阿里云
专有云大数据版

产品简介

产品版本：V2.1.0

文档版本：20180730

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的”现状“、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云文档中所有内容，包括但不限于图片、架构设计、页面布局、文字描述，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表本文档中的内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含”阿里云”、Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 阿里云文档中所有内容，包括但不限于图片、页面设计、文字描述，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表本文档中的内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包

含”阿里云”、Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

7. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 **禁止**： 重置操作将丢失用户配置数据。
	该类警示信息可能导致系统重大变更甚至故障，或者导致人身伤害等结果。	 **警告**： 重启操作将导致业务中断，恢复业务所需时间约10分钟。
	用于警示信息、补充说明等，是用户必须了解的内容。	 **注意**： 导出的数据中包含敏感信息，请妥善保存。
	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 **说明**： 您也可以通过按**Ctrl** + **A**选中全部文件。
>	多级菜单递进。	**设置** > **网络** > **设置网络类型**
粗体	表示按键、菜单、页面名称等UI元素。	单击 **确定**。
`courier` 字体	命令。	执行 `cd /d C:/windows` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *`Instance_ID`*
[]或者[a|b]	表示可选项，至多选择一个。	`ipconfig` *`[-all|-t]`*
{}或者{a|b}	表示必选项，至多选择一个。	`swich` *`{stand | slave}`*

目录

1 大数据轻量专有云简介

1.1 什么是大数据轻量专有云

大数据轻量专有云(Apsara Stack Insight)是面向中小型企业用户，提供大数据端到端、全链路业务的轻量级软硬件一体化解决方案。

大数据轻量专有云基于阿里云飞天分布式操作系统Apsara开发，能够提供完备的大数据计算服务能力以及丰富的大数据应用。它的出现极大地降低了客户使用大数据产品的成本和门槛，这也使阿里云大数据产品普惠到各行各业。

大数据轻量专有云目前已经实现一键快捷部署、规模扩展和统一运维管控平台，并达到国家安全等保三级要求。

1.2 产品优势

大数据轻量专有云作为轻量级的解决方案，其产品优势主要体现在以下几个方面。

轻量级

大数据轻量专有云提供轻量级的软硬件部署，硬件部署最少只需要服务器15台，典型配置为7(飞天底座)+6(MaxCompute)+2(DataWorks)模式，无需DSW等网络设备。软件交付部署时间可以在12小时内完成，去除了中间件和其他冗余的产品。此外，还能够提供强大的向上扩展能力。

模块化

按需搭配计算服务，实现大数据产品灵活组合。可水平扩展已部署服务的计算节点，也可水平扩展部署新的计算服务。配置简单，易于复制。此外，用户还可以根据自身业务定制产品形态，如离线计算、实时计算、流计算等。

开放性

兼容开放标准，提供丰富的SDK包和RESTful API接口。用户可以使用开放接口来灵活访问专有云提供的各种云服务

端到端

各大数据产品形成了完成的数据链路闭环，涵盖了数据集成、数据治理， 数据开发、数据挖掘、数据展示各个环节。

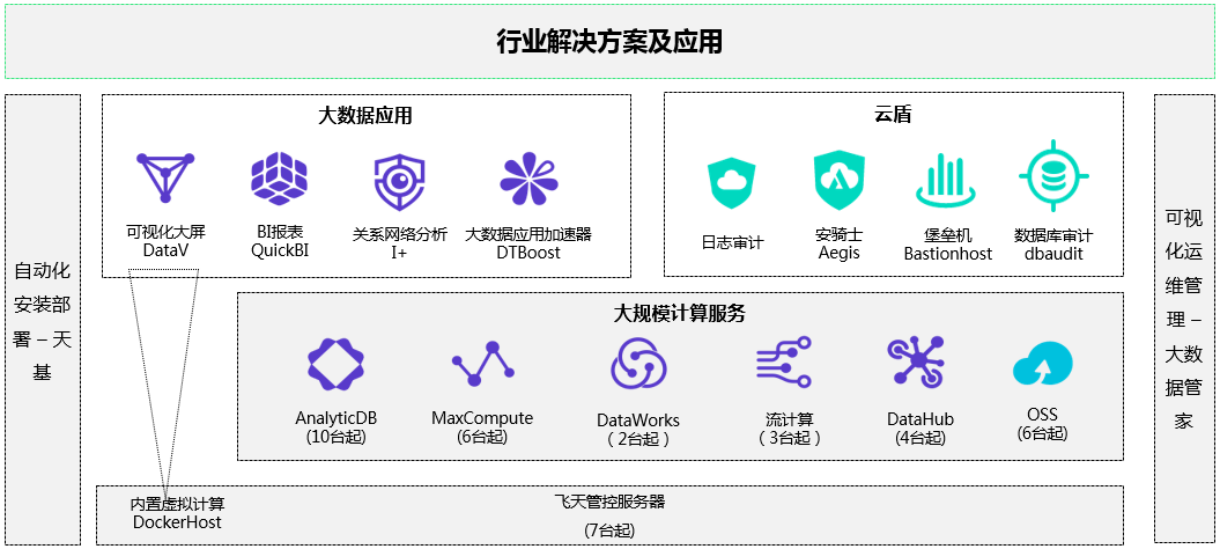
1.3 产品架构

1.3.1 系统架构

大数据轻量专有云的系统架构如图 *1-1:* 大数据轻量专有云架构图所示，底层硬件设备基于标准的IDC基础架构、x86服务器、以太网交换机的大数据产品。在硬件服务器之上，大数据轻量专有云搭载飞天操作系统，形成大规模集群。大规模计算服务是基于飞天操作系统开发的大数据计算平台。在计算平台之上，大数据轻量专有云提供大数据应用服务。

大数据轻量专有云支持自动化安装部署和可视化运维。运维人员可通过工具便捷的安装部署大数据轻量专有云，并可通过可视化运维管理工具维护大数据轻量专有云中的各产品，极大的提升运维人员的工作效率。

图 1-1: 大数据轻量专有云架构图



1.3.2 网络架构

大数据轻量专有云采用专有云万兆 V1.0-Mini架构，如图 *1-2:* 网络架构图所示。带内网络由四台万兆交换机组成，其中两台ASW（接入交换机）做堆叠，两台ISW（出口交换机）40G互联，所有物理机bond0双上行10G光纤连接分别堆叠ASW，每组ASW 各160G分别上联每台ISW带外网络为前兆网络，OMR连接所有其他交换机的管理口并连接ops1和ops2的eth0，OASW则连接每台服务器的BMC管理口网络架构当前限制为2组（4台ASW），最大支持96台物理服务器。

图 1-2: 网络架构图

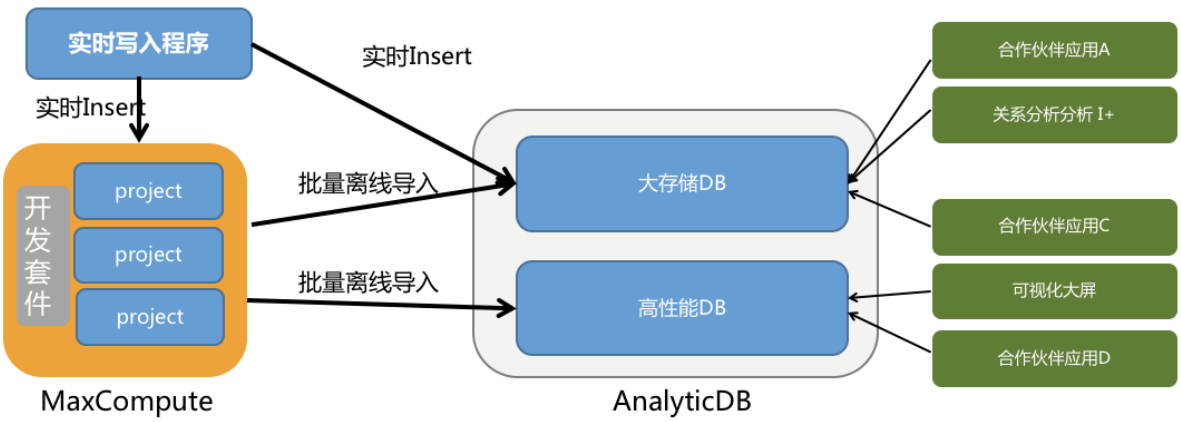
表 1-1: 网络架构角色说明

角色名称	所属模块	作用
ISW（互联交换机）	外网接入模块	出口交换机，互联ISP或用户网络骨干。
ASW （接入层交换机）	数据交换模块	接入交换机，接入云服务器，上行互联交换机ISW。
OMR	带外管理交换机	用于ops机器管理其他交换机设备
OASW	带外接入交换机	接服务器的BMC接口，管理带外网络

1.3.3 典型应用

在使用大数据轻量专有云过程中，大数据轻量专有云各产品会形成数据链路闭环。大数据轻量专有云各产品形成的数据链路闭环，如图 *1-3:* 数据链路闭环示意图所示。

图 1-3: 数据链路闭环示意图



将源数据写入到AnalyticDB的方式如下：

- 支持从MaxCompute同步数据到AnalyticDB：
  1. 用户离线数据通过DataWorks提供的数据集成功能，从源数据库中将数据同步到MaxCompute的project中；用户实时数据通过实时写入程序实时写入到MaxCompute的project中。
  2. 用户根据业务逻辑在DataWorks中开发任务，可以是ETL类任务、与业务逻辑相关的任务、多个有依赖性的任务。

3. 用户开发同步任务将数据批量离线同步到AnalyticDB。

- 支持通过实时写入程序将实时数据以标准INSERT语句写入到AnalyticDB。

关系网络分析I+、大数据应用加速器DTBoost及合作伙伴都对接到分析型数据库，由分析型数据库提供高并发、低时延的多维查询功能。

AnalyticDB支持创建高性能和大存储两种数据库实例。高性能实例仅使用SSD盘作为数据存储；大存储实例使用SSD盘与SATA盘混合存储。因此，大存储实例单GB的成本更低。

1.4 应用场景

大数据轻量专有云能够为不同规模、同行业的用户提供灵活的、可扩展的行业解决方案。本章节将重点介绍如下三个应用场景。

金融云

金融云是服务于银行、证券、保险、基金等金融机构的行业云，采用独立的机房集群提供满足一行三会监管要求的云产品，并为金融客户提供更加专业周到的服务。通过自建/共建楼模式 ，满足中大型金融机构需要完全物理隔离的独立云机房需求，能够将大数据平台输出到客户的。

司法云

司法云是服务于地市级中级法院甚至高院的行业云，依托MaxCompute、OSS、DataWorks，再加上人工智能服务，实现各级法院对智慧立案、智慧庭审、智慧办案、智慧公诉、智慧执行等新时代下的“智慧法院”建设目标。

边缘计算云

在金融、能源、安全等场景下，越来越多的诉求是在多个边缘节点上完成数据的加工处理，然后再同步到中心节点作汇聚处理。特别是在视频处理中，边缘计算扮演着越来越重要的角色。边缘计算云以Blink/Datahub为核心，可以高可靠、高性能实时处理海量数据，满足多种场景下对小规模实时视频处理的诉求。

2 对象存储OSS

2.1 什么是对象存储OSS

对象存储服务（Object Storage Service，简称 OSS）提供海量、安全、低成本、高可靠的云存储服务。OSS可以被理解成一个即开即用，无限大空间的存储集群。

OSS 将数据文件以对象/文件（Object）的形式上传到存储空间（Bucket）中。OSS 提供的是一个 Key-Value 键值对形式的对象存储服务。用户可以根据Object的名称（Key）唯一地获取该Object的内容。

相比传统自建服务器存储，OSS 在可靠性、安全性、成本和数据处理能力方面都有着突出的优势。使用 OSS，您可以通过网络随时存储和调用包括文本、图片、音频和视频等在内的各种非结构化数据文件。

2.2 产品优势

OSS与自建存储对比的优势

对比项	对象存储 OSS	自建服务器存储
可靠性	• 服务可用性高。 • 规模自动扩展，不影响对外服务。 • 数据持久性高。 • 数据自动多重冗余备份。	• 受限于硬件可靠性，易出问题，一旦出现磁盘坏道，容易出现不可逆转的数据丢失。 • 人工数据恢复困难、耗时、耗力。
安全	• 提供企业级多层次安全防护。 • 多用户资源隔离机制，支持同城容灾。 • 提供多种鉴权和授权机制，以及白名单、防盗链、主子账号、STS临时授权访问功能。	• 需要另外购买清洗和黑洞设备。 • 需要单独实现安全机制。
数据处理能力	提供图片处理功能。	需要额外采购，单独部署。

OSS具备的其他各项优势

- 方便、快捷的使用方式

提供标准的 RESTful API 接口（部分接口与Amazon S3 API兼容）、丰富的 SDK 包、客户端工具、控制台。您可以像使用文件一样方便地上传、下载、检索、管理用于 Web 网站或者移动应用的海量数据。

- 不限文件数量和大小。您可以根据所需存储量无限扩展存储空间，解决了传统硬件存储扩容问题。
- 支持流式写入和读出。特别适合视频等大文件的边写边读业务场景。
- 支持数据生命周期管理。您可以自定义将到期数据批量删除。

- 强大、灵活的安全机制

  灵活的鉴权、授权机制。提供 STS 和 URL 鉴权和授权机制，以及白名单、防盗链、主子账号功能。

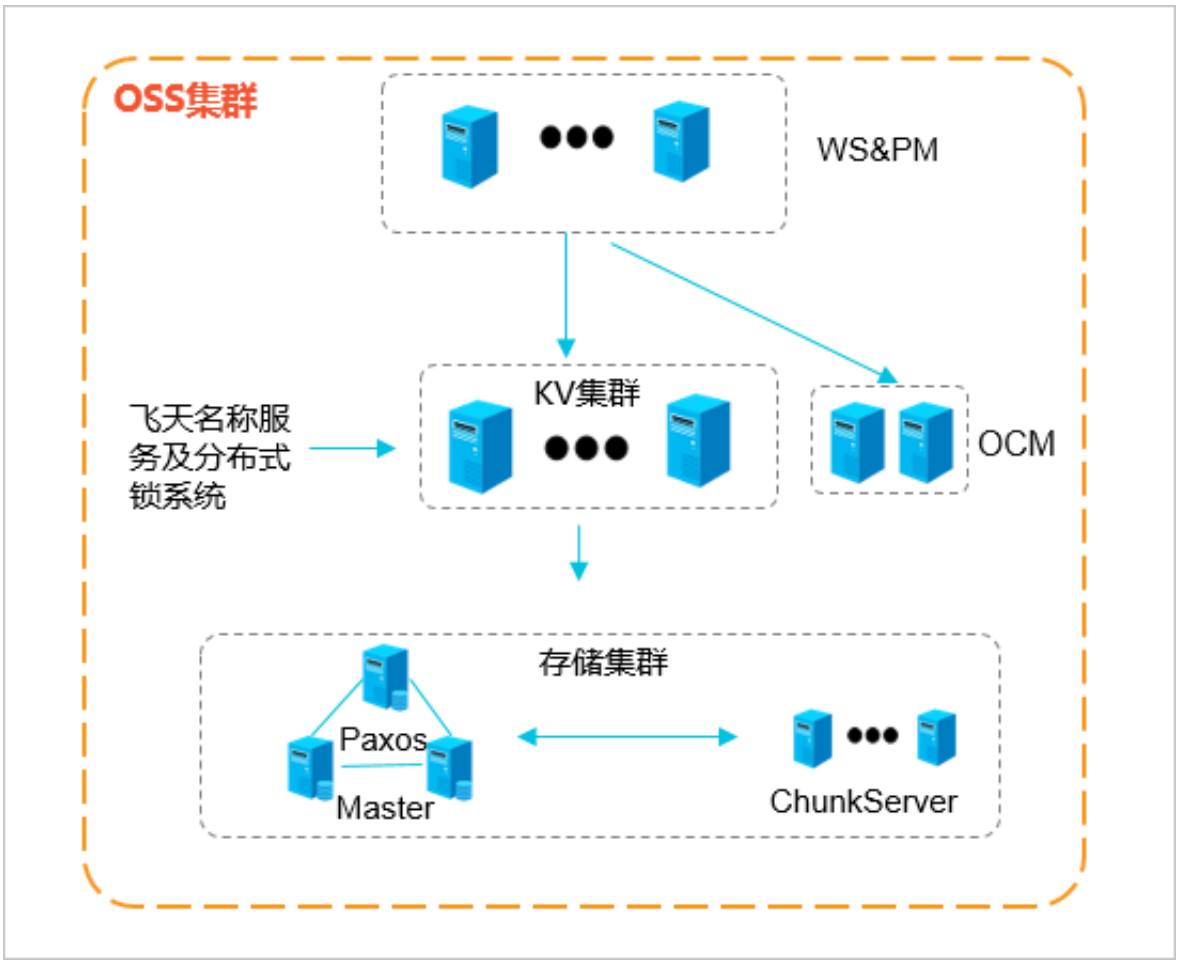
- 丰富的图片处理服务

  支持 jpg、png、bmp、gif、webp、tiff 等多种图片格式的转换，以及缩略图、剪裁、水印、缩放等多种操作。

2.3 产品架构

对象存储 OSS 是构建在阿里云飞天平台上的一种存储解决方案。其基础是飞天平台的分布式文件系统，分布式任务调度等基础设施。这些基础设施提供了 OSS 以及其他阿里云服务所需的分布式调度、高速网络、分布式存储等重要特性。OSS 的架构如下图所示：

图 2-1: OSS 架构图



- WS&PM（协议接入层）：负责接收用户使用 RESTful协议发来的请求，进行安全认证。如果认证通过，用户的请求将被转发到 Key-Value引擎继续处理。如果认证失败，直接返回错误信息给用户。
- KV集群：负责数据结构化处理，即按照 Key（对象名称）来查找或存储数据，并支持大规模并发的请求。当协调服务集群变更导致服务被迫改变运行物理位置时，可以快速协调找到接入点。
- 存储集群：元数据存储在 Master 上，Master 之间采用分布式消息一致性协议（Paxos）保证元数据的一致性。从而实现高效的文件分布式存储和访问。

2.4 功能特性

OSS提供以下主要功能：

表 2-1: OSS主要功能

类别	功能	描述
存储空间	创建存储空间	在上传任何文件到 OSS 之前，您需要首先创建存储空间来存储文件。
	删除存储空间	如果您不再需要存储空间，请将其删除以免进一步产生费用。
	修改存储空间读写权限	OSS 提供权限控制 ACL（Access Control List），您可以在创建存储空间的时候设置相应的 ACL 权限控制，也可以在创建之后修改 ACL。
	设置静态网站托管	将存储空间配置成静态网站托管模式，并通过存储空间域名访问该静态网站。
	设置防盗链	为了减少您存储于 OSS 的数据被其他人盗链而产生额外费用，OSS 支持设置基于 HTTP header 中表头字段 referer 的防盗链方法。
	管理跨域资源共享	OSS 提供 HTML5 协议中的跨域资源共享 CORS 设置，帮助您实现跨域访问。
	设置生命周期	定义和管理存储空间内所有对象或对象的某个子集的生命周期。设置生命周期一般用于文件的批量管理和自动碎片删除等操作。
对象（文件）	上传文件	您可以上传任意类型文件到存储空间中。
	新建文件夹	您可以像管理 Windows 文件夹一样管理 OSS 文件夹。
	搜索文件	在存储空间或文件夹中搜索具有相同的名称前缀的文件。
	获取文件访问地址	通过获取已上传文件的地址进行文件的分享和下载。
	删除文件	删除单个文件或批量删除文件。
	删除文件夹	删除单个文件夹或批量删除文件夹。
	修改文件读写权限	您可以在上传文件的时候设置相应的 ACL 权限控制，也可以在上传之后修改 ACL。
	管理碎片	删除存储空间内的全部或部分碎片文件。
图片处理	图片处理	对保存在OSS上的图片进行格式转换、剪裁、缩放、旋转、水印、样式封装等各种处理。
API	API	提供 OSS支持的 RESTful API 操作和相关示例。

类别	功能	描述
SDK	SDK	提供主流语言 SDK 的开发操作和相关示例。

2.5 应用场景

图片和音视频等应用的海量存储

OSS可用于图片、音视频、日志等海量文件的存储。各种终端设备、Web网站程序、移动应用可以直接向OSS写入或读取数据。OSS支持流式写入和文件写入两种方式。

网页或者移动应用的静态和动态资源分离

利用BGP带宽，OSS可以实现超低延时的数据直接下载。

离线数据归档存储

依靠低成本、高可用的OSS对象存储，可以将企业内部长期需要离线归档的数据转存至OSS。

2.6 使用限制

限制项	说明
存储空间（bucket）	• 同一用户创建的存储空间总数不能超过10个。 • 存储空间一旦创建成功，名称和区域不能修改。
上传文件	• 通过控制台上传、简单上传、表单上传、追加上传的文件大小不能超过5GB，要上传大小超过5GB的文件必须使用分片上传(Multipart Upload）上传方式。分片上传方式上传的文件大小不能超过48.8TB。 • OSS支持上传同名文件，但会覆盖已有文件。
删除文件	• 文件删除后无法恢复。 • 控制台批量删除文件的上限为50个，更大批量的删除必须通过API或SDK实现。
生命周期	每个存储空间的生命周期配置最多可容纳1000条规则。
图片处理	• 对于原图： - 图片格式只能是：jpg、png、bmp、gif、webp、tiff。 - 文件大小不能超过20MB。 - 使用图片旋转或裁剪时图片的宽或者高不能超过4096。 • 对于缩略后的图：

限制项	说明
	- 宽与高的乘积不能超过4096x4096。 - 单边长度不能超过4096。

2.7 基本概念

本部分将向您介绍 OSS 中涉及的几个基本概念，以便于您更好地理解 OSS 产品。

对象/文件（Object）

对象是 OSS 存储数据的基本单元，也被称为 OSS 的文件。对象由元信息（Object Meta），用户数据（Data）和文件名（Key）组成。对象由存储空间内部唯一的 Key 来标识。对象元信息是一个键值对，表示了对象的一些属性，比如最后修改时间、大小等信息，同时用户也可以在元信息中存储一些自定义的信息。

对象的生命周期是从上传成功到被删除为止。在整个生命周期内，对象信息不可变更。重复上传同名的对象会覆盖之前的对象，因此，OSS 不支持修改文件的部分内容等操作。

说明：
如无特殊说明，本文档中的对象、文件称谓等同于 Object。

存储空间（Bucket）

存储空间是您用于存储对象（Object）的容器，所有的对象都必须隶属于某个存储空间。您可以设置和修改存储空间属性用来控制访问权限、生命周期等，这些属性设置直接作用于该存储空间内所有对象，因此您可以通过灵活创建不同的存储空间来完成不同的管理功能。

- 同一个存储空间的内部是扁平的，没有文件系统的目录等概念，所有的对象都直接隶属于其对应的存储空间。
- 每个用户可以拥有多个存储空间。
- 存储空间的名称在 OSS 范围内必须是全局唯一的，一旦创建之后无法修改名称。
- 存储空间内部的对象数目没有限制。

强一致性

Object 操作在 OSS 上具有原子性，操作要么成功要么失败，不会存在有中间状态的Object。OSS 保证用户一旦上传完成之后读到的 Object 是完整的，OSS 不会返回给用户一个部分上传成功的 Object。

Object 操作在 OSS 上同样具有强一致性，用户一旦收到了一个上传（PUT）成功的响应，该上传的 Object 就已经立即可读，并且数据的三份副本已经写成功。不存在一种上传的中间状态，即 read-after-write 却无法读取到数据。对于删除操作也是一样的，用户删除指定的 Object 成功之后，该 Object 立即变为不存在。

强一致性方便了用户架构设计，可以使用跟传统存储设备同样的逻辑来使用OSS，修改立即可见，无需考虑最终一致性带来的各种问题。

OSS与文件系统的对比

OSS 是一个分布式的对象存储服务，提供的是一个 Key-Value 对形式的对象存储服务。用户可以根据 Object 的名称（Key）唯一地获取该Object的内容。虽然用户可以使用类似 test1/test.jpg 的名字，但是这并不表示用户的 Object 是保存在test1 目录下面的。对于 OSS 来说，test1/test.jpg 仅仅只是一个字符串，和 a.jpg 这种并没有本质的区别。因此不同名称的 Object 之间的访问消耗的资源是类似的。

文件系统是一种典型的树状索引结构，一个名为 test1/test.jpg 的文件，访问过程需要先访问到 test1 这个目录，然后再在该目录下查找名为 test.jpg 的文件。因此文件系统可以很轻易的支持文件夹的操作，比如重命名目录、删除目录、移动目录等，因为这些操作仅仅只是针对目录节点的操作。这种组织结构也决定了文件系统访问越深的目录消耗的资源也越大，操作拥有很多文件的目录也会非常慢。

对于 OSS 来说，可以通过一些操作来模拟类似的功能，但是代价非常昂贵。比如重命名目录，希望将 test1 目录重命名成 test2，那么 OSS 的实际操作是将所有以 test1/ 开头的 Object 都重新复制成以 test2/ 开头的 Object，这是一个非常消耗资源的操作。因此在使用 OSS 的时候要尽量避免类似的操作。

OSS 保存的 Object 不支持修改（追加写 Object 需要调用特定的接口，生成的 Object 也和正常上传的 Object 类型上有差别）。用户哪怕是仅仅需要修改一个字节也需要重新上传整个 Object。而文件系统的文件支持修改，比如修改指定偏移位置的内容、截断文件尾部等，这些特点也使得文件系统拥有广泛的适用性。但另外一方面，OSS 能支持海量的用户并发访问，而文件系统会受限于单个设备的性能。

因此，将 OSS 映射为文件系统是非常低效的，也是不建议的做法。如果一定要挂载成文件系统的话，建议尽量只做写新文件、删除文件、读取文件这几种操作。使用 OSS 应该充分发挥其优点，即海量数据处理能力，优先用来存储海量的非结构化数据，比如图片、视频、文档等。

3 云盾

3.1 什么是云盾

专有云云盾是防护云上安全的一套专有云安全解决方案。

在云计算环境下，随着技术的发展，传统以检测技术为主的边界安全防护不能充分保障云上业务的安全。专有云云盾结合云计算平台强大的数据分析能力以及专业的安全运营团队，从网络层、应用层、主机层等多个层面为用户提供一体化的安全防护服务。

在专有云大数据版中，云盾包含安骑士与安全审计两大功能模块。

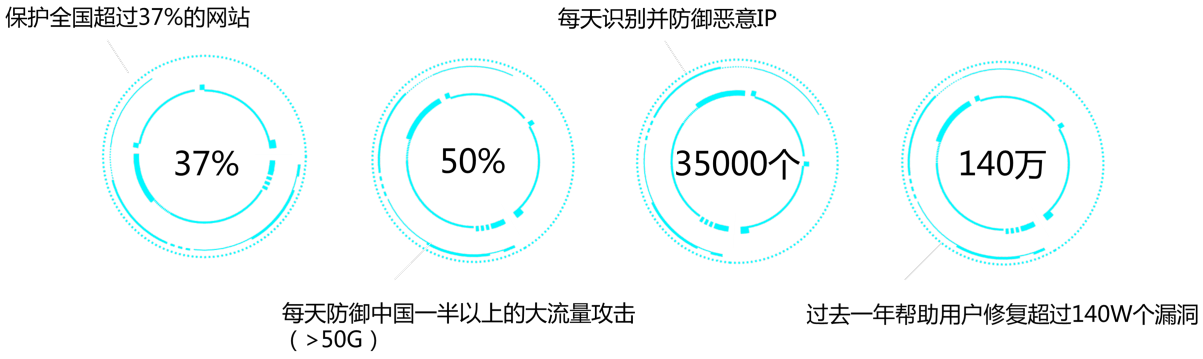
3.2 产品优势

3.2.1 云上安全先行者

阿里安全团队从2005年起护航阿里巴巴集团内部所有业务系统的信息安全，不断积累安全经验。自2011年首次推出云盾产品，全方位保障云上安全，成为云上安全先行者。

云盾保护全国超过37%的网站，每天防御中国一半以上的大流量攻击，每天识别并防御35,000个恶意IP，过去一年帮助用户修复超过140万个漏洞。

图 3-1: 云盾护航处理海量互联网数据



3.2.2 权威认证，安全可靠

依靠平台自身的安全特性以及云盾为云上客户提供的攻击防御特性，先后取得了国内外多项云安全认证。

图 3-2: 获得的安全认证

- 全球首家获得云安全国际认证金牌（CSA STAR Certification）的云服务供应商。
- 全国首家获得ISO27001信息安全管理体系国际认证的云安全服务供应商。
- 全国首个通过公安部等级保护测评（DJCP）的云计算系统。
- 电子政务云平台首批通过党政部门云服务网络安全审查（增强级）。
- 全国首家云等保试点示范平台。
- 金融云高分通过等保四级测评，成为全国首个四级云平台。

3.2.3 体系完整，技术领先

十年攻防，一朝成盾。在经历了为阿里巴巴集团自身业务十年来的安全护航后，阿里巴巴积累了大量的安全研究成果、安全数据、安全运营和安全管理方法，形成了一支专业的云安全专家团队。 云盾是集合这些安全专家多年攻防经验开发出来的面向云计算平台安全最佳实现的成熟体系，可有效保护专有云用户云平台、云网络环境和云业务系统的安全。

3.2.4 与传统安全产品对比

特点	传统安全产品	云盾
顶级互联网企业安全能力完整输出	传统安全厂商都有各自擅长的产品，但在其他产品上存在短板，无法形成整体安全防护体系。	阿里巴巴集团在与黑客攻击对抗过程中沉淀了大量的情报能力，及时发现流行的互联网攻击行为以及0 Day攻击手段，为用户提供完整的安全能力。
提前研判、预知风险爆发	传统安全厂商缺少完整的监控业务场景，无法准确研判风险的爆发。	对于重大漏洞、重大安全事件能够进行分析并及时响应，避免安全问题的爆发。

特点	传统安全产品	云盾
弹性扩容，与硬件解耦合	基于自身定制的硬件设备实现；软件化后的安全产品，也是基于虚拟化平台上的虚拟主机实现。	• 硬件解耦合：采用云架构设计，所有功能模块都基于通用的X86硬件平台，对硬件无依赖性。 • 弹性扩容：在性能不足时，无需改造网络结构，直接平滑地扩展硬件数量即可。
网络和主机的联动	通过设备的叠加来实现功能的完整覆盖。各设备之间不能有效的联动，一般只能通过管理平台将各个设备的日志、状态进行统一收集、展现。	提供完整的网络安全、应用安全、主机安全的互联网防护能力，各防护组件互相联动形成整体防护体系，对已知的攻击行为达到最佳的防护效果。
兼容所有的IDC环境，和云平台解耦合	大部分的传统安全厂商仍然基于硬件盒子的形态提供安全产品。在SDN技术越来越普及的情况下，这样的形态无法与云环境兼容。	采用“网络出口检测 + 服务器操作系统联动”架构；采用数据分析方式发现安全威胁。通过这种架构及方式，避开了IDC内部复杂的网络结构，完全兼容所有的IDC环境。

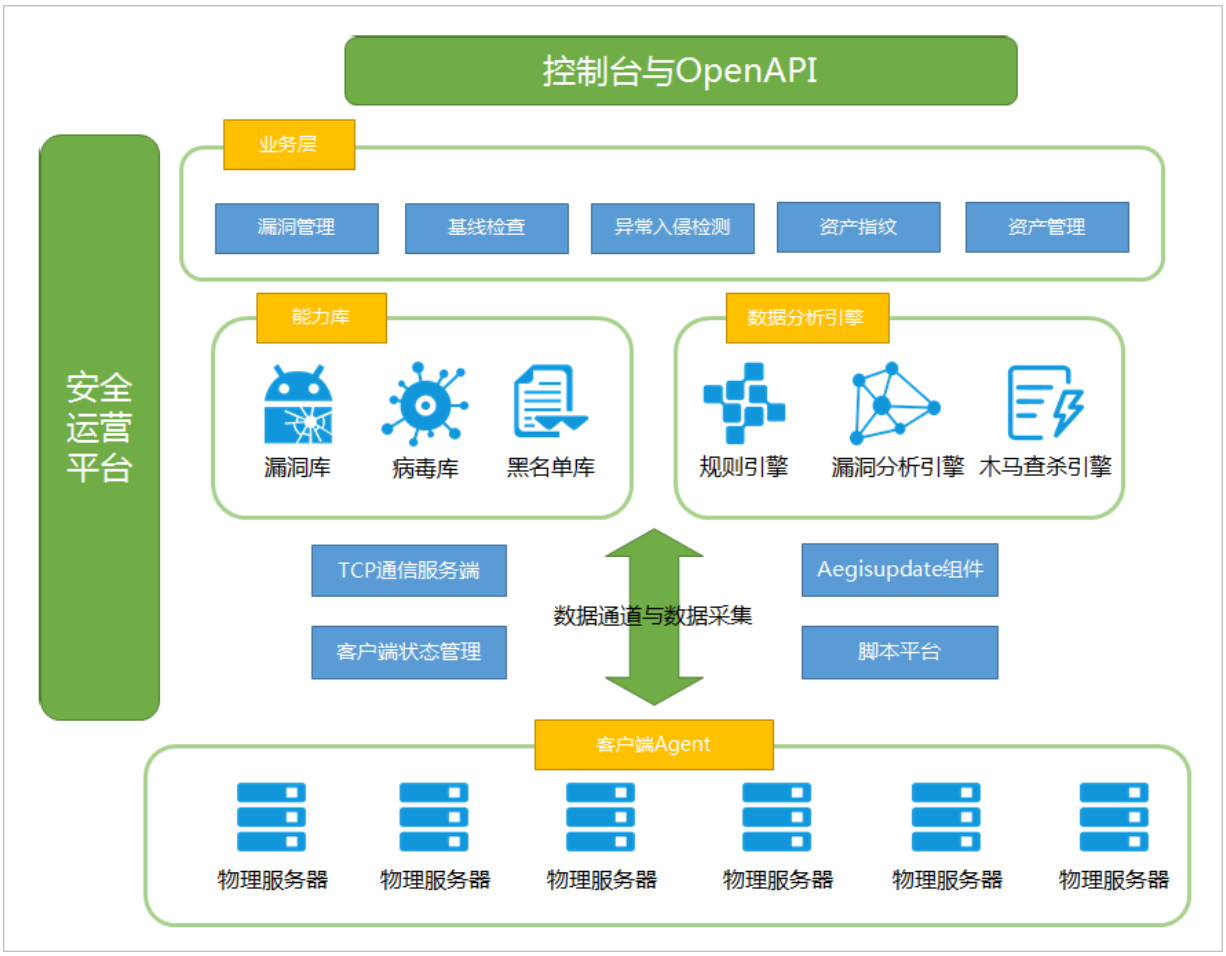
3.3 产品架构

在专有云大数据版中，云盾由安骑士与安全审计两大功能模块组成。

安骑士

安骑士功能模块的产品结构如图 *3-3:* 安骑士产品结构图所示。

图 3-3: 安骑士产品结构图

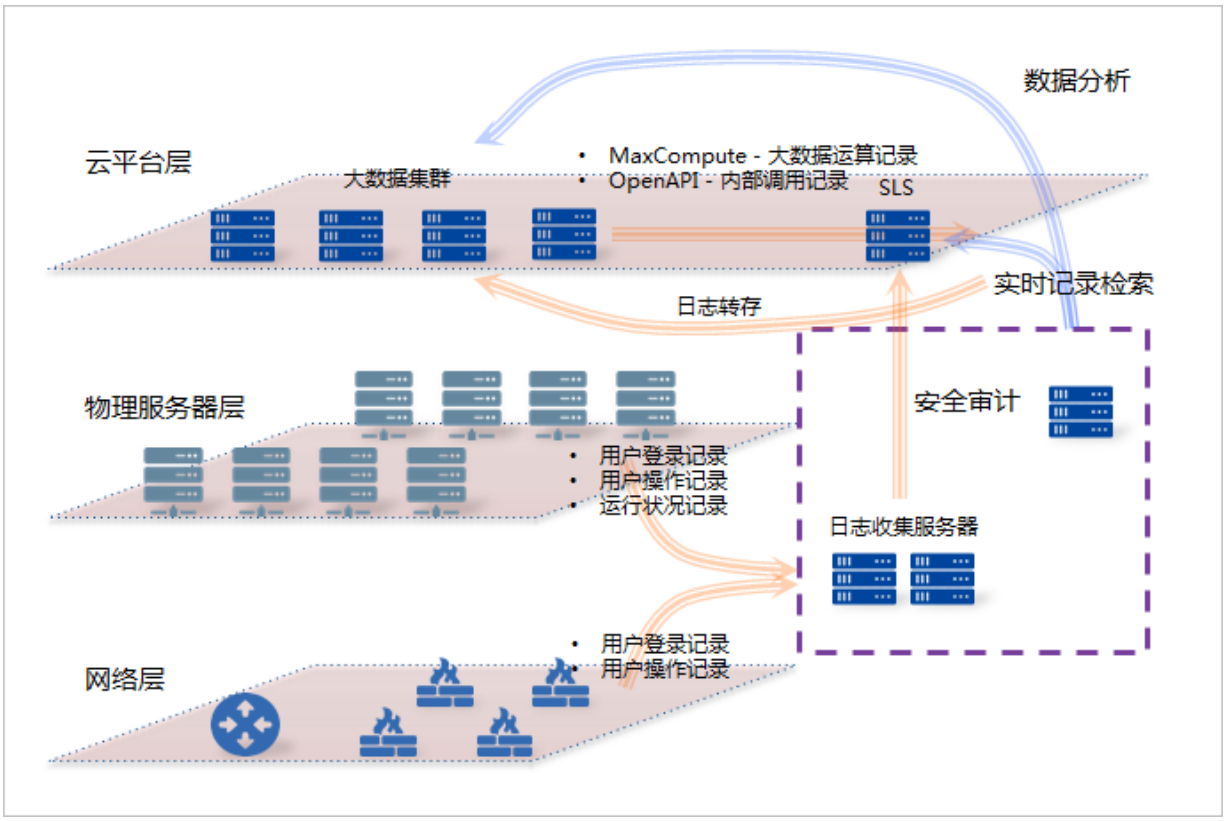


安骑士模块采用Client-Server架构，在物理服务器上的客户端Agent与服务端通过TCP长连接进行通信，通过HTTP方式从服务端获取需要的脚本、规则、安装包等文件。基于客户端收集到的主机的信息，通过日志监控、文件分析、特征扫描等手段，为专有云环境中的主机提供漏洞管理、基线检查、入侵检测、资产管理等安全防护措施。

安全审计

安全审计功能模块的产品结构如图 *3-4:* 安全审计产品结构图所示。

图 3-4: 安全审计产品结构图



安全审计模块通过收集网络设备、物理服务器、MaxCompute产品、OpenAPI调用的日志记录，基于所设定的审计策略，对审计日志和审计事件进行分析并以图表方式呈现，帮助用户直观了解专有云环境面临的风险和威胁。

3.4 功能特性

云盾不同于传统的软硬件安全产品，它采用纵深防御、多点联动的云安全架构，完全基于专有云的云计算环境研发，从网络层、应用层、主机层等多个层面为用户提供全面的、一体化的云安全防护能力。

主要功能

云盾所包含的详细功能如表 *3-1:* 云盾功能列表所示。

表 3-1: 云盾功能列表

模块	功能项	功能说明
安骑士	网站后门查杀	通过规则匹配对云服务器中存在的脚本后门进行精准查杀，并可手动对脚本后门进行隔离。

模块	功能项	功能说明
	暴力破解攻击拦截	对黑客进行暴力破解的行为进行实时检测和拦截。
	异常登录告警	通过分析和比对用户常用登录设置，对疑似的非常用登录行为进行告警。
	漏洞扫描	对云服务器主机进行扫描，发现主机软件漏洞并提供漏洞修复方式。
	漏洞修复	对云服务器主机中的应用和操作系统的高危漏洞提供一键修复，包括Web应用漏洞修复、系统文件修复等。
	基线检查	对云服务器主机中的安全基线进行检查，包括账户安全检测、弱口令检查、以及配置项风险检测，以达到企业级服务器安全准入标准。
	资产指纹	对云服务器主机的端口、账号、进程、应用软件等进行清点，全面了解主机资产的运行状态并有效进行回溯分析。
	日志检索	将散落的云服务器主机的进程、网络、系统登录等日志集中管理，帮助在主机出现问题时一站式搜索相关日志，快速定位问题根源。
安全审计	原始日志采集	支持采集以下日志： • MaxCompute运算记录日志、主机日志。 • 用户侧控制台操作日志、运维侧控制台操作日志。 • 网络设备日志。
	审计查询	根据**审计类型**、**审计对象**、**操作类型**、**操作风险级别**、**是否告警**与**创建时间**进行审计日志查询。同时，支持审计日志的全文检索。
	策略设置	支持根据**发起者**、**目标**、**命令**、**结果**与**原因**参数设置审计规则，对原始日志进行高危操作识别并告警。

3.5 使用限制

无

3.6 基本概念

Web 应用防火墙

基于大数据挖掘技术，实现Web层的安全攻击实时侦测和实时拦截。

密码暴力破解

根据密码命名规则的部分条件确定密码的大致范围，并在此范围内对所有可能的情况逐一验证，直到全部情况验证完毕。

网站后门

一段运行在服务器端的网页代码，主要以ASP和PHP代码为主，攻击者通过这段代码，在服务器端进行某些危险的操作，获得某些敏感的技术信息或者通过渗透，提权获得服务器的控制权。

4 MaxCompute

4.1 什么是MaxCompute

大数据计算服务（MaxCompute）是阿里巴巴内部发展的一个高效能、低成本，高可用的**EB级**大数据计算服务，在集团内部每天处理超过EB级的数据量。MaxCompute是面向大数据处理的分布式系统，主要提供结构化数据的存储和计算，是阿里巴巴云计算整体解决方案中最核心的主力产品之一。

多租户、数据安全、水平扩展等特性是MaxCompute的核心设计目标，采用抽象的作业处理框架为不同用户对各种数据处理任务提供统一的编程接口和界面。

MaxCompute主要服务于批量结构化数据的存储和计算，可以提供海量数据仓库的解决方案以及针对大数据的分析建模服务。MaxCompute的目的是为用户提供一种便捷的分析处理海量数据的手段。用户可以不必关心分布式计算细节，从而达到分析大数据的目的。

MaxCompute产品特点如下：

- 采用分布式架构，规模可以根据需要平行扩展。
- 自动存储容错机制，保障数据高可靠性。
- 所有计算在沙箱中运行，保障数据高安全性。
- 以RESTful API的方式提供服务。
- 支持高并发、高吞吐量的数据上传下载。
- 支持离线计算、机器学习两类模型及计算服务。
- 支持基于SQL、Mapreduce、Graph、MPI等多种编程模型的数据处理方式。
- 支持多租户，多个用户可以协同分析数据。
- 支持基于ACL和policy的用户权限管理，可以配置灵活的数据访问控制策略，防止数据越权访问。
- 支持Spark的增强应用，即Spark on MaxCompute。

4.2 产品优势

国内唯一的大数据云服务平台，真正的数据分享平台

- 能够同时做到数据仓库、数据挖掘、数据分析、数据分享。
- 阿里集团内部使用的统一数据处理平台，支持阿里贷款、数据魔方、DMP（阿里巴巴广告联盟）、余额宝等多款产品。

MaxCompute支持的集群及用户规模极大，同时能够支持极高的作业并发数

- 单一集群规模可以达到10000+服务器（保持80%线性扩展）。
- 单个MaxCompute可以多集群部署100万服务器以上，无限制（线性扩展略差），支持同城多数据中心模式。
- 10000+用户数，1000+项目应用，100+部门（多租户）。
- 100万以上作业（目前单日平均提交任务），20000以上并发作业。

海量运算触手可得

用户不必关心数据规模增长带来的存储困难、运算时间延长等烦恼，MaxCompute根据用户的数据规模自动扩展集群的存储和计算能力，使用户专心于数据分析和挖掘，最大化发挥数据的价值。

服务“开箱即用”

用户不必关心集群的搭建、配置和运维工作，仅需简单的几步操作，用户便可以在MaxCompute中上传数据、分析数据并得到分析结果。

数据存储安全可靠

采用多副本技术、读写请求鉴权、应用沙箱、系统沙箱等多层次数据存储和访问安全机制来保护用户的数据，使其不丢失、不泄露、不被窃取。

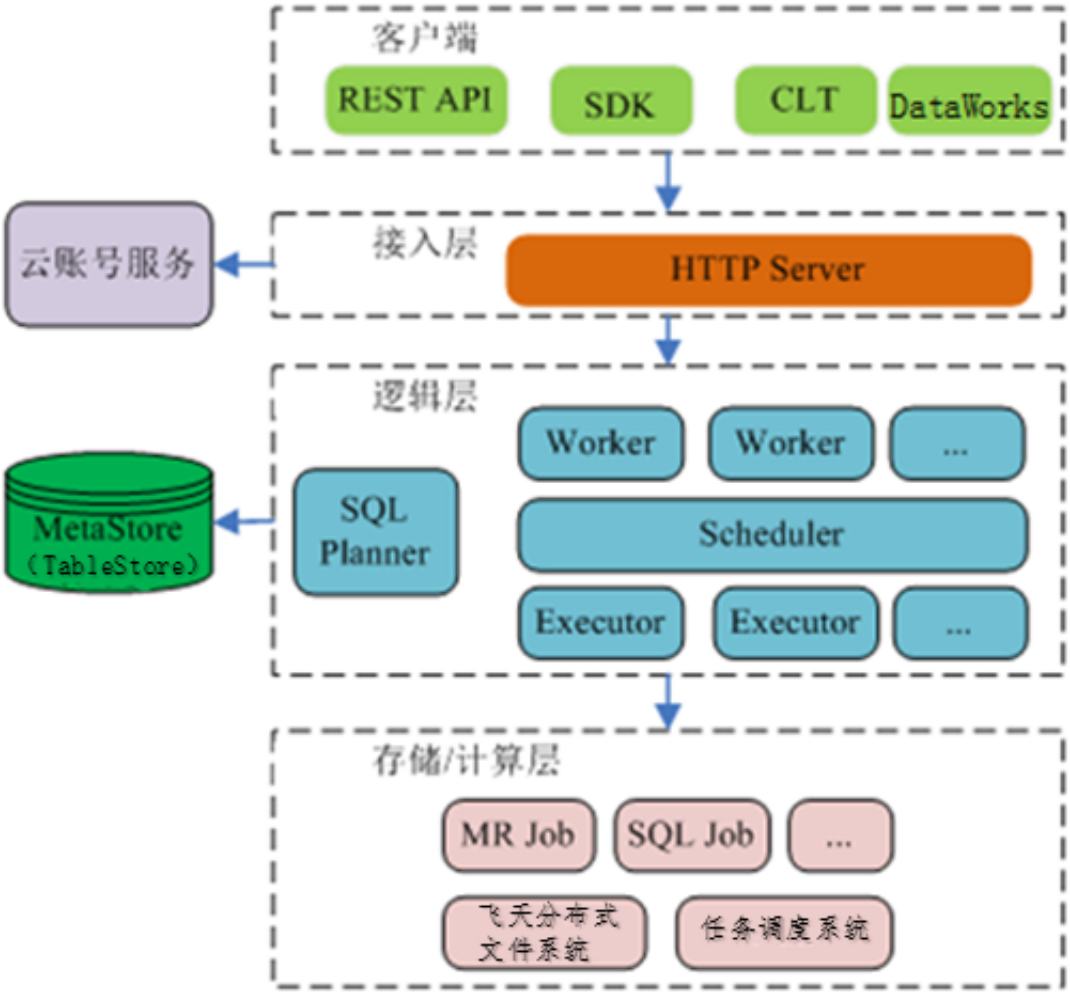
多用户协作的多租户机制

通过配置不同的数据访问策略，用户可以让组织中的多名数据分析师协同工作，并且每人仅能访问自己权限许可内的数据，在保障数据安全的前提下最大化工作效率。

4.3 产品架构

MaxCompute的产品架构如图 *4-1: MaxCompute*产品架构图所示。

图 4-1: MaxCompute产品架构图



MaxCompute由四部分组成，分别是**客户端**、**接入层**、**逻辑层**及**计算层**，每一层均可平行扩展。

MaxCompute的客户端有以下几种形式：

- **API**：以RESTful API的方式提供离线数据处理服务。
- **SDK**：对RESTful API的封装，目前有Java等版本的实现。
- **CLT**（**Command Line Tool**）：运行在Window/Linux下的客户端工具，通过CLT可以提交命令完成Project管理、DDL、DML等操作。
- **DataWorks**：提供了上层可视化ETL/BI工具，用户可以基于DataWorks完成数据同步、任务调度、报表生成等常见操作。

MaxCompute接入层提供HTTP（HTTPS）服务、Load Balance、用户认证和服务层面的访问控制。

MaxCompute逻辑层是核心部分，实现用户空间和对象的管理、命令的解析与执行逻辑、数据对象的访问控制与授权等功能。逻辑层包括两个集群：调度集群与计算集群。调度集群主要负责用户空间和对象的管理、Query和命令的解析与启动、数据对象的访问控制与授权等功能；计算集群主要负责task的执行。控制集群和计算集群均可根据规模平行扩展。在调度集群中有Worker、Scheduler和Executor三个角色，其中：

- **Worker处理所有RESTful请求**：包括用户空间（project）管理操作、资源（resource）管理操作、作业管理等，对于SQL、MapReduce、Graph等启动Fuxi任务的作业，会提交Scheduler进一步处理。
- **Scheduler负责instance的调度**：包括将instance分解为task、对等待提交的task进行排序、以及向计算集群的Fuxi master询问资源占用情况以进行流控（Fuxi slot满的时候，停止响应Executor的task申请）。
- **Executor负责启动SQL/MR task**：向计算集群的Fuxi master提交Fuxi任务，并监控这些任务的运行。

简单的说，当用户提交一个作业请求时，接入层的Web服务器查询获取已注册的Worker的IP地址，并随机选择某些Worker发送API请求。Worker将请求发送给Scheduler，由其负责调度和流控。Executor会主动轮询Scheduler的队列，若资源满足条件，则开始执行任务，并将任务执行状态反馈给Scheduler。

MaxCompute存储与计算层为阿里云自主知识产权的云计算平台的核心构件，图中仅列出了若干主要模块。

4.4 功能特性

4.4.1 Tunnel

4.4.1.1 基本概念

Tunnel是MaxCompute提供的数据通道服务，各种异构数据源都可以通过Tunnel服务导入MaxCompute或从MaxCompute导出。它是MaxCompute数据对外的统一通道，提供高吞吐、持续稳定的服务。

Tunnel提供了Restful API接口，提供了Java SDK，可以方便用户编程。目前Tunnel仅支持表（不包括视图 View）数据的上传下载。

4.4.1.2 Tunnel特点

- 数据进出MaxCompute的通道。

- 高并发上传下载。
- 服务能力水平扩展。
- 可支持每天1P吞吐量。
- 分为批量及实时两种模式。
- 实时模式支持pub/sub（发布/订阅）模型。
- 基于MaxCompute Tunnel的工具有TT，CDP，Flume，Fluentd等。
- 支持对表的读写，不支持视图。
- 写表是追加（Append）模式。
- 并发以提高总体吞吐量。
- 避免频繁提交。
- 上传数据时，目标分区必须存在。
- 实时上传模式。

4.4.1.3 Tunnel数据上传下载

Tunnel命令

```
odps@ > tunnel upload log.txt test_project.test_table/p1="b1",p2="b2
 “;
```

```
odps@ > tunnel download test_project.test_table/p1="b1",p2="b2" log.
txt;
```

使用说明

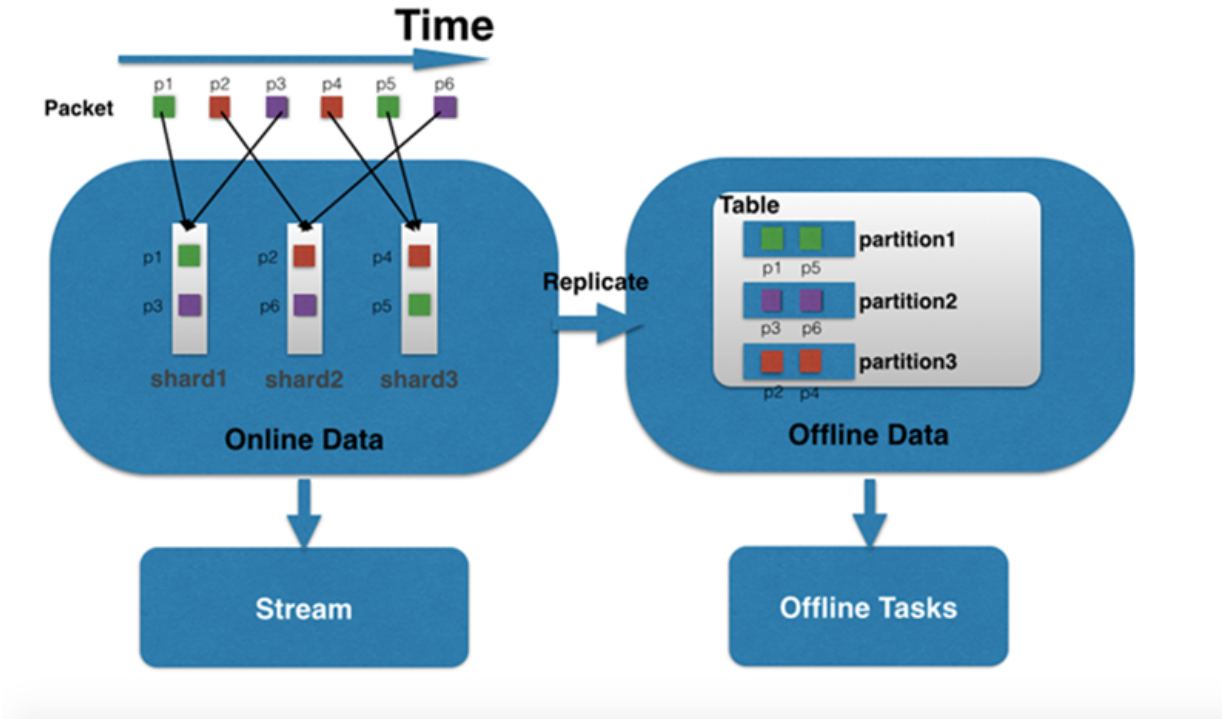
- 基于Tunnel SDK的一个命令行工具，可用于本地文本文件上传到MaxCompute或下载表数据到本地。
- 表的分区要先建好。
- DataX、CDP、TT等已基于Tunnel实现了更为完善的工具，可用于支持MaxCompute与关系数据库的数据交互。
- 日志数据可以使用Flume、Fluentd工具导入。
- 特殊的场景用户可以基于Tunnel实现自定义的工具。

实时上传

- 小batch上传。
- 高QPS。

- 毫秒级latency。
- 可订阅。

图 4-2: 实时上传



4.4.2 SQL

4.4.2.1 基本概念

MaxCompute SQL采用的是类似于SQL的语法，可以看作是标准SQL的子集，但不能因此简单的把MaxCompute SQL等价成 一个数据库，它在很多方面并不具备数据库的特征，如事务、主键约束、索引等。目前在MaxCompute中允许的最大SQL长度是2MB。

MaxCompute SQL离线计算模式适用于海量数据（TB级别），实时性要求不高的场合。它的每个作业的准备、提交等阶段均需要花费较长时间，因此要求每秒处理几千至数万笔事务的业务是不能用MaxCompute SQL完成的。准实时场景可以使用MaxCompute SQL的在线计算模式处理。

4.4.2.2 SQL特点

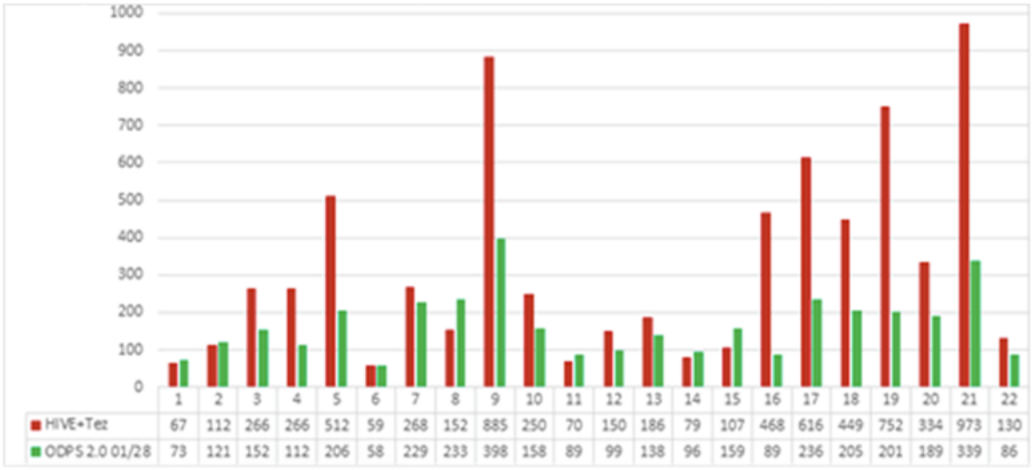
- 适用于大数据量处理（T级别到P级别）。
- 延迟比较高，每个SQL的运行时间在几十秒到小时级别。
- 语法类似Hive的HQL，是在标准SQL的基础上有所扩展。

- 没有事务，没有主键。
- 不支持UPDATE、DELETE。

4.4.2.3 与开源对比

- TPCH 1 TB 数据，MaxCompute与Hive（Apache-hive-1.2.1-bin +TEZ-ui-0.70. with CBO）相比， MaxCompute提升95.6%。

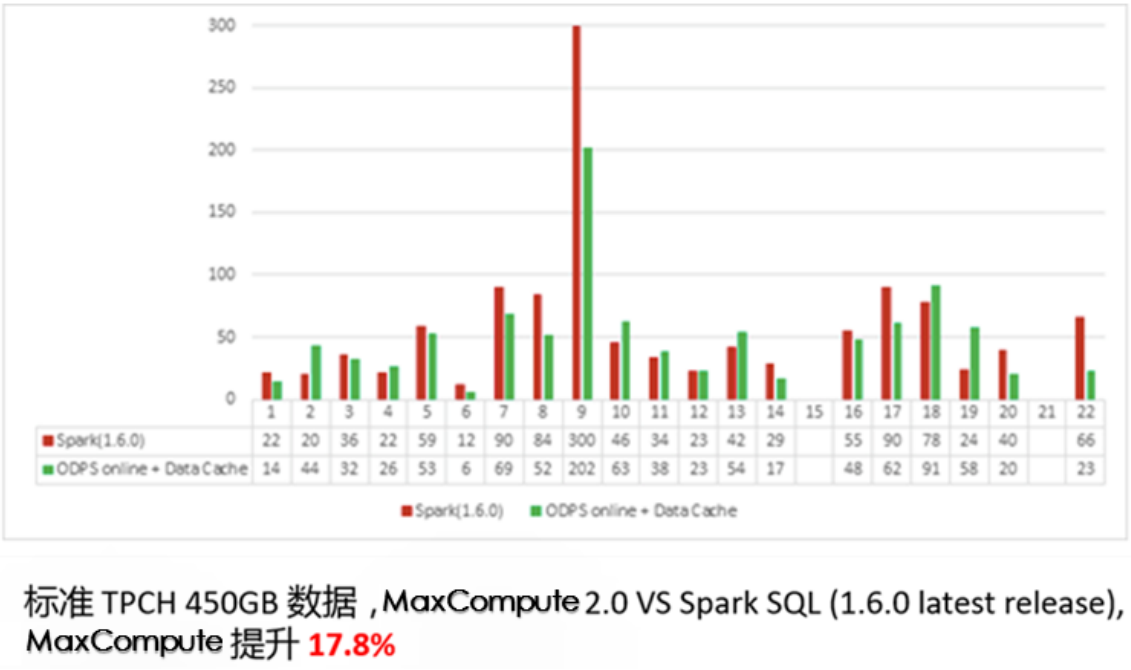
图 4-3: MaxCompute 2.0 VS Hive



TPCH 1 TB 数据，MaxCompute 2.0 VS Hive(Apache-hive-1.2.1-bin +TEZ-ui-0.70. with CBO)
MaxCompute提升95.6%

- 标准TPCH 450GB数据，MaxCompute与Spark SQL（1.6.0 latest release）相比，MaxCompute提升 17.8%。

图 4-4: MaxCompute 2.0 VS Spark SQL



4.4.3 MapReduce

4.4.3.1 基本概念

MapReduce是一种编程模型，基本等同Hadoop中的MapReduce。用于MaxCompute大规模数据集（TB级别）的并行运算。

MaxCompute提供了MapReduce编程接口。用户可以使用MapReduce提供的接口（Java API）编写MapReduce程序来处理MaxCompute中的数据。

说明：
由于MaxCompute的所有数据都被存放在表中，因此MaxCompute MapReduce的输入、输出只能是表，不允许用户自定义输出格式，不提供类似文件系统的接口。

4.4.3.2 MapReduce特点

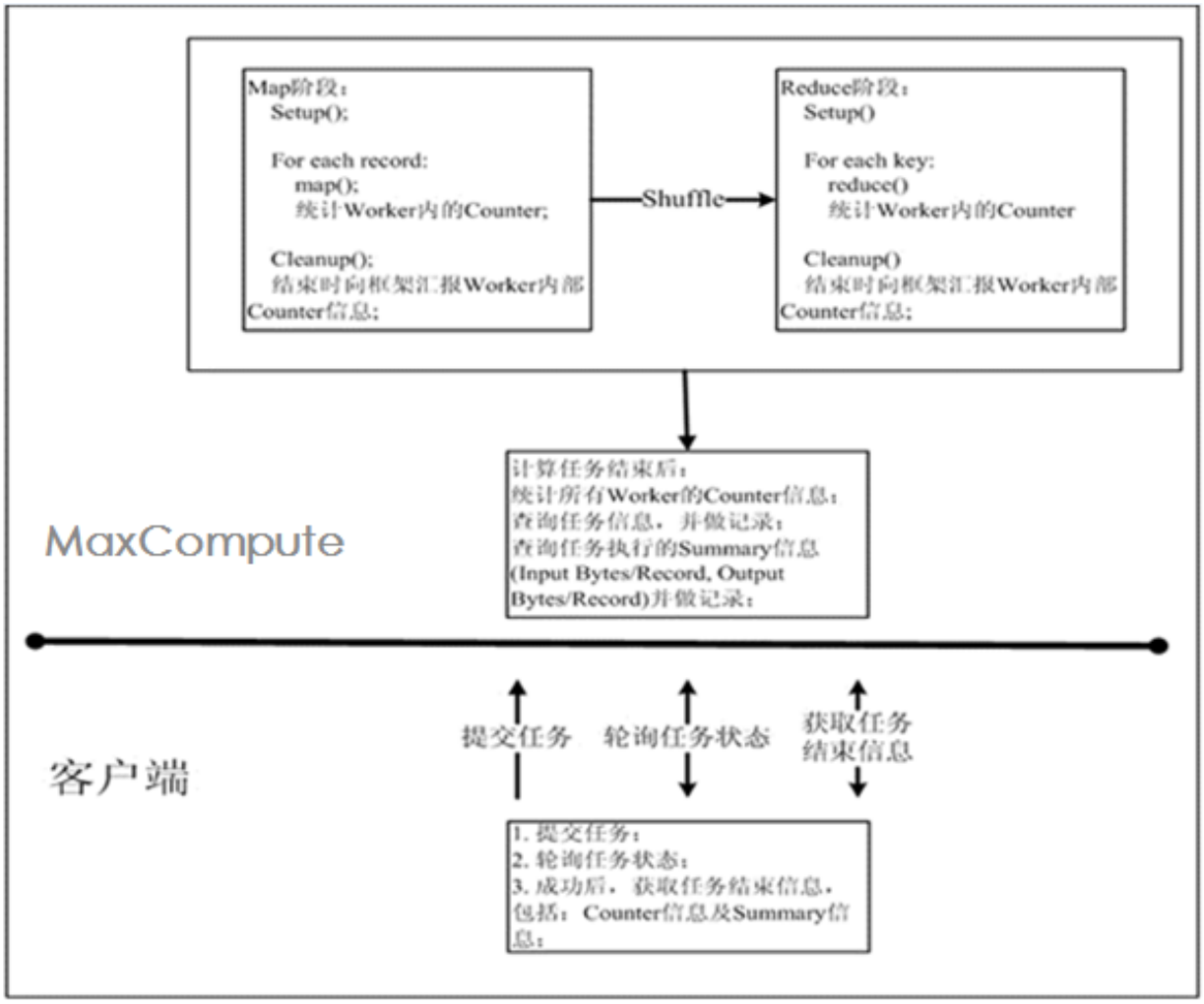
- 输入输出仅支持MaxCompute内置类型。
- 可以输入多表或输出多表到不同分区。
- 可以读资源（Resource）。
- 不支持输入view。
- 编译MR程序需要JDK 1.8环境。

- 受限的沙箱安全环境。

4.4.3.3 MaxCompute MR过程

MaxCompute MR的过程如下图所示。

图 4-5: MR过程



4.4.3.4 Hadoop MR VS MaxCompute MR

Hadoop MR与MaxCompute MR的对比如下表所示。

表 4-1: Mapper/Reducer

Mapper/Reducer	
Hadoop MapReduce	MaxCompute MapReduce

map（InKey key,InputValue value,OutputCollector<OutKey,OutValue> output,Reporter reporter）	map（long key,Record record,TaskContext context）
Reduce（InKey key,Iterator<InValue> values,OutputCollector<OutKey,OutValue> output,Reporter reporter）	reduce（lRecord key,Iterator<Record> values,TaskContext context）

图 4-6: MR

```
@Override
public void map(long recordNum, Record record, TaskContext context)
    throws IOException {
  for (int i = 0; i < record.getColumnCount(); i++) {
    word.set(new Object[] { record.get(i).toString() });
    context.write(word, one);
  }
```

4.4.4 Graph

4.4.4.1 基本概念

Graph是MaxCompute提供的面向迭代的图计算处理框架，为用户提供类似Pregel的编程接口，用户可以基于Graph框架提供的接口Java SDK开发高效的机器学习或数据挖掘算法。

图计算作业使用图进行建模，通过迭代对图进行编辑、演化，最终求解出结果。

4.4.4.2 Graph特点

- 图计算编程模型（类似Google Pregel）。
- 数据装载到内存，在迭代次数较多时优势明显。
- 可用于开发机器学习算法。
- 可以支持100亿顶点和1500亿边的规模。
- 典型应用。
  - Pagerank。
  - K-Means聚类。
  - 一度、二度关系，最短路径等。
- Graph作业处理数据是一个图。
- 原始数据存储在Table中，用户自定义的GraphLoader将Table中的数据加载为点和边。
- 迭代计算。

4.4.4.3 Graph关系网络模型

关系网络引擎为关系型数据的挖掘提供了多种针对业务优化的关系网络模型，帮助用户快速实现对关系网络数据的复杂挖掘。

社区发现

- 引擎输入：关系数据。
- 引擎输出：ID、社区ID。
- 计算逻辑：利用全局网络连接最优找到N个社区，社区内部足够紧密，社区间足够稀疏。

半监督分类

- 引擎输入：问题ID。
- 引擎输出：疑似问题ID、权重。
- 计算逻辑：利用已有问题ID（某一类或多类），根据整个网络连接关系判断全局中疑似此类（或多类）的ID以及权重。

孤立点检测

- 引擎输入：关系数据。
- 引擎输出：孤立点ID、权重。
- 计算逻辑：利用关系网络中连接关系判断是否有相对孤立的节点并输出。

关键点挖掘

- 引擎输入：关系数据。
- 引擎输出：关键点ID、类别。
- 计算逻辑：利用关系网络中连接关系计算网络中关键类型节点（中间度、影响力、媒介能力）。

N度关系

- 引擎输入：关系数据。
- 引擎输出：可检索的关系网络。
- 计算逻辑：利用关系网络中连接关系整理多维关系，并建立索引方便查询某ID的具体关联情况。

4.4.5 云产品联通

4.4.5.1 非结构化数据的访问与处理

MaxCompute团队依托MaxCompute系统架构，引入非结构化数据处理框架，解决MaxCompute SQL面对MaxCompute表外的各种用户数据时（例如：OSS上的非结构化数据或者来自TableStore

的非结构化数据），需要首先通过各种工具导入MaxCompute表，才能在其上面进行计算，无法直接处理的问题。

用户只需要通过一条简单的DDL语句，在MaxCompute上创建一张外部表，建立MaxCompute表与外部数据源的关联，即可以为各种数据在MaxCompute上的计算处理提供入口。并且创建好的外部表可以像普通的MaxCompute表一样使用，充分利用MaxCompute SQL的强大计算功能。

4.4.6 增强特性

4.4.6.1 Spark on MaxCompute

4.4.6.1.1 基本概念

Spark on Maxcompute是阿里云开发的基于Maxcompute平台无缝运行使用spark的增强方案，其很大程度上丰富了Maxcompute产品的功能体系。

Spark on MaxCompute为用户提供原生Spark的使用体验以及Spark原生的组件及API；提供存取MaxCompute数据源的能力；提供多租户场景更好的安全能力；提供管控平台，让Spark作业与MaxCompute作业共享资源、存储、用户体系，保证高性能和低成本。 Spark配合MaxCompute产品能够构建更加完善高效的数据处理解决方案，同时社区Spark的应用可以无缝的运行在**Spark on MaxCompute**之上。

4.4.6.1.2 Spark特点

处理MaxCompute以及非结构化数据源

- Scala、Python、Java、R语言API支持处理MaxCompute的Table。
- Spark sql、Spark Mllib、Graphx、Streaming组件支持处理MaxCompute的Table。
- 非结构支持处理阿里云OSS的数据。

友好的使用体验及管控

- 通过兼容Yarn以及HDFS API，使得**Spark on MaxCompute**具有社区Spark on Yarn原生的提交方式。
- Spark sql、Spark Mllib、Graphx、Streaming组件全部支持。
- Spark可以与MaxCompute的Sql、Graph等组件共用构建解决方案。
- 可以访问原生Spark的UI。
- 可以直接使用MaxCompute强大的管控能力。
- 在支持社区Spark之外，还提供Client模式、livy、HUE等附加的工具。

扩展性

Spark与MaxCompute共享集群资源，可以享受MaxCompute大集群的扩容能力。

4.4.6.1.3 Spark功能

Spark on MaxCompute的产品功能如下所示。

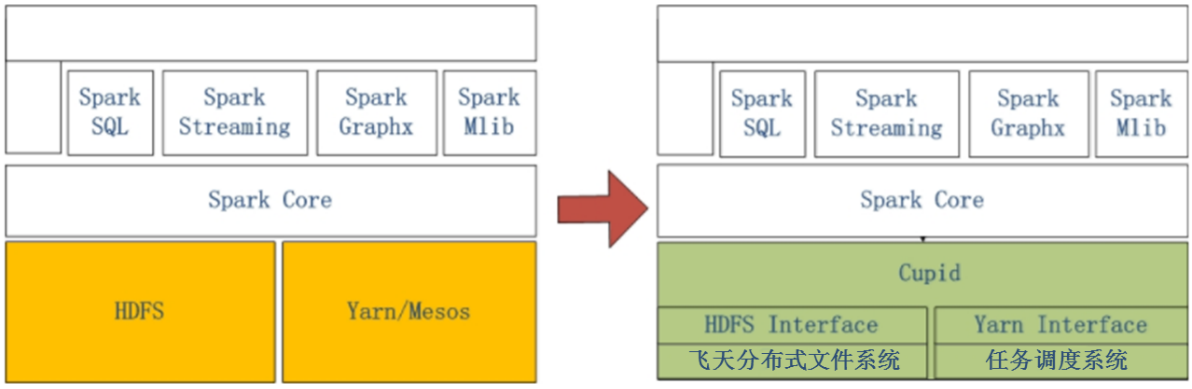
表 4-2: 产品功能列表

分类	特性	说明
分布式集群	集群部署 集群监控	提供运维平台页面监控集群、节点情况。
数据处理组件	可使用Spark Sql、Mllib、Graphx、Streaming组件	提供原生Spark的组件。
作业管控	资源统一管理、生命周期管理、权限验证	通过兼容Yarn API来实现。
数据源	非结构化数据 MaxCompute的table数据源	具有MaxCompute上sql以及MapReduce同样的数据处理性能。
安全管理	用户身份认证、数据鉴权、多租户作业隔离	通过权限认证以及安全沙箱加强Spark的安全能力。

4.4.6.1.4 Spark架构

Spark on MaxCompute与**原生Spark**的架构图比对如下所示。

图 4-7: Spark on MaxCompute与原生Spark架构比对



说明：

左侧是原生spark的架构图，右侧是**Spark on MaxCompute**的架构图。

从上图可以看出，整个产品包含Spark原生的计算能力以及相关的管控平台、运维平台、调度系统、安全能力、数据互通。在管控上面Spark是通过MaxCompute的一个CupidTask实例拉起；资源申请是MaxCompute提供一层Yarn API的兼容；安全方面使用MaxCompute的沙箱能力；数据处理方面打通数据和元数据。具体的模块介绍如下：

- MaxCompute控制集群使用CupidTask实例拉起Spark的Driver，后续Driver通过Yarn API的兼容层去向统一资源管理器FuxiMaster申请资源。
- MaxCompute控制集群管理Spark运行实例消耗的用户quota、Spark实例的生命周期、可访问数据源的权限管理。
- 在MaxCompute的计算集群通过父子进程的方式来拉起Spark的Driver以及Executor，并且让Spark的代码运行在MaxCompute的安全沙箱中，从而保证多租户场景的安全性。
- 系统通过MaxCompute的Proxyserver来提供Spark原生的UI界面，同时用户可以使用MaxCompute相关管控组件来管理作业信息。

4.4.6.1.5 Spark优势

支持Spark完整生态

使用体验跟开源保持一致。

跟MaxCompute完全整合

做到资源统一、数据统一和安全统一。

兼有Spark和飞天操作系统的优势

既有Spark的灵活性、易用性；又有飞天操作系统的高可用、高扩展性和稳定性。

支持多租户机制

既能保证物理机器的高性能，也能实现大集群的统一资源调度，从而降低成本。

4.5 应用场景

本章节将简要的介绍一些MaxCompute的典型应用场景。

4.5.1 场景一：使用成本低，数据上云周期短

使用场景：针对某从事新能源电力领域的数据信息服务公司的业务需求，搭建新能源产业互联网大数据应用服务云平台。

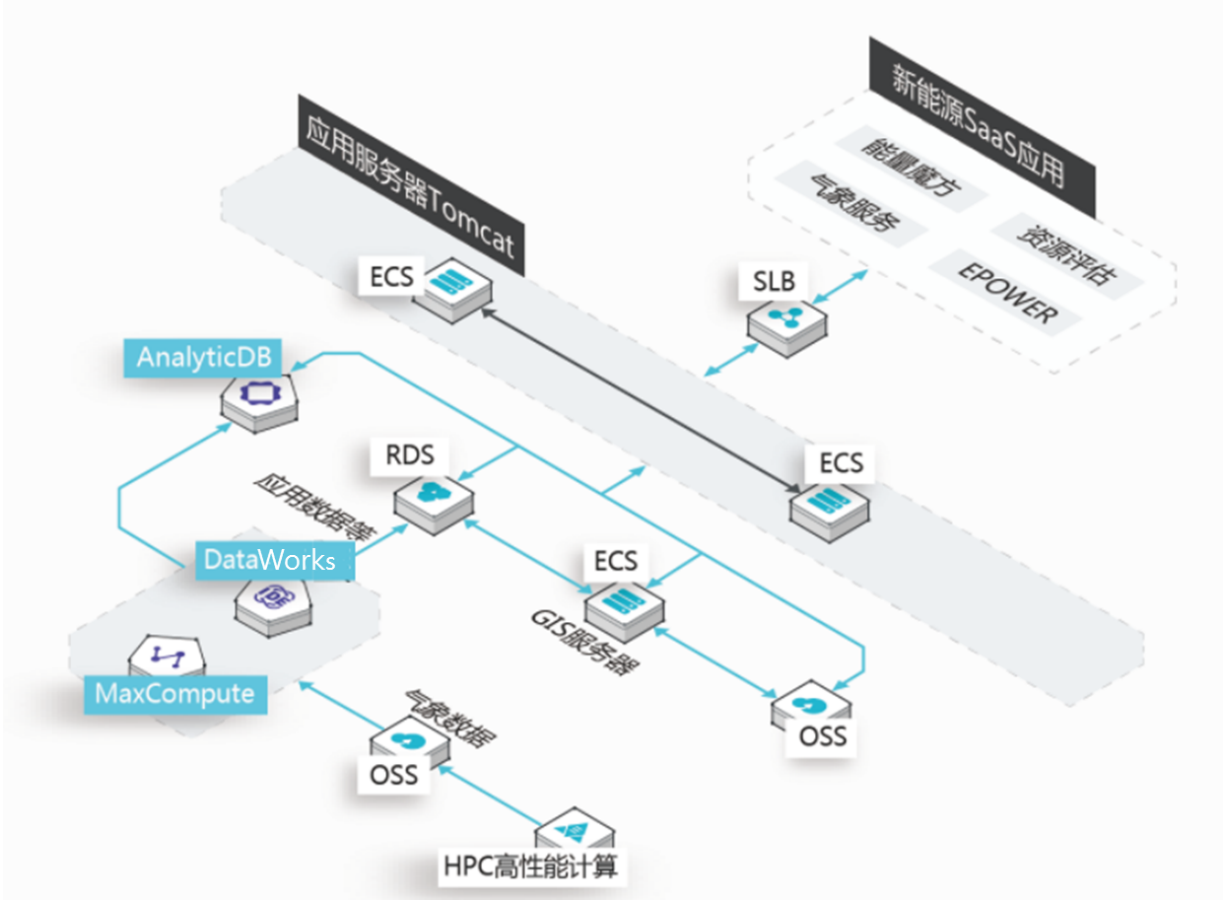
目标达成：3个月内业务全面交付云端，数据处理时间不到原来自建方式的1/3，并确保云上新能源电力数据安全无忧。

客户收益：

- **让企业更专注于业务**：用了不到3个月时间，就将业务全面的交付云端，让云端的海量资源真正为业务服务。
- **降低投资、运维成本**：极大减少了自建大数据平台的物力投入、人力运维投入和研发投入。
- **安全稳定**：全方位服务能力及其稳定安全的表现确保数据上云万无一失。

应用架构图：

图 4-8: 应用架构图



架构解读：使用MaxCompute、分析型数据库、DataWorks、SLB、ECS、OSS、RDS及HPC搭建云端平台。

- 使用MaxCompute进行大数据计算和分析。
- 使用分析型数据库（AnalyticDB，原名ADS）存放上亿条记录级数据，支持业务实时数据访问及展示。
- 使用DataWorks进行数据同步、数据开发、离线任务调度运维等。
- 使用阿里云SLB负载均衡，实现用户终端实时高性能接入。
- 使用阿里云ECS部署Web应用、地图服务等应用。
- 使用阿里云OSS对象存储进行海量气象数据、地图文件存储。
- 使用阿里云RDS数据库存储业务应用、地图应用数据。
- 使用阿里云HPC高性能计算进行气象数据计算。

4.5.2 场景二：提升开发效率，降低存储和计算成本

使用场景：针对某以“做卓越的天气服务公司”为目标的新兴移动互联网公司的业务需求，提供天气查询业务和广告业务的海量日志分析服务。

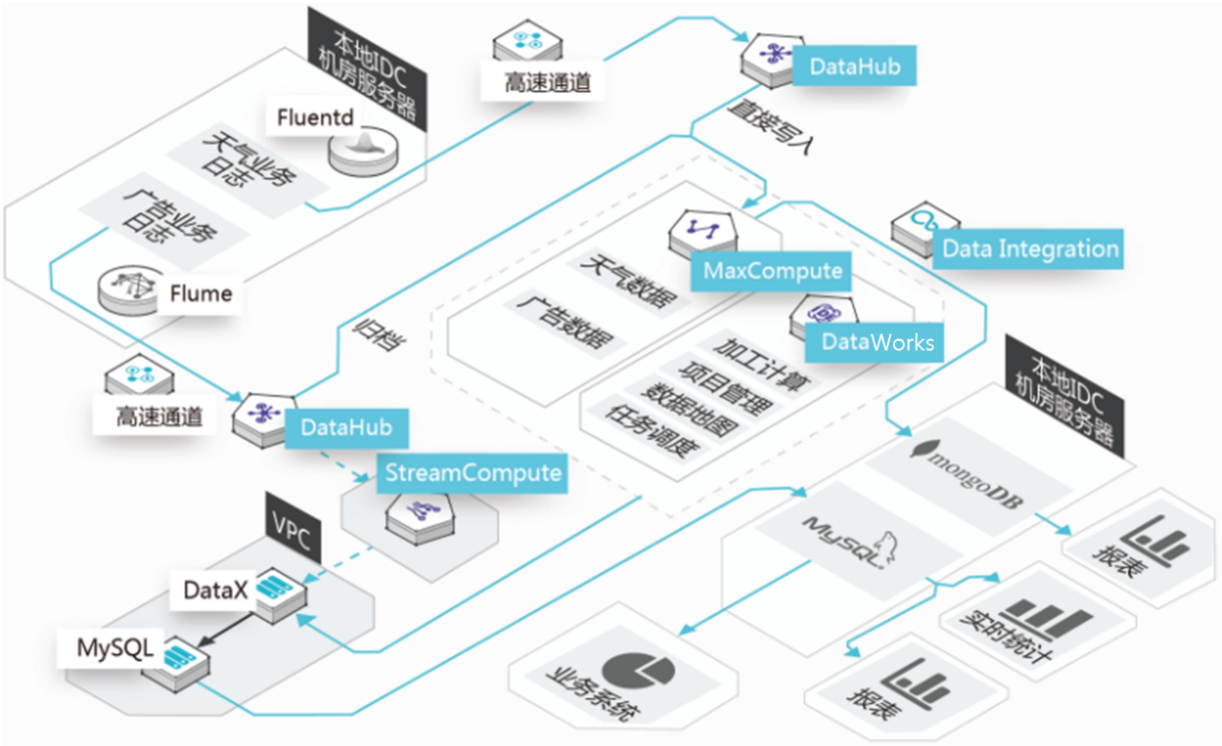
目标达成：该互联网公司日志分析业务迁移到MaxCompute后，开发效率提升了超过5倍，存储和计算费用节省了70%，每天处理分析2TB的日志数据，更高效的赋能其个性化运营策略。

客户收益：

- **提高工作效率**：日志数据全部通过SQL进行分析，工作效率提升了5倍以上。
- **提升存储利用率**：整体存储和计算的费用比之前节省70%，性能和稳定性也有提升。
- **个性化的服务**：可以借助MaxCompute上的机器学习算法，对数据进行深度挖掘，为用户提供个性化的服务。
- **降低大数据使用门槛**：MaxCompute提供多种开源软件的插件，轻松完成数据上云。

应用架构图：

图 4-9: 应用架构图



架构解读：天气查询业务日志分析如架构图上半部分所示，广告业务日志分析如架构图下半部分所示。

4.5.3 场景三：盘活海量数据，实现百万用户精细化运营

使用场景：针对某专注美甲行业的社区型垂直电商APP的业务需求，使用MaxCompute为其搭建大数据平台，主要应用在其业务监控、业务分析、精细化运营和推荐四个方面。

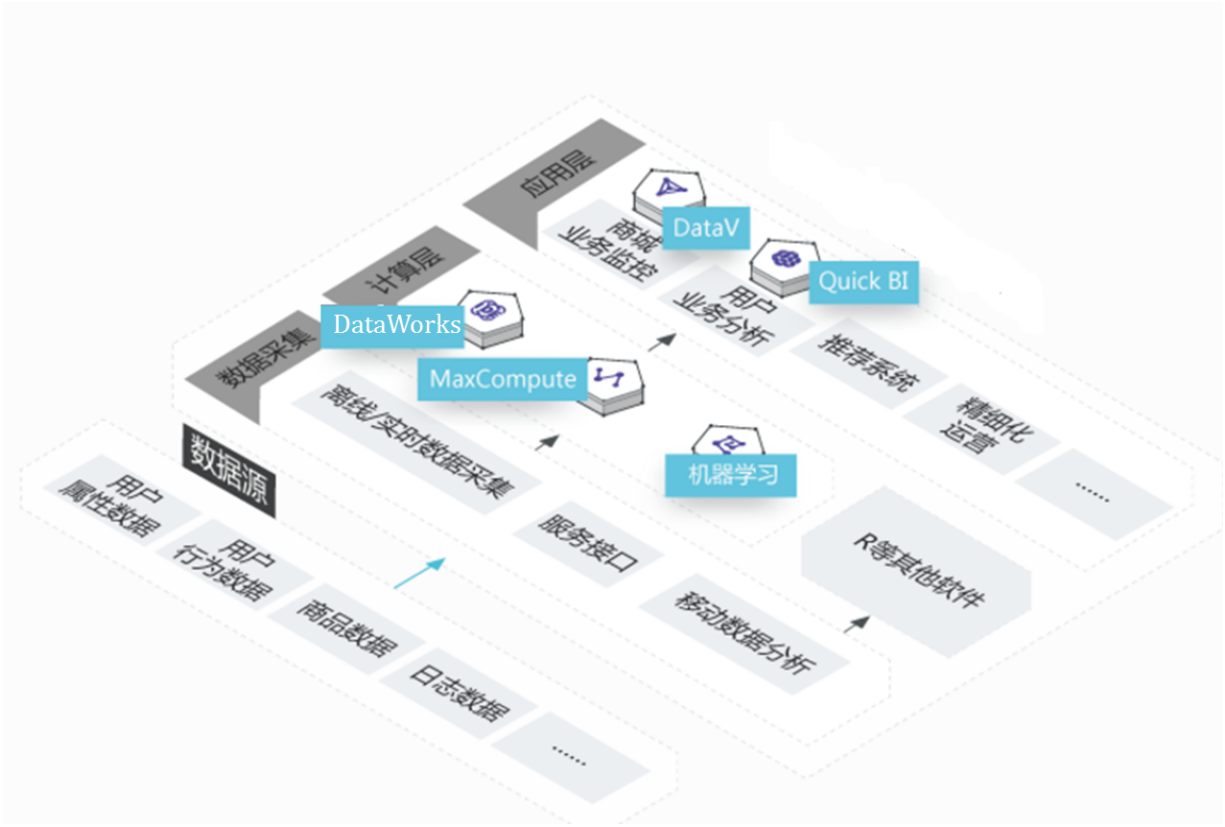
目标达成：该电商APP使用MaxCompute搭建的大数据平台后，通过MaxCompute的计算能力实现了针对百万用户的精细运营，业务上做到了更敏捷、更智能、更具洞察力，并且能够快速响应新业务的数据及分析需求。

客户收益：

- **提升业务洞察能力**：通过MaxCompute计算能力实现了针对百万用户的精细化运营。
- **业务数据化**：对业务数据分析能力提升并有效监控，更好的业务赋能。
- **快速响应业务需求**：MaxCompute生态满足新业务数据分析需求的“随机应变”能力。

应用架构图：

图 4-10: 应用架构图



架构解读：整体架构分为数据采集层、计算层以及应用层。

- 数据采集层-数据采集、清洗、处理：数据源主要包括云数据库RDS、移动数据分析（Mobile Analytics）日志、服务接口调用的数据。以精细化运营为例，用户属性数据存放于RDS，用户行为数据来源于移动数据分析的日志数据。使用大数据开发套件DataWorks把分布在多个数据源的数据集合一起，进行清洗和加工。
- 计算层-数据分析挖掘：使用大数据开发套件DataWorks的定时任务调度功能，自动完成计算任务并将结果同步回传到数据库；IDE、机器学习以及R等工具主要解决具体的业务分析；MaxCompute用于海量数据的存储和计算引擎。
- 应用层-实际应用：使用DataV制作业务看板进行实时业务监控；推荐系统用于其个性化业务的个性化推荐；Quick BI用于业务分析；精细化运营用于用户洞察及精准营销。

4.5.4 场景四：大数据精准营销

使用场景：针对某以精准营销广告技术与服务为长的互联网企业的业务需求，为其搭建核心的大数据精准营销平台。

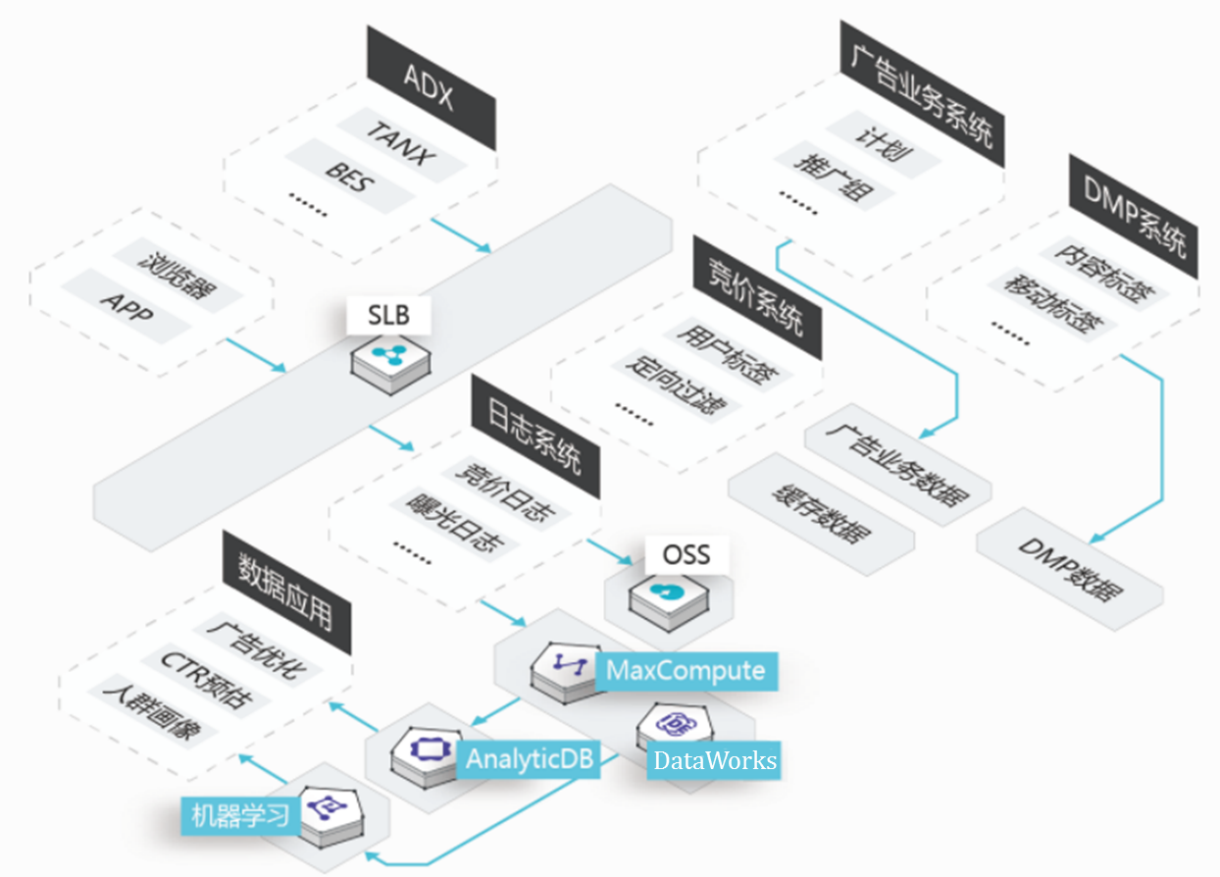
目标达成：该企业基于MaxCompute，搭建了核心的大数据精准营销平台，所有的日志数据存储在MaxCompute并通过DataWorks进行离线调度和分析。

客户收益：

- **高效低成本的海量数据分析**：对海量日志数据进行统计分析，在满足同等业务需求基础上能够减少一半的支出，有效地节约了成本开支，帮助创业型企业快速成长。
- **数据查询分析的实时性**：MaxCompute帮助该企业确立技术优势，打破了海量数据处理分析和实时查询分析的技术瓶颈，每天通过MaxCompute收集、分析和存储20多亿条的访客行为；同时，还会根据用户需求在亿级日志表中做毫秒级查询。
- **低门槛的机器学习平台**：作为精准营销广告提供商，算法模型的好坏直接与最终收益挂钩，因此选择具有低门槛、易上手特点的MaxCompute的机器学习平台可以起到事半功倍的效果。

应用架构图：

图 4-11: 应用架构图



架构解读：利用MaxCompute存储日志数据并通过DataWorks进行离线调度和分析，利用分析型数据库实现数据的实时查询和计算处理，同时通过学习机器完成开源到MaxCompute的迁移。

- 日志数据全部存储在大数据计算服务MaxCompute中。
- 大部分离线统计需求都在大数据开发套件DataWorks中开发，将数据使用做到极简，只要使用者会写SQL，就可以制作并导出自己需要的报表，满足了公司大部分的业务需求。
- 分析型数据库能够满足在亿级数据中做毫秒级查询，在即席查询及数据分析方面，能够满足数据实时计算处理的需求。
- 通过机器学习组件，逐步将开源大数据平台中的机器学习相关业务应用迁移到基于MaxCompute的机器学习平台之上。

4.6 使用限制

无。

4.7 基本概念

项目空间

项目空间是MaxCompute的基本组织单元，它类似于传统数据库的Database或Schema的概念，是进行多用户隔离和访问控制的主要边界，一个用户可以同时拥有多个项目空间的权限。

> **说明**：
> 通过安全授权，可以在一个项目空间中访问另一个项目空间中的对象，例如：表（Table）、资源（Resource）、函数（Function）、实例（Instance）。

表

表是MaxCompute的数据存储单元，它在逻辑上也是由行和列组成的二维结构，每行代表一条记录，每列表示相同数据类型的一个字段，一条记录可以包含一个或多个列，各个列的名称和类型构成这张表的Schema 。

> **说明**：
> MaxCompute的表格分两种类型：外部表及内部表。

分区表

分区表指的是在创建表时指定分区空间，即指定表内的某几个字段作为分区列。在使用数据时如果指定了需要访问的分区名称，则只会读取相应的分区，避免全表扫描，提高处理效率，降低费用。

数据类型

MaxCompute表中的列必须是下列描述的任意一种类型：Tinyint、Smallint、Int、Bigint、String、Float、Boolean、Double、Datetime、Decimal、Varchar、Binary、Timestamp、Array、Map、Struct。

资源

资源（Resource）是MaxCompute的特有概念。用户如果想使用MaxCompute的自定义函数（UDF）或MapReduce功能，需要依赖资源来完成。

说明：
MaxCompute资源的类型包括：File类型、Table类型（MaxCompute中的表）、Jar类型（编译好的Java Jar包）及Archive类型（通过资源名称中的后缀识别压缩类型，支持的压缩文件类型包括：.zip/.tgz/.tar.gz/.tar/jar）。

函数

MaxCompute为用户提供了SQL计算功能，用户可以在MaxCompute SQL中使用系统的内建函数完成一定的计算和计数功能。但当内建函数无法满足要求时，用户可以使用MaxCompute提供的Java编程接口开发自定义函数（User Defined Function，简称UDF）。

说明：
自定义函数（UDF）又可以进一步分为标量值函数（UDF），自定义聚合函数（UDAF）和自定义表值函数（UDTF）三种。

任务

任务（Task）是MaxCompute的基本计算单元，SQL及MapReduce功能都是通过任务（Task）完成的。

任务实例

在MaxCompute中，部分任务在执行时会被实例化，以MaxCompute实例（简称实例或Instance）的形式存在。

资源配额

配额（Quota）分为存储和计算两种。对于存储配额，在MaxCompute中可以设置一个project中允许使用的存储上限，在接近上限到一定程度时会触发报警。对于计算配额，有内存和CPU两方面，即在project中同时运行的进程所占用的内存和CPU资源不可以超过指定的上限。

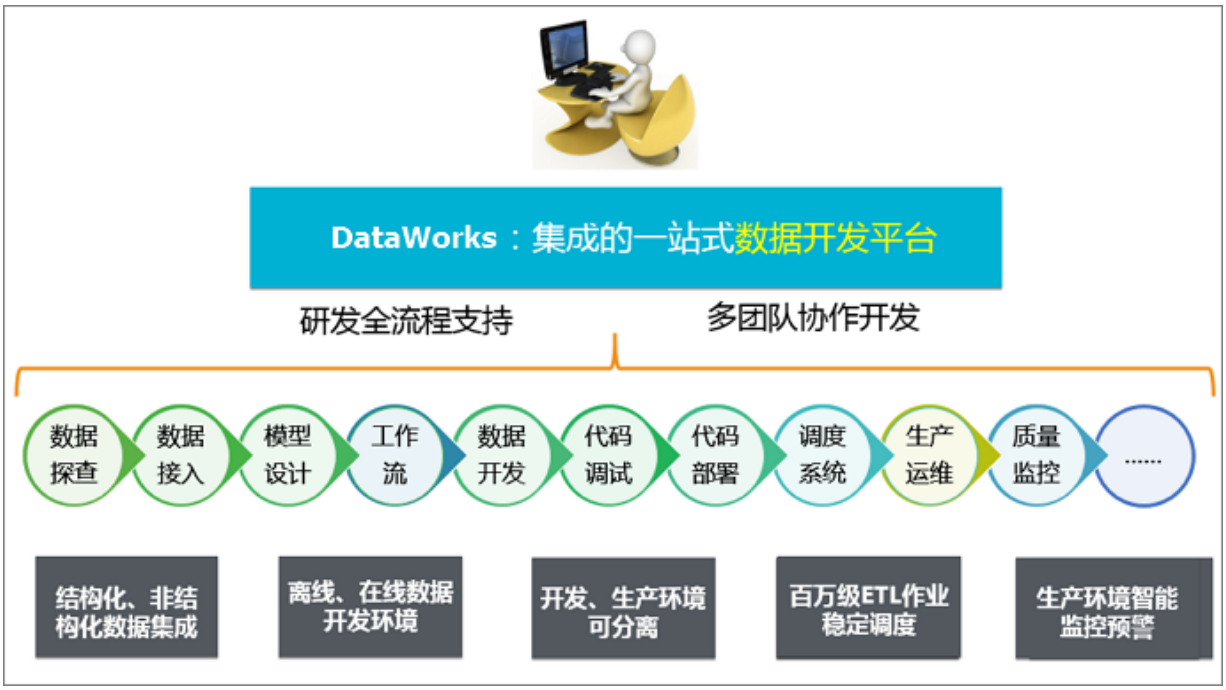
5 DataWorks

5.1 什么是DataWorks

DataWorks是阿里云数加重要的Paas平台产品，提供全面托管的工作流服务，一站式开发管理的界面，帮助政府部门、事业单位专注于数据价值的挖掘和探索。使用DataWorks，可对数据进行数据传输、数据转换等相关操作，从不同的数据存储引入数据，对数据进行转化处理，最后将数据提取到其他数据系统。

DataWorks是新一代一站式的大数据开发平台，是一个各类型生产车间齐备的大数据工场，它以MaxCompute为核心数据计算与存储引擎，集成了数据集成、数据开发、生产运维、实时分析、资产管理、数据质量、数据安全、数据共享等核心数据工艺，承上启下，让可视化数据开发成为可能，让组件化共享协作开发、启动数据研发的群殴模式，让业务容器减少数据干扰，让每一个开发者被中台航母的各项功能赋能。

图 5-1: DataWorks概述



5.2 产品优势

5.2.1 超大规模计算处理能力

数据资源平台与底层计算平台天然集成，轻松处理海量数据。

- 万亿级数据JOIN，百万级并发Job，作业I/O可达PB级/天。
- 离线调度支持百万级任务量，实时监控告警。
- 提供功能强大易用的SQL、MR引擎，兼容大部分标准SQL语法。
- 采用三重备份、读写请求鉴权、应用沙箱、系统沙箱等多层次数据存储和访问安全机制保护您的数据，确保不丢失、不泄露、不被窃取。

5.2.2 一站式的数据工场

- 一个平台，提供数据从集成、加工、管理、监控、输出服务的全流程所有功能。
- 提供可视化工作流程设计器功能。
- 多人协同作业机制，分角色进行任务开发、线上调度、运维、数据权限管理等功能，数据及任务无需落地即可完成复杂的操作流程。

5.2.3 海量异构数据源快速集成能力

支持400对异构数据源的离线同步，支持分钟级、小时级、天级等的调度。

5.2.4 Web化的软件服务

可在互联网/内部网络环境下直接使用，无需安装部署，拎包入驻，开箱即用。

5.2.5 多租户权限模型

多租户模型确保您的数据被安全隔离，以租户为单位进行统一的权限管控、数据管理、调度资源管理和成员管理工作。

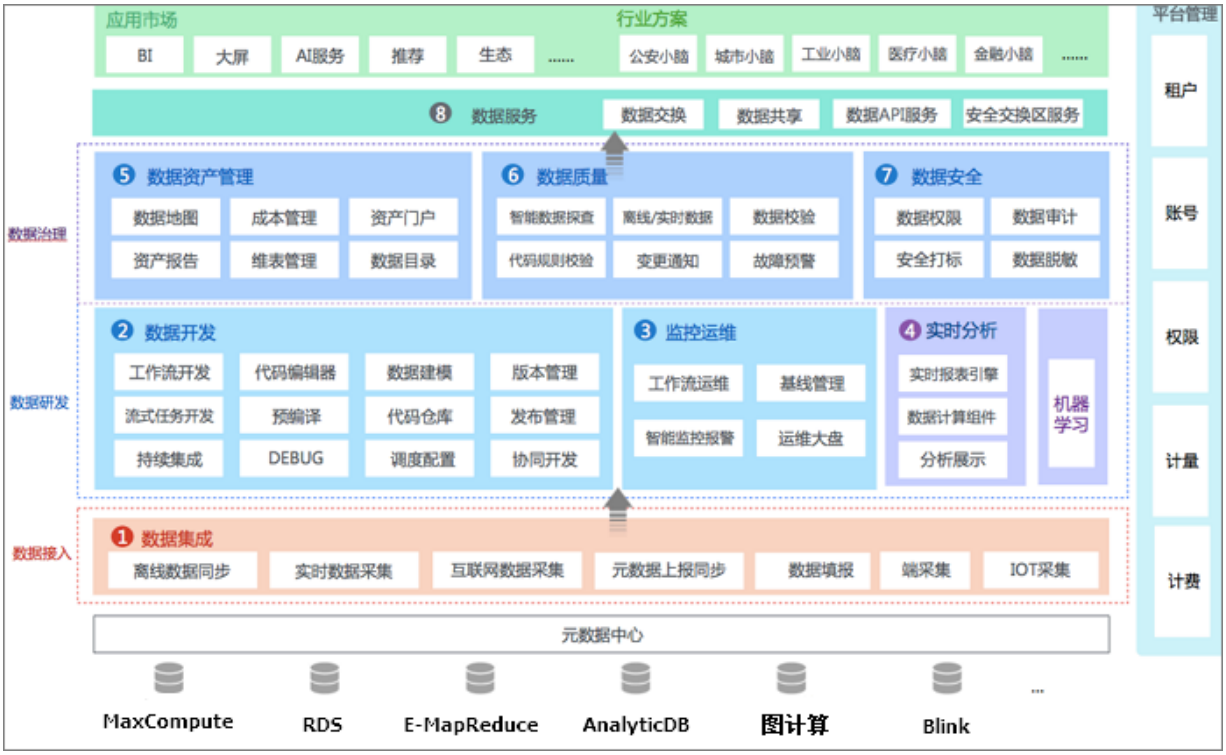
5.2.6 开放的平台

所有模块已实现组件化、服务化，您可基于DataWorks的Open API来定制开发扩展功能。

5.3 产品架构

5.3.1 功能架构

图 5-2: 功能架构图



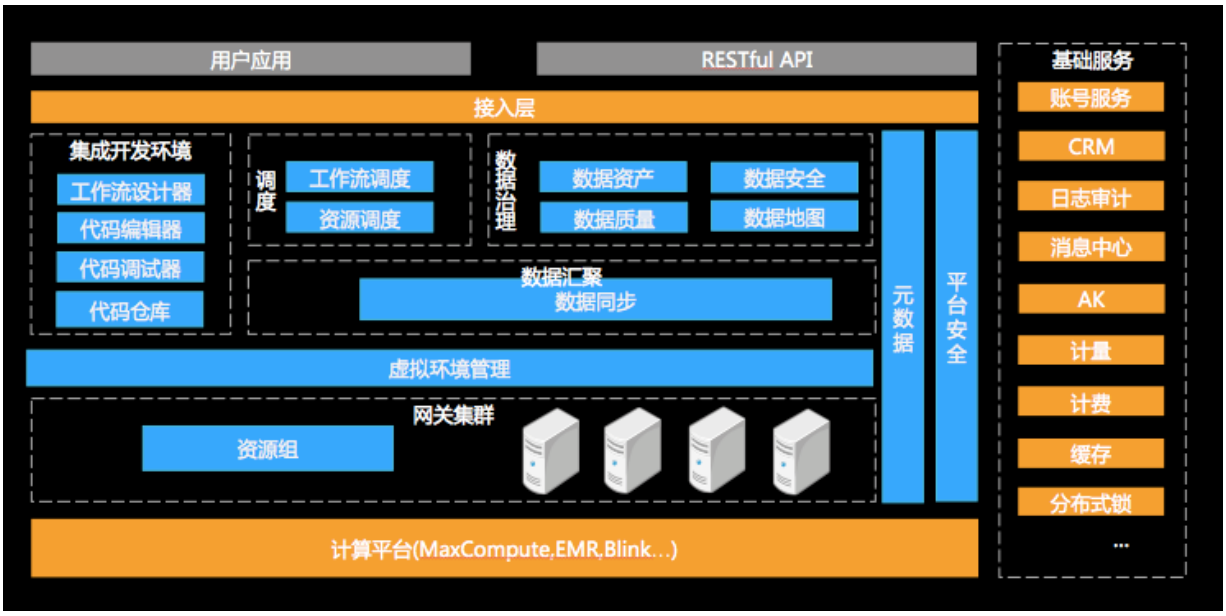
DataWorks提供以下八大服务模块：

- 数据集成：异构数据集成，将海量的数据从各种源系统汇集到大数据平台。
- 数据开发：数据仓库设计和ETL开发过程。
- 监控运维：ETL线上作业的运维监控。
- 实时分析：实时探查和分析数据。
- 数据资产管理：元数据管理、数据地图、数据血缘、数据资产大图等。
- 数据质量：数据质量探查、监控、校验和评分体系。
- 数据安全：数据权限管理，数据的分级打标、脱敏，以及数据审计。
- 数据服务：数据共享和数据交换，数据API服务。

5.3.2 系统架构

数据资源平台以数据为基础，以数据全链路加工流程为核心，提供数据汇聚、研发、治理、服务等多种功能，既可满足平台用户的数据需求，又能为上层应用提供各种行业解决方案。

图 5-3: 数据资源平台架构图



5.3.3 安全架构

数据资源平台的安全架构，由平台自身的安全实现层、平台内置的安全服务层和租户可选的安全产品层构成。

- 平台自身的安全实现层：保障平台在代码实现和部署配置时的产品自身安全性。
- 平台内置的安全服务层：为租户和其用户提供平台基础性的安全服务能力，如：租户资源隔离、身份认证、权限鉴别和日志合规审计等。
- 租户可选的安全产品层：为租户和其用户提供可选的、已集成的安全产品或工具，帮助租户根据其自行定义的安全策略对其拥有的系统、数据进行安全防护和运维管理。

5.3.4 多租户模型

数据资源平台拥有自己的多租户权限模型。

- 弹性的存储和计算资源，租户可按需申请资源配额，独立管理自己的资源。
- 租户独立管理自有的数据、权限、用户、角色，彼此隔离，以确保数据安全。

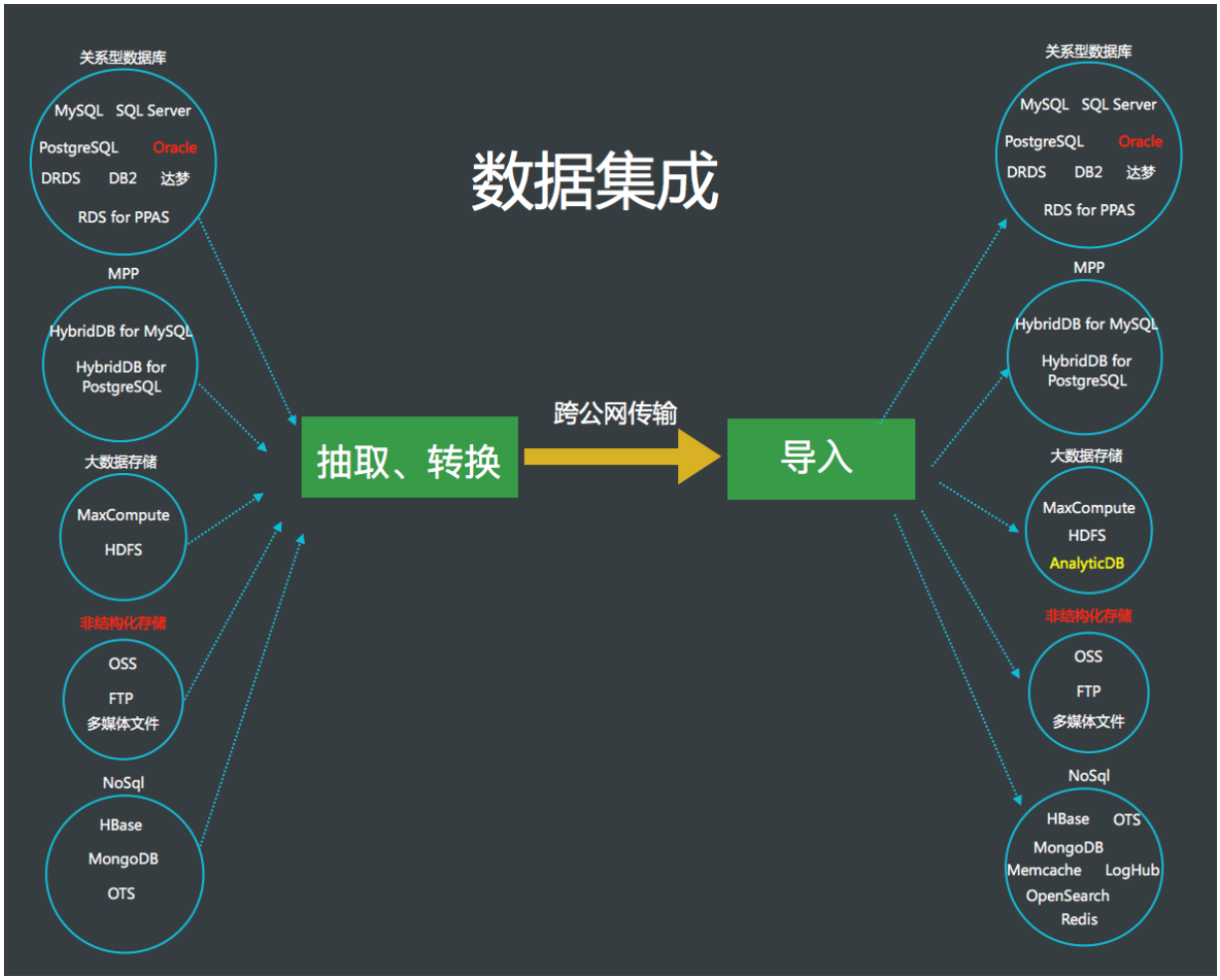
5.4 功能特性

5.4.1 数据集成

数据集成提供对业务方数据库进行抽取监控功能，能对数据源头的数据资源能够进行统一清点，并能够在复杂网络情况下对异构的数据源进行数据同步与集成，包括对关系型数据库、NoSQL数据

库、大数据数据库、文本存储（FTP）等数据库类型支持，支持离线数据的批量、全量、增量同步，支持分钟、天、小时、周、月来自定义同步时间。

图 5-4: 数据集成



5.4.1.1 支持多种数据通道

5.4.1.1.1 元数据信息同步

元数据信息是整个平台数据的基础，数据集成系统可从各个业务系统完成MySQL、SQLServer、Oracle、MaxCompute等20多种常见数据库的元数据信息的收集，避免对整个整体数据资产的情况不清楚，帮助数据管理者直接通过元数据进行后续资产的盘点及重点数据的同步。

5.4.1.1.2 关系型数据库同步服务

支持MySQL、SQLServer、Oracle、DRDS、PostgreSQL、DB2、RDS for PPAS等关系型数据库的读写操作。

5.4.1.1.3 NoSQL数据库同步服务

支持HBase、MongoDB、Table Store等NoSQL数据库的读写操作。

5.4.1.1.4 MPP数据库同步服务

支持HybridDB for MySQL、HybridDB for PostgreSQL等MPP数据库的读写操作。

5.4.1.1.5 大数据数据库同步服务

支持MaxCompute、HDFS的读写操作，并支持Analytic DB（ADS）的写操作。

5.4.1.1.6 非结构化存储同步服务

支持OSS、FTP的读写操作。

说明：
在数据集成中，数据源之间的数据传输是正交关系，目前已经支持400多对数据源的数据互传。

5.4.1.2 数据流入管控

支持各种数据类型的转换，精确识别脏数据，进行过滤、采集、展示，为您提供可靠的脏数据处理，让您准确把控数据流入内容，提供作业全链路的流量、数据量、脏数据探测和运行时汇报。

5.4.1.3 传输速度

单通道插件性能优化，可充分使用单机网卡能力，并使用分布式模型架构，可保障数据吞吐量水平扩展，能够提供GB级、TB级的数据流量。

5.4.1.4 控制友好

数据集成提供精准流控保证，支持通道、记录流、字节流三种流控模式，并提供完备的容错处理，支持线程级别、进程级别、作业级别多层次局部/全局的重跑。

5.4.1.5 同步插件

支持以插件的方式部署采集工具至数据源端的服务器，完成数据信息的同步采集工作。

5.4.1.6 跨网络传输

支持各种复杂网络环境下的数据传输，如本地跨私网环境、VPC环境、网闸环境等。

说明：
对长链路传输通过协议加速能更高效、稳定的传输大批量数据。

5.4.2 数据开发

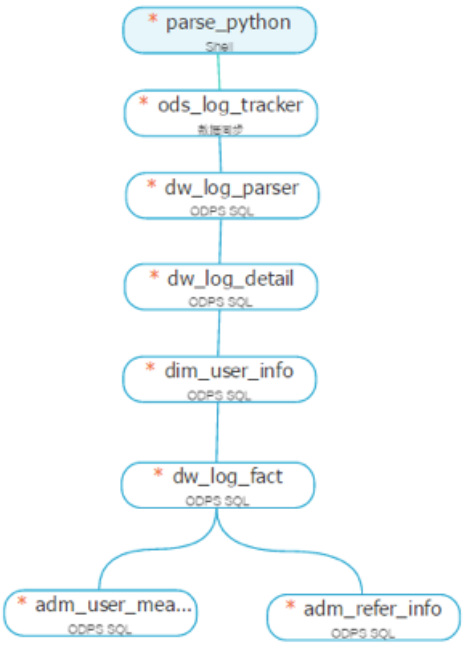
当底层数据进行聚合后，数据仍然出于零散的状态，数据是无法直接为上层智能算法和DI应用提供对应数据的，此时需要对数据进行汇聚加工。数据管理和开发人员需要在数据资源平台建立对应的数据中心，进行对应数据的加工。

数据开发为数据使用者提供一站式的集成开发环境，可满足数据资源平台下，数据开发者进行ETL开发、数据挖掘算法开发、数据主题库建设等需求。

5.4.2.1 工作流设计器

帮助您配置数据开发节点任务，包含ODPS SQL、ODPS MR、Shell、机器学习、数据同步、虚拟节点任务。可以被工作流任务或其他节点任务依赖，并能够被调度系统调度，完成数据仓库的建设。

图 5-5: 工作流设计器



5.4.2.1.1 节点任务

节点任务包含ODPS_SQL、ODPS_MR、Shell、机器学习、数据同步、虚拟节点任务，可以被工作流任务或其他节点任务依赖，并能够被调度系统调度。

5.4.2.1.2 任务属性配置

单击展开工作流任务/接单任务设计器中的属性区，进入属性配置界面。其主要包括基本属性、调度属性、依赖属性和跨周期依赖属性（节点任务包含版本属性）。

5.4.2.1.3 历史版本

可以查看任意版本的节点代码（仅限ODPS SQL、ODPS MR、Shell等节点类型），回滚节点的历史版本。

5.4.2.1.4 任务管理

任务管理提供了简单易用的发布功能，您可以选择工作流任务（单个或多个节点）、资源、对象进行打包发布，但发布包不包含运行环境、数据源连接配置信息、报警信息等。

5.4.2.2 代码开发编辑器

代码开发编辑器为开发人员提供代码编辑界面，帮助完成数据开发任务。

图 5-6: 代码开发编辑器

```
CREATE TABLE IF NOT EXISTS ods_log_tracker(
ip STRING COMMENT 'client ip address',
user STRING,
time DATETIME,
request STRING COMMENT 'HTTP request type + requested path without args + HTTP protocol version',
status BIGINT COMMENT 'HTTP reponse code from server',
size BIGINT,
referer STRING,
agent STRING)
COMMENT 'Log from coolshell.cn'
PARTITIONED BY(dt STRING);
```

5.4.2.2.1 SQL编程

MaxCompute提供基于Web端的SQL编程，包括辅助编程（自动SQL提示、格式化、代码高亮等）、代码调试运行等编程功能。

5.4.2.2.2 MR编程

MaxCompute支持将MR编译的JAR包以资源的形式上传并在ODPS_MR节点中引用的形式来使用。

5.4.2.2.3 Resource资源文件

支持将本地资源文件上传的类型，如下所示。

- JAR包：编辑好的Java jar包，供UDF或ODPS_MR程序调用。
- Python程序：供UDF程序调用。
- file文件：支持自定义参数的Shell脚本、xml配置文件、txt配置文件等资源文件。
- Archive类型：通过资源名称中的后缀识别压缩类型，支持的压缩文件类型包括.zip/.tgz/.tar.gz/.tar/jar。

5.4.2.2.4 UDF函数注册

UDF包含Java UDF和Python UDF两种，需先上传JAR包和Python程序后再进行UDF函数注册，您即可使用自定义函数进行数据开发。

5.4.2.2.5 Shell脚本编程

提供在线的shell脚本编程与调试环境。

5.4.2.3 代码管理与团队协作

代码管理与协作让数据开发可以多人同时进行编辑协作，提高开发效率。代码管理提供工作流任务和代码的锁机制，保证同一个工作流任务或代码在同一时间内只能被一个用户所编辑。您也通过获取锁的方式来得到编辑权限，并实时发送系统提示信息给对应用户。

同时数据开发也提供代码版本管理，您每一次提交的节点或工作流任务的版本都会被系统记录保存下来，支持查看任意两个版本的对比。

5.4.3 监控运维

5.4.3.1 系统概述

数据资源平台上数据量庞大、数据类型多样、数据业务复杂，数据处理任务也非常多，数据处理环节和流程周期长，需要支持高并发、多周期、支持多种数据处理环节的统一数据任务调度机制，按照策略进行数据任务调度。

监控运维为数据开发者和维护者提供一站式的数据运维管控能力，您可自主管理作业的部署、作业优先级、以及生产监控运维。平台提供数据监控运维、任务运行情况监控、异常情况告警、日常运维数据统计等功能。

5.4.3.2 运维概览

运维概览主要用来展示调度任务的指标数据情况，目前包含任务完成情况、任务运行情况、任务执行时长排行、调度任务数量趋势、近一月出错排行、任务类型分布和30天基线破线次数排行。

5.4.3.3 任务运维

可视化展示调度任务DAG图，极大地方便您对线上任务进行运维管理。

- 支持任务运行状态监控告警，支持单任务重跑、多任务重跑、kill、置成功、暂停等操作。
- 支持两种模式选择：包括列表、DAG模式。
- 可以针对周期运行、测试运行、手动运行任务查看任务运行状态。

- 可以针对任务进行重跑、停止、查看运行日志、查看节点代码、查看节点属性。

5.4.3.4 监控告警

监控告警是调度任务的监控保障系统，当任务出现错误的时候，系统会通过预定义的方式告知您任务失败。

您可以按照自己定义的规则来配置告警规则，及时调整任务产出，保障产出数据的及时性和可用性。

5.4.4 平台管理

平台管理主要从系统层面，为管理者对参与数据资源平台使用的用户进行对应管控，项目空间作为代码管理、成员管理、角色和权限分配的基本单元，每个团队都可具有独立的项目空间。您加入项目空间并被分配相关权限之后，才能够查看或编辑代码。

说明：

一个用户可以同时加入多个项目空间，在不同的项目空间中被授予不同的角色。

5.4.4.1 组织管理

显示组织详情信息以及组织Owner账号、AK信息，并可对组织对应人员进行成员管控。

5.4.4.2 项目管理

数据资源平台管理员可对您的项目空间进行列表展现，并提供创建、配置、激活、禁用项目空间的对应管理功能，方便数据资源层管理员对项目空间进行整体管控。

5.4.4.3 项目成员管理

项目成员列表，以列表的形式显示本项目的成员的名称、登录名称、成员角色等信息。

- 支持模糊搜索项目成员，并可将用户移出本项目空间。
- 用户加入项目空间，仅项目管理员可以新增项目成员，支持模糊匹配查找的方式将用户添加至本项目空间。

说明：

在添加用户到项目空间时，必须为其指定至少一种角色。

项目管理员可以主动清退用户。

说明：
用户移出本项目后失去之前在本项目分配的所有权限。

5.4.4.4 权限管理

权限管理主要完成平台用户、角色、权限等管理由统一的管理提供，对应权限特征如角色权限表所示。

表 5-1: 角色权限表

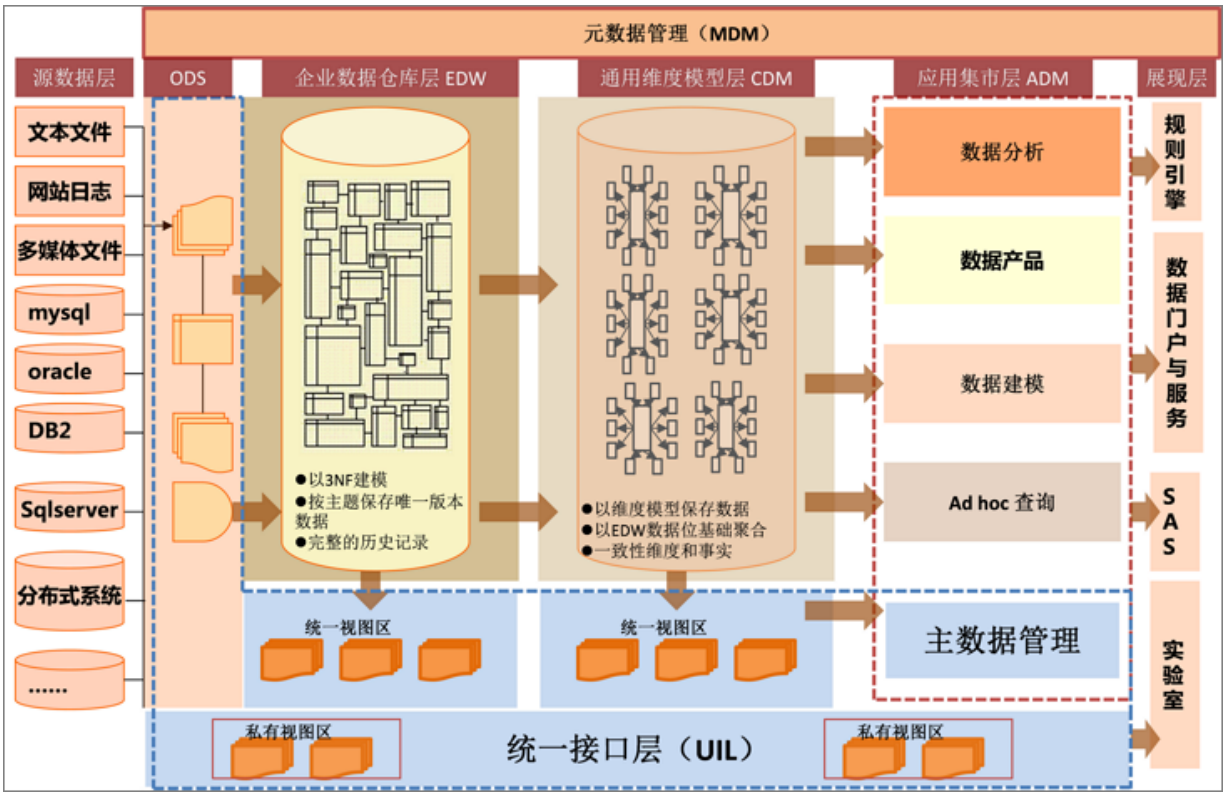
数据资源平台角色	平台权限特征
数据项目管理员	指数据项目空间的管理者，可对该项目空间的基本属性、数据源、当前项目空间计算引擎配置和项目成员等进行管理，并为项目成员赋予项目管理员、开发、运维、部署、访客角色 。
数据开发	开发角色的用户能够创建工作流、脚本文件、资源和 UDF，新建/删除表，同时可以创建发布包，但不能执行发布操作 。
数据运维	运维角色的用户由项目管理员分配运维权限；拥有发布及线上运维的操作权限，没有数据开发的操作权限 。
数据部署	部署角色与运维角色相似，但是它没有线上运维的操作权限 。
数据访客	访客角色的用户只具备查看权限，没有权限进行编辑工作流和代码等操作 。

5.5 应用场景

5.5.1 云上数仓

大型企业可在专有云环境下使用DataWorks来构建超大型的数据仓库。

图 5-7: 数据采集到应用



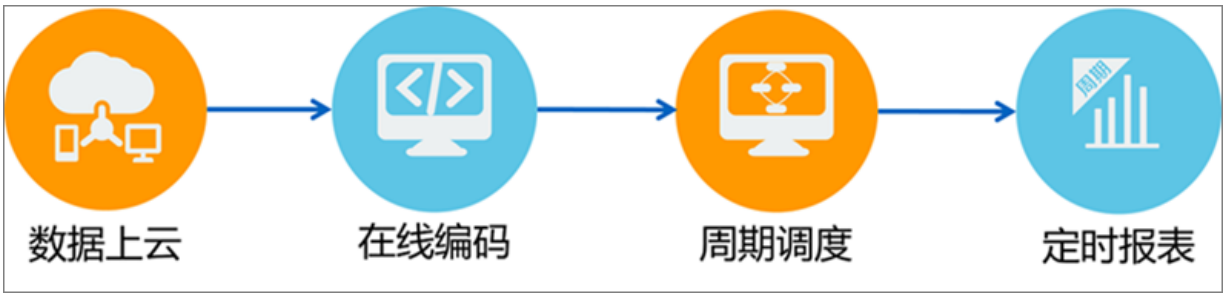
DataWorks为这类客户提供卓越的海量数据集成能力。

- 海量存储：可支持PB、EB级别的数据仓库，存储规模可线性扩展。
- 数据集成：支持多种异构数据源的数据同步和整合，消除数据孤岛。
- 数据开发：基于MaxCompute（原ODPS）的大数据开发，支持SQL、MR等编程框架，以及贴近业务场景的白屏化工作流设计器。
- 数据管理：基于统一的元数据服务来提供数据资源管理视图，以及数据权限审批流程。
- 离线调度：可以提供多时间维度的周期性调度能力，支持每天百万级的调度并发，并对任务调度实时监控，对错误及时告警。

5.5.2 BI应用

本节将为您介绍如何使用DataWorks做报表。

图 5-8: BI应用流程



基于网络日志，完成如下分析需求。

- 统计网站的PV（浏览次数）、UV（独立访客），按用户的终端类型（如Android、iPad、iPhone、PC等）分别统计，并给出这一天的统计报表。
- 网站的访问来源，了解网站的流量从哪里来。

截取一条真实的数据如下：

```
xx.xxx.xx.xxx - - [12/Feb/2014:03:15:52 +0800] "GET /articles/4914.html
HTTP/1.1" 200 37666
"http://xxx.cn/articles/6043.html" "Mozilla/5.0 (Windows NT 6.2; WOW64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/xx.x.xxxx.xxx Safari/537.
36" -
```

新建表：在导入数据之前，需要先创建一张MaxCompute目标表，把表名命名为ods_log_tracker。

图 5-9: 建表

图 5-10: 各表依赖

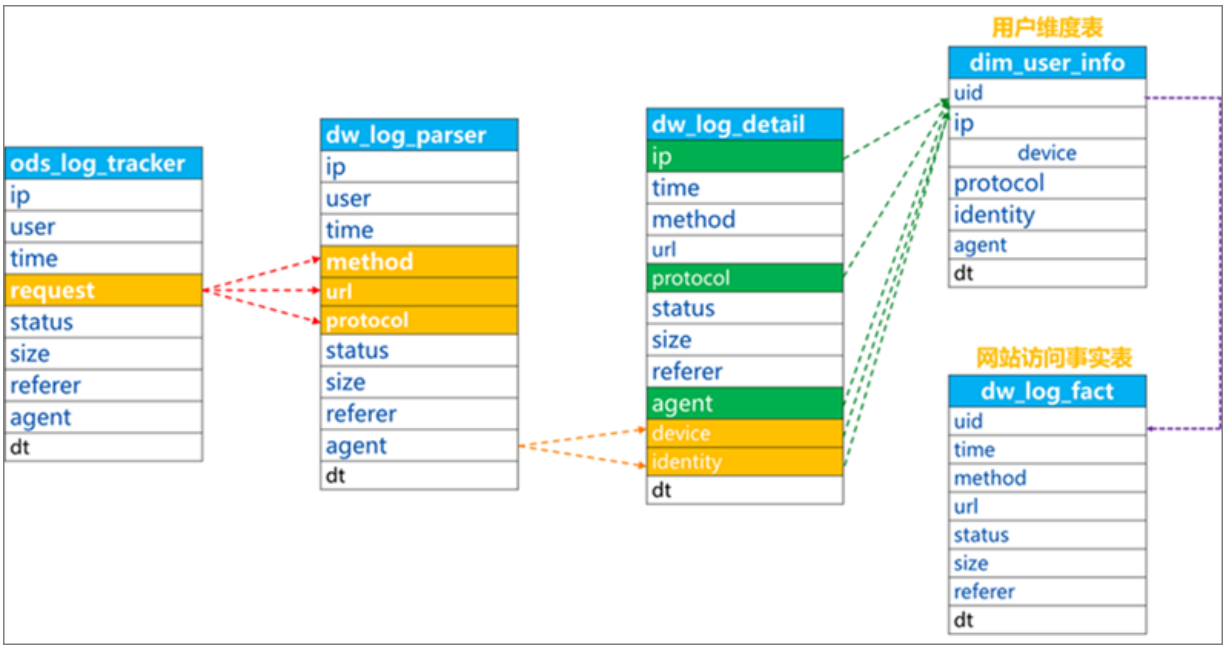


图 5-11: 新建任务

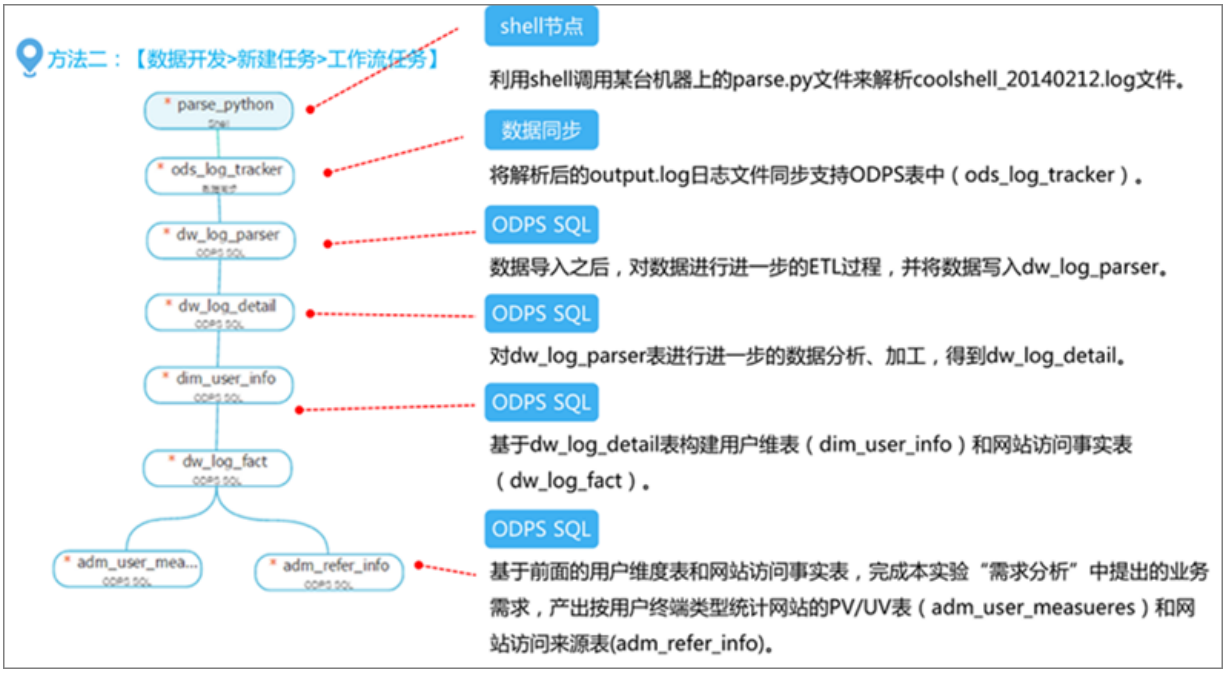
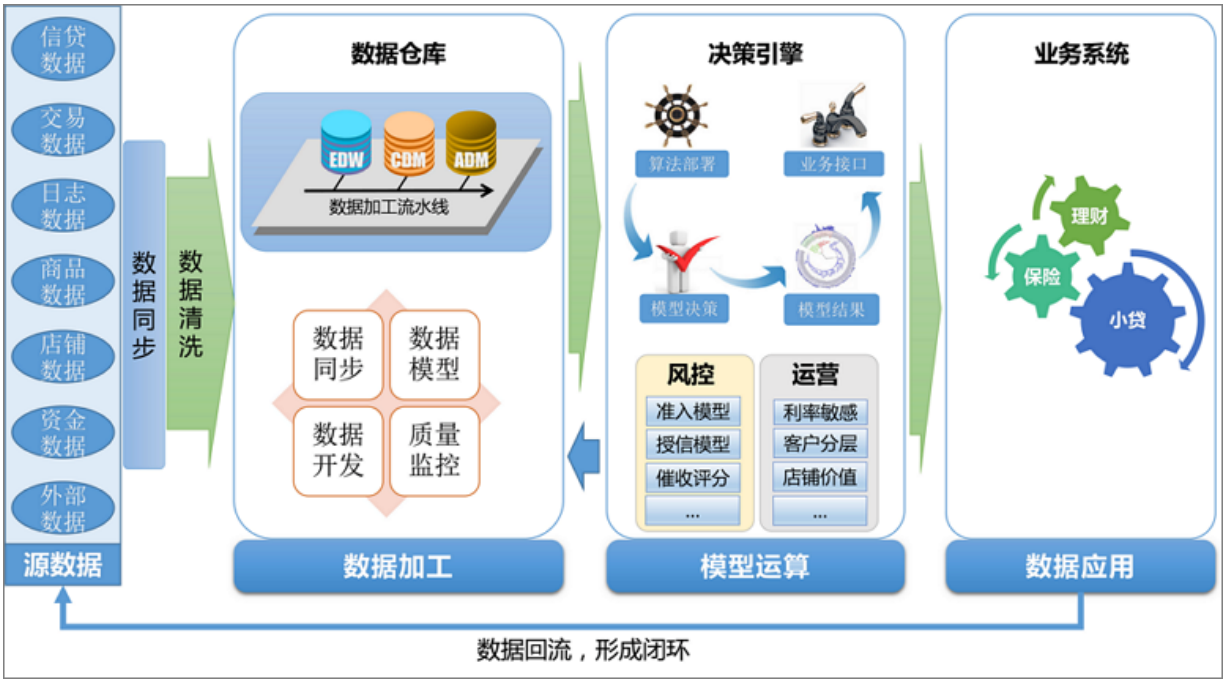


图 5-12: 开发过程

5.5.3 数据化运营

- 创新业务：通过数据挖掘建模和实时决策系统，将大数据加工结果直接应用于业务系统。
- 中小企业：基于DataWorks可快速使用和分析数据，助力企业的经营决策。

图 5-13: 阿里小贷的数据业务运营模式图



5.6 使用限制

无

5.7 基本概念

任务（Task）

任务是指定义对数据执行的操作，分为节点任务（node task）、工作流任务（flow task）和内部节点（inner node）三种类型。

实例（Instance）

在调度系统中的任务经过调度系统、手动触发运行后会生成一个实例，实例代表了某个任务在某时某刻执行的一个快照，实例中会有任务的运行时间、运行状态、运行日志等信息。

提交（Submit）

提交是指开发的节点任务、工作流任务从开发环境发布到调度系统的过程。完成提交后，相应的代码、调度配置全部合并到调度系统中，调度系统按照相关配置进行调度操作。

说明：

未提交的节点任务、工作流任务不会进入到调度系统。

脚本开发（Script）

脚本开发是提供给数据分析使用的一个代码存储空间，脚本开发的代码无法发布到调度系统，无法进行调度参数配置，仅可以进行一些数据查询分析的工作。

6 分析型数据库AnalyticDB

6.1 什么是分析型数据库

分析型数据库AnalyticDB（原名 ADS）是阿里巴巴针对海量数据分析自主研发的实时高并发在线分析RT-OLAP（Realtime OLAP）云计算服务，支持对千亿级数据进行即时的（毫秒级）多维分析透视和业务探索。

说明：

联机分析处理OLAP（Online Analytical Processing）系统是相对联机事务处理OLTP（Online Transaction Processing）系统而言的，它擅长对已有的海量数据进行多维度的、复杂的查询和分析，适用于分析型数据库系统。OLTP系统擅长事务处理，数据操作保持着严格的一致性和原子性，支持频繁的数据插入和修改，通常用于MySQL、Microsoft SQL Server等关系型数据库系统。

AnalyticDB是一套RT-OLAP系统，具有以下特点：

- 兼容 MySQL、现有的商业智能 BI （Business Intelligence）工具和ETL（Extract-Transform-Load）工具，可以经济、高效、轻松地分析与集成您的所有数据。
- 采用关系模型存储，可以使用SQL进行自由灵活的计算分析，无需预先建模。
- 采用分布式计算技术，具有强大的实时计算能力。在处理百亿条甚至更多量级的数据上，AnalyticDB的性能可以达到甚至超越MOLAP类系统。AnalyticDB在数百毫秒内可以完成百亿级的数据计算，使用者可以根据自己的想法在海量数据中进行自由的探索，而不是根据预先设定好的逻辑查看已有的数据报表。
- 兼具实时和自由的海量数据计算能力，拥有快速处理千亿级别的海量数据的能力。在AnalyticDB系统中，数据分析使用的数据是业务系统中产生的全量数据而不再是抽样的，这使得数据分析结果具有最大的代表性。
- 能够支撑较高并发查询量，同时通过动态的多副本数据存储计算技术也保证了较高的系统可用性，所以AnalyticDB能够直接作为面向最终用户（End User）的产品（包括互联网产品和企业内部的分析产品）的后端系统。当前AnalyticDB已普遍应用于拥有数十万至上千万最终用户的互联网业务系统中，例如淘宝数据魔方、淘宝指数、快的打车、阿里妈妈达摩盘（DMP）、淘宝美食频道等。

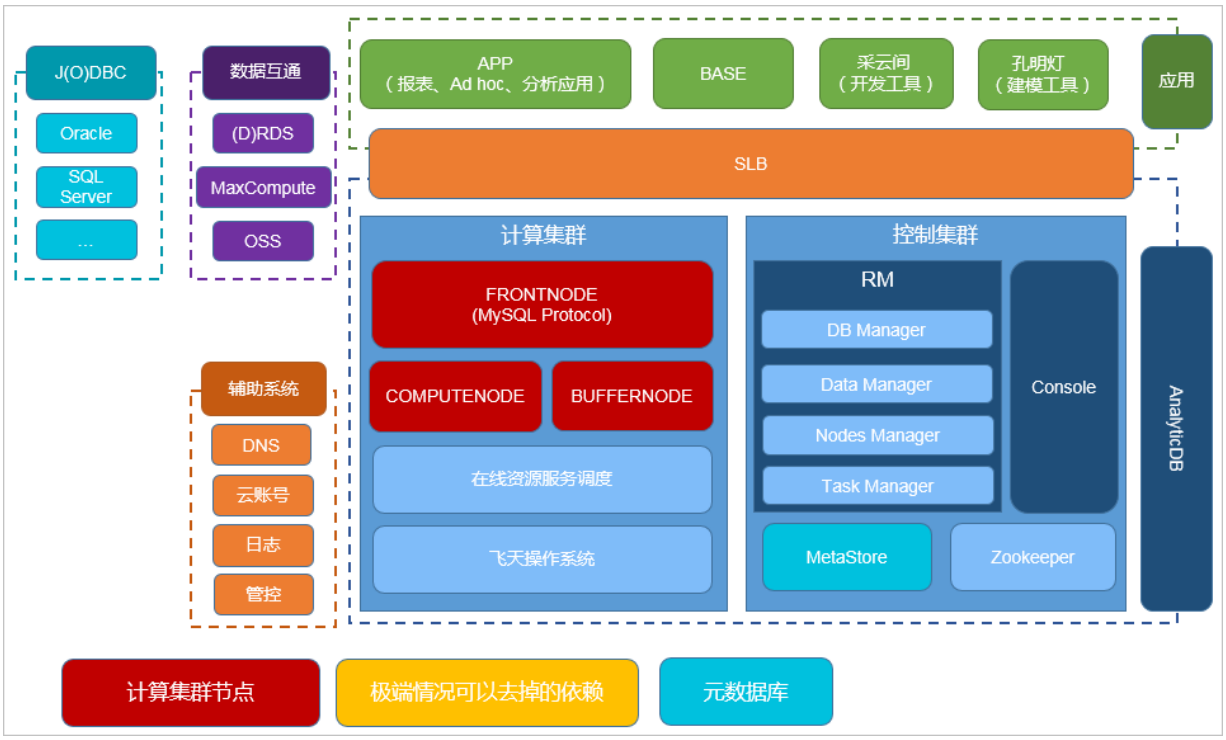
作为海量数据下的实时计算系统，AnalyticDB给使用者带来了极速的、自由的大数据在线分析计算体验。

6.2 产品优势

优势	描述
海量数据计算能力	最高支持计算单表万亿记录、PB级别的数据。
全量数据分析	数据分析中使用的数据不再是抽样的，而是全量数据，分析结果具有最大的代表性。
查询响应极速	支持在毫秒级内对百亿级数据进行多维透视。
高并发、高可用	支持高并发查询量，并且通过动态的多副本数据存储计算技术来保证系统的高可用性，能够直接作为面向最终用户（End User）产品（包括互联网产品和企业内部的分析产品）的后端系统。
查询形式自由灵活	支持通过SQL灵活的对海量数据进行多维分析、数据透视、数据筛选。
数据导入多通道并行	支持离线通道、在线通道双模式并行数据导入，导入性能随集群规模线性扩展。
安全机制精细	支持精确到列级别的权限管理和超细粒度的用户操作审计，通过公私钥机制保护数据安全。
兼容性良好	全面兼容MySQL协议（包括数据元信息）、兼容商业分析工具和应用、内置支持多种数据源数据快速接入，大幅度降低业务系统和商业软件的接入成本。

6.3 产品架构

AnalyticDB是基于MPP架构并融合了分布式检索技术的分布式实时计算系统，构建在飞天操作系统之上。AnalyticDB的主体部分主要由底层依赖、计算集群、控制集群和外围模块组成，具体如下图所示。



底层依赖

底层依赖包括：

- 飞天操作系统：用于资源虚拟化隔离、数据持久化存储、构建数据结构和索引。
- MetaStore：阿里云RDS关系数据库或阿里云表格存储，用于存储分析型数据库的各类元数据（注意并不是实际参与计算用的数据）。
- 开源Apache ZooKeeper模块：用于对各个组件进行分布式协调。

计算集群

计算集群是计算资源实际包括的内容，均可进行横向扩展。计算集群运行在飞天操作系统上，通过在线资源调度模块来调度计算资源。计算集群包括：

- FRONTNODE：用于处理用户连接接入认证、鉴权，查询路由和分发路由，以及提供元数据查询管理服务。
- COMPUTENODE：用于进行实际的数据存储与计算的 。
- BUFFERNODE：用于处理数据实时更新、数据缓冲和实时数据写入版本控制。

控制集群

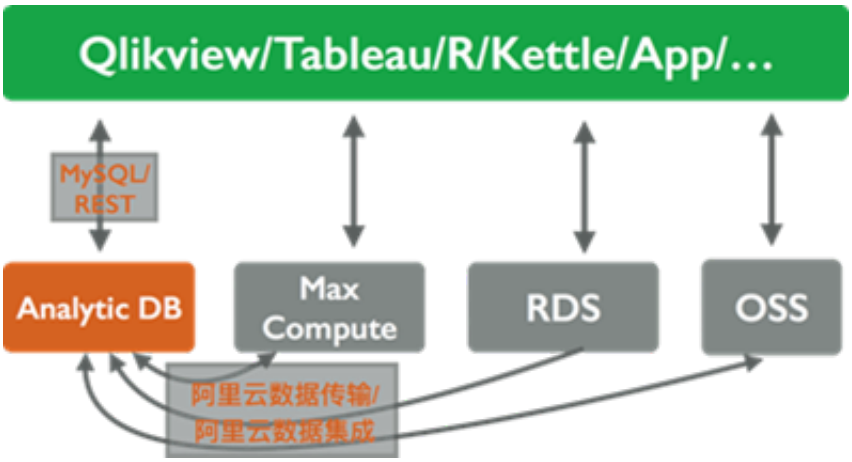
控制集群（即资源管理器RM）用于控制计算集群中数据库资源分配、数据库内数据和计算资源的分布、飞天集群上的计算节点管理、数据库后台运行的任务管理等。控制集群实际上由多个模块组成，一个控制集群可以同时管理部署于不同机房的多套计算集群。

外围模块

外围模块主要包括：

- 阿里云负载均衡：用于管理Front Node的分组和负载均衡。
- 阿里云DNS系统：用于发布数据库域名。
- 阿里云账号系统。
- AnalyticDB控制台（Admin Console）。
- 用户控制台（DMS for Analytic DB）。

外围模块与外部系统交互如下图所示。



- 支持从MaxCompute批量导入数据，也支持快速批量导出海量数据到MaxCompute。
- 支持实时的将(D)RDS的数据同步到分析型数据库中（需借助外部同步工具）。
- 支持从OSS批量导入数据，也支持快速批量导出海量数据到OSS。

AnalyticDB主要支持的客户端、驱动、编程语言和中间件如下：

- 客户端和驱动：支持MySQL 5.1/5.5/5.6系列协议的客户端和驱动，如MySQL 5.1.x jdbc driver、MySQL 5.3.x odbc connector(driver)、MySQL 5.1.x/5.5.x/5.6.x 客户端。
- 编程语言：JAVA、Python、C/C++、Node.js、PHP、R（RMySQL）。
- 中间件：Websphere Application Server 8.5、Apache Tomcat、JBoss。

6.4 功能特性

6.4.1 DDL

数据库管理：

- 通过DDL创建数据库。
- 通过DDL删除数据库。
- 查看全部有权限的数据库列表（show databases）。
- 查看和管理每个数据库的访问信息（域名、端口等信息）。
- 通过DDL对数据库使用的ECU资源进行扩容、缩容。
- 通过DDL创建表组和修改表组属性。
- 通过DDL创建表。
- 通过DDL在已创建的表中增加列。
- 通过DDL修改表属性。
- 通过DDL修改索引。
- 支持Create-table-as-Select创建临时表。

6.4.2 DML

6.4.2.1 SELECT

- 和标准MySQL的Query兼容达90%。
- 支持表达式、函数、别名、列名、case when等列投射形式。
- 支持From表名as别名，Join表名as别名。
- 支持事实表之间的Join（若需加速Join则有限定条件）和事实表与维度表的Join（几乎无限制）。
- 支持多个on条件的Join（若需加速Join则其中必须包含一个一级分区列）。
- 过滤条件（where）中，支持and和or表达式组合、支持函数表达式、支持between、is等多种逻辑判断和条件组合。
- 支持多列group by，并且支持case when等列投射表达式产生的别名进行group by，支持常见的聚合函数。
- 支持order by表达式、列，并支持正序和倒序。
- 支持having。

- 支持子查询（建议不超过3层），支持在特定条件下的两个子查询的Join，支持过滤条件中的in中使用数据全部源自维度表的子查询，通过Full MPP Mode支持任意的in子查询。
- 支持带有一级分区列的多列的[count]distinct，在Full MPP Mode下支持任意列的[count]distinct。
- 支持常数列。
- 支持union/union all，有限定条件的支持minus/intersect。

6.4.2.2 INSERT/DELETE

- 支持对已定义主键的实时写入表进行INSERT、DELETE操作。
- 在资源足够的情况下，单表可支撑5万次每秒以上的INSERT操作，数据插入后数分钟内生效。
- 多种机制保障写入成功的数据不会丢失，insert支持overwrite、ignore两种模式。
- 支持insert into...select from。

6.4.3 存储模式

AnalyticDB支持两种存储模式：

存储模式	描述	优点	缺点	是否支持模式切换
高性能存储模式	采用全SSD（或Flash卡）作为计算用数据存储，采用内存作为数据和计算的动态缓存实例，可运行于双千兆或双万兆的网络服务器上。	计算性能好、查询并发能力强。	存储成本较高。	不支持。
大存储模式	采用分布式存储的SATA磁盘作为计算用数据存储，采用SSD和内存两级作为数据和计算的动态缓存实例。必须运行于双万兆的网络服务器上。	存储成本低。	查询并发能力相对较弱，一次性计算较多行列时性能较差。	支持切换到高性能存储模式。

6.4.4 计算引擎

AnalyticDB支持两套计算引擎：

计算引擎	描述	优点	缺点
COMPUTENode Local /Merge（简称LM）	原有计算引擎。	计算性能好、并发能力强。	对部分跨一级分区列的计算支持较差。
Full MPP Mode（简称MPP）	新增的计算引擎。优点是计算功能全面，支持跨一级分区列的计算	计算功能全面，支持跨一级分区列的计算，可以通过全部TPC-H查询测试用例（22个）和60%以上的TPC-DS查询测试用例。	计算性能相对LM引擎较差，计算并发能力相对LM引擎很差。

当开启MPP引擎时，AnalyticDB自动对查询（Query）进行路由，并将LM引擎不支持的Query路由到MPP引擎，以兼顾AnalyticDB的性能和通用性。用户也可以通过Hint来指定某个Query 使用的计算引擎。

6.4.5 系统资源管理

AnalyticDB通过ECU（弹性计算单元）进行资源管理。通过操作系统底层技术和飞天操作系统提供的分布式资源调度能力，AnalyticDB为每个数据库实例创建完全独立的FRONTNODE、COMPUTENODE、BUFFERNODE进程。每个数据库至少拥有FRONTNODE、COMPUTENODE、BUFFERNODE进程各两个（双副本双活）。

您可以通过控制ECU型号来控制FRONTNODE、COMPUTENODE、BUFFERNODE进程的配置。通过ECU型号可以区分的资源包括CPU 核数（支持独占和共享）、内存大小（独占）、SSD大小（独占）、网络带宽（独占）、SATA数据逻辑大小（仅大存储实例的COMPUTENODE可选）。

您可以通过ECU的数量来控制一个数据库实例所要启用的COMPUTENODE数量；而通过ECU上配置的COMPUTENODE与FRONTNODE和BUFFERNODE的比例，系统会自动启用相应数量的FRONTNODE和BUFFERNODE。通过这种方式可以达到容量水平伸缩的目的。

FRONTNODE、COMPUTENODE、BUFFERNODE进程默认混部在同一批物理机上，您也可以通过参数配置，强制让不同的角色运行于不同的物理机上。

除此之外，AnalyticDB的后台任务、数据库AM等也会占用一定量的系统资源。

6.4.6 权限与授权

授权模型：

- 支持标准MySQL模式的权限模型。

- 支持对数据库、表组、表、列四个级别进行ACL授权。
- 支持数据库Owner授权给任意合法账号。
- 支持单独控制或全局控制数据库创建权限。
- 支持超级管理员、系统管理员等角色。
- 支持每个级别授予不同的权限。
- 支持ADD USER/REMOVE USER语句添加和删除用户。
- 支持GRANT语句进行授权。
- 支持REVOKE语句进行权限回收。
- 支持SHOW GRANTS ON语句查看各级对象上的用户权限。
- 支持LITS USERS语句查看全部有权限的用户。
- 超级管理员：集群初始建立时指定的账号，具有任命系统管理员和数据库管理员的权限，无其他权限。
- 系统管理员：由超级管理员任命，具有查看和操作SYSDB的权限。
- 数据库管理员：由超级管理员任命，具有为其他用户创建数据库和删除其他用户的数据库的权限。
- 数据库Owner：数据库的所有者，具有一个数据库的全部权限，并可以授权一般用户访问自己的数据库。

6.4.7 数据导入导出

支持快速导入导出海量数据：

- 支持任何SELECT语句的查询输出。
- DUMP DATA语句支持将大量数据快速导出到OSS等DFS中以及MaxCompute。
- 支持类BULKLOAD模式导入MaxCompute、OSS、RDS中用户存放的数据。
- 内置支持使用LOAD DATA语句进行导入。
- 内置支持导入数据Owner校验，保证导入安全。

6.4.8 元数据

6.4.8.1 information_schema

- 最大限度兼容MySQL标准的数据库、表、列等信息，元数据完全可被您使用并可进行交互。
- 数据导入的记录和进度均可在元数据库进行查询。
- 提供ECU运行状态，以及ECU扩容、缩容记录表。

- 元数据按照数据库进行隔离，您无法访问无权使用的元数据。

6.4.8.2 performance_schema

- 提供实时元仓，可进行SQL粒度的查询审计以及分钟粒度的插入性能统计。
- 提供分钟级别更新的QPS、RT、请求数、数据量大小等实时性能监测。
- 元数据按照DB进行隔离。

6.4.8.3 sysdb

- 面向系统管理员和运维人员的元数据库。
- 支持查看AnalyticDB全部模块的运行状态、运行历史记录等，拥有数十张各个主题的系统元数据表。
- 支持查看系统的运行状态，并在有需要时可以进行修改。
- 支持查看或修改系统各组件的参数，运行计算集群升级、降级、扩容、缩容、挂起命令。

6.4.9 特色功能

6.4.9.1 特色函数

- 支持高性能的根据地理坐标范围筛选数据（方形和圆形圈选、点距离计算等）。
- 支持智能分段统计函数（指标的自动分段）。
- 支持快速多列聚合函数。
- 根据列的数据分布智能为全部列建立索引，无需您进行任何操作。
- 对完全不需进行检索的列，您可手动关闭智能索引。

6.4.9.2 智能缓存和CBO优化

- 拥有多层智能缓存，最大限度的利用内存加速计算，但是可计算的数据量大小不受内存大小限制。
- 拥有智能的CBO优化器，可以根据数据分布情况和您的SQL执行情况动态优化计算执行计划，让您从SQL写法优化中解脱。
- 拥有智能的分布式长尾处理技术，大幅度降低分布式系统中单节点繁忙以及网络等不确定性因素对响应时间的影响。

6.4.9.3 Quota控制

- 支持每次导入数据量、单表导入次数、并发任务数、DB导入次数、DB导入总数据量等控制。
- 支持ECU总数据量控制、ECU库存控制、单次扩容ECU数量控制、每天扩容次数控制。

- 支持DB的总表数、总实时表数、总表组数、单表最大列数、单表分区数（一级、二级）控制。
- 支持系统元数据库（SYSDB、ADMIN DB）流控和风控。
- 支持大存储模式实例，使用SATA进行计算数据存储，使用SSD和内存作为缓存加速热点数据查询。

6.4.9.4 Hint和小表广播

- 支持通过Hint干预执行计划，如计算引擎的选定和索引使用的控制。
- 支持小表广播模式Join，在小表（物理表或虚表）Join大表时通过 Hint指定小表广播，在不符合加速Join条件时亦可获得比较好的Join性能。

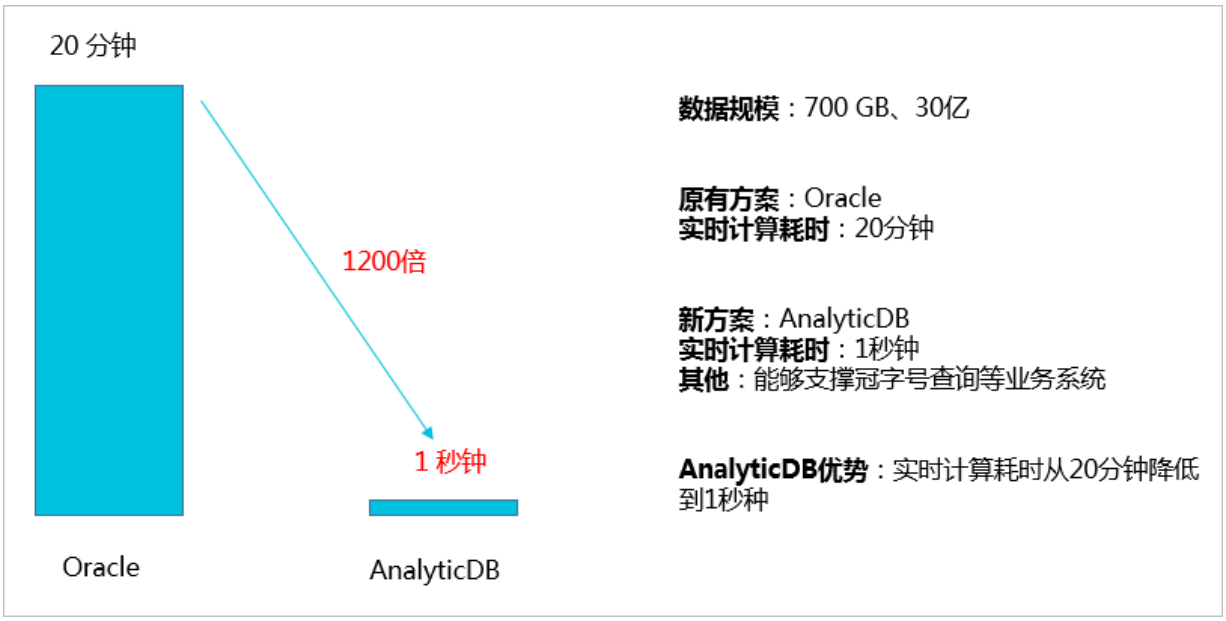
6.5 应用场景

AnalyticDB常见的应用场景如下表所示。

应用场景	描述
电商行业	A-CRM、爆款选品、自动化运营、SKU组合分析等。
O2O	数据分析和CRM系统、地理围栏系统。
广告行业	数字营销，M-DMP系统。
金融行业	实时多维数据分析、交易流水查询系统、报表系统等。
大安全	人群透视分析，潜在关键元素挖掘，关系网络分析，明细查询等。
交通、交警	车辆卡口数据分析和研判。
物流和物联网	车联网数据分析、企业安监数据分析、传感器数据存储和检索、物流实时数据仓库。

6.5.1 某银行

某银行原有方案和AnalyticDB方案对比如下图所示。



6.5.2 某交警

某交警应用示例如下图所示。

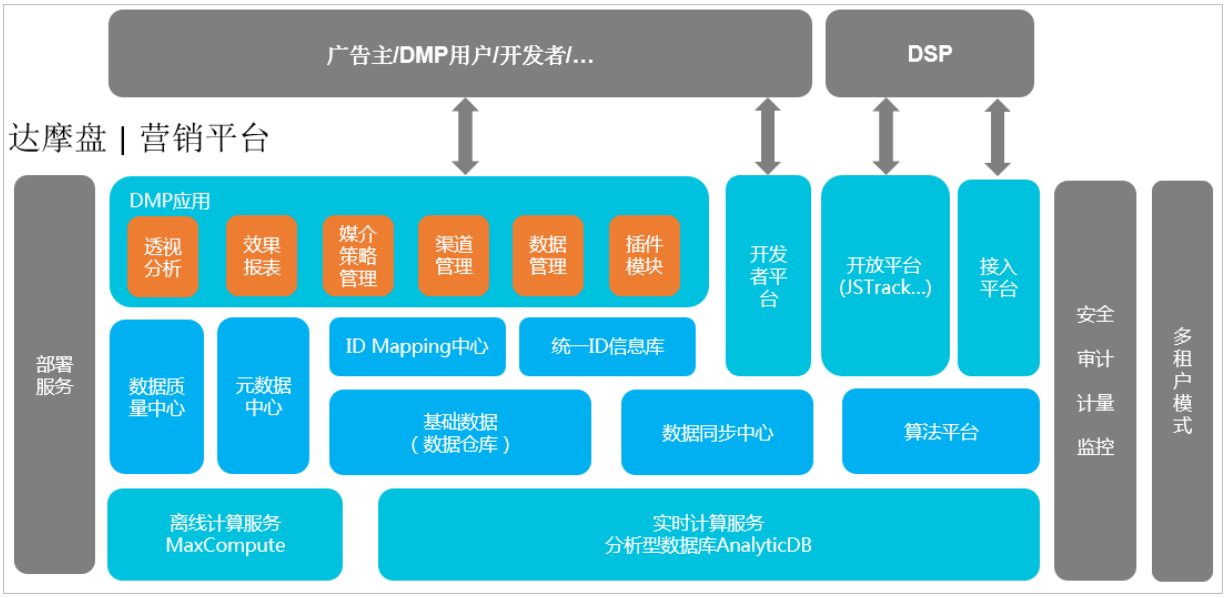
该交警业务系统特点如下：

- 海量数据：全市仅交通卡口过车记录表达到300亿~500亿级别（保存半年），折合数据20TB~30TB。
- 飞速增长：市级系统每天数据增量1000万条。

- 复杂查询：有多个部门需要查询，多种插叙方式。业务应用查询复杂，包括单表查询、多表查询（join）、模糊查询（like）、轨迹分析（in）、区域碰撞（intersect）、短时过车（having count）、多用户等多种应用场景，对表结构设计、内存使用效率、CPU使用效率等要求较高，对查询的并发性也有要求。

6.5.3 阿里妈妈DMP

阿里妈妈的广告DMP（Data Management Platform）应用架构如下:



在DMP系统中，大数据处于整个系统的核心位置，具体说明如下：

- MaxCompute进行用户数据清洗、标签挖掘。
- AnalyticDB承接了广告主对大数据的透视和人群管理的计算工作。AnalyticDB具有海量数据极速导出功能，已圈定的用户群数据可通过AnalyticDB导出到查询速度更快的KV（Key-Value）存储系统中。
- 定向引擎根据KV存储系统的数据服务于DSP（Demand-Side Platform）系统。

6.6 使用限制

AnalyticDB当前版本存在一些使用限制，您需要进行合理的设计以规避这些限制。

数据表限制

限制项	最大值	说明及实践
一级分区数	256个	表记录数超过10亿条时，建议采用二级分区方案。

限制项	最大值	说明及实践
二级分区数	1000个	建议小于90，需要在where子句增加二级分区条件。
单表最大列数	1024列	超过1024列时，建议拆分为两个或者多个表。
varchar长度	8KB	超过8KB时，建议使用clob存储

处理能力限制

单个AnalyticDB数据库的处理能力与所分配的资源数量有关。AnalyticDB提供良好的线性扩展能力，当遇到可用资源不足时，需要考虑资源扩容，相关说明及实践如下：

限制项	最大值	说明及实践
实时表每日更新二级分区上限	30	建议低于3。对于需要更新历史二级分区数据的情况，需要合理考虑二级分区的设置，建议每日更新数据跨越的二级分区数不超过3个。如果是补录历史数据的场景，建议分批次逐天完成历史数据补录。
表/分区单次导入最大数据量	N/A	建议小于 *ECU*个数**diskSize* **0.2*，其中diskSize为对应的ECU磁盘容量大小。

6.7 基本概念

用户

用户是AnalyticDB数据库的使用者，可通过云账号进行登录数据库。在AnayticDB数据库中，不同用户可以被授予不同的权限，用户的操作也均可以被细粒度审计。

数据库

数据库是组织、存储和管理数据的仓库。在AnalyticDB中，数据库是租户隔离的基本单位，是用户所关心的最大单元，也是用户和分析型数据库系统管理员的管理职权的分界点。

分析型数据库系统管理员最小可管理数据库粒度的参数，未经用户授权，无法查看和管理数据库的内部结构和信息。用户只能查看大多数数据库级别的参数，但不能修改。

在AnalyticDB中，一个数据库对应一个用于访问的域名和端口号，并且有且只有一个Owner（即创建者）。

在AnalyticDB中，用户的宏观资源配置是以数据库为粒度进行的，所以在创建数据库时分析型数据库通过业务预估的QPS、数据量、Query类型等信息智能的分配数据库初始资源。

表组

表组是一系列可发生关联的表的集合，是AnalyticDB为方便管理关联数据和资源配置而引入的概念。在AnalyticDB中，表组可分为普通表组和维度表组，说明如下：

- 普通表组：
  - 普通表组是数据物理分配的最小单元，数据的物理分布情况通常无需用户关心。
  - 一个普通表组最大支持创建256个普通表。
  - 数据库中数据的副本数（指数据在AnalyticDB中同时存在的份数）必须在表组上进行设定，同一个表组的所有表的副本数一致。
  - 只有同一个表组的表才支持快速HASH JOIN。
  - 同一个表组内的表可以共享一些配置项（例如：查询超时时间）。如果表组中的单表对这些配置项进行了个性化配置，那么在进行表关联时会用表组级别的配置进行覆盖单表的个性化配置。
  - 同一个表组的所有表的一级分区（即HASH分区）的分区数建议一致。
- 维度表组：
  - 用于存放维度表（一种数据量较小，但能和任何表进行关联的表）。
  - 维度表组是创建数据库时自动创建的，一个数据库有且只有一个，不可修改和删除。

表

AnalyticDB支持标准的关系表模型。在AnalyticDB中，表通常分为普通表（也称事实表）和维度表，说明如下：

- 普通表：创建时必须至少指定一级分区（即HASH分区）列、分区相关信息，同时还需要指定该普通表所属的表组。普通表支持根据若干列进行数据聚集（聚集列），以实现高性能查询优化。单表最大支持1024个列，可支持数千亿行甚至更多的数据。
- 维度表：创建时不需配置分区信息，但会消耗更多的存储资源，单表数据量大小受限。维度表最大可支持千万级的数据条数，可以和任意表组的任意表进行关联。

普通表最多支持两级分区，一级分区是 HASH分区，二级分区是LIST 分区。HASH 分区和LIST分区说明如下：

- HASH 分区：
  - 根据导入操作时已有的一列内容进行散列后进行分区。
  - 一个普通表至少有一级HASH分区，分区数最小支持8个，最大支持256个。
  - 多张普通表进行快速 HASH JOIN ，JOIN KEY必须包含分区列，并且这些表的HASH分区数必须一致。
  - 数据装载时，仅包含HASH分区的数据表会全量覆盖历史数据。
- LIST分区：
  - 根据导入操作时所填写的分区列值来进行分区。同一次导入的数据会进入同一个LIST分区，因此LIST分区支持增量的数据导入。
  - 一个普通表默认最大支持365*3个二级分区。

在AnalyticDB中，两级分区表同时支持HASH JOIN和增量数据导入。对于任一分区形态的表，查询操作均不强制要求指定分区列，但查询操作指定分区列或分区列范围时可能提高查询性能。

在AnalyticDB中，根据表的数据更新方式，表分为批量更新表和实时更新表，说明如下：

- 批量更新表：适合将离线系统（如MaxCompute）产生的数据批量导入到分析型数据库，供在线系统使用。
- 实时更新表：支持通过INSERT、DELETE语句增加和删除单条数据，适合从业务系统直接写入数据。

注意：

- 分析型数据库不支持读写事务。
- 数据实时更新后，一分钟左右才可查询。
- 分析型数据库遵循最终一致性。

列

- 支持boolean、tinyint、smallint、int、bigint、float、double、varchar、date、timestamp等多种MySQL标准数据类型。
- 支持多值列multivalue（AnalyticDB特有的数据类型列），高性能存储和查询一个列中的多种属性值信息。

- 支持删除列的自动化索引，无需手动追加建立HashMap索引。

ECU

ECU是弹性计算单元，是Analytic DB的资源调度和计量的基本单位。

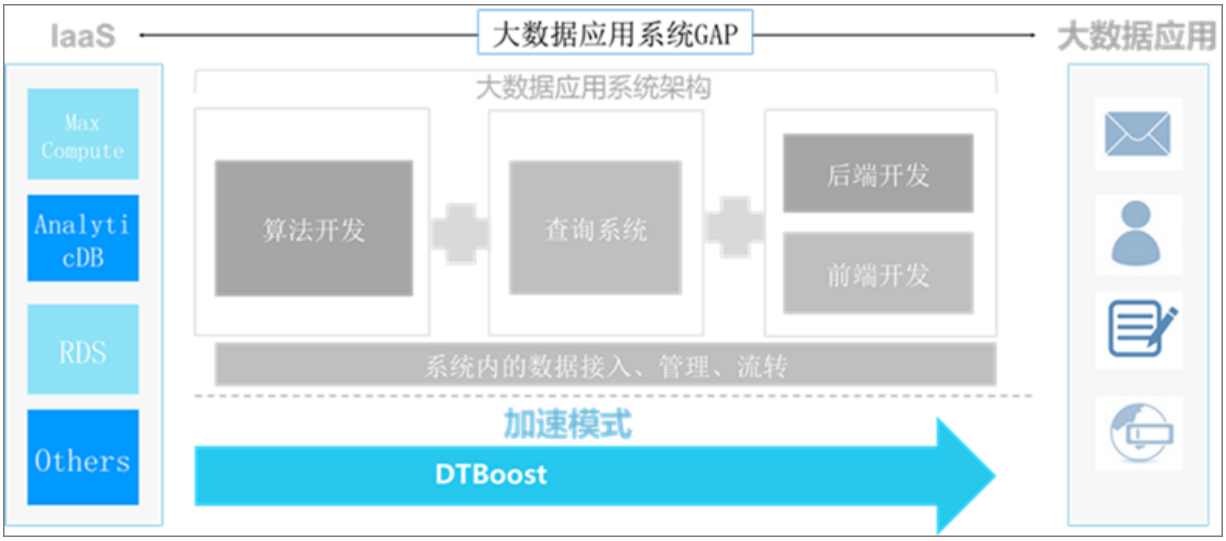
ECU可配置不同的型号，每种型号的ECU可配置CPU核数（最大、最小）、内存空间、磁盘空间、网络带宽等多种资源隔离指标。Front Node、Compute Node、Buffer Node均由ECU进行资源隔离。Compute Node的ECU数量和型号需由用户配置（弹性扩容/缩容），Front Node、Buffer Node的数量和型号系统自动根据ComputeNode的情况换算和配置。Analytic DB出厂时已根据机型配置预设了多种最佳的ECU型号。

7 大数据应用加速器DTBoost

7.1 什么是DTBoost

阿里云数据加速器（DTBoost）从大数据应用落地点出发，提供了一套大数据应用开发套件，能够帮助开发者从业务需求的角度有效地整合阿里云各个大数据产品，大大降低搭建大数据应用系统当中绝大部分的系统工程工作，在相应行业应用解决方案的结合下，能够让不是很熟悉大数据应用系统开发的程序员也可快速为企业搭建大数据应用，从而实现大数据价值的快速落地。

图 7-1: 大数据应用系统



DTBoost是以标签中心为基础，建立跨多个云计算资源之上的统一逻辑模型，开发者可以在**标签**这种逻辑模型视图上结合画像分析、规则预警、文本挖掘、个性化推荐、关系网络等多个业务场景的数据服务模块，通过接口的方式进行快速的应用搭建。这种方式有以下好处：

- 可屏蔽掉应用开发人员对于下层多个计算存储资源的深入理解与复杂的系统对接工作。
- 通过数据服务的形式，有助于IT部门对数据使用进行管理，避免资源的重复和冗余。

简单来说，因为大数据计算能力的增强，开发者只需要把需要使用的数据在模型当中进行管理后，即可通过API方式进行相应的计算，并对接到产品界面端。或通过提供的界面配置功能直接生成可以独立部署的代码，以快速搭建相应的大数据产品。

7.2 产品优势

新一代企业级大数据应用平台

DTBoost以加速企业内业务数据化进程为目标，提供物理数据管理的逻辑抽象模型（OLT），并提供画像分析、营销引擎、规则预警、推荐引擎、可视化等基础数据应用引擎，加速数据业务的快速落地。

适用于多种场景

- **基于行为等明细数据的分析**

  自由地分析各类行为明细数据，进行的分析可能是任意维度之间的交叉关系，很难进行预先的计算。

- **从半结构化数据中抽取特征**

  灵活分析还意味着能够与预测、评分、文本特征提取等算法技术相结合，进行广度与深度兼备的分析。能够借助一些偏好计算、文本挖掘类的算法能够从这些半结构化的数据当中对用户互相的特征进行深度的挖掘。

- **交互式的查询分析**

  分析是在数据当中探索有用的信息，能够根据不断调整的筛选条件、维度组合、下钻上聚能够快速返回结果，直到获取到足够多的信息，这对查询速度的高响应提出了要求。

开放平台

DTBoost不仅提供UI交互界面，并提供一组强大API接口，可以方便地完成系统集成，大大降低应用开发成本。

Web化的软件服务

在互联网/内部网络环境下安装部署，即可使用。

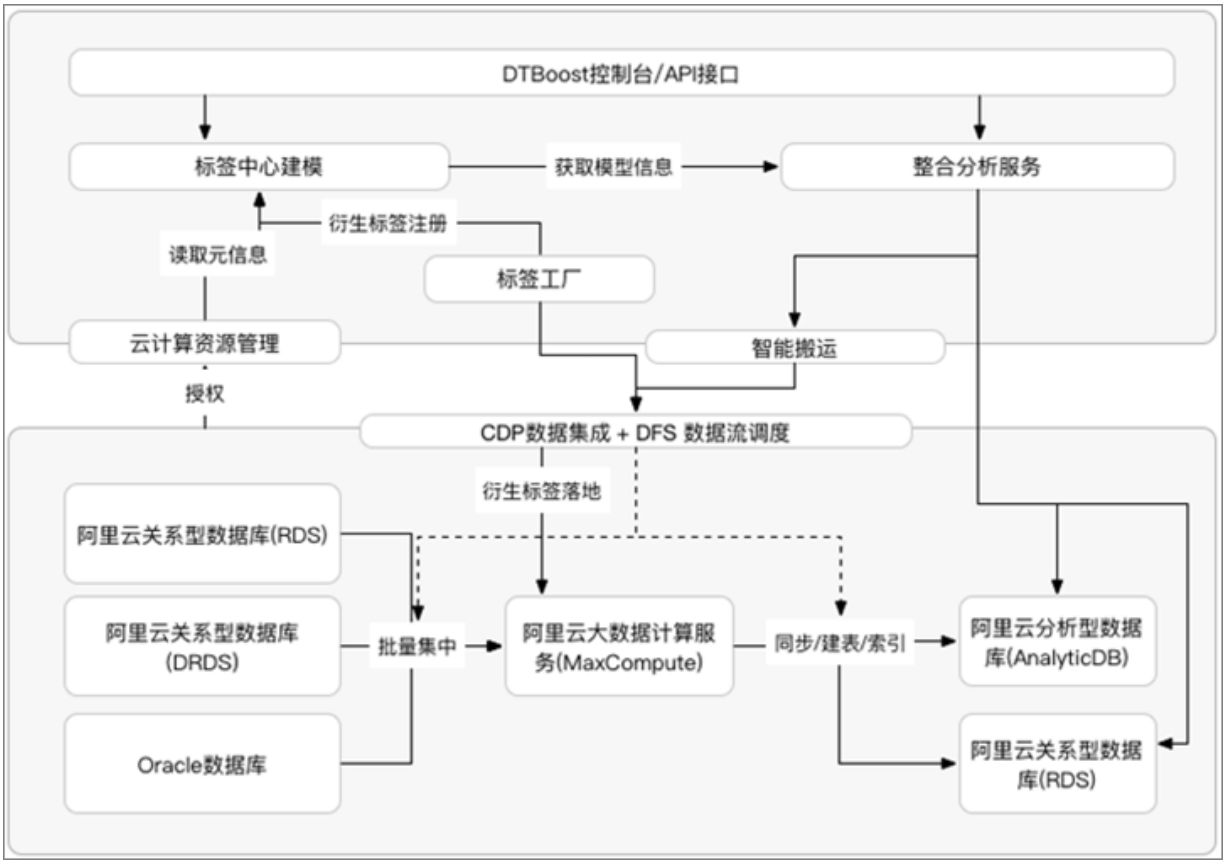
7.3 产品架构

在大数据环境下，一个数据应用往往需要通过多个计算资源来配合完成，最简单来说，一般数据需要先在离线环境当中进行离线加工处理（ETL），再同步至在线数据库当中进行在线分析查询（OLAP）。那么标签中心所能够做的就是与多个数据库进行通信，获取多个计算存储资源的数据元信息后进行逻辑建模，并把各个数据服务模块接口传入的指令解析后将真实的计算命令传给每一个计算资源。

以最常见的OLAP分析场景来看，一般需要从业务库当中将数据进行抽取，加载到大数据（离线）计算服务MaxCompute当中，然后进行相应的加工、衍生后，再把所需要分析的数据同步到在线分析库（在大数据量下通常会使用分析型数据库AnalyticDB）中。

下面以DTBoost数据服务模块中的整合分析为例，为您介绍DTBoost的总体架构。

图 7-2: 产品架构图



如图 7-2: 产品架构图所示，您从DTBoost控制台或API进入，通过把自己的云计算资源授权给DTBoost后，即可通过DTBoost读取各个云计算资源中的数据元信息。

经过建模配置后，在相关的数据服务模块中可以进行手工/自动触发标签中心的智能搬运模块，通过把相关的数据同步调度任务发送给数据流服务DFS（Data Flow Service）和数据整合CDP，来对所需要整合的数据以标签粒度来进行业务库到离线数据仓库的批量大集中，以及到在线分析数据库的同步、建表、索引工作。

数据准备完成后，便可通过相关的数据服务API接口或者在控制台上基于标签模型视图之上进行相关的计算。对于当中需要离线计算加工的部分，一些常用的加工可以通过标签工厂来对标签进行批量的衍生（如常见的聚合、筛选组合等）落地到大数据计算服务（MaxCompute）中。

整个过程可以看作DTBoost在大数据平台之上对各个计算资源之间满足常见业务场景的架构方案进行了系统集成，简化了各个系统之间手工对接等过程。

7.4 功能特性

7.4.1 标签中心

以逻辑模型的建立提到耗时耗力的传统数据仓库物理模型，为分布在多个计算存储资源上的明细数据建立统一的**实体>关系>标签**模型，并将数据一键加速整合入阿里云分析型数据库等在线分析库当中，为分析应用做好数据准备。

功能简介

- **数据资源整合**

  标签中心提供一种业务视角的数据发现、模型探索的工具，便于业务人员、开发人员、数据管理人员透视企业的数据资产。

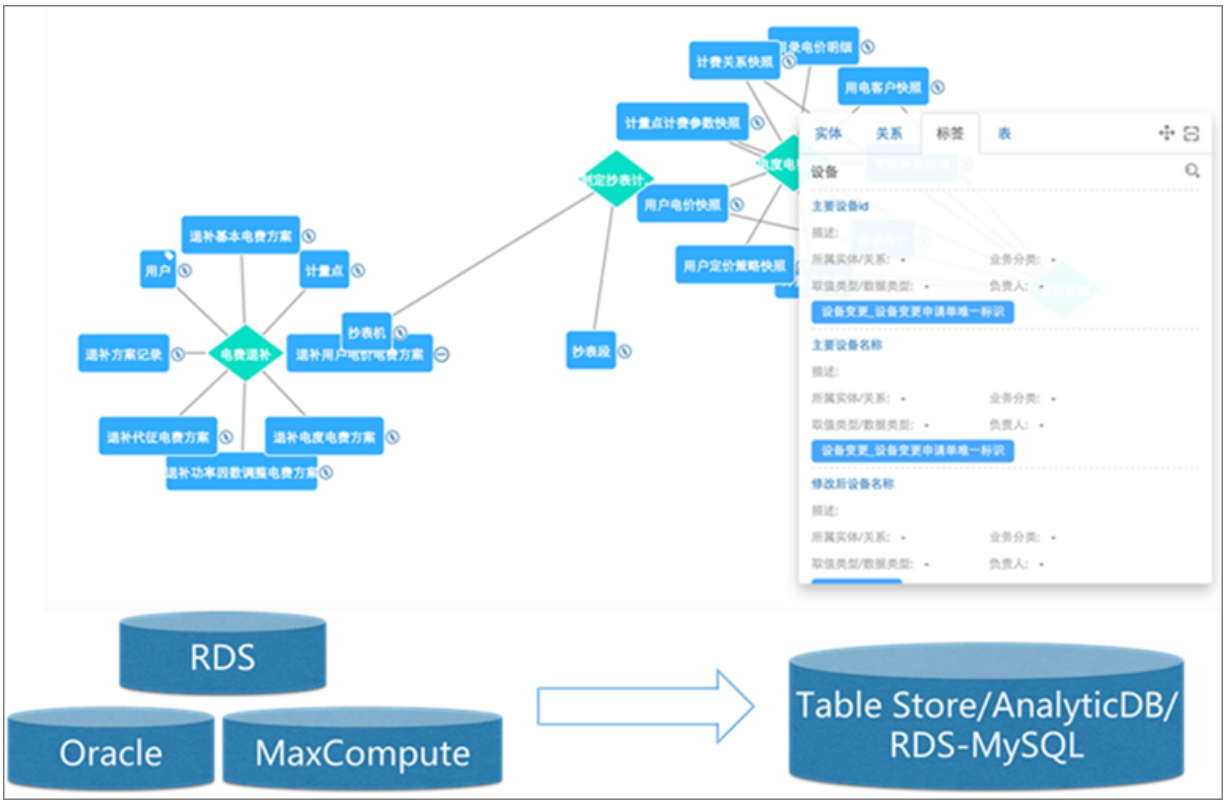
- **为数据服务提供视图支撑**

  为多个计算引擎上的数据提供一个统一的数据视图，结合数据服务能够屏蔽下层多个计算资源的对接与数据同步。

- **数据访问管理**

  可以通过逻辑层对数据访问权限进行有效控制，比物理表的访问管理更加安全有效。

图 7-3: DTBoost功能



产品特性

- **业务视角管理**

  围绕实体>关系>标签这三个元素进行建模，是从业务的角度出发对数据进行组织管理，而不是从表的概念出发进行建模，便于应用层对数据运用和管理的理解、操作，以近似于概念模型的形态透出，让人人都能看得懂。

- **跨计算的统一逻辑模型**

  传统建模的数据来源和模型的使用一般在同一数据库当中，而大数据环境下因为数据采集类型的多样性，和数据计算的多样性使得来源和使用分散在不同的计算存储资源当中，数据产生与加工首先就可能分布在不同的数据库当中，其次同一份数据需要进行跨流式、Adhoc类多维分析、离线算法加工等多种方式的计算，数据需要能在多个存储和计算资源当中自由流转。

- **灵活拓展性**

  表/标签之间的逻辑关系的建立也是在逻辑层上完成的，这就使得模型的维护是可以动态建设的，便于模型的维护和管理，而无需在物理层将数据进行归并后再使用。每一个标签之间可以独立使用，这种离散的列化操作方式也使的数据的使用上更为灵活。

7.4.2 整合分析

在逻辑模型上可通过快速自主封装可实时计算的分析接口透出到前端，或是在线可视化界面配置生成可以独立部署交互式分析应用代码，实现敏捷的交互式分析应用搭建。

功能简介

- 能够提供可视化的界面配置工具，在标签中心所管理的模型基础上，生成交互式分析应用。
- 能够支持交互式筛选、钻取、计数、列表、统计分布、地理分布等分析操作。内置多种可视化图表，包括常规条形图、饼状图、折线图、地图、表格等。
- 拖拽查询配置支持多种查询模式，支持子查询的配置。支持筛选数据导出配置，方便与其他系统打通。
- 所生成应用以源代码方式提供，能够自由修改界面样式，独立部署。

图 7-4: 整合分析功能

图 7-5: 分析结果

产品特性

表 7-1: 一般BI工具和整合分析对比表

对比项	一般BI工具	整合分析
查询灵活度	一般以一层关联居多	与标签中心模型配合，可自定义配置多层子查询
分析交互	以固定报表查看为主	强化交互式SQL模型不定由您自己选择的分析
可拓展性	拓展性较低，以固定产品	开发应用层的源代码，可以自行进行集成拓展
系统集成性	一般作为独立分析工具，不与其他系统集成	可方便地完成下游系统的对接（如广告投放系统）

7.4.3 规则引擎

使用者以图形化操作形式，结合专家业务经验配置自定义规则，规则引擎实时处理数据，当自定义规则命中时、或当算法模型捕捉到业务数据中的特定模式时，实时返回结果。

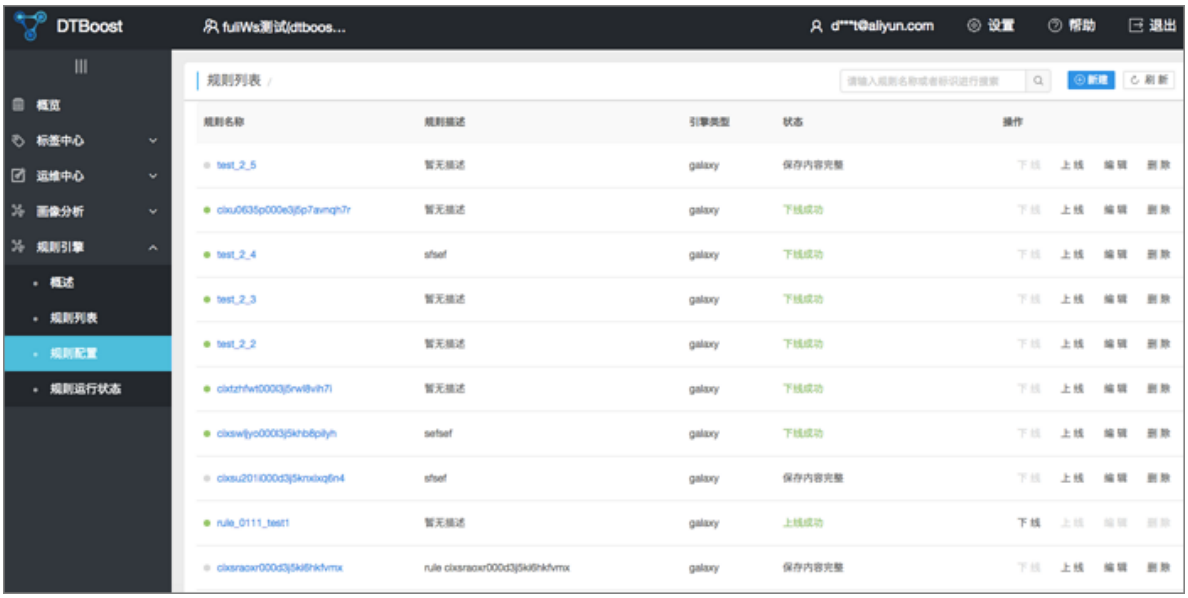
基于流计算的规则引擎适用于业务决策的实时评估，包括风险拦截和预警、资源分配、流程改进等，特别适用于在状态或行为数据频繁变化的场景下，针对特定的业务目标进行实时决策。

功能简介

- **规则列表**

  规则列表展示了用户创建的规则的列表信息，并可以对已上线的规则进行**下线**操作，对已下线的规则进行**上线**、**编辑**、**删除**等操作。

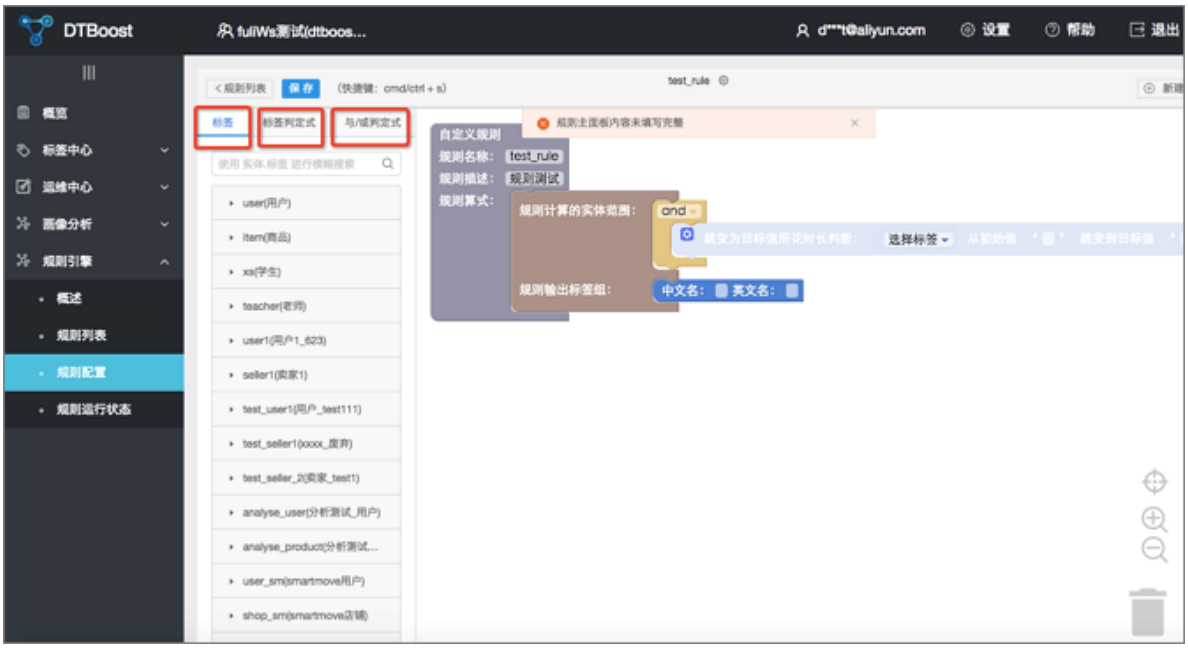
  图 7-6: 规则列表



- **规则配置**

  用户可以创建规则并对其进行配置操作，目前支持标签、标签判定式、与/或判定式三种自定义规则。

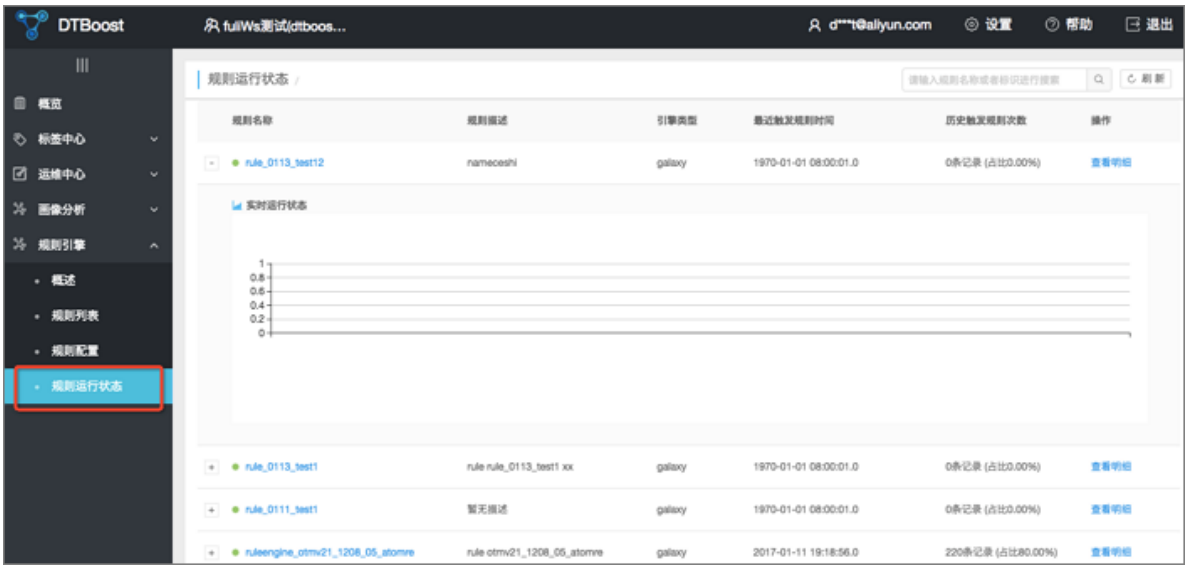
图 7-7: 自定义规则



- **规则运行状态**

  对于已经上线的规则，可以在规则运行状态页面查看规则实时运行的状态和相关的历史触发次数。

图 7-8: 规则运行状态



您也可以查看规则的明细信息，如最近触发时间、数据产生时间和规则明细信息。

图 7-9: 规则运行状态明细

产品特性

- **拥有海量实时数据流处理能力**

  规则引擎使用了高效的流计算引擎，规则所对应的数据量每秒可达到上万条，规则所对应的数据量每秒可达到上万条。例如在工业场景中，如果有上万台设备，每个设备上有数十个传感器，若在规则引擎中进行规则配置，基于传感器数值对设备进行监控，可以支持每个传感器每隔1秒上传状态数据。

- **支持十万级别规则并发**

  规则引擎支持十万级别规则的并发执行。例如在电商行业中，若用户出现违规行为，需要实时告警。告警出现延时则可能让欺诈得逞，带来钱款的损失。对于这个场景，需要由数十名小二制定数万条规则。由这些规则同时对海量电商用户进行监控，发现其中的违规欺诈行为，每个用户每时每刻的行为都将接受数万规则的检测。

- **支持时间划窗、时空轨迹等流式计算复杂规则**

  规则引擎支持基于时间窗口的规则计算，可以在一段时间窗口内进行统计或计算。例如，判断设备温度在1小时内升高是否超过100%，或判断潜在用户在过去3天购买兴趣是否包含3C数码。

  同时，规则引擎还支持判断预设的多个条件是否按照预设顺序被触发。例如风险卖家先提取了全部现金，再拒绝了全部退款申请。

- **支持规则热升级，升级前后状态数据不丢失**

  规则引擎中的规则通常按照专家的经验进行配置，然后根据实际执行的结果或实际的数据进行调整。规则引擎提供了规则配置热升级的功能，可以确保在规则参数修改后，规则执行的中间状态不会丢失，对于包含时间窗口的规则尤为有用，同时也可有效避免规则上下线过程中的漏报。

7.4.4 标签工厂

标签工厂的设计思路是进行数据的再次开发，开发出来的数据落地后可以继续二次开发，不断满足不同的业务场景需求。设计中根据原有OLT标签体系中的标签数据进行再加工生成新标签数据并落地到OLT标签体系。

功能简介

- **计划**

  计划（scheduler）是用户创建的每个生产标签的任务，是指根据原有OLT标签系统数据通过SQL、TQL或算法生成新的标签数据并关联到对应的OLT模型的一条记录内容。它包含有原有的OLT标签、SQL或TQL或算法部分、以及新产生标签数据并关联到对应的OLT模型的关系映射。

  目前支持图形界面配置计划任务、文本直接写TQL配置计划任务、算法配置计划任务三种形式。

  计划通过提交、执行就可以将产生的新的标签数据落地到OLT标签系统。

- **任务**

  计划执行之后，就会在任务列表里面有对应的记录，任务就是计划的执行记录。

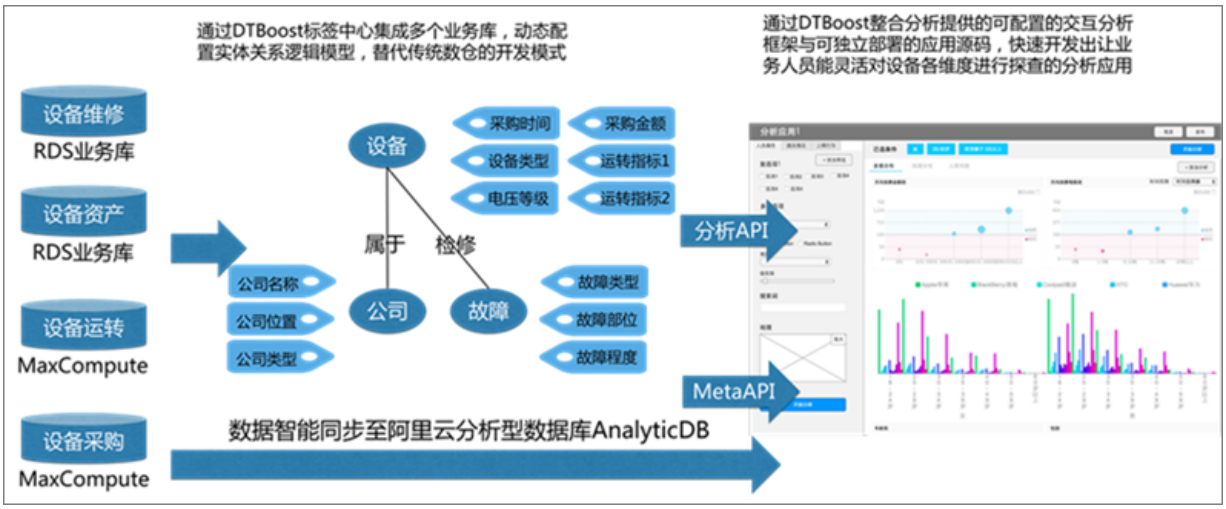
  标签工厂任务列表展示了计划的执行人、提交时间、开始执行时间、执行结束时间、执行状态，同时支持执行参数和日志的查看、执行任务根据执行日期和计划名称（ID）进行全局搜索等功能。

7.5 应用场景

DTBoost适用于以下场景。

- 结构化数据全量数据200G以上。
- 希望把多个业务系统/小数仓的数据进行整合打通。
- 分析场景围绕一个主体展开（人、设备、车辆...）。
- 数据维度（包括衍生出来的维度）超过10个以上。
- 业务人员希望自己来进行分析，而不是只看固定报表。

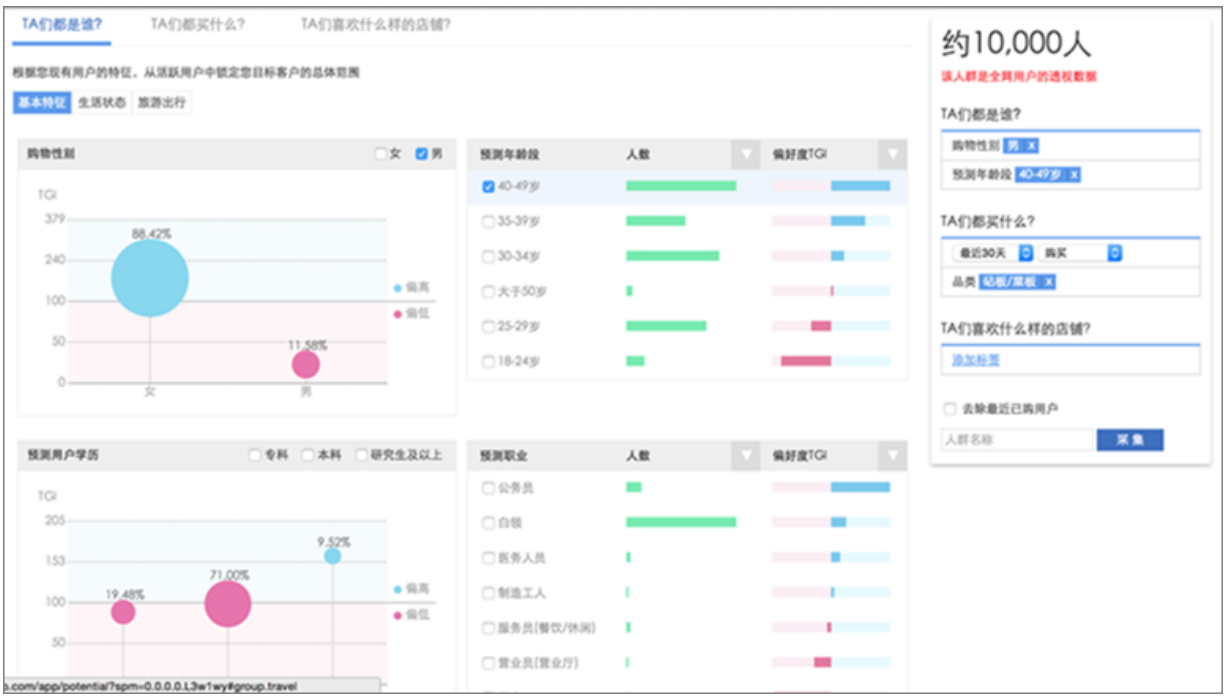
图 7-10: DTBoost应用



7.5.1 画像分析

整合用户收藏、成交、点击、注册信息、定位、以及衍生加工的标签等多份数据，全方位分析用户各种行为之间的关联性，从而更有效的设计交叉销售、营销内容、人群定向等运营策略。

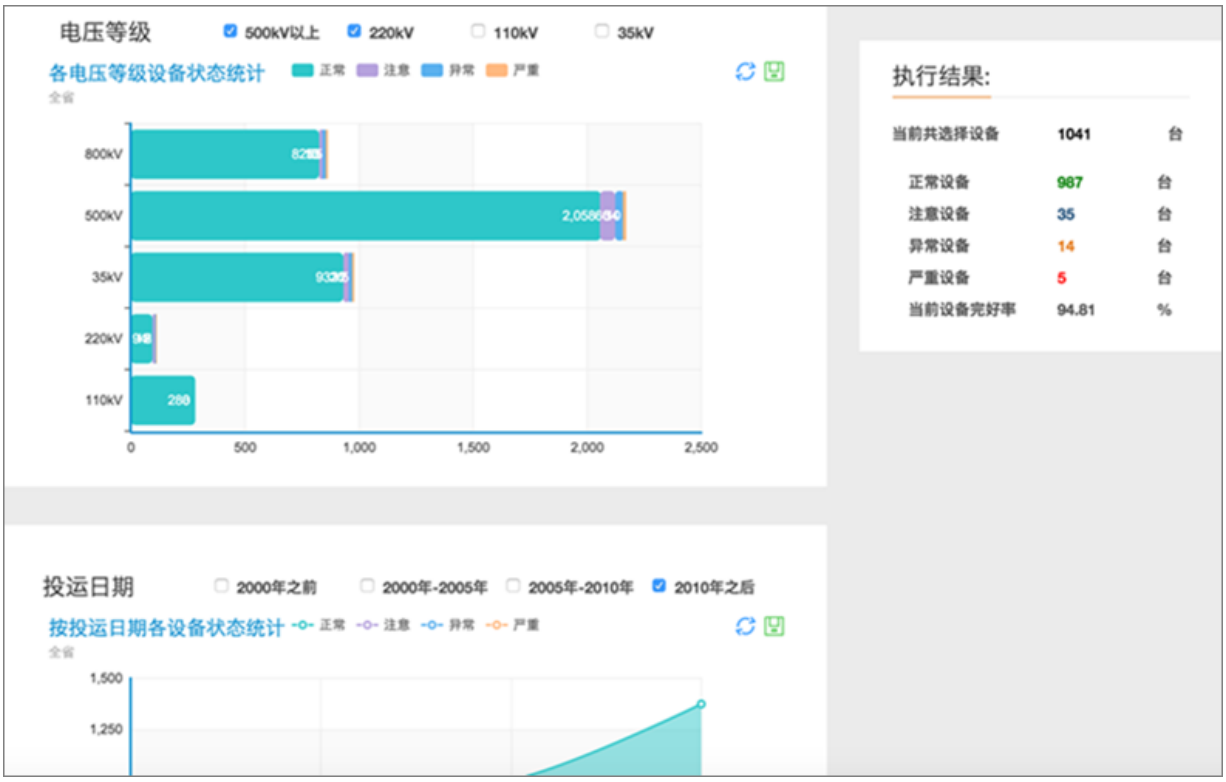
图 7-11: 画像分析



7.5.2 设备履历

打通设备在采购、运维、检修、报废、技改等多个环节的数据，能全方位的对设备资产情况进行分析梳理，以及剖析各种外界的数据对设备状况的影响，大大提升设备资产管理水平。

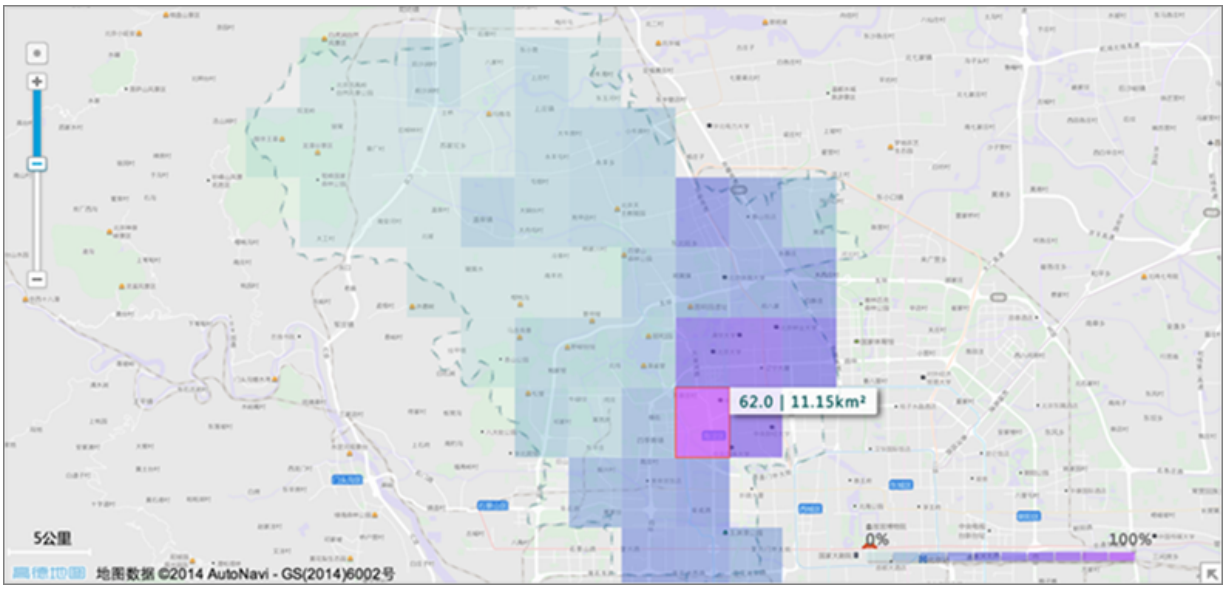
图 7-12: 设备履历



7.5.3 地理分析

结合地理位置信息，无需在每个地理网格上的分布提前进行预计算，直接在POI明细数据之上进行多种多样的筛选、汇总分析，大大提高了时空数据分析的灵活度。

图 7-13: 地理分析



7.6 使用限制

无

7.7 基本概念

实体（Object）

实体是分析的主体，是外部物理世界真实事物的抽象，比如安全场景的公民、火车、酒店、网吧、案件等。

关系（Link）

关系建立在两个及两个以上的实体上。比如安全场景的公民火车出行、公民酒店入住、公民网吧上网、公民与案件的犯罪关系。

标签（Tag）

标签挂接在某个实体或者某个关系上。

- 公民的姓名、年龄、籍贯等基础标签属于公民这个实体。
- 火车出行的时间、出发站、目的地属于公民火车出行这个关系上的标签。
- 酒店住宿入住时间、酒店住宿天数属于公民酒店住宿关系，以此类推。

TQL

标签查询语言（Tag Query Language）是分析接口支持的一种查询语言，基于实体、关系、标签的一种类似SQL的查询语法。其语法和SQL中的select语法非常相识，但不支持update、insert、delete等其他语法。

JQL

结构化标签查询语言（JSON Query Language）是分析接口支持查询输入语言，基于实体、关系、标签的描述，JSON格式的语法。其语法表达的含义和TQL相同。

8 大数据管家BCC

8.1 什么是大数据管家

大数据管家Big Data Manager（原名BCC）是为大数据产品量身定做的运维管理平台。当前运维的大数据产品包括MaxCompute（原名ODPS）、DataWorks（原名大数据开发套件）、AnalyticDB（原名ADS）、Quick BI、IPlus（关系网络分析）、OSS（对象存储）、Apsara（飞天操作系统），并为这些产品提供服务监控、日常自检、日志搜索、运维、配置以及服务特性运维功能，维护产品稳定性。

大数据管家以服务组件的形式给与产品提供运维功能，每个服务组件包含产品树结构、配置、自动化服务自检、工作流、包管理、日志搜索、指标信息和Metrics信息，还包括各服务组件自定义的一些功能。

通过大数据管家的赋能，专有云驻场人员可以轻松地管理大数据产品，比如查看大数据产品的运行指标，修改大数据产品运行配置，对大数据产品进行自检，搜索日志等功能。

8.2 产品优势

集群健康状况监控

支持对大数据产品集群的设备、资源以及服务进行状态监控、配置管理，收集与展现集群的实时运行状态信息。

数据化分析资源变化趋势

支持收集集群的设备、资源与服务的实时运行状态和历史数据，并对收集的信息进行聚合分析，以分析和评估集群的健康状态。如果分析和评估结果存在风险，还可以实时推送给相关责任人。

图形化界面运维操作管理

可视化展现系统的各类运行信息，以及常见的运维操作。

面向开发者的性能优化建议

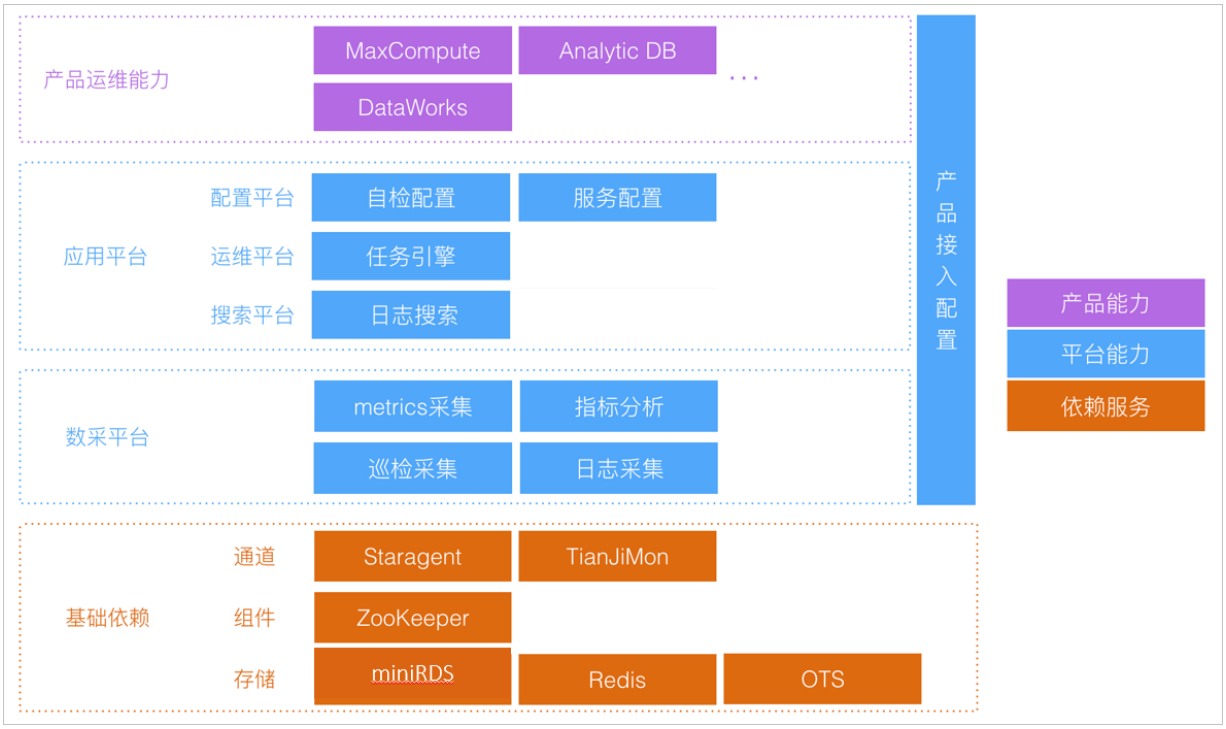
针对MaxCompute、AnalyticDB等大数据计算引擎，提供产品特有的性能优化分析与建议，提升开发者效率。

8.3 产品架构

大数据管家采用微服务设计理念，在此之上提供数据整合、接口整合以及功能整合的统一整合平台，暴露统一标准的服务接口。基于这样的设计理念，在大数据管家上的不同产品的页面操作和运维体验完全一致，减轻现场运维学习成本，降低运维风险。

整个系统主要由基础依赖、数采平台、应用平台、以及产品运维能力组成，如图 *8-1: 产品架构*所示。

图 8-1: 产品架构



8.3.1 基础依赖

大数据管家采用了集团统一的通道服务staragent和TianJiMon来完成远程命令和远程数据采集指令，通过zookeeper来协调主从服务，保证服务可用性。在存储方面，大数据管家主要使用miniRDS来存储元数据，使用redis作为缓存服务，使用OTS来存储大量的自检采集信息，提升吞吐量。

8.3.2 数采平台

依赖TianJiMon，大数据管家开发了针对实时运维数据采集的采集脚本和针对服务状态采集的巡检脚本，通过回收这些脚本采集的数据，可以快速了解服务整体运行数据和状态，通过一定规则的聚

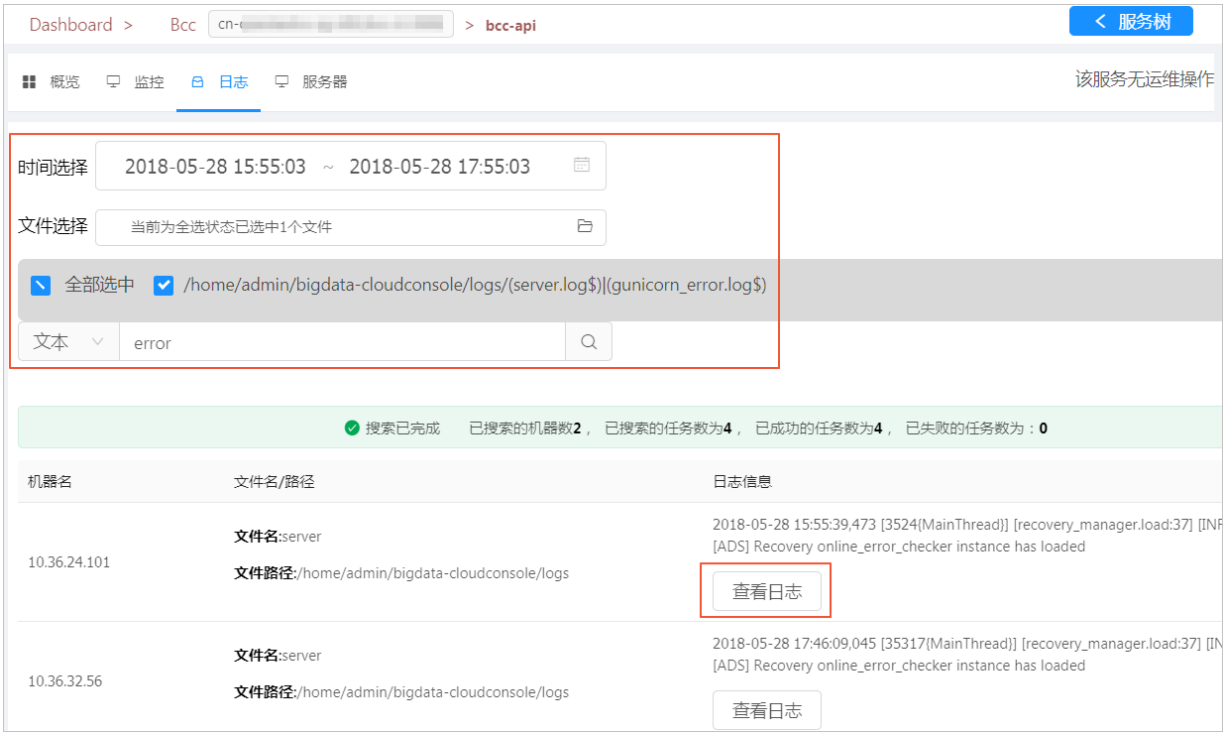
合、筛选和甄别发现有价值信息。日志采集则是依赖staragent完成实时采集任务，让您可以实时分布式抓取日志信息。

8.3.3 应用平台

应用平台根据底层提供的能力以及数据采集的数据信息，构筑出多个提供上层服务能力的平台，这里主要包含配置平台、运维平台以及搜索平台。

- 配置平台可以方便的让您了解指定服务的所有配置信息，主要包含自检的配置和服务本身的配置，并且这些配置可以在这里修改提交。
- 运维平台主要提供运维的能力，并且可以智能识别您当前所在的服务而动态填写参数。
- 搜索平台主要提供了日志搜索能力。日志搜索则提供了针对指定服务的指定日志路径的实时日志搜索能力，如图 *8-2: 日志搜索*所示。

图 8-2: 日志搜索



8.3.4 产品运维能力

配置平台、运维平台以及搜索平台并不意味着就是所有大数据管家上提供的产品运维能力，因为部分跟产品有关的定制化运维功能则依托在每一个产品运维能力本身上。他们都会深度的与具体需要运维的产品融合和定制，达到一致性和特殊性的统一。例如AnalyticDB 会在日志平台的能力上，再

提供更高层次封装的针对其服务的查询分析，并结构化展示分析数据，让您可以快速了解日志信息。这些定制都是透明的，您在具体运维场景上会顺利地了解这些定制内容。

8.4 功能特性

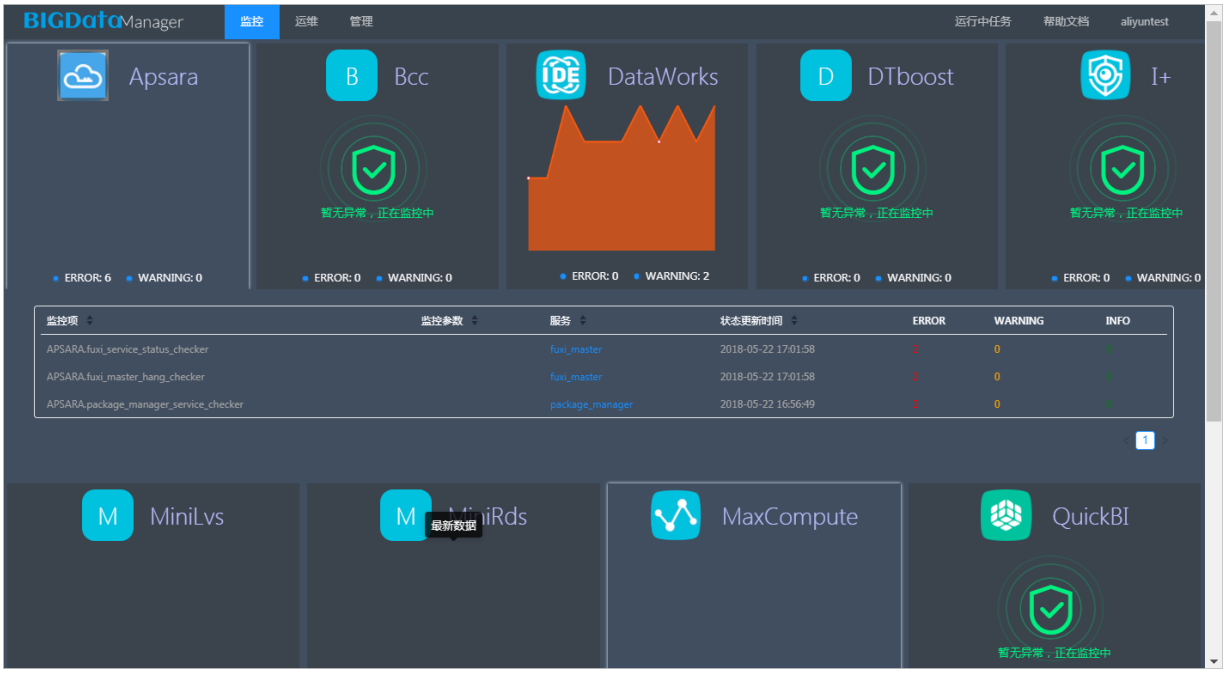
下面对大数据管家的主要功能做一些介绍，详细的功能使用可以参考《运维指南》中的**大数据管家**章节。

8.4.1 监控

大数据管家支持自动对Apsara、AnalyticDB、MaxCompute等所有监控的产品进行业务健康检查，并展示检查结果。

登录大数据管家后，大数据管家首页即**监控**页面，如图 *8-3:* 监控概览所示。

图 8-3: 监控概览



如果产品存在告警和系统错误，则提示告警和系统错误数量，同时还支持查看告警和系统错误详情。

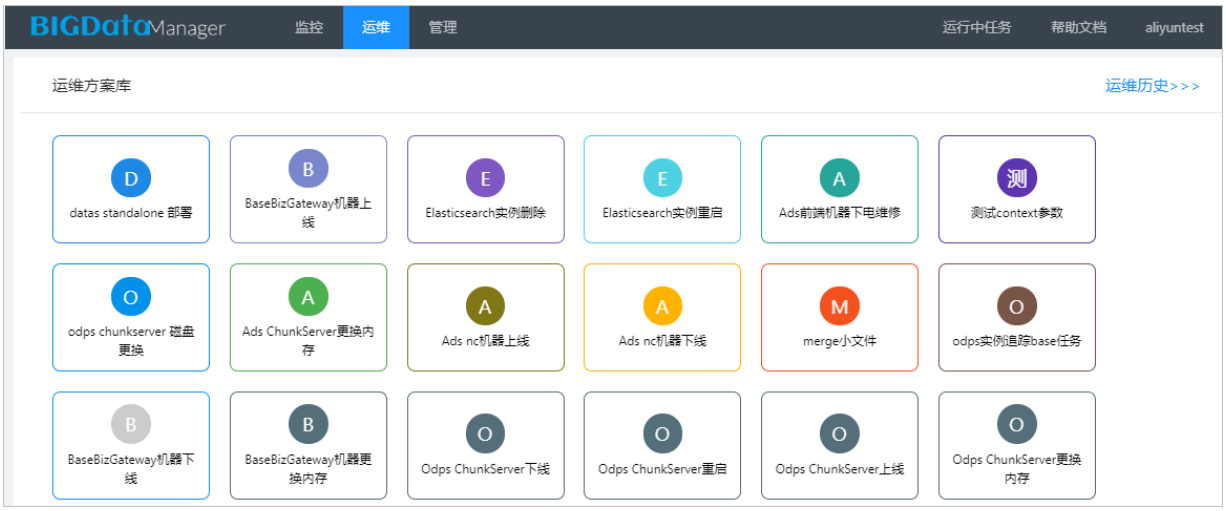
监控页面是各产品运维的入口，您可通过单击某产品图标中的产品名称进入该产品的运维界面。

8.4.2 运维

运维是大数据管家提供的常用运维案例库。通过运维案例库模板，您只需要填写少量参数即可为您的实例环境创建一个运维任务。

运维案例库界面如图 *8-4: 运维案例库*所示。

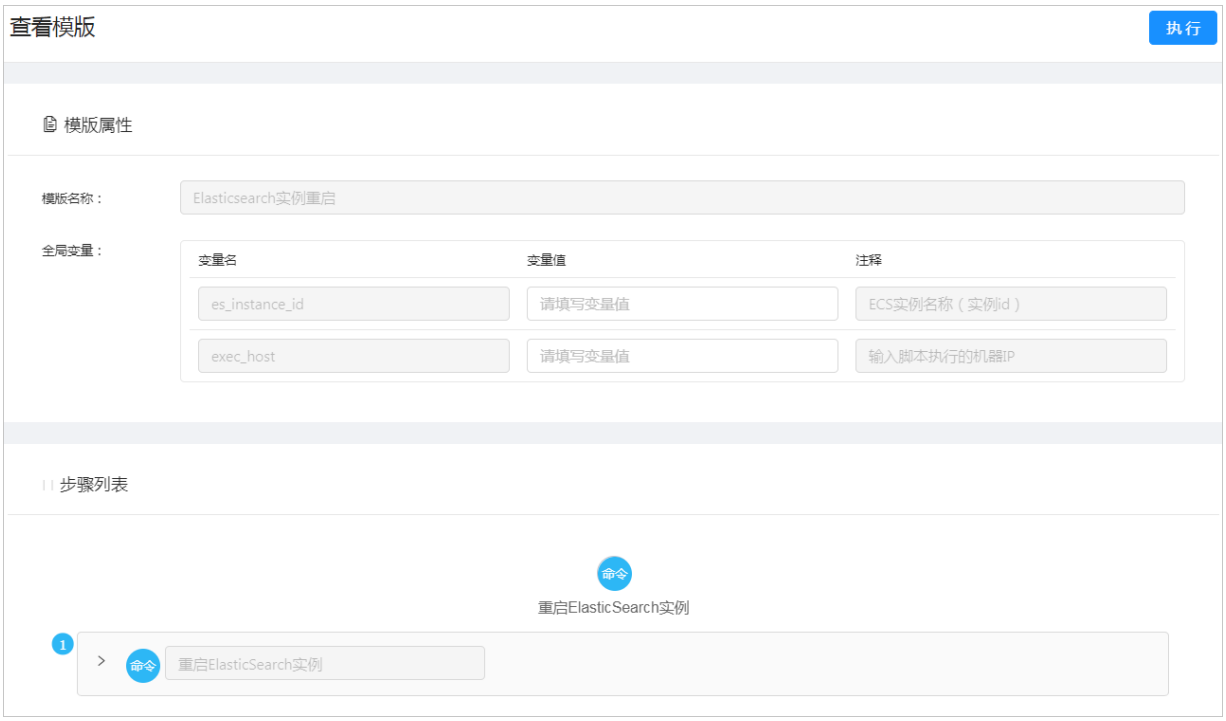
图 8-4: 运维案例库



每个运维案例均包括**模板属性**和**步骤列表**。**模板属性**和**步骤列表**中大部分参数均已固定，不可修改，用户只需要根据实际情况填写实例相关的参数即可执行。

运维案例界面示例如图 *8-5: 运维案例界面-示例*所示。

图 8-5: 运维案例界面-示例



模板属性是实例环境的参数，您需要根据实际环境填写相关的设备和实例参数。

步骤列表的操作步骤包括脚本步骤和人工步骤。脚本步骤会自动执行，执行过程中自动映射**模板属性**中填写的参数。人工步骤需要用户根据步骤描述对实例进行相应操作。人工步骤正确执行完成后，系统会继续执行脚本步骤，直到执行完所有步骤。

8.4.3 管理

大数据管家提供docker管理、云账号管理、oss管理、租户管理、计算引擎和热升级管理的功能，方便您对各产品进行运维管理。

docker管理

Docker是一个应用容器引擎，大数据管家为开发人员提供docker管理功能，方便开发人员快速的更新产品服务包。与在天基更新产品服务包相比，通过docker管理功能更新产品服务包时，您不需要更新整个docker的镜像，只需要替换docker中有更新的文件即可。

云账号管理

云账号是大数据各产品的运维、管理、配置账号。在使用各产品前，您首先需要在大数据管家开通一个云账号。大数据管家支持新增、修改云账号，并支持授权和回收云账号在大数据管家上对各产品进行运维的权限。

说明：

云账号创建后，不支持删除，请勿随意添加云账号。

oss管理

大数据管家支持管理对象存储服务OSS（Object Storage Service），支持bucket创建、删除、查看。

租户管理

租户管理DataWorks特有功能，用于管理使用DataWorks的租户，即DataWorks的项目空间拥有者，每个租户关联一个云账号。租户可以添加项目中的其他成员为租户用户，每个租户用户也必须关联一个云账号。在租户管理页面，您可查看的租户信息包括：租户名称、责任人、云账号、联系人、联系电话、联系邮箱、描述、用户列表。

说明：

，一个云账号仅可关联一个租户或租户用户。

计算引擎

计算引擎是分配给租户的计算资源。大数据管家支持管理租户的计算引擎。

热升级管理

热升级管理用于对大数据管家的监控项进行在线热升级，不会中断业务。

8.4.4 运行中任务

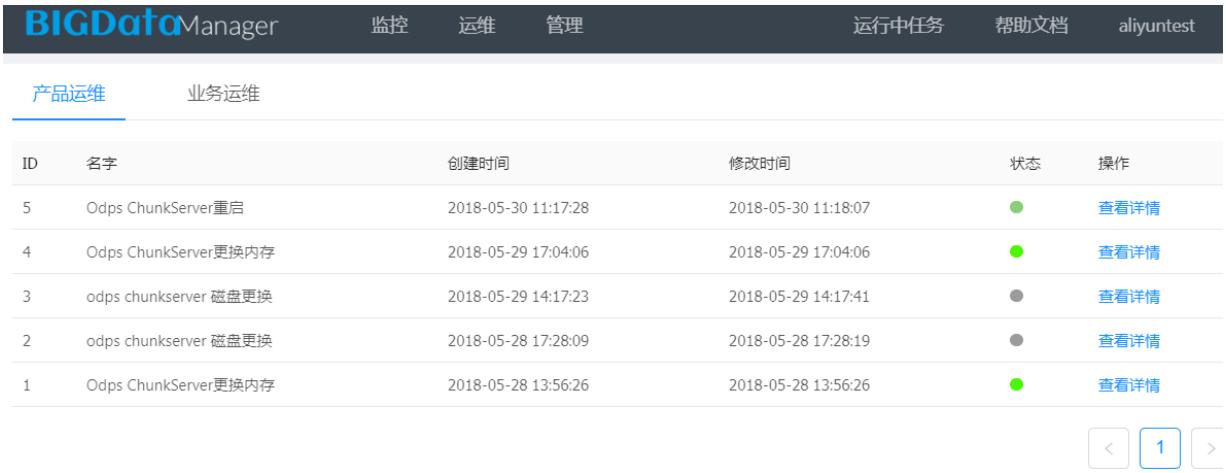
大数据管家中的任务分为业务运维任务和产品运维任务。运行中任务是大数据管家系统中当前没有完成的工作流任务列表，包括执行中和执行失败的工作流。

大数据管家支持查看运行中的任务，如下：

图 8-6: 运行中任务

单击**更多**，进入任务列表界面，如图 *8-7: 任务列表*所示。

图 8-7: 任务列表



任务列表中的任务包括产品运维和业务运维两种，产品运维任务是通过运维案例库创建的产品相关的任务实例，业务运维任务是各具体产品的运维功能创建的业务相关的任务。

在任务执行过程中，产品运维任务和业务运维任务在不同阶段会显示不同的状态，每种状态的图颜色不同，方便你快速查找任务。

通过**查看详情**，您可查看任务的详情。在任务详情界面，您可以对执行异常的任务，进行重试、回滚、取消等操作。

8.4.5 产品界面

通过单击**监控**页面各产品图标中的产品名称，您可进入相应产品的操作维护界面。产品操作界面通常包括概览、监控、服务器等功能，不同产品的功能项可能会有不同。

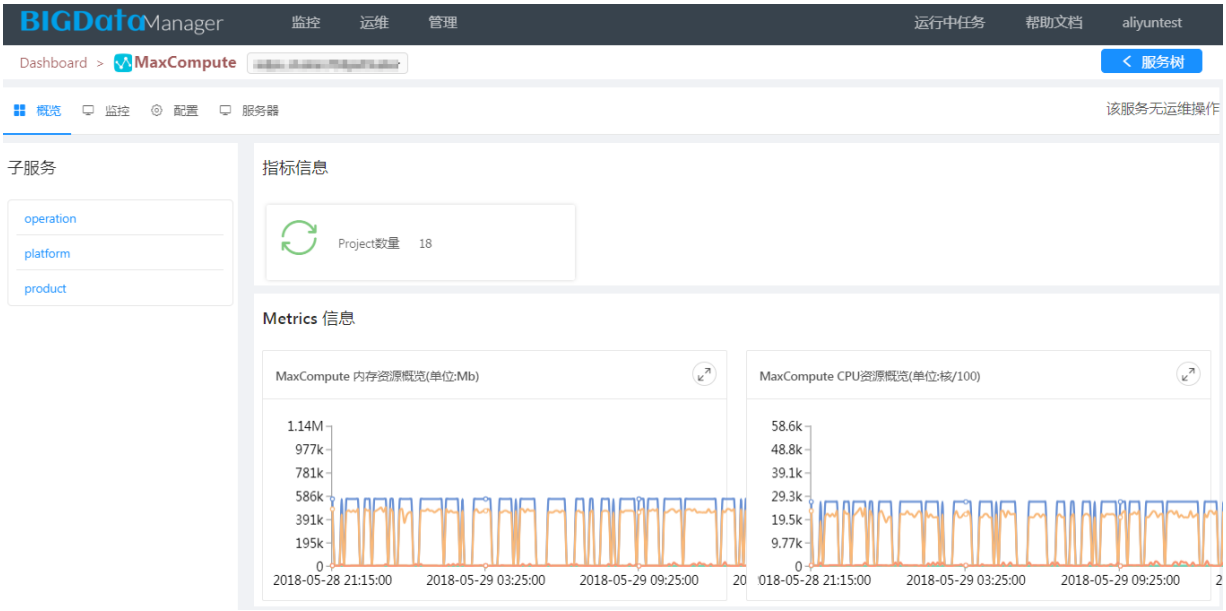
概览

概览页面显示集群的子服务、指标信息和Metrics信息，不同集群的子服务、指标信息和Metrics不同。

概览能够直观的展示当前集群的状况，可通过点击集群下拉列表中的集群名称切换集群。

您还可以通过单击左侧子服务，查看子服务的指标信息和Metrics信息。

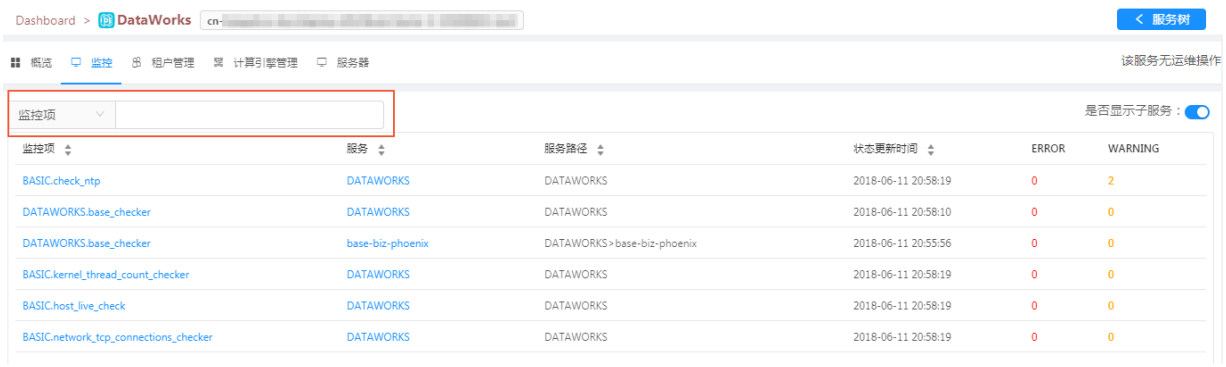
图 8-8: 概览



监控

监控页面大数据库管家所监控的产品均有的页面，用于展示所监控产品的监控项及监控状态。

图 8-9: 监控页面



在**监控**页面，您可通过**监控项**、**服务**、**运行实例**等进行过滤搜索。

通过单击**监控**页面中的**监控项**名称，您可查看监控项实例的历史趋势、监控项信息、主机列表以及监控项在大数据管家平台上的历史趋势。

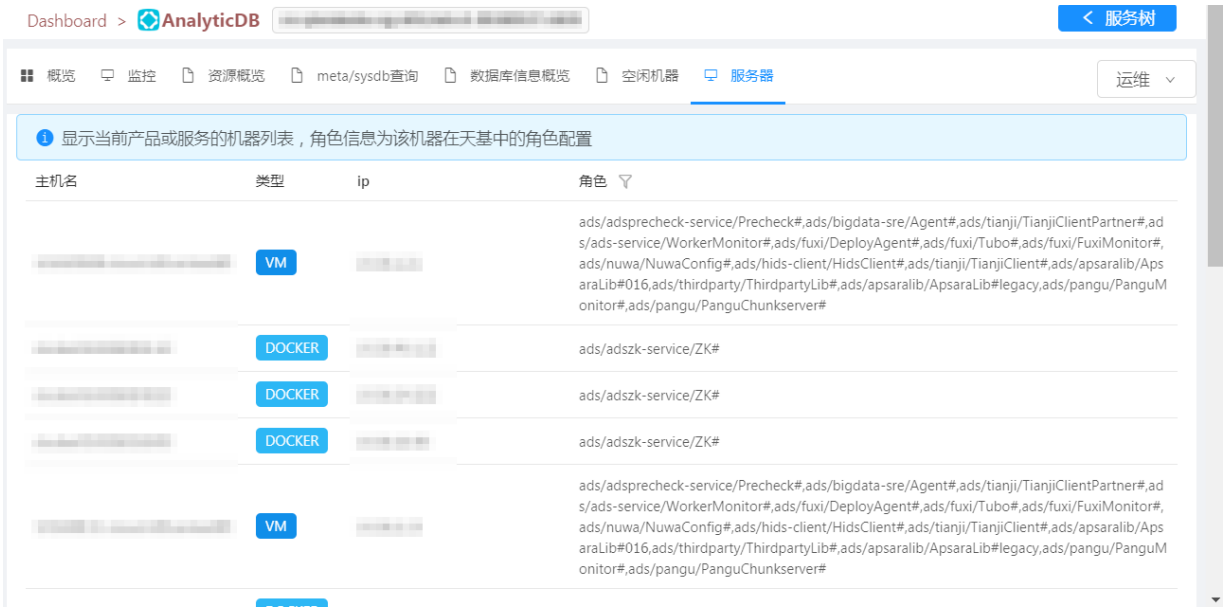
通过主机列表，您还可以查看以下信息：

- 该主机在该监控项实例上的当前输出内容。
- 关联服务：鼠标指向**关联服务**，则显示关联的服务在该监控项实例上的当前报错数量。
- 该主机在该监控项实例上的历史执行情况和输出。

服务器

服务器页面是大数据管家展示产品或服务的机器列表的页面。

图 8-10: 服务器页面



服务器页面展示当前产品或服务的机器列表，机器信息包括主机名、类型、IP地址和角色。其中角色信息是该机器在天基中的角色配置。

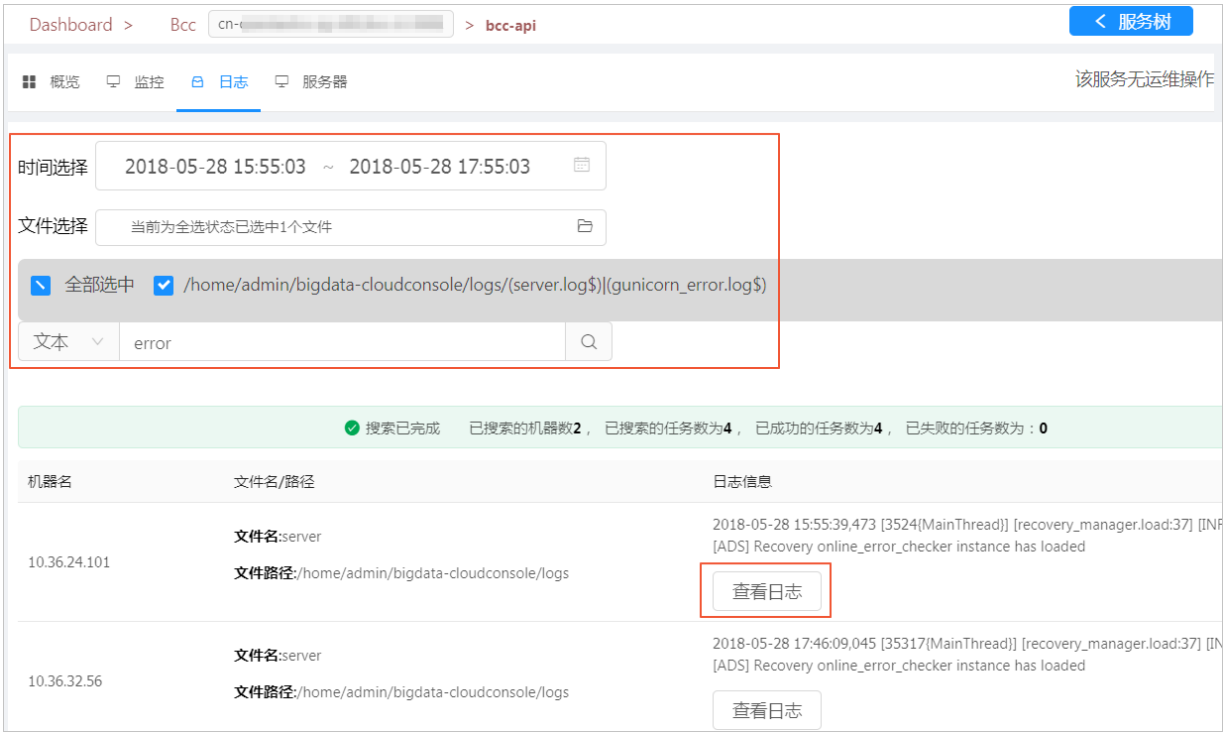
配置

配置是部分产品或产品的子服务所具有的功能，用于对已部署的产品服务进行相关的配置和操作，从而实现您的实际运维需求。不同产品配置功能不同，总的配置类型包括工作流配置、服务配置，自检配置、指标配置、表配置、图配置、日志搜索配置、节点信息配置、采集插件配置、健康配置、基础配置、私有配置、配置项配置。

日志

日志是部分产品或产品子服务所具有的功能，用于获取已部署的产品服务中的各个worker的日志。大数据管家支持对日志进行筛选过滤，您可以方便快捷的查找到所需的日志内容。

图 8-11: 日志



8.5 应用场景

专有云企业版+大数据产品

如果您部署了阿里云专有云Enterprise版且同时部署了任一大数据产品（例如：MaxCompute、AnalyticDB等），则您需要使用大数据管家来对已部署的所有大数据产品进行运维管理。

专有云大数据版+大数据产品

如果您部署了阿里云专有云大数据版，则您需要部署大数据管家来作为用户侧、运营侧和运维侧的统一管控平台，提供全量的面向用户的管理操作。

8.6 使用限制

无。

8.7 基本概念

产品服务树

产品服务树是产品的组织结构体现形式，每个服务组件（产品Product）都是一个独立存在的个体，都拥有任意多个服务（Service），所有服务的组织形成产品服务树。

当进入产品页面时，单击右上角的**服务树**，会左移出来一个新的页面，显示当前产品的服务树并定位到当前界面展示的服务节点；如果当前服务节点包含子服务，则还会显示当前服务节点的所有子服务。服务树中含有服务列表以及对应的节点，通过单击服务或者节点同步切换左侧的产品的服务或者节点。

工作流

工作流是事先根据一系列过程规则定义的处理流程和步骤过程，是封装好的一种框架，能够自动执行。工作流可以解决一些繁琐或者重复性工作。

9 关系网络分析I+

9.1 什么是关系网络分析

关系网络分析，又名Alibaba Cloud I+（以下简称I+）是基于关系网络的大数据可视化分析平台，在阿里巴巴、蚂蚁金服集团内广泛应用于反欺诈、反作弊、反洗钱等风控业务，面向公安、税务、海关、银行、保险、互联网等提供行业解决方案。

I+产品围绕**大数据多源融合、计算应用、可视分析、业务智能**设计实现，结合关系网络、时空数据、地理制图学建立可视化表征，揭示对象关联及与时间、空间密切相关的模式及规律。目前产品已具备关系分析、路径分析、群集分析、共同邻居、骨干分析、伴随分析、时序分析、行为分析、统计分析、地图分析等功能，以可视化的方式有效融合计算机的计算能力和人的认知能力，获得对于大规模复杂网络的洞察力，帮助用户更为直观、高效地获取知识和信息。

I+产品是依托于关系网络，从时间、空间维度来提供交互式分析的大数据处理平台。主体业务模块有：搜索、关系网络，时空分析三大部分。同时产品主要着力于安全相关领域，权限控制也是产品重要模块。

9.2 产品优势

海量数据实时挖掘

I+能够在百亿节点、千亿边、万亿记录的PB量级数据中，按照用户的业务指令进行关系挖掘和时空计算，并且实时交互响应。

模型认知万物相连

I+用OLEP模型认知世界万物相连，以实体（Object）、关联（Link）和事件（Event）显像表征，以属性（Property）实现异构数据的理解和整合。

业务场景灵活赋能

I+自带以OLEP为核心的中枢控制，通过业务配置和感知实现人机交互学习，赋能公安、欺诈、金融、税务等不同场景业务研判。

可视分析高效体验

I+全面分析潜在用户体验要素和业务痛点，沉淀出数据、交互、结果的分阶可视化体验和协同共享，使得有证可查，有据可说。

智能深入以人为本

I+精准智能化帮助业务员思考学习，解决常见的业务难题，目前已有涉恐指数、伴随分析、亲密度、涉毒指数等深度训练模型。

重大项目深度考验

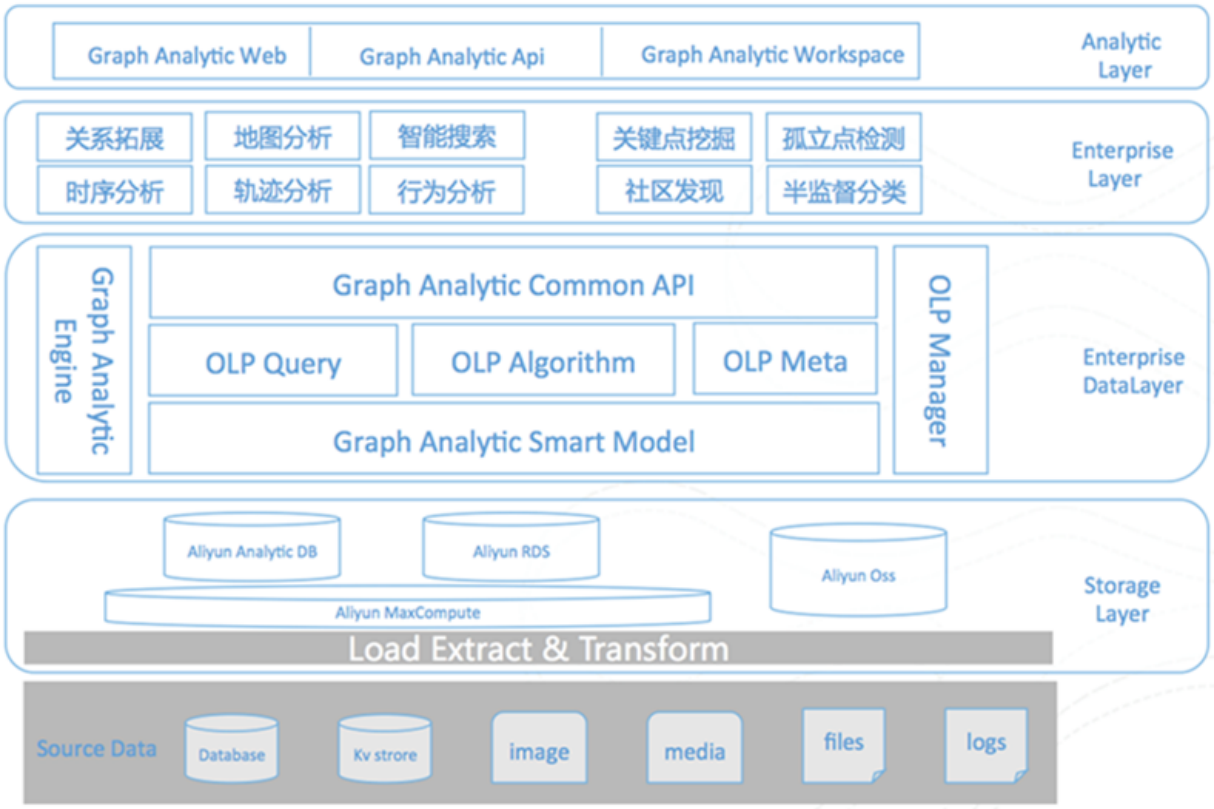
I+已经作为亮点应用参与多个国家级重点项目，在安保、反恐、关税等领域让客户耳目一新，业务价值得到深度考验和认可。

9.3 产品架构

9.3.1 系统架构

关系网络分析软件采用组件化、服务化设计理念，多层次体系架构。数据存储计算平台建立在阿里云自主研发的大数据基础服务平台（数加平台）上，支持PB / EB级别数据的存储和计算，具有强大的数据整合、处理、分析、计算能力。

图 9-1: 系统架构



整个系统分为存储计算层、数据服务层、业务应用层、分析展现层。

- **存储计算层**

基于阿里云大数据平台，支持多种开放数据源。计算平台分为离线和在线，离线计算为MaxCompute，实现数据的整合、处理，在线计算实现数据的实时计算，包括分析型数据库AnalyticDB、图数据库（BigGraph）、流计算（StreamCompute）。

- **数据服务层**

  按照关系域、关系类型、关系事项抽象出的“实体 - 属性 - 关系”模型，提取自然对象关系、社会对象关系、空间对象关系，进行业务关系逻辑建模，通过逻辑业务定义整合异域多源的数据，支持逻辑模型的灵活管理和维护。数据服务引擎为业务应用层提供统一的业务逻辑查询语言，执行各种复杂的关系网络查询、算法分析。

- **业务应用层**

  将关联网络、时空网络、搜索网络、信息立方、智能研判、协作共享、动态建模等多业务业务应用封装成API接口，提供给分析层调用。

- **应用分析层**

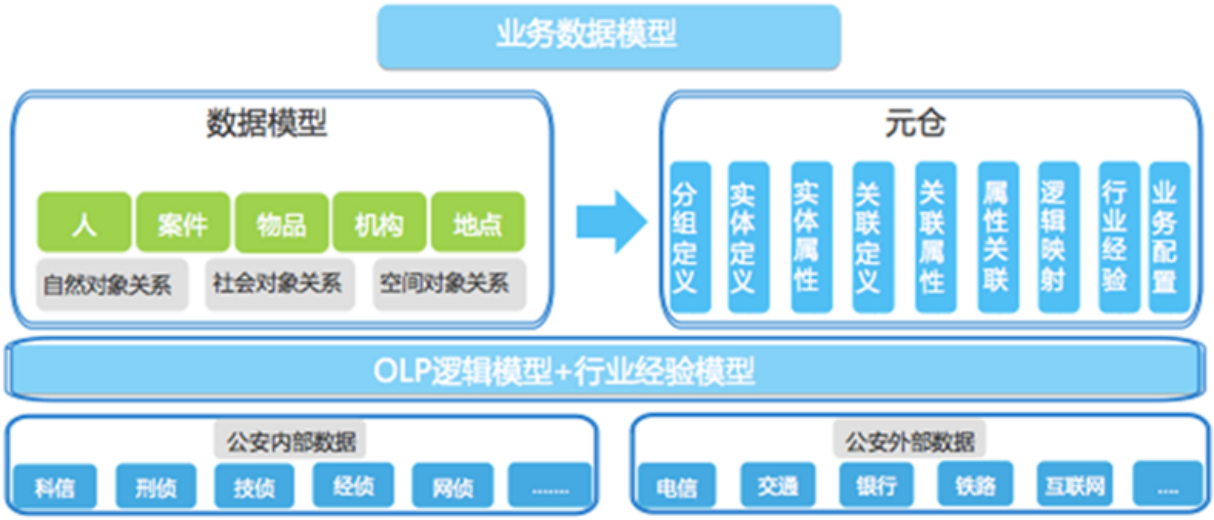
  提供多元智能可视化交互分析界面，支持多种终端。提供外部API接口及可视化组件服务，支持第三方系统的接入。

9.3.2 OLEP模型

OLEP 模型是以实体（Object）、关联（Link）和事件（Event）对现实世界建模，通过属性（Property）实现异构数据的整合。OLEP模型是I+重要的数据模型，是I+接入关系实体数据的关键，是I+构建万物连接的基础。

以公安行业为例，公安行业用到的数据包括公安内部数据和公安外部数据，在这些物理数据之上抽象OLEP逻辑模型和行业经验模型，最终将数据模型映射为实体定义、实体属性、关联定义、关联属性等元数据定义，如图 *9-2: OLEP模型*所示。

图 9-2: OLEP模型



以高铁为例，张三、李四、王五乘坐上海至杭州的高铁，他们之间可能存在同车次、同车厢、同出发地、同达到地等关系，如图 *9-3:* 高铁示例所示。

图 9-3: 高铁示例

9.4 功能特性

三大功能模块

9.4.1 搜索

搜索作为I+三大独立模块之一，可以作为研判人员查找对象的工具，该对象包含手机，身份证等不同对象实体，方便研判人员快速查看对象信息。同时搜索出的对象也可以作为单独对象节点添加到关系分析和GIS分析中去，可以作为研判起点。

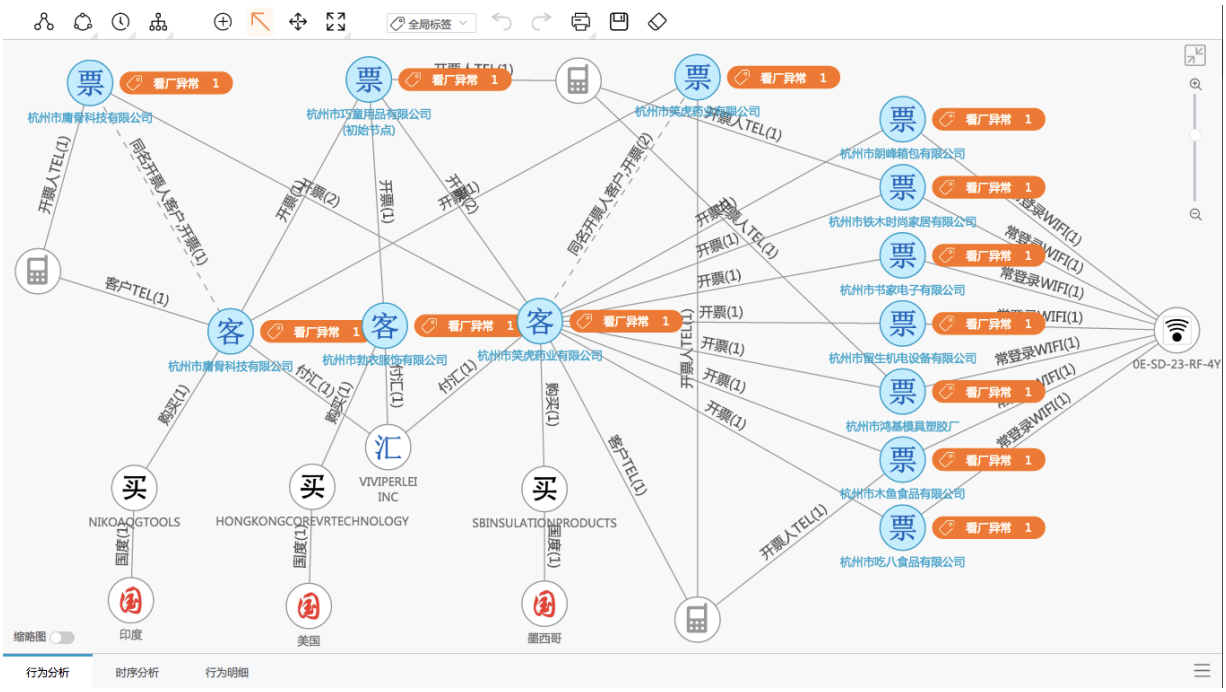
图 9-4: 搜索

9.4.2 关系网络

I+关系网络分析平台提供了多种关系网络分析方式，可让您方便快捷的从复杂的关系网络中挖掘出有用的情报。I+关系网络分析的主要功能包括：关联反查、群体分析、共同邻居、骨干分析、血缘分析、信息立方、群体统计、标签统计等。

下图是一个通过I+关系网络分析平台形成的客户与票据之间的关系网络，从图中可以清晰看出客户和票据的关系。

图 9-5: 关系网络



关联反查

以任意单个或一批对象为起点进行关系的无限拓展分析。关联反查，可帮助实现信息的无限关联。情报分析工作的核心是从大量的、没有关联的信息中发现少量的关联性线索和情报，即将信息转换为可操作情报的过程。关联反查提供两种方式：简单和高级。

群体分析

分析一批相同或不同类型的对象内部之间的关联关系，包含直接关系和间接关系。

共同邻居

分析一批相同或不同类型的对象共同联系对象。

路径分析

分析两个对象之间的关系路径。

骨干分析

针对团伙网络，通过智能业务算法，探索关系网络中核心骨干节点。

血缘分析

以家族户号为血缘脉络，展示所有人之间的血缘关联。

时序分析

在时间维度上详细展示每个事件发生细节。

信息立方

- 行为分析

  展示事件在时间维度上发生的分布频率情况。

- 行为明细

  将事件的明细信息（原数据记录按规则筛选）展示出来。

- 对象信息

  汇总关系网络中的实体，并按实体类型分类。

- 统计信息

  统计关系网络中的关系和实体，包含总体分布、对象熟悉和关系属性。

群体统计

群集统计是统计关系网络分析中的群集分布。这里面群集是指一群体对象节点，任意两两对象节点在拓扑上拥有联通路径，其中合并节点作为一个联通桥，在拓扑上认为其内部全部联通。

标签统计

标签统计是为了统计关系网络中的对象节点的标签信息。在I+系统中，标签分为两类：系统标签和用户标签。系统标签是指业务系统对某些节点定义的标签，如红黑名单等。而用户标签是指每个I+用户对某些节点通过I+添加的标签。

图区布局

关系网络分析图区支持矩阵布局、圆环布局、横直线布局、竖直线布局、力导向布局、层次布局。

右键操作

关系网络分析图区中的信息包括对象、关系内容，映射的网络结构中的图。其中点和边是核心元素，所有的分析都是基于图对其中的点和边做操作。右键操作是围绕着对关系网络编辑分析设计的功能点。

目录管理

目录管理可以对目录和分析进行管理操作，其主要功能如下：

- 对于目录，支持新建分析、删除目录、重命名目录和新建目录操作。
- 对于分析，支持打开、删除和重命名操作。
- 支持通过搜索对内容进行检索定位。
- 支持将个人分析内容分享给他人延续分析。

协作共享

协作共享是产品提供的一种新分析模式，可以将个人的分析共享给其他人，将个人的思路和经验传递给其他分析用户，同时可以将其他分析用户的智慧和经验统一起来，形成多人协作，团体共同推进的局面，能够将团队融合为一个整体。

协作共享的角色分为发起者、协同者。整体流程为：发起者对个人的分析内容分享，分享过程中会设定分享范围，指定分享对象。协同者在接收到分析内容后，可以延续分析，同时将结果保存，形成新的分析版本。

同时产品支持对协同分析的管理，包括删除、重命名，历史版本等管理。

9.4.3 地图分析

GIS模块为I+情报分析平台三大主要模块之一。GIS模块的前端界面如下所示：

图 9-6: GIS模块的前端界面一

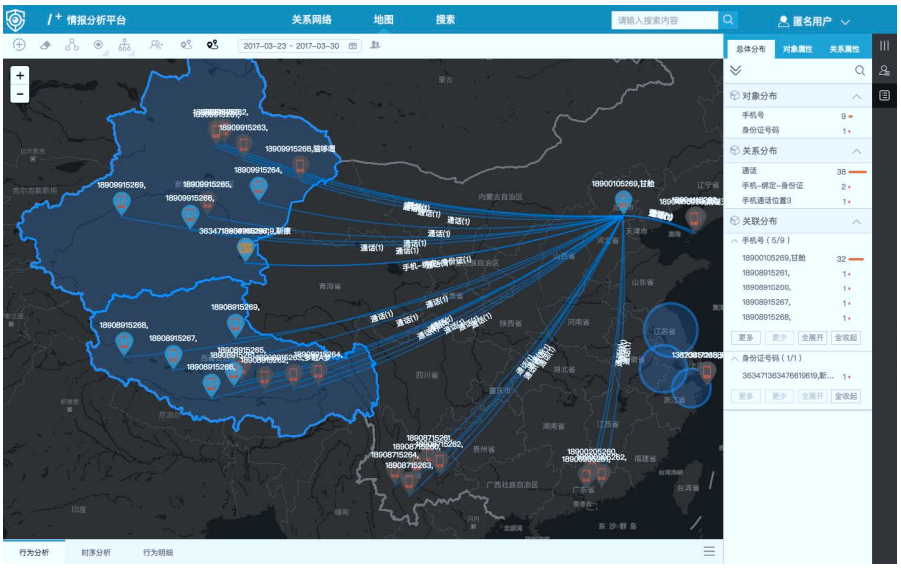
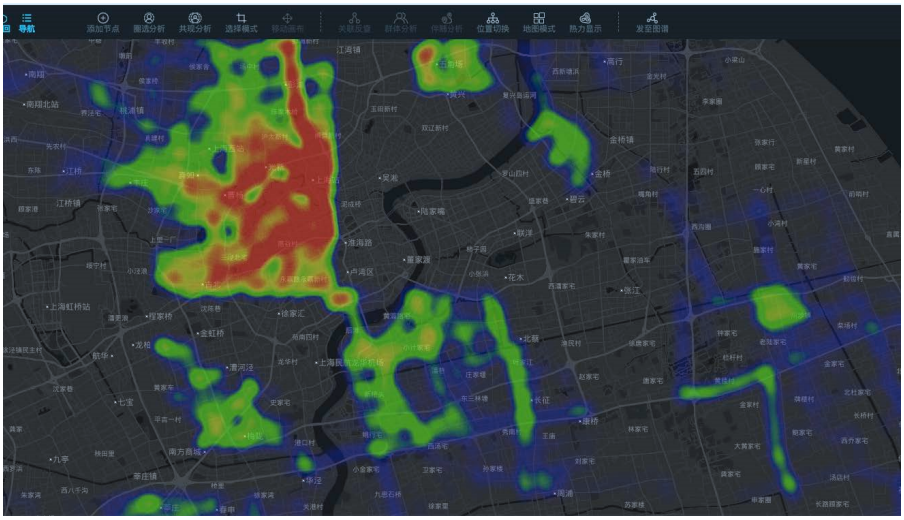


图 9-7: GIS模块的前端界面二



GIS模块的功能如下所示。

空间关系网络

以对象的坐标位置布局关系。辅以多维统计分析和不同层次的关系探索，构建时空网络，支持自定义业务坐标，建立不同的地图模式，方便业务差异化展示。

动态轨迹

自定义不同对象实体的动态轨迹任务，支持点或线轨迹模式切换和视觉可视差异化表现。

圈地选人

支持用户自定义地理位置和空间范围，搜寻时空交错中显现的对象。

伴随分析

构建降纬hash空间，极速计算潜在伴随对象，支持业务时空自定义。

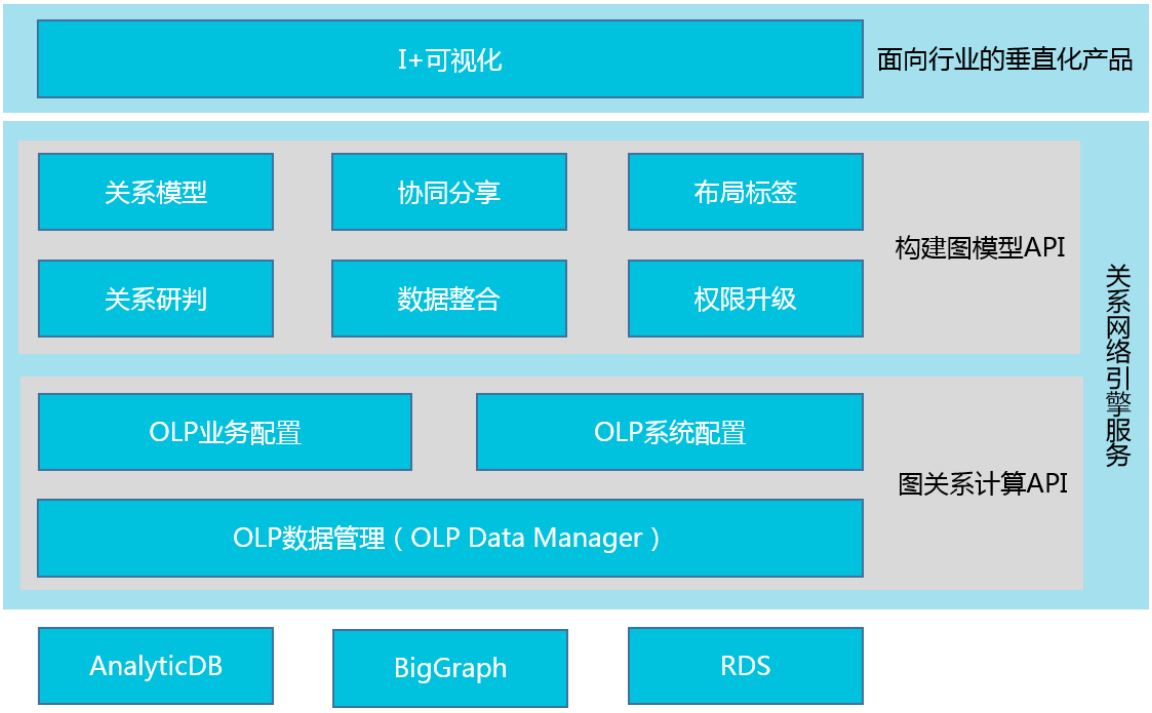
共现分析

以自定义时空为基础，海量数据中探查较高概率共同出现的对象。

9.4.4 关系引擎

关系引擎的逻辑功能结构图如下：

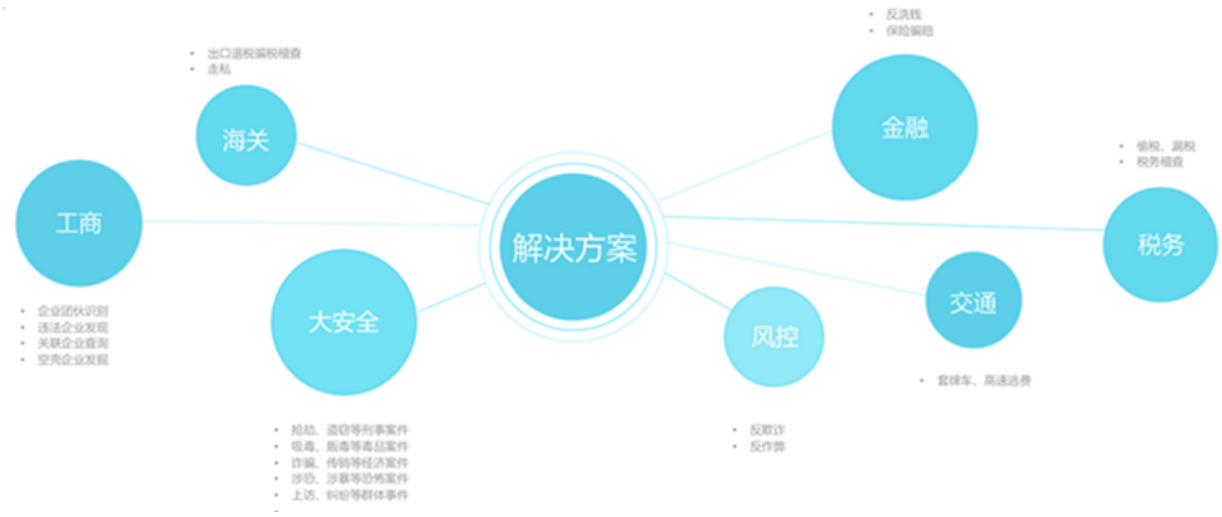
图 9-8: 关系引擎



- 关系网络、搜索、地图模块所有的计算API开放。
- 支持基本的关系点和关系边的基本运算。

9.5 应用场景

图 9-9: 应用场景



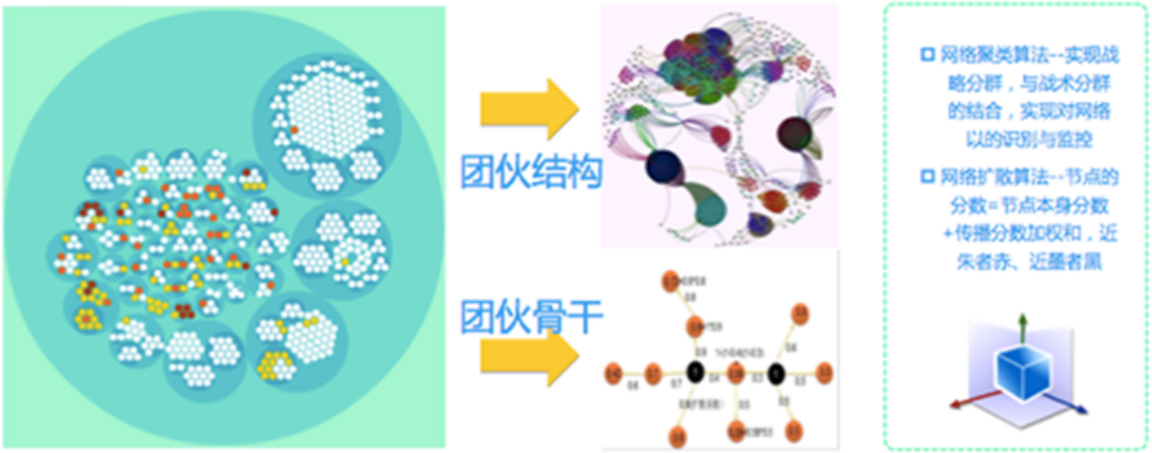
9.5.1 智能关系网络

I+提供智能的关系网络分析，可以方便快捷的帮您分析出各对象之间的关系。以下是团伙关系分析和转账交易分析的案例。

团伙关系分析

I+可通过对象之间的关联关系，分析出对象的团伙结构；通过网络拓扑图结构，分析出关系网络中的骨干成员，如下图所示。

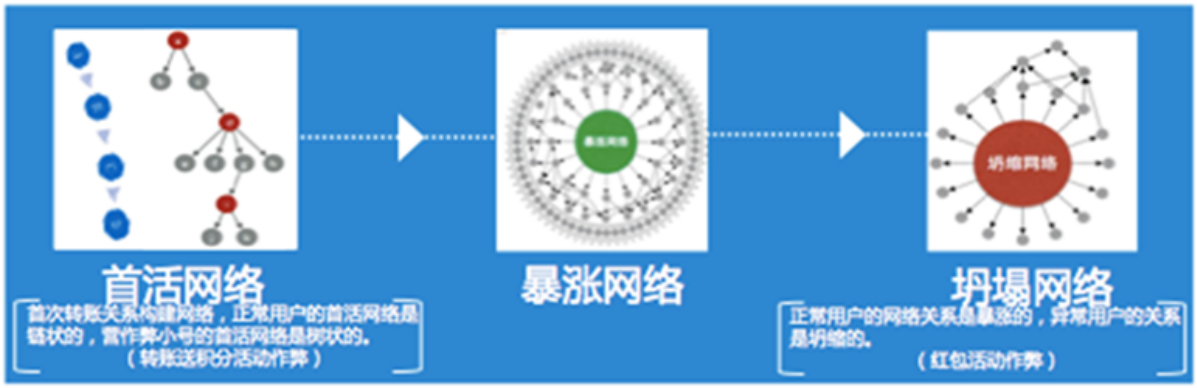
图 9-10: 团伙关系分析



转账交易分析

I+可从用户间的转账交易中分析出可能存在的异常转账（例如：营销作弊等），示例如下图所示。

图 9-11: 转账交易分析



首活网络：首次转账关系构建的网络，正常用户的首活网络是链状的，营销作弊的首活网络是树状的（例如转账送积分活动作弊）。对于营销作弊的转账，您可以选取首次活跃为转账交易的买方和卖方构建树状首活网络，以探索首活网络的网络性质（如网络大小、众数占比、生长速度等）。

暴涨网络和坍塌网络：从网络中资金出度最大的一部分节点出发，顺着资金链路往下探索，正常网络的生长状态是暴涨的，营销作弊网络的生长是坍塌的（最后会汇集到一个账户）。

9.5.2 行业风控

I+的行业风控应用场景如下：

图 9-12: 行业风控图

- 关系模型：建立自然人、账户、设备、环境间的联系，并通过数据挖掘算法识别关系属性（例如：强弱、影响力、类型等），并挖掘关键人物及其子群研究。
- 关系引擎：将关系数据转化成标准化的引擎和接口服务，以使更多的业务场景享受关系网络带来的价值。
- 可视化产品：更友好、直观的展示对象之间的关联，更易于使用。
- 应用方案：通过在风险控制以及关系网络推荐等场景中的业务应用方案，使得关系网络可以不断有业务上的驱动。

9.5.3 公安安防

公安安防客户可以通过I+来构建自己的信息系统，对安全信息进行查询分析，最终基于可视化页面展示，如下图所示。

图 9-13: 公安安防图

9.6 使用限制

无。

9.7 基本概念

关联反查

以任意单个或一批对象为起点进行关系的无限拓展分析。关联反查，可帮助实现信息的无限关联。情报分析工作的核心是从大量的、没有关联的信息中发现少量的关联性线索和情报，即将信息转换为可操作情报的过程。关联反查提供两种方式：简单和高级。

群体分析

分析一批相同或不同类型的对象内部之间的关联关系，包含直接关系和间接关系。

共同邻居

分析一批相同或不同类型的对象共同联系对象。

路径分析

分析两个对象之间的关系路径。

骨干分析

针对团伙网络，通过智能业务算法，探索关系网络中核心骨干节点。

血缘分析

以家族户号为血缘脉络，展示所有人之间的血缘关联。

时序分析

在时间维度上详细展示每个事件发生细节。

信息立方

- 行为分析

  展示事件在时间维度上发生的分布频率情况。

- 行为明细

  将事件的明细信息（原数据记录按规则筛选）展示出来。

- 对象信息

  汇总关系网络中的实体，并按实体类型分类。

- 统计信息

  统计关系网络中的关系和实体，包含总体分布、对象熟悉和关系属性。

群体统计

群集统计是统计关系网络分析中的群集分布。这里面群集是指一群体对象节点，任意两两对象节点在拓扑上拥有联通路径，其中合并节点作为一个联通桥，在拓扑上认为其内部全部联通。

标签统计

标签统计是为了统计关系网络中的对象节点的标签信息。在I+系统中，标签分为两类：系统标签和用户标签。系统标签是指业务系统对某些节点定义的标签，如红黑名单等。而用户标签是指每个I+用户对某些节点通过I+添加的标签。

10 Quick BI

10.1 什么是Quick BI

Quick BI是一个基于云计算的灵活的轻量级的自助BI工具服务平台。

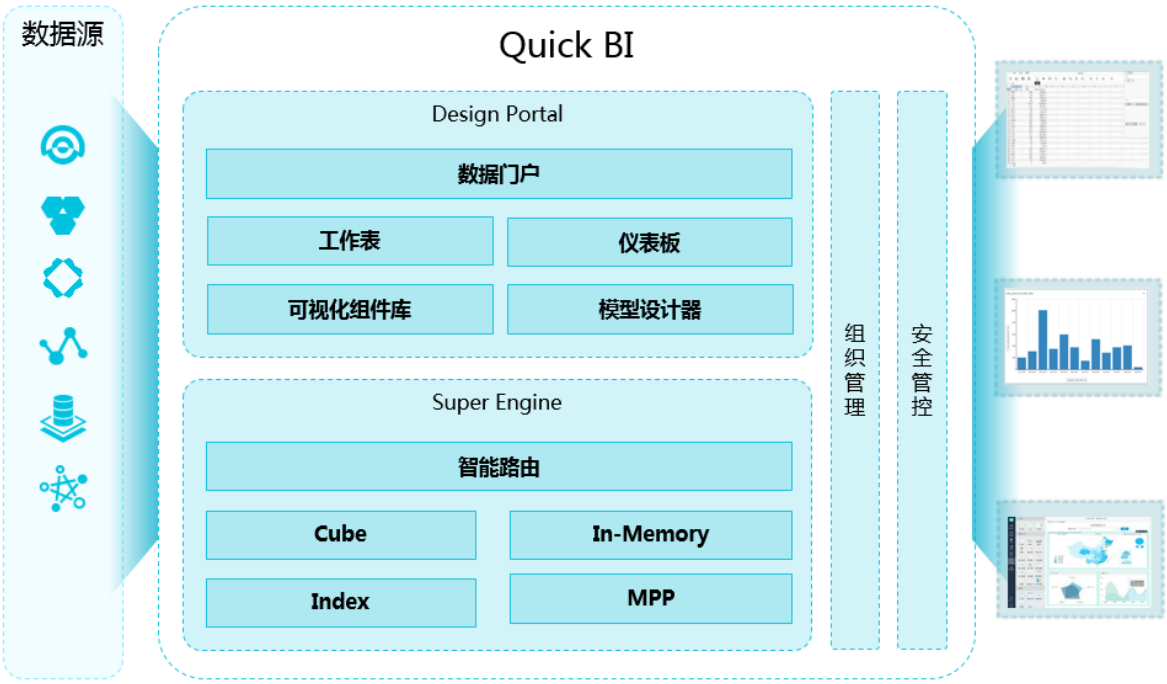
Quick BI支持众多种类的数据源，既可以连接MaxCompute（ODPS）、RDS、AnalyticDB、HybridDB（Greenplum）等云数据源，也支持连接ECS上您自有的MySQL数据库，同时，还支持连接来自VPC的数据源。Quick BI可以为您提供海量数据实时在线分析服务，通过提供智能化的数据建模工具，极大降低了数据的获取成本和使用门槛，通过拖拽式的操作和丰富的可视化图表控件，帮助您轻松自如地完成数据透视分析、自助取数、业务数据探查、报表制作和搭建数据门户等工作。

Quick BI不止是业务人员看数据的工具，更能让每个人都成为数据分析师，帮助企业实现数据化运营。

10.2 产品架构

Quick BI产品架构如下图所示。

图 10-1: Quick BI架构



Quick BI的主要模块和相关功能。

- **数据连接模块**

  负责适配各种云数据源，包括MaxCompute、RDS（MySQL、PostgreSQL、SQL Server）、AnalyticDB、HybridDB（MySQL、PostgreSQL）等，封装数据源的元数据或者数据的标准查询接口。

- **数据预处理模块**

  负责针对数据源的轻量级 ETL 处理。目前主要是支持MaxCompute的自定义SQL功能，未来会扩展到其他数据源。

- **数据建模**

  负责数据源的OLAP建模过程，将数据源转化为多维分析模型，支持维度（包括日期型维度、地理位置型维度）、度量、星型拓扑模型等标准语义，并支持计算字段功能，允许您使用当前数据源的SQL语法对维度和度量进行二次加工。

- **工作表/电子表格**

  负责在线电子表格（Workbook）的相关操作功能，涵盖行列筛选、普通或高级过滤、分类汇总、自动求和、条件格式等数据分析功能，并支持数据导出，以及文本处理、表格处理等功能。

- **仪表板**

  负责将可视化图表控件组装为仪表板。支持线图、饼图、柱状图、漏斗图、树图、气泡地图、色彩地图、指标看板等17种图表，支持查询条件、TAB、IFRAME、PIC和文本框5种基本控件，支持图表间数据联动效果。

- **数据门户**

  负责将仪表板组装为数据门户，支持内嵌链接（仪表板）和外嵌链接（第三方URL），支持模板和菜单栏的基本设置。

- **QUERY引擎**

  负责针对数据源的查询过程。

- **组织权限管理**

  负责组织管理和工作空间管理的两级权限架构体系管控，以及工作空间下的用户角色体系管控，实现基本的权限管理，实现同一份报表，不同的人可以看不同的内容。

- **行级权限管理**

  负责数据的行级粒度权限管控，实现同一张报表，不同的人可以看不同的数据。

- **分享/公开**

支持将工作表/电子表格、仪表板、数据门户分享给其他的用户，支持将仪表板公开到互联网供非登录用户查看。

10.3 功能特性

Quick BI提供以下功能：

无缝集成云上数据库

支持阿里云多种数据源，包括MaxCompute、RDS（MySQL、PostgreSQL、SQL Server）、AnalyticDB、HybridDB（MySQL、PostgreSQL）等。

图表

丰富的数据可视化效果。系统内置柱状图、线图、饼图、雷达图、散点图等多种可视化图表，满足不同场景的数据展现需求，同时自动识别数据特征，智能推荐合适的可视化方案。

分析

多维数据分析。基于Web页面的工作环境，通过拖拽式的操作和类似于Excel的页面展示，实现数据的一键导入和实时分析，并可以灵活地切换数据分析的视角，无需您重新建模。

快速搭建数据门户

通过拖拽式操作、强大的数据建模和丰富的可视化图表，帮助您快速搭建数据门户。

实时

支持海量数据的在线分析。您无需提前进行大量的数据预处理，大大提高数据的分析效率。

数据权限

内置组织成员管理功能，支持行级数据权限，满足不同人看不同的报表，以及同一份报表，不同的人查看不同的内容。

10.4 产品优势

Quick BI的总体优势可以总结为多、快、强大和易用。

多

支持RDS、MaxCompute、AnalyticDB等多种云数据源。

快

亿级数据秒级响应。

强大

内置完整的电子表格工具，可以让您轻松制作复杂的中国式报表。

易用

丰富的数据可视化功能，自动识别数据特征，自动为您推荐合适的图表。

10.5 使用限制

无。

10.6 应用场景

10.6.1 数据及时分析与决策

能够解决：

- 取数难

  业务人员需要经常找技术人员写SQL语句取数，并查看各个维度的数据来做决策。

- 报表产出效率低，维护难

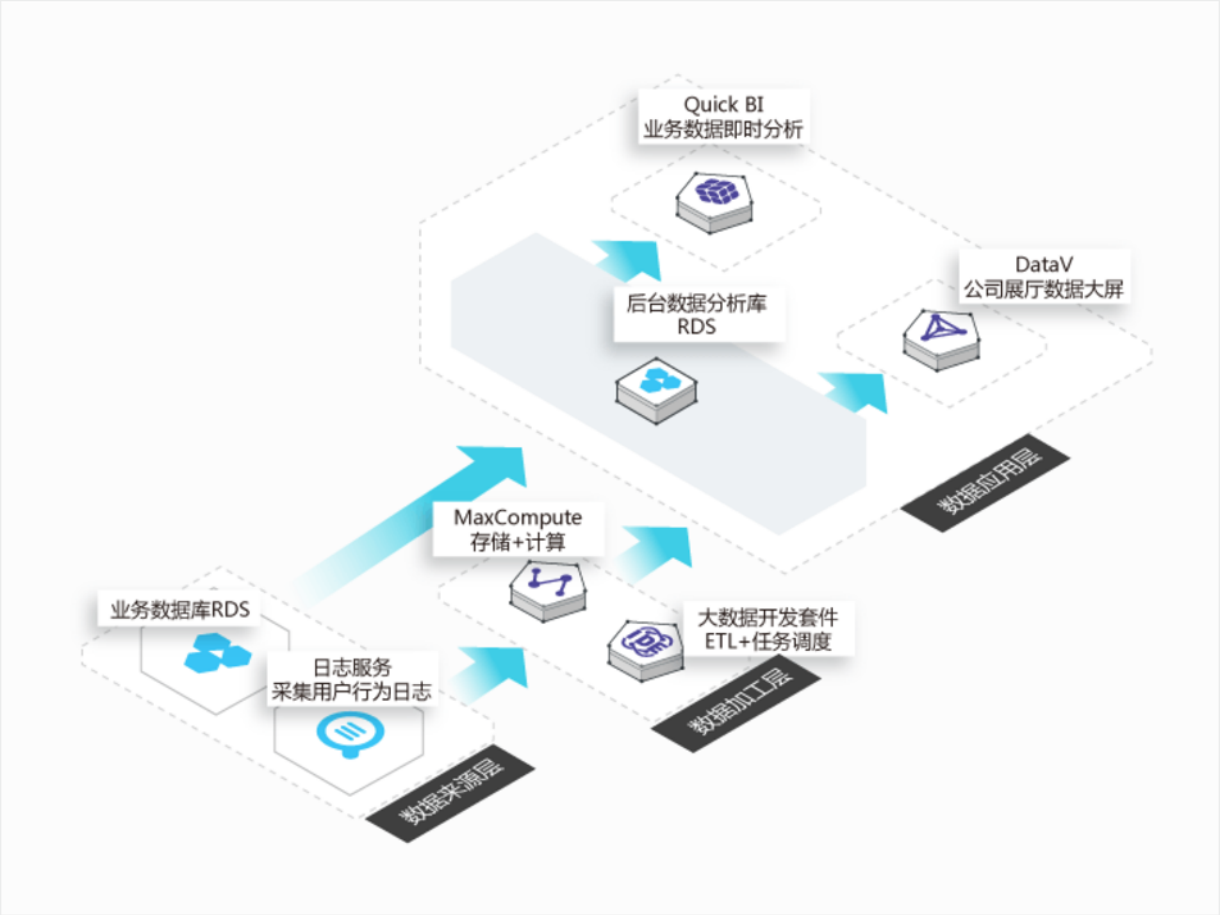
  后台分析系统的数据报表变更，编码研发周期长，维护困难。

- 图表效果设计不佳，人力成本高

  使用 HighChart 等工具做报表，界面效果不佳，人力维护成本高。

推荐搭配使用：RDS + Quick BI

图 10-2: 数据及时分析与决策



10.6.2 报表与自有系统集成

能够实现：

- 上手快

  上手简单、快捷，满足不同岗位的数据需求，学习门槛低。

- 极大提高看数据的效率

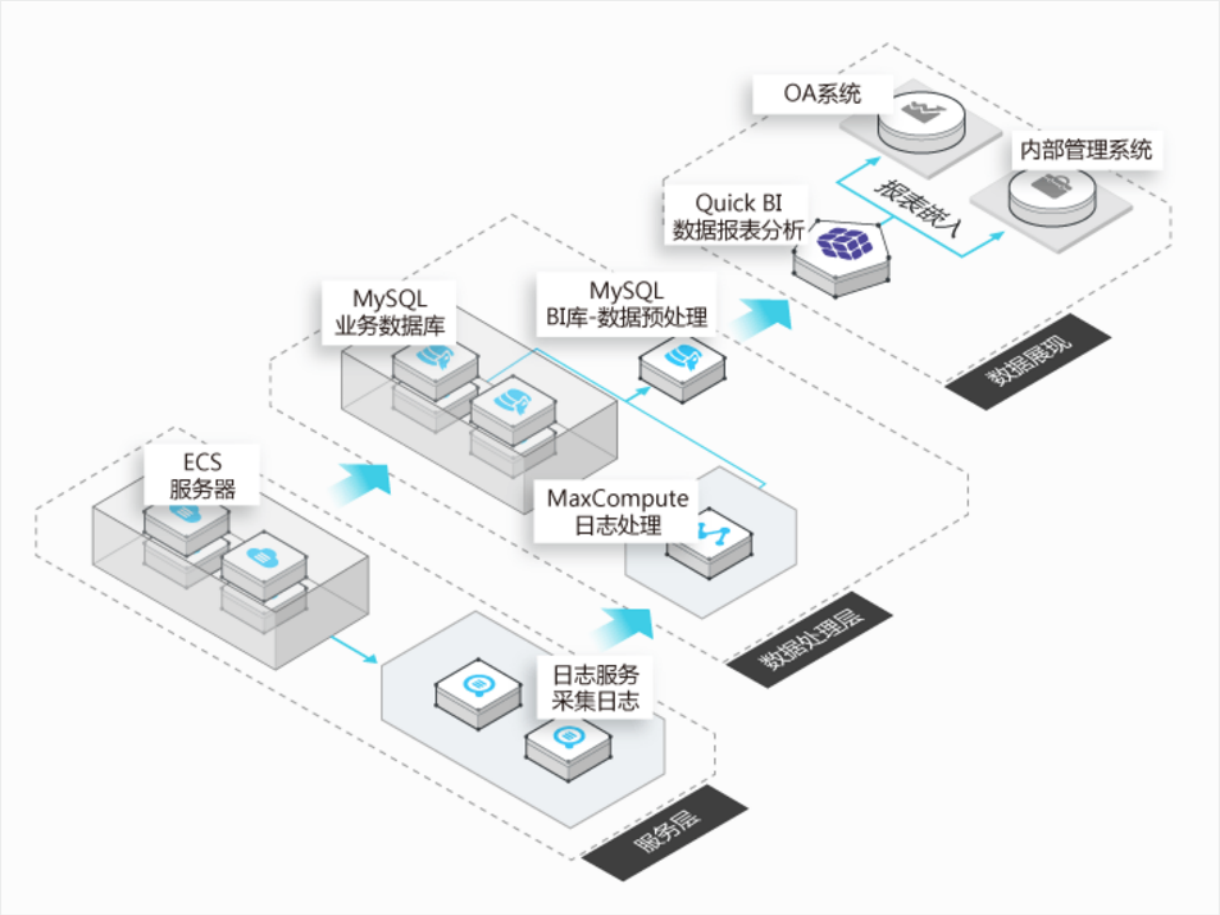
  与内部系统集成，可结合进行数据分析，极大提高看数据的效率。

- 统一系统入口

  解决员工使用多系统的麻烦，利于使用与控制。

推荐搭配使用：RDS + Quick BI

图 10-3: 报表与自有系统集成



10.6.3 交易数据权限管控

能够实现：

- 数据权限行级管控

  轻松实现同一份报表，上海区经理只看到上海的相关数据。

- 适应多变的业务需求

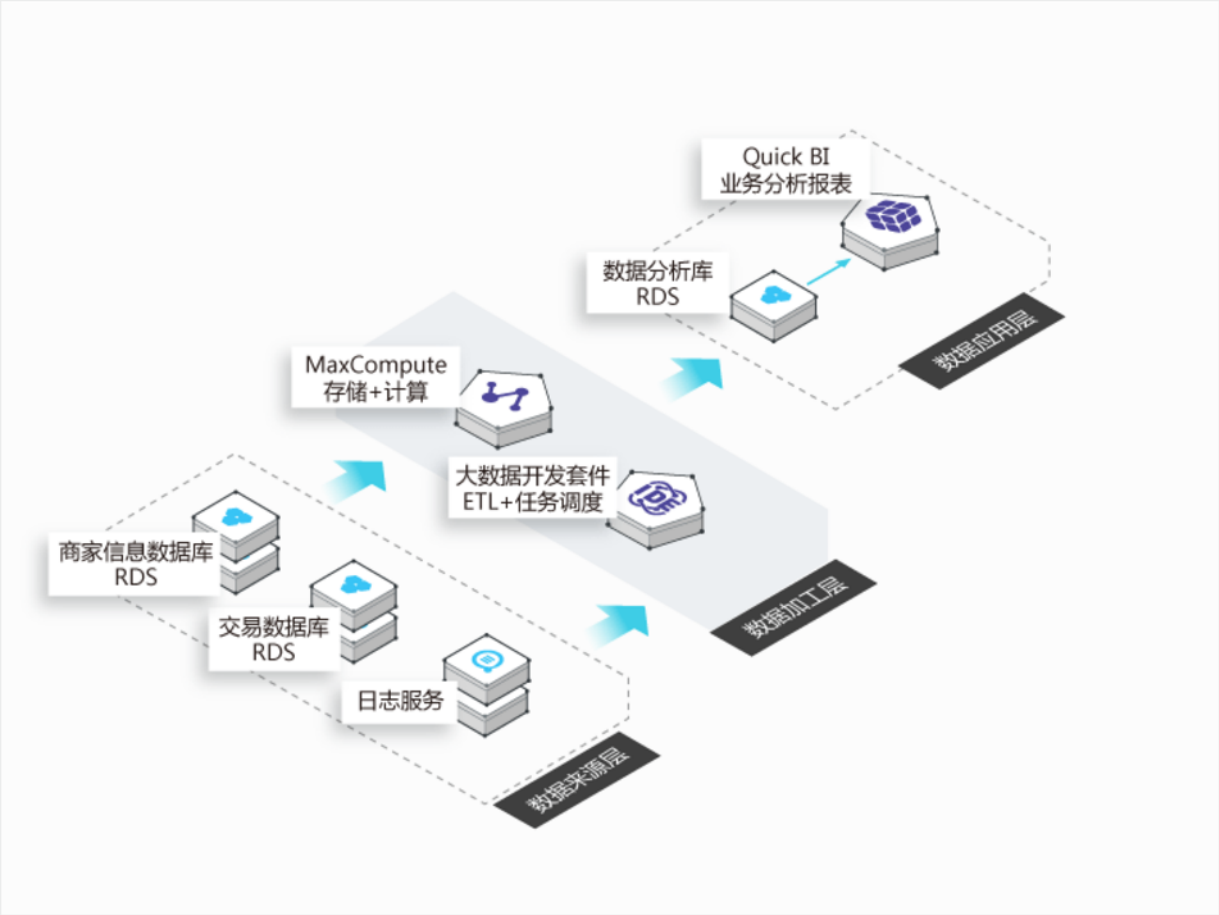
  统计指标经常根据业务发展而频繁变动，负担重，响应慢。

- 跨源数据集成及计算性能保障

  充分利用云上 BI 的底层能力，解决跨源数据分析及计算性能瓶颈问题。

推荐搭配使用：Log + RDS + Quick BI + MaxCompute

图 10-4: 交易数据权限管控



11 流计算StreamCompute

11.1 什么是流计算

阿里云流计算（Alibaba Cloud StreamCompute，以下简称流计算）是一种提供实时流数据计算服务的通用计算平台。

业务背景

目前对信息高时效性、可操作性的需求不断增长，这要求软件系统在更少的时间内能处理更多的数据。传统的大数据处理模型将在线事务处理和离线分析从时序上将两者完全分割开来，但显然该架构目前已经越来越落后于人们对于大数据实时处理的需求。

流计算的产生即来源于对于上述数据加工时效性的严苛需求： **数据的业务价值随着时间的流失而迅速降低**，**因此在数据发生后必须尽快对其进行计算和处理**。而传统的大数据处理模式对于数据加工均遵循传统日清日毕模式，即以小时甚至以天为计算周期对当前数据进行累计并处理，显然这类处理方式无法满足数据实时计算的需求。在诸如实时大数据分析、风控预警、实时预测、金融交易等诸多业务场景领域，批量（或者说离线）处理对于上述对于数据处理时延要求苛刻的应用领域而言，是完全无法胜任其业务需求的。而流计算作为一类针对流数据的实时计算模型，可有效地缩短全链路数据流时延、实时化计算逻辑、平摊计算成本，最终有效满足实时处理大数据的业务需求。

什么是流数据

从广义上说，所有大数据的生成均可以看作是一连串发生的离散事件。这些离散的事件以时间轴为维度进行观看就形成了一条条事件流/数据流。不同于传统的离线数据，流数据是指由数千个数据源**持续生成的数据**，流数据通常也以数据记录的形式发送，但相较于离线数据，流数据普遍的规模较小。流数据产生源头来自于源源不断的事件流，例如客户使用您的移动或 Web 应用程序生成的日志文件、网购数据、游戏内玩家活动、社交网站信息、金融交易大厅或地理空间服务，以及来自数据中心内所连接设备或仪器的遥测数据。

阿里云流计算的特点

通常而言，阿里云流计算（StreamCompute）具备以下三大特点：

- 实时（realtime）且无界（unbounded）的数据流

  流计算面对计算的数据源是实时且流式的，流数据是按照时间发生顺序地被流计算**订阅**和**消费**。且由于数据发生的持续性，数据流将长久且持续地集成进入流计算系统。例如对于网站的访问点

击日志流，只要网站不关闭，其点击日志流将一直不停产生并进入流计算系统。因此，对于流系统而言，数据是实时且不终止（无界）的。

- 持续（continuos）且高效的计算

  流计算是一种**事件触发**的计算模式，触发源就是上述的无界流式数据。一旦有新的流数据进入流计算，流计算立刻发起并进行一次计算任务，因此整个流计算是持续进行的计算过程。

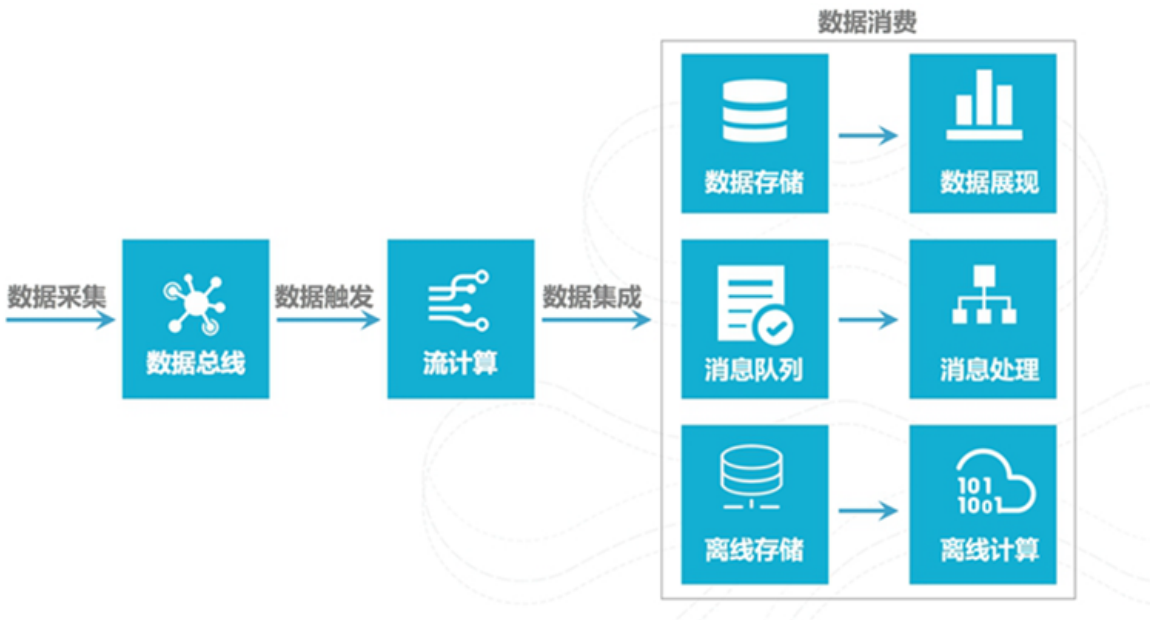
- 流式（streaming）且实时的数据集成

  流数据触发一次流计算的计算结果，可以被直接写入目的数据存储，例如将计算后的报表数据直接写入RDS进行报表展示。因此流数据的计算结果可以类似流式数据源一样持续写入目的数据存储。

11.2 全链路流计算

不同于现有的离线/批量计算模型，阿里云的全链路流计算整体上更加强调数据的实时性，包括**数据实时采集**、**数据实时计算**、**数据实时集成**。三大类数据的实时处理逻辑在全链路上保证了流式计算的低时延。全链路流计算示意图如图 *11-1:* 全链路流计算所示。

图 11-1: 全链路流计算



1. **数据采集**

   使用流式数据采集工具将数据**流式且实时**地采集并传输到大数据消息Pub/Sub系统后，该系统将为下游流计算提供源源不断的事件源去触发流式计算任务。

2. **流式计算**

流数据作为流计算的触发源，驱动流计算的运行。因此，**一个流计算任务必须至少使用一批流数据作为数据源**。一批进入的数据流将直接触发下游流计算的一次流式计算处理，并针对单批次流式数据得出计算结果。

3. **数据集成**

   流计算将计算的结果数据直接写入目的数据存储，其中包括多种数据存储，包括数据存储系统、消息投递系统，甚至直接对接业务规则告警系统发出告警信息。不同于批量计算产品（例如阿里云MaxCompute或者开源Hadoop），**流计算自带数据集成模块，可以将结果数据直接写入到目的数据存储。**

4. **数据消费**

   流计算一旦将结果数据投递到目的数据源后，后续的数据消费从系统划分来说，和流计算已经完全解耦。用户可以使用数据存储系统访问数据，使用消息投递系统进行信息接收，或者直接使用告警系统进行告警。

11.3 流计算和批量计算区别

相对于批量大数据计算，流（式）计算整体上还属于较为新颖的计算概念。本章将从用户/产品层面为您介绍两类计算方式的区别。

> **说明**：
> 本文的说明并非严谨的科学/理论解释，更详细的理论解析请参见Wikipedia的相关内容*Stream Processing*。

11.3.1 批量计算

目前绝大部分传统数据计算和数据分析服务均是基于批量数据处理模型： 使用ETL系统或者OLTP系统构造数据存储，在线的数据服务（包括Ad-Hoc查询、DashBoard等服务）通过构造SQL语言访问上述数据存储并取得分析结果。

这套数据处理的方法论伴随着关系型数据库在工业界的演进而被广泛采用。但在大数据时代下，伴随着越来越多的人类活动被信息化、进而数据化，越来越多的数据处理要求实时化、流式化，当前这类处理模型开始面临实时化的巨大挑战。

传统的批量数据处理通常基于如下处理模型。

1. 使用ETL系统或者OLTP系统构造原始的数据存储，以提供给后续的数据服务进行数据分析和数据计算。如图 *11-2:* 批量计算所示，用户装载数据，系统将根据自己的存储和计算情况，对于装

载的数据进行索引构建等一系列查询优化工作。因此，对于批量计算，**数据一定需要预先加载到计算系统**，**后续计算系统才在数据加载完成后方能进行计算。**

图 11-2: 批量计算

2. 用户/系统**主动**发起一个计算任务（例如MaxCompute的SQL任务，或者Hive的SQL任务）并向上述数据系统进行请求。此时计算系统开始调度（启动）计算节点进行大量数据计算，该过程的计算量可能巨大，耗时长达数分钟乃至于数小时。同时，由于数据累计的不可及时性，上述计算过程的数据一定是历史数据，无法保证数据的新鲜。您可以根据自己的需要随时调整计算SQL，甚至于使用Ad-Hoc查询，可以做到即时修改即时查询。
3. 计算结果返回，计算任务完成后将数据以结果集形式返回给您，或者可能由于计算结果数据量巨大保存在数据计算系统中，您可以进行再次数据集成到其他系统。一旦数据结果巨大，整体的数据集成过程较长，耗时可能长达数分钟乃至于数小时。

批量计算是一种批量、高时延、主动发起的计算任务。 使用批量计算的顺序如下所示。

1. 预先加载数据。
2. 提交计算任务，并且可以根据业务需要修改计算任务，再次提交任务。
3. 计算结果返回。

11.3.2 流式计算

不同于批量计算模型，流式计算更加强调计算数据流和低时延，流式计算数据处理模型如下所示。

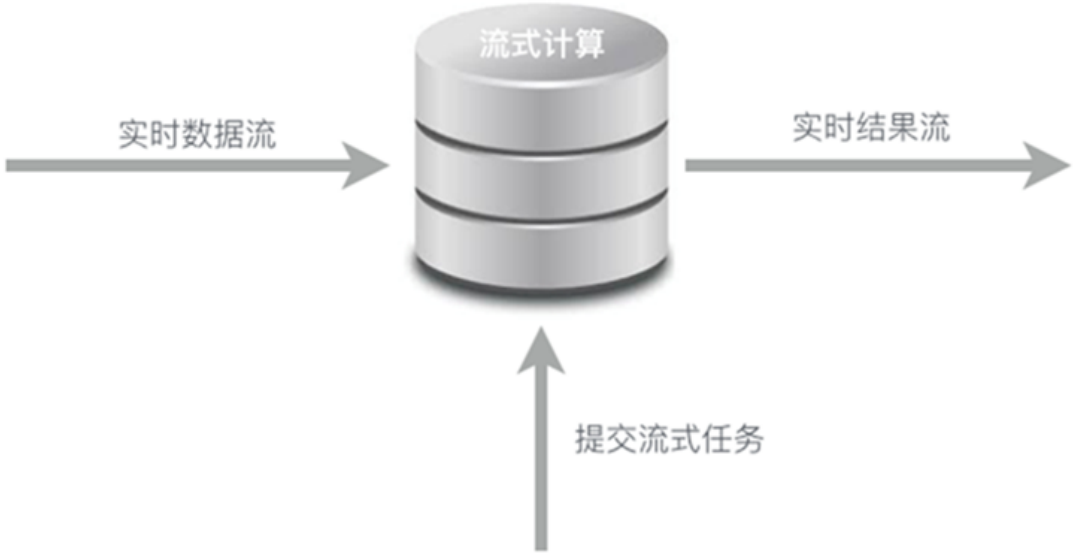
1. 使用实时数据集成工具，将数据实时变化传输到流式数据存储（即消息队列，如DataHub）。此时数据的传输变成实时化，将长时间**累积**大量的数据平摊到每个时间点不停地小批量实时传输，因此数据集成的时延得以保证。

此时数据将源源不断写入流数据存储，不需要预先加载的过程。同时流计算对于流式数据不提供存储服务，数据是持续流动，在计算完成后就立刻丢弃。

2. 数据计算环节在流式和批量处理模型差距更大，由于数据集成从**累积**变为**实时**，不同于批量计算**等待数据集成全部就绪后才启动计算任务**，流式计算任务是一种常驻计算**服务**，一旦启动将一直处于**等待事件触发**的状态，一旦有小批量数据进入流式数据存储，阿里云流计算立刻计算并迅速输出结果。同时，阿里云流计算还使用了增量计算模型，将大批量数据分批进行增量计算，进一步减少单次运算规模并有效降低整体运算时延。

   从用户角度，对于流式任务，必须预先定义计算逻辑，并提交到流计算系统中。在整个运行期间，流计算作业逻辑不可更改。当您停止当前作业运行后再次提交作业时，之前已经计算完成的数据无法重新再次计算。
3. 不同于批量计算结果数据需等待整体数据计算结果完成后，批量将数据传输到在线系统。流式计算任务在每次小批量数据计算后可以立刻将数据写入在线/批量系统，进而做到实时计算结果的实时化展现。

图 11-3: 流式计算



流计算是一种持续、低时延、事件触发的计算任务，使用流计算的顺序如下所示。

1. 提交流计算任务。
2. 等待流式数据触发流计算任务。
3. 计算结果持续不断对外写出。

11.3.3 模型对比

流计算与批量计算两类计算模型的差别，如表 *11-1:* 模型对比表所示。

表 11-1: 模型对比表

对比指标	批量计算	流式计算
数据集成方式	预先加载数据。	实时加载数据实时计算。
使用方式	业务逻辑可以修改，数据可重新计算。	业务逻辑一旦修改，之前的数据不可重新计算（流数据易逝性）。
数据范围	对数据集中的所有或大部分数据进行查询或处理。	对滚动时间窗口内的数据或仅对最近的数据记录进行查询或处理。
数据大小	大批量数据。	单条记录或包含几条记录的微批量数据。
性能	几分钟至几小时的延迟。	只需大约几秒或几毫秒的延迟。
分析	复杂分析。	简单的响应函数、聚合和滚动指标。

由于当前流计算的整个计算模型较为简单，在大部分大数据处理场景下，流计算是批量计算的**有效增强**，特别在对于事件流处理时效性上，**流计算对于大数据计算是一个不可或缺的增值服务。**

11.4 产品优势

相较于其他流计算产品，阿里云流计算提供一些极具竞争力的产品优势，您可以充分利用这些优势，方便快捷地解决自身业务实时化大数据分析的问题。

强大的实时处理能力

不同于其他开源流计算中间件只提供粗陋的计算框架，大量的流计算细节需要业务人员重新实现。阿里云流计算集成诸多全链路功能，方便您进行全链路流计算开发，如下所示：

- 强大的流计算引擎。
  - 阿里云流计算提供提供标准的StreamSQL，支持各类失败场景的自动恢复，保证故障情况下数据处理的准确性。
  - 支持多种内建的字符串、时间、统计等类型函数。
  - 精确的计算资源控制，彻底保证您的作业的隔离性。
- 关键性能指标超越开源Flink的3到4倍，数据计算延迟优化到秒级乃至毫秒级，单个作业吞吐量可做到百万级别，单集群规模在数千台。

- 深度整合DataHub数据存储，无需额外的数据集成工作，阿里云流计算可以直接读写上述产品数据。

托管的实时计算服务

不同于开源或者自建的流式处理服务，阿里云流计算是完全托管的流式计算引擎，阿里云可针对流数据运行查询，无需预置或管理任何基础设施。在阿里云流计算，您可以享受一键启用的流式数据服务能力。阿里云流计算天然集成数据开发、数据运维、监控告警等服务，方便您以最小成本试用和迁移流式计算。

提供完全租户隔离的托管运行服务，从最上层工作空间到最底层执行机器，提供最有效的隔离和全面防护，让您放心使用流计算。

良好的流式开发体验

支持标准SQL（产品名称为BlinkSQL），提供内建的字符串、时间、统计等各类计算函数，替换业界低效且复杂的Flink开发，让更多的BI人员、运营人员通过简单的BlinkSQL可以完成实时化大数据分析和处理，让实时大数据处理普适化、平民化。

提供全流程的流式数据处理方案，针对全链路流计算提供包括数据开发、数据运维、监控告警等不同阶段辅助套件，让数据开发仅需三步，即可完成流式计算任务上线。

低廉的人力和集群成本

大量优化的SQL执行引擎，会产生比手写原生Storm任务更高效且更廉价的计算任务，无论开发成本和运行成本，阿里云流计算均要远低于开源流式框架。考虑编写一个复杂业务逻辑下Storm任务Java代码行数，针对这个任务的调试、测试、调优、上线工作成本，以及后续长期对于Storm、Zookeeper等开源软件的运维成本。如果使用阿里云流计算，上述问题完全交由阿里云平台承担，您可完全聚焦业务，快速实现市场目标。

11.5 产品架构

11.5.1 业务架构

目前流计算定义为一套轻量级提供SQL表达能力的流式数加加工处理引擎。其业务架构如下：

- **数据产生**

  生产数据发生源通常为服务器日志、数据库日志、传感器、第三方数据，这份流式数据将作为流计算的驱动源进入数据集成模块。

- **数据集成**

提供用以进行数据发布和订阅的数据总线进行数据采集工具的集成，可以集成大数据计算的DataHub。

- **数据计算**

  阿里云流计算通过订阅数据集成提供的流式数据，驱动流计算的运行。

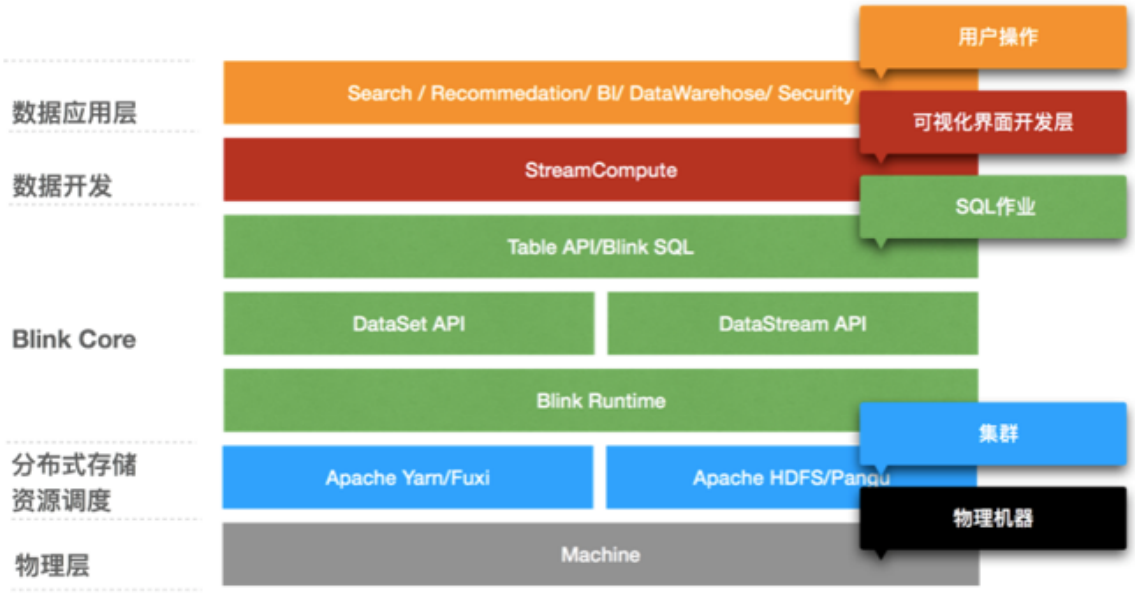
- **数据存储**

  流计算本身不带有任何存储，流计算将流式加工计算的结果写入DataHub数据存储。

11.5.2 技术架构

流计算是一个实时的增量计算平台，其能提供类似SQL的语言，通过MapReduceMerge计算模型（简称MRM）完成增量式计算。流计算具有比较完善的FailOver机制，能保证在各种异常情况下数据的精确性，如技术架构图所示。

图 11-4: 技术架构图



流计算主要由5部分组成。

- **数据应用层**

  主要提供了开发平台，便于新业务的开发和作业提交，系统提供了完善的监控告警系统，在作业出现延迟时及时通知到业务方，同时您可以通过Blink UI等系统了解线上作业的运行情况和性能瓶颈，从而能够及时、更好的优化作业。

- **数据开发**

  该层主要负责Blink SQL的解析和逻辑及物理执行计划的生成，并最终将执行计划转化成可执行的DAG。该层会依据SQL得到的DAG生成由不同Model组成的有向图，用以处理具体的业务逻辑，通常一个Model会包含以下三部分。

  - Map：进行数据过滤、分发（group）或Join（MapJoin）等操作。
  - Reduce：完成一个batch内的聚合计算（流计算将流数据打包成一个个批任务来进行处理，每个批任务内会有多条数据记录）。
  - Merge：将该批任务内的计算结果与以前的结果state进行merge操作得到新的state，在n个批任务处理完成后进行checkpoint操作（n值可配置），从而将该state持久化化到State系统中（如HBase、Tair等）。
- **Blink Core**

  提供多种计算模型、Table API和Blink SQL。下层支持DataStream API和DataSet API。最下层需要有负责资源调度的Blink Runtime来保证作业运行稳定。
- **分布式资源调度**

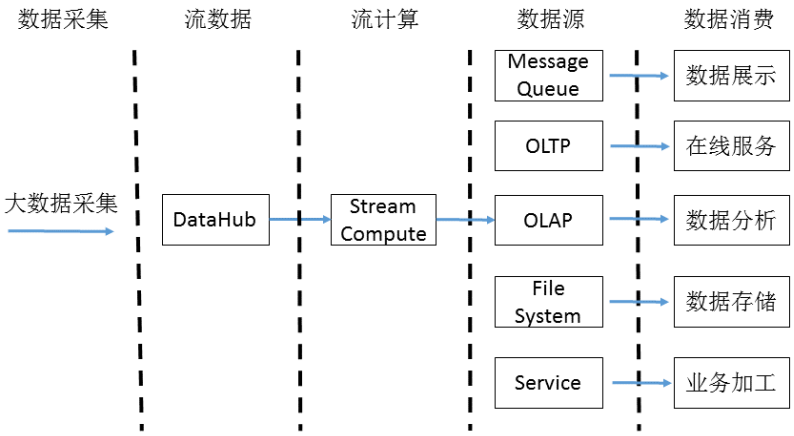
  整个流计算集群是构建在Gallardo调度系统之上，其本身也是流计算能够有效运行和出错恢复的重要保证。
- **物理层**

  物理层是指阿里云给与的强大的硬件机器的集群支持。

11.5.3 业务流程

在阿里云流计算使用前，对流式数据处理全链路有简单认识可以方便您梳理业务流程，制定相应的系统设计方案。阿里云流计算全流程系统架构如图 *11-5: 系统架构*所示。

图 11-5: 系统架构



- **数据采集**

  广义的实时数据采集指您使用流式数据采集工具将数据**流式且实时地采集并传输到大数据Pub/Sub系统**，该系统将为下游流计算提供源源不断的事件源去触发流式计算任务的运行。阿里云大数据生态中提供了诸多针对不同场景领域的流式数据 Pub/Sub 系统，阿里云流计算天然集成上图中诸多Pub/Sub系统，以方便用户可以轻松集成各类流式数据存储系统。由于部分数据存储和流计算模型不能一一匹配，也需要使用其他类型的数据存储系统做中转。DataHub提供了多类数据（包括日志、数据库 BinLog、IoT数据流等）从数据源上传到DataHub的工具、界面，以及和一些开源、商业软件的集成，请参见DataHub相关介绍文档，即可获取丰富多样的数据采集工具。

- **流式计算**

  流数据作为流计算的触发源驱动流计算运行。因此，**一个流计算任务必须至少使用一个流数据作为数据源**。同时，对于一些业务较为复杂的场景，流计算还支持和静态数据存储进行关联查询。例如针对每条DataHub流式数据，流计算将根据流式数据的主键和RDS中数据进行关联查询（即join查询）。同时，阿里云流计算还支持针对多条数据流进行关联操作，StreamSQL支持阿里集团量级的复杂业务也不在话下。

- **实时数据集成**

为尽可能减少数据处理时延，同时减少数据链路复杂度，阿里云流计算将计算的结果数据可不经其他过程直接写入目的数据源，从而最大程度降低全链路数据时延，保证数据加工的新鲜度。为了打通阿里云生态，阿里云流计算天然集成了DataHub数据存储。

- **数据消费**

  流式计算的结果数据进入各类数据源后，您可以使用各类个性化的应用消费结果数据。

  - 可以使用数据存储系统访问数据。
  - 可以使用信息投递系统进行信息接收。
  - 可以使用告警系统进行告警。

11.6 功能特性

阿里云流计算具有以下功能：

- **数据采集和存储**

  常言道“巧妇难为无米之炊”，所有的大数据分析系统都基于一个前提：数据需要采集进入大数据系统。为最大化利用用户现有的流式存储系统，阿里云流计算对接了DataHub流式存储，让用户可以不用进行数据采集、数据集成，即可享受现有的数据流式存储。

  为方便用户管理数据存储，通过提前注册数据存储，用户能够享受到更多一站式流计算开发平台提供的便利性。

- **数据开发**

  - 提供全托管的在线开发平台，集成多种SQL辅助功能，包括BlinkSQL语法检查、BlinkSQL智能提示和BlinkSQL语法高亮。

    - **BlinkSQL语法检查**

      用户在修改IDE文本后即可进行自动保存，保存操作可以触发SQL语法检查功能。语法校验出错误后，将在IDE界面提示出错行数、列数以及错误原因。

    - **BlinkSQL智能提示**

      用户在输入BlinkSQL过程中，IDE提供包括关键字、内置函数、表/字段智能记忆等提示功能。

    - **BlinkSQL语法高亮**

      针对BlinkSQL关键字，提供不同颜色的语法高亮功能，以区分BlinkSQL不同结构。

  - 支持SQL版本管理。

数据开发涵盖了日常开发工作的关键领域，包括代码辅助、代码版本。数据开发模块提供了一个代码版本管理功能。用户每次提交即可生成一个代码版本，代码版本为追踪修改以及日后回滚所用。

- 提供一整套数据存储管理的便捷工具，用户通过在“开发”注册数据存储，即可享受到多种遍历的数据存储服务，包括数据预览、DDL辅助生成。
  - 数据预览

    数据开发页面中，为各类数据存储类型提供数据预览功能。使用数据预览可以有效辅助用户洞察上下游数据特征，识别关键业务逻辑，快速完成业务开发工作。
  - DDL辅助生成

    流计算DDL生成工作大部分均属于比较机械的翻译工作，即将需要映射的数据存储DDL语句人工翻译为流计算的DDL语句。流计算提供辅助生成DDL工作，进一步减少用户手工编写流式作业的复杂度，有效降低人工编写SQL的错误率，并最终提供流计算业务产出效率。
- 支持使用标准SQL进行实时数据清洗、统计汇总、数据分析 ，支持通用的聚合函数，支持流数据和静态数据关联查询。
- 提供了一套模拟的运行环境，用户可以在调试环境中自定义上传数据，模拟运行，检查输出结果。

- **数据运维**

  阿里云流计算提供以下运维监控功能：作业状态、数据曲线、FailOver、CheckPoints、JobManager、TaskExecutor、血缘关系和属性参数。
- **性能调优**
  - 自动性能调优

    该功能帮助用户解决作业吞吐量不足、作业全链路的反压等性能调优的问题。
  - 手动性能调优

    手动性能调优的内容主要有3种类型：
    - 资源调优：主要对作业中的Operator的并发数（parallelism）、CPU（core）、堆内存（heap_memory）等参数进行调优；
    - 作业参数调优：主要对作业中的miniBatch等参数进行调优；
    - 上下游参数调优：主要对作业中的上下游存储参数进行调优。

- **监控报警**

  为了更好的服务用户，帮助用户实时监控Job的健康度，流计算对接了云监控平台。云监控服务可用于收集获取云资源的监控指标或用户自定义的监控指标，探测服务可用性，以及针对指标设置警报，使您全面了解云上的资源使用情况、业务的运行状况和健康度，并及时收到异常报警做出反应，保证应用程序顺畅运行。流计算现支持以下四种类型报警：

  - 业务延时
  - 读入RPS
  - 写出RPS
  - FailoverRate

11.7 产品定位

阿里云流计算提供类标准的StreamSQL语义协助您简单轻松完成流式计算逻辑的处理。同时，受限于SQL代码功能有限无法满足某些特定场景的业务需求，阿里云流计算同时为部分授信用户提供全功能的UDF函数，帮助您完成业务定制化的数据处理逻辑。如果您在流数据分析领域，可直接使用StreamSQL+UDF完成大部分流式数据分析处理逻辑，目前的流计算**更擅长于做流式数据分析、统计、处理**，对于非SQL能够解决的领域，例如复杂的迭代数据处理、复杂的规则引擎告警则不适合现有的流计算产品去解决。

目前流计算擅长解决的几个领域的应用场景，如下所示：

- 实时的网络点击PV、UV统计。
- 统计交通卡口的平均5分钟通过车流量。
- 水利大坝的压力数据统计和展现。
- 网络支付涉及金融盗窃固定行为规则的告警。

阿里云流计算曾经对接过，但发现无法满足的情况，如下所示：

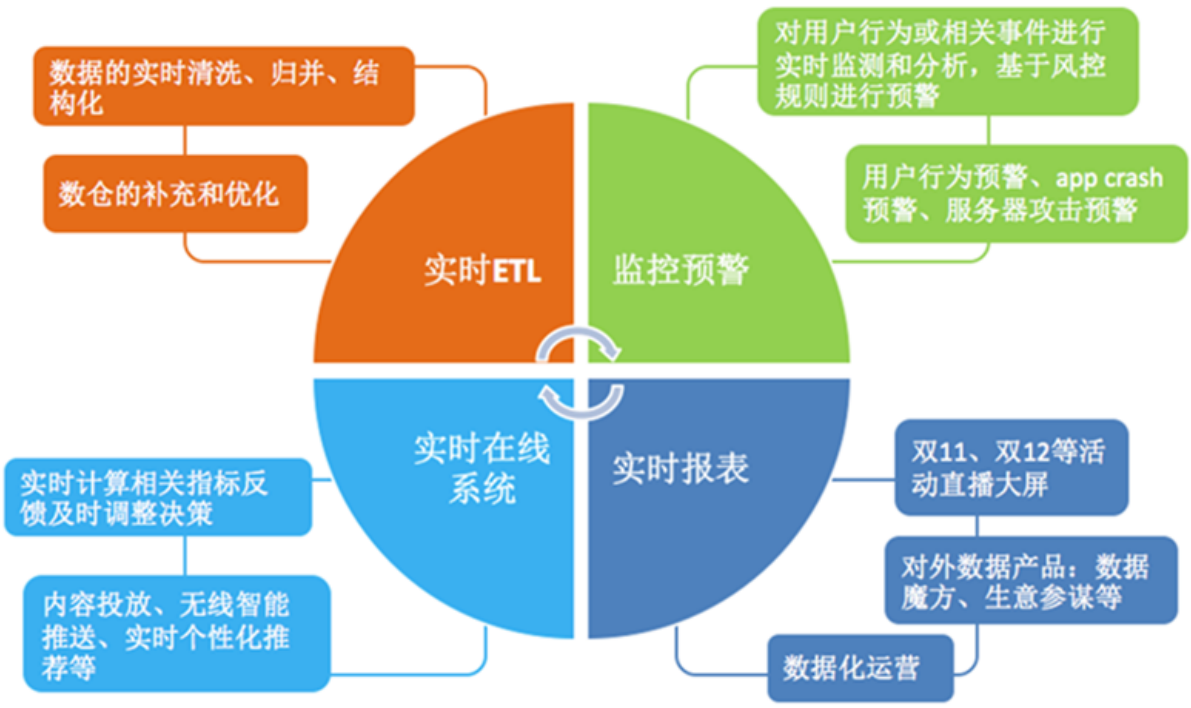
- Oracle存储过程使用阿里云流计算替换：流计算无法从功能上完全替换掉Oracle存储过程，两者面向问题领域不一致。
- 现有的Spark作业无缝迁移到流计算： Spark部分涉及流计算可以考虑改造并迁移到流计算，您可以完全省去运维Spark和开发Spark的各类成本，但无法做到Spark作业无缝迁移到流计算。
- 多种复杂规则引擎告警：针对单一一条数据存在多条复杂规则告警，且该规则在系统运行时变化。这类应该有专门的规则引擎系统解决，当前流计算面对不是该问题域。

当前流计算对外接口定义为StreamSQL/UDF，提供服务于流式数据分析、统计、处理的一站式开发工具，面向的用户人群更多是数仓开发人员、数据分析师，这类用户不希望更多参与底层代码开发，而希望**简单编写流计算SQL即可完成自身流式数据分析业务。**

11.8 应用场景

流计算使用StreamSQL主打流式数据分析场景，如图 *11-6:* 使用场景所示。

图 11-6: 使用场景



- **实时ETL**

  集成流计算现有的诸多数据通道和SQL灵活的加工能力，对流式数据进行实时清洗、归并、结构化。作为离线数仓有效的补充和优化，作为数据实时传输的可计算通道。

- **实时报表**

  实时化采集、加工流式数据源，实时监控和展现业务、客户各类指标，让数据化运营实时化。

- **监控预警**

  对系统和用户行为进行实时检测和分析，实时监测和发现危险行为。

- **在线系统**

实时计算各类数据指标，并利用实时结果及时调整在线系统相关策略。在各类内容投放、无线智能推送领域有大量场景。

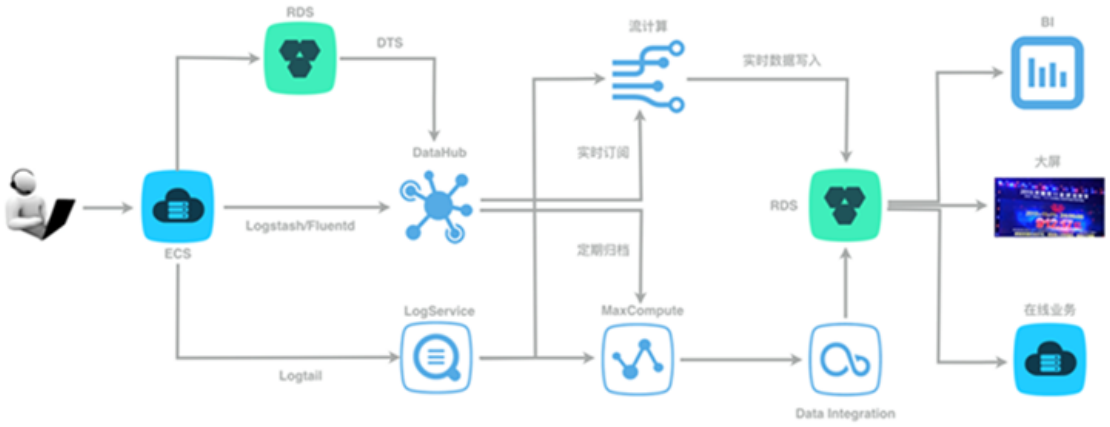
11.8.1 电商活动运营

阿里云流计算脱胎于阿里集团电商行业大数据架构，可以说天生适合电商行业各类流式数据分析和报表支持。从电商行业来看，对于流式数据实时处理需求主要集中在以下几个方面。

- 对于用户行为的实时分析，例如交易大屏、用户大屏等一系列用户行为大屏展示。使用传统的离线分析不仅整体速度慢时延高，而且可能对于在线系统存在一定压力，从而系统稳定性得不到完全保证。
- 对于用户、业务、系统的实时监控，例如全站交易时段曲线图可以协助运营或者技术人员了解当前时间点全站交易情况，如果交易出现不正常波动（例如突然下跌），应该立刻触发报警机制以方便相关人员介入排查问题，以减少交易波动对于公司业务的巨大影响。
- 对于一些大促活动的实时监控，例如双十一、618等电商大促，运营人员需要实时获取到各类指标信息，用以迅速决定是否需要更换大促运营方案。

使用阿里云流计算，结合阿里云各类计算、存储系统，可以方便支持上述各类个性化的流式数据分析需求。不同于其他数据分析系统，阿里云流计算既满足业务灵活性，同时又采用SQL保证业务开发的低门槛。

图 11-7: 电商活动运营架构图



11.9 使用限制

无。

11.10 基本概念

项目空间（Project）

项目空间是阿里云流计算最基本的业务组织单元，是用户管理集群、作业、资源、人员的基本单元。管理员可以选择创建项目，同时也可以将用户加入其他Project中。流计算的项目空间通过阿里云RAM主子账号支持多人协作。

作业（Job）

类似一个MaxCompute/Hadoop Job，一个流计算的作业描述了一个完整的流式数据处理业务逻辑，是流式计算的基础业务单元。

流计算单元（CU）

在阿里云流计算中，作业的流计算单元为CU。一个CU描述了一个流计算作业最小运行能力，即在限定的CPU、内存、IO情况下对于事件流处理的能力。一个流计算作业可以指定在1个或者多个CU上运行。

当前流计算定义 `1 CU = 1 Core CPU + 4G MEM`。

StreamSQL

不同于诸多开源的流式数据处理系统提供非常底层的编程API，阿里云流计算提供更加高层更加面向业务化的StreamSQL（标准SQL语法上提供了关于流式处理的语法扩展），方便数据开发人员使用标准化的SQL即可完成流式数据计算加工的业务流程。因此，阿里云流计算适合面向更大众的数据分析人员快速、方便地完成一个流式数据处理业务。

UDF

类似于Hive UDF函数，StreamSQL提供了标准化的流式数据处理能力同时，对于部分业务特殊自定义处理逻辑，建议您使用UDF函数表达。目前阿里云流计算仅支持Java的UDF函数扩展。

资源（Resource）

阿里云流计算支持UDF（User Define Function，即用户自定义函数），当前UDF函数仅支持使用Java语言表达，对于每个用户上传的Jar，流计算定义为一个Resource。

数据采集（Data Collection）

广义的数据采集指将数据从数据源产生方收集并传输进入到大数据处理引擎的过程，在阿里云流计算，数据采集原则上遵循上述定义，但更加聚焦为将流式数据从数据源产生方收集并传输进入数据总线的过程。

数据存储（Data Store）

阿里云流计算定义为一种轻量级计算引擎，**本身不带有任何业务数据存储系统。**阿里云流计算均是使用外部数据存储作为数据来源和数据目的端进行使用。阿里云流计算将DataHub数据存储定义为外部的数据存储。

数据加工（Data Develop）

流式计算的开发过程（即编写BlinkSQL的过程），将其定义为数据加工。阿里云流计算提供一整套包括开发、调试的在线IDE，服务流式数据加工过程。

数据运维（Data Operation）

流计算作业的在线运维定义为数据运维。阿里云流计算提供一整套管控平台，方便您进行流式数据的运维管控。

12 实时数据分发平台DataHub

12.1 什么是DataHub

阿里云实时数据分发平台（DataHub）是流式数据（Streaming Data）的处理平台，提供对流式数据的发布（Publish）、订阅（Subscribe）及分发功能，让用户可以轻松构建基于流式数据的分析和应用。

DataHub服务可以对各种移动设备、应用软件、网站服务、传感器等产生的大量流式数据进行持续不断的采集、存储和处理。用户可以编写应用程序或者使用流计算引擎来处理写入到DataHub的流式数据（例如：实时web访问日志、应用日志、各种事件等），并且能够产出各种实时的数据处理结果（例如：实时图表、报警信息、实时统计等）。

DataHub服务基于阿里云自研的飞天操作平台，具有高可用、低延迟、高可扩展、高吞吐的特点。DataHub与阿里云流计算引擎StreamCompute无缝连接，用户可以轻松使用SQL进行流数据分析。

DataHub服务也提供分发流式数据到各种云产品的功能，目前支持分发到MaxCompute、OSS等。

12.2 产品优势

高吞吐

单**主题**（**Topic**）最高支持每日TB级别的数据量写入；每个**分片**（**Shard**）最高支持每日8000万Record级别的数据量写入。

实时性

通过DataHub ，用户可以实时的收集通过各种方式生成的数据并进行实时的处理，对用户的业务产生快速的响应。

易用性

- DataHub提供丰富的SDK包，包括C++、Java、Pyhon、Ruby、Go等语言。
- DataHub服务也提供Restful API规范，用户可以使用自己的方式实现接口访问。
- 除了SDK以外，DataHub还提供一些常用的客户端插件，包括：Fluentd、LogStash、Flume等，用户可以使用这些客户端工具向DataHub中写入流式数据。
- DataHub同时支持强Schema的结构化数据和无类型的非结构化数据，用户可以自由选择。

高可用

- 规模自动扩展，不影响对外服务。
- 数据自动多重冗余备份。

动态伸缩

每个**主题**（`Topic`）的数据流吞吐能力可以动态扩展和减少，最高可达到每主题256000 Records/s的吞吐量。

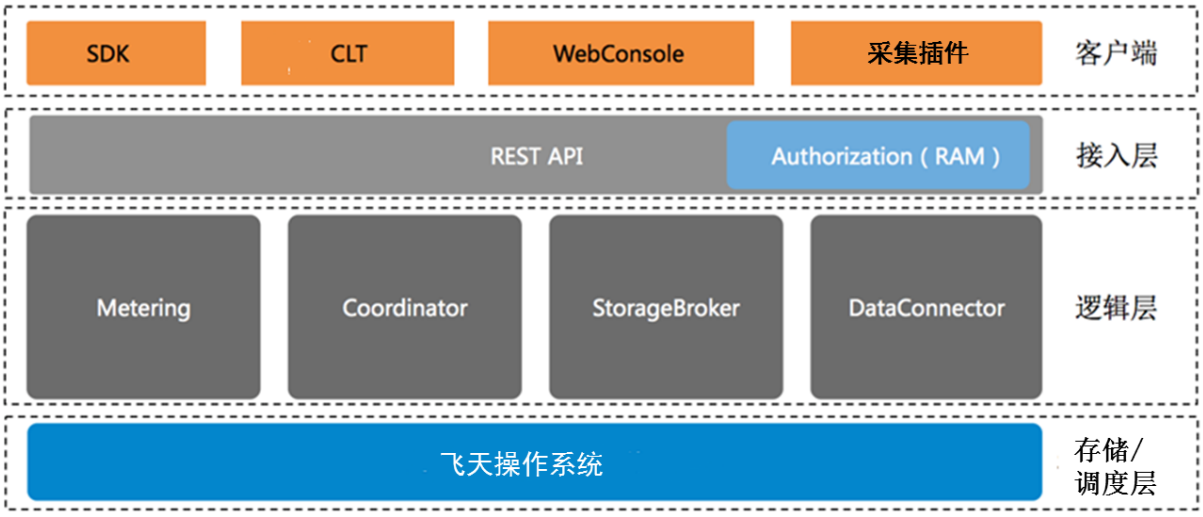
高安全性

- 提供企业级多层次安全防护，多用户资源隔离机制。
- 提供多种鉴权和授权机制及白名单、主子账号功能。

12.3 产品架构

DataHub的产品架构如图 *12-1: DataHub产品架构图*所示。

图 12-1: DataHub产品架构图



DataHub从功能上可分为4层，分别为**客户端**、**接入层**、**逻辑层**及**存储/调度层**。

客户端

DataHub的客户端有以下几种形式：

- SDK：提供C++、JAVA、Pyhon、Ruby、Go等语言的SDK。
- CLT（Command Line Tool）：支持在Windows/Linux/Mac下运行的命令行工具，可以通过命令进行project/topic的管理。

- Web Console：可视化页面，可以在页面上进行project/topic管理、创建订阅、查看shard状态、topic性能监控、管理Connector等操作。
- 采集插件：包括Logstash、Fluntd、Flume以及Oracle GoldenGate。

接入层

DataHub在接入层提供Http（Https）服务和RAM鉴权，支持水平扩展。

逻辑层

DataHub逻辑层处理整个服务的核心逻辑，提供Project/Topic管理、数据读/写、点位存储、流量统计、数据对接等核心功能，根据基本功能可以分为以下几个模块：StorageBroker、Metering、Coordinator、DataConnector。

- StorageBroker：提供最核心的数据读写功能，使用飞天分布式文件系统的LogFile为存储模型，比传输的WAL降低一半集群磁盘读写；一份数据三份存储，单机异常不丢数据；支持跨机房容灾；实时数据写入缓存，保证实时消费场景；历史数据有独立的读缓存，保证读历史数据多份订阅的性能。
- Metering：支持shard维度使用时长计费。
- Coordinator：提供点位存储功能，单机QPS高达150000，支持水平扩展。
- DataConnector：提供数据同步功能，支持将DataHubk的数据自动同步到阿里云产品线的其它服务，目前支持以下服务：MaxCompute、OSS、AnalyticDB、RDS、TableStore、Elasticsearch。

存储/调度层

- 存储：存储层基于飞天分布式文件系统的LogFile做为存储模型，Append模式写入数据，支持SSD混合存储，支持timestamp做为数据索引，单shard写单个飞天分布式文件系统文件，基于时间轴切分存储数据。
- 调度：基于任务调度系统的调度模块，结合partition模式和machine模式实现业务维度的调度算法，即按流量配置每台机器上的负责的shard；不占用任务调度系统的CPU/Memory单元，单机partition数据无上限；能够实现毫秒级Failover，支持热升级。

12.4 功能特性

12.4.1 数据队列

DataHub的基本功能，单shard内数据保序；DataHub对Shard的读/写性能提供SLA的保障；单topic的性能以shard数为单位水平扩展。

12.4.2 点位存储

支持消费应用将消费点位保存到DataHub服务，保证消费应用在Failover后可以从保存的点位进行消费。

12.4.3 数据同步

DataHub中的数据自动同步到阿里云其它服务，支持标done功能，确认某一个时间点之前的数据已经全部同步完成。

流式数据同步DataConnector

DataConnector是将DataHub服务中的流式数据同步到其他云产品中的功能，目前支持将Topic中的数据实时/准实时同步到MaxCompute、OSS、ElasticSearch、RDS for MySQL、AnalyticDB、TableStore中。

用户只需要向DataHub中写入一次数据，并在DataHub服务中配置好同步功能，即可以在各个云产品中使用这份数据。数据同步支持at least once语义，在网络服务异常等小概率场景下可能会导致目的端的数据产生重复。

DataConnector支持的系统

下面是DataConnector支持数据同步的各系统的相关描述。

表 12-1: DataConnector支持的系统

目标系统	时效性	描述
MaxCompute	准实时，通常情况有5分钟延迟	同步Topic中流式数据到离线MaxCompute表，字段类型名称需一一对应，且DataHub中必须包含一列（或多列）MaxCompute表中分区列对应的字段。
OSS	实时	同步数据到对象存储OSS指定Bucket的文件中，将以csv格式保存。
ElasticSearch	实时	同步数据到ElasticSearch指定Index中，Shard之间数据同步不保证时序，所以需将同样ID的数据写入相同的Shard中。
RDS for MySQL	实时	同步数据到指定的RDS for MySQL表中。
AnalyticDB	实时	同步数据到指定的AnalyticDB表中。
TableStore	实时	同步数据到指定的TableStore表中。

12.4.4 扩容缩容Merge/Splits

DataHub支持为Topic动态扩容/缩容，一般通过SplitShard/MergeShard来实现。

Datahub具有服务弹性伸缩功能，用户可根据实时的流量调整Shard数量，来应对突发性的流量增长或达到节约资源的目的。

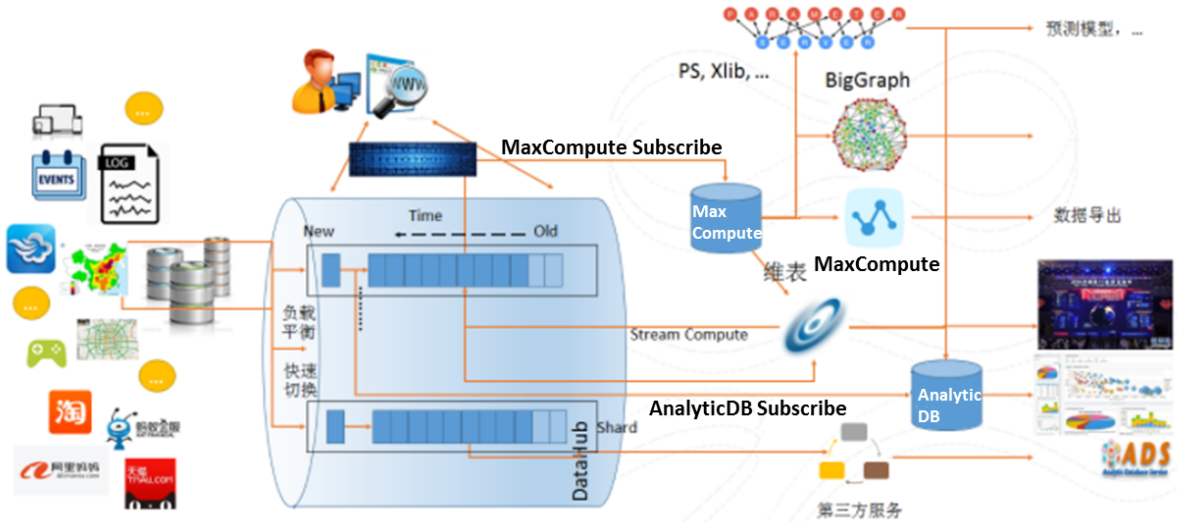
例如：在双11大促期间，大部分Topic数据流量会激增，平时的Shard数量可能完全无法满足这样的流量增长。此时可以对其中一些Shard进行SplitShard扩容操作，最大可扩容至256个Shard，按目前的流控限制足以达到256MB/s的流量，用以应对数据流量的激增。

而在双11大促后，数据流量下降，多余的Shard会占用没有必要的quota，因此此时可以进行MergeShard缩容操作，每两个Shard合并为一个，直至合适为止。

12.5 应用场景

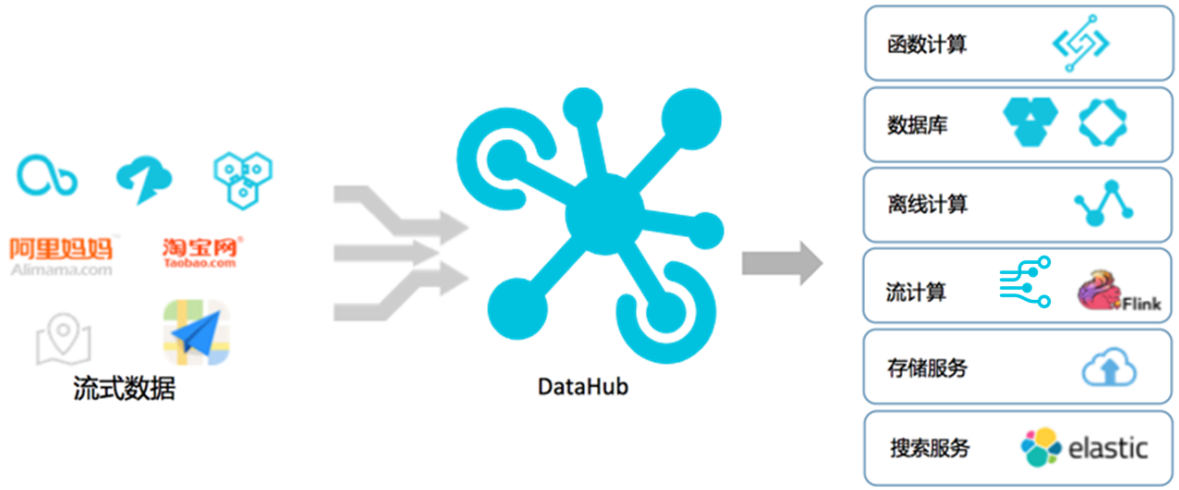
DataHub作为一个流式数据处理服务平台，结合阿里云众多云产品，可以构建一站式的数据处理服务。

图 12-2: DataHub应用



12.5.1 数据上云入口

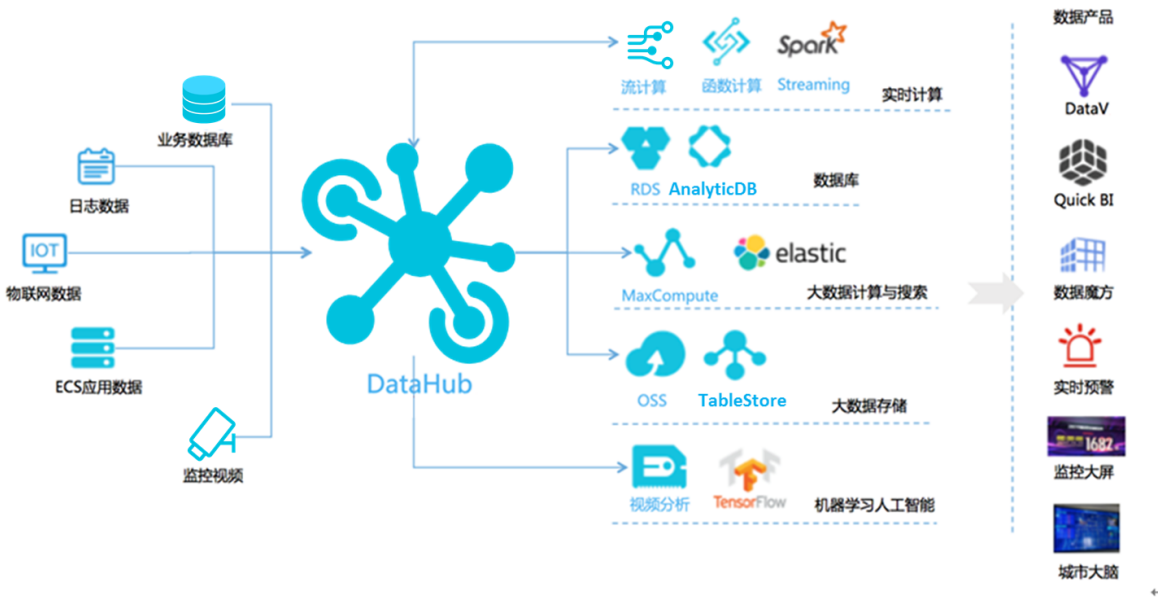
图 12-3: 数据上云入口



DataHub做为数据上云的入口，可以简化用户将数据上传到阿里云的接口，让用户做到一次写入多处使用，无需对接多种云服务的接口。

12.5.2 数据采集通道

图 12-4: 数据采集通道



DataHub提供丰富的采集端插件，支持多种业务数据的采集，让用户可以方便的将现有的业务数据采集到DataHub，在云端进行更多的处理和使用。目前DataHub支持的采集端包括日志采集（

Logstash/Fluntd/Flume）、数据库binlog采集（dts/oracle goldengate）、视频采集（GB28181协议的监控/安防视频）。

12.5.3 流计算StreamCompute

StreamCompute是阿里云提供的流计算引擎，提供使用类SQL的语言来进行流式计算。DataHub和StreamCompute无缝结合，可以作为StreamCompute的数据源和输出源。

图 12-5: DataHub和StreamCompute无缝结合

12.5.4 流处理应用

用户可以编写应用订阅DataHub中的数据，并进行实时的加工，并且将加工后的结果输出。

用户还可以把应用计算产生的结果输出到DataHub中，并使用另外一个应用来处理上一个应用生成的流式数据，来构建数据处理流程的DAG。

12.5.5 流式数据归档

用户的流式数据可以归档到MaxCompute中，通过创建DataHub Connector，指定相关配置，即可创建将Datahub中流式数据定期归档的同步任务。

12.6 使用限制

表 12-2: 使用限制

限制项	限制范围	限制说明
活跃shard数	(0,10]	每个topic中活跃shard数量最大不超过10个。
总shard数	(0,512]	每个topic中总shard数量最大不超过512个。
Http BodySize	最大不超过4MB	Http请求中body大小最大不超过4MB。
单个String长度	最大不超过1MB	数据中单个String字段长度最大不超过1MB。
Merge/Split频率限制	5s	每个新产生的shard在5s内不允许进行Merge/Split操作。
QPS限制	最多不超过1000次	每个Shard写入QPS限制（非Record/s，Batch写入同一Shard仅计算为1次）最大不超过1000次。
Throughput限制	最大不超过1MB	每个Shard写入每秒吞吐最大不超过1MB。
Project限制	最多不超过5个	每个云账号能够创建的Project上限最多不超过5个。
Topic限制	最多不超过20个	每个Project内能创建的Topic上限最多不超过20个，例如有特殊请求请联系管理员。
Topic Lifecycle限制	[1,7]	每个Topic中数据保存的最大时长为7天，最小时长为1天。

12.7 基本概念

Project

项目（Project）是DataHub数据的基本组织单元，下面包含多个Topic。需要注意，DataHub的项目空间与MaxCompute的项目空间是相互独立的，用户在MaxCompute中创建的项目不能复用于DataHub，需要独立创建。

Topic

Topic是DataHub订阅和发布的最小单位，用户可以用Topic来表示一类或者一种流数据。更多详情请参见：*Project及Topic数量限制*。

Topic Lifecycle

表示一个Topic中写入数据在系统中可以保存的最长时间，以天为单位，最小值为1，最大值为7。

Shard

Shard表示对一个Topic进行数据传输的并发通道，每个Shard会有对应的ID。每个Shard会有多种状态，具体可以参见下方的状态说明列表。每个Shard启用以后会占用一定的服务端资源，建议按需申请Shard数量。

 **说明**：

表 12-3: 状态说明

状态	说明
Opening	Topic刚创建，所有shard会处于Opening状态直至准备完成，不可读写。
Active	Shard通道打开后，状态会变为Active，此时表示Shard正常可读写。
Closing	Shard进行了Split/Merge操作，后台正在关闭该通道，该状态Shard不可读写。
Closed	Shard在Split/Merge操作完成后，会变为Closed状态，此时Shard为只读状态。

Shard Hash Key Range

每个Shard都有的属性，包括开始和结束的Key范围，写入数据的时候具有相同Key的数据会落到同一个Shard上。对一个Shard的Key范围是左闭右开。

Shard Merge

Shard合并，可以把相邻的Key Range连接的Shard合并成一个Shard。

Shard Split

Shard分裂，可以把一个Shard分裂成Shard Key Range相连接的两个Shard。

Record

用户数据和DataHub端交互的基本单位。

RecordType

Topic的数据类型，目前支持Tuple与Blob两种类型。Tuple类型的Topic支持类似于数据库的记录的数据，每条记录包含多个列；Blob类型的Topic仅支持写入一块二进制数据。

说明：

- Tuple类型下目前只支持写入数据是有格式的数据，具体支持以下几种数据类型。

  表 12-4: Tuple数据类型

类型	含义	值域
Bigint	8字节有符号整型。 **说明**： 请用户不要使用整型的最小值（-9223372036854775808），该数值为系统保留值。	-9223372036854775807 ~ 9223372036854775807
String	字符串，只支持UTF-8编码。	单个String列最长允许1MB
Boolean	布尔型。	可以表示为True/False、true/false、0/1
Double	8字节双精度浮点数。	-1.0 10^{308} ~ 1.0 10^{308}
TimeStamp	时间戳类型。	表示到微秒的时间戳类型

- Blob类型下仅支持写入一块二进制数据作为一个Record，数据将会以BASE64编码传输。