阿里云
专有云大数据版

技术白皮书

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 对象存储OSS

1.1 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。

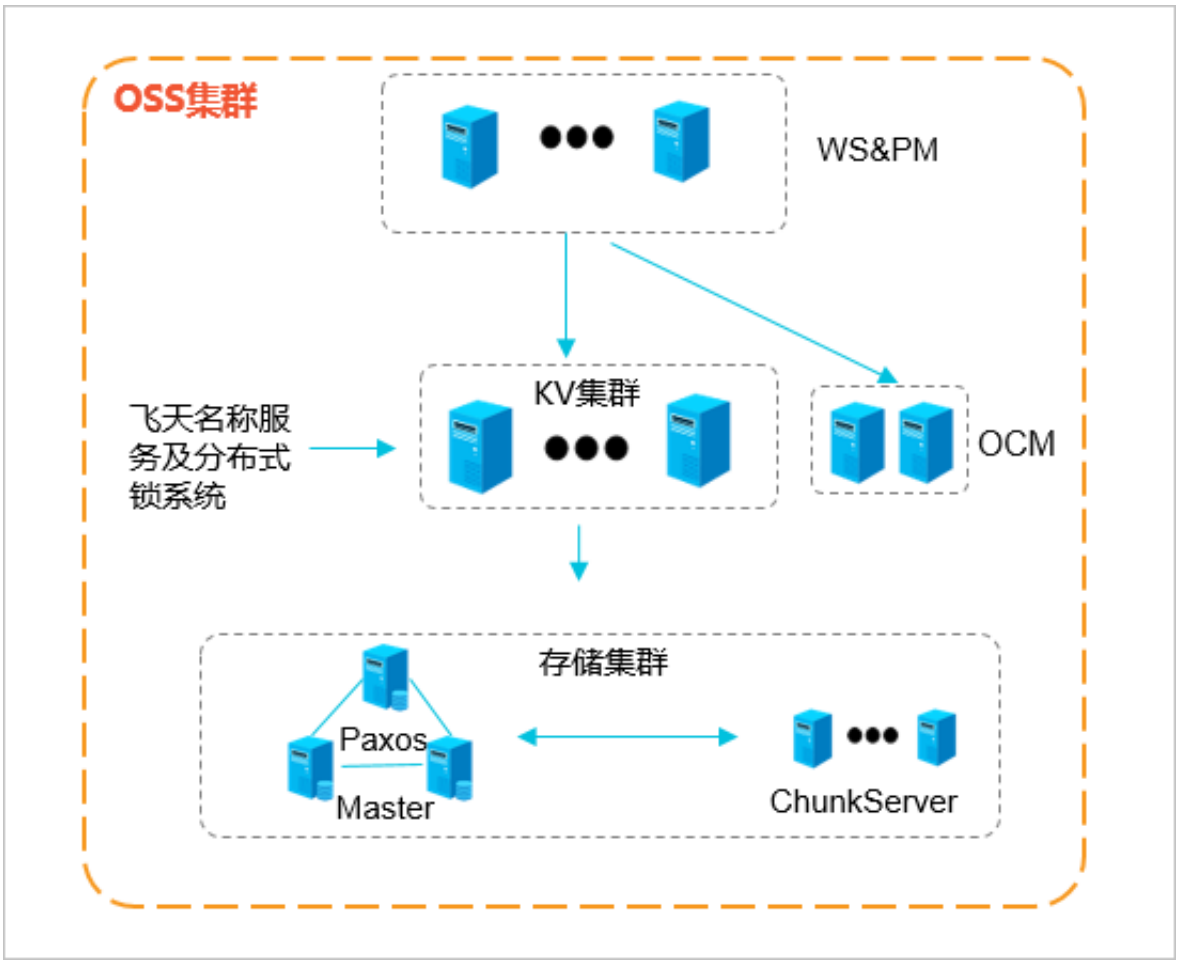
相比传统自建服务器存储，OSS 在可靠性、安全性、成本和数据处理能力方面都有着突出的优势。使用 OSS，您可以通过网络随时存储和调用包括文本、图片、音频和视频等在内的各种非结构化数据文件。

1.2 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。该基础设施提供了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 1-1: OSS架构图



- 最上层是协议接入层，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。
- 数据访问层负责数据结构化处理，即按照Key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。
- 最底层是持久存储层，即大规模分布式文件系统。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，保证数据在系统中有3个备份以及在软硬件错误发生以后的故障恢复。OSS系统的这一设计提供了高可用性和高数据可靠性。

1.3 功能特性

存储空间概览

展示请求者所拥有的所有Bucket，在通过HTTP访问OSS服务地址时将默认展示您所拥有的所有Bucket。

设置并查询Bucket访问权限

- Private（私有权限）：只有该存储空间的创建者或者授权对象可以对该存储空间内的文件进行读写操作，其他人在未经授权的情况下无法访问该存储空间内的文件。
- Public-read（公共读，私有写）：只有该存储空间的创建者可以对该存储空间内的文件进行写操作，任何人（包括匿名访问）可以对该存储空间中的文件进行读操作。
- Public-read-write（公共读写）：任何人（包括匿名访问）都可以对该存储空间中的文件进行读写操作，所有这些操作产生的费用由该存储空间的创建者承担，请慎用该权限。

创建/删除 Bucket

一个用户默认最多创建 10 个 Bucket，当 Bucket 创建数量超过 10 时将返回错误信息。新创建的 Bucket 命名要符合Bucket命名规范，否则返回错误标志。如果创建的 Bucket 不存在，系统按照 Bucket 名称创建 Bucket，并返回成功标志；如果要创建的Bucket已存在，且请求者是所有者，则保留原来 Bucket，并返回成功标志；如果要创建的 Bucket 已存在，且请求者不是所有者，则返回失败标志。Bucket删除成功的条件有如下几个：Bucket 存在，访问者对 Bucket 有删除权限，Bucket 为空。

列出 Bucket 中的所有 Object

根据 Bucket 名称列出此 Bucket 下的所有 Object 信息，访问者必须具有对相应 Bucket 的操作权限，当访问的 Bucket 不存在时返回错误信息。

OSS 支持前缀查询，可以设置一次最大返回的文件数量（最大支持设置1000）。

上传/删除 Object 文件

上传 Object 到指定的 Bucket 空间下。在满足如下条件下 Object 会上传成功：Bucket 存在、访问者拥有对 Bucket 相应的操作权限。当 Bucket 中存在同名 Object 文件时将会覆盖掉原来的 Object 文件。根据 Object 名称删除某个特定 Object，访问者必须有此 Object 相应的操作权限。

获取 Object 文件或元信息

取得 Object 文件内容信息或者元信息，访问者需要对 Object 有相应的操作权限。

访问 Object

OSS 支持通过 URL 方式访问某文件。

日志及监控操作

用户可以选择开启 Bucket 的日志记录功能，一旦开启，OSS 会按照小时粒度推送日志。用户可以通过 OSS 控制台查询存储空间、流量、请求等信息。

2 云盾

2.1 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。

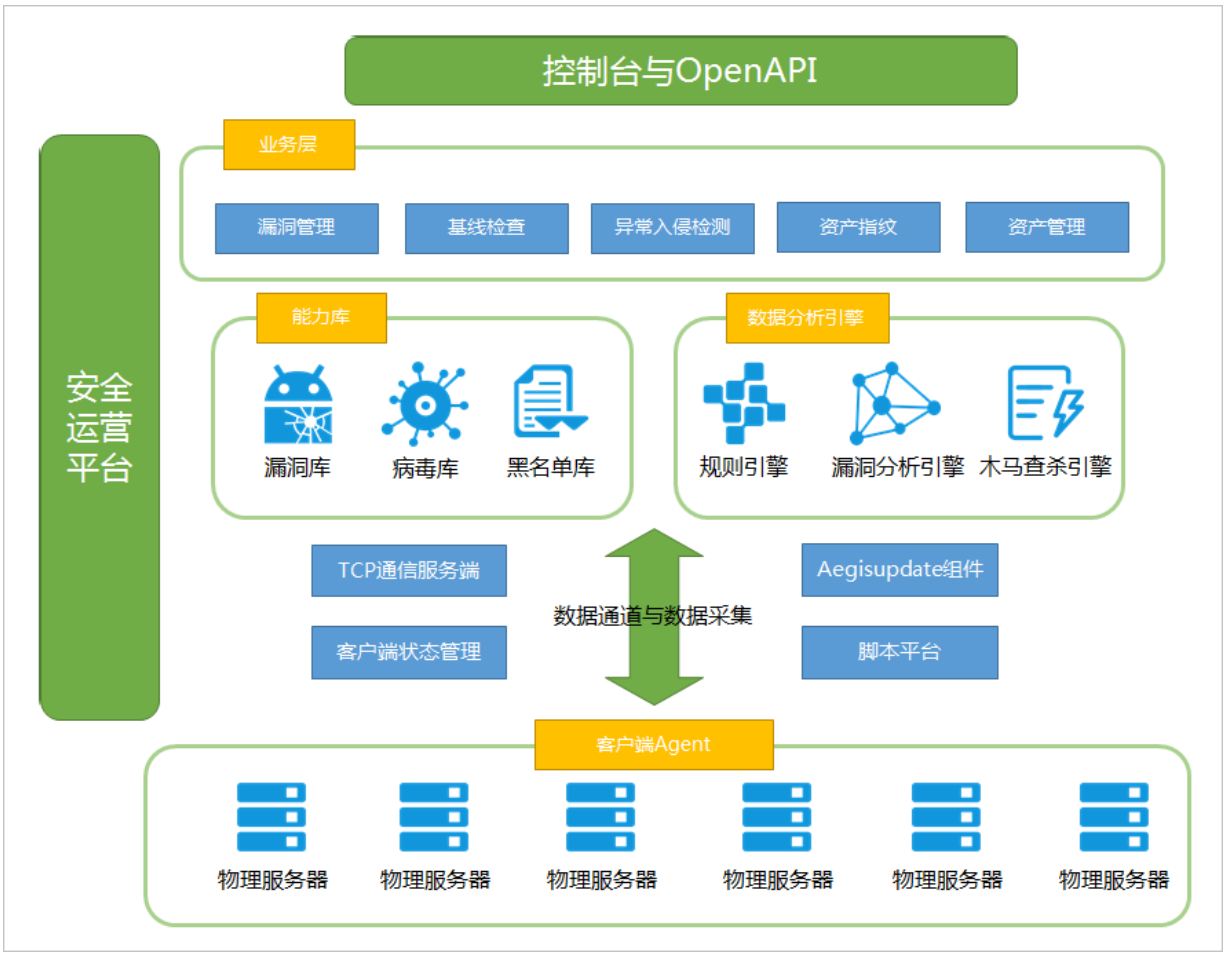
2.2 产品架构

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 *2-1:* 安骑士产品结构图所示。

图 2-1: 安骑士产品结构图

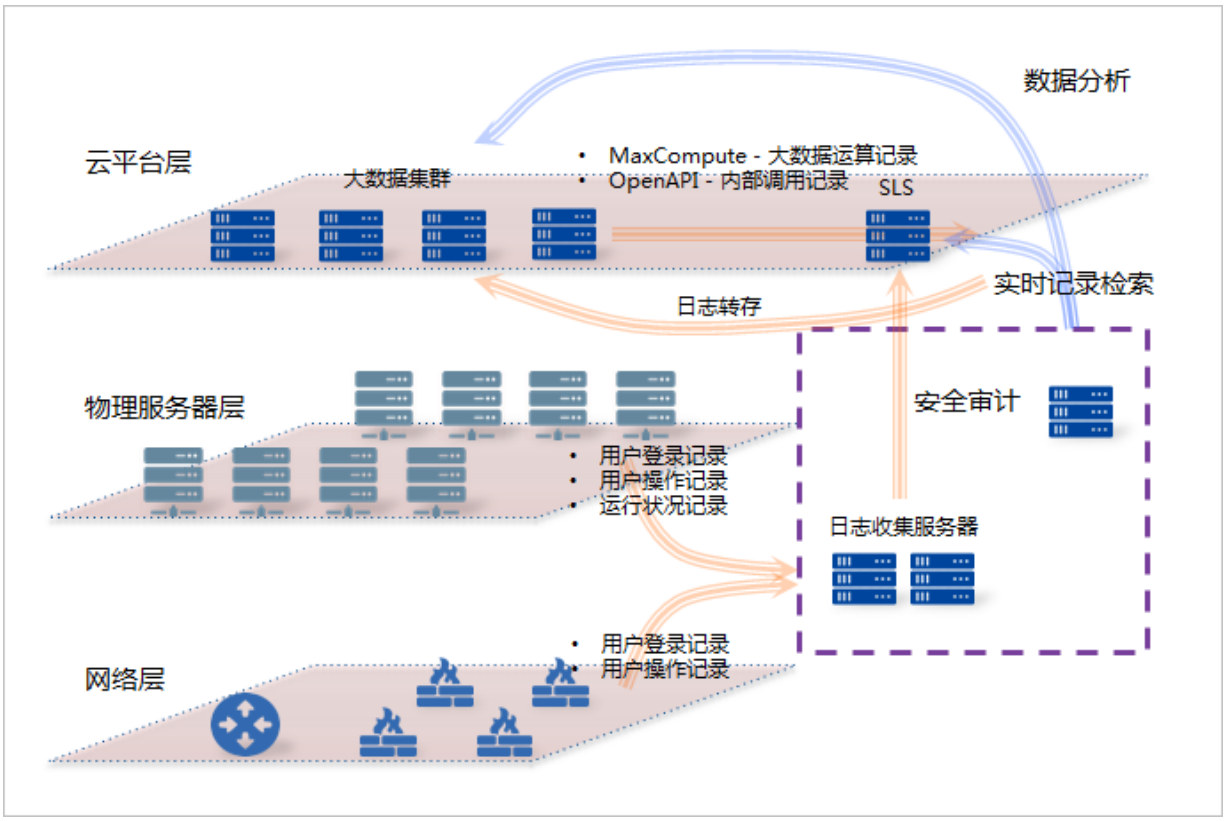


安骑士模块采用Client-Server架构，在物理服务器上的客户端Agent与服务端通过TCP长连接进行通信，通过HTTP方式从服务端获取需要的脚本、规则、安装包等文件。基于客户端收集到的主机的信息，通过日志监控、文件分析、特征扫描等手段，为专有云环境中的主机提供漏洞管理、基线检查、入侵检测、资产管理等安全防护措施。

安全审计

安全审计功能模块的产品结构如图 2-2: 安全审计产品结构图所示。

图 2-2: 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。

2.3 功能特性

2.3.1 安骑士

安骑士模块通过日志监控、文件分析、特征扫描等手段，为服务器提供漏洞管理、基线检查、入侵检测、资产管理等安全防护措施。安骑士模块分为客户端和服务器端，客户端配合安骑士服务器，监测针对主机系统层和应用层的攻击行为及漏洞信息，实时防护主机安全。

功能模块

安骑士模块提供以下功能：

功能分类	功能项	功能说明
漏洞管理	Linux软件漏洞检测	通过检测服务器上安装软件的版本信息，与CVE官方的漏洞库进行匹配，检测出存在漏洞的软

功能分类	功能项	功能说明
		件（包括SSH、OpenSSL、MySQL等软件漏洞）并给您推送漏洞信息，提供修复建议。
	Windows漏洞检测与修复	通过订阅微软官方更新源，若发现您服务器存在高危的官方漏洞未修复，将为您推送微软官方补丁进行修复（如“SMB远程执行漏洞”）。 **说明**： 默认系统只推送高危漏洞，安全更新和低危漏洞您可以手动更新。
	Web-CMS漏洞检测与修复	同步阿里云安全情报源，通过目录及文件的检测方案，检测Web-CMS软件漏洞，提供云盾自研补丁（支持修复例如Wordpress、Discuz等软件漏洞），且支持漏洞修复及回滚操作。
	配置型、组件型的漏洞检测	检测无法通过版本匹配和文件判断方式发现的漏洞，精准识别软件高危配置漏洞，例如Redis未授权访问、ImageMagick等配置型、组件型漏洞。
基线检查	账号安全基线检查	• 检测SSH、RDP、FTP、MySQL、PostgreSQL、SQLServer的弱口令账号。 • 检测云服务器中可疑的隐藏账号、克隆账号等风险账号。 • 检测Linux系统服务器中的密码策略合规性。 • 检测云服务器中的空密码账户。
	系统配置检测	对系统组策略、登录基线策略、注册表配置风险进行检测，包括： • 检测Linux系统服务器的定时任务中是否包含可疑的自启动项目。 • 检测Windows系统服务器中的自启动项。 • 检测系统共享配置。 • 检测Linux系统服务器的SSH登录安全策略配置。 • 检测Windows系统服务器中账号相关的安全策略。
	数据库安全基线检查	支持检测服务器上的Redis服务是否对公网开放、是否存在未授权访问漏洞并向系统关键文件写入异常数据的情况。

功能分类	功能项	功能说明
	合规对标检测	按照CIS-Linux Centos7最新基线标准进行系统层面基线合规检测。
入侵检测-异常登录	异地登录告警	自动记录所有登录记录，通过分析和记录用户常用登录位置，识别常用的登录区域（精确到地市级），对疑似的非常用地登录行为进行告警，且支持自定义常用登录地配置。
	非白名单IP登录告警	配置登录白名单IP后，对来自非白名单IP的登录事件进行告警。
	非法时间登录告警	配置合法登录时间后，对非合法时间段内的登录事件进行告警。
	非法账号登录告警	配置合法登录账号后，对非合法账号的登录事件进行告警。
	暴力破解登录拦截	对非法暴力破解密码的异常登录行为进行识别，实时检测并拦截暴力破解攻击，避免被黑客多次猜解密码而导致入侵。支持对SSH和RDP服务的暴力破解行为进行监控。
入侵检测-网站后门查杀	网站后门（Webshell）查杀	通过自研网站后门查杀引擎对云服务器中存在的脚本后门进行精准查杀（支持配置定时查杀和实时防护扫描策略），识别包括ASP、PHP、JSP编写的脚本后门文件，并可手动对脚本后门文件进行隔离。
入侵检测-主机异常进程	进程异常行为检测	检测诸如反弹Shell、JAVA进程执行CMD命令、bash异常文件下载等进程异常行为。
资产管理	资产分组	支持对服务器进行最多四级分组，并可按照地域、在线状态等信息进行筛选，且支持资产标签管理功能。
	资产指纹	• **端口监听**：对端口监听信息收集和呈现，对变动进行记录，便于清点端口开放情况 • **账号管理**：收集账户及对应权限信息，清点特权账户、发现提权行为 • **进程管理**：通过进程快照信息的收集和呈现，清点合法进程、发现异常进程 • **软件管理**：清点软件安装信息，在高危漏洞爆发时快速定位受影响资产

功能分类	功能项	功能说明
主机日志	日志检索	• **进程相关日志**： - 进程启动：进程一旦启动，记录该启动事件的详细信息 - 进程快照：记录某一时刻的进程全量日志抓取并存储 • **网络相关日志**： - 五元组日志：主机端和网络端同时采集的连接五元组信息 - Web日志：从网络上抓取的HTTP的访问日志（暂不支持HTTPS） - DNS日志：对外请求的DNS解析日志（暂不支持内网DNS） • **其他日志**： - 系统登录：通过SSH、RDP方式的系统登录流水日志 - 端口监听快照：某一时刻的所有对外监听端口的快照数据 - 账号快照：某一时刻的所有账号信息的快照数据

工作原理

安骑士客户端程序包含Webshell特征库、Webshell隔离模块、补丁特征库和补丁修复模块：

- Webshell特征库的功能是用于检测文件是否符合特征库的特征。如果符合，安骑士客户端将木马文件发送至安骑士服务器端，由安骑士服务器端结合更多的特征库进一步分析是否为木马文件。
- Webshell隔离模块的功能是安骑士客户端通过向安骑士服务器端确认所发现的文件为木马文件后，对该文件进行隔离。
- 补丁特征库的功能是用于检测文件是否符合特征库的特征。如果符合，安骑士客户端将漏洞文件发送至安骑士服务器端，由安骑士服务器端结合更多的特征库进一步分析是否为漏洞文件。
- 补丁修复模块的功能是安骑士客户端通过向安骑士服务器确认文件存在漏洞文件后，对该文件进行修复。

安骑士客户端分为Windows版本和Linux版本，客户端自动连接到安骑士服务器端进行在线升级。

安骑士服务器端包括Aegis-server、Defender和Aegis-health-check三个模块：

- Aegis-server包括通讯模块和客户端检查模块，主要功能是与安骑士客户端进行交互，收集木马文件、补丁文件等信息，并向Defender模块反馈异地登录信息、暴力破解行为和暴力破解成功信息。
- Defender的主要功能是进行异地登录分析、暴力破解分析和暴力破解是否成功的分析。
- Aegis-health-check的主要功能是进行基线检查，通过Aegis-server向客户端下发基线检查的命令，并且通过Aegis-server接收客户端返回的数据，更新基线检查的状态和结果。

应用场景

安骑士适用于下列场景的主机安全防护：

- **使用通用软件建站的场景**

  在这种场景下，极可能因通用软件的漏洞被利用而导致被黑客入侵，使用安骑士进行漏洞检测，一旦发现漏洞可快速进行一键修复。

- **使用Web应用服务的场景**

  无论是内部Web服务还是外部Web服务，黑客都可能通过Web服务窃取网站的核心数据，使用安骑士可有效阻止黑客从外部的攻击和内部的渗透。

性能指标

为保证服务器性能，安骑士客户端包含以下性能限制：

- **正常工作状态**： 安骑士客户端程序CPU占用为1%，内存占用为50M。
- **工作峰值状态**： 安骑士客户端程序CPU占用为10%，内存占用为80M。超过该峰值时，安骑士客户端程序将自动暂停。

2.3.2 安全审计

安全审计模块提供基于云计算平台的一体化审计解决方案。对标信息系统安全等级保护基本要求，安全审计模块从物理服务器层面、网络设备层面、云计算平台应用层面分别进行审计，实现行为日志的收集、存储、分析、报警等功能。

功能模块

安全审计模块提供以下功能：

功能项	功能说明
原始日志采集	支持收集MaxCompute运算记录日志、主机syslog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syslog日志。

功能项	功能说明
	说明： 网络设备日志需要进行手动接入。
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。

工作原理

安全审计主要由Auditlog核心组件完成。Auditlog收集网络设备、物理服务器、RDS、OpenAPI、控制台的日志记录，云安全中心通过调用Auditlog的API接口获取审计日志数据、审计策略和审计事件，对审计日志和审计事件进行分析并以图表方式呈现，帮助用户直观了解系统面临的风险和威胁。

功能优势

安全审计模块具有以下特点和优势：

- **行为日志全面无死角**

  覆盖云计算的多个业务和物理服务器主机，从各个角度对行为进行收集，保证不会因为覆盖面不够导致的审计缺失。日志收集中心集中、准实时、同步回收行为日志。

- **日志存储可靠**

  日志的存储基于云计算存储业务，通过集群化三备份，保障存储安全稳定性。存储空间也可快速扩充。

- **海量数据实时查询**

  通过对海量日志数据构建全文索引，具备大量数据的快速检索查询能力。当前最多可支持500亿条日志的同时索引。

2.4 产品价值

跟云平台深度耦合的安全方案

十年攻防，一朝成盾。在经历了阿里巴巴集团自身业务十年来的安全护航以及阿里云六年安全运营保障，阿里巴巴积累了大量的安全研究成果、安全数据和安全运营方法，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一套专门面向云平台的攻击防护产品，可有效地保护专有云上用户的云网络环境和业务系统的安全。

云盾的各组件是软件虚拟化，具有较为广泛的硬件兼容性，可以快速的部署扩容和投入使用，适应云计算弹性的特点。

提供租户维度的安全自服务感知

云平台是面向租户维度的，租户可以通过云安全中心控制台查看租户侧自身的安全防护情况，生成简单的报表，并且通过外部资源配置可以实现短信和邮件方式的告警通知。

安全能力输出

云盾的防护策略和数据源自于多年的积累，百万级的用户每天面临多达几十万次的各种攻击，阿里充分利用了这些安全攻防的数据积累，每天对10多TB的安全数据进行分析。分析结果形成恶意IP库、恶意行为库、恶意样本库、安全漏洞库等基础安全能力，并及时应用到云盾的各个防护模块中，提升云盾防护能力，为用户带来更好的安全保障。

3 MaxCompute

3.1 什么是MaxCompute

大数据计算服务（MaxCompute）是基于飞天操作系统分布式平台，由阿里云自主研发的海量数据离线处理服务。MaxCompute提供针对TB/PB级别数据、实时性要求不高的批量处理能力，主要应用于日志分析、机器学习、数据仓库、数据挖掘、商业智能等领域。

3.2 产品特性和核心优势

产品特点

- 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，是阿里巴巴大数据平台的基础计算平台。MaxCompute中的多租户、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。
- 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。

产品优势

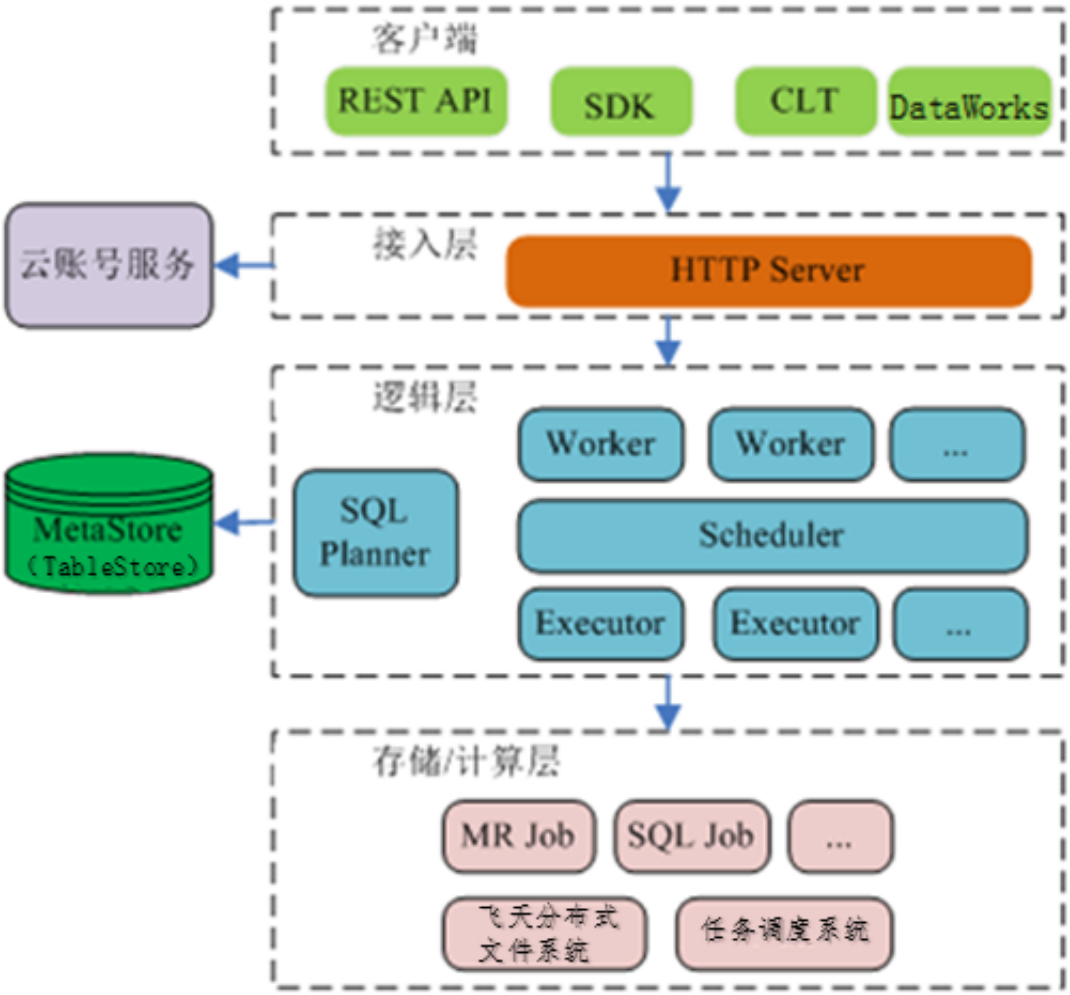
- **海量运算触手可得**：用户不必关心数据规模增长带来的存储困难、运算时间延长等烦恼，根据用户的数据规模自动扩展集群的存储和计算能力，使用户专心于数据分析和挖掘，最大化发挥数据的价值。
- **服务“开箱即用”**：用户不必关心集群的搭建、配置和运维工作，仅需简单的几步操作，用户便在MaxCompute中上传数据、分析数据并得到分析结果。

- **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**：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提高工作效率。

3.3 产品架构

MaxCompute的功能架构如图 *3-1: MaxCompute*功能架构图所示：

图 3-1: MaxCompute功能架构图



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。

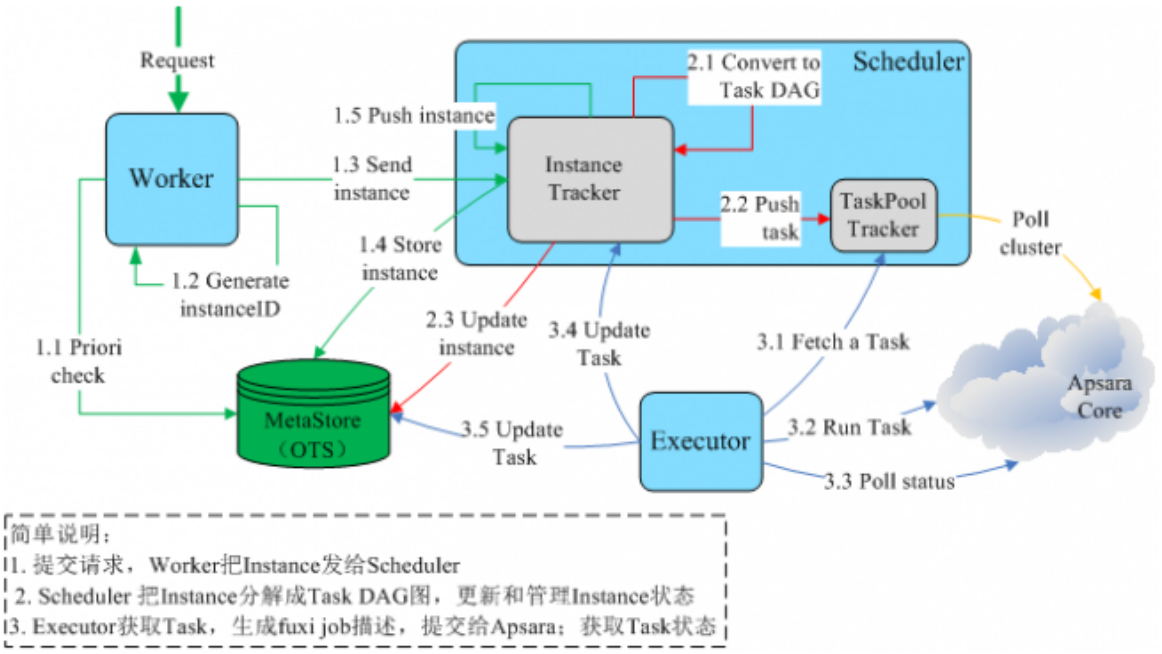
MaxCompute接入层提供HTTP（HTTPS）服务、Load Balance、用户认证和服务层面的访问控制。

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。其对应的业务执行逻辑图如下图所示：

图 3-2: 业务执行逻辑图



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，是飞天操作系统内核，运行在和控制集群独立的计算集群上。上面的MaxCompute架构图中仅列出了若干飞天内核主要模块，例如：飞天分布式文件系统、任务调度系统等。

其中，**飞天分布式文件系统**是一个分布式文件系统，其设计目标是将大量通用机器的存储资源聚合在一起，为用户提供大规模和可靠的分布式存储服务，是飞天操作系统内核的重要组成部分。

飞天分布式文件系统目前包括三台master和多台chunkserver，前者负责文件meta信息的存储和管理，后者负责数据存储。为了保证可靠性，同一个数据块会被存储在多个chunkserver上，一般情况下飞天分布式文件系统保存的数据会有3份副本，所有的MaxCompute数据文件都保存在飞天分布式文件系统上，在 /product/aliyun/odps飞天分布式文件系统目录下可以找到。需要注意，其中master是热备，同一时间只有一台在工作。

- **master**：

  1. PanguMaster维护了整个文件系统的元数据，其中包括命名空间、文件到数据块的映射及数据块的存储地址等。
  2. 作为整个分布式文件存储系统的大脑，PanguMaster控制着系统层面的活动如孤立数据块的垃圾回收、chunckserver间的数据合并、判断chunkserver是否健康、恢复由于chunkserver宕机所造成的数据块丢失等。

3. PanguMaster同时还负责调节同一时间片中多台客户端对数据的访问来维护同一集群中数据的完整性。
4. PanguMaster只对客户端提供元数据相关的操作，而数据传输相关的通讯都直接在chunkservers间进行。

- **chunkserver**：飞天分布式文件系统上的文件会被切成多个固定大小的存储单元，每个存储单元都叫一个chunk。存储数据块chunk的机器称为chunkserver，对每个数据块，创建之初，pangumaster都会分配一个128位的ID，飞天分布式文件系统客户端根据ID来读取存储在磁盘上的数据块。

同时，为了更好地支持MaxCompute表中的结构化数据，MaxCompute使用统一的数据文件格式，实现了一种特殊格式的飞天分布式文件系统文件，称为CFile。

- CFile是一种基于列存储的文件格式，其主要目的是为了降低离线数据处理过程中的无效磁盘读取操作。文件中的数据以列为单位聚簇组织（聚簇被称为Block），并在存储到文件系统之前进行压缩，减少了存储空间的占用。在离线数据处理的场景中，用户只需要读取待处理的数据，避免了无用的磁盘操作，提高读取的磁盘效率，同时减少了网络带宽的使用。
- CFile文件的存储结构逻辑上可以分为三个区域，分别为数据区（Data区）、索引区（Index区）以及元信息区（Header区）。其中，数据区存储的是按照列划分，以Block为组织单位的用户数据；索引区存储的是每列的数据Block对应的索引，其中包含每个Block在文件中的起始位置、Block压缩后的长度以及Block内的数据个数（对非定长的数据类型如string而言）。元信息区存储了该文件中每一列的元信息，例如该列的索引在文件中的起始位置以及索引长度、该列的类型信息、压缩方法等，以及文件中用户数据的行数以及版本号等。
- 目前支持如下五种原始的数据类型（即MaxCompute支持该五种类型）：
  - BIGINT，8字节有符号整型。
  - BOOLEAN，布尔型，包括TRUE/FALSE。
  - DOUBLE，8字节双精度浮点数。
  - STRING，字符串，需要注意在MaxCompute中的函数会假设STRING中存的是UTF8编码的字符串，其他编码格式可能会导致异常。
  - DATETIME日期类型，格式为**`YYYY-MM-DD HH:mm:SS`**，例如：2012-01-02 10:09:25。

而**任务调度系统**则是飞天操作系统平台内核中负责资源管理和任务调度的模块，也为应用开发提供了一套编程基础框架。其主要功能是充分利用整个集群的硬件资源来服务用户和系统的计算需求。任务调度系统同时支持两种应用类型的计算：低延迟的在线服务和高吞吐的离线处理，分别称为Fuxi Service和Fuxi Job，可以对应Hadoop中的Yarn。

- **Fuxi Service**：Service是任务调度系统上的常驻进程，由用户申请创建及销毁，任务调度系统不会主动销毁Service进程。
- **Fuxi Job**：Job是任务调度系统上的临时任务，一旦任务结束，资源就会被释放，由任务调度系统回收。

在资源管理上，任务调度系统负责调度和分配集群的存储、计算等资源给上层应用，支持计算资源额度、访问控制和作业优先级，保证有效的资源共享。在任务调度上，任务调度系统提供了数据驱动的多级流水线并行计算框架，类似于MapReduce编程模式，适用于海量数据处理和大规模计算等复杂应用。

任务调度系统目前包括两台master和多台tubo，其中master是冷备，同时只有一台在工作。而每台计算节点上都启动了一个tubo进程，用来管理单机资源，例如汇总整机的可用的CPU、内存、硬盘、网络，同时记录每台机器已被占用的资源，每个机器上的tubo进程会将这些资源汇报给FuxiMaster，由FuxiMaster统一管理及调度。

3.4 产品价值

MaxCompute与传统数据库相比，具有如下的价值点：

表 3-1: 价值点对比

价值点	传统数据库	MaxCompute
系统扩展能力	共享磁盘技术，难以超过100节点，分库分表技术将导致应用数据碰撞、比对等大规模计算带来海量数据的数据交换开销，极大限制应用的分析能力。	超过10000节点，支持超过1.5EB数据量。例如阿里双11活动中，6小时内MaxCompute处理数据量超过300PB。
数据类型支持	不适用于非结构化计算。	同时支持结构化、非结构化数据处理。
高可用性	无冗余存储，传统备份恢复对PB级数据量不可用，单点磁盘故障会导致全库不可用。	无共享的多副本技术，无单点故障。
复杂计算能力	不支持迭代计算，不支持图计算。共享磁盘技术，复杂计算导致节点间大量数据交换，带宽难以支持。	分布式存储，支持MR、SQL、迭代计算、MPI、图计算等多种计算框架。
并发能力	一个大规模计算任务可能会消耗掉全部系统资源（如索引计算），存	具有完善的多租户资源隔离和资源管理工具，用户对集群资源一目了然，并且很方便的管

价值点	传统数据库	MaxCompute
	在网络、热点盘（数据字典）等多种瓶颈，无法支持高并发访问。	理每个业务使用的资源量，可以支持超过1万的并发访问。
性能支持	索引机制导致实时入库数据难以支持分析型应用，海量数据碰撞比对、分析预测计算时间可能超过24小时，性能无法满足需求。	关注与海量数据的并行计算能力，支持数据实时入库可用、具备高性能大规模离线计算、海量数据实时多维分析、流计算等多种高性能计算能力。

3.5 功能描述

3.5.1 Tunnel

Tunnel是MaxCompute提供的数据通道服务，各种异构数据源都可通过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。

3.5.2 SQL

MaxCompute SQL是一种结构化查询语言，语法和Oracle/MySQL/Hive SQL类似，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。

MaxCompute SQL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。

3.5.3 MapReduce

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

用户可以使用MapReduce提供的接口（Java API）编写MapReduce程序处理MaxCompute中的数据。概念“Map（映射）”和“Reduce（归约）”和它们的主要思想，都是从函数式编程语言和矢量编程语言中借鉴来的特性。它极大地方便了编程人员在不会分布式并行编程的情况下，将自己的程序运行在分布式系统上。

当前的软件实现是指定一个Map（映射）函数，用来把一组键值对映射成一组新的键值对，指定并发的Reduce（归约）函数，用来保证所有映射的键值对中的每一个共享相同的键组。

MaxCompute MapReduce特性：

- Hadoop-style，针对MaxCompute场景设计（用于处理Table和Volume）。
- 输入输出仅支持MaxCompute内置类型。
- 可以输入多表，输出多表或到不同分区。
- 可以读资源（Resource）。
- 不支持输入view。
- 受限的沙箱安全环境。

3.5.4 Graph

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框架开发高效的机器学习或数据挖掘算法。

在互联网环境下，存在很多海量图结构的数据，例如社交网络、物流信息等，这类图计算模型的典型特点是迭代，整个计算过程是通过一轮一轮反复迭代求解，最后达到一个收敛状态。例如对于需要迭代学习模型参数的机器学习算法而言，图计算模型比MapReduce有天然优势。在实际应用中，用户将问题抽象成图，然后以顶点为中心，通过超步进行迭代更新。

MaxCompute Graph目前提供两种模式：

- 离线模式：适用于计算规模较大的场景，类似于MapReduce作业，每次运行完成加载和计算两个过程。
- 交互模式：适用于计算规模较小的场景，用户实现UDF，然后通过命令行方式交互。

在离线模式下，加载和计算是两个独立的步骤，数据加载后会常驻内存，用户可以对数据执行不同的计算逻辑。例如风控部门每天会加载一次数据，运营人员会对这份数据执行不同的查询逻辑，查看数据之间的关系。

在阿里巴巴内部，MaxCompute Graph已经有很多应用，例如实现带权重的PageRank算法计算支付宝用户身边影响力指数；实现变分贝叶斯EM模型，基于用户购买的商品属性信息，推测用户的汽车品牌分布等。

3.5.5 非结构化数据的访问及处理

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。

3.5.6 增强功能

3.5.6.1 Spark on MaxCompute

3.5.6.1.1 开源平台——Cupid

MaxCompute是阿里云自主研发的大数据解决方案，其规模和稳定性在业界都是领先的。大数据开源社区当前也非常活跃，应对各类需求的系统快速出现并发展。为了更好服务用户，同时也为了丰富MaxCompute的生态，MaxCompute团队研发了Cupid平台，把MaxCompute跟开源社区连接到一起，兼顾开源社区的多样性和飞天操作系统的规模和稳定性。

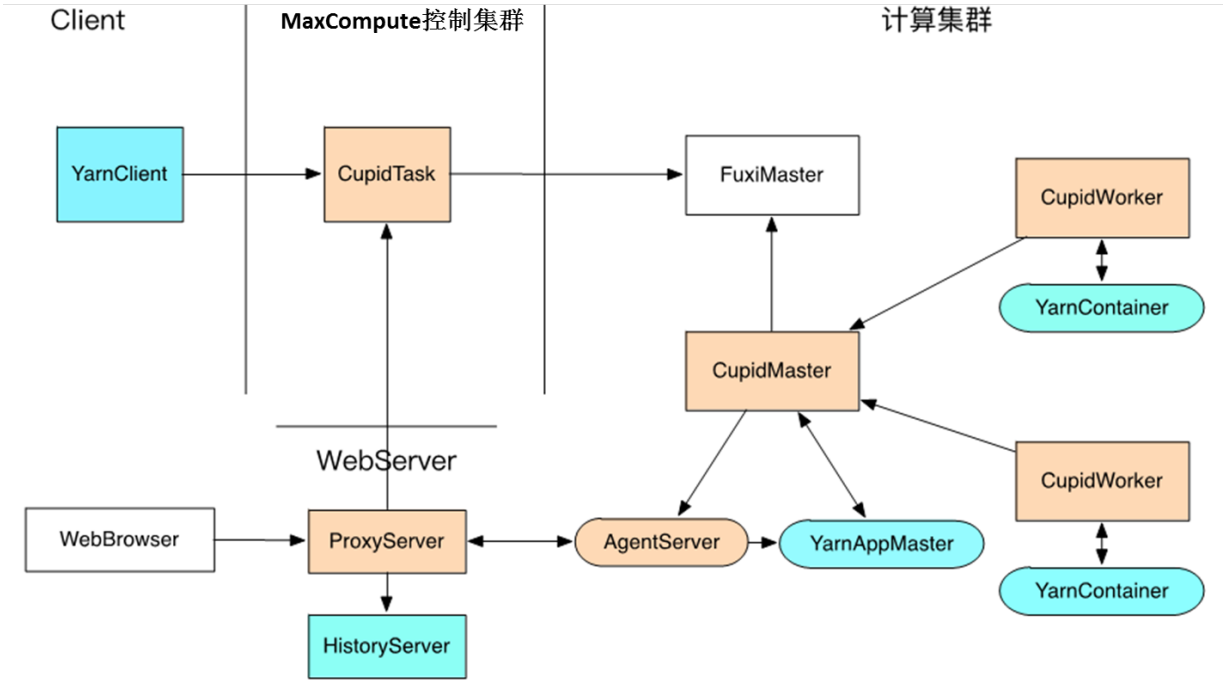
开源社区和飞天操作系统的软件栈比较相似，但也有不同之处。

对于分布式文件系统，开源社区大多使用HDFS，飞天操作系统里面则是使用飞天分布式文件系统；对于分布式调度系统，开源社区用的比较多的是Yarn，飞天操作系统里面则是使用任务调度系统；在此之上是应对各类场景的计算引擎。为了能够把开源应用（例如Spark）运行在MaxCompute系统中，Cupid首先要做兼容。

3.5.6.1.1.1 兼容Yarn

Yarn面向应用的接口主要有三个：YarnClient、AMRMClient和NMClient。YarnClient用于客户在应用端提交作业到集群；AMRMClient用于AppMaster向ResourceManager申请和释放资源；NMClient用于启动和停止应用的Container。

图 3-3: Yarn on MaxCompute



一个Yarn上App执行到MaxCompute平台的流程如上图所示，其中黄色部分是Cupid平台的组件，浅蓝色部分是开源组件。具体流程如下：

1. 用户通过YarnClient访问MaxCompute控制集群，提交作业给FuxiMaster。
2. FuxiMaster启动一个CupidMaster，CupidMaster按照Yarn的协议把YarnAppMaster启动起来。
3. YarnAppMaster通过CupidMaster跟FuxiMaster交互，申请和释放资源。
4. 启动新的Container时，任务调度系统的tubo会先启动一个CupidWorker，CupidWork根据Yarn的协议启动Container。

说明：
YarnAppMaster中一般都会有UI，Cupid通过代理机制实现。

3.5.6.1.1.2 兼容FileSystem

开源社区里面大多使用HDFS作为分布式存储，Hadoop社区提供了FileSystem接口，目前阿里云的OSS，亚马逊的s3都兼容了这个接口。MaxCompute团队同样实现了飞天分布式文件系统兼容FileSystem接口，提交到MaxCompute集群的开源job同样可以原封不动运行起来，并且获得原生飞天分布式文件系统的性能。

说明：

需要注意，飞天分布式文件系统不直接对外提供服务，飞天分布式文件系统上面的数据只能作为作业中间数据，例如checkpoint之类的。如果用户想要使存入的数据在其他环境同样能够访问，目前可以选择使用OSS。

3.5.6.1.1.3 DiskDrive

开源应用大多会使用本地文件系统来进行数据处理，例如spark shuffle、storage等。在大集群环境下，磁盘是一种非常重要的系统资源，应该被统一管理，才能保证高可用、高性能以及安全性。在飞天操作系统中，磁盘都是被飞天分布式文件系统统一管理。为了在飞天分布式文件系统的基础上提供本地文件系统接口，Cupid把网盘引入到MaxCompute，设计并实现了DiskDriverService系统。

3.5.6.1.2 功能扩展

由于MaxCompute提供了Cupid框架来支持开源应用，因此，Spark可以在MaxCompute上面运行起来。为了更好的方便用户使用，也为了更好地跟MaxCompute融合，Spark on MaxCompute需要对Spark进行功能扩展，主要包括：保证Spark开源程序的安全隔离；跟MaxCompute数据打通；在多租户的集群中提供交互式。

3.5.6.1.2.1 安全隔离

Spark作业提交到MaxCompute计算集群中，为防止用户恶意代码对系统进行攻击，Spark作业需要运行在安全沙箱里面。整体使用父子进程架构，Cupid框架代码运行在父进程中，Spark代码运行在子进程中。Spark需要访问系统服务时，可以通过父子进程通信的方式，由父进程代理访问。

3.5.6.1.2.2 数据互通

运行在MaxCompute中的Spark的一个优势是，Spark作业和MaxCompute作业在整个集群资源上是统一的，可以做到相互之间数据直接访问，而不需要跨级群拖数据。

数据互通包括两个方面：元数据和存储数据，并且Spark存储MaxCompute数据必须通过MaxCompute账号体系进行鉴权。**Spark on MaxCompute**实现了OdpsRDD、OdpsDataFrame，可以满足用户对Spark这两种API的需求。同时Spark SQL可以直接访问MaxCompute元数据进行SQL优化，并且在物理层直接存取MaxCompute数据格式。

3.5.6.1.2.3 Client模式

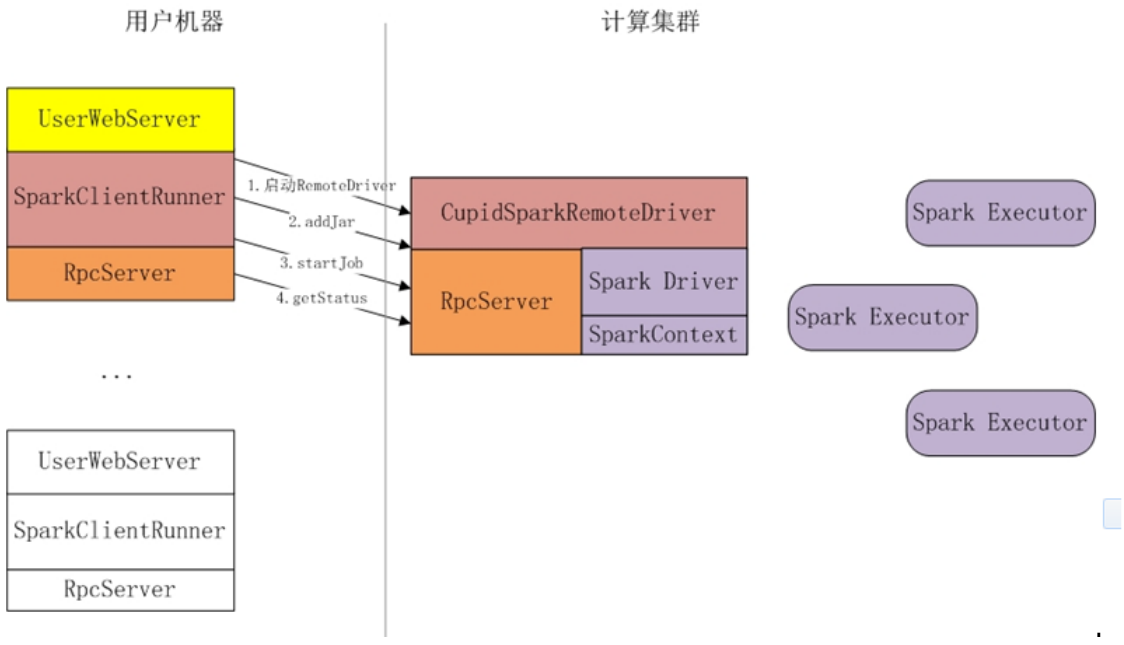
社区spark生产主要使用**`yarn-cluster`**、**`yarn-client`**两种模式。**`yarn-cluster`**模式将spark作业提交到集群运行，运行完毕后客户端打印状态日志。这种模式无法向一个Spark Session动态多次提交作业，且客户端无法获取每个job的状态及结果；而**`yarn-client`**模式，主要解决spark交互式场景问题，需要在客户端机器启动Driver，无法将Spark Session作为一个服务。

因此MaxCompute团队基于**Spark on MaxCompute**开发了`Client`模式来解决上述的问题。该模式具有以下特点：

1. 客户端轻量级，不用再启动spark的Driver。
2. 客户端有一套API向MaxCompute集群的同一个Spark Session动态提交作业，并监控状态。
3. 客户端可以通过监控作业状态及结果，构建作业之间的依赖关系。
4. 用户可以动态编译应用程序jar，通过客户端提交到原有的Spark Session运行。
5. 客户端可以集成在业务的WebServer中，并且可以进行水平扩展。

`Client`模式会先通过CupidSparkClientRunner在MaxCompute集群启动一个Spark Session。后续通过CupidSparkClientRunner在客户端提交作业、获取作业状态及运行结果，同时各个作业之间可以共享cache的数据等；也可以构建多个CupidSparkClientRunner与同一个Spark Session交互，运行时的框架图如下图所示。

图 3-4: Spark Client模式



具体使用Spark Client模式的执行流程如下所示：

1. 向MaxCompute集群提交启动CupidSparkRemoteDriver，并获取SparkClientRunner对象。
2. 通过SparkClientRunner添加要执行作业的jar包到RemoteDriver。
3. SparkClientRunner使用job的classname向RemoteDriver提交运行。
4. SparkClientRunner通过提交作业返回的job id来监控作业的状态。

3.5.6.1.2.4 Spark生态支持

Spark拥有丰富的生态：Mllib、Streaming、PySpark、SparkR、Graphx、SQL。**Spark on MaxCompute**对Spark的完整生态提供支持，使用方式跟开源社区保持一致。此外，**Spark on MaxCompute**也支持Spark UI以及历史日志的访问。

3.6 应用场景

MaxCompute主要面向三类大数据处理场景：

- 基于SQL构建大规模数据仓库和企业BI系统。
- 基于MapReduce和MPI的分布式编程模型开发大数据应用。
- 基于统计和机器学习算法，开发大数据统计模型和数据挖掘。

3.6.1 搭建数据仓库

图 3-5: 搭建数仓

使用MaxCompute可以轻松打造一个云端数据仓库，借助MaxCompute的分区、数据表统计、表的生命周期等功能，用户可以很方便的实现数据仓库的历史信息增强存储、冷热表区分、数据质量控制等场景。

阿里金融数仓团队基于MaxCompute构建了一个完善复杂、功能强大的数据仓库体系，包含六个层次：源数据层、ODS层、企业数据仓库层、通用维度模型层、应用集市层和展现层。

- 源数据层处理各个来源数据，包括淘宝、支付宝、B2B、外部数据等。

- ODS作为数据导入的临时存储层。
- 企业数据仓库层采用3NF建模方式，按主题（如商品、店铺）进行划分，包括完整的历史数据。
- 通用维度模型以维度建模方式构建面向通用业务应用的模型层，不以满足特定的应用为目的。通用维度模型层的目的是屏蔽业务需求变化，以一致性维度和事实的方式为上层提供数据。
- 应用集市层是面向需求，构建满足某一应用需求的数据集市。
- 展现层提供一些数据门户（Portal）和服务等，供应用访问。

在这个体系架构中，不可避免会涉及元数据管理等其他方面。

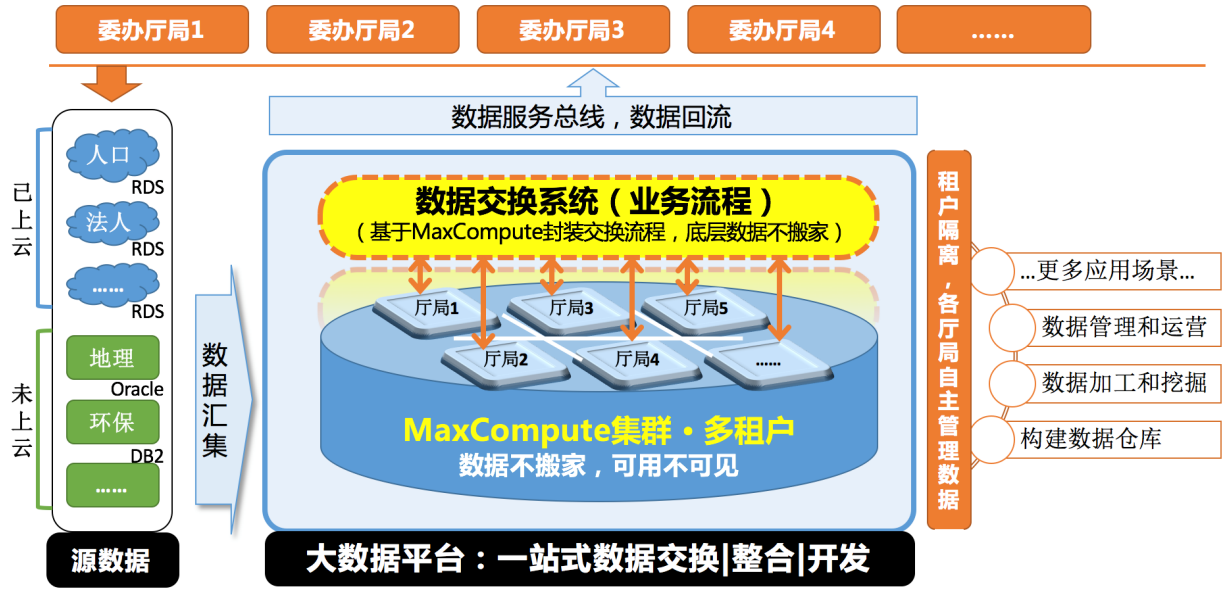
金融的数据仓库主要是基于MaxCompute SQL完成离线计算，并通过一系列指标规则和算法完成离线决策，输出结果给在线决策使用。

跟传统数据库相比，基于MaxCompute搭建数仓，有以下不同之处：

- **历史数据存储**：MaxCompute天生支持大数据，不必像传统数据库那样将历史数据转储到廉价存储媒介。
- **分区方式**：传统数据库支持的分区方式更丰富一些（例如支持范围分区），但在数仓场景下，MaxCompute目前支持的分区方式基本够用。不管用哪个数据库搭建数仓，表分区的设计理念和原则一致。
- **大宽表**：MaxCompute按字段存储，建大宽表有天然优势。
- **数据整合**：传统数据库都用存储过程来加工、整合数据，MaxCompute则需要将逻辑拆分成一段段SQL，虽然实现途径不同，但算法是一致的。经过几年的使用比较，采用分段SQL的方式更清晰和高效，而存储过程的方式更灵活且具备处理复杂逻辑的能力。

3.6.2 大数据共享及交换

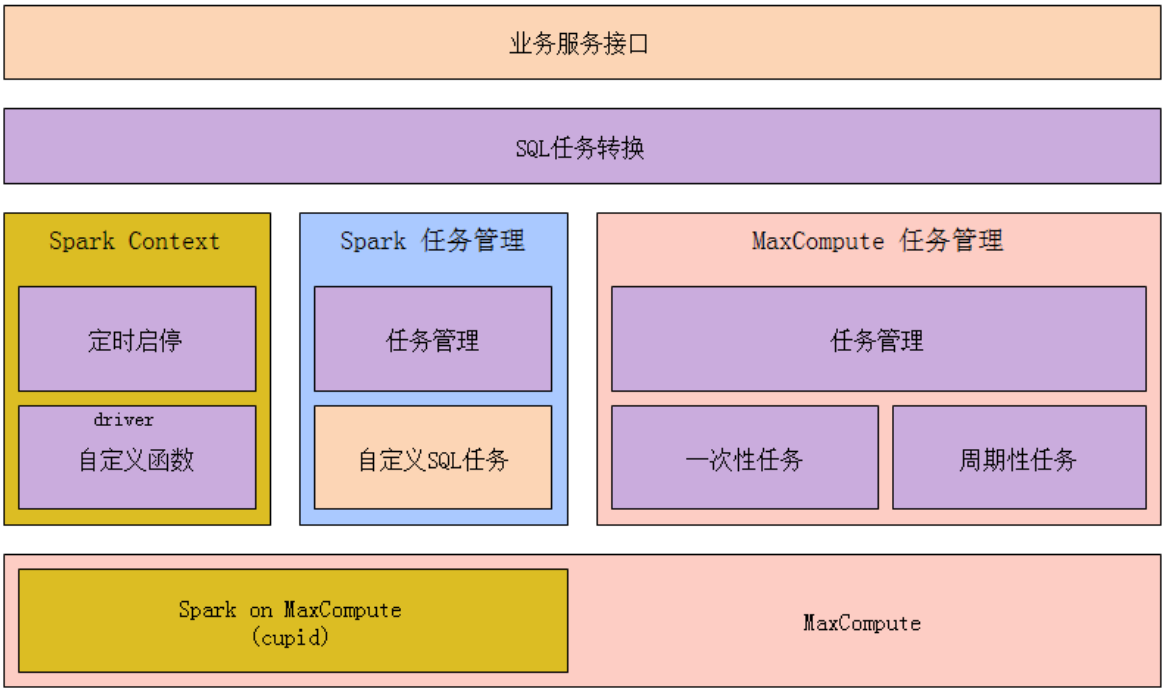
图 3-6: 大数据共享及交换



MaxCompute具有丰富的多种权限管理方式、灵活数据访问控制策略。MaxCompute提供丰富的授权管理手段，包括ACL、角色授权、 Policy授权、跨Project授权以及Label机制，可以提供精确到列级别的安全方案，满足一个组织或者跨组织间的授权需求。安全要求较高的项目，可以提供项目保护机制，防止数据泄露，而且对用户的任何操作都记录日志便于事后追溯审计。

3.6.3 Spark on MaxCompute典型实践

图 3-7: 典型实践



Spark on MaxCompute支持Client Mode的业务计算平台应用，应用框架如上图所示。

4 DataWorks

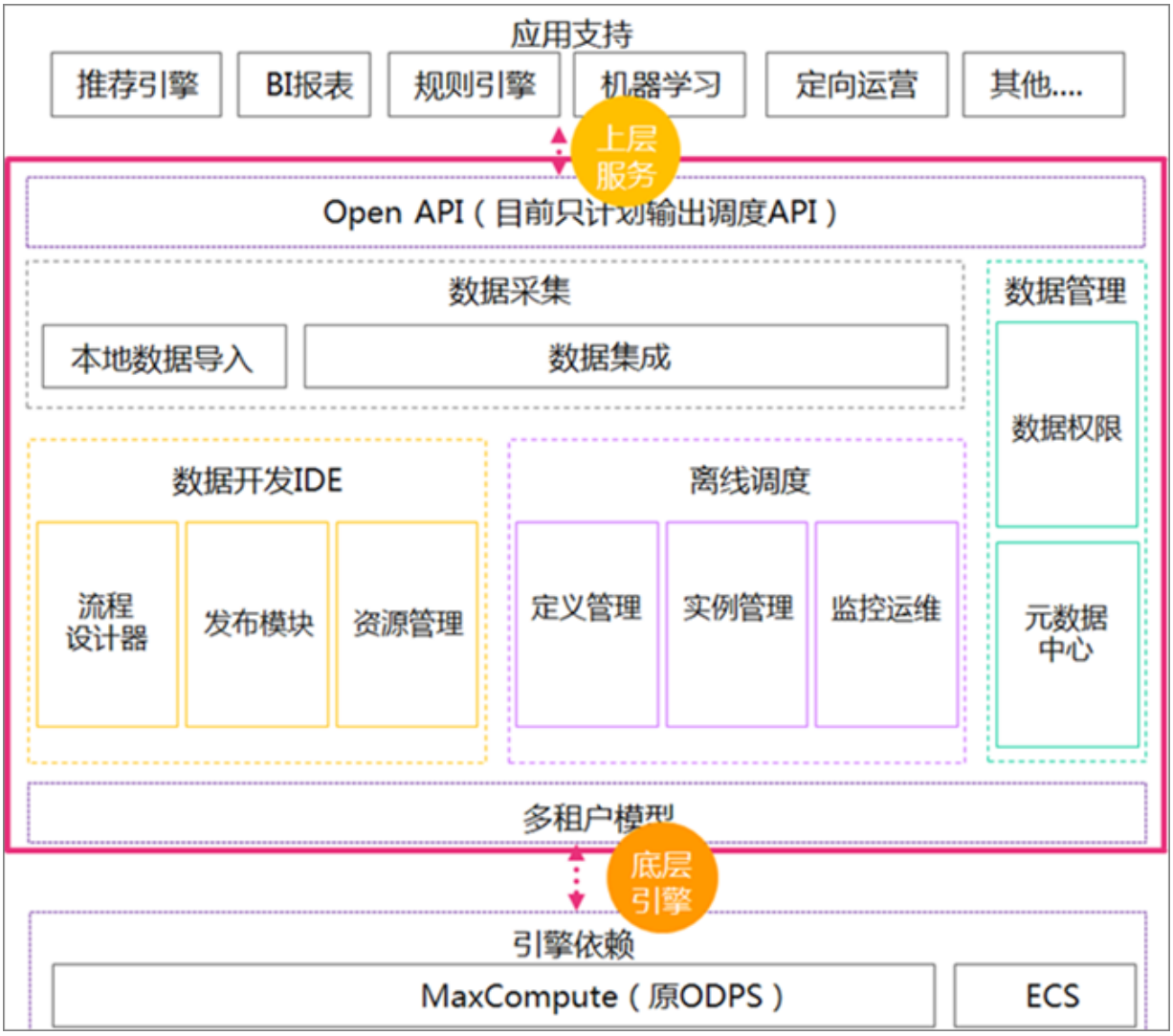
4.1 产品概述

DataWorks是阿里云推出的一款大数据领域平台级产品。面向企业和个人用户提供端到端的一站式大数据开发、管理、离线调度和应用解决方案。

DataWorks以让更多人使用更多数据为使命，提供DT时代的大数据基础服务。

- 让大型企业构建PB、EB级别的数据仓库，实现超大规模数据集成，对数据进行资产化管理，通过对数据价值的深度挖掘实现业务的数据化运营。
- 让中小企业和个人用户快速构建数据应用，助力中小企业的数据业务创新。

图 4-1: 产品组成



DataWorks产品由数据开发IDE、离线调度系统、数据集成工具和数据管理四大部分组成。

- **数据开发IDE**：提供开箱即用的一站式数据开发工具，满足在线SQL、MR、Shell的编码工作，并提供多人协同开发和代码版本管理功能。通过可视化的流程设计工具可以满足快速构建数仓调度的依赖关系。
- **离线调度系统**：提供百万级的离线任务调度能力，以及在线运维、在线日志查询、调度状态监控报警等一系列功能。
- **数据集成工具**：提供海量异构数据源的数据集成能力，打通阿里云80%的数据库及存储设备的数据链路，以及常用关系型数据库、FTP、HDFS等多种数据链路，并且提供周期性定时集成的能力。
- **数据管理系统**，提供对MaxCompute（原ODPS）中以公司为单位的全量数据的管理能力，并提供权限管理、数据血缘和元数据查看等功能。

4.2 产品特性和核心优势

超大规模数据处理能力

DataWorks与阿里云大数据服务MaxCompute（原ODPS）天然集成，单个集群的规模可达5000台，并且具备跨机房的线性扩展能力，轻松处理海量数据。离线调度支持百万级任务量，实时监控告警。

核心指标：

- 万亿级数据JOIN，百万级并发job，作业I/O可达PB级/天。
- 具备跨集群（机房）数据共享能力，支持万级别的集群数，扩容不受限制。
- 提供功能强大易用的SQL、MR引擎，兼容大部分标准SQL语法。
- MaxCompute（原ODPS）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来保护用户的数据，使其不丢失、不泄露、不被窃取。

一站式的数据开发环境

数据开发、离线调度、调度运维、监控告警、数据管理全流程串通。

核心指标：

- 一个产品，提供全流程所有功能。
- 提供可视化工作流程设计器功能，类似Kettle的工具，支持用户对流程进行设计并编辑。
- 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。

海量异构数据源快速集成能力

提供11种异构数据源的数据读取能力，12种异构数据源的数据写入能力。并且提供脏数据过滤，流量控制等功能。

核心指标：

- 提供mysql、oracle、sqlserver、postgresql、rds、drds、MaxCompute、ftp、oss、hdfs、dm、sysbase的数据读取能力。
- 提供mysql、oracle、sqlserver、postgresql、rds、drds、MaxCompute、ads、ocs、oss、hdfs、dm、sysbase的数据写入能力。
- 提供脏数据过滤、流量控制能功能。
- 可以周期性调度，周期性数据集成能力。

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。

多租户权限模型

多租户模型确保用户的数据被安全隔离，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。

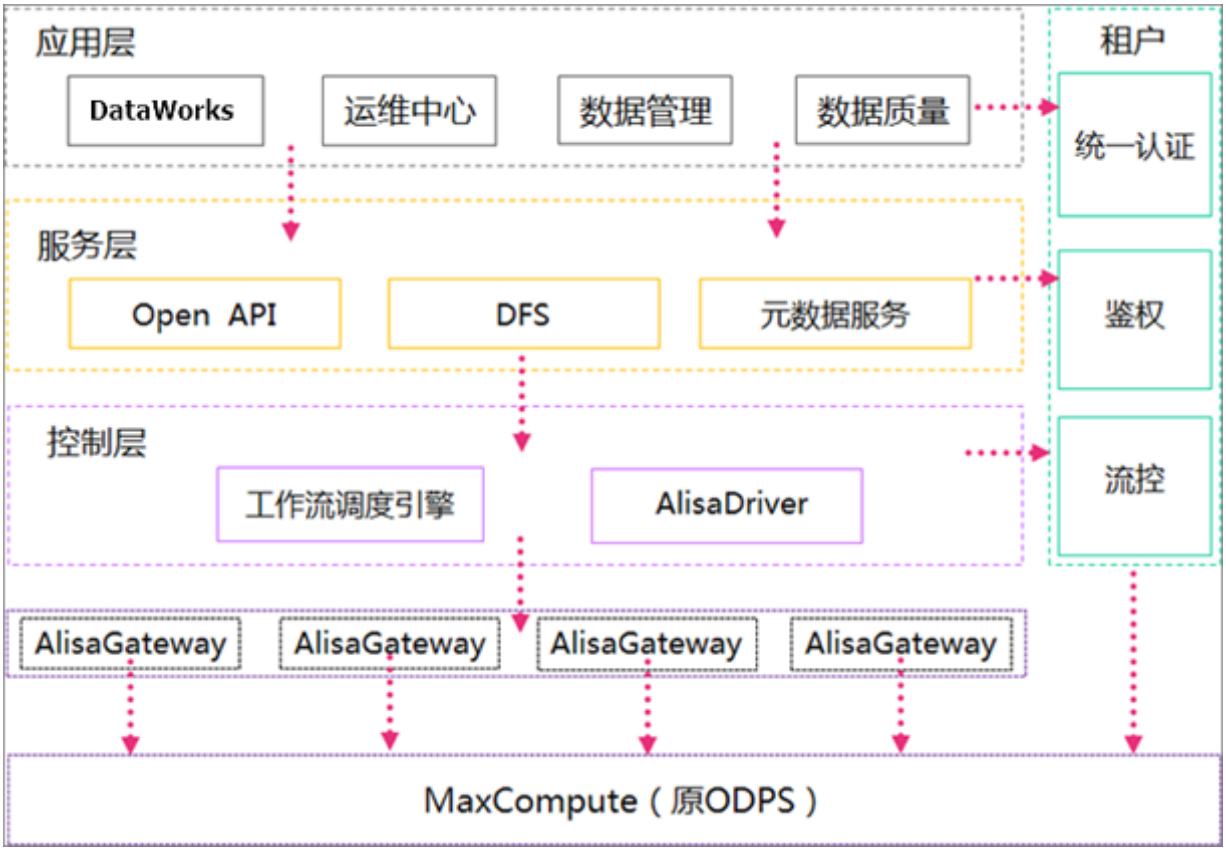
开放的平台

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。

4.3 系统架构

系统开放性架构

图 4-2: 系统架构图



DataWorks采用组件化、服务化设计理念，分为三层：

- **控制层**：DataWorks离线加工的核心，工作流调度引擎承接着整个DataWorks的调度，包括工作流的转实例、工作流调度，AlisaDriver主要协调、控制所有任务的执行。
- **服务层**：为应用层或外部其他应用提供服务。
- **应用层**：基于底层服务直接和用户进行交互，为用户提供可视化的操作的界面。

安全架构

DataWorks的安全架构，是由平台自身的安全实现层、平台内置的安全服务层、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产品自身的安全性。
- **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，比如租户资源隔离、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。

多租户模型

DataWorks拥有自己的多租户权限模型：

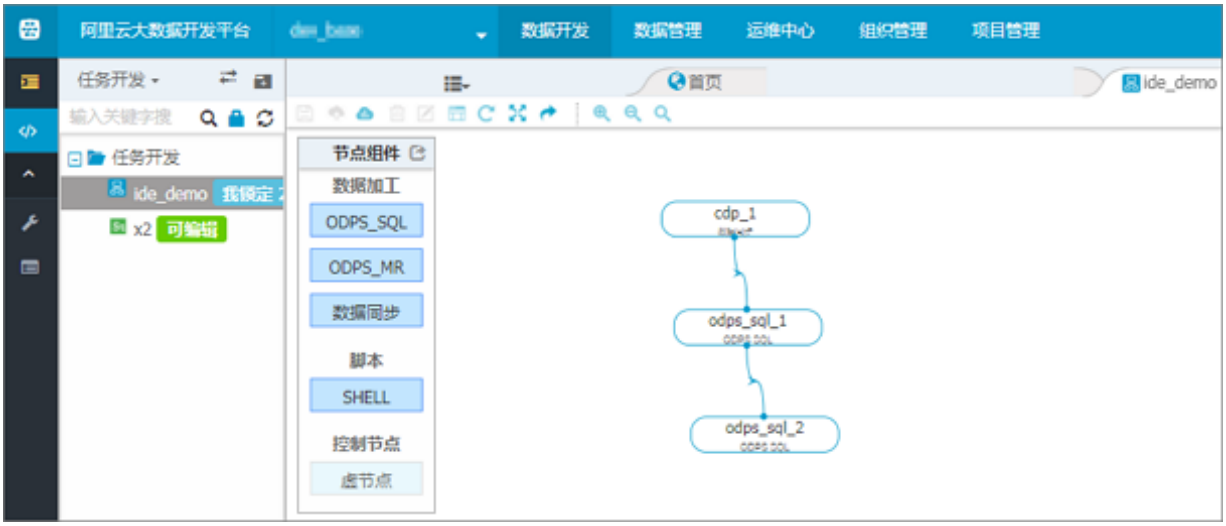
- 弹性的存储和计算资源：租户可按需申请资源配额，独立管理自己的资源。
- 租户独立管理自有的数据、权限、用户、角色，彼此隔离，以确保数据安全。

4.4 功能描述

4.4.1 数据开发IDE

DataWorks的数据开发IDE模块，提供一站式的集成开发环境，可满足大数据环境下的快速数仓建模、数据查询、ETL开发、算法开发等需求，并提供多人在线协同开发、文件版本控制等功能。

图 4-3: 数据开发



功能特性：

- 提供可视化工作流程设计器功能，类似Kettle的工具，支持用户对流程进行设计并编辑，对流程中的每一个任务节点进行相应的开发工作。
- 提供本地数据上传功能，支持本地文本数据快速上云。
- 提供海量异构数据源的数据快速集成能力。

> **说明**：
>
> 目前数据同步任务支持的**数据源类型**包括：

- 取数据支持的数据源：mysql、oracle、sqlserver、postgresql、rds、drds、maxcompute、ftp、oss、hdfs、dm、sysbase。
- 写数据支持的数据源：mysql、oracle、sqlserver、postgresql、rds、drds、maxcompute、ads、ocs、oss、hdfs、dm、sysbase。

- 提供Web IDE编程和调试环境，支持多种程序类型： SQL、MR、SHELL（有限支持）、数据同步等。
- 跨项目发布：快速将任务及代码部署到其他项目的调度系统。
- 协同开发：代码版本管理，多人协同模式下的代码锁管理和冲突检测机制。
- 提供MaxCompute（原ODPS）表搜索、资源搜索引用、自定义函数搜索引用、数据查询功能，您可轻松索引数据。

4.4.2 数据管理

数据管理为您提供租户范围内数据表搜索、数据表详情查看、数据表权限管理、收藏数据表等功能。详细操作请参见《*DataWorks*用户指南中的数据管理》。

4.4.3 调度系统

离线调度系统为您提供百万量级任务的离线调度服务，并提供可视化运维界面、在线日志查询、监控告警等功能。详细操作请参见《*DataWorks*用户指南》。

功能特性：

- 调度系统可支撑的job数量达到百万级。
- 执行框架采用分布式架构，并发作业数可线性扩展。
- 支持多时间粒度的调度周期：分钟、小时、日、周、月、年。
- 支持节点空跑、暂停、一次性运行等特殊状态控制。
- 可视化展示调度任务DAG图，极大地方便用户对线上任务进行运维管理。
- 支持实时任务运行状态监控告警功能，短信、邮件的告警方式。
- 支持单任务重跑、多任务重跑、结束进程、置成功、暂停等线上运维操作功能。
- 支持补数据（串行执行多周期实例）。
- 提供全局的任务统计信息汇总界面，任务统计内容包括：总调度任务数、出错调度任务数、运行调度任务数、计算资源消耗Top10调度任务、计算时间消耗Top10调度任务、任务类型分布等信息。

4.4.4 数据集成

提供多种异构数据源的快速集成服务，为跨平台的异构数据提供快速数据整合的能力。

功能特性：

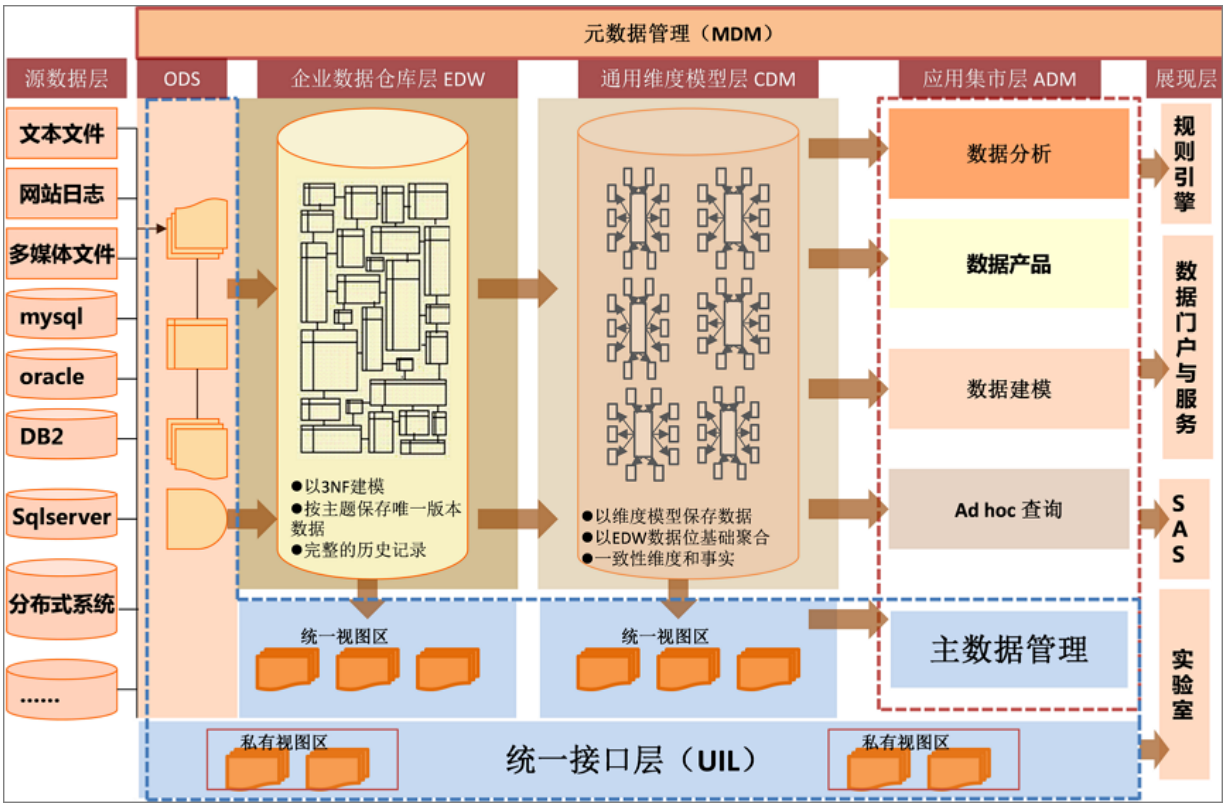
- 支持以多种数据通道（并且持续增长中）
  - 取数据支持的数据源：mysql、oracle、sqlserver、postgresql、rds、drds、MaxCompute、ftp、oss、hdfs、dm、sysbase。
  - 写数据支持的数据源：mysql、oracle、sqlserver、postgresql、rds、drds、MaxCompute、ads、ocs、oss、hdfs、dm、sysbase。
- 可靠的数据质量
  - 完美支持各种数据类型的转换，保证不失真。
  - 能精确识别脏数据，进行过滤、采集、展示，为用户提供可靠的脏数据处理，让用户准确把控数据质量。
  - 提供作业全链路的流量、数据量、脏数据探测和运行时汇报。
- 强劲的传输速度
  - 极致优化的单通道插件性能，单进程一定能够打满单机网卡（200MB/s）。
  - 全新的分布式模型，吞吐量无限水平扩展，我们能够提供GB级、乃至于TB级数据流量。
- 友好的控制体验
  - 精确且强大的流控保证，支持通道、记录流、字节流三种流控模式。
  - 完备且健全的容错处理，能够做到线程级别、进程级别、作业级别多层次局部/全局的重试。
- 清晰的内核设计
  - 专家级的框架设计经验，执行引擎更加强大，内核可以仅仅修改配置即可完成升级。
  - 更加清晰易用的插件接口，让插件开发人员专注于业务开发，而不再关注框架细节。

4.5 应用场景

4.5.1 大型数据仓库搭建

大型企业可在私有云环境下使用DataWorks来构建超大型的数据仓库。

图 4-4: 构建数据仓库



DataWorks为这类客户提供卓越的海量数据集成能力。

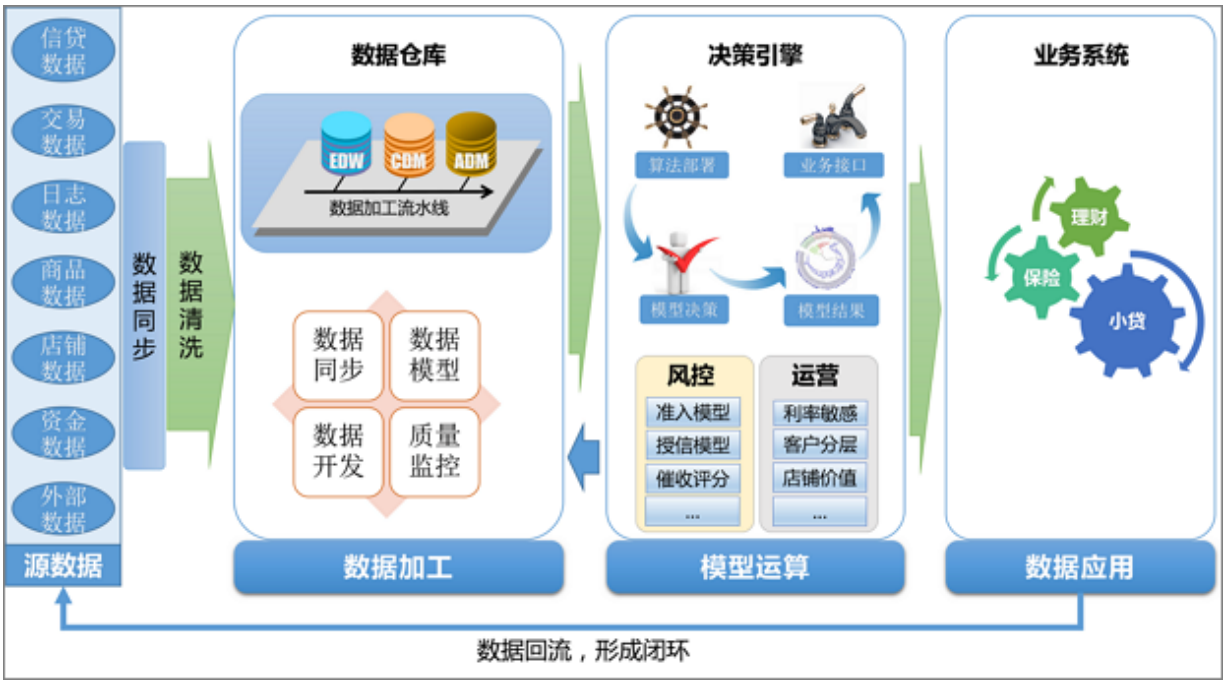
- 海量存储：可支持PB、EB级别的数据仓库，存储规模可线性扩展。
- 数据集成：支持多种异构数据源的数据同步和整合，消除数据孤岛。
- 数据开发：基于MaxCompute（原ODPS）的大数据开发，支持SQL、MR等编程框架，以及贴近业务场景的白屏化工作流设计器。
- 数据管理：基于统一的元数据服务来提供数据资源管理视图，以及数据权限审批流程。
- 离线调度：可以提供多时间维度的周期性调度能力，支持每天百万级的调度并发，并对任务调度实时监控，对错误及时告警。

4.5.2 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平台快速使用和分析数据，助力企业的经营决策。

以下展示阿里小贷的数据业务运营模式：

图 4-5: 数据化运营


信贷数据
交易数据
日志数据
商品数据
店铺数据
资金数据
外部数据
源数据
数据同步
数据清洗
数据仓库
EDW
CDM
ADM
数据加工流水线
数据同步
数据模型
数据开发
质量监控
数据加工
决策引擎
算法部署
业务接口
模型决策
模型结果
风控
准入模型
授信模型
催收评分
...
运营
利率敏感
客户分层
店铺价值
...
模型运算
业务系统
理财
保险
小贷
数据应用
数据回流，形成闭环

5 分析型数据库AnalyticDB

5.1 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。

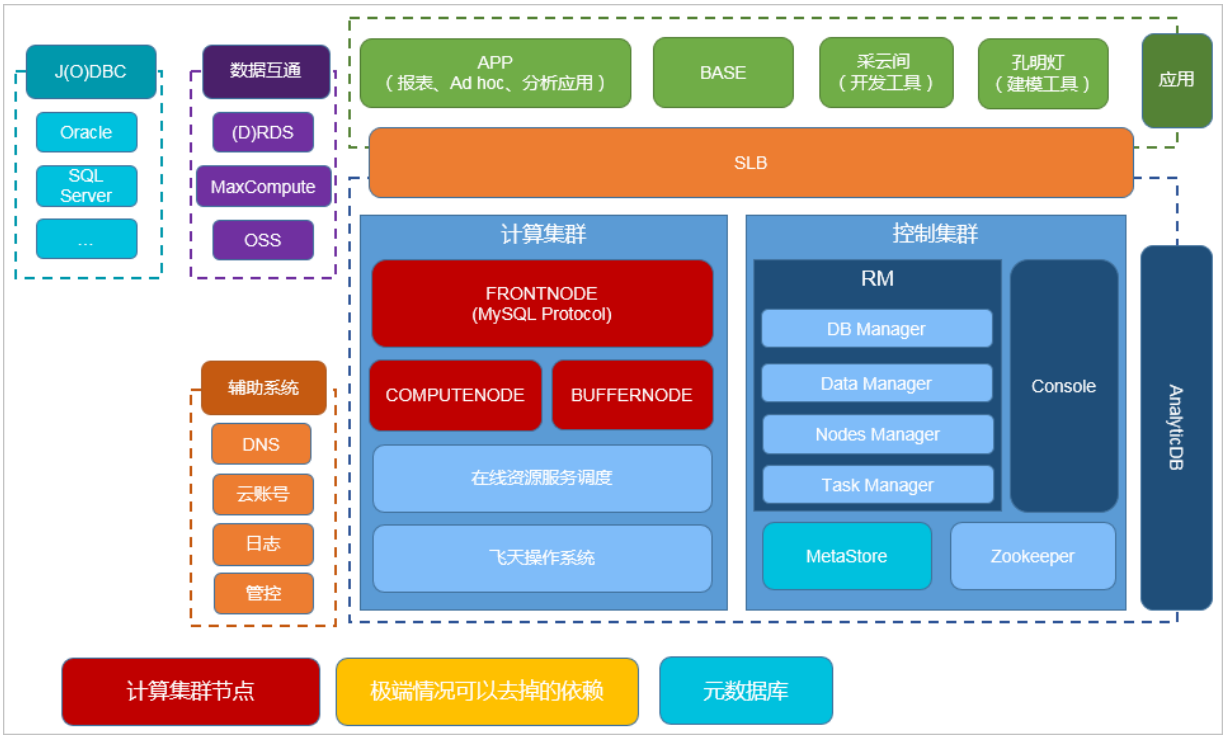
AnalyticDB是一套RT-OLAP系统，具有以下特点：

- 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。

5.2 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

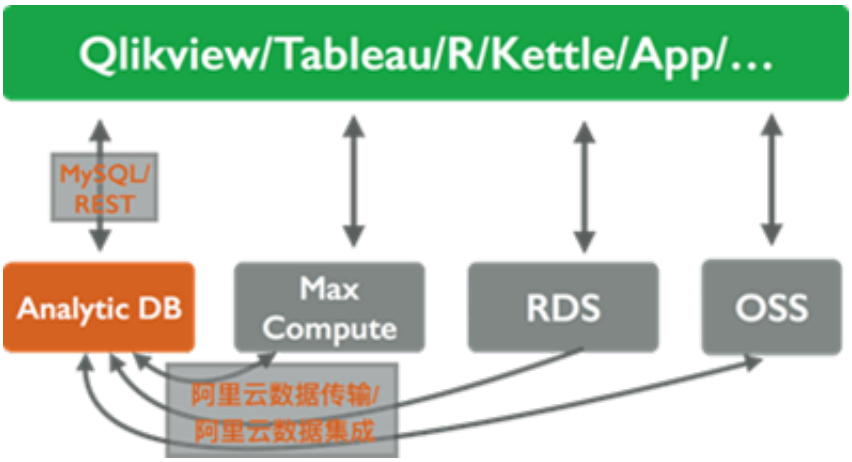
控制集群（即资源管理器RM）用于控制计算集群中数据库资源分配、数据库内数据和计算资源的分布、飞天集群上的计算节点管理、数据库后台运行的任务管理等。控制集群实际上由多个模块组成，一个控制集群可以同时管理部署于不同机房的多套计算集群。

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。

5.3 功能特性

5.3.1 实体

5.3.1.1 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。

5.3.1.2 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。

5.3.1.3 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。

5.3.1.4 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。

5.3.1.5 维度表

- 支持和任意表组的任意表以任意列进行Join加速。
- 最大可支持千万级的数据条数。
- 无需指定分区方式。

5.3.1.6 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。

5.3.1.7 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。

5.3.2 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。

5.3.3 DML

5.3.3.1 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。

5.3.3.2 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。

5.3.4 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。

5.3.5 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。

5.3.6 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。

5.3.7 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。

5.3.8 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。

5.3.9 元数据

5.3.9.1 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。

5.3.9.2 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。

5.3.9.3 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。

5.3.10 管理控制台

5.3.10.1 用户控制台（DMS for AnalyticDB）

- 支持阿里云账号登录与鉴权。
- 提供图形化的数据库创建与管理、表组/表的创建与管理功能。
- 提供友好的SQL查询调试功能，并提供图形化的执行计划展示。
- 提供友好的用户与权限查看、管理功能。
- 提供友好的数据导入导出、导入导出状态查询和导入导出进度查询界面。
- 提供两分钟内实时系统性能报告的展示以及最近七天内小时粒度详细的离线性能报告展示。
- 提供数据库资源可视化。
- 提供扩容、缩容、查看扩容/缩容状态和历史的功能。

5.3.10.2 运维管理控制台（大数据管家）

- 用于系统后台运维人员管理系统资源、监控系统运行状态和修改系统参数。
- 提供图形化的界面展示各个数据库的存储计算资源，以及在物理节点上的占用、分布。
- 提供图形化的查看和管理系统参数功能。
- 提供图形化的查看和管理数据导入全链路状态功能。
- 提供图形化的分布式日志提取和查看工具。

5.3.11 特色功能

5.3.11.1 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。

5.3.11.2 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。

5.3.11.3 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。

5.3.11.4 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。

5.4 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。

优势	描述
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 大数据应用加速器DTBoost

6.1 前言

随着近五年互联网和大数据技术的蓬勃发展，各类数据产品应运而生。从阿里大数据的应用发展来看，阿里依然面临很多挑战。

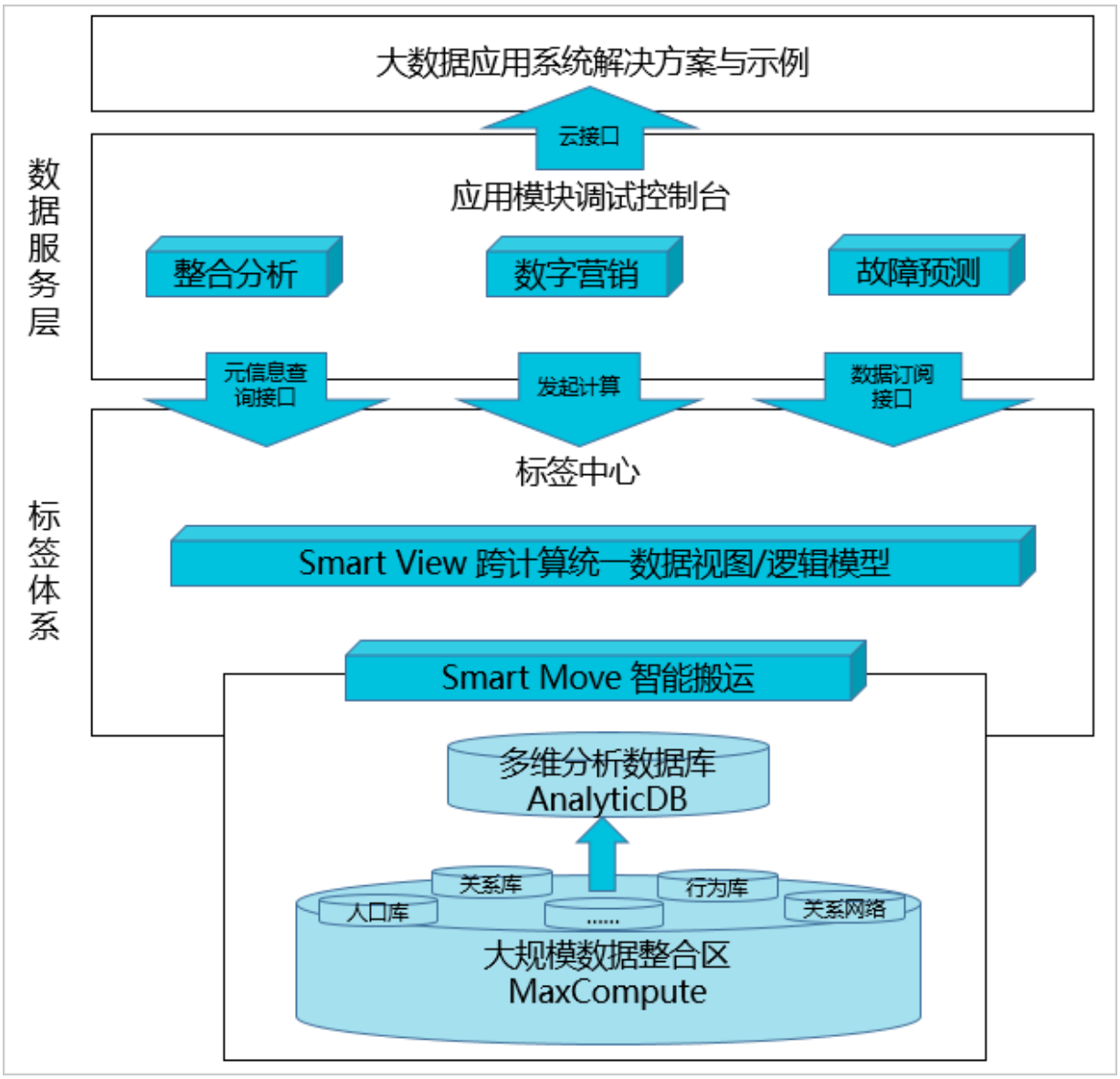
- 为应对数据量高速的增长，衍生出多样的分布式数据计算与存储技术，以解决各类应用场景下的难题，而非传统IT架构当中只需要单一数据库即可支撑整个企业的数据分析报表问题。如何对各类数据的积累进行有效的整合与管理，各个业务库的数据之间如何打通在多个计算存储资源上合理的分布管理也成为一大难题。
- 大数据在各个行业当中的应用，如数字广告、互联网金融、电子商务、在线风控等场景当中，一个数据应用需要囊括报表分析、行为预测、实时监控、信用评分、个性化推荐、文本挖掘、时空数据等各类大数据技术方法的综合运用，而不仅仅是做企业经营的报表统计。
- 当下对运用数据的用户不再局限于专业的数据分析师、数据仓库工程师，更多的是能够让非技术背景的业务人员能够以他能够理解的方式灵活的探查数据。
- 若想运用好大数据，就需要对企业的IT架构、技术人员的综合能力提出更高的要求。
  - 需要了解各个专业分布式计算和存储资源的特性。
  - 需要深入了解业务人员使用数据的场景，并制作出面向业务的数据产品。
  - 需要将计算和存储资源针对数据分析、算法服务等多种应用场景进行合理的架构。

阿里云DTBoost数据加速器产品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的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能够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。

6.2 产品概述

DTBoost产品组件如图 *6-1: DTBoost产品组件*所示。

图 6-1: DTBoost产品组件



概括来讲，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、规则引擎、营销引擎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。

这种方式的好处如下所示：

- 应用开发人员不必对下层多个计算存储资源进行深入理解，减少对复杂系统的对接工作。
- 通过数据服务的形式透出，有助于IT部门对数据使用的管理，以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，开发者只需把要使用的数据在模型当中进行管理后，即可通过API进行相应的计算，然后对接到产品界面，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快速搭建相应的大数据产品。

整个产品系列包括以下几个模块。

- **标签中心**
  - 云计算资源管理
  - 模型探索
  - 模型管理
- **画像分析**
  - 分析服务
  - 界面配置
- **规则引擎**
  - 服务层
  - 控制层
  - 执行层

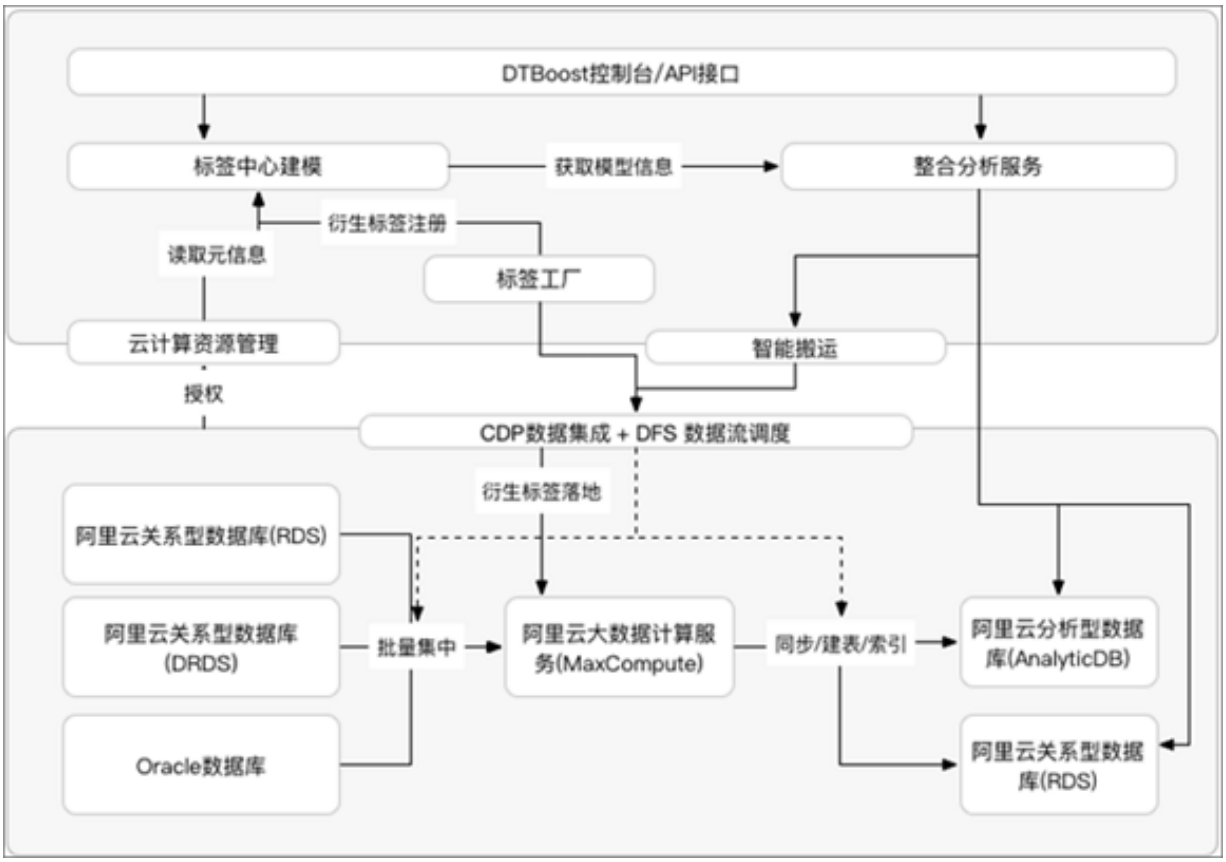
6.3 产品架构

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，再同步至在线数据库当中进行在线分析查询。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。

下文将以画像分析为例，为您介绍DTBoost的总体架构。

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）计算服务中进行集中，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。

图 6-2: 技术架构图



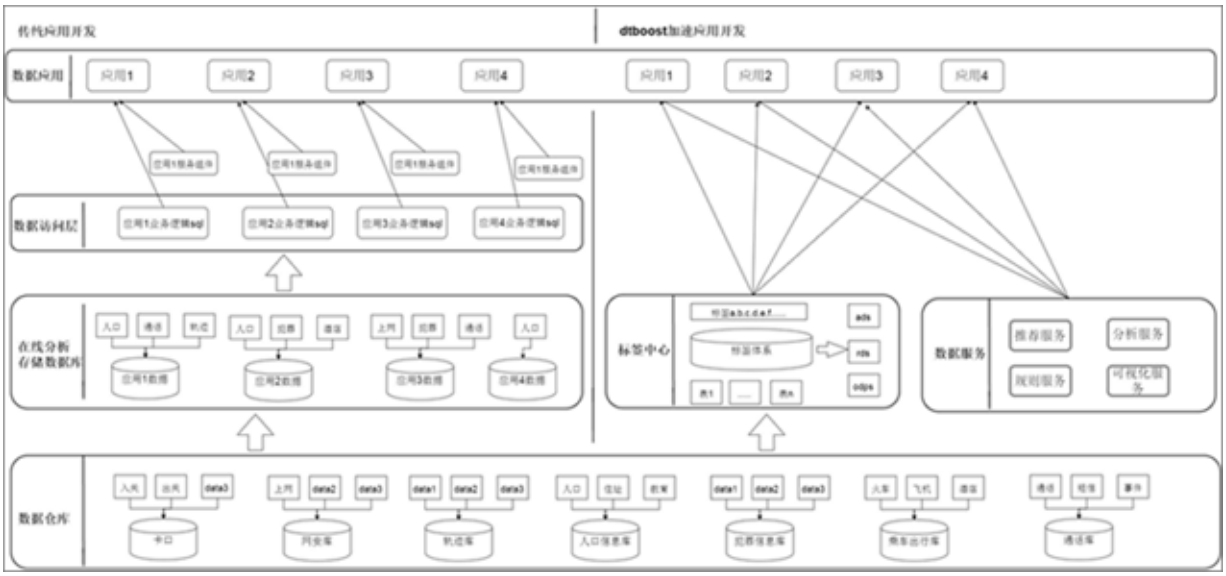
您从DTBoost控制台或API进入，通过把自己的云计算资源授权给DTBoost后，即可通过DTBoost读取各个云计算资源中的数据元信息。经过建模配置后，在相关的数据服务模块中可以进行手工/自动触发标签中心的智能搬运模块，通过把相关的数据同步调度任务发送给数据流服务DFS（Data Flow Service）和数据集成，来对所需要整合的数据以标签粒度来进行业务库到离线数据仓库的批量集中，以及到在线分析数据库的同步、建表、索引工作。在数据准备完成之后，就可以通过相关的数据服务API接口或者在控制台上基于标签模型视图之上进行相关的计算。对于当中需要离线计算加工的部分，一些常用的加工可以通过标签工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。

整个过程可以看做DTBoost在大数据平台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。

6.4 应用场景

您除了通过控制台对各个模块进行配置操作以外，各个模块从数据元信息到数据服务的操作处理都可以透过开放API整合入自己的应用系统当中。这种服务化的方式一方面提高了系统整合的便利性，另一方面也对企业数据应用管理上提供了便利。

图 6-3: 加速开发流程



如图 *6-3:* 加速开发流程所示，从企业IT架构上来看，IT或者数据部门可以通过DTBoost以数据服务化的方式把计算资源、数据资源、数据计算方法打包在一起，提供给业务部门开发、外部合作伙伴。一方面对应用开发者来说即开通即使用，方便快捷。另一方面从IT部门来说，对于平台的资源管控更加有效，一定程度上降低了数据的冗余存储与加工，特别针对于业务算法、消费者画像这些需要使用到明细数据计算的场景，既能够使用到明细数据，又不会影响到原始数据的生产，不造成大数据量的冗余拷贝，还能够降低数据使用的门槛，提供了有力的支撑。

6.5 功能模块

6.5.1 标签中心

6.5.1.1 概念说明

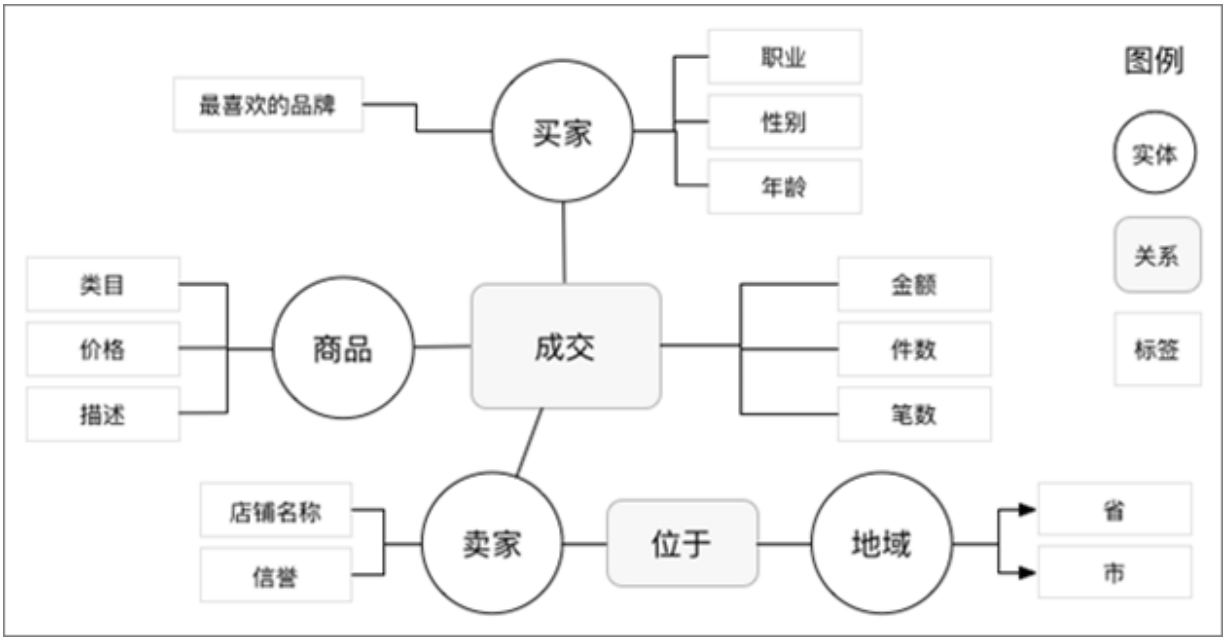
标签中心的作用是在现有的数据表之上构建跨计算存储的逻辑模型，直接让您在视图层上对数据进行管理、加工、查询，屏蔽下层的多个大数据计算存储资源，简化数据的使用。当整个数据架构越复杂，越是需要多个计算存储资源组合使用的场景下，标签中心的价值就越为明显。

标签建模的方法来源于阿里巴巴用户画像体系，广泛应用于精准营销、个性化推荐、用户画像、信用评分等需要基于明细数据进行计算的大数据应用当中。所谓标签就是对用户这一对象的一个最小描述单元，代表着所描述对象某一个具体的客观事实的抽象表达，如属性（性别，标签值男、女；年龄，标签值实际年龄），行为（成交金额、收藏次数、位置定位），或者是兴趣（对于多个关键词的偏好度），是一种以业务视角出发的数据建模方法，标签既可能是数值、也可能是枚举值，也可以是多个Key-Value组织的列，还可能是多字段组成的事实表（如对象、时间、谓

语、宾语）。从概念模型上讲，标签体系是指围绕多个实体对象（如买家、卖家、商品、企业、设备），以及实体之间的关系（如成交、检修、位于等），建立标签化描述的方法。

标签建模如图 *6-4:* 标签建模所示。

图 6-4: 标签建模



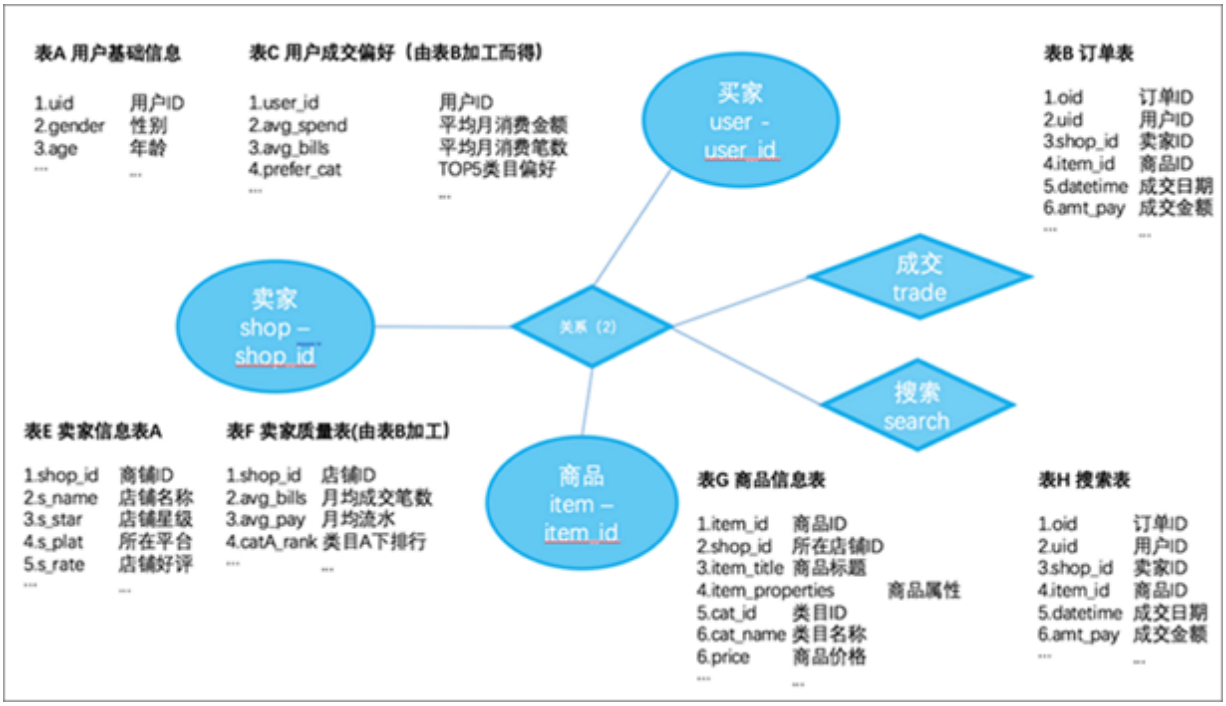
这种建模方式看起来可能类似于角模型（Anchor）或者是图模型（Graph），其实并不一样。传统的建模过程是根据业务需求设计概念和逻辑模型，再根据逻辑模型对物理数据表进行加工和规整。而标签建模是在已有的物理数据/模型之上直接建立逻辑模型，通过各个数据服务的代理解析，让用户可以在视图上直接进行各类的计算，不需要预先对物理数据进行大规模的加工处理，即用即算。

但需要明确的是，总体来说，标签仍然是建立在物化数据之上，因为在跨计算的语境之下可能会面临多个计算的查询语言和性能的差异，建立在逻辑请求上的标签很可能会无法执行，所以总体来讲定义的每一个标签还是需要对应到落地的物理表上。但在DTBoost当中，可以在相应的数据服务当中以某一个计算查询逻辑定义为一个临时标签使用，但关系到跨计算之时还是需要将之物化，避免错误发生的可能。

标签模型是围绕实体（Entity）、关系（Link）、标签（Tag）三大元素对分布在不同数据库中的数据进行网络化的建模方式。实体用于描述某个客观的对象，如设备-人员-地址等，对应到物理数据表上一般就是属性表，有一个主键来代表每一个对象，剩下的每一列就是标签即描述对象的属性。那么关系是表示对象和对象之间的联系、事件、行为，一般对应到物理数据表上一般就是事实流水表，如成交-检修-乘车等。

实体关系建模如图 6-5: 实体关系建模所示。

图 6-5: 实体关系建模



相比于指标-维度体系，这种建模方式更适用于对于明细数据描述和表达。明细数据很大一部分都是事实表，引入关系的概念对应到流水事实表上，把多个实体之间的关系很好的呈现表达，既有利于管理也方便分析时的表达，在对业务端呈现上也更接近于概念模型的设计一样可被一般人理解。

在经过建模转化之后，可以将上表中的模型逻辑关系转化为图 6-6: 实体关系管理所示。成交表对应到关系结点上，金额和时间是关系上的标签，用户表和商品表对应到买家和商品两个实体上，性别、年龄是买家的标签。这种建模方式非常便于各类基于明细行为、关系数据进行分析的场景。

图 6-6: 实体关系管理

您可以在标签中心页面下看到标签中心的几大功能，包括**云计算资源、实体关系标签、模型探索**和**标签同步**。

6.5.1.2 适用场景

标签中心是跨计算存储、可在物理模型之上逻辑动态建模、与数据服务结合面向大数据应用开发的数据建模、数据管理工具，并能够通过可视化的方法清晰的展现企业的数据模型视图。

标签中心适用于以下场景。

- **数据模型探索管理**

  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，结合数据服务能够方便地进行业务逻辑计算操作。

- **数据权限管理**

  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。

6.5.1.3 功能组件

6.5.1.3.1 云计算资源管理

云计算资源管理能够支持与多个计算存储资源通信，并可获取元信息。

目前DTBoost支持对以下计算存储资源进行管理。

- Oracle数据库
- 阿里云关系型数据库（RDS）
- 阿里云大数据计算（MaxCompute）
- 阿里云分析型数据库（AnalyticDB）
- 阿里云表格存储（Table Store）
- 阿里云流式计算（StreamCompute）

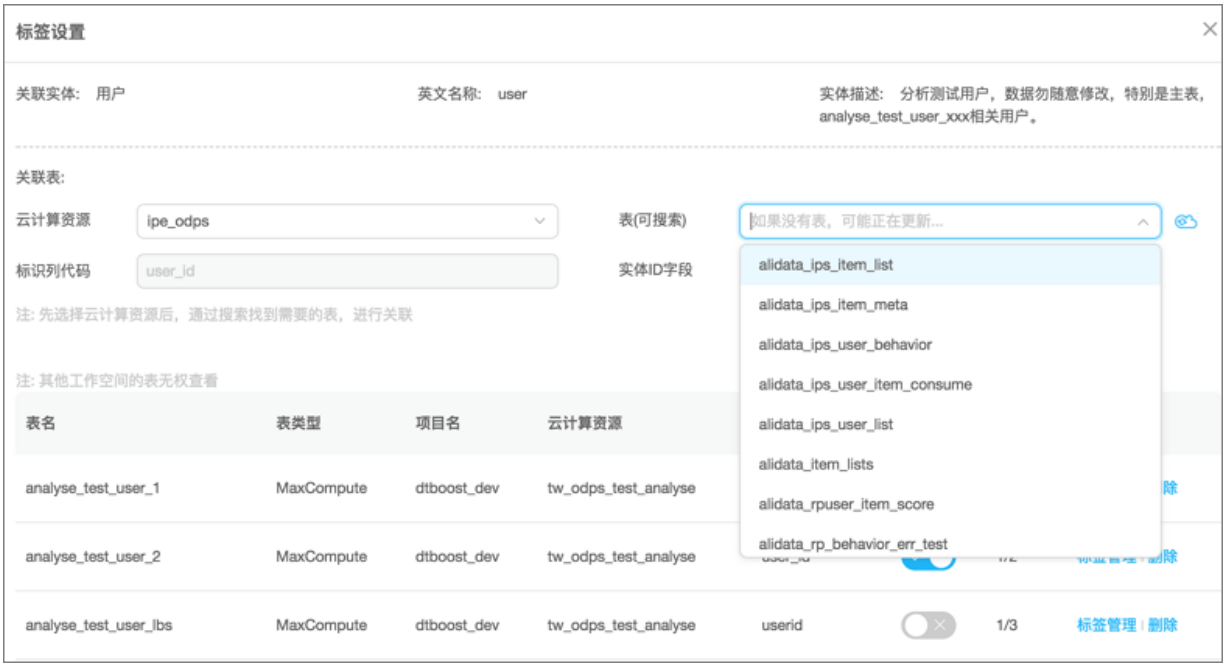
6.5.1.3.2 模型管理

实体关系管理

实体/关系管理是标签中心当中对逻辑模型进行配置的主要功能，能够读取不同来源的数据库的元信息，整合为实体或者关系。

描述同一个实体（主键）的多张表可以在逻辑层上聚合在一个实体下，形成一张**大宽表**，如图 *6-7: 实体关系建模示意*所示。

图 6-7: 实体关系建模示意



关系的建立则是可以把联合主键表看作为关系，将多个实体关联起来。其余的描述字段则根据相应的情况定义为标签，如图 *6-8: 关系定义*所示。

图 6-8: 关系定义

标签管理

标签管理模块能够对所有的标签进行查看、检索和修改，如图 *6-9:* 标签管理和图 *6-10:* 设置标签所示。

图 6-9: 标签管理

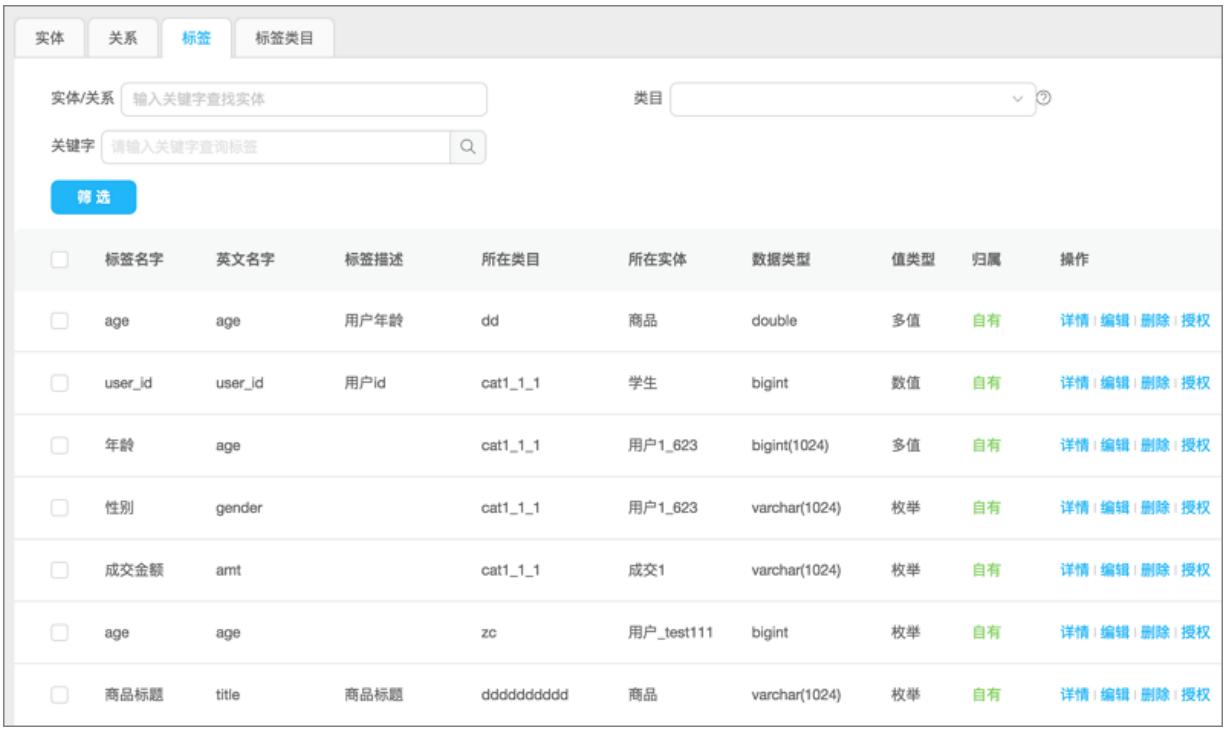


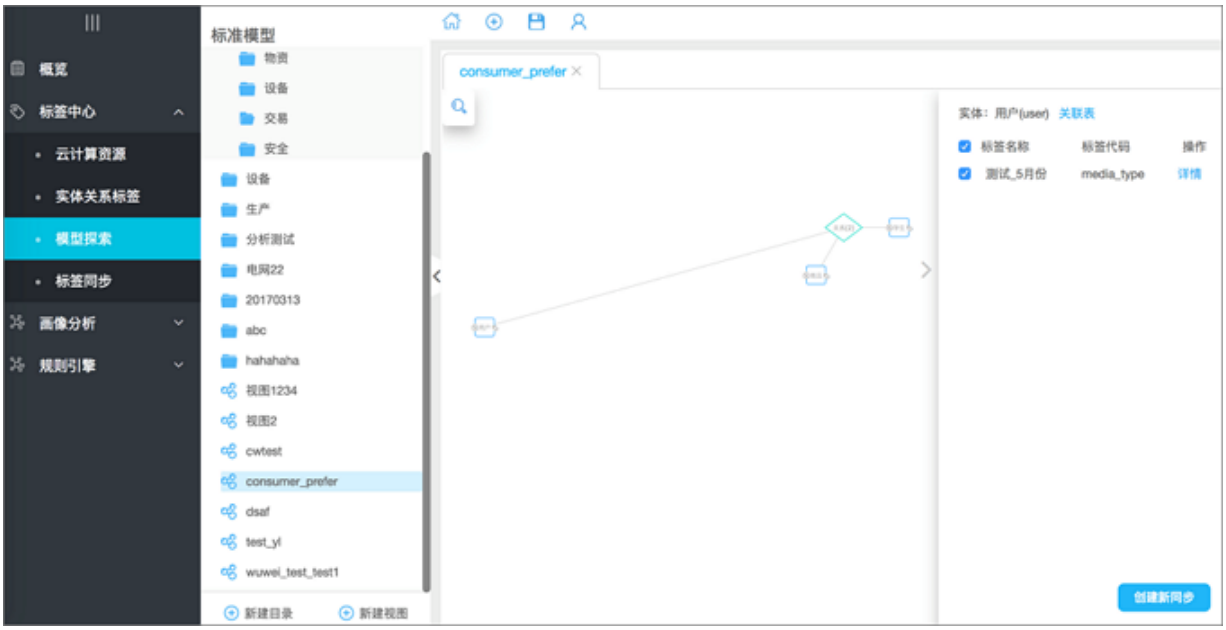
图 6-10: 设置标签

6.5.1.3.3 模型探索与数字订阅

模型探索部分可以通过关系图的方式查看所有的实体，实体与实体之间的联通关系及其属性，以及实体/关系下关联的标签情况。

如图 *6-11:* 实体关系模型探索所示，通过模型探索可以对整个标签模型进行全局的分析查看。

图 6-11: 实体关系模型探索



标签数据订阅是DTBoost处理跨计算数据流转的重要功能之一。在相应的数据服务需要使用到数据的时候，标签中心提供了将分散在多个存储当中的数据订阅至数据服务需要计算的位置的功能。对于同步且相应时间要求高的场景来说，需要用户在相应的数据服务当中进行提前的手工订阅操作，对于异步或者请求相应要求不高的同步的计算场景来说，这个订阅过程对于用户来说透明。

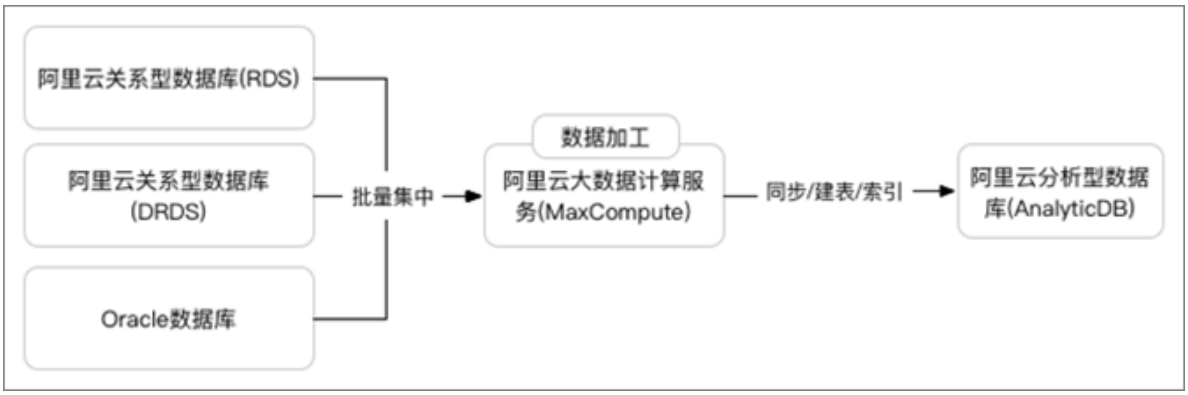
标签中心的技术架构如图 *6-12:* 标签中心技术架构所示。

图 6-12: 标签中心技术架构

智能搬运内置了针对几套典型的架构路径。

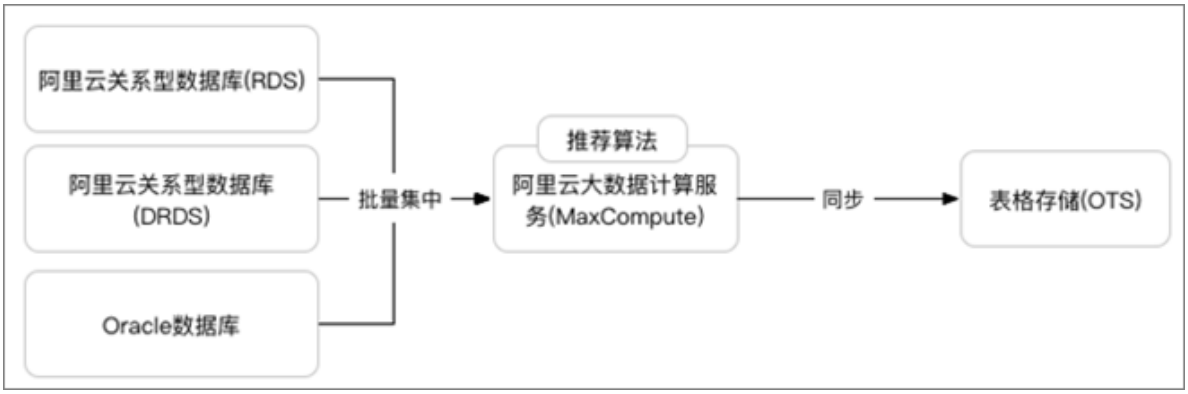
- 批量大数据在线分析，如图 *6-13:* 批量大数据在线分析所示。

图 6-13: 批量大数据在线分析



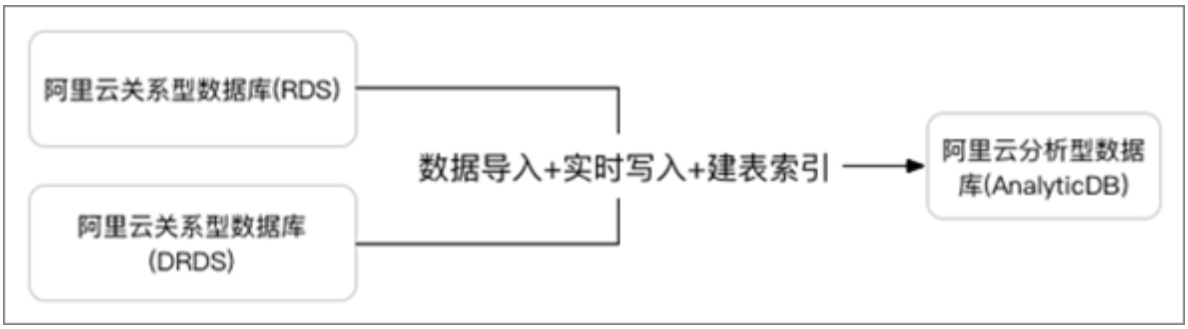
- 批量数据算法计算在线查询，如图 *6-14:* 批量数据算法计算在线查询所示。

图 6-14: 批量数据算法计算在线查询



- 实时大数据在线分析，如图 *6-15:* 实时大数据在线分析所示。

图 6-15: 实时大数据在线分析



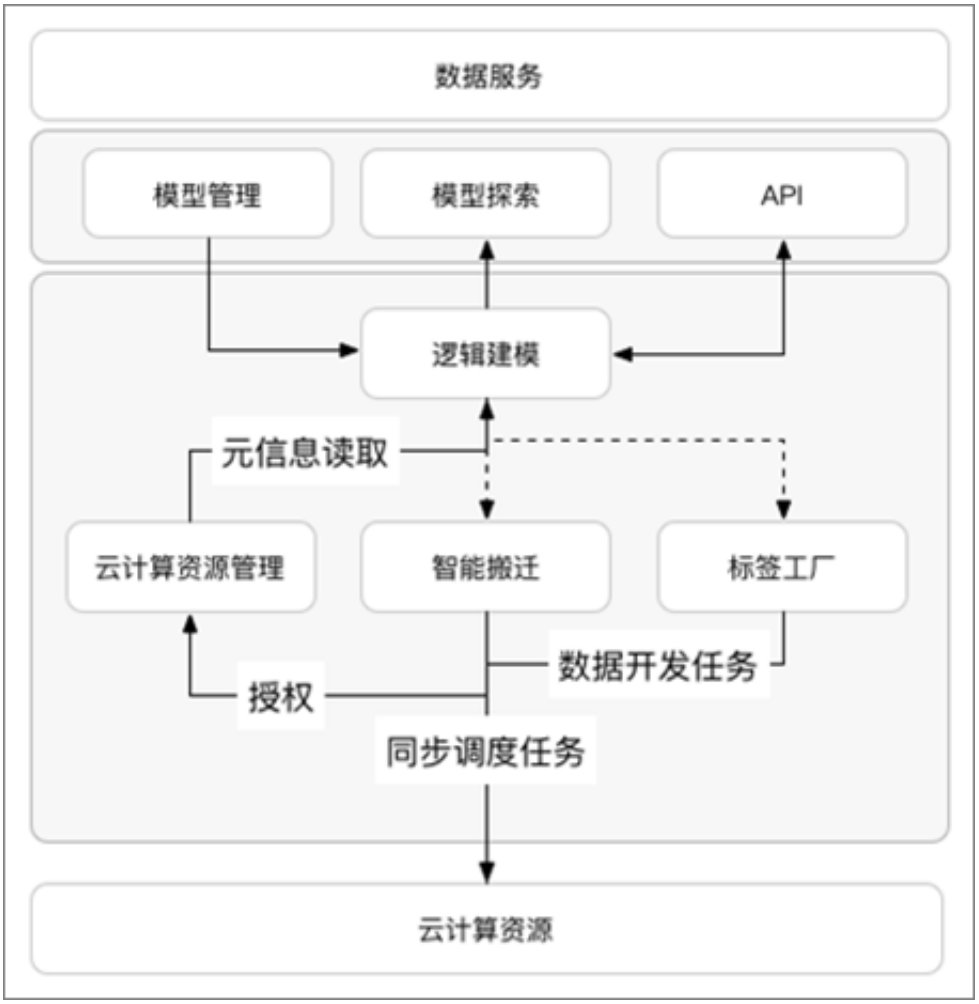
  - 对于整合分析这类 OLAP/ADHOC 场景来说，提供了将 Oracle，关系型数据库（MySQL）等业务库中的数据同步至大数据计算（MaxCompute）中，再订阅到所使用的分析库中，如阿里云分析型数据库（AnalyticDB）、关系型数据库（RDS）等。

- 对于规则引擎这类流式计算的场景来说，提供了将离线数据、流式数据进行归并，将规则所需要的离线历史数据订阅至阿里云表格存储当中，并根据规则计算结果订阅至所需要的存储计算资源（MySQL/MaxCompute/AnalyticDB等）中。
- 对于目前尚未以标准方式提供的订阅路径，可以进行相应的定制。

6.5.1.4 技术架构

技术架构如图 *6-16:* 标签中心组件架构所示。

图 6-16: 标签中心组件架构



6.5.1.5 产品特性

对于这种数据体系的规划上来说，往往是由业务驱动的，是累积增加的，随着不同的业务板块的开展会逐渐纳入更多的数据源。如果按照传统数据仓库的做法会面临以下问题。

- 需要不断地在物理层进行数据表的归并，下层表的频繁变化可能会造成数据使用的不稳定。

- 当标签的需求越来越多，因为不可能无限制的在物理层将数据拼在一张宽表当中，那分散的数据表也会越来越多，会造成检索和管理的困难。
- 在不同的应用当中不可能是整表使用，往往是需要多张表中的某几列，那多个应用不断的抽取再整合也会造成管理和检索的困难。
- 标签可能是实时数据、也有可能是离线数据，数据存储方式不同同样造成管理使用的困难。

标签体系建模和传统的BI分析建模相比，具有以下特性。

- **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。

  所以标签体系是把多个计算当中的拷贝在逻辑视图上进行唯一映射，即一个标签对应到多个计算当中的物理字段。

- **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。

从另一方面来说，计算能力的增强和数据使用场景的丰富，更多的数据计算是需要直接作用在明细的行为数据上，而非只是对指标的多维统计。传统数据集市建模的**指标>维度**体系就略显狭窄。标签的定义上涵盖了多种数值类型，既可以是单列，也可以是维度 + 标签组成的复合标签（这种方式通常用于描述某种行为），赋予应用操作上更大的灵活度。

6.5.2 画像分析

6.5.2.1 适用场景

数据服务是架设在标签视图之上的业务功能模块，可以通过界面化的配置或者API的操作能够以标签为粒度对跨计算资源的数据进行统一的业务计算操作。透过数据服务+逻辑建模的组合，既节省

工作量又有很好的扩展性。特别是对于大数据环境需要整合多个系统数据的前提下，很难一次把所有数据需求全部规划完整，那么这种动态逻辑建模的方式就有非常好的扩展性。

从应用的角度来说，由标签模型视图层隔开明细数据复杂的数据结构，能够在相对扁平的标签体系之上进行明细数据上计算和查询，对于数据开发与应用开发的分工流程上来说也更为合理。

图 6-17: 画像分析

画像分析作为DTBoost数据服务的其中之一，其所适用的场景主要是结合阿里云分析型数据库（Analytics DataBase），将您分布在多个存储资源的数据整合起来，在标签模型上构建大数据画像类的交互式分析应用，让您的业务人员可以自由灵活的分析这些对象各种属性与行为之间的关联性。可以广泛应用于工业设备画像分析、企业经营画像分析、用户行为画像分析等多个场景当中。

大数据画像类分析应用有以下特性。

- **基于行为等明细数据的分析**

  在过去以各项KPI指标计算为主要分析目的背景下，很容易把所有的指标计算提前构建。随着数据采集和使用场景的丰富，业务人员希望能够自由地分析各类行为明细数据，如查看不同客户属性在各个商品类目下消费的偏好和关联购买的情况，或者不同时间采购的不同类型、属性、地域设备的故障率与检修情况，还能够把多个维度细分下的具体客户/设备清单进行查看。业务人员进行的分析可能是任意维度之间的交叉关系，就很难进行预先的计算。

- **从半结构化数据中抽取特征**

从另一点来说，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。往往很多的画像特征如抽象的兴趣，如喜欢动漫、爱美一族等风格兴趣偏好类的特征，通常需要通过算法从用户的点击、收藏、购买行为与相关物品的文本描述当中进行特征抽取。这就需要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**

  业务人员希望得到的分析是在数据当中探索有用的信息，如发现影响消费者购买的可能因素，或者故障设备的关联因素，这就需要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。这就对查询速度的高响应提出了要求。

  在这种交互式的分析场景下也对整体界面的组织提出了要求，业务人员关心的是在不断探索中获得的数据洞察，如果还需要用户进行复杂的报表配置或者是数据结构/技术上的学习理解，就会大大影响数据探索发现的过程。各种数据的分析还需要与各种类型的可视化形态结合，除了常规的图表外，可能还需要各种尺度特别是城市内尺度的地图图表，表达拓扑关系的关系网络图表，以及能展示文本特征的图表。

从以上几点来看，交互式数据分析产品的开发变得非常有挑战性。

- 需要充分理解数据结构，才能把跨表查询的逻辑与界面交互进行有机的组织。
- 需要了解多个专业存储与计算资源的特性，把不同计算产出的结果组织到同一个分析界面中。
- 需要熟悉各类的可视化图表与分析控件的使用方法，结合到不同类型的分析中。

6.5.2.2 功能组件

画像分析提供了两大块的功能，分析服务层部分用户可以在控制台中完成。

6.5.2.2.1 接口调试

分析服务接口模块可以让您在此进行分析语句的调试和自助化封装数据分析接口。画像分析的查询表达都是建构在实体关系模型之上的。

您也可以对接口进行调试，查看查询的结果/执行错误、语法每一步解析所耗费的时间以及所解析的真实SQL语句，来帮助您调试分析接口。

6.5.2.2.2 界面配置

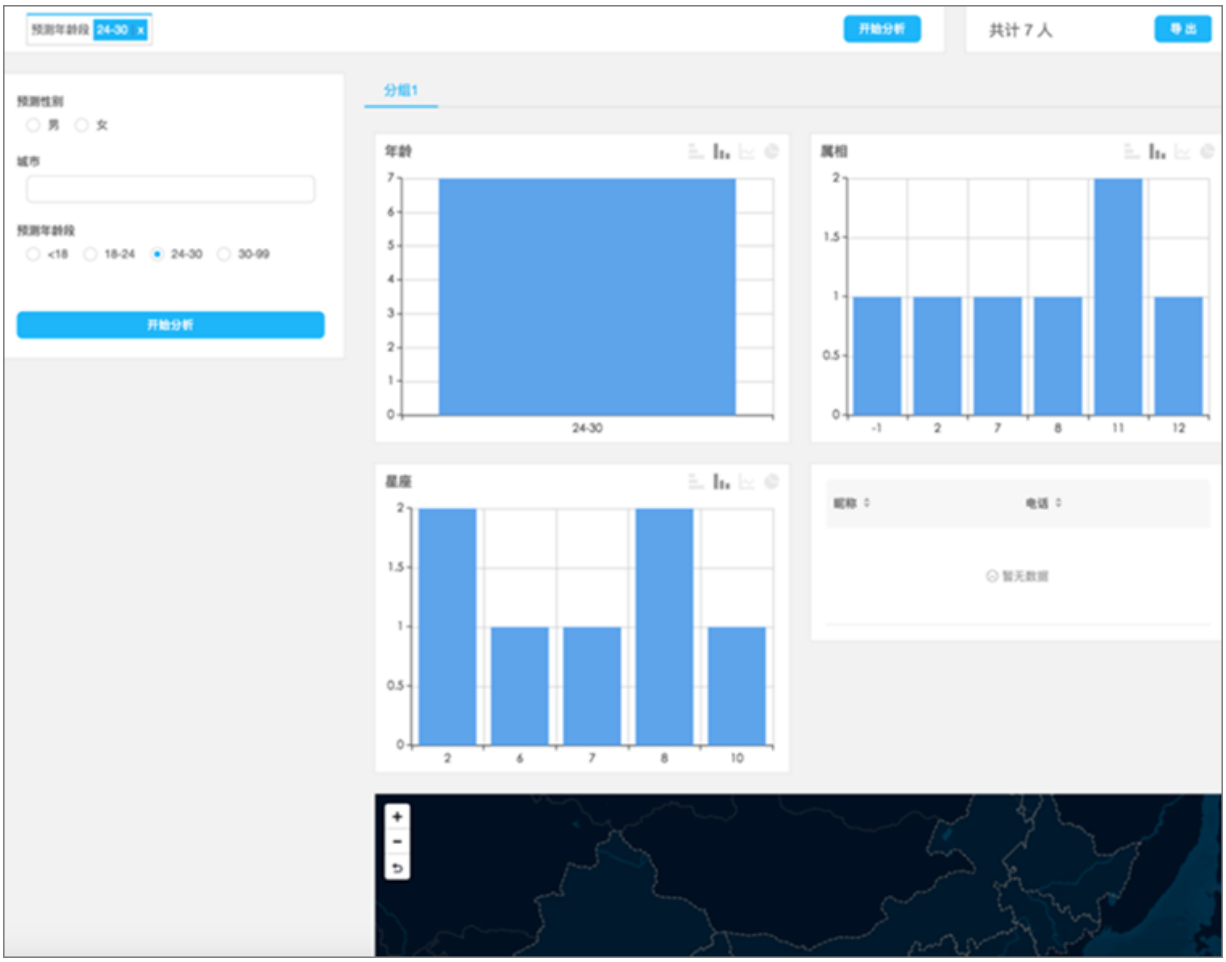
如图 *6-18:* 筛选条件配置所示，通过画像分析界面搭建工具，灵活配置交互式画像分析界面。对筛选出来的特定的分析对象进行多维透视，并进一步钻取分析，并可以将分析筛选出来的对象导出到

其他系统当中，如结合广告投放系统进行精准营销。整个界面的代码是完全开放的，可以无缝与您的现有系统进行整合。

图 6-18: 筛选条件配置

如图 *6-19:* 分析应用所示，上层提供的交互分析应用框架是以源代码的方式提供，您只需把整个应用运行，会根据配置文件的填写自动渲染出一个可进行交互式分析的界面。您可以进行代码的修改进行样式的改造。多个配置文件可以通过配置从不同的URL进行路由，让不同的用户可以看到不同的分析应用。

图 6-19: 分析应用



6.5.2.3 典型应用

6.5.2.3.1 用户全景画像

在受众分析、CRM、用户行为、人口分析等场景下，通常需要对这些人群的明细行为数据进行分析。

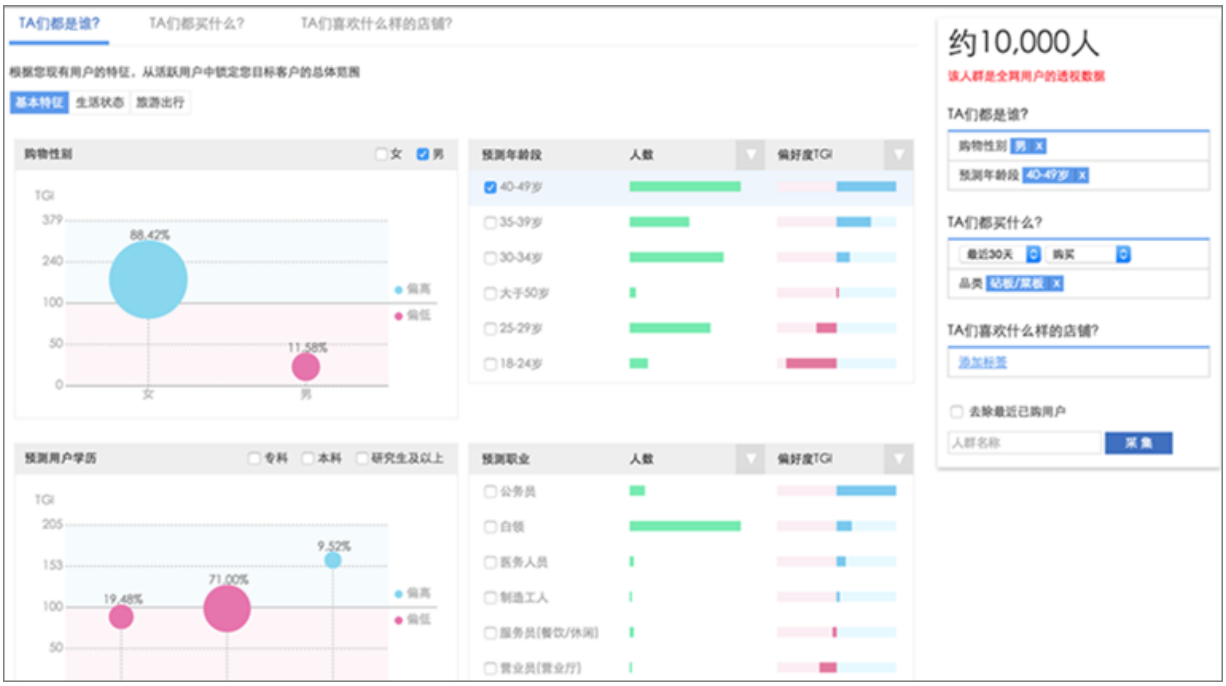
分析人员在进行分析时，不同的用户属性和用户行为之间会形成多种组合，所以无法按传统数据分析的建模方法提前预算好所有组合。

例如用户根据消费者行为选取想要分析的目标群体，如图 *6-20:* 选择目标群体所示。

图 6-20: 选择目标群体

对于选取的目标群体，分析其各项特征，发现分布密集的特征，如图 *6-21: 分析各项特征*所示。

图 6-21: 分析各项特征

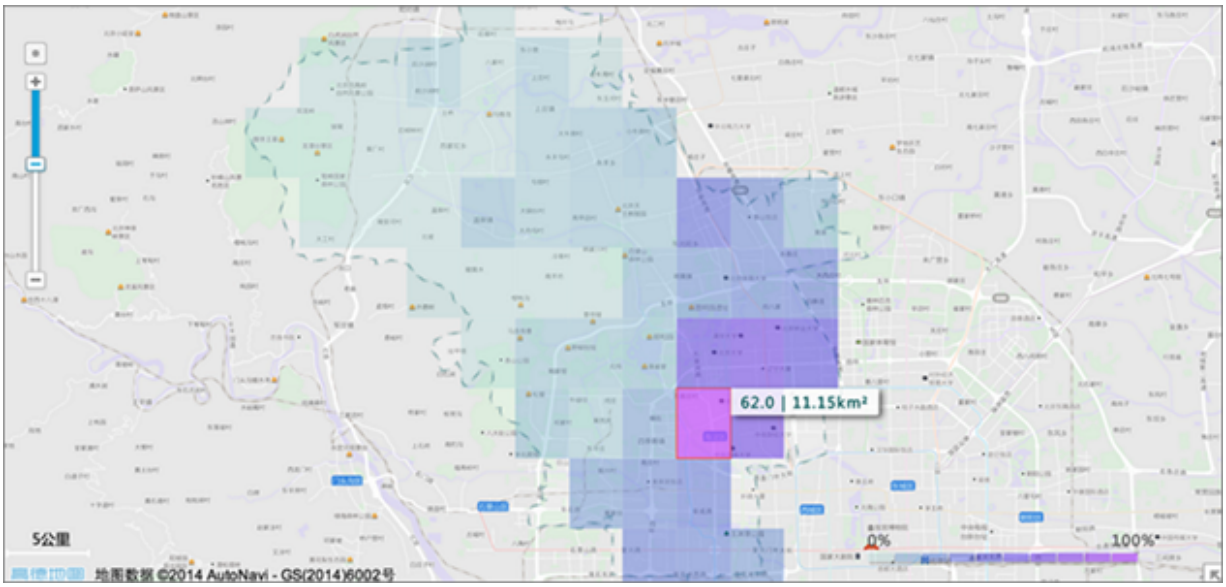


再根据这些典型特征，找到还未购买的群体，如图 *6-22: 找到未购买的群体*所示。

图 6-22: 找到未购买的群体

针对这类群体，对接下游系统，如图 *6-23:* 对接下游系统所示。

图 6-23: 对接下游系统



6.5.2.3.2 设备全履历

例如，设备全履历如图 *6-24:* 电压等级、图 *6-25:* 地级供电公司和图 *6-26:* 设备明细所示。

图 6-24: 电压等级

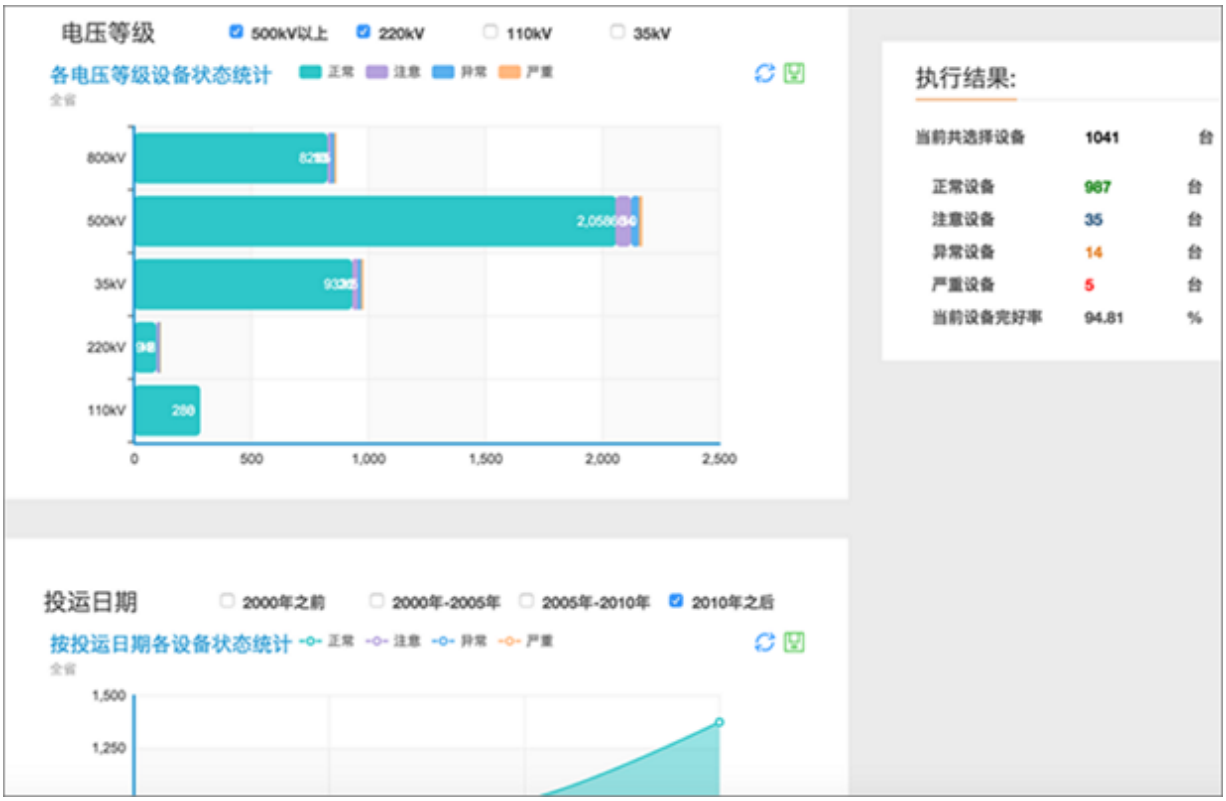


图 6-25: 地级供电公司

图 6-26: 设备明细

序号	设备编号	电压等级	生产厂家	设备名称	投运年份	所属公司	设备型号	所属区县	设备类型	设备状态	消缺日期
1	GLKG10605	220kV	厂商E	隔离开关10605	2000/03/19	杭州	10605	临安市供电公司	隔离开关	严重	-
2	GLKG10641	220kV	厂商A	隔离开关10641	2001/02/17	杭州	10641	建德市供电公司	隔离开关	严重	-
3	GLKG11603	110kV	厂商C	隔离开关11603	2016/04/14	湖州	11603	德清县供电公司	隔离开关	严重	-
4	GLKG11604	110kV	厂商D	隔离开关11604	2010/06/15	湖州	11604	湖州市本市局	隔离开关	严重	-
5	GLKG12601	35kV	厂商A	隔离开关12601	2007/03/18	嘉兴	12601	桐乡市供电公司	隔离开关	严重	-
6	GLKG13602	35kV	厂商B	隔离开关13602	2004/02/17	金华	13602	东阳市供电公司	隔离开关	严重	-
7	GLKG14609	35kV	厂商D	隔离开关14609	2008/06/15	丽水	14609	庆元县供电公司	隔离开关	严重	-
8	GLKG14610	35kV	厂商E	隔离开关14610	1996/06/18	丽水	14610	龙泉市供电公司	隔离开关	严重	-
9	GLKG14627	35kV	厂商B	隔离开关14627	2015/06/14	丽水	14627	景宁县供电公司	隔离开关	严重	-
10	GLKG14628	800kV	厂商C	隔离开关14628	2009/06/15	丽水	14628	松阳县供电公司	隔离开关	严重	-

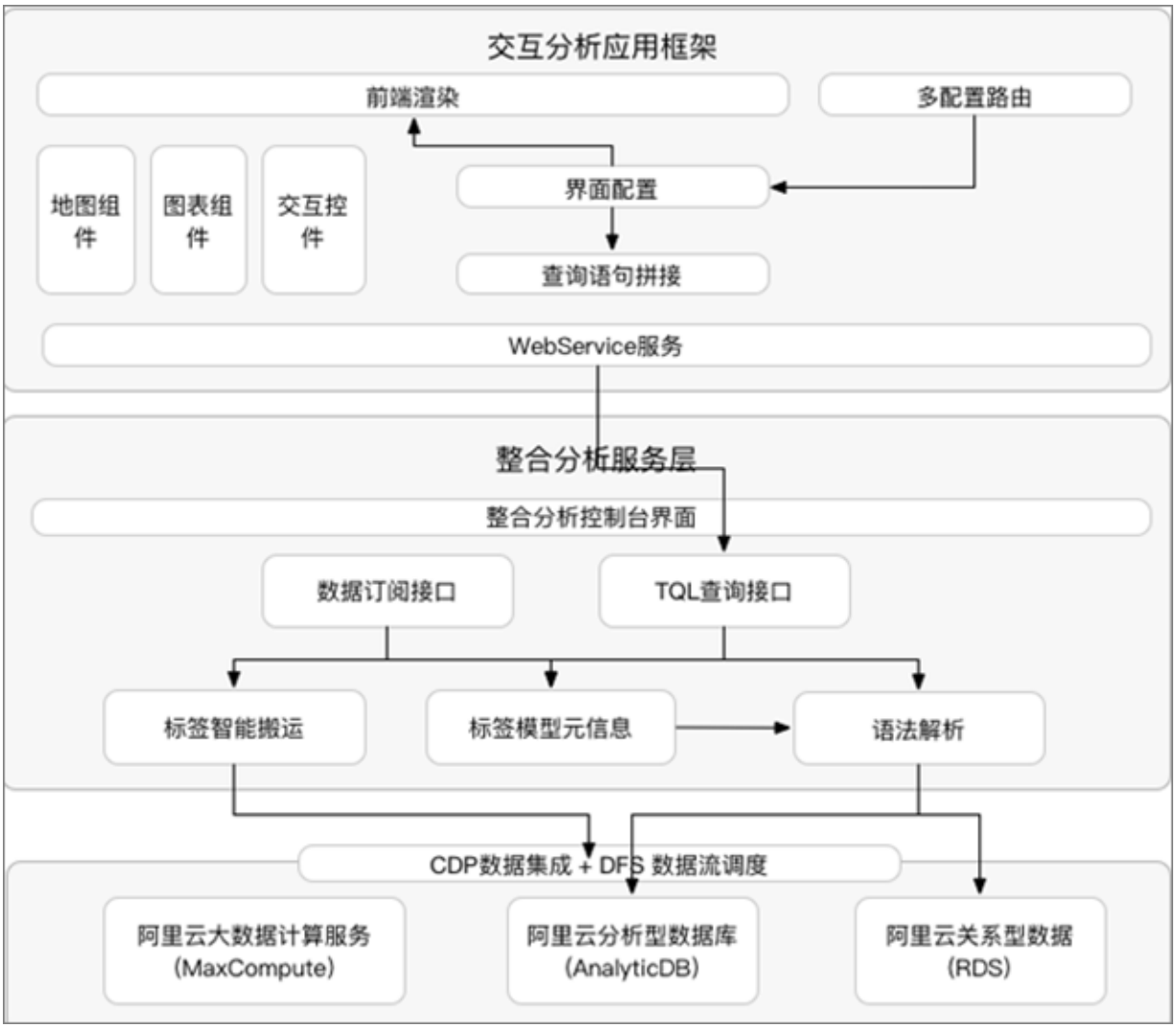
总数：1 / 3 页

首页 | 上一页 | 下一页 | 尾页

6.5.2.4 技术架构

画像分析的技术架构如图 *6-27:* 技术架构图所示。

图 6-27: 技术架构图



从技术架构来讲分为两个部分，在服务层部分通过TQL（Tag Query Laugage）查询接口接收您输入的参数，结合标签模型中的元信息会进行真实SQL的语法解析，然后把相应的SQL传送给相应的计算资源进行计算，通过接口获得返回的计算结果。使用Debug模式同时返回所解析的真实SQL以及每一步计算所耗费的时长，可以用于优化相应的查询语句。此处的语法解析主要是把您在扁平化的标签模型视图上的查询逻辑，翻译为真实的多个表之间关联JOIN的查询。

数据订阅部分可以通过界面实现数据的一键搬迁，控制台界面会调用数据订阅的接口获取相应数据的元信息，调用标签中心底层智能搬运的接口，在相应的计算存储资源会自动建表建立索引后，触发调度任务进行数据同步。

在交互分析的应用框架层，会提供应用开发配置框架的源代码。其中包括相应的前端组件、WebService后端服务、界面配置文件以及根据界面配置文件配置的查询接口。同时配置文件还能够进行相应的路由配置，让不同的页面URL可以路由到不同的配置，让不同的人看到不同的界面。

6.5.2.5 产品特性

DTBoost的特性如下所示。

- **一键数据整合**

  针对不同的分析主体，您可将多个存储资源当中的多个标签，通过一键数据整合功能同步至在线查询数据库中，并可进行索引操作，像管理一张表一样管理不同的数据源。兼容的在线分析数据库既可以是阿里云分析型数据库（Analytics DataBase），也可以是阿里云RDS关系型数据库。

- **Web开发友好**

  Web应用开发者直接通过与画像分析查询和标签元信息API接口的交互，结合阿里云DataV或是其他图表组件，即可快速搭建自己的分析型数据产品。

- **查询表达简单**

  在扁平化的标签体系上，一定程度简化了表关联和子查询的表达，让Web应用开发人员更加关注在应用逻辑而非数据表的组织逻辑。查询参数提供JSON对象模式，也提供与SQL相似的TQL模式。

- **与DTBoost其他模块无缝结合**

  由于标签体系下，多个模块之间共享同一个标签视图，以及同一个标签在不同的存储计算资源能够自动搬迁，使得画像分析能够与DTBoost的算法模块、特征工程模块、实时预警模块产出的数据有一致性的表达，相互打通无缝结合。

- **交互式分析应用框架**

  以SDK代码的方式提供分析界面配置工具，即刻生成交互式的分析应用。相比传统的BI工具，配置出来的分析界面像一个独立的交互分析产品，可以整合到您整体的分析系统当中，更容易让您对实体的属性、行为、地理出行等进行准确的分析。

6.5.3 规则引擎

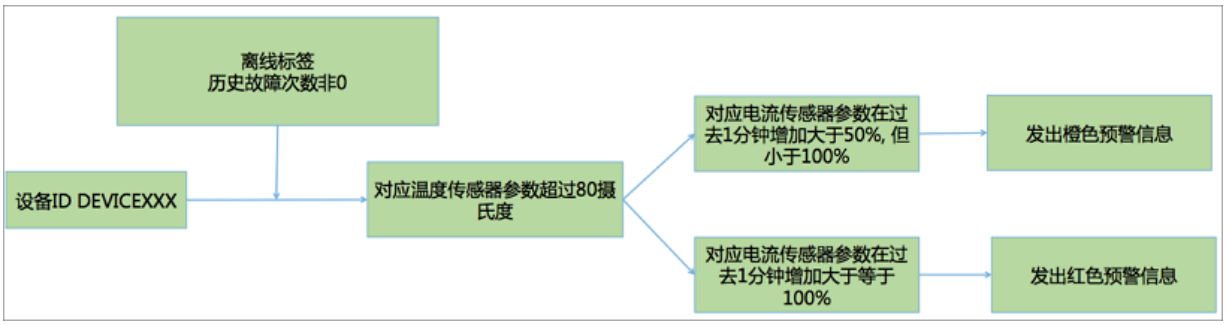
6.5.3.1 概念说明

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。

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。

规则引擎核心的概念即规则。一个可能的规则结构如图 *6-28:* 规则结构展示所示。

图 6-28: 规则结构展示



规则通常来说是一个if...then...结构，即条件与行为构成了一个完整的规则。在上面的示例中，包括如下几个条件。

1. 设为ID为DEVICEXXXX
2. 该设备历史故障次数非0
3. 该设备对应温度传感器参数超过80
4. 对应电流传感器参数在过去1分钟增加大于50%小于100%
5. 对应电流传感器参数在过去1分钟增加大于等于100%

而规则的行为，列出如下：

- 发出橙色预警信息
- 发出红色预警信息

当满足条件1、2、3、4时，发出橙色预警信息。当满足条件1、2、3、5时，发出红色预警信息。同时，规则条件的组成部分包括数据与对应计算逻辑。

在规则引擎中，数据在物理上就是一个个的表，被抽象为DTBoost中的标签。计算逻辑就是基于数据需要执行的计算流程，在规则引擎中被抽象为Function，比如最简单的大于小于等于，复杂一点的可以是跳变，多次跳变等。

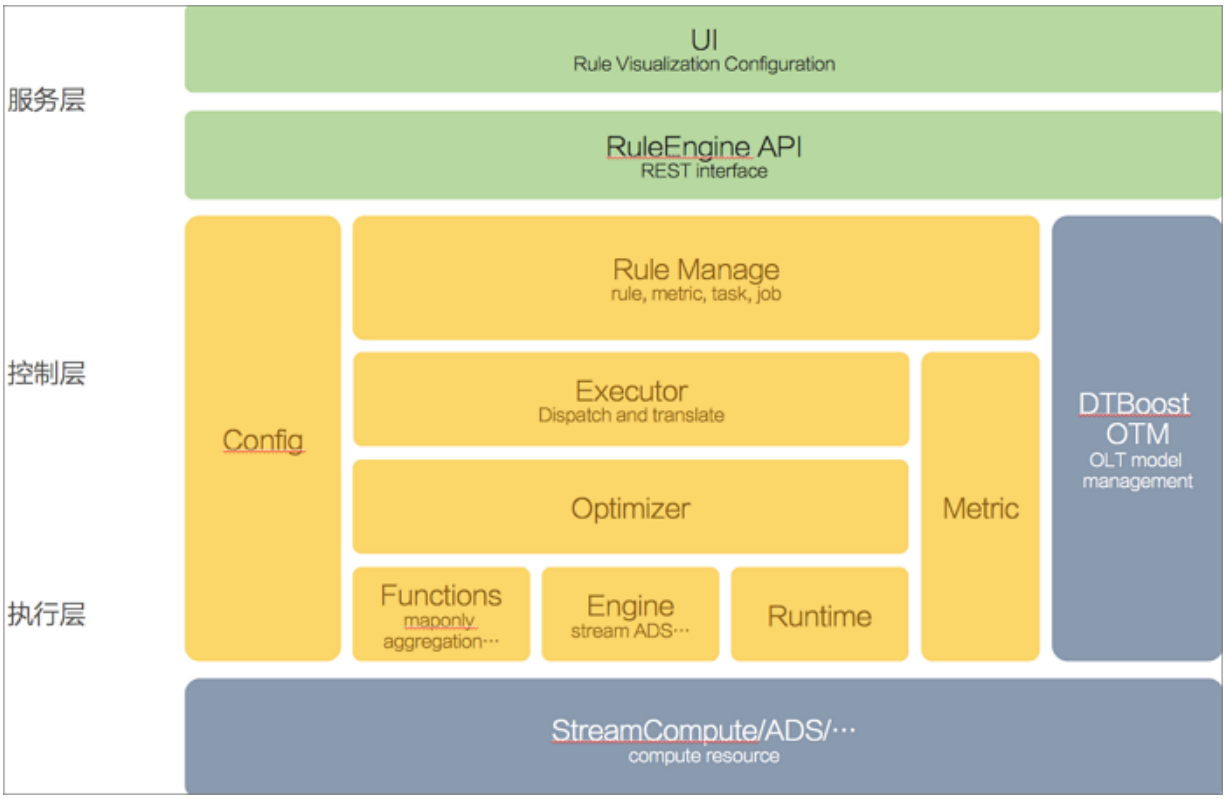
实际规则的执行，就是使用标签所描述的数据，去放到规则配置的Function 中进行执行，对得到的结果进行判断。

6.5.3.2 技术架构

6.5.3.2.1 功能模块

规则引擎的功能模块拆解，如图 *6-29:* 规则引擎功能模块所示。

图 6-29: 规则引擎功能模块



规则引擎整体分为三层：

- **服务层**

  服务层包含UI，提供了基本的规则管理功能。UI模块下的API模块提供了一组RESTful风格的API。规则引擎同时可以通过UI和API向外提供服务。您可通过UI访问规则引擎，也可通过API完成系统间的调用。同时，如果您有深度定制的需求，规则引擎的API模块提供了完备的功能供您基于不同行业需求进行二次开发。

- **控制层**

  控制层包括对规则管理，执行，优化等功能。并且提供了Metric采集功能，能够查看所有规则运行相关 Metric。

  说明：
  Metric是一种运维统计指标，可统计每个规则的输入数据量、产出预警量、延迟情况、处理速度等信息。

- **执行层**

执行层包括了规则实际提交到计算引擎，以及运行态的计算逻辑。此处提供了Function 扩展模块，可以通过对Function进行扩展，以支持新的判断条件。

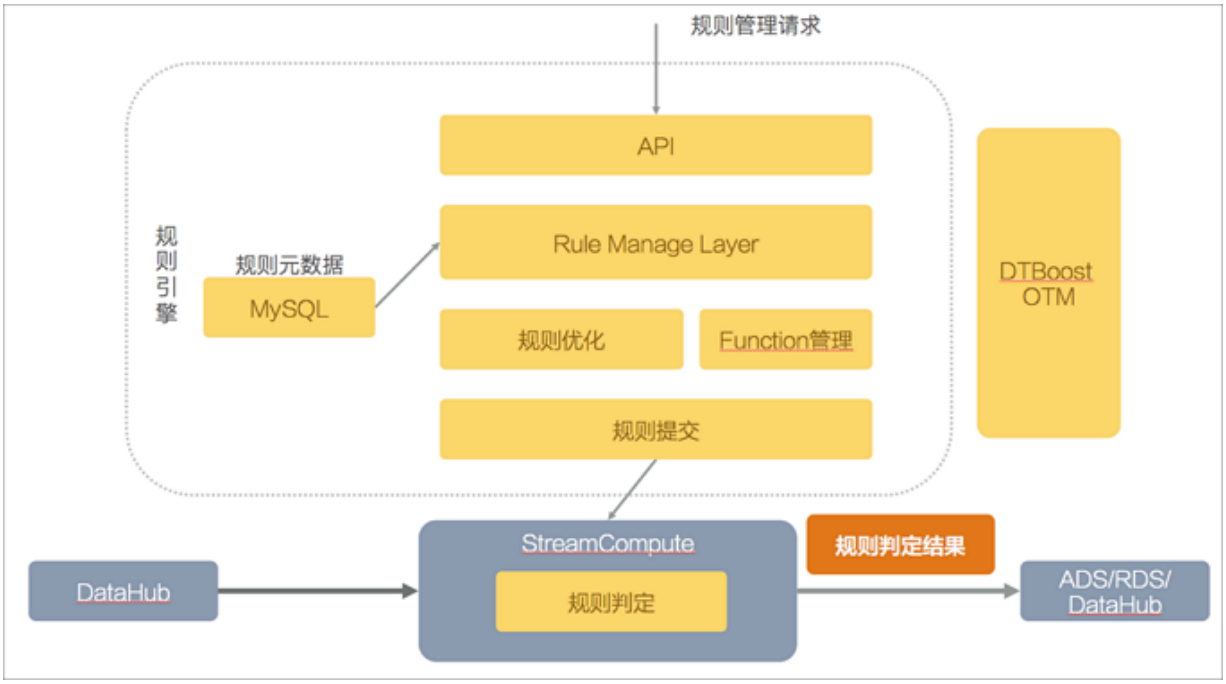
最底层的灰色模块为外部的计算引擎。

侧面灰色的OTM（Open Tag Management）模块是DTBoost所提供的标签管理模块，用于对云计算资源、实体关系以及标签进行管理。

6.5.3.2.2 规则提交流程

规则引擎中规则的提交执行流程，如图 *6-30:* 规则提交流程所示。

图 6-30: 规则提交流程



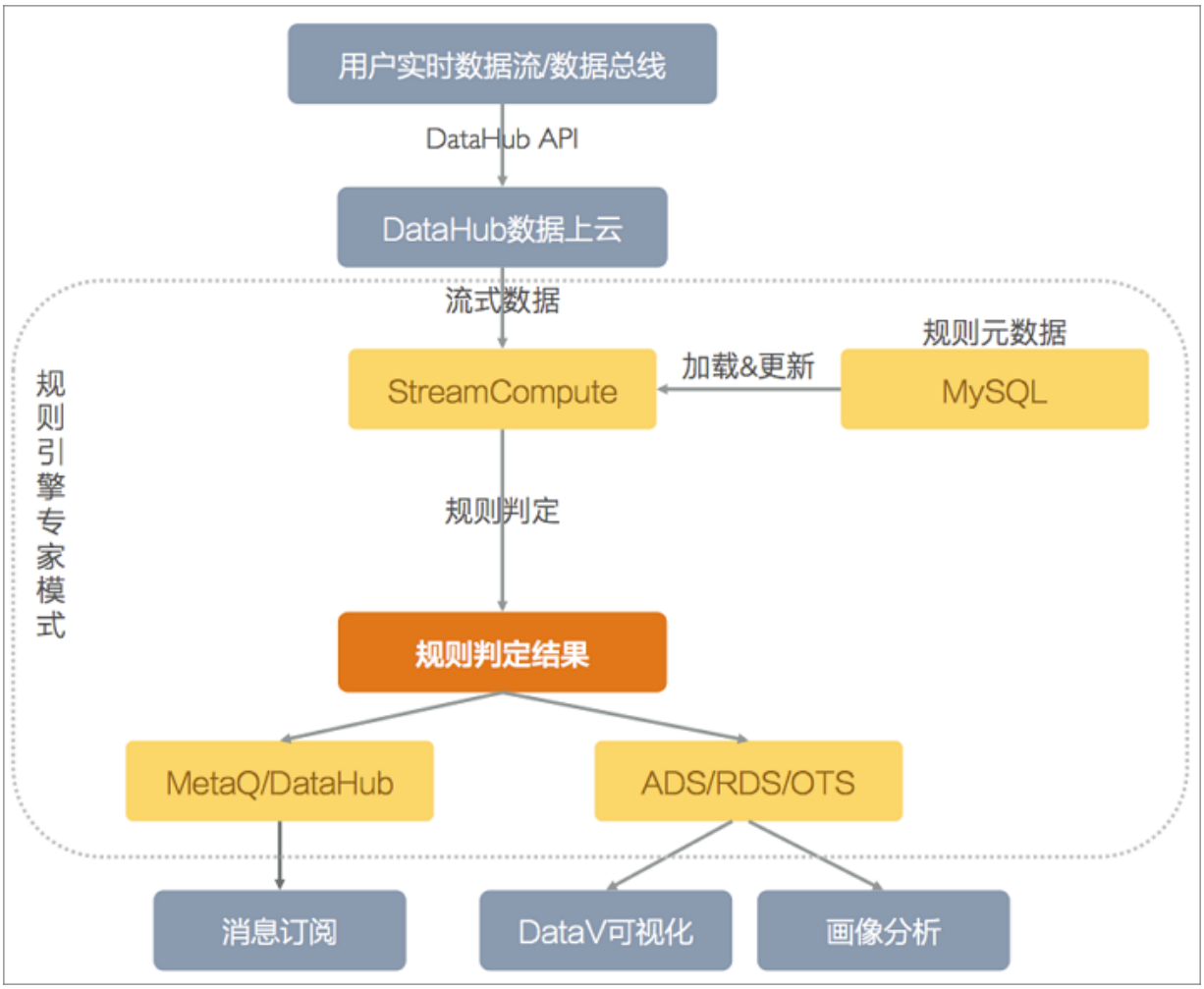
规则通过管理模块进行元数据注册，然后进行翻译，优化，并转化为对应 function 逻辑。同时，根据规则配置的计算引擎类型，提交到对应的计算引擎，作为计算引擎中的作业进行执行。

在规则执行过程中，不断消费上游发来的实时数据，进行规则 function 执行，并输出规则判定结果到对应下游目标存储中。

6.5.3.2.3 规则运行流程

规则运行流程，如图 *6-31:* 规则数据流所示。

图 6-31: 规则数据流



6.5.3.3 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 大数据管家BCC

7.1 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。

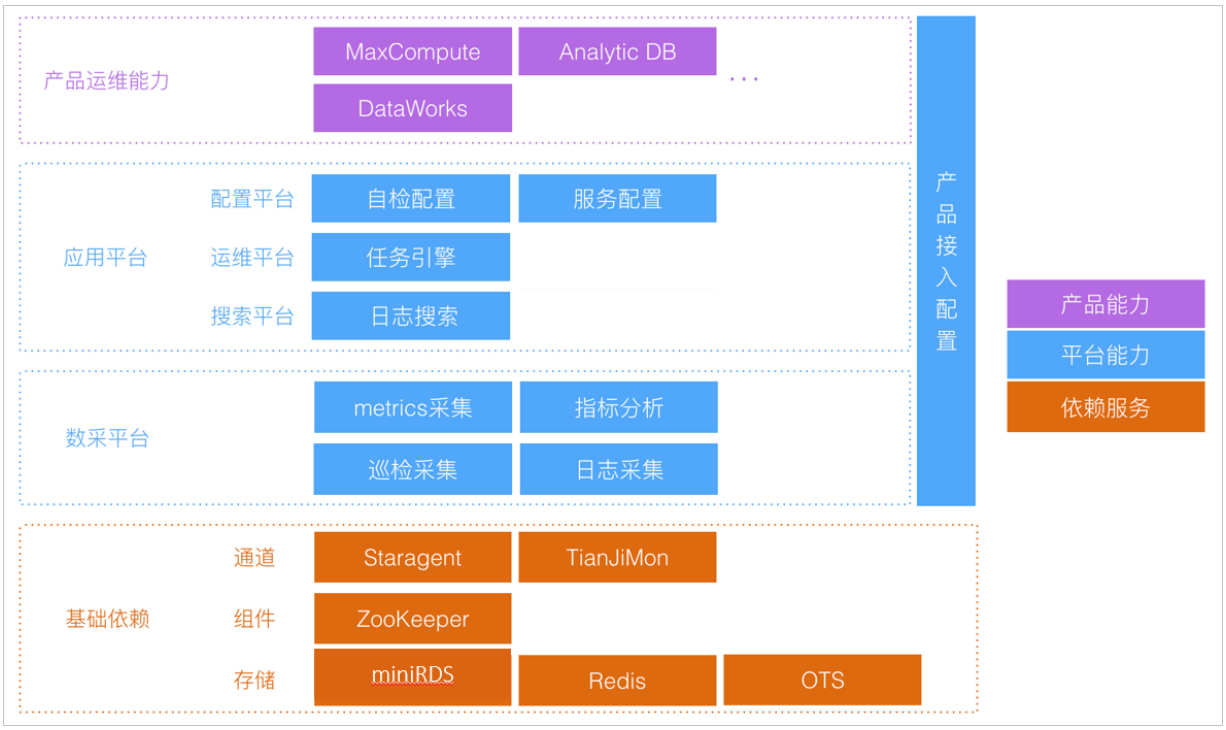
通过大数据管家的赋能，专有云驻场人员可以轻松地管理大数据产品，比如查看大数据产品的运行指标，修改大数据产品运行配置，对大数据产品进行自检，搜索日志等功能。

7.2 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 *7-1:* 产品架构所示。

图 7-1: 产品架构



7.2.1 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。

7.2.2 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。

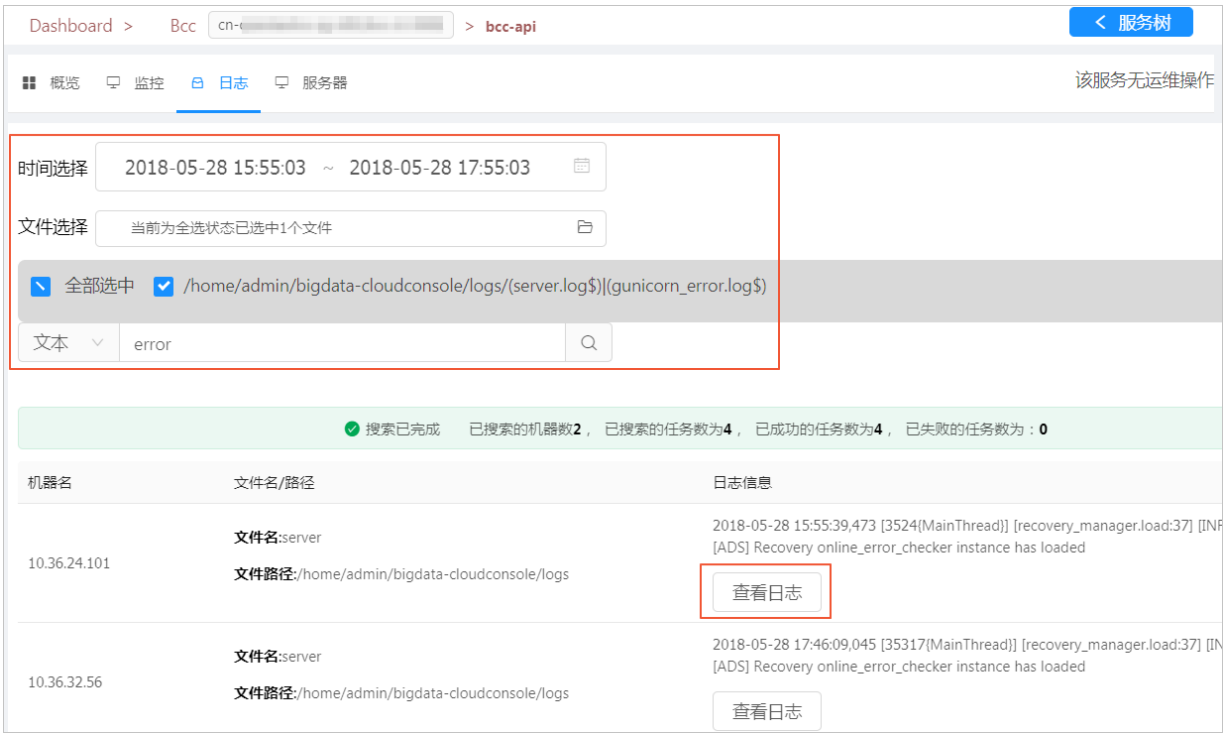
7.2.3 应用平台

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 7-2: 日志搜索所示。

图 7-2: 日志搜索



7.2.4 产品运维能力

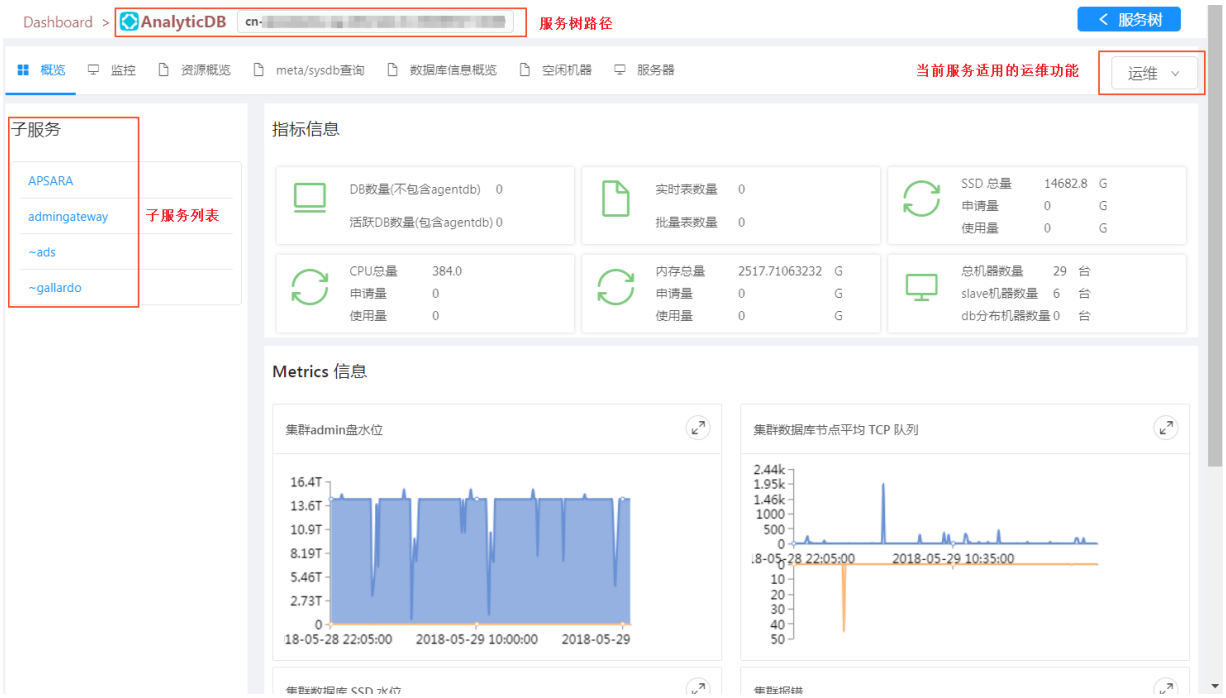
配置平台、运维平台以及搜索平台并不意味着就是所有大数据管家上提供的产品运维能力，因为部分跟产品有关的定制化运维功能则依托在每一个产品运维能力本身上。他们都会深度的与具体需要运维的产品融合和定制，达到一致性和特殊性的统一。例如AnalyticDB 会在日志平台的能力上，再提供更高层次封装的针对其服务的查询分析，并结构化展示分析数据，让您可以快速了解日志信息。这些定制都是透明的，您在具体运维场景上会顺利地了解这些定制内容。

7.3 功能特性

7.3.1 层次化服务管控

大数据管家主要使用层次化服务树来分解应用，也同时引导您了解被运维产品的特性，如图 7-3: 服务管控所示。

图 7-3: 服务管控



层次化服务树主要是对被运维应用的一种树形产品模块梳理，这种梳理可能面向服务组件梳理，也可能面向业务场景梳理，梳理出来后，每一个层次上的节点我们都称之为服务。

在这个树里您可以轻松的发现这个服务的父服务，以及它所依赖的子服务。对于服务本身的所有操作，都可以在这个服务的页面上找到。

服务树主要有以下几个特性。

- 产品树里面的服务可配置并实时动态更新。
- 支持跨产品依赖。
- 飞天、机器信息统一配置并快速接入。
- 按照产品树做信息展示和信息汇聚。

7.3.2 自我管控

大数据管家有自我管控能力，可以在应用平台层或底层依赖服务出现问题时显示问题定位和原因，方便您快速恢复大数据管家。

7.3.3 管控服务列表

大数据管家下管理的产品及它们的原名如下表所示。

产品名称	原名
MaxCompute	ODPS
AnalyticDB	ADS
DataWorks	Base
DTBoost	-
IPlus	I+
Quick BI	-
Apsara	-
Big Data Manager	BCC

7.4 故障处理

常见故障处理

- **登录故障**

  如出现无法登录的情况，请先清理浏览器的缓存、cookie，重新尝试登录。

  如果登录框提示登录异常，请根据提示异常检查以下内容。

  - 是否密码不对
  - 是否账号锁定
  - 是否账号被关停

- **其他异常**

  请寻找技术支持。

8 关系网络分析I+

8.1 产品概述

阿里云关系网路分析软件（简称I+）是阿里云推出的一款基于关系网络的海量数据智能可视化研判平台。

产品面向公安、工商、税务、海关、银行、保险、互联网金融等领域的大数据情报分析，为案件分析研判、反洗钱、反欺诈、反腐反贪、关联交易等调查分析提供强力支撑，帮助分析人员洞察数据中的关键信息，快速、智能的寻找破案线索及有价值的情报。

8.2 产品架构

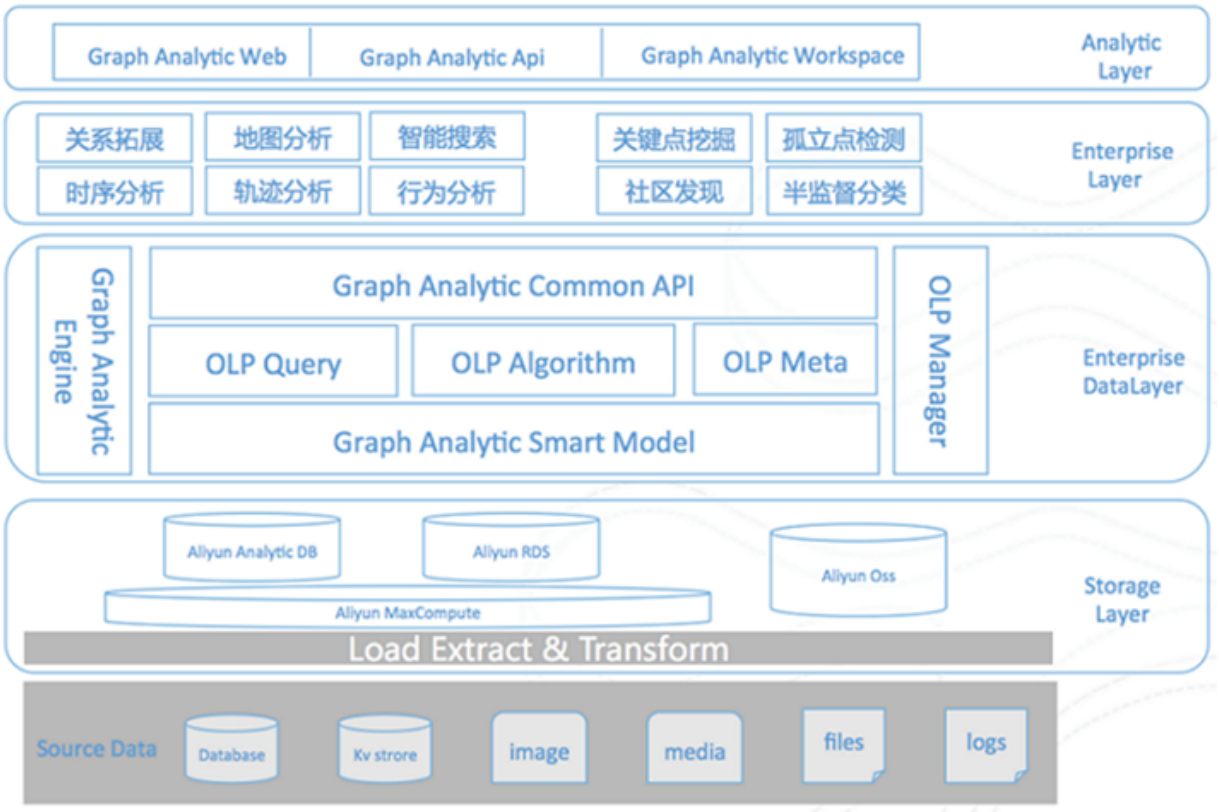
阿里云关系网路分析软件（Graph Analytic）（简称I + ）采用组件化、服务化设计理念，多层次体系架构。

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。

I+ 的系统架构如图 *8-1:* 系统架构图所示。

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。

图 8-1: 系统架构图



- **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。

8.3 功能特性

8.3.1 关系网络

关系网络是I + 研判分析平台的核心模块，所有实体之间的关系拓扑，业务计算，可视化布局，以及图交互操作在其中。同时有三大辅助分析组件用户空间、时序分析和信息立方补充关系网络研判，使之可以涵盖大部分研判业务场景。

8.3.2 时空网络

时空网络引入时间、空间维度，将实体、关联等数据和 GIS 地图结合，发掘出空间关联关系，并利用机器学习，智能计算同轨伴随、空间活动等信息。

8.3.3 搜索网络

搜索网络提供对象信息检索的功能，用户在分析过程中逐步递进拓展信息，从线索关键词开始细化分析，如**电话号码**：**138*****001**，**姓名**：**张三**，**住址**：**杭州市西湖区文三路**号**等。搜索提供检索工具帮助用户快速定位信息，同时作为关系网络和时空分析的入口，可以将检索对象信息引入到**关系网络**和**时空网络**继续拓展分析。

8.3.4 信息立方

产品提供图形、表格等多视角信息呈现方式，包括行为分析、时序分析、行为明细、网络统计、属性分布统计等，帮助用户进行多维信息洞察。

8.3.5 智能研判

智能研判是I + 平台上旨在研判过程中为业务提供智能化线索的算法应用。这些算法是I + 算法同学深入研判领域如公安、反恐以及税务中沉淀出的业务智能。目前I + 已经沉淀的伴随分析、涉恐指数、吸毒指数等应用，在多个项目中取得重大成果。

8.3.6 动态建模

复杂时空大数据往往以复杂关联网络形式存在，基于本体论和语义网技术，产品中设计实现大数据背景下通用领域抽象数据模型。用对象、关系的链接抽象的方式来组织刻画数据，将数据析解成对象（Object）、属性（Property）和事件（Event）以及对象间的关联关系（Link），形成OLEP数据模型。OLEP数据模型支持用户快速组织整合数据，同时支持业务模型灵活扩展。

8.4 产品优势

8.4.1 超大规模计算及存储

I+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级别数据规模的处理。计算存储平台包括大数据计算服务（MaxCompute）、分析型数据库（AnalyticDB）、分布式图计算（BigGraph）等。

大数据计算服务（MaxCompute）

阿里云大数据服务 MaxCompute，单个集群的规模可达5000台，并且具备跨机房的线性扩展能力，轻松处理海量数据。离线调度支持百万级任务量，实时监控告警。

核心指标：

- 万亿级数据 join，百万级并发 job，作业 I/O 可达 PB 级/天。
- 具备跨集群（机房）数据共享能力，支持万级别的集群数，扩容不受限制。
- 提供功能强大易用的 SQL、MR 引擎，兼容大部分标准 SQL 语法。
- MaxCompute（原ODPS）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用户的数据不丢失、不泄露、不被窃取。

分析型数据库（AnalyticDB）

阿里云分析型数据库（Analytic DB）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。

核心指标：

- 拥有快速处 理迁移级别海量数据的能力，使得数据分析中使用的数据可以不再是抽样的，而是业务系统中产生的全量数据，使得数据分析的结果具有最大的代表性。
- 采用分布式计算技术，拥有强大的实时计算能力，通常可以在数百毫秒内完成百亿级的数据计算，使得使用者可以根据自己的想法在海量数据中自由的进行 探索。
- 支撑较高并发查询量，并且通过动态的多副本数据存储计算技术来保证较高的系统可用性。

分布式图计算软件（BigGraph）

阿里云分布式图计算软件（BigGraph）是一款低延时高可用的分布式图计算产品，适用于大数据上的交互式图分析场景。

核心指标：

- 分布式存储

- 分布式计算
- 兼容 Gremlin 图遍历语言
- 基于多备份的高可用机制

8.4.2 跨数据源建模，多源整合计算，灵活高效部署

与耗时耗力的传统数据仓库物理模型不同，I+为分布在多个计算存储资源上的明细数据建立统一的OLEP逻辑模型。用户在对业务数据进行理解和梳理后，可以通过I+管理后台进行业务分析功能的建模和定义，即自定义业务逻辑模型、配置逻辑模型和实际物理数据存储的映射关系、配置应用场景的参数等。当数据源或业务有变动时，逻辑模型可以进行灵活修改并能够实时部署生效。

以目前I+实际落地的项目来看，在充分理解业务数据的基础上，通常在1-2天就能完成系统的部署和应用的上线使用。

8.4.3 智能算法组件集成，挖掘数据价值

I + 平台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和算法运行计算，平台结合经典战法，融入机器学习、业务算法，实现智能分析，如犯罪系数、伴随分析、骨干分析、路径分析等，帮助用户快速实现对关系网络数据的复杂挖掘。

8.4.4 智能可视化交互，提升用户体验

I + 可视化交互界面结合关系网络、地图分析、信息立方、时序行为面板等多个维度的信息交互提供给分析用户，方便用户从网络、时间、空间三个维度视角来探查分析。同时提供各种常用应用工具的操作，支持用户常用操作习惯，提升用户体验。

8.4.5 高度参数配置化，实现灵活的项目定制

I+平台支持包括数据源、业务模型，样式图标，用户角色权限，技术参数、系统参数等基本上所有可配置参数的配置化管理。用户可以根据项目的实际情况在管理控制台灵活自定义配置，满足具体项目的实际业务需求。

8.5 产品价值

目前阿里云关系网路分析软件（简称I + ）已经作为亮点应用参与了多个国家级重点项目，涉及包括公安、金融、国税、保险、互联网金融等多个行业多个领域，尤其在安保、反恐、反洗钱、税务欺诈等细分领域的应用让客户耳目一新，业务价值得到深度考验和认可。

下面列举一些典型行业的典型应用案例，说明阿里云关系网络分析软件在真实场景中如何通过交互式的所见及所得的方式，让用户在复杂数据环境中理清数据之间的关联和逻辑关系，并通过直观和友好的用户界面展现给用户。

说明：
案例截图来源于真实应用系统，相关的案例敏感数据已做脱敏处理。

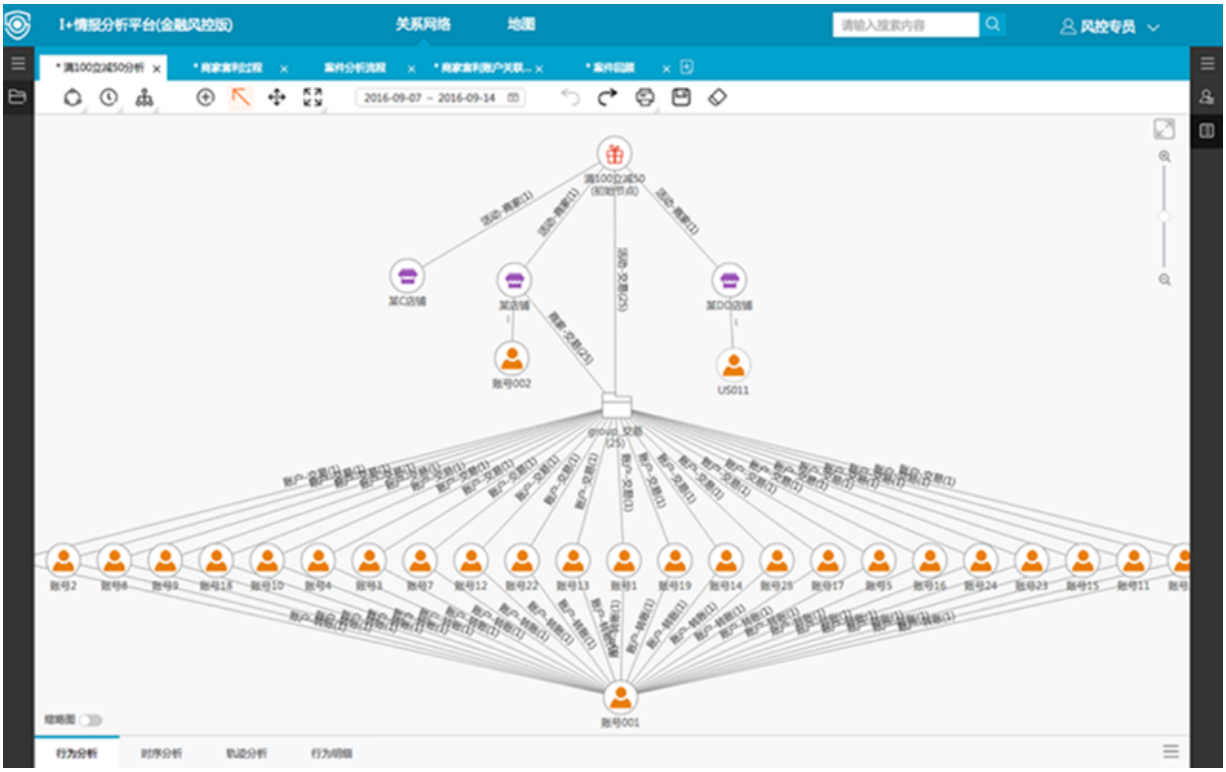
8.5.1 金融行业应用

此案例描述的是互联网金融风控专员如何通过阿里云关系网路分析软件（简称I + ）进行**营销反作弊**的分析。

某互联网企业投放大额资金在线下店铺开展买100立减50的纳新营销活动，部分店铺企图套取营销资源，通过事先批量注册小号并在活动前进行转账激活，活动当天在店铺购买物品产生虚假交易，骗取营销资源并获利。

该互联网企业综合利用转账、位置、设备、环境等信息，构建可疑关系网络，通过I+情报分析平台进行快速定位分析，如图 *8-2: 营销反作弊分析*所示，最终追回被套利资源，避免了营销资源的浪费，同时对相应的环境及账号进行布控，对该店铺进行相应的处罚。

图 8-2: 营销反作弊分析



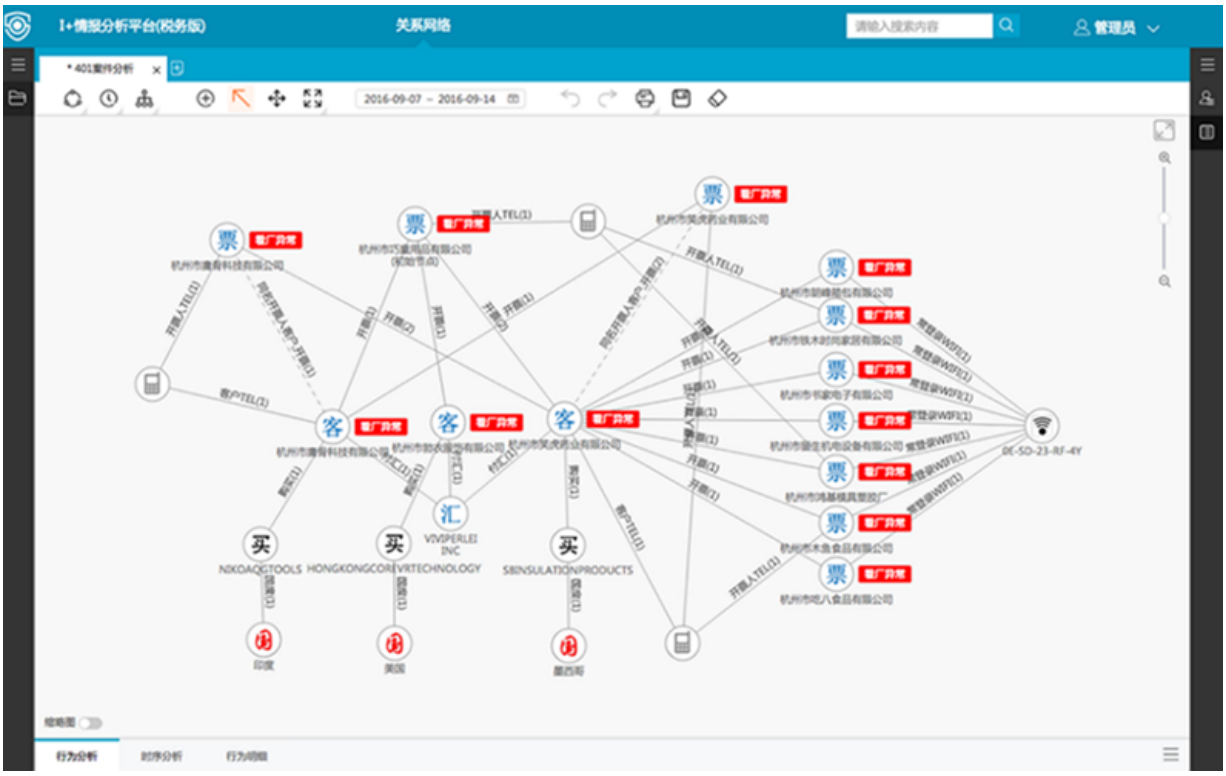
8.5.2 税务行业应用

此案例描述的是税务分析人员如何通过阿里云关系网路分析软件（简称I + ）进行**出口骗税**的分析。

某外贸综合服务平台为中小企业提供专业、低成本的通关、外汇、退税以及配套物流和金融服务，但是部分企业存在投机取巧、造假骗税的情况，骗税金额不断攀升，对平台的外贸资质带来重大负面影响。

分析人员利用I+情报分析平台的关联反查、群体分析、骨干分析、共同邻居、信息立方以及行为分析等功能，如图 *8-3:* 出口骗税分析所示，从被举报的企业入手，查获了一个涉及13家企业的大规模骗税团伙，该团伙在平台上注册不同角色，内部间虚开发票、作假交易，向贸易平台申报退税金额达到上亿人民币，实地看厂调查后，对该团伙实施了相应的处罚，并追回损失。

图 8-3: 出口骗税分析



9 Quick BI

9.1 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。

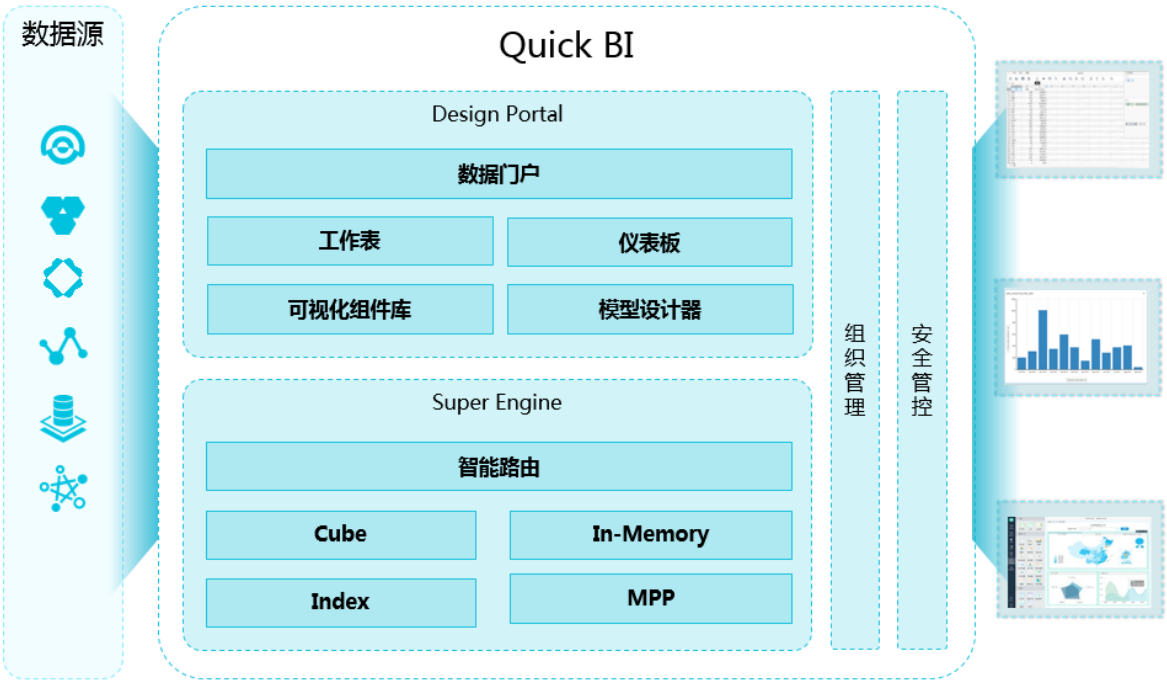
Quick BI支持众多种类的数据源，既可以连接MaxCompute（ODPS）、RDS、AnalyticDB、HybridDB（Greenplum）等云数据源，也支持连接ECS上您自有的MySQL数据库，同时，还支持连接来自VPC的数据源。Quick BI可以为您提供海量数据实时在线分析服务，通过提供智能化的数据建模工具，极大降低了数据的获取成本和使用门槛，通过拖拽式的操作和丰富的可视化图表控件，帮助您轻松自如地完成数据透视分析、自助取数、业务数据探查、报表制作和搭建数据门户等工作。

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。

9.2 产品架构

Quick BI产品架构如下图所示。

图 9-1: 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。

9.2.1 部署方案

通过天基进行 Quick BI 自动化部署。

9.2.2 组件及作用

Quick BI 分为如下几类服务角色。

- base-biz-yunbi-dbinit：进行元数据初始化
- quickbi-redis-slave： redis 缓存 slave
- quickbi-redis-master：redis 缓存 master
- base-biz-yunbi-executor： quickbi agent 服务
- base-biz-yunbi： quickbi 前台页面 web 服务
- ServiceTest： 自动化测试服务

9.3 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。

9.4 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 流计算StreamCompute

10.1 产品历程

阿里云流计算源自于阿里集团内部双十一实时大屏业务，从最开始支持双十一大屏展现和部分实时报表业务的实时数据业务团队，历经约5年的长期摸索和发展，到最终成长为一个独立稳定的云计算产品团队。阿里云流计算期望将阿里集团本身沉淀多年的流计算产品、架构、业务能够以云产品的方式对外提供服务，助力更多的企业实时化自身大数据业务。

最初阿里集团支撑双十一大屏等业务同样采用的是开源的Storm作为基础系统支持，并在上面开发相关Storm代码。这个时期的实时业务处于萌芽阶段，规模尚小。数据开发人员使用Storm原生API开发流式作业，开发门槛高，系统调试难，存在大量重复的人工工作。

阿里集团的工程师针对这类大量重复工作，开始考虑进行业务封装和抽象。首先我们希望有个流计算和批处理的一体化处理方案。Spark和Flink都具有流和批处理能力，但是他们的做法是相反的。Spark Streaming是把流转化成一个个小的批来处理，这种方案的一个问题是我们需要的延迟越低，额外开销占的比例就会越大，这导致了Spark Streaming很难做到秒级甚至亚秒级的延迟。Flink是把批当作一种有限的流，这种做法的一个特点是在流和批共享大部分代码的同时还能够保留批处理特有的一系列的优化。因为这个原因，如果要用一套引擎来解决流和批处理，那就必须以流处理为基础，所以我们决定先选择一个优秀的流处理引擎。从功能上流处理可以分为无状态的和有状态两种。在流处理的框架里引入状态管理大大提升了系统的表达能力，让您能够很方便地实现复杂的处理逻辑，是流处理在功能上的一个飞跃。

任何技术的发展一定遵循小众/创新到大众/普及的成长轨迹，而从小众到大众、从创新到普及的转折点一定在于技术的功能成熟和成本降低。阿里工程师开始思考如何更大程度降低数据分析产品门槛从而普及到更多的用户。Flink在架构上有很多创新，是非常领先的，但是在工程上的实现有一些不足支出，不同的Job的任务可能会运行在同一个进程里，这样大大降低了系统运算的性能，阿里的工程师为了解决这个问题，重新实行了Yarn的结合，完全解决了这些问题，另外Flink是通过checkpoint的机制来保证一致性的，但原有的机制效率比较低，导致在状态（增量计算保存的状态）较大的时候不可用，Blink大大优化了checkpoint，能够高效地处理很大的状态。稳定性和可扩展性在生产上都是至关重要的，通过在大集群上的锤炼，Blink解决了一系列这方面的问题和瓶颈，已经成为一个能够支撑核心业务的计算引擎。同时我们扩展了Flink的 SQL层，使得它能够比较完备地支持较复杂的业务，正是不断的去发现、去创新，才有了今天阿里云流计算的核心计算引擎（Blink）。

10.2 产品特点

StreamCompute的底层引擎Blink是基于Flink开发的，继承了Flink的所有优点并改进了Table API，使其更完整，因此您可以使用相同的SQL进行批处理和流式处理。 更强大的YARN模式，仍然100%兼容Flink的API和更广泛的生态系统。

图 10-1: 流计算技术栈及StreamCompute和竞品的关系

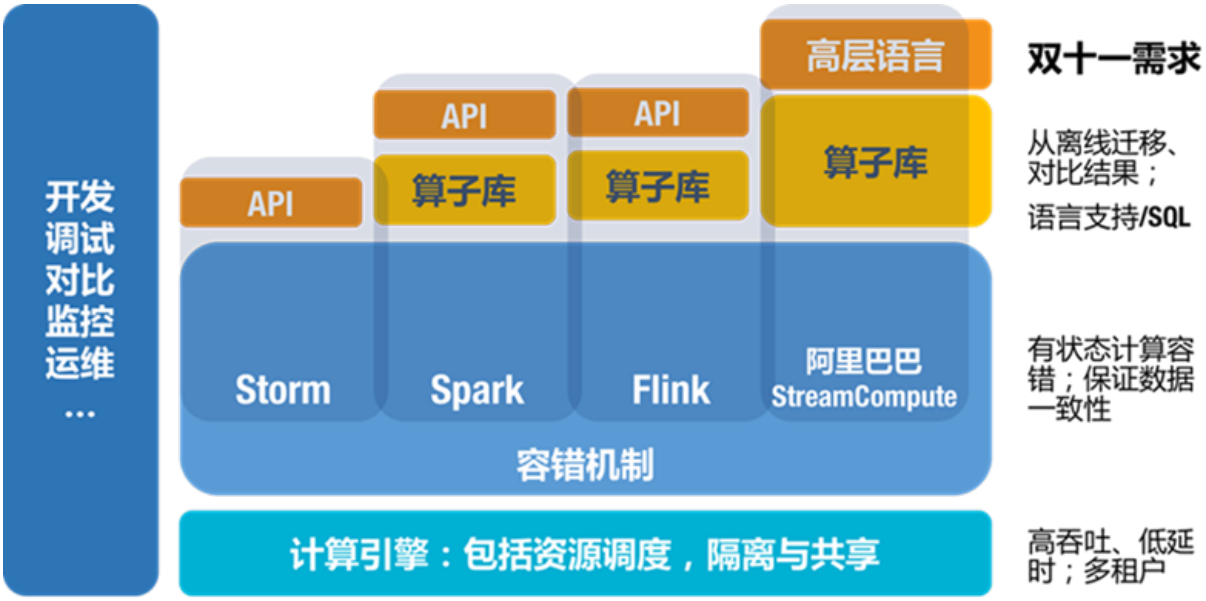


图 *10-1: 流计算技术栈及StreamCompute和竞品的关系*展示了流计算产品的技术栈，阿里云StreamCompute在世界级挑战的业务场景锤炼中，逐步形成了如下竞争优势。

- **功能强大**

  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造轮子重新实现。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，包括：

  - 强大的流计算引擎。
    - 提供流式计算的标准StreamSQL，支持各类Fail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处理、时间、统计等类型函数。
    - 精确的计算资源控制，彻底保证多租户之间作业的隔离性。
  - 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亚秒级，单个作业吞吐量可做到百万(记录/秒)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)，提供内建的字符串处理、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、监控预警等不同阶段辅助套件，让数据开发仅需三步，即可完成流式计算作业上线。

- **成本低廉**

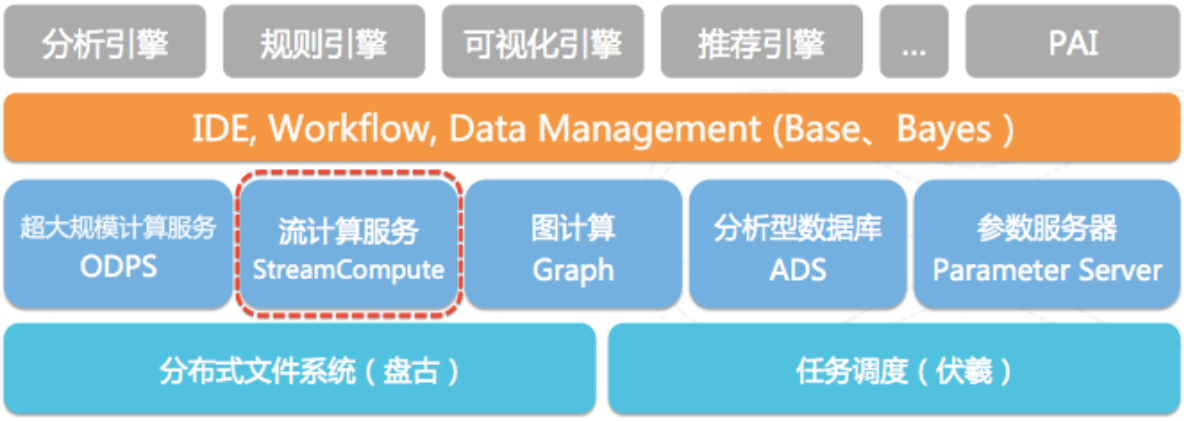
  大量优化的SQL执行引擎，会产生比手写原生Flink作业更高效且更廉价的计算作业，无论开发成本和运行成本，阿里云流计算均要远超开源流式框架。

10.3 产品战略地位及发展路线

阿里云流计算在阿里集团内部有数千台规模的集群，服务20多个BU的数百个实时应用，日均消息处理数千亿，流量近PB级别，成为阿里集团最核心的分布式计算服务之一。

StreamCompute 在阿里云计算平台中的位置如图 *10-2:* 流计算战略地位所示。

图 10-2: 流计算战略地位



阿里云流计算后续主要在以下几个方面重点加强。

- 计算引擎：重点针对性能提升、多种消息处理语义支持等方面进行优化。
- 编程接口：提供更丰富的API支持，支持多语言，兼容开源系统API，比如Storm API、Beam API等。
- 语言：丰富Streaming场景的SQL表达能力，增加对Temporal、CEP等语法和语义的支持。
- 产品：对StreamCompute的可调试性、一键部署、热升级、培训体系等方面持续进行完善。

10.4 产品架构

10.4.1 业务架构

阿里云流计算是指一套轻量级的提供SQL表达能力的流式数据加工处理引擎。其业务架构如下：

- **数据产生**

  生产数据发生源，通常在服务器日志、数据库日志、传感器、第三方数据均是数据产生方，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**

  提供流式数据集成的用以进行数据发布和订阅的数据总线，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数据存储DataHub。

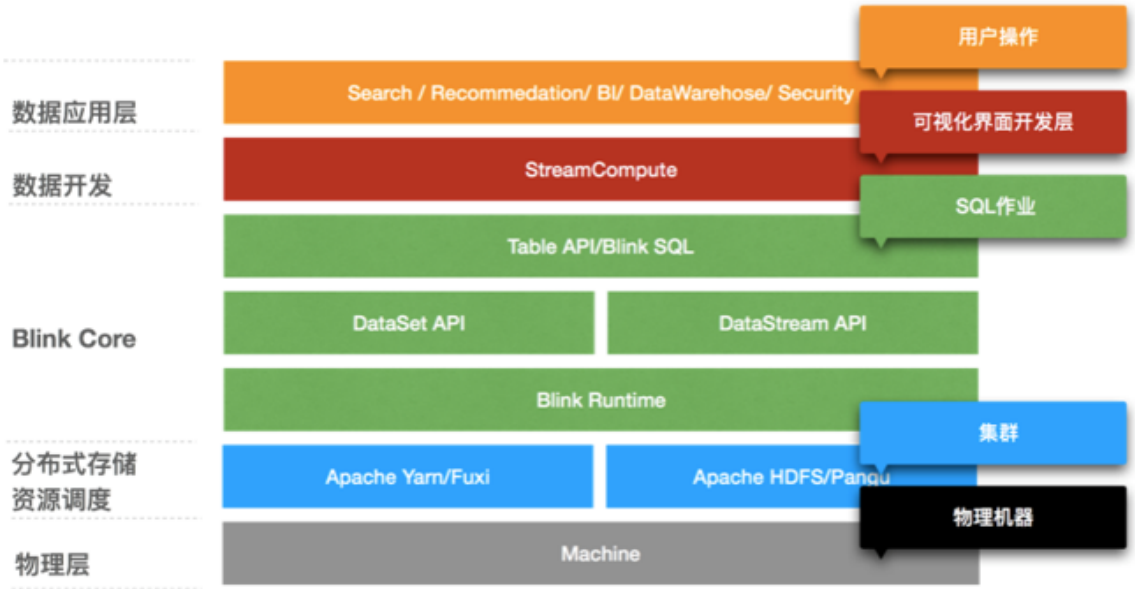
- **数据消费**

  不同的数据存储可以进行多样化的数据消费。提供消息队列的数据存储可以用作报警，提供关系型数据库的可以提供在线业务支持等等。

10.4.2 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 10-3: 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 实时数据分发平台DataHub

11.1 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。

11.2 产品特性和核心优势

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

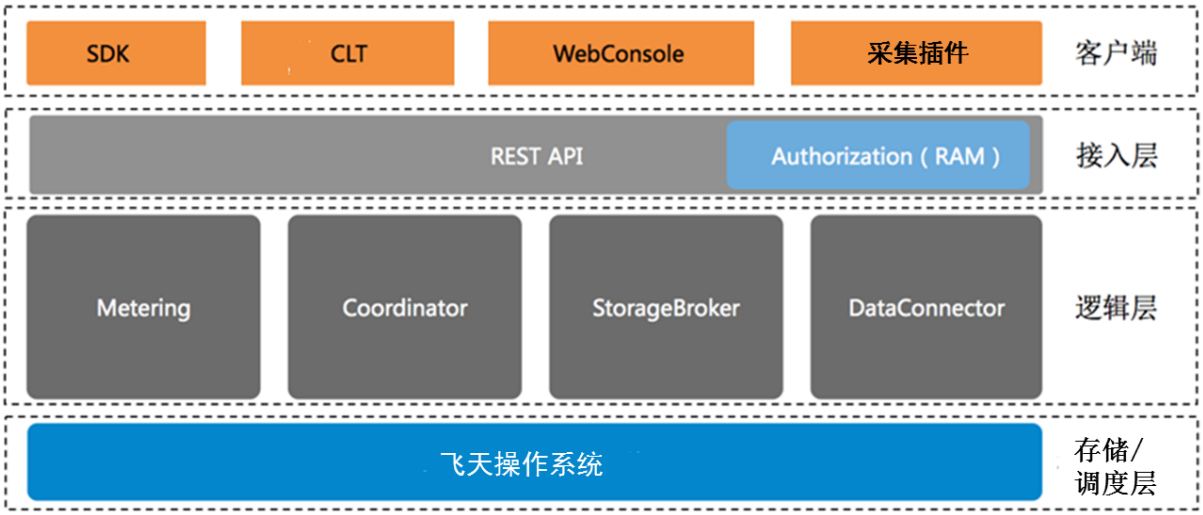
- 提供企业级多层次安全防护，多用户资源隔离机制。
- 提供多种鉴权和授权机制及白名单、主子账号功能。

11.3 产品架构

11.3.1 功能架构

DataHub的产品功能架构如图 *11-1: DataHub产品功能架构图*所示：

图 11-1: DataHub产品功能架构图



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。

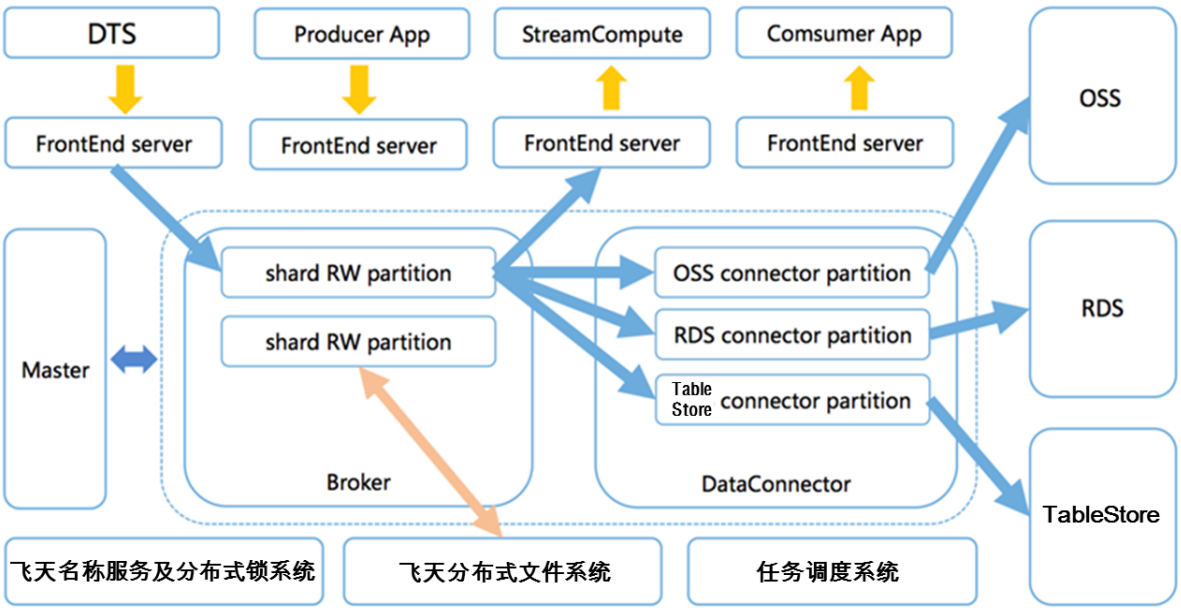
存储/调度层

- 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。

11.3.2 技术架构

DataHub的技术架构如图 *11-2: DataHub技术架构图*所示：

图 11-2: DataHub技术架构图



上图为数据从产生到消费一个基本流程。

1. DataHub中Shard做为数据管理的最小单位，逻辑上可以看做一个先进先出队列。
2. 物理上每个Shard对应飞天分布式文件系统上一组Logfile。
3. Master是以Shard为单位进行调度，每个shard会被调度到一台Broker机器上，每台Broker负责多个Shard读写。
4. Frontend server会根据请求中的project/topic/shard组合定位到一台broker并将数据请求转发到Broker上。
5. DataHub Connector做为服务的内部模块可以直接从Broker上读取数据，并将数据转发给其它服务。

数据采集

DataHub支持多种数据写入方式：使用SDK开发的数据应用、DataHub的数据采集插件（LogStash/Fluntd/Flume）、数据采集服务（DTS）、流计算（StreamCompute）产生的数据。

Frontend Server

提供Rest API接口和鉴权服务，做为DataHub服务的统一入口层，支持水平扩展。

Master

提供Meta管理服务和Shard调用服务，支持project和topic基本的增删改查、shard分裂/合并以及shard调度。

Broker

负责每个Shard中数据的读/写功能，包括数据的索引、缓存、数据文件的组织和管理。

Connector

Connector的功能是从DataHub中将数据转发到阿里云的其它服务中。针对不同的服务Connector提供不同的功能点，例如MaxCompute的自动创建分区、OSS的自动切分文件等功能。

11.4 产品价值

DataHub具有如下的价值点：

表 11-1: 价值点

价值点	DataHub
安全性	高（基于阿里云RAM权限体系）
运维难度	容易（自动化的上下线）
资源隔离	租户隔离
生态整合	丰富
规模扩展性	强（经历过双十一的考验）
读写功能	索引机制导致实时入库数据难以支持分析型应用，海量数据碰撞比对、分析预测计算时间可能超过24小时，性能无法满足需求。
高可用	丰富
与阿里云上下游产品整合	无缝

11.5 功能描述

11.5.1 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。

11.5.2 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。

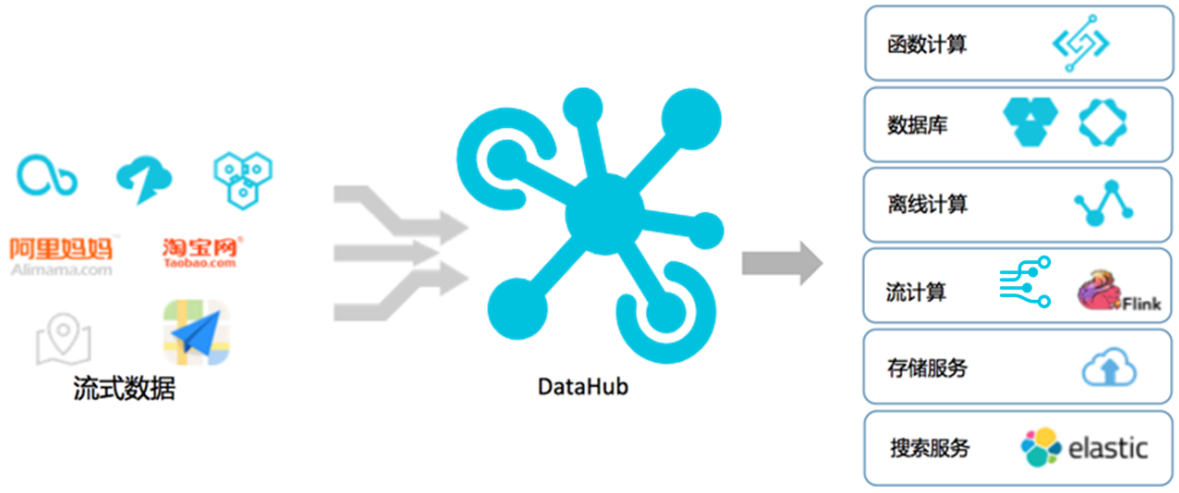
11.5.3 数据同步

DataHub中的数据自动同步到阿里云其它服务，目前支持的服务包括：MaxCompute、OSS、AnalyticDB、RDS、TableStore、Elasticsearch、FunctionCompute。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。

11.6 应用场景

11.6.1 数据上云入口

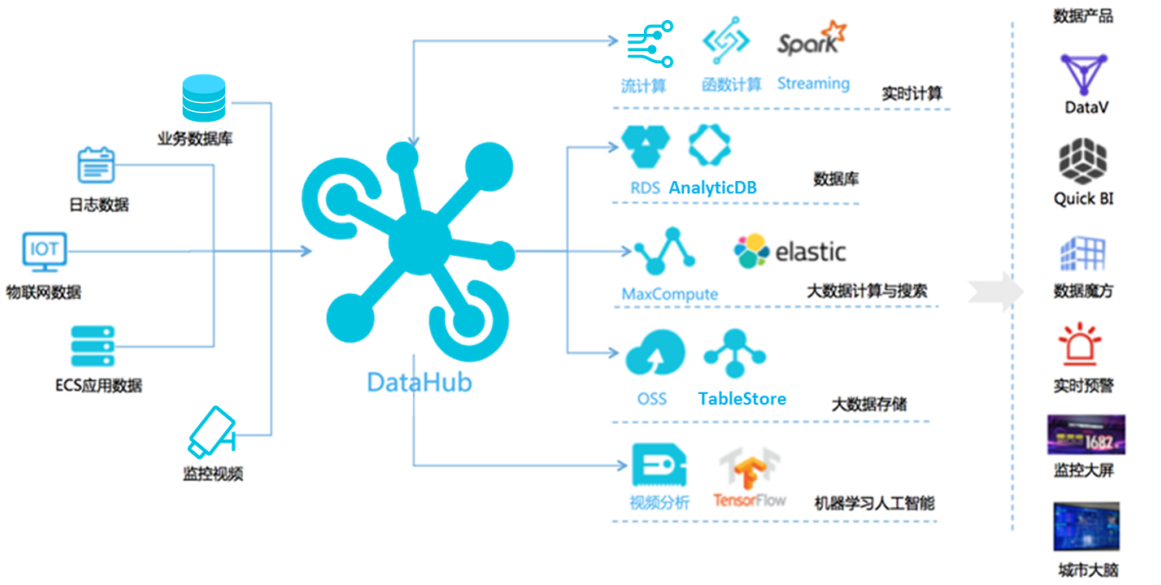
图 11-3: 数据上云入口



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。

11.6.2 数据采集通道

图 11-4: 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）。