

开发者社区 × 飞天大数据平台

APISARA BIG DATA PLATFORM

# 大数据工程师 必读手册

像阿里巴巴一样  
玩转大数据

BIG DATA

十位阿里巴巴大数据专家深度分析  
飞天大数据平台八款产品最新玩法  
2019不容错过的大数据手册

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



阿里巴巴大数据计算



DataWorks 开发者交流钉钉群



扫码关注阿里技术

# 目录

<b>开篇</b>	<b>4</b>
AI 加持的阿里云飞天大数据平台技术揭秘	4
<b>计算存储引擎</b>	<b>19</b>
飞天大数据平台计算引擎 MaxCompute 最新特性	19
飞天大数据平台实时计算 Flink on Kubernetes 最新特性	36
飞天大数据平台 E-MapReduce 4.0 最新特性	43
<b>大数据 &amp; AI 开发平台</b>	<b>55</b>
飞天大数据平台智能开发云平台 DataWorks 最新特性	55
飞天大数据平台机器学习 PAI 最新特性	77
<b>搜索与推荐</b>	<b>88</b>
飞天大数据平台 OpenSearch 最新特性	88
飞天大数据平台 Elasticsearch 最新特性	101
飞天大数据平台智能推荐 AIRec 最新特性	110

# 开篇

## AI 加持的阿里云飞天大数据平台技术揭秘



**关涛**

阿里云智能计算平台  
事业部研究员



**徐晟**

阿里云智能计算平台  
事业部资深技术专家

摘要: 2019 云栖大会大数据 & AI 专场, 阿里云智能计算平台事业部研究员关涛、资深专家徐晟来为我们分享《AI 加持的阿里云飞天大数据平台技术揭秘》。本文主要讲了三大部分, 一是原创技术优化 + 系统融合, 打破了数据增长和成本增长的线性关系, 二是从云原生大数据平台到全域云数仓, 阿里开始从原生系统走入到全域系统模式, 三是大数据与 AI 双生系统, 讲如何更好的支撑 AI 系统以及通过 AI 系统来优化大数据系统。

说到阿里巴巴大数据, 不得不提到的是 10 年前王坚博士率领建构的飞天大数据平台, 十年磨一剑, 今天飞天大数据平台已是阿里巴巴 10 年大平台建设最佳实践的结晶, 是阿里大数据生产的基石。飞天大数据平台在阿里巴巴集团内每天有数万名数据和算法开发工程师在使用, 承载了阿里 99% 的数据业务构建。同时也已经广泛应用于城市大脑、数字政府、电力、金融、新零售、智能制造、智慧农业等各领域的大数据建设。

大家知道飞天大数据平台是从 2009 年发展到今天的系统。飞天平台从发展的角度来讲经历了 2 个阶段, 第一个阶段是从 2009 年到 2015 年, 那个阶段我们主要解决的是稳定性问题、可用的问题和异用的问题, 从这个角度, 到 2015 年的时候我们完成了一个非常有名的项目叫登月, 登月就是把阿里巴巴所有的数据都融合在一起

放到那个平台上，我们构建了中台体系，在那个时候阿里巴巴的数据进入到腾飞的过程，每月大概是几百个 PB 的数据涌入到这样的平台里来，那个时候的数据比较小，大概是数千台的服务器、几百个基架的状态。当时每个月就会有那么多的服务器开始摆进来，从 2015 年开始，也就是从海量的数据爆炸开始，系统走过这个稳定时期之后，后续的关键问题是什么？



在 2015 年的时候，我们开始关注到数据的海量增长对系统带来了越来越高的要求，随着深度学习的需求增长，数据和数据对应的处理能力是制约人工智能发展的关键问题，我们在给客户聊到一个摆在每个 CIO/CTO 面前的现实问题——如果数据增长 10 倍，应该怎么办？图中数字大家看得非常清晰，非常简单的拍立淘系统背后是 PB 的数据在做支撑，阿里小蜜客服系统有 20 个 PB，大家每天在淘宝上日常使用的个性化推荐系统，后台要超过 100 个 PB 的数据来支撑后台的决策，10 倍到 100 倍的数据增长是非常常见的。从这个角度上来讲，10 倍的数据增长通常意味着什么问题？

第一，意味着 10 倍成本的增长，如果考虑到增长不是均匀的，会有波峰和波谷，可能需要 30 倍弹性要求；第二，实际上因为人工智能的兴起，二维结构性的关系型数据持续性增长的同时，带来的是非结构化数据，这种持续的数据增长里面，一半的增长来自于这种非结构化数据，我们除了能够处理好这种二维的数据化之后，我们如何做好多种数据融合的计算？第三，阿里有一个庞大的中台团队，如果说我们的数据增长了 10 倍，我们的团队是不是增长了 10 倍？如果说数据增长了 10 倍，数据的关系复杂度也超过了 10 倍，那么人工的成本是不是也超过了 10 倍以上，我们的飞

天平台在 2015 年后就是围绕这三个关键性的问题来做工作的。

## 原创技术优化 + 系统融合

当阿里巴巴的大数据走过 10 万台规模的时候，我们已经走入到技术的无人区，这样的挑战绝大多数公司不一定能遇到，但是对于阿里巴巴这样的体量来讲，这个挑战是一直摆在我们面前的。

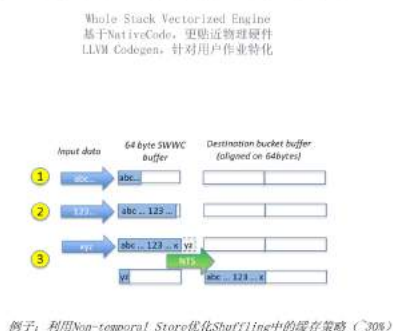


大家可以看到，2015 年的时候，我们整个的体系建立起来之后，就开始做各种各样的 Benchmark，比如 2015 年 100TB 的 Sorting，2016 年我们做 Cloud-Sort，去看性价比，2017 年我们选择了 Bigbench。如图是我们最新发布的数据，在 2017、2018 和 2019 年，每年都有一倍的性能提升，同时我们在 30TB 的规模上比第二名的产品有一倍的性能增长，并且有一半的成本节省，这是我们的计算力持续上升的优化趋势。



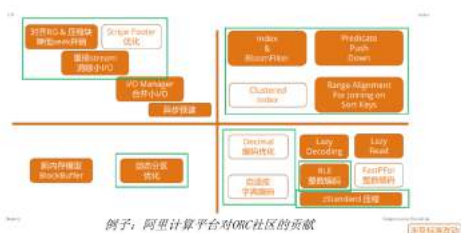
那么，计算力持续升级是如何做到的？如图是我们经常用到的系统升级的三角理论，最底层的计算模型是高效的算子层和存储层，这是非常底层的基础优化，往上面要找到最优的执行计划，也就是算子组合，再往上是新的方向，即怎么做到动态调整与自学习的调优。

### 单一算子和引擎框架的极致优化



### 存储层优化（AliORC）

兼容开源Apache ORC的增强版列存格式  
支持大数据数仓（ODPS）、机器学习（PAI）和分析型查询等  
性能卓越：读取性能对比开源Java ORC均快50%  
是ORC社区过去两年最大贡献者（2W+行代码，多个标准改动）



我们先来看单一算子和引擎框架的极致优化，我们用的是比较难写难维护的框架，但是因为它比较贴近物理硬件，所以带来了更极致的性能追求。对于很多系统来说可能5%的性能提升并不关键，但对于飞天技术平台来讲，5%的性能提升就是5千台的规模，大概就是2~3亿的成本。如图做了一个简单的小例子做单一算子的极致优化，在shuffle子场景中，利用Non-temporal Store优化shuffling中的缓存策略，在这样的策略上有30%的性能提升。

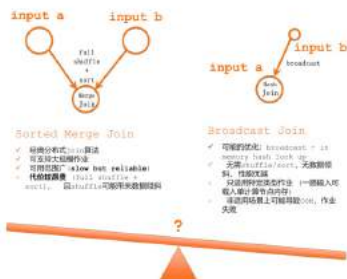
除了计算模块，它还有存储模块，存储分为4个象限。一四象限是存储数据本身的压缩能力，数据增长最直接的成本就是存储成本的上升，我们怎么做更好的压缩和编码以及indexing？这是一四象限做的相关工作；二三象限是在性能节省上做的相关工作，我们存储层其实是基于开源ORC的标准，我们在上面做了非常多的改进和优化，其中白框里面都有非常多的标准改动，我们读取性能对比开源Java ORC均快50%，我们是ORC社区过去两年最大贡献者，贡献了2W+行代码，这是我们在算子层和存储层的优化，这是最底层的架构。

## 灵活的算子组合

Dynamic Execution

4种Join - SortMergeJoin, NestLoopJoin, HashJoin, BroadcastJoin  
3种Shuffling - MRShuffling, GraySortShuffling, ShufflingService  
3种作业运行模式 - Batch, ServiceMode, BubbleMode  
多种硬件支持 - CPU, GPU, FPGA\*  
多种存储介质支持 - SATA, SSD, Optane\*

总计超过100种原子算子支持，在静态优化的基础上，**具备执行阶段动态选择**



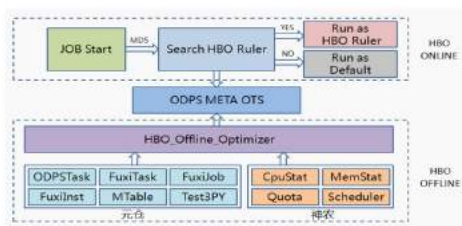
但是从另外一个层面上来讲，单一的算子和部分的算子组合很难满足部分的场景需求，所以我们就提到灵活的算子组合。举几个数字，我们在Join上有4种模式，有3种Shuffling模式提供，有3种作业运行模式，有多种硬件支持和多种存储介质支持。图右是怎样去动态判别Join模式，使得运算效率更高。通过这种动态的算子组合，是我们优化的第二个维度。

## 基于历史信息的自学习回归优化

History/Learn Based Optimization

飞天平台在CBO的基础上，基于历史信息自学习来优化新作业的运行  
(CBO+HBO)

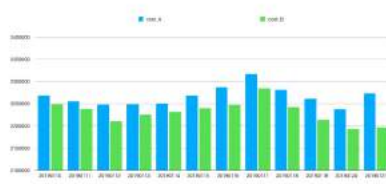
上线阿里内部系统三年来，**提升资源利用率1倍以上**



## 全面自适应优化

Adaptive Execution

针对特定数据采用不同的编码和压缩方式  
自动数据分层和Caching  
自动冷数据归档



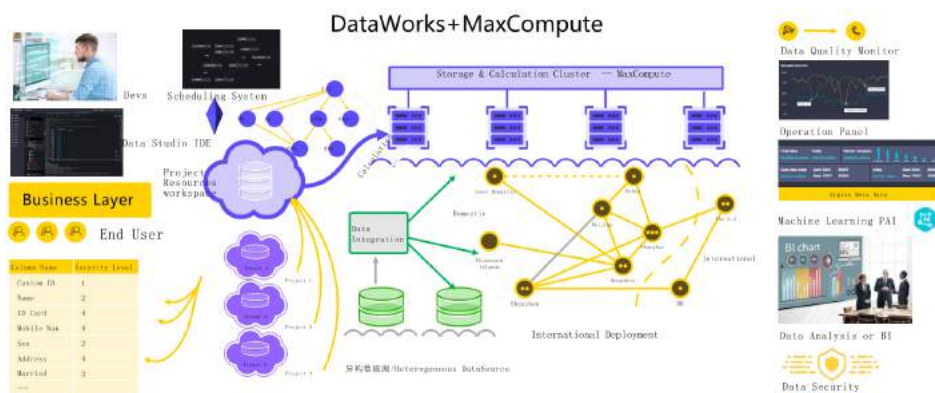
例子：通过自动自适应冷归档，过去一年为阿里经济体节省超过1亿成本

从引擎优化到自学习调优是我们在最近1年多的时间里花精力比较多的，我们在考虑如何用人工智能及自学习技术来做大数据系统，大家可以想象学骑自行车，刚开始骑得不好，速度比较慢甚至有的时候会摔倒，通过慢慢的学习，人的能力会越来越好。对于一个系统而言，我们是否可以用同样的方式来做？当一个全新的作业提交到这个系统时，系统对作业的优化是比较保守的，比如稍微多给一点资源，那么我选择的执行计划会相对比较保守一点，使得至少能够跑过去，当跑过之后就能够搜集到信

息和经验，通过这些经验再反哺去优化数据，所以我们提出一个基于历史信息的自学习回归优化，底层是如图的架构图，我们把历史信息放在 OFFLINE system 去做各种各样的统计分析，当作业来了之后我们把这些信息反哺到系统之中去，让系统进行自学习。通常情况下，一个相似的作业大概跑了 3 到 4 次的时候，进入到一个相对比较优的过程，优指的是作业运行时间和系统资源节省。这套系统大概在阿里内部 3 年前上线的，我们通过这样的系统把阿里的水位线从 40% 提升到 70% 以上。

另外图中右侧也是一个自学习的例子，我们怎么区分热数据和冷数据，之前可以让用户自己去 set，可以用一个普通的 configuration 去配置，后来发现我们采用动态的根据作业方式来做，效果会更好，这个技术是去年上线的，去年为阿里节约了 1 亿 + 人民币。从以上几个例子上来讲引擎层面和存储层面做的极致性能优化，性能优化又带来了用户成本的降低，在 2019 年 9 月 1 号，飞天大数据平台的整体存储成本降低了 30%，同时我们发布了基于原生计算的新规格，可以实现最高 70% 的成本节省。

以上都是在引擎层面的优化，随着 AI 的普惠优化，AI 的开发人员会越来越多，甚至很多人都不太具备代码的能力，阿里内部有 10 万名员工，每天有超过 1 万员工在飞天大数据平台上做开发，从这个角度上来讲，不仅系统的优化是重要的，平台和开发平台的优化也是非常关键的。



计算引擎对大家来说看不见摸不着，我们要去用它肯定希望用最简单的方式，先来看一下 Maxcompute 计算引擎。首先我们需要有用户，用户怎么来使用？需要资源隔离，也就是说每个用户在系统上面使用的时候会对应着账号，账号会对应着权限，这样就把整套东西串联起来。今天我的用户怎么用？用哪些部分？这是第一部分。第二部分是开发，开发有 IDE，IDE 用来写代码，写完代码之后提交，提交之后存在一个调度的问题，这么多的资源任务顺序是什么？谁先谁后，出了问题要不要中断，这些都由调度系统来管，我们的这些任务就有可能在不同的地方来运行，可以通过数据集成把它拉到不同的区域，让这些数据能够在整个的平台上跑起来，我们所有的任务跑起来之后我们需要有一个监控，同时我们的 operation 也需要自动化、运维化，再往下我们会进行数据的分析或者 BI 报表之类的，我们也不能够忘记 machine learning 也是在我们的平台上集成起来的。最后，最重要的就是数据安全，这一块整个东西构成一个大数据引擎的外沿 + 大数据引擎本身，这一套我们称之为单引擎的完备大数据系统，这一套系统我们在 2017 年的时候就具备了。



2018 年的时候我们做什么？2018 年我们在单引擎的基础上对接到多引擎，我们整个开发链路要让它闭环化，数据集成可以把数据在不同的数据源之间进行拖动，我们把数据开发完之后，传统的方式是再用数据引擎把它拖走，而我们做的事情是希望这个数据是云上的服务，这个服务能够直接对用户提供的数据，而不需要把数

据整个拖走，因为数据在传输过程中有存储的消耗、网络的消耗和一致性消耗，所有的这些东西都在消耗用户的成本，我们希望通过数据服务让用户拿到他想要的东西。再往下，如果数据服务之上还有自定义的应用，用户还需要去建一个机房，搭一个 web 服务，然后把数据拿过来，这样也很麻烦，所以我们提供一个托管的 web 应用的云上开发平台，能够让用户直接看到所有的数据服务，在这个方向上来说，我们就可以构建任意的数据智能解决方案。

## 全域大数据平台产品架构

One platform for all the data



到 2019 年，我们会把理念再拓展一层，首先对于用户来说是用户交互层，但是用户的交互层不仅仅是开发，所以我们会把用户分成两类，一部分叫做数据的生产者，也就是写任务、写调度、运维等，这些是数据的生产者，数据的生产者做好的东西给谁呢？给数据的消费者，我们的数据分散在各个地方，所有的东西都会在治理的交互层对数据的消费者提供服务，这样我们就在一个新的角度来诠释飞天大数据平台。除了引擎存储以外，我们有全域的数据集成进行拉动，统一的调度可以在不同的引擎之间来切换协同工作，同时我们有统一的元数据管理，在这之上我们对数据的生产者和数据的消费者也都进行了相应的支持，那么这个整体就是全域的大数据平台产品架构。

## 云原生平台到全域云数仓

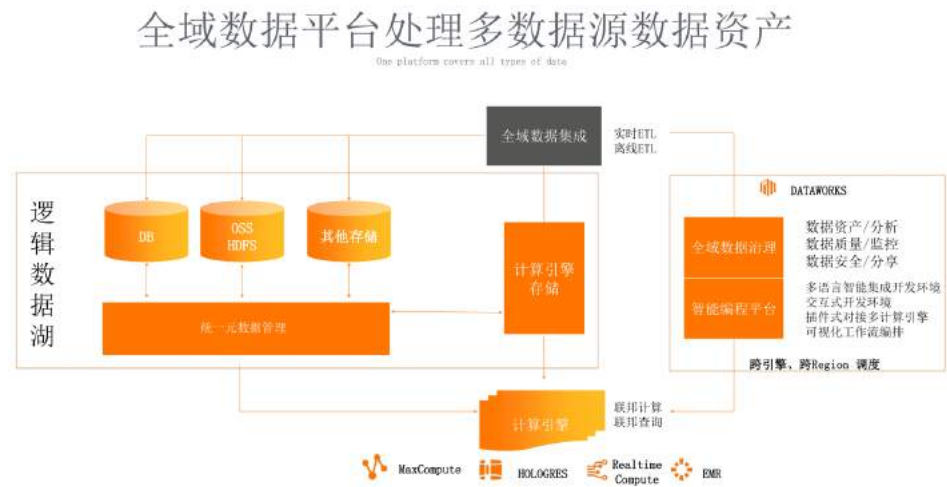
我们整个平台都是云原生的，云原生有哪些技术呢？



飞天大数据平台在 10 年前就坚持云原生的数据，云原生意味着三件事情，第一开箱即用、不用不花钱，这个和传统的买硬件方式有非常大的不同；第二我们具备了秒级自适应的弹性扩展，用多少买多少；第三因为是云上的框架，我们很多运维和安全的東西由云自动来完成了，所以是安全免运维的。从系统架构上讲，飞天大数据包括传统的 CPU、GPU 集群，以及平头哥芯片集群，再往上是我们的伏羲智能调度系统和元数据系统，再往上我们提供了多种计算能力，我们最重要的目标就是通过云原生设计把 10 万台在物理上分布在不同地域的服务器让用户觉得像一台计算机。我们今天已经达到了 10 年前的设计要求，具备了更强的服务扩展能力，能够支撑 5 到 10 年的数据进步的发展。

我们充分利用云原生设计的理念，支持大数据和机器学习的快速大规模弹性负载需求。我们支撑 0 ~ 100 倍的弹性扩容能力，去年开始，双十一 60% 的数据处理量来自于大数据平台的处理能力，当双 11 巅峰来的时候，我们把大数据的资源弹回来让给在线系统去处理问题。从另外一个角度来讲，我们具备弹性能力，相比物理的 IDC 模式，我们有 80% 成本的节省，按作业的计费模式，我们提供秒级弹性伸缩的同时，不使用不收费。相比自建 IDC，综合成本只有 1/5。除了坚持原生之外，我们最近发现，随着人工智能的发展，语音视图的数据越来越多了，处理的能力就要加

强，我们要从二维的大数据平台变成全域的数据平台。



如图所示，业界有一个比较火的概念叫数据湖，我们要把客户多种多样的数据拿到一起来进行统一的查询和管理。但是对于真正的企业级服务实践，我们看到一些问题，首先数据的来源对于客户来说是不可控的，也是多种多样的，而且很大程度上没有办法把所有的数据统一用一种系统和引擎来管理起来，在这种情况下我们需要更大的能力是什么呢？我们今天通过不同的数据源，可以进行统一的计算和统一的查询和分析，统一的管理，所以我们提出一个更新的概念叫逻辑数据湖，对于用户来说，不需要把他的数据进行物理上的搬迁，但是我们一样能够进行联邦计算和查询，这就是我们讲的逻辑数据湖的核心理念。

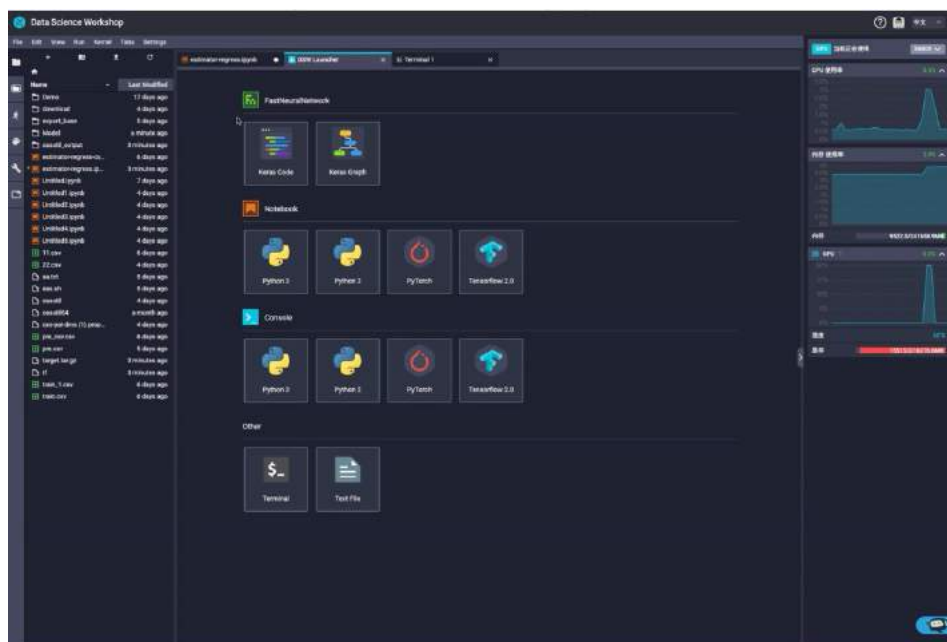
为了支撑这件事情，我们会有统一的元数据管理系统和调度系统，能够让不同的计算引擎协同起来工作，最后把所有的工作汇聚到全域数据治理上面，并且提供给数据开发者一个编程平台，让他能够直接的产生数据，或者是去定制自己的应用。那么，通过这样的方式，我们把原来的单维度大数据平台去做大数据处理，拓展到一个全域的数据治理，这个数据其实可以包含简单的大数据的，也可以包含数据库的，甚至是一些 OSS 的 file，这些我们在整个的平台里面都会加以处理。



如图为飞天大数据的产品架构，下面是存储计算引擎，可以看到我们除了计算引擎自带的存储之外还有其它开放的 OSS，还有 IOT 端采集的数据和数据库的数据，所有数据进行全域数据集成，集成后进行统一的元数据管理，统一的混合任务调度，再往上是开发层和数据综合治理层，通过这种方式，我们立体化的把整个大数据圈起来管理。

## 大数据与 AI 双生系统

提到了大数据我们肯定会想到 AI，AI 和大数据是双生的，对于 AI 来说它是需要大数据来 empower 的，也就说 bigdata for AI。下面可以通过一个 demo 来看我们怎么来做这件事情。对于 AI 的开发工程师来说，他们比较常用的方式是用交互式的 notebook 来进行 AI 的开发，因为它比较直观，但是如何把大数据也进行交互式开发，并且和 AI 来绑定，下面来看一下这个简单的例子。



如图是我们 DSW 的平台，我们可以直接的用一个 magic 命令，connect 到现存的 maxcompute 集群，并且选择 project 后，可以直接输入 sql 语句，这些都是智能的。然后我们去执行，结果出来之后我们可以对 feature 进行相应的分析，包括可以去改变这些 feature 的横纵坐标做出不同的 charts，同时我们甚至可以把生成的结果直接 web 到 excel 方式进行编辑和处理，处理完之后我们再把数据拉回来，也可以切换到 GPU 或者 CPU 进行深度学习和训练，训练完了之后，我们会把整个的代码变成一个模型，我们会把这个模型导入到一个相应的地方之后提供一个 Web 服务，这个服务也就是我们的在线推理服务。整套流程做完之后，甚至我们可以再接数据应用，可以在托管的 WEB 上构建，这就是大数据平台给 AI 提供数据和算力。

## AI优化的大数据系统 - 让大数据自动驾驶

AI Embedded Bigdata System



可用的

(面对双十一的负载, 系统仍然可用)



高性能低成本的

(面对日益增长的数据和计算, 能够打破数据增长与成本增长的线性关系)



智能的

(DB+在大数据时代无法延续, 系统自主学习优化, 进一步降低用户上手和调优的代价)

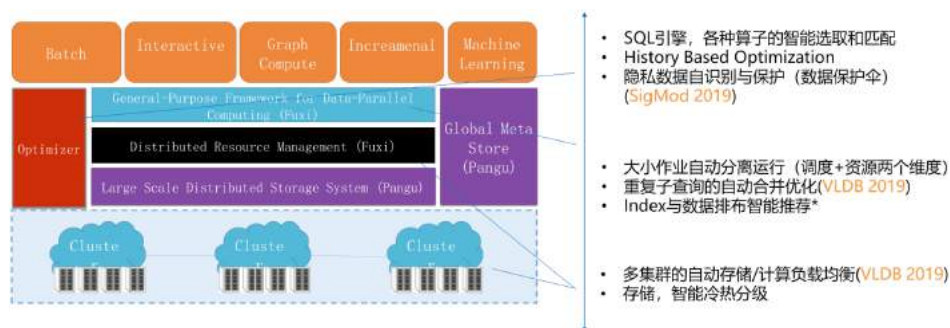
大数据和 AI 是双生系统, AI 是一个工具层, 可以优化所有的事情。我们希望飞天的大数据平台能够赋能给 AI。我们在最开始的时候希望 build 一个可用的系统, 能够面临双 11 的弹性负载仍然是可用的。通过这些年的努力, 我们追求极致的性能, 我们能够打破数据的增长和成本增长的线性关系, 我们也希望它是一个智能的, 我们希望更多的数据开发工程师来支持它, 我们需要更复杂的人力投入来理解他, 我们希望有更强的大数据来优化大数据系统。

## AI加持的大数据计算能力 Auto Data Warehouse



我们提出一个概念叫 Auto Data Warehouse，我们希望通过智能化的方式把大数据做得更聪明。整体上可以分成 3 个阶段：

- 第一阶段是计算层面和效率层面，我们尝试寻找计算的第一层原理，我们去找百万到千万级别里面的哪些作业是相似的，因此可以合并，通过这种方式来节省成本，还有当你有千万级别的表之后，究竟哪些表建索引全局是最优的，以及我们怎么去做冷热的数据分层和做自适应编码。
- 第二阶段是资源规划，AI 和 Auto Data Warehouse 可以帮助我们做更好的资源优化，包括我们有 3 种的执行作业模式，哪一种模式更好，是可以通过学习的方式学出来的，还有包括作业的运行预测和自动预报警，这套系统保证了大家看得到或者看不到的阿里关键作业的核心，比如每过一段时间大家会刷一下芝麻信用分，每天早上九点阿里的商户系统会和下游系统做结算，和央行做结算，这些基线是由千百个作业组成的一条线，从每天早上凌晨开始运行到早上八点跑完，系统因为各种各样的原因会出现各种的状况，可能个别的机器会宕机。我们做了一个自动预测系统，去预测这个系统是否能够在关键时间点上完成，如果不能够完成，会把更多的资源加进来，保证关键作业的完成。这些系统保证了我们大家日常看不见的关键数据的流转，以及双十一等重要的资源弹性。
- 第三阶段是智能建模，当数据进来之后和里面已有的数据究竟有多少的重叠？这些数据有多少的关联？当数据是几百张表时，搞 DBA 手工的方式可以调优的，现在阿里内部的系统有超过千万级别的表，我们有非常好的开发人员理解表里面完全的逻辑关系。这些自动调优和自动建模能够帮助我们在这些方面做一些辅助性的工作。



这是 Auto Data Warehouse 系统架构图, 从多集群的负载均衡到自动冷存, 到中间的隐形作业优化, 再到上层的隐私数据自动识别, 这是我和蚂蚁一起开发的技术, 当隐私的数据自动显示到屏幕上来, 系统会自动检测并打码。我们其中的三项技术, 包括自动隐私保护, 包括重复子查询自动合并优化, 包括多集群的自动容灾, 我们有 3 篇 paper 发表, 大家有兴趣的话可以去网站上读一下相关的论文。

# 计算存储引擎

## 飞天大数据平台计算引擎 MaxCompute 最新特性



曲宁

阿里云智能计算平台  
事业部产品专家

摘要：距离上一次 MaxCompute 新功能的线上发布已经过去了大约一个季度的时间，而在这一段时间里，MaxCompute 不断地在增加新的功能和特性，比如参数化视图、UDF 支持动态参数、支持分区裁剪、生成建表 DDL 语句功能等功能都已经得到了广大开发者的广泛使用。那么，近期 MaxCompute 究竟还有哪些新特性呢？本文就为大家揭晓答案。

### MaxCompute 与阿里云大数据产品解决方案

在介绍 MaxCompute 新功能前，我们先快速对阿里云的大数据产品解决方案进行介绍，以便不熟悉 MaxCompute 的朋友能快速建立认知。

阿里云大数据解决方案中包含了数据接入、数据存储及处理分析、数据服务以及在线应用等这样的几个维度。通常的情况下，基于 MaxCompute 和阿里云大数据解决方案搭建的系统会通过 DataWorks 实现离线多源异构数据的同步，并向 MaxCompute 大数据平台加载数据。与此同时，借助于 DTS 日志服务、Kafka 消息队列服务实现对实时数据的收集。之后，通过流式计算服务实现对于数据的实时计算和分析，并将数据投递到实时在线的服务或者回流到统一的数据仓库服务中去。数

据落盘保留下来之后，将进行数据仓库相关的处理分析，加工成为可以被业务消费、高质量的数据集。同时，利用机器学习平台可以开展包含数据准备、模型训练、模型部署在线推理在内的完整智能应用。

在数据服务 (data serving) 维度，阿里云大数据产品解决方案中也提供了多种的服务，包括了关系型数据库、分析型数据库、ES 等，这些服务能够帮助用户加速在面向在线应用场景下的数据消费。同时，阿里云大数据产品解决方案还能够与阿里云线上的 Quick BI、DataV 以及第三方客户自行购买的 BI 等工具进行结合。在云上大数据场景下，DataWorks 则承担的是整体的数据开发、编排调度以及数据管理的职能。

## What's New? MaxCompute 产品近期发布预览

本次分享面对的主要群体是对于阿里云 MaxCompute 产品有所了解并且有一定使用经验的客户，因此所介绍的内容会比较细致，但不会过多展开相关背景及原理介绍，更多地会面向 MaxCompute 已有的问题以及新推出的特性本身进行分享。

### What's New? MaxCompute 产品近期发布概览

已发布	新功能	大数据计算服务 - MaxCompute 国际Region Spark商业化发布	0 / 0 / 0
		MaxCompute国际Region推出内建Spark服务，开发者可在统一数据和权限体系下使用全托管的Spark进行数据分析和处理。	
	新功能	大数据计算服务 - 大数据计算服务 MaxCompute 西南成都节点正式开服售票。	2 / 2 / 0
		大数据计算服务 MaxCompute 西南成都节点开服售票，对包中云资源规格进行升级，在西南成都节点部署。	
	新功能	大数据计算服务 - UDTF/UDAF支持动态参数	1 / 1 / 0
		UDAF 和 UDTF 支持动态参数，对Reserve函数的语法做了扩展，使UDAF的编写更加便捷。	
	新功能	大数据计算服务 - UDF支持分区裁剪	0 / 0 / 0
		帮助用户降低计算成本。	
	新功能	大数据计算服务 - 生成建表DDL语句功能	0 / 0 / 0
		方便用户快速复制表的schema。	
正在灰度发布	新功能	大数据计算服务 - MaxCompute-云监控	0 / 0 / 0
		可以在阿里云的云监控服务中配置告警规则，以监控MaxCompute的运维使用情况。	
定向内测	新功能	大数据计算服务 - MaxCompute预付费套餐(计算资源+存储资源)发布	2 / 2 / 0
		MaxCompute的预付费套餐将推出，购买套餐将包含计算资源和存储资源。	
	新功能	大数据计算服务 - MaxCompute 19年6月特性: SQL新功能介绍、数仓教程、搭建数据平台	0 / 0 / 0
		文档详细讲解了最新的产品功能使用、最佳实践、教程等，通过文档大数据开发工程师可以快速上手学习和使用。	
		新增日期提取EXTRACT(datepart from timestamp)、JSON处理JSON_TUPLE的内置函数	
		支持指定表的列数认证: DEAFULT VALUE	
		支持LIMIT OFFSET	
		支持对分区合并降低分区数量/一次删除多个分区	
		元数据开放服务info_schema	
		实时审计日志服务	

近期以来，MaxCompute 大约每三个月就会迭代一个大版本发布到线上，而中间则会有很多个小版本。到 8 月份的时候，已经距离上次 MaxCompute 线上发布会经过了大约一个季度的时间，因此需要再做一次新特性的发布。所以本次分享不仅涵盖了 MaxCompute 针对日常需求的功能发布，也包括了大版本发布的内容。

本次所要介绍的 MaxCompute 产品近期的发布情况主要包括三个部分，首先是近几个月已经陆续发布上线，并且产品文档已经完备的功能，希望希望通过本次介绍让开发者能够更好地了解这些新的功能；其次是目前 MaxCompute 在线上所正在做的大版本升级中已经实现的一批灰度升级项目，本次也会对于其中一些比较成熟的功能进行分享；最后就是一些即将面向更大规模的用户进行发布的功能，也就是目前还处于定向内测阶段的功能。

## 新 Region 开服：西南成都节点正式开服、国际 Region 提供 Spark 服务

随着阿里云西南成都节点的正式开服，大数据计算服务 MaxCompute 也正式在西南成都节点开服售卖。与此同时，MaxCompute 也提供了很多国际的 Region，阿里云根据用户需求的强烈程度优先在香港、德国、新加坡、印度和美西这五个国际 Region 推出了 Spark 服务。

### 新Region开服：西南成都节点正式开服、国际region提供Spark服务



- 新region：大数据计算服务 MaxCompute 西南成都节点开服售卖。可在中文站购买成都节点资源，在控制台按需求开通project
- 功能国际化推广：MaxCompute国际Region香港、德国、新加坡、印度、美西推出内建Spark服务

#### 发布功能

MaxCompute国际Region香港、德国、新加坡、印度、美西推出内建Spark服务。

支持原生多版本Spark作业，社区原生Spark运行在MaxCompute里面，完全兼容Spark的API，支持多Spark版本同时运行。提供原生的Spark WebUI供用户查看。统一的计算资源，MaxCompute Spark像MaxCompute SQL/MR等任务类型一样，运行在MaxCompute项目开通的统一计算资源中。统一的数据和权限管理。完全遵循MaxCompute项目的权限体系，在访问用户权限范围内安全地查询数据。与开源系统相同的使用体验。MaxCompute Spark会和开源保持相同的体验和用户使用习惯，比如开源应用的UI，在线交互等。开源UI对于开源应用的调试至关重要，MaxCompute Spark提供原生的开源实时UI，并且能查询历史日志。某些开源应用提供交互式，也就是把后台引擎拉起后，进行实时交互。

## 新功能：SQL-参数化视图

MaxCompute 近期发布上线的版本围绕着 SQL 核心功能的一些细节做了大量的优化和提升，其中一点就是参数化视图。MaxCompute 传统的视图 (VIEW) 中实现了一定的封装与重用，但是并不接受调用者传递的任何参数，例如：调用者无法对视图读取的底层表进行数据过滤或传递其它参数，导致代码重用能力低下。MaxCompute 近期发布上线的版本的 SQL 引擎支持带参数的视图，支持传入任意表或者其它变量来定制视图的行为，从而增强了视图的可用性和复用度。

### 新功能：SQL-参数化视图



#### 什么是参数化视图？

MaxCompute 传统的视图 (VIEW) 中，实现了一定的封装与重用，但是并不接受调用者传递的任何参数，例如：调用者无法对视图读取的底层表进行数据过滤或传递其它参数，导致代码重用能力低下。MaxCompute 当前的 SQL 引擎支持带参数的视图，支持传入任意表或者其它变量，定制视图的行为。

#### 创建参数化视图

```
-- view with parameters
-- param @a -> a table parameter
-- param @b -> a string parameter
-- returns a table with schema (key string,value string)

create view if not exists pvi(@a table (k string,v bigint), @b string)
as
select srcp.key,srcp.value
from srcp join @a on srcp.key=a.k and srcp.p=@b;
```

#### 增强功能

##### 支持用 subquery 作为参数的值

```
SELECT * FROM param_view((SELECT name, id, age from students));
```

##### 定义 view 时，支持 ANY 类型

```
CREATE VIEW paramed_view (@a TABLE(name STRING, id ANY, age BIGINT))
AS
SELECT * FROM @a WHERE name = 'foo' and age < 25;
```

##### 定义 view 时，TABLE 定义支持代表可变长列表

```
CREATE VIEW paramed_view (@a TABLE(key STRING, * ANY), @b TABLE(key STRING, * STRING))
AS
SELECT a.* FROM @a JOIN @b ON a.key = b.key;

-- 请参见，
SELECT name, address
FROM param_view(1);
SELECT school, name, age, address
FROM student(), school()
WHERE age < 20;
```

## 新功能：SQL-UDTF/UDAF 支持动态参数

新发布的 MaxCompute 版本的 SQL 能够支持 UDF 相关的动态参数。如下图所示的代码所示，其中含有一个命名为 JsonTuple 的 UDTF。这里 JsonTuple 的业务需求就是首先读取一个 JSON 串，其中包含了一系列 JSON 内容，并且需要解析其中某些节点的信息。

## 新功能: SQL-UDTF/UDAF支持动态参数



• MaxCompute对Resolve注解的语法做了如下扩展

- 可以在参数列表中使用\*, 表示接受任意长度、任意类型的输入参数。

例如@Resolve[double,\*>String]表示接受第一个是DOUBLE类型, 后接任意长度、任意类型的参数列表。

- 可以在参数列表中使用any关键字, 表示任意类型的参数。

例如@Resolve[double,any>string]表示接受第一个是DOUBLE类型, 后接任意类型的参数列表。

- UDTF的返回值可以使用星号, 表示返回任意个STRING类型。返回值的个数与alias个数有关。

例如@Resolve[\*ANY,ANY>DOUBLE,\*], 调用方式是UDTF(x,y) as [a,b,c], 此处as后面设置了三个alias, 即a、b、c。

```
import com.aliyun.odps.udf.UdfException;
import com.aliyun.odps.udf.UDTF;
import com.aliyun.odps.udf.annotation.Resolve;
import org.json.JSONException;
import org.json.JSONObject;

@Resolve("STRING,*>STRING,*")
public class JsonTuple extends UDTF {
    private Object[] result = null;
    @Override
    public void process(Object[] input) throws UdfException {
        if (result == null) {
            result = new Object[input.length];
        }
        try {
            JSONObject obj = new JSONObject((String)input[0]);
            for (int i = 1; i < input.length; i++) {
                // 返回要求变长部分都是STRING。
                result[i] = String.valueOf(obj.get((String)(input[i])));
            }
            result[0] = null;
        } catch (JSONException ex) {
            for (int i = 1; i < result.length; i++) {
                result[i] = null;
            }
            result[0] = ex.getMessage();
        }
        forward(result);
    }
}
```

面对像 JsonTuple 这样的函数设计, 虽然给定了一个 JSON, 但是可能需要根节点的参数, 也可能需要根节点 + 子节点或者多个子节点的参数去提取并解析 JSON 字符串中的信息, 此时就要求自定义函数可以支持动态数量参数, 因此函数最好能够支持用户动态的参数输入, 也就是可以根据用户的动态参数输入提取相应的信息。MaxCompute 的 UDTF 和 UDAF 在参数列表中支持使用 \* 的模式, 表示接受任意长度、任意类型的输入参数, 从而满足了上述场景的需求。

## 新功能: SQL-UDT 支持资源 (Resource) 访问

User Defined Type, 简称 UDT, 它属于最近几个版本 MaxCompute 的 SQL 里面推出的比较新的功能, 目前应用也不是非常广泛。UDT 是 MaxCompute 2.0 之后主张使用的 NewSQL 能力, 其允许在 SQL 中直接引用第三方语言的类或者对象, 获取其数据内容或者调用其方法。目前, MaxCompute 的 SQL 中的 UDT 提供的是对于 Java UDT 的支持。举例而言, 如果想要在 MaxCompute 中获取整数中的最大值, 通常需要自己写一个 UDF 函数并通过其他语言实现出来, 而其实在像 Java 这样的高级语言中的一些内置方法就能够实现, 因此可以简单地使用 Select Java 库中的一个方法来获得相应的能力。这是之前所发布的 UDT 本身的能力, 而今天要发

布的新特性则是 UDT 对于 MaxCompute 中资源访问的支持。MaxCompute 中有很多对象，其中一种就是资源，比如一些数据文件。MaxCompute 可以使用 SQL 通过 Resource 读取文本文件，也可以通过 Set 参数的方式将自定义的 JAR 作为 UDT 能够直接访问的一个库，并在 UDT 当中使用用户自定义的第三方库。

## 新功能: SQL-UDT 支持资源 (Resource) 访问



User Defined Type, 简称UDT。UDT允许在SQL中直接引用第三方语言的类或者对象，获取其数据内容或者调用其方法。

```
SELECT java.lang.Integer.MAX_VALUE;
```

- 通过在UDT中获取ExecutionContext对象来访问Resource  
MaxCompute UDT支持获取 ExecutionContext 对象，可以在SQL中通过 `com.aliyun.odps.udt.impl.UDTExecutionContext.get()` 静态方法获取 ExecutionContext 对象，从而访问当前的 ExecutionContext，进而访问资源（例如文件资源和表格资源）。

### UDT访问Resource示例

```
set odps.sql.session.resources=1.txt;

select new String(com.aliyun.odps.udt.UDTExecutionContext.get().readResourceFile('1.txt')) text;
```

\* odps.sql.session.java.imports 指定默认的java package，可以指定多个，用逗号隔开。和java的import语句类似。可以提供完整类路径，如 `java.math.BigInteger`，也可以使用 `*`，暂不支持 `static import`。

如功能简介中的UDT jar包。用UDT来使用还可以如下写法：

```
1. set odps.sql.type.system.odps2=true;
2. set odps.sql.session.resources=odps-test.jar;
3. set odps.sql.session.java.imports=com.aliyun.odps.test.*; -- 指定默认的package
4. select new IntegerMaxValue().evaluate();
```

## 新功能: SQL-UDF 支持分区裁剪

MaxCompute 本身有大量的分区表，在分区的使用中一个很重要的优化点就是通过分区裁剪的方式在查询中过滤分区。在过滤分区的时候，有时需要使用一些自定义的逻辑，通过一些自定义的功能来确定分区的范围。当增加一个 UDF 时往往会发生全表扫描，这是因为在提交查询的时候还不能确定提交的分区到底是哪个，因此会发生全表扫描，进而带来比较高的费用和更重的负载。

## 新功能:SQL-UDF支持分区裁剪



### 分区剪裁合理性评估

您可以通过explain select语句查看分区裁剪是否生效。普通的UDF或join的分区条件写法都有可能致分区裁剪不生效，详情请参见[分区剪裁合理性评估](#)。

### UDF支持分区剪裁的实现原理

支持的方式是将UDF语句先当做一个小作业执行，再将执行的结果替换到原来UDF出现的位置。

### UDF支持分区剪裁的两种方式

➤ 在编写UDF的时候，UDF类上加入Annotation。

```
@com.aliyun.odps.udf.annotation.UdfProperty(isDeterministic=true)
```

➤ 在SQL语句前设置Flag: `set odps.sql.udf.ppr.deterministic = true;`

注意：该操作执行的原理是做执行结果回填，但是结果回填存在限制，即最多回填1000个Partition。因此，如果UDF类加入Annotation，则可能会导致出现超过1000个回填结果的报错。此时您如果需要忽视此错误，可以通过设置Flag: `set odps.sql.udf.ppr.to.subquery = false;`全局关闭此功能。关闭后，UDF分区裁剪也会失效。

新发布的 MaxCompute 版本中的 SQL 已经能够支持 UDF 分区裁剪的能力了。其逻辑是当提交的 SQL 存在 UDF 的时候，会首先将 UDF 单独提取出来执行，将 UDF 的执行结果拿到之后再和原来的查询语句进行拼装再提交。这样一来，当存在动态不确定的信息时，首先会解析动态信息，再确定分区范围，从而进行分区裁剪，尽量减低数据扫描所产生的费用。其开发方式就是在 UDF 中加一些参数，就能够使得 UDF 支持分区裁剪。

## 新功能: SQL-生成建表 DDL 语句功能

很多开发者都有将项目中数据表的 DDL 导出出来的需求，新版本的 MaxCompute 也提供了兼容 Hive 的 SHOW Create table + 的方式。从下图中的案例也可以看出，使用 SHOW Create table + 之后就能够获得该表的 DDL 的脚本，通过这种方式就能够实现重建表或者做表结构备份的工作。

## 新功能:SQL-生成建表DDL语句功能



- 查看建表语句的语法

```
SHOW CREATE TABLE <table_name>;
```

- 示例

```
27
28 SHOW create table public_data.ods_enterprise_share_basic;
29
```

```
OK
2019-07-16 17:15:10 start to get jobId:
2019-07-16 17:15:10 get jobId:20190716091510637gnetfv21
ID = 20190716091510637gnetfv21
CREATE TABLE IF NOT EXISTS public_data.ods_enterprise_share_basic(code STRING,name STRING,industry STRING,area STRING,pe STRING,outstanding STRING,t
otals STRING,totalsassets STRING,liquidsassets STRING,fixedassets STRING,reserved STRING,reservedpershare STRING,eps STRING,bvps STRING,pb STRING,time
consumed STRING,condp STRING,percondp STRING,rev STRING,profit STRING,gpr STRING,npr STRING,holders_num STRING) PARTITIONED BY (ds STRING) STORED AS C
FILE TBLPROPERTIES ('comment'='股票基本信息');
OK
```

## 新功能: SQL-支持指定表的列默认值

新版本的 MaxCompute 中也支持了指定表的列默认值。其实熟悉大数据发展历程的同学们都应该有所了解，数据库中存在很多比较完善技术，比如在校验机制、约束条件上都提供了各种默认值可供开发者选择。而大数据技术在这些方面所做的远远不够，目前的发展趋势是在数据的质量和数据的校验等方面工作中，大数据技术越来越靠近数据库技术。在 MaxCompute 最新的版本发布中提供的 DEFAULT VALUE 就允许了用户创建数据表的时候去指定默认值。

## 新功能:SQL-支持指定表的列默认值



- 功能说明

**DEFAULT VALUE** 允许用户创建table的时候指定默认值，**insert**的时候如果不指定该列，则插入默认值。

- 示例

```
1. CREATE TABLE t (a bigint default 10, b bigint);
2.
3. INSERT INTO TABLE t(b) SELECT 100;
4. --等同于
5. INSERT INTO TABLE t(a, b) SELECT 10, 100;
```

a	b
10	100

在上图的代码示例中，创建了一个表 T，表 T 中包含了 a、b 两列，这两列的数据类型都是 bigint，并且 a 的默认值设为 10。当向这个表中插入数据记录并且只对于字段 b 进行插数据的时候，MaxCompute 会自动地在结果当中将 a 的值按照默认值补齐，其等效于同时在 a 列中插入 10 并且在 b 列中插入 100。这样的功能在很多场景下用处还是比较大的，比如某些字段默认设置为 0 或者为空的情况。

## 新功能：SQL—支持 LIMIT OFFSET

之前的时候，很多用户也有这样的一个呼声，就是希望在 MaxCompute 的 SQL 上面做到像 MySQL、PG 这些数据库一样的分页功能，能够定位到在多少行之后取多少行数据，这样的功能在数据库中是非常通用的。以前，在 MaxCompute 的 SQL 里面通过为每一行数据打一个唯一标示 Row Number，再用 Between And 这样的方式来变相地实现分页功能。而在新版本的 MaxCompute 中，SQL 本身就已经原生地支持了 LIMIT OFFSET 的语法，可以更加容易地实现分页的需求。

### 新功能: SQL-支持LIMIT OFFSET



#### • 功能说明

**OFFSET** 语句 和 **ORDER BY LIMIT**语句配合，可以指定跳过**OFFSET**数目的行。如下面的语句将 **src** 按照 **key** 从小到大排序后，输出第**11**到第**20**行（**OFFSET 10** 指定跳过前**10**行，**LIMIT 10** 指定最多输出**10**行）

```
SELECT * FROM src ORDER BY key LIMIT 10 OFFSET 10;
```

#### • 可以解决的问题

1. 借助row\_number()函数作为递增唯一标识进行过滤查询

```
select * from (select row_number() over() as row_id, * from orders_delta) where row_id between 10 and 20;
```

row_id	key	value	other_fields
11	10000000000000000000	10000000000000000000	...
12	10000000000000000000	10000000000000000000	...
13	10000000000000000000	10000000000000000000	...
14	10000000000000000000	10000000000000000000	...
15	10000000000000000000	10000000000000000000	...
16	10000000000000000000	10000000000000000000	...
17	10000000000000000000	10000000000000000000	...
18	10000000000000000000	10000000000000000000	...
19	10000000000000000000	10000000000000000000	...
20	10000000000000000000	10000000000000000000	...

通过row\_number()对数据进行排序及唯一标识编号，然后根据该标识选取每次查询的分页范围。

## 新功能：SQL—新增内置函数 JSON\_TUPLE、EXTRACT

接下来介绍 MaxCompute 的 SQL 中两个比较实用的函数——JSON\_TUPLE

和 EXTRACT。MaxCompute 本身在之前对于字符串的处理只有一个 getObject() 内建函数，而只有这样的一个函数是比较痛苦的，因为在提取 JSON 中多个节点值的时候往往需要多次调用这个内建函数来获取某个节点的值。

## 新功能:SQL-新增内置函数JSON\_TUPLE、EXTRACT



### 查看函数定义

#### JSON\_TUPLE

函数说明：

```
1. STRING, TUPLE(STRING json, STRING key[, STRING key2,...])
```

函数说明如下：

返回数据中一个节点的JSON字符串中，按照输入的一维键（key1, key2,...）返回各个键对应的字符串。

参数说明：

- json：STRING类型，即原始的json格式字符串。
- key：STRING类型，用于提取json中的json，一次只能输入一个，不能以\$开头。

函数例：返回值为STRING类型。

### 示例

```
1. table school;
2.
3. -- 创建表
4. create table school (
5.   id int,
6.   name string,
7.   class string,
8.   teacher string,
9.   student string,
10.  primary key(id),
11.  partitioned by (year, month, day, hour, minute)
12. )
13.
14. -- 插入数据
15. insert into school values (1, '张三', '高一', '李四', '王五', 2019, 10, 10, 10, 10);
```

```
1. SELECT
2.   json_tuple(school, 'class', 'teacher') AS
3.   (class, teacher)
4. FROM school;
5.
6. -- 输出结果
7. class teacher
8. -----
9. 高一 李四
```

```
1. SELECT
2.   json_tuple(school, 'class', 'teacher') AS
3.   (class, teacher)
4. FROM school;
5.
6. -- 输出结果
7. class teacher
8. -----
9. 高一 李四
```

### 查看函数定义

#### EXTRACT

函数说明：

```
1. DAT EXTRACT(datetime, time_column)
```

函数说明如下：

返回日期的一维。

参数说明：

datetime：DATE, TIME, MONTH, DAY, HOUR, MINUTE, 返回日期、时间、月份、小时、分钟。

time\_column：返回日期。

### 示例

```
1. SET odds.sql.type.system odds2=true;
2. SELECT extract(YEAR FROM '2019-10-10 11:11:11') year;
3.
4. -- 输出结果
5. year
6. ----
7. 2019
```

```
1. SELECT extract(MONTH FROM '2019-10-10 11:11:11') month;
2.
3. -- 输出结果
4. month
5. ----
6. 10
```

而新增的 JSON\_TUPLE 函数则解决了这些问题。使用 JSON\_TUPLE 首先可以获得一个 JSON 字符串，该函数的参数则可以是多个动态的参数，比如想要获取 10 个节点的值就可以带 10 个参数，而且还适合使用嵌套式数据，因此 JSON\_TUPLE 将来也会成为开发者使用较多的内建函数。此外，新版本 MaxCompute 的 SQL 中还支持了 EXTRACT 函数，能够按年、按月、按日、按小时、按分钟来截取日期。因为对于日期的处理也属于高频操作，因此 EXTRACT 函数也将会被高频地使用。

## 新功能 - 分区管理

除了前面提到的 MaxCompute 在 SQL 中的一些增强功能之外，新版本的 MaxCompute 还提供了一些其他的能力，比如分区管理的能力。在分区管理部分，一个比较重要的场景就是由于开发者往往喜欢使用较多的分区，而当数据量大的时候

可能会造成多级嵌套分区，因此表的分区数量就会非常庞大。而 MaxCompute 本身对于分区数量存在一个上限，当数据分区过多的时候就需要将历史过往的分区进行合并，并且删除不必要的分区。

## 新功能-分区管理



### 分区合并

MaxCompute 有分区数量上限6W的限制，当分区数量过多时，可使用合并分区功能，对数仓数据进行归档，降低分区数量。合并分区功能会将同一个表下多个分区数据快速合并成一个分区，并删除之前分区，把数据移动到指定的分区下。语法格式如下，

```
1. ALTER TABLE <tableName> MERGE [IF EXISTS] PARTITION(<predicate>) [, PARTITION(<predicate2>) ...] 0
   VERWRITE PARTITION(<fullPartitionSpec>) [PURGE];
```

具体使用说明请参考文档[DML语句-分区/列操作](#)

### Drop partitions

支持一次性增加或者删除多个分区，语法格式如下：

```
1. ALTER TABLE t ADD [IF NOT EXISTS] PARTITION (p = '1') PARTITION (p = '2');
2. ALTER TABLE t DROP [IF EXISTS] PARTITION (p = '1'), PARTITION (p = '2')
3. --注意ADD多个分区之间没有逗号，DROP的多个分区间有逗号。
```

分区管理的最佳实践就是在数据比较“热”的情况下，分区切分得可能会比较细，当数据沉淀时间比较长的时候，过往的历史数据被访问的频率就会比较低，因此可以通过分区归并的方式减少分区的数量。过去删除分区的时候也是比较痛苦的，需要每次通过 drop partition 来删除一个分区，而现在 MaxCompute 在 drop partition 命令里面也支持了一次删除多个分区的能力，可以通过一次操作删除多个分区。

## 新功能 – 预付费资源监控报警

预付费资源监控报警是 MaxCompute 近期发布的一个新功能。阿里云 Max-Compute 的很多用户使用的都是预付费的固定 CPU 大小的资源组，在这种场景下，用户往往会更加关注购买的资源是否过多，在业务高峰期能否将其充分地利用起来，是否需要扩容或者缩容等。与此同时，用户还需要对于一些关键任务进行监控，

比如在作业排队比较严重，产生积压的情况时，也希望能够通过监控报警系统获得排队积压信息，并且及时告警，从而更加利于人工及时干预，保证当核心业务出现问题时能够得到及时处理。

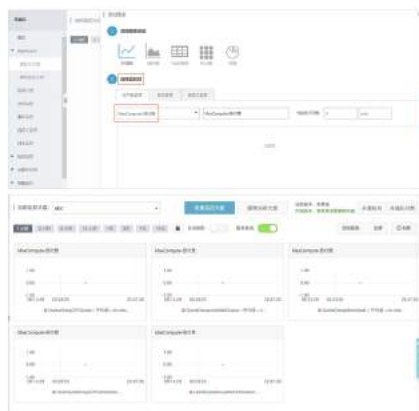
## 新功能-预付费资源监控报警



MaxCompute预付费资源的使用情况，已经接入阿里云的云监控服务，可以配置监控图表和自定义报警信息。

### • 监控指标

监控指标	说明
配额组CPU使用量	Quota组维度的指标，是指每分钟Quota组CPU的使用量快照。
配额组作业等待数	Quota组维度的指标，是指每分钟Quota组中在排队中的作业总数。
配额组内存使用量	Quota组维度的指标，是指每分钟Quota组内存的使用量快照。
配额组整体CPU使用率	用户在单个Region整体资源组维度的指标，是指每分钟Quota组CPU的使用占比。
配额组整体内存使用率	用户在单个Region整体资源组维度的指标，是指每分钟Quota组内存的使用占比。



MaxCompute 所提供的预付费资源监控报警能力基于阿里云所提供的云监控服务，并且结合了配额组 CPU 使用量、作业等待等待数、内存使用量、整体 CPU 使用率等一些关键指标，通过对以上这些指标进行规则配置，就能实现对于关键事件的短信通知以及实时告警，从而帮助用户更好地管理和使用资源。

## 新功能 -IP 白名单支持 IPV6

目前，IPV6 已经成为阿里云整体都会支持的能力，而 MaxCompute 本身也支持了 IPV6。对于 IP 白名单这个安全特性而言，过去 MaxCompute 支持 IPV4，目前也扩展支持了 IPV6，使用方式与原本的 IPV4 差异不大，通过白名单列表的设置就能够同时支持 IPV4 和 IPV6。

## 新功能-IP白名单支持IPv6



MaxCompute支持Project级别的IP白名单，目前已支持IPv6格式

### • IP白名单列表的三种形式：

#### ➢ 单纯IP

例如：101.132.236.134、FE80:0202:B3FF:FE1E:8329。

#### ➢ 子网掩码

例如：100.116.0.0/16、  
FE80:0101:4567:F456:0202:B3FF:1111:1111/126。

#### ➢ 网段

例如：101.132.236.134-101.132.236.144、  
FE80:0101:4567:F456:0202:B3FF:FE1E:8330-  
FE80:0101:4567:F456:0202:B3FF:FE1E:8331。

### • 使用命令行工具设置IP白名单的方法示例

```
setproject odps.security.ip.whitelist=FE80:0202:B3FF:FE1E:8329;
```

### • 关闭了IP白名单功能的方法

```
setproject odps.security.ip.whitelist=;
```

## 新规格：预付费套餐（计算资源 + 存储资源）发布

前面为大家介绍了 MaxCompute 在新版本的发布中的一些功能，接下来将为大家介绍 MaxCompute 的一种新规格。新版本的 MaxCompute 中推出了预付费套餐（计算资源 + 存储资源）的规格。对于 MaxCompute 而言，主要有两种付费形式，其中一种是按量付费，这种方式具有极致的弹性，并且能够自动伸缩；另外一种是按作业付费，这种方式比较符合期望的财务支出，在性价比方面更具有优势。

## 新规格：预付费套餐(计算资源+存储资源)发布



### • 收费规则

预付费套餐的售卖方式为混合付费，在开通时以包年包月方式购买计算资源（主要为CU）和存储资源，使用时，公网下载流量和超出的存储容量按使用量收费。

- **套餐费用：**开通时直接选择购买的套餐进行付费。
- **存储收费：**目前只收取表（Table）存储的容量费用，每日按购买的套餐有固定量的存储，超出的量另付费。
- **上传收费：**数据上传至MaxCompute不产生费用。
- **下载收费：**仅对通过公网下载进行按量收费。

规格类型	计算资源（CU）	存储资源	上传/下载资源
存储密集型160套餐	160	150TB，超出部分按量付费	无限制，按量付费
存储密集型320套餐	320	300TB，超出部分按量付费	无限制，按量付费
存储密集型600套餐	600	500TB，超出部分按量付费	无限制，按量付费

本次新发布的预付费打包套餐和之前的固定资源配额都属于固定规格预付费方式。预付费套餐的售卖方式为混合付费，在开通时以包年包月方式购买计算资源（主要为 CU）和存储资源。使用时，公网下载流量和超出的存储容量按使用量收费。总体而言，预付费套餐实际是将一定大小计算资源和存储资源打包销售的优惠套餐，相比于企业客户在线下自行搭建的大数据常用配置的计算和存储规模，阿里云本次所推出的预付费套餐将会提供一个十分具有竞争力的价格，这样一来既能够保证企业每月的财务支出比较稳定，同时也能够提供较高的性价比。

## 体验优化：文档持续优化及最佳实践指引

MaxCompute 团队一直以来都非常注重产品文档的质量和体验，也在进行着持续地优化，并在社区中也和广大的开发者保持着密切的交流和互动，获取了开发者在 MaxCompute 产品使用以及产品如何应用到生产的过程中的一些需求。MaxCompute 团队近期也将一些重点的、高频的问题沉淀下来，形成了文档化、案例形式的实践指导，发布到了官网上面。

## 体验优化：文档持续优化及最佳实践指引



针对使用过程中的常见问题，“精准定向”通过文档和案例进行实践指导，优化产品使用体验



这些文档包含了如何基于 MaxCompute 构建和优化数据仓库，同时也提供了一些数字化运营的具体案例，比如如何基于 MaxCompute 搭建互联网在线的运营分

析平台。此外，因为 Python 生态广受欢迎，开发者也是人数众多，因此 PyODPS 也受到了很多 Python 开发者的支持，通过前一个阶段的反馈，MaxCompute 团队也发现 PyODPS 在文档方面存在一些不足之处，因此近期也做了专项整理。MaxCompute 团队整理了 PyODPS 的一些指导文档和具体示例，大家可以到官网上去查看最新发布的指导文档和案例。

## 新功能内测：元数据服务 Information\_Schema

前面分享的是 MaxCompute 新版本中的一些线上可用的功能，而在实际上还有一些功能目前处于内部定向邀请测试阶段，这些功能也是社区中很多开发者以及企业客户所非常看重的功能。这里为大家介绍两个处于内测阶段的功能，其中第一个就是元数据服务。MaxCompute 本身是一个多租户的系统，用户使用的时候非常希望将更多的数据以更加友好的方式开放给使用者，能够帮助使用者自助地了解项目中的一些信息，其中比较常见的就是元数据信息，比如 Table、Partition、Column、UDF、Resource、User 等信息。以往这些信息都是通过命令的方式获取，但是当元数据量非常大的时候，很难通过这种方式去获得或者检索和过滤信息。而在传统的数据库技术领域有非常多的解决方案，因此 MaxCompute 中也提供了 Information\_Schema 这样的一套服务，在遵循规范的基础之上也添加了一些 MaxCompute 特有的视图和字段，既能够满足开发者原有的一些使用习惯，也能够更好地结合 MaxCompute 本身的一些特点。

## 新功能内测：元数据服务 Information\_Schema

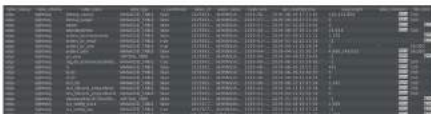


MaxCompute 的 Information Schema 提供了项目元数据及使用历史数据等信息。在 ANSI SQL-92 的 Information Schema 基础上，添加了面向 MaxCompute 服务特有的字段及视图。



系统表分类	View	时效性/保留策略
元数据系统表	TABLES	准实时视图
	PARTITIONS	准实时视图
	COLUMNS	准实时视图
	UDFS	准实时视图
	RESOURCES	准实时视图
	UDF_RESOURCES	准实时视图
	USERS	准实时视图
	ROLES	准实时视图
	USER_ROLES	准实时视图
	PACKAGE_RESOURCES	准实时视图
	INSTALLED_PACKAGES	准实时视图
	SCHEMA_PRIVILEGES	准实时视图
	TABLE_PRIVILEGES	准实时视图
	COLUMN_PRIVILEGES	准实时视图
	UDF_PRIVILEGES	准实时视图
	RESOURCE_PRIVILEGES	准实时视图
使用历史表	TABLE_LABELS	准实时视图
	COLUMN_LABELS	准实时视图
	TABLE_LABEL_GRANTS	准实时视图
	COLUMN_LABEL_GRANTS	准实时视图
	TUNNELS_History	准实时，保留 14 天

## 项目元数据查询



## 作业历史明细查询

```
select * from information_schema.tasks_history where ds='20190813'
and status='Terminated' and task_type='SQL' limit 20;
```



通过内建服务，MaxCompute 首先会为用户提供一个 Information\_Schema 库，用户可以去自助地去访问相关的元数据信息，比如项目中的表格、字段、视图、创建人、创建时间以及表的大小和生命周期等。通过这种方式就能够对于表的全局信息有一个很好的把控。与此同时，MaxCompute 也会提供准实时的作业历史明细的查询能力。作业历史包含了日常的实例信息、SQL 的作业明细等。以往是通过 show instance 等命令获取近期执行作业的信息，而如今可以通过对于视图的查询方式来获取本项目所有作业信息，并且能够通过一些过滤条件筛选出自己所需要的作业信息。所查询出来的信息则包括了项目类、项目名称、作业提交时间、作业状态、作业的 SQL 语句、数据的扫描量、复杂度以及资源消耗等信息，基于对于这些信息的统计就能够实现很多的性能优化以及诊断等相关工作。元数据服务 Information\_Schema 这样的能力也会在近期面向全体云上用户进行公测。

## 新功能内测：基于 ActionTrail 的行为审计日志服务

另一个要介绍的处于内测阶段的功能就是基于 ActionTrail 的行为审计日志服务。对于阿里云 MaxCompute 团队而言，经常会有企业提交工单希望能够帮助他们检查一下某张表的数据被谁删除了，或者某个账号近期是否进行了数据下载。因

此，MaxCompute 也将要推出基于 ActionTrail 的行为审计日志服务，这款服务能够完整地记录项目内的用户操作行为，并通过接入阿里云 Action Trail 服务将 MaxCompute 用户行为日志实时推送给客户，满足客户实时审计、问题回溯分析等需求。被审计的行为包含了对于表的创建、删除以及对于表结构的变更、写入数据、下载数据等，而日志信息也包含了企业比较关心的客户端信息、操作的具体 SQL 内容、提交人、提交时间等关键信息。

## 新功能内测：基于 ActionTrail 的行为审计日志服务



完整记录项目内的用户操作行为，并通过接入阿里云 Action Trail 服务将 MaxCompute 用户行为日志实时推送给客户，满足客户实时审计、问题回溯分析等需求。



MaxCompute

行为审计日志服务

完整记录项目内的用户操作行为，并通过接入阿里云 Action Trail 服务将 MaxCompute 用户行为日志实时推送给客户，满足客户实时审计、问题回溯分析等需求。

操作时间	操作人	操作内容	操作结果	操作详情
2019-08-01 10:00:00	root	createTable	成功	createTable (createTable)
2019-08-01 10:00:00	root	createTable	成功	createTable (createTable)
2019-08-01 10:00:00	root	createTable	成功	createTable (createTable)

```

1. createTable (createTable)
2. createTable (createTable)
3. createTable (createTable)
4. createTable (createTable)
5. createTable (createTable)
6. createTable (createTable)
7. createTable (createTable)
8. createTable (createTable)
9. createTable (createTable)
10. createTable (createTable)
11. createTable (createTable)
12. createTable (createTable)
13. createTable (createTable)
14. createTable (createTable)
15. createTable (createTable)
16. createTable (createTable)
17. createTable (createTable)
18. createTable (createTable)
19. createTable (createTable)
20. createTable (createTable)
21. createTable (createTable)
22. createTable (createTable)
23. createTable (createTable)
24. createTable (createTable)
25. createTable (createTable)
26. createTable (createTable)
27. createTable (createTable)
28. createTable (createTable)
29. createTable (createTable)
30. createTable (createTable)
31. createTable (createTable)
32. createTable (createTable)
33. createTable (createTable)
34. createTable (createTable)
35. createTable (createTable)
36. createTable (createTable)
37. createTable (createTable)
38. createTable (createTable)
39. createTable (createTable)
40. createTable (createTable)
41. createTable (createTable)
42. createTable (createTable)
43. createTable (createTable)
44. createTable (createTable)
45. createTable (createTable)
46. createTable (createTable)
47. createTable (createTable)
48. createTable (createTable)
49. createTable (createTable)
50. createTable (createTable)
51. createTable (createTable)
52. createTable (createTable)
53. createTable (createTable)
54. createTable (createTable)
55. createTable (createTable)
56. createTable (createTable)
57. createTable (createTable)
58. createTable (createTable)
59. createTable (createTable)
60. createTable (createTable)
61. createTable (createTable)
62. createTable (createTable)
63. createTable (createTable)
64. createTable (createTable)
65. createTable (createTable)
66. createTable (createTable)
67. createTable (createTable)
68. createTable (createTable)
69. createTable (createTable)
70. createTable (createTable)
71. createTable (createTable)
72. createTable (createTable)
73. createTable (createTable)
74. createTable (createTable)
75. createTable (createTable)
76. createTable (createTable)
77. createTable (createTable)
78. createTable (createTable)
79. createTable (createTable)
80. createTable (createTable)
81. createTable (createTable)
82. createTable (createTable)
83. createTable (createTable)
84. createTable (createTable)
85. createTable (createTable)
86. createTable (createTable)
87. createTable (createTable)
88. createTable (createTable)
89. createTable (createTable)
90. createTable (createTable)
91. createTable (createTable)
92. createTable (createTable)
93. createTable (createTable)
94. createTable (createTable)
95. createTable (createTable)
96. createTable (createTable)
97. createTable (createTable)
98. createTable (createTable)
99. createTable (createTable)
100. createTable (createTable)

```

# 飞天大数据平台实时计算 Flink on Kubernetes 最新特性



张荣

阿里云智能计算平台  
事业部产品专家

## Flink 产品介绍

目前实时计算的产品已经有两种模式，即共享模式和独享模式。这两种模式都是全托管方式，这种托管方式下用户不需要关心整个集群的运维。其次，共享模式和独享模式使用的都是 Blink 引擎。这两种模式为用户提供的主要功能也类似，

1. 都提供开发控制台；
2. 开发使用的都是 Blink SQL，其中独享模式由于进入了用户的 VPC，部署在用户的 ECS 上，因此可以使用很多底层的 API，如 UDX；
3. 都提供一套的开箱即用的 metric 收集、展示功能；
4. 都提供作业监控和报警功能。
5. 最后，在收费模式上，共享模式和独享模式用户所承担的都是硬件加软件（独享模式是软件（VPC））的费用。

## Flink on Kubernetes 模式介绍及对比

在共享和独享这两种模式的基础上，阿里云实时计算团队于 2019 年 9 月中旬会推出一个新的模式，Flink on K8S，其与前两种模式区别主要在于：

- 托管模式：集群以半托管模式部署在用户 ECS 和 K8S 上，用户对该集群用完全的掌控能力。

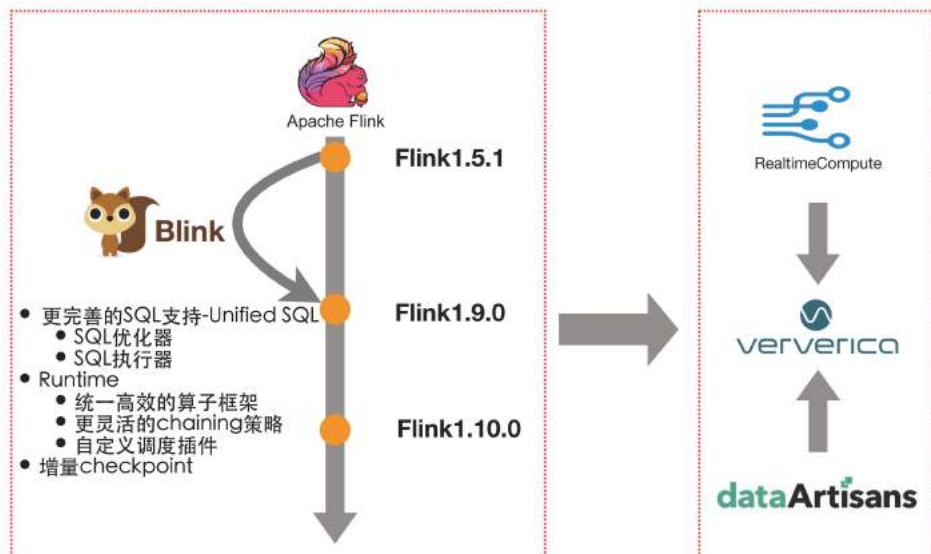
- 引擎版本：直接使用开源 Flink 版本
- 提供功能：提供开发控制台支持用户提交并控制作业；支持 Flink 1.6/1.7/1.8 等版本；也提供 metric 收集、展示、作业监控、报警功能；提供其他可插拔的增值功能。
- 收费模式：Flink on K8S 模式下，推广期间软件完全免费。

各种模式对比如下：

	共享模式	独享模式	Flink on K8S
托管模式	全托管	全托管	半托管
引擎版本	Blink	Blink	Flink
提供功能	1. 开发控制台 2. Blink SQL 3. metric收集/展示 4. 作业监控/报警	1. 开发控制台 2. Blink SQL+UDX 3. metric收集/展示 4. 作业监控/报警	1. 开发控制台 2. Flink1.6/1.7/1.8/1.9 3. metric收集/展示 4. 其他增值功能
收费模式	硬件+软件	硬件+软件（VPC）	软件（VPC）免费

引擎方面，Blink 是由阿里云实时计算团队于 2016 年从 Flink 1.5.1 拉出来的分支，在这之后的三年多的时间里，该分支被进行了一系列的改造：1) SQL 优化器和执行器的改造，目前有更完善的 SQL 支持，提供了 Unified SQL；2) 在 Runtime 上，提供了统一高效的算子框架、更加灵活的 chaining 策略和自定义调度插件；3) 提供增量 Checkpoint。

2019 年 1 月份，阿里巴巴决定将 Blink 的所有优化功能贡献给社区，经过六个多月的改造，Blink 的部分基本功能已经合并到 Flink 1.9.0 中，与此同时，阿里也收购了 Flink 创始团队成立的公司 dataArtisans。在此之后，两个团队将共同维护一个新的品牌 Ververica，该品牌推出新的界面平台 Ververica Platform 来为用户提供服务。



## Why Flink+Kubernetes

Flink 之所以选择 K8S 来作为底层的资源管理来为用户提供服务主要原因有以下几点:

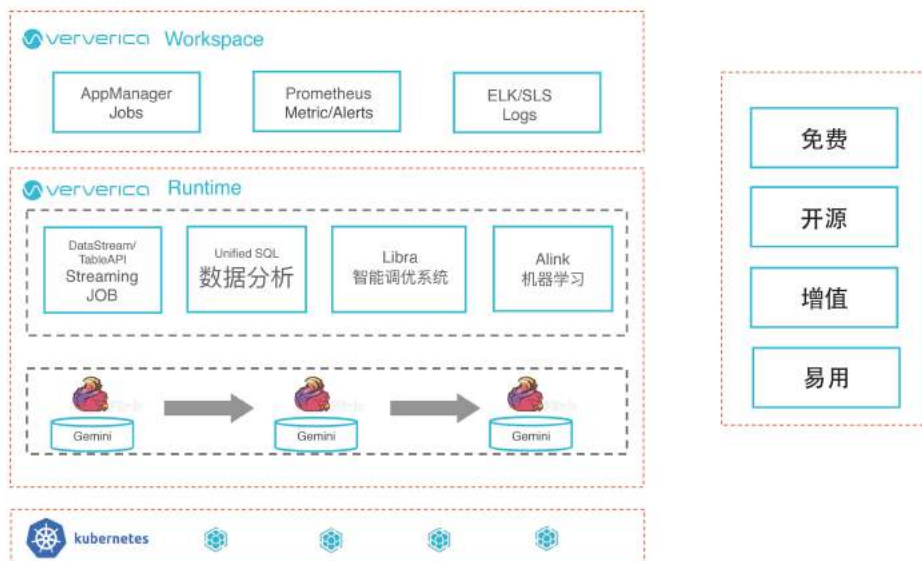
- Flink 特性: 首先 Flink 是大数据类应用, 与传统大数据应用如 Spark、Hadoop、MapReduce 以及 Hive 等不同的是, Flink 是常驻进程, 其类似于在线业务的 App, 作业发布后修改频率比较低, 这就要求执行作业的 worker 长时间稳定运行。另外, 与其他批处理作业相比, 流作业任务一般应用于实时风控和实时推荐的业务场景下, 其重要度更高, 稳定性要求也更高。
- K8S 优势: K8S 设计的初衷是为在线应用服务, 目标是为了帮助在线应用更好地发布和管理, 实现资源隔离; 其次, 目前 K8S 具备一定的生态优势, 目前很多用户已经开始或尝试开始使用 K8S 来管理在线应用; K8S 可以很好地集成其他集群维护工具, 如监控工具普罗米修斯, 同时在资源弹性方面, K8S 可以很方便地进行扩缩容。

Flink特性	K8S优势
<ol style="list-style-type: none"><li>1. 常驻进程</li><li>2. 任务重要度高<ol style="list-style-type: none"><li>1. 实时风控</li><li>2. 实时推荐</li></ol></li></ol>	<ol style="list-style-type: none"><li>1. 资源隔离</li><li>2. 生态优势</li><li>3. 资源弹性</li></ol>

## Ververica Platform 介绍

Ververica Platform 平台所包含主要功能模块如下图所示：

- K8S 集群：需要用户在阿里云上创建 ACK 集群（阿里云官网搜索 ACK 进入产品主页了解详情）。
- 可插拔组件：1) APP Manager。用户可通过 APP Manager 界面对作业进行提交和管理；2) 开箱即用的指标收集、展示及报警组件，该组件集成了 Prometheus 的功能。3) 日志收集、分析、展示组件
- 增值功能：在后续产品规划中，产品会提供一些列增值功能。首先是 Libra 智能调优系统。熟悉 Flink 的用户可能都知道，目前 Flink 的调优比较麻烦，尤其在开源 Flink 中，用户需要多次预估波峰流量来设置 Flink 作业的资源，否则在波峰的时候可能会出现作业延时。而所推出的 Libra 智能调优系统功能会根据波峰波谷的特点来自动调整 Flink 资源，在没有人为干预的情况下提高资源利用率，同时降低波峰时的作业延时。除此之外，平台还将提供 Alink 机器学习组件、Gemini 等增值功能。



总体而言，Ververica Platform 的特点是**免费、开源、增值和易用**。首先用户只需要支付 ECS 的费用便可以使用 Ververica Platform 平台的所有功能；其次 Flink Core 是开源的，用户无需担心其兼容性和因为被某个平台绑架而产生的问题，并且开源 Flink 的功能可以无缝迁移到该平台上；此外，Ververica Platform 将提供一系列增值功能，整个平台易用性较高。

下图是 Ververica Platform 的平台界面，通过该界面用户可以创建并提交一个 Flink 作业。用户可以设置 Flink 作业的名称、初始化状态、Flink 版本（目前支持 1.6/1.7/1.8）、Jar 包地址以及开源资源配置（如并发度、Job Manager 的 CPC 内存等），点击提交后可以很方便地在 K8S 上运行一个 Flink 作业。

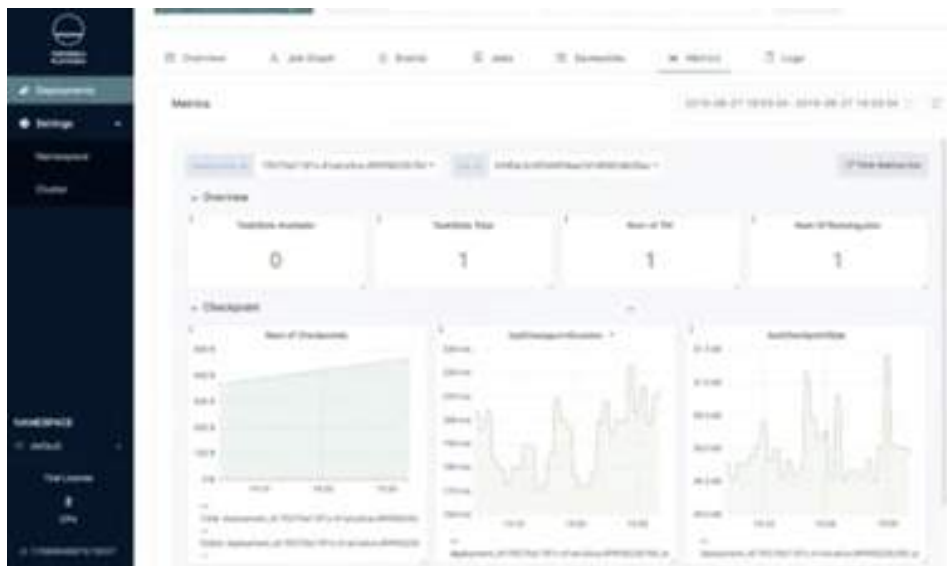


下图展示了一个已经在 K8S 上运行的作业，用户可以浏览整个作业的配置信息、拓扑图、在 K8S 上的 Events、Jobs、Savepoints 的状态。



用户还可以在界面上直观地查看作业默认的 metrics 和 logs 信息，从而简化作业问题排查的复杂度。此外，该平台还支持直接显示 Flink Web UI。目前的平台功

能比较简洁，后续会将平台自动调优、Alink 机器学习等方面的功能纳入进来。



以上是阿里云 Flink on Kubernetes 产品形态的新功能，欢迎大家试用体验。试用流程：登陆阿里云实时计算详情页，点击购买‘Flink 云原生’产品即可申请公测资格。

# 飞天大数据平台 E-MapReduce 4.0 最新特性



王晓平

阿里云智能计算平台  
事业部高级产品经理

本次的分享主要围绕以下三个方面：

## 一、EMR 产品概述

## 二、EMR 产品新特性

## 三、EMR Road Map

### 一、产品概述

#### 1. E-MapReduce (EMR) 是什么

E-MapReduce 是阿里云上云原生的开源大数据平台，一款利用开源大数据解决用户分析问题的大数据产品。在 2016 年 6 月，E-MapReduce 开始公测，2016 年 9 月开始商业化，目前为止 E-MapReduce 已经商业化三年多的时间。在 2019 年 8 月，阿里发布了 E-MapReduce (EMR) 敏捷版。在 2019 年 10 月，阿里巴巴将发布 EMR4.0 版本。EMR4.0 版本的 Hadoop 将会升级到 3.1.x。



EMR是阿里云上**云原生开源**大数据平台。

**Time line**

- 16 Jun EMR 上线公测
- 16 Sep EMR 正式商业化
- 17 Sep EMR 集成AI商业化和算证
- 19 Aug EMR 数据湖上线
- 19 Oct EMR 4.0 即将发布, Hadoop升级到3.1.x

**Product Features**

**100% 开源 组件**

- 产品所使用的大数据组件均为100%开源组件。
- 根据开源社区的进度版本升级, Bug Fix。
- 对Spark, Hadoop, Kafka等组件均有性能和易用性的增强。

**Cost optimization**

- 支持集群的弹性伸缩
- 支持OSS作为数据湖存储系统
- 支持EC2实例
- 支持通过作业启动临时集群

**Enterprise features**

- 采用MT Karbonis + Apache Ranger 用户鉴权和权限管理
- 提供EMR API监控集群和作业运行状况

## 2. EMR 产品特性:

EMR 产品特性主要展现在以下三个方面。

**100% 开源组件:** EMR 产品所使用的大数据组件为 100% 开源组件。对用户来说, 学习成本和迁移成本都相对较低。此外, EMR 会根据社区的进度和 Bug Fix 进行升级, 保证产品是根据社区持续升级, 实现较好的用户体验。还有, EMR 对 Spark、Hadoop、Kafka 等组件均有性能和应用性的增强。以 Spark 为例, EMR 比较早的集成了 Spark AE, 推测执行的自动化和特性, 同时包括基于 Spark 来进行物化视图的自研的产品功能。

**成本优势:** EMR 产品相对于用户自建 Hadoop 更具有成本优势。首先, EMR 集群支持弹性伸缩。此处的弹性伸缩指两方面。比如可以根据一个时间点, 大部分用户在用开源大数据 Hadoop 时在每天晚上 12 点定时跑一个较大的日报任务, 而日报任务可能并不需要全天都跑, 可能在早上就会跑完。此时使用弹性伸缩有利于节省计算资源。同时也可以采用按负载伸缩的方式, 设置 YARN Root 的队列的某一个指标进行伸缩。用户可以根据自己实际的 application 排队情况及 container 分配的情况进行伸缩。其次, EMR 支持 OSS 作为数据存储系统。除 HDFS 之外可以使

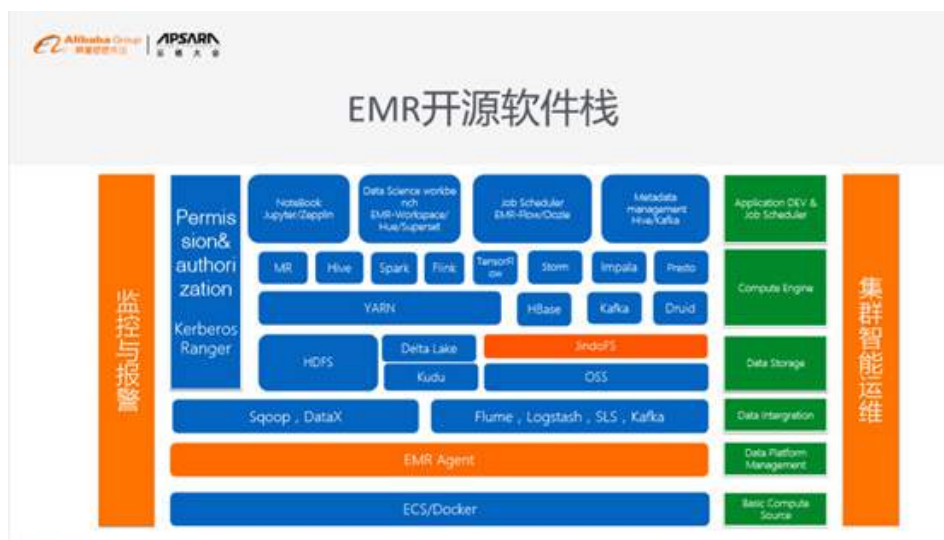
用 OSS，将 HDFS 的 schema 改为 OSS 路径直接读写 OSS。相对来说，可以实现用户的数据分层存储，当用户的热数据放在 HDFS 里，冷数据则可以放在 OSS 中。同时实现 OSS 的生命周期的管理。比如说相对温数据放在一个标准型里，但如果用户的数据是一年前或者几年前的数据，则可以放到归档型或者低频型的更便宜的 OSS 存储中。第三个特性是 EMR 在弹性伸缩的基础上同时支持竞价实例。ECS 提供了一种比较便宜的实例类型，即竞价实例。用户可以自己出一个价格来实现成本节省。考虑到数据可靠性的影响，竞价实例只能用在计算节点上，若上面有 HDFS 的数据存储，则不允许使用竞价实例或者弹性伸缩。最后，EMR 可以通过作业来拉起一个临时集群。当用户只是想要跑一个大数据的任务，可以通过集群模板设置集群的配置。当作业被触发时，EMR 会先创建好临时集群，作业执行完毕，临时集群会被释放掉。通过临时集群的方式实现比较灵活的集群创建和管理。

**企业级特性：**EMR 采用了 MIT Kerberos 和 Apache Ranger 实现用户鉴权和用户权限管理。同时提供了 EMR APM 实现集群和作业的监控。

### 3. EMR 开源软件栈

下图是 EMR 目前支持的开源软件栈。最底部是 EMR 的 Docker 和 ECS。通过 ECS 是目前在公有云上的主要形态，专有云可以通过 Docker 的方式，在 ECS 之上部署一个 EMR Agent，主要负责集群管理和集群运行数据的采集，检测集群端前是否出现问题。数据导入时 EMR 提供了 Sqoop，使用开源的 DataX 组件、Flume 组件或 SLS 将数据导入到 EMR 上。数据存储这一块 EMR 支持 HDFS 和 OSS。在 OSS 之上又开发了 jindoFS，jindoFS 可以用 OSS 作为数据湖的概念，通过 jindoFS catch 的方式做数据加速，提升 OSS 读取数据的效率。EMR 最新版本支持 Kudu 和 Delta Lake，帮助用户有效的构建实时数仓。用户利用 Kudu 或 Delta Lake+ 流式计算，如 Spark 或 Flink 实现实时用户画像、实时推荐等功能。资源调度方面，EMR 目前主要使用 YARN。计算引擎方面，EMR 提供了较多的计算引擎，如 Spark、Flink、TensorFlow、Storm、Impala 和 Presto 等。同时在应用层面提供了开源的 Jupyter、NoteBook、Zepplin 以及 Hue。EMR 是一个半托管

的形态，用户自建的系统也可以无缝迁移。阿里自研的 EMR-Flow 可实现工作流的调度。



## 4. EMR 集群特性

目前 EMR 支持五种集群类型。目前五种集群类型的购买是独立的，在购买集群的首页可以选择一个集群类型，如 Hadoop，Kafka，或者 ZooKeeper 集群。

**Hadoop 大数据平台：**一般用户都会购买一个 Hadoop 平台，然后再实现批计算、流式计算、数据仓库、Ad hoc 查询、NoSQL 和 HBase 等。

**Kafka：**若用户希望实现一个半托管的 Kafka，可以购买 EMR Kafka 系统实现消息系统。

**TensorFlow：**TensorFlow 在深度学习、机器学习方面非常流行。EMR 可以实现 TensorFlow On YARN 或 TensorFlow On Spark 执行。

**Druid：**EMR 对于实时 OLAP 有一个独立的集群类型 druid。

**ZooKeeper：**在集群功能较大的情况下，用户可以购买独立的 ZooKeeper 集

群实现分布式协调系统。如 Kafka、Druid、HDFS、YARN 或 HBase 等都需要 ZooKeeper。



## 5. EMR 半托管形态

如下图，EMR 目前是半托管形态。用户可以通过 ssh login 的方式登录 ECS 节点，在 ECS 节点上部署 EMR 的服务，如工作流调度、智能诊断、弹性伸缩、资源池等功能。EMR 有更低的迁移成本。如果用户是 Hadoop 基础体系，可拥有更低的迁移成本和学习成本。在用户人力资源不充沛的情况下可以通过 EMR 有效实现运维成本的节省。



## 6. EMR 集群节点类型

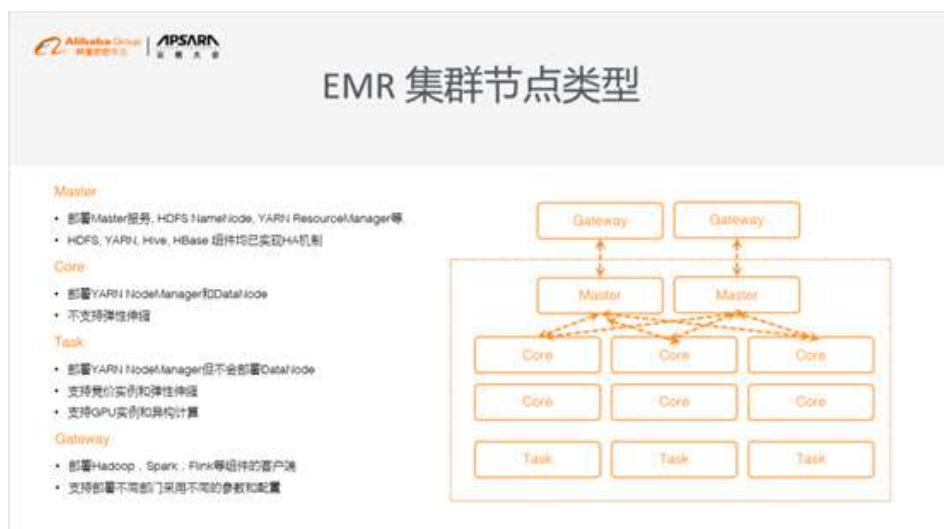
EMR 集群节点类型主要分以下四种。

**Master 节点:** Master 节点部署了 HDFS Name Node 和 YARN Resource Manager。此外，相应的 HDFS、YARN、Hive、HBase 等组件均已实现了 HA 机制。

**Core 节点:** Core 节点主要部署 YARN NodeManager 和 DataNode。Core 节点不支持弹性伸缩。

**Task 节点:** 用户可以使用 Task 进行弹性伸缩或者使用竞价实例的购买方式。如果用户有深度学习需求，可以通过 YARN 调度 GPU 资源，同时也可以 Task 节点选择购买 GPU 实例。

**Gateway 节点:** Gateway 节点部署 Hadoop、Spark、Flink 等组件的客户端，支持部署不同部门采用不同的参数和配置。因为用户可能会在不同客户端采用不同配置，从而实现用户隔离，用户可以购买 Gateway 节点实现资源配置上的隔离，同时减轻 Master 节点上的工作压力。



## 二、产品新特性

### 1. EMR 基础能力升级

**计算能力升级：**阿里云刚刚发布了第六代企业级实例，既 CPU 从 Skylake 升级到 Cascade Lake，睿频从 **2.7GHz 加速提升到 3.2GHz**。由下图可发现，相对于第五代企业级实例，第六代企业级实例 CPU 性能和内存读写性能都有显著提升。阿里云在 2019 年九月底上线第六代企业级实例，提供更高的计算性能和更低的产品价格，将技术的红利让给用户。

**存储能力升级：**EMR 已支持 ESSD 云盘。ESSD 云盘全称是企业级 SSD 云盘。ESSD 云盘的单盘 IOPS 有较大提升，在 EMR Kudu, HBase 等场景下有较好的表现和性能。ESSD 云盘分为 PL1、PL2、PL3 三种，目前 EMR 支持 PL1 级的 ESSD 云盘，从性能上 PL1 级已经基本满足用户需求。下图右下对比图分别是 ESSD 云盘和 SSD 云盘在块存储方面官网的标称的性能指标。通过对比图可以发现 ESSD 在单盘 IOPS 和单盘吞吐性能上都有较大幅度的提升。



**Alibaba Group**  
阿里巴巴集团



**APSARA**  
聚星 聚力

## EMR 基础能力升级

### 计算能力升级

- 适配阿里云ECS第六代企业级实例
  - SkyLake至Cascade Lake
  - 睿频从2.7Ghz加速提升至3.2Ghz
- 更强计算性能
- 更低产品价格

### 存储能力升级

- 支持ESSD，提升单盘IOPS性能
- 在EMR Kudu，HBase等数据量级不大，但对IOPS要求高的场景推荐使用
- 支持PL1 ESSD




#### 网络能力升级

支持VPC、VSwitch、EIP、NAT网关、云企业网、云防火墙、云盾、云安全中心、云审计、云监控、云日志服务、云原生网关、云原生网络、云原生安全、云原生应用、云原生数据、云原生AI、云原生IoT、云原生边缘、云原生生态、云原生未来

#### 安全能力升级

支持VPC、VSwitch、EIP、NAT网关、云企业网、云防火墙、云盾、云安全中心、云审计、云监控、云日志服务、云原生网关、云原生网络、云原生安全、云原生应用、云原生数据、云原生AI、云原生IoT、云原生边缘、云原生生态、云原生未来

## 2. EMR 软件栈升级

EMR Hadoop 目前支持 Hadoop2.7.2 和 Hadoop2.8.5 两个版本，社区 EMR3.0 版本逐渐成熟之后 EMR4.0 会将版本升级到 Hadoop3.1.2 版本。同时，EMR 4.0 新增支持 Kudu1.1.0 和 Delta Lake 0.3。EMR4.0 中 Apache Flink 升级至 1.8.1。在 Apache 社区 Flink 升级至 1.9.0 或 1.9.1 后，EMR 会再考虑 Apache Flink 升级。EMR4.0 支持 OpenLDAP。如果用户使用的是 OpenLDAP，则可以进行平滑迁移。下图右侧是当前 EMR 已经支持的一些版本，这些版本在 EMR 4.0 发布后会实时保持和社区更新同步，保证用户平稳使用。



### 3. EMR 生态集成

EMR+DataWorks: EMR 作为一个云原生软件, 在云上有一系列生态集成, 如OSS 的集成和ES 的集成。一部分用户同时使用了 EMR 和 Elastic Research 或云 Elastic Research。EMR 已经集成了 Elastic Research 组件, 可实现数据的打通。如下图左侧, 创建 DataWorks 项目时选择计算引擎, 之前只能选择 Maxcompute 和实时计算, 现在多了两个选择, 一个是 EMR, 另一个是刚开始公测的交互式分析。用户在创建 DataWorks 作业时, 可以将作业运行在一个已经存在的 Hadoop 集群上。在写的时候只要相应的输入 accessKey 和 accessID, 输入 EMR 的集群 id, 则可以将这个任务调度到 EMR 集群上。下图右侧展示了若用户目前以 EMR 作为计算引擎, 只要用户购买了 DataWorks 相应付费版软件, 就可以实现数据血缘跟踪, 表和相应列的源数据信息的查看。



EMR+PAI: 生态集成的另外一方面是阿里云上非常受欢迎的深度学习平台 PAI。目前大数据 +AI 是一个非常重要的场景。EMR 已经能够实现 TensorFlow On YARN 和 TensorFlow On Spark，可以通过 YARN 实现对 TensorFlow 作业的调度，作业可以跑在 CPU 以及 GPU 上。同时，TensorFlow 也可以跑在 Spark RDD 上进行计算。EMR+PAI 集成主要体现在将 PAI 这一个企业级分布式的深度学习计算框架部署在 EMR 上。因为 PAI TensorFlow 完全兼容于开源的 TensorFlow，同时在显存和分布式通信方面有深度优化。另一方面，PAI 的计算模型 PAI-FastNN 集成了 11 种经典的神经网络模型，能够跑在 EMR 上，实现开箱即用的效果。



## EMR 生态集成

EMR + PAI

### 大数据+AI

EMR已具备的AI能力

- TensorFlow on YARN
- TensorFlow on Spark

### PAI TensorFlow

- 企业级分布式深度学习计算框架
- 完全兼容开源TensorFlow的PAI-TF，在显存、分布式通信等方面做了深度优化。
- PAI模型工厂PAI-FastNN，内置11种业内经典神经网络，开箱即用。

## 4. EMR 数据迁移

EMR 正在与阿里云上另一款灾备产品 HDR-BD 进行集成实现数据灾备和迁移。目前支持社区 Hadoop2.7.x/3.1.x，HDP，CDH 版本。Source 和 Target 集群双活，实现 HDFS 和 Hive 数据的迁移，同时支持迁移到 OSS 上。



## EMR 数据迁移

### 阿里云HDR-BD双活数据迁移

- 支持社区Hadoop 2.7.x/3.1.x，HDP，CDH版本迁移到EMR
- Source/Target集群双活，实时迁移HDFS、Hive数据，支持OSS

### 三、Road Map

EMR 在未来几个月的重点工作首先是在本地盘实例运维。大部分用户会使用 ECS 的大数据实例。大数据实例最大的特点是存储成本相对较低，对于数据量比较大的用户是一种比较好的成本优化方式。对于大数据来说，几千或者几万块盘在管理时如果出现单盘的损坏，对作业的稳定性会产生比较大的影响。EMR 的主动运维能力主要体现在用户单键点击便可实现对磁盘的隔离，确定对坏盘的较为安全的处理方式。此外，EMR 与 PAI 和 DataWorks 会有比较深入的融合，构建开源的大数据 AI 平台以及开源大数据数据仓库。在未来，EMR 同时会基于 Kudu, Delta 以及目前较为流行的 Flink, Spark Streaming 等计算框架实现实时数仓的构建。在 2019 年底或 2020 年初，EMR 会推出第二代企业级大数据实例。第二代企业级大数据实例的计算性能会大幅提升，同时成本会持平或小幅节省，将更大的成本优化的空间贡献给客户。



# 大数据 & AI 开发平台

## 飞天大数据平台智能开发云平台 DataWorks 最新特性



刘天鹗

阿里云智能计算平台  
事业部高级产品经理

摘要：自 2009 年 DataWorks 立项，伴随着阿里巴巴集团登月计划、公共云和专有云的发布，直到 2018 年 V2.0 的发布，DataWorks 已经走过了十年的历程。本文主要分享了 DataWorks 商业化增值版本中高级功能，分别介绍了 DataWorks 基础版，标准版，专业版和企业版中具备的特色功能和每个功能点所适应的场景，帮助用户选择适合的 DataWorks 版本，更好的解决所面临的问题。

## 一、DataWorks 背景介绍

### 1. DataWorks 的发展历史

自 2009 年 DataWorks 立项，伴随着阿里巴巴集团登月计划、公共云和专有云的发布，直到 2018 年 V2.0 的发布，DataWorks 已经走过了十年的历程。整个过程经历了几个关键的节点，2009 年到 2013 年期间，DataWorks 具备调度 Hadoop 集群任务的能力。在此之后，由于数据量不断的增加，Hadoop 集群已无法支撑阿里巴巴集团海量增长的数据，阿里开始自研 MaxCompute+DataWorks。2013 年

之后，DataWorks 开始支持 Maxcompute 任务的调度。自此，阿里巴巴集团基于 Maxcompute+DataWorks，构建了整个阿里的数据中台。



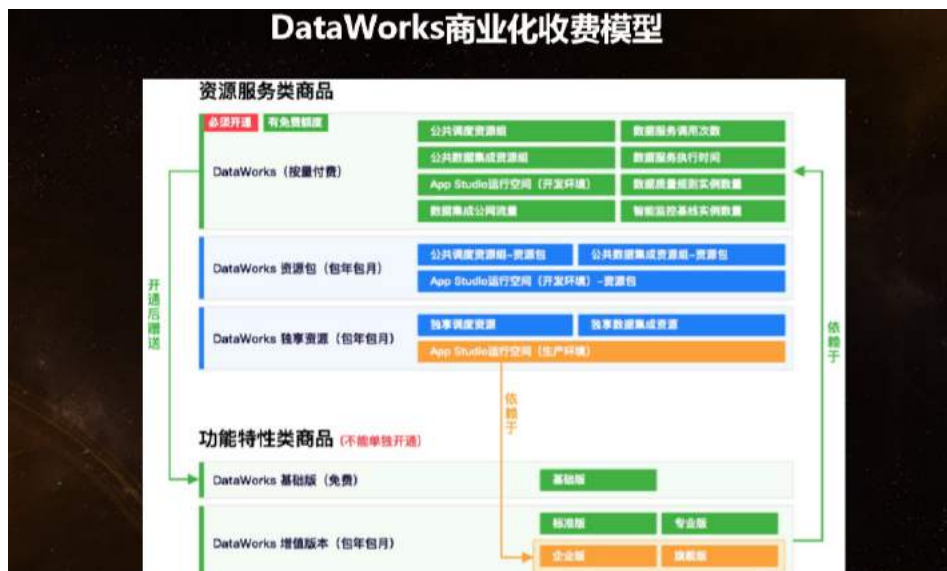
## 2. DataWorks—一站式大数据研发 + 治理平台

作为阿里巴巴的一站式大数据的研发平台，DataWorks 具备的基础能力可以分为两大部分，数据研发和数据治理。在 2018 年上半年之前，大部分用户使用 DataWorks 产品的目的是做数据研发工作，基于 DataWorks 的数据研发工作通常是将数据源通过数据集成传到 Maxcompute，再通过离线计算 Data Studio 任务进行定时调度。从 2018 年下半年开始，发布的 DataWorks V2.0 将集团内部数据治理功能无缝部署到了公共云，使每位公共云用户都能够在 DataWorks 基础版中具备较为完善的数据治理能力，包括数据血缘、数据质量监控、任务监控、数据审计，以及安全数据权限管控的功能。



为了让用户体验到更好的商业级服务，DataWorks 在 2019 年 6 月份实行了商业化。DataWorks 收费文档中展示了整个收费模型，该模型将 DataWorks 划分为两种类型的商品，分别是：资源服务类商品和功能特性类商品。功能特性类商品提供给用户的是 DataWorks 各模块的功能点，如：用户打开 DataWorks 界面便能看到数据开发 - 节点类型、运维中心 - 智能监控等功能，这些界面都是基于 DataWorks 的某个版本来呈现给用户的，DataWorks 基础版 / 标准版 / 专业版 / 企业版等。如果在此版本之上，如用户使用到其中的某些特定功能点，则会按照按量付费的方式进行扣费。

（注意：任何用户都必须开通 DataWorks 按量付费商品，开通后即可永久赠送 DataWorks 基础版。）



## 二、DataWorks 高级功能场景化介绍

### 1. DataWorks 基础版

DataWorks 基础版具备较为实用的特性，可以帮助用户快速构建数据仓库。DataWorks 基础版可以覆盖大数据研发的全生命周期，包括数据接入、数据开发、调度生产、可视化运维、数据质量监控，表权限管理，数据服务 API 构建，到数据最终呈现到某个应用的开发等全部模块。值得一提的是，在数据接入部分 DataWorks 新增了“批量上云”功能，如用户的数据存在多个 MySQL 数据库，每个数据库包含多个 DB，每个 DB 都有 n 多张表。此时用户可以通过批量上云功能，上传 EXCEL 形式的数据，快速建立起多个数据同步任务进行数据的快速上云。目前批量上云方式支持 ORACLE，MySQL 以及 SQL server。基础版目前也有数据质量监控功能，用户可以设置自定义的规则进行检查。



## 2. DataWorks 标准版

针对基础版，DataWorks 标准版在研发方面增加了较多复杂且专业化的节点类型，同时对实时 Flink 引擎也有了更好的可视化支持。DataWorks 标准版更多的是面向大数据体系发展比较快的企业，企业在数据体系发展比较快时，数据质量问题、安全问题通常会逐步暴露，出现相关质量、安全隐患，因此在数据治理方面，DataWorks 标准版提供了相应的能力帮助用户解决问题。

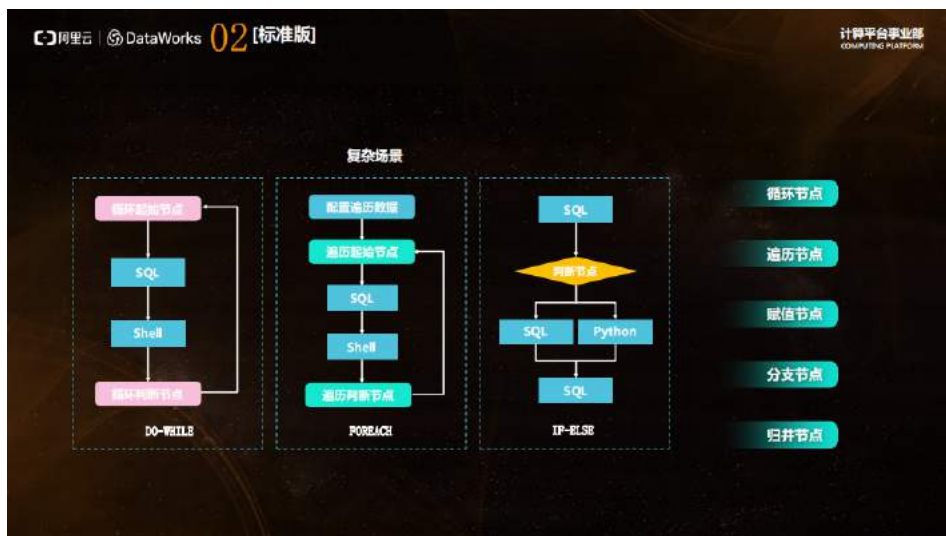
如下是 DataWorks 标准版场景化案例：

### 场景一、分支节点实现特定时间执行任务

比如，用户会提出诉求，既能否根据某个特殊的判断决定任务工作流的下游是否要跑某个任务。如下图，用户要判断今天是不是每个月最后一日，而每个月的最后一天并不是固定的 30 号。



此时，如果通过 DataWorks 传统的简单场景的线性依赖，很难实现用户诉求。DataWorks 标准版提供了几种负载类型的节点，包括循环、遍历、赋值、分支和归并节点，让用户自由组合节点，以便实现更复杂的场景。



上述场景如何通过复杂节点实现？如下图所示，案例中需要使用两种节点，赋值节点和分支节点。赋值节点中用户可以通过 Python 判断今天是不是这个月的最后一

天，如果是，赋值节点负责将“1”作为一个传参传给下游，如果不是则传“0”。分支节点接收到上游传参“1”，则执行左边分支，如果接收到“0”，执行右边的分支。



## 场景二、SQL 代码复用

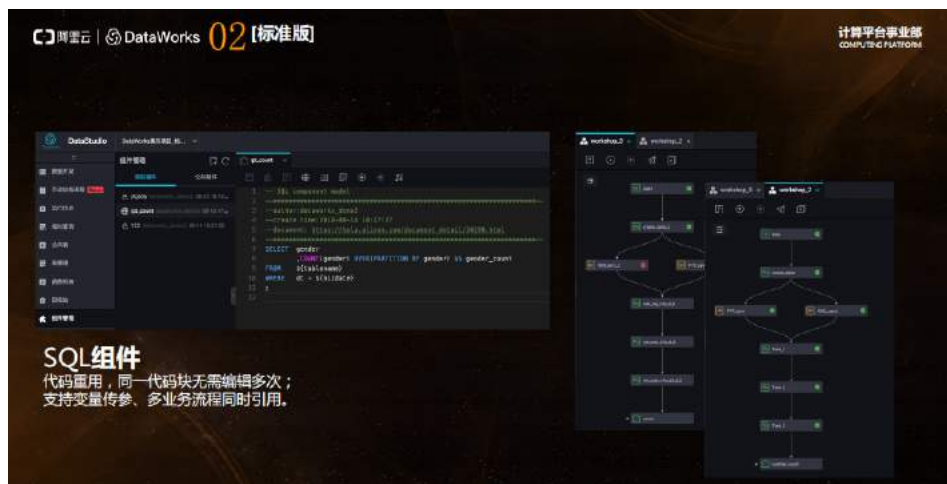
随着用户 SQL 代码规模越来越大，某些代码段会成为一个公用的代码段。用户需要将这些代码做成公用代码供其他人复用。传统方法是复制代码，而传统的做法存在较高的风险。复制代码一方面效率较低，另外很容易对代码进行剪切，原代码会被改动。

**场景** SQL代码块需要给他人重复利用，复制粘贴效率太低且容易误操作。需解决SQL代码复用的问题。

```

select
    p.channel,
    rownumber() order_id
from
    (
        select
            s1.channel,
            s1.id
        from
            (
                select
                    trim(channel) as channel,
                    min(id) as id
                from
                    ods_trd_trade_base_dd
                where
                    channel is not null
                    and
                    dt = ${tp_yyyymmdd}
                    and
                    trim(channel) <> ''
                group by
                    trim(channel)
            ) s1
        left outer join
            ods_trade_channel s2
        on
            s1.channel = s2.trade_channel_edesc
        where
            s2.trade_channel_edesc is null
        order by
            id
    ) p
    
```

DataWorks 标准版提供了 SQL 组件功能，允许用户定义一个公用的 SQL 代码块，并且在其中设置自己需要的变量。其他人需要这个 SQL 代码块，就将其作为一个节点投入到一个业务流程，同时对设置的参数进行赋值，实现 SQL 代码的复用。



### 场景三、实时场景

某些企业需要构建实时的数据仓库，而实际上，往往需要技术非常娴熟的一批团队才能够搭建起整个流计算的系统。Flink 的整个流程包括，用户在本地进行流任务的开发调参，开发结束后上传到 Flink 集群，再通过命令行运行。同时，运维过程也非常麻烦，如果出现超时或其它问题，Flink 需要在多个地方检查日志，排查难度非常高。



DataWorks 标准版提供了 DAG 开发以及 SQL/DAG 互转功能。用户可以对每个处理的数据和处理动作都拖入一个节点，将节点串联成一个工作流，依次对数据进行处理。如下图，点开每个节点的结构，如第一个 datahub 是个数据源，输入时会会有一个字段，到第二个固定链被切分，切分为多个字段。用户可以通过可视化的方式在线进行配置字段。在配置完整整个流程之后，点击右上角切换 SQL 模式，将其转化为 Flink SQL。DataWorks 标准版除了支持图形化的方式配置 Flink 流计算的处理流程，也支持用户使用原生的 SQL 进行任务的编写。

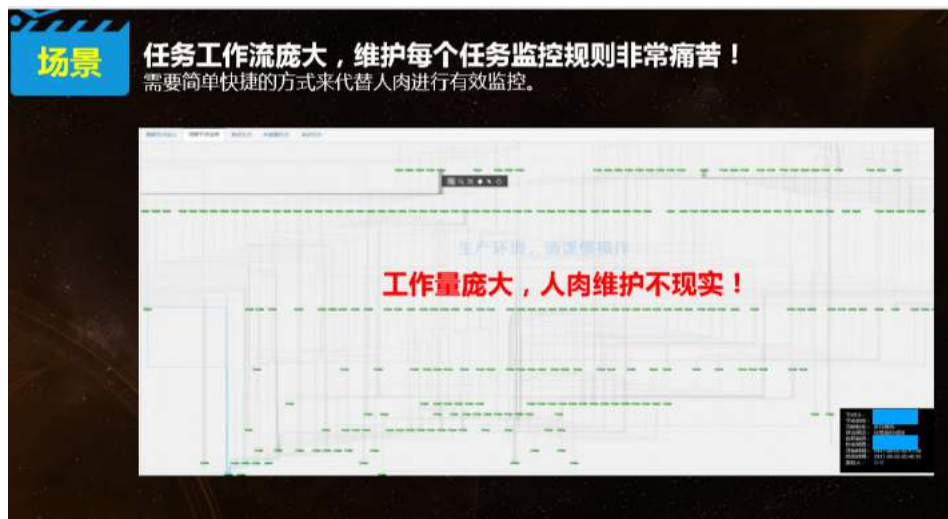


另外，DataWorks 专业版中具备本地调试功能，在每个节点用户可以点击采样看到采样的时间点采样数据。如果不符合预期，用户可以及时做出调整。传统 Flink 不具备此功能，只能等到所有任务都执行结束，才可以查看结果数据是否符合预期。用户在还没有将整个任务上线时，本地调试功能便能够检查数据的生产是否符合预期。在快速构建实时流计算体系时，DataWorks 大大降低了用户使用实时流计算引擎的门槛。

#### 场景四、任务监控场景

随着用户的工作量越来越大和越来越复杂，人肉维护每个任务监控变得非常痛

苦。下图中展示的密密麻麻的节点，是阿里集团内在运营中心展开的 DAG 截图。传统监控配置手段对每个任务之间配置的超时进行报错，但当工作流的体量非常大时，传统的配置手段不是很现实。由于每个任务的数据体量不固定，导致任务时间增长，用户无法对每个任务的报警阈值做调整。这时需要一个既快捷又智能的方式，用户只需配置一次，在此之后合理监控，对整个工作流实现整体监控的功能。



DataWorks 标准版的智能监控功能，允许用户仅对产出数据的最后一个节点配置一个基线时间，之后 DataWorks 智能监控系统在后端针对往常的每个任务的运行时间以及 SQL 语义进行分析，规划出关键路径，分析每一个节点的最晚开始和结束时间。一旦上游的某一个中间任务出现变慢或者出错的情况，智能监控系统中报警的机制会通过邮件、短信或者钉钉推送给用户。用户在接收到中间任务出现的事件时就可以快速响应，快速处理任务所出现的问题。如果问题能够得到及时解决，最终产出的任务就能够保证在用户定好的时间点产出。既对整个工作流设置基线，只要保证基线不破线，则智能监控功能可以保证数据准时产出。

阿里云 | DataWorks 02 [标准版]

计算平台事业部  
COMPUTING PLATFORM

智能监控

监控报警的痛点

监控数量  
监控所有任务是不现实的

配置难度  
为每个任务配置监控规则极为繁琐

报警时间  
每个任务所需报警的时间都不一样

智能监控核心功能

- 智能识别关键路径，合理设定报警阈值
- 任务异常产生事件，自动评估事件影响范围，通知相应人员
- 灵活报警方式配置，支持钉钉群机器人

智能基线监控

## 场景五、寻找脏数据的源头

随着用户的数据量飞速增长，数据质量问题也会凸显。DataWorks 标准版提供了让用户自己配置，自定义规则或者预设模板来监控数据质量的功能。如下图，一张用户消费表，以往每天用户花费是 100 多万，忽然有一天统计到的数据是 2000，出现了不符合预期的情况。

阿里云 | DataWorks 02 [标准版]

场景

某交易数据字段今日产出异常值，需找脏数据的源头。  
需要简单快捷的方式来代替人肉进行有效监控。

User_count	Order_count	Order_amount	bizdate
564	1242	1234876	20190810
567	1120	1164965	20190811
578	1235	2000	20190812

DataWorks 标准版中数据血缘功能能够帮用户很好的解决数据质量的问题。数据血缘功能可以为用户提供数据地图，能够识别每一张表的上游表和下游表，每一个字段的上游字段和下游字段，既以字段的形式呈现数据的历史。数据地图也有其它的用处，用户只要在一个阿里云主账号下面的同一个 region，所有被加到当前项目的用户都可以在数据地图中搜索表，也可以在每张表的详细信息中看得到当天这张表的产出信息。数据地图对于日常表的使用也是必不可少的一个功能，可以查看字段说明、每日分区产出等信息。

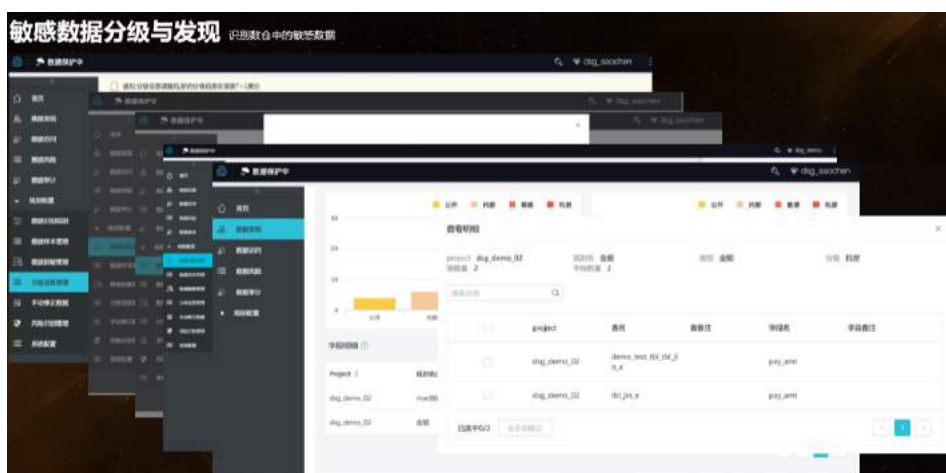


## 场景六、数据审计

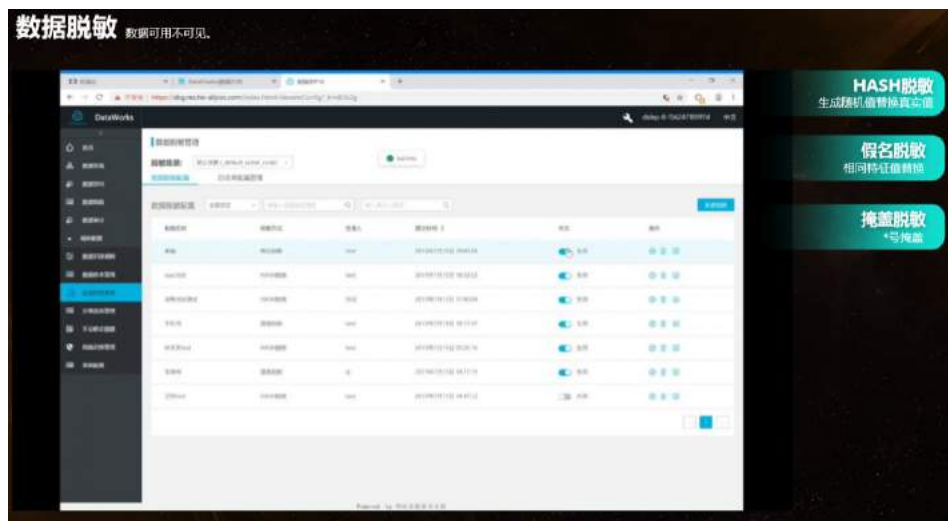
在保证数据准时产出、产出准确之后，如何保证数据不丢失？为此企业管理者通常会推动企业构建数据审计体系。DataWorks 数据保护伞模块帮助用户对数据敏感级别进行分级和分类，以可视化（图形化）方式展示敏感数据，并且审计内部人员对敏感数据的访问行为，同时保证敏感数据可用不可见（既数据脱敏）。DataWorks 标准版提供了以上四个功能，帮助用户落地最基本的数据审计的能力。



首先，DataWorks 标准版提供了一个能够让用户自己定义字段、进行字段扫描的能力，对字段的敏感级别进行分类。如下图，用户可以在页面中创建几个分级，包括公开、内部、敏感和机密。然后设定数据类型，只要扫描到表中有 payment 字段，就将其定位为金额字段。设定的规则自第二天便开始生效，从零点开始进行扫描，扫描结束后展示整个项目中关于金额类的字段。



敏感数据访问行为审计功能将访问的行为记录下来，帮助管理者发现访问敏感字段的用户的行为，必要时管理者与相关人员进行一定的沟通，建立内部沟通的机制。数据脱敏目前有三种脱敏方式，包括 HASH，假名和掩盖脱敏。



DataWorks 标准版目前已经覆盖了大部分数据研发和基本的数据治理的功能。对于数据业务发展非常快，并且对数据治理有一定要求的企业，在初期可以选择 DataWorks 标准版来满足自己的诉求。

### 3. DataWorks 专业版

DataWorks 专业版提供可拓展数据服务的 API，能够提供更灵活、更高可用的服务能力。同时在数据治理方面，DataWorks 专业版的安全能力也有了一定的加强。

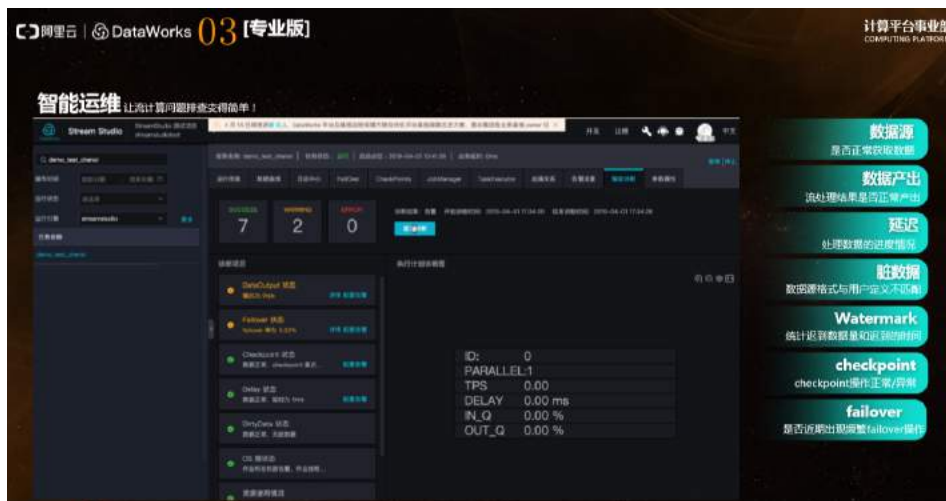
#### 场景一、API 对接报表系统

第一个场景，企业有不同数据结构的报表系统，需要以可视化方式展示数据。如一家银行需要对公司年度放贷的同比增长近做柱状图统计，此时银行正好有一套可视化的系统，其中有一个柱状图的组件，组件要求返回接收 JSON 格式的对象包数组数据结构即可正常显示。这时可以使用数据服务中的过滤器和函数功能对 API 返回





如下图所示，DataWorks 的智能运维功能对数据源、数据产出、延迟、脏数据、Watermark、Checkpoint、failover 提供了可视化检查，发现任何的错误都可以及时抛给用户，帮助用户定位错误。



### 场景三、非通用脱敏数据审计

DataWorks 专业版在数据安全方面有了本质的提升。企业中不常见的敏感数据，

需要纳入敏感数据的行列进行审计。这时需要有个功能是能够让用户通过选择某一类敏感数据的类型匹配字段，自定义的扫描字段。如下图，Bankcard id 是银行卡号字段，通过 DataWorks 标准版功能就能够识别到此字段，并列为敏感字段。但下表中 Product description 字段是用户购买的商品，其中某些记录值是敏感的，某些则不是，这时无需将整个字段都列为敏感信息。DataWorks 专业版通过自定义内容扫描的方式来正则匹配，用户也可以选择自定义，点击内容扫描，只要是满足标准的正则的规则都可以进行匹配。这满足了在特定场景下敏感数据识别的诉求。



**场景** 企业特有（非通用）敏感数据需纳入敏感数据进行审计。  
需识别不具有普遍性的敏感数据。

name	Order id	Product id	Product desc (非通用敏感数据)	Bankcard id (敏感可识别敏感数据)	Order date
Jack	23435234124	41235	私人用品-酒	6220000102034567897	2019-07-11
Rose	87677546454	64313	运动用品-篮球	6222000102034567897	2019-08-12

#### 场景四、审计敏感数据访问行为

管理者不仅要审计每个人员对敏感数据的访问行为，还需要审查具备某个表权限的人员的风险行为，检查是否在非预期的时间使用了敏感数据。DataWorks 专业版可以帮管理者识别高危访问行为，在页面上呈现出访问的 SQL，访问时间和访问的敏感数据。若不符合预期，管理员需要其进行内部审核。



## 4. DataWorks 企业版

DataWorks 企业版增加了数据服务 API 编排能力。数据中台解决的问题是“存”、“通”、“用”问题。既需要将数据规范进行集中存储，使各系统间的数据统一流通，进行统一规划加工，同时将最终数据供业务方使用。而使用数据的业务方需求各式各样、千奇百怪，如数据中台可以快速慢速业务方诉求，则该数据体系在企业内就会赢得更多信任，具有越来越重要的企业战略地位；如不能及时满足用户的需求或不符合预期，则数据体系会越来越被边缘化。如何满足业务方纷繁复杂，各种各样的需求？DataWorks 企业版数据服务目前提供了服务编排的能力，可以满足各种比较复杂的场景。

### 场景一、应对临时需求

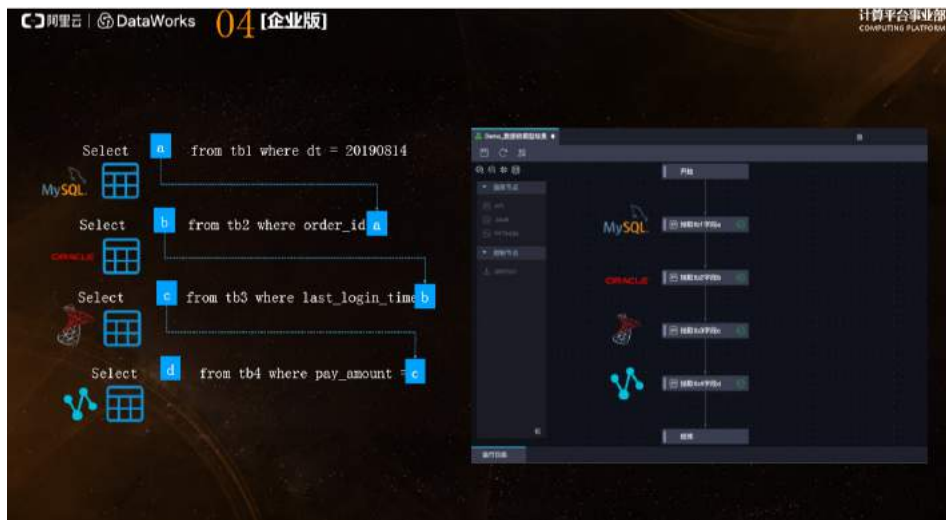
某些临时性需求，如：审计人员会随机不定时地抽取各个数据源进行对比；或者风控系统要从多张表中取得相互依赖的数据，最终取得最下游的数据进行决策，都是较为复杂的业务诉求。

如下图所示，审计人员要将几张表 join 之后进行对比，或从多张表获得相互依

赖的数据，找出最下游的数据进行决策。每一个表 select 的结果，都是下一张表 where 后面的条件。从数据开发角度较为常规的方法是将数据拉到数据仓库中，对表进行 join 后产出，最终得到结果进行对比。但这非常费时费力，同时会耗费计算存储资源，并且任务的维护性价比不是很高。



DataWorks 企业版提供的数据服务编排能力允许用户构建多个 API，将 API 构建成复杂逻辑的复合服务的功能。用户只需传入一个参数，下游 API 便可以根据每个上游 API 的产出作为输入，最终产出结果。如下图示例，在数据库里面创建一个编排的流程，拖入四个 API，每个 API 都会抽一张表的一个字段，抽到的结果传入下一个 API 作为入参。API 接到入参进行一系列的计算之后，将结果作为出参传给下游作为入参。经过一系列流程，就可以实现整个逻辑，用户不需要将所有数据都导到数据仓库里进行统一加工。数据服务支持多种数据源，包括 MySQL，Oracle 或者其它引擎。



在数据服务里也可以构建如下图所示的工作流。下图左边是 join 的 API，API 将两张表 join 后返回两个字段，最终形成字段表。最后拖入 Python 对两个 API 返回的结果进行对比，对比出来输出结果。如此便满足风控系统的审计人员不定时抽取多张表复查数据对比的诉求。



## 场景二、安全能力升级

DataWorks 企业版在数据治理方面有了安全能力的升级，提供了场景 PII 敏感数据的识别模型。用户只需设置一个字段类型，DataWorks 企业版就可以帮用户去识别某些敏感字段。DataWorks 企业版中已经预设了部分敏感字段的算法，包括邮箱、座机、IP 等。DataWorks 企业版有一个特点，既进行安全知识的输出，如用户打开这个功能开关，便可以实现对同一天多次使用高相似度的 SQL 进行查询，帮助用户一键使用。DataWorks 企业版目前提供个性化需求，用户可自建样本库，定义敏感数据，在内容扫描模块中维护的自建样本库。



## 场景三、定制化开发

部分用户的诉求是对 DataWorks 界面进行企业的定制化开发，DataWorks 企业版通过 APPStudio 模块中高级 API，进行界面的二次开发。目前，DataWorks 企业版为用户提供了 部分 API，用户可以通过 DataWorks 企业版中的 AppStudio 模块来进行调用，并构建如下图所示的数据资产页面。



### 三、购买建议

DataWorks 基础版：一般适用于构建数据仓库的需求非常紧迫，并且人员有缺口的情况。因为 DataWorks 基础版提供了可视化数据批量上云的能力，同时数据开发页面也支持可视化，包括 SQL 编写，工作流构建，依赖关系构建和任务运维。

DataWorks 标准版：适用于数据体系发展比较快的企业。当企业数据量增加，任务的体量成规模的增加，安全治理的问题凸显。DataWorks 标准版能够满足用户基本的数据治理的能力以及一些高级的数据开发能力。

DataWorks 专业版：适用于需要构建相对成熟的数据服务体系的企业。如果安全要求较高，DataWorks 专业版提供的风险识别能力是非常必要的。

DataWorks 企业版：当用户认为当前的数据体系提供服务的时候不是特别灵活，可以尝试使用 DataWorks 企业版数据编排服务，以满足整个数据中台对于业务方的各种各样的诉求。DataWorks 企业版也会赋予用户越来越多数据安全和数据审计方面的能力。同时二次定制的数据开发的能力也会越来越完善。

# 飞天大数据平台机器学习 PAI 最新特性



郁环宇

阿里云智能计算平台  
事业部产品经理

本次分享主要围绕以下五个方面：

- PAI 产品简介
- 自定义算法上传
- 数加智能生态市场
- AutoML2.0 自动调参
- AutoLearning 自动学习

## 一、PAI 产品简介与功能发展

### 1. PAI-Studio

在下方 PAI 的架构图中，最底层的是 PAI 的计算框架和数据资源。PAI 可支持 MaxCompute、OSS、HDFS 和 NAS 等多种数据资源。在数据资源和多种计算框架基础之上，诞生出了 PAI 的最早形态：PAI-Studio——可视化建模实验平台，Studio 中包括了两百多种算法，覆盖了数据预处理，特征工程，模型训练，预测评估等全链路的实验流程操作。用户可在 PAI-Studio 中以拖拽的方式构建实验，而无需复杂的操作。此外，PAI 内置了鲲鹏计算框架，可支持百亿特征，百亿样本的超大规模矩阵训练。在最初建立过程中，PAI-Studio 的用户定位为中级的算法工程师，即一个不需要很高的技术门槛就可以上手使用的算法平台。有了可视化建模 Studio，PAI 就可以拥有为用户提供业务支持，如构建推荐系统、金融风控、疾病预测或新闻分类等的能力。

## PAI产品简介——产品架构图



## 2. PAI-EAS

然而，从 Studio 中算法和实验的构建，到真正成为用户可用的模型服务，其中间还存在一个门槛，即如何将模型部署为在线服务。用户一般需要耗费较大的精力在此之上。为了解决这个问题，PAI 平台随后推出了 PAI-EAS 模型在线服务功能，为用户提供 EAS 在线服务的一键部署能力，大大简化操作，连接模型构建与生产服务。除了一键部署，PAI-EAS 模型在线服务功能还支持版本控制、蓝绿部署和弹性扩容等特性功能。通过 PAI-EAS 模型在线服务，用户可以方便地进行构建实验，并将模型进行在线部署，最后应用到实际业务当中。

## 3. PAI-DSW

在 PAI 之后的发展过程中，出现了新的需求，即有的工程师希望在整个实验构建过程中有更大的自主发挥空间。为此，PAI 推出了 PAI-DSW 版块，其特点是：使用 notebook 进行建模、内置 Jupyter 开发环境、内部深度优化的 TensorFlow、可视化编辑神经网络。由需求的初衷不难发现，PAI-DSW 和 PAI-Studio 最大不同点就在于它为拥有更多算法背景技术的工程师提供更大的发挥空间，因此适合于高级

的算法工程师。

## 4. PAI-AutoLearning

在解决了高级、中级算法工程师的需求之后，PAI 又进一步推出了专门为初级算法工程师量身设计的全新 PAI-AutoLearning 功能（详细功能特性会在下文介绍）。继而，拥有不同的算法能力的工程师都可以在 PAI 找到适合自己使用的产品类别。

## 5. 智能生态市场

通过以上 PAI 的功能设计，用户可以根据自身特点迅速找到适合自己的板块，并快速部署服务到业务中去，但这些都是需要用户自身来开发完成。随着 AI 行业应用的不断发展，如何让借助他方的能力、智慧，来快速解决自己的业务需求，又成为了一个新的问题解决思路。为此，PAI 推出了——智能生态市场功能版块。用户可以在智能生态市场中寻找自己业务问题的解决方案（如模型、算法或者应用等），快速获取能力，避免了不必要的开发人力资源的耗费。拥有对应技术的开发人员或公司，又可以在智能生态市场中一展才能，售卖发布商品，并获取相应回报及品牌。

## 二、自定义算法上传

自定义算法上传是 PAI-Studio 内的一个功能。机器学习在实际的应用过程当中，有千万种与行业结合的可能性和应用场景，因此用户会有多种个性化的需求。尽管 PAI-Studio 为用户提供了两百多种算法组件，但依然不可能满足每一个用户的每一个需求。通过自定义算法上传功能，用户可以开发自己的算法组件，方便后续使用。

自定义算法上传包含三个特性。首先，自定义算法上传功能兼容 Spark 生态，支持 Spark 和 PYSpark 框架，在这个框架下用户可以任意开发自己所需要的算法。其次，自定义算法上传功能支持便捷发布。为了避免在自定义算法发布

过程中，花费的精力大于最终带来的效率收益，PAI 在自定义算法上传功能设

计的过程当中，着重突出便捷发布，提供分钟级的算法发布体验。自定义算法上传第三个特性是可视化配置。从上传算法包，到成为真实的可使用组件之间还包含一个过程，即配置组件的算法参数以及相关配置。PAI 为用户提供了可视化的在线操作配置的方式。



### 三、数加智能生态市场

#### 1. 智能生态市场角色

智能生态市场是大数据和 AI 领域的一个淘宝平台，其最大的作用是连接了开发方和业务方。在此之前，开发方有自己的能力和技术，业务方有需求和想要解决的问题。但两方一直没有办法很好的连接在一起，通过数加智能生态市场平台，将开发方的技术和业务方的需求进行对接，两方都可以通过平台获取利益。首先，开发方可以打通产品快速发布售卖，收获品牌效益，同时掌握市场动态需求。另一方面，业务方可以通过智能市场更便捷的获取适合自己的业务解决方案。降低探索新业务的成本，扩展基础业务，快速实现公司技术的优化。

## 数加智能生态市场



如下图，在智能生态市场链路出现以前，用户只有两种构建业务解决方案的途径，一个是通过自己使用机器学习 PAI 来开发，其中包括实验的构建、模型部署和应用等步骤；二是选择行业 ISV，ISV 通过在行业中的经验为客户构建出不同的，可部署在实际业务中的模型，解决客户的业务需求。但这两种方案的前者需要投入较大的人力成本，后者需要较大的财力成本。有了数加智能生态市场之后，用户可以选择第三条平衡人力和财力成本的新途径，通过向算法模型开发者购买最新的算法模型或者解决方案，解决自己的实际业务问题。这恰恰是智能生态市场所建立的功能目标，以及它可以为客户和开发者带来的价值。

## 数加智能生态市场



## 2. 数加智能生态市场的子版块

数加智能生态市场包括四大板块。第一个是市场版块，在市场板块中用户可以进行大数据智能商品的买卖交易。即有业务需求的用户可以找到业务问题的高效解决方案，开发者可以通过技术的传播和共享，获取回报和品牌效益。第二个板块是培训板块，培训板块可以提供市场相关平台的使用指导和案例讲解，为用户提供实用的培训，快速上手大数据的智能业务，实现零基础入门。第三个板块是认证板块，主要面向开发者同学。在认证板块中开发者同学可以获取 Apsara Clouder 技能的认证证书，通过认证提升开发的专业技能，并获得官方资格的评定。第四个板块是论坛版块，大数据智能用户以及相关兴趣爱好者都可以在论坛相互交流，获取最新的前沿技术和知识。通过以上四个板块，PAI 为用户构建出了一个生态圈。智能生态市场为用户提供培训，为开发的同學提供认证。用户和开发同学之间通过市场建立互相买卖的交易关系，并且获取各自的需求。同时三者之间可以构成一个论坛的有机生态。



## 3. 商品发布流程简介

下图是数加智能生态市场主界面。开发者通过主界面进行卖家入驻，填写个人信息，提交审核。通过审核之后，卖家拥有了在市场上售卖商品的资格。卖家可以进

行商品发布，首先需要选择商品的种类，如发布解决方案商品，输入商品名称，商品介绍以及来源渠道。在来源渠道里中填写商品的跳转链接，由于解决方案商品属于比较开放式的商品的类型，开发者可以发布自己相关的名片，方便更进一步的沟通和联系。此外，还可以在数加智能生态市场中发布算法商品。算法商品和自定义算法上传功能是相通的。自定义算法上传配置完之后有一个发布按钮，点击发布按钮就可以发布到数加智能生态市场。商品通过审核之后，可以点击上架操作，上架操作完成之后页面中生成一个商品的智能客服机器人，其中已经预置了商品的基本问答，在客户想要来了解商品信息时，可以通过智能客服机器人进行初步的商品信息获取，若智能客服机器人无法提供足够的信息量，还可以通过预留的钉钉号进行进一步的沟通联系。

## 数加智能生态市场



## 四、AutoML2.0 自动调参

### 1. AutoML2.0 特性

AutoML 自动调参功能是 PAI-Studio 中的核心功能之一。PAI-Studio 可以进行可视化的建模操作，但模型构建好之后，如何进行参数调整，并达到最好的实验效果，是用户经常面临的问题。AutoML 即为了解决这个调参问题。AutoML2.0

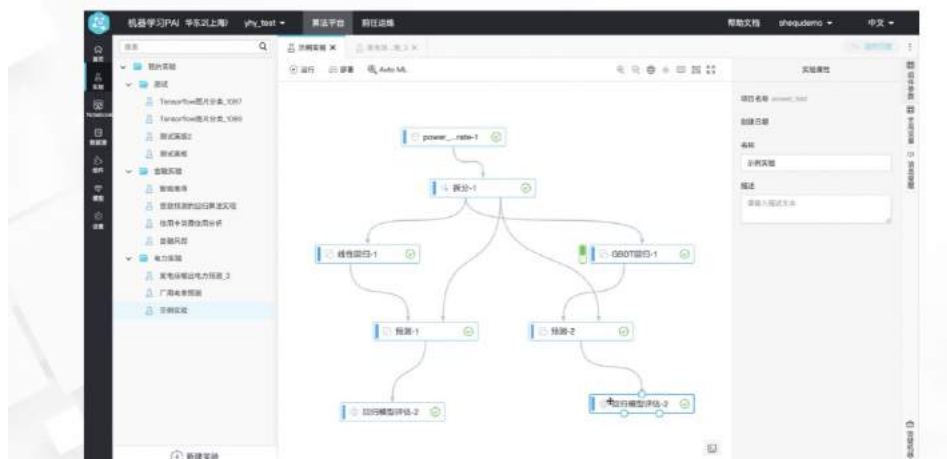
自动调参功能包含以下特性：首先是一键自动调参数，包括自动调整参数、模型评估和模型传导。其次，AutoML2.0 支持七种调参算法，如 GritSearch、Random Search、PBT、Gause、Evolutionary Search 等常见的调参算法。实践证明，AutoML2.0 可以帮助用户节省 90% 的工作量，大幅降低了用户在建立机器学习模型过程中所花费的时间以及人力成本。



## 2. 自动调参实践示例

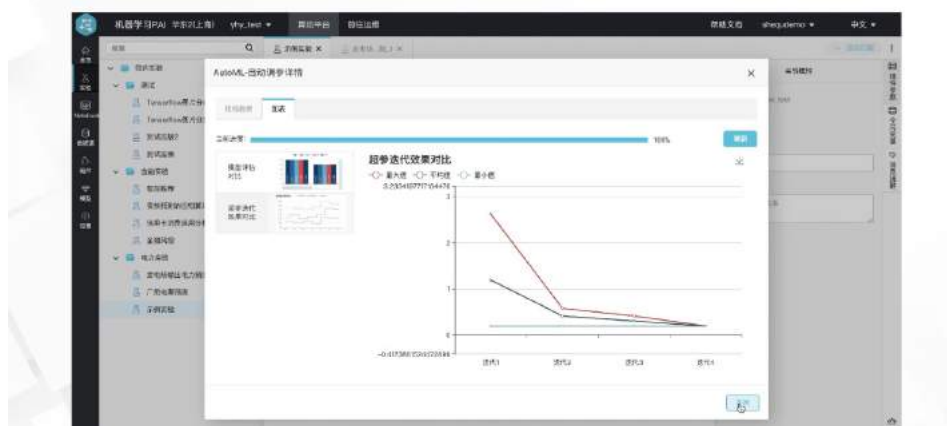
在这里，我以 GBDT 回归模型调参为例，这里我们选择 Evolutionary 优化调参方式，下面需要配置 Evolutionary 优化调参方式所需要配置的搜索的样本数目，探索次数，收敛系数，定义参数范围。配置结束后，便会自动生成模型，示例中一共生成了 11 个模型，那如何从 11 个模型中选择最好的模型？自动调参页面中为用户提供了选择的标准，MAE，即通过 MAE 对生成的模型的质量高低进行排序，用户可以保存前几名模型，进行进一步的操作。

## AutoML2.0 自动调参



如下图，在自动调参详情页可以看到多次迭代之后，对模型效果的改善是非常直观的。随着迭代次数的逐步增加，模型的效果越来越好。纵坐标是 MAE 值，MAE 值越低，代表模型的误差越低。随着迭代次数上升，模型的质量也越来越高，这证明了 AutoML 调参功能的有效性。

## AutoML2.0 自动调参



## 五、AutoLearning 自动学习

### 1. AutoLearning 特性

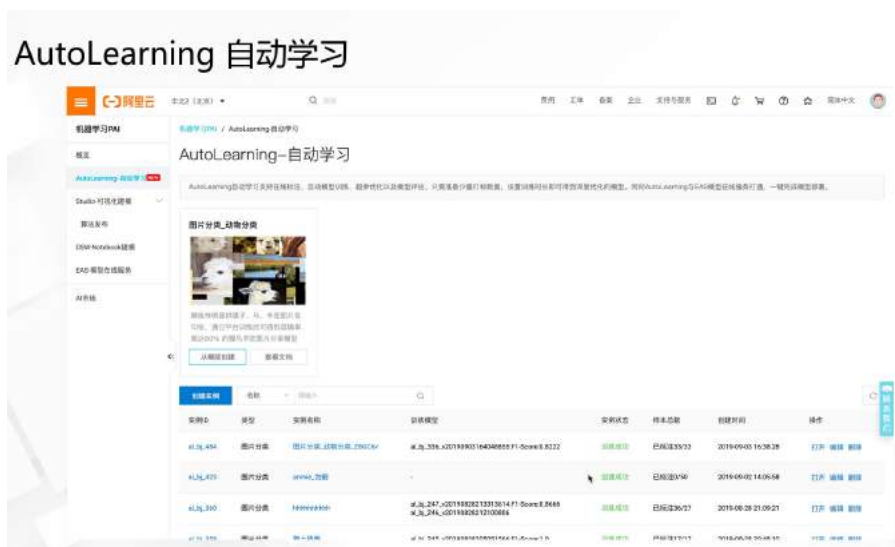
PAI-AutoLearning 自动学习功能是 PAI 最新推出的功能之一。在 AutoLearning 自动学习功能出现之前，PAI 中的 PAI-Studio，PAI-DSW 分别为中级算法工程师和高级算法工程师对应解决了他们的实验构建需求。而 PAI-AutoLearning 则通过提供小白级、零门槛的一个工具，解决了初级或入门级算法工程师的实验构建需求，使更多的人参与到机器学习，使用机器学习为自己的业务带来价值。AutoLearning 自动学习功能有以下亮点。首先是零门槛使用，即功能开箱即用，对于小白同学特别友好。第二是最低基于 5 张图片就可进行一次学习。通过强大的迁移学习框架，PAI-AutoLearning 可实现少量数据的有效学习，学习训练的结果依然非常出色。第三个亮点是 AutoLearning 实现了一站式解决方案。一站式包括从数据标注、模型训练、模型部署的整个过程，帮助小白用户也可以快速地入门机器学习的实际应用。



### 2. AutoLearning 自动学习使用实例

在 AutoLearning 自动学习功能板块，点击从模板创建。模板创建是适用于新手快速入门的一个功能体验，用户可以迅速的体验到整个自动学习的功能特性。页面中预置好了实验中需要的训练数据，如图片分类实验，数据集中包含不同的动物种类，

通过训练可以生成准确识别动物种类的深度学习的模型。



首先，在深度学习训练之前需要对深度学习算法提供有效的数据。有效数据需要对数据进行标注。因此，第一步是对数据的打标，AutoLearning 自动学习板块内置集成数据标注功能，帮助用户快速的进入数据打标。第一步是在图片中添加标签，动物种类包括羊驼、马和骡子，快速勾选同一类的动物，将其标记为羊驼、马或骡子。在开始训练页面中输入训练时长，训练时长是决定最终模型训练效果的一个关键的因素，训练的时间则越长模型效果越好。Auto Learning 板块特色是可以在短时间内快速训练出较为精准的模型，只需十分钟就可以训练出一个训练分类效果不错的模型。相比于传统深度学习模型训练这点的改进非常明显。此外，开始训练页面中另外一个选项是增量训练。增量训练表示是否在原有的训练模型基础上继续进行进一步的训练。在模型训练及评估界面，可以看到训练好的模型结果，模型指标包括准确率、精准率，召回率值，表示模型对当前的训练结果的有效性程度。用户可以点击上传新的图片，检验模型预测性能。那么验证好的模型该如何运用到实际生产当中去呢，Auto Learning 自动学习功能已一站式业务构建流程，用户在此界面可直接点击前往 EAS 部署就可以将模型部署为服务，应用到实际的生产中进行产出。

# 搜索与推荐

## 飞天大数据平台 OpenSearch 最新特性



李雪莹

阿里巴巴搜索推荐  
事业部高级产品经理

摘要：阿里云开放搜索（OpenSearch）是一款结构化数据搜索托管服务，其能够提供简单、高效、稳定、低成本和可扩展的搜索解决方案。OpenSearch 以平台服务化的形式，将专业搜索技术简单化、低门槛化和低成本化，让搜索引擎技术不再成为客户的业务瓶颈，以低成本实现产品搜索功能并快速迭代。本文将为大家介绍 OpenSearch 的最新推出的电商查询语义理解和搜索算法平台两个新功能。

本次分享将主要围绕以下三个方面：

1. OpenSearch 产品简介
2. 新功能介绍 – 电商查询语义理解
3. 新功能介绍 – 搜索算法平台和人气模型

### 一、OpenSearch 产品简介

OpenSearch 产品上云之前在阿里巴巴集团内部一直都扮演着搜索中台的角色。除了淘宝、天猫之外，阿里巴巴集团的其他核心业务产品比如飞猪、菜鸟等也都是通



手使用，并将关键路径流程化、可视化出来，仅需要几步就能够完成基础服务的搭建。如果业务经常会出现波峰波谷情况，比如 618 或者双 11 大促，都可以在 OpenSearch 上随时扩缩容，并且能够及时生效，无需提前申请配额，因此也非常方便。

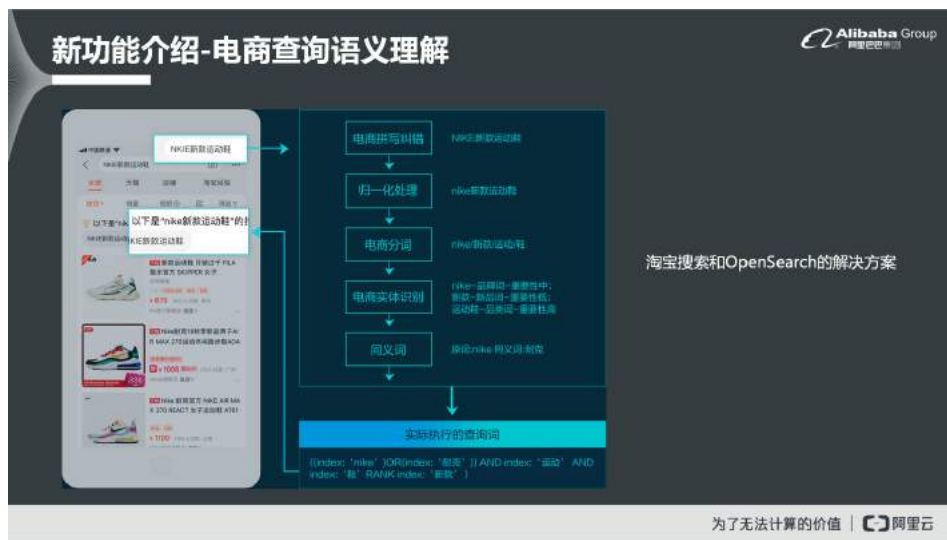
## 二、新功能介绍 – 电商查询语义理解

首先为大家介绍 OpenSearch 的电商查询语义理解这个新功能。之所以要实现电商查询语义理解这个功能，是因为在电商场景里面，搜索引导的成交往往占比达到 60% 以上，因此电商对于搜索的精准度要求非常严格。但是在实际的搜索场景中，用户输入的关键词往往非常口语化，而站内商品的标题、详情描述、类目以及标签等都属于比较书面化和规范化的文本，因此就导致查询词和商品信息无法匹配。如果站点没有实现语义的 Mapping，用户往往搜不到其想要的商品，对于平台而言，这就使得无结果率或者少结果率都会非常高。基于上述的背景，OpenSearch 就把淘宝搜索内部的智能语义 Mapping 功能集成到了 OpenSearch 内部，称之为查询语义理解，该功能专门为电商领域提供服务，因此垂直性也非常强。



## 淘宝搜索和 OpenSearch 的解决方案

这里为大家介绍一个实际案例，来帮助大家理解电商场景的语义查询 Mapping 功能对于实际业务的具体作用。下图中，用户在搜索框输入了“NKIE 新款运动鞋”，这里的“NKIE”其实是输入错误的“NIKE”，而搜索返回的结果其实都是符合用户实际需求的 NIKE 的新款运动鞋。那么，在这个过程中系统做了哪些工作呢？首先，系统会对于用户输入的关键词进行拼写纠错，将其修改为正确的关键词，并做一个大小写的归一化处理。之后使用分词器对其进行处理，对于分词的语义单元实现实体的标签识别，这里的作用就是抓住核心的东西来进行搜索，保证搜索结果既丰富又不会跑题。系统在最后还会进行同义词的拓展来扩大搜索结果的召回量。因此，系统在最后实际进行查询的词并不是用户所输入的“NKIE 新款运动鞋”，而是如图中所示的实际执行的查询词。这就是电商查询语义理解在实际业务中的作用。



## 核心功能亮点

本次 OpenSearch 所集成的电商查询语义理解功能具有以下的三个核心亮点。这里之所以强调电商领域，是因为 OpenSearch 在原本通用的查询语义理解基础上提供了针对于电商场景的纵深优化。

- **电商拼写纠错**: OpenSearch 系统会对于用户输入的查询串进行拼写错误检查, 计算查询词是否需要进行纠错, 并给出纠错建议。对于明确的拼写错误会直接改写原始查询串, 然后进行检索; 对于可能的拼写错误将仍然使用原始查询串进行检索。
- **电商命名实体识别 (NER)**: OpenSearch 系统将查询词分词后对每个语义单元进行需求识别, 每个实体会被打上一个类型标签 (共 36 种标签), 比如品牌、品类、新品等。在系统中, 标签重要性大致分为三档: 高、中、低。标签重要性低的实体在查询中会被忽略, 以扩大召回率; 标签重要性高的实体会直接影响文本相关性的计算和类目预测训练。比如“耐克修身连衣裙”, 实体识别的结果为“耐克 / 品牌 / 中”、“修身 / 款式元素 / 低”、“连衣裙 / 品类 / 高”。并且如果系统中默认的标签重要性不符合用户预期, 用户也可以直接进行调配。
- **灵活干预**: 电商场景中查询语义理解功能是基于淘宝搜索多年沉淀的大数据训练, 绝大多数情况下会为电商场景提供良好的查询分析效果, 但是不同的业务场景都会有自身垂直的查询词可能未覆盖或者分析错误, 所以开放搜索也支持可视化上传、管理干预词条, 支持自定义词库的上传, 支持灵活配置, 即干预即生效, 简单快捷, 方便用户在业务中更加灵活地使用这一功能。

## 新功能介绍-电商查询语义理解



### [ 核心功能亮点 ]



**电商拼写纠错**

系统会对用户输入的查询串进行拼写错误检查, 计算查询词是否需要进行纠错, 并给出纠错建议。对于明确的拼写错误会直接改写原始查询串, 然后进行检索; 对于可能的拼写错误将仍然使用原始查询串进行检索。  
例如: 查询词“连衣裙”, 经过拼写纠错会改写为“连衣裙”再进行检索。



**电商命名实体识别 (NER)**

系统将查询词分词后对每个语义单元进行需求识别, 每个实体会被打上一个类型标签 (共36种标签)。标签重要性低的实体在查询中可能会被忽略; 标签重要性高的实体会直接影响文本相关性的计算和类目预测训练。比如“耐克修身连衣裙”, 实体识别的结果为“耐克/品牌/中”“修身/款式元素/低”“连衣裙/品类/高”。




**灵活干预**

电商场景的查询语义理解功能是基于淘宝搜索多年沉淀的大数据训练, 绝大多数情况下会为电商场景提供良好的查询分析效果, 但是不同的业务场景都会有自身垂直的查询词可能未覆盖或者分析错误, 所以开放搜索也支持可视化上传、管理干预词条, 即干预即生效, 简单快捷。

为了无法计算的价值 |  阿里云

## 电商查询语义理解客户案例

这里为大家分享一个使用 OpenSearch 电商查询语义理解的实际客户案例，该客户的业务属于垂直领域的电商行业，搜索引导的成交占比全部成交的 60% 以上，因此搜索是站内最重要的功能。而用户在站内搜索的无结果率却接近 60%，也就是说每天海量搜索 PV 都是 0 转化，这是非常巨大的浪费。基于上述背景，该客户内部明确立项对于在两个月的时间内对于搜索效果进行优化，而其面对的最大挑战就是业务非常垂直并且圈子内用户输入的查询词和商品的官方名称往往千差万别。此外，该客户所售卖的商品很多来自海外，最开始名字为英文，而当引入国内之后，客户的用户有时会直接音译成为中文来称呼商品名称，所以站内的热搜词往往会商品的别名、缩写、音译词等。而且客户原本自建的开源服务没有对搜索关键词做智能的语义理解，甚至有些实体名词分词还是错误的，因此造成了搜索无结果率居高不下。

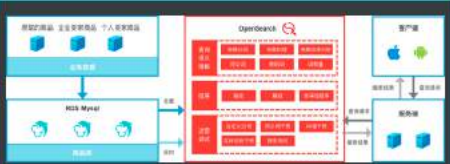


### 新功能介绍-电商查询语义理解的客户案例


#### 客户面临的问题和挑战

- 客户业务属于非常垂直的电商行业，搜索引导的成交占比全部成交的60%以上，是站内最重要的功能。由于站内用户的热搜词往往是商品的别名、缩写、音译词等等，这些查询词和商品官方的名称千差万别，客户原本自建的开源服务没有对搜索关键词做智能的语义理解，甚至有些实体名词分词还是错误的，这就导致站内**搜索的无结果率接近60%**，即每天海量搜索PV都是0转化，是非常巨大的浪费。
- 客户内部明确搜索效果优化周期需要在2个月完成，而技术人力只有1个。解决搜索结果召回问题的主要路径需要建立起搜索关键词的智能语义理解服务，而从0到1的实现该能力不仅需要多个精通NLP的算法专家，还需要至少几十万量级有人工标注的数据物料做训练，现阶段**没有人做数据，市面上也没有现成的开源模型**。

#### 使用OpenSearch的配置方案及效果



- OpenSearch的**搜索结果率**基本在90%左右，而自建服务的结果召回率只有31%左右，对比原先基于开源自建**提升了190%**。
- 搜索引导成交转化率**OpenSearch对比原先基于开源自建**提升了9%**。

为了无法计算的价值 |  阿里云

基于上述问题，客户方的技术人员开始调研解决方案。客户内部明确要求对于搜索效果的优化周期要在两个月之内完成，而技术人力却只有一个。解决搜索结果召回问题的主要路径就是建立起搜索关键词的智能语义理解服务，而从 0 到 1 的实现该能力不仅需要多个精通 NLP 的算法专家，还需要至少几十万量级有人工标注的

数据物料做训练，而现阶段，客户方面既没人也没数据，市面上也没有现成的开源插件。基于这样的效率和投入成本的考虑，客户就选择使用 OpenSearch 对于之前的搜索功能进行改造和升级。该客户使用了 OpenSearch 的电商分词、电商查询语义理解之外，还将一些自定义的补丁词库放到了 OpenSearch 上，而词库可以由运营同学直接上传，而不需要为技术人员增加额外的开发工作量。经过不到 2 个月的时间，全流程实现完成之后，使用 OpenSearch 的部分服务的结果召回率基本达到了 90% 左右，而自建服务的结果召回率只有 31% 左右，两者进行对比就能够发现使用 OpenSearch 在结果召回率方面具有较大的优势。除此之外，使用 OpenSearch 使得搜索引导成交转化率相比于原本的开源自建服务提升了 9%。

### 三、新功能介绍 – 搜索算法平台和人气模型

#### 搜索算法平台

之所以要实现搜索算法平台功能，是因为当搜索业务发展到一定阶段，用户行为数据对于搜索排序效果的提升将会承担重要的角色。受限于行为日志数据的庞大和繁杂，这个时候往往需要机器通过统计学习算法，对大量的历史数据进行学习从而生成经验模型。利用经验模型将行为数据量化为合理的分数，最终在排序中使用。想要在业务中高效实现这一整套机器学习流程，往往会面临以下两个问题：

- 繁杂的数据采集和预处理工作：每天亿级的数据都需要通过自动化的方式进行采集、存储和处理，对于这些原始数据还需要进行统计、分析和处理来发现用户行为的基本规律，这就要求开发者具备长分析能力和长期经验积累，比如发现原始数据正负不均衡时，如何构建出正负样本数均衡的样本数据，这些还需要一定的算法工作经验。
- 大海捞针般的算法调参：面对错综复杂的算法参数，开发者们往往需要花费非常多的时间进行调试，而这就如同大海捞针一般。当找到一个看似靠谱的参数组合之后，这个参数组合是否最优还会存在一定不确定性。

## 新功能介绍-搜索算法平台

当搜索业务发展有一定阶段，用户行为数据对于搜索排序效果的提升将会承担重要的角色。受限于行为日志数据的庞大和繁杂，往往需要机器通过统计学算法，对大量的历史数据进行学习从而生成经验模型，利用经验模型将行为数据量化成合理的分数，最终在排序中使用。想要在业务中高效实现这一整套机器学习流程，往往会面临：



I'm fine.

繁杂的数据采集和预处理工作



大海捞针般的算法调参

为了无法计算的价值 | 阿里云

基于以上两个令人极为头疼的问题，OpenSearch 通过一个高可视化的方式支持用户直接完成数据预处理、特征工程、算法调参、模型评估、模型管理模型上线等一系列繁杂的算法日常工作，并且还集成了阿里巴巴集团内部比较成熟的排序算法模型。用户可以通过整个算法平台自行训练高质量的算法排序模型并且应用到 OpenSearch 的算法排序中，之后将模型上线到实际业务并进行 AB Test 对比，得到整体实验效果并且回收，实现全链路 OpenSearch 的完全打通，并且以可视化、流程式地进行呈现。对于没有接触过这些算法的开发者而言，也可以直接上手这个 OpenSearch 平台。并且该平台专注于搜索算法模型训练，因此会内置一些阿里巴巴集团已经经过多年实战经验沉淀下来的模型，比如比较成熟的人气模型。并且在未来，类目预测等模型也会陆续上线到 OpenSearch 平台上。总之，用户可以在平台上完成整个算法模型训练以及上下游的流程。

## 新功能介绍-搜索算法平台

### [ 核心功能亮点 ]



#### 简单+灵活

模型训练的上下游链路全打通，以可模块化、流程式的方式呈现，无需理解复杂的算法逻辑，从未做过算法也可使用；

训练数据集、数据预处理、特征工程、算法参数等环节均可定制可迭代，告别全黑盒封闭的状态。



#### 高质量淘宝搜索模型

- 人气模型：已上线，淘宝搜索最基础的算法模型，用来衡量物品/文档的受欢迎程度。
- 类目预测模型：下个月即将上线算法平台，用来预测用户需求的类目指向，优化多类目指向关键词场景下结果的精准度。



#### 专业模型质量分析评估

- 模型评估报告：模型训练完成后都会产出评估报告，使用不同的算法生成的评估指标也不一样，比如使用分类算法，那么指标会包括AUC值、ROC曲线、混淆矩阵、特征权重等。
- AB Test：支持一键生成对比实验，快速验证模型对业务的影响。
- 业务指标分析报告：最专业的衡量搜索业务效果指标呈现。


为了无法计算的价值 | 阿里云

此外，对于上线的模型是否优秀而言，平台还会产出一个较为专业的评估报告，通过这些评估报告和指标使得模型的效果可以一目了然。对于模型上线之后用户是否买单而言，OpenSearch 还支持一键生成对比 AB Test 实验，能够支持用户快速迭代模型和改善效果。对于模型上线效果的回归而言，平台还能提供业务指标分析的报表。

## 人气模型

人气模型简单而言就是衡量用户对于站内每个商品或者每个文件的受欢迎程度，也就是静态质量分数。可以将其加入排序，使其成为排序的一个因子，通过这个因子来直接影响排序效果。OpenSearch 会将四个维度的特征加入训练，包括了实体维度、时间维度、行为维度以及统计维度。

## 新功能介绍-搜索算法平台-人气模型




### 是什么？

- 是淘宝搜索最基础的排序算法模型，量化每个商品的静态质量及受欢迎的程度的值称之为商品人气分。但其实这个模型对于其他的搜索场景也有**复用性**，在非商品搜索场景中通过人气模型也可以计算出doc的受欢迎程度。（离线计算类模型）

### 哪些特征参与训练？

- 实体维度：商品/doc、品牌、商家、叶子类目、一级类目等。
- 时间维度：1天、3天、7天、14天、30天、时间衰减加权等。
- 行为维度：曝光、点击、收藏、加购、购买、评论、点赞等。
- 统计维度：数量、人数、频率、点击率、转化率等。

每个特征从以上4个维度中各取一到两个进行组合，再从历史数据中统计该组合特征最终的特征值。比如，商品（实体）最近1天（时间）的曝光（行为）量（统计指标）；商品所在店铺（实体）最近30天（时间）的销量（行为+统计维度）等等。由以上方法产生的特征数量级相当于4个维度的笛卡尔积。

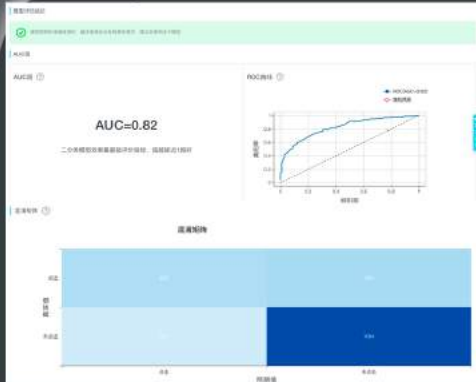


为了无法计算的价值 | 阿里云

## 模型评估报告


OpenSearch 会针对每个模型在训练完成后自动产生一个模型评估报告，通过这个质量来评估模型的效果，比如分类算法模型的评估报告中就会包括以下指标，比如整个模型是否适合在业务中使用，此外还有一些细化的指标如 AUC 值、ROC 曲线、混淆矩阵、特征权重等。对于一些有算法经验的同学也可以直接通过这些指标值来发现问题，并进行模型的下一轮调优。

## 新功能介绍-搜索算法平台-模型评估报告



模型训练完成后都会产出模型的效果评估报告，通过指标来评估模型的效果，来决策下一步继续迭代还是部署使用。分类算法报告中会包含以下指标：

- **AUC值**：Area Under Curve，ROC曲线下与坐标轴围成的面积，通常在0.5-1之间，数值越大效果越好。
- **ROC曲线**：（receiver operating characteristic curve）根据一系列不同的二分类方式（分界值或决定阈），以真阳性率（灵敏度）为纵坐标，假阳性率（1-特异度）为横坐标绘制的曲线。
- **混淆矩阵**：每一列代表了预测类别，每一行的总数表示预测为该类别的数据的数目；每一行代表了数据的真实归属类别，每一行的数据总数表示该类别的数据实例的数目。
- **特征权重**：每个特征对于模型的效果产生的实际的影响量化值。



为了无法计算的价值 | 阿里云

## 如何使用

如果想要使用搜索算法平台，首先需要通过 OpenSearch 的 SDK 上传行为数据，之后就可以直接在算法平台上创建和训练模型，模型训练完成之后就可以查看模型评估报告。如果模型评估报告建议使用该模型，就可以直接将模型部署到 OpenSearch 的应用中，进而将其应用到排序表达式中。



## 搜索业务效果评估

将模型应用到排序表达式中之后就需要对于搜索业务效果进行评估了，传统方式往往是结合线下人工评估和线上流量测试。线下人工评估往往会抽取一些具有代表性的查询关键词，组成一个规模适当的关键词集合。针对这个关键词集合，从排序模型的产出结果中查询对应的结果，进行人工标注相关性。对人工评测的结果数据，使用预定义好的评价计算公式比如 DCG 等，用数值化的方法来评价模型的结果和标注的理想结果的接近程度。如果人工评估认为搜索业务的效果是变好的，此时就可以进行线上流量测试。为了真实验证一个排序模型的好坏，就需要使用刚才提到的 AB Test 机制来检验和对比。在用户搜索时，由测试机制根据一定的策略来自动决定用户的分组号，保证自动抽取导入不同分组的流量具有可对比性，然后让不同分组的用户看到

的不同排序模型所提供的结果，用户在不同模型下的行为将被记录下来，这些行为数据通过数据分析形成一系列指标，通过比较，最后就形成了不同模型之间孰优孰劣的结论。

### 新功能介绍-搜索算法平台-搜索业务效果评估



#### 线下人工评估

人工抽取具有代表性的查询关键词，组成一个规模适当的关键词集合。针对这个关键词集合，从排序模型的产出结果中查询对应的结果，进行人工标注相关性（从广义的层面，相关性可以理解为用户查询的综合满意度）。对人工评测的结果数据，使用预定义好的评价计算公式比如DCG等，用数值化的方法来评价排序模型的结果和标注的理想结果的接近程度。



#### 线上流量测试

为了真实验证一个排序模型的好坏，需要有AB Test机制来检验和对比，在用户搜索时，由机制根据一定的策略来自动决定用户的分组号，保证自动抽取导入不同分组的流量具有可对比性，然后让不同分组的用户看到的不同排序模型提供的结果。用户在不同模型下的行为将被记录下来，这些行为数据通过数据分析形成一系列指标，通过比较，最后就形成了不同模型之间孰优孰劣的结论。

为了无法计算的价值 | 阿里云

对于 OpenSearch 而言，其无法支持第一种线下人工评估，因此用户只能自己抽取 Query 来进行人工评估。当人工评估完成之后，OpenSearch 可以支持线上的 AB Test 端到端一键配置，可以支持多场景多分组下流量划分进行试验，充分满足了单一应用但覆盖多样业务场景的实验需求。并且使得实验配置和上下线状态可以灵活调整，即操作即生效。并且可以实现报表的天级产出，帮助客户进行决策。并且能够实现全流程可视化界面操作，能够使得用户实现快速接入，简单无门槛，并且产品和运营人员也可以直接使用。

### 新功能介绍-搜索算法平台-搜索业务效果评估

AB Test

新建AB Test

查看AB Test

商品列表

1. 基础版-人气分析算法效果评估

2. 进阶版-人气分析算法效果评估

实验名称	实验状态	实验配置	实验数据	更新时间	操作
1. 基础版-人气分析算法效果评估	进行中	搜索算法: 1.0.0.0	10%	2019-03-04 12:04:31	编辑 删除 下线
2. 进阶版-人气分析算法效果评估	进行中	搜索算法: 1.0.0.0	10%	2019-03-04 12:04:31	编辑 删除 下线

新建实验

端到端的将AB Test流程实现平台化

- 支持多场景多分箱下流量划分进行实验, 充分满足单一应用但覆盖多样业务场景的实验需求。
- 实验配置和上下线状态灵活可调整, 即操作即生效。
- 报表天级产出。
- 全流程可视化界面操作, 快速接入, 简单无门槛, 产品和运营人员可自行使用, 解放技术精力。

为了无法计算的价值 | 阿里云

当整个优化完成之后, OpenSearch 还提供了一套专业搜索业务质量报表指标体系, 该体系针对于电商场景划分了 5 个体系, 即流量指标、点击指标、用户分析指标、Query 分析指标以及成交指标。

### 新功能介绍-搜索算法平台-搜索业务效果评估

流量指标

点击指标

用户分析

Query分析

成交指标

流量指标

UV: 86,172

人均PV: 7.4

IPV: 206,490

IPV-UV: 22,264

CTR: 0.324

无结果率: 6.251

流量指标

7日: +3%

30日: +4%

7日: -6%

30日: +0.8%

7日: +3%

30日: +1%

7日: +0.1%

30日: +1.2%

7日: -

30日: +1%

流量指标

2019-03-04

2019-03-05

2019-03-06

2019-03-07

2019-03-08

2019-03-09

2019-03-10

2019-03-11

2019-03-12

2019-03-13

2019-03-14

2019-03-15

2019-03-16

2019-03-17

2019-03-18

2019-03-19

2019-03-20

2019-03-21

2019-03-22

2019-03-23

2019-03-24

2019-03-25

2019-03-26

2019-03-27

2019-03-28

2019-03-29

2019-03-30

2019-03-31

2019-04-01

2019-04-02

2019-04-03

2019-04-04

2019-04-05

2019-04-06

2019-04-07

2019-04-08

2019-04-09

2019-04-10

2019-04-11

2019-04-12

2019-04-13

2019-04-14

2019-04-15

2019-04-16

2019-04-17

2019-04-18

2019-04-19

2019-04-20

2019-04-21

2019-04-22

2019-04-23

2019-04-24

2019-04-25

2019-04-26

2019-04-27

2019-04-28

2019-04-29

2019-04-30

2019-05-01

2019-05-02

2019-05-03

2019-05-04

2019-05-05

2019-05-06

2019-05-07

2019-05-08

2019-05-09

2019-05-10

2019-05-11

2019-05-12

2019-05-13

2019-05-14

2019-05-15

2019-05-16

2019-05-17

2019-05-18

2019-05-19

2019-05-20

2019-05-21

2019-05-22

2019-05-23

2019-05-24

2019-05-25

2019-05-26

2019-05-27

2019-05-28

2019-05-29

2019-05-30

2019-05-31

2019-06-01

2019-06-02

2019-06-03

2019-06-04

2019-06-05

2019-06-06

2019-06-07

2019-06-08

2019-06-09

2019-06-10

2019-06-11

2019-06-12

2019-06-13

2019-06-14

2019-06-15

2019-06-16

2019-06-17

2019-06-18

2019-06-19

2019-06-20

2019-06-21

2019-06-22

2019-06-23

2019-06-24

2019-06-25

2019-06-26

2019-06-27

2019-06-28

2019-06-29

2019-06-30

2019-07-01

2019-07-02

2019-07-03

2019-07-04

2019-07-05

2019-07-06

2019-07-07

2019-07-08

2019-07-09

2019-07-10

2019-07-11

2019-07-12

2019-07-13

2019-07-14

2019-07-15

2019-07-16

2019-07-17

2019-07-18

2019-07-19

2019-07-20

2019-07-21

2019-07-22

2019-07-23

2019-07-24

2019-07-25

2019-07-26

2019-07-27

2019-07-28

2019-07-29

2019-07-30

2019-07-31

2019-08-01

2019-08-02

2019-08-03

2019-08-04

2019-08-05

2019-08-06

2019-08-07

2019-08-08

2019-08-09

2019-08-10

2019-08-11

2019-08-12

2019-08-13

2019-08-14

2019-08-15

2019-08-16

2019-08-17

2019-08-18

2019-08-19

2019-08-20

2019-08-21

2019-08-22

2019-08-23

2019-08-24

2019-08-25

2019-08-26

2019-08-27

2019-08-28

2019-08-29

2019-08-30

2019-08-31

2019-09-01

2019-09-02

2019-09-03

2019-09-04

2019-09-05

2019-09-06

2019-09-07

2019-09-08

2019-09-09

2019-09-10

2019-09-11

2019-09-12

2019-09-13

2019-09-14

2019-09-15

2019-09-16

2019-09-17

2019-09-18

2019-09-19

2019-09-20

2019-09-21

2019-09-22

2019-09-23

2019-09-24

2019-09-25

2019-09-26

2019-09-27

2019-09-28

2019-09-29

2019-09-30

2019-10-01

2019-10-02

2019-10-03

2019-10-04

2019-10-05

2019-10-06

2019-10-07

2019-10-08

2019-10-09

2019-10-10

2019-10-11

2019-10-12

2019-10-13

2019-10-14

2019-10-15

2019-10-16

2019-10-17

2019-10-18

2019-10-19

2019-10-20

2019-10-21

2019-10-22

2019-10-23

2019-10-24

2019-10-25

2019-10-26

2019-10-27

2019-10-28

2019-10-29

2019-10-30

2019-10-31

2019-11-01

2019-11-02

2019-11-03

2019-11-04

2019-11-05

2019-11-06

2019-11-07

2019-11-08

2019-11-09

2019-11-10

2019-11-11

2019-11-12

2019-11-13

2019-11-14

2019-11-15

2019-11-16

2019-11-17

2019-11-18

2019-11-19

2019-11-20

2019-11-21

2019-11-22

2019-11-23

2019-11-24

2019-11-25

2019-11-26

2019-11-27

2019-11-28

2019-11-29

2019-11-30

2019-12-01

2019-12-02

2019-12-03

2019-12-04

2019-12-05

2019-12-06

2019-12-07

2019-12-08

2019-12-09

2019-12-10

2019-12-11

2019-12-12

2019-12-13

2019-12-14

2019-12-15

2019-12-16

2019-12-17

2019-12-18

2019-12-19

2019-12-20

2019-12-21

2019-12-22

2019-12-23

2019-12-24

2019-12-25

2019-12-26

2019-12-27

2019-12-28

2019-12-29

2019-12-30

2019-12-31

2020-01-01

2020-01-02

2020-01-03

2020-01-04

2020-01-05

2020-01-06

2020-01-07

2020-01-08

2020-01-09

2020-01-10

2020-01-11

2020-01-12

2020-01-13

2020-01-14

2020-01-15

2020-01-16

2020-01-17

2020-01-18

2020-01-19

2020-01-20

2020-01-21

2020-01-22

2020-01-23

2020-01-24

2020-01-25

2020-01-26

2020-01-27

2020-01-28

2020-01-29

2020-01-30

2020-01-31

2020-02-01

2020-02-02

2020-02-03

2020-02-04

2020-02-05

2020-02-06

2020-02-07

2020-02-08

2020-02-09

2020-02-10

2020-02-11

2020-02-12

2020-02-13

2020-02-14

2020-02-15

2020-02-16

2020-02-17

2020-02-18

2020-02-19

2020-02-20

2020-02-21

2020-02-22

2020-02-23

2020-02-24

2020-02-25

2020-02-26

2020-02-27

2020-02-28

2020-02-29

2020-03-01

2020-03-02

2020-03-03

2020-03-04

2020-03-05

2020-03-06

2020-03-07

2020-03-08

2020-03-09

2020-03-10

2020-03-11

2020-03-12

2020-03-13

2020-03-14

2020-03-15

2020-03-16

2020-03-17

2020-03-18

2020-03-19

2020-03-20

2020-03-21

2020-03-22

2020-03-23

2020-03-24

2020-03-25

2020-03-26

2020-03-27

2020-03-28

2020-03-29

2020-03-30

2020-03-31

2020-04-01

2020-04-02

2020-04-03

2020-04-04

2020-04-05

2020-04-06

2020-04-07

2020-04-08

2020-04-09

2020-04-10

2020-04-11

2020-04-12

2020-04-13

2020-04-14

2020-04-15

2020-04-16

2020-04-17

2020-04-18

2020-04-19

2020-04-20

2020-04-21

2020-04-22

2020-04-23

2020-04-24

2020-04-25

2020-04-26

2020-04-27

2020-04-28

2020-04-29

2020-04-30

2020-05-01

2020-05-02

2020-05-03

2020-05-04

2020-05-05

2020-05-06

2020-05-07

2020-05-08

2020-05-09

2020-05-10

2020-05-11

2020-05-12

2020-05-13

2020-05-14

2020-05-15

2020-05-16

2020-05-17

2020-05-18

2020-05-19

2020-05-20

2020-05-21

2020-05-22

2020-05-23

2020-05-24

2020-05-25

2020-05-26

2020-05-27

2020-05-28

2020-05-29

2020-05-30

2020-05-31

2020-06-01

2020-06-02

2020-06-03

2020-06-04

2020-06-05

2020-06-06

2020-06-07

2020-06-08

2020-06-09

2020-06-10

2020-06-11

2020-06-12

2020-06-13

2020-06-14

2020-06-15

2020-06-16

2020-06-17

2020-06-18

2020-06-19

2020-06-20

2020-06-21

2020-06-22

2020-06-23

2020-06-24

2020-06-25

2020-06-26

2020-06-27

2020-06-28

2020-06-29

2020-06-30

2020-07-01

2020-07-02

2020-07-03

2020-07-04

2020-07-05

2020-07-06

2020-07-07

2020-07-08

2020-07-09

2020-07-10

2020-07-11

2020-07-12

2020-07-13

2020-07-14

2020-07-15

2020-07-16

2020-07-17

2020-07-18

2020-07-19

2020-07-20

2020-07-21

2020-07-22

2020-07-23

2020-07-24

2020-07-25

2020-07-26

2020-07-27

2020-07-28

2020-07-29

2020-07-30

2020-07-31

2020-08-01

2020-08-02

2020-08-03

2020-08-04

2020-08-05

2020-08-06

2020-08-07

2020-08-08

2020-08-09

2020-08-10

2020-08-11

2020-08-12

2020-08-13

2020-08-14

2020-08-15

2020-08-16

2020-08-17

2020-08-18

2020-08-19

2020-08-20

2020-08-21

2020-08-22

2020-08-23

2020-08-24

2020-08-25

2020-08-26

2020-08-27

2020-08-28

2020-08-29

2020-08-30

2020-08-31

2020-09-01

2020-09-02

2020-09-03

2020-09-04

2020-09-05

2020-09-06

2020-09-07

2020-09-08

2020-09-09

2020-09-10

2020-09-11

2020-09-12

2020-09-13

2020-09-14

2020-09-15

2020-09-16

2020-09-17

2020-09-18

2020-09-19

2020-09-20

2020-09-21

2020-09-22

2020-09-23

2020-09-24

2020-09-25

2020-09-26

2020-09-27

2020-09-28

2020-09-29

2020-09-30

2020-10-01

2020-10-02

2020-10-03

2020-10-04

2020-10-05

2020-10-06

2020-10-07

2020-10-08

2020-10-09

2020-10-10

2020-10-11

2020-10-12

2020-10-13

2020-10-14

2020-10-15

2020-10-16

2020-10-17

2020-10-18

2020-10-19

2020-10-20

2020-10-21

2020-10-22

2020-10-23

2020-10-24

2020-10-25

2020-10-26

2020-10-27

2020-10-28

2020-10-29

2020-10-30

2020-10-31

2020-11-01

2020-11-02

2020-11-03

2020-11-04

2020-11-05

2020-11-06

2020-11-07

2020-11-08

2020-11-09

2020-11-10

2020-11-11

2020-11-12

2020-11-13

2020-11-14

2020-11-15

2020-11-16

2020-11-17

2020-11-18

2020-11-19

2020-11-20

2020-11-21

2020-11-22

2020-11-23

2020-11-24

2020-11-25

2020-11-26

2020-11-27

2020-11-28

2020-11-29

2020-11-30

2020-12-01

2020-12-02

2020-12-03

2020-12-04

2020-12-05

2020-12-06

2020-12-07

2020-12-08

2020-12-09

2020-12-10

2020-12-11

2020-12-12

2020-12-13

2020-12-14

2020-12-15

2020-12-16

2020-12-17

2020-12-18

2020-12-19

2020-12-20

2020-12-21

2020-12-22

2020-12-23

2020-12-24

2020-12-25

2020-12-26

2020-12-27

2020-12-28

2020-12-29

2020-12-30

2020-12-31

专业搜索业务质量报表指标体系

- 流量指标: PV、UV、曝光次数、人均搜索PV、Query数、人均Query数、独立Query数、商品数/店铺数
- 点击指标: IPV、IPV-UV、人均IPV、浏览-点击、曝光-点击、CTR、无结果率、Top3结果CTR、Top5结果CTR、点击商品数/店铺数
- 用户分析指标: 搜索次日留存率、7日留存率、流失率
- Query分析指标: 高频词列表(含对应PV、UV、CTR)、低频词列表(含对应PV、UV、CTR)、无结果词列表(含对应PV、UV)、少结果词列表(含对应PV、UV、CTR)
- 成交指标(电商场景): 搜索引导GMV、成交UV、成交转化率、点击-成交

为了无法计算的价值 | 阿里云

借助于这些效果评估指标, 客户可以非常专业并且直接地看到经过自己的一轮又一轮的优化, 搜索效果的提升能够体现在哪里以及存在的问题在哪里。

# 飞天大数据平台 Elasticsearch 最新特性



钱雨欣

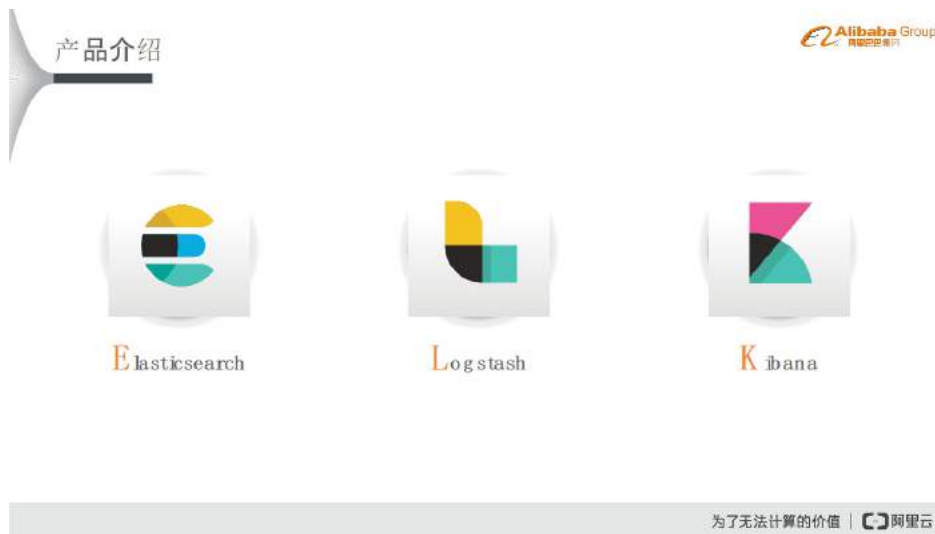
阿里巴巴搜索推荐  
事业部产品经理

摘要：阿里云 Elasticsearch 提供 100% 兼容开源 Elasticsearch 的功能，以及 Security、Machine Learning、Graph、APM 等商业功能，致力于数据分析、数据搜索等场景服务。与开源社区背后商业公司 Elastic 战略合作，为客户提供企业级权限管控、安全监控告警、自动报表生成等场景服务。本文中，阿里云产品专家沐泽为大家介绍了阿里云 Elasticsearch 产品的相关情况。

---

## 产品介绍

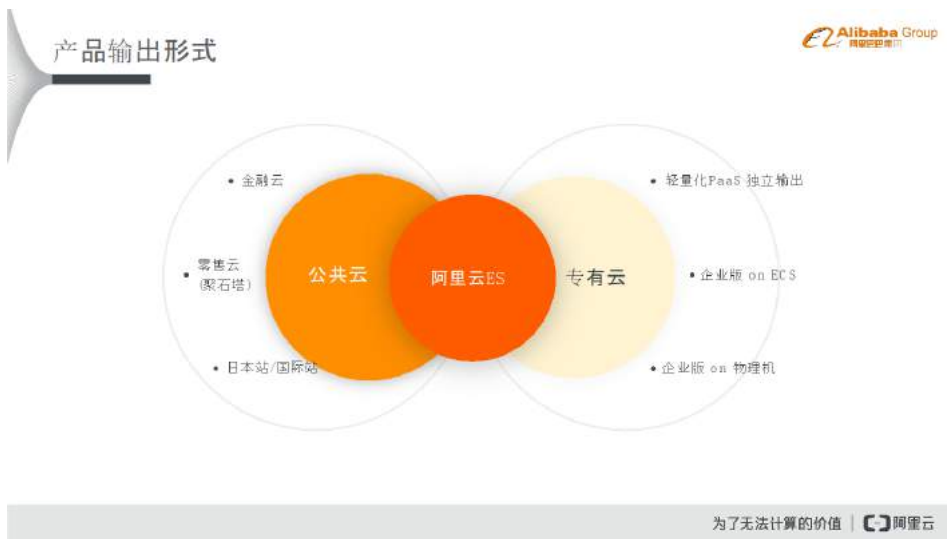
Elasticsearch (简称 ES) 是 2010 年推出的一款开源产品，本质上是一个实时的分布式实时搜索与分析引擎。随着这些年来 Elasticsearch 生态的演进，逐渐发展成为 ELK 即 Elasticsearch、Logstash、Kibana 的生态。Elasticsearch 属于搜索引擎，Logstash 负责数据的采集、转化以及输出，Kibana 则提供了强大的数据可视化功能。对于 Elasticsearch 而言，其在 DB-Engines 中的开源数据库排行榜中位列第一。可以看出，Elasticsearch 受到了广泛的认可，并且也有大量的开发者正在使用。



阿里云 Elasticsearch 提供了全托管的 Elasticsearch 服务，并且 100% 兼容开源版本，并且对于内核进行了针对性优化，提供了商业功能（原 ‘X-Pack’），即开即用，高可用服务，弹性伸缩，按需付费。在下图中，在阿里云 Elasticsearch 的可靠性、安全性、系统托管等方面与友商的产品进行了对比。在可靠性方面，阿里云 Elasticsearch 具有 99.9% 的数据可靠性，并且会定时地向 OSS 进行数据备份，方便用户在数据出现问题的时候进行恢复。此外，通过同城多活，提供了较强的容灾能力。在开源差异部分，阿里云 Elasticsearch 也做了大量的工作。在内核性能优化部分，不仅做了存储与计算分离，还提供了 ECS 本身的调优。在 Index Build 服务部分，Elasticsearch 本身支持高并发的数据写入加速，这样会使得数据的写入和查询相互影响，阿里云 Elasticsearch 则通过 Index Build 服务离线地构建索引，并将原生索引切换成比较小的分片并与线上索引进行 Merge，这样就避免了用户线上集群产生较大的 I/O 开销，在一定场景下保障了高并发写入场景下的查询服务的稳定性。在智能运维方面，阿里云 Elasticsearch 提供了 EYou 智能运维系统，能够帮助用户运维和监控集群并且进行智能分析，方便用户更好地了解集群的健康状况，并且还提供了预警以及改进建议等功能。此外，阿里云 Elasticsearch 近期还集成了阿里达摩院的 NLP 分词器和分析器，能够更好地完成业务的分析和检索任务。在商业

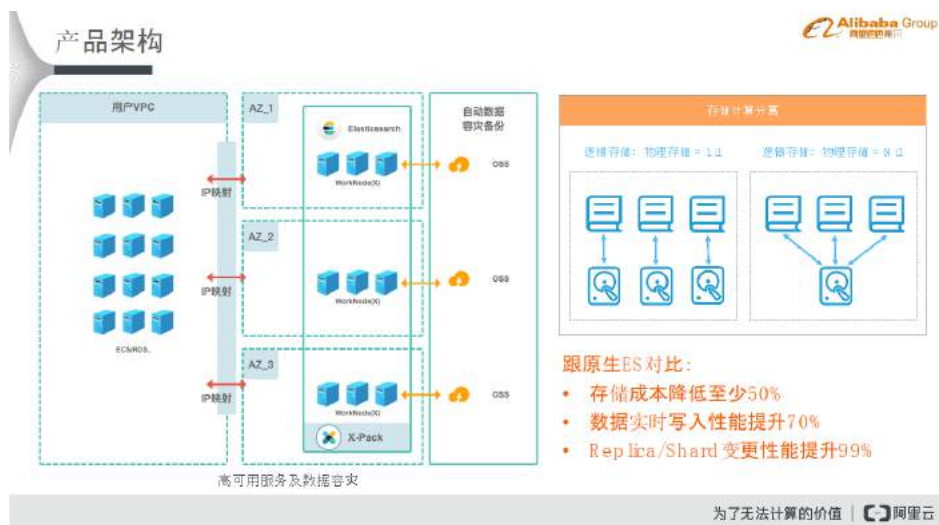


方面。在公共云上，阿里云 Elasticsearch 支持了金融云、零售云以及菜鸟云，并且在日本站和国际站进行了售卖。在专有云方面，8月底的时候阿里云 Elasticsearch 也提供了轻量 PaaS 独立输出，并且可以在企业版 On ECS 和企业版 On 物理机上部署。



## 产品架构

在产品架构部分，阿里云 Elasticsearch 部署在 ECS 网段，相当于购买了大量的 ECS 服务器拉起了 ES 镜像。对用户而言，可以购买很多的 ES 集群，每个 ES 集群中都会有很多的 Node，每个 Node 就是一台 ECS。整个 ECS 部署在系统方 VPC 内，并且支持跨可用区的同城容灾能力，也就是说在同一个区域下面，可以在不同的可用区内部署服务，通过阿里云 VPC 和用户 VPC 之间的 IP 映射使得每个集群的 Node 分布在不同的可用区之内。



除此之外，在保证容灾方面，用户的数据节点会定时地向 OSS 做快照备份，当用户的数据出现问题的时候，可以快速通过 OSS 实现数据恢复。整体的数据存储类型支持高效云盘、SSD 云盘以及本地磁盘。在存储计算分离方面，阿里云 Elasticsearch 近期也在内核方面进行了优化。本身 Elasticsearch 索引为了方便存储需要做分片，为了提升查询效率，每个分片会有多个副本，但是这样属于用空间换时间的方式，因此会造成大量的数据冗余，为用户带来很高的存储成本。另外一方面，为了提升查询效率，用户在写入数据的时候，就会增加更多的内存开销，进而造成写入速度较慢。在这样的背景之下，阿里云 Elasticsearch 做了存储与计算分离的内核优化，将用户数据的多个副本进行分片映射到同一块的物理介质之上，与原生的 ES 相比，阿里云 Elasticsearch 的存储成本降低至少 50%，数据写入实时性能提升 70%，Replica/Shard 变更性能提升 99%，以上这些能力都是开源版本的 ES 所不具备的。

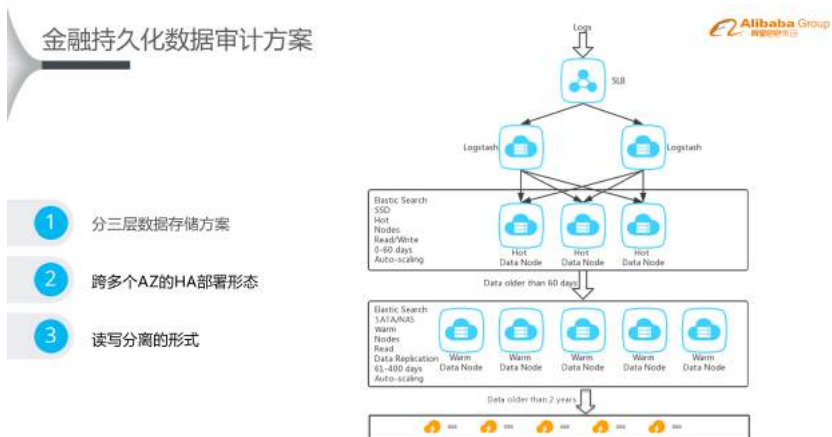
## 公共云可售卖区域

目前，阿里云 Elasticsearch 除了美东、英国和迪拜三个区域还没有部署售卖之外，在全球范围内的其他阿里云数据中心都已经部署售卖了，未来也会在更多的区域进行开放。



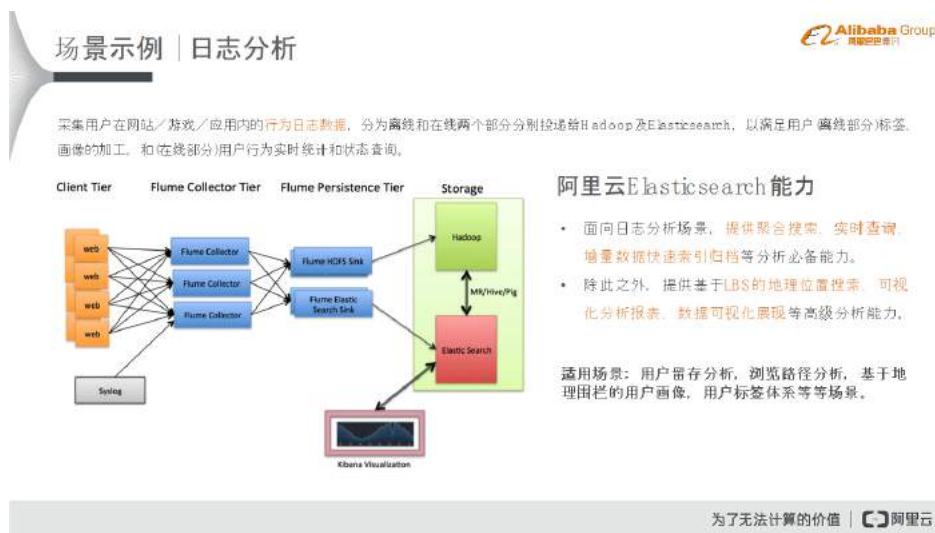
## 金融持久化数据审计方案

这里介绍一个实际案例，这是阿里云为一家信用卡结算公司设计的金融持久化数据库审计方案。该客户存在金融数据监管需求，因此数据需要存储的时间较长，因此造成数据量非常大。因此，阿里云提供了金融持久化数据库审计方案中为用户提供了一个三层的数据存储方案，用户近期的 Hot Data 会在第一层 ECS 存储大约 2 个月的时间，当变成 Warm Data 或者更老的数据之后，就会存储到下层 ECS 或者 OSS 之上，这样一方面保障了用户在使用阿里云 Elasticsearch 时的数据查询时效性，使得其不会被大数据量所影响，另外一方面也大大降低了用户的存储成本。



## 场景示例 – 日志分析

这里针对于日志分析场景进行更进一步的介绍。在日志分析方面，会采集用户在网站 / 游戏 / 应用内的行为日志数据，分为离线和在线两个部分分别投递给 Hadoop 及 Elasticsearch，以满足用户（离线部分）标签、画像的加工，和（在线部分）用户行为实时统计和状态查询。阿里云 Elasticsearch 在日志分析场景下提供了很多对应的能力，面向日志分析场景，提供聚合搜索、实时查询、增量数据快速索引归档等分析必备能力。除此之外，阿里云 Elasticsearch 基于 X-Pack 服务提供了基于 LBS 的地理位置搜索、可视化分析报表、数据可视化展现等高级分析能力。进一步可以实现用户留存分析，浏览路径分析，基于地理围栏的用户画像，用户标签体系等数据查询、统计以及分析场景。



## Elasticsearch 如何处理日志

日志数据的来源有很多，比如日志文件、数据库、传感器以及 Web API 等，而利用这些日志数据实现日志搜索和日志分析会存在很多常见的需求，主要包括五点，即集中收集与存储、日志搜索、分析聚合及可视化、安全、角色管理以及可伸缩性。

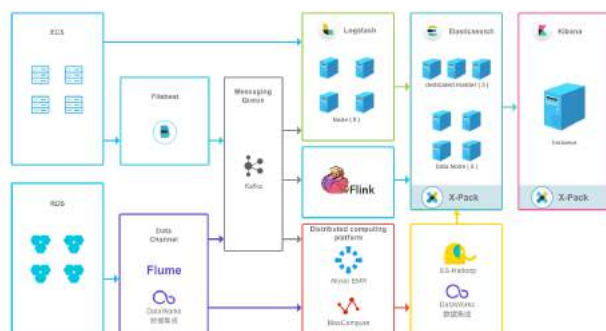


- 在集中收集与存储日志数据方面，阿里云 Elasticsearch 会对于常规日志数据进行采集，包括日志文件、日志系统、网络拥堵等其他常见的日志数据。阿里云 Elasticsearch 通过收集和汇拢数据以及离线 Hadoop 数据迁移能够比较快捷地集中日志数据并存储到 Elasticsearch 中构建索引。
- 在日志搜索方面，阿里云 Elasticsearch 能够支持全文检索、元数据搜索、指标 / 标签搜索以及地理位置搜索等。
- 在分析聚合和可视化方面，当数据聚合到阿里云 Elasticsearch 里面去之后，可以通过 sum、average、min/max 等聚合函数实现聚合分析，并且可以通过 X-Pack 实现机器学习分析，也可以借助 Kibana 实现在线数据可视化。并且在阿里云 Elasticsearch 中，用户可以直接通过 Kibana 控制台实现配置以及可视化面板的创建。
- 在安全和角色管理方面，阿里云 Elasticsearch 提供了 RBAC 的用户权限以及 TLS/SSL 交互式安全协议，并且能够实现实时监控和触发告警，能够帮助用户进行实时预防。此外，基于 X-Pack 的商业功能能够提供自动数据报表以及触发式报表等服务，帮助用户更好地进行数据管理和查询。
- 在可伸缩性方面，阿里云 Elasticsearch 能够支持弹性扩容，因为 ES 的节点是对等的，因此能够实现快速拷贝和弹性扩容，实现不同规模下的数据管理。

## 阿里巴巴 Elasticsearch 的产品生态架构

数据会从 RDS 等数据存储中过来，通过 Flume、EMR、MaxCompute 等下游计算引擎进行加工和处理，完成画像或者标签的工作，最后索引到 Elasticsearch 中去。阿里云 Elasticsearch 是兼容整个大数据生态的，并且也能够无缝地对接整个阿里云的产品生态，进而非常方便地完成数据的处理工作。此外，还可以通过 Kibana 可以帮助用户更好地实现数据的可视化管理。

阿里云Elasticsearch的产品生态架构



- 完全兼容开源大数据生态—用户放心
- 无缝对接阿里云产品生态—用户方便

# 飞天大数据平台智能推荐 AIRec 最新特性



闫勳勉

阿里云智能计算平台  
事业部高级解决方案  
工程师

## 一、智能推荐 (AIRec) 简介

### AIRec 背景介绍

智能推荐 (简称 AIRec) 基于阿里巴巴集团领先的大数据和人工智能技术, 结合阿里巴巴在电商、内容、新闻、视频直播和社交等多个行业领域的积累, 为全球企业及开发者提供云推荐服务。AIRec 能够有效解决用户偏好和物品之间的曝光选择问题。AIRec 产品与淘宝、天猫推荐的底层是同根共源的, 是淘宝、天猫推荐技术的首次对外输出。目前对于电商而言, 获客等成本非常高, AIRec 也希望能够解决以上这些问题, 帮助用户去深度运营好每一个流量, 将每一个流量的价值都充分地挖掘出来。因此 AIRec 在促活、留存、收入等整个流量生命周期中都可以提现它的价值。

### AIRec - 背景介绍

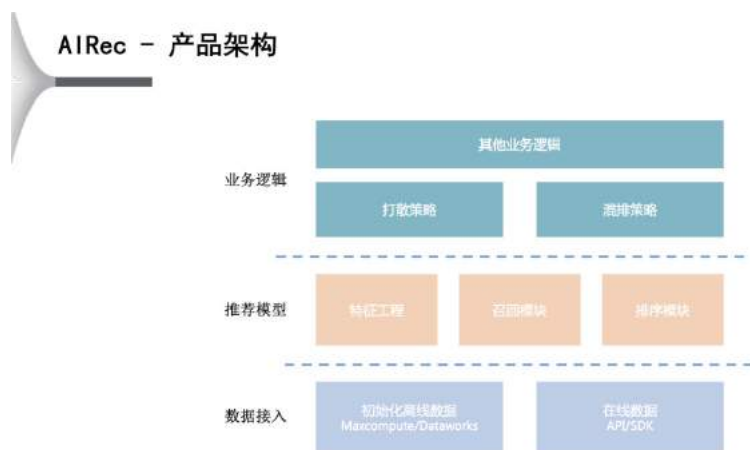
#### AIRec产品背景：

智能推荐 (简称AIRec) 基于阿里巴巴集团领先的大数据和人工智能技术, 结合阿里巴巴在电商、内容、新闻、视频直播和社交等多个行业领域的积累, 为全球企业及开发者提供云推荐服务。有效解决用户偏好和物品之间的曝光选择问题。



## AIRec 产品架构

阿里云智能推荐 AIRec 产品的实现方式整体上大致分为三层。其中最底层是数据接入层，对于这一层而言，一般需要客户提供两种数据，初始化离线数据和在线数据。初始化离线数据将用于整个推荐实例的启动，而在线数据则用于当服务启动之后在后续实时收集用户的行为，以及更新和回传。在数据接入层之上是推荐模型层，这里主要分为三部分，第一部分是特征工程，主要是对于用户的特征、商品的特征进行一定的挖掘；第二部分是召回模块；第三部分是排序模块。在推荐模型之上是业务逻辑层，各个应用厂家或者应用的场景不同，业务逻辑也会比较复杂，推荐系统往往无法覆盖所有的业务场景，AIRec 提供便捷的打散、混排等功能来满足尽可能多的场景需求。在此基础之上，业务方也可以根据 AIRec 推荐的结果来进行第二次封装来满足业务上更为复杂的需求。

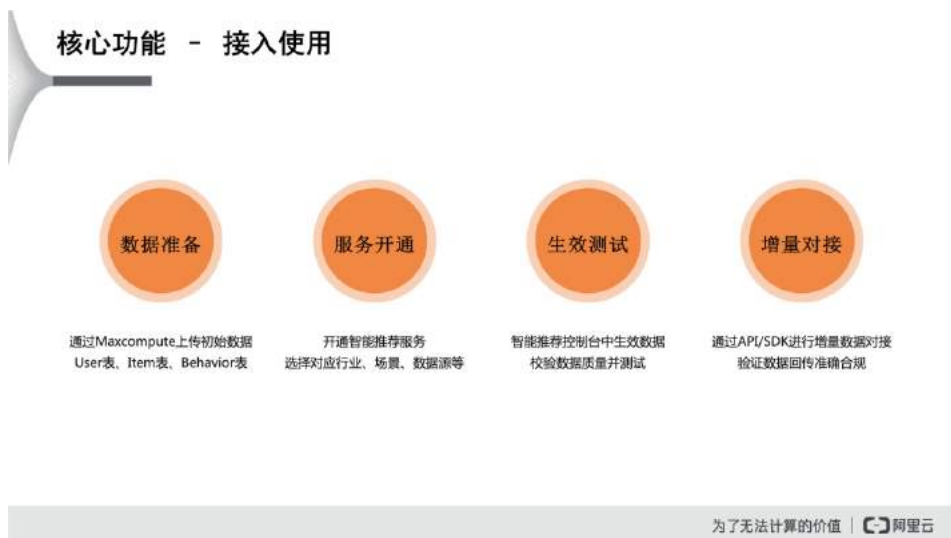


## 二、核心功能介绍

### AIRec 核心功能 - 接入使用

整个 AIRec 接入使用主要分为四个部分：数据准备、服务开通、生效测试和增

量对接。在数据准备方面，目前 AIRec 可以通过 MaxCompute 进行初始数据上传，MaxCompute 可以提供海量 PB 级别的数仓及数据分析处理能力。AIRec 初始时需要三张表，分别为 User 表、Item 表和 Behavior 表，User 表存储了用户的基本信息、用户 ID 以及是否给这个用户进行推荐以及一些用户特征维度的信息。针对于表中的这些字段，客户可以根据自己的数据埋点以及数据收集情况酌情地进行上传，并且表中的字段也是分为必填、选填和推荐等类别。当然，这些数据如果填写的越饱满，对于后期的效果调优而言，就会越有优势。同样的，Item 表存储了物品信息，Behavior 表则存储了用户与物品之间的关联信息，数据中的点击和展示是必须要上传的。在数据准备完成之后就可以开通 AIRec 服务了，开通完成之后可以进行对于行业、场景以及数据源的选择。选择完成之后，就可以在数据生效的页面检查数据生效的质量，同时在数据生效之后进行数据的测试。当测试完成之后就可以确认 AIRec 实例已经正常地运行起来了，在此基础之上可以对接服务端的 SDK 或者 API 实现增量数据的回传。



## AIRec 核心功能 - 行业

阿里云智能推荐 AIRec 现在向外输出的行业主要包括四种，即电商行业、新闻

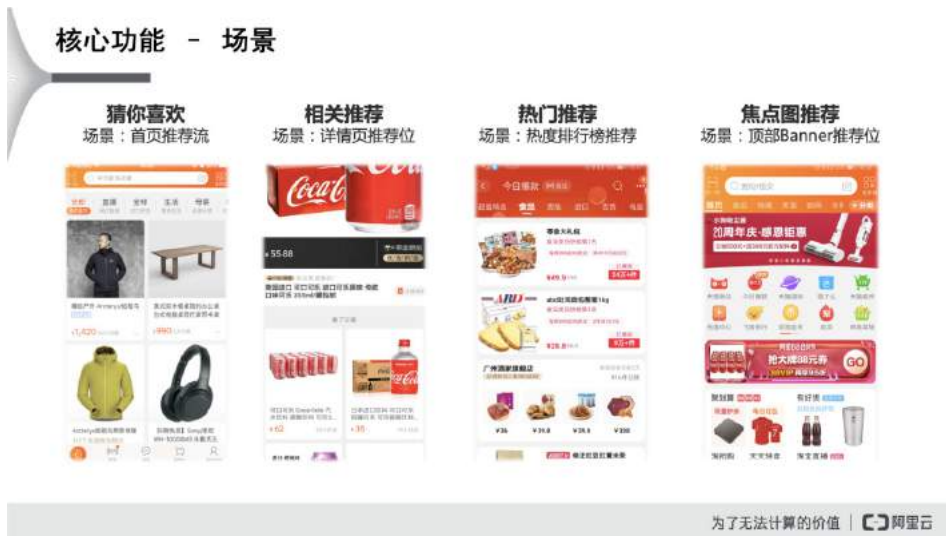
行业、内容行业以及视频行业。之所以对于行业进行了划分，这是因为目标行业的特点不同会导致用户的关注点不同，这也就导致了运营人员所关注的指标也各不同，因此这些因素最终也会影响推荐系统对于模型的选用。举例而言，对比一下电商型和内容型行业，前者比较关注点击率、GMV 成交值以及购买率等业务指标，而后者除了常见的关注点击率之外，还会更加关注人均点击量、用户停留时间以及浏览深度等业务指标。



## AIRec 核心功能 - 场景

这里的场景可以理解为推荐所在的位置。根据推荐所在的位置不同，推荐系统使用模型和手段也会有所不同。比如在淘宝的首页推荐上“猜你喜欢”这部分属于一个比较综合的推荐位置，因此需要综合一些基于用户浏览历史行为的召回，也需要符合用户的兴趣，还需要在照顾热点的同时增加一些对于新品的关注度，因此这部分属于一个比较综合的推荐。而商品详情页的“相关推荐”位置则不同，其更加看重物品与物品之间的关联关系，这个关联可能是商品存在相似关系，也有可能商品之间存在类似于“啤酒和尿布”的关系。“热门推荐”就是热度排行榜了，其背后有着比较成熟的热度算法。而位于页面顶部 Banner 推荐位的“焦点图推荐”则是对于重要活动或

者产品的推荐。



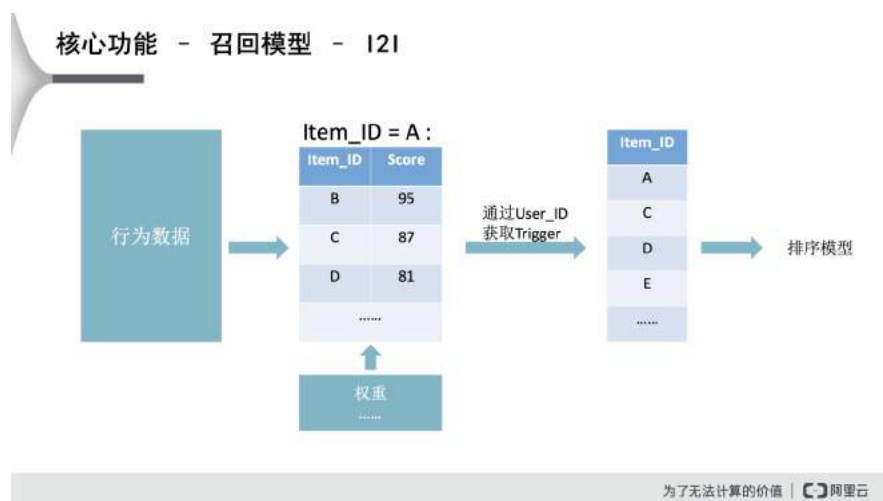
## AIRec 核心功能 – 召回模型

如下图的左侧的推荐模型所示，其最下面的输入可以认为是全量的 Item 列表，通过召回可能筛选出了一批和用户相关的 Item，经过规则过滤将其放入排序层，之后根据用户行为、用户特征以及商品特征进行排序并计算分数，并将最终算分的结果放入到业务逻辑层进行封装，并 Push 给终端用户，使得终端用户能够看到 Push 的结果。下图中右侧是召回系统的简单的实现方式示意图，对于召回而言，想要判断结果是否与用户的兴趣相关，就可以通过不同的链路实现不同兴趣的召回，比如 Item 2 Item、热度、新品、语义相似、User 2 X 2 Item 等。



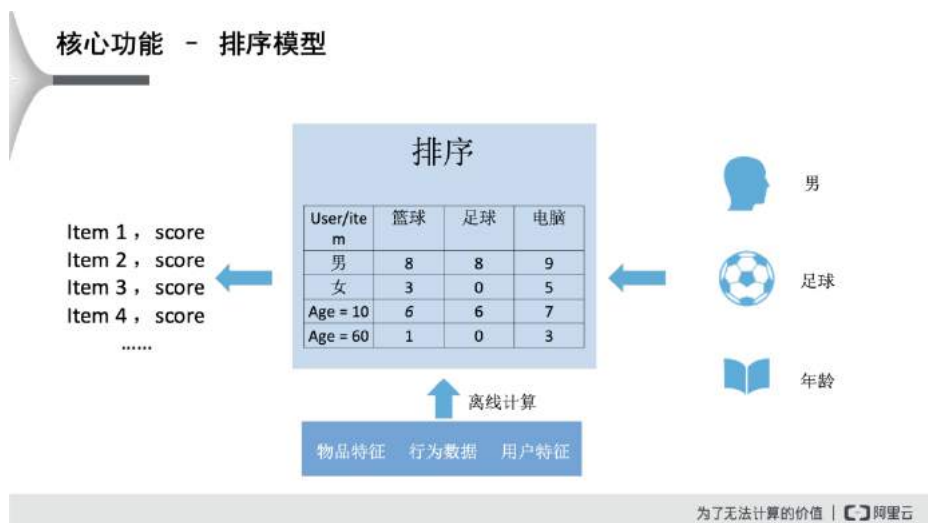
## 召回模型 I2I

在 I2I 这个召回模型中，将用户的行为分为了长期行为和短期行为，进而分别训练出一些表格，这些表可能是 Item\_ID=A 的一些商品可能有一些关联的 Item\_ID，并且这些 Item\_ID 也会有对应的分值，而一些商品的权重可能会影响最终的分值。当这个表计算完成之后，后续如果有一个 User 过来，请求就会带来一个 User\_ID，根据这个 User\_ID 就能获一个 Trigger，进而获取一个 Item\_ID 的列表将其推荐给排序模型，并且这个排序模型还能够接受其他的召回链路的推荐。



## AIRec 核心功能 – 排序模型

如下图所示的是排序模型的简单原理。AIRec 会将用户特征、行为数据以及商品特征全部都进行离线计算，生成针对用户场景的排序模型。举例而言，当一个男生过来，他会带有一些用户特征和行为特征，根据这些内容就可以在模型表上找到对应的分值，将特征所代表的分值进行相加并进行整体倒排，之后就会得到一个排序完成的 Item 表格，最后将这些东西作为排序层的输出给到上面的业务逻辑层。



## AIRec 核心功能 – 打散混排

业务逻辑层中有两个能够控制的点，就是混排和打散。混排可以配置一些比例，而打散则可以配置一些层级和窗口等。

## 核心功能 - 打散混排

### 混排功能

创建混排配置

名称: Text 4/30

名称	配置	操作
image	50 %	删除
article	20 %	删除
video	30 %	删除
item	40 %	删除

[添加混排配置](#)

通过配置混排的 item 配置参数包含 item 的占比数据，所有配置的占比加起来等于 100%。

### 打散功能

创建打散配置

名称: Text 4/30

层级: 2

窗口: 2

为了无法计算的价值 | 阿里云

这里简单介绍一下混排和打散的具体功能：

- 混排功能：一方面是为了让喜好多多种物品类型 (item\_type) 的客户平衡其分发，另一方面是为了让用户在浏览过程中保持物品类型上丰富的多样性，提升用户的惊喜度。AIRec 的混排功能目前支持 image、article、video、short video、item、recipe 等 6 种类型，后面将会增加更多的延展。
- 打散功能：是为了保障在推荐的物品流里相同类目的物品不重复出现，防止用户疲劳，保持新鲜度。使用 AIRec 时可按物品类目（物品表 category\_level 字段）来打散，可根据用户具体业务的物品层级来设置打散。配置打散时选择相应的物品类目，再填写配合该类目层级的窗口期 x，即 x 个物品推荐流里最多出现一次该类目的物品。

这里举个例子，如下图所示，我们对于体育文章、文娱文章以及经济文章可以设置窗口为 3，也就是使得每 3 篇文章里面只能出现 1 次同类型的文章，这样就实现了通过类目进行打散。而混排就是设置文章大约占总量的 50%，短视频和商品大约各占据 25%。通过以上的混排和打散就能够满足推荐的诉求。

## 核心功能 - 打散混排

## 混排功能

一方面是为了让喜好多种物品类型(item\_type)的客户平衡其分发, 一方面是为了让用户在浏览过程中保持物品类型上丰富的多样性, 提升用户惊喜度。

混排目前有image、article、video、shortvideo、item、recipe等6种类型, 后面将会增加更多延展。

如果目前的混排类型不满足您业务的需求, 可提工单或联系您的销售找到我们。

## 打散功能

是为了保障在推荐的物品流里相同类目的物品不重复出现, 防止用户疲劳, 保持新鲜度。

用户可按物品类目(物品表category\_level字段)来打散, 可根据您具体业务的物品层级来设置打散。

配置打散时选择相应的物品类目, 再填写配合该类目层级的窗口期x, 即x个物品推荐流里最多出现一次该类别的物品。



为了无法计算的价值 | 阿里云

## AIRec 核心功能 - 数据仪表盘

阿里云智能推荐 AIRec 可以通过数据仪表盘实现对于数据的跟踪, 包括做分桶测试等。此外, 在 AIRec 上还可以实现基于阿里巴巴的推荐和基于自建的推荐的对比, 并查看数据走向。目前, AIRec 支持的数据指标包括 PV\_CTR、UV\_CTR、PV/UV、Click/UV、活跃的用户数以及活跃的商品数等。

## 核心功能 - 数据仪表盘

## 核心指标:

- 渠道: 行为表中trace\_id即流量分桶标识
- 场景ID: 通常用于接入产品存在多Tab场景
- PV\_CTR: 浏览点击率
- UV\_CTR: 用户点击率
- PV/UV: 平均用户浏览量
- Click/UV: 平均用户点击量
- 有点击用户数
- 活跃商品数



为了无法计算的价值 | 阿里云

### 三、案例及场景

#### 应用案例——内容论坛

客户接入了 AIRec 两部分场景——首页的“猜你喜欢”和详情页面的“相关推荐”。客户使用 AIRec 解决两个主要痛点，一是提高首页“猜你喜欢”的用户点击率，二是期望解决论坛内文章的标题党问题。针对于这两个问题，AIRec 帮助他们做了两步的优化，首先是解决标题党的问题，增加了浏览时长、点赞、收藏等重点行为特征来解决问题，也就是将标题党的文章进行降权，使得其排序结果较低。第二步就是调整了时间衰减等参数，对于热度文章进行降权等操作，调整了排序模型使得业务可干预推荐效果。最终达到的效果就是该论坛在其原有系统的基础之上使得用户点击率提升了 50%。



#### 应用案例——使用架构

前面所提到的论坛客户同时接入了阿里云的搜索和推荐两款产品，从而可以形成比较好的化学反应。推荐产品可以承接首页的流量，使得首页整体流量的转化率得以提升。此外，在搜索方面还接入了 OpenSearch 产品，能够满足用户高质量精准

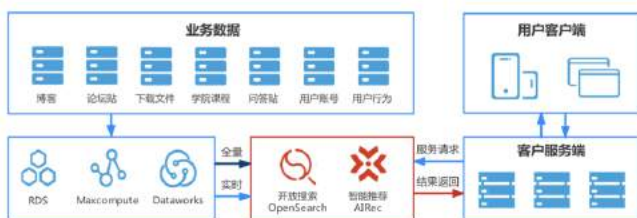
搜索的需求，能够提高有具体浏览目标用户的转化率。通过这两个服务整体地从流量承接以及精准搜索解决了用户的需求，同时用户对于搜索结果的行为也会让智能推荐更加精准。下图中展示了用户使用的架构，会将主流场景下的行为包括数据都传递到 RDS 上，并放到 MaxCompute DataWorks 上进行全量导入，后续还会通过 API 和服务端 SDK 进行实时数据反馈。数据经过 AIRec 和 OpenSearch 两个产品就可以和客户的服务端进行交互，客户的服务端则会进行业务封装并给到最终的用户。

## 应用案例 - 使用架构

### 背景：

客户同时使用推荐、搜索产品

- 推荐：AIRec，用于首页流量承接，提高首页流量整体转化率。
- 搜索：OpenSearch，满足用户高质量搜索需求，提高有具体浏览目标用户的转化率。



## 应用案例——电商

这里再介绍一个电商类客户的应用案例。该案例来自一个国内垂直类电商客户，阿里云 AIRec 为其打造的推荐产品应用在 APP、小程序等多个渠道中。针对于这样的情况，阿里云 AIRec 也为其实现了两部分场景，一部分是“猜你喜欢”，另外一个“相关推荐”，并最终提升了两个场景点击率。之前他们的场景中存在问题就是 SKU 比较少，用户量级相对而言也不是非常大，因此从整体来看在这样的场景下 AIRec 能够帮助客户达到点击率的提升，效果还是不错的。总体经过了一个月左右的调优，使得点击率提升了 1 倍左右。





## 阿里技术

扫一扫二维码图案，关注我吧



阿里云实时计算



阿里巴巴大数据计算



DataWorks 开发者交流钉钉群



扫码关注阿里技术