与各种数据源的相互同步

- 机器学习 PAI

实现直接基于 MaxCompute 完成数据处理、模型训练、服务部署的一站式机器学习

- QuickBI

对 MaxCompute 表数据进行报表制作，实现数据可视化分析

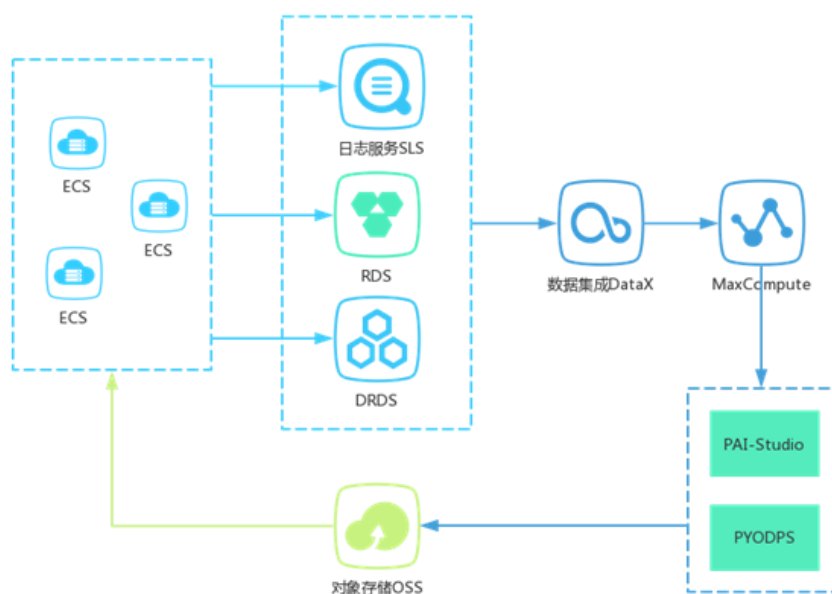
- 表格存储

阿里云自研的分布式 NoSQL 数据存储服务，MaxCompute 离线计算的机器学习特征可以很方便的写入，以供在线模型使用

相反，如果完全基于开源的 Hadoop 框架，从服务部署、可视化开发、代码管理、任务调度、集群运维等多方面，均需要大量的人力来开发与维护。基于 MaxCompute，不论是人力成本，还是计算成本，还是运维成本，都已经降到了最低。

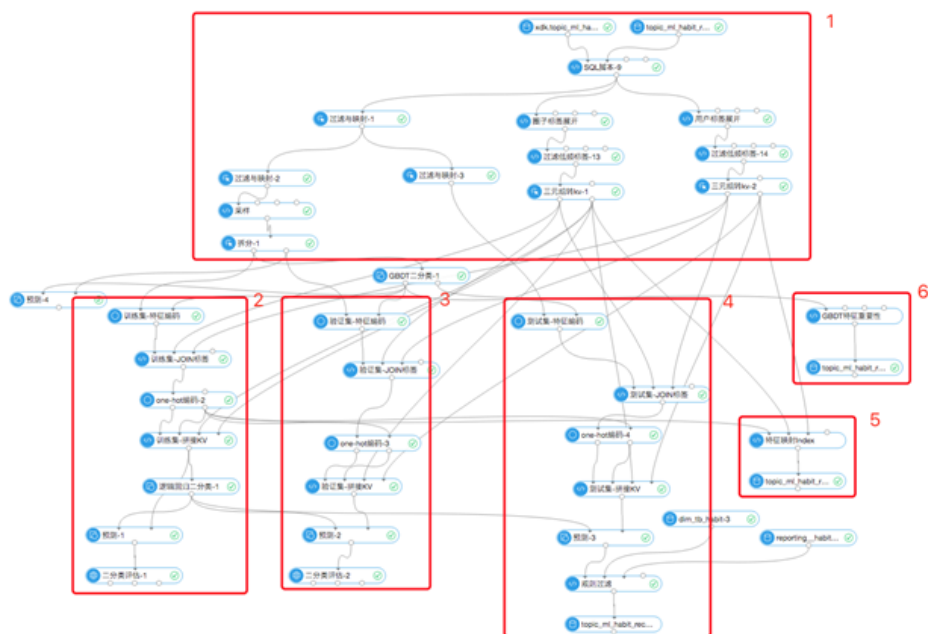
二、小打卡推荐系统架构

小打卡的整个技术架构完全基于阿里云实现。埋点日志、业务数据分别由日志服务、RDS/DRDS 承担收集和存储任务，通过数据集成同步到 MaxCompute，之后便可基于 PAI 实现机器学习任务。其物理结构如下图所示。



机器学习 PAI-Studio 提供了数据预处理、特征工程、机器学习、深度学习、文本分析等丰富的机器学习组件，并且计算结果直接以宽表的形式存储于 MaxCompute，极大的减轻了算法开发的工作量。

我们基于 PAI，实现了 GBDT+LR 算法，任务流如下图所示：



图中采用了丰富的 PAI 机器学习组件，主要包括 6 个部分：1- 特征加工，2- 训练模型，3- 验证模型，4- 测试模型，5- 特征映射关系，6- 特征重要性。开发完成后，可以加载到 DataWorks 中进行调度，运行完成会生成 GBDT 模型文件 (pmml 格式)、LR 模型文件 (pmml 格式)、特征映射表，以便线上使用。由于我们后端主服务均在华北 1，而 PAI 的模型在线部署在华东 2，存在着公网访问问题，因此我们暂时无法使用 PAI 的模型在线部署功能，建议大家将两部分放在同一地域。鉴于此问题，幸好 PAI 的同学提供了将 pmml 模型文件写入 oss 的脚本，相关代码如下：

pmml 模型写入 oss

```
pai --name modeltransfer2oss
```

```
-DmodelName=xlab_m_GBDT_LR_1_1806763_v0  
-DossPath="oss://test.oss-cn-shanghai-internal.aliyuncs.com/model/"  
-Darn="acs:ram::123456789:role/aliyunodpspaidefaultrole"  
-Doverwrite=true  
-Dformat=pmml;
```

因此我们可以将 pmml 文件写入 oss，然后后端服务读取 pmml 模型文件，自行创建模型在线预测。对于解析 pmml 模型文件，虽然有开源项目支持 pmml 模型加载，但是由于 pmml 过于通用，导致性能存在问题，因此我们定制化自解析模型。

三、如何结合开源算法

遗憾的是，PAI 提供的机器学习算法仍然有限，如果想要使用开源项目来实现算法部分怎么办呢？我们对此也做了尝试，我们则结合 MaxCompute+PAI+xLearn 实现了基于 FM 算法的 CTR 预估模型。

由于 xLearn 需要在单独的一台 ECS 上执行，那么问题就来了：

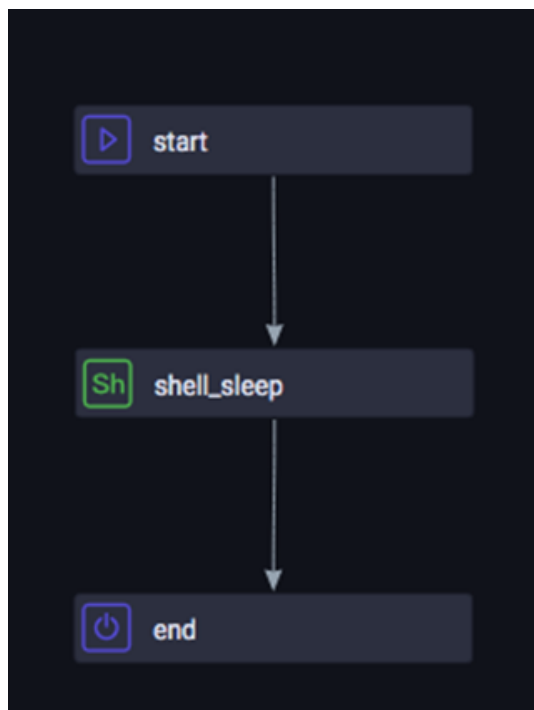
1. 如何从 MaxCompute 拉取数据，又如何上传结果？

MaxCompute 提供了 pyodps，可以很方便的使用 python 读写 MaxCompute 的离线表。因此，我们在 python 循环检测 PAI 任务的特征工程结果表是否生成完成。生成完成，则启动下载数据和算法训练任务。算法执行结束后，则将模型文件通过 pyodps 写入 MaxCompute，当然也可以使用 tunnel 工具来完成。

2. 如何周期性调度？DataWorks 上的任务与 ECS 上的任务，如何形成依赖关系？

对于拉取，我们通过循环实现了 python 与 MaxCompute 表的依赖关系，那么算法训练完成了，DataWorks 的下游怎么知道呢？幸好，DataWorks 提供了 do-

while 控制流组件，我们可以在 while 条件中检测模型表相应分区的数据是否存在了，在 do 组件中，则采用 shell 执行 sleep。跳出 while 后，则下游也开始正常执行了。之后便可以将模型和映射文件同步写入到 oss，以供后端使用了。其中 do-while 结构如下：



四、总结

除了排序算法之外，我们还使用 PAI 和 SQL，分别实现了 item_cf 和 user_cf 的召回算法，整个推荐系统完全打通。基于 MaxCompute 利用 DataWorks 调度系统，我们实现了推荐算法模型的每日自动更新。相比于模型不更新，我们对比了 60 天前的模型，新的模型效果提升 10% 左右。

期待阿里云提供越来越丰富的功能，例如基于 MaxCompute 的 Spark 机器学习。

小打卡：快速部署大数据计算平台实践

李跃凯 小打卡 数据负责人

简介：小打卡目前已为 3000 万用户提供体验服务 3.4 亿人次，内容消费 7.4 亿人次。在庞大的用户量背后，MaxCompute 产品的搭建显得尤为重要。

一、业务背景

小打卡致力于帮助用户成为更好的自己。目前，已为 3000 万用户提供体验服务 3.4 亿人次，内容消费 7.4 亿人次。

在小打卡上线初期，业务分析所需的数据主要是通过查询 mysql 库表。

现在，小打卡的主要业务分析需求，包括业务报表，用户行为分析，A/B/n 实验评估，个性化推荐，数据服务等全部是借助于阿里云的大数据平台来满足。

选择阿里云大数据产品的原因：

1. 成本低

享用阿里云超大规模的云计算资源，按照实际需要采购存储和计算资源。

企业无需组建专门的大数据平台部署和运维团队，在业务发展初期，极大的降低了拥有大数据平台的各项成本。

2. 效率高

企业通过阿里云官网了解并采购所需的大数据产品，快速搭建适合业务的平台架构。

阿里云大数据提供开发生产环境隔离的集成开发环境，以及完善的调度 / 监控 / 数据管理等工具能力，提高数据仓库的开发效率。

企业可以快速构建大数据平台的功能模块，快速相应业务需求。

3. 性能按需采购

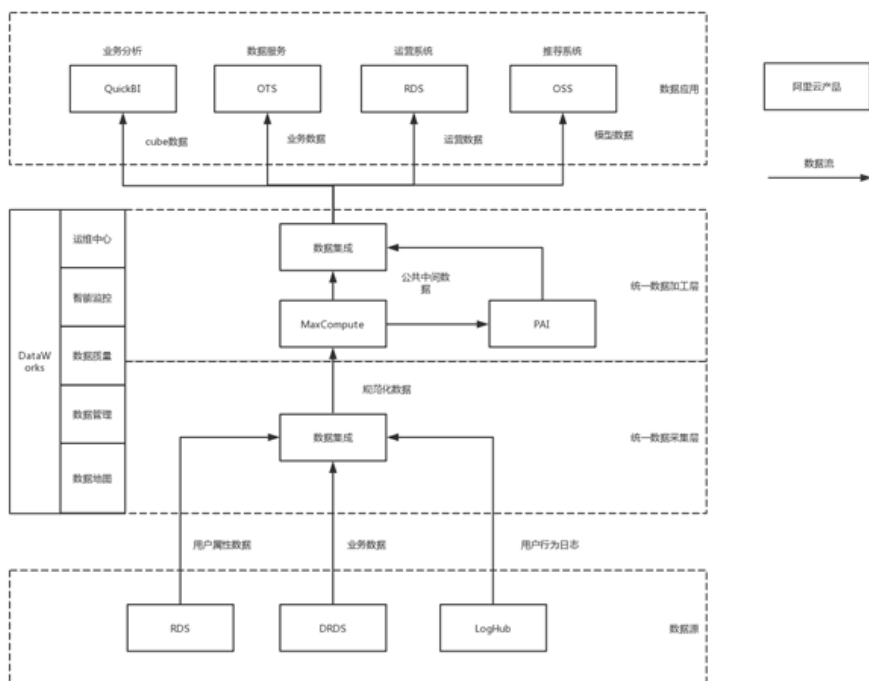
阿里云大数据的 I/O 及计算能力弹性伸缩，可以支持 TB/PB/EB 级数据规模，千万级别复杂任务调度和万兆的网络同步速率。

4. 安全

阿里云大数据提供云上数据的安全，以及企业租户之间的安全隔离，大数据项目不同角色的权限管理和各种数据资源的权限管理。

二、大数据实践

小打卡基于阿里云大数据产品实现的离线数仓架构：



面向小打卡的各项业务场景，所需的基本产品搭配：

1. 用户行为分析：DataWorks + 数据集成 + MaxCompute + Quick BI
2. 数据化运营：DataWorks + 数据集成 + MaxCompute + RDS
3. 线上数据服务：DataWorks + 数据集成 + MaxCompute + OTS/API 网关
4. 推荐系统：DataWorks + 数据集成 + MaxCompute + PAI + OSS

实践 case：

1. 业务分析需求：监控每小时访问小打卡小程序的新增用户数及活跃用户数
2. 阿里云大数据产品搭配：DataWorks + 数据集成 + MaxCompute + Quick BI

DataWorks：DataWorks（数据工场，原大数据开发套件）是阿里云数加重要的 PaaS 平台产品，提供数据集成、数据开发、数据管理、数据治理、数据分享等全方位的产品服务，一站式开发管理的界面，帮助企业专注于数据价值的挖掘和探索。

MaxCompute：大数据计算服务（MaxCompute，原名 ODPS）是一种快速、完全托管的 EB 级数据仓库解决方案。

数据集成：数据集成是阿里集团对外提供的稳定高效、弹性伸缩的数据同步平台。致力于提供复杂网络环境下、丰富的异构数据源之间数据高速稳定的数据移动及同步能力。

Quick BI：Quick BI 是阿里云旗下产品，是一个基于云计算致力于大数据高效分析与展现的轻量级自助 BI 工具服务平台。

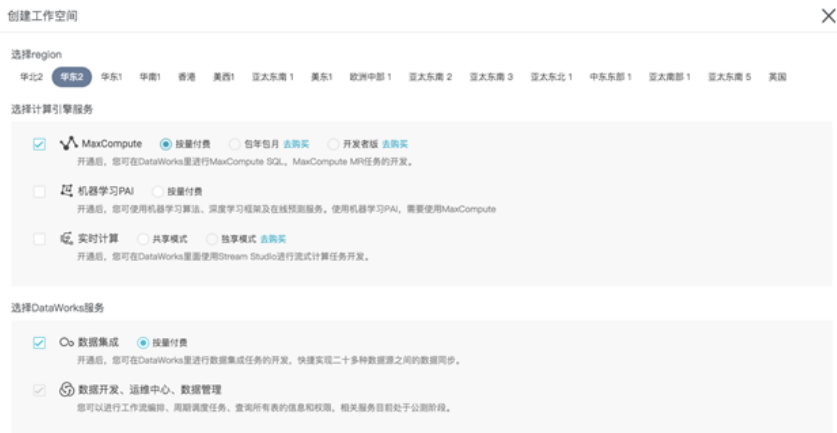
3. 用 2-3 个工作日，快速搭建大数据平台：

步骤一：注册阿里云账号，开通 DataWorks 及 Quick BI

步骤二：快速建设数仓

1. 在 DataWork 控制台创建工作空间

小打卡将原始数据的采集层和中间公共数据的加工层分别部署在独立的项目中。

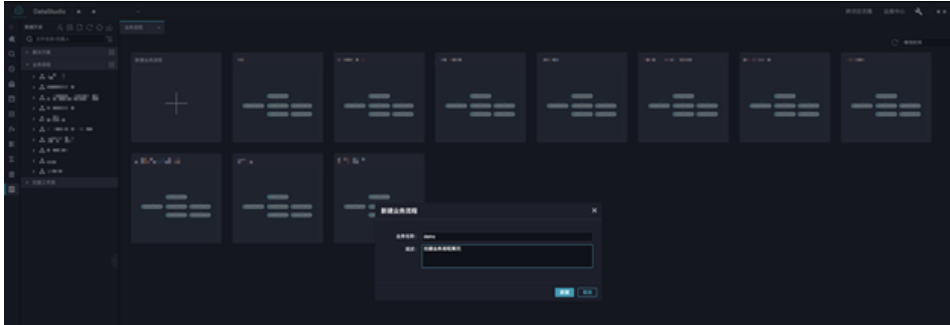


2. DataWorks 平台中集成了数据集成和 MaxCompute 等大数据产品，可以一站式的开发。



在 DataStudio(大数据集成开发环境工具) 中建设数据仓库，创建业务流程，物理模型，数据集成任务及 ETL 任务。

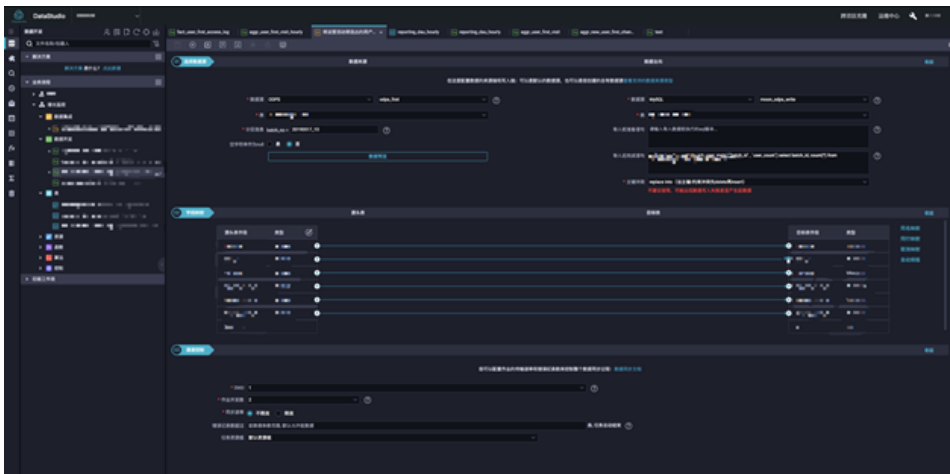
业务流程帮助企业总结业务的一般流程，来有效组织相互依赖的数据流，数据集成任务，ETL 任务，数据表和 UDF 等其他资源。



数据集成可以帮助企业从异构数据源采集数据并沉淀到数据仓库

阿里云的数据集成提供丰富的数据源支持：

- 文本存储 (FTP/SFTP/OSS/ 多媒体文件等)
- 数据库 (RDS/DRDS/MySQL/PostgreSQL 等)
- NoSQL (Memcache/Redis/MongoDB/HBase 等)
- 大数据 (MaxCompute/AnalyticDB/HDFS 等)
- MPP 数据库 (HybridDB for MySQL 等)





3. 将数据接入 Quick BI，交付业务部门使用。

目前由于小打卡分析业务的团队规模较小，仅有 10-20 人的规模，所以当前的架构是直接读取 MaxCompute 中的数据。这样做的好处是省钱，非常省钱，部署也是非常的快速。

但是缺点则是查询速度较慢，只能维持在秒级，且报表的查询并发度有瓶颈。后续随着分析团队规模的增加，会适时的优化架构，引入分析型数据库产品 ADB 来提供毫秒级的速度和高并发的查询性能。



基于 MaxCompute 搭建社交好友推荐系统

翟永东 驻云科技 资深架构师

摘要：本次由阿里云驻云科技资深架构师翟永东带来了“基于 MaxCompute 搭建社交好友推荐系统”为主题的分享，主要对大数据在好友推荐系统中的应用、好友推荐系统的分析模型、好友推荐系统在阿里云上的实现方式和 MaxCompute 技术进行了精彩的介绍。

大数据在好友推荐系统中的应用

给大家分享一下基于 MaxCompute 搭建社交好友推荐系统，使用 MaxCompute 阿里的大数据计算的方法可以做哪些事情，如果说是以社交好友的推荐，来给大家去演示一下。好友推荐系统它的一个场景介绍，现在大家都在讲大数据，如果想去使用这些数据，我们认为它需要具备三个要素，第一个要素是海量的数据，数据量越多越好，只有数据量达到了足够大，我们才能够成为一个数据里面潜在去挖掘出来。第二个是处理数据的能力，有了这样很高的快速处理数据的能力，可以让我们更快的去把数据里面的信息挖掘出来。第三个是商业变现的一个场景，我们采集大数据的时候，并不是数据越多越好，一定要有一个具体的场景。以推荐系统为例来看一下大数据的一个应用。

大数据在好友推荐系统中的应用

支付宝



Linkin



云栖社 为了无法计算的价值 | 阿里云

左边是支付宝，在支付宝一打开的时候，下面会有一栏推荐可能是你的好友，一般的话下面的那些人都是你认识的，可能还没加他们为好友。右侧是 Linkin，它是一个求职社交网站，Linkin 也会给你这样的一个推荐，会告诉你哪一些用户是你潜在的好友，而且 Linkin 会告诉你这个好友跟你是一度的关系的还是两度的关系或者是三度的关系。潜在关联性高的，会在前面直接显示出来，潜在关联性没有那么高的也会在后面显示出来，这两个都是典型的一个好友推荐。

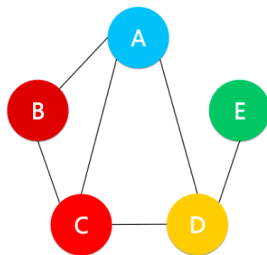
进行好友推荐的时候，怎么给用户进行推荐，首先这两个人是非好友的关系，接着我们去看一下他们俩潜在共同好友的处理，通过这种方式去给用户推送，比方说潜在好友数量多，我就认为这两个人是好友关系，就是通过这种方式来实现的。

好友数据的描述

假定好友关系是双向的：

原始数据每条Record包含两个字段，user和friends，use唯一表示一个用户，friends是该用户的好友示例数据如下：

user	friends
A	E C D
E	A C
C	A B D
D	A C E
E	D



云栖社 为了无法计算的价值 | 阿里云

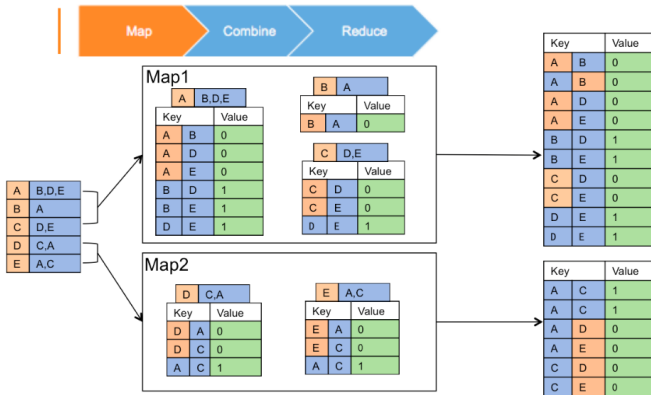
上图的右侧是人与人之间的一个社交关系的服务，比如说 A 跟 B 是一个好友，我们可以通过这五个方式画出来，让机器去分析这些数据，需要把右边这种社交的关系，转换成机器可以识别的数据，转换成左侧这样的二维表的数据，比如说 A 跟 B、C、D 他们之间是好友，我们左侧是 A 跟 B、C、D 是好友关系，剩下这些也是类似的，这样就可以把这个表传到机器里面进行分析，比方说通过分析之后，发现 A 跟 E 有一个共同好友，B 跟 D 有两个共同好友，然后 C 跟 E 有一个共同好友。这个时候就可以推荐 B 跟 D 他两个是一个潜在的好友，而排在前面，A 跟 E 或者 C 跟 E 排在概率往下，稍微低一些，潜在好友多的排在前面，潜在好友少的排在后面，通过这种方式来进行排列，这个是我们期望的结果。

好友推荐系统的分析模型

我们怎么来去计算呢？我们一般使用方式是什么呢？使用的是 MapReduce 这样一个计算模型，MapReduce 是一种编程模型，用于大规模数据集的并行运算，它由三部分组成分别是 Map、Combine、Reduce。

以好友推荐这样的一个场景为例。

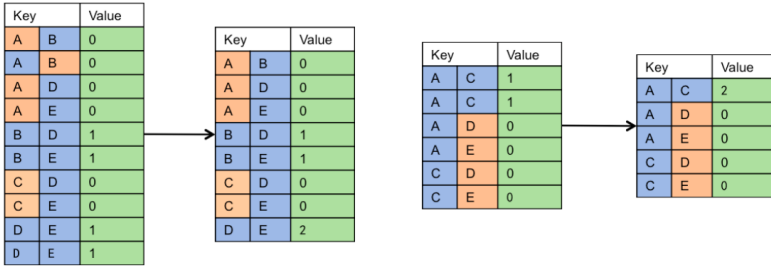
好友推荐系统的Map阶段



云栖社 为了无法计算的价值 | 阿里云

首先输入左侧机器可以识别的数据，输入之后，在 Map 端先把数据做一个拆分，拆分成两份不同的数据，在拆分的同时把它转换成 key、value 的类型，比方说 A、B、D、E 这几行数据转换成什么呢？A 跟 B，然后 value 是零，零代表他们两个已经是好友。如果两个不是好友的话，自定义这一行数据，B 跟 D 不是好友，就把他的值视为 1。下面的 B、E，还有 D 跟 E 也是 1。把原来一行数据转换成 Key、Value 这个形式的数据，类似于右边这样的数据，上面是 key、value 的一个类型，下面也是类似的。这个是在 Map 做的事情，把这个数据通过两个 key、value 进行一个拆分，转化成 key、value 这样的类型。

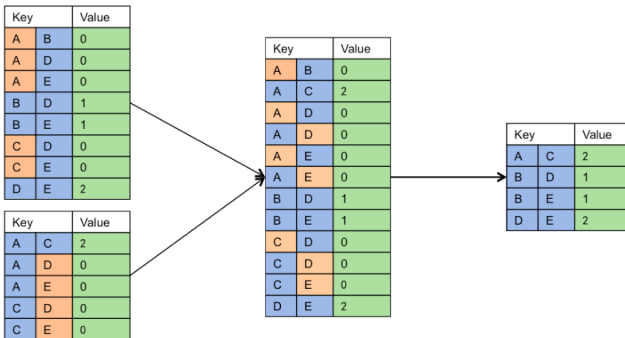
好友推荐系统的Combine阶段



云栖社区 yunqishe.com 为了无法计算的价值 | 阿里云

Combine 是对数据先做一个本地的汇总，先看到有一些数据是重复的，比如说 A 跟 B 是零，A 跟 B 是零，出现了两次，这个时候就存一个就可以。其他类似的，这样我把这些数据在本地做完汇总，类似于这张表，这两个数据。

好友推荐系统的Reduce阶段



云栖社区 yunqishe.com 为了无法计算的价值 | 阿里云

接着是第三步是 Reduce 阶段，Reduce 是对这些数据进行一个汇总，把两边

数据汇总到一起，然后对每一个 Key 值对应唯一的一个 value 值做一个汇总，这个就是它最终计算的一个结果。如果两个用户已经是好友了，Value 值是零的话，不需要再给他推荐。所以说 A、B 如果是零的话就别掉，只需要知道它的 value 值是大于零的，有潜在好友，同时这两个人目前还是非好友的关系，这个就达到了想要的效果。

好友推荐系统在阿里云上的实现方式



好友推荐阿里云实现整个的架构是怎么样的呢？比方现在有一个社交软件是一个业务系统，前端使用阿里云的云服务器 ECS 去部署整个的社交的软件的应用，入库的一些数据存到阿里的 RDS，这个就是当前的一个社交应用系统。业务系统里面产生了一个数据，怎么来对数据进行分析，首先需要在数据库里边把这个数据提取出来，提取到阿里云的大计算服务 MaxCompute 里面，很类似于我们传统做数仓的时候 ETL 的一个过程，会利用阿里云的大数据开发平台对数据进行分析 and 处理。

使用它可以快速便捷的去开发我们数据植入或者数据这样的一个流程，这个就是会使用大数据开发平台和大数据制造，结果是一个数据分析结果，还需要前端的应用

数据对分析出来的结果展示出来。

MaxCompute 的技术特点

对于 MaxCompute 的一些技术特点主要有一下几点：

- (1) 分布式：分布式集群、跨集群技术、可灵活扩展。
- (2) 安全性：从安全性来讲具有自动存储纠错、沙箱机制、多分备份。
- (3) 易用：具有标准 API、全面支持 SQL、上传下载工具。
- (4) 权限控制：多租户管理、用户权限策略、数据访问策略。

MaxCompute 的使用场景

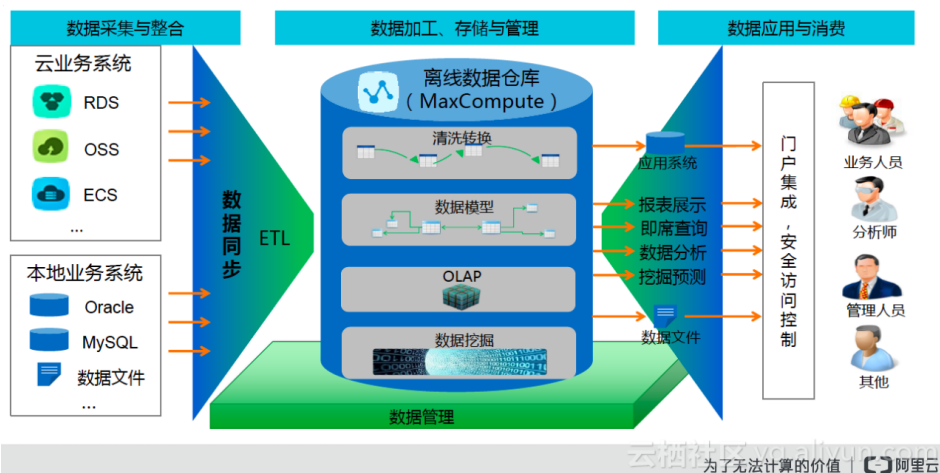
对于 MaxCompute 的使用的场景，可以使用 MaxCompute 搭建自己的一个数据仓库，同时，MaxCompute 还可以提供一种分布式的应用系统，比方说可以通过图计算，或者通过有效的宽幅的方式，可以搭建一个工作流；比方说数据分析并不是说只分析一天就不分析了，其实是周期性的。如果数据每天要分析一次，可以在 MaxCompute 里面生成那样的任务工作流，设置一个周期性的调度，每天要让它调度一次，MaxCompute 可以按照设计好的工作流，调动周期，然后去运行；MaxCompute 在机器学习里面也是有用的，因为机器学习会用到 MaxCompute 分析出来的数据，其他相类似的服务对数据进行分析处理，分析出来的结果数据放到机器学习平台里面，让机器通过一些算法一些模型，去学习这里边的数据，生成一个希望达到的一个模型。

大数据开发套件 DataIDE

另外一个除了 MaxCompute 之外还有一个会用到一个大数据开发操作 DataIDE，大数据开发套件 DataIDE（现名：数据工场 DataWorks）提供一个高效、安全的离线数据开发环境。为什么介绍它呢？是因为 DataIDE 只是对数据任务工作流的一个开发，其实底层的数据处理，数据分析，都是在 MaxCompute 上完成，可

以简单理解为 DataIDE 就是一个图象化的数据开发的服务，它是为了帮助我们更好去使用 MaxCompute。也可以看到，这我们可以在 DataIDE 进行一个开发，不需要直接在 MaxCompute 里面进行开发了，在 MaxCompute 开发的一个效果，跟在 DataIDE 里面开发的效果对比。

DataIDE的应用场景



MaxCompute 的应用开发流程

MaxCompute 的应用开发流程一共需要六步分别是：

- (1) 安装配置环境
- (2) 开发 MR 程序
- (3) 本地模式测试脚本
- (4) 导出 jar 包
- (5) 上传到 MaxCompute 项目空间
- (6) 在 MaxCompute 中使用 MR

下面我们以一个好友推荐的事例来详细讲解一下这个过程。首先需要去安装

MaxCompute 客户端，使用它的好处是可以在本地通过命令的方式去远程使用阿里云的 MaxCompute，在本地只需要配置 MaxCompute 信息就可以。另外还需要去配置自己的一个开发环境，因为现在阿里云的 MaxCompute 主要是两种语言，一种是 Java 一种是 Eclipse。然后新建项目，在开发新建项目的时候，大家可以看到这个红包，这个红包就是需要配置本地的客户端的信息。在进入到写代码的过程。

接下来就是简单的测试，开发之后要测试，这个代码是不是按照设想的方式去工作的。接着这边输入的是一个测试数据，这个输出的数据类别，就是输出的这样的一个表格，表格有三列，第一类是用户 A，第二类是用户 B，第三类是两个潜在的共同好友的数量，只需要关注这三个数据就可以，然后就可以测试。接着第三个本地运行的数据的代码，运行的结果就是通过本地的开发测试，在本地测试的时候这边有一个数据，你第一步需要选择是使用哪一个的一个项目处理。第二个要选择输入表和输出表，要告诉他输出表是哪个，输出表的目的是什么，告诉这个程序，你输出的结果保存在表里面，配置好点击运行这个结果就出来了。

本地开发测试成功之后，接着要把它打成一个 Jar 包，然后上传到阿里云上，就是上传到 MaxCompute 的集群里边。第二个打完 Jar 包以后添加资源，下面就把刚刚输出的 Jar 包，通过资源的管理，把刚刚输入的 Jar 包上传上来。本地开发测试好的一个 MR 的 Jar 包已经上传到 MaxCompute 集群里边。

上传好了之后就可以使用它，去新建一个任务，然后这个任务去起个名字，这个任务跟哪一个 Jar 包相关联，接着是 OPENBMR，我们选的是 MR 的程序，所以里面选的是 OPENMR 模块，生成这样的一个任务，进入到编辑页面，在编辑页面里面首先告诉它，这个 OPENMR 这样的一个任务，使用的是上传的好友推荐的一个 Jar 包，最下面告诉它 Jar 包里面的程序的逻辑是什么，在这个里面制定好之后点击运行结果就会出来。这个就是我们在本地开发测试，把资源上传到 MaxCompute 的集群里面，接着在集群里面去使用我在本地开发好的 Jar 包，这个就是整个的一个开发和部署的一个流程。

发光的二次元——克拉克拉上云实践

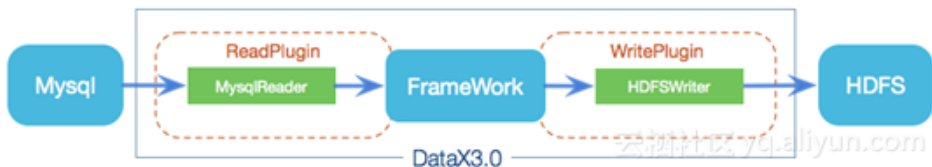
田亮 深圳市有信网络 大数据团队负责人

克拉克拉 (KilaKila) 是国内专注二次元、主打年轻用户的娱乐互动内容社区软件。KilaKila 推出互动语音直播、短视频配音、对话小说等功能，满足当下年轻用户个性化、碎片化的文娱需求。随着业务规模增长，海量数据存储与计算的瓶颈也日益突出，由于单台服务器的处理能力有限，海量数据的分析需要分布式计算模型。分布式的计算模型对数据分析人员要求较高且不易维护：数据分析人员不仅需要了解业务需求，同时还需要熟悉底层分布式计算模型。MaxCompute 提供完善的数据导入方案以及多种经典的分布式计算模型，可快速解决克拉克拉所面临的海量数据的计算问题，有效降低企业成本并保障数据安全。对于使用方的我们不必关心分布式计算和维护细节便可轻松完成大数据分析，最终我们采用阿里云 MaxCompute 方案进行数据上云。

一、数据上云

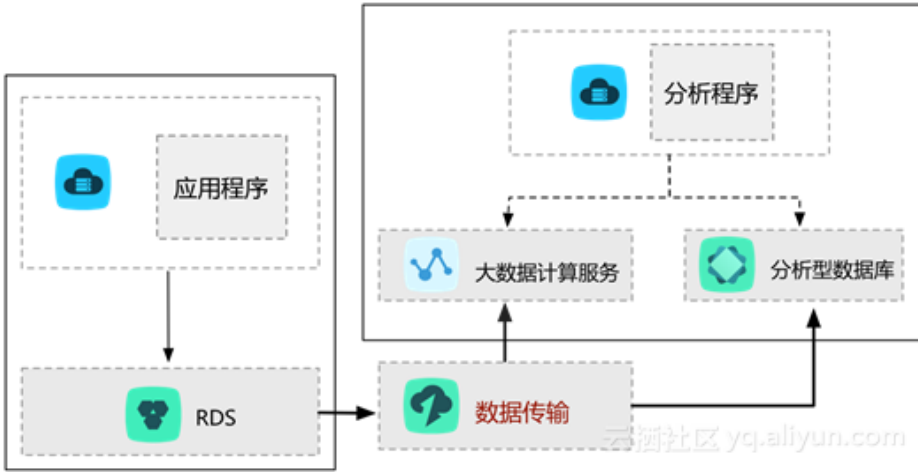
克拉克拉 (KilaKila) 数据源主要分为两部分：1. 结构化业务数据表 (MySQL)；
2. 非结构化用户行为日志

- 结构化业务库上云主要使用阿里云 datax、DTS 两种工具。



图一 datax 数据同步

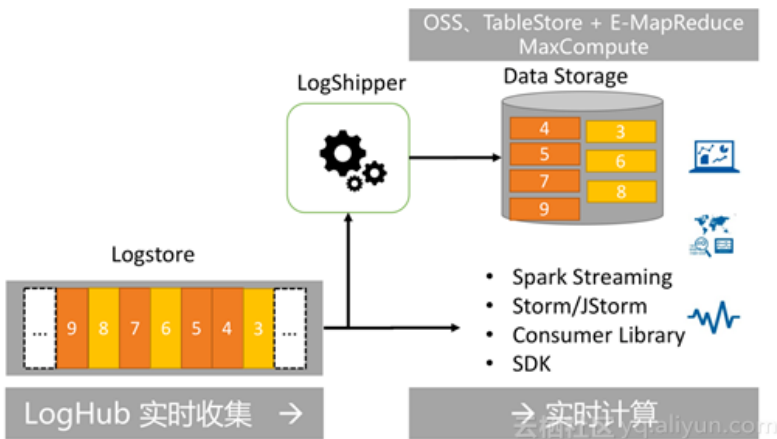
datax 原理是将 MySQL 数据完全透明化同步至 MaxCompute 数据表中，是表对表的映射。



图二 DTS 数据同步

DTS 主要以 MySQL binlog 方式来进行数据同步，该同步的优势在于不会影响线上生产环境 MySQL 库 I/O 压力，与此同时对于删除操作也会进行实时同步。

- 用户行为日志上云，该日志主要使用阿里云日志服务 (Log Service) 组件。该组件除了满足将海量日志存储在 MaxCompute 数据仓库中外，也解决了实时业务计算需求，如 storm 集群。



图三 Log Service workflow

Project名称	注释	地域	创建时间	操作
kila-transcode	视频转码服务	华北2 (北京)	2018-11-13 19:36:02	修改 全球加速 删除
kilakila-bigdata	视频转码服务	华北2 (北京)	2018-08-07 14:49:31	修改 全球加速 删除
hd-live	直播服务	华北2 (北京)	2018-06-19 17:44:51	修改 全球加速 删除
uxin-klive-jp	直播服务	华北2 (北京)	2018-03-14 17:01:14	修改 全球加速 删除
uxin-wk	直播服务	华北2 (北京)	2017-09-20 15:40:00	修改 全球加速 删除
wk-pro-test	直播服务	华北2 (北京)	2017-09-14 14:45:56	修改 全球加速 删除
hongdou-logcenter	直播服务	华北2 (北京)	2017-05-25 16:48:36	修改 全球加速 删除
uxin-app-hangzhou	直播服务	华东1 (杭州)	2016-10-09 15:15:19	修改 全球加速 删除
uxin-app-shenzhen	直播服务	华南1 (深圳)	2016-10-09 15:14:55	修改 全球加速 删除
uxin-live	直播服务	华北2 (北京)	2016-09-21 17:14:50	修改 全球加速 删除
uxin-app	直播服务	华北2 (北京)	2016-09-21 17:14:34	修改 全球加速 删除
uxin-sdk	直播服务	华北2 (北京)	2016-09-21 17:14:15	修改 全球加速 删除

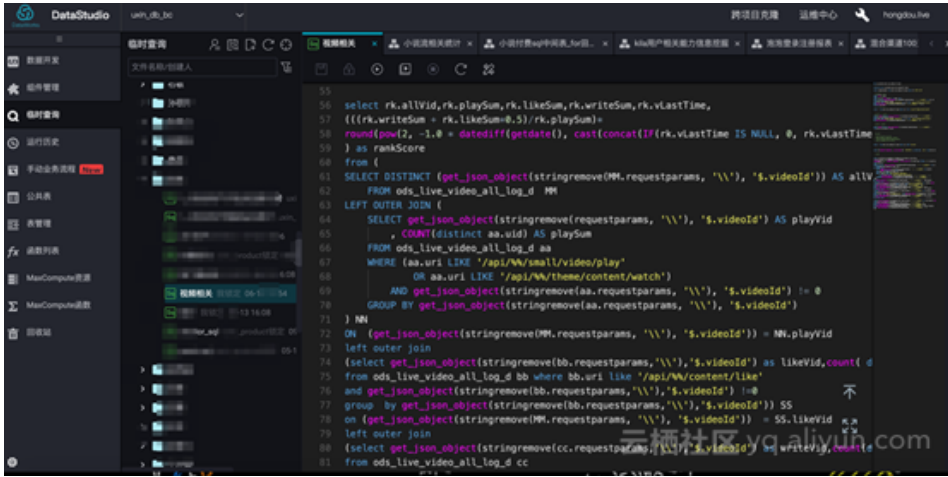
图四 日志服务各 Topic 日志源

The screenshot shows the Log Service console for the topic 'uxzb_api'. It displays a list of log entries with the following columns: ID, Time, and Source. The log content is visible in the main area, showing structured data with fields like 'source', 'hostname', and 'path'. The interface also includes navigation and search options at the top and bottom.

图五 克拉克拉用户行为日志片段

二、数据计算

为满足各类批处理计算业务需求，基于 MaxCompute 建立了小时维度、天维度、月维度等不同维度的工作流以及业务流程。在数据开发阶段我们使用 MaxCompute 的 Web datawork 进行任务开发、任务运维、任务监控等。



图六 datawork SQL 开发

在 datawork 中可以创建 SQL 脚本，通过编写 SQL 语言满足我们的临时查询需求。



图六 datawork workflow

在 datawork 中可以根据业务场景配置 workflow，解决生产环节中所面临的定时计算任务等诉求。



图七 datawork 运维中心

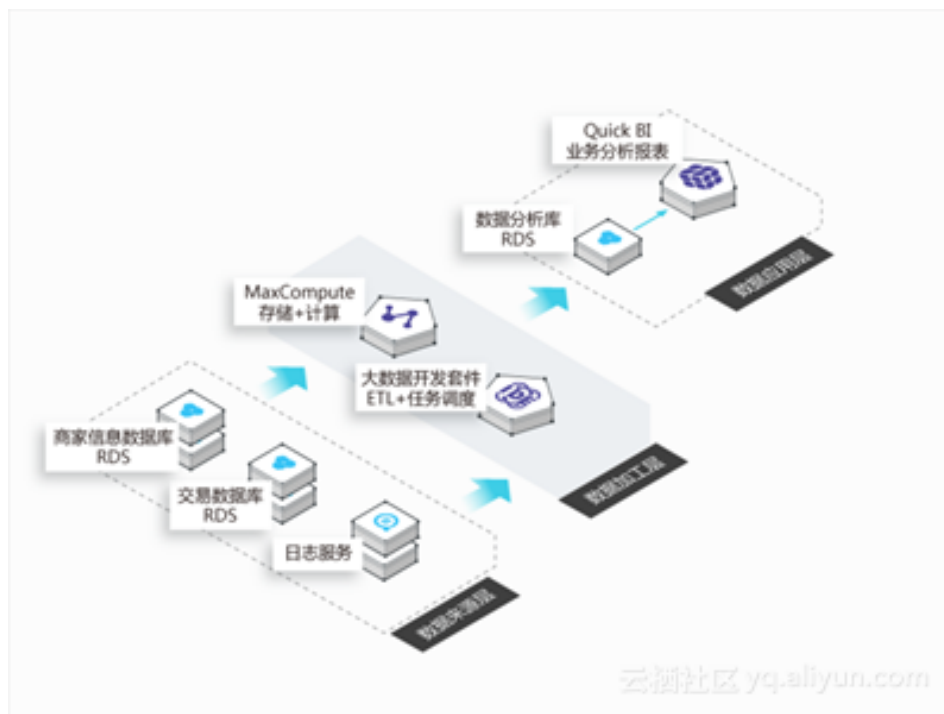
运维中心可实时监控任务的状态，对应不同的任务状态进行不同情况的报警；

综上所述，得益于 datawork 强大的数据开发多样性和丰富的内置函数，我们可以针对不同业务场景进行快速的项目开发和上线。

三、克拉克拉业务应用场景

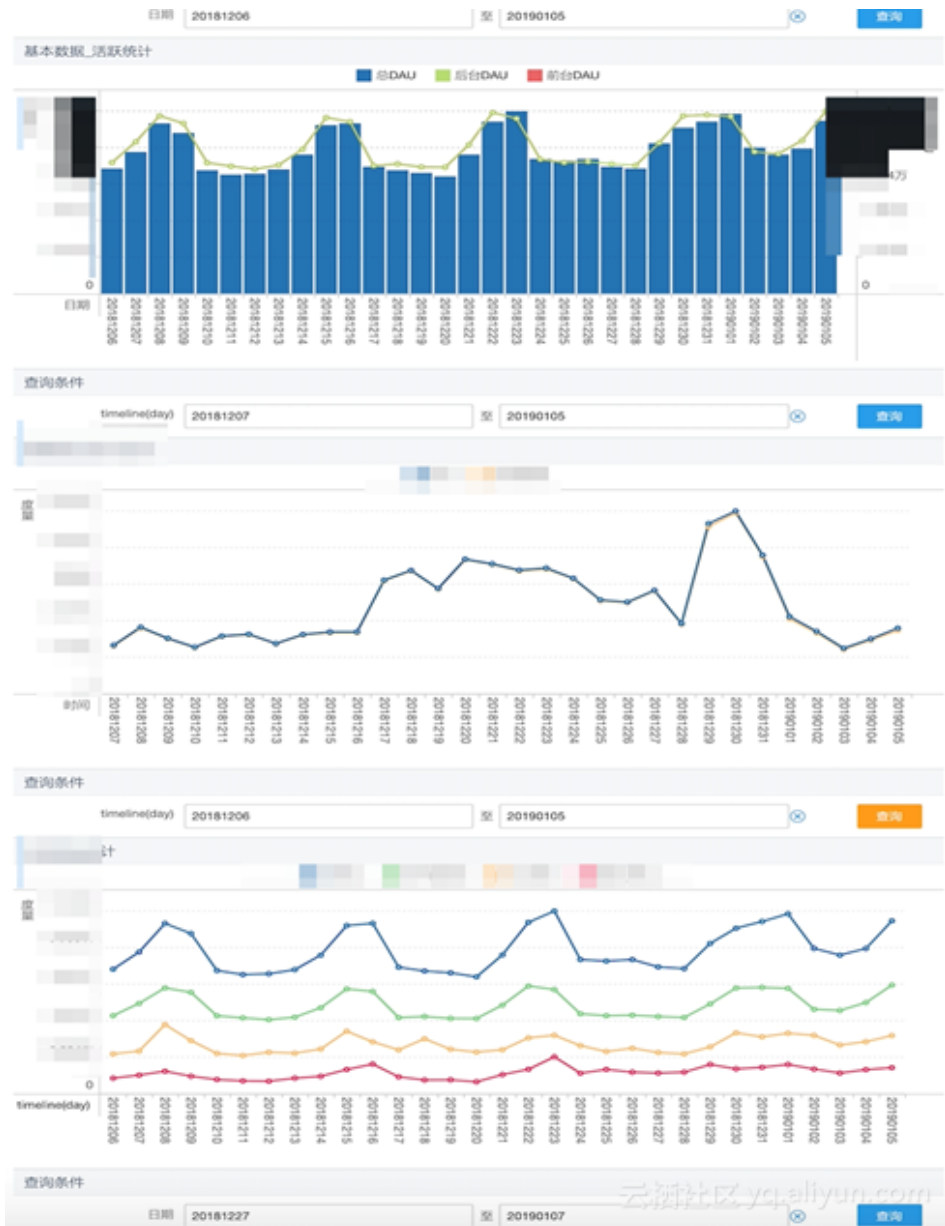
克拉克拉 (KilaKila) 在实际生产环境中，主要有以下四个数据业务应用场景。如下：

1. 报表系统



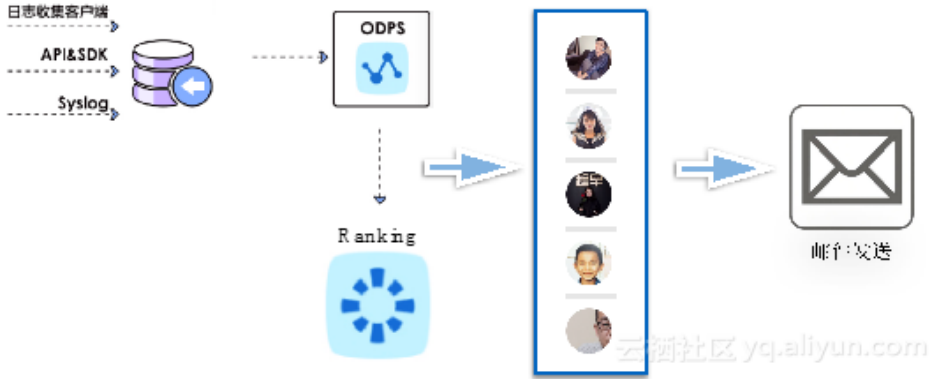
图八 克拉克拉 BI 报表系统架构

基于 MaxCompute 与 quickBI 组合快速构建了业务报表系统。该系统通过 datawork 部署系列报表计算任务，最终的计算结果可存储 MaxCompute 表中以及 MySQL 表中。quickBI 通过数据源关联即可实现前端趋势图灵活展示。



图八 克拉克拉 BI 报表系统

2. 克拉克拉 (KilaKila) 榜单业务

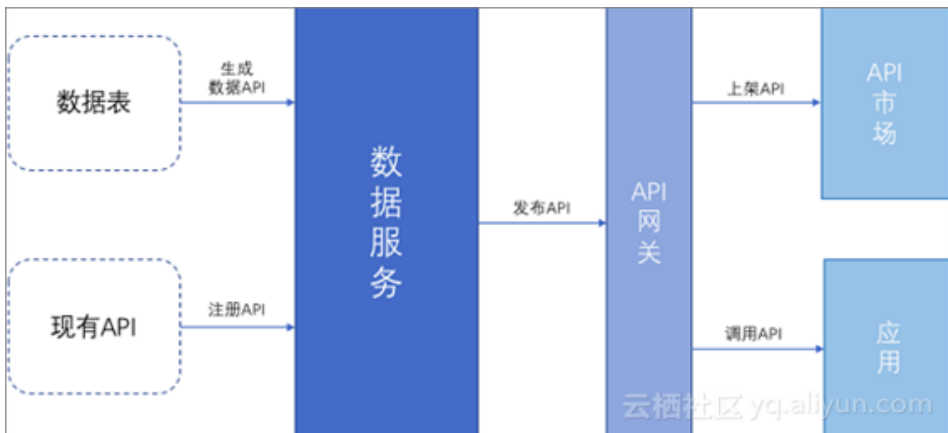


图九 克拉克拉榜单业务计算场景

通过 MaxCompute 满足了克拉克拉主播排行榜、热门榜单等排序业务。

3. 对外开放接口平台 (Restful API)

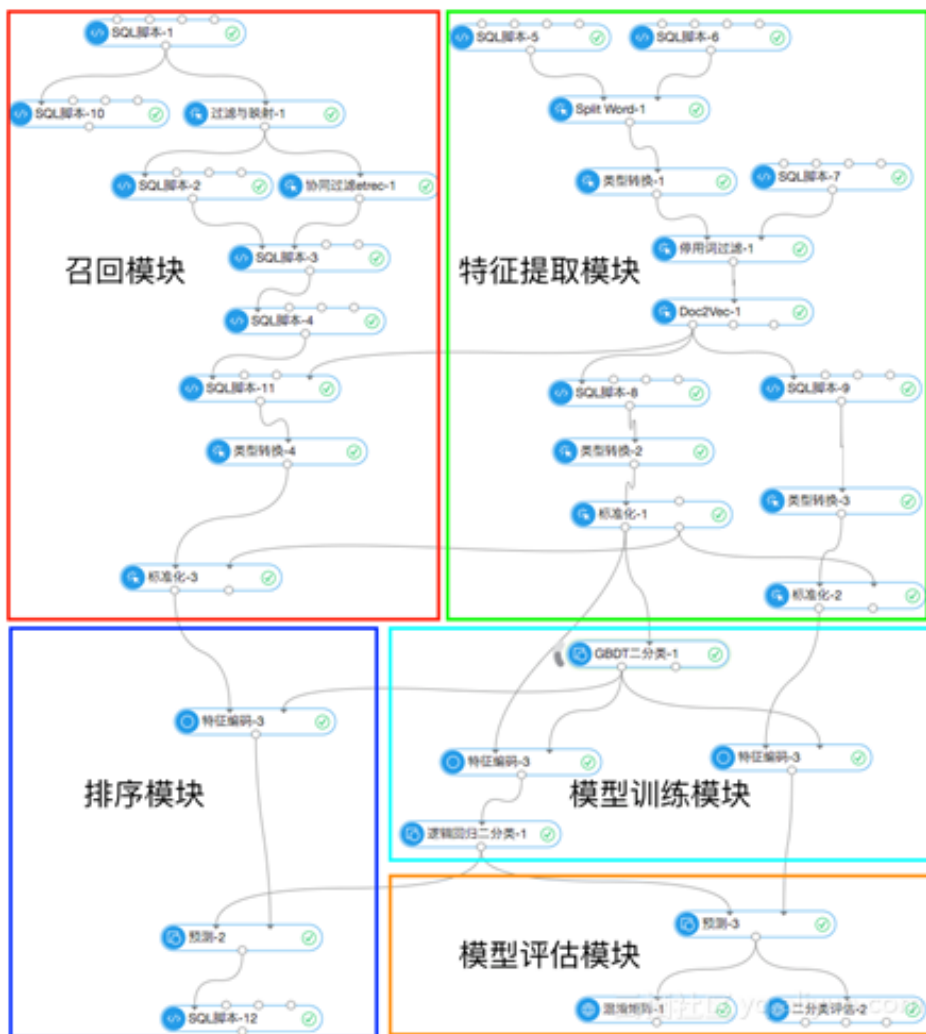
为了满足第三方通过 API 接口获取克拉克拉数据仓库中的用户标签数据，基于 MaxCompute Lightning 架构搭建开放接口服务，通过该服务的网关鉴权机制保证了授权访问和防 DDOS 攻击。



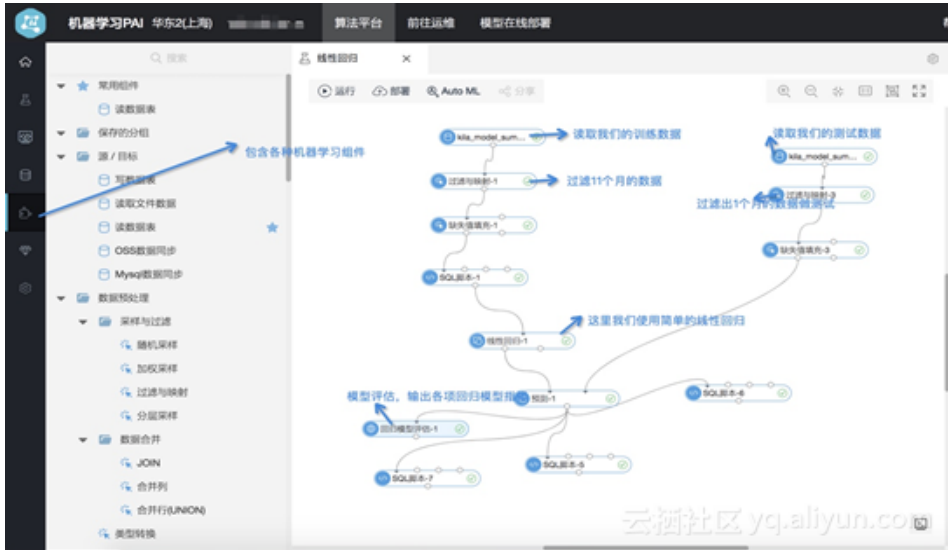
图九 克拉克拉开放接口结构

4. 算法业务

阿里云机器学习平台是构建在阿里云 MaxCompute (原 ODPS) 计算平台之上, 集数据处理、建模、离线预测、在线预测为一体的机器学习平台。克拉克拉 (KilaKila) 预测算法业务、推荐业务等相关项目均建立在机器学习平台基础上, 项目上线快且效果符合预期。



图九 克拉克拉短视频推荐业务



图十 克拉克拉渠道 ROI 预测业务

通过阿里云所提供大数据解决方案，除了解决了克拉克拉所遇到的大数据挑战，也极大的缩短了克拉克拉 (Kilakila) 诸多数据项目从想法到落地的研发周期。

更多内容：

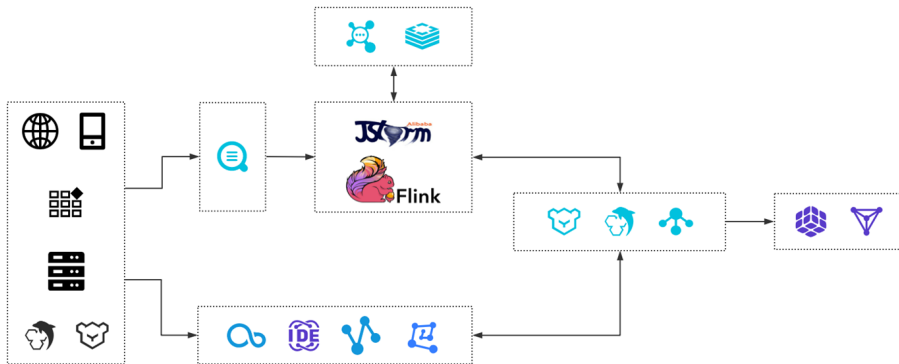
1. [《克拉克拉：基于阿里云 PAI 实现渠道 ROI 投放预测系统》](#)
2. [《克拉克拉 \(KilaKila\)：借力阿里云快速构建短视频推荐系统》](#)
3. [《克拉克拉 \(KilaKila\)：大规模实时计算平台架构实战》](#)

万师傅大数据体系实践案例

王昕岩 万师傅 大数据开发部经理

一、整体架构

每当我在思考技术选型方案的时候，翻翻阿里云的官网，总能找到我想要的东西。于是，我们的大数据体系就变成了这样，如图：



二、离线

2.1 选型原则

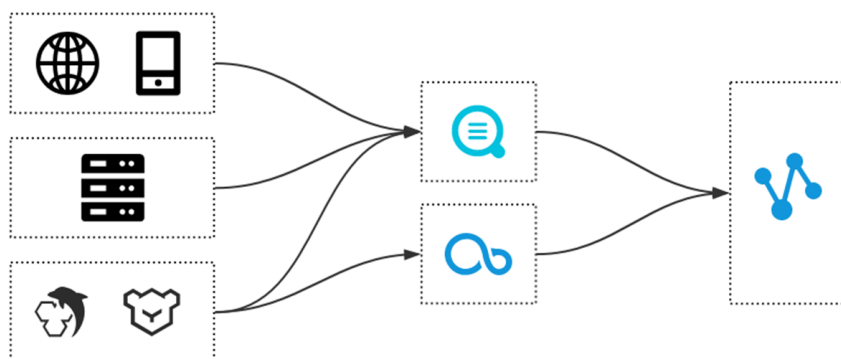
团队成员，大都是 Hive 方向或是算法方向出身。为追求上手简单、专注数据的分析和挖掘、减少不必要的学习成本和费用成本，使用了阿里云 MaxCompute。

2.2 数据采集

数据源共包含三类：

- (1) 关系型数据库中的数据；
- (2) 服务器上的日志文件；
- (3) 前端埋点日志；

采集方式如图：



关系型数据库中的数据，使用 dataworks 中的“数据集成”功能，定时离线同步到 MaxCompute 中；

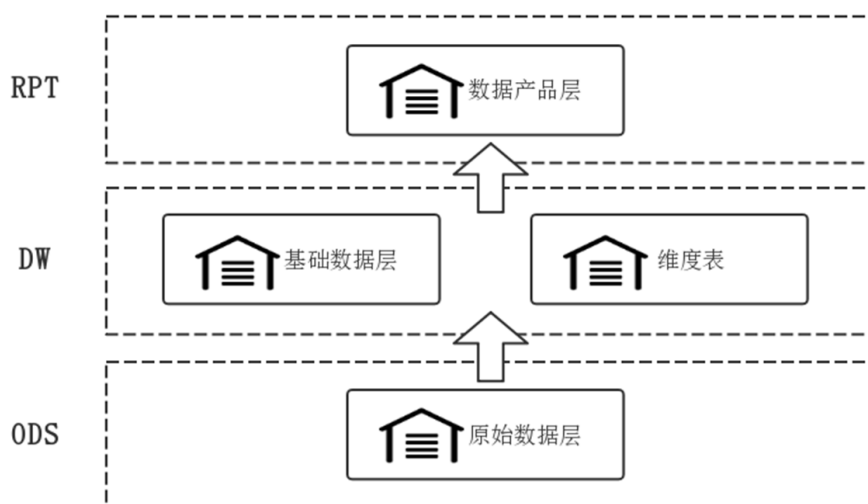
其他两类数据，以及关系型数据库的 Binlog，直接使用了万能的“日志服务 SLS”。WebTracking 支持直接收集 HTML、H5、iOS 和 Android 的日志；Logtail 支持收集服务器上的日志文件，以及关系型数据库的 Binlog。数据都收集过来之后，再定时将数据投递到 MaxCompute 中；

如上两个步骤，完成了三类数据的收集。比业界常见的 Flume+Kafka、Kettle、

Logstash 等方式，上手更快、维护更简单。

2.3 数据仓库

2.3.1 分层



数据仓库的分层模型，大体的思路和网上烂大街的数仓分层原则相似，总体分 ODS、DW、RPT 三层。具体实践的过程中，根据我们的实际情况，慢慢形成了我们自己的风格。

ODS 层，大部分是和数据源中的数据一模一样的，也有极少部分经过了简单的 ETL、或者只截取了与统计有关的字段。数据已采用了其他备份方式，所以这里不再需要使用 MaxCompute 做冷备。

DW 层是最核心的数据仓库层。由于公司技术正在朝着微服务转型，系统、数据库拆分得越来越细，对数据的统计分析很不利。所以我们依靠数据仓库层，将相关的数据放到一起，便于上层的开发、更有利于日常的临时数据需求的快速响应。数据仓

库层的数据结构，不会随着微服务系统和数据的拆分而变化，让系统拆分对于这套离线数据分析的影响终结在这一层，不渗透到更上层。

RPT 层的具体做法，市面上有很多种。根据我们的实际情况，决定采用按业务划分的方式。曾经我们也尝试过按数据产品划分，但是时间长了，出现了几个严重的问题。首先，不同数据产品中对于相同指标的定义混乱，导致各个部门对于数据没有一个统一的概念。其次，技术上的系统拆分的影响范围，随着数据产品的增多而大面积扩大，极易出现修改遗漏的现象。

2.3.2 DATAWORKS

配套 MaxCompute 一起使用的 Dataworks，是一个全能型的可视化工具，集成了几乎一切我们使用 MaxCompute 时需要配套的功能，也解决了很多开源产品中无法解决的痛点，例如：可视化调度、智能监报告警、数据权限控制等。

实际使用时，我们的数据在 MaxCompute 中的流转，全部是通过 MaxCompute SQL 节点和机器学习节点进行的。定时依赖 + 调度依赖 + 跨周期依赖，也让方案的设计变得更灵活。

业务流程是按实际业务模块划分、没有按照数据产品划分，这样可以解决“找任务难”、“不同团队对相同指标的定义不一致”等问题。

当某个业务有变更时，可以快速定位到需要配合修改的任务都有哪些，有效地避免了遗漏。

技术文档的同步更新一直是业界难以解决的痛点，数据字典也不例外。按照业务模块划分了之后，有新增指标时，更容易发现是否已有相同或相似的指标，即使数据字典更新不及时也不会有大影响。

三、实时

3.1 选型原则

团队初始成员均为 Java 出身，并且我们当前没有、未来也不准备拥有自己的 Hadoop 集群。综合考虑，采用了阿里开源的 JStorm 作为核心的流式计算引擎，同时也在尝试业界最新的 Flink，为未来做准备。至于没有使用阿里云商业版的“实时计算”，完全是出于成本考虑，在我们的场景下，自建 JStorm 集群的成本会远低于使用“实时计算”。

与核心的流式计算引擎相配套的中间件及数据存储，使用的全部都是阿里云的产品，开箱即用、省去运维烦恼。

3.2 实践

3.2.1 消息队列

消息队列类的产品，主要使用了“日志服务 SLS”和“消息队列 RocketMQ”两种。

“日志服务 SLS”这款产品，大于等于开源组合 ELK，不仅有日志采集、搜索引擎、分析展示，还有消息队列、监报告警等功能，价格也很合理。尤其，这几个功能的组合，可以轻松实现业务日志告警、nginx 监控等等使用传统方式要开发很久的需求。如果单纯作为消息队列使用，还可以关闭索引，以节省费用。

“消息队列 RocketMQ”的使用，主要看中了“定时延时消息”这一功能，可以实现很多定时延时任务的需求场景。

3.2.2 缓存

Redis，不需要过多介绍。

3.2.3 数据库

阿里云包含了非常多的数据库类产品，根据我们的实际需求，主要使用了以下几款：

(1) RDS for MySQL，与 MySQL 一致，不需要过多介绍；

(2) PolarDb，阿里云自研的云原生数据库，与 RDS 价格一致。对于我们使用者来说，它是一个可以支持更高读并发、单实例容量更大的 MySQL。可以帮助我们建立离线数据中心，也解决了“所有数据库的查询都要先经过 Redis 缓存”的问题，节省了少量 Redis 的费用；

(3) TableStore，这款产品的初衷应该是想要对标开源的 HBase，主要用于单一索引、庞大数据量、单条或小范围检索、高并发、低延时的查询场景。在单条查询时，性能几乎可以媲美 Redis，而且也拥有 TTL 功能。被我们大量使用在用户画像、幂等校验等场景中；

其他产品，例如 DRDS、AnalyticDb，或 MongoDB、Elasticsearch 等，由于目前的场景不需要，所以没有投入使用。

四、数据展示

4.1 选型原则

前端产品的选型原则很简单，由于我们的团队没有专门的前端开发，所以只能选择阿里云的产品、或者免费的、可对接的开源产品。

4.2 实践

阿里云的可视化产品主要有两个：QuickBI 和 DataV。我们都有使用。

QuickBI 主要用于日常的数据展示、分析，帮助运营、产品等部门进行决策；

DataV 主要用于“非交互式”的数据展示场景，例如展会、大屏等。

基于实时计算 (Flink) 的商场实时客流分析系统——上海鸥新

郭华 (付空) 阿里云实时计算高级产品经理

客户简介

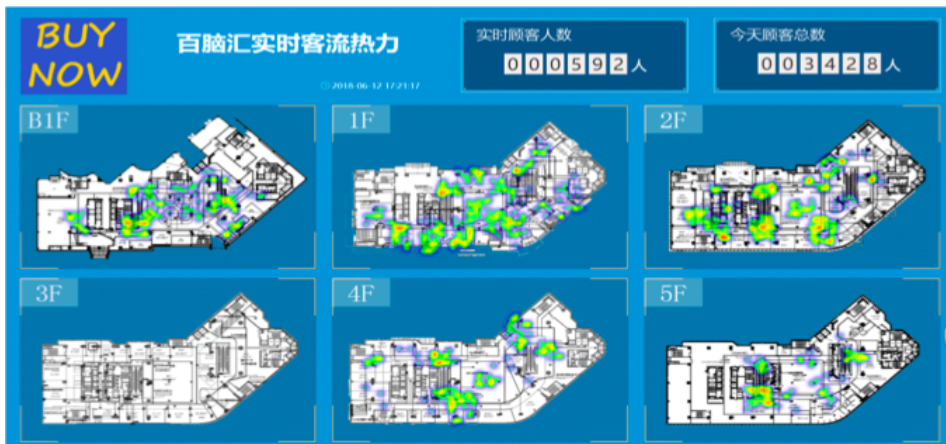
上海鸥新软件有限公司专注于室内定位技术和客流统计与分析的研发，如室内定位引擎、客流统计与分析系统。在用户导入客流系统的同时，为商业零售实体店提供了覆盖，微信上网，定时定地点向客户进行精准化商业信息推送等一体化解决方案。

产品简介

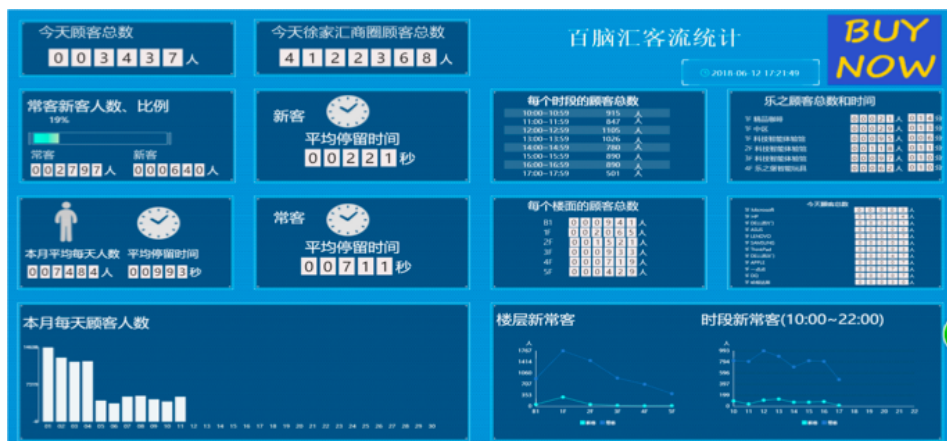
实时客流分析产品主要是基于 Wi-Fi 的客流统计与分析系统，如下图

- 实时热力图

通过实时客流分析系统，制作每个楼层的实时热力图，不同颜色代表客流人数的密集程度。



• 实时客流统计图

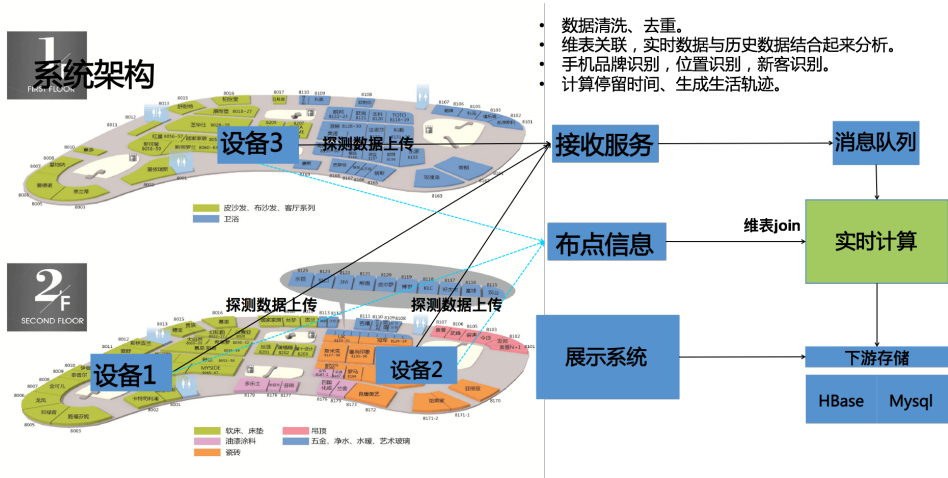


实时客流分析产品主要服务于商场运营方，提供的功能包括：

- 实时客流分析：商场热力图、店铺热力图、客流量、新老客占比、停留时间和客流时间分布等，为运营决策提供数据支持；
- 精准推送：将 Wi-Fi 采集到的地址跟现有数据库进行碰撞，针对碰撞出的用户建立用户画像，根据来店情况进行精准推送；
- 位置定向广告：跟商场线下广告屏幕打通，设定地理围栏与规则，命中规则后个性化推荐广告。

系统架构

整个系统的数据源都是 Wi-Fi，Wi-Fi 设备的布点是系统成功的关键。在 Wi-Fi 铺设的过程中会预先记录好设备的位置（所属楼层，平面坐标，所属店铺等），且根据业务情况来 Wi-Fi 之间是否重叠；如果要精确，需要多点定位，否则尽量 Wi-Fi 的范围不重叠，防止数据互相污染。



数据流程

1. 使用 Wi-Fi 采集设备信息。
2. 把采集的数据经过 SLB 发送到接收服务器。
3. 接受服务器把数据发送到消息队列 (DataHub)。
4. 实时计算订阅 DataHub 数据。
5. 把设备采集到的用户信息跟设备的地理位置信息进行关联。
6. 完成处理，然后把结果写出供下游使用。

实时计算的处理

- 数据清洗、去重。
- 维表关联，用户 mac 地址与设备地理信息关联，实时数据与历史数据关联。
- 手机品牌识别，位置识别，新客识别。
- 计算停留时间、生成轨迹。

说明 数据收集与清洗部分是整个系统的基础，在这些数据的基础上可再进行精准推送和位置广告等服务。

总结

鸥新商场实时客流分析平台涉及多台线下设备(2000台设备),实时计算每秒处理输入30K条数据,每秒输出20K条处理后的数据,整体延迟为秒级,整体收益包括:

- 运维成本:免运维,阿里云提供高保障;
- 对接上下游:直接注册,免开发;
- 开发成本:SQL开发,效率高,门槛低,原来单作业Java开发3天的工作量降低到1天内,且BUG少,整个系统重构只需一周。

这套系统打通了线下与线上,为商场的运营方得到了不同维度的数据支持,提高了运营活动的效果,为在顾客打造更好的购物体验的同时也提升商场的整体营收。

商场实时客流分析系统是IoT技术与大数据实时处理技术结合起来的典型案例。

注:本文部分内容来自鸥新蔡新峰的分析,特此感谢。

如果你有类似场景需求,欢迎联系付空。

基于 MaxCompute 构建 Noxmobi 全球化精准营销系统

杨洋 北京多点在线 高级架构师



本文分享将主要围绕以下三个方面：

- 一、行业及公司背景介绍
- 二、广告业务和系统
- 三、相关技术及 MaxCompute 应用

一、行业及公司背景介绍

行业介绍：什么是数字营销

目前全球广告市场规模约 6000 亿，其中约 30% 为互联网和移动互联网广告，所以这块蛋糕也是很大的。数字营销、互联网广告、在线广告、计算广告、程序化广告等所说的基本上都是同一件事情，这些主要解决的问题就是如何在互联网媒体上投放广告的问题。

行业介绍：什么是数字营销？

行业介绍

什么是数字营销

广告主（需求方）：如何成本更低，效果更好。效果如何定义，品牌or效果。

媒体（供应方）：如何收益更高。长期or短期。

中间商：连接供需，赚差价。规模效应和垄断。

数字营销，应用最新的互联网技术来提高效率。

广告主：减少浪费，让最合适的人看到广告。

媒体：把广告位卖给最需要的人。

中间商：高效撮合交易。

云栖社区 yq.aliyun.com

数字营销主要有三方参与者：广告主、媒体和中间商。

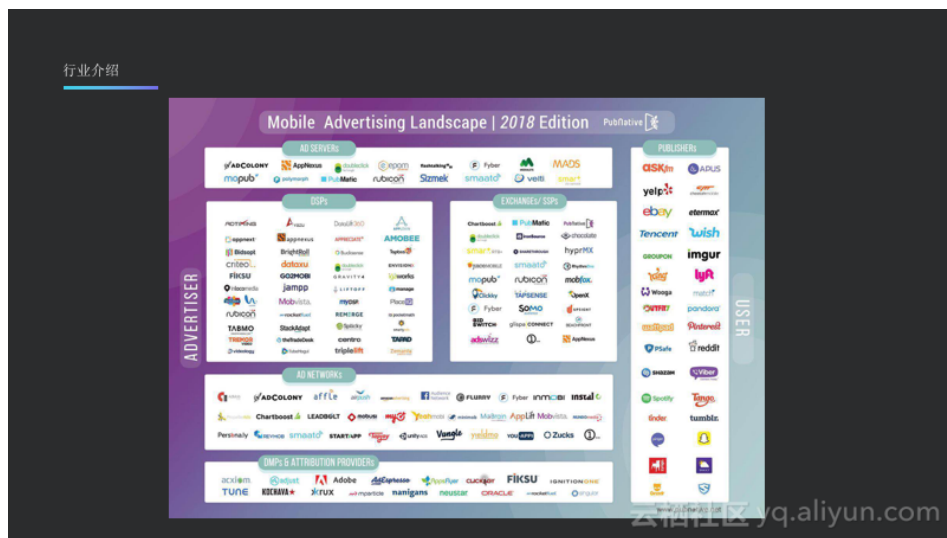
广告主（需求方）：其需求往往是面对流量和库存的，其最大需求就是如何成本更低，效果更好。而效果的定义往往是不同的，一般分为品牌和效果两种。对于品牌而言，比如可口可乐打广告，人们看到广告可能不会立即去买一瓶喝，但是品牌广告会深植在人们的心里，进而产生长期的效应；而效果广告比如在头条上看到一个广告，用户当时看到有兴趣可能会点击之后进行下载了，这样就能发生直接的转化效果，这种就叫做效果广告。

媒体（供应方）：其需求比较简单，就是如何收益更高。收益具有长期和短期之分，媒体往往不能因为短期的高利益而放弃媒体的形象，这样才能让长期的利益更高。

中间商：主要就是通过连接供需来赚差价。有些广告宣传的中间商不赚差价基本上是不可能的，只不过是怎么赚和什么时候赚的问题。赚取差价的主要手段就是规模效应和垄断，而无论是广告主和媒体只要能够垄断就能够具有更高的议价权。

总而言之，数字营销就是应用最新的互联网技术来提高营销效率，会应用比较前

沿的人工智能、机器学习、经济学上的博弈论等。广告主会追求减少浪费，让最合适的人看到广告，这也是数字营销相比传统营销的特点。媒体则是将广告为卖给最需要的人，最需要的人则将会通过出价体现自己的需要。中间商则是高效地撮合交易。如下图所示的是广告行业的简单生态图，图中的 DSP 大概就是前面提到的需求方，也就是广告主。其右边是流量方也就是媒体，下面这部分则是赚取差价的中间商。



二、广告业务和系统

Nox 夜神介绍: Three major product systems support to capture high -LTV users with low cost

Nox夜神介绍

Three major product systems support to capture high-LTV users with low cost

Online advertising

More than 300 million users covered globally

Offline self-media promotion

More than 100,000 KOL signed
500 million users covered

Self-inflicted traffic from pan-entertainment

More than 200 million generic users across the world

云栖社区 yq.aliyun.com

上图中最左边是 Noxmobi 广告平台；中间的是 Influencer，在国外基本上就是网红；最右边的 NoxPlayer 和 NoxCleaner 就是 Nox 自有的媒体流量，多点在线的广告也主要会投放在这些广告上。

Nox 夜神介绍: Manage Every Marketing Stage

Nox夜神介绍

Manage Every Marketing Stage

Pre-Launch	Launch	Post-Launch
Accumulate Followers	Make It Louder	Do The Best
CPL KOL Compatible Test	CPT/CPI KOL User Operations	CPI OMI User Operations

云栖社区 yq.aliyun.com

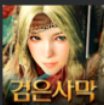
如上图所示的是夜神在发行前期、中期和后期所应该使用的相应产品。

Nox 夜神介绍: Pre-Launch and Launch Marketing

广告位示例

Pre-Launch and Launch Marketing


Black Desert



Device: Android
Area: Korea
Marketing: CPL



CPL
Promotion Period: 40 days
Impressions: 120w+/day
Registers: 300+/day
Total Registers: 12000+

라그나로크M: 영원한 사랑



Device: Android
Area: Korea
Marketing: CPI

CPI
Promotion Period: 15 days
Impressions: 200+ million
Clicks: 310000+
Installs: 20000+
CTR: 0.155%, CVR: 6.4%

云栖社区 yq.aliyun.com

如上图所示的就是 Nox 流量上的广告位，图中右侧是模拟器的启动图和 Launcher 部分的广告位，这些广告也不会非常影响用户的体验。Nox 主要是做海外的生意，而国内和海外的广告市场是不同的，所以也是分开来做的。

NoxInfluencer 示例

NoxInfluencer 示例

Compare with other promotion channels:
cost efficient with massive exposure, 2-3 times high return



Unique features recommended+warm-up
+interaction: achieve the highest CR

KEEP -An fitness app with social attribute

Product : KEEP Promotion Background Influencer No. : 2 Area : America Promotion Period : 12 days Video Length : 10mins Content : Features recommended	Results exhibition views : 116731 (7 days) Direct click : 3545 CTR=3.78% , CR=35.59% Direct user addition : 2316+ (7 days) CPI=\$3 (IOS) CPM=\$60
---	--

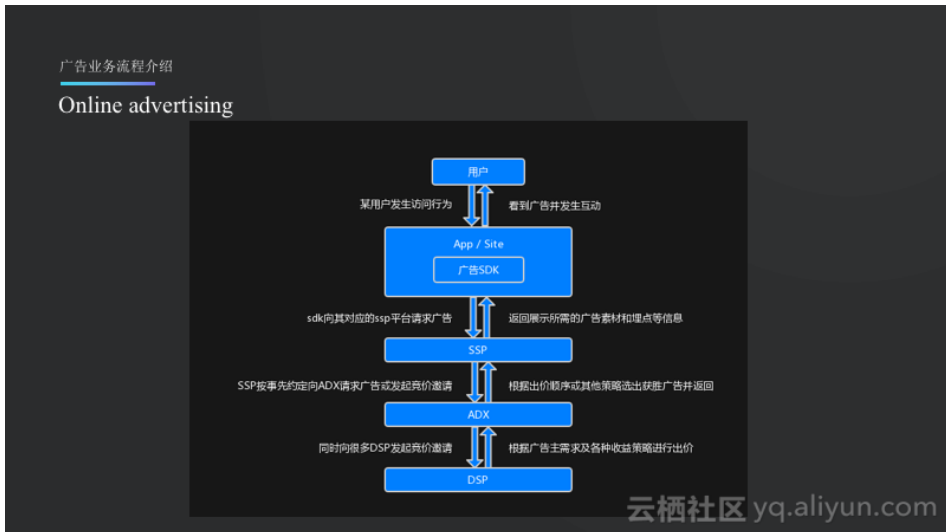



00:45

云栖社区 yq.aliyun.com

如上图所示的是目前主推的 Influencer，目前也有一波流量红利，找网红推送 APP，右图是 Nox 为 Keep 做的广告，比如在健身教练所发出的视频或者直播，给教练一些收益，一起来做视频，也可以看到 Keep 的广告收益也是非常好的。目前 Nox 也对于自己的流量端产品用 Influencer 进行营销，花费大约 10 万，使得自然新增达到了每天 1 万，APP 开发者都知道这其实是一个非常可观的数字，因为自然新增能够将应用在 GooglePlay 上的排名推到很高的位置，使得 NoxCleaner 在一些国家直接达到工具排行榜的前三。

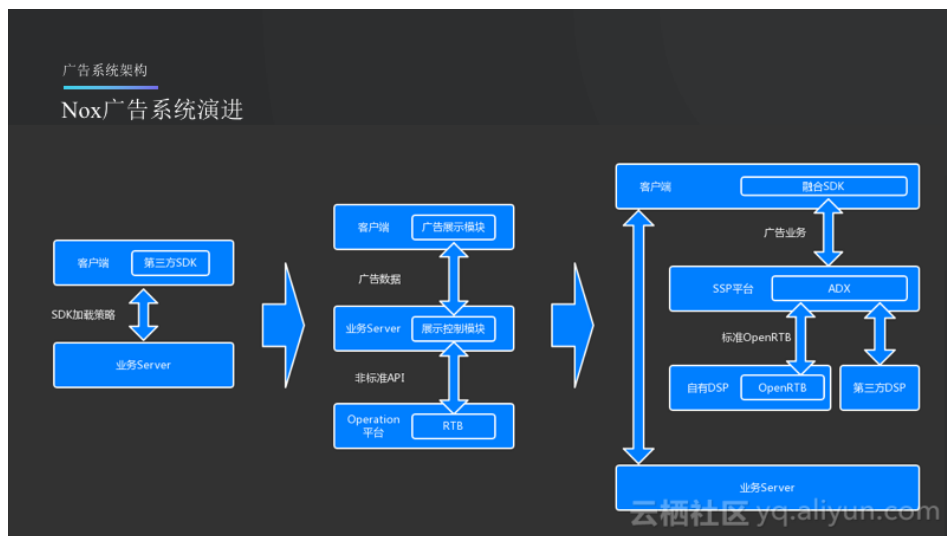
广告业务流程介绍：Online advertising



在线广告业务的基本流程可以从用户的访问开始，用户首先打开一个 APP 或者访问一个页面，当发生了用户访问行为之后，一般在 APP 里面会有广告的 SDK，之后 SDK 就会触发展示然后去请求广告的内容。SDK 一般会与一个 SSP 关联，属于某一家 SSP，然后去 SSP 上请求广告。SSP 再对接 ADX，也就是之前所提到的中间商，其主要负责撮合交易。ADX 又会对接很多的 DSP，并向很多的 DSP 发起竞价邀请，DSP 则将会代表广告主决定是否需要投广告以及广告的出价是多少，并在 ADX 进行多家的比较，做一次竞价的拍卖，价高者得，但是按照第二高价收费，再

将获胜者一步步返回给用户，用户最终就会看到赢得本次竞价的广告。广告后续还会发生一些点击、下载、安装等行为，并进行进一步跟踪。

Nox 广告系统演进



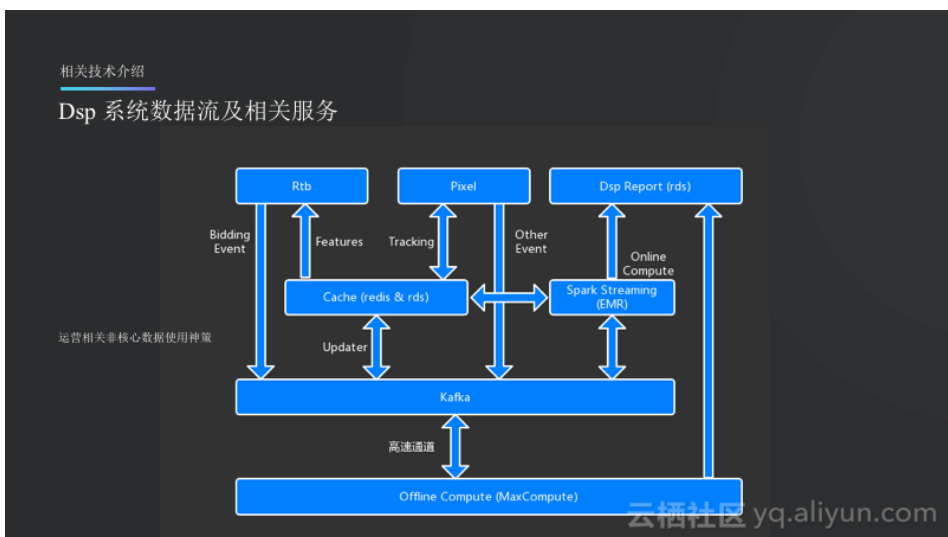
如上图所示的是 Nox 广告系统的演进情况，首先可能与小开发者一样直接集成第三方的 SDK，而第三方的 SDK 的广告是完全不能被控制的，只不过会每天给 Nox 一定的收益，而在这个阶段其实收益也是比较低的。后来，Nox 就自己做了一个广告展示模块，由业务同学去接国外的离线 Offer，在国外某些广告主或者品牌会放出来一些离线的 Offer，这样的收益就比第一阶段高出了很多，这个阶段的系统会自己在客户端做展示模块，而在业务 Server 里面则会有相应的展示控制模块，再往后就是在接到 Offer 之后做一些简单的 CTL 预估，之后再做一些比较并将收益比较高的投放出去。第三个阶段系统展现图中的中心部分就是广告业务，最外层的两个模块就是原来的业务系统，最外层两个模块之间的连接就会变得很细，基本上只做流量切分的工作，而中心部分就是前面所提到的 DSP、SSP 以及集成的 ADX 模块，从而形成一个完整的广告生态。在这里，除了自己本身的 DSP 还会和第三方的 DSP 进行竞价保证流量的收益最大化。此外，SDK 还可以做成融合的，虽然其不开放，

但是将做成展现其他第三方 SDK 也会产生一些收益。

三、相关技术及 MaxCompute 应用

DSP 系统数据流及相关服务

SSP 和 ADX 相对而言比较简单，因此在这部分将会重点讲述一下 DSP 系统数据流。Rtb 就是 Runtime Bidding，也是 DSP 里面比较重要的一个部分，主要是做实时竞价的。所以广告流程一般而言就是从 Rtb 模块开始，在竞价的过程中会产生 Bidding 的 Event，并将 Bidding 的 Event 都输入到 Kafka 里面，Kafka 里面会有订阅的消息，一直更新大的 Cache，这里面的数据会通过流式计算做成 feature 再反馈给 Rtb 系统。Pixel 就是事件服务，比如发生了展现或者安装之后就会访问到这个服务，而这个服务将会 Tracking 到本次广告的所有 Session，这样的数据也会流入 Kafka，同样由 Updater 和流式计算进行处理。而从 Kafka 里面会另外分出来一只数据流通过高速通道回流到中心节点，也就是 MaxCompute 上面。MaxCompute 会进行一些离线的报表计算和特征，报表就会输出到 DSP Report 上面（比如 RDS）。



其实 Nox 设计的所有广告的核心数据将会走这样的一套比较复杂的流程，而一些运营相关的非核心数据则会主要使用了自己搭建在阿里云上的神策这款 BI 服务，其也属于企业级的应用服务。

流式计算 Spark Streaming 应用

流式计算 Spark Streaming 主要用于实现实时的报表以及实时特征的计算。因为业务的主要要求是必须稳定并且能够实现 7*24 小时的可用。可以接受秒级延迟，比如广告投出去了，晚 10 秒钟展现在报表里也是没问题的。可根据吞吐量横向扩展，比如突然新接了几家 SSP，突然变得流量很大，不能在这个时候让系统挂掉。此外，因为业务在全球都有，所以需要全球的聚合任务，需要通过一个平台看到各个国家的数据。



Nox 选择的方案就是：Spark Streaming 能够将上面几项需求全部满足，另外就是配合 Kafka、RDS 以及 Redis 做输出。在部署上面，要实现小集群独占，这里所用到的就是阿里云 EMR，其可以帮助客户托管集群，Nox 只需要在阿里云 EMR 上面申请一个小集群，比如三到五台机器，这些机器申请之后就不再释放掉了，会一直独占着，并且 7*24 小时地跑流式计算任务。原始日志压缩流式回传，这

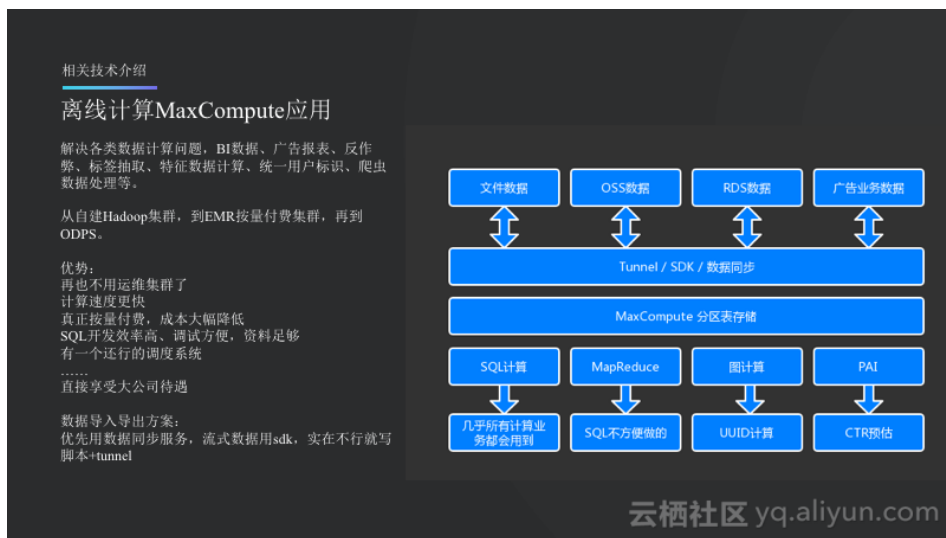
个是因为 Nox 在各个数据中心都有 Bidder 或者 Pixel 的服务，会产生很多数据，之前的一种方案是在每个中心先将数据计算成半成品，之后在进行回传，这样所用的带宽就会比较小，但是如果采用这样方案，那么所有的功能都需要开发两套，在本地先计算，之后传回来再进行聚合计算，这样就会比较复杂，因此最终决定将日志进行压缩，以流式方式进行回传，这样的方案在验证之后发现所占的带宽不是很大，而因为是流式传输，因此带宽也比较平稳，虽然这里所用的带宽属于高速通道带宽，因此成本也可以接受。而压缩则使用了 Kafka，其是能够支持压缩协议的。此外，中心节点部署能够方便开发。

上图中最底层就是阿里云的 EMR 托管服务，在其上是 DSP 平台和 SSP 平台，他们的集群是分开的，如果流量特别大，某一个平台被打挂掉了，另外一个平台是不会受到影响的。而托管服务的好处就是能够托管很小的集群，对于企业而言也没有什么成本。Kafka 里面输入的就是 Event 的 Topic，之后还会输出回 Kafka，这样 Updater 再将 Kafka 里面的数据放到 Redis 或者 RDS 用于构建模型和计算报表。这样的设计的唯一问题就是比较依赖于高速通道，这样稳定性和扩展性就有可能受到限制。

离线计算 MaxCompute 应用

离线计算部分，Nox 主要使用了 MaxCompute。几乎使用了 MaxCompute 来解决各类数据计算问题，BI 数据、广告报表、反作弊、标签抽取、特征数据计算、统一用户标识、爬虫数据处理等。其实一开始，Nox 也是自建 Hadoop 集群，购买了阿里云的 ECS 搭建集群，从最开始的 6 台一直到后来的十几台，这时候实在扛不住了，机器经常宕机，因为使用的是 Spark，因此内存很容易占满，某一天用户突然增多了，数据就没了。此外，这样的成本也非常高，因为当时主要运行 BI 数据，所以基本上都是在晚上运行的，而白天机器则处于空闲状态，因此成本很高。后来采用了 EMR 的按量付费集群，晚上申请之后跑数据，但是白天能够释放掉，但是这样的过程则是比较漫长的，需要 10 到 20 分钟。后来 Nox 开始接触到 MaxCompute，使用起来非常好，其带来了许多优势。首先，不再需要运维集群了，此外其计算速度很快，虽然说 Spark 的计算速度很快，但是小集群的 Spark 和大集群的 Hadoop

是无法比拟的，所以大集群的 Hadoop 其实计算速度是很快的。MaxCompute 是真正的按量付费，因此成本也能够大大降低，而自建 Hadoop、使用 EMR 以及使用 MaxCompute 的成本是成量级降低的。差距也是非常大的。主要使用 SQL 开发，效率比较高，也便于调试，文档也比较清晰。此外，MaxCompute 还提供了一个还不错的调度系统，如果是自己搭建这样调度系统还是比较困难的。



对于数据的导入和导出而言，因为 Nox 有很多海外的服务，有些服务是不能覆盖到的。所以 Nox 采取的策略是优先使用数据同步服务，而流式数据则使用 SDK，当数据同步和 SDK 都不合适就写脚本 +tunnel 导入和导出数据。

如上图所示的文件数据主要是爬虫，因为一些服务的日志是达到 OSS 上的，并且有一些外部数据也是先上传到 OSS 上面的。RDS 则是什么都有的，广告业务数据就是前面所提到的，这些数据会选择合适的方式统一进入到 MaxCompute 的分区表里面。SQL 计算基本上都会用到，MaxCompute 则是在 SQL 写起来很费力或者运行很慢的情况下使用。图计算使用的并不多，只是会在计算同一用户 UUID 的情况下使用，这里应用了最小连通域的算法，比如一个用户使用了多个设备，则需要将这些设备统一地关联到同一个用户身上。而 PAI 平台对于广告的 CTL 预估非常重要。

特征计算和标签抽取

如下图所示的某第三方 DMP 的对外标签体系的示例，大概分了几类，比如人口学、设备信息等大类，在每个大类下面还会有多个标签。特征一般而言就是连续值，标签则是指将连续值做一些规则之后所打的标签。举例而言，定义特征，最近一周内活跃天数，则有 0~7 的取值。而定义标签规则，则是一周内活跃 0 天、1 天、2~3 天、4~5 天、6~7 天的分别是不活跃、低活跃、中活跃、高活跃、极高活跃用户。当在做好定义之后，就要看大家的 SQL 写的是好是坏了。之前在不支持 with 的时候，SQL 代码一般都要写很大一堆，而且很难改动。此外，在写 SQL 的时候注意代码分隔还是很重要的。另外的一些建议就是优先使用内建函数，虽然一些内建函数和 UDF 的功能差不多，可能一个 UDF 能够实现两三个内建函数的功能，但是效率却相差了很多，虽然使用内建函数会让代码看起来丑一些，但是绝对比 UDF 运行速度要快得很多，所以在内建函数无法满足需求的时候再去考虑 UDF，实在不行就可以用 MapReduce 实现，比如对一大批特征做等频离散化。

相关技术介绍

特征计算和标签抽取

大部分特征和标签使用 SQL 计算

定义特征，最近一周内活跃天数，则有 0~7 的取值

定义标签规则，例如：
一周内活跃 0 天、1 天、2~3 天、4~5 天、6~7 天的分别是不活跃、低活跃、中活跃、高活跃、极高活跃用户

做好定义，然后就是展现 SQL 技巧的时候了
多用 with，注意代码风格

优先使用内建函数，无法满足的时候考虑使用 UDF、UDAF、UDTF，还不行或者跑的太慢的时候使用 MapReduce，比如对一大批特征做等频离散化

尚未实现平台化，还需写代码，好像有个服务能实现上面的功能，待试用

某第三方 DMP 对外标签体系（部分示例）

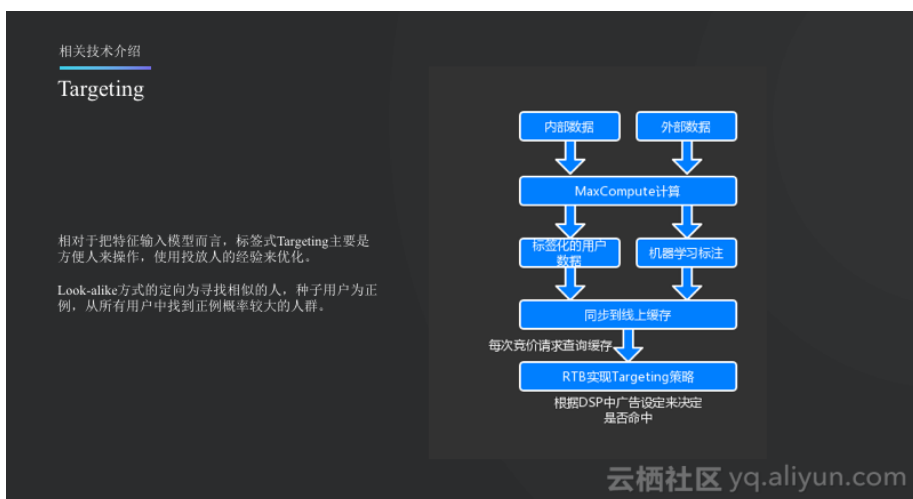
标签大类	标签	标签值（示例）
人口学	性别	男、女、未知
人口学	年龄	年龄段
人口学	消费能力	超高、高、中、低
人口学	作息时间	作息
设备信息	激活时间	时间戳
设备信息	设备型号	设备品牌&型号
设备信息	操作系统	操作系统版本
设备信息	操作系统语言	语言编码
设备信息	真机概率	概率值（数值）
设备信息	羊毛党预警	概率值（数值）
设备信息	视频会员	是否
设备信息	所属运营商	运营商
设备信息	3/4G 触点占比	占比
关联设备	家庭-OTT	OTT-ID
关联设备	移动设备	设备ID
关联设备	电脑	CookieID
关联设备	同性人	SuperID
地理信息	常住城市	城市
地理信息	工作区域	区域
地理信息	工作流动性	固定/同城流动/跨城商旅
地理信息	居住区域	小区
地理信息	休息时段活动区域	区域
地理信息	深夜活动	经常、偶尔、无

云栖社区 yq.aliyun.com

上述这些都是 DMP 的标准功能，但是 Nox 目前还没将其实现平台化，都是使用标签写的。而阿里云上有标签服务，目前也在考虑使用。

Targeting

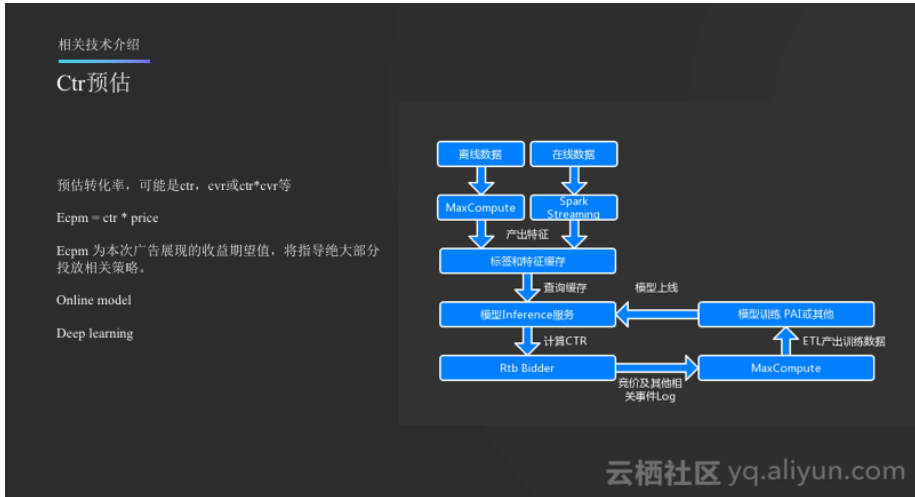
所谓 Targeting 就是人群定向，相对于把特征输入模型而言，标签式 Targeting 主要是方便人来操作，使用投放人的经验通过标签定向的方式来优化表达。比如在广告投放初期可以比较好地缩放盲打范围。另外一种则是 Look-alike 方式，Look-alike 方式的定向为寻找相似的人，种子用户为正例，从所有用户中找到正例概率较大的人群。以上就是定向的两种主要方式。



如上图所示的定向的主要做法就是将内部和外部的数据输入到 MaxCompute 里面，经过各种计算将标签化的数据或者用机器学习标注好的数据同步到线上并缓存好。之后在进行实时竞价 RTB 里面查询缓存，命中之后就在 DSP 里面由广告主配置，命中了就投放广告，否则就不投放。

Ctr 预估

Ctr 预估是 Nox 投入比较多的一项工作。Ctr 预估并不一定是预估 Ctr，还可以去预估 Cvr 甚至是 Ctr 和 Cvr 的乘积，这些统称为 Ctr 预估。其作用是首先计算 Ecpm 的值，Ecpm 就是每次展现的期望收益，期望是一个概率论上的概念，其等于用单价乘上本次收益可能的概率。Ecpm 也将指导绝大部分投放相关策略。



如上图所示，主要分为两条线，一条是离线数据会走 MaxCompute，而在线则会走 Spark Streaming。总体最后会输入到标签和特征的大 Cache 里面，Cache 中的数据有一部分直接加载到内存里面，另外一部分比如用户特征无法加入内存就会在 Inference 服务查询缓存，这里就会组合出一个特征的向量，用来计算 Ctr 的值，并将值返回给 Rtb 的 Bidder，在 Bidder 做一系列的策略，最终得出竞价的决策。竞价完成之后，是否参与、是否竞价成功以及是否展现等日志都会灌入回 MaxCompute 或者 Spark Streaming 进行模型训练，最终对已有模型进行更新，形成一个整体的闭环。Online model 需要使用 Spark Streaming，而 Deep learning 就需要用 TensorFlow 等了。

Pacing

Pacing 比较复杂一些，其就是不止考虑单次展现的收益，而要在单次竞价时考虑对全局收益的影响，比如考虑一天之内总收益如何，比如可能将转化率最高的 Offer 在一天开始的两小时内都投完了，但是其他的 Offer 都没有投出去，这样计算下来总收益并不如将全部广告都投完的收益高。Pacing 的整体思路就是通过流量分层和分时的统计和预估，用数学方法来保证全局收益的最大化。Nox 则根据 Yahoo 的论文实现了自己的方案，这里面最核心的就是将 Ctr 估算准确，并将分层

和分时的各种统计值计算好，然后按照其策略执行即可。

相关技术介绍

Pacing

不止考虑单次展现的收益，而要在单次竞价时考虑对全局收益的影响。

通过对流量分层和分时的统计和预估，用数学方法来保证收益最大化。

方案来源于Yahoo

Handwritten notes on a piece of paper detailing mathematical models for pacing. The notes include a flowchart starting with 'Target: R_t ' and '逐时且分层'. It contains several equations: $R = C^{(t)} - C^{(t-1)}$, $C = B^{(t)} + \frac{B_m - \sum_{i=1}^K B_i^{(t)}}{K-m}$, and trial rate formulas. There are also notes about 'CINR 策略', 'CPL 分层', and 'goal'.

Nox 目前也在寻求更多的合作伙伴，希望更多与具有出海意向的开发者进行深度合作。

Business Partners

Business partnerships with globally renowned gaming games companies

A grid of logos for various gaming companies including Tencent Games, Kakao, Netmarble, Garena, ENJOY GAME, Nexon, KUNLUN, 巨人网络, Bilibili, GAMEVIL, 蜗牛游戏, 心动网络, 4:33 CREATIVE LAB, 盛大游戏, DIGITAL SKY, 猎豹移动, 完美世界, IGG, G-bits, and PARAMIDA Games.



阿里技术

扫一扫二维码图案，关注我吧



阿里云实时计算



MaxCompute 开发者交流钉钉群



DataWorks 开发者交流钉钉群



扫码关注阿里技术