阿里云
专有云Enterprise版

技术白皮书

产品版本：V3.0.0

文档版本：20171101

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. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，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、发行或公开发表阿里云网站、产品程序或内容。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. 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

表 1: 格式约定

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 云服务器ECS

1.1 产品概述

本节对云服务器的基本概念进行简单的介绍。

云服务器（Elastic Compute Service，简称ECS）是处理能力可弹性伸缩的计算服务，它的管理方式比物理服务器更简单高效。根据业务需要，您可以随时创建实例、扩容磁盘或释放任意多台云服务器实例。

云服务器ECS实例（以下简称ECS实例）是一个虚拟的计算环境，包含CPU、内存等最基础的计算组件，是云服务器呈献给每个用户的实际操作实体。ECS实例是云服务器最为核心的概念。其他的资源，比如磁盘、IP、镜像、快照等，只有与ECS实例结合后才能使用。

云服务器ECS示意图如图 *1: ECS* 示意图所示。

图 1: ECS 示意图

1.2 产品架构

云服务器ECS系统由虚拟化平台与分布式存储、控制系统、运维及监控系统组成。

1.2.1 虚拟化平台与分布式存储

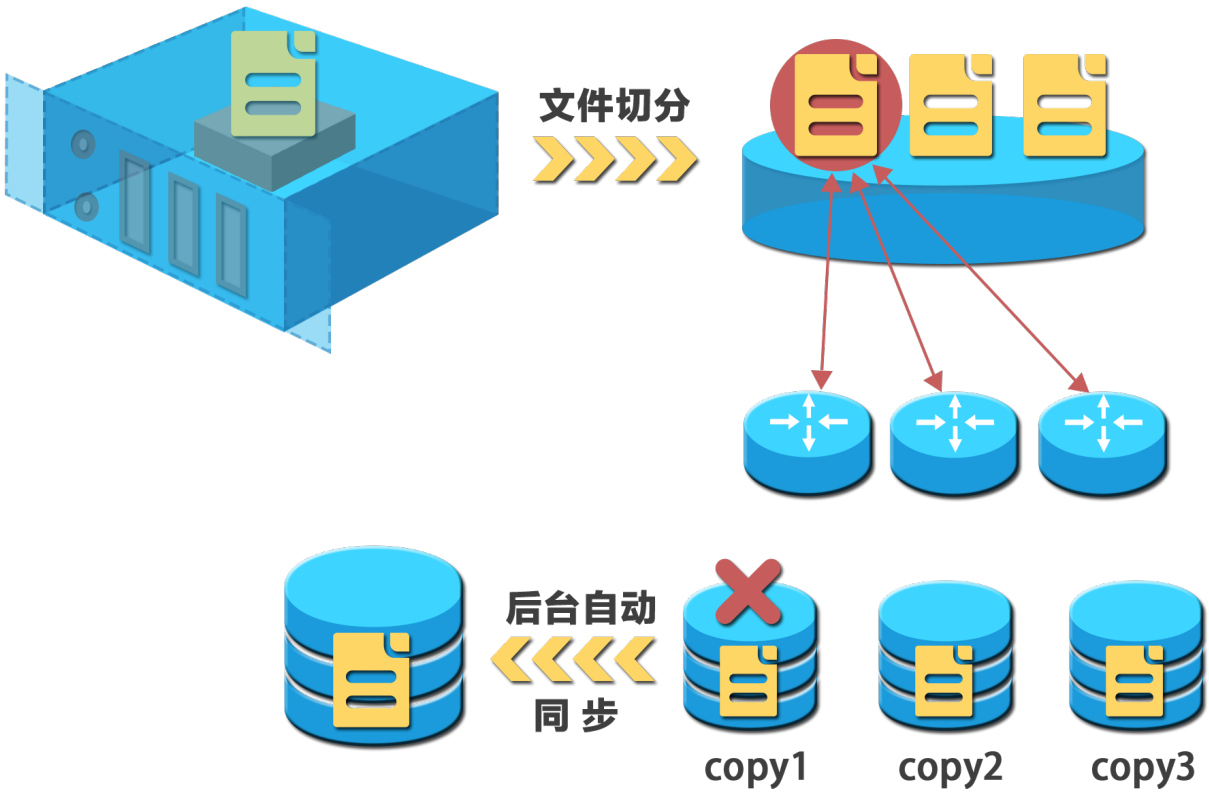
虚拟化是ECS的基础。阿里云采用KVM虚拟化技术，将物理资源进行虚拟化，通过虚拟化后的虚拟资源，对外提供弹性计算服务。

ECS包括两个重要的模块：计算资源模块和存储资源模块。

- 计算资源指CPU、内存、带宽等资源，通过将物理服务器上的计算资源虚拟化再分配给ECS使用。一台ECS的计算资源只能位于一台物理服务器上。当一台物理服务器上资源耗尽时，只能在另外的物理服务器上创建ECS。通过资源的QoS，保证同一台物理服务器上不同ECS互不影响。
- 存储采用了大规模分布式存储系统，将整个集群中的存储资源虚拟化后，整合在一起对外提供服务。同一台ECS的数据，保存在整个集群中。在分布式存储系统中，每份数据都提供三份副本，当单份数据损坏后，可实现数据的自动拷贝。

具体原理图见下图。

图 2: 虚拟化平台与分布式存储原理图



1.2.2 控制系统

控制系统是弹性计算平台的核心，它决定着ECS启动在哪一台物理服务器上且ECS的所有功能及信息都需要通过控制中心统一处理与维护。

控制系统由以下模块组成：

- **数据采集**

  负责整个虚拟化平台的数据采集，包括计算资源、存储资源、网络资源等使用情况。通过数据采集可以对集群的资源使用情况进行统一的监控管理，并作为资源调度的一个重要的依据。

- **资源调度系统**

  决定ECS启动的位置，在创建ECS时，会根据物理服务器的资源负载情况，合理地调度ECS。且在ECS发生故障时，决定ECS再次启动的位置。

- **ECS管理模块**

  管理及控制ECS，例如启动、关闭、重启云服务。

- **安全控制模块**

  进行整个集群的网络安全监控与管理。

1.3 功能特性

ECS是弹性计算产品的核心部分。它主要为用户提供计算能力服务。创建并启动一台ECS只需数分钟，且ECS一经创建即有特定的系统配置。与传统服务器相比，大大提升了用户业务开展的效率。

使用ECS与传统托管物理服务器使用方法完全相同，用户对ECS有完全控制权，可通过远程的方式或API的方式（控制面板）来对ECS进行一系列基本操作。

ECS的计算能力可用虚拟CPU、虚拟MEM来表示；磁盘存储能力可用云磁盘容量来衡量。区别于传统服务器，ECS有较为灵活的机器配置。

用户可以根据需求灵活配置ECS，在服务器运行过程中，如果现有服务器配置不能满足业务需求，可随时调整服务器配置。

ECS的生命周期从ECS创建到ECS释放。当ECS释放后，所有的数据将彻底删除，不可找回。

阿里云专有云ECS控制面板管理界面一般会包括：

- **资源概览**

提供了您已购买并且实例资源还在服务周期内的数量、运行状态、实例资源到期提醒、云盾报警提醒总览。

- **实例列表**

  查看和管理您已购买并且还在服务周期内的实例；提供在线重启、停止、启动、释放、远程登录ECS、更换系统盘、重置密码、变配（升级、降级）的操作；提供查看实例运行监控信息、配置信息。

- **磁盘列表**

  查看和管理您创建的磁盘；提供在线重新初始化磁盘、创建快照、设置自动快照策略、释放磁盘、挂载/卸载磁盘的功能；提供查看磁盘性能监控信息、配置信息。

- **快照列表**

  查看和管理您创建的快照信息；提供在线回滚磁盘、创建自定义镜像、删除快照的功能。

- **镜像列表**

  查看和管理您创建或被分享的快照信息；提供镜像复制、镜像分享、镜像删除等功能。还可以查看系统镜像信息、自定义镜像信息、分享镜像信息。

2 容器服务

2.1 产品概述

容器服务（Container Service）是一种高性能可伸缩的容器管理服务，支持在一组阿里云云服务器上通过Docker 容器来运行或编排应用。

容器服务免去了您对容器管理集群的搭建，整合了负载均衡、专有网络等云产品，让您可以通过控制台或简单的API（兼容Docker API）进行容器生命周期管理。

2.1.1 容器技术

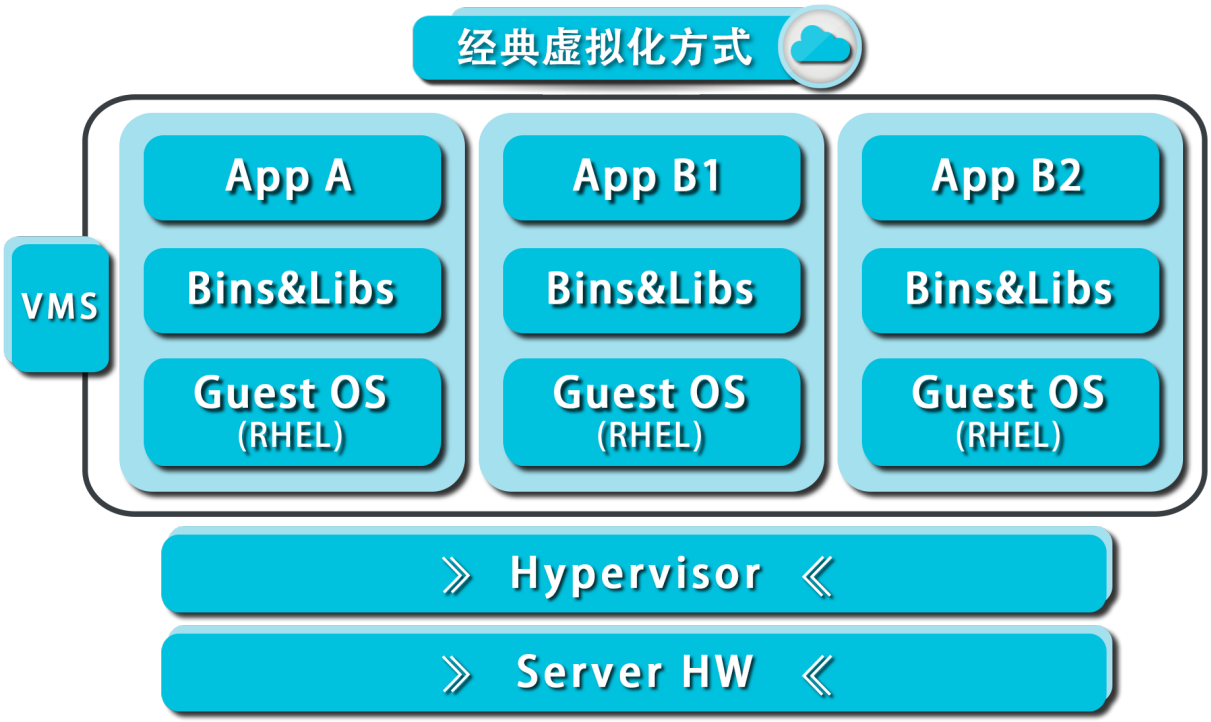
容器技术是一种轻量级的操作系统级虚拟化技术。用户通过容器镜像来交付应用，其中包含了应用程序及所需的运行时依赖。容器镜像具有良好的可移植性，可以在不同环境下保证部署的一致性。容器之间运行时相互隔离，具有相当好的安全性。

容器技术避免了不同应用在同一个环境中可能存在的版本冲突，以及同一个软件在不同环境中可能存在的运行环境不一致的问题。所有容器共享宿主机的操作系统内核，这使得容器比虚拟机更轻量级，可以快速启动，并进行细粒度的资源控制。

容器技术与虚拟化

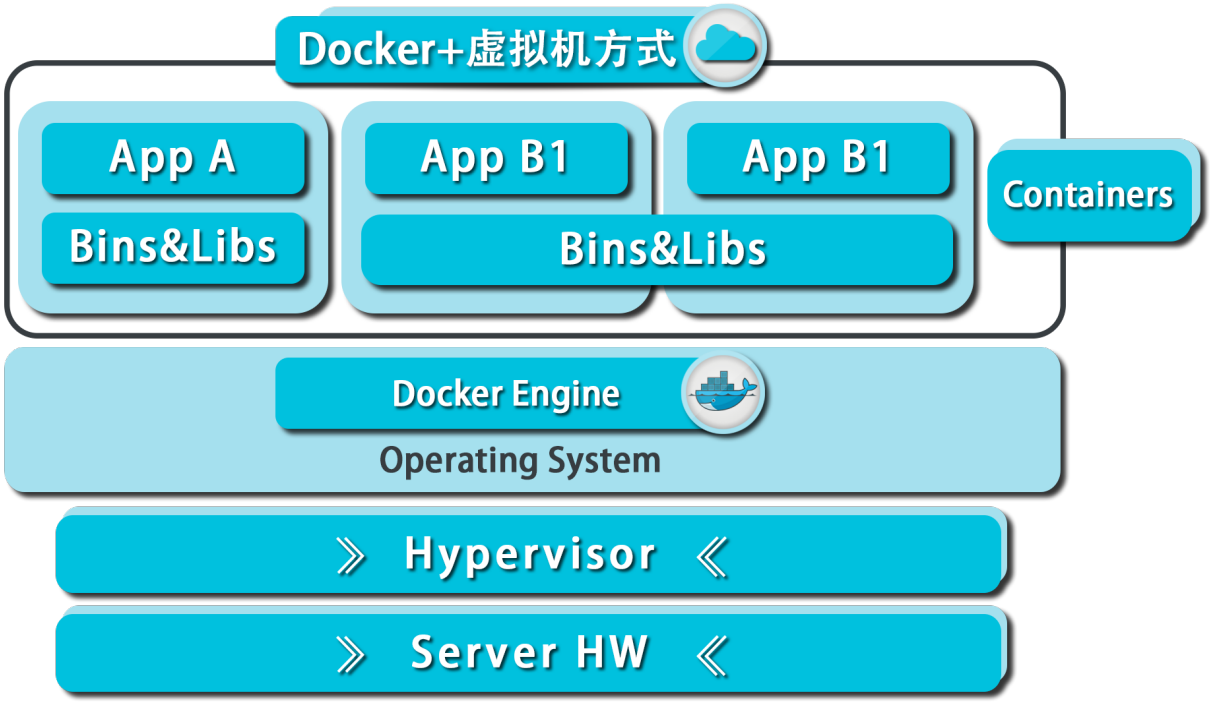
容器技术和传统的虚拟化并不冲突。传统的虚拟化方式是将操作系统到应用的所有要素全部包含在一起，如下图所示。

图 3: 虚拟化



容器只将应用的代码和运行环境打包，镜像分层可以与同样环境下的进行复用，非常简单。

图 4: Docker+虚拟化



结合容器和虚拟化技术，可以利用虚拟机提供弹性基础架构，提供更好的安全隔离，动态热迁移能力；同时还可以利用容器技术实现简化应用部署、运维，实现弹性应用架构。

技术特点

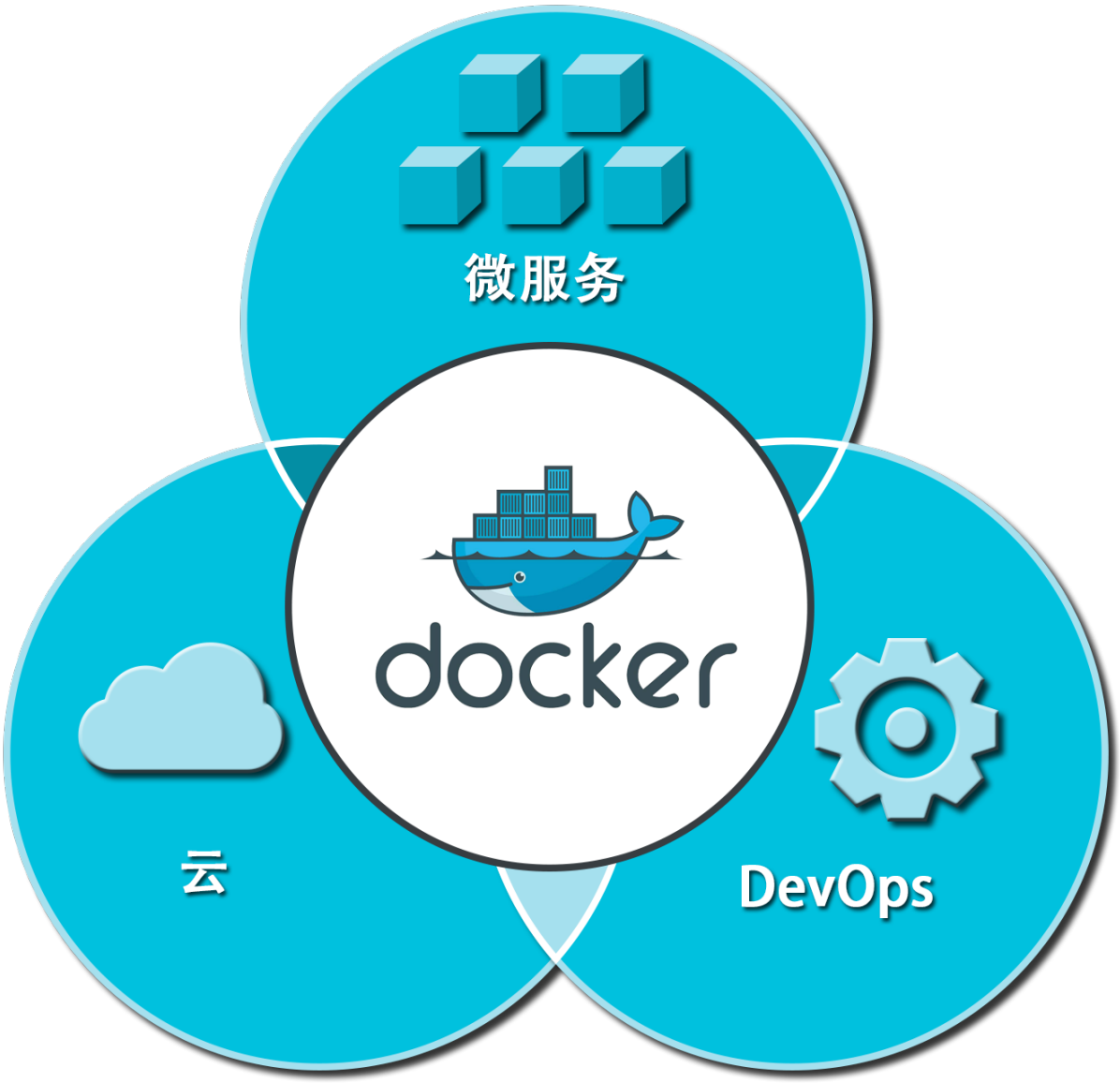
容器技术的特点是：敏捷、可移植、可控。

- **敏捷**：简单快速是容器技术吸引开发人员的重要特性，一致性的交付能力使得软件的交付更快，开发企业更敏捷。
- **可移植性**：开发人员可以把容器化的应用从开发转移到测试，最终到生产环境。在这个过程中同样的镜像运行结构一致。这意味着计算能力可以跨越数据中心边界进行部署，从而让混合云中的计算能力迁移真正可行。
- **可控**：生产环境的应用需要保证 SLA，要有完善的管理能力，安全和监控能力。容器技术应用环境标准化了，开发人员可以利用自动化工具来管理基础架构和应用，保证了所有操作自动化、可控、可回溯。

应用场景

容器技术可以应用在非常多的场景。不过针对这三点要求很高的 DevOps、云应用管理和微服务的应用场景讨论得比较多，研究也比较充分。

图 5: 应用场景



2.2 产品架构

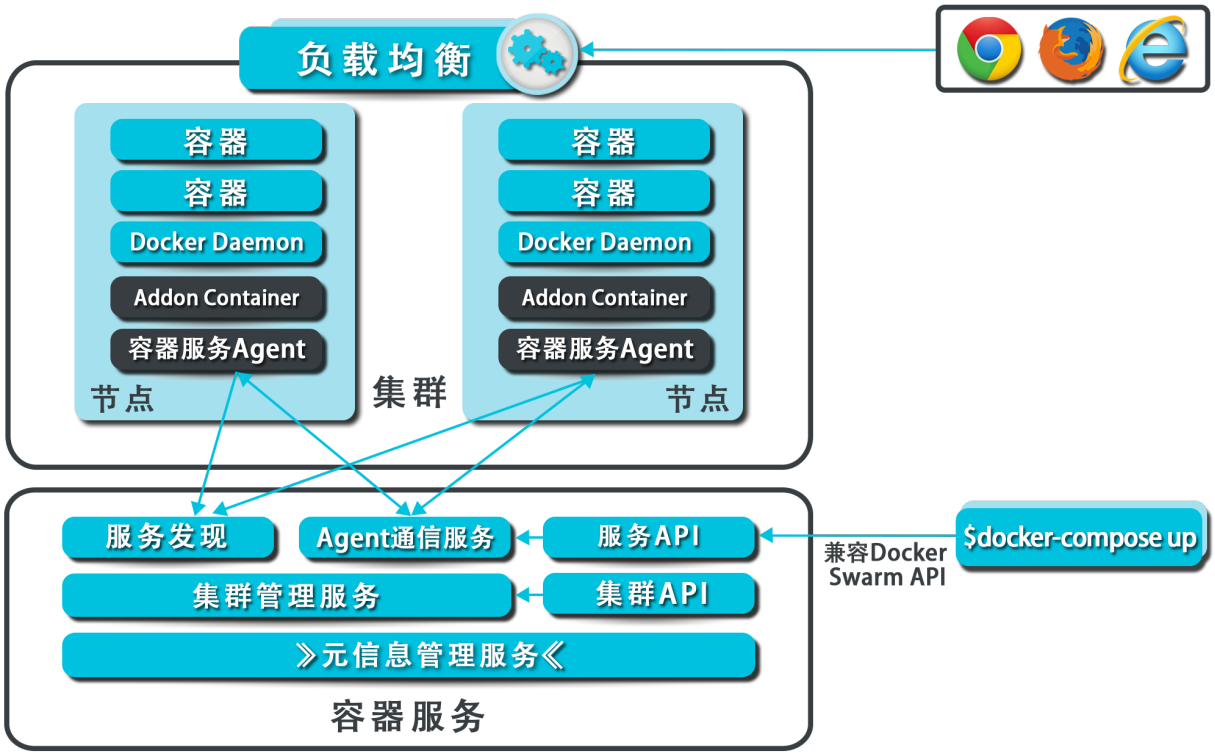
阿里云容器服务完全兼容Docker原生编排基础，兼容Docker Swarm集群管理，全兼容Docker Compose模板编排应用，并在此基础上做了大量的扩展和针对阿里云的深度优化；支持通过图形化界面和Open API管理集群和容器应用。

在底层架构上，您拥有并独占云服务器或物理机，保证了安全可控，支持定制安全组和专有网络VPC 安全规则。

为了保证您的应用的低成本上云，容器服务实现了兼容标准Docker API的程序接口，兼容所有的Docker镜像和Compose模板，支持应用无缝迁云。为第三方能力扩展，实现了灵活可定制的扩展机制。

容器服务的系统架构如下所示。

图 6: 系统架构



其中：

- **集群管理服务**：提供Docker集群管理和调度。
- **服务发现**：提供Docker的状态等元数据存储。
- **Agent通信服务**：提供每台宿主机和集群管理服务之间的通信服务。
- **集群**API：对外暴露阿里云统一的Open API能力。
- **服务**API：对外暴露兼容Docker Swarm的API能力。

容器服务的能力栈如下图所示。容器服务构建在云基础设施之上，和阿里云能力深度整合并支持三方扩展，可以支持不同的应用类型。

图 7: 功能架构

2.3 功能特性

集群管理

- 您可以根据自己的需求，选择不同的地域创建和删除集群。
- 多种服务器托管方式。
- 支持将已创建的云服务器添加到指定集群。

一站式容器生命周期管理

- **网络**：支持跨宿主机容器间互联，支持通过container name或hostname定义的域名互访。
- **存储**：支持数据卷管理，支持OSSFS。
- **日志**：支持日志自动采集。
- **监控**：支持容器级别和VM级别的监控。
- **调度**：支持跨可用区高可用和异常节点的reschedule等策略。
- **路由**：支持4层和7层的请求转发和后端绑定。
- **子账号**：支持集群级别的RAM授权管理。

兼容标准Docker API

兼容标准的Docker Swarm和Docker Compose协议，无缝地将已有系统从线下迁移至云上。

阿里云环境特有的增值能力，更好的体验

- 扩展Compose模板定义，增强生命周期管理。
- 整合负载均衡，提供容器的访问能力。
- 高可用调度策略，轻松打通上下游交付流程。
- 支持服务级别的亲和性策略和横向扩展。
- 支持跨可用区高可用和灾难恢复。
- 支持集群和应用管理的OpenAPI，轻松对接持续集成和私有部署系统。

高效可靠

- 支持海量容器秒级启动。
- 支持容器的异常恢复和自动伸缩。
- 支持跨可用区的容器调度。

3 对象存储OSS

3.1 什么是 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。它可以理解为一个即开即用，无限大空间的存储集群。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。

OSS 将数据文件以对象/文件（object）的形式上传到存储空间（bucket）中。 您可以进行以下操作：

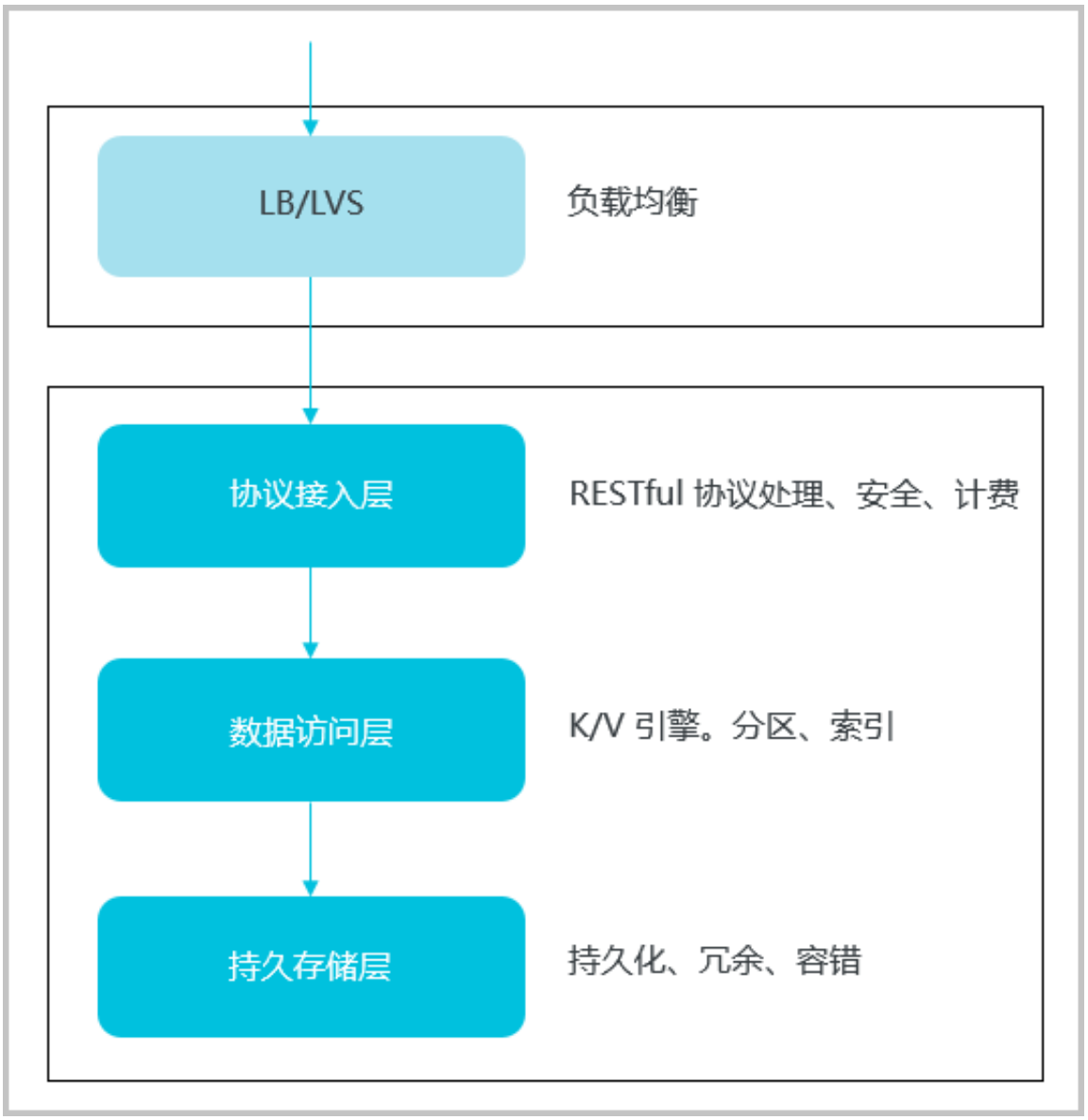
- 创建一个或者多个存储空间
- 每个存储空间中添加一个或多个文件
- 通过获取已上传文件的地址进行文件的分享和下载
- 通过修改存储空间或文件的属性或元信息来设置相应的访问权限
- 通过云控制台执行基本和高级 OSS 任务
- 通过开发工具包 SDK 或直接在应用程序中进行 RESTful API 调用执行基本和高级 OSS 任务。

3.2 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 8: 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不低于99.9% 可用性和99.99999999% 数据可靠性。

3.3 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。

4 消息服务

4.1 产品概述

阿里云消息服务（Message Service，简称 MNS）是一种高效、可靠、安全、便捷、可弹性扩展的分布式消息服务。MNS 能够帮助应用开发者在他们应用的分布式组件上自由地传递数据、通知消息，构建松耦合系统。

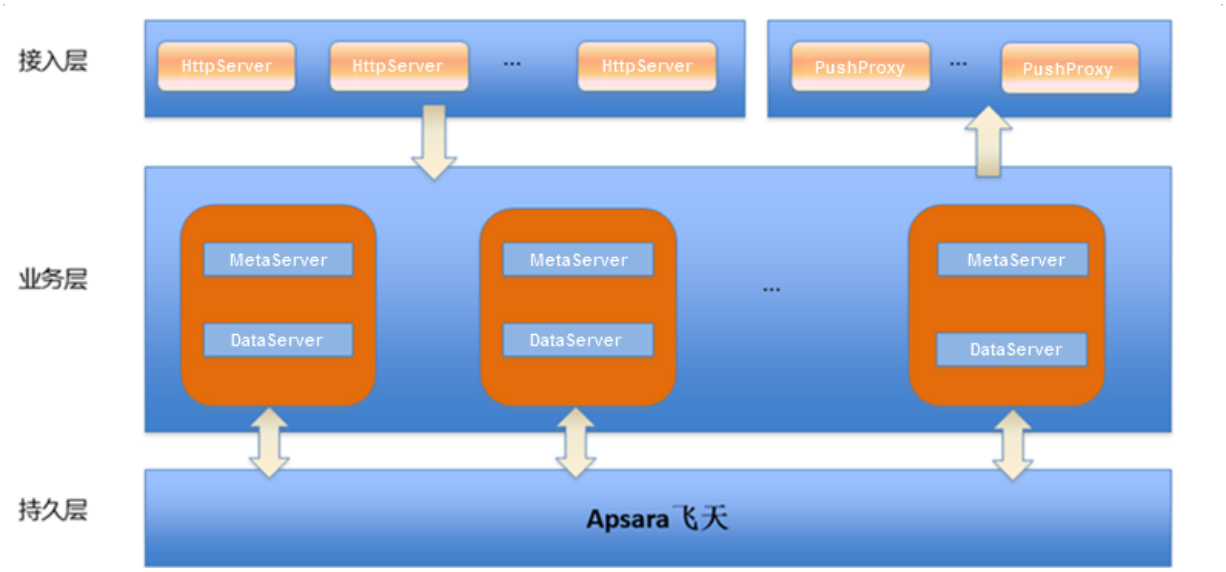
消息服务提供了队列模型，旨在提供高可靠高并发的一对一消费模型，即队列中的每一条消息都只能够被某一个消费者进行消费。队列模型就如同一家旋转寿司店，寿司店中有多个寿司师傅（生产者）在制作精美的寿司（消息），每一份寿司都是独特的，每位顾客（消费者）同时从传送带上拿取中意的寿司进行食用（消费）。

4.2 产品架构

阿里云 MNS 基于阿里云自主研发的飞天分布式存储系统，旨在提供高效、可靠、安全、便捷、可弹性扩展的分布式消息服务。帮助应用开发者在他们应用的分布式组件上自由地传递数据、通知消息，构建松耦合系统。

系统架构

图 9: 系统架构



前端服务器 HttpServer

用户通过发起 HTTP 请求发送到 HttpServer 上的 Tengine 模块，Tengine 模块本地端口转发给 FastCgi 模块，然后 FastCgi 模块再通过一系列验权、鉴权以及数据预处理之后转发到数据服务器。

元数据服务器 MetaServer

元数据服务器 MetaServer 主要用于用户队列的元数据管理，例如队列创建时间、队列修改时间、消息计数等元数据信息。其中各个用户的队列元数据都是依照特定的切分算法分别加载到不同的元数据服务器上。

数据服务器 DataServer

数据服务器 DataServer 主要用于用户队列的数据处理及存储。一般每个队列都由多台数据服务器同时提供服务。

Apsara 飞天分布式系统

元数据服务器以及数据服务器中的所有数据均通过高速通道存入 Apsara 飞天分布式系统，每份数据均持有三份拷贝，确保了用户数据的高可靠性。

4.3 功能特性

丰富的队列属性配置

消息服务提供了丰富的队列属性配置选项，您可以进行队列属性的个性化配置来满足不同的应用场景，支持普通队列、延迟队列、优先级队列等多种队列模式。

支持海量并发访问

支持多个生产者和消费者并发访问同一个队列，并能确保某条消息在取出之后的特定时间段内，无法被其他消费者获得。可以根据业务需求自由伸缩并发访问数。

消息投递保障

在消息有效期内，确保消息至少能被成功消费一次。用户间资源隔离，确保您队列中的消息不会被非法获取。

分布式事务消息

完善的分布式环境下事务消息解决方案。

支持日志管理

可以通过日志管理的方式，查看每一条消息发送、接收和删除的完整生命周期。用户可以通过日志管理，方便的进行问题调查。

支持云监控

用户可以通过云监控查看队列情况，并且可以自定义报警项，当队列情况不符合期望时，能够及时知晓。

5 表格存储TableStore

5.1 什么是表格存储

5.1.1 技术背景

DT 时代下的数据特点

随着移动互联网的普及并深入到各个行业和领域中，互联网应用呈现出如下几个非常显著的特点和趋势：

- 应用需要存储和处理的数据量接近指数级增长，比如微博、社交事件、图片、访问日志等。
- 诸如手机等移动设备的普及以及物联网设备的增加，结构化数据的存储将面临越来越高的写入并发。
- 需要处理的数据没有严格的 schema，更趋向于半结构化，数据的字段会动态的变化。
- 用户的访问存在明显的热点和高峰，比如各种大促期间，应用的用户访问量会在瞬间达到非常高的值。
- 由于移动互联网无时无刻都在接入用户，用户对互联网应用的可用性要求也非常高，很难接受故障导致的服务不稳定甚至是计划中的服务停机。
- 大量的数据信息对计算分析提出了更高的要求。

传统 IT 软件解决方案的挑战

使用传统 IT 软件解决方案很难面对这些新的趋势和挑战，主要体现在如下几个方面：

- 规模可扩展

  传统的软件如关系型数据库很难处理这样快速增长的数据量，不管是在数据写入的吞吐率还是在数据量变大之后的访问效率上，都存在巨大的瓶颈。于是使用传统数据库的方案不得不进行手工和静态的分库分表策略，而这个策略也意味着很大的系统维护代价，特别是在增加节点进行扩容的时候，需要对已有的数据进行重新的切分和迁移，在这个过程中服务的性能、稳定性和可用性都很难得到很好的保证，而且整个过程是非常复杂的。

- 数据模型变更

  传统数据库处理的数据都具有严格的 schema，数据中包含的列数通常是固定的，很少去修改。频繁修改表 schema 和列数的设置，会对服务的可用性产生较大的影响，因此传统方案在面对结构越来越松散的互联网应用的数据时显得很力不从心。

- 快速伸缩

  传统的解决方案中，业务的访问压力是比较平稳的，系统不会经常面临资源需要快速调整（扩容和减容）的情况，因此一旦发生这种情况，就需要很大的代价，比如对数据进行重新切分，对切分之后的数据进行迁移，一旦业务压力下降之后，为了避免资源利用率低的问题，又要对多余的机器进行下线处理，又会经历数据的再一次搬迁，整个过程极其复杂且效率低下。

- 运维保障

  使用传统的软件方案需要专门来处理机器硬件（网络、磁盘）等设备发生故障时的服务恢复，需要处理硬件的更换，需要处理软件的版本升级，配置调优和更新，要让这些过程对应用透明，不影响服务的可用性，需要有专门的运维保障和系统工程师团队才能达到。不管是从人才招聘还是从成本投入上来说，这些工作对于快速发展的企业都是巨大的挑战。

- 计算瓶颈

  在现有的业务系统中，我们通常使用的是 OLTP （OnLine Transaction Processing，联机事务处理）系统来对数据进行处理和分析，如 MySQL、Microsoft SQL Server 等关系数据库系统。这些关系数据库系统擅长事务处理，在数据操作中保持着严格的一致性和原子性，能够很好支持频繁的数据插入和修改，但是，一旦需要进行查询或计算的数据量过大，达到数千万甚至数十亿条，或需要进行的计算非常复杂，OLTP 类数据库系统便力不从心了。

5.1.2 表格存储技术

表格存储是构建在阿里云分布式操作系统飞天之上的 NoSQL 数据存储服务，通过将数据表进行分区并且将数据分区调度到不同的节点上进行服务，从而提供可扩展的能力。在单机的硬件出问题时，表格存储服务通过心跳机制快速发现有问题的节点，并且把该节点上数据分区快速迁移到健康的节点上继续服务，从而达到服务的快速恢复能力。

数据分区和负载均衡

表中每一行主键的第一列称为数据分片键（Partition Key），系统根据数据分片键的范围将表切分为多个分区，这些分区被系统均匀地调度到不同的存储节点上。当单个数据分区内的数据不断增加到一定程度时，分区会自动分裂成两个更小的分区，从而将数据和访问 load 分散到两个分区上，这两个分区会被调度到不同的节点上，从而将访问 load 分散到不同的节点上，最终达到了单表数据规模和访问压力的线性扩展。

技术指标：表格存储支持的单表最大能够到 PB 级别，最多能够提供百万级别的并发读写能力。

单机故障自动恢复

在表格存储的存储引擎中，每个节点都会服务一批不同表的数据分区，这些分区的分布和调度信息由一个 Master 节点来负责管理，并且这个 Master 节点也会监控每个服务节点的健康状态。当发现某个服务节点出现问题时， Master 就会将原先分配给这个服务节点的数据分区调度到其他健康的节点上。由于数据分区的迁移只是逻辑上的迁移，不涉及实际数据的移动，所以当出现单机故障时，服务能够在很短的时间内恢复。

技术指标：单个机器节点的故障只影响部分数据分区的服务，并且能够在分钟级别恢复。

同城及异地灾备

为满足业务对安全性和可用性的需求，表格存储提供了同城及异地双集群主备容灾，容灾粒度为实例级别。容灾实例下的数据表上的数据插入、更新、删除操作会异步复制到备实例下的同名表，主备实例的数据保持一致的时间取决于主备集群的网络环境，在理想的网络环境下，为毫秒级别延时。所以在进行手动切换时，需要先停止对主集群的读写，并等待所有数据备份完成，再切换服务到备集群。主备切换之后，1 个小时内不能再进行主备切换，且需要清理原集群数据并重新设置备集群信息。

同城双集群主备容灾中，应用访问主备集群的表格存储的服务域名不变，即发生切换之后，应用程序无需更改。异地双集群主备容灾中，主备集群的服务域名会不同，在发生切换情况下，应用程序需要更改访问表格存储的域名。

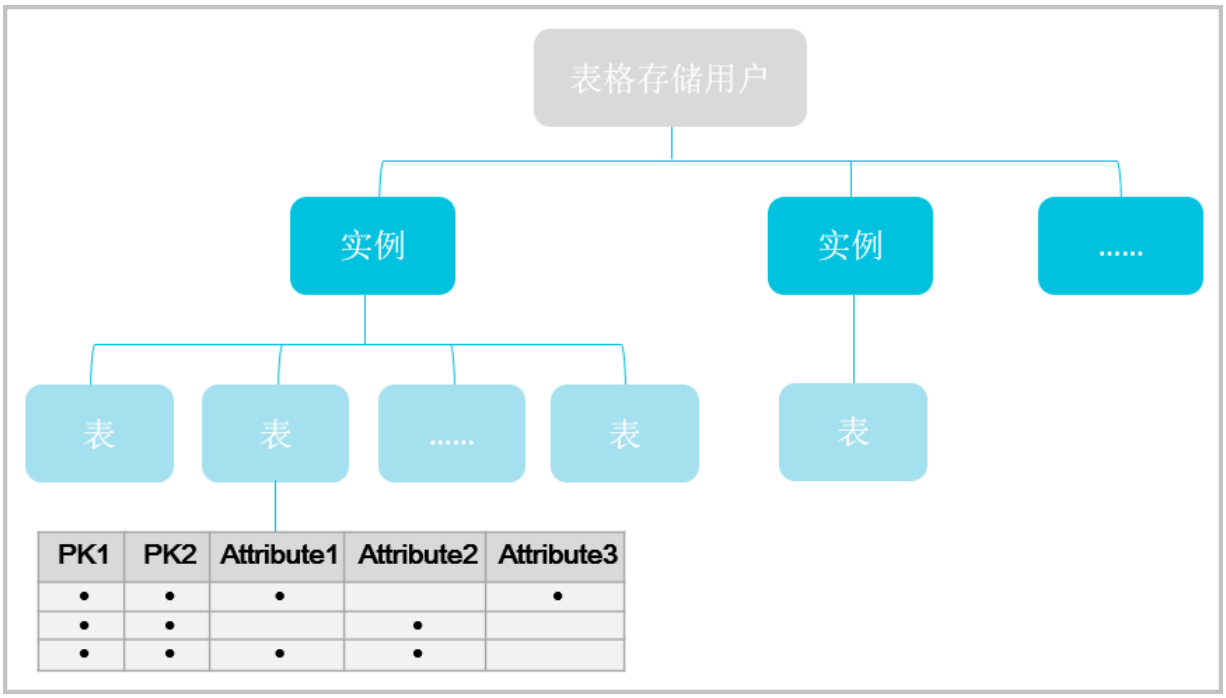
技术指标：RTO 小于 2 分钟，RPO 小于 5 分钟，RCO 为 1。

5.2 功能特性

5.2.1 用户和实例

用户和实例架构图如下图所示。

图 10: 用户和实例架构图

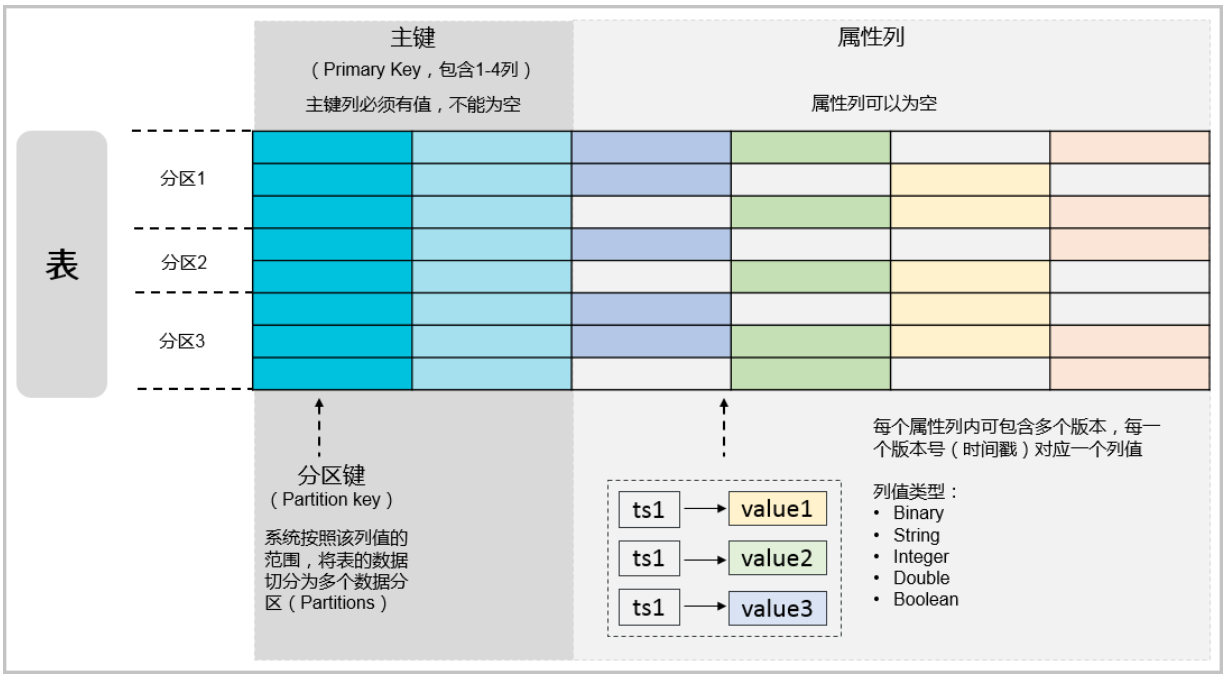


- 通过云账号进行登录。
- 用户的操作均可被细粒度审计。
- 用户通过实例来组织资源，一个用户可以创建多个实例，每个实例可以创建、管理多张数据表。
- 实例是多租户隔离的基本单位。
- 不同的用户可以授予不同的权限。

5.2.2 数据表

数据表结构图如下图所示。

图 11: 数据表结构图



- 数据表是资源分配的最小单元。
- 表是行的集合，行由主键和属性组成。
- 表根据第一个主键列大小对数据进行分片。
- 表中的所有行都必须包含相同数目和名称的主键列。
- 每行包含的属性列的数目、名字和数据类型也可以不同。
- 单表支持最大 1024 列。
- 单表可支持千亿行甚至更多数据。
- 单表数据规模可达到 PB 级别。

5.2.3 数据分片

- 表根据第一个主键列大小对数据进行分片。
- 第一个主键列值在同一个分片范围内的行会被分配到同一个数据分片。
- 表格存储服务会根据特定的规则对分片进行分裂和合并，以达到更好的负载均衡。
- 同一个分片键下的数据建议不超过 1 GB。

5.2.4 表的常用命令与函数

操作表的常用命令

- ListTable：列出实例下所有的表。
- CreateTable：创建数据表。

- DeleteTable：删除表。
- DescribeTable：获取表的属性信息。
- UpdateTable ：更新表的预留读/写吞吐量配置。

操作表中数据的常用函数

- GetRow：读取单行数据。
- PutRow：新插入一行数据。
- UpdateRow：更新一行数据。
- DeleteRow：删除一行数据。
- BatchGetRow：批量读取一张或者多张表的多行数据。
- BatchWriteRow：批量插入、更新、删除一张表或者多张表的多行数据。
- GetRange：读取表中一个范围内的数据。

5.2.5 授权与权限控制

表格存储的权限

结合访问控制服务和专有网络，表格存储支持如下的权限控制：

- 表级别的授权操作。
- API 粒度的权限控制。
- 支持 IP 限制、https、MFA（多因素认证）、访问时间限制等多种鉴权条件。
- 临时授权访问（STS）。
- 支持专有网络（VPC）访问控制。

云控制台

- 支持云平台账号登录与鉴权。
- 提供图形化的实例创建、管理和删除的功能。
- 提供图形化的数据表的创建、管理、调整预留读写吞吐量和删除的功能。
- 提供表级别的监控信息展示。

5.3 产品优势

表格存储（Table Store）是构建在阿里云飞天分布式系统之上的 NoSQL 数据存储服务，提供海量结构化数据的存储和实时访问。表格存储以实例和表的形式组织数据，通过数据分片和负载均衡技术，达到规模的无缝扩展。表格存储向应用程序屏蔽底层硬件平台的故障和错误，能自动从各类错

误中快速恢复，提供非常高的服务可用性。表格存储管理的数据全部存储在 SSD 中并具有多个备份，提供了快速的访问性能和极高的数据可靠性。用户在使用表格存储服务时，只需要按照预留和实际使用的资源进行付费，无需关心数据库的软硬件升级维护、集群缩容扩容等复杂问题。

表格存储有如下特点：

- **扩展性**

  表格存储的表数据量没有上限，随着表数据量的不断增大，表格存储会进行数据分区调整从而为该表配置更多的存储并提供更高的并发访问能力。

- **数据可靠性**

  表格存储通过存储多个数据备份及备份失效时的快速恢复，提供极高的数据可靠性，数据可靠性为 99.99999999%。

- **高可用性**

  通过自动的故障检测和数据迁移，表格存储对应用屏蔽了机器和网络的硬件故障，提供高可用性，服务可用性为 99.9%。

- **管理便捷**

  应用程序无需关心数据分区的管理、软硬件升级、配置更新、集群扩容等繁琐运维任务。

- **访问安全性**

  表格存储提供多种权限管理机制，并对应用的每一次请求都进行身份认证和鉴权，以防止未授权的数据访问，确保数据访问的安全性。

- **强一致性**

  表格存储保证数据写入强一致，写操作一旦返回成功，应用就能立即读到最新的数据。

- **灵活的数据模型**

  表格存储的表无固定格式要求，每行的列数可以不相同，支持多种数据类型，如 Integer、Boolean、Double、String、Binary。

6 云数据库RDS版

6.1 产品概述

阿里云关系型数据库（Relational Database Service，简称RDS）是一种稳定可靠、可弹性伸缩的在线数据库服务。基于阿里云分布式文件系统和高性能存储，云数据库提供了容灾、备份、恢复、监控、迁移等方面的全套解决方案，彻底解决数据库运维的烦恼。

云数据库MySQL版基于Alibaba的MySQL源码分支，经过双11高并发、大数据量的考验，拥有优良的性能和吞吐量。除此之外，云数据库MySQL版还拥有经过优化的读写分离、数据压缩、智能调优等高级功能。

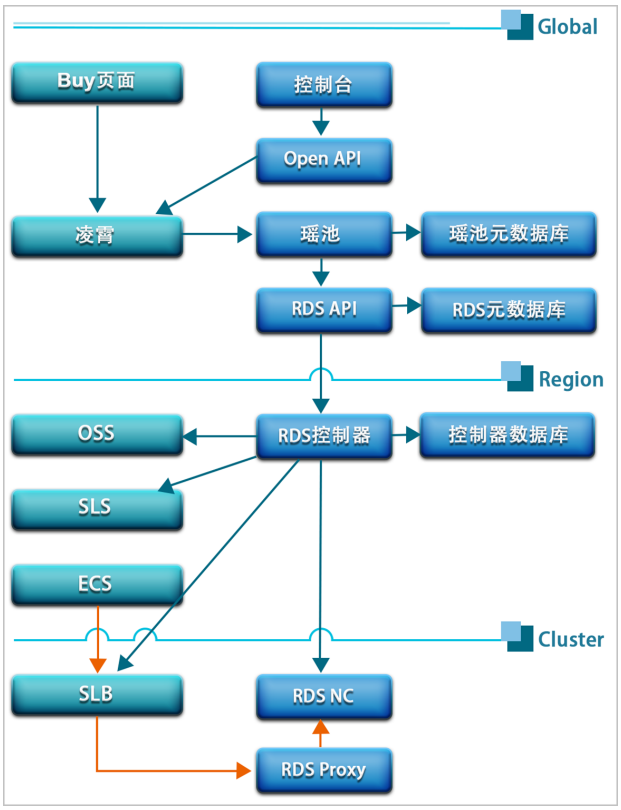
MySQL是全球最受欢迎的开源数据库，作为开源软件组合LAMP（Linux + Apache + MySQL + Perl/PHP/Python）中的重要一环，广泛应用于各类应用。

Web2.0时代，风靡全网的社区论坛软件系统Discuz和博客平台Wordpress均基于MySQL实现底层架构。Web3.0时代，阿里巴巴、Facebook、Google等大型互联网公司都采用更为灵活的MySQL构建了成熟的大规模数据库集群。

6.2 产品架构

云数据库RDS版的系统架构如下。

图 12: RDS系统架构



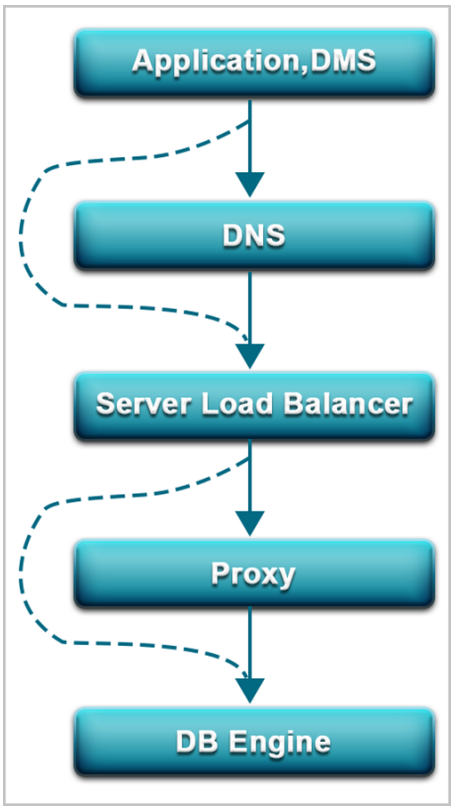
6.3 功能特性

高可用服务RDS主要包括6大核心服务：数据链路服务、调度服务、备份服务、高可用服务、监控服务、迁移服务。

6.3.1 数据链路服务

数据链路服务主要提供数据操作，包括表结构和数据的增删改查。

图 13: RDS 数据链路服务



6.3.1.1 DNS

DNS模块提供域名到IP的动态解析功能，以便屏蔽RDS实例IP地址变化带来的影响。

举例来说：

某RDS实例的域名为test.rds.aliyun.com，而这个域名对应的IP地址为10.1.1.1。某程序连接池中配置为test.rds.aliyun.com或10.1.1.1，都可以正常访问RDS实例。

当该RDS实例发生了可用区迁移或者版本升级后，IP地址就可能变为10.1.1.2。如果程序连接池中配置的是test.rds.aliyun.com，仍然可以正常访问RDS实例。如果程序连接池中配置的是10.1.1.1，就无法访问RDS实例了。

6.3.1.2 SLB

SLB模块提供实例IP地址（包括内网和外网IP），以便屏蔽物理服务器变化带来的影响。

举例来说：

某RDS实例的内网IP地址为10.1.1.1，对应的Proxy或者DB Engine运行在192.168.0.1上。在正常情况下，SLB模块会将访问10.1.1.1的流量重定向到192.168.0.1上。

当192.168.0.1发生了故障，处于热备状态的192.168.0.2接替了192.168.0.1的工作。此时SLB模块会将访问10.1.1.1的流量重定向到192.168.0.2上，RDS实例仍旧正常提供服务。

6.3.1.3 Proxy

Proxy模块提供数据路由、流量探测和会话保持等功能，该模块还在不断发展中。

- 数据路由功能：支持大数据场景下的分布式复杂查询聚合和相应的容量管理。
- 流量探测功能：降低SQL注入的风险，在必要情况下支持SQL日志的回溯。
- 会话保持功能：解决故障场景下的数据库连接中断问题。

6.3.1.4 DB Engine

RDS全面支持主流的数据库协议，具体情况如下表所示：

表 2: RDS支持的数据库协议

RDBMS	Version
MySQL	5.6（含只读实例）
MS SQLServer	2008R2
PostgreSQL	9.4
PPAS	9.3
ORACLE	SQL语法和存储过程

6.3.1.5 DMS

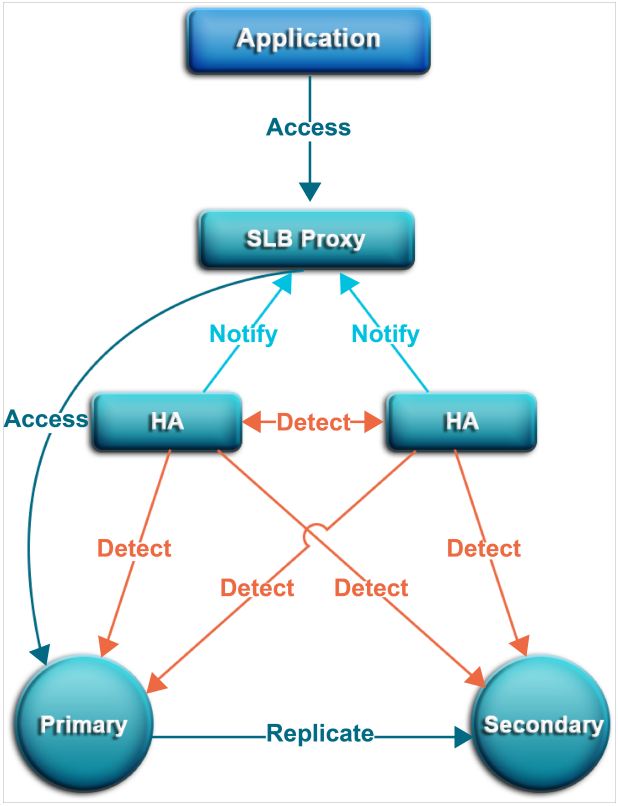
DMS（Data Management Service，简称DMS）是一个访问管理云端数据的Web服务，提供了数据管理、对象管理、数据流转和实例管理等功能。目前支持MySQL、MS SQLServer、PostgreSQL和ADS等数据源。

6.3.2 高可用服务

高可用服务主要保障数据链路服务的可用性，除此之外还负责处理数据库内部的异常。

另外，高可用服务由多个 HA 节点提供，本身具有高可用的特点。

图 14: RDS 高可用服务



6.3.2.1 Detection

Detection模块负责检测DB Engine的主节点和备节点是否提供了正常的服务。

通过间隔为8-10秒的心跳信息，HA节点可以轻易获得主节点的健康情况。再结合备节点的健康情况和其他HA节点的心跳信息，Detection模块可以排除网络抖动等异常引入的误判风险，在30秒内完成异常切换操作。

6.3.2.2 Repair

Repair模块负责维护DB Engine的主节点和备节点之间的复制关系，还会修复主节点或者备节点在日常运行中出现的错误。如：

- 主备复制异常断开的自动修复
- 主备节点表级别损坏的自动修复
- 主备节点 Crash 的现场保存和自动修复

6.3.2.3 Notice

Notice模块负责将主备节点的状态变动通知到SLB或者Proxy，保证用户访问正确的节点。

举例来说：

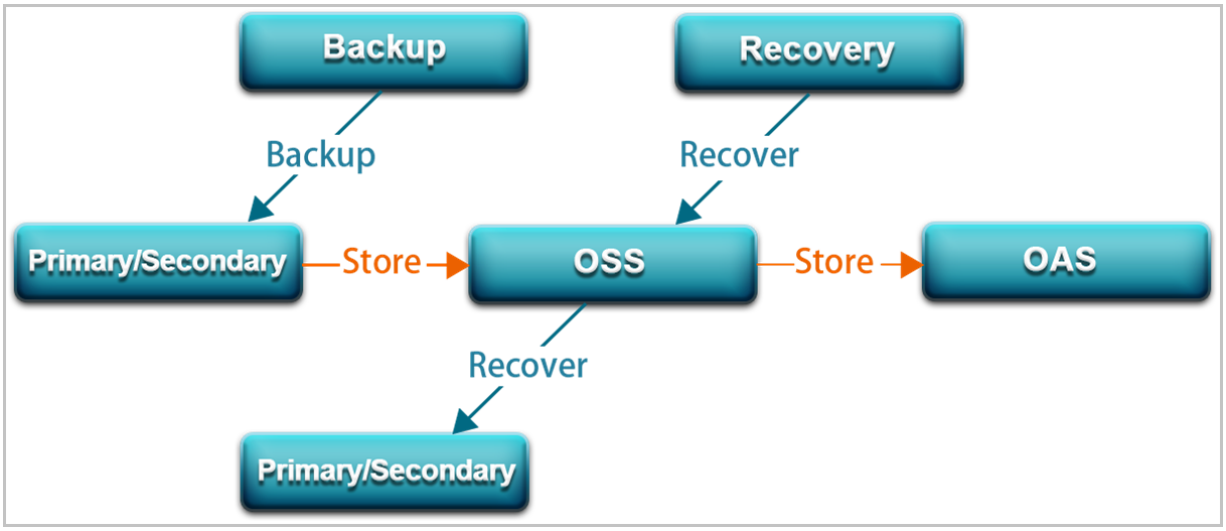
Detection模块发现主节点异常，并通知Repair模块进行修复。Repair模块进行了尝试后无法修复主节点，通知Notice进行流量切换。Notice模块将切换请求转发至SLB或者Proxy，此时用户流量全部指向备节点。

与此同时，Repair在别的物理服务器上重建了新的备节点，并将变动同步给Detection模块。Detection模块开始重新检测实例的健康状态，并通过。

6.3.3 备份服务

备份服务主要提供数据的离线备份、转储和恢复。

图 15: RDS备份服务



6.3.3.1 Backup

Backup模块负责将主备节点上面的数据和日志压缩并上传到OSS上面。在备节点正常运作的情况下，备份总是在备节点上面发起，以避免对主节点的服务带来冲击；在备节点不可用或者损坏的情况下，Backup模块会通过主节点创建备份。

6.3.3.2 Recovery

Recovery模块负责将OSS上面的备份文件恢复到目标节点上。

- 回滚主节点功能：客户发起数据相关的误操作后可以通过回滚功能按时间点恢复数据。
- 修复备节点功能：在备节点出现不可修复的故障时自动新建备节点来降低风险。
- 创建只读实例功能：通过备份来创建只读实例。

6.3.3.3 Storage

Storage模块负责备份文件的上传、转储和下载。

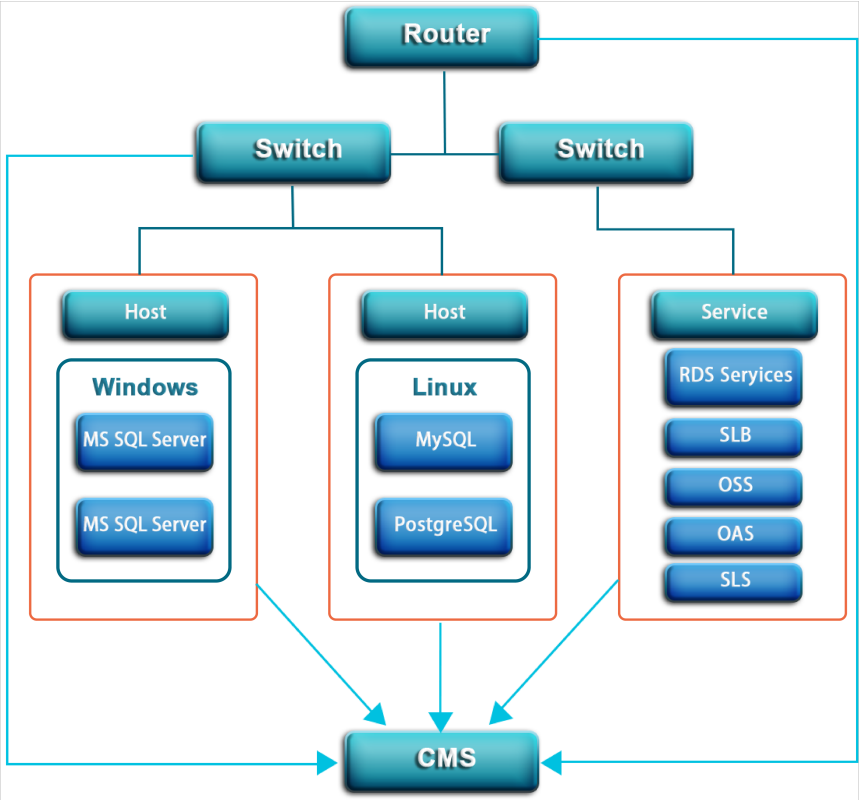
目前备份数据全部上传至OSS进行存储，客户可以根据需要获取临时链接来下载。

在某些特定场景下，Storage模块支持将OSS上面的备份文件转储至归档存储来提供更长时间和更低费用的离线存储。

6.3.4 监控服务

监控服务主要提供服务、网络、操作系统和实例层面的状态跟踪。

图 16: RDS监控服务



6.3.4.1 Service

Service模块负责服务级别的状态跟踪。

举例来说：

Service模块会监控SLB、OSS等RDS依赖的其他云产品是否正常，包括功能和响应时间等。另外对RDS内部的服务，Service也会通过日志来判定是否正常运作。

6.3.4.2 Network

Network模块负责网络层面的状态跟踪。

举例来说：

- ECS与RDS之间的连通性监控。
- RDS物理机之间的连通性监控。
- 路由器和交换机的丢包率监控。

6.3.4.3 OS

OS模块负责硬件和OS内核层面的状态跟踪。

举例来说：

- 硬件检修：OS模块会不断检测CPU、内存、主板、存储等设备的工作状态，并预判是否会发生故障，并提前进行自动报修。
- OS内核监控：OS模块会跟踪数据库的所有调用，并从内核态分析调用缓慢或者出错的原因。

6.3.4.4 Instance

Instance模块负责RDS实例级别的信息采集。

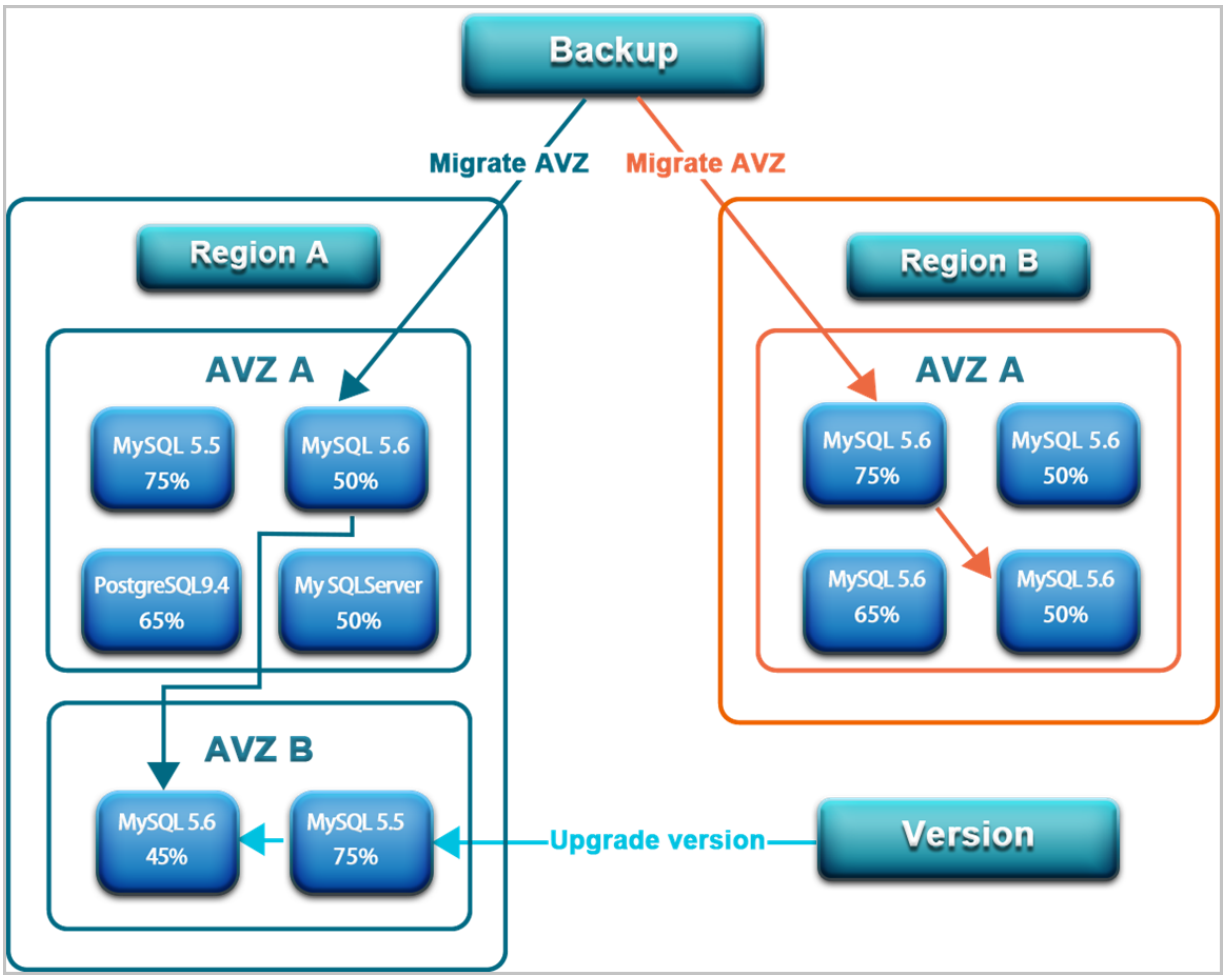
举例来说：

- 实例的可用信息。
- 实例的容量和性能指标。
- 实例的SQL执行记录。

6.3.5 调度服务

调度服务主要提供资源调配和实例版本管理。

图 17: RDS调度服务



6.3.5.1 Resource

Resource模块主要负责RDS底层资源的分配和整合，对用户而言就是实例的开通和迁移。

举例来说：

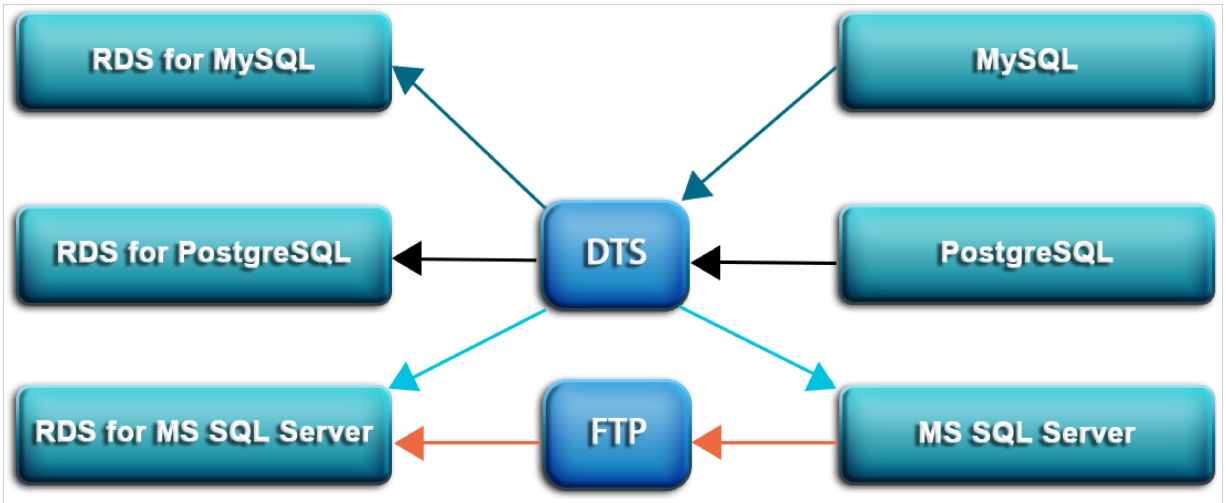
用户通过控制台或者OpenAPI创建实例，Resource模块会计算出最适合的物理服务器来承载流量。RDS实例跨可用区迁移的情况类似。

在经过长时间的实例创建、删除和迁移后，Resource模块会计算可用区内的资源碎片化程度，并定期发起资源整合以提高可用区的服务承载量。

6.3.6 迁移服务

调度服务主要帮助用户把数据从自建数据库迁移到RDS里面。

图 18: RDS 迁移服务



6.3.6.1 FTP

FTP模块主要负责RDS for MS SQL Server的全量迁移上云。

用户在自建的MS SQL Server数据库上进行一次备份后，可以通过FTP客户端将备份文件上传至RDS提供的专用FTP上，FTP模块会将备份还原到指定的RDS实例上。

7 云数据库Redis版

7.1 产品概述

阿里云数据库 Redis 版（ApsaraDB for Redis）是兼容开源 Redis 协议的 Key-Value 类型在线存储服务。它支持字符串（String）、链表（List）、集合（Set）、有序集合（SortedSet）、哈希表（Hash）等多种数据类型，及事务（Transactions）、消息订阅与发布（Pub/Sub）等高级功能。通过“内存+硬盘”的存储方式，云数据库Redis版在提供高速数据读写能力的同时满足数据持久化需求。

除此之外，云数据库 Redis 版作为云计算服务，其硬件和数据部署在云端，有完善的基础设施规划、网络安全保障、系统维护服务。所有这些都无需用户考虑，确保用户专心致力于自身业务创新。

7.2 功能特性

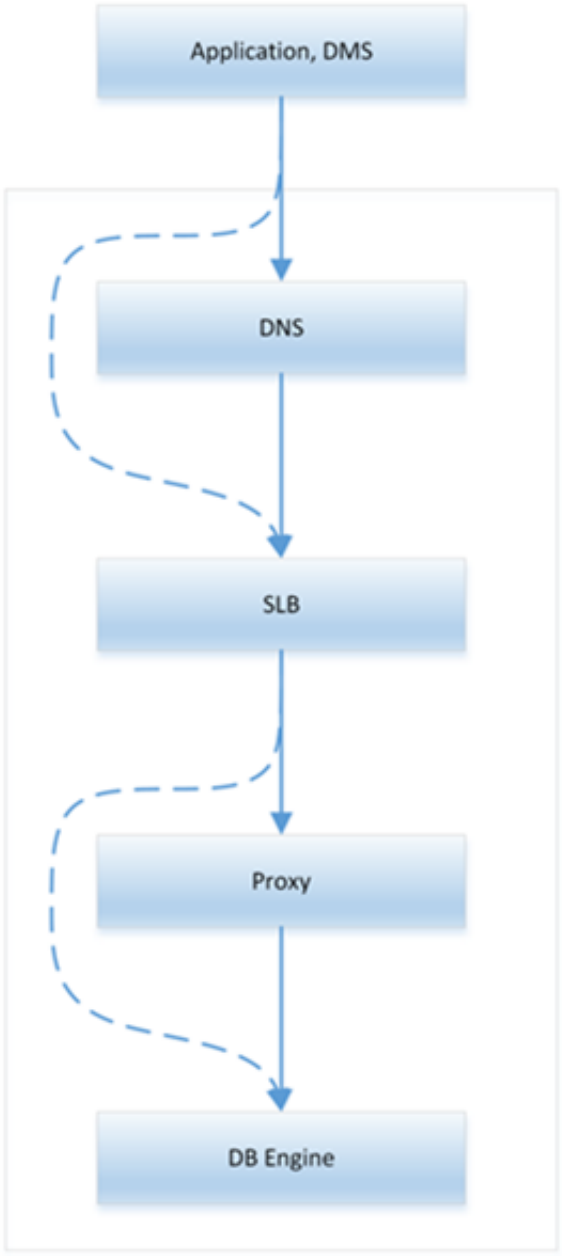
高可用服务Redis主要包括6大核心服务：

- 数据链路服务
- 调度服务
- 备份服务
- 高可用服务
- 监控服务
- 迁移服务

7.2.1 数据链路服务

数据链路服务主要提供数据操作，包括数据的增删改查。

您可以通过应用程序连接Redis服务，也可以通过阿里云Redis提供的数据管理工具（DMS）进行图形化数据管理。



7.2.1.1 DNS

DNS模块提供域名到IP的动态解析功能，以便屏蔽RDS实例IP地址变化带来的影响。

举例来说：

某Redis实例的域名为**test.kvstore.aliyun.com**，而这个域名对应的IP地址为**10.1.1.1**。

某程序连接池中配置为**test. kvstore.aliyun.com** 或**10.1.1.1**，都可以正常访问云数据库Redis实例。

当该云数据库Redis实例发生了跨机故障迁移或者版本升级后，IP地址就可能变为**10.1.1.2**。

如果程序连接池中配置的是**test.rds.aliyun.com**，仍然可以正常访问云数据库Redis实例。

如果程序连接池中配置的是**10.1.1.1**，就无法访问实例了。

7.2.1.2 SLB

SLB模块提供实例IP地址，以便屏蔽物理服务器变化带来的影响。

举例来说：

某云数据库Redis实例的内网IP地址为**10.1.1.1**，对应的Proxy或者DB Engine运行在**192.168.0.1**上。在正常情况下，SLB模块会将访问**10.1.1.1**的流量重定向到**192.168.0.1**上。

当**192.168.0.1**发生了故障，处于热备状态的**192.168.0.2**接替了**192.168.0.1**的工作。此时SLB模块会将访问**10.1.1.1**的流量重定向到**192.168.0.2**上，Redis实例仍旧正常提供服务。

7.2.1.3 Proxy

Proxy模块提供数据路由、流量探测和会话保持等功能，该模块还在不断发展中。

- 数据路由功能：Redis支持集群版架构，可进行分布式路由复杂查询，分区策略。
- 流量探测功能：降低利用Redis漏洞进行网络攻击的风险。
- 会话保持功能：解决故障场景下的数据库连接中断问题。

7.2.1.4 DB Engine

阿里云云数据库Redis版支持标准协议：

引擎	Version
Redis	兼容2.8和3.0geo

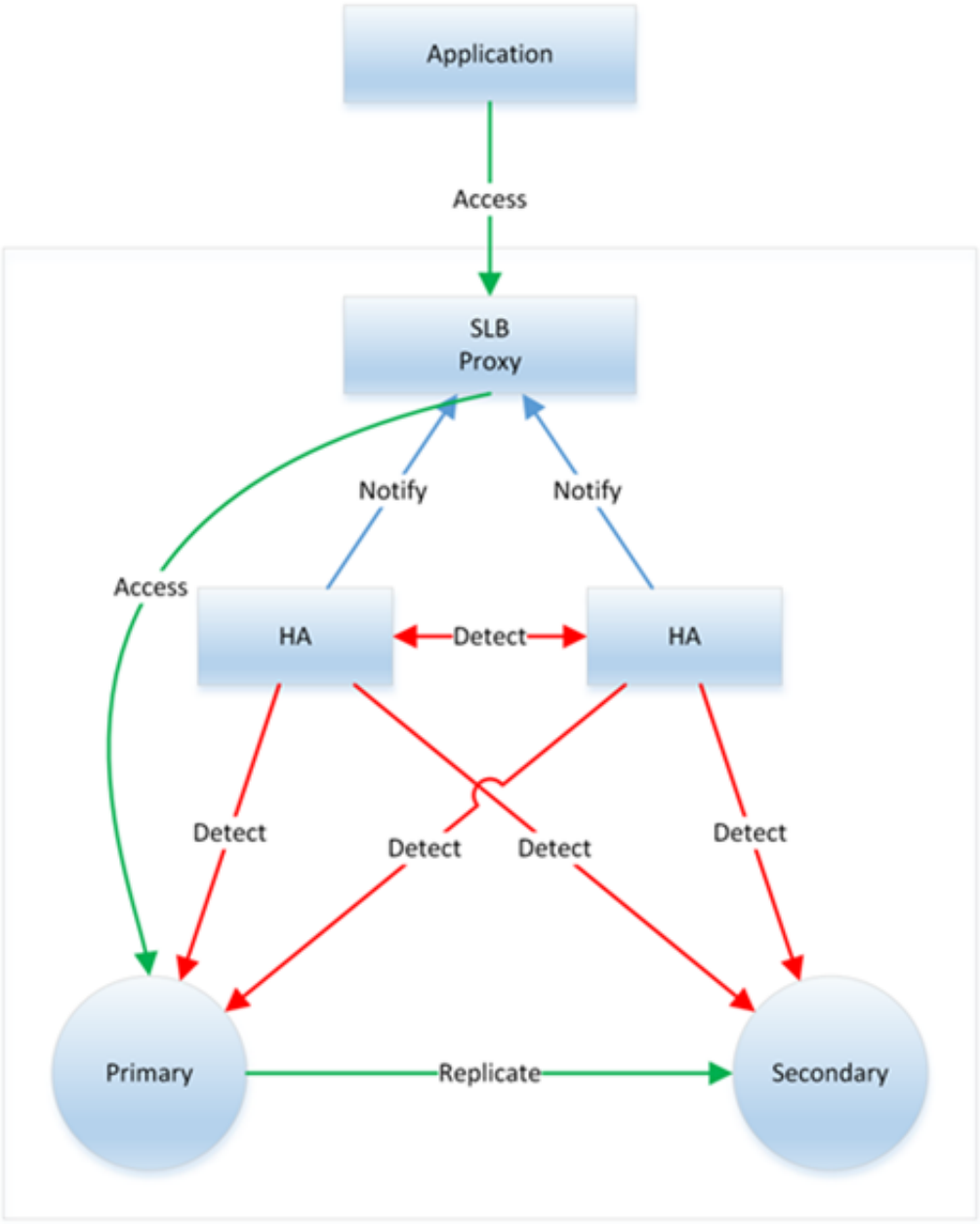
7.2.1.5 DMS

DMS（Data Management Service，简称DMS）是一个访问管理云端数据的Web服务，提供了数据管理、对象管理、数据流转和实例管理等功能。

7.2.2 高可用服务

高可用服务主要保障数据链路服务的可用性，除此之外还负责处理数据库内部的异常。

另外，高可用服务由多个HA节点提供，本身具有高可用的特点。



7.2.2.1 Detection

Detection模块负责检测DB Engine的主节点和备节点是否提供了正常的服务。

通过间隔为8秒～10秒的心跳信息，HA节点可以轻易获得主节点的健康情况。再结合备节点的健康情况和其他HA节点的心跳信息，Detection模块可以排除网络抖动等异常引入的误判风险，在30秒内完成异常切换操作。

7.2.2.2 Repair

Repair模块负责维护DB　Engine的主节点和备节点之间的复制关系，还会修复主节点或者备节点在日常运行中出现的错误。如：

- 主备复制异常断开的自动修复
- 主备节点表级别损坏的自动修复
- 主备节点Crash的现场保存和自动修复

7.2.2.3 Notice

Notice模块负责将主备节点的状态变动通知到SLB或者Proxy，保证用户访问正确的节点。

举例来说：

Detection模块发现主节点异常，并通知Repair模块进行修复。Repair模块进行了尝试后无法修复主节点，通知Notice进行流量切换。Notice模块将切换请求转发至SLB或者Proxy，此时用户流量全部指向备节点。

与此同时，Repair在别的物理服务器上重建了新的备节点，并将变动同步给Detection模块。Detection模块开始重新检测实例的健康状态，并通过。

7.2.3 监控服务

监控服务主要提供服务、网络、操作系统和实例层面的状态跟踪。

7.2.3.1 服务层面监控

单独的Service模块负责服务层面监控。

举例来说：

Service模块会监控SLB等云数据库Redis版依赖的其他云产品是否正常，包括功能和响应时间等。

7.2.3.2 网络层面监控

网络层面的Network模块负责网络层面的状态跟踪。

举例来说：

ECS与云数据库Redis之间的连通性监控。

云数据库Redis物理机之间的连通性监控。

路由器和交换机的丢包率监控。

7.2.3.3 操作系统层面监控

操作系统OS模块负责硬件和OS内核层面的状态跟踪。

举例来说：

硬件检修：OS模块会不断检测CPU、内存、主板、存储等设备的工作状态，并预判是否会发生故障，并提前进行自动报修。

OS内核监控：OS模块会跟踪数据库的所有调用，并从内核态分析调用缓慢或者出错的原因。

7.2.3.4 实例层面监控

实例层面监控的Instance模块负责云数据库Redis实例级别的信息采集。

举例来说：

实例的可用信息。

实例的容量和性能指标。

7.2.4 调度服务

调度服务主要提供资源调配，主要负责云数据库Redis底层资源的分配和整合，对用户而言就是实例的开通和迁移。

举例来说：

用户通过控制台创建实例，调度服务会计算出最适合的物理服务器来承载流量。

在经过长时间的实例创建、删除和迁移后，调度服务会计算可用区内的资源碎片化程度，并定期发起资源整合以提高可用区的服务承载量。

8 负载均衡SLB

8.1 产品概述

负载均衡（Server Load Balancer）是将访问流量根据转发策略分发到后端多台云服务器（Elastic Compute Service，简称 ECS） 的流量分发控制服务。通过流量分发扩展应用系统对外的服务能力，通过消除单点故障提升应用系统的可用性。

负载均衡服务通过设置虚拟服务地址，将添加的ECS虚拟成一个高性能、高可用的应用服务池。根据应用指定的方式，将来自客户端的网络请求分发到云服务器池中。

该服务地址都是每个负载均衡实例独占的，更改转发策略不会导致负载均衡服务地址的变更。您可以将域名解析到负载均衡的服务地址，对外提供服务。除非必要，不建议您删除负载均衡服务。删除了负载均衡服务以后，相应的服务配置和服务地址将会被释放掉，数据一旦删除，不可恢复。

负载均衡服务由以下三个部分组成：

- **负载均衡实例**（Load Balancer）

  如果您想使用负载均衡服务，必须先创建一个负载均衡实例。一个负载均衡实例可以添加多个监听和后端服务器。

- **监听器**（Listener）

  在使用负载均衡服务前，您必须为负载均衡实例添加一个监听，指定监听规则和转发策略，并配置健康检查。

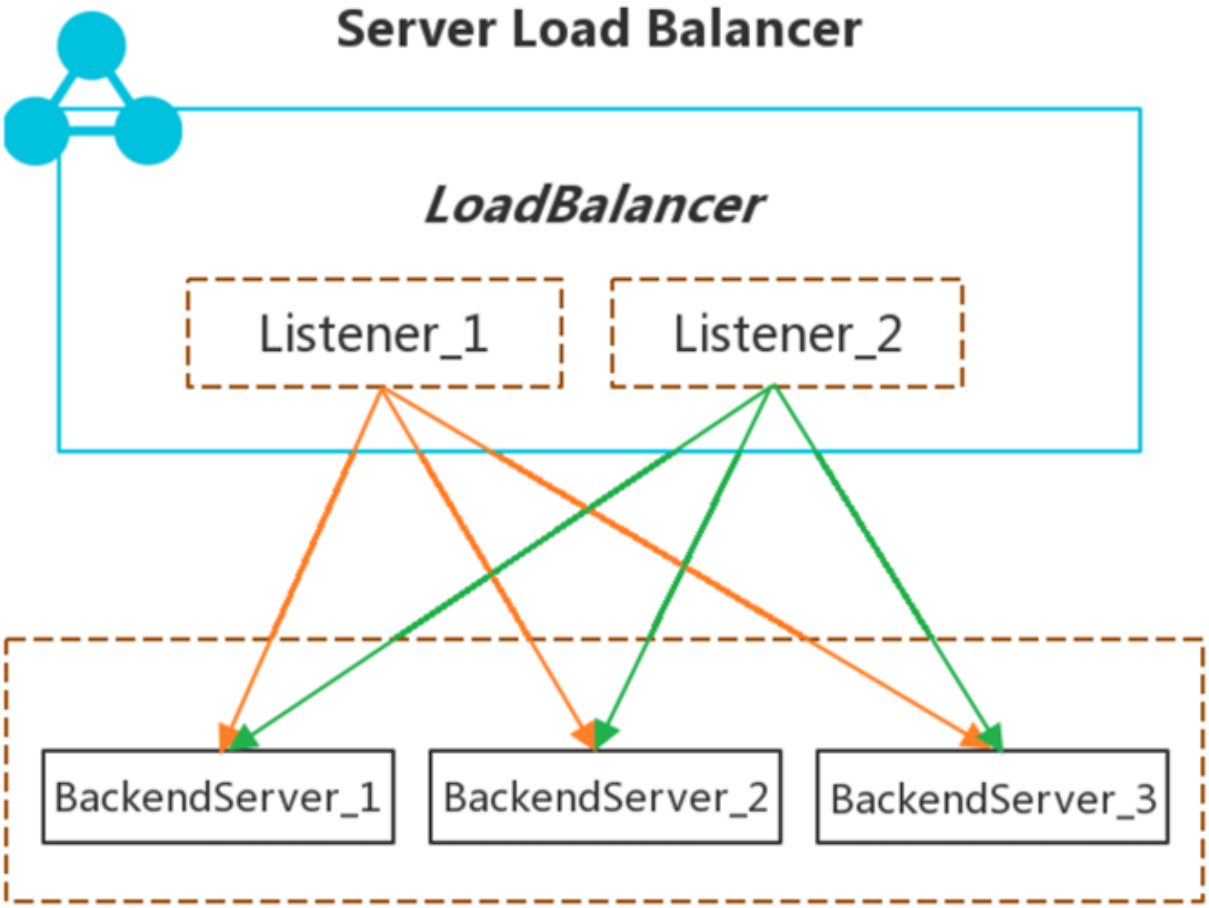
  针对不同的需求，您可以配置四层（TCP/UDP）或七层（HTTP/HTTPS）监听。

- **后端服务器**（Backend Server）

  一组接收前端请求的ECS实例。

如下图所示，来自客户端的请求经过负载均衡实例后，系统根据配置的监听规则，将请求转发到对应的后端ECS实例上。

图 19: 负载均衡构成

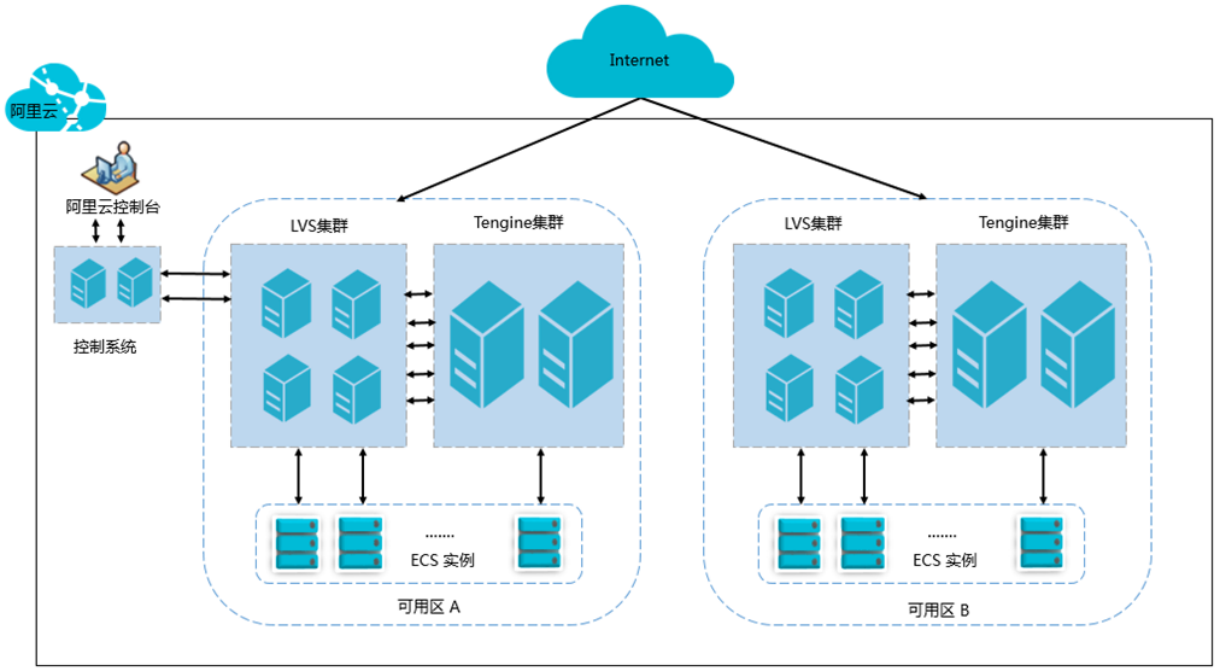


8.2 产品架构

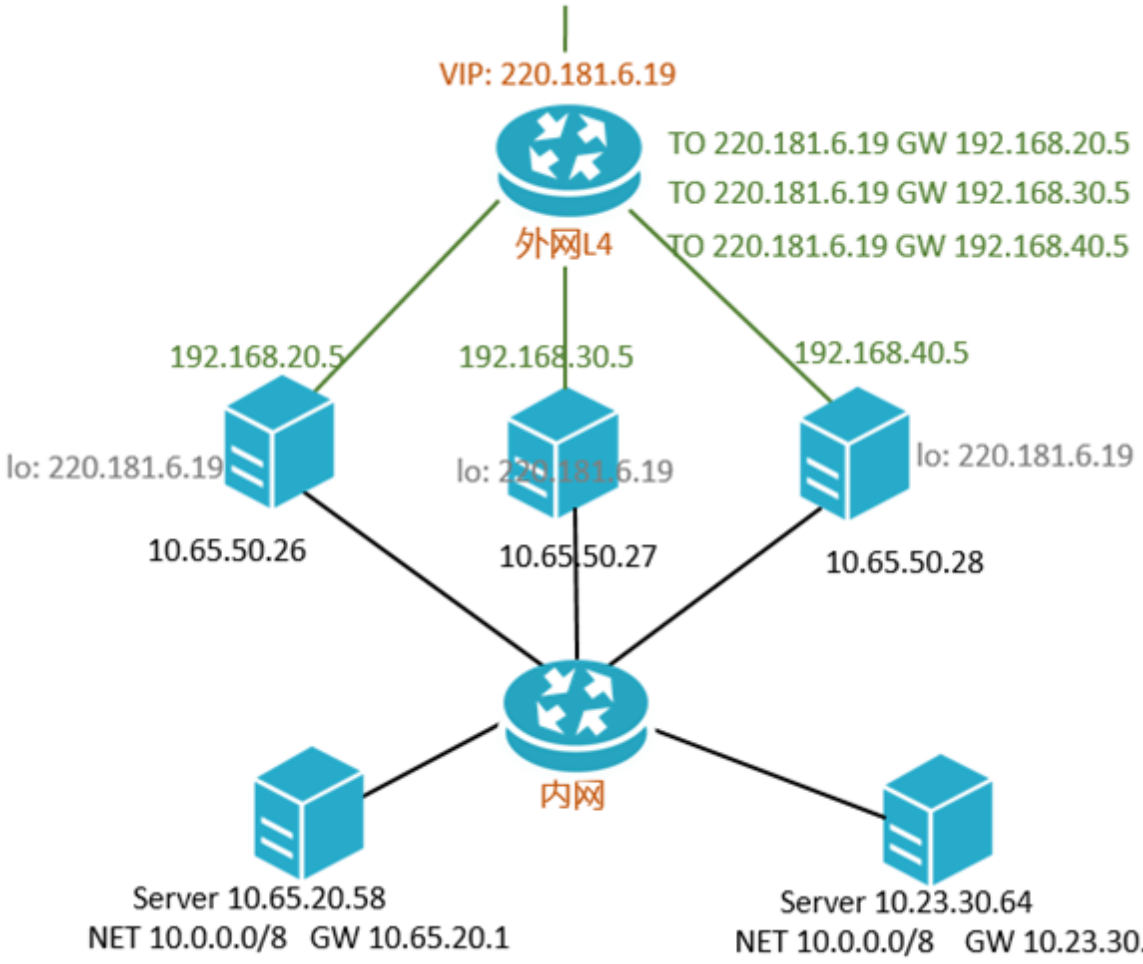
负载均衡采用集群部署，可实现会话同步，以消除服务器单点，提升冗余，保证服务稳定。专有云当前提供四层（TCP协议和UDP协议）和七层（HTTP和HTTPS协议）的负载均衡服务。

- 四层采用开源软件 LVS（Linux Virtual Server）+ keepalived的方式实现负载均衡，并根据云计算需求对其进行了定制化。
- 七层采用Tengine实现负载均衡。Tengine是由淘宝网发起的Web服务器项目，它在Nginx的基础上，针对大访问量网站的需求，添加了很多高级功能和特性。

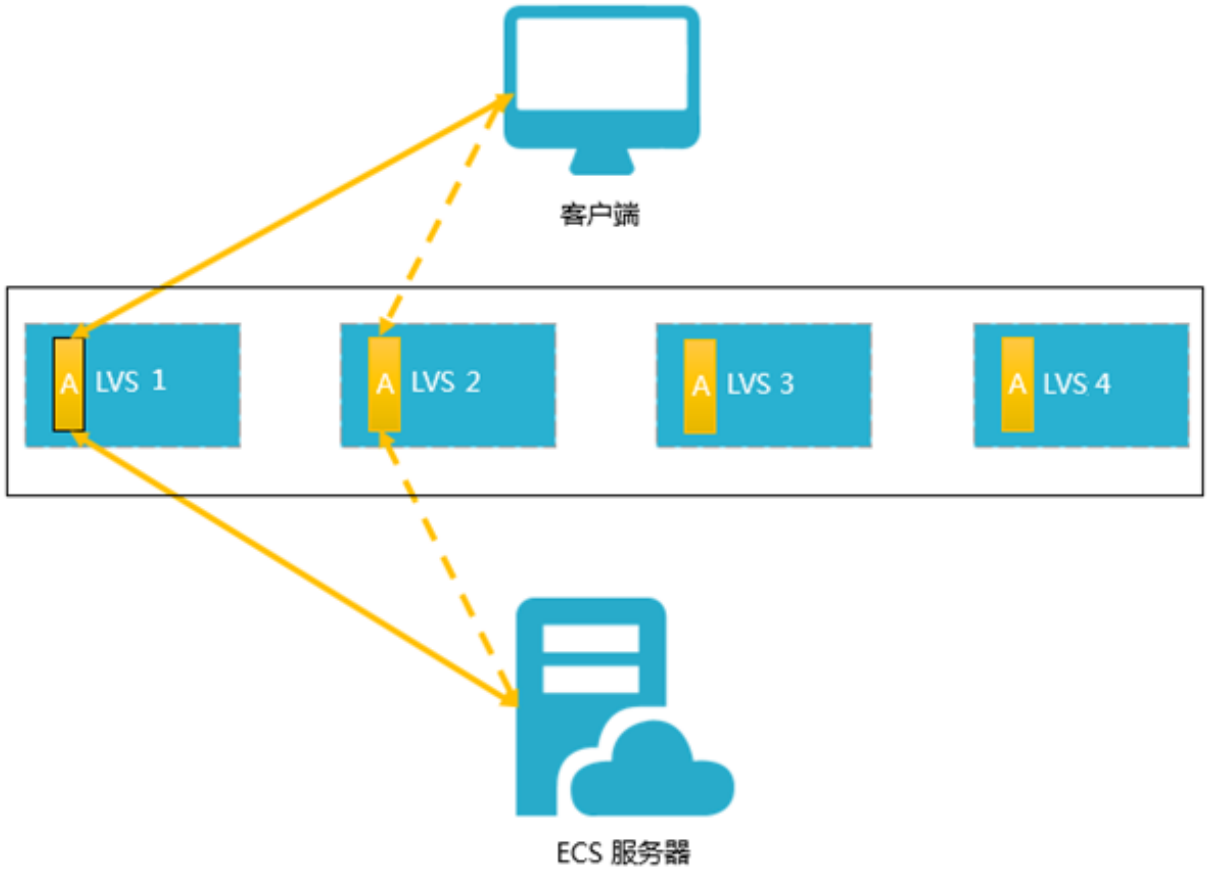
图 20: 负载均衡架构



如下图所示，各个地域的四层负载均衡实际上是由多台LVS机器部署成一个LVS集群来运行的，采用集群部署模式极大的保证了异常情况下负载均衡服务的可用性、稳定性与可扩展性。



LVS集群内的每台LVS上都会话，通过组播报文同步到该集群内的其他 LVS机器上，从而实现LVS集群内部各台机器间的会话同步。如下图所示，在LVS1上面建立的会话A，当客户端向服务端传输三个数据包后，会开始同步到其他LVS机器上，图中实线表示现有的连接，图中虚线表示当LVS1出现故障或进行维护时，这部分流量会走到一台可以正常运行的机器LVS2上，这能够保证负载均衡集群支持热升级，机器故障和集群维护时最大程度对用户透明，不影响用户业务。



8.2.1 四层负载均衡LVS技术特点

官方LVS存在的问题

LVS是全球最流行的四层负载均衡开源软件，由章文嵩博士在1998年5月创立，可以实现LINUX平台下的负载均衡。LVS是基于linux netfilter框架实现（同iptables）的一个内核模块，名称为IPVS (IP Virtual Server)，其钩子函数分别HOOK在LOCAL_IN和FORWARD两个HOOK点。

在云计算大规模网络环境下，官方LVS存在如下问题：

- 问题1：LVS支持NAT/DR/TUNNEL三种转发模式。上述模式在多vlan网络环境下部署时，存在网络拓扑复杂，运维成本高的问题。

- 问题2：和商用负载均衡设备（如F5）相比，LVS缺少DDoS攻击防御功能。
- 问题3：LVS采用PC服务器，使用常用软件keepalived的VRRP心跳协议进行主备部署，其性能无法扩展。
- 问题4：LVS常用管理软件keepalived的配置和健康检查性能不足。

四层负载均衡LVS定制化功能

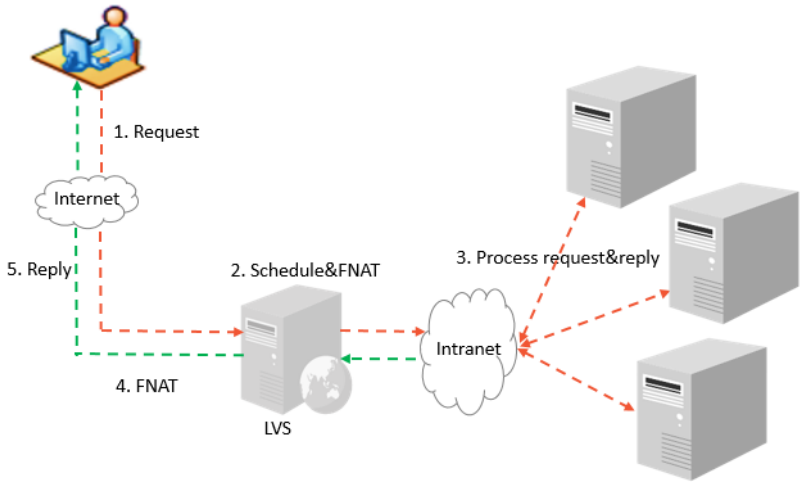
为了解决上述问题，阿里巴巴在官方LVS基础上进行了定制化。 Ali-LVS开源地址*https://github.com/alibaba/LVS*。

- 定制1：新增转发模式FULLNAT，实现LVS和Real Server间跨vlan通讯。
- 定制2：新增synproxy等攻击TCP标志位DDoS攻击防御功能。
- 定制3：采用LVS集群部署方式。
- 定制4：优化keepalived性能。

定制1：FULLNAT技术

- FULLNAT实现主要思想：引入local address（内网IP地址），cip-VIP转换为lip-rip，而lip和rip均为IDC内网IP，可以跨vlan通讯。
- IN/OUT的数据流全部经过LVS。为了保证带宽，采用万兆（10G）网卡。
- FULLNAT转发模式，当前仅支持TCP协议。

图 21: FULLNAT转发



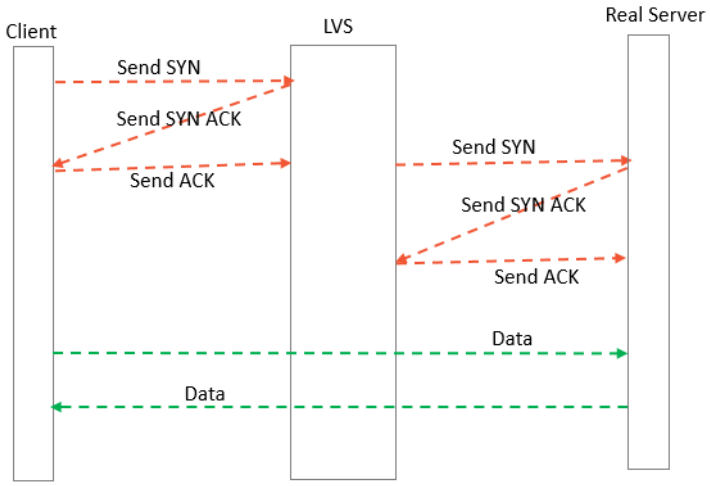
定制2：SYNPROXY技术

LVS针对TCP标志位DDoS攻击和Synflood攻击，利用synproxy模块进行防御。实现主要思想：参照Linux TCP协议栈中syncookies的思想，LVS代理TCP三次握手。

代理过程如下：

1. Client发送syn包给LVS。
2. LVS构造特殊seq的synack包给client，client回复ack给LVS。
3. LVS验证ack包中ack_seq是否合法；如果合法，则LVS再和Realserver建立3次握手。

图 22: LVS代理TCP三次握手



针对ACK、 FIN和RST Flood攻击，LVS查找连接表，如果不存在，则直接丢弃。

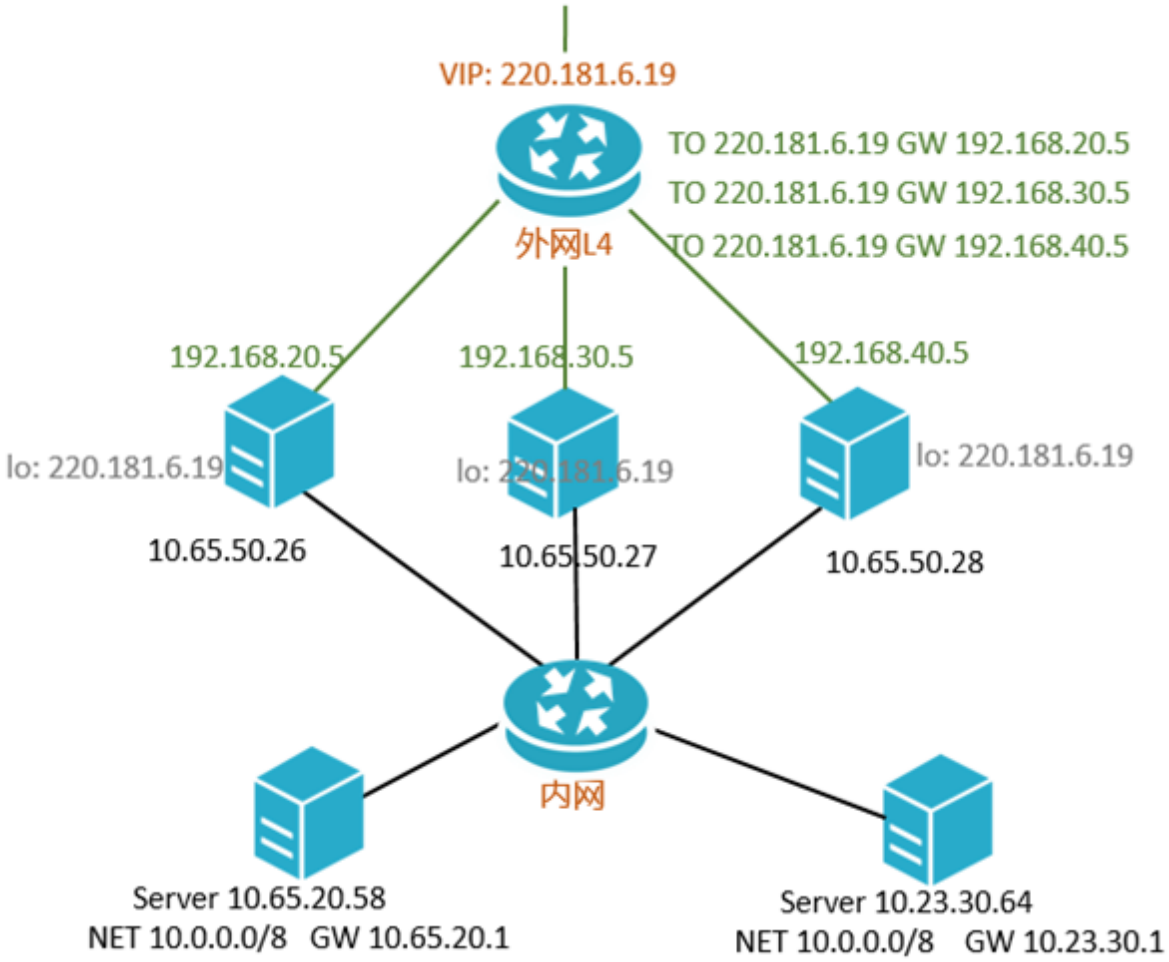
定制3：集群部署方式

LVS集群部署方式实现的主要思想：LVS和上联交换机间运行OSPF协议，上联交换机通过ECMP等价路由，将数据流分发给LVS集群，LVS集群再转发给业务服务器。

集群部署方式极大地保证了异常情况下负载均衡服务的稳定性。

- 健壮性：LVS和交换机间运行OSPF心跳。一个VIP配置在集群的所有LVS上，当一台LVS不可用时，交换机会自动发现并将其从ECMP等价路由中剔除。
- 可扩展：如果当前LVS集群无法支撑某个VIP的流量，LVS集群可以进行水平扩容。

图 23: 集群部署



定制4：keepalived优化

对LVS管理软件keepalived进行了全面优化，包括：

- 优化了网络异步模型，select改为epoll方式。
- 优化了reload过程。

四层负载均衡的特点

综上所述，四层负载均衡有如下特点：

- 高可用：LVS集群保证了冗余性，无单点。
- 安全：LVS自生攻击防御+云盾，提供了近实时防御能力。
- 健康检查：对后端ECS进行健康检查，自动屏蔽异常状态的ECS，待该ECS恢复正常后自动解除屏蔽。

8.2.2 七层负载均衡Tengine技术特点

Tengine是阿里巴巴发起的Web服务器项目，其在Nginx的基础上，针对大访问量网站的需求，添加了很多高级功能和特性。Nginx是当前最流行的7层负载均衡开源软件之一。Tengine开源地址*http://tengine.taobao.org/*。

Tengine定制化功能

针对云计算场景，Tengine定制的主要特性如下：

- 继承Nginx-1.4.6的所有特性，100%兼容Nginx的配置。
- 动态模块加载（DSO）支持。加入一个模块不再需要重新编译整个Tengine。
- 更加强大的负载均衡能力，包括一致性hash模块、会话保持模块，还可以对后端的服务器进行主动健康查，根据服务器状态自动上线下线。
- 监控系统的负载和资源占用从而对系统进行保护。
- 显示对运维人员更友好的出错信息，便于定位出错机器。
- 更强大的防攻击（访问速度限制）模块。

七层负载均衡特点

采用Tengine作为负载均衡的基础模块，阿里七层负载均衡产品有如下特点：

- 高可用：Tengine集群保证了冗余性，无单点。
- 安全：多维度的CC攻击防御能力。
- 健康检查：对后端ECS进行健康检查，自动屏蔽异常状态的ECS，待该ECS恢复正常后自动解除屏蔽。
- 支持7层会话保持功能。
- 支持一致性hash调度。

8.3 功能特性

协议支持

当前提供四层（TCP协议和UDP协议）和七层（HTTP和HTTPS协议）的负载均衡服务。

健康检查

支持对后端ECS进行健康检查，自动屏蔽异常状态的ECS，待该ECS恢复正常后自动解除屏蔽。

会话保持

提供会话保持功能，在Session的生命周期内，可以将同一客户端的请求转发到同一台后端ECS上。

调度算法

支持轮询、最小连接数两种调度算法。

- 轮询：按照访问次数依次将外部请求依序分发到后端ECS上。
- 最小连接数：连接数越小的后端服务器被轮询到的次数（概率）也越高。

访问控制

支持白名单控制，通过设置负载均衡监听，仅允许特定IP访问，适用于用户的应用只允许特定IP访问的场景。

证书管理

针对HTTPS协议，提供统一的证书管理服务，证书无需上传到后端ECS，解密处理在负载均衡上进行，降低后端ECS CPU开销。

实例类型

支持外网或内网类型的负载均衡服务。您可以根据业务场景来选择配置对外公开或对内私有的负载均衡服务。

外网类型的负载均衡默认使用经典网络；内网类型的负载均衡服务可以选择使用经典网络或专有网络。

管理方式

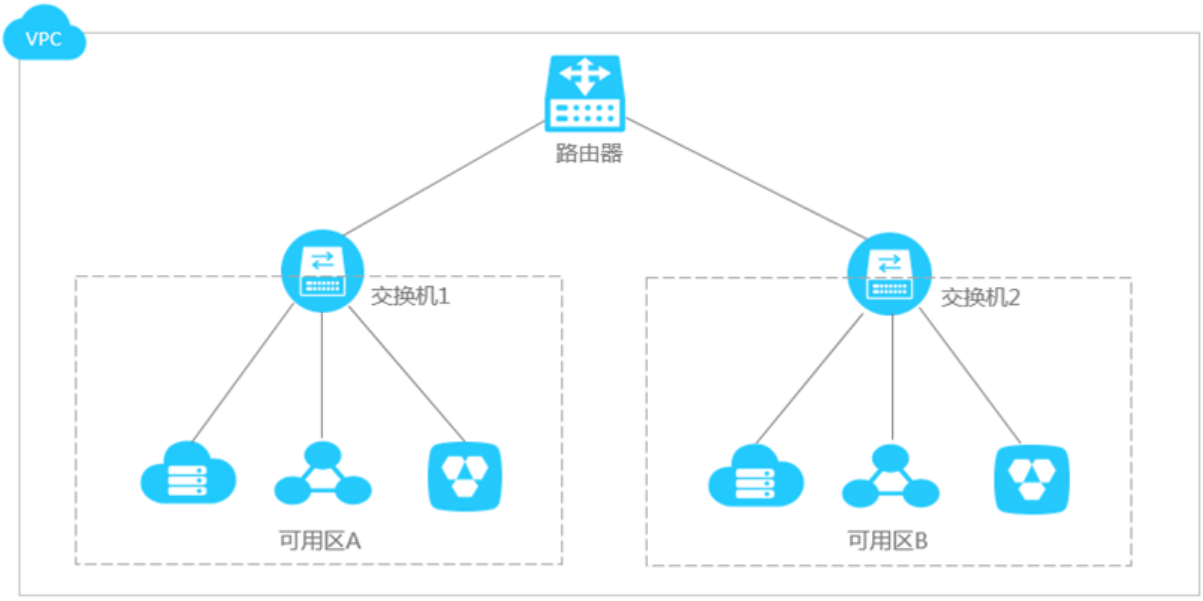
提供控制台、API、SDK多种管理方式。

9 专有网络VPC

9.1 产品概述

专有网络VPC（Virtual Private Cloud），帮助您基于阿里云构建出一个隔离的网络环境。您可以完全掌控自己的虚拟网络，包括选择自有IP地址范围、划分网段、配置路由表和网关等。此外您也可以通过专线 、VPN等连接方式将VPC与传统数据中心组成一个按需定制的网络环境，实现应用的平滑迁移上云。

图 24: 专有网络



专有网络和经典网络

阿里云提供如下两种网络类型：

- 经典网络

  经典网络类型的云产品，统一部署在阿里公共基础内，规划和管理由阿里云负责，更适合对网络易用性要求比较高的客户。

- 专有网络

  专有网络是一个可以自定义隔离专有网络，您可以自定义这个专有网络的拓扑和IP地址，适用于对网络安全性要求较高和有一定的网络管理能力的客户。

经典网络和专有网络的功能差异如下表所示。

表 3: 经典网络和专有网络功能对比

功能	经典网络	专有网络
二层逻辑隔离	不支持	支持
自定义私网网段	不支持	支持
私网IP规划	经典网络内唯一	专有网络内唯一，专有网络间可重复
自建VPN	不支持	支持
私网互通	账号内相同地域内互通	专有网络内互通，专有网络间隔离
路由表	不支持	支持
交换机	不支持	支持
自定义路由器	不支持	支持
SDN	不支持	支持
隧道技术	不支持	支持
自建NAT网关	不支持	支持

9.2 产品架构

背景信息

随着云计算的不断发展，对虚拟化网络的要求越来越高，弹性 (scalability)、安全 (security)、可靠 (resilience)、私密 (privacy)，并且还要求极高的互联性能 (performance)，因此催生了多种多样的网络虚拟化技术。

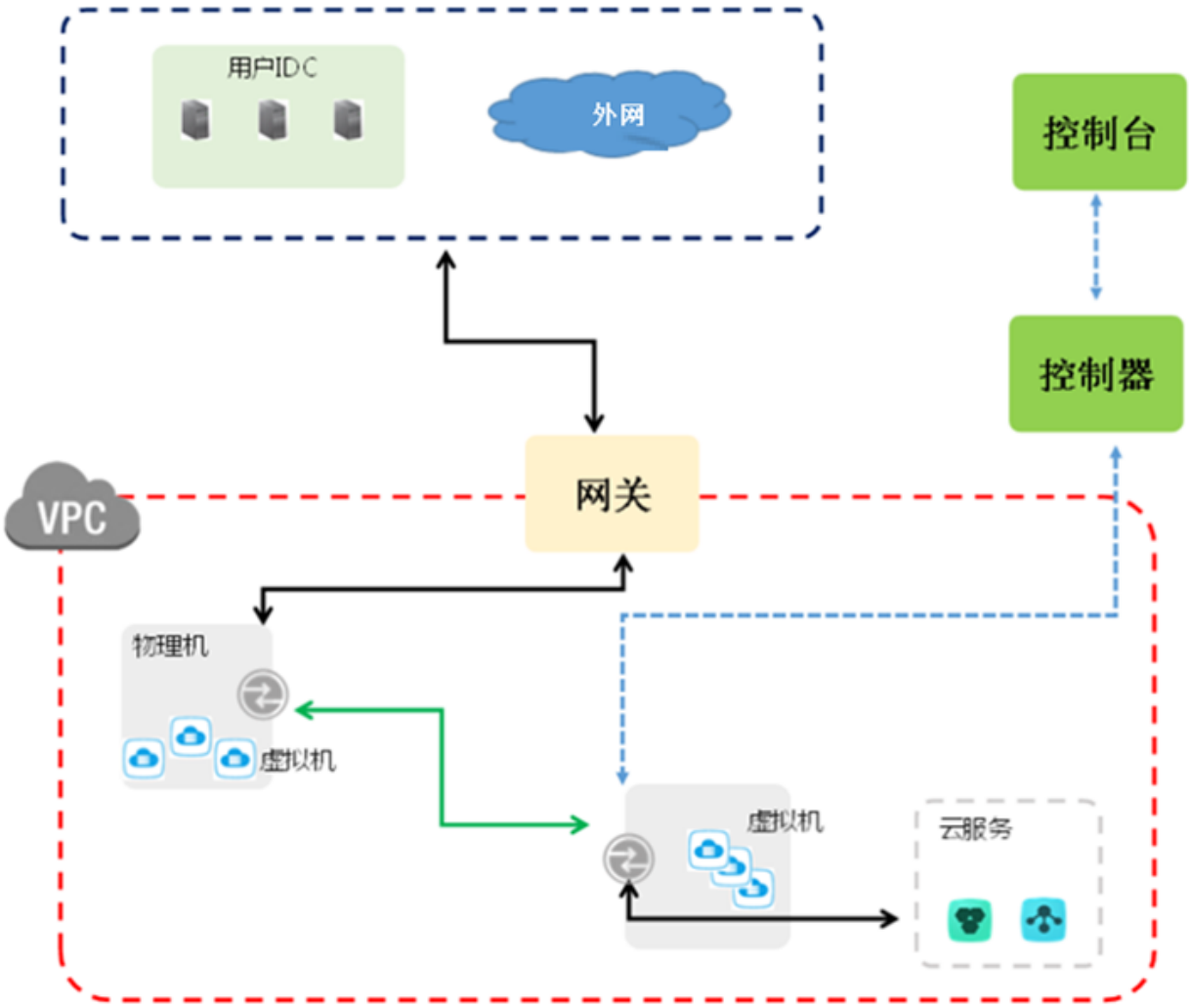
比较早的解决方案，是将虚拟机的网络和物理网络融合在一起，形成一个扁平的网络架构，例如大二层网络。这种类似的方案，随着虚拟化网络规模的增大，ARP 欺骗、广播风暴、主机扫描等问题会越来越严重。为了解决这些问题，出现了各种网络隔离技术，把物理网络和虚拟网络彻底隔开。其中一种技术是用户之间用 VLAN 进行隔离，但是 VLAN 的数量最大只能支持到 4096 个，无法支撑公共云的巨大用户量。

原理描述

基于目前主流的隧道技术，专有网络 VPC 隔离了虚拟网络。每个 VPC 都有一个独立的隧道号，一个隧道号对应着一张虚拟化网络。一个 VPC 内的 ECS 之间的传输数据包都会加上隧道封装，带有唯一的隧道 ID 标识，然后送到物理网络上进行传输。不同 VPC 内的 ECS 因为所在的隧道 ID 不同，本身处于两个不同的路由平面，从而使得两个不同的隧道无法进行通信，天然的进行了隔离。

基于隧道技术，阿里云的研发团队自研了交换机，软件自定义网络（Software Defined Network，简称 SDN ）技术和硬件网关，在此基础上实现了 VPC 产品。

逻辑架构



如上图所示，在 VPC 架构里面包含交换机、网关和控制器三个重要的组件。

- 交换机和网关组成了数据通路的关键路径，控制器使用自研的协议下发转发表到网关和交换机，完成了配置通路的关键路径，整体架构里面，配置通路和数据通路互相分离。
- 交换机是分布式的结点，网关和控制器都有集群部署并且是多机房互备的，所有链路上都有冗余容灾，提升了 VPC 产品的整体可用性。
- 交换机和网关性能在业界都是领先的，自研的 SDN 协议和控制器，能轻松管控云上成千上万张虚拟网络。

在产品上，除了给用户一张独立的虚拟化网络，阿里云还为每个 VPC 提供了独立的路由器、交换机组件，让用户可以更加丰富的进行组网。针对有内网安全需求的用户，还可以使用安全组技术在一

个 VPC 进行更加细粒度的访问控制和隔离。缺省情况下，VPC 内的 ECS 只能和本 VPC 内其他 ECS 通信，或者和 VPC 内的其他云服务之间进行通信。用户可以使用阿里云提供的 VPC 相关的 EIP 功能、高速通道功能，使得 VPC 可以和 Internet、其他 VPC、用户自有的网络（如用户办公网络、用户数据中心）之间进行通信。

9.3 功能特性

私网网段

在创建VPC和交换机时您需要指定专有网络和其子网的网段。每个专有网络只能指定一个网段，网段范围如下表所示。

表 4: 专有网络网段

网段	可用主机数	备注
192.168.0.0/16	65532	去除系统占用地址
172.16.0.0/12	1048572	去除系统占用地址
10.0.0.0/8	16777212	去除系统占用地址

交换机

交换机是组成专有网络的基础网络设备，它可以连接不同的云产品实例。创建专有网络之后，您可以通过添加交换机为专有网络划分一个或多个子网。在创建交换机时，您也要指定交换机的网段，交换机的网段可以和它所属的VPC网段一样或者是其VPC网段的子集，子网掩码必须在16到29之间。

路由器

路由器是一个专有网络的枢纽。作为专有网络中重要的功能组件，它可以连接VPC内的各个交换机，同时也是连接VPC与其它网络的网关设备。

每个路由器中维护一张路由表，它会根据具体的路由条目的设置来转发网络流量。您创建VPC时，系统会自动为VPC创建一个路由器。删除VPC时，系统也会自动删除对应的路由器。目前不支持直接创建和删除路由器。

路由表

路由表是指路由器上管理路由条目的列表。新建VPC时，系统会自动创建一个路由表。删除VPC时，系统也会自动删除对应的路由表。不支持直接创建和删除路由表。

说明： 每个路由器只能有一个路由表。路由表中的路由条目会影响VPC中的所有云产品实例。

路由条目

路由表中的每一项是一条路由条目，路由条目定义了通向指定目标网段的网络流量的下一跳地址，路由条目包括系统路由和自定义路由两种类型的路由条目。

专有网络创建时，系统会自动创建1条系统路由条目，用于专有网络内的云产品实例访问专有网络外的云服务。创建交换机，系统也会创建1条对应的系统路由条目，目的地址为所创建交换机的网段。您可以创建和删除自定义路由条目。

选路规则

路由表中采用最长前缀匹配作为流量的路由选路规则。最长前缀匹配是指IP网络中当路由表中有多条条目可以匹配目的IP时，采用掩码最长（最精确）的一条路由作为匹配项并确定下一跳。

例如，某专有网络中路由表中路由条目如下表所示。

表 5: 路由表示例

目标网段	下一跳类型	下一跳地址	类型
100.64.0.0/10	-	-	System
192.168.0.0/24	-	-	System
0.0.0.0/0	Instance	i-12345678	Custom
10.0.0.0/24	Instance	i-87654321	Custom

目的地址为 100.64.0.0/10和192.168.0.0/24的两条路由均为系统路由，前者为系统保留的地址段，后者为专有网络中为交换机配置的系统路由。

目的地址为0.0.0.0/0和10.0.0.0/24的两条路由为自定义路由，表示将访问0.0.0.0/0地址段的流量转发至ID为i-12345678的ECS实例，将访问10.0.0.0/24地址段的流量转发至ID为 i-87654321的ECS实例。根据最长前缀匹配规则，在该专有网络中，访问10.0.0.1的流量会转发至 i-87654321，而访问10.0.1.1的流量会转发至i-12345678。

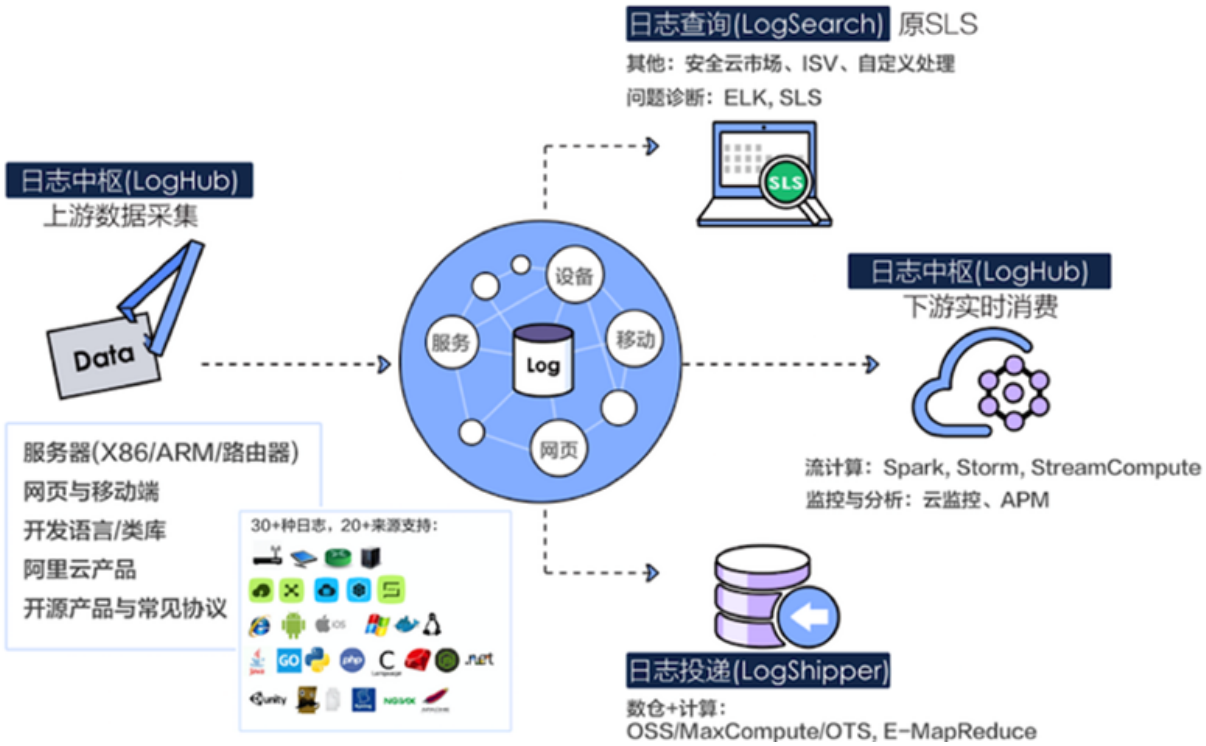
10 日志服务

10.1 产品概述

日志服务（Log Service，Log）是针对日志类数据场景的一站式解决方案，解决海量日志数据采集/订阅、转储与查询功能。

- 实时采集与订阅：通过客户端、API、Tracking JS，Library 等手段实时采集来自多个渠道日志数据。数据写入后，可以进行实时订阅读取。例如通过 Spark Streaming、Storm、Consumer Library 等接口对数据进行实时处理。
- 日志投递：将实时日志数据通过一定规则，目录映射，字段制定等写入大规模存储系统。
- 日志索引查询：实时索引日志数据，并提供实时、海量存储查询引擎。可以基于时间、关键词、上下文等任意维度进行日志查询。

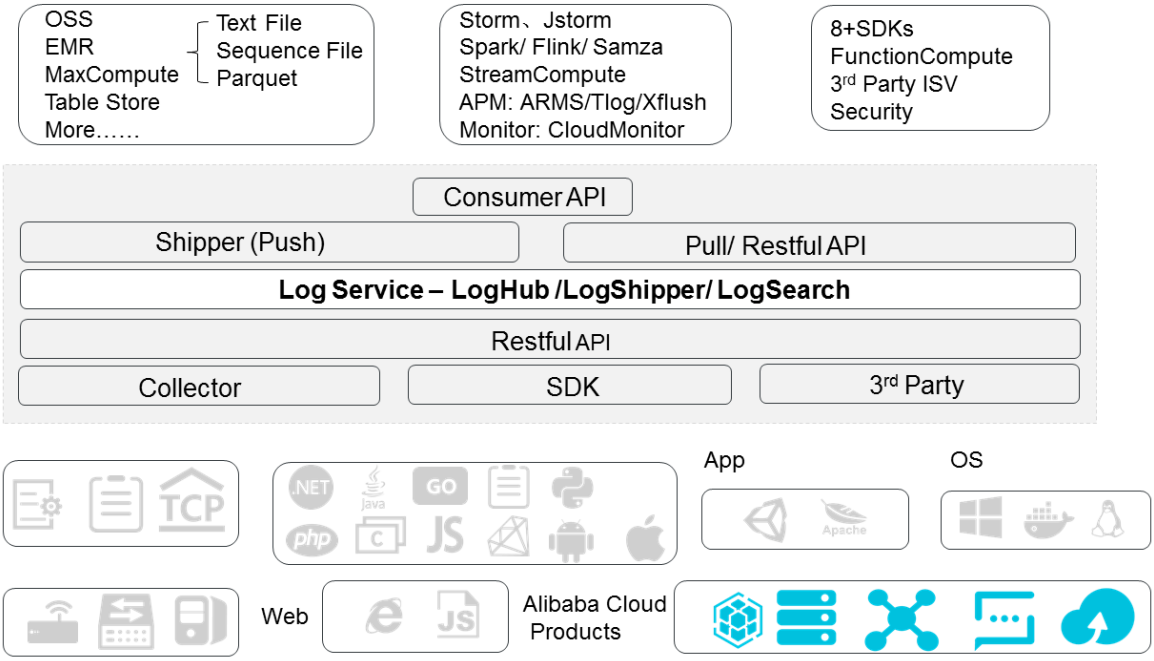
日志服务提供弹性伸缩、自动扩容等能力，可以水平支持 PB 级数据。



10.2 产品架构

日志服务的架构如下图所示。

图 25: 产品架构



Logtail

帮助您快速收集日志的Agent。其特点如下所示：

- 基于日志文件、无侵入式的收集日志
  - 只读取文件。
- 安全、可靠
  - 支持文件轮转不丢失数据。
  - 支持本地缓存。
  - 网络异常重试。
- 方便管理
  - Web端可视化配置
- 完善的自我保护
  - 实时监控进程CPU、内存消耗，限制使用上限。

前端服务器

采用LVS + Nginx构建的前端机器。其特点如下所示：

- HTTP、REST协议
- 水平扩展

- 流量上涨时可快速通过增加前端机来提高处理能力。

- 高吞吐、低延时
  - 纯异步处理，单个请求异常不会影响其他请求。
  - 内部采用专门针对日志的Lz4压缩，提高单机处理能力，降低网络带宽。

后端服务器

后端是分布式的进程，部署在多个机器上，完成实时对Logstore数据的持久化、索引、查询以及投递至MaxCompute。整体后端服务的特点如下所示：

- 数据高安全性：
  - 您写入的每条日志，都会被保存3份。
  - 任意磁盘损坏、机器宕机情况下，数据自动复制修复。
- 稳定服务：
  - 进程崩溃和机器宕机时，Logstore会自动迁移。
  - 自动负载均衡，确保无单机热点。
  - 严格的Quota限制，防止单个用户行为异常对其他用户产生影响。
- 水平扩展：
  - 以分区（Shard）为单位进行水平扩展，用户可以按需动态增加分区来增加吞吐量。

10.3 功能特性

日志实时采集（LogHub）

实时采集与消费。通过 30+ 方式实时采集海量数据、下游实时消费。

- 使用 logtail 采集日志：稳定可靠、安全、全平台（Linux、Windows、Docker)、高性能&低资源占用。
- 通过 API/SDK 采集日志：灵活方便，可扩展，支持 10+ 种语言、移动端。
- 云产品日志采集：支持云服务器（Elastic Compute Service，ECS）、容器服务（Container Service）、消息服务（Message Service，MNS）、内容分发系统（Content Delivery Network，CDN）等云产品的日志接入。一键打通，便捷高效。
- 其他方式：Syslog、Unity3D、Logstash、Log4j、Nginx 等。

日志实时消费（LogHub）

流计算、协同消费库、多语言支持。

- 功能完善：覆盖 Kafka 100% 功能，并提供保序、弹性伸缩、根据时间段 Seek 等功能。
- 稳定可靠：写入即可消费；99.9% 以上可用性；数据多份拷贝；秒内弹性伸缩；低成本。
- 使用便捷：支持 Spark Streaming，Storm，Consumer Library（一种自动负载均衡的编程模式），SDK 订阅等。

日志投递（LogShipper）

稳定可靠的日志投递。将日志中枢数据投递至存储类服务进行存储与大数据分析。

- 对象存储（Object Storage Service，OSS）：将日志投递到 OSS，利用 E-MapReduce 进行分析。
- 大数据计算（MaxCompute）：将日志投递到 MaxCompute 进行分析。
- 表格存储（Table Store）：将日志投递到表格存储。

日志查询（LogSearch）

实时索引、查询数据。对日志中枢创建索引，提供基于时间、关键词进行检索。

- 大规模：PB 级实时索引（写入 1 秒内即可查）；查询每秒过十亿级日志。
- 查询灵活：支持关键词、模糊、跨 Topic 查询、及上下文查询。

10.4 产品价值

帮助快速搭建面向海量日志数据解决方案。

解决如下典型场景下的问题：数据采集、实时计算、数仓与离线分析、产品运营与分析、运维与管理等场合。

- 数据收集与消费（Data Collection & Consumption）
- 数据清洗与流计算（ETL/Stream Processing）
- 数据仓库对接（Data Warehouse）
- 事件溯源（Event Sourcing/Tracing）
- 日志管理（LogManagement）

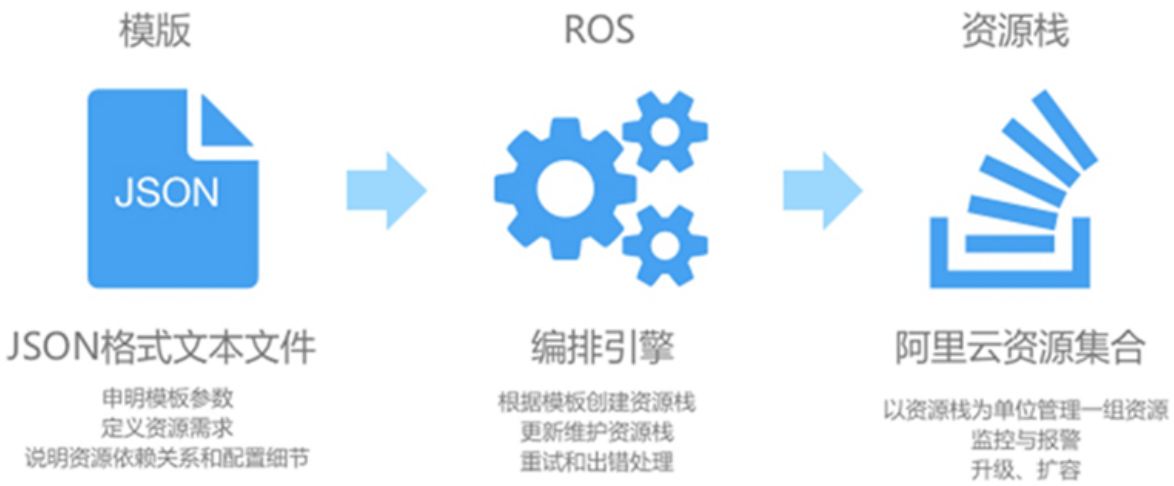
11 资源编排

11.1 产品概述

阿里云资源编排服务（ROS）是一款帮助阿里云用户简化云计算资源管理和自动化运维的服务。您遵循ROS定义的模板规范，编写模板文件，在模板中定义所需云计算资源的集合及资源间的依赖关系、资源配置细节等，ROS通过编排引擎自动完成所有资源的创建和配置，以达到自动化部署、运维的目的。编排模板是一种用户可读、易于编写的文本文件，用户可以通过SVN、Git等版本控制工具来控制模板的版本，以达到控制基础设施版本的目的，您可以通过API、SDK等方式把ROS的编排能力与自己的应用整合，做到基础设施即代码（Infrastructure is Code）。

编排模板同时也是一种标准化的资源和应用交付方式，您可以通过编排模板交付包含云资源和应用的整体系统和解决方案。ISV可以通过这种交付能力，轻松的整合阿里云的资源和ISV的软件系统，达到统一交付的目的。

编排服务是通过资源栈（Stack）这样的逻辑集合来统一管理一组云资源，所以，对于云资源的创建、删除、克隆等操作都可以以资源组为单位来完成。在DevOps实践中，可以很轻松的克隆开发、测试、线上环境。同时，也可以更容易做到应用的整体迁移和扩容。



11.2 功能特性

ROS是云计算中很重要的一个服务，您的整个基础架构全部包含在编辑好的ROS模板中，无论何时何地通过ROS就能在云上构建出自己的基础架构，正真做到基础设施即代码（Infrastructure is Code）。与直接调用各产品的OpenAPI相比，大大提高了您展开业务的效率。

ROS层面所看到都是stack，在stack之下是资源，这些资源也可以通过各产品的控制台访问。通过ROS的控制台可以操作ROS的stack和stack的一组资源。

ROS专有云控制台一般会包括：

- **资源栈管理**

  提供您已经创建的资源栈的概览信息，可以查看stack的详细信息，浏览stack的资源信息、事件信息以及原始模板。以及可适用于某一个stack的操作，现在支持的操作有：重新创建、健康检查、更新堆栈。重新创建是指用该stack的原始模板重新创建一个stack，可以指定不同的参数。健康检查是指检查该栈中的资源状态是否可用；更新堆栈是指通过修改原始模板，更新堆栈中的资源。

- **新建资源栈**

  通过ROS模板创建一个全新的stack。

- **资源类型**

  列出ROS当前所支持的所有的资源类型。

- **ECS实例相关信息**

  当前各可用区中ECS所支持的规格、镜像，通过单击ECS规格行中的**创建**按钮，ROS可以很快的创建出您想要的ECS资源。

11.3 产品价值

ROS使用灵活、简便、不需要关注个产品的调用逻辑，可为您节省很高的开发运维成本。

- 模板简单易懂；
- 实现资源关系的编排；
- 实现基础架构一键交付；
- 资源安组运维成本小；
- 资源更新操作简便快捷。

下面表格列出了ROS和传统OpenAPI相比所具有的优势。

表 6: ROS与传统OpenAPI对比表

对比项	ROS	openAPI
部署交付周期	快速编辑模板	开发调试数天

对比项	ROS	openAPI
配置动态伸缩	修改模板实现伸缩	开发调试数天
底层API升级	修改模板实现	开发调试数天
组资源更新	修改模板实现	开发调试数天
可迁移性	相同模板直接创建	开发调试

ROS具有灵活性、方便性且低成本的特点。使您有更多时间关注自己的核心业务，做到基础设施即代码（Infrastructure is Code）。在DevOps实践中，可以很轻松的克隆开发、测试、线上环境，同时，也可以更容易做到应用的整体迁移和扩容。

12 云盾基础版

12.1 产品概述

阿里云云盾是阿里巴巴集团多年来安全技术研究积累的成果，结合阿里云云计算平台强大的数据分析能力以及阿里云专业的安全运营团队，为云用户提供多层面、一体化的安全防护服务。

云盾基础版由网络流量监控系统、主机入侵防御系统、安全审计、和集中管控系统四大功能模块组成。

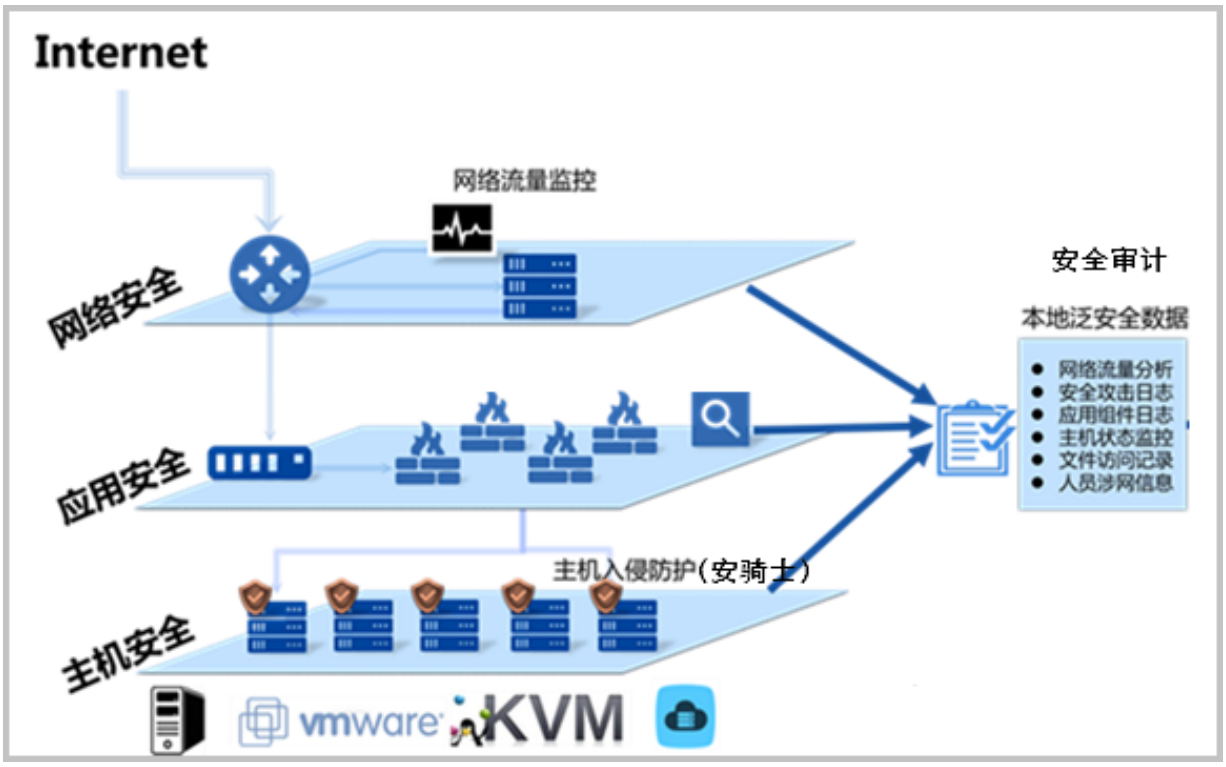
依靠阿里云平台自身的安全特性以及云盾为云上客户提供的攻击防御特性，阿里云先后取得了国内外多项云安全认证：

图 26: 阿里云获得的安全认证

- 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
- 阿里云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
- 金融云通过等保四级测评，全国首个四级云平台

12.2 产品架构

图 27: 云盾基础版在专有云中的部署逻辑拓扑结构



- **网络流量监控**（Beaver）

  网络流量监控模块部署在专有云的网络边界，通过流量镜像的方式将出入专有云的所有网络流量进行逐包检测分析，分析结果将作为云盾其他防护模块的参考依据。同时，网络流量监控模块自带旁路式Web防火墙功能，对明文的Web攻击请求进行旁路阻断。

- **主机入侵防护**（**安骑士**）

  主机入侵防御模块部署在云服务器上，实现对云服务器上的Web木马进行查杀、对密码暴力破解进行拦截、对异常的登录行为进行告警，同时还能对高危漏洞进行修复。

- **安全审计**

  安全审计模块对标信息系统安全等级保护基本要求，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，实现行为日志的收集、存储、分析、报警等功能。

- **集中管控**

  集中管控系统部署在云服务器集群中，负责对云盾所有安全模块的集中策略管理以及进行统一的日志分析。

12.3 功能特性

云盾基础版由网络流量监控、主机入侵防御、安全审计功能模块组成。

12.3.1 云盾基础版功能

12.3.1.1 网络流量监控

网络流量监控模块是阿里云安全自主研发的毫秒（ms）级攻击监控产品。通过对专有云入口镜像流量包的深度解析，实时地检测出各种攻击和异常行为，并与其他防护模块联动防护。网络流量监控模块在整个云盾防御体系中，提供了丰富的信息输出与基础的数据支持。

网络流量监控模块提供以下功能：

模块	功能项	说明
网络流量监控	DDoS攻击检测	通过流量镜像方式，旁路检测云边界流量中的DDoS攻击。
	流量统计	对云产品使用流量进行计量，生成流量图。
	网络层Web攻击拦截	根据内嵌的Web匹配规则，对常见的Web攻击进行网络层拦截、旁路阻断。
	IP黑名单	对添加的IP黑名单库进行旁路TCP阻断。
	恶意主机识别	对云计算平台内部的恶意主机进行识别和告警。

12.3.1.2 主机入侵防御

主机入侵防御模块是阿里云自主研发，专门面向阿里云云服务器的安全防护模块。主机入侵防御模块默认部署在云服务器上，联动云盾云安全防护体系中的其他安全模块在主机层面为您提供多项安全防护能力。

主机入侵防御模块提供密码暴力破解防御、木马文件检测和处理、异地登录告警等功能。

主机入侵防御模块具有以下特点和优势：

- **海量的暴力破解防御能力**

  基于大数据处理能力，实时识别出暴力破解攻击行为，日均拦截暴力破解行为超过8.5亿次。

- **精准的Web木马检测**

  利用动静结合的检测方式，对网页木马进行分析检测。

  通过HTML和Javascript引擎对可疑的代码进行解析，利用基于阿里云数百万量级的恶意文件特征库进行静态模式匹配识别。同时，通过模拟浏览器对被检测页面进行访问，动态地分析代码的恶意行为，从而发现未知木马、及时主动隔离，实现木马检测的零误报。

- **异地登录告警**

基于用户的登录行为模型，准确识别出异地（精确到地市级）登录行为，对疑似的非管理员登录系统行为通过手机短信进行告警。

- **高危漏洞检测与修复**

  借助阿里云大数据计算能力以及阿里巴巴安全研究团队，主机入侵防御模块可第一时间获取0Day漏洞的技术细节，并在主机上完成高危漏洞修复。覆盖范围包括：Web应用漏洞修复、系统文件修复等。

12.3.1.3 安全审计

安全审计模块是基于云计算平台的一体化解决方案。对标信息系统安全等级保护基本要求，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，实现了行为日志的收集、存储、分析、报警等功能。

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宿主机，从各个角度对行为进行收集，确保了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。目前，已支持500亿条日志的同时索引。

安全审计模块提供以下功能：

模块	功能项	说明
安全审计	网络审计	对云平台网络设备的登录和操作进行审计。
	物理服务器审计	对云平台物理服务器的登录和操作进行审计。
		对平台内部API调用，命令执行的操作进行审计。
	云平台审计	对云平台内部的各个API调用操作审计。
	云产品审计	云产品ECS虚拟机登录审计。
		云产品RDS数据库操作审计。

模块	功能项	说明
		云产品ODPS的操作审计。

12.4 产品价值

12.4.1 云环境下的安全威胁

云计算的虚拟化资源池、弹性架构、服务可度量、灵活接入和按需服务等特性让计算资源（包括网络，服务器，存储，应用软件，服务）变得像自来水一样随时、随地、随需可得，极大的优化了IT资源效率，但同时也对云上用户的IT系统安全性提出了新的挑战。

12.4.1.1 DDoS攻击威胁

分布式拒绝服务攻击（Distributed Denial of Service，简称DDoS攻击）是对云计算环境影响最大的系统可用性威胁， 2011年至2013年连续三年被云安全联盟（Cloud Security Alliance，简称CSA）收录为云端十大安全威胁之一。

DDoS攻击在云端的表现：攻击者利用互联网上大量存在的僵尸网络主机，向云上服务器发起大量的正常服务请求导致云服务器过载，从而影响正常用户的访问。

大量的DDoS攻击，轻则导致被攻击的云服务器无法正常提供服务，影响客户的在线业务，重则导致整个云环境网络不稳定，影响云环境的可用性。另一方面，云平台本身拥有非常强大的计算能力和基础带宽资源，攻击者也可以通过利用云上服务器发起DDoS攻击。

DDoS攻击防护的最佳安全实践是通过云防御来解决。

12.4.1.2 网络入侵威胁

云计算平台是互联网的基础设施，用户的业务都是以数据的形式承载于云计算平台上，数据是云平台上最重要的资产。攻击者往往以数据为攻击目标，通过各种网络渗透攻击手段获取或者篡改客户的业务数据，从而达到非法目的。因此，在云平台上防御攻击者的入侵行为是云计算环境下保护用户业务系统安全的重中之重。

在云环境中常见的网络入侵行为有：

- 攻击者通过暴力破解或者其他方式获得操作系统、服务或者Web应用的访问权限，从而非法登录系统，直接获取敏感数据。或者在登录系统后直接篡改系统中的敏感数据以达到个人非法目的。
- 攻击者利用系统或者Web应用漏洞发起远程攻击，窃取敏感数据。例如，攻击者利用Web系统漏洞，上传WebShell后门程序，从而获得服务器操作权限，利用系统权限直接下载数据库文

件（即，拖库攻击）。攻击者通过同样的手段也可以直接修改系统中的敏感数据。例如，利用Web应用漏洞发起SQL注入攻击，直接篡改数据库中存储的敏感数据。

- 攻击者在云服务器上放置监听程序，监听网络中的数据包，从而获取敏感数据。同样的，攻击者在云服务器上放置WebShell等木马后门程序，利用所获得的信息发起中间人攻击，或者钓鱼攻击。

12.4.1.3 内部威胁

随着业务系统访问、网络应用行为日益频繁，系统维护人员能够直接接触重要业务系统，产生内部威胁的概率也越来越高。这些内部行为，安全防护体系往往不能及时发现、定位源头，给企业带来了极大的困扰。

常见的内部威胁有：

- 内部系统维护人员对业务应用系统的越权访问、违规操作，损害业务系统的运行安全。
- 重要业务数据库，被员工或系统维护人员篡改牟利、外泄，给企业造成巨大的经济损失。

12.4.2 云盾产品价值

云盾在阿里云专有云出口，通过网络流量监控系统，在网络层对恶意的攻击行为进行识别，实时地阻断网络攻击行为。在主机层对Web木马和恶意文件进行实时查杀，避免云服务器被攻击者利用。实时拦截暴力破解行为，并对异常的登录行为进行告警，避免攻击者利用弱口令登录系统窃取或者破坏客户业务数据。

云盾由多个功能模块组成，在专有云网络出口、专有云网络中、专有云服务器上实现纵深防御，多点联动。为了方便您集中管理和实时掌握云平台安全风险，云盾提供了统一的管理视图，您可以在云盾集中管控系统上对所有安全防护模块中的安全策略进行统一管理，同时还可以在集中管控系统上对日志进行关联分析。

纵深防御的安全体系架构

云盾由涵盖网络安全、主机安全、应用安全、弱点分析等多层次安全防护模块组成，在云边界、云网络中、云服务器上形成一套纵深的防御体系，通过集中管控的管理中心协调调度，综合各模块提供的安全信息，做出最准确的判断，并且可以在最合适的位置检测和阻断恶意的攻击行为，有效地保护了云环境不受外界攻击者的侵扰，保障用户业务系统的安全。

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，可有效地保护公有云和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；云边界、云网络上防御模块采用的是旁路的架构，贴切云的业务，对云平台业务影响最小化；云服务器上的防御模块是虚拟化，适应虚拟机灵活的特点。

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，云盾也提供了租户自服务的控制台Portal，租户可以在上面查看自己相关的安全防护情况，生成简单的报表。利用合理的外部资源配置可以自动化的接收告警短信和邮件。

阿里云安全能力输出

云盾的防护策略和数据源自于多年的积累，阿里云上多达百万级的用户，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，每天对阿里云上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，为您带来更好的安全保障。

13 云盾高级版

13.1 产品概述

阿里云云盾是阿里巴巴集团多年来安全技术研究积累的成果，结合阿里云云计算平台强大的数据分析能力以及阿里云专业的安全运营团队，为云用户提供多层面、一体化的安全防护服务。

云盾基础版由网络流量监控系统、主机入侵防御系统、安全审计、和集中管控系统四大功能模块组成。云盾高级版在基础版的基础上增加了DDoS清洗、云防火墙、WAF和态势感知等功能，结合阿里云专业的安全运营服务为云用户提供了入侵防御、安全审计、态势感知和集中管控等一站式安全保障。

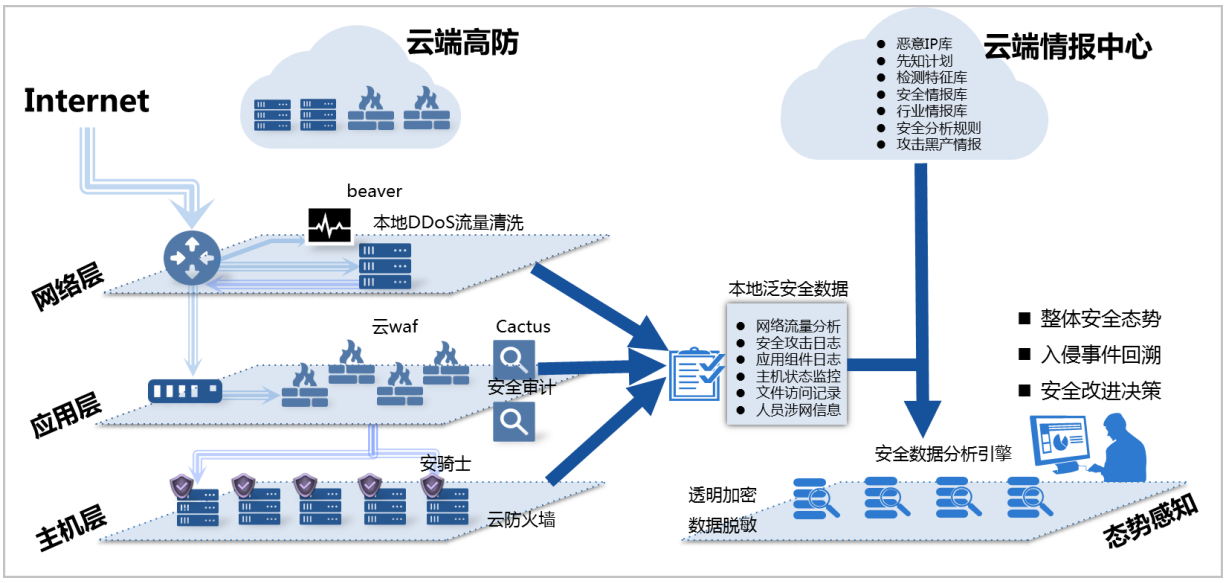
依靠阿里云平台自身的安全特性以及云盾为云上客户提供的攻击防御特性，阿里云先后取得了国内外多项云安全认证：

图 28: 阿里云获得的安全认证

- 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
- 阿里云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
- 金融云通过等保四级测评，全国首个四级云平台

13.2 产品架构

图 29: 云盾高级版在专有云中的部署逻辑拓扑结构



- **网络流量监控**（Beaver）

  网络流量监控模块部署在专有云的网络边界，通过流量镜像的方式将出入专有云的所有网络流量进行逐包检测分析，分析结果将作为云盾其他防护模块的参考依据。同时，网络流量监控模块自带旁路式Web防火墙功能，对明文的Web攻击请求进行旁路阻断。

- **主机入侵防护**（**安骑士**）

  主机入侵防御模块部署在云服务器上，实现对云服务器上的Web木马进行查杀、对密码暴力破解进行拦截、对异常的登录行为进行告警，同时还能对高危漏洞进行修复。

- **安全审计**

  安全审计模块对标信息系统安全等级保护基本要求，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，实现行为日志的收集、存储、分析、报警等功能。

- **DDoS攻击防御**

  DDoS清洗提供DDoS攻击流量检测、DDoS攻击过滤和集中策略管理功能。

- **Web应用防火墙**（**云**WAF）

  Web应用防火墙 (简称WAF)保护网站的应用程序避免遭受常见Web漏洞的攻击，包括常见Web应用攻击（例如SQL注入、XSS跨站脚本攻击等）、也包括CC这种影响网站可用性的资源消耗型攻击。同时，云WAF模块也支持根据网站实际业务制定精准的防护策略，用于过滤对您网站有恶意的Web请求。

- **云防火墙**

  云防火墙模块是基于云计算环境东西向流量微隔离需求的云访问控制系统。

- **弱点分析**（Cactus）

  弱点分析模块是专门为构建在云服务器上的Web应用进行漏洞扫描的安全模块。

- **态势感知**

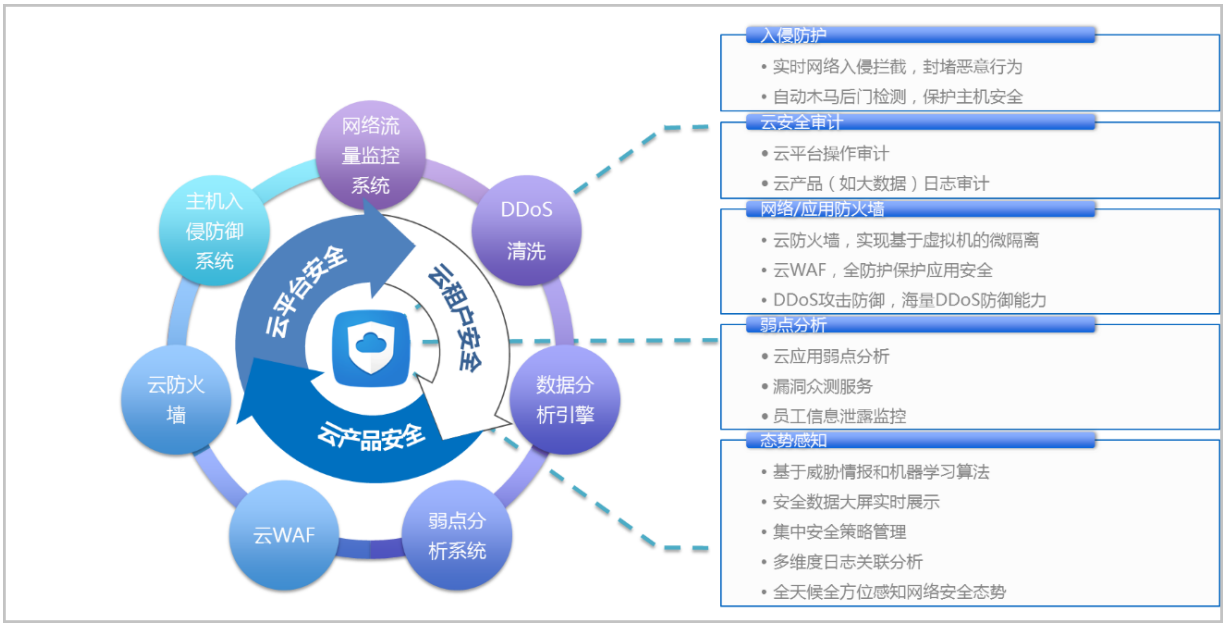
  态势感知系统通过汇集网络流量、主机端信息，结合从云端下发的威胁情报和大数据分析模型，在本地部署的大数据集群中进行威胁态势分析。

13.3 功能特性

云盾基础版由网络流量监控、主机入侵防御、安全审计功能模块组成；云盾高级版在基础版之上增加了DDoS攻击防御、云防火墙、云WAF、弱点分析和态势感知功能，结合阿里云专业的安全运营服务为云用户提供了入侵防御、安全审计、态势感知和集中管控等一站式安全保障。

云盾高级版功能如下图所示：

图 30: 云盾高级版功能组件



13.3.1 云盾高级版功能

云盾高级版包含基础版所有功能，并在此基础之上增加了主机入侵防护（高级版）、DDoS清洗、云防火墙、Web应用防火墙、弱点分析和态势感知等功能。

13.3.1.1 网络流量监控

网络流量监控模块是阿里云安全自主研发的毫秒（ms）级攻击监控产品。通过对专有云入口镜像流量包的深度解析，实时地检测出各种攻击和异常行为，并与其他防护模块联动防护。网络流量监控模块在整个云盾防御体系中，提供了丰富的信息输出与基础的数据支持。

网络流量监控模块提供以下功能：

模块	功能项	说明
网络流量监控	DDoS攻击检测	通过流量镜像方式，旁路检测云边界流量中的DDoS攻击。
	流量统计	对云产品使用流量进行计量，生成流量图。
	网络层Web攻击拦截	根据内嵌的Web匹配规则，对常见的Web攻击进行网络层拦截、旁路阻断。
	IP黑名单	对添加的IP黑名单库进行旁路TCP阻断。
	恶意主机识别	对云计算平台内部的恶意主机进行识别和告警。

13.3.1.2 主机入侵防御（高级版）

主机入侵防御（高级版）模块采用C/S架构，其中云服务器上部署Client端负责信息搜集，由统一的Server端进行检测数据分析。

主机入侵防御（高级版）模块实现以下功能：

- 通过日志监控、文件分析、特征扫描等手段，提供账号暴力破解防护、WebShell查杀、异地登录报警等防入侵措施。
- 针对性防御SQL Server、MySQL、SSH、RDP、FTP等服务的暴力破解攻击。
- 主机异常登录报警。
- 主机高危漏洞的检测和修复。

主机入侵防御（高级版）模块在基础版功能之上增加了安全基线检查功能，您可以手动或自动对服务器做安全巡检。检测内容包括特定DDoS木马、可疑进程、系统帐户安全、数据库合规、文件访问控制、进程访问控制、文件完整性、Apache配置合规、弱口令等。

主机入侵防御（高级版）模块提供以下功能：

模块	功能项	说明
主机入侵防御	网站后门查杀	通过规则匹配和动态解析方式，对云服务器中存在的后门木马进行精准查杀。

模块	功能项	说明
	恶意文件基本查杀	基于恶意文件样本库，对服务器中存在的恶意文件或进程进行查杀和隔离。
	异地登录告警	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）。对疑似的非管理员登录系统行为通过手机短信进行告警。
	可疑账户检测	对服务器管理员的行为信息进行分析，对可疑的账户进行检测和告警。
	密码暴力破解拦截	对黑客进行暴力破解的行为进行实时检测和拦截，支持在Windows和Linux环境下 对SSH、RDP、FTP、MySQL、SQL Server等常见服务的暴力破解行为进行监控。
	漏洞扫描	基于主机扫描，发现主机漏洞并给出漏洞修复方式。
	漏洞修复	对云服务器中的应用和部分Windows系统的高危漏洞进行一键修复。覆盖范围包括：Web应用漏洞修复、系统文件修复等。
	安全基线	对主机端进行的所有操作进行审计记录。

13.3.1.3 弱点扫描

弱点扫描模块是阿里云自主研发，为构建在云服务器上的Web应用进行漏洞扫描的安全模块。弱点分析模块基于无状态扫描技术，并与网络流量安全监控联动，结合动态检测和静态匹配两种扫描模式，为您提供自动化、高性能的精准Web漏洞扫描能力。

弱点扫描模块具有以下特点和优势:

- **扫描速度快**
  - 采用无状态扫描技术，可在5 MB带宽条件下，并发每秒扫描10,000个IP地址。
  - 具备全网扫描能力，能在10个小时左右对全国互联网进行一次完整扫描。
  - 拥有对第三方专属漏洞扫描的专利技术，通过指纹识别的方式对第三方CMS系统进行快速扫描。
- **漏洞覆盖全**
  - 通过与网络流量监控模块联动，实时监控网络上的URL请求，对所有目标URL和接口进行全覆盖扫描。

- 支持针对30多种通用的Web漏洞、以及流行的150多种专属Web应用漏洞扫描，扫描范围覆盖检测的漏洞类型覆盖OWASP、WASC、CNVD的漏洞分类。
- 支持常见系统和数据库的弱口令扫描，如FTP、SSH、MySQL、SQL Server、MongoDB、Redis等。
- 支持恶意篡改检测，支持Web2.0、AJAX、各种脚本语言、PHP、ASP、.NET和Java等环境，支持复杂字符编码、Chunk、Gzip、Deflate等压缩方式、多种认证方式（Basic、NTLM、Cookie、SSL等）。
- 支持代理、HTTPS、DNS绑定扫描。

同时，借助阿里云大数据计算能力，云盾每天对阿里云平台上海量的攻击行为进行数据挖掘和分析，实时获取最新的攻击行为样本和情报及时发现0Day漏洞，形成安全漏洞库并应用到弱点分析系统上，进一步保障云盾弱点分析模块的及时性和全面性。

弱点扫描模块模块提供以下功能：

模块	功能项	说明
弱点分析	通用Web漏洞扫描	针对SQL注入、跨站攻击、文件包含、代码执行、信息泄露等漏洞进行扫描。
	第三方专属漏洞扫描	针对常用的第三方Web应用组件漏洞扫描，例如Discuz!，WordPress，DedeCms, Phpcms等。
	弱口令扫描	支持常见系统、应用和数据库的弱口令扫描功能，如RDP、FTP、SSH、MySQL、SQL Server、MongoDB等。
	网页恶意链接扫描	扫描Web页面，发现恶意植入的盗链。

13.3.1.4 安全审计

安全审计模块是基于云计算平台的一体化解决方案。对标信息系统安全等级保护基本要求，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，实现了行为日志的收集、存储、分析、报警等功能。

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宿主机，从各个角度对行为进行收集，确保了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。目前，已支持500亿条日志的同时索引。

安全审计模块提供以下功能：

模块	功能项	说明
安全审计	网络审计	对云平台网络设备的登录和操作进行审计。
	物理服务器审计	对云平台物理服务器的登录和操作进行审计。
		对平台内部API调用，命令执行的操作进行审计。
	云平台审计	对云平台内部的各个API调用操作审计。
	云产品审计	云产品ECS虚拟机登录审计。
		云产品RDS数据库操作审计。
		云产品ODPS的操作审计。

13.3.1.5 DDoS清洗

DDoS清洗模块是，阿里云基于自主开发的大型分布式操作系统和十余年安全攻防的经验，为广大云平台用户提供基于云计算架构设计和开发的云盾海量DDoS攻击防御产品。

DDoS清洗模块具有以下特点和优势:

- **全面覆盖常见DDoS攻击类型**

  DDoS清洗模块帮助您抵御各类基于网络层、传输层及应用层的各种DDoS攻击（包括CC、SYN Flood、UDP Flood、UDP DNS Query Flood、(M)Stream Flood、ICMP Flood、HTTP Get Flood等所有 DDoS 攻击方式），并实时短信通知您的网站防御状态。

- **快速自动响应，一秒内进入防护状态**

  DDoS清洗模块采用全球领先的检测和防护技术，能在一秒内完成攻击发现、流量牵引和流量清洗全部动作。在防护触发条件上，不仅依赖流量阈值，还对网络行为进行统计判断，精准识别DDoS攻击，大幅减少网络抖动现象，全面保障您在遇到DDoS攻击时业务的持续性。

- **高弹性、高冗余的DDoS防御能力**

DDoS清洗模块每个最小单元支持20 Gbps的攻击流量过滤。得益于云计算架构的高弹性和大冗余特点，DDoS攻击防御模块可在云环境内部中无缝扩容，实现DDoS攻击防御能力的高弹性。

在专有云环境中部署的DDoS清洗模块支持与阿里云DDoS攻击高防服务联动，可将DDoS攻击防护能力最大扩容到1,000+ Gbps，避免因专有云出口带宽限制导致的防护性能瓶颈。

- **双向防护，避免云资源被滥用**

  DDoS清洗模块不仅防护来自于云外的DDoS攻击，还能及时发现云内资源被滥用的非法行为。一旦发现云内有服务器被利用向外发起DDoS攻击，网络流量监控模块与主机安全入侵防御模块联动，限制被滥用的云服务器的网络访问行为，并产生告警，实现对内部主机的有效管控。

DDoS清洗模块提供以下功能：

模块	功能项	说明
DDoS清洗	海量DDoS清洗能力	完美防御SYN Flood、ACK Flood、ICMP Flood、UDP Flood、NTP Flood 、SSDP Flood、DNS Flood、HTTP Flood、CC攻击。
	应用层DDoS防护	具备应用层抗DDoS攻击的能力，通过重认证、身份识别、验证码等多种手段精确识别恶意访问和真实访问者，针对网站类CC和游戏类CC攻击均可防御。
	弹性扩展	与阿里云DDoS攻击高防服务联动，将DDoS攻击防护能力最大扩容到1000+ Gbps。

13.3.1.6 Web应用防火墙

Web应用防火墙（简称WAF），是阿里云自主研发的一款网站安全防护产品，它能够保护网站的应用程序避免遭受常见Web漏洞的攻击。这类攻击既有诸如SQL注入、XSS跨站脚本等常见Web应用攻击，也有CC攻击这种影响网站可用性的资源消耗型攻击。同时，WAF模块也允许根据网站实际业务制定精准的防护策略，用于过滤对您网站有恶意的Web请求。

WAF模块防护的流量定位在HTTP/HTTPS的网站业务上，允许您在WAF的管理界面中自主导入证书与私钥，从而实现业务的全链路加密，避免数据在链路中被监听的可能，同时也满足了您对HTTPS业务的安全防护需求。

WAF模块的防护体系主要分为两部分：

- 阿里云的大数据分析平台：基于阿里云的大数据核心竞争力建立的威胁情报库与网站可信模型，能够轻松识别正常、异常的流量。

- 攻击特征匹配与统计分析的安全防护策略：WAF自身内置的通用防护规则，能够正常检测并阻断出OWASP Top 10中常见的Web漏洞攻击；同时，精准防护的能力能够为您量身打造属于网站定制防护策略，过滤网站指定的恶意Web请求流量。

WAF模块支持防护场景下的规则排序，以及精准防护与其他安全防护策略的生效关系调整（即，匹配精准防护规则后，是否还要继续过CC与Web通用防护策略）。精准防护作为自定义添加的防护策略，在请求的匹配上优先级永远排在第一。

WAF模块提供以下功能：

模块	功能项	说明
Web应用防火墙（WAF ）	Web常见攻击防护	防御OWASP 常见威胁；针对GET、POST常见HTTP请求，针对不同的网站业务，提供高、中、低三种规则策略，实现SQL注入、XSS跨站、WebShell上传、后门隔离保护、命令注入、非法HTTP协议请求、常见Web服务器漏洞攻击、核心文件非授权访问、路径穿越、扫描防护等安全防护功能。
	缓解CC攻击	对单一源IP的访问频率进行控制、重定向跳转验证、人机识别等。
		针对海量慢速请求攻击、识别异常响应码、IP访问、URL异常分布、异常Referer、User-agent的请求，结合精确访问控制过滤。
		充分利用阿里云大数据安全优势，建立威胁情报与可信访问分析模型，快速识别恶意流量。
	精准访问控制	提供友好的配置控制台界面，支持IP、URL、Referer、User-Agent等HTTP常见字段的条件组合，打造强大的精准访问控制策略，并支持盗链防护、网站后台保护等防护场景。
		与Web常见攻击防护、CC防护等安全模块打造多层综合保护机制；轻松依据需求，识别可信与恶意流量。

13.3.1.7 云防火墙

云防火墙模块是阿里云自主研发，基于云计算环境东西向流量微隔离需求的云访问控制系统。

云防火墙模块提供以下功能：

模块	功能项	说明
云防火墙	微隔离	在复杂的云环境里，微隔离是安全的基础设施，为您带来基本的业务安全域管理。
	可视化	在可视环境下（业务流向的可视），您可以快速完成精细化的微隔离部署，并进行策略调整和优化。
	基于角色+标签的资产定义	去IP化的资产定义，更加贴近您的业务，让您实现基于角色+标签的资产管理。
	分布式架构	无需引流，彻底摆脱云计算平台集中式防火墙的引流之苦；适应各种虚拟化环境。

13.3.1.8 态势感知

态势感知是一个大数据安全分析平台，它通过机器学习和数据建模发现潜在的入侵和攻击威胁，从攻击者的角度，有效捕捉高级攻击者使用的0Day漏洞攻击、新型病毒攻击事件，以及正在发生的安全攻击行为有效的展示，帮助您实现业务安全可视和可感知，解决因网络攻击导致数据泄露的问题，并通过溯源服务追踪黑客身份。

日志采集

态势感知日志采集分两大类：一类是业务级数据采集，分别为HTTP流量采集、五元组数据采集、主机Syslog日志采集；另一类是阿里云产品采集，分别为ECS实例登录日志采集，ECS实例操作命令日志采集，RDS实例执行语句采集，ODPS实例操作日志采集。

态势感知，通过应有规则分析、机器学习建模分析两种不同角度的分析方法，将采集到的日志进行聚合分析。值得一提的是，态势感知系统的威胁分析、以及紧急事件判定，有60%以上的结果都来自于机器学习。

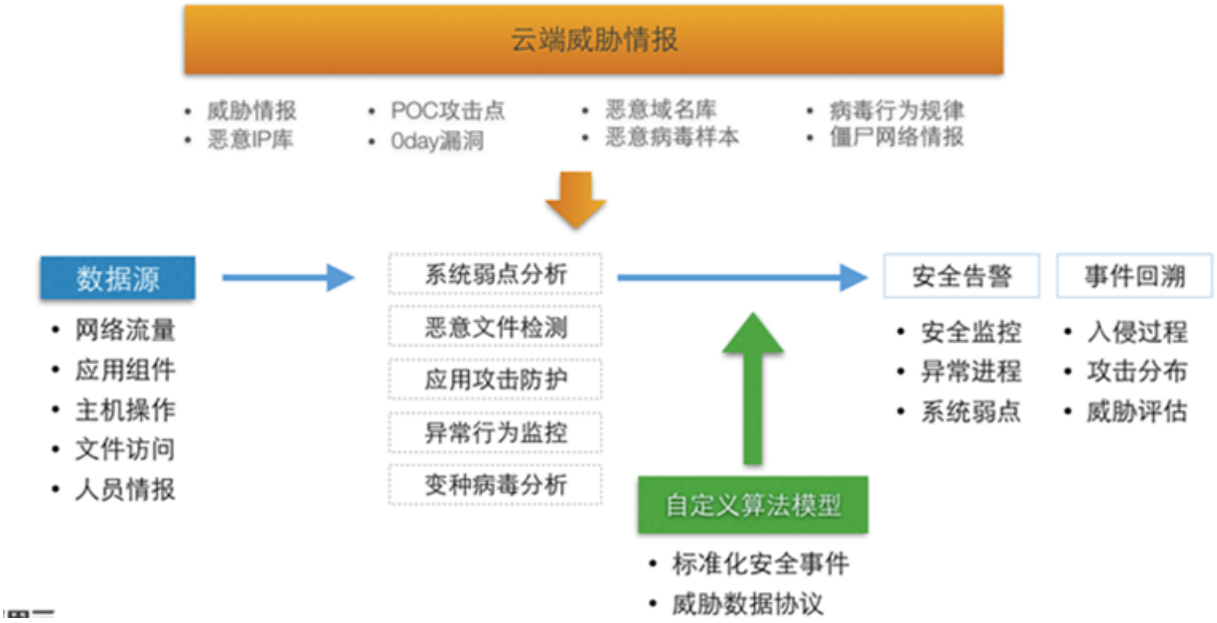
情报来源

态势感知威胁情报体系来源分两类：一类是通过数据分析得出的情报，如发起针对性攻击的恶意IP库、黑客攻击手段、黑客信息等；另一类是通过外部收集，如IP信誉库、0Day漏洞库、病毒、后门、弱口令库等已知的样本信息。

产品结构

态势感知系统与云盾各个功能模块之间的结构如下图：

图 31: 态势感知与其它云盾功能模块结构



功能特性

态势感知具有以下功能：

- 对已知入侵威胁安全态势的感知。
- 对未知威胁安全态势的感知。
- 对恶意文件（WebShell、木马、恶意执行脚本）的态势感知。
- 对资产自身脆弱性的态势感知。
- 对业务系统的安全态势监控和展示。

态势感知具有以下特点和优势：

- **大数据威胁分析**

  作为一个集合了大数据和安全的跨界产品，态势感知不仅拥有PB级别的大数据分析和计算能力，而且通过机器学习，汇集全网安全数据和威胁情报，建立了完整的智能的安全威胁模型，并作用到百万客户的实际业务场景中。通过对海量数据的收集、分析与展现，帮助您获得无与伦比的全局可见性和安全智能性，抵御来自各个维度和领域的新型安全威胁。

  态势感知用大数据分析的方法，用智能化的机器学习和建模分析，聚焦数据中心云计算用户面临的定向 Web 应用攻击、面向系统的暴力破解、黑客入侵行为、应用层主机层漏洞等多个方面的新威胁和新的安全趋势。

- **大屏展现**

  云盾态势感知基于互联网可视化技术，将大数据威胁分析成果以直观的图形呈现于大屏上，作为云计算平台安全决策的支撑工具。

13.4 产品价值

13.4.1 云环境下的安全威胁

云计算的虚拟化资源池、弹性架构、服务可度量、灵活接入和按需服务等特性让计算资源（包括网络，服务器，存储，应用软件，服务）变得像自来水一样随时、随地、随需可得，极大的优化了IT资源效率，但同时也对云上用户的IT系统安全性提出了新的挑战。

13.4.1.1 DDoS攻击威胁

分布式拒绝服务攻击（Distributed Denial of Service，简称DDoS攻击）是对云计算环境影响最大的系统可用性威胁， 2011年至2013年连续三年被云安全联盟（Cloud Security Alliance，简称CSA）收录为云端十大安全威胁之一。

DDoS攻击在云端的表现：攻击者利用互联网上大量存在的僵尸网络主机，向云上服务器发起大量的正常服务请求导致云服务器过载，从而影响正常用户的访问。

大量的DDoS攻击，轻则导致被攻击的云服务器无法正常提供服务，影响客户的在线业务，重则导致整个云环境网络不稳定，影响云环境的可用性。另一方面，云平台本身拥有非常强大的计算能力和基础带宽资源，攻击者也可以通过利用云上服务器发起DDoS攻击。

DDoS攻击防护的最佳安全实践是通过云防御来解决。

13.4.1.2 网络入侵威胁

云计算平台是互联网的基础设施，用户的业务都是以数据的形式承载于云计算平台上，数据是云平台上最重要的资产。攻击者往往以数据为攻击目标，通过各种网络渗透攻击手段获取或者篡改客户的业务数据，从而达到非法目的。因此，在云平台上防御攻击者的入侵行为是云计算环境下保护用户业务系统安全的重中之重。

在云环境中常见的网络入侵行为有：

- 攻击者通过暴力破解或者其他方式获得操作系统、服务或者Web应用的访问权限，从而非法登录系统，直接获取敏感数据。或者在登录系统后直接篡改系统中的敏感数据以达到个人非法目的。
- 攻击者利用系统或者Web应用漏洞发起远程攻击，窃取敏感数据。例如，攻击者利用Web系统漏洞，上传WebShell后门程序，从而获得服务器操作权限，利用系统权限直接下载数据库文件（即，拖库攻击）。攻击者通过同样的手段也可以直接修改系统中的敏感数据。例如，利用Web应用漏洞发起SQL注入攻击，直接篡改数据库中存储的敏感数据。

- 攻击者在云服务器上放置监听程序，监听网络中的数据包，从而获取敏感数据。同样的，攻击者在云服务器上放置WebShell等木马后门程序，利用所获得的信息发起中间人攻击，或者钓鱼攻击。

13.4.1.3 内部威胁

随着业务系统访问、网络应用行为日益频繁，系统维护人员能够直接接触重要业务系统，产生内部威胁的概率也越来越高。这些内部行为，安全防护体系往往不能及时发现、定位源头，给企业带来了极大的困扰。

常见的内部威胁有：

- 内部系统维护人员对业务应用系统的越权访问、违规操作，损害业务系统的运行安全。
- 重要业务数据库，被员工或系统维护人员篡改牟利、外泄，给企业造成巨大的经济损失。

13.4.2 云盾产品价值

云盾在阿里云专有云出口，通过网络流量监控系统，在网络层对恶意的攻击行为进行识别，实时地阻断网络攻击行为。在主机层对Web木马和恶意文件进行实时查杀，避免云服务器被攻击者利用。实时拦截暴力破解行为，并对异常的登录行为进行告警，避免攻击者利用弱口令登录系统窃取或者破坏客户业务数据。

云盾由多个功能模块组成，在专有云网络出口、专有云网络中、专有云服务器上实现纵深防御，多点联动。为了方便您集中管理和实时掌握云平台安全风险，云盾提供了统一的管理视图，您可以在云盾集中管控系统上对所有安全防护模块中的安全策略进行统一管理，同时还可以在集中管控系统上对日志进行关联分析。

纵深防御的安全体系架构

云盾由涵盖网络安全、主机安全、应用安全、弱点分析等多层次安全防护模块组成，在云边界、云网络中、云服务器上形成一套纵深的防御体系，通过集中管控的管理中心协调调度，综合各模块提供的安全信息，做出最准确的判断，并且可以在最合适的位置检测和阻断恶意的攻击行为，有效地保护了云环境不受外界攻击者的侵扰，保障用户业务系统的安全。

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，可有效地保护公有云和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；云边界、云网络上防御模块采用的是旁路的架构，贴切云的业务，对云平台业务影响最小化；云服务器上的防御模块是虚拟化，适应虚拟机灵活的特点。

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，云盾也提供了租户自服务的控制台Portal，租户可以在上面查看自己相关的安全防护情况，生成简单的报表。利用合理的外部资源配置可以自动化的接收告警短信和邮件。

阿里云安全能力输出

云盾的防护策略和数据源自于多年的积累，阿里云上多达百万级的用户，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，每天对阿里云上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，为您带来更好的安全保障。

14 企业级分布式应用服务EDAS

14.1 产品概述

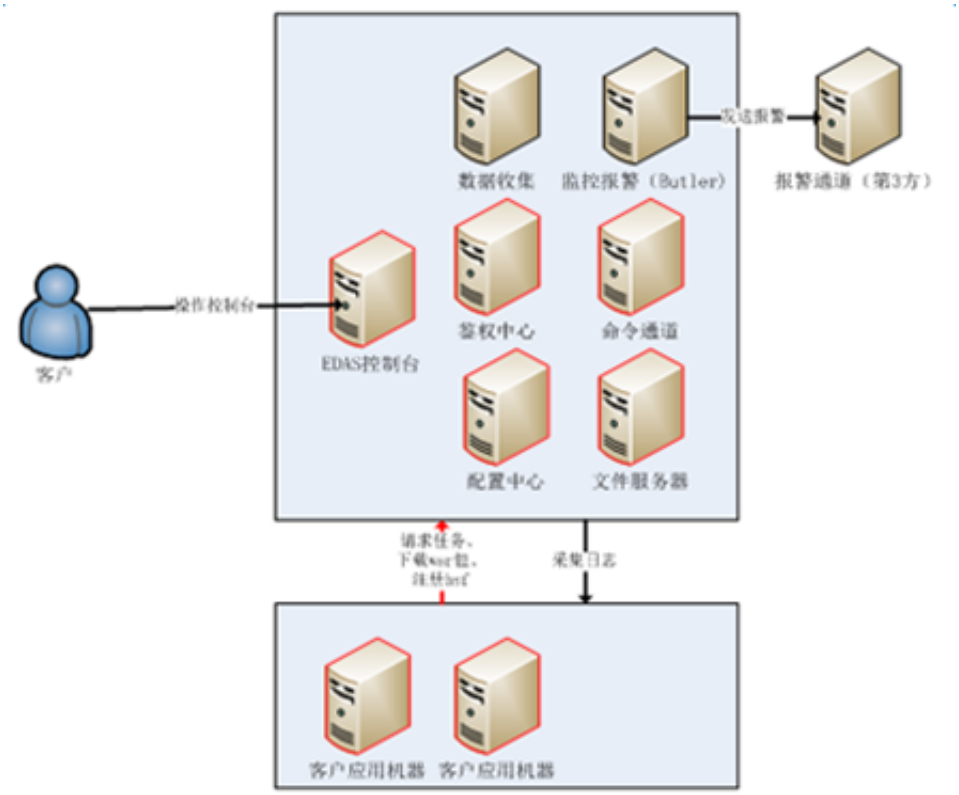
企业级分布式应用服务（Enterprise Distributed Application Service，简称 EDAS）是企业级互联网架构解决方案的核心产品。EDAS 充分利用阿里云现有资源管理和服务体系，引入中间件成熟的整套分布式计算框架（包括HSF 或 Dubbo 分布式服务化框架、服务治理、运维管控、链路追踪和稳定性组件等），以应用为中心，帮助企业级客户轻松构建并托管分布式应用服务体系。

图 32: EDAS产品示意图

14.2 产品架构

企业级分布式应用服务EDAS由控制台、数据采集系统、配置注册中心和鉴权中心等系统组成，整体系统架构如图 *33: EDAS系统架构*所示。

图 33: EDAS系统架构



EDAS控制台

EDAS控制台是使用EDAS系统功能的操作界面，是唯一可以让您直接使用的系统。通过控制台可以实现资源管理、应用生命周期管理、运维管控及服务治理、立体化监控及数字化运营等。

EDAS控制台包含两个组件，EDAS Console和EDAS Admin。

- Console是供客户访问的操作界面。
- Admin组件主要用来执行后台定时任务，例如：定时同步ECS数据、定时自动扩缩容等。

数据采集系统

数据收集系统负责实时收集EDAS集群及所有客户应用机器的系统运行状态，调用链日志等，并进行实时汇总计算存储到HBase及HiStore中。HBase存储实时计算的结果，HiStore存储调用链的详细数据，存储的数据作为监控报警及调用链查看的基础数据。

数据收集系统包含采集配置中心、JStorm实时采集节点、HBase、HiStore。

- 采集配置中心，主要是用来配置采集规则、采集的切分规则、采集的目标节点等。配置完成之后生成采集任务推送到ZooKeeper 。
- JStorm实时采集节点，是任务真正执行的节点，任务会分发到每个采集节点执行，采集时会主动访问客户应用机器的8182端口进行日志数据的拉取，日志拉取后会进行实时的分析和计算。

- HBase用于存储实时计算后的各种数据，目前默认保留2000小时，在实际场景中可根据您的需求进行调整。
- HiStore用于存储调用链的详情日志，以供调用链查询，目前默认保留7天时间。

运维监控系统

Butler系统是EDAS对外输出的主要日常监控及报警工具，提供EDAS所有组件的日常巡检及报警等工作。在基础硬件及网络完善的情况下，可以实时监控EDAS系统各个组件的运行状态，如果发现异常可以触发报警通知运维人员及时进行故障排查。

14.2.1 EDAS控制台

EDAS控制台是供用户使用EDAS系统功能的操作界面，是唯一可以让客户直接使用的系统。用户通过控制台可以实现资源管理、应用生命周期管理、运维管控及服务治理、立体化监控及数字化运营等。

EDAS控制台包含2个组件，EDAS Console和EDAS Admin。

- Console是真正供客户访问的操作界面。
- Admin组件主要用来执行后台定时任务，例如：定时同步ECS数据、定时自动扩缩容等。

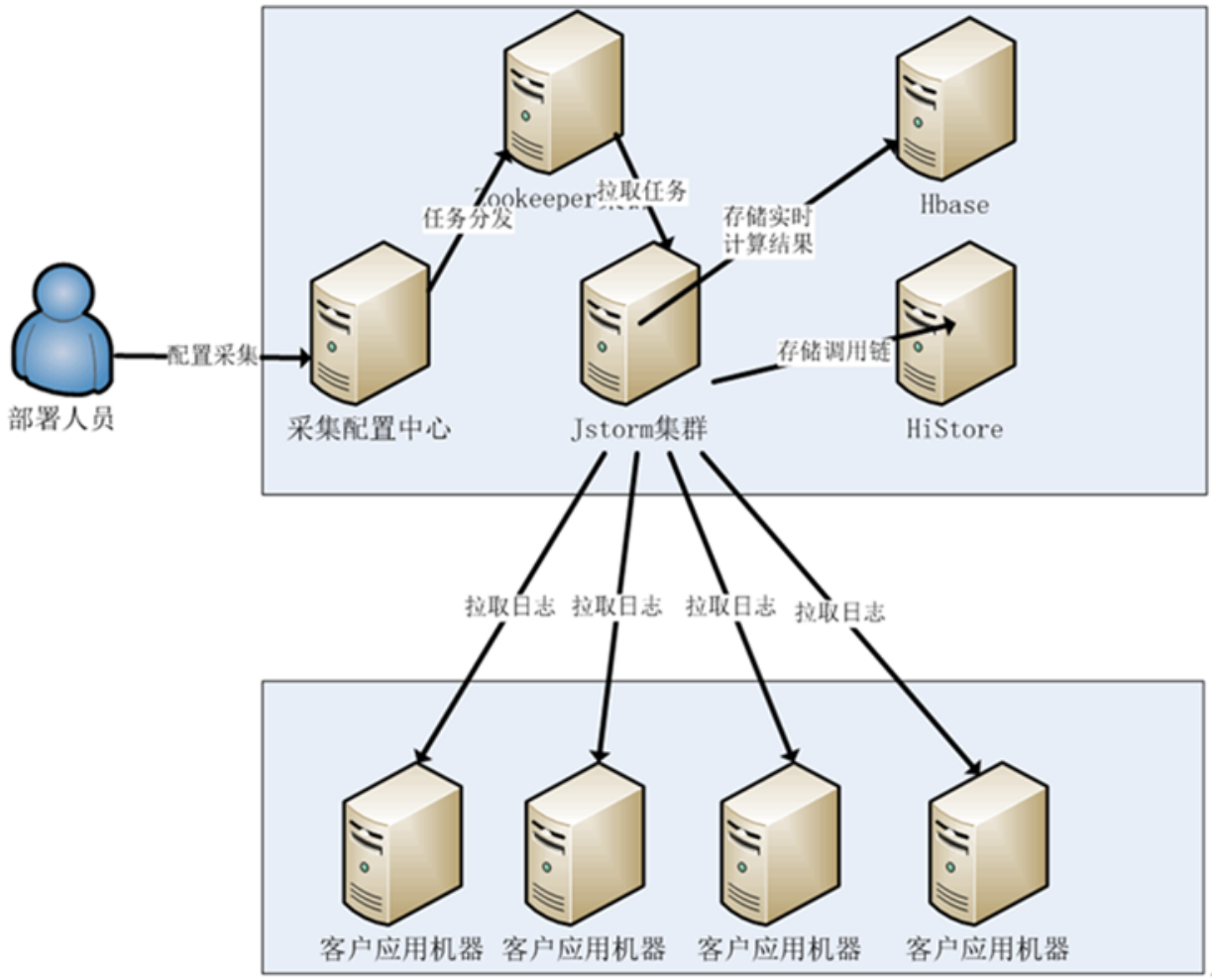
14.2.2 数据收集系统

数据收集系统负责实时收集EDAS集群及所有客户应用机器的系统运行状态，调用链日志等，并进行实时汇总计算存储到HBase及HiStore中。HBase存储实时计算的结果，HiStore存储调用链的详细数据，存储的数据作为监控报警及调用链查看的基础数据。

数据收集系统包含采集配置中心、JStorm实时采集节点、HBase和 HiStore。

- 采集配置中心，主要是用来配置采集规则、采集的切分规则、采集的目标节点等。配置完成之后生成采集任务推送到ZooKeeper。
- JStorm实时采集节点，是任务真正执行的节点，任务会分发到每个采集节点执行，采集时会最主动访问客户应用机器的8182端口进行日志数据的拉取，日志拉取后会进行实时的分析和计算。
- HBase用于存储实时计算后的各种数据，目前默认保留2000小时，对外输出时根据情况进行调整。
- HiStore用于存储调用链的详情日志，用于调用链查询，目前默认保留7天时间。

图 34: 数据收集流程图



14.2.3 运维（Butler）系统

Butler系统是EDAS对外输出的主要日常监控及报警工具，提供EDAS所有组件的日常巡检及报警等工作。在基础硬件及网络完善的情况下，可以实时监控EDAS系统各个组件的运行状态，如果发现异常可以触发报警通知运维人员进行及时故障排查。

14.2.4 配置注册中心系统

配置注册中心是HSF（RPC框架）服务发布及订阅的中心服务器，也是分布式配置配置推送的中心服务器。

配置注册中心包含地址服务器（Address Server）、服务注册中心（ConfigServer）、配置推送中心（Diamond Server）：

- 地址服务器，底层是Tengine，配置了服务注册中心和配置推送中心的地址列表。HSF或配置推送时，首先需要连接地址服务器获取对应的地址列表后，才能通信。

- 服务注册中心，HSF服务提供者发布的服务注册到本中心，HSF消费者从本中心订阅对应的服务，获取提供服务的IP列表，进行服务调用。
- 配置推送中心，提供了配置信息管理功能，通过配置客户端发布和订阅相应信息，确保配置信息在多个系统保持实时同步。

14.2.5 鉴权中心系统

为保证各个用户直接的数据安全，使用鉴权系统对用户的数据进行权限控制。用户登录也使用鉴权系统的单点登录系统进行登录。

鉴权系统需要依赖地址服务器、配置推送中心，目前可以和配置注册中心共用这2个组件。

14.2.6 命令通道系统

命令通道系统是远程发送相关指令到客户应用机器执行控制中心。

命令通道系统包含操作控制台、命令通道管理节点、命令通道服务器：

- 操作控制台是供运维人员登录进行机器状态查询或者管理的操作界面。
- 管理节点提供链接管理分配功能，客户端首先访问管理节点获取可以长链接的服务器，然后再与服务器建立长连接。
- 服务器是真正与客户端产生长链接并进行命令下发的节点。

14.2.7 文件系统

文件系统用于存放客户上传的WAR包及JDK、Ali-Tomcat等必须组件。

专有云部署时，使用Nginx+FTP搭建的文件系统。文件系统内上传的WAR包每个应用只会保留7个最新的。

14.3 功能特性

14.3.1 全面兼容Apache Tomcat容器

作为EDAS平台应用运行的基础容器，EDAS Container集成了阿里巴巴中间件技术栈，在容器启动、容器监控、稳定性及性能上得到了极大的提升。同时，EDAS Container全面兼容Apache Tomcat。

14.3.2 以应用为中心的中间件PaaS平台

应用基本管理和运维

在EDAS平台可视化的管控界面上，您可以一站式完成应用生命周期的管控，包括创建、部署、启动、停止、扩容、缩容和应用下线等，实现对应用的全流程管理。依托阿里巴巴平台超大规模集群运维管理经验，轻松运维上千个实例的应用。

弹性伸缩

EDAS支持手动和自动两种方式来实现应用的扩容与缩容；通过对CPU、内存和负载的实时监控来实现对应用的秒级扩容和缩容。

主子账户体系

EDAS独创主子体系，你可以根据自己企业的部门划分、团队划分和项目划分在EDAS平台上建立对应的主子账号关系；同时，ECS资源也以主子账号关系进行划分，以便进行资源的分配。

角色与权限控制

应用的运维通常涉及应用研发负责人、应用运维负责人和底层机器资源负责人。不同的角色对于一个应用的管理操作各不一致，因此EDAS提供了角色和权限控制机制，方便您为不同的账号定义各自的角色，并分配相应的权限。

14.3.3 丰富的分布式服务

分布式服务框架

自2007年，伴随着阿里巴巴电商平台大规模分布式改造的持续进行，自主研发的分布式服务框架HSF（High Speed Framework）和Dubbo应运而生。HSF是一款面向企业级互联网架构的分布式服务框架，以高性能网络通信框架为基础，提供了诸如服务发布与注册、服务调用、服务路由、服务鉴权、服务限流、服务降级和服务调用链路跟踪等一系列久经考验的功能特性。

分布式配置管理

集中式系统变成分布式系统后，如何有效的对分布式系统中每一个机器上的配置信息进行有效的实时管理成了一个难题。EDAS提供高效的分布式配置管理，能够将分布式系统的配置信息在EDAS控制台上集中管理起来，做到一处配置，处处使用。更重要的是，EDAS允许您在控制台上对配置信息进行修改，在秒级时间内就能够实时通知到所有的机器。

分布式任务调度

任务调度服务允许您配置任意周期性调度的单机或者分布式任务，并能对任务运行周期进行管理，同时提供对任务的历史执行记录进行查询。适用于诸如每天凌晨2点定时迁移历史数据，每隔5分钟进行任务触发，每个月的第一天发送系统月报等任务调度场景。

14.3.4 运维管控与服务治理

服务鉴权

HSF服务框架致力于保证每一次分布式调用的稳定与安全。在服务注册、服务订阅以及服务调用等每一个环节，都进行严格的服务鉴权。

服务限流

EDAS可以对每一个应用提供的众多服务配置限流规则，以实现对服务的流控，确保服务能够稳定运行。限流规则可以从QPS和线程两个维度进行配置，确保系统再应对流量高峰时能以最大的支撑能力平稳运行。

服务降级

与服务限流相反，每一个应用会调用许多外部服务，对于这些服务配置降级规则可以实现对劣质服务的精准屏蔽，确保应用自身能够稳定运行，防止劣质的服务依赖影响应用自身的服务能力。EDAS从响应时间维度对降级规则进行配置，在应对流量高峰时合理地屏蔽劣质依赖。

14.3.5 立体化监控与数字化运营

分布式链路跟踪

EDAS鹰眼监控系统能够分析分布式系统的每一次系统调用、消息发送和数据库访问，从而精准发现系统的瓶颈和隐患。

服务调用监控

EDAS能够针对应用的服务调用情况，对服务的QPS、响应时间和出错率进行全方面的监控。

IaaS基础监控

EDAS能够针对应用的运行状态，对机器的CPU、内存、负载、网络和磁盘等基础指标进行详细的监控。

14.4 性能指标

指标项	规格要求
访问处理性能	在简单调用场景下，不考虑不确定的服务提供方响应时间，1KB单个服务请求消息大小，RPC服务调用每CPU核QPS 4000。可线性扩展，单个注册中心支持20000个。
基本功能	提供集成多款互联网中间件的Java容器，全面兼容Apache Tomcat。
	提供应用生命周期管理，包括应用发布、启动、停止、扩容等。
	提供对硬件基础指标的监控。
	提供对Java容器的监控。
	提供完善的服务鉴权机制。
	提供分布式RPC调用，消息和多种数据源的事务处理。
	提供完整的针对服务链路和系统指标的日志、巡检、监控和链路跟踪。
	提供分布式系统配置推送。
可靠性	数据系统采用多级缓存和主备存储方案。
高可用性	组件集群化，包括负载均衡、网关、缓存、数据库服务节点、数据节点，可用性不低于99.9% 。
扩展性	支持服务节点的无间断扩容与缩容。
技术成熟	基于阿里内部长期使用与沉淀的高可用高性能分布式集群技术产品构建，团队成员具有处理这一领域问题的丰富经验。

15 分布式关系型数据库DRDS

15.1 产品概述

分布式关系型数据库服务（ Distributed Relational Database Service，简称DRDS ）是阿里巴巴集团自主研发的中间件产品，专注于解决单机关系型数据库扩展性问题，具备轻量(无状态)、灵活、稳定、高效等特性。DRDS兼容MySQL协议和语法，支持分库分表、平滑扩容、服务升降配、透明读写分离和分布式事务等特性，具备分布式数据库全生命周期的运维管控能力。

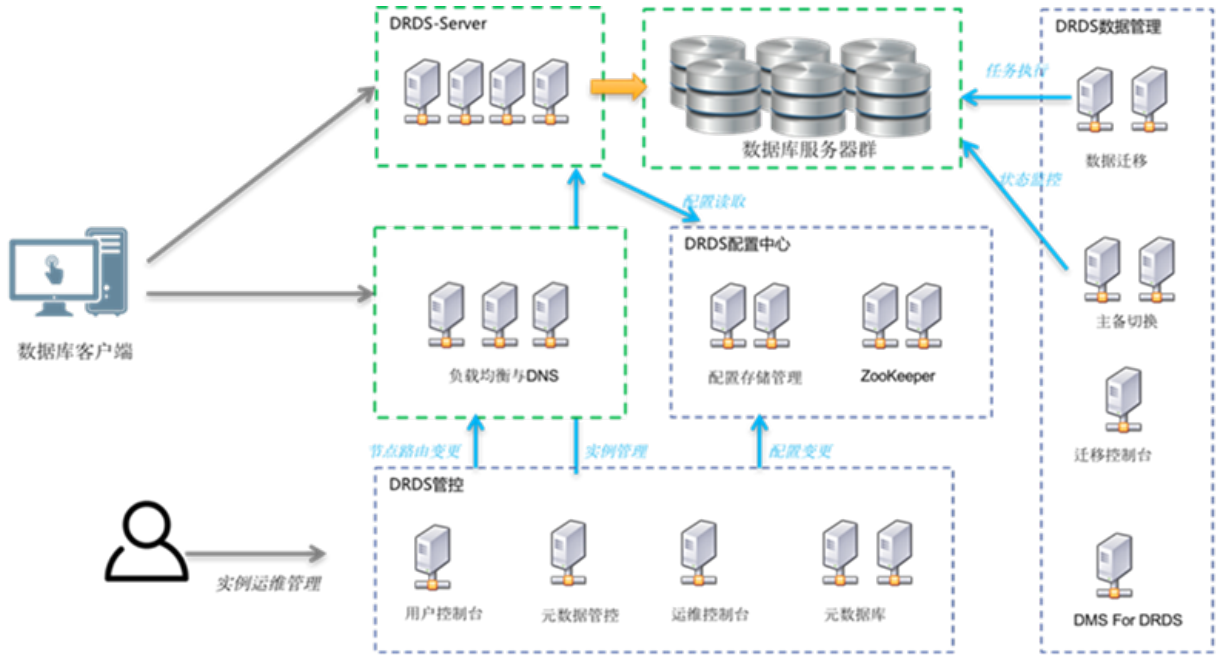
DRDS实现了一套独立逻辑的SQL解析、优化、执行、结果处理的数据库服务，与淘宝TDDL产品共享分布式数据库拆分逻辑和经验，实现常用聚合函数、ORDER BY/GROUP BY/LIMIT M,N 、分布式JOIN、子查询等分库分表下的支持。

DRDS主要应用场景在大规模在线数据操作上，通过贴合业务的拆分方式，将操作效率提升到极致，有效满足用户在线业务对关系性数据库要求。

15.2 产品架构

分布式关系型数据库服务DRDS的系统架构如图 *35: DRDS系统架构图*所示。

图 35: DRDS系统架构图



DRDS Server

DRDS Server是DRDS的服务层，由多个服务节点组成服务集群，提供分布式数据库服务，包括读写分离、SQL路由执行、结果合并、动态数据库配置、全局唯一ID服务等功能。

数据库服务集群

DRDS的数据库服务集群主要是负责基础的数据存储，如MySQL。通过MySQL主备复制实现高可用，配合数据管理的主备切换系统实现动态数据库故障转移。

负载均衡与DNS

DNS是域名解析服务。DNS将底层DRDS实例集群的IP屏蔽，通过域名的方式对外服务。用户的请求通过DNS解析到实例集群IP，通过均衡负载进行流量的分配。当某个实例节点出现故障或者新增服务节点时，都能够通过均衡负载保证底层节点的流量均衡分配。

DRDS配置中心

DRDS配置中心是负责配置存储和管理的系统，提供配置存储、查询、通知功能，在DRDS中主要存储数据库源数据、拆分规则、DRDS开关等配置。

15.2.1 DRDS数据管理

DRDS数据管理模块由数据迁移、主备切换、迁移控制台和DMS For DRDS组成，能够实现对数据库数据的常规管理以及数据安全功能的支持。

数据迁移

数据迁移模块包含全量数据迁移和增量数据迁移模块，实现源数据到目的数据的平滑迁移。数据库扩容和小表广播的等场景都需要借助数据库迁移模块完成。

数据迁移控制台

数据迁移控制台主要负责对数据迁移任务的管理和监控，能够帮助运维人员及时发现数据迁移任务的异常。控制台主要功能有：数据迁移任务的执行状态展示、执行时间统计、执行进度、异常状态报警。

主备切换

主备切换是自主研发的MySQL主备切换系统。该系统通过数据库机器上安装Agent ，执行心跳SQL等多种手段，综合判定MySQL主备活性，并根据这个判定结果执行主备切换。

DMS For DRDS

数据管理服务（Data Management Service，简称DMS）是图形化的分布式数据库运维平台，支持建表、表结构变更、数据增删改查等可视化功能。

15.2.2 DRDS管控

DRDS管控是提供给DBA以及DRDS集群管理人员的运维管控系统。

用户控制台

DRDS用户控制台，提供实例管理、库表管理、读写分离配置、平滑扩容、DRDS服务监控展现、IP白名单安全功能。

元数据管控

元数据管控是DRDS的运维支撑系统，支持进行数据库配置、读写权重、连接参数等管理以及库表拓扑、拆分规则等管理。

运维控制台

运维控制台是DRDS实例管理控制台，提供以下功能：

- DRDS实例依赖的所有资源管理，包括虚拟机、负载均衡、域名等资源；
- DRDS实例创建、修改、销毁的功能；
- DRDS实例状态监控，包括QPS、活跃线程、连接数、各节点网络IO、各节点CPU占用等指标。

元数据库

管控信息的存储数据库，负责存储数据实例的配置信息并同步到配置中心，进行配置推送。

15.3 功能特性

15.3.1 数据拆分

DRDS核心架构原理是数据拆分。DRDS将数据库数据拆分后存储到多个单机数据库上，对外保持逻辑的一致性。拆分后的数据库称为分库，对应的表称为分表。每个分库负责一份拆分数据的读写操作，分散整体访问压力。DRDS通过增加分库的数量和迁移相关数据，实现数据库的扩容，提高DRDS系统的总体容量。

数据拆分需要选择拆分维度，作为数据分布的依据。

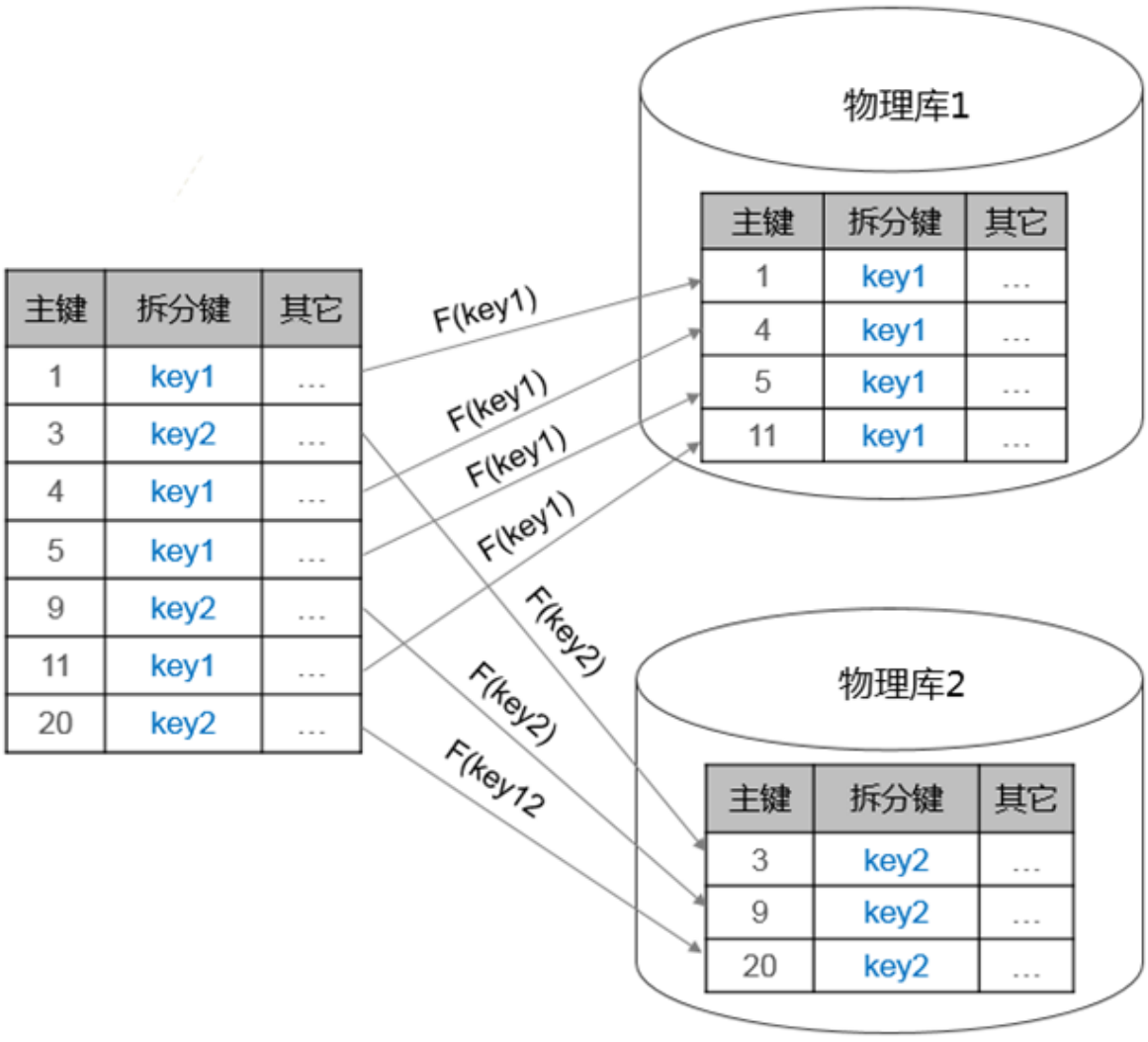
例如一个电商订单信息表，如果按照订单ID做数据拆分，那么相同订单ID的数据就会被拆分到同一个数据库存储节点上；如果按照用户ID做数据拆分，那么同一个用户的订单就会分布到同一个数据库存储实例存储节点上。

拆分维度要根据实际业务的场景确定，有如下几个指导原则：

- 最大程度保证数据库节点的数据量和访问量均衡。
- 单条SQL操作尽量落到单个物理数据库节点上执行。
- 不同SQL的查询落到不同的数据库节点上。
- 减少多个节点之间的网络传输。
- 易于扩展。

数据拆分如图 *36:* 数据拆分图所示。

图 36: 数据拆分图



15.3.2 平滑扩容服务

DRDS扩容通过增加RDS/MySQL实例数，将原有的分库迁移到新的RDS/MySQL实例上，达到扩容的目标。

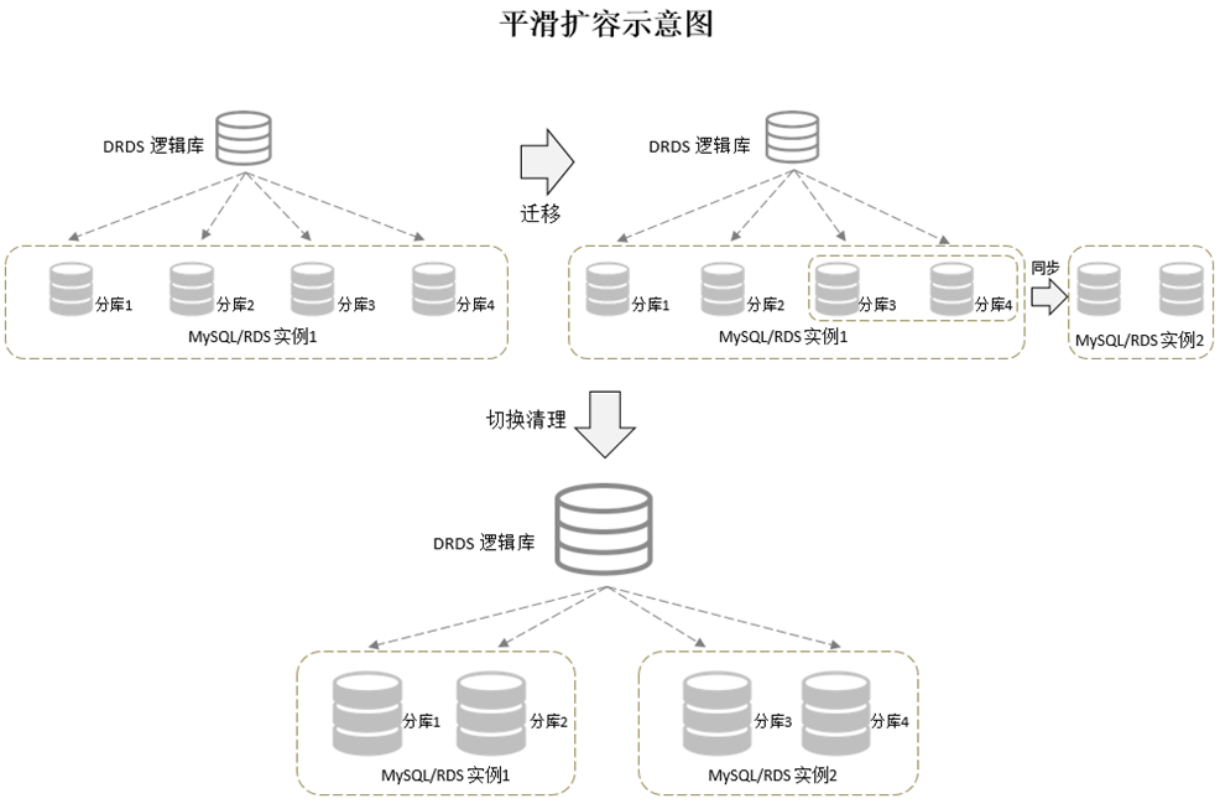
DRDS扩容原理

步骤如下：

1. 创建扩容计划，在新增加的RDS/MySQL实例上创建新的分库。
2. 全量迁移，系统选择当前时间之前的一个时间点，将这个时间点之前的数据进行全量的数据复制迁移。
3. 增量数据同步，完成全量迁移后，基于全量迁移时间点之后的增量变更日志进行增量同步，最终原分库和目标分库数据实时同步。
4. 应用停写和路由切换，增量达到实时同步后，业务选定时间进行切换，为确保数据严格一致，建议应用停服(也可以不停，但可能面临同一条数据高并发写入覆盖问题)，引擎层进行分库规则的路由切换，将后续流量转向新库，切换过程秒级完成。

分库迁移示意图如图 *37:* 扩容示意图所示。

图 37: 扩容示意图



扩容平滑切换

为了保证数据本身的安全，便于扩容回滚，在路由规则切换完成后，数据同步依然会运行，直到数据运维人员确认服务正常后在控制台主动发起旧分库数据的清理。

整个扩容过程对上层的业务正常服务基本没有影响，切换时如果应用不停服，建议操作选择在数据库访问低谷期进行，降低同一条数据并发更新覆盖的概率。

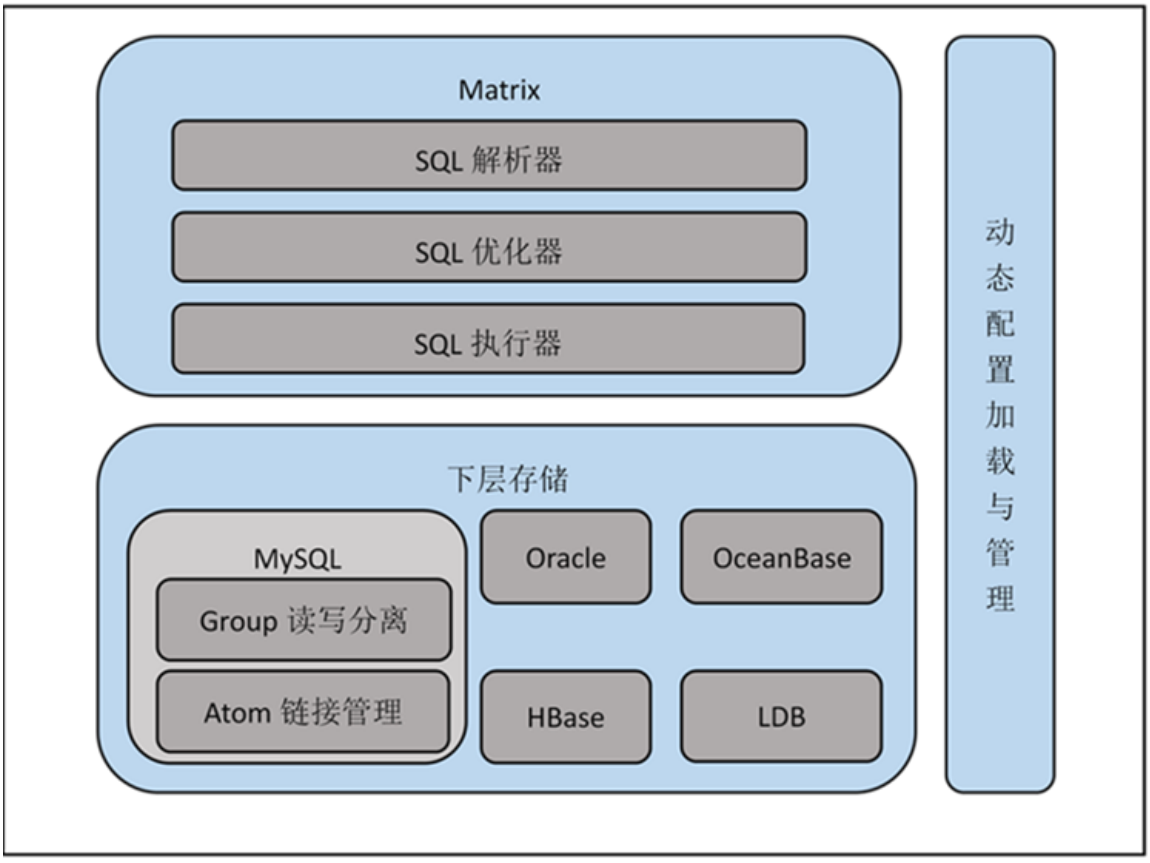
15.3.3 分布式MySQL执行引擎

DRDS分布式SQL引擎目标是实现与单机数据库SQL引擎的完全兼容，实现SQL的智能下推。

智能下推过程包括SQL分析、SQ优化、SQL路由和数据聚合计算。

分布式引擎包含SQL解析、优化、执行和合并四个核心功能，如图 *38:* 分布式引擎核心功能图所示。

图 38: 分布式引擎核心功能图



智能下推核心原则有如下几个：

- 减少网络传输。

- 减少计算量，尽量将计算下推到下层的数据节点上，让计算在数据所在的机器上执行。
- 充分发挥下层存储的全部能力。

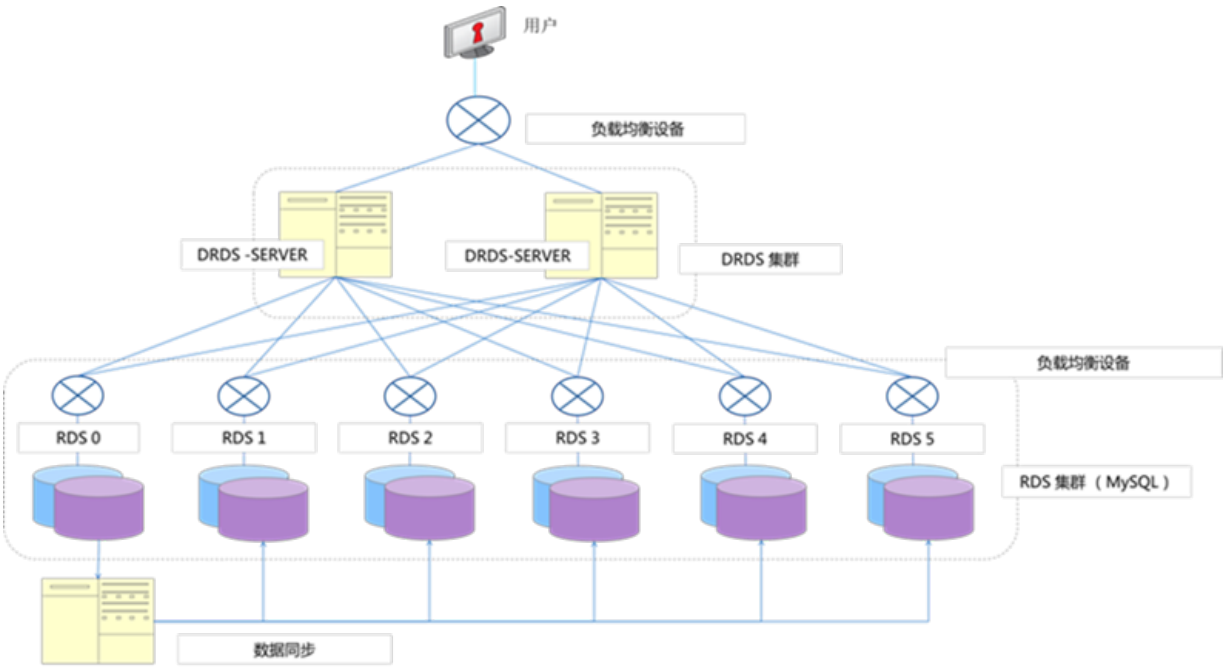
基于以上原则实现的SQL引擎，就可以做到服务能力线性扩展。

例如一个简单的AVG操作，一些开源分布式数据库模型是把AVG直接下发到所有的存储节点。这种简单下推方式会造成语法不兼容，执行结果错误。DRDS对于AVG函数操作，会将逻辑AVG SQL解析优化为SUM和COUNT的SQL然后进行下推，由底层的数据库实例节点完成SUM和COUNT的计算，在引擎层将各个存储节点的SUM和COUNT结果聚合计算，最终计算出AVG。

15.3.4 弹性扩展

DRDS Server层通过集群方式部署，由多个服务节点构成一个服务实例，通过负载均衡与DNS对外提供服务。DRDS的多个服务节点之间无状态同步，均衡处理外部请求。当服务集群的处理能力不足的时候，支持实时增加服务节点，扩展服务能力。DRDS服务层资源都利用率比较低的情况下，支持降低集群规模，降低服务层服务能力，做到服务能力弹性扩展。如图 *39:* 弹性扩展示意图所示。

图 39: 弹性扩展示意图



15.3.5 分布式JOIN支持

DRDS支持分布式JOIN ，提供相应的优化策略。

DRDS的分布式JOIN基于Nested Loop算法，有如下执行步骤：

1. 对于JOIN的左右两个表，首先从JOIN的左表（又叫驱动表）取出数据。
2. 将所取出数据中的JOIN列的值放到右表并进行IN查询，完成JOIN过程。

如果参与JOIN的多表数据切分纬度不同，数据就会按照不同的拆分维度分散在不同的数据库实例上，JOIN操作会跨多个物理分库执行，需要进行多个实例之间大量数据传输，SQL的执行效率就得不到保证。

通过以下优化方案可以提升分布式JOIN效率：

- 参与JOIN操作的数据表尽量要保持拆分维度的统一，让JOIN操作发生在单物理数据库实例上。
- 选择数据量少的表作为JOIN的左表，减少对右表做IN查询就次数就越少。
- 右表的JOIN列为拆分键并且建索引。

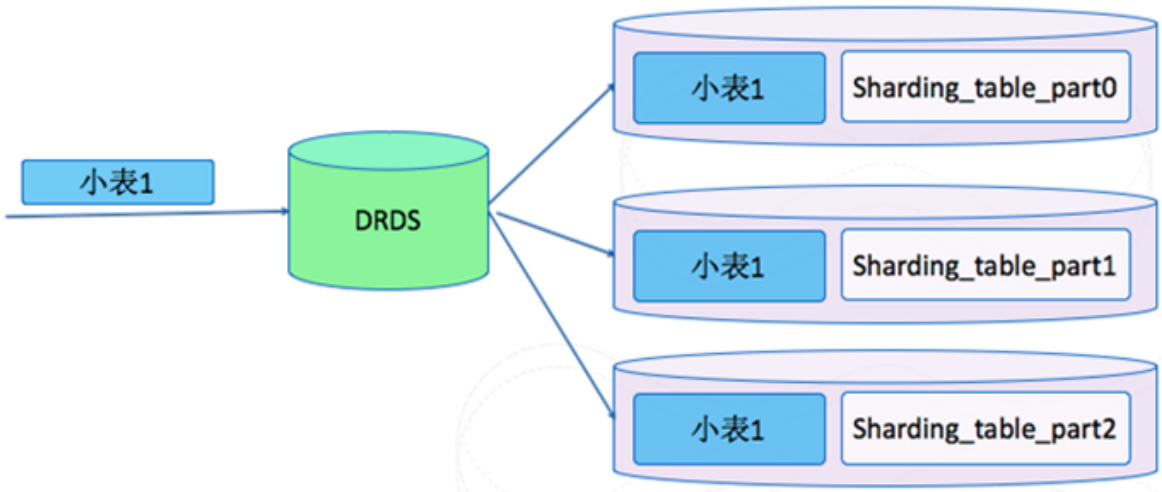
15.3.6 小表广播

分布式数据库场景中，存在一些源信息表，数据量比较小，更新频度也很低，这些表是不需要进行拆分的。源信息表通常采用非拆分的单表模式，单表模式下一个逻辑表的数据统一存储在一个分库中，将这些表定义为小表。

当小表和分库分表进行JOIN的时候，基于Nested Loop算法的原则，小表作为JOIN的驱动表可以减少右表IN查询的次数。

DRDS提供的小表广播的功能，可以将小表的数据实时同步到分库上，将分布式JOIN转化为单机的JOIN操作，减少DRDS Server的计算量，降低数据在多个底层实例之间的传输，提升分布式JOIN的效率。如图 *40:* 小表广播示意图所示。

图 40: 小表广播示意图



15.3.7 读写分离

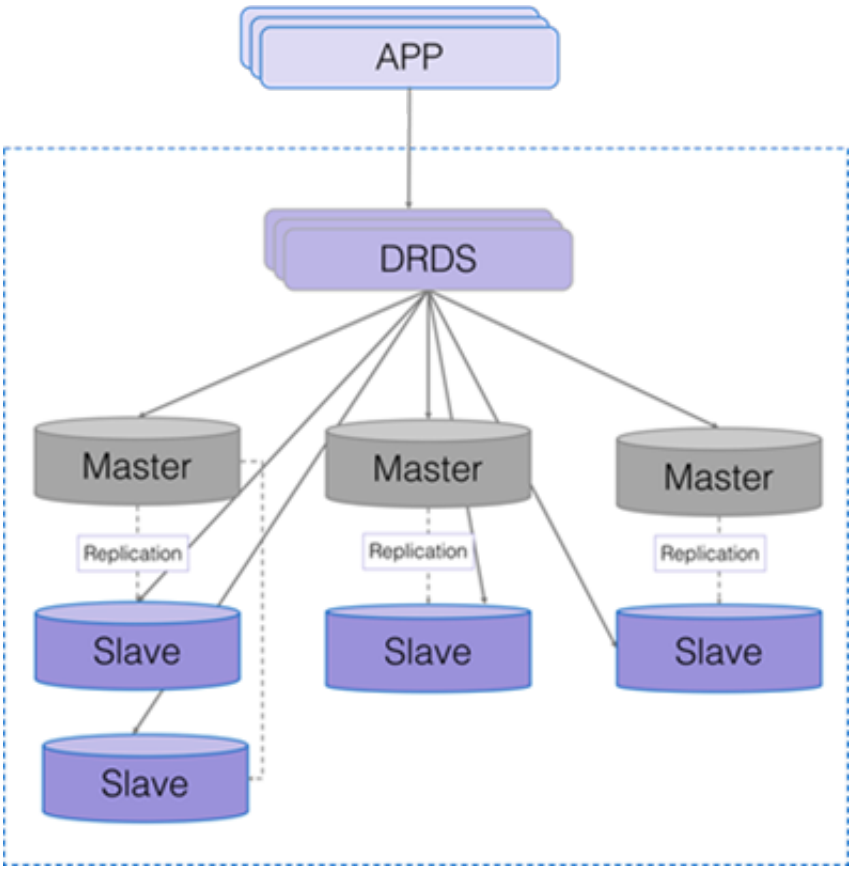
DRDS的读写分离功能是一种对应用透明的读写分离实现。

读写分离流量分配与扩展

DRDS读写分离应用层不需要修改任何代码，只需要在DRDS控制台中增加只读实例和调整读权重，即可将读流量按照需要的比例在主实例与多个只读实例之间调整，写操作则统一走主实例。

添加只读实例可以使读性能线性提升。例如在初始有一个只读实例的情况下，挂载一个只读实例，读性能提升至原来2倍，挂载2个只读实例为单个主库的3倍。如图 *41:* 读写分离流量分配与扩展示意图所示。

图 41: 读写分离流量分配与扩展示意图



读实例上读所操作的数据都是从主实例上异步同步的，存在毫秒级别延迟，对于实时性要求特别强的SQL可以通过DRDS Hint指定主库执行，如下所示：

```
/*TDDL:MASTER/select * from tddl5_users;
```

DRDS支持通过SHOW NODE指令查看实际读流量分布。如图 *42: SHOW NODE*指令查看实际读流量分布所示。

图 42: SHOW NODE指令查看实际读流量分布

```
mysql> show node;
+------+------------------+-------------------+-------------------+--------------------+--------------------+
| ID   | NAME             | MASTER_READ_COUNT | SLAVE_READ_COUNT | MASTER_READ_PERCENT | SLAVE_READ_PERCENT |
+------+------------------+-------------------+-------------------+--------------------+--------------------+
| 0    | USERDATABASE_RDS |                10 |                 2 |                 83% |                17% |
+------+------------------+-------------------+-------------------+--------------------+--------------------+
1 row in set (0.00 sec)
```

非拆分模式的读写分离使用

DRDS的读写分离可以在非拆分模式下独立使用。

DRDS控制台上创建DRDS数据库时，在选定一个数据库实例的情况下，可以选择将底层数据库实例下的一个逻辑数据库直接引入DRDS做读写分离，不需要做数据迁移。

15.3.8 分布式事务

强一致事务

DRDS支持分布式强一致事务。

分布式数据库架构下会出现跨库的分布式事务。分布式事务会在多个分库上进行事务分支的执行和状态同步，相比单机事务，分布式跨库事务的吞吐量和延迟会大大增加。事务执行涉及的分库越多，边界越大，事务执行时间也会相应增加，性能就会出现线性衰减。

分布式数据库的事务的原则是尽量通过优化让事务在单库中执行。在单库中执行事务可以保持事务ACID特性的同时，同时具备事务扩展能力。

最终一致事务

最终一致事务是解决分布式事务的效率问题有效办法。

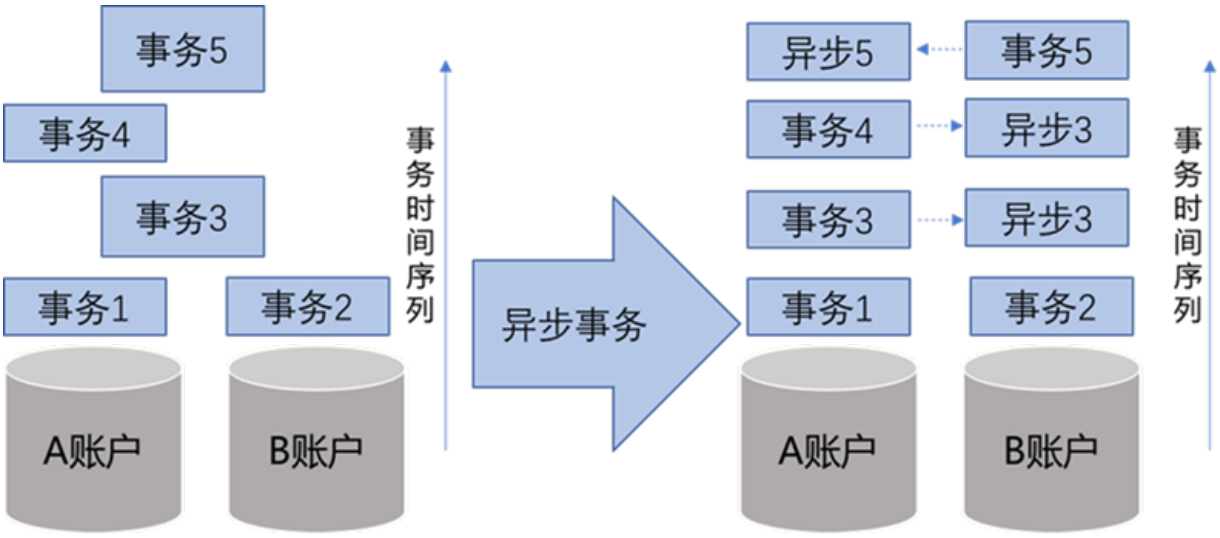
最终一致事务的原理是优先保证核心事务分支的正向执行，然后对事务的中间状态进行保存，其他的事务分支异步执行。分支事务执行完成后达到最终的事务数据一致，避免跨库事务时间序列执行的阻塞，提升事务吞吐量。

如图 *43:* 一致事务示意图所示，事务3是一个转账事务，负责从A账户转账到B帐户。

如果是强一致事务（左图），那么在事务3执行时，必须锁定A账户和B帐户事务相关数据，其他事务限制操作，在左右两个分库必须全部完成后，才可以执行事务4。

如果是最终一致事务（右图），则可将事务3的转账事务，拆分为A账户扣款和B帐户增加款项两个事务分支，两个事务分支在所在分库上顺序执行，事务4等其他事务无需等待事务3的所有事务分支全部完成就可以执行，当两个异步事务分支都完成时，整个转账事务最终完成。事务5也是同样的逻辑，整体上可以满足最终事务一致性。

图 43: 一致事务示意图



15.3.9 SQL兼容性

DRDS高度兼容MySQL协议和语法，但由于分布式数据库和单机数据库存在较大的架构差异，存在SQL使用限制，相关兼容性和SQL限制描述如下。

DRDS SQL限制

SQL大类限制

- 暂不支持用户自定义数据类型、自定义函数。
- 暂不支持视图、存储过程、触发器、游标。
- 暂不支持BEGIN...END、LOOP...END LOOP、REPEAT...UNTIL...END REPEAT、WHILE...DO...END WHILE等复合语句。
- 暂不支类似IF，WHILE等流程控制类语句。

小语法限制

DDL

- CREATE TABLE tbl_name LIKE old_tbl_name不支持拆分表。
- CREATE TABLE tbl_name SELECT statement不支持拆分表。

DML

- 暂不支持SELECT INTO OUTFILE/INTO DUMPFILE/INTO var_name。
- 暂不支持INSERT DELAYED Syntax。
- 暂不支持非WHERE 条件的 Correlate Subquery。
- 暂不支持SQL中带聚合条件的Correlate Subquery。

- 暂不支持SQL中对于变量的引用和操作，比如SET @c=1，@d=@c+1； SELECT @c，@d。

数据库管理

- SHOW WARNINGS Syntax不支持LIMIT/COUNT的组合。
- SHOW ERRORS Syntax不支持LIMIT/COUNT的组合。

DRDS SQL兼容

协议兼容

DRDS支持MySQL Workbench ， Navicat For MySQL ， SQLyog等主流客户端。

DDL语法兼容

- CREATE TABLE Syntax
- CREATE INDEX Syntax
- DROP TABLE Syntax
- DROP INDEX Syntax
- ALTER TABLE Syntax
- TRUNCATE TABLE Syntax

DML语法兼容

- INSERT Syntax
- REPLACE Syntax
- UPDATE Syntax
- DELETE Syntax
- Subquery Syntax
- Scalar Subquery
- Comparisons Subquery
- Subqueries with ANY， IN，or SOME
- Subqueries with ALL
- Row Subqueries
- Subqueries with EXISTS or NOT EXISTS
- Subqueries in the FROM Clause
- SELECT Syntax

Prepare语法兼容

- PREPARE Syntax
- EXECUTE Syntax
- DEALLOCATE PREPARE Syntax

数据库管理语法兼容

- SET Syntax SHOW Syntax
- KILL 'PROCESS_ID' (DRDS不支持KILL QUERY指令，只支持KILL 'PROCESS_ID')
- SHOW COLUMNS Syntax
- SHOW CREATE TABLE Syntax
- SHOW INDEX
- SHOW TABLES Syntax
- SHOW TABLE STATUS Syntax
- SHOW TABLE STATUS Syntax
- SHOW TABLES Syntax
- SHOW VARIABLES Syntax
- SHOW WARNINGS Syntax
- SHOW ERRORS Syntax

注意: 其他SHOW指令会默认下发到DB处理，结果数据没有进行分库数据合并。

数据库工具指令

- DESCRIBE Syntax
- EXPLAIN Syntax
- USE Syntax

DRDS自定义指令

- SHOW SEQUENCES / CREATE SEQUENCE / ALTER SEQUENCE / DROP
- SEQUENCE 【 DRDS全局唯一数字序列管理】
- SHOW PARTITIONS FROM TABLE 【查询表的拆分字段】
- SHOW TOPOLOGY FROM TABLE 【查询表的物理拓扑】
- SHOW BROADCASTS 【查询所有广播表】

- SHOW RULE [FROM TABLE] 【查询表拆分定义】
- SHOW DATASOURCES 【查询后端DB连接池定义】
- SHOW DBLOCK / RELEASE DBLOCK 【分布式LOCK定义】
- SHOW NODE 【查询读写库流量】
- SHOW SLOW 【查询慢SQL列表】
- SHOW PHYSICAL_SLOW 【查询物理DB执行慢SQL列表】
- TRACE SQL_STATEMENT / SHOW TRACE 【跟踪SQL整个执行过程】
- EXPLAIN [DETAIL/EXECUTE] SQL_STATEMENT 【分析DRDS执行计划和物理DB上的执行计划】
- RELOAD USERS 【同步DRDS控制台用户信息到DRDS SERVER】
- RELOAD SCHEMA 【清理DRDS对应DB库数据缓存，比如SQL解析/语法树/表结构缓存】
- RELOAD DATASOURCES 【重建后端与所有DB的连接池】

数据库函数

- 带拆分键的SQL ，所有MySQL函数支持
- 不带拆分键的SQL，部分函数支持。
- 操作符函数

Function	Description
AND，&&	Logical AND.
=	Assigna value (as part of a SET statement, or as part of the SET clause in an UPDATE statement).
BETWEEN... AND...	Check whether a value is within a range of value.
BINARY	Cast a string to a binary string.
&	Bitwise AND.
~	Bitwise inversion.
^	Bitwise XOR.
DIV	Integer division.
/	Division operator.
<=>	NULL-safe equal to operator.
=	Equal operator.
>=	Greater than or equal operator.

Function	Description
>	Greater than operator.
IS NOTNULL	NOT NULL value test.
ISNOT	Test a value against a boolean.
ISNULL	NULL value test.
IS	Test a value against a boolean.
<<	Left shift.
<=	Less than or equal operator.
<	Less than operator.
LIKE	Simple pattern matching.
-	Minus operator.
%,	MOD Modulo operator.
NOTBETWEEN... AND...	Check whether a value is not within a range of values.
!= , <>	Not equal operator.
NOTLIKE	Negation of simple pattern matching.
NOTREGEXP	Negation of REGEXP.
NOT , !	Negates value.
OR	Logical OR.
+	Addition operator.
REGEXP	Pattern matching using regular expressions.
>>	Right shift.
RLIKE	Synonym for REGEXP.
*	Multiplication operator.
-	Change the sign of the argument.
XOR	Logical XOR.
Coalesce	Return the first non-NULL argument.
GREATEST	Return the largest argument.
LEAST	Return the smallest argument.
STRCMP	Compare two strings.

- 流程控制函数

Function	Description
CASE	Case operator.
IF()	If/else construct.
IFNULL()	Null if/else construct.
NULLIF()	Return NULL if expr1 =expr2.

- 数学函数

Function	Description
ABS()	Return the absolute value.
ACOS()	Return the arc cosine.
ASIN()	Return the arc sine.
ATAN2()，ATAN()	Return the arc tangent of the two arguments.
ATAN()	Return the arc tangent.
CEIL()	Return the smallest integer value not less than the argument.
CEILIG()	Return the smallest integer value not less than the argument.
CONV()	Convert numbers between different number bases.
COS()	Return the cosine.
COT()	Return the cotangent.
CRC32()	Compute a cyclic redundancy check value.
DEGREES()	Convert radians to degrees.
DIV	Integer division.
EXP()	Raise to the power of.
FLOOR()	Return the largest integer value not greater than the argument.
LN()	Return the natural logarithm of the argument.
LOG10()	Return the base-10 logarithm of the argument.
LOG2()	Return the base-2 logarithm of the argument.
LOG()	Return the natural logarithm of the first argument.
MOD()	Return the remainder.
%,MOD	Modulo operator.

Function	Description
PI()	Return the value of pi.
POW()	Return the argument raised to the specified power.
POWER()	Return the argument raised to the specified power.
RADIANS()	Return argument converted to radians.
RAND()	Return a random floating-point value.
ROUND()	Round the argument.
SIGN()	Return the sign of the argument.
SIN()	Return the sine of the argument.
SQRT()	Return the square root of the argument.
TAN()	Return the tangent of the argument.
TRUNCATE(	Truncate to specified number of decimal places.

- 字符串函数

Function	Description
ASCII()	Return numeric value of left-most character.
BIN()	Return a string containing binary representation of a number.
BIT_LENGTH()	Return length of argument inbits.
CHAR_LENGTH()	Return number of characters in argument.
CHAR()	Return the character for each integer passed.
CHARACTER_LENGTH()	Synonym for CHAR_LENGTH().
CONCAT_WS()	Return concatenate with separator.
CONCAT()	Return concatenated string.
ELT()	Return string at index number.
EXPORT_SET()	Return a string such that for every bit set in the value bits, you get an onstring and for every unset bit, you get an offstring.
FIELD()	Return the index (position) of the first argument in the subsequent arguments.
FIND_IN_SET()	Return the index position of the first argument within the second argument.
FORMAT()	Return a number formatted to specified number of decimal places.

Function	Description
HEX()	Return a hexadecimal representation of a decimalor string value.
INSERT()	Insert a substring at the specified position up to the specified number of characters.
INSTR()	Return the index of the first occurrence of substring.
LCASE()	Synonymfor LOWER().
LEFT()	Return the leftmost number of characters as specified.
LENGTH()	Return the length of a string inbytes.
LIKE	Simple pattern matching.
LOCATE()	Return the position of the first occurrence of substring.
LOWER()	Return the argument in lowercase.
LPAD()	Returnthe string argument, left-padded with the specified string.
LTRIM()	Remove leading spaces.
MAKE_SET()	Return a set of comma-separated strings that have the corresponding bit in bits set.
MID()	Return a substring starting from the specified position.
NOTLIKE	Negation of simple pattern matching.
NOTREGEXP	Negation of REGEXP.
OCT()	Return a string containing octal representation of a number.
OCTET_LENGTH()	Synonym for LENGTH().
ORD()	Return character code for leftmost character of the argument.
POSITION()	Synonym for LOCATE().
QUOTE()	Escape the argument for use in an SQL statement.
REPEAT()	Repeat a string the specified number of times.
REPLACE()	Replace occurrences of a specified string.
REVERSE()	Reverse the characters in a string.
RIGHT()	Return the specified rightmost number of characters.
RPAD()	Append string the specified number of times.
RTRIM()	Remove trailing spaces.
SPACE()	Return a string of the specified number of spaces.

Function	Description
STRCMP()	Compare two strings.
SUBSTR()	Return the substring as specified.
SUBSTRING_INDEX()	Return a substring from a string before the specified number of occurrences of the delimiter.
SUBSTRING()	Return the substring as specified.
TRIM()	Remove leading and trailing spaces.
UCASE()	Synonym for UPPER().
UNHEX()	Return a string containing hex representation of a number.
UPPER()	Convert to uppercase.

- 时间函数

Function	Description
ADDDATE()	Add time values (intervals) to a date value.
ADDTIME()	Add time.
CURDATE()	Return the current date.
CURRENT_DATE()	CURRENT_DATE Synonyms for CURDATE().
CURRENT_TIME()	CURRENT_TIME Synonyms for CURTIME().
CURRENT_TIMESTAMP()	CURRENT_TIMESTAMP Synonyms for NOW().
CURTIME()	Return the current time.
DATE_ADD()	Add time values (intervals) to a date value.
DATE_FORMAT()	Format date as specified.
DATE_SUB()	Subtract a time value (interval) from a date.
DATE()	Extract the date part of a date or datetime expression.
DATEDIFF()	Subtract two dates.
DAY()	Synonym for DAYOFMONTH().
DAYNAME()	Return the name of the weekday.
DAYOFMONTH()	Return the day of the month(0-31).
DAYOFWEEK()	Return the weekday index of the argument.
DAYOFYEAR()	Return the day of the year(1-366).
EXTRACT()	Extract part of a date.

Function	Description
FROM_DAYS()	Convert a day number to a date.
FROM_UNIXTIME()	Format UNIX timestamp as a date.
GET_FORMAT()	Return a date format string.
HOUR()	Extract the hour.
LAST_DAY()	Return the last day of the month for the argument.
LOCALTIME()	LOCALTIME Synonym for NOW().
LOCALTIMESTAMP, LOCALTIMESTAMP()	Synonym for NOW().
MAKEDATE()	Create a date from the year and day of year.
MAKETIME()	Create time from hour, minute, second.
MICROSECOND()	Return the microseconds from argument.
MINUTE()	Return the minute from the argument.
MONTH()	Return the month from the date passed.
MONTHNAME()	Return the name of the month.
NOW()	Return the current date and time.
PERIOD_ADD()	Add a period to a year-month.
PERIOD_DIFF()	Return the number of months between periods.
QUARTER()	Return the quarter from a date argument.
SEC_TO_TIME()	Converts seconds to 'HH:MM:SS' format.
SECOND()	Return the second(0-59).
STR_TO_DATE()	Convert a string to a date.
SUBDATE()	Synonym for DATE_SUB() when invoked with three arguments.
SUBTIME()	Subtract times.
SYSDATE()	Return the time at which the function executes.
TIME_FORMAT()	Format as time.
TIME_TO_SEC()	Return the argument converted to seconds.
TIME()	Extract the time portion of the expression passed.
TIMEDIFF()	Subtract time.
TIMESTAMP()	With a single argument, this function returns the date or datetime expression; with two arguments, the sum of the arguments.

Function	Description
TIMESTAMPADD()	Add an interval to a datetime expression.
TIMESTAMPDIFF()	Subtract an interval from a datetime expression.
UNIX_TIMESTAMP()	Return a UNIX timestamp.
UTC_DATE()	Return the current UTC date.
UTC_TIME()	Return the current UTC time.
UTC_TIMESTAMP()	Return the current UTC date and time.
WEEKDAY()	Return the weekday index.
WEEKOFYEAR()	Return the calendar week of the date(1-53).
YEAR()	Return the year.

- 类型转换函数

Function	Description
BINARY	Cast a string to a binary string.
CAST()	Cast a value as a certain type.
CONVERT()	Cast a value as a certain type.

16 消息队列MQ

16.1 产品概述

消息队列（Message Queue，简称 MQ）是阿里巴巴集团中间件技术部自主研发的专业消息中间件。产品基于高可用分布式集群技术，提供消息发布订阅、消息轨迹查询、定时（延时）消息、资源统计、监控报警等一系列消息云服务，是企业级互联网架构的核心产品。MQ 历史超过 9 年，为分布式应用系统提供异步解耦、削峰填谷的能力，同时具备海量消息堆积、高吞吐、可靠重试等互联网应用所需的特性，是阿里巴巴双11使用的核心产品，每年天猫双十一全天提供 99.99% 可用性。

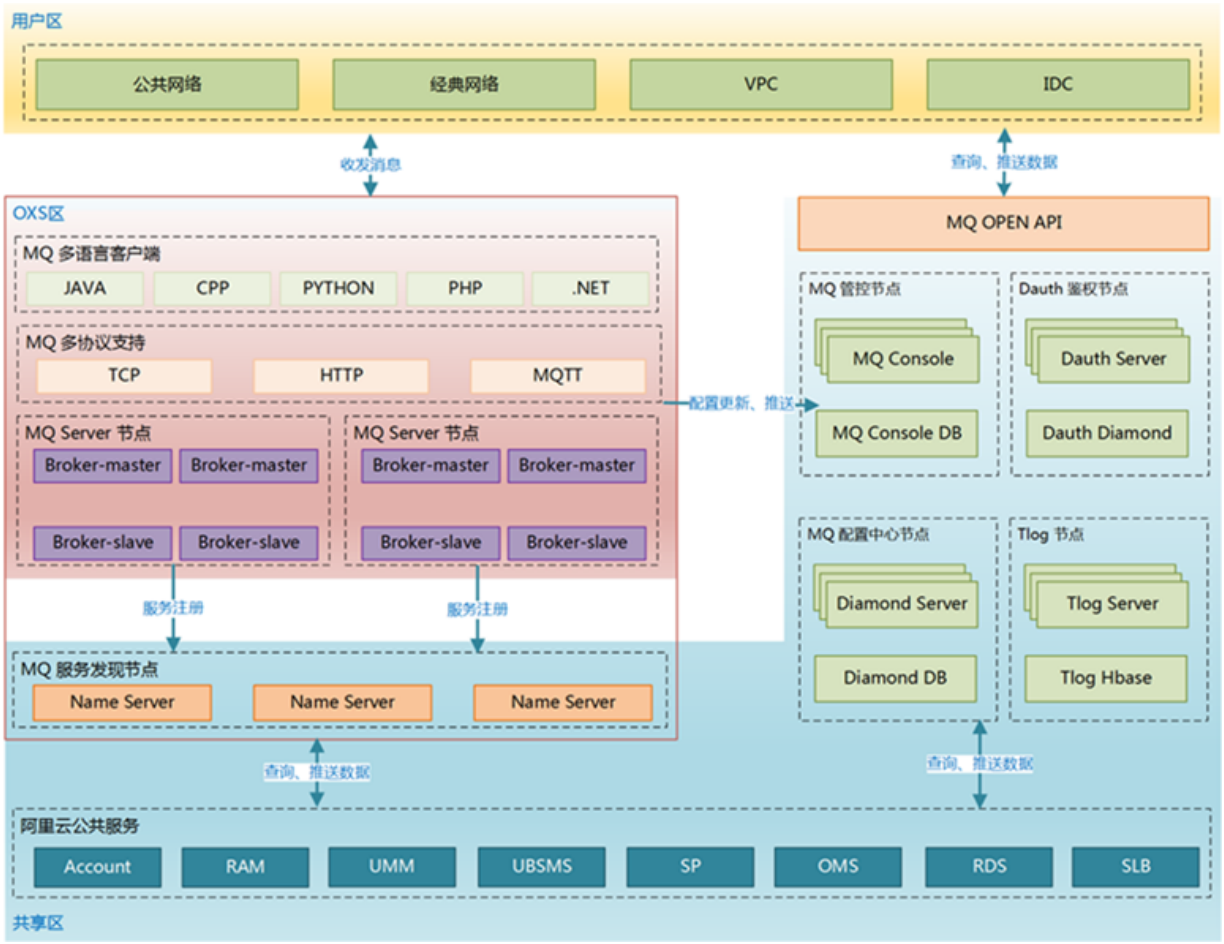
MQ 目前提供 TCP、HTTP、MQTT 等协议层面的接入方式，支持 Java、C++ 以及 .NET 不同语言，方便不同编程语言开发的应用快速接入 MQ 消息云服务。您可以将应用部署在阿里云 ECS、企业自建云，或者嵌入到移动端、物联网设备中与 MQ 建立连接进行消息收发，同时本地开发者也可以通过公网接入 MQ 服务进行消息收发。

图 44: MQ产品示意图

16.2 产品架构

消息队列MQ由MQ Broker 、MQ服务发现系统、MQ管控平台、MQ鉴权系统（DAuth）、MQ OpenAPI等系统模块组成，整体系统架构如图 *45: MQ系统架构*所示。

图 45: MQ系统架构



MQ Broker

消息队列服务器（MQ Broker）是消息队列的核心处理模块，由多个服务节点组成服务集群，负责消息的收、发以及消息的存储。

MQ Broker支持集群化、多副本部署，主备复制实现高可用。提供高可用、稳定高效、可线性扩容的消息服务能力。

MQ Broker将消息队列将主题信息、订阅信息等注册到Name Server上，提供服务。

MQ服务发现节点

服务发现节点（Name Server）主要负责消息队列服务的注册与查找，是实现消息队列服务弹性部署和线性扩展的核心。

Name Server几乎无状态节0点，节点之间不需要进行数据同步，可集群部署横向扩展。

MQ管控平台

MQ管控平台为用户提供消息的Topic管理、发布管理、订阅管理、消息查询、消息轨迹查询、资源报表、以及监控告警等一整套完备的运维功能。

用户可以通过MQ管控平台，快速接入消息队列、对用户资源进行管理、问题排查以及通过监控告警等服务帮助用户及时的发现问题。

MQ鉴权系统

MQ鉴权系统（DAuth）为消息队列提供统一登录服务以及安全访问控制。包括资源的权限控制、跨账号与子账号访问控制、资源授权等功能。

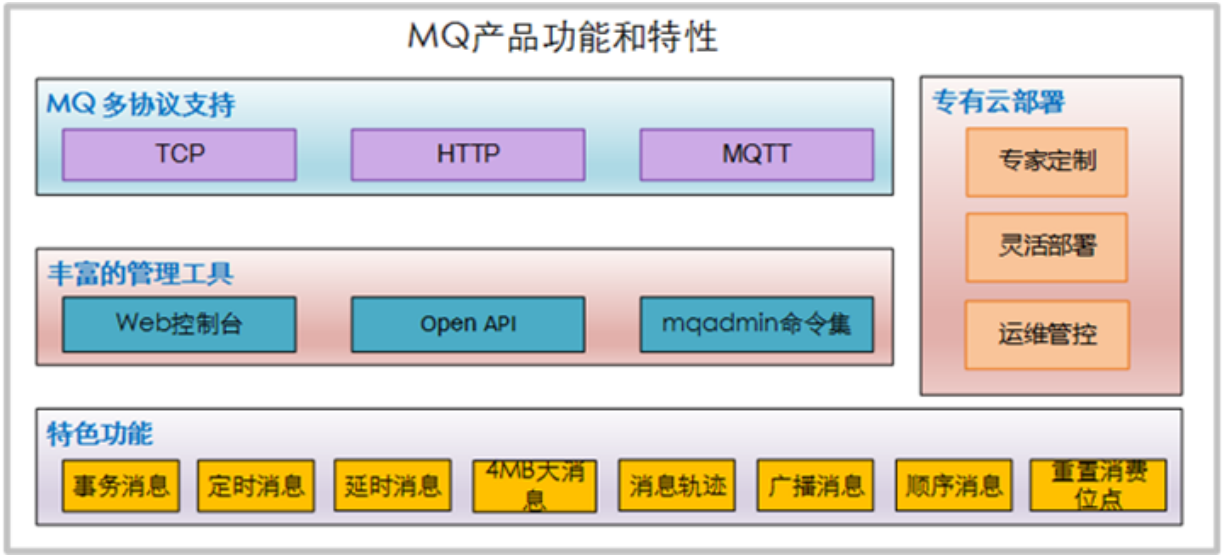
MQ OpenAPI

为方便用户接入、自主运维，MQ OpenAPI通过HTTP/HTTPS接口的形式提供一整套API服务，用户可以通过OpenAPI创建Topic、发布者信息、订阅者信息、消息查询以及消费者状态查询等。

16.3 功能特性

MQ提供了多种协议和开发语言的接入方式以及多维度的管理工具，同时针对不同的应用场景提供了一系列的特色功能。MQ功能特性示意图如图 *46: MQ功能特性示意图*所示。

图 46: MQ功能特性示意图



16.3.1 多协议支持

MQ提供了多种协议和开发语言的接入方式，包括：

- 支持HTTP协议：支持RESTful风格HTTP协议完成收发消息，可以解决跨语言使用MQ问题。
- 支持TCP协议：区别于HTTP简单的接入方式，提供更为专业、可靠、稳定的TCP协议的SDK接入。
- 支持MQTT协议：支持主动推送模型，多级Topic模型支持一次触达1000万+ 终端，可广泛应用于物联网和社交即时通信场景。

16.3.1.1 支持HTTP协议

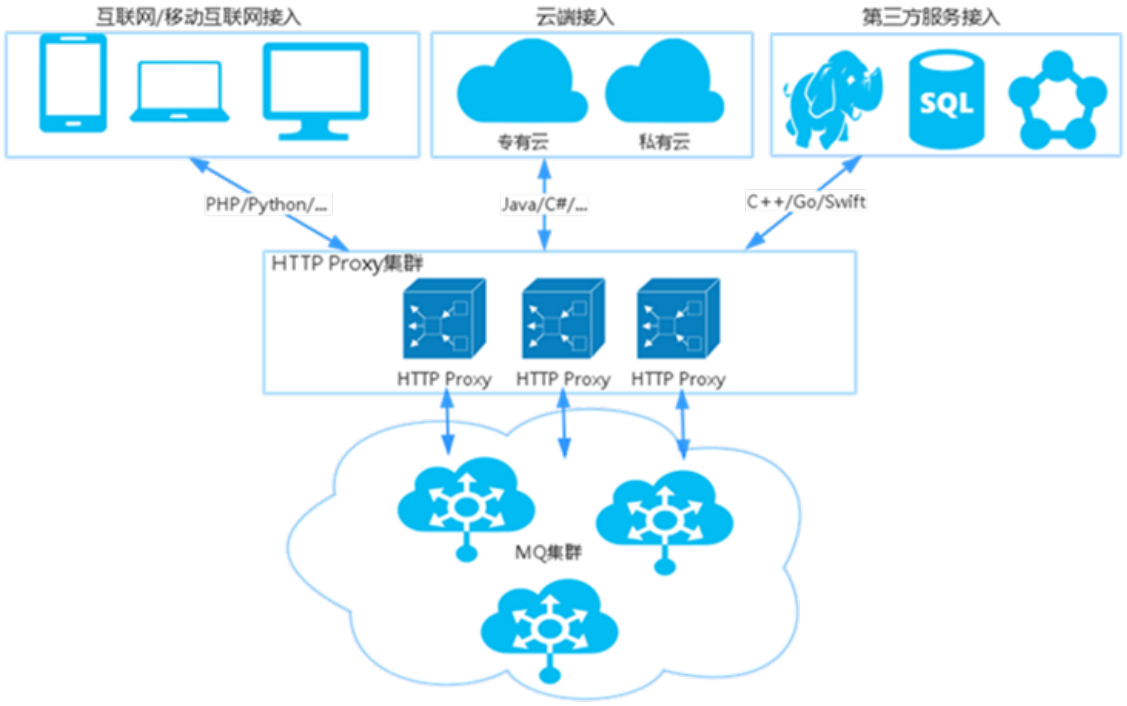
随着云计算相关技术的快速发展，很多应用都开始了云端部署。基于HTTP提供服务的MQ中所有消息都是以HTTP协议为载体，通过使用HTTP的常用接口来对消息队列进行增删查。

HTTP接入主要有以下几大优势：

- 消息队列HTTP接入模式最大优势跨语言跨网络访问消息队列；
- 解决异构网络环境下的服务相互访问屏障，对于没有提供相关操作消息队列SDK的环境中，使用HTTP方式接入更为方便；
- 消息队列HTTP接入方式在使用上简单，上手快。

HTTP接入应用场景：

HTTP接入方式应用的场景主要依托于客户的业务场景，假设客户的业务场景或者部分模块是基于HTTP协议并且需要通信服务，就可以使用MQ服务。



16.3.1.2 支持TCP协议

区别于HTTP简单的接入方式，提供更为专业、可靠、稳定的TCP协议的SDK接入。

TCP接入主要有以下几大优势：

- 长连接方式提供更高的服务性能；
- 长轮询的实现方式，使得消息收发具有更高的实时性；
- 官方提供JAVA、C++、.NET、PHP四种高可靠SDK接入；
- 支持3种消息发送方式，消息场景全覆盖：可靠同步、可靠异步、oneway方式；
- 支持事务消息、定时消息、顺序消息等特色功能；
- 支持集群方式与广播方式订阅；
- 更为完整的运维配套支持，包括消息堆积、消息轨迹、消费者运行状态信息等。

消息发送

TCP协议支持三种消息发送方式：可靠同步发送、可靠异步发送、单向（Oneway）发送。本文介绍了每种实现的原理、使用场景以及三种实现的异同，同时提供了代码示例以供参考。

- **可靠同步发送**

原理：同步发送是指消息发送方发出数据后，会在收到接收方发回响应之后才发下一个数据包的通讯方式。

应用场景：此种方式应用场景非常广泛，例如重要通知邮件、报名短信通知、营销短信系统等。

- **可靠异步发送**

原理：异步发送是指发送方发出数据后，不等接收方发回响应，接着发送下个数据包的通讯方式。MQ的异步发送，需要用户实现异步发送回调接口（SendCallback），在执行消息的异步发送时，应用不需要等待服务器响应即可直接返回，通过回调接口接收务器响应，并对服务器的响应结果进行处理。

应用场景：异步发送一般用于链路耗时较长，对RT响应时间较为敏感的业务场景，例如用户视频上传后通知启动转码服务，转码完成后通知推送转码结果等。

- **单向（Oneway）发送**

  原理：单向（Oneway）发送特点为只负责发送消息，不等待服务器回应且没有回调函数触发，即只发送请求不等待应答。此方式发送消息的过程耗时非常短，一般在微秒级别。

  应用场景：适用于某些耗时非常短，但对可靠性要求并不高的场景，例如日志收集。

  下表概括了三者的特点和主要区别。

发送模式	发送TPS	发送结果反馈	可靠性
同步发送	快	有	不丢失
异步发送	快	有	不丢失
单向发送	最快	无	可能丢失

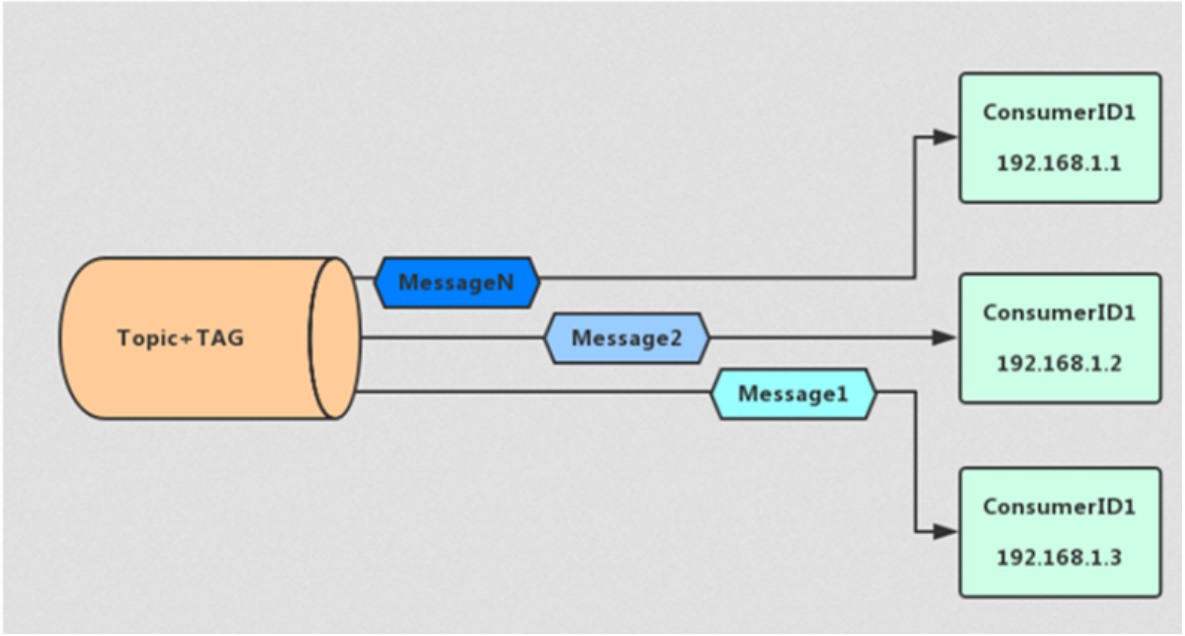
消息订阅

MQ是基于发布订阅模型的消息系统。在MQ消息系统中消息的订阅方订阅关注的Topic ，以获取并消费消息。由于订阅方应用一般是分布式系统，以集群方式部署有多台机器。因此MQ约定以下概念。

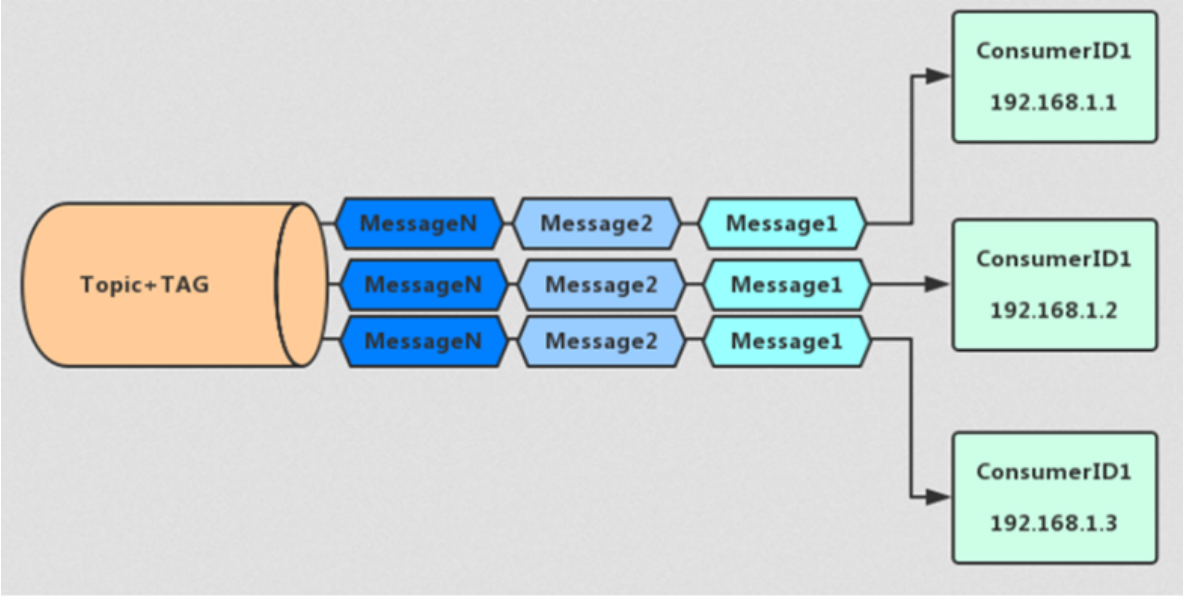
- 集群：MQ约定使用相同Consumer ID的订阅者属于同一个集群，同一个集群下的订阅者消费逻辑必须完全一致（包括Tag的使用），这些订阅者在逻辑上可以认为是一个消费节点。
- 集群消费：当使用集群消费模式时，MQ认为任意一条消息只需要被集群内的任意一个消费者处理即可。
- 广播消费：当使用广播消费模式时，MQ会将每条消息推送给集群内所有注册过的客户端，保证消息至少被每台机器消费一次。

两种消费模式对比如下：

- 集群消费模式



- 广播消费模式



16.3.2 特色功能

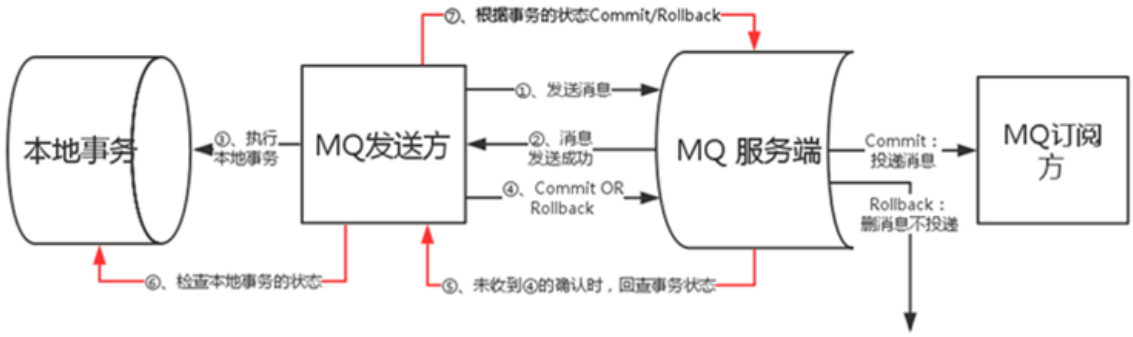
- 事务消息，实现类似X/Open XA的分布事务功能，以达到事务最终一致性状态；
- 定时（延时）消息，允许消息生产者指定消息进行定时（延时）投递，最长支持40天；
- 顺序消息，支持消息的全局顺序与局部顺序；
- 消息过滤，支持消费者根据Tag在MQ服务端完成消息过滤。

16.3.2.1 事务消息

MQ提供类似X/Open XA的分布事务功能，通过MQ事务消息能达到分布式事务的最终一致。

MQ事务消息交互流程如图 *47: MQ事务消息交互流程图*所示：

图 47: MQ事务消息交互流程图



其中：

- 发送方向MQ服务端发送消息。
- MQ Server将消息持久化成功之后，向发送方ACK确认消息已经发送成功，此时消息为半消息。
- 发送方开始执行本地事务逻辑。
- 发送方根据本地事务执行结果向MQ Server提交二次确认（Commit或是Rollback），MQ Server收到Commit状态则将半消息标记为可投递，订阅方最终将收到该消息；MQ Server收到Rollback状态则删除半消息，订阅方将不会接受该消息。
- 在断网或者是应用重启的特殊情况下，上述步骤4提交的二次确认最终未到达MQ Server，经过固定时间后MQ Server将对该消息发起消息回查。
- 发送方收到消息回查后，需要检查对应消息的本地事务执行的最终结果。
- 发送方根据检查得到的本地事务的最终状态再次提交二次确认，MQ Server仍按照步骤4对半消息进行操作。

16.3.2.2 定时（延时）消息

- **定时消息**：Producer将消息发送到MQ服务端，但并不期望这条消息立马投递，而是推迟到在当前时间点之后的某一个时间投递到Consumer进行消费，该消息即定时消息。
- **延时消息**：Producer将消息发送到MQ服务端，但并不期望这条消息立马投递，而是延迟一定时间后才投递到Consumer进行消费，该消息即延时消息。

定时（延时）消息适用于如下一些场景：

- 消息生产和消费有时间窗口要求：比如在电商交易中超时未支付关闭订单的场景，在订单创建时会发送一条MQ延时消息，这条消息将会在30分钟以后投递给消费者，消费者收到此消息后需要判断对应的订单是否已完成支付；如支付未完成，则关闭订单，如已完成支付则忽略。
- 通过消息触发一些定时任务，比如在某一固定时间点向用户发送提醒消息。

在使用方式上，定时消息、延时消息的使用在代码编写上存在略微的区别：

- 发送定时消息需要明确指定消息发送时间点之后的某一时间点作为消息投递的时间点。
- 发送延时消息时需要设定一个延时时间长度，消息将从当前发送时间点开始延迟固定时间之后才开始投递。

16.3.2.3 顺序消息

消息队列MQ的支持顺序消息，消息的发送与订阅保持有序，其中包括：全局有序和分块有序。

- **全局有序**：所有的消息以消息达到消息队列服务器时的顺序进行消息的消费。
- **分块有序**：根据业务指定的sharding_key进行分块，同一块内消息以消息达到消息队列服务器时的顺序进行消息的消费，不同块之间无顺序关系。

两种有序消息的特点：

- **全局有序**：无论是消息的发送，或是消息的订阅，都必须是单实例单线程运行，无法横向扩展，性能相对较弱。
- **分块有序**：根据sharding_key进行分块，块内部单并发运行，块与块之间并发运行，可以横向扩展，性能相对较高。

16.3.2.4 消息过滤

消息队列MQ支持消费者根据Tag在MQ服务端完成消息过滤，从而满足用户对于订阅不同消息类型的需求。

所谓Tag，即消息标签、消息类型，用来区分某个MQ的Topic下的消息分类。MQ允许消费者按照Tag对消息进行过滤，确保消费者最终只消费到他关心的消息类型。

16.3.3 MQ应用场景

MQ可应用在多个领域，包括异步通信解耦、企业解决方案、金融支付、电信、电子商务、快递物流、广告营销、社交、即时通信、手游、视频、物联网、车联网等。

MQ可以应用但不局限于以下业务场景：

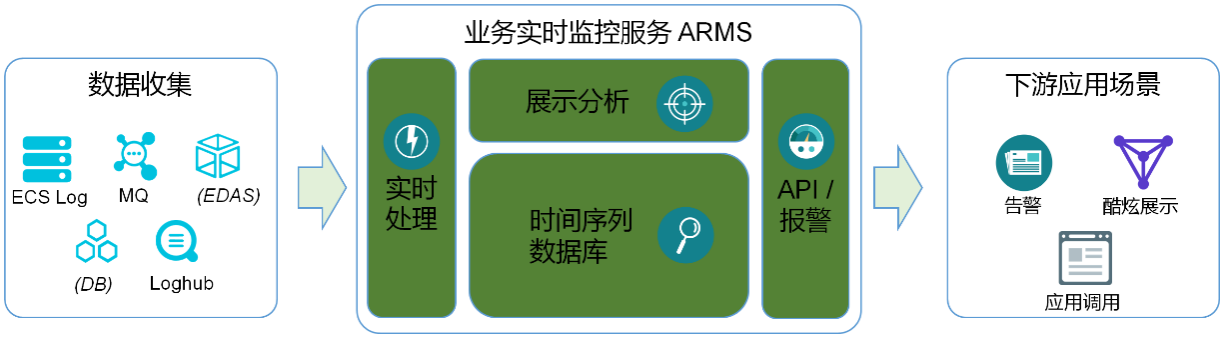
- 一对多，多对多异步解耦，基于发布订阅模型，对分布式应用进行异步解耦，增加应用的水平扩展能力。
- 削峰填谷，大促等流量洪流突然来袭时，MQ可以缓冲突发流量，避免下游订阅系统因突发流量崩溃。
- 日志监控，作为重要日志的监控通信管道，将应用日志监控对系统性能影响降到最低。
- 消息推送，为社交应用和物联网应用提供点对点推送，一对多广播式推送的能力。
- 金融报文，发送金融报文，实现金融准实时的报文传输，可靠安全。
- 电信信令，将电信信令封装成消息，传递到各个控制终端，实现准实时控制和信息传递。

17 企业实时监控服务ARMS

17.1 产品概述

业务实时监控服务 (Application Real-Time Monitoring Service，简称 ARMS) 是一款端到端一体化实时监控解决方案的 PaaS 级阿里云产品。通过该产品，您可以基于海量的数据迅速便捷地通过定制化为企业带来秒级的业务监控和响应能力。ARMS 产品孵化于阿里巴巴内部业务，经过长时间考验，目前已被广泛用于阿里内外的商品、物流、风控和各种云产品的各类业务监控场景。

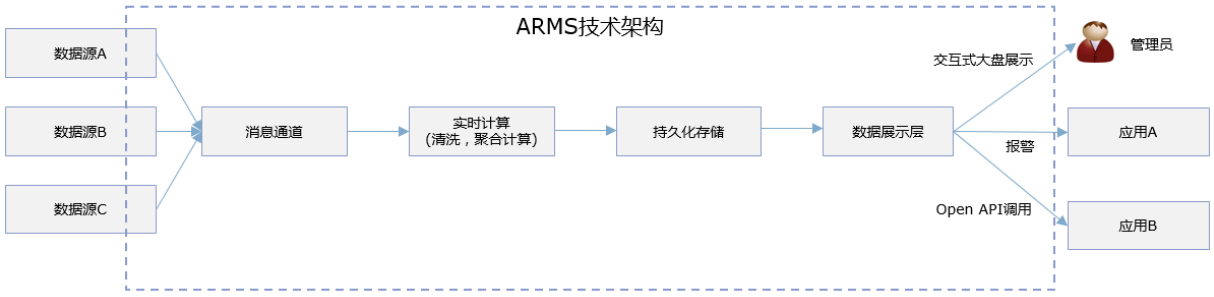
图 48: ARMS产品示意图



17.2 产品架构

ARMS的技术架构由下面几个模块组成：

图 49: ARMS技术架构



在完整的一个监控任务中，数据流依次经过以下技术栈：

- 从数据源流入数据通道，作统一管理和缓存作用。
- 从数据通道流入实时计算引擎进行实时计算。
- 计算结果流入持久化存储平台作统一存储。

- 通过数据展示层对数据进行各类导出，包括Open API直接读取、报表展示、报警通知等。

17.2.1 数据源

在ARMS中，数据源负责为ARMS提供数据输入。数据源包括以下几种方式：

- StarAgent数据源： 通过StarAgent完成在服务器上的日志的增量拉取，例如日志文件。适用所有可以网络直连互通的网络环境。
- MQ ：通过配置 MQ 相应 Topic 的接收端，将指定的数据以消息方式传输到 ARMS计 算节点。

ARMS 在收集数据时，需要在已定义的数据源中通过建立数据采集规则来实现数据采集。以采集服务器上的日志为例，采集规则包括从哪些数据源中收集，对应的文件路径等。

17.2.2 数据通道

从数据源中流出的数据首先进入数据通道。数据通道在ARMS中相当于一个数据队列。它主要起到以下作用：

保证从数据源中流出的数据能立即被ARMS接收到，为下游的实时计算层充当缓冲层。

当计算节点出现任何异常时，相应时间点数据能统一从日志通道重新发送给计算层，以保证所有数据至少被实时计算层处理过一次。

日志通道在ARMS上对于用户是透明不可见的。您只需控制数据源和实时计算层的计算逻辑，日志通道的各类管理由ARMS自动完成。

17.2.3 实时计算

在数据通道中的数据会被ARMS计算层节点实时读取。ARMS实时计算层的实时计算能力由基于阿里巴巴内部开发且开源的实时计算引擎JStorm提供，实现毫秒级的流式计算处理能力。

ARMS 的实时计算层并不要求您基于实时计算引擎编写流式计算程序，而是通过拖拽方式方便地定义出实时计算任务。在 ARMS 中，您只需要基于交互界面做以下事情：

- 定义数据的清洗逻辑：例如一行日志数据是以“|”或是 “；” 符号进行切分(清洗的一种逻辑)，切分后的 Key-Value 设定等。
- 定义数据集的聚合计算逻辑：例如数据以何种维度进行聚合（Group By），聚合的计算方式（Sum，Max）等。

在监控任务中，通过定义清洗和聚合计算逻辑来产出计算结果，最终的计算结果会形成一个个特定的数据集，进行持久化的存储。

17.2.4 持久化存储

作为一款致力于提供端到端实时监控解决方案的产品，ARMS除了提供实时计算能力以外，同时提供了将计算结果持久化以供下游使用的能力。

在 ARMS 中，持久化存储通过列式存储实现，以达到高吞吐和高扩展的性能要求。在实时计算中产生的一个个数据集以 ARMS 特定优化的数据结构方式存储在列式存储中。存储具体结构对外部透明，不需要用户干预。您只需要在实时计算中通过控制产生的数据集和相关属性(索引键，保存周期等)来控制哪些数据需要进行持久化存储，并管理对应的存储空间。

ARMS 持久化到后端列式存储的数据结构对外部透明。您可以通过访问数据展示层来间接访问列式存储的数据结果。

17.2.5 数据展示层

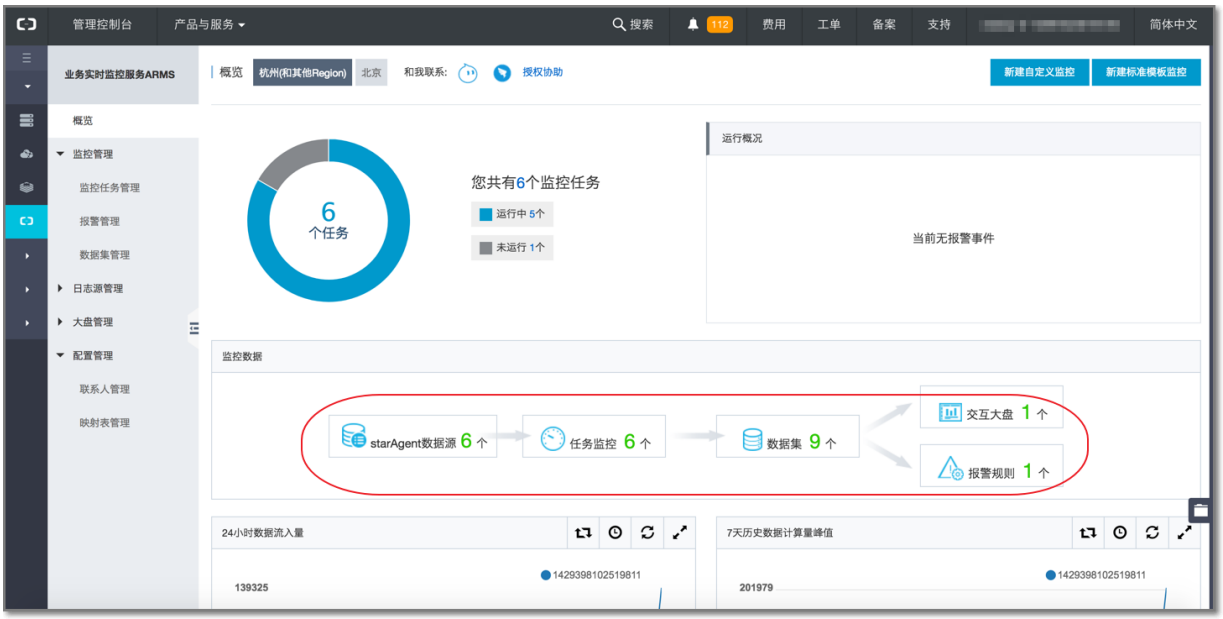
ARMS的监控数据结果一般可通过以下三种方式进行利用。

- RESTful API：最直接的方式。通过 RESTful API 基于数据集定义的各类查询 Key 对数据结果进行访问。
- 交互式大盘：ARMS 中用户基于数据集自定义的一组交互式报表。一个交互式大盘可用于多个数据集的不同图表形式的展示。查询时间跨度可自定义。
- 报警通知：通过定义报警策略，定期对数据集结果进行抓取和匹配，对符合特征的事件进行包括 Email ，短信方式的报警。

17.3 功能特性

在ARMS中，最重要的功能和对应任务流程和术语如图 *50: ARMS*对应功能流程图所示。

图 50: ARMS对应功能流程图



其中：

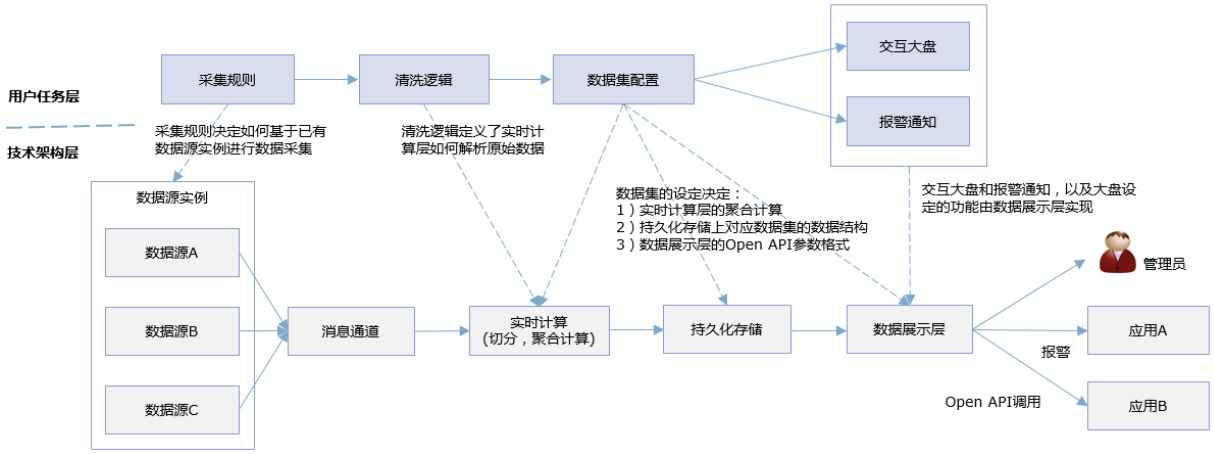
- 数据源：ARMS 获取数据的地方，目前主要支持StarAgent数据源采集增量数据、MQ数据源采集等。
- 监控任务：一个任务代表 ARMS 从数据抓取、数据处理、数据存储到结果展示和导出的一个实例。
- 数据集：代表一个监控业务的数据结果，其结果可以用 Open API 导出。数据集直接被报表控件和报警规则依赖。
- 交互大盘：基于数据集自定义的一组交互式报表。
- 报警规则：定义如何从既有数据集中产生报警。

17.3.1 关键流程

在 ARMS 中，最关键的流程是定义监控任务。通过定义一个监控任务，可以利用数据源产生一系列的监控结果，包括生成数据集、报表控件和报警规则。

如图 *51: ARMS创建任务流程图*所示，创建一个任务所需要的大致流程和组件，以及组件在 ARMS 技术架构中的依赖方式如下。

图 51: ARMS创建任务流程图



下文对任务组件作简要说明。

- 采集规则(必选)：定义数据如何从不同的数据源实例进行采集。
- 清洗逻辑(必选)：定义如何解析采集到的数据。
- 数据集配置(必选)：通过配置数据集来定义任务如何基于采集到的数据做聚合计算，持久化存储，以及 Open API 访问输出。
- 报警通知(可选)：基于数据集，通过定义报警含义和通知方式，提供报警能力。
- 交互大盘(可选)：通过集成数据集和报警通知数据提供可视化大盘能力。

17.3.2 监控任务

在ARMS中，监控任务主要分为两大类。

- **预定义任务**：如异常堆栈监控，商品销售量统计等。通过创建这类任务，您可以直接使用预定义的清洗逻辑、数据集、报表控件的组件，快速组装出一个针对特定场景的监控任务。
- **定制任务**：根据提示步骤，一步步手动定制任务的各类组件，组装出一个完整监控任务。

创建了监控任务后可在相应的任务管理界面进行管理。除了查看、删除以外，还可以针对监控任务进行起停操作。任务只有被启动的时候ARMS才会进行数据采集、计算和存储数据。当任务被停止的时候，以上工作也会被停止。

17.3.3 采集规则定义

采集规则定义了数据如何从数据源实例中进行采集。您可以基于已定义的数据源建立采集规则。

- StarAgent数据源：直接选取填写目标IP列表即可。
- MQ数据源：直接填写您的Topic和Topic所在的Region即可。

17.3.4 数据清洗定义

每个监控任务对应一个清洗逻辑。清洗逻辑定义如何解析采集到的数据。对于文本类数据，ARMS支持多种数据清洗方式。例如：

- 通过特定分隔符如“ | ” “，” “=” 对数据进行清洗，从而清洗出不同的 Key-Value (KV) 。一个极简的例子包括：itemID=abc|amount=100 的数据日志会被清洗成 itemID 为 String abc，amount 为 int 100，一共两组 Key-Value 。
- 支持基于JSON格式的数据，通过解析 Hash 数据结构清洗出不同的 KV 。
- 支持用户自定义的清洗逻辑，如基于不同清洗符的清洗嵌套等。

17.3.5 数据集配置管理

数据集是 ARMS 中实时监控数据计算和持久化的重要概念。一个监控任务可对应一个或多个数据集。

创建数据集

在定义了清洗逻辑以后，通过以下方法定义数据集：

- 直接创建：创建一个数据集，并定义其维度 (Open API 查询 Key) ，统计值 (Value) 。
- 间接创建：通过创建一个报表控件并定义控件要展示的值和维度，或通过创建一个报警通知并定义要监控的值和维度，来间接创建一个数据集。

数据集与实时计算逻辑和数据导出格式

无论使用直接创建还是间接创建，当创建了一个数据集以后，定义的维度 (Key) 和统计值 (Value) 将直接决定数据在 ARMS 中如何进行实时计算，以及其 Open API 的查询参数组合和返回值方式。

一个极简单的例子，例如某电商想统计各类商品的各个时刻的实时销售额，用于实时展示和事后统计。其设计的统计维度和统计值为：

- 查询维度为时间 (TimeStamp) 和商品类目 ID(String)。
- 统计值为销售额 (Sum(Int))。

那么：

- 首先，在实时计算中，ARMS 在后面的 JStorm 引擎中会针对大量的输入数据作基于时间和商品类目 ID 作类似于 Reduce 的计算，在计算中对销售额做 Sum 操作。
- 计算后的结果，根据聚合粒度实时在存储层中持久化。其对应的查询 Open API 中的必选查询 Key 为时间和销售类目 ID，返回的 Int 值表示指定时刻和商品类目的销售总额。

数据集的聚合粒度和保存周期

在进行数据集配置时，您可以定义聚合的粒度，例如 1 分钟聚合一次还是 1 小时聚合一次，以及响应的数据的保存周期。一个数据集的聚合粒度和保存周期设置将直接影响其在 ARMS 的持久化存储层的存储容量。

对于大多数场景，您可能想定义以下聚合和保存的方式组合。这样既可以保证最近时刻的数据精确性，又可以满足长期的统计工作需求，而且还最大限度利用了空间。

- 1 分钟的数据聚合频率，保存 7 天。
- 1 小时的数据聚合频率，保存 30 天。
- 1 天的数据聚合频率，保存 3 年。

配置了数据集以后，可以通过数据集管理界面对数据集进行管理，包括启动/停止操作。

- 数据集启动操作将保证在对应任务启动时，对应的计算将被执行且结果持久化到存储层。
- 数据集停止时，即便其对应的任务在启动状态，数据集对应的计算也不会被执行。数据集停止期间未被处理的数据流亦无法被回溯执行。

17.3.6 报警规则管理

您可以直接创建一个报警规则，或者基于一个现有的数据集创建一个报警规则。

报警规则是对一个现有数据集的处理定义，包括：

- 需要判断的指标阈值。
- 超过阈值后的处理规则。

17.4 技术规格

指标项	规格要求
服务能力	• 单集群总配置为120Core计算能力， 40TB裸存储能力。 • 集群标称负载下计算能力峰值约为400MB/s, 平均延迟为3秒以内。
基本功能	接入端支持MQ ，文件日志。
	基于任意日志或数据格式的清洗（切分）功能，能切分的格式包括KV、Json、Exception、以及其他各类特定日志，如Nginx、Apache HTTP以及其他各类用户自定义日志。
	基于数据清洗结果的监控算子支持Sum、Count、Max/Min、抽样、TopN、Count Distinct，以及其他基本算子如+-*/、同比、环比等。监控结果支持多维度查询。

指标项	规格要求
	监控结果（数据集）支持API查询，支持报表展示，支持报警检测。
	监控实时性达到 3 秒以内，大盘刷新在15秒以内，报警实时率在2分钟以内。
	支持各类在线图表格式，包括饼图、柱状图、翻牌器、折线图、面积图等；支持交互式图表，时间和关键输出指标可交互输入；在线图表可自动刷新，频率不低于10秒一次。
	数据集支持上钻下钻操作。
	计算和存储支持在线扩容。
	报警支持短信、邮件和其他定制命令接口格式；报警级别定制；报警内容可基于数据集结果任意定制。
	报警可过滤，防止报警风暴。
	支持数据生命周期管理，可区分时间力度，保存周期可定制。例如按分钟聚合的结果保存7天，小时粒度保存一个月，天粒度保存一年。
开放性	运维管理平台可以支持两种类型的API： • 数据API ，支持对数据结果的查询。 • 管理API ，支持对系统管理和运维的自动化集成。
安全隔离性	有租户概念，不同租户之间数据和任务不可见。
	不同租户之间有性能QoS ，单个租户的并发不影响其他租户的性能。
	不同租户之间的数据生命周期管理互不影响。
可靠性	服务SLA 99.9%。
	数据SLA 99.9999%。
技术成熟	在阿里巴巴内部经过 5 年实践考验，包括基础架构监控、电商监控、物流监控等场景。

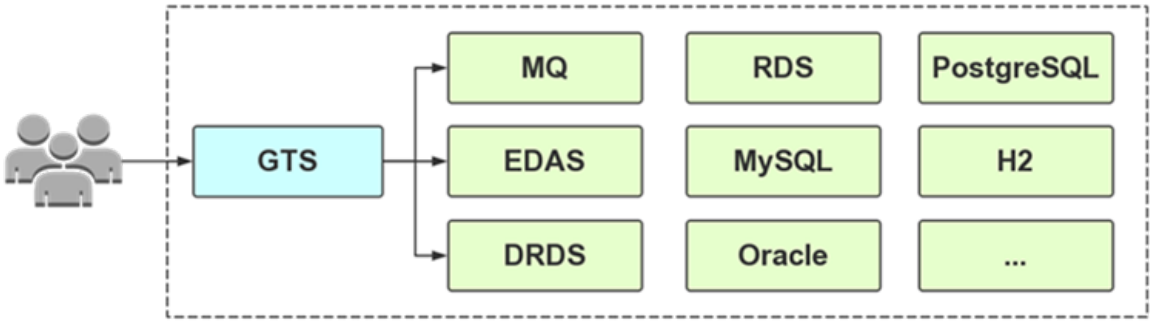
18 全局事务服务GTS

18.1 产品概述

全局事务服务（Global Transaction Service，简称GTS ）是一款高性能、高可靠、接入简单的分布式事务中间件，用于解决分布式环境下的事务一致性问题。

传统的事务主要是指单机数据库的ACID特性，GTS在支持分布式数据库事务的基础上，将事务的范围拓展到了多种资源，让分布式环境下的多个资源的操作加入事务的范畴，赋予了分布式资源操作ACID特性。

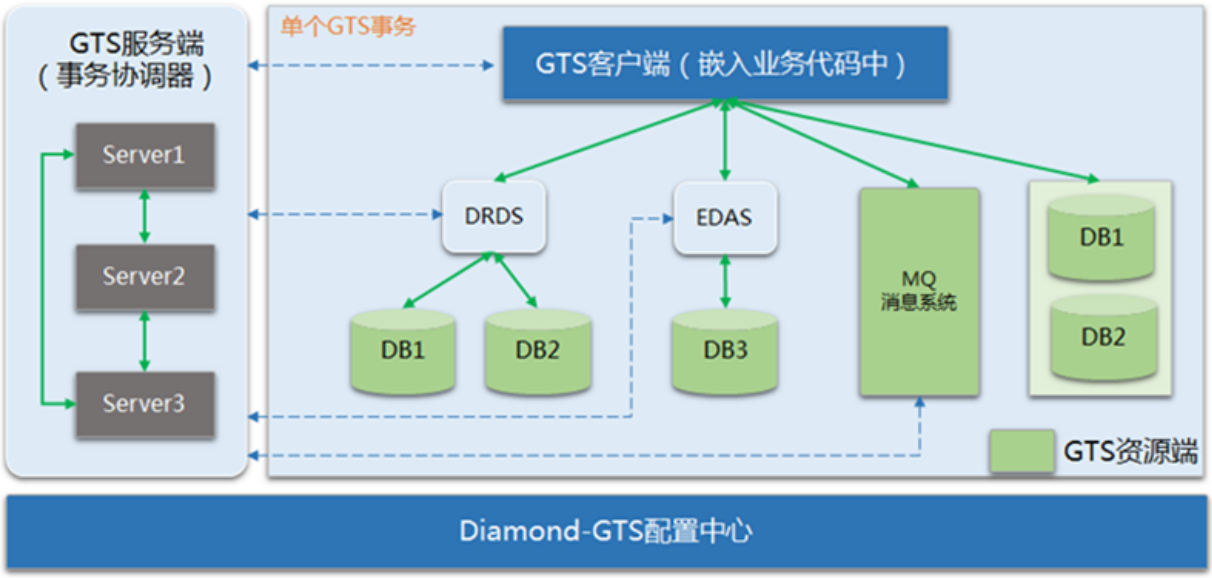
该产品支持DRDS（Distributed Relational Database Server，分布式关系型数据库服务）、RDS（Relational Database Server，关系型数据库服务）、Oracle、MySQL、PostgreSQL、H2等多种数据源，并可以配合使用EDAS（Enterprise Distributed Application Service，企业级分布式应用服务）、Dubbo及多种私有RPC框架，同时还兼容 MQ（Message Queue，消息队列）等中间件产品，能够轻松实现分布式数据库事务、多库事务、消息事务、服务链路级事务及各种组合，策略丰富且兼顾易用性和性能。



18.2 产品架构

全局事务服务GTS主要由GTS服务端、GTS客户端和GTS资源端三大部分组成，整体的系统架构如图 *52: 系统架构*所示。

图 52: 系统架构



GTS服务端

GTS服务端是分布式事务的协调者，提供高可用、高可靠、稳定高效的事务协调能力。

主要负责分布式事务的推进，包括：

- 为GTS客户端发起的分布式事务请求分配全局唯一的事务ID。
- 处理GTS资源端提交的事务分支注册及状态。
- 负责全局事务的提交或回滚。

GTS客户端

GTS客户端是分布式事务的发起方，部署在您的代码中，是您使用GTS的接口。主要负责：

- 发起事务，界定事务边界。
- 通知GTS服务端开始或者提交一个分布式事务。
- 从配置中心获取GTS服务端列表。
- 控制GTS资源端执行业务的数据操作。

GTS资源端

GTS资源端（包括数据库、消息系统等）负责具体的资源操作，在操作过程中，记录必要的事务日志并将执行状态汇报给GTS服务端。

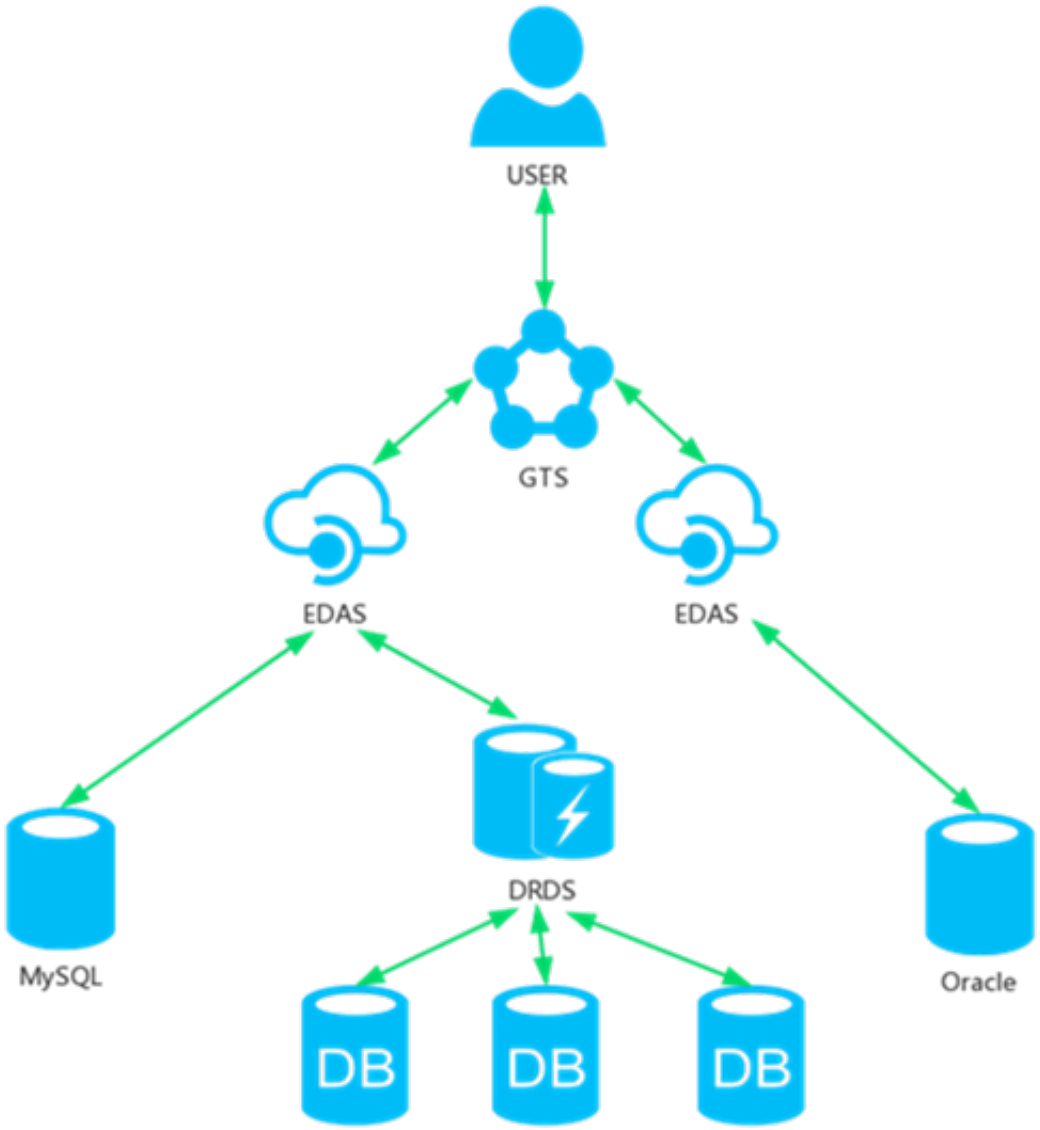
18.3 功能特性

能力开放场景

GTS支持DRDS、RDS、Oracle、MySQL、PostgreSQL、H2等多种数据源，并可以配合使用EDAS、Dubbo及多种私有RPC框架，同时还兼容MQ消息队列等中间件产品，能够轻松实现分布式数据库事务、多库事务、消息事务、服务链路级事务及各种组合，策略丰富且兼顾易用性和性能。

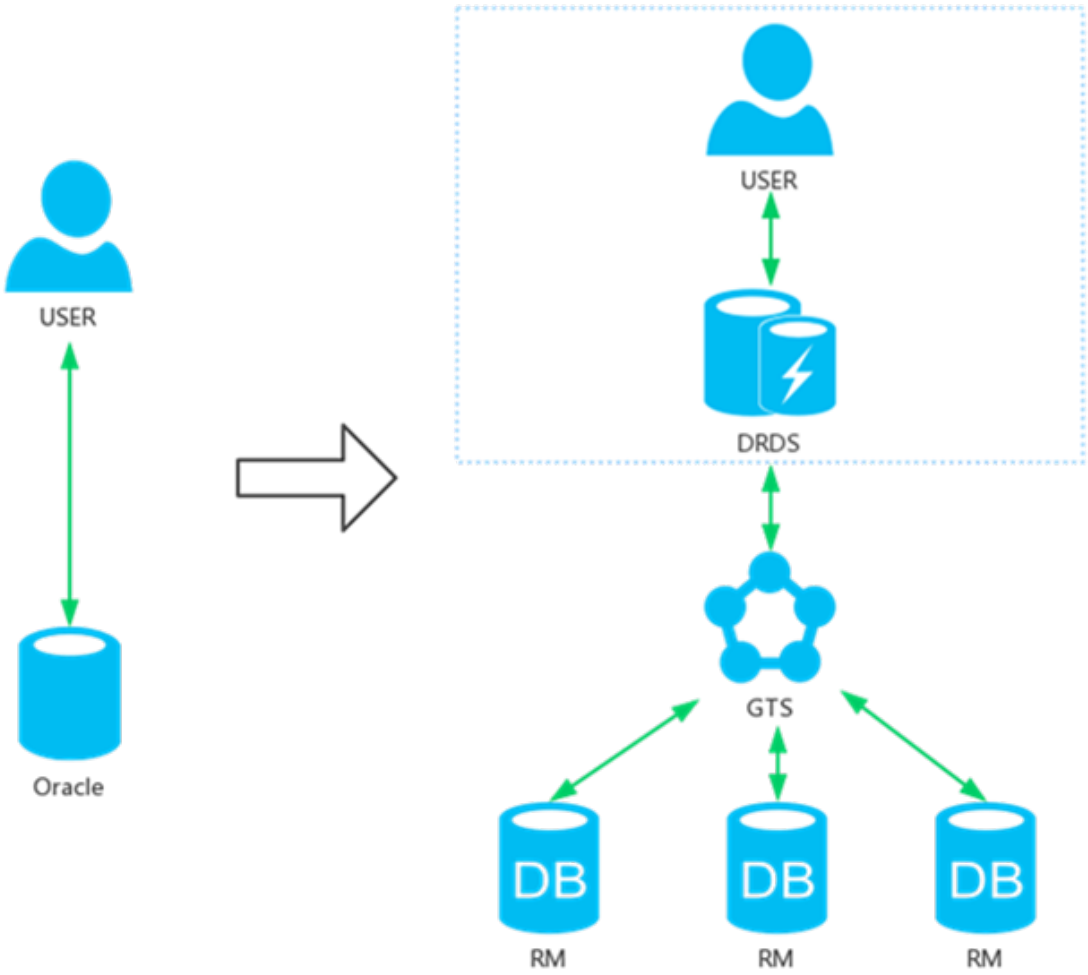
分布式服务事务解决方案

GTS支持Dubbo和HSF两种服务的事务，产品上已经与EDAS打通，提供跨库、跨服务的分布式事务支持，实现业务链路级别的分布式事务。开发简单，只需要在客户端声明一个注解，界定事务边界。例如一个库存调配案例，A店库存需要减1，B 店库存需要加1，库存调用为服务化接口。GTS可以保证两次服务调用在一个事务中完成。



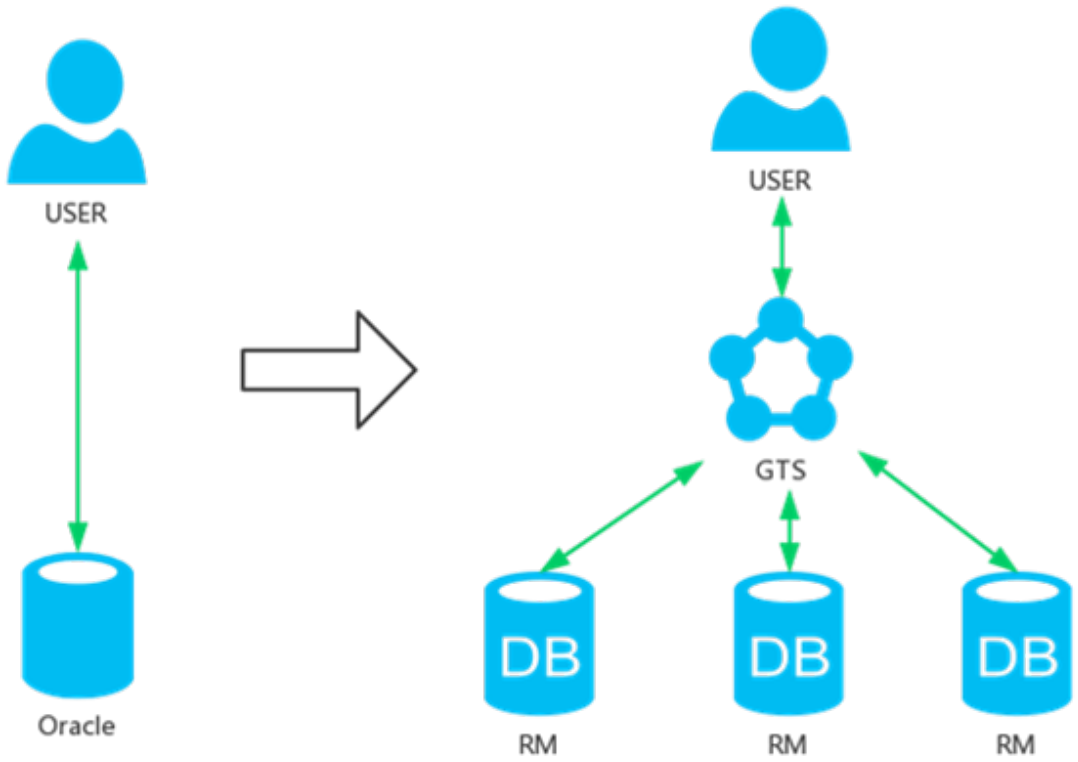
DRDS事务解决方案

GTS提供标准的SQL语法，以极低的开发成本，为分布式数据库DRDS提供分布式事务能力，是企业从单机数据库迁移到分布式数据库的利器。例如一个彩票追号案例，系统使用的是DRDS数据库，数据分库键为彩票号，追号会让这些号都落入不同的数据分库中，GTS可以保证这些分库中的数据操作在一个事务中。



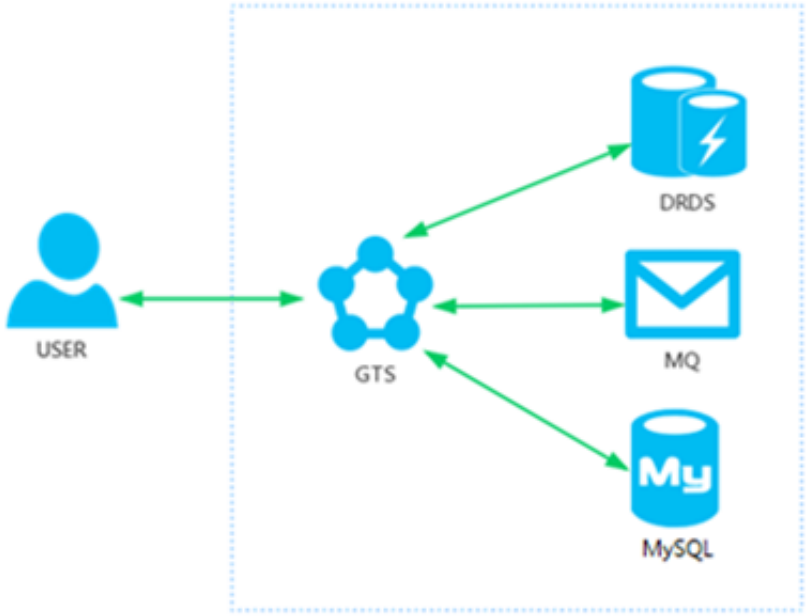
RDS事务解决方案

GTS提供标准的SQL语法，以极低的开发成本，帮助您实现跨RDS数据库的事务，让您的数据库水平拆分和垂直拆分没有事务的后顾之忧。假设您有两组历史数据分别存在两个RDS数据库中，有业务希望对两个RDS数据库中的操作放到一个事务中，GTS可以轻松解决这类跨数据库的事务问题。



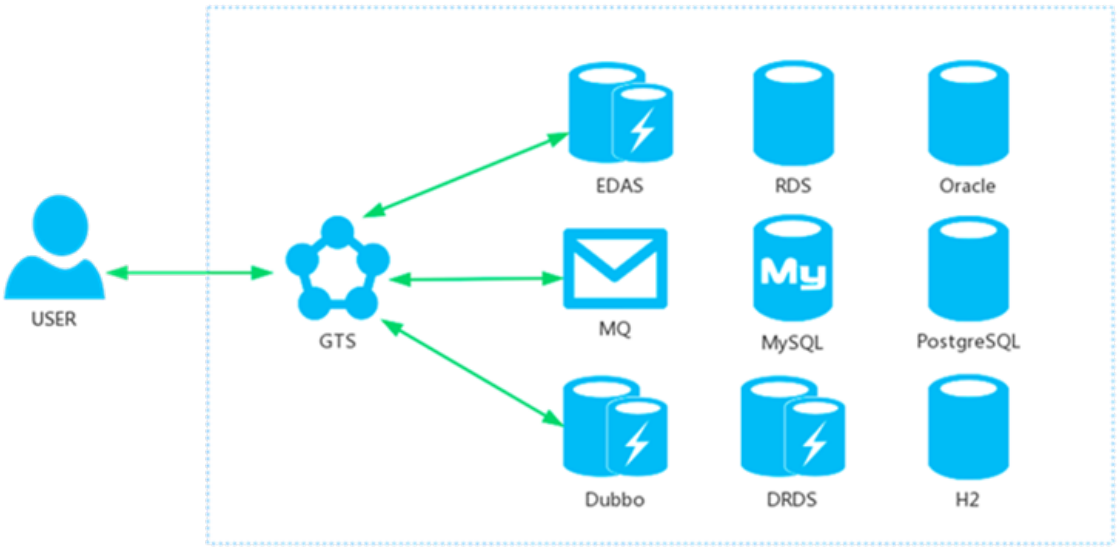
MQ事务解决方案

GTS已经和MQ产品打通，提供事务消息处理能力，业务链路完成时发送消息，任一阶段异常时回滚消息。开发简单，在客户端声明一个注解，界定事务边界，调用MQ的API发送消息。例如一个银行扣款的案例中，业务希望在操作完用户数据库后发送一个消息通知用户，两件事要么同时成功，要么同时失败。GTS可以将扣钱操作和消息要放到一个事务中，保证操作一致性。



混合事务解决方案

以极简单的接口实现跨服务（EDAS、Dubbo）、跨数据库（DRDS、RDS、MySQL）、跨消息（MQ）的通用分布式事务管理。例如一个下单用券的案例，业务要先在订单数据库中新增一个订单，再调用代金券服务新增一个使用，最后发消息通知客户。GTS可以将服务、消息及数据库操作放到一个事务中，保证三个资源的一致。



18.4 技术规格

表 7: 技术特性参数列表

指标项	规格要求
访问处理性能	在网络通信状况良好，不考虑不确定的网络异常的情况下，GTS单分支RT 2ms以内。在某些场景下，3台4c8g的普通服务器组成的集群可以支撑3万TPS以上的分布式事务。
基本功能	支持DRDS、RDS、Oracle、MySQL、PostgreSQL、H2等多种数据库的跨数据库事务。
	支持EDAS、Dubbo及多种私有RPC框架的跨服务事务，并深度兼容了EDAS服务框架。
	支持MQ消息队列的事务消息。
可靠性	中间状态多份落盘存储，经过严格断电测试，严格保证数据一致性。
高可用性	GTS具有同region高可用特性，即使突发事件造成集群中某一台机器挂掉，GTS仍然能够提供原本一半的服务能力。
技术成熟	基于阿里内部长期使用与沉淀的高可用高性能分布式集群技术产品构建，团队成员具有处理这一领域问题的丰富经验。

19 云服务总线CSB

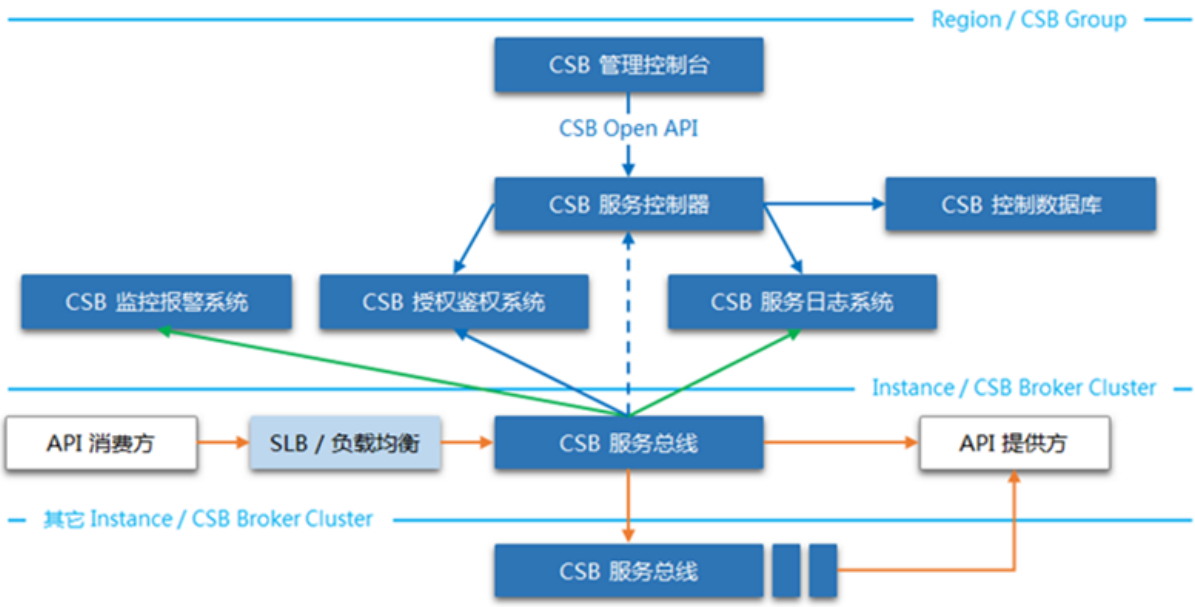
19.1 产品概述

云服务总线 (Cloud Service Bus，简称 CSB) 是处理系统间服务互通和管理的计算服务，面向专有云和专有域，帮助企业在自己的多个系统之间，或者与合作伙伴以及第三方的系统之间实现跨系统跨协议的服务能力互通。各个系统以发布、订阅服务API的形式相互开放，并对服务API进行统一管理和组织，围绕API互动，实现企业内部各部门之间，以及企业与合作伙伴或者第三方开发者之间业务能力的融合、重塑、和创新。

云服务总线CSB主要针对需要对各系统间服务访问和对外开放进行管理和控制的场景，比如安全授权、流量限制等。

19.2 产品架构

云服务总线CSB由服务总线系统、管控系统、运维监控系统组成，整体系统架构如下图所示。



服务总线系统

服务总线系统提供高可用、稳定高效、可线性扩容的服务能力和服务访问控制。

服务总线系统从服务控制器获得服务定义、服务控制、以及服务授权信息，用以处理API消费方的服务调用请求，包括认证鉴权、协议转换、流量控制等服务控制能力，以及和其他服务总线实例协调处理跨多个实例的级联式服务调用。

在服务处理过程中，服务总线系统由授权鉴权系统支持完成对服务调用的认证鉴权，服务处理日志会推送到服务日志处理系统，服务总线系统的状况也会实时推送到监控报警系统。

管控系统

管控系统提供了灵活的服务管理和组织功能。

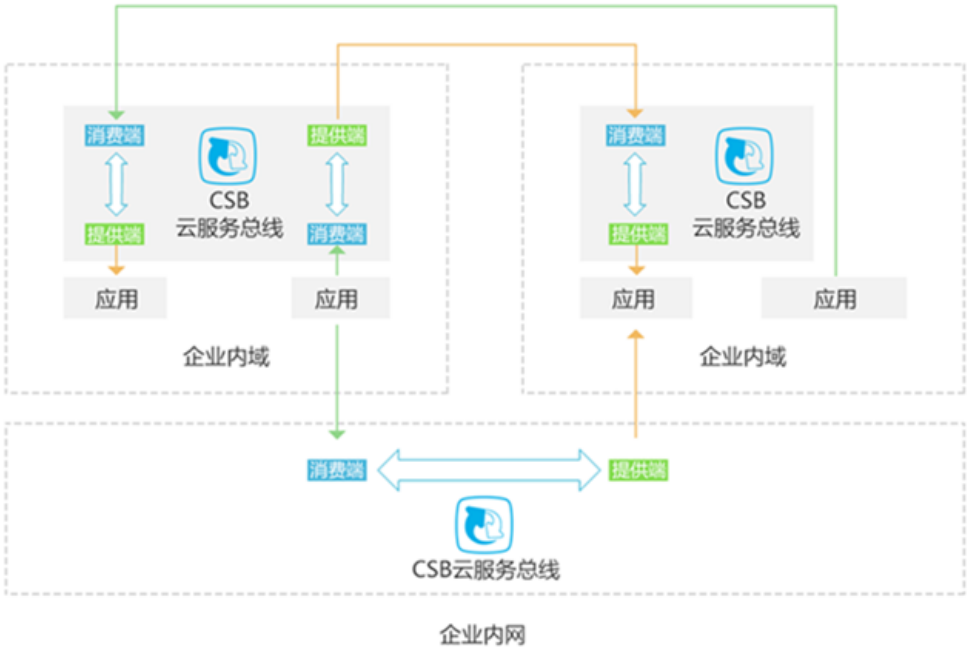
您可以使用控制台来进行服务发布和订阅，以及对已发布、已订阅的服务进行管理和组织。和控制台一样，得到适当授权的用户或者系统也可以直接调用由服务控制器提供的CSB OpenAPI ，进行服务和平台的管理。

19.3 功能特性

19.3.1 能力开放场景

面向企业专有云，其典型场景为内部（各个部门、地区、业务系统之间）互通以及内外（与合作伙伴、第三方开发组织等）互通。下图所示就是一个典型的用CSB支持内部互通混合内外互通场景：

- 一个域的应用通过另一个域的CSB实例访问其内部应用服务。
- 两个域的CSB实例构成的桥接通道实现互通。
- 各个域之间通过企业统一的CSB实例实现互通。



实际上，云服务总线CSB产品还支持更复杂的多实例，甚至多群组的联动场景。对应分级、分层，或者有特殊的系统间隔离要求，以及更复杂的多个环境如企业内网和公共云的混合场景。

19.3.2 API服务总线

可在能力开放平台建设中提供基本的实时服务控制能力，包括：

- 协议转换：支持常用协议服务的接入和开放，处理协议转换和接口映射。
- 认证鉴权：判断是否是合法用户，是否已授权访问当前服务。
- 服务控制：支持黑白名单，以及对调用者对当前服务的访问流量限制检查。

关于服务控制，还有其他特色功能，例如请求校验、响应缓存、服务路由，以及定制化报文转换等功能，正在逐步产品化整理推出。

其中请求校验可以按照用户指定的规则来检查请求内容，判断是否是合理请求，检查入参的格式、取值方位、组合等等，来过滤无效请求，降低对后端服务提供方的压力；而响应缓存可以按用户指定的规则，对指定的API服务，在一段时间内直接缓存回复相同内容的请求，在大量消费者的情况下，可以极大降低对后方提供者的压力。

19.3.2.1 协议转换

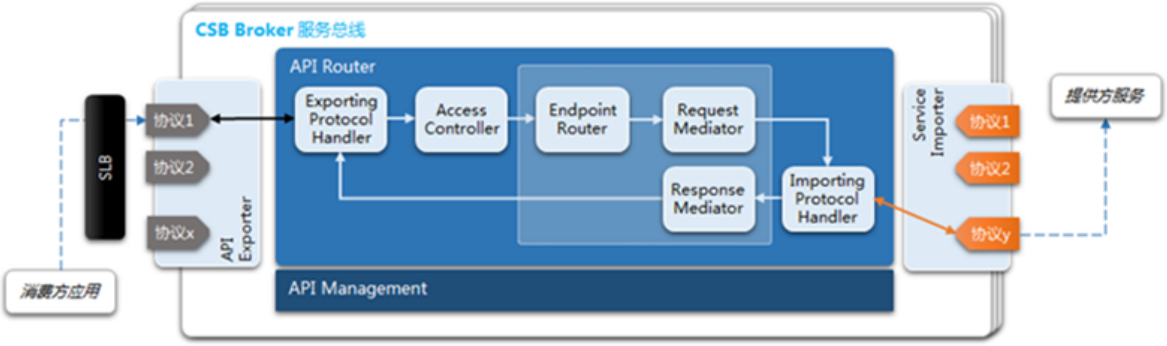
需要连通的各方系统很可能是基于不同的技术架构构建，提供的对接接口使用的协议各有不同，有的可能是HTTP OpenAPI ，有的可能是Web Service ，有的可能是应用互联网架构新建的系统，例

如Dubbo或者EDAS的HSF服务，甚至可能是完全私有定制化的协议。要实现互通，就必须解决多个不同的协议之间如何转换的问题；同时对服务的提供方而言，所提供的现存服务的协议是确定的，但开放出去，很可能面临着完全不同的消费偏好，需要支持一个服务用不同的多种协议同时进行开放。

协议转换包括两个部分：

- 将一种协议的请求转换成另一种协议的请求。

  例如HSF服务开放成了HTTP OpenAPI，CSB接收到HTTP OpenAPI请求时，需要能把请求内容转换成对指定HSF服务的访问请求，在收到HSF的服务回复后，再转换成HTTP OpenAPI的响应结果。



  还要考虑特殊情况，例如接入的服务使用了特殊的协议，或者需要遵循服务提供方的安全处理，这时就需要产品提供定制化的协议实现的支持。

  在CSB上发布服务的时候，在指定了接入的服务协议后，可以选择用哪些协议开放出去。对应不同的接入类型，即服务提供方的原有协议类型，需要提供不同的接入信息让CSB知道如何能访问这个服务。如果需要，CSB可以提供定制化服务支持特殊的协议的接入和开放。

- 将按消费方根据开放定义发出的请求，映射到接入方（提供方）的接口

  服务开放出来的接口，有可能和服务自己实际的接口是不一致的。例如有些入参或者出参并不想暴露，或者使用不同名字，等等。这时就需要在服务自己的接口（接入接口）和开放接口间做映射。CSB提供的基本方式是以服务自己的接口作为基础来指定映射。

19.3.2.2 认证鉴权

服务的开放要安全可控。云服务总线CSB的授权鉴权组件支持对接企业自己的账号系统，CSB产品提供身份认证和访问鉴权，检查服务调用者是否是合法用户，以及是否已授权可以消费该服务。

19.3.2.3 服务控制

除了基于IP的黑白名单，以及其他正在产品化的请求校验、响应缓存、服务路由等功能，CSB支持的典型的服务控制就是流量控制，可指定访问频度限制，即每秒最多调用次数。首先要支持的，是用户的单个消费凭证对单个API的访问频度限制，其次还要考虑系统级的保护。所谓系统级的保护就是防止整体服务消费频度过高，在压力大时进行调整限流，回绝部分访问请求，保持系统高效的处理效率。

19.3.3 API管理组织

API服务的管理针对两个不同的角色：发布者和消费者。从发布者角度出发，希望能对自己发布的服务有很好组织，及时掌握所发布服务的订阅、消费情况，可以方便地管理对所发布服务的订阅授权；从消费者的角度出发，希望能方便地定位找到自己想要的服务，选择质量或者服务水平更适合自己的服务，方便地管理自己的订阅，及时掌握所订阅服务的消费情况等等。

服务管理除了服务本身的生命周期管理，如注册发布、启用停用、下线注销等，还要支持服务的订阅审批、授权鉴权，以及服务目录的管理。主要的角色和操作有：

- 服务发布者：服务组管理、发布服务、发布管理。
- 服务订阅者：消费凭证管理、订阅服务、订阅管理。
- 系统管理员：实例管理、用户管理。

19.3.3.1 服务发布

发布API有如下几个关键概念。

- 接入服务：提供API对应的后端服务信息，让CSB能访问到这个已有的服务。
- 开放服务：指明API开放的协议，以及开放的接口和后端服务接口如何对应。
- 访问控制：指明API开放的策略，是否限流，对谁可见，访问是否需要授权。

目前，CSB用服务分组实现了对于用户所发布服务的原子分类，至于更复杂的服务目录结构的组织方式，往往具体的业务需求有很大差异，CSB建议业务方视具体场景单独构建。

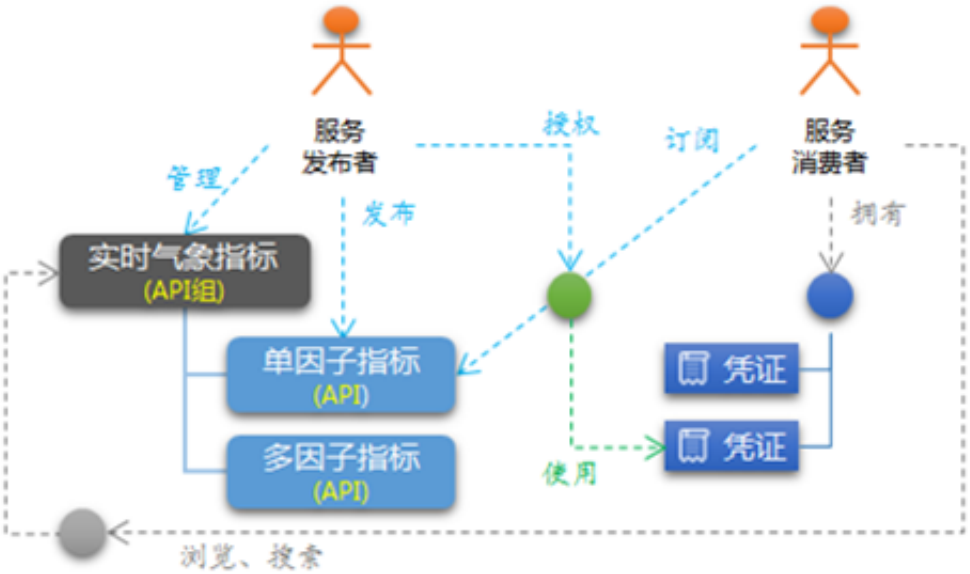
19.3.3.2 服务授权

在云服务总线CSB中，用户是对等的，没有从属概念，只有授权关系。用户可以拥有属于自己的一个或多个CSB实例，具有这些CSB实例的管理员权限。可以定义用户分组，控制其他用户对这些CSB实例的使用权限，即其他用户是否能在某个实例上发布和订阅服务。实际在企业中，往往只有少量用户会拥有CSB实例，大多数用户只是被授权访问使用CSB实例。

用户在取得某个CSB实例的访问权限后，可以在该实例上发布或者订阅服务。服务的发布者就是该服务的拥有者，可以审批授权其他用户的订阅申请。类似的，CSB用户在这里同样是对等的，CSB只关心两个用户之间针对具体服务的授权关系，并不关心用户在企业组织结构中的实际层级和隶属关系。

订阅API有如下几个关键概念。

- 消费凭证：API消费方应用在调用API时用的凭证，一个用户可有多个凭证。
- 订阅服务：API消费者使用自己的某个消费凭证来订阅API 。
- 授权服务：API发布者对API订阅授权，包括允许、禁止、流量控制等。
- 搜索服务：API消费者通过服务名、服务组名来搜索和浏览可订阅的API 。



云服务总线CSB还有群组的概念，对应于相对隔离的管理环境。例如企业的内部数据中心和阿里云公共云的某个地域（region）即是不同的群组。相应地，也有CSB群组管理员的角色，与CSB实例

管理员不同，只有群组管理员可以应用户请求创建CSB实例，以及设置跨实例的服务链路。例如在阿里云公共云环境中，CSB产品运维团队即是该群组的管理员。

19.3.3.3 服务消费

云服务总线CSB提供服务消费的计量统计，可以让服务的发布者和消费者进行查看，了解服务消费量的实时统计，以及服务消费质量信息。CSB提供从服务发布者角度看到的服务被调用详情，包括整体的服务调用量，各种类型的错误占比，近一段时间的消费曲线，以及各个消费者的消费统计等等；类似地，CSB也提供从服务消费者角度的服务调用详情。

服务消费计量也可以用来做服务计费。目前CSB没有计划提供服务调用的定价计费功能。

API消费方应用调用API有多种方式，例如HTTP API、HSF API、Web Service API等。其中HSF API的消费调用沿用HSF原有的服务调用方式，无需任何专用SDK ，如有必要可以指定CSB的服务IP进行HSF服务调用。而调用HTTP API则需要使用CSB Client SDK ，目前提供了Java版本的HTTP Client SDK。

19.3.3.4 API运维监控

日志监控

提供完整的服务和系统的日志、巡检和监控，包括对CSB所有组件系统指标的监控以及运行环境指标的监控，例如：单机和集群的负载、内存和CPU的使用率，JVM的线程、内存以及GC情况等等。还有自身服务处理的工作情况，包括各种异常情况的记录。还可以配置巡检规则，对指定的指标定期巡检并制定报警规则。此外，服务链路的分析信息也很重要，有助于快速定位是哪个环节什么原因出现问题。

系统管理

包括实例及实例上发布规则的管理，以及用户访问授权、用户组定义管理等等。

注意： 有些管理功能是只有群组管理员才有权限的，例如跨实例调用链路的定义，对用户发起的实例创建请求的审批等。

19.4 技术规格

指标项	规格要求
访问处理性能	在简单协议互联场景下，不考虑不确定的服务提供方响应时间，1KB单个服务请求消息大小，API服务节点每CPU核QPS 1000。可线性扩展，理论上支持1000个API服务节点。

指标项	规格要求
基本功能	多协议服务互联，支持HTTP OpenAPI、HSF、Dubbo、Web Service等常用协议。
	提供完备的服务安全调用，包括加密，鉴权，流量控制保护，黑白名单等访问控制。
	提供多个服务节点集群之间的级联API发布管理。
	支持从发布、订购、消费到注销的API全生命周期管理。
	提供完善的服务访问授权管理机制。
	提供及时详细的服务消费统计与质量报告。
	提供完整的针对服务链路和系统指标的日志、巡检和监控。
	提供系统用户管理，以及跨实例链路的规则配置管理。
可靠性	数据系统采用多级缓存和主备存储方案。
	API服务系统采用无状态的集群结构，采用多级缓存系统，后端的数据库宕机，不影响开放平台提供服务。
高可用性	组件集群化，包括负载均衡、网关、缓存、数据库服务节点、数据节点，可用性不低于99.9% 。
扩展性	支持服务节点的无间断扩容。
技术成熟	基于阿里内部长期使用与沉淀的高可用高性能分布式集群技术产品构建，团队成员具有处理这一领域问题的丰富经验。

20 MaxCompute

20.1 产品概述

大数据计算服务（MaxCompute）是基于飞天分布式平台，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。

20.2 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。

产品优势

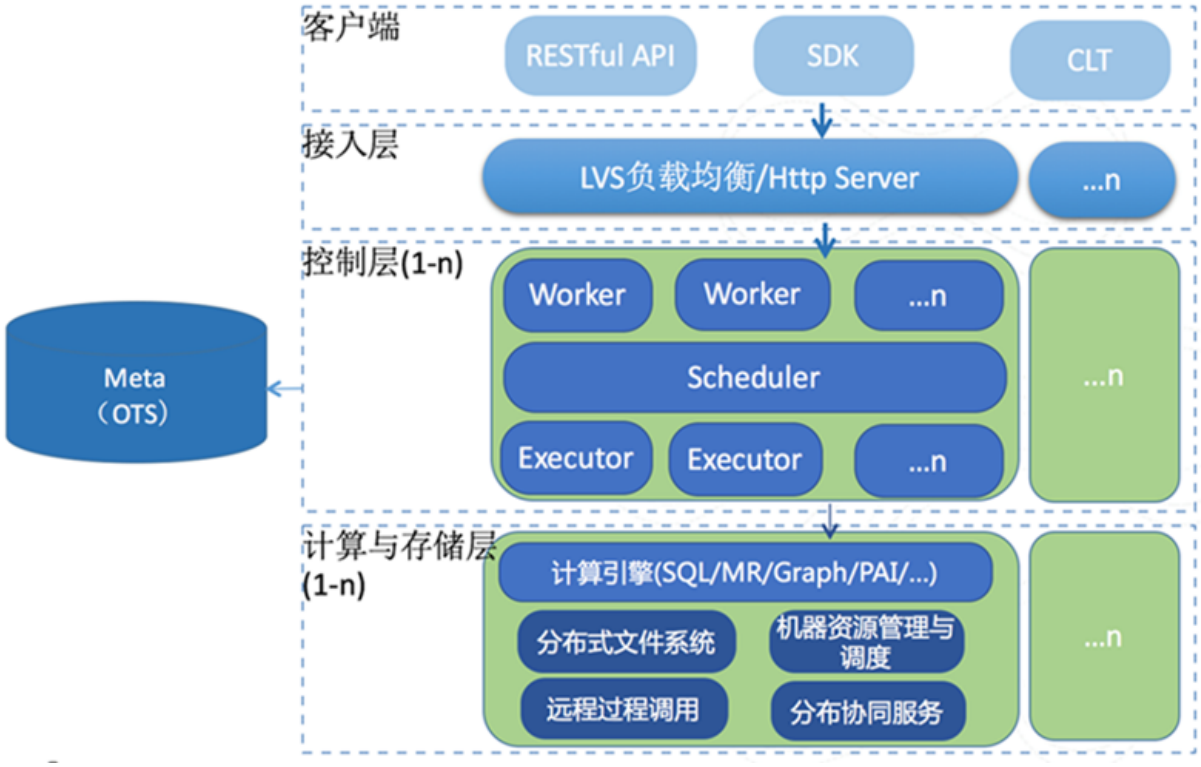
- **海量运算触手可得**：用户不必关心数据规模增长带来的存储困难、运算时间延长等烦恼，根据用户的数据规模自动扩展集群的存储和计算能力，使用户专心于数据分析和挖掘，最大化发挥数据的价值。
- **服务“开箱即用”**：用户不必关心集群的搭建、配置和运维工作，仅需简单的几步操作，用户便在MaxCompute中上传数据、分析数据并得到分析结果。

- **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工作效率。

20.3 系统架构

MaxCompute的体系架构如图 *53: MaxCompute架构图*所示：

图 53: MaxCompute架构图



MaxCompute由四部分组成，分别是**客户端**、**接入层**、**控制层**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；
- IDE：Data IDE提供了上层可视化ETL/BI工具，用户可以基于Data IDE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。

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。

20.4 功能描述

20.4.1 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。

20.4.2 SQL

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。

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，熟悉传统数据库或Hive的编程人员会很容易上手。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。

20.4.3 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。

20.4.4 Graph

Graph是MaxCompute提供的面向迭代的图处理计算框架，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，比如社交网络、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。比如对于需

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。

在分析模式下，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。比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，比如实现带权重的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。

20.5 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。

20.5.1 搭建数据仓库

图 54: 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。

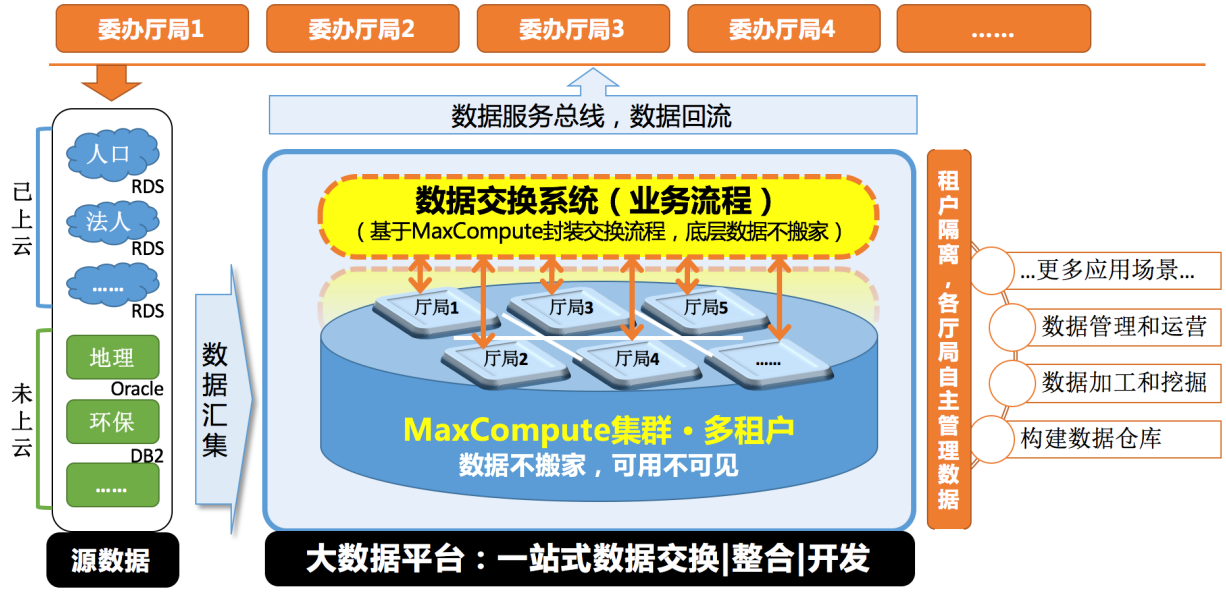
跟传统数据库相比，基于MaxCompute搭建数仓，有以下不同之处：

- **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。

20.5.2 大数据共享及交换

图 55: 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。

20.6 性能与可靠性

做为海量处理数据平台，MaxCompute单一集群规模可以达到10000+台服务器，而且支持多集群技术（用户访问方式不变，MaxCompute各个角色均可平行扩展）。

计算性能方面除MaxCompute采用C++研发本身语言性能优势外，还在引擎层面引入基于代价优化和运行时LLVM及向量化处理等技术和手段，使得计算效率上优于同类型的开源产品。

MaxCompute的数据可靠性是通过多副本技术实现的，MaxCompute底层的分布式文件系统通过将数据副本分布到不同失效的多台硬件上，能够保证正常概率损坏的硬件下的数据安全。单节点故障或磁盘单点故障时均不影响作业的正常运行。

21 大数据开发套件

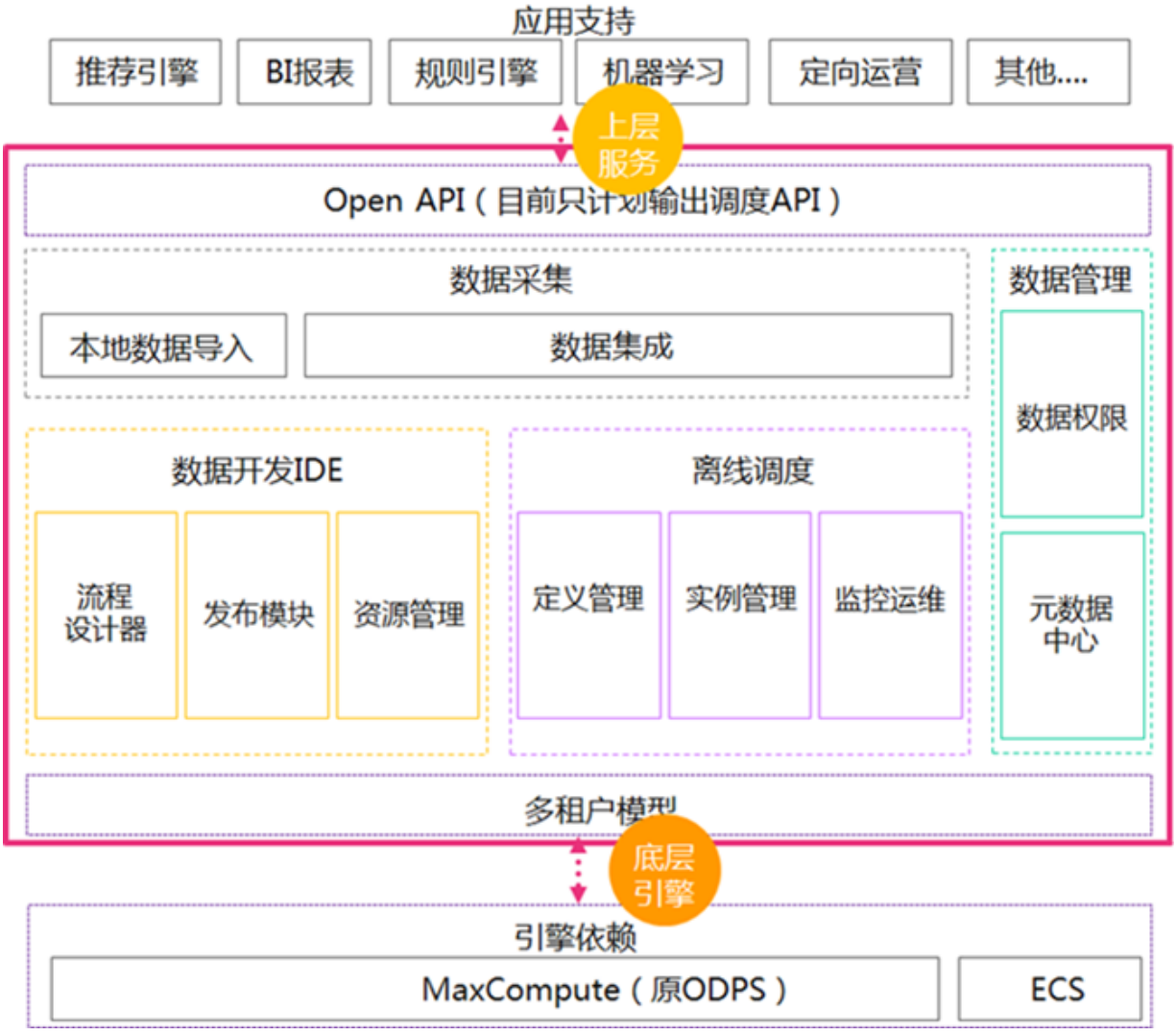
21.1 产品概述

阿里云大数据开发平台（Data IDE）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。

Data IDE以 **enable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：

- enable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；
- enable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 56: 产品组成



Data IDE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。

21.2 产品特性和核心优势

超大规模数据处理能力

Data IDE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
  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。

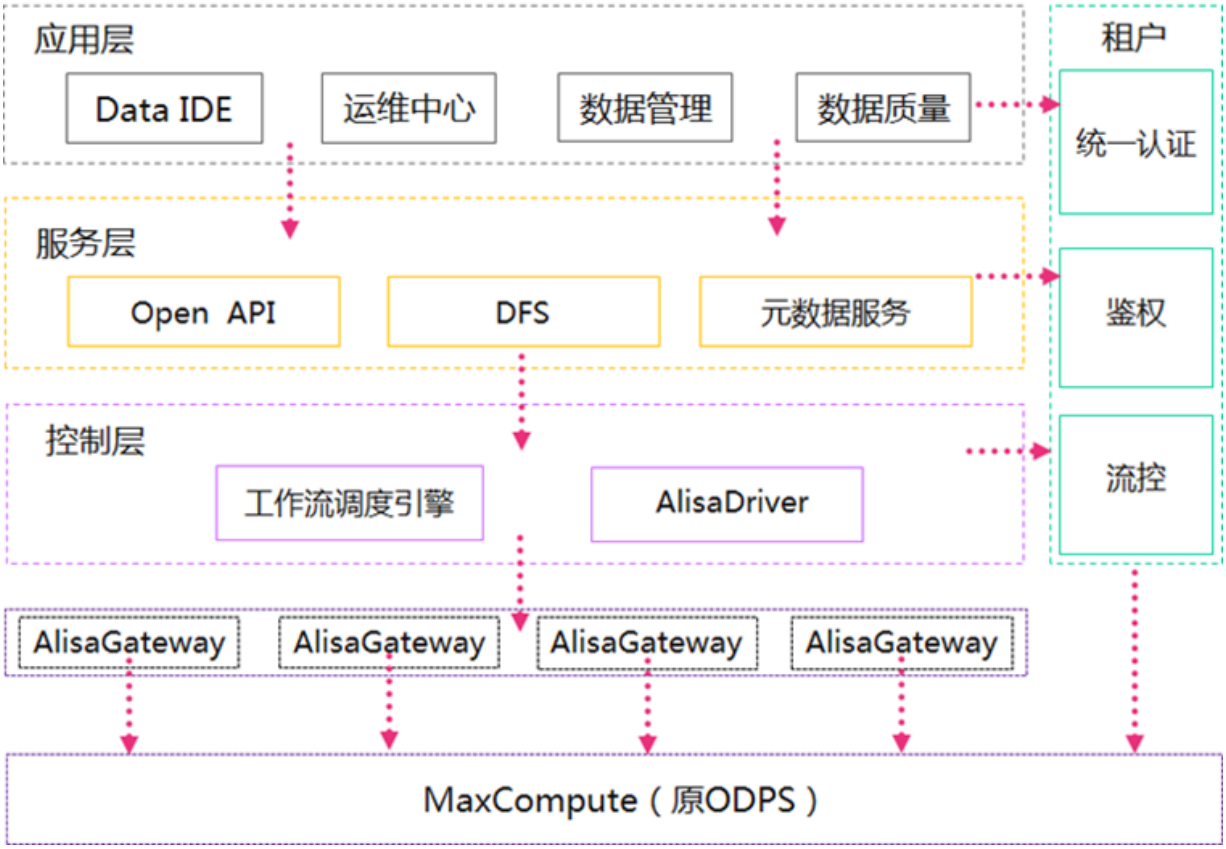
开放的平台

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

21.3 系统架构

系统开放性架构

图 57: 系统架构图



Data IDE采用组件化、服务化设计理念，分为三层：

- **控制层**：Data IDE离线加工的核心，工作流调度引擎承接着整个Data IDE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

Data IDE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，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

Data IDE拥有自己的多租户权限模型：

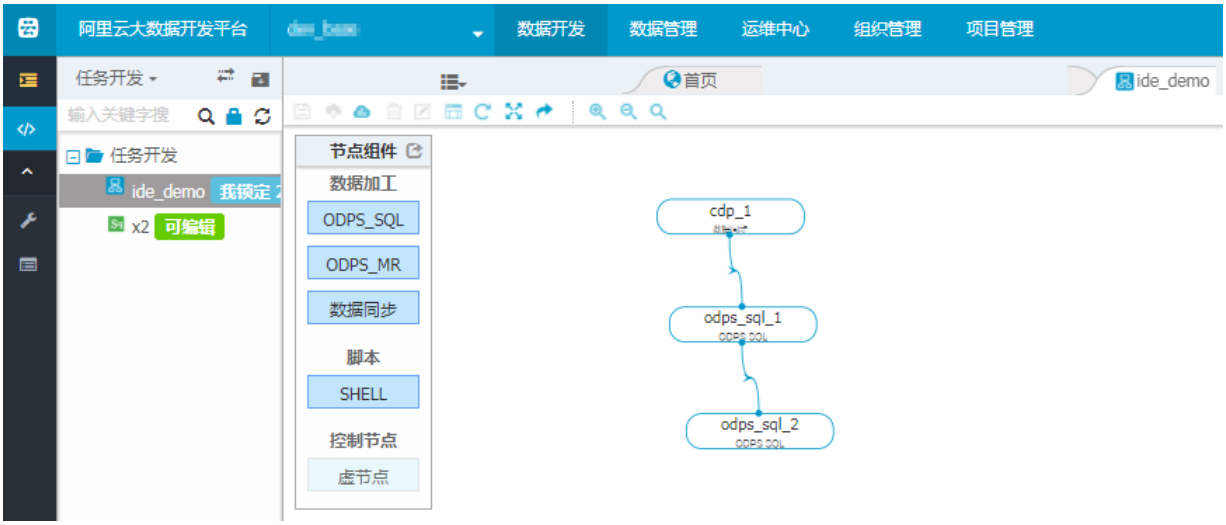
- 弹性的存储和计算资源：租户可按需申请资源配额，独立管理自己的资源；
- 租户独立管理自有的数据、权限、用户、角色，彼此隔离，以确保数据安全。

21.4 功能描述

21.4.1 数据开发IDE

Data IDE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 58: 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、

- 写数据支持的数据源：

  mysql、oracle、sqlserver、postgresql、rds、drds、maxcompute、ads、ocs、oss、hdfs

- 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，用户可轻松索引数据。

21.4.2 数据管理

数据管理为用户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：《*Data#IDE*用户指南中的数据管理》。

21.4.3 调度系统

离线调度系统为用户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：《*Data#IDE*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。

21.4.4 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：

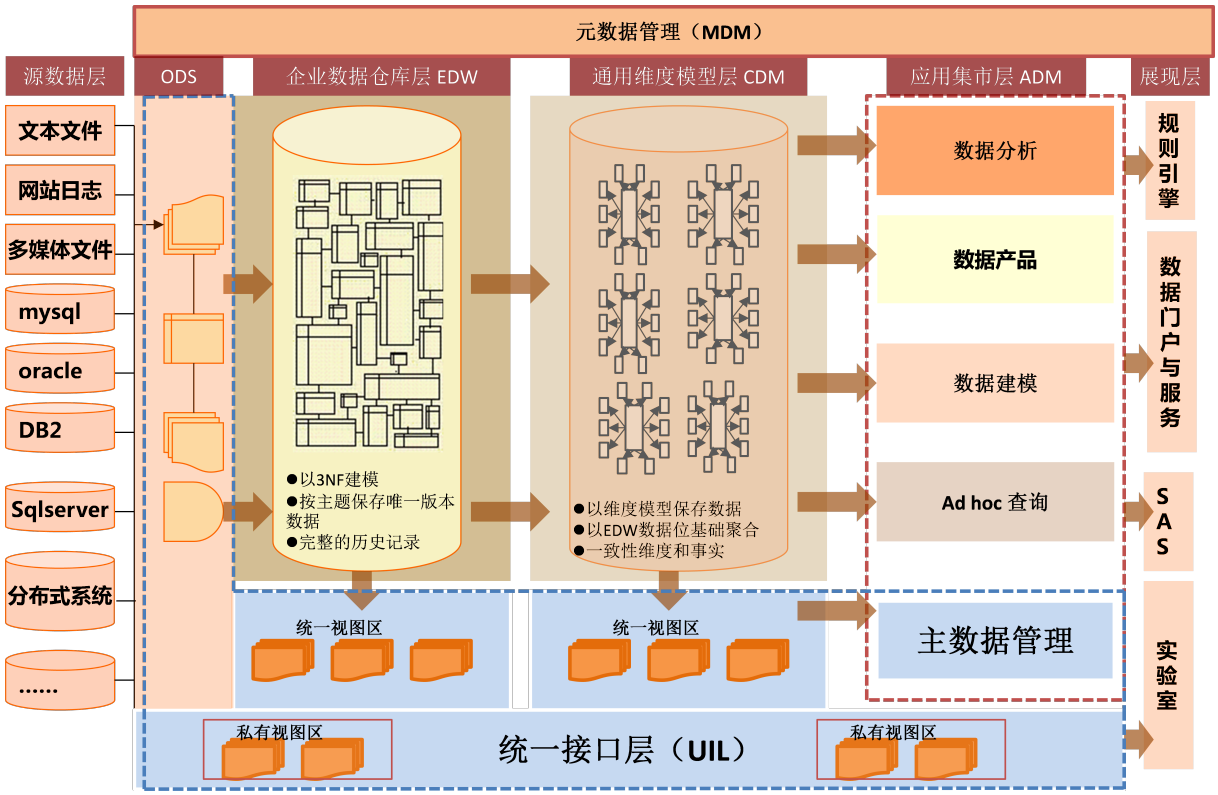
- 支持以多种数据通道（并且持续增长中）
  - 取数据支持的数据源：
    mysql、oracle、sqlserver、postgresql、rds、drds、MaxCompute、ftp、oss、hdfs、dm、sysbase；
  - 写数据支持的数据源：
    mysql、oracle、sqlserver、postgresql、rds、drds、MaxCompute、ads、ocs、oss、hdfs、dm、sysbas
- 可靠的数据质量
  - 完美支持各种数据类型的转换，保证不失真；
  - 能精确识别脏数据，进行过滤、采集、展示，为用户提供可靠的脏数据处理，让用户准确把控数据质量；
  - 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。

21.5 应用场景

21.5.1 大型数据仓库搭建

大型企业可在私有云环境下使用Data IDE来构建超大型的数据仓库。

图 59: 构建数据仓库



Data IDE为这类客户提供卓越的海量数据集成能力：

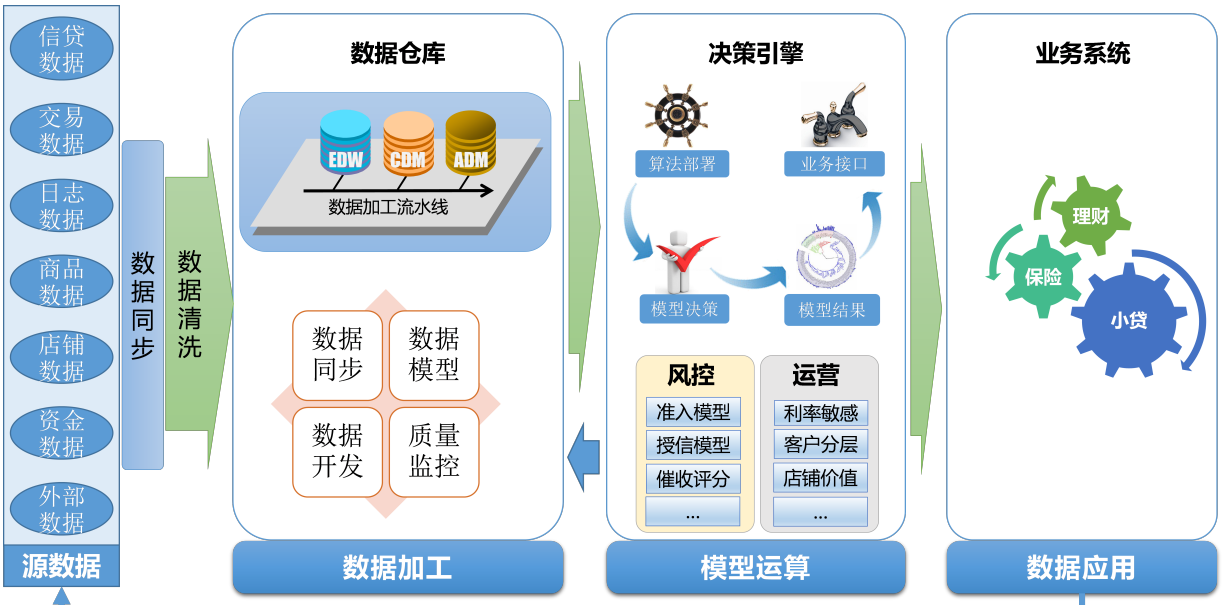
- 海量存储：可支持PB、EB级别的数据仓库，存储规模可线性扩展；
- 数据集成：支持多种异构数据源的数据同步和整合，消除数据孤岛；
- 数据开发：基于MaxCompute（原ODPS）的大数据开发，支持SQL、MR等编程框架，以及贴近业务场景的白屏化工作流设计器；
- 数据管理：基于统一的元数据服务来提供数据资源管理视图，以及数据权限审批流程；
- 离线调度：可以提供多时间维度的周期性调度能力，支持每天百万级的调度并发，并对任务调度实时监控，对错误及时告警。

21.5.2 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Data IDE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 60: 数据化运营

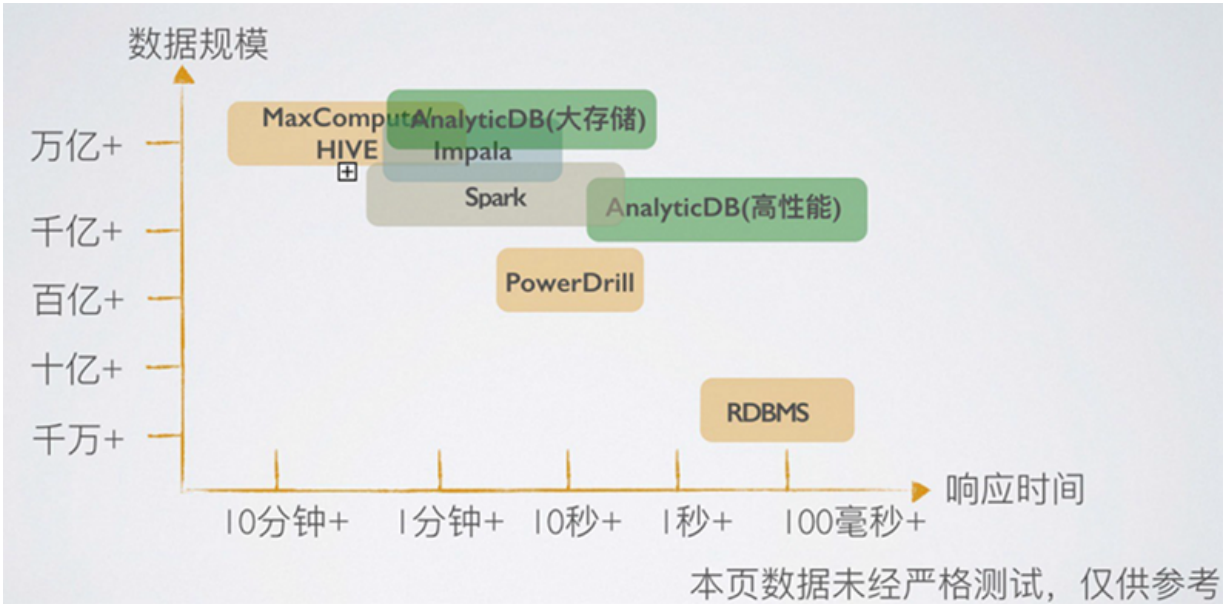

信贷数据
交易数据
日志数据
商品数据
店铺数据
资金数据
外部数据
源数据
数据同步
数据清洗
数据仓库
EDW
CDM
ADM
数据加工流水线
数据同步
数据模型
数据开发
质量监控
数据加工
决策引擎
算法部署
业务接口
模型决策
模型结果
风控
准入模型
授信模型
催收评分
...
运营
利率敏感
客户分层
店铺价值
...
模型运算
业务系统
理财
保险
小贷
数据应用
数据回流，形成闭环

22 分析型数据库

22.1 产品概述

根据IDC 2013年发布的数字宇宙研究报告（Digital Universe）显示，在接下来的8年中，我们所产生的数据量将超过40ZB（泽字节）。作为大数据特征中最重要的Volume（容量）、Velocity（数据生产速度）的两个原始特征都在发生急剧变化，使得数据处理从业务系统的一部分演变得愈发独立，企业需要加速数据分析和挖掘过程，并由报表展现为主到强调数据洞察转型，让数据直接快速产生价值（Value）。在业务系统中，我们通常使用的是OLTP（OnLine Transaction Processing，联机事务处理）系统，如MySQL，Microsoft SQL Server等关系数据库系统。这些关系数据库系统擅长事务处理，在数据操作中保持着严格的一致性和原子性，能够很好支持频繁的数据插入和修改，但是，一旦需要进行查询或计算的数据量过大，达到数千万甚至数十亿条，或需要进行的计算非常复杂的情况下，OLTP类数据库系统便力不从心了。

图 61: 产品对比图



分析型数据库（Analytic DB，原ADS）和主流数据系统进行对比（数据未经严格测试，仅供参考）时，便需要OLAP（On-Line Analytical Processing，联机分析处理）系统进行处理。从广义上，OLAP系统是针对OLTP系统而言，并不特别关心对数据进行输入、修改等事务性处理，而是关心对已有大量数据进行多维度的、复杂的分析的一类数据处理系统。在具体的产品中，通常将OLAP系统分为MOLAP、ROLAP和HOLAP三类。

多维OLAP（Multi-Dimensional OLAP，简称 MOLAP），是预先根据数据待分析的维度进行建模，在数据 的物理存储层面以“立方体”（Cube）的结构进行存储，具有查询速度快等优点，但是数据必须预先建模，无法依据使用者的意愿进行即时灵活的修改。而关系型OLAP（Relational OLAP，简称ROLAP），则使用类似关系数据库的模型进行数据存储，通过类似SQL等语言进行查询和计算，优点是数据查询计算自由，可以灵活的根据使用者的要求进行分析，但是缺点是在海量数据的情况下分析计算缓慢。至于HOLAP，则是MOLAP和ROLAP的混合模式。

而分析型数据库，则是一套RT-OLAP（Realtime OLAP，实时OLAP）系统。在数据存储模型上，采用自由灵活的关系模型存储，可以使用SQL进行自由灵活的计算分析，无需预先建模，利用分布式计算技术，分析型数据库可以在处理百亿条甚至更多量级的数据上达到甚至超越MOLAP类系统的处理性能，真正实现百亿数据毫秒级计算。

分析型数据库让海量数据和实时与自由的计算可以兼得，实现了速度驱动的大数据商业变革。一方面，分析性数据库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。更重要的是，Analytic DB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，而不是根据预先设定好的逻辑查看已有的数据报表。

更加重要的是，由于分析型数据库能够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，所以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。如淘宝数据魔方、淘宝指数、快的打车、阿里妈妈达摩盘（DMP）、淘宝美食频道等拥有 数十万至上千万最终用户的互联网业务系统中，都使用了分析型数据库。

分析型数据库作为海量数据下的实时计算系统，给使用者带来极速自由的大数据在线分析计算体验，最终期望为大数据行业带来巨大的变革。

22.1.1 存储模式

在0.9版本中，分析型数据库拥有两种存储模式，对应不同的成本和业务模型：

高性能存储模式实例：使用全SSD（或Flash卡）进行计算用数据存储，使用内存作为数据和计算的动态缓存的实例。可在双千兆或双万兆网络服务器上良好运行，具有计算性能好、查询并发能力强的优点，缺点是存储成本较高。

大存储模式实例：采用SATA磁盘进行分布式存储，作为计算用数据存储，使用SSD和内存两级作为数据和计算的动态缓存的实例。必须在双万兆网络服务器上才能良好运行，具有存储成本低的优点，但查询并发能力相对较弱，一次性计算较多行列时性能较差。

您创建数据库实例时，可以选择使用的存储模式，一经选定无法中途更改。

22.1.2 系统资源管理

分析型数据库通过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提供的分布式资源调度能力，分析型数据库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，来决定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Compute Node可选）。

您可以通过控制ECU的数量，来决定一个数据库实例所启用的ComputeNode数量，从而通过ECU类型上所配置的比例来按比例启动若干个FrontNode和BufferNode，从而达到容量的水平伸缩的目的。

FrontNode、ComputeNode、BufferNode进程可以混部在同一批物理机（默认），也可以配置参数强制不同的角色在不同的物理机上运行。

除此之外，分析型数据库的后台任务、数据库AM等也会占用一定量的系统资源。

22.1.3 计算引擎

在0.9.5或更新的分析型数据库版本中，拥有两套计算引擎：

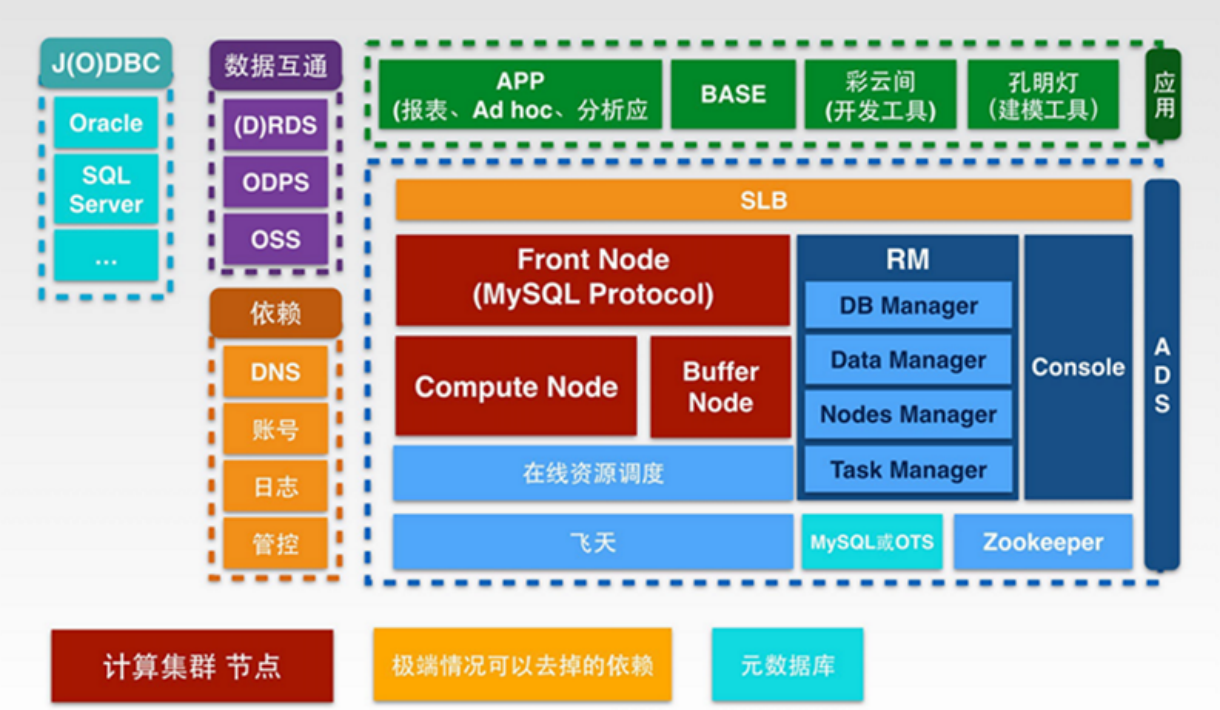
Compute NodeLocal/Merge（简称LM）：先前版本的旧计算引擎，优点是计算性能很好，并发能力强，缺点是对部分跨一级分区列的计算支持较差。

FullMPP Mode（简称MPP）：0.9.5版本新增的计算引起，优点是计算功能全面，对跨一级分区列的计算有良好的支持，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。缺点是计算性能相对LM引擎较差，并且计算并发能力相对很差。

在开启Full MPP Mode引擎功能的数据库中，分析型数据库会自动对查询Query进行路由，将LM引擎不支持的查询路由给MPP引擎，尽可能兼顾性能和通用性，用户也可以通过Hint自行决定某个Query 使用什么样的计算引擎。

22.2 产品架构

图 62: 产品架构图

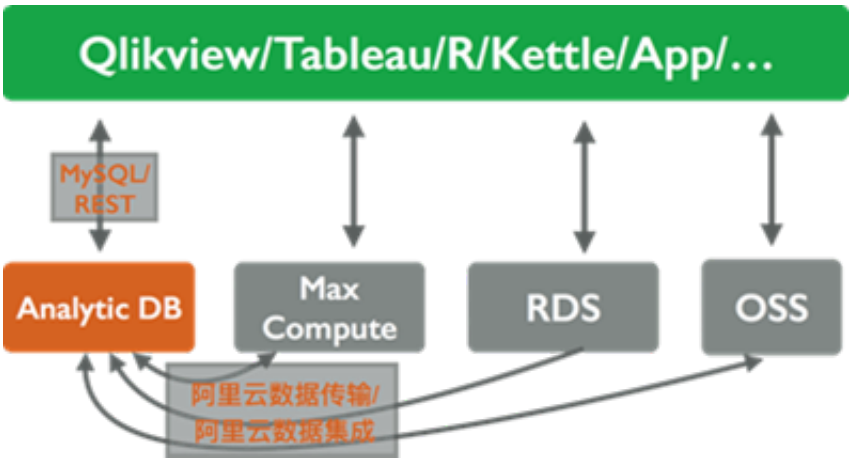


Analytic DB（原 ADS）是构建在阿里云分布式操作系统飞天之上的基于MPP架构并融合了分布式检索技术的分布式实时计算系统。如图所示，Analytic DB的主体部分主要由四个部分组成：

- **底层依赖**：包括用于进行资源虚拟化隔离、数据持久化存储、构建数据结构和索引而使用的阿里云飞　天分布式操作系统套件，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）的阿里云RDS关系数据库或阿里云表格存储，以及用于各个组件间进行分布式协调的开源Apache ZooKeeper模块；
- **计算集群**：是计算资源实际包括的内容，均可进行横向扩展。包括用于处理用户连接接入认证、鉴权、查询路由与分发以及提供元数据查询管理服务的Front　Node、用于进行实际的数据存储与计算的Compute　Node、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的Buffer Node。计算集群运行在阿里云超大规模分布式操作系统飞天上，并通过在线资源调度模块来使用飞天调度计算资源；
- **控制集群**：暨资源管理器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、管理飞天集群上的计算节点、管理数据库后台运行的任务等。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位于不同机房部署的多套计算集群；

- **外围模块**：如用于管理Front Node的分组和负载均衡的阿里云负载均衡、用于发布数据库域名的阿里云DNS系统、阿里云账号系统、Analytic DB的控制台（Admin Console）和用户控制台（DMS for Analytic DB）等。

图 63: 系统交互



在对外部系统的交互上，分析型数据库能够从MaxCompute批量导入数据，并且可以快速批量导出海量数据到MaxCompute；可以实时的将RDS的数据同步到分析型数据库中（需借助外部同步工具）；计划在0.10可以通过CDP从OSS拉取数据并快速批量导出海量数据到OSS。

对于前端业务，分析型数据库允许任何遵循MySQL 5.1/5.5/5.6系列协议的客户端和驱动进行连接。例如：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、 java、python、C/C++、Node.js、PHP、R（RMySQL）、Websphere Application Server 8.5、Apache Tomcat、JBoss等。

22.3 功能特性

22.3.1 实体

22.3.1.1 用户

- 通过云账号进行登录；
- 不同用户可以被授予不同的权限；
- 用户的操作均可被细粒度审计。

22.3.1.2 数据库

- 租户隔离的基本单位；
- 可基于数据库对计算资源进行物理或虚拟化隔离；

- 可基于数据库进行大量的系统配置；
- 数据库的创建用户为数据库的owner或数据库管理员。

22.3.1.3 表组

- 资源分配的最小单元；
- 可进行副本数、分片数、超时时间等多种配置；
- 同表组内的表可进行Join加速（维度表可和任何表组的表进行Join加速）。

22.3.1.4 事实表

- 支持标准的关系表模型；
- 表支持多级分区以及多种分区类型；
- 支持根据若干列进行数据聚集，以实现高性能查询优化；
- 单表支持最大1024列；
- 单表可支持数千亿行甚至更多的数据。

22.3.1.5 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。

22.3.1.6 列

- 支持boolean、tinyint、smallint、int、bigint、float、double、varchar、date、timestamp等多种MySQL标准数据类型；
- 分析型数据库特有类型多值列multivalue，高性能存储和查询一个列中的多种属性值信息；
- 可针对列配置不建立自动化索引，0.8版本前支持追加建立HashMap索引，0.9版本无需手动追加建立HashMap索引。

22.3.1.7 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，Analytic DB出厂时已根据机型配置预设了多种最佳的ECU型号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。

22.3.2 DDL

数据库管理：

- 通过DDL创建数据库；
- 通过DDL删除数据库；
- 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、缩容操作3.2.2表组和表管理；
- 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。

22.3.3 DML

22.3.3.1 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。

22.3.3.2 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。

22.3.4 权限与授权

授权模型：

- 支持标准MySQL模式的权限模型；
- 可以对数据库、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；
- 支持可创建数据库权限单独控制或外挂（目前只支持UMM）控制；
- 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；
- 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list users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。

22.3.5 Data Pipeline

海量数据快速导出：

- 支持任何SELECT语句的查询输出；
- DUMP DATA可以将大量数据快速导出到TFS（对内）、OSS（对外，暂未上线）等DFS中以及MaxCompute；
- DUMP DATA性能可达到1000万条数据10s内导出完毕3.5.2数据的导入；
- 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命令进行导入；
- 内置支持导入数据owner校验，保证导入安全。

22.3.6 特色功能

22.3.6.1 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；
- 对全部列根据列的数据分布智能建立索引，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。

22.3.6.2 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。

22.3.6.3 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。

22.3.6.4 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。

22.3.7 元数据

22.3.7.1 information_schema

- 最大限度兼容MySQL标准的db、表、列等信息，并且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DB进行隔离，您无法访问无权使用的元数据。

22.3.7.2 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。

22.3.7.3 sysdb

- 面向系统管理员和运维人员的元数据库；
- 可以观察Analytic DB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；
- 可以查看系统运行的运行时状态，并在有需要时可以进行修改；
- 可以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。

22.3.8 管理控制台

22.3.8.1 用户控制台（DMS for Analytic DB）

- 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运行界面以及查看状态与进度的界面；
- 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DB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。

22.3.8.2 运维管理控制台（Admin Console、Tesla）

- 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DB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；
- 提供白屏化的分布式日志提取和查看工具。

23 流计算

23.1 产品概述

Alibaba Cloud StreamCompute（阿里云流计算）是运行在阿里云平台上的流式大数据分析平台，提供给用户在云上进行流式数据实时化分析工具。使用阿里云 StreamSQL，用户可以轻松搭建自己的流式数据分析和计算服务，彻底规避掉底层流式处理逻辑的繁杂重复开发工作。利用阿里云流计算提供的全链路流式数据开发套件，用户可以享受到从数据采集、数据加工、数据消费全流程一站式解决方案，最大化实时化自身业务。

23.2 产品历程

阿里云流计算脱胎于阿里集团内部双十一实时大屏业务，在阿里集团内部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4、5年的长期摸索和发展，到最终成长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，助力更多中小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，存在大量重复的人肉工作。

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。工程师们基于 Storm 的 API 开发出大量可复用的数据统计组件 ，例如实现了简单过滤、聚合、窗口等等作为基础的编程组件，并基于这类组件提供了一套 XML 语义的业务描述语言。基于这套设计，流式计算用户可以使用 XML 语言将不同的组件进行拼装描述，最终完成一整套完整的流计算处理流程。基于 XML +Storm 组件的编程方式，从底层上避免了用户大量的重复开发工作，同时亦降低了部分使用门槛。但我们的数据分析人员仍然需要熟悉整套编程组件和 XML 描述语法，这套编程方式离分析人员最熟悉的 SQL 方式仍然差距甚远。

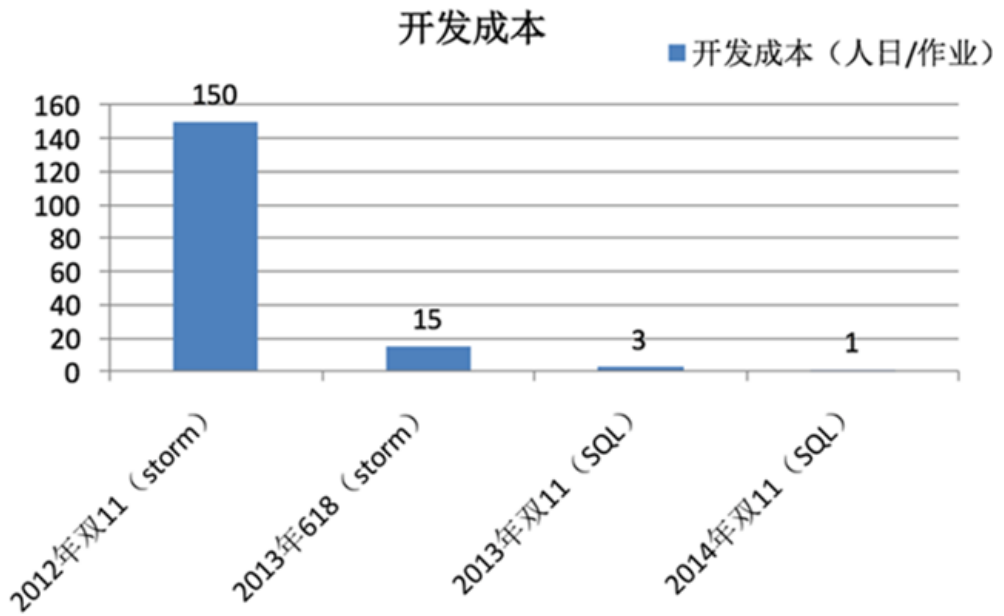
任何技术的发展一定遵循 **小众/创新** 到 **大众/普及** 的成长轨迹，而从小众到大众，从创新到普及的转折点一定在于技术的功能成熟和成本降低。阿里工程师开始思考如何更大程度降低数据分析产品门槛从而普及到更多的用户。得益于关系型数据库几十年沉淀的用户群体，使用经典的 SQL 模式去计算和处理数据一则可以对标 SQL 功能从而提炼我们的技术成熟度，二则可以利用用户熟悉的 SQL 模型可极大降低用户上手使用流计算的门槛。因此，阿里工程师最终开发一套 StreamSQL 替换了原有的 XML+ 组件的编程方式，这套系统成为今天阿里云流计算的核心计算引擎（Galaxy）。当前

这套系统以单机群数千台机器规模，在阿里集团内部服务20+BU，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流式计算集群。

当前阿里云流计算在原有 Galaxy 系统基础上，更加丰富和提升了用户的使用体验，包括提供一整套的开发平台，完整的流式数据处理业务流程。使用阿里云流计算，受益于阿里大数据多年的技术和业务沉淀，用户可以完全享受到阿里集团最新最前沿的计算引擎能力，业务上可规避阿里集团多年在流式大数据的试错和教训，让用户自身可以更快、更轻松地实时化大数据处理流程，助力业务发展。

阿里集团工程师使用阿里云流计算开发工期对比如图 *64:* 开发成本所示。

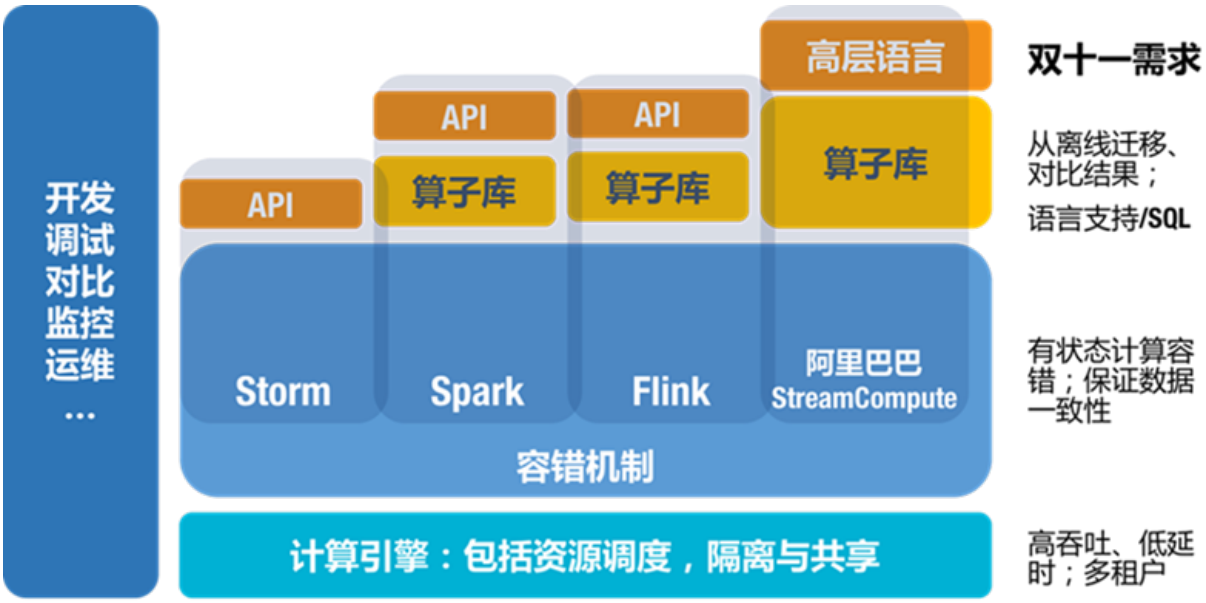
图 64: 开发成本



23.3 产品特点

相较于其他流计算产品，阿里云流计算起步较早，应用场景丰富，经过海量数据和多样性业务的打磨，系统功能和整体架构和提供一些极具竞争力的产品优势，用户可以充分利用阿里云流计算提供的产品优势，方便快捷的解决自身业务实时化大数据分析的问题。

图 65: 流计算技术栈及StreamCompute和竞品的关系



- **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，方便用户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。
  - 丰富多样的数据采集工具，涵盖从无结构化日志采集到结构化的数据库变更采集，从简单易懂的一键式拖拽上传数据到开放可定制化的编程 SDK，让用户随心所欲采集并上传业务流式数据。
  - 深度整合各类云数据存储，包括 MaxCompute、DataHub、Log Service、RDS、Table Store、AnalyticDB 等各类数据存储系统，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。
  - 林林总总的数据展现套件，覆盖各类数据化报表展示组件；同时针对流式计算特有翻牌器、实时大屏等场景提供定制化显示组件，用实时化的大数据助力业务发展。

- **性能优越**

  关键指标超越 Storm 的性能六到八倍，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数百万级别，单集群规模在数千台，计算能力线性扩展。

- **简易好用**

支持标准 SQL（产品名称为：StreamSQL），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 Storm 开发，让更多的BI人员、运营人员通过简单的 StreamSQL 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采集、数据开发、数据展现等不同阶段辅助套件，让实时数据开发不再高不可攀。

- **成本低廉**

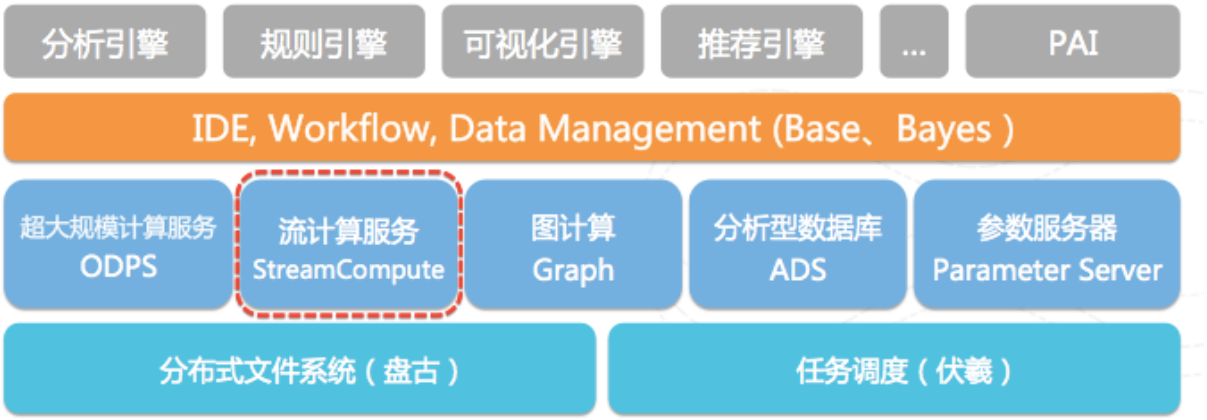
  大量优化的 SQL 执行引擎，会产生比手写原生 Storm 任务更高效且更廉价的计算任务，无论开发成本和运行成本，阿里云流计算均要远超开源流式框架。

23.4 阿里云流计算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+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 66: 流计算战略地位所示。

图 66: 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投入：

- 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。

23.5 产品架构

23.5.1 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，如图 *67: 业务架构*所示。

图 67: 业务架构

- **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，包括可以集成大数据计算的 DataHub、连接物联网信息的 IOTHub、和对接 ECS 日志的 Log Service。

- **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，包括关系型数据库、NoSQL 数据库、OLAP 系统等等。

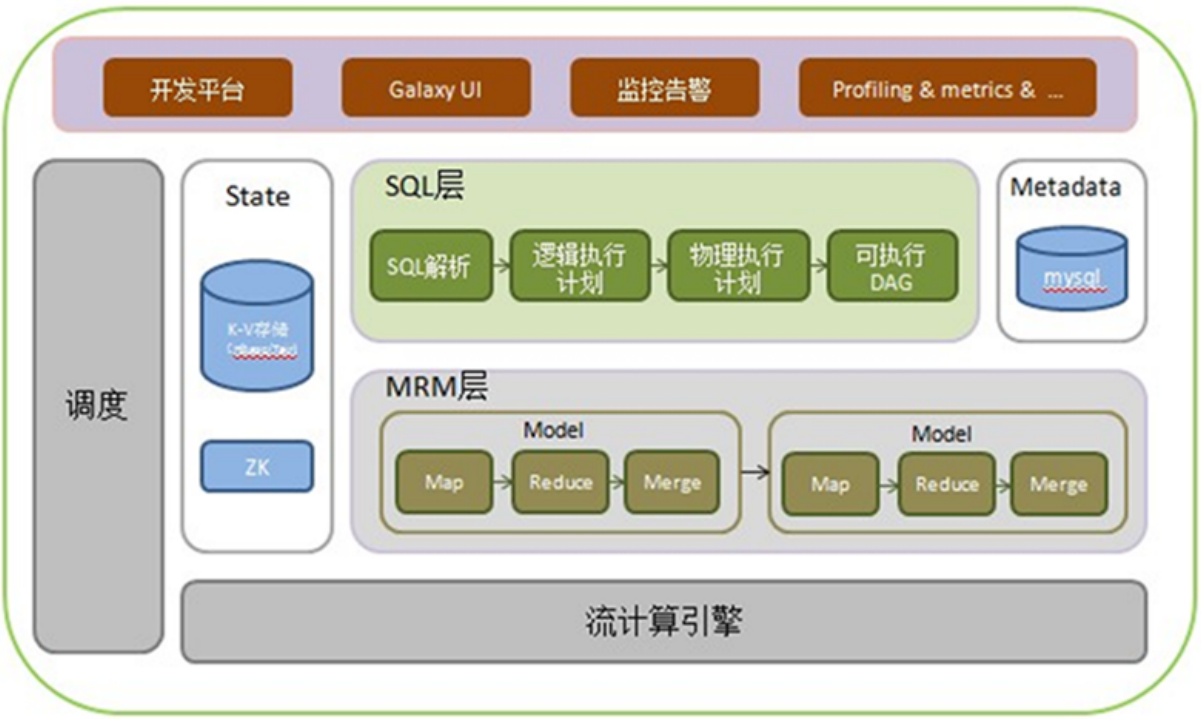
- **数据消费**

  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。

23.5.2 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，如图 *68:* 技术架构所示。现简要介绍下流计算的整个技术架构。

图 68: 技术架构



流计算主要由七部分组成：

- **用户接口层**：主要提供了开发平台，便于用户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，同时用户可以通过 Galaxy UI、Profiling 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；
- **SQL**：该层主要负责 Galaxy 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；
- **MRM**：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 batch 来进行处理，每 个batch 内会有多条数据记录）；

  - Merge：将该 batch 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 batch 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）
- **流计算引擎**：目前流计算是构建在底层的流计算引擎之上，MRM 将上述的有向图转化成相应的 Topology，然后进行数据的处理与计算，但其对流计算引擎只是弱依赖，在未来会支持更多的计算引擎。
- **Metadata**：对于提交的每个流计算作业，都会在 Metadata 系统中有相应的元数据，从而便于作业的管理；
- **State**：在介绍 MRM 的时候有提到过，这个是用来持久化流计算处理的中间结果状态的系统，通过 State 的存储流计算可以实现较完善的容错，保证在各种异常情况下数据的精确性；
- **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。

24 大数据应用加速器

24.1 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，从阿里自身大数据的应用发展来看可以看到几方面的挑战：

- 一方面为了应对数据量高速的增长，衍生出各类的分布式数据计算与存储技术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就可以支撑整个企业的数据分析报表问题；各类数据的积累如何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；
- 另一方面，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；
- 并且，当下对运用数据的用户也不只是局限在专业的数据分析师、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。

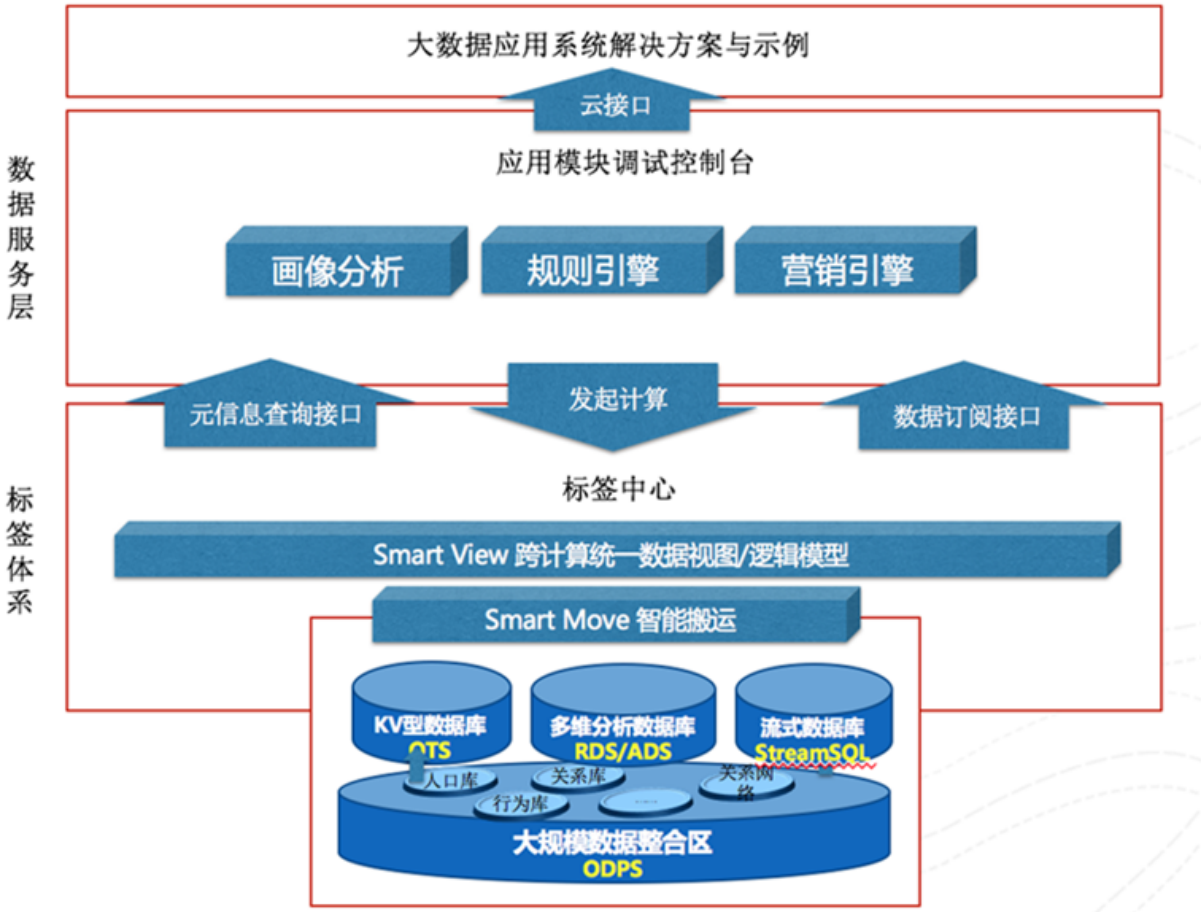
在这三方面之下，如果想要运用好大数据就对企业的 IT 架构、技术人员综合能力的要求提出了更高的挑战，既要能了解各个专业分布式计算和存储资源的特性，又要求能将这些资源针对数据分析、算法服务等多种应用场景进行合理的架构，还要能够针对业务人员使用数据的场景足够了解并告诉相应，制作出面向业务的数据产品。

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。

24.2 产品概述

DTBoostv2.0产品组件如图 *69: DTBoost v2.0产品组件*所示。

图 69: DTBoost v2.0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。

这种方式的好处在于：

- 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透出也有助于 IT 部门对数据使用的管理，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对接到产品界面端上，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
- 整合分析
  - 分析服务
  - 界面配置

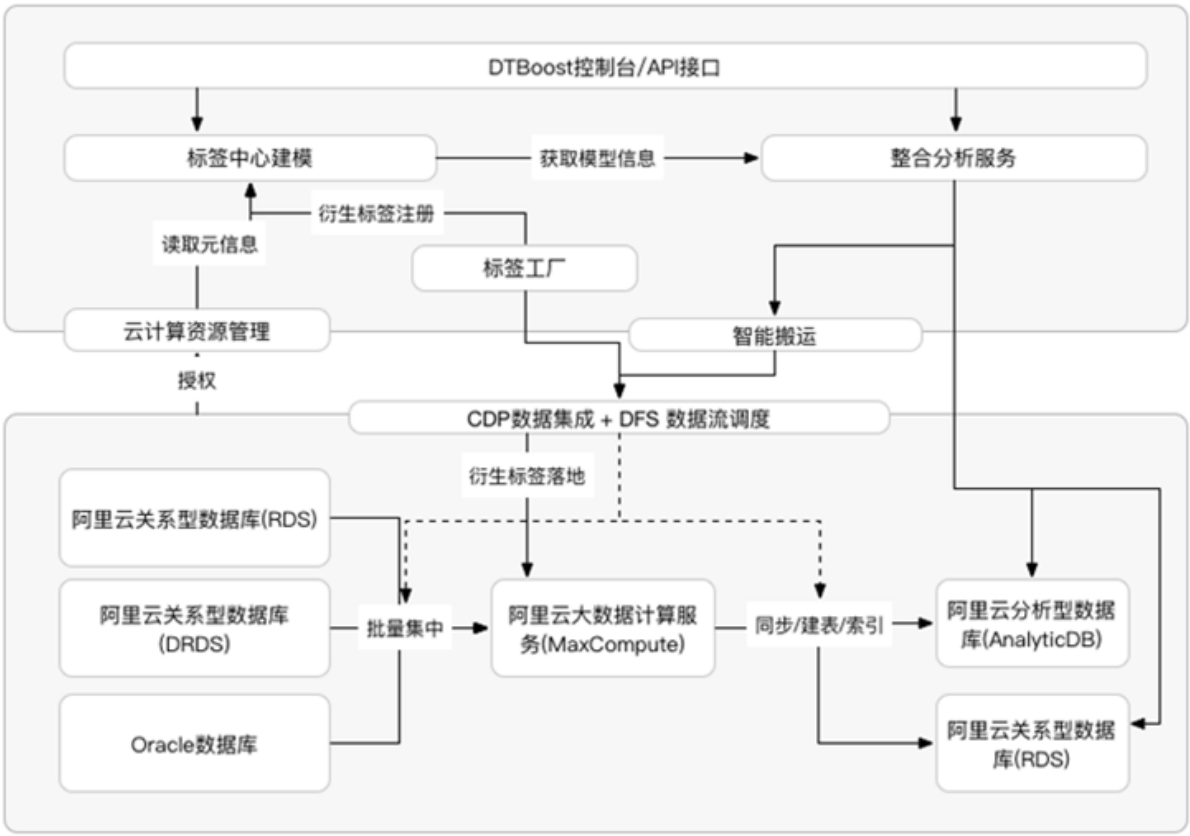
24.3 架构总览

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。

下面以其中以 DTBoost 数据服务模块当中整合分析作为案例来解释总体的架构。

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 MaxCompute 当中进行集中，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）当中。

图 70: 标签中心技术架构



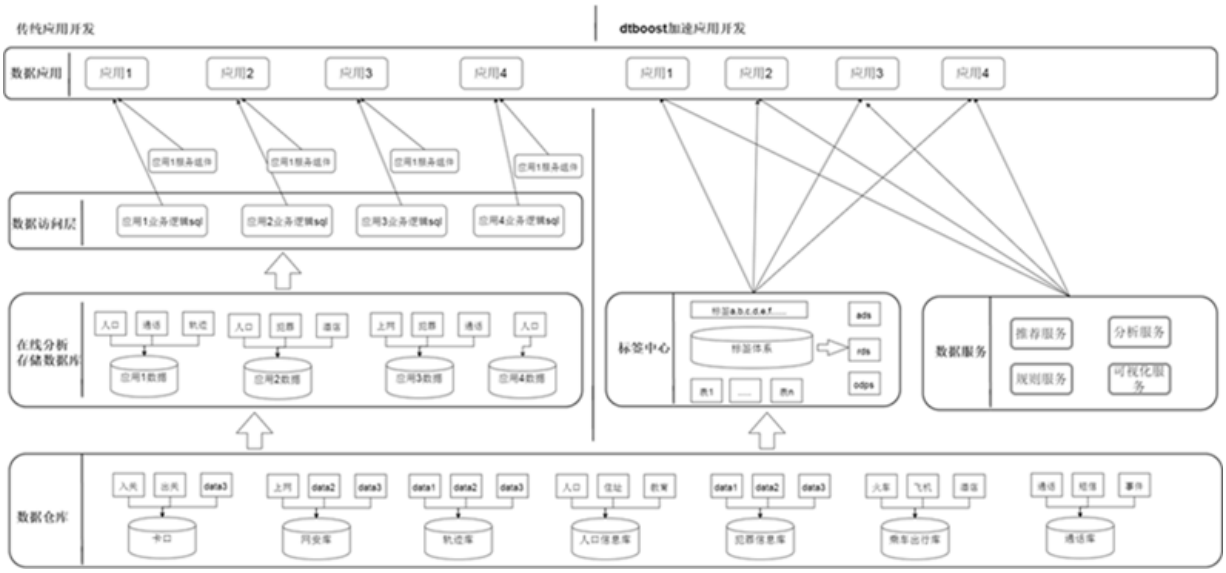
如图 *70:* 标签中心技术架构所示，用户从 DTBoost 控制台或 API 进入，通过把自己的云计算资源授权给 DTBoost 后，就可以通过 DTBoost 读取各个云计算资源中的数据元信息。经过建模配置后，在相关的数据服务模块中可以进行手工/自动触发标签中心的智能搬运模块，通过把相关的数据同步调度任务发送给数据流服务 DFS（Data Flow Service）和数据整合 CDP，来对所需要整合的数据以标签粒度来进行业务库到离线数据仓库的批量大集中，以及到在线分析数据库的同步、建表、索引工作。在数据准备完成之后，就可以通过相关的数据服务 API 接口或者在控制台上基于标签模型视图之上进行相关的计算。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当中（MaxCompute）。

整个过程可以看做 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。

24.4 场景概要

用户除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 71: 加速开发流程



如图 *71: 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。

24.5 功能模块

24.5.1 标签中心

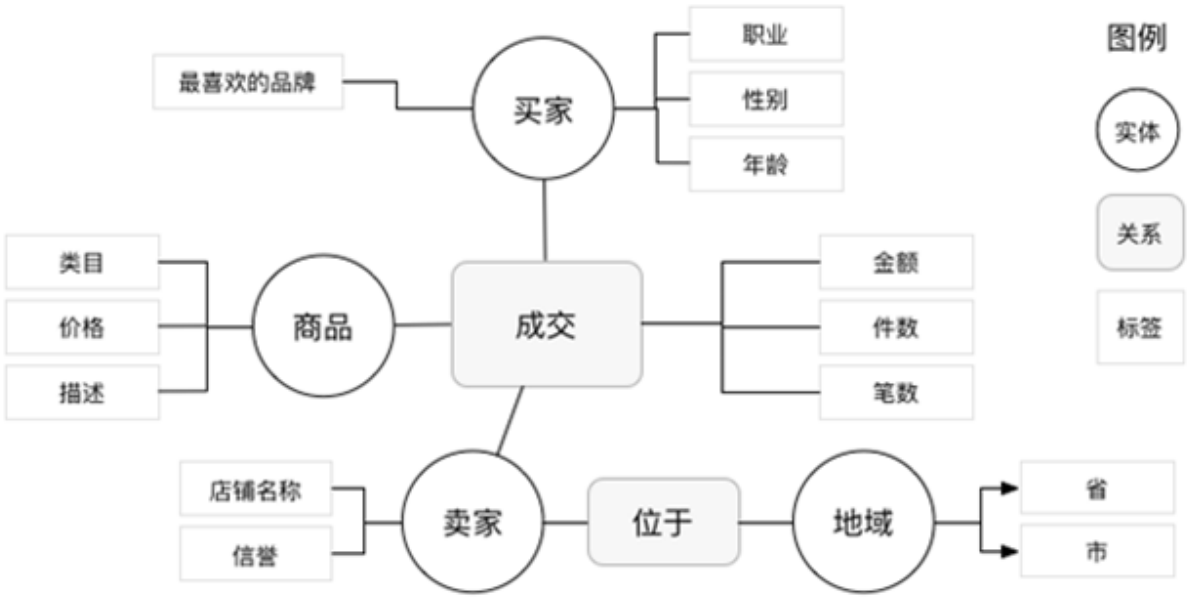
24.5.1.1 概念说明

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，直接让用户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，标签体系就是围绕多个实体对象，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-位于等等，建立标签化描述的方法。

标签建模如图 72: 标签建模所示。

图 72: 标签建模



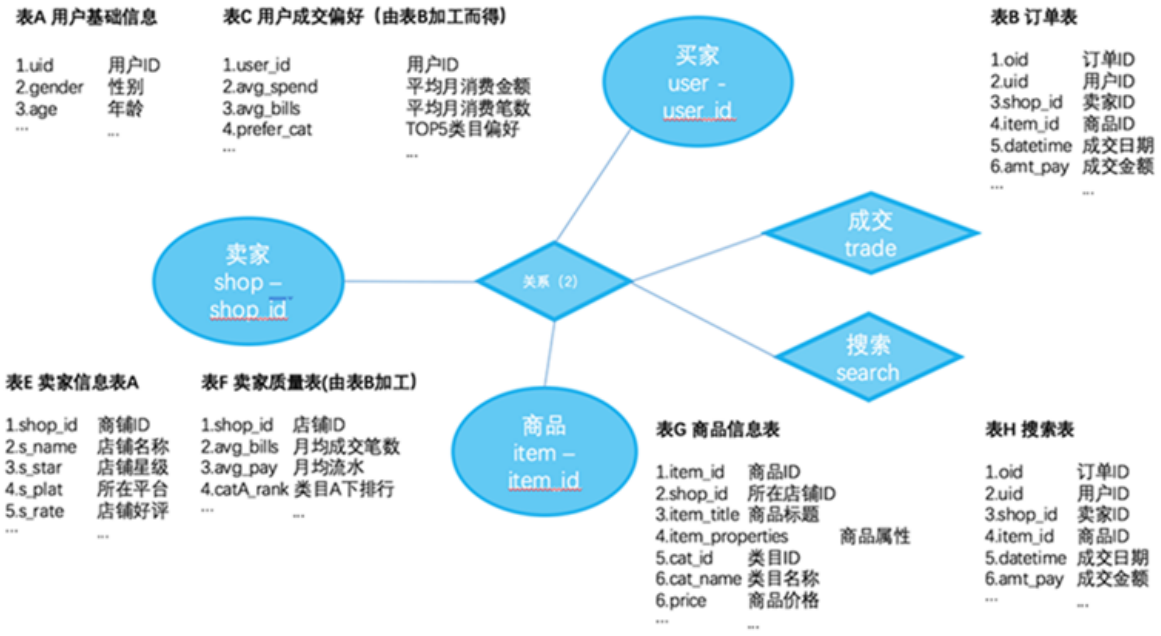
这种建模方式看起来可能类似于角模型（Anchor）或者是图模型（Graph），其实并不然。传统的建模过程是根据业务需求设计概念和逻辑模型，再根据逻辑模型对物理数据表进行加工和规整。而标签建模是在已有的物理数据/模型之上直接建立逻辑模型，通过各个数据服务的代理解析，让用户可以在视图上直接进行各类的计算，不需要预先对物理数据进行大规模的加工处理，即用即算。

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v2.0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 73: 实体关系建模所示。

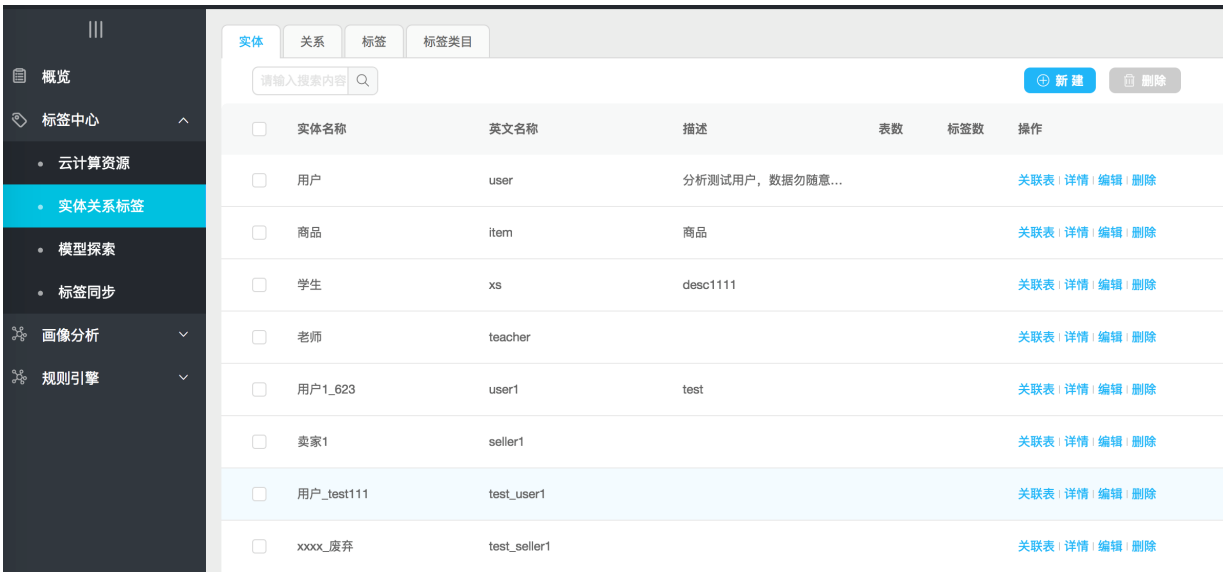
图 73: 实体关系建模



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 74: 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 74: 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，包括模型管理、云计算资源管理和模型探索。

24.5.1.2 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的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

24.5.1.3 功能组件

24.5.1.3.1 云计算资源管理

云计算资源管理就是支撑与多个计算存储资源通信，与元信息获取的基本功能模块。

目前 DTBoost 支持与以下计算存储资源的管理：

- 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）
- 阿里云数据中枢（DataHub）
- 阿里云流式计算（StreamCompute）

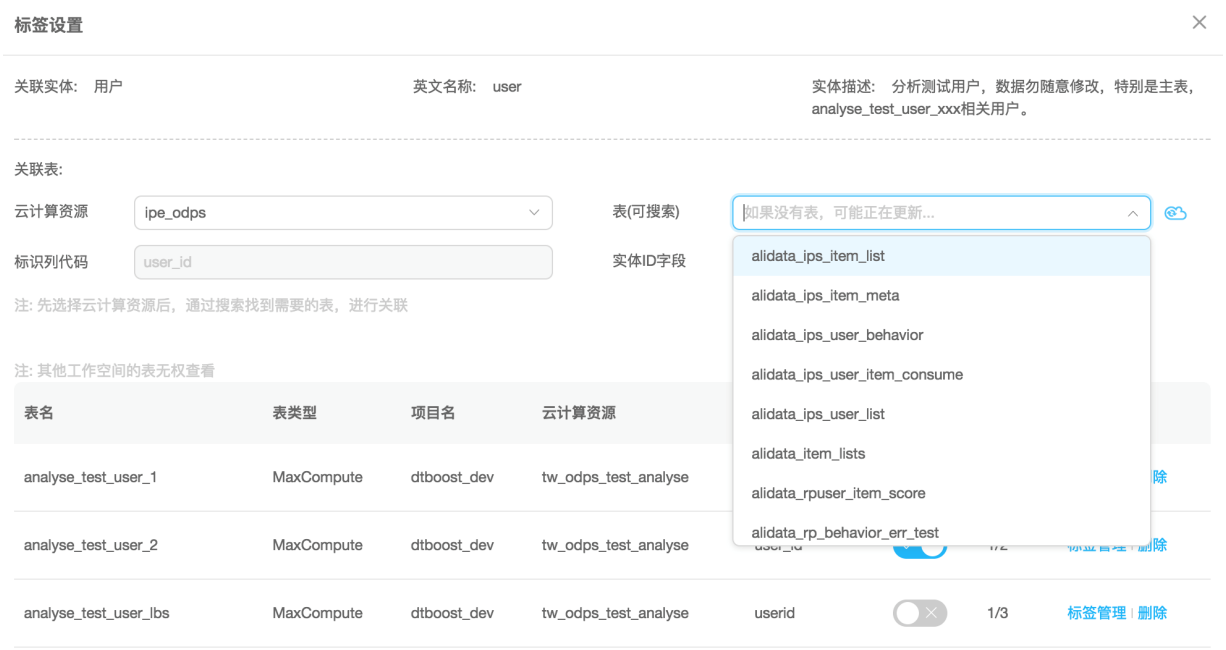
24.5.1.3.2 模型管理

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 *75: 实体关系建模示意*所示。

图 75: 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 *76: 关系定义*所示。

图 76: 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 *77: 查看实体详情*和图 *78: 设置标签*所示。

图 77: 查看实体详情

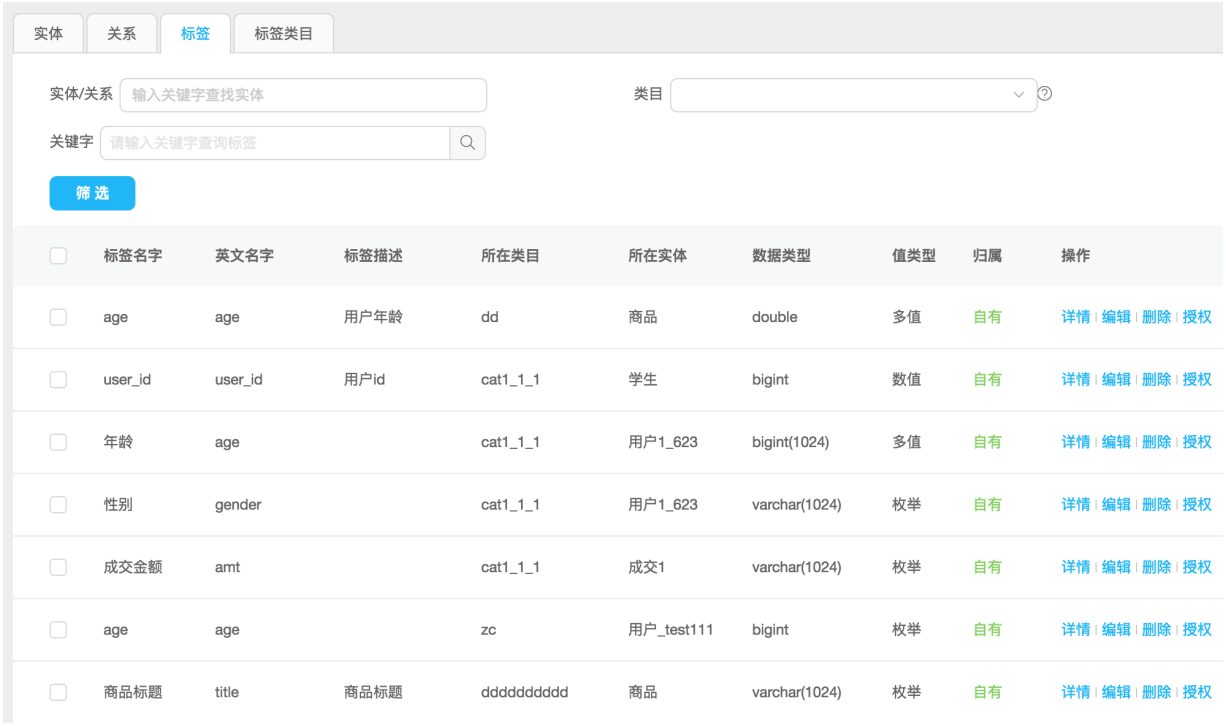
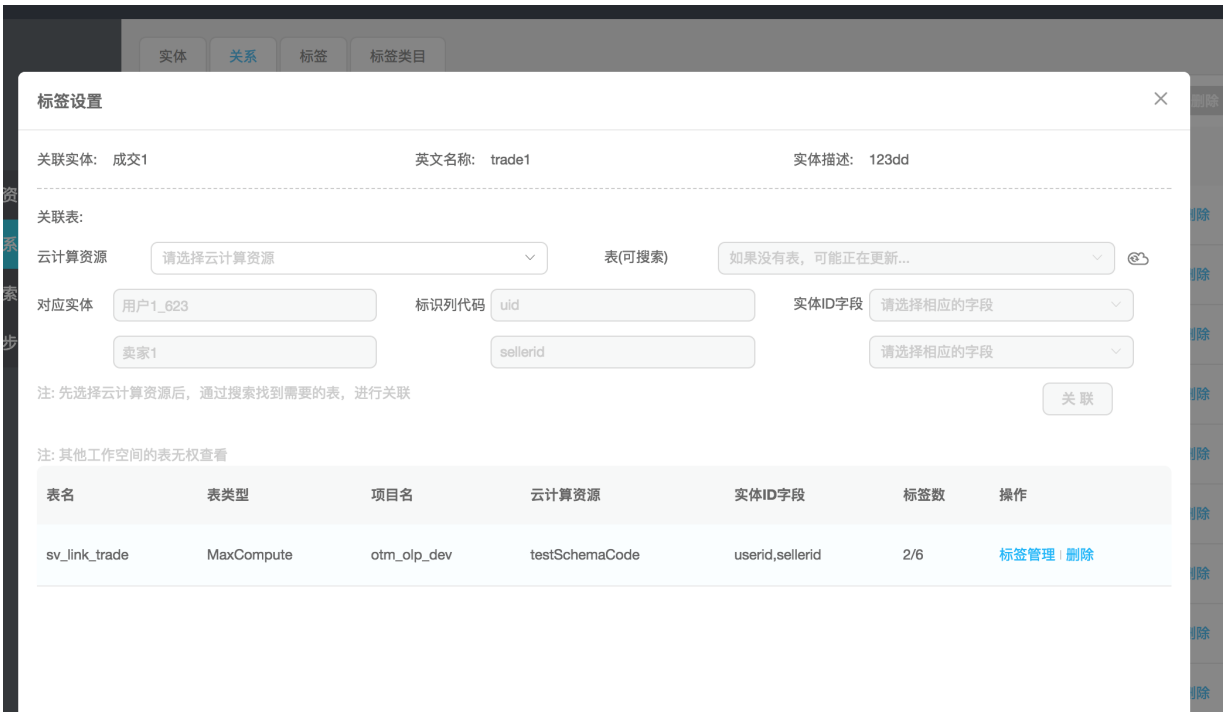


图 78: 设置标签

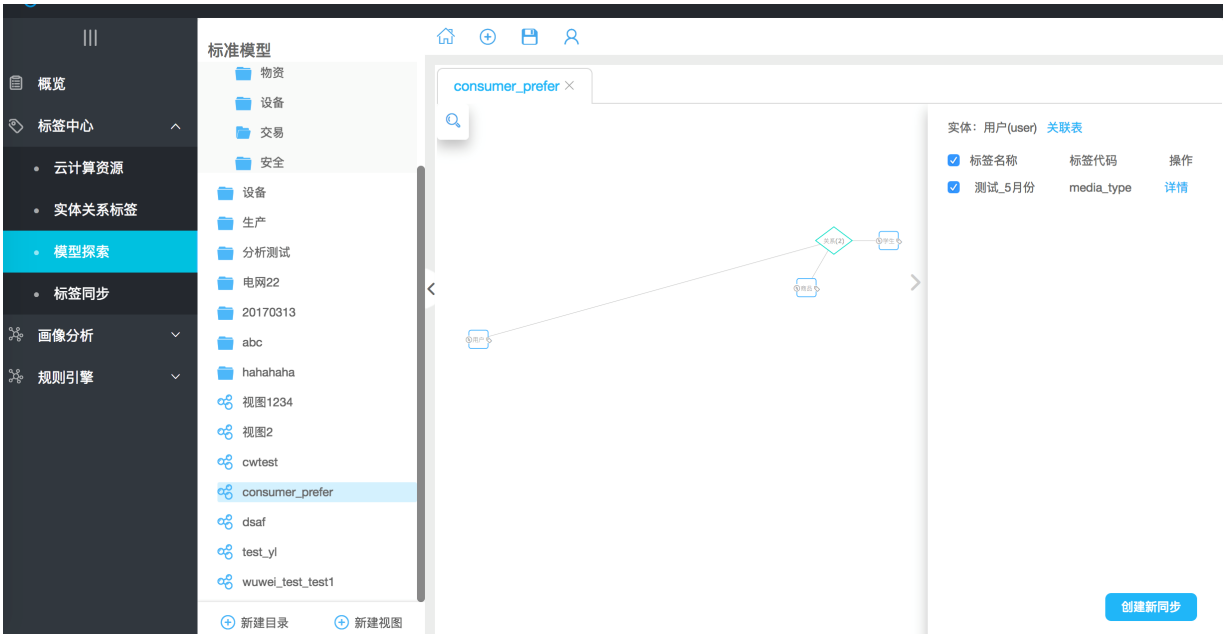


24.5.1.3.3 模型探索与数字订阅

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 79: 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。

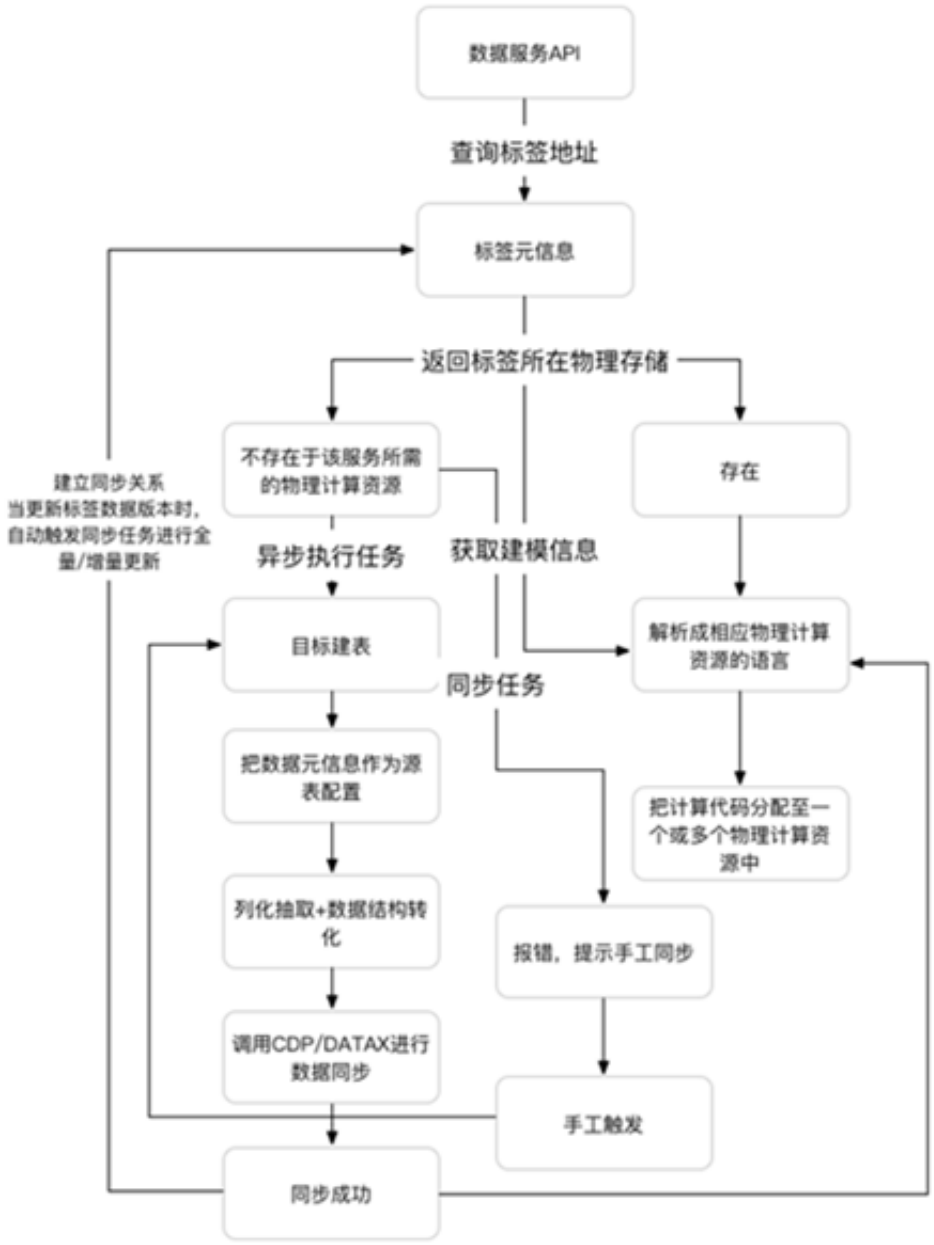
图 79: 实体关系模型探索



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 80: 标签中心技术架构所示。

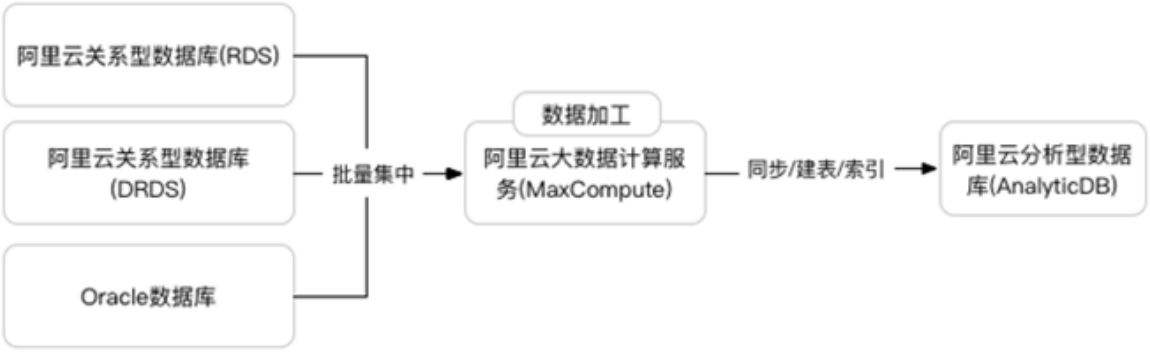
图 80: 标签中心技术架构



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 *81:* 批量大数据在线分析所示

图 81: 批量大数据在线分析

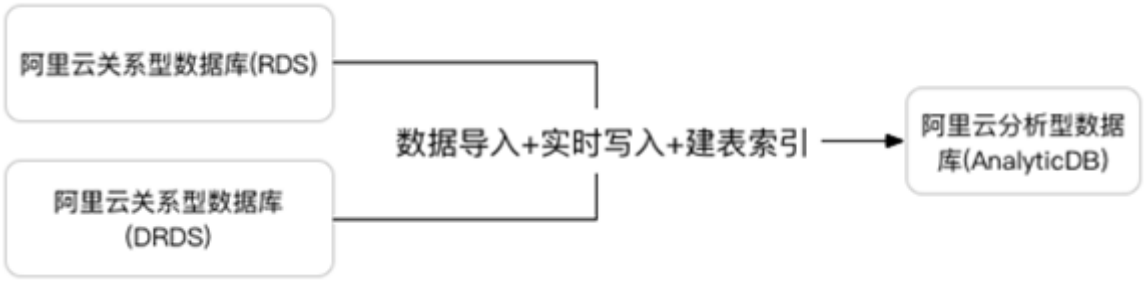


- 批量数据算法计算在线查询，如图 *82:* 批量数据算法计算在线查询所示

图 82: 批量数据算法计算在线查询

- 实时大数据在线分析，如图 *83:* 实时大数据在线分析所示

图 83: 实时大数据在线分析



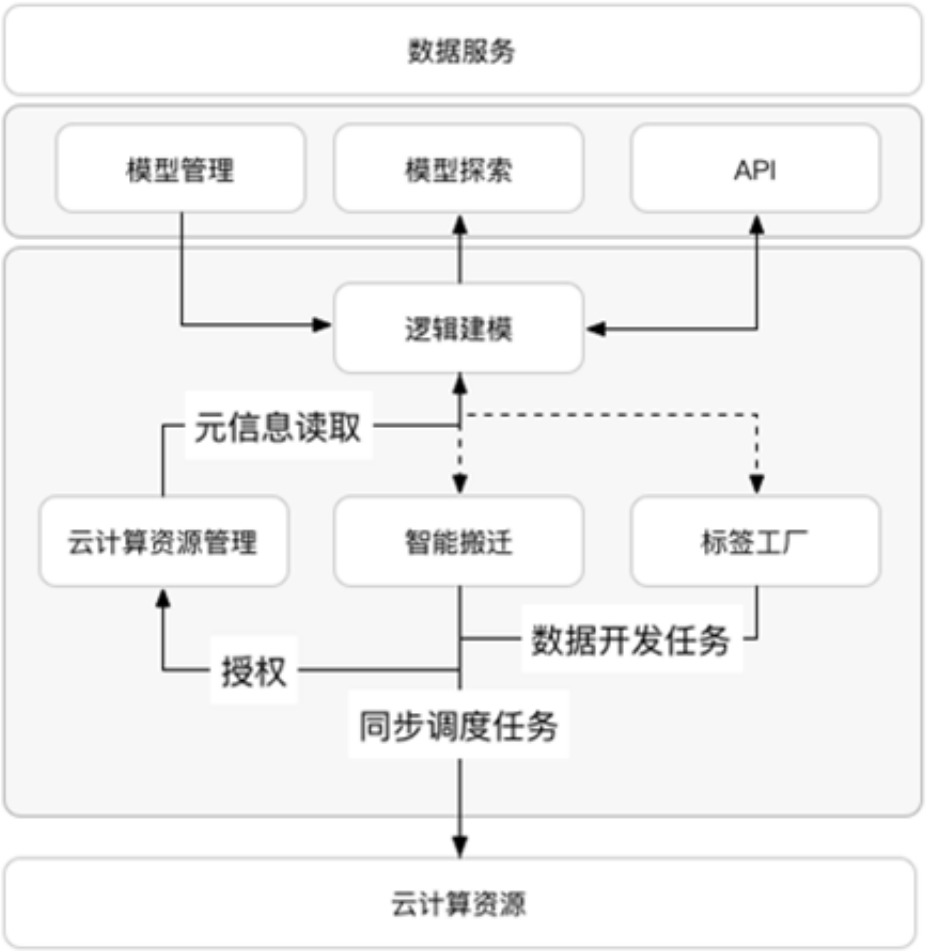
  - 对于整合分析这类 OLAP/ADHOC 场景来说，提供了将 Oracle，关系型数据库（MySQL）等业务库中的数据同步至大数据计算（MaxCompute）中，再订阅到所使用的分析库中，如阿里云分析型数据库（AnalyticDB）、关系型数据库（RDS）等。

- 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。

24.5.1.4 技术架构

技术架构如图 *84:* 标签中心组件架构所示。

图 84: 标签中心组件架构



24.5.1.5 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几个问题：

1. 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；

2. 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。
3. 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。
4. 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。

由此，标签体系建模和传统BI分析建模有几大特性：

- 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

  呈上，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。

24.5.2 整合分析

24.5.2.1 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里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 85: 整合分析

如图 *85:* 整合分析所示，整合分析作为 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如下几个特性：

- 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，应用开发当中既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；还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当中；也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当中。DTBoost 整合分析模块就是面向这类场景为您提供以下功能，帮助您加速应用开发的难度。

24.5.2.2 功能组件

整合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。

24.5.2.2.1 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。

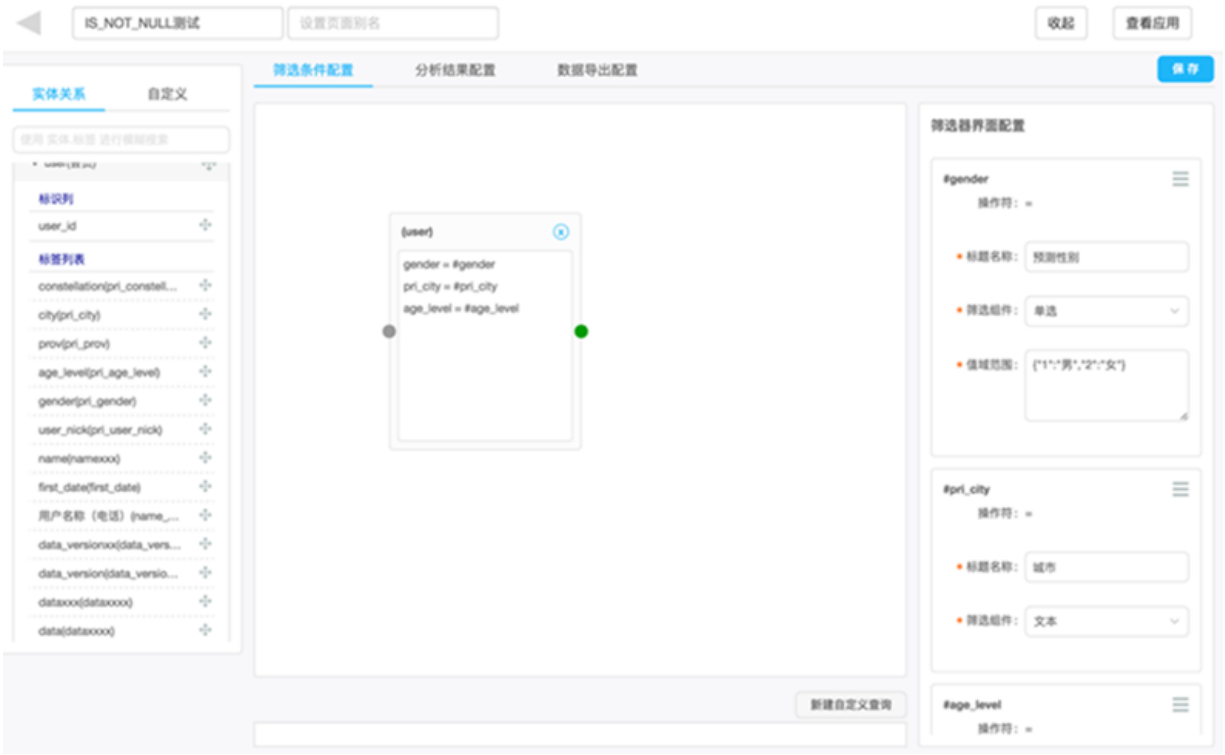
您也可以对接口进行调试，查看查询的结果/执行错误、语法每一步解析所耗费的时间以及所解析的真实 SQL 语句，来帮助您调试分析接口。

24.5.2.2.2 界面配置

如图 *86:* 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

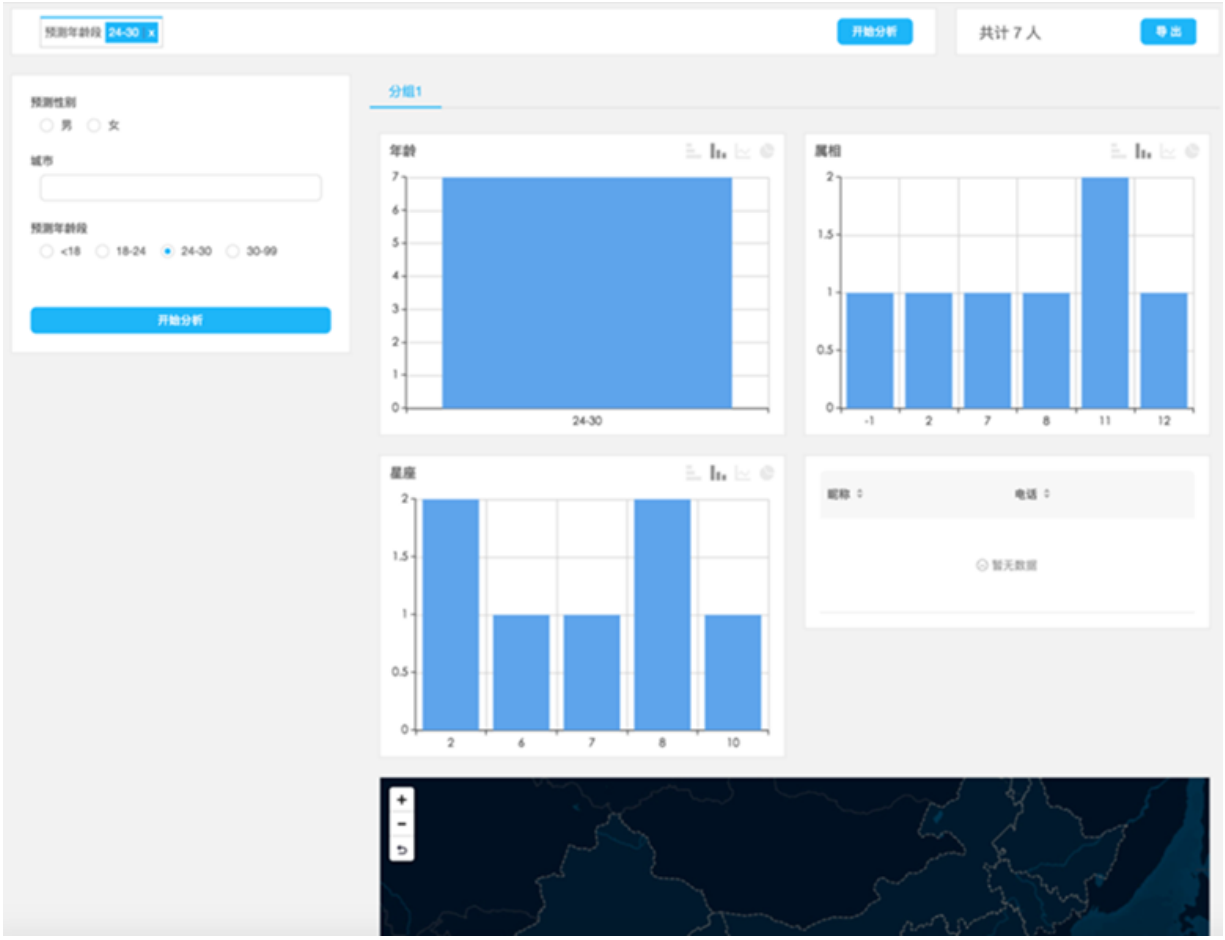
他系统当中，如结合广告投放系统进行精准营销。整个界面的代码是完全开放的，可以无缝与您的现有系统进行整合。

图 86: 筛选条件配置



如图 *87: 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，用户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。用户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 87: 分析应用



24.5.2.3 典型应用

24.5.2.3.1 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。

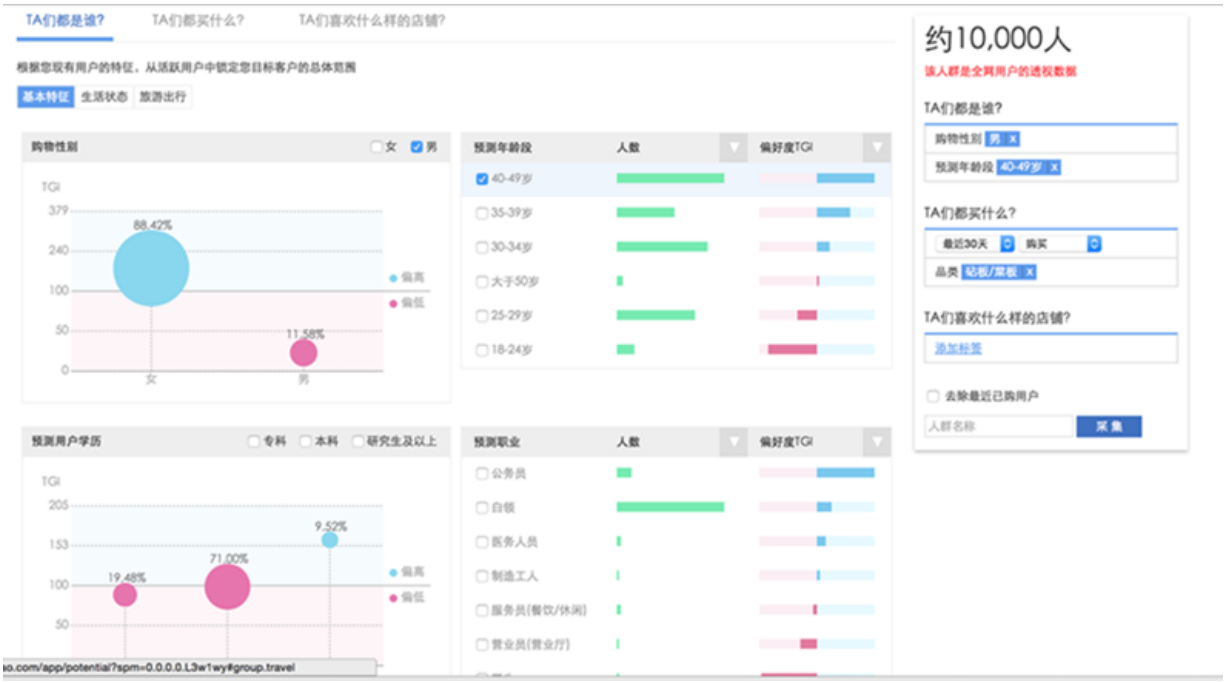
分析人员在进行分析的时候，可能多种行为与多种用户属性之间的组合筛选变化多样，无法按传统数据分析建模方法提前预算好所有组合。

例如，用户根据消费者行为选取想要分析的目标群体，如图 *88:* 选择目标群体所示。

图 88: 选择目标群体

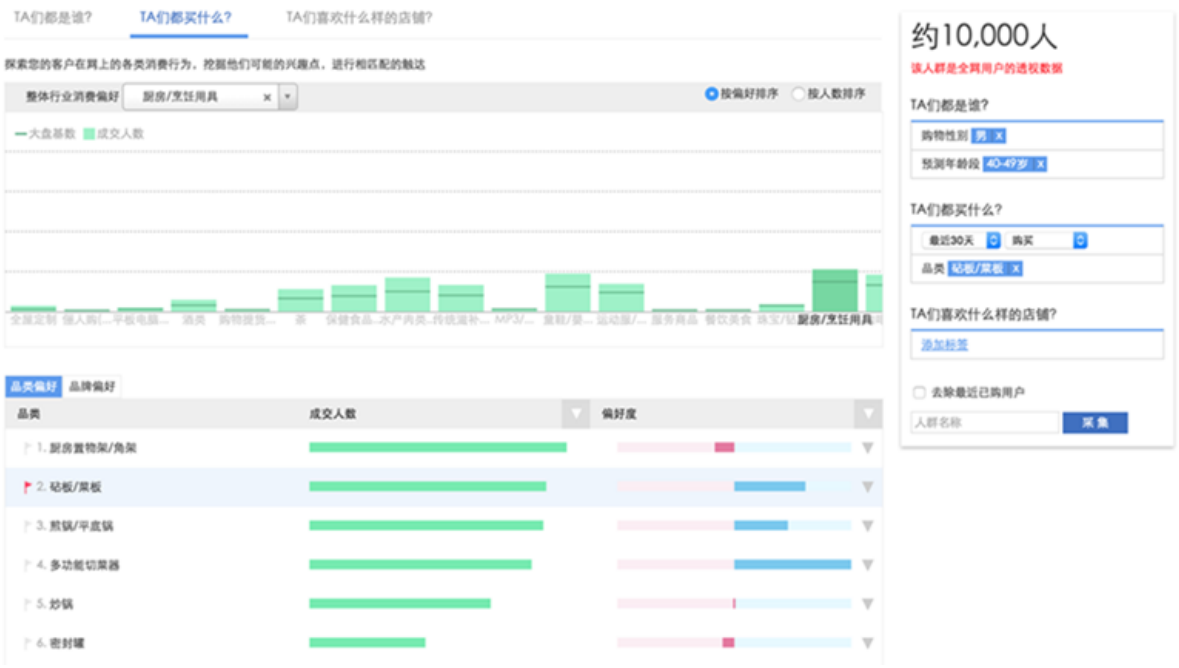
对于选取的目标群体，分析其各项特征，发现分布密集的特征，如图 *89:* 分析各项特征所示。

图 89: 分析各项特征



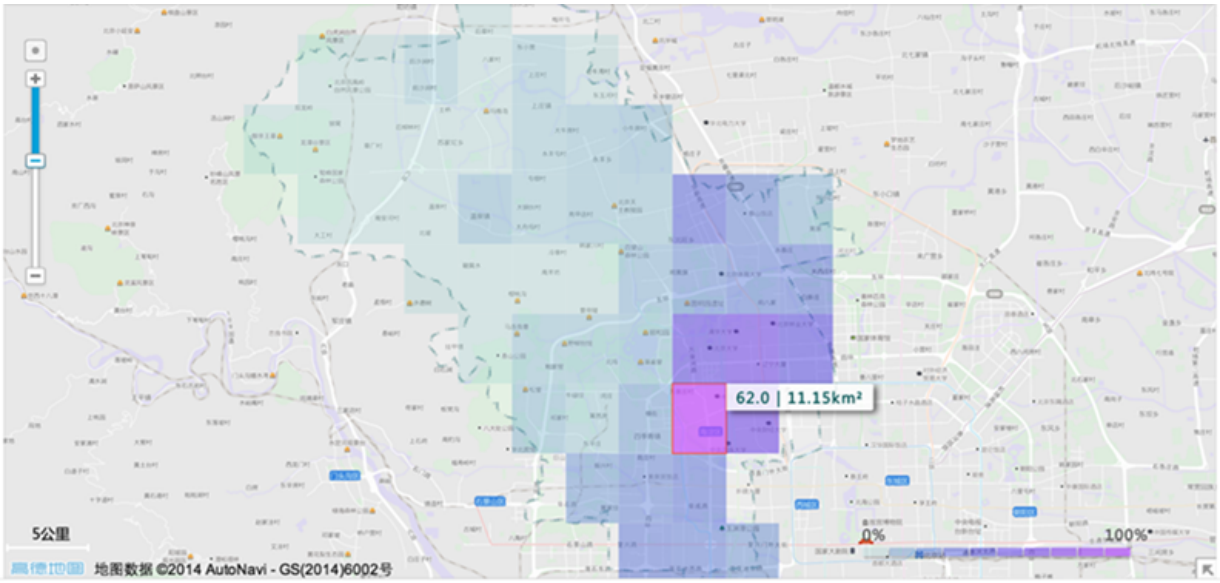
再根据这些典型特征，找到还未购买的群体，如图 *90:* 找到未购买的群体所示。

图 90: 找到未购买的群体



针对这类群体，对接下游系统，如图 *91: 对接下游系统*所示。

图 91: 对接下游系统



24.5.2.3.2 设备全履历

例如，设备全履历如图 *92: 电压等级*、图 *93: 地级供电公司*和图 *94: 设备明细*所示。

图 92: 电压等级

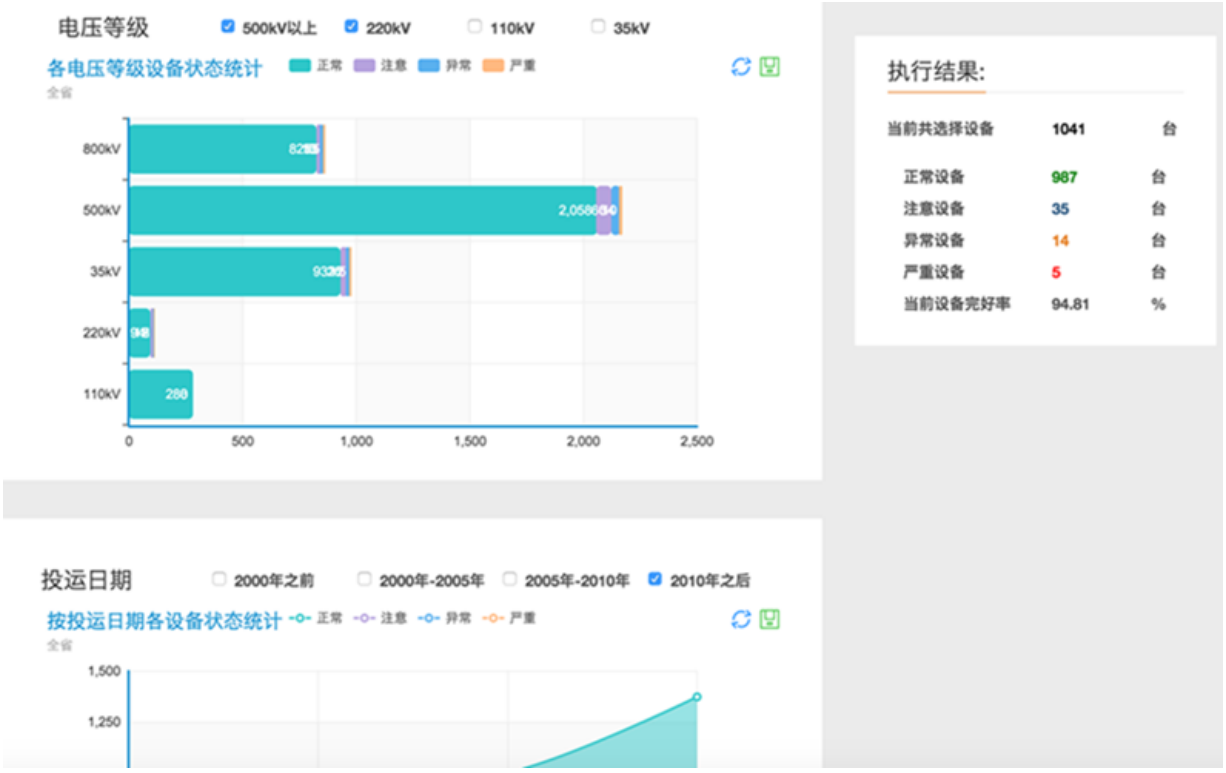


图 93: 地级供电公司

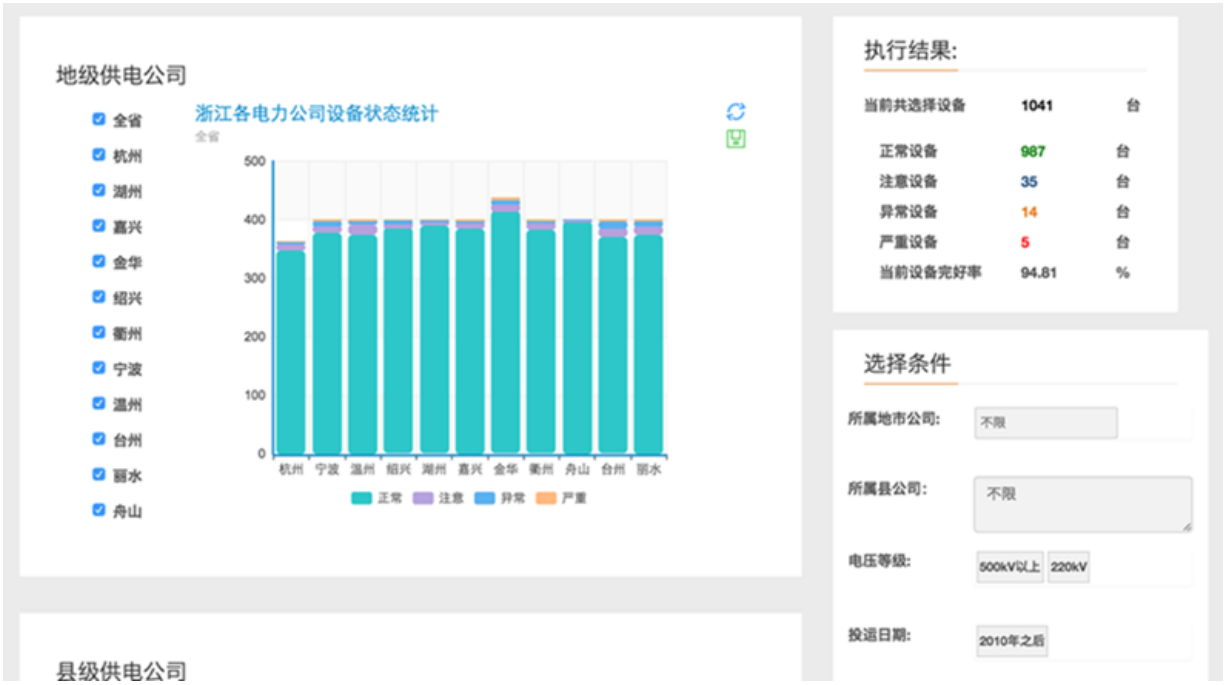
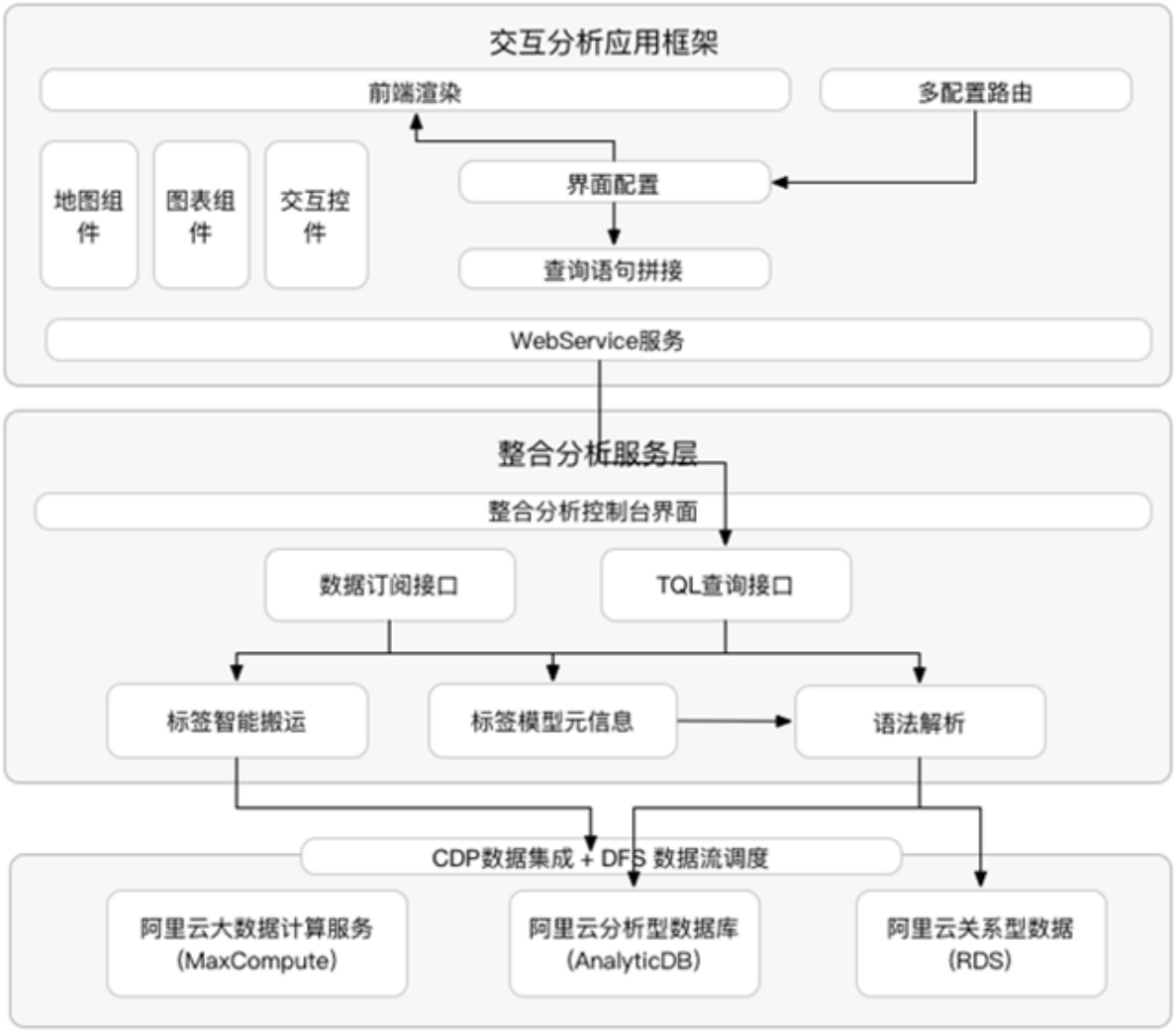


图 94: 设备明细

24.5.2.4 技术架构

整合分析的技术架构如图 *95:* 整合分析技术架构所示。

图 95: 整合分析技术架构



从技术架构来讲分为两个部分，在服务层部分通过 TQL 查询接口接收用户输入的参数，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用户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渲染相应界面同时翻译为相应的查询的接口。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。

24.5.2.5 产品特性

DTBoost 具备如下特性：

- 一键数据整合

  针对不同的分析主体，您可一键完成分布在多个存储资源当中的多个标签到在线查询数据库当中的同步、索引工作，像管理一张表一样的管理不同数据源。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 应用开发者者直接通过与整合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，即可以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 相似的的 TQL（Tag Query Laugage）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，使得整合分析能够与 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，可以整合入您整体的分析系统当中，更容易让用户灵活的洞察，对实体的属性、行为、地理出行等进行灵活的分析。

24.5.3 规则引擎

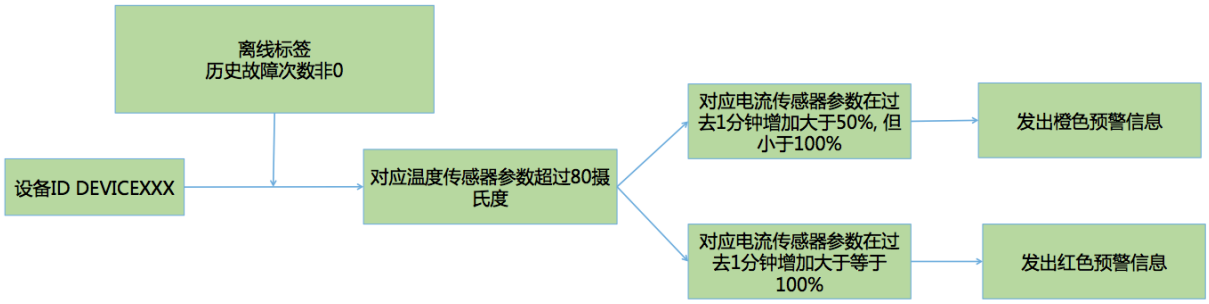
24.5.3.1 概念说明

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 *96:* 规则结构展示所示。

图 96: 规则结构展示



规则通常来说为一个 if...then... 结构，即条件与行为构成了一个完整的规则。在上面的例子里，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。同事，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。

24.5.3.2 适用场景

24.5.3.2.1 自定义规则实现电机设备异常预警

- 问题：

  普通检修人员经验不足，难以根据电机设备读数判断是否出现异常，需要逐设备开箱逐部件检测，耗费人力巨大。

- 需求：

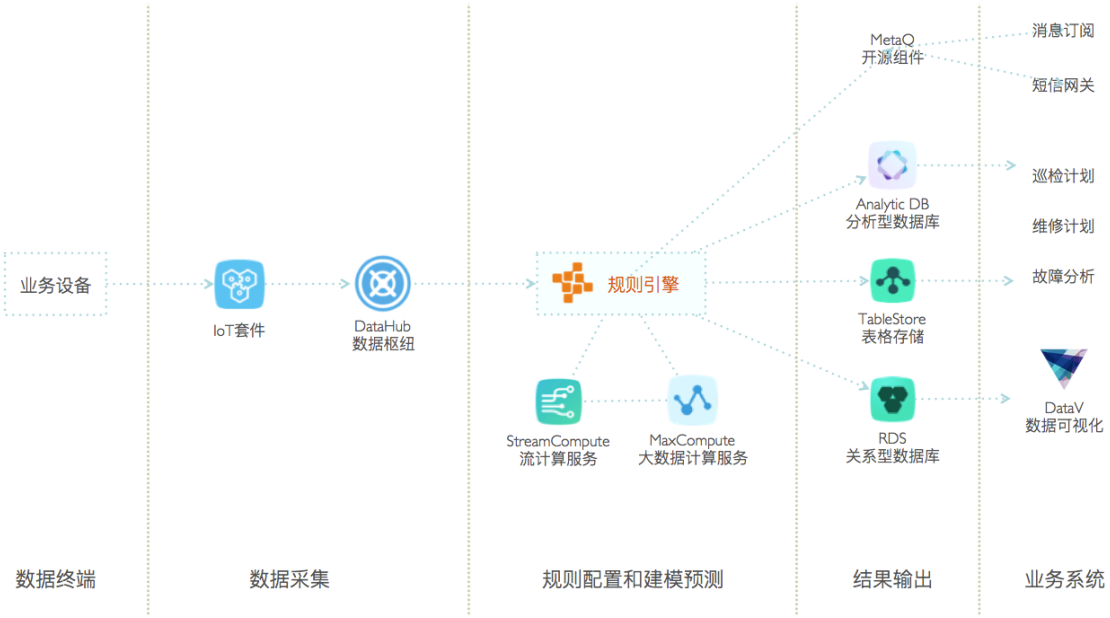
  凭借电机设备读数实时了解设备异常概率，以决定是否开箱检测，减少人力资源投入。

- 异常判断考虑的数据维度：

  设备型号、运行时长、维修次数、当地温度等。

- 解决方案：

图 97: 工业设备异常检测解决方案架构



24.5.3.2.2 智能规则实现交易异常监控

- 问题：

  恶意差评、炒信、欺诈、套保、商家卷款跑路……

  电商领域异常行为举不胜举。

- 需求：

提前识别新兴恶意行为模式，不做事后诸葛。

实时触发应对措施，有效止损。

防止恶意用户测试和规避规则。

- 算法考虑的数据维度：

  用户某一时间段的下单频率。

  某一时间段的规则赔付频率。

  某一时间段的积分使用的量。

- 解决方案：

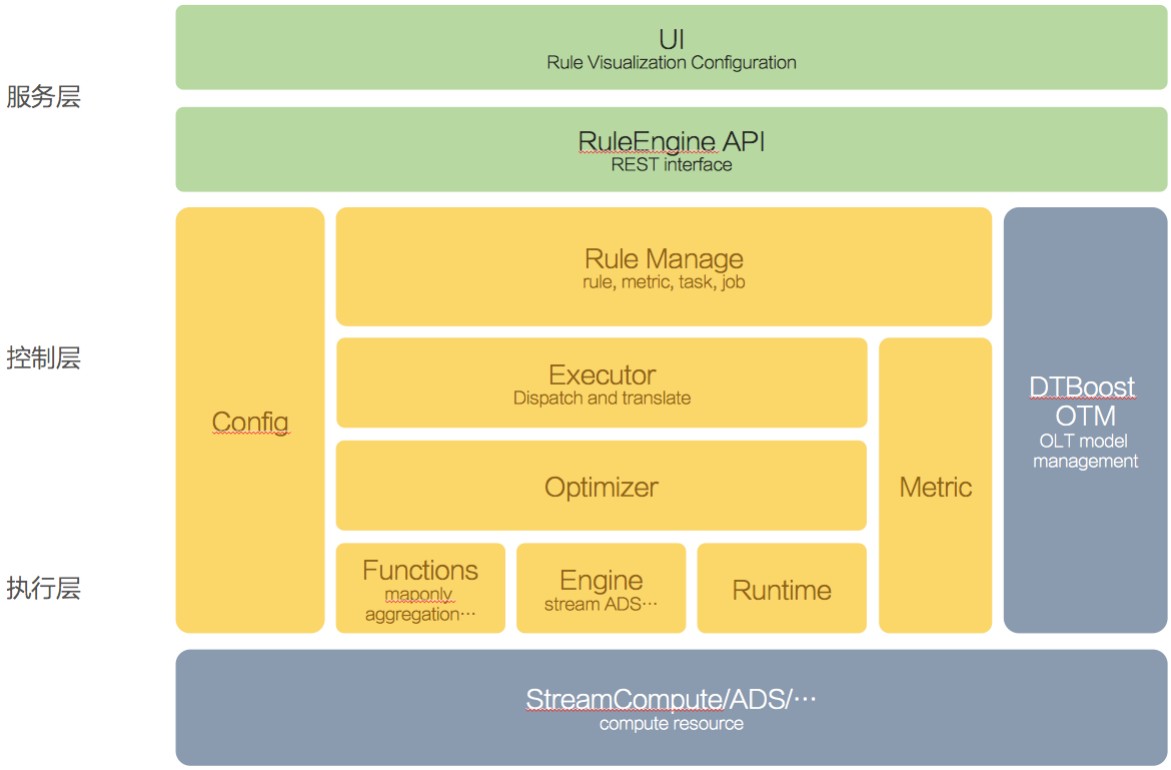
图 98: 电商恶意行为检测解决方案架构

24.5.3.3 技术架构

24.5.3.3.1 功能模块

规则引擎的功能模块拆解，如图 *99:* 规则引擎功能模块所示。

图 99: 规则引擎功能模块



规则引擎整体分为三层：

- 服务层
- 控制层
- 执行层

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。用户可以通过 UI 访问规则引擎，也可通过 API 完成系统间的调用。同时，如果用户有深度定制的需求，规则引擎的 API 模块提供了完备的功能供用户基于不同行业需求进行二次开发。

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。

执行层包括了规则实际提交到计算引擎的 engine，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。

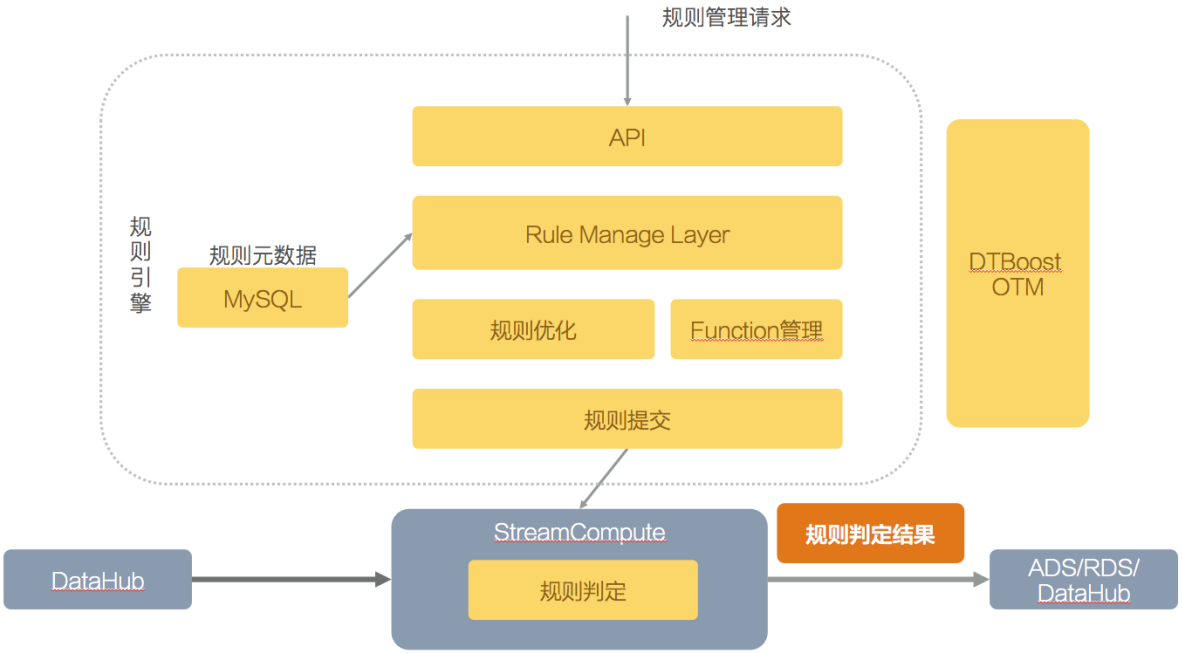
最底层的灰色模块为外部的计算引擎。

侧面灰色的 OTM 模块为 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。

24.5.3.3.2 规则提交流程

规则引擎中规则的提交执行流程，如图 *100: 规则提交流程*所示。

图 100: 规则提交流程



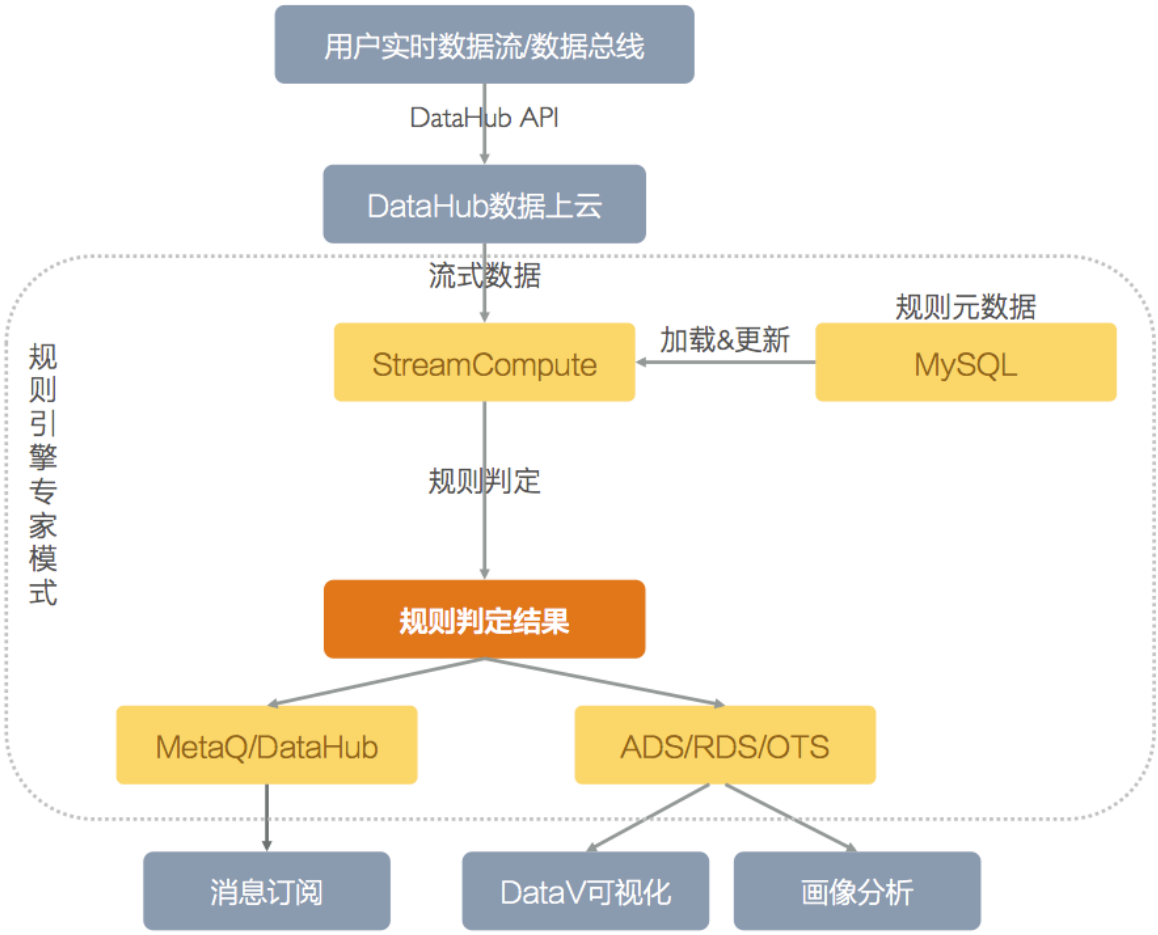
规则通过管理模块进行元数据注册，然后进行翻译，优化，并转化为对应 function 逻辑。同时，根据规则配置的计算引擎类型，提交到对应的计算引擎，作为计算引擎中的作业进行执行。

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。

24.5.3.3.3 规则运行流程

规则运行流程，如图 *101: 规则数据流*所示。

图 101: 规则数据流



24.5.3.4 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，规则所对应数据流可达到上万条每秒的数据量，同时可以确保99.99%的事件从发生到规则产出全流程不超过5秒。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是按照专家经验所配置，会有按照实际执行结果进行微调，或者随着实际数据变化进行对应调整的需求。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。

25 Quick BI

25.1 产品概述

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。

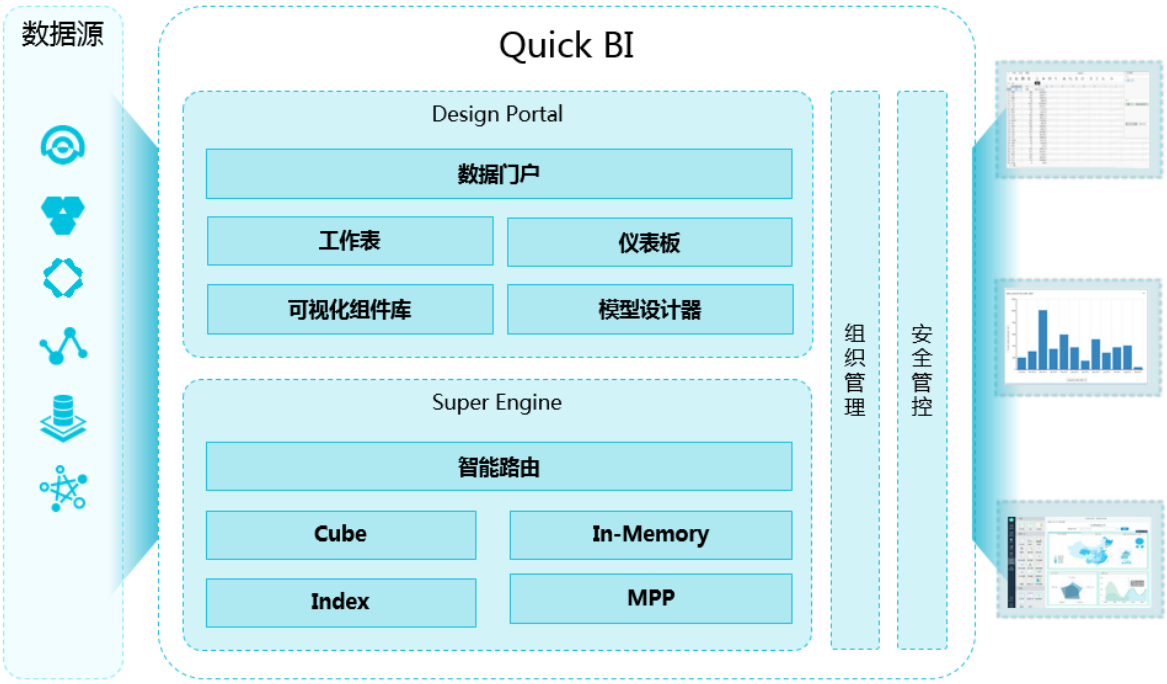
Quick BI支持众多种类的数据源，既可以连接MaxCompute（ODPS）、RDS、AnalyticDB、HybridDB（Greenplum）等云数据源，也支持连接ECS上您自有的MySQL数据库。Quick BI可以为您提供海量数据实时在线分析服务，通过提供智能化的数据建模工具，极大降低了数据的获取成本和使用门槛，通过支持拖拽式操作和提供丰富的可视化图表控件，帮助您轻松自如地完成数据透视分析、自助取数、业务数据探查、报表制作和搭建数据门户等工作。

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。

25.2 产品架构

Quick BI产品架构如下图所示。

图 102: 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，包括但不限于MaxCompute、RDS（MySQL、PostgreSQL、SQL Server）、AnalyticDB、HybridDB（MySQL、PostgreSQL）等，封装数据源的元数据/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，允许用户使用当前数据源的SQL语法对维度和度量进行二次加工。

- **工作表**

  负责在线电子表格（webexcel）的相关操作功能，涵盖行列筛选、普通/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、表格处理等丰富功能。

- **仪表板**

  负责将可视化图表控件拖拽式组装为仪表板，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和文本框4种基本控件，支持图表间数据联动效果。

- **数据门户**

  负责将仪表板拖拽式组装为数据门户，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-工作空间> 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，实现不同人看不同报表。

- **行级权限管理**

  负责数据的行级粒度权限管控，实现不同人看同一张报表展现不同数据。

- **转让/分享/公开**

支持将工作表、仪表板、数据门户转让或分享给其他登录用户访问，支持将仪表板公开到互联网供非登录用户访问。

25.3 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，包括但不限于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17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，智能推荐合适可视化方案。

分析

多维数据分析。基于Web页面的工作环境，拖拽式、类似于Excel的操作方式，一键导入、实时分析，可以灵活切换数据分析的视角，无需重新建模。

快速搭建数据门户

拖拽式操作、强大的数据建模、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，大大提高分析效率。

安全管控数据权限

内置组织成员管理，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人看到不同数据的需求。

25.4 产品优势

Quick BI的总体优势可总结为多，块，强大和易用。

多

支持RDS、MaxCompute、AnalyticDB等多种数据源。

快

亿级数据秒级响应。

强大

内置完整的电子表格工具，可以让您轻松完成复杂的中国式报表的制作。

易用

丰富的数据可视化功能，自动识别数据特征，自动智能为您生成最合适的图表。

26 Quick BI

26.1 产品概述

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。

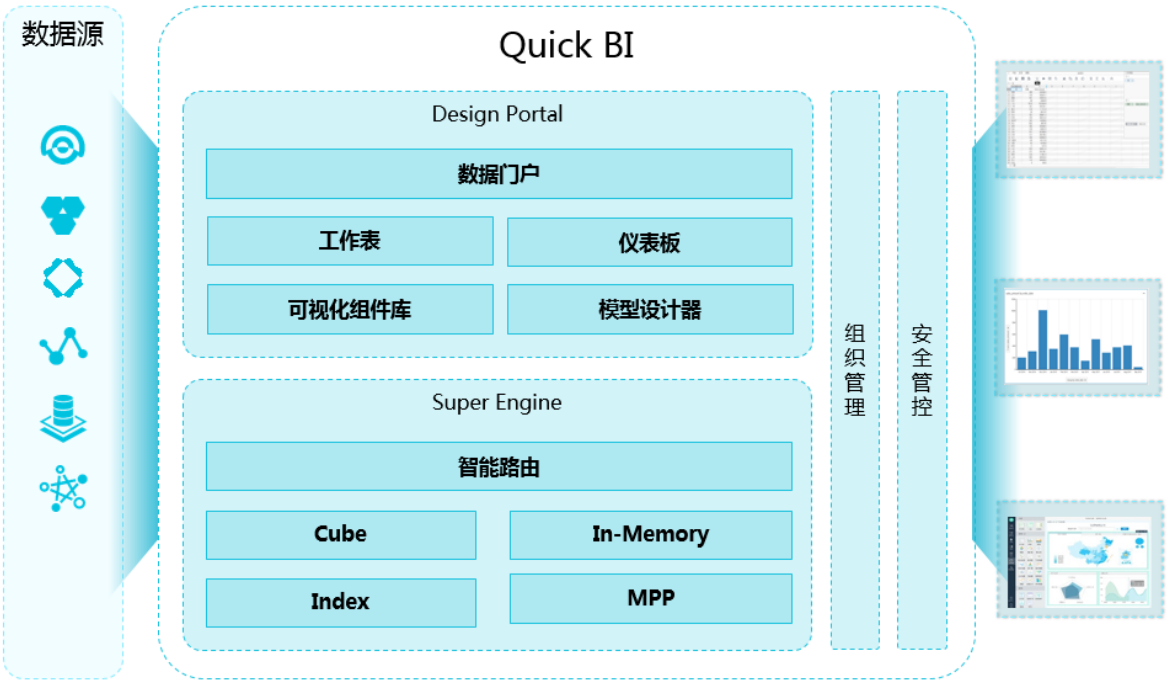
Quick BI支持众多种类的数据源，既可以连接MaxCompute（ODPS）、RDS、AnalyticDB、HybridDB（Greenplum）等云数据源，也支持连接ECS上您自有的MySQL数据库。Quick BI可以为您提供海量数据实时在线分析服务，通过提供智能化的数据建模工具，极大降低了数据的获取成本和使用门槛，通过支持拖拽式操作和提供丰富的可视化图表控件，帮助您轻松自如地完成数据透视分析、自助取数、业务数据探查、报表制作和搭建数据门户等工作。

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。

26.2 产品架构

Quick BI产品架构如下图所示。

图 103: 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，包括但不限于MaxCompute、RDS（MySQL、PostgreSQL、SQL Server）、AnalyticDB、HybridDB（MySQL、PostgreSQL）等，封装数据源的元数据/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，允许用户使用当前数据源的SQL语法对维度和度量进行二次加工。

- **工作表**

  负责在线电子表格（webexcel）的相关操作功能，涵盖行列筛选、普通/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、表格处理等丰富功能。

- **仪表板**

  负责将可视化图表控件拖拽式组装为仪表板，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和文本框4种基本控件，支持图表间数据联动效果。

- **数据门户**

  负责将仪表板拖拽式组装为数据门户，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-工作空间> 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，实现不同人看不同报表。

- **行级权限管理**

  负责数据的行级粒度权限管控，实现不同人看同一张报表展现不同数据。

- **转让/分享/公开**

支持将工作表、仪表板、数据门户转让或分享给其他登录用户访问，支持将仪表板公开到互联网供非登录用户访问。

26.3 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，包括但不限于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17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，智能推荐合适可视化方案。

分析

多维数据分析。基于Web页面的工作环境，拖拽式、类似于Excel的操作方式，一键导入、实时分析，可以灵活切换数据分析的视角，无需重新建模。

快速搭建数据门户

拖拽式操作、强大的数据建模、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，大大提高分析效率。

安全管控数据权限

内置组织成员管理，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人看到不同数据的需求。

26.4 产品优势

Quick BI的总体优势可总结为多，块，强大和易用。

多

支持RDS、MaxCompute、AnalyticDB等多种数据源。

快

亿级数据秒级响应。

强大

内置完整的电子表格工具，可以让您轻松完成复杂的中国式报表的制作。

易用

丰富的数据可视化功能，自动识别数据特征，自动智能为您生成最合适的图表。

27 关系网络分析

27.1 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。

27.2 产品架构

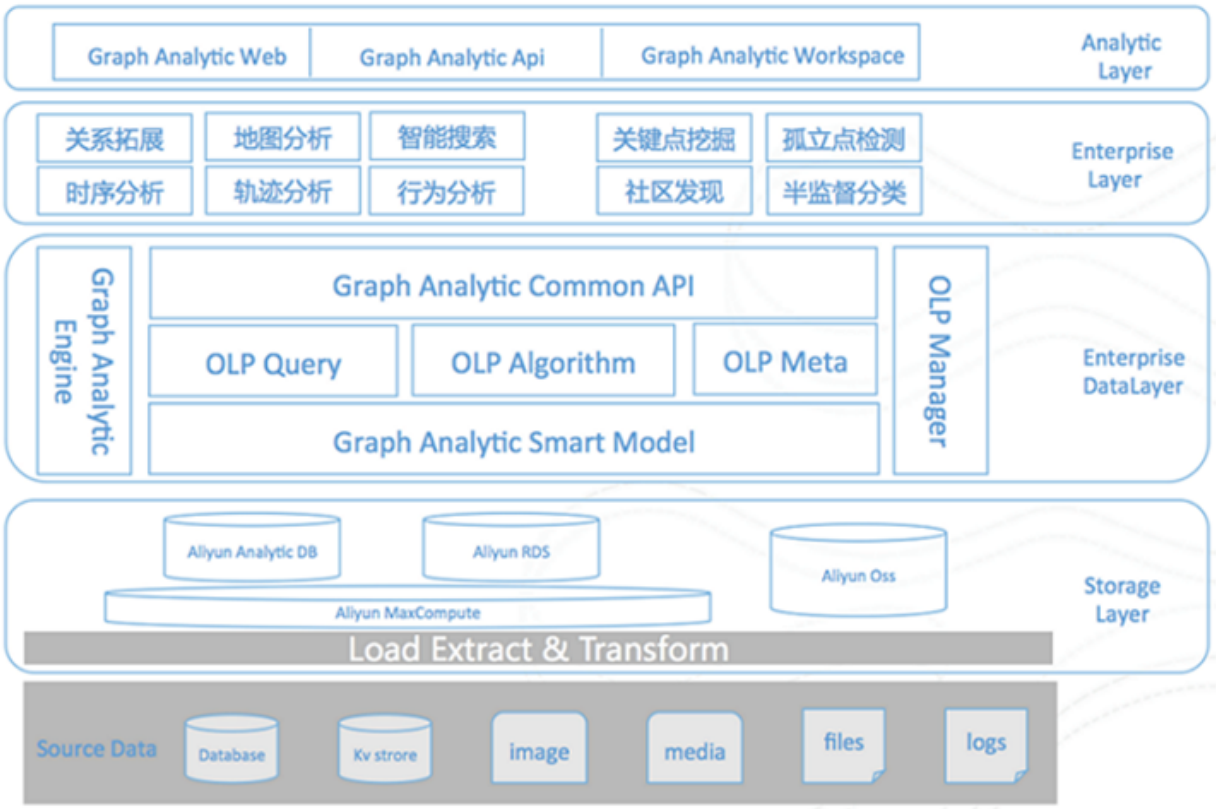
阿里云关系网路分析软件（Graph Analytic）（简称I +）采用组件化、服务化设计理念，多层次体系架构。

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 *104: 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 104: 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。

27.3 功能特性

27.3.1 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。

27.3.2 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。

27.3.3 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。

27.3.4 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。

27.3.5 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。

27.3.6 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）以及对象间的关联关系（Link），形成OLP数据模型。OLP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。

27.4 产品优势

27.4.1 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

27.4.2 跨计算数据整合建模，灵活高效部署

I + 平台以逻辑模型的建立替代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-属性”的逻辑模型。用户在业务数据理解和梳理后，通过I+管理后台轻松配置定义业务逻辑模型，配置逻辑模型和实际物理数据存储的映射关系，以及应用场景的参数即可以完成业务分析功能的建模和定义。逻辑模型支持根据数据源的变动或业务变动，进行灵活的修改和实时部署生效。

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天就能完成系统部署应用上线使用。

27.4.3 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。

27.4.4 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。

27.4.5 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。

27.5 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。

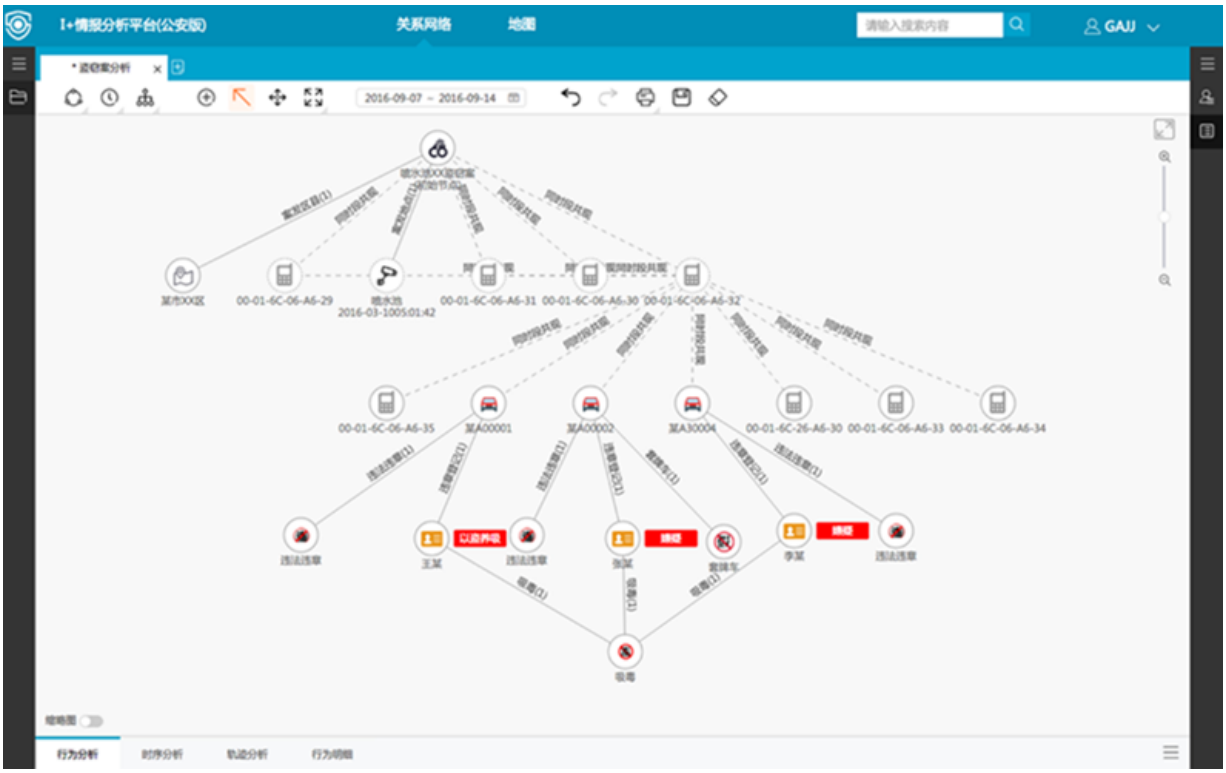
说明: 案例截图来源于真实应用系统，相关的案例敏感数据已做脱敏处理。

27.5.1 公安行业应用

此案例描述的是公安办案人员如何通过阿里云关系网路分析软件（简称I + ）快速侦破**以盗养吸**案件。

2016年以来，某地连续发生店铺被盗案件，店铺柜台、抽屉遭到不同程度的破坏，现金遗失、被盗。办案人员根据掌握的虚拟账号位置信息、车辆轨迹信息，结合时间、空间元素，利用阿里云关系网路分析软件（简称I + ）对多起案件进行串并分析，如图 *105: 盗窃案分析*所示，发现在多个案发现场、案发时段均出现相同的虚拟账号，根据已掌握的信息中未能直接分析得到虚拟账号对应的人员，但办案人员结合车辆轨迹、虚拟账号移动轨迹进行时空伴随分析，关联挖掘虚拟账号对应的车辆，进一步反查发现车主为吸毒前科人员，具有较大作案嫌疑，为整个案件的侦破提供了重要线索。

图 105: 盗窃案分析



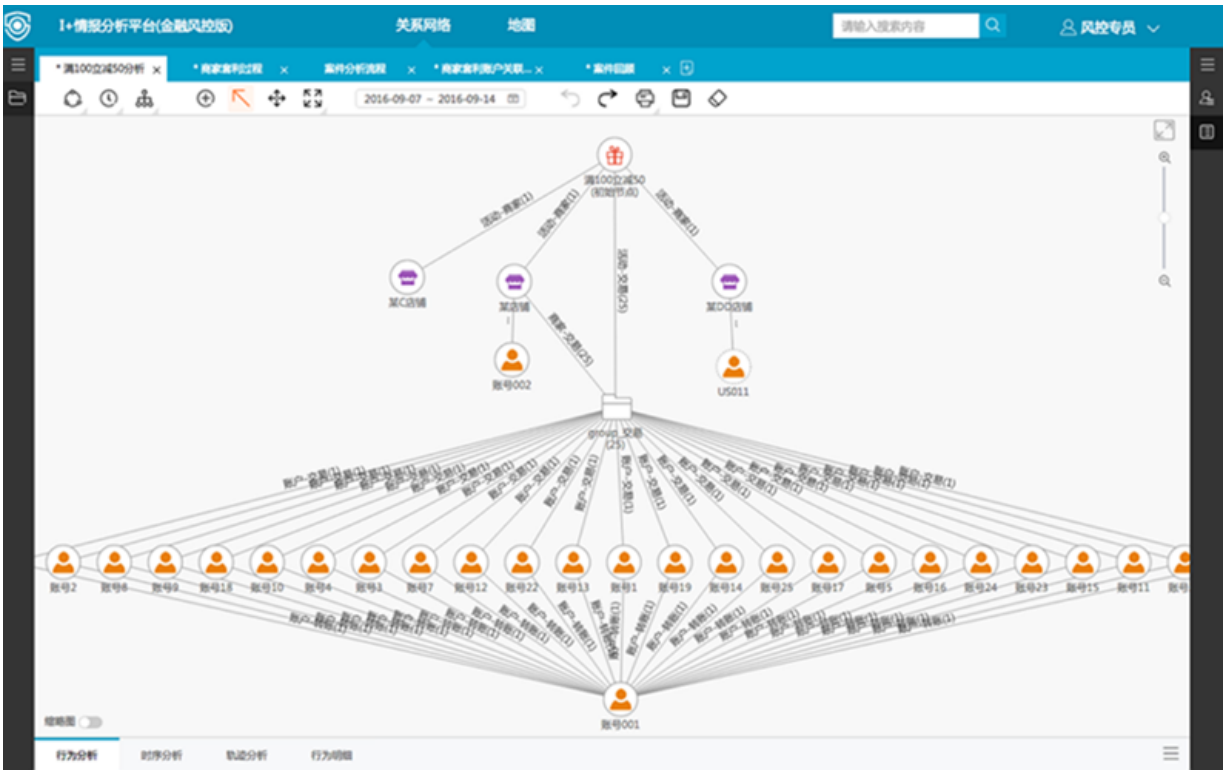
27.5.2 金融行业应用

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 *106: 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 106: 营销反作弊分析



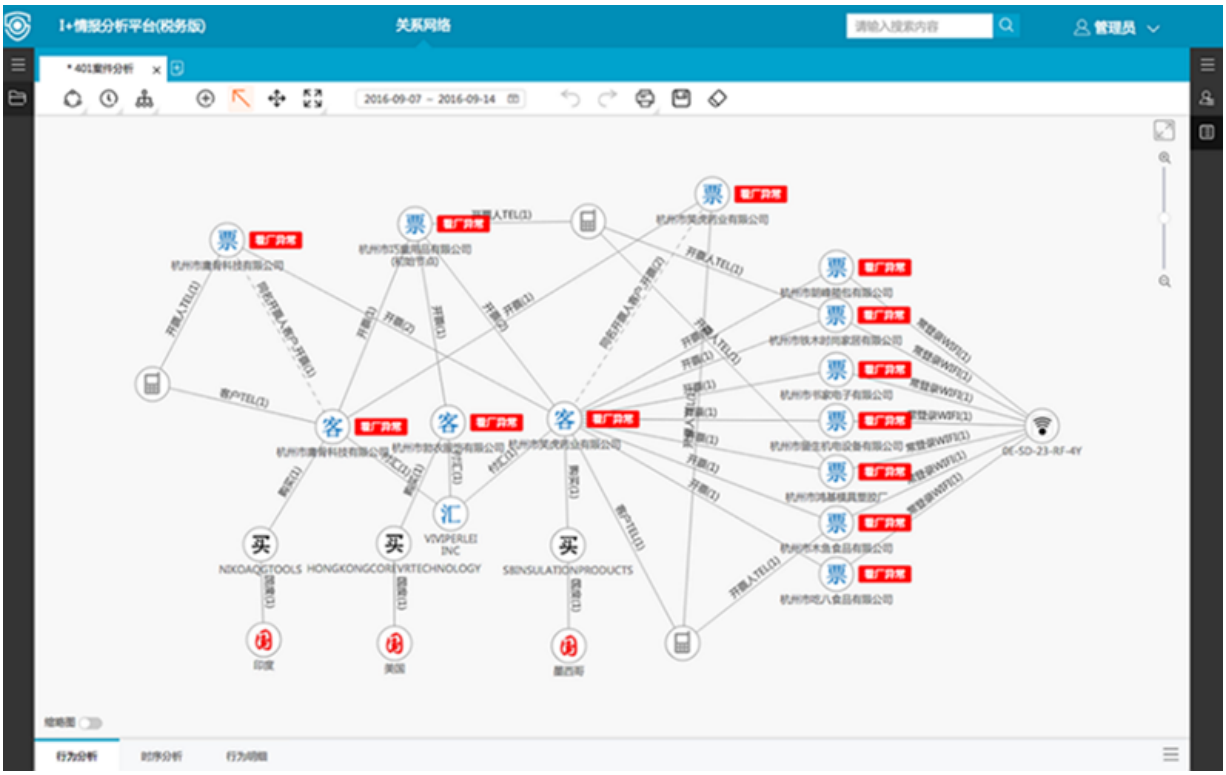
27.5.3 税务行业应用

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 *107:* 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 107: 出口骗税分析



28 采云间（DPC）

28.1 数据集成平台

28.1.1 产品概述

阿里云数据集成平台（Data Integration Platform，简称DIP）是蚂蚁大数据平台解决方案的一个分支产品，DIP基于MaxCompute，提供强大的产品功能，支持复杂的企业级数据集成，涵盖数据的集成、ETL开发、任务调度、发布部署和运维等功能，完成数据的价值重构和可信赖交付，帮助企业从大数据中获得更多价值。

28.1.2 产品架构

阿里云数据集成平台系统架构如图 *108: 阿里云数据集成平台系统架构*所示。

图 108: 阿里云数据集成平台系统架构

阿里云数据集成平台的基础架构分为五层：

- **用户层**：系统的用户包括实际的操作用户，也包括应用用户，支持不同角色的用户来访问。系统对不同用户，开放不同的权限；
- **接口层**：系统提供PC Web，API，SDK等方式，给外部客户使用；
- **功能层**：系统提供数据工厂，任务调度，数据字典和分发平台等功能，涵盖了从数据采集，数据加工到数据发布的全流程；

- **服务层**：系统通过云台屏蔽了不同PAAS平台的差异，提供基础统一的数据服务；
- **基础层**：一般是IAAS层和PAAS层提供的基础能力。

28.1.3 功能特性

28.1.3.1 ETL开发

ETL开发服务提供基于大数据平台的ETL开发的编程环境。

ETL设计与开发是大数据的DW和BI建设的关键环节之一，负责系统数据的生成，将数据在EDW数据架构的层次之间进行加工传输。

ETL开发主要包含的功能模块如下：

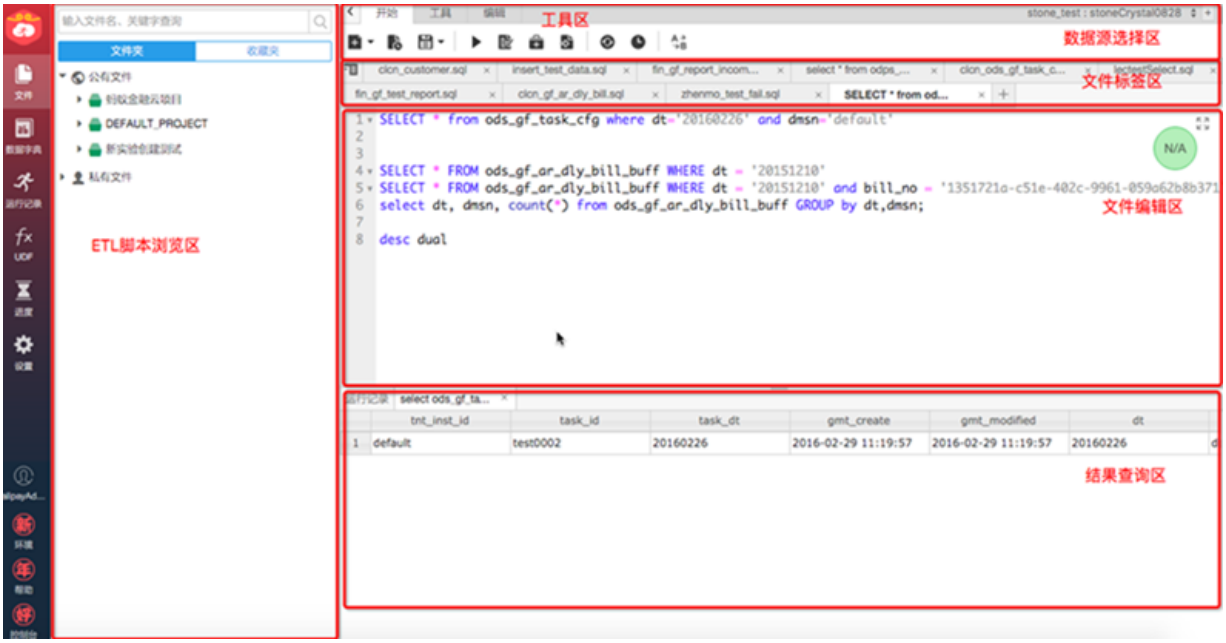
- 脚本开发
- 数据字典
- 数据管道

28.1.3.1.1 脚本开发

脚本开发模块提供脚本编辑和调试的开发环境，可以进行数据清洗和加工的脚本开发，也可以作为灵活查询的工具，执行查询命令。

如图 *109:* 脚本开发界面所示为脚本开发界面，脚本开发具备如下功能特性：

图 109: 脚本开发界面



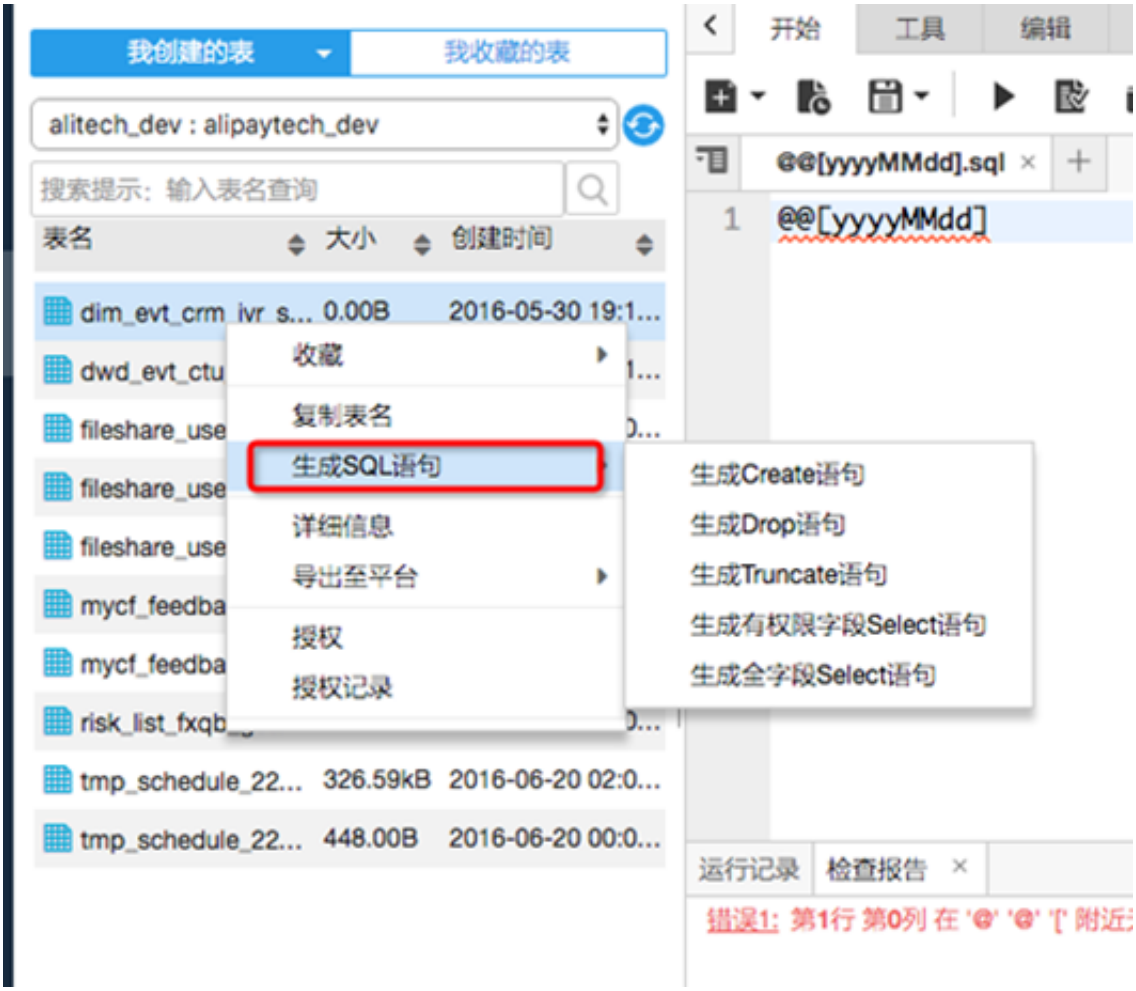
- 提供个人和团队进行ETL脚本协同开发。
- 方便的SQL执行和调试功能，可以对执行结果集进行各类操作，满足灵活查询的需要。
- 以文件树的方式记录、编辑和管理代码脚本，支持文件加锁和版本管理。
- 支持UDF函数上传，支持数据分析师上传自己的数据进行关联查询。

28.1.3.1.2 数据字典

数据字典功能可以查看表清单和字段信息，也可以收藏个人关心的表。

如图 *110:* 数据字典所示，数据字典具备如下功能特性：

图 110: 数据字典



- 可以查询指定数据源的所有的表和我创建的表。
- 显示查看表的详情，包括创建人、存储和字段清单。
- 分组收藏关心的表，方便使用。
- 自动生成常见SQL。

- 可以把表授权给他人，并查看授权记录。

28.1.3.1.3 数据管道

数据管道支持将数据导入到MaxCompute，也支持将数据从MaxCompute中导出。

如图 *111:* 数据管道所示，数据管道具备如下功能特性：

图 111: 数据管道

- 支持最大500M文件的上传，支持csv和txt格式。
- 支持查询结果导出（默认不开通，需要设置）。

28.1.3.2 任务管理

任务管理服务提供数据任务的全生命周期管理功能，可以周期性的运行数据采集、加工等任务，保证数据的日常开发。

任务管理主要包含的功能模块如下：

- 任务配置
- 任务监控
- 报警配置

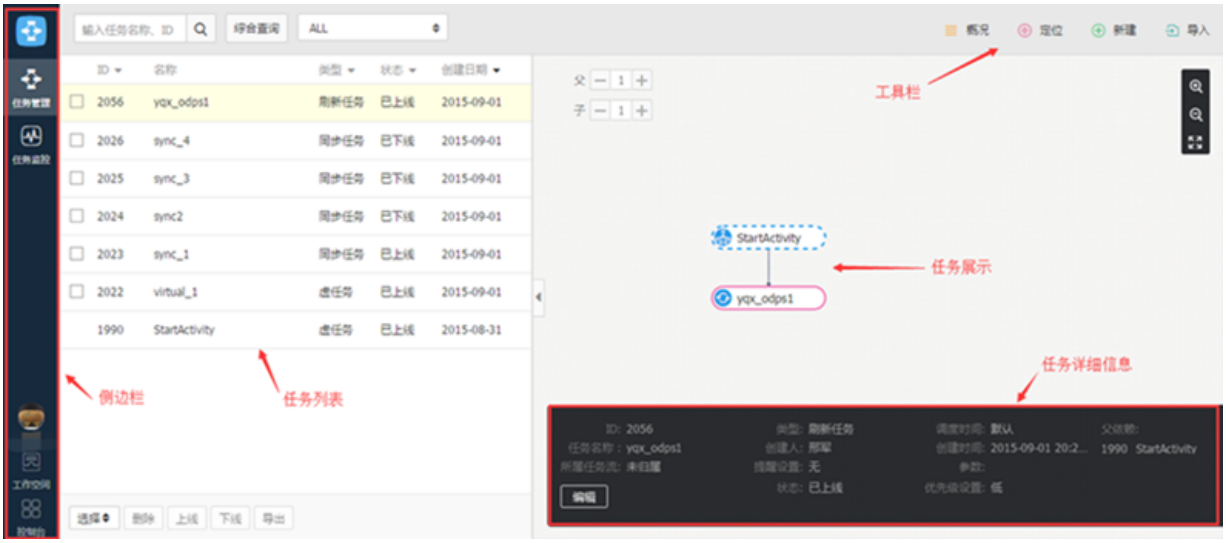
28.1.3.2.1 任务配置

任务配置支持任务的创建、编辑、上线和删除等操作，可以灵活设置任务依赖。

如图 *112: 任务配置*所示，任务配置具备如下功能特性：

- 云端数据同步和加载，通过简单易用的配置，支持RDS、Table Store、MaxCompute等异构数据库之间的双向同步。
- 图形化任务调度的设置，支持复杂的任务调度规则，包括任务定义、任务上下线、任务挂起、补数据、任务监控和异常报警等。
- 支持自动解析依赖，省却手动配置工作。

图 112: 任务配置



28.1.3.2.2 任务监控

任务监控提供任务运行的实时监控大图，方便查看任务执行进度，也可快速发现和解决出错任务。

如图 *113: 任务监控*所示，任务监控具备如下功能特性：

- 实时数据刷新，第一时间发现问题。
- 可视化的任务流显示，方便查找异常任务。

- 支持任务的挂起、跳过和重试等功能。

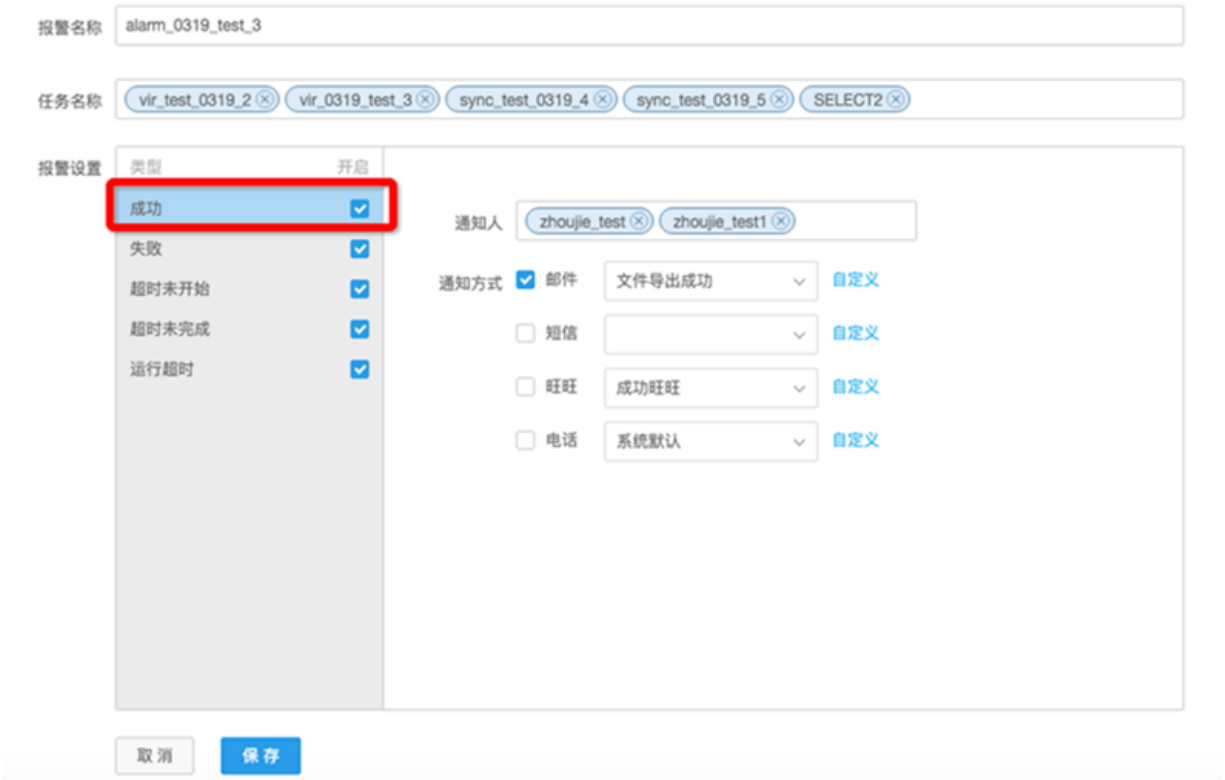
图 113: 任务监控

28.1.3.2.3 报警配置

报警配置功能支持设置报警规则，方便在第一时间在任务执行出现异常情况时，发出报警。

如图 *114:* 报警配置所示，报警配置具备如下功能特性：

图 114: 报警配置



- 支持多种渠道进行提醒。
- 支持多种预警规则类型：成功、失败、超时未开始、超时未结束和运行时长超时。
- 支持任务的挂起、跳过和重试等功能。

28.1.3.2.4 发布部署

发布部署支持把已经开发好的任务和脚本，发布到其他环境。

发布部署支持同集群发布和跨集群发布。

在发布部署模块，您可以：

- 创建发布包

  选择待发布的任务，设置发布目标环境，如图 *115:* 创建发布包所示。

  图 115: 创建发布包

- 发布监测

  系统会自动检测发布包中的任务，在目标环境的依赖关系是否完整，数据源是否存在，如图 *116: 发布监测*所示。

  图 116: 发布监测

- 发布审核

  发布管理员可以审阅待发布的发布包，以决定是否需要发布通过，如图 *117:* 发布审核所示。

  图 117: 发布审核

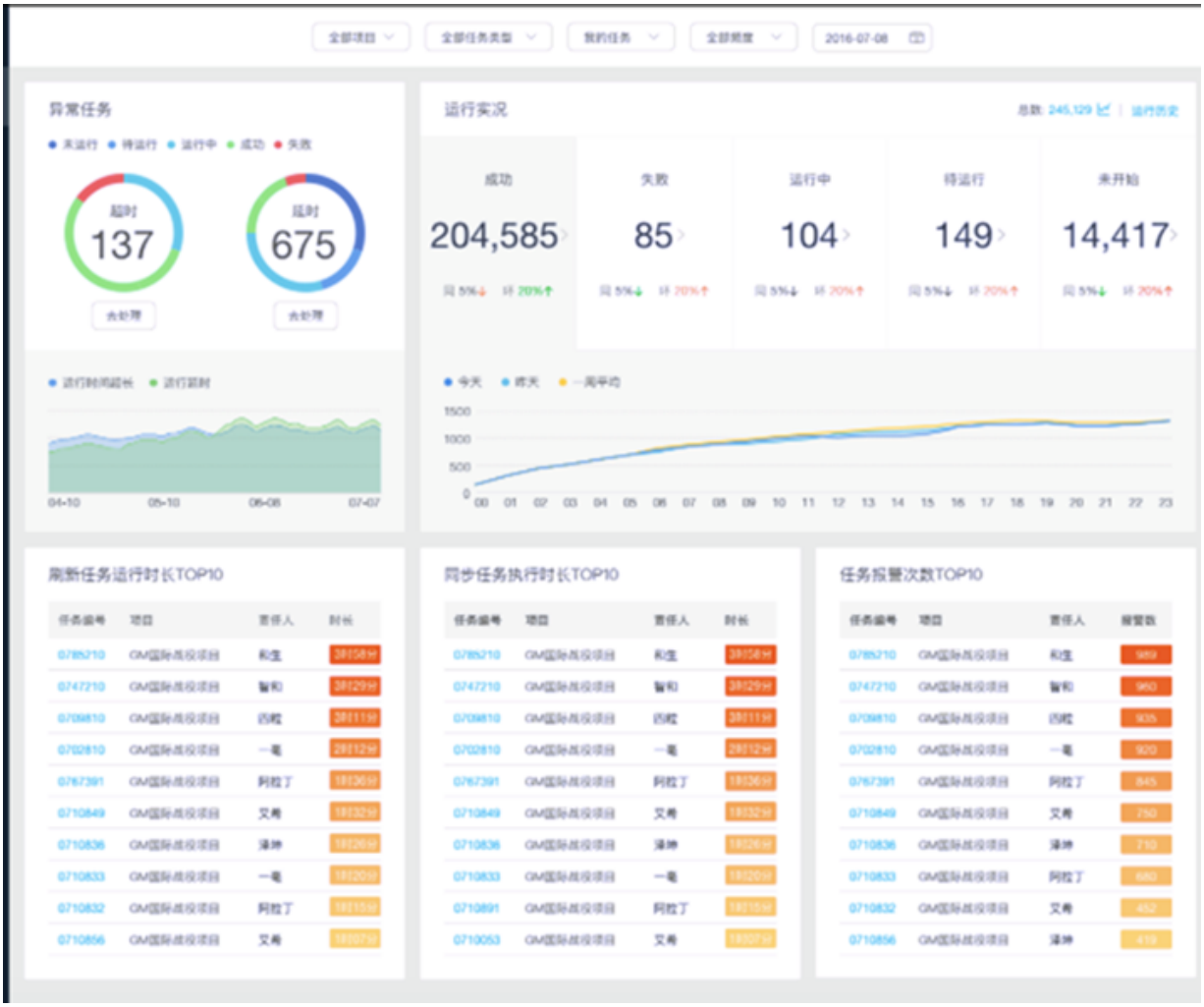
28.1.3.2.5 智能运维

如图 *118:* 智能运维所示，智能运维提供运维监控大屏，实时显示当前任务执行的情况：

- 异常任务：包括任务运行超时和启动延迟。
- 任务实况：显示任务的实时运行情况。
- 刷新任务运行时长Top10：显示运行时长Top10的刷新任务。

- 同步任务运行时长Top10：显示运行时长Top10的同步任务。
- 任务报警Top0：显示报警次数Top10的任务。

图 118: 智能运维



28.2 机器学习平台

28.2.1 产品概述

阿里云采云间机器学习平台是一套基于MaxCompute（原ODPS）的数据挖掘、建模、预测的工具。提供算法开发、分享、模型训练、部署、监控等一站式算法服务。

用户可以通过可视化的操作界面来操作整个实验流程，同时也支持PAI命令，让用户通过命令行来操作实验。阿里云机器学习平台沉淀了阿里巴巴的机器学习算法体系和经验，从数据的预处理、到机器学习算法、模型的评估和预测动能。

在专有云方面，阿里云机器学习平台的运行需要依赖于MaxCompute，通过将算法包部署到MaxCompute集群中，用户通过阿里云机器学习平台调用算法，实现算法的应用和计算引擎的解耦。

阿里云机器学习平台丰富的算法和技术保障支持也给用户解决自身业务场景带来了更多的可能性和想象空间。在DT时代，通过使用阿里云机器学习平台可以真正实现数据驱动业务的目的。

28.2.2 产品架构

如图 *119:* 阿里云机器学习平台系统架构所示，阿里云机器学习平台的基础架构分为四层，用户通过阿里云机器学习平台调用下一层的模型和算法，然后系统将相应的算法转换成对应的计算类型，比如说对于两个表的join操作，会自动生成SQL的workflow传到底层MaxCompute（原ODPS）进行运算和操作。所有算法都是以plugin的形式存放在底层的计算引擎当中，用户在使用算法的时候只关心算法的调用即可，真正实现了算法和计算引擎的解耦。

图 119: 阿里云机器学习平台系统架构

28.2.3 功能特性

28.2.3.1 完善的数据挖掘组件

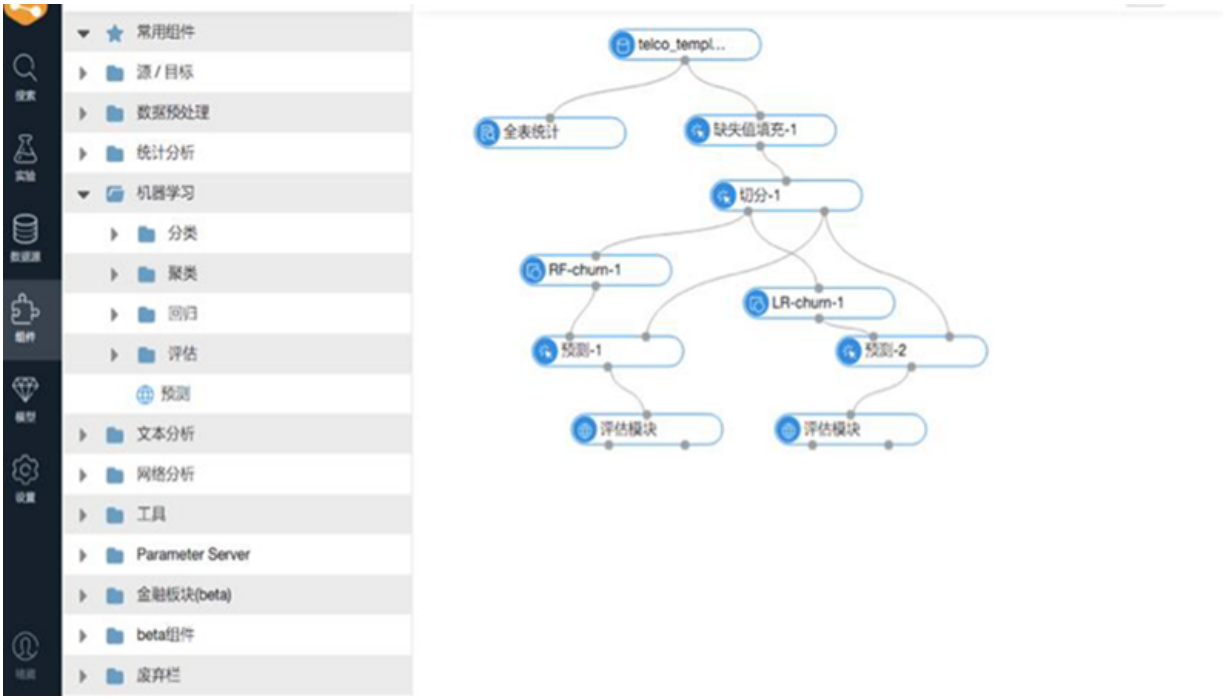
阿里云机器学习平台提供整套的数据挖掘组件，包括数据的预处理、特征抽取、模型训练、预测和评估。用户将数据导入平台，可以借助实验组件灵活的拼装自己的试验流程来解决自身的业务场景。

28.2.3.2 可视化建模

如图 *120:* 可视化建模所示，在操作界面，通过拖拉拼接实验。

从左边的组件框中拖拉组件到右边的实验区进行实验的搭建。

图 120: 可视化建模



28.2.3.3 数据可视化

对于每个输出型组件，都可以通过右键组件来查看可视化输出模型，如图 *121:* 数据可视化所示。

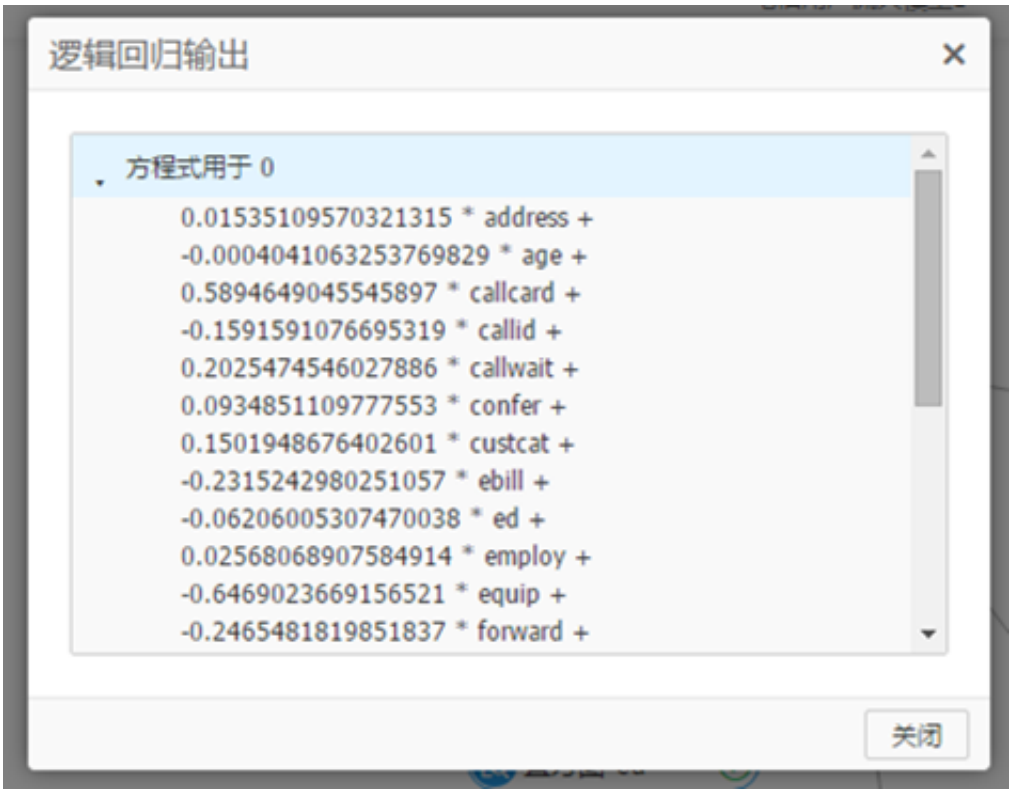
可视化输出有多种表示方法，包括折线图、点图和柱形图。

图 121: 数据可视化

28.2.3.4 模型可视化

在最左侧的菜单页签中选择模型页签，有查看模型选项，在相应的机器学习算法文件夹下就可以查看相应的模型结果，如图 *122:* 模型可视化所示。

图 122: 模型可视化


逻辑回归输出
方程式用于 0
0.01535109570321315 * address +
-0.0004041063253769829 * age +
0.5894649045545897 * callcard +
-0.1591591076695319 * callid +
0.2025474546027886 * callwait +
0.0934851109777553 * confer +
0.1501948676402601 * custcat +
-0.2315242980251057 * ebill +
-0.06206005307470038 * ed +
0.02568068907584914 * employ +
-0.6469023669156521 * equip +
-0.2465481819851837 * forward +
关闭

29 机器学习PAI

29.1 产品概述

阿里云机器学习平台是一套基于MaxCompute（原ODPS）的数据挖掘、建模、预测的工具。提供算法开发、分享、模型训练、部署、监控等一站式算法服务。用户可以通过可视化的操作界面来操作整个实验流程，同时也支持PAI命令，让用户通过命令行来操作实验。阿里云机器学习平台沉淀了阿里巴巴的机器学习算法体系和经验，从数据的预处理、到机器学习算法、模型的评估和预测动能。

在专有云方面，阿里云机器学习平台的运行需要依赖于MaxCompute，通过将算法包部署到MaxCompute集群中，用户通过阿里云机器学习平台调用算法，实现算法的应用和计算引擎的解耦。

阿里云机器学习平台丰富的算法和技术保障支持也给用户解决自身业务场景带来了更多的可能性和想象空间。在DT时代，通过使用阿里云机器学习平台可以真正的实现数据驱动业务的目的。

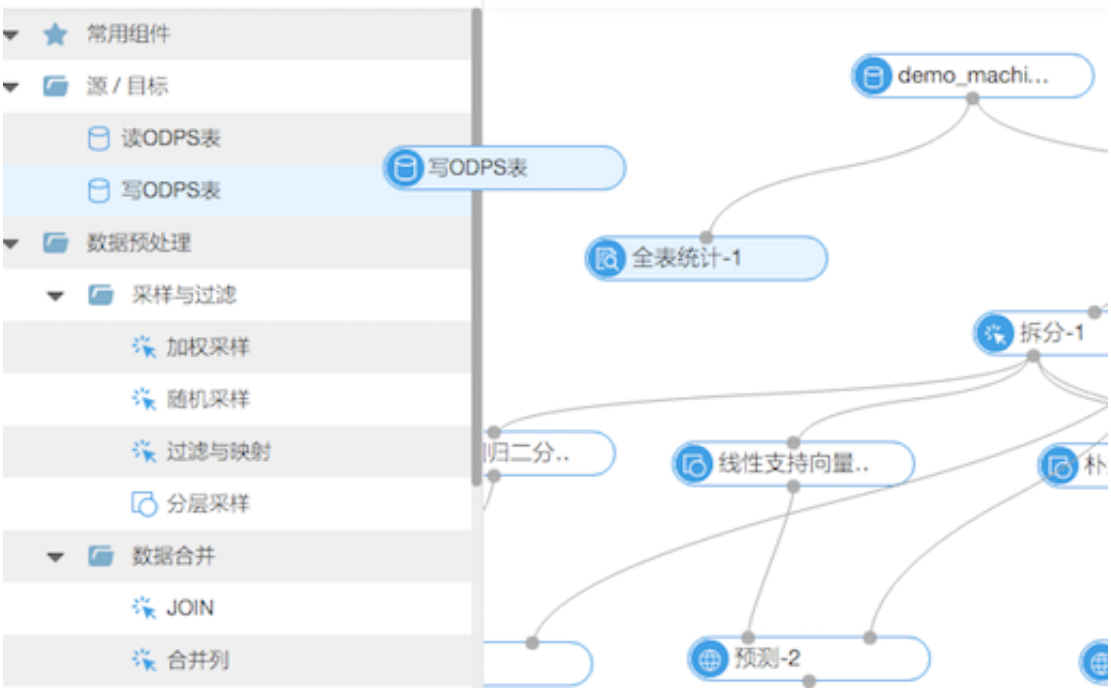
29.2 产品特性和核心优势

阿里云机器学习平台的产品主要优势可以概括为以下几方面：

良好的交互设计

通过拖拽的方式搭配实验，并且提供了数据模型的可视化功能。缩短了用户与数据的距离，真正实现了数据的触手可及。同时也提供了命令行工具，方便用户将算法嵌入到自身的工程中。

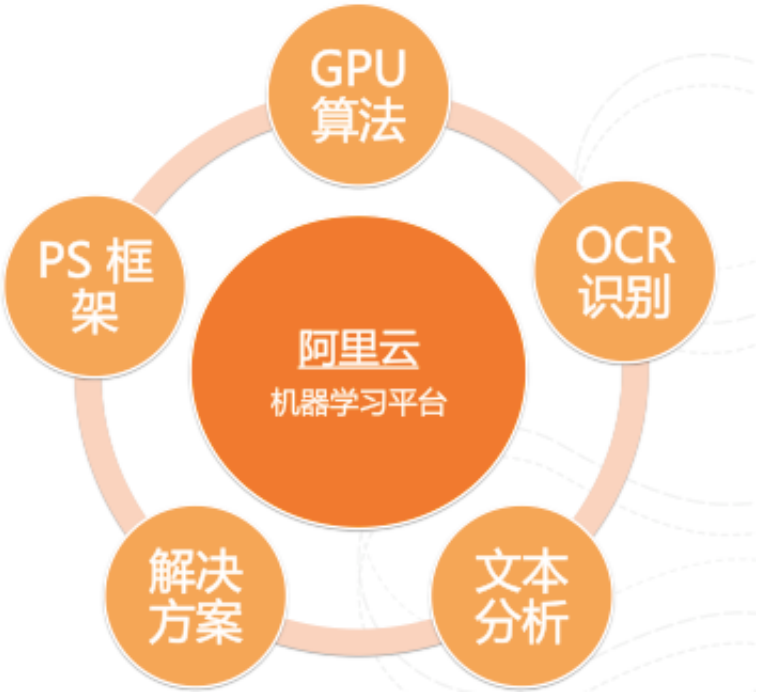
图 123: 操作界面



优质、丰富的机器学习算法

平台上边的机器学习算法都是经过阿里大规模业务锤炼的。从算法的丰富性角度来看，阿里云机器学习平台不仅提供了基础的聚类、回归等机器学习算法，也提供了文本分析、特征处理的算法。

图 124: 算法框架



与阿里系的融合

使用阿里云机器学习平台计算的模型直接存储在 MaxCompute（原ODPS）上。

可以配合其它阿里云的产品组件加以利用。

图 125: 阿里云数加产品图

优质的技术保障

阿里云机器学习算法平台的背后是阿里巴巴 IDST 的算法科学家和阿里云的技术保障团队，在使用过程中遇到任何问题都可以到工单系统提交工单或者直接与相关接口人联系。

图 126: 工单系统

29.3 系统架构

阿里云机器学习平台的基础架构分为四层，用户通过阿里云机器学习平台调用下一层的模型和算法，然后系统将相应的算法转换成对应的计算类型，比如说对于两个表的 join操作，会自动生成SQL 的 workflow 传到底层 MaxCompute（原 ODPS）进行运算和操作。所有算法都是以plugin 的形式存放在底层的计算引擎当中，用户在使用算法的时候只关心算法的调用即可，真正实现了算法和计算引擎的解耦。

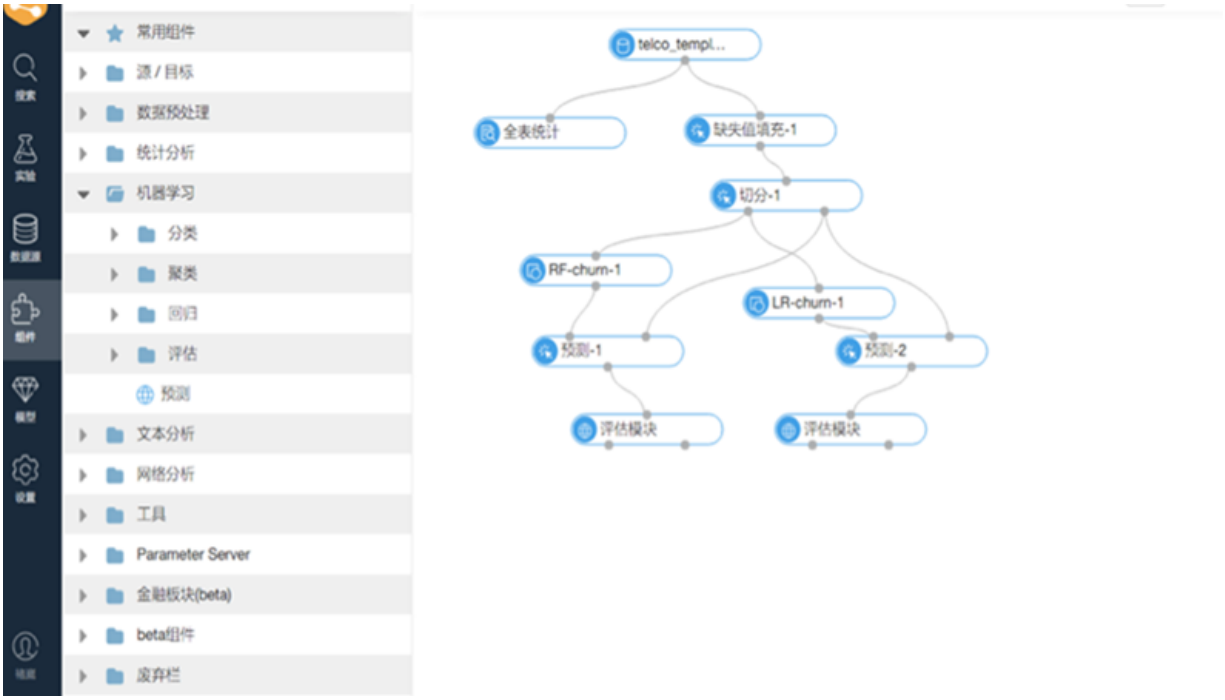
29.4 功能描述

完善的数据挖掘组件

提供整套的数据挖掘组件，包括数据的预处理、特征抽取、模型训练、预测和评估。用户将数据导入平台，可以借助实验组件灵活的拼装自己的试验流程来解决自身的业务场景。

可视化建模

操作界面：通过拖拉拼接实验。从左边的组件框中拖拉组件到右边的实验区进行实验的搭建。

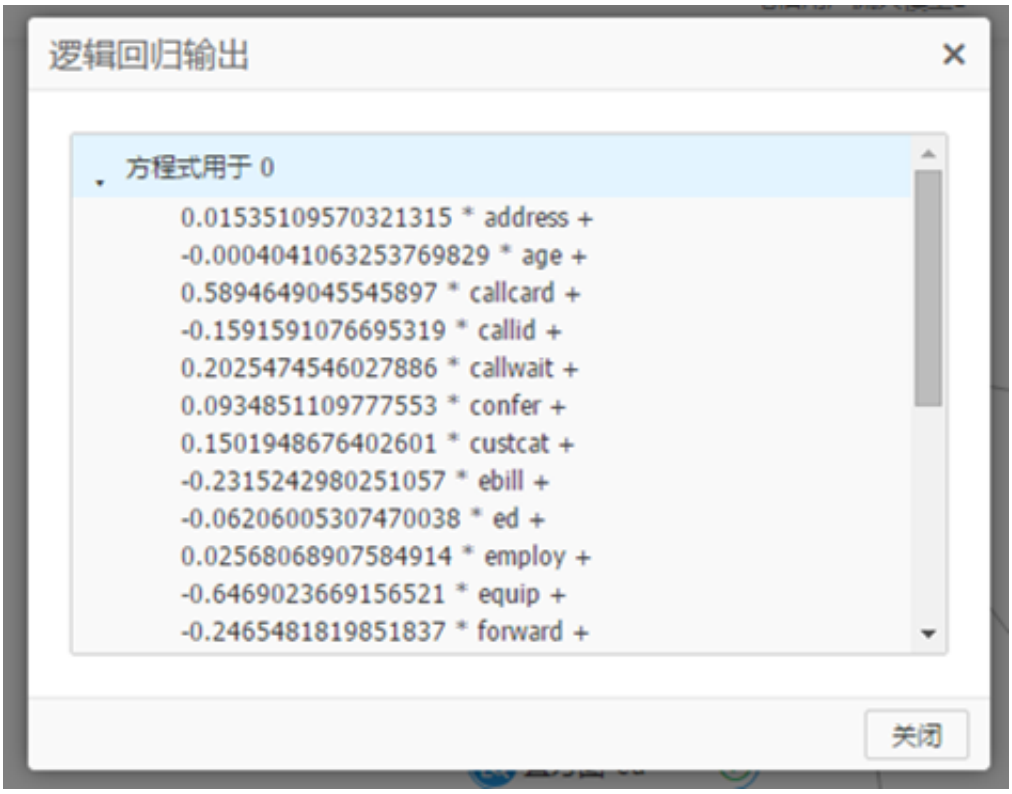


数据可视化

数据的可视化：对于每个输出型组件，都可以通过右键组件来查看可视化输出模型。可视化输出有多种表示方法，有折线图、点图和柱形图。

模型可视化

模型的可视化管理：在最左侧的菜单页签中选择模型页签，有查看模型选项，在相应的机器学习算法文件夹下就可以查看相应的模型结果。



算法组件

机器学习产品部署完成之后， 在可视化界面上可包括如下算法组件：

表 8: 算法组件

一级目录	二级目录	算法组件	说明
源/目标			机器学习 PAI 输入数据源。
		读 MaxCompute 表	读取数据源。
		写 MaxCompute 表	写数据源。
数据预处理	采样与过滤	加权采样	按照一定比例采样。
		随机采样	按照随机采样。
		过滤与映射	利用 sql where 过滤。
		分层采样	按照等级抽样。
	数据合并	拆分	按照比例拆分数据。
		合并列	合并两个表的两列。
		UNION	SQL UNION。
	其它	标准化	对表的某一列进行标准化处理。
		归一化	归一化是一种简化计算的方式，即将有量纲的表达式，经过变换，化为无量纲的表达式，成为标量。
		增加序列号	在数据表中增加自增 id 列。
特征工程	特征变换	主成分分析 PCA	降维算法。
	特征重要性评估	线性特征重要性	对应算法的特征评估。
		随机森林特征重要性	
统计分析		百分位	计算某列的百分位。
		全表统计	计算全表的每个字段的统计信息，包括缺省

一级目录	二级目录	算法组件	说明
			值、最大值、最小值、方差、偏值等。
		皮尔森系数	计算两字段（数值型）的皮尔森相关系数。
		直方图	多个字段查看直方图。
		散点图	数据点在直角坐标系平面上的分布图。
		相关系数矩阵	计算多个字段的相关系数矩阵。
机器学习	二分类	线性支持向量机	支持向量机 SVM 是一个有监督的学习模型，通常用来进行模式识别、分类、以及回归分析。
		逻辑回归二分类	通过逻辑回归算法对数据进行训练生成二分类模型，属于有监督的机器学习。
		GBDT二分类	GBDT 是一种迭代的决策树算法，该算法由多颗决策树组成，所有树的结论累积起来做最终答案。
	多分类	逻辑回归多分类	线性回归的多分类。
		随机森林	随机森林指的是利用多棵树对样本进行训练并且预测的一种分类器。
		朴素贝叶斯	朴素贝叶斯法是基于贝叶斯定理与特征条件独立假设的分类方法。
	回归	GBDT 回归	利用 GBDT 树状结构最回归。

一级目录	二级目录	算法组件	说明
	聚类	K均值聚类	聚类相似度是利用各聚类中对象的均值所获得的一个中心对象来进行计算的。
	评估	二分类评估	对于上述算法的评估和预测。
		多分类评估	
		回归模型评估	
		聚类模型评估	
		混淆矩阵	
	预测	预测	
	关联推荐	协同过滤 etrec	etrec 是一个 item base 的协同过滤算法，输入为两列，输出为 item 之间相似度 topK。
文本分析		分词	对指定的文本内容列进行分词，目前仅支持中文淘宝分词和互联网分词。
		词频统计	在分词基础上，按行保序输出对应文章 ID 对应的词，统计指定文章 ID 列对应内容的词频。
		TF-IDF	TF-IDF 是一种统计方法，用以评估一个词对于文件集或一个语料库中的其中一份文件的重要程度。
		PLDA	给出每篇文档中主题的概率真分布。
		Word2Vec	将词表转为向量。
		三元组转 KV	将给定的三元组（row, col, value）转

一级目录	二级目录	算法组件	说明
			成 kv 格式（row, [col_id,value]）
		文本摘要	使用算法在文章中提取文摘。
		关键词提取	从指定的文本中提取和这篇文章意义最相关的词组。
		句子拆分	按照句子标点符号进行拆分。
工具		MaxCompute SQL	执行 MaxCompute SQL。
网络分析		K-Core	一个图的 Kcore 是指反复去除度小于或等于 K 的节点后，所剩余的子图。
		单源最短路径（SSSP）	计算最短路径。
		PageRank	计算网页 Rank 值。
		标签传播聚类（LabelPropagationClustering）	基于图的半监督学习方法，依赖其邻居节点的标签信息，影响程度由节点相似度决定，并通过传播迭代更新达到稳定。
		标签传播分类（LabelPropagationClassification）	该算法为半监督分类算法，原理为用已标记节点的标签信息去预测未标记节点的标签信息。
		Modularity	Modularity 是一种评估社区网络结构的指标，来评估网络结构中划分出来的社区的紧密程度，往往0.3以上是比较明显的社区结构。

一级目录	二级目录	算法组件	说明
		最大联通子图（maximalConnectedComponent）	在无向图 G 中，从连通的顶点寻找可以顶点相通的子图，寻找到的子图数据量称为最大联通子图。
		点聚类系数（nodeDensity）	在无向图 G 中，计算每一个节点周围的稠密度，星状网络稠密度为0，全联通网络稠密度为1。
		边聚类系数（edgeDensity）	在无向图 G 中，计算每一条边周围的稠密度。
		计数三角形（triangleCount）	在无向图 G 中，输出所有三角形。
		树深度（treeDepth）	对于众多树组成的网络，输出每个节点所处深度和树id。

29.5 应用场景

泰坦尼克号沉船事件：通过分析泰坦尼克号沉船事件幸存者和丧生者的数据，判断拥有什么样的属性的人有更大的概率获救。

数据准备

图 127: 数据准备

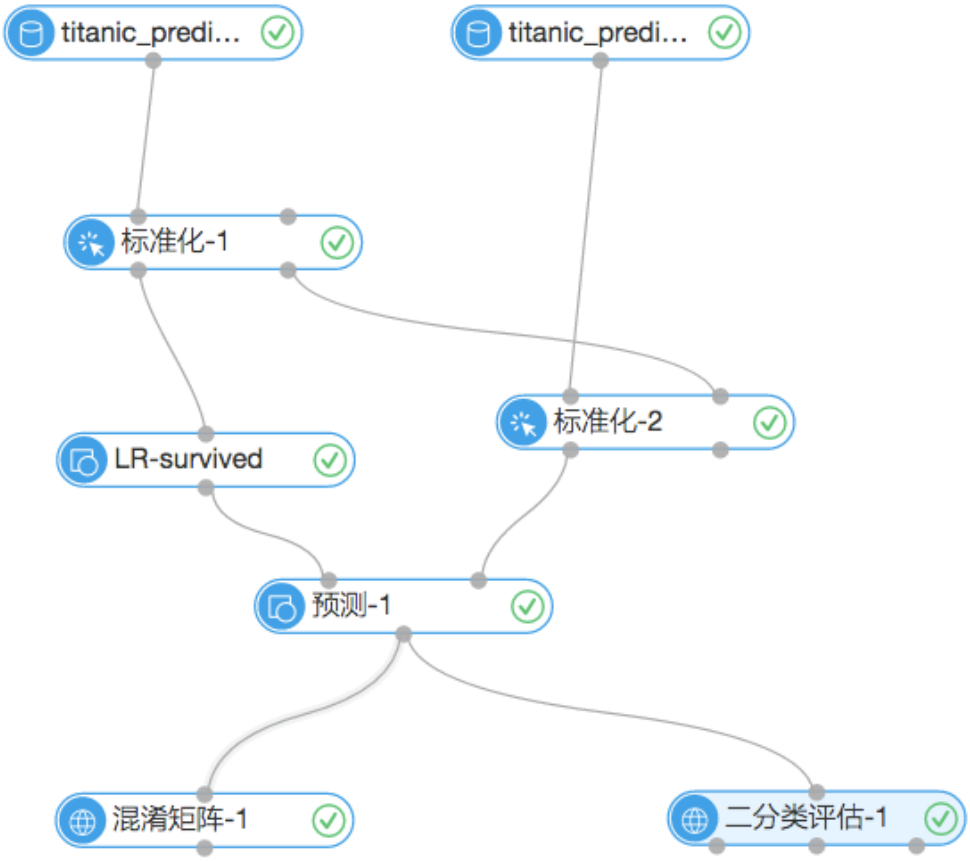
passengerid	survived	pclass	sex	age	sibsp	parch	fare	cabin	embarked
1.0	0.0	3.0	1.0	22.0	1.0	0.0	7.25	0.0	1.0
2.0	1.0	1.0	0.0	38.0	1.0	0.0	71.2833	1.0	2.0
4.0	1.0	1.0	0.0	35.0	1.0	0.0	53.1	1.0	1.0
5.0	0.0	3.0	1.0	35.0	0.0	0.0	8.05	0.0	1.0
7.0	0.0	1.0	1.0	54.0	0.0	0.0	51.8625	1.0	1.0
8.0	0.0	3.0	1.0	2.0	3.0	1.0	21.075	0.0	1.0
9.0	1.0	3.0	0.0	27.0	0.0	2.0	11.1333	0.0	1.0
10.0	1.0	2.0	0.0	14.0	1.0	0.0	30.0708	0.0	2.0

将数据导入MaxCompute：

- passengerId：用户的ID号
- survived： 乘客是否获救；1表示获救，0表示没有获救。目标队列（target）
- pclass：乘客的社会阶层；1表示Upper，2表示Middle，3表示Lower
- sex：乘客的性别；1表示男，0表示女
- age：乘客的年龄
- sibsp：乘客在船上的配偶数量或兄弟姐妹数量
- parch：乘客在船上的父母或子女数量
- fare：乘客的船费
- cabin：是否住在独立的房间；1表示是，0为否
- embarked：表示乘客上船的码头距离泰坦尼克出发码头的距离，数值越大表示距离越远

搭建实验流程

图 128: 搭建实验流程



1. 首先将数据集按照7：3进行拆分，一部分作为 Titanic 训练集，一部分作为预测集。
2. 将数据进行标准化处理，去除量纲对于数据造成的干扰。
3. 训练数据通过逻辑回归 LR 算法生成模型。
4. 对预测集进行预测。
5. 通过 ROC 曲线和混淆矩阵来对结果进行评估。

评估结果

- 混淆矩阵结果

图 129: 混淆矩阵结果

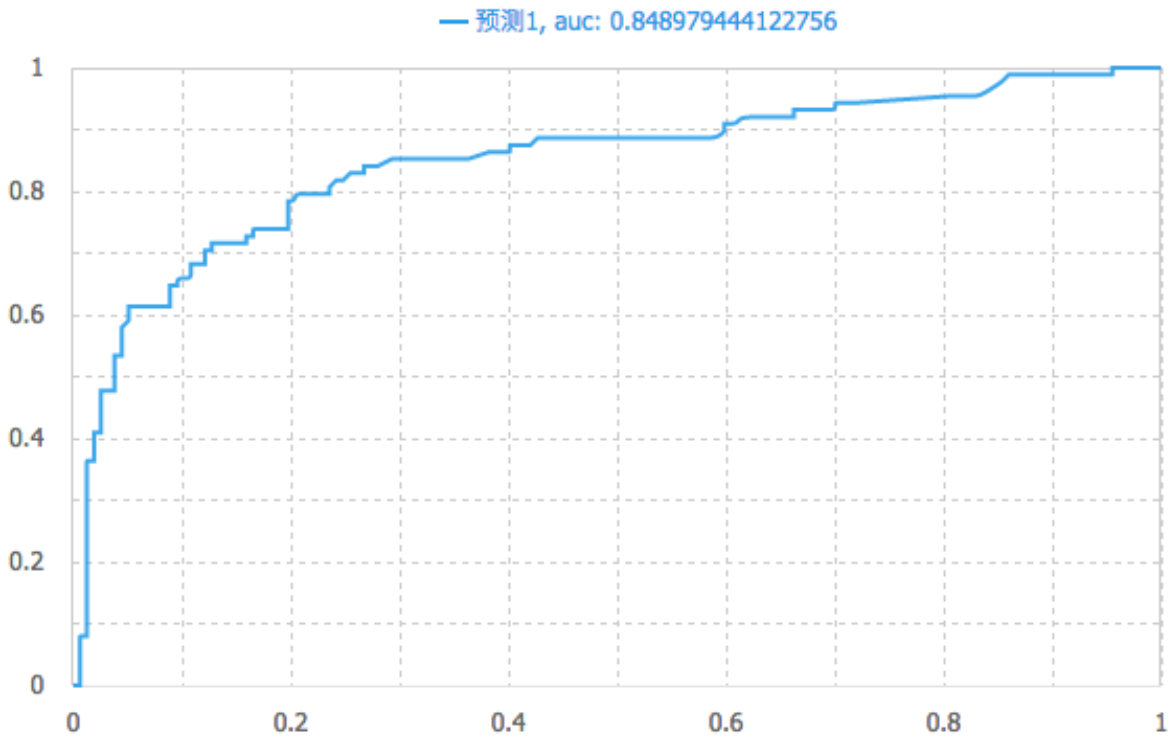
预测1

模型	正确个数	错误个数	总计	正确率	召回率	F1指标
1	65	27	92	70.652%	73.864%	72.222%
0	130	23	153	84.967%	82.803%	83.871%

1. 正确率：预测正确的正例个数占预测为正例的比例，即 P=TP/(TP+FP)。
2. 召回率： 预测正确的正例个数占实际正例的比例，即 R=TP/(TP+FN)。
3. F1 指标：P 和 R 的中权调和平均，即 F1=2PR/(P+R)。当 F1 较高时说明实验方法比较理想。

- ROC 曲线评估结果

图 130: ROC曲线评估结果



1. TPR=TP/(TP+FN)
2. FPR=FP/(FP+TN)
3. ROC 曲线
   a. Y轴：TPR; X轴：FPR
   b. (0,1)：FRP=0,TPR=1。FN=0,FP=0，将所有样本都正确分类

c. (1,0)：即，FPR=1,TPR=0，预测结果相反，正变负，负变正

d. (0,0)：FRP=TPR=0，将所有样本划分为负样本

e. (1,1)：所有样本划分为正样本

f. ROC 曲线越接近左上角，该分类器越好

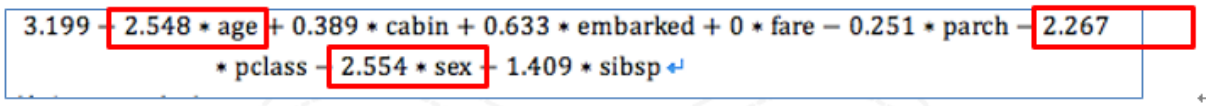
4. y=x，随机猜测，一半正样本，一半负样本

5. AUC：ROC 曲线下面积，应该介于0.5~1之间。AUC 越大，分类器越好

模型分析

逻辑回归生成模型：

图 131: 模型分析



根据 Logical regression 的特性，model 输出的是每个特征的线性组合。3.199为常数项，不予考虑。其它系数绝对值越大说明对结果影响越大。通过这一结论得出 age、sex 和 pclass 对于结果影响最大。

根据 sigmoid 函数得出负号系数的绝对值越大其结果的正例可能性越大。所以我们可以得出结论，age、pclass 和 sex 的值越小，目标值越大。

也就是有钱人家的女人和小孩有更大的获救概率。

通过真实数据比对也印证了我们的分析结果：

图 132: 分析结果

age	cabin	embarked	fare	parch	pclass	sex	sibsp	survived	prediction_result
26.0	0.0	1.0	7.925	0.0	3.0	0.0	0.0	1.0	1
30.0	0.0	3.0	8.4583	0.0	3.0	1.0	0.0	0.0	0
58.0	1.0	1.0	26.55	0.0	1.0	0.0	0.0	1.0	1
14.0	0.0	1.0	7.8542	0.0	3.0	0.0	0.0	0.0	1
35.0	0.0	1.0	26.0	0.0	2.0	1.0	0.0	0.0	0
15.0	0.0	3.0	8.0292	0.0	3.0	0.0	0.0	1.0	1
19.0	1.0	1.0	263.0	2.0	1.0	1.0	3.0	0.0	0
30.0	0.0	3.0	7.8792	0.0	3.0	0.0	0.0	1.0	1
30.0	0.0	3.0	7.75	0.0	3.0	0.0	0.0	1.0	1
30.0	0.0	2.0	7.2292	0.0	3.0	1.0	0.0	1.0	0
21.0	0.0	1.0	8.05	0.0	3.0	1.0	0.0	0.0	0
3.0	0.0	2.0	41.5792	2.0	2.0	0.0	1.0	1.0	1
30.0	0.0	3.0	15.5	0.0	3.0	1.0	1.0	0.0	0
30.0	0.0	3.0	7.75	0.0	3.0	0.0	0.0	1.0	1
30.0	0.0	2.0	21.6792	0.0	3.0	1.0	2.0	0.0	0