阿里云
大数据轻量专有云

产品简介

产品版本：V1.1.0

文档版本：20180327

为了无法计算的价值 | 阿里云

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

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 产品概述

大数据轻量专有云（Apsara Stack Insight）是阿里巴巴基于阿里云飞天分布式操作系统研发的大数据产品集合，是阿里云大数据产品在小规模集群上应用的解决方案。

大数据轻量专有云包括大数据计算服务MaxCompute、分析型数据库AnalyticDB、大数据开发套件DataWorks、大数据应用加速器和关系网络分析Graph Analytics 五款核心产品。

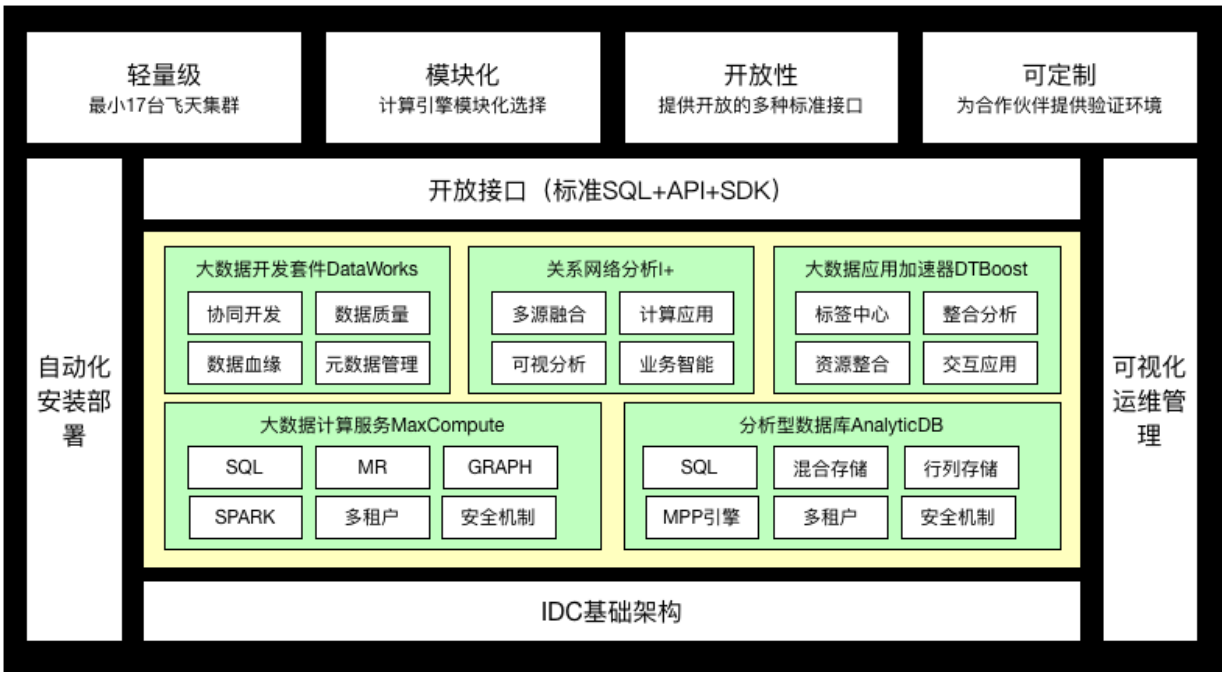
大数据轻量专有云极大的降低了客户使用阿里云大数据产品的成本和门槛，这将使阿里云大数据产品普惠到各行各业，尤其是市级、地区级的单位和企业。

2 系统架构

2.1 产品架构

大数据轻量专有云是基于标准的IDC基础架构、x86服务器、以太网交换机的大数据产品。

大数据轻量专有云的产品架构如下图所示。



在硬件服务器之上，大数据轻量专有云搭载飞天操作系统，形成大规模集群。大数据计算服务MaxCompute和分析型数据库AnalyticDB是基于飞天操作系统开发的大数据计算平台。在计算平台之上，大数据轻量专有云提供大数据开发套件DataWorks、关系网络分析I+ 和大数据应用加速器DTBoost。

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

大数据轻量专有云主要特点如下：

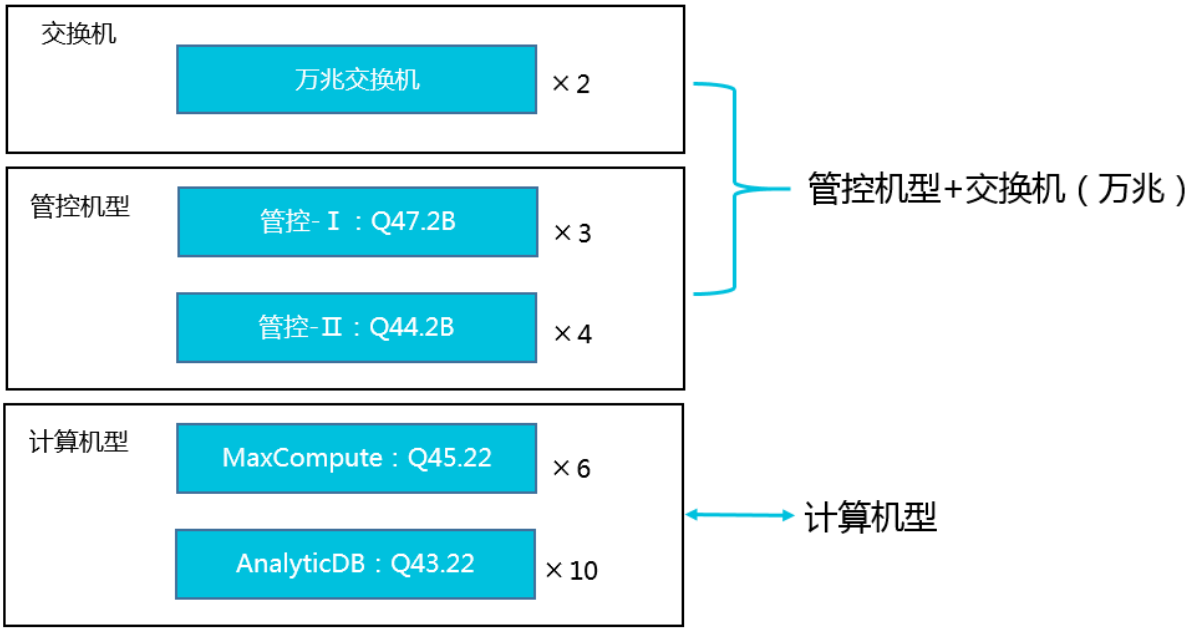
- 轻量化：支持15台~96台服务器的飞天集群。
- 模块化：支持模块化组合交付，常用组合如下：
  - Maxcompute与DataWorks组合
  - Maxcompute、AnalyticDB与DataWorks组合

  - AnalyticDB
  - AnalyticDB与I + 组合
  - Maxcompute 、DataWorks、AnalyticDB、I + 与DTBoost组合
- 开放性：提供多种标准接口。
- 可定制：为合作伙伴提供预验证的环境。

2.2 交付形态

大数据轻量专有云对硬件的要求，以及计算机型对服务器的要求。

大数据轻量专有云的交付形态如下图所示。



硬件要求：至少包含2台万兆交换机、7台管控服务器。DataWorks 、I+和 DTBoost 软件部署在7台管控服务器中，不需要额外的服务器。

计算机型对服务器要求：MaxCompute至少需要6台物理服务器，AnalyticDB至少需要10台服务器。

3 系统容量

介绍大数据轻量专有云中AnalyticDB和MaxCompute的性能参数。

AnalyticDB性能参数

类型	机器数（台）	最大数据量（TB）	查询（多维）	查询（复杂度）	写入（rec/s）	写入（MB/s)
高性能	10~30	10~50	200~1000并发	70~350并发	8万~40万	30~150
大存储	10~30	100~500	20~100并发	10~50并发	8万~40万	30~150

MaxCompute性能参数

机型	机器数（台）	理论最大数据量	单机磁盘利用率	压缩比
Q45.2B	6~30	336.6TB~1PB	0.668	3

4 产品优势

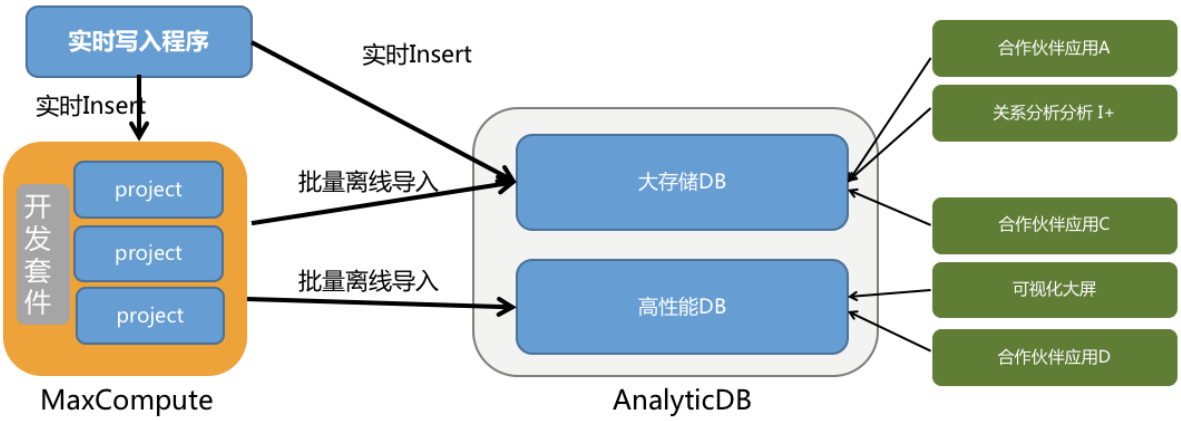
大数据轻量专有云的主要优势如下：

- 大数据轻量专有云的集群规模小，输出的产品能力强，能够形成数据链路闭环。
- MaxCompute能够高效的处理海量数据的离线分析，在阿里巴巴集团内稳定运行。
- MaxCompute提供多种授权管理手段，包括ACL、角色授权、Policy授权、跨Project授权和Label机制，可以提供精确到列级别的安全方案。
- AnalyticDB支持同时新建纯ssd存储实例、ssd与sata混合存储实例，两种实例可以在同一集群内同时存在，极大得提高了用户对集群的使用效率。
- AnalyticDB支持离线导入数据和实时写入数据，实时写入生产可达到200万行每秒。
- DataWorks支持多人协同开发，支持多种任务调度模型，支持统一数据及权限管理。

5 典型场景

在使用大数据轻量专有云过程中，大数据轻量专有云各产品会形成数据链路闭环。

大数据轻量专有云各产品形成的数据链路闭环，如下图所示。



将源数据写入到AnalyticDB的方式如下：

- 支持从MaxCompute同步数据到AnalyticDB：
  1. 用户离线数据通过数据开发套件DataWorks提供的数据集成功能，从源数据库中将数据同步到MaxCompute的project中；用户实时数据通过实时写入程序实时写入到MaxCompute的project中。
  2. 用户根据业务逻辑在数据开发套件中开发任务，可以是ETL类任务、与业务逻辑相关的任务、多个有依赖性的任务。
  3. 用户开发同步任务将数据批量离线同步到AnalyticDB。
- 支持通过实时写入程序将实时数据以标准INSERT语句写入到AnalyticDB。

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

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

6 MaxCompute

6.1 什么是MaxCompute

大数据计算服务（MaxCompute）是阿里巴巴内部发展的一个高效能、低成本，高可用的**EB级**大数据计算服务，在集团内部每天处理超过EB级的数据量。

MaxCompute是面向大数据处理的分布式系统，主要提供结构化数据的存储和计算，是阿里巴巴云计算整体解决方案中最核心的主力产品之一。

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

MaxCompute主要服务于批量结构化数据的存储和计算，可以提供海量数据仓库的解决方案以及针对大数据的分析建模服务。

MaxCompute的目的是为用户提供一种便捷的分析处理海量数据的手段 。用户可以不必关心分布式计算细节，从而达到分析大数据的目的。

MaxCompute产品特点如下：

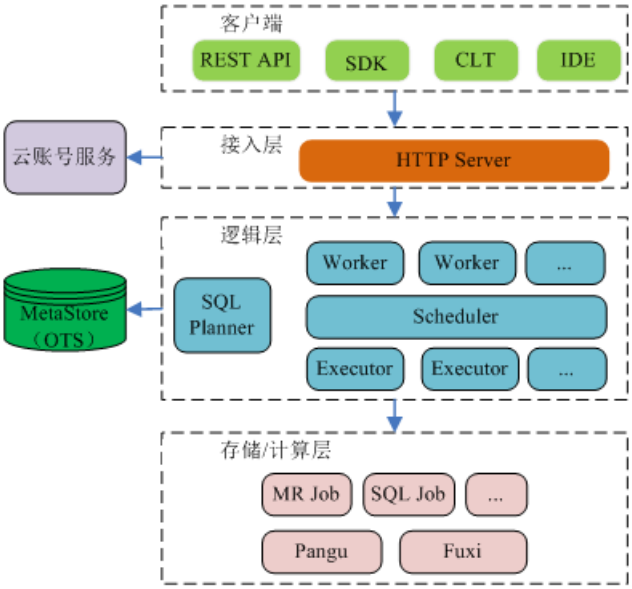
- 采用分布式架构，规模可以根据需要平行扩展。
- 自动存储容错机制，保障数据高可靠性。
- 所有计算在沙箱中运行，保障数据高安全性。
- 以RESTful API的方式提供服务。
- 支持高并发、高吞吐量的数据上传下载。
- 支持离线计算、机器学习两类模型及计算服务。
- 支持基于SQL、Mapreduce、Graph、MPI等多种编程模型的数据处理方式。
- 支持多租户，多个用户可以协同分析数据。
- 支持基于ACL和policy的用户权限管理，可以配置灵活的数据访问控制策略，防止数据越权访问。
- 支持Spark的应用，即Spark on MaxCompute。

6.2 产品架构

6.2.1 功能架构

MaxCompute的功能架构如图 *6-1: MaxCompute*功能架构图所示：

图 6-1: MaxCompute功能架构图



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

MaxCompute的客户端有以下几种形式：

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

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

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

- **Worker处理所有RESTful请求**：包括用户空间（project）管理操作、资源（resource）管理操作、作业管理等，对于SQL、MapReduce、Graph等启动Fuxi任务的作业，会提交Scheduler进一步处理。

- **Scheduler负责instance的调度**：包括将instance分解为task、对等待提交的task进行排序、以及向计算集群的Fuxi master询问资源占用情况以进行流控（Fuxi slot满的时候，停止响应Executor的task申请）。
- **Executor负责启动SQL/MR task**：向计算集群的Fuxi master提交Fuxi任务，并监控这些任务的运行。

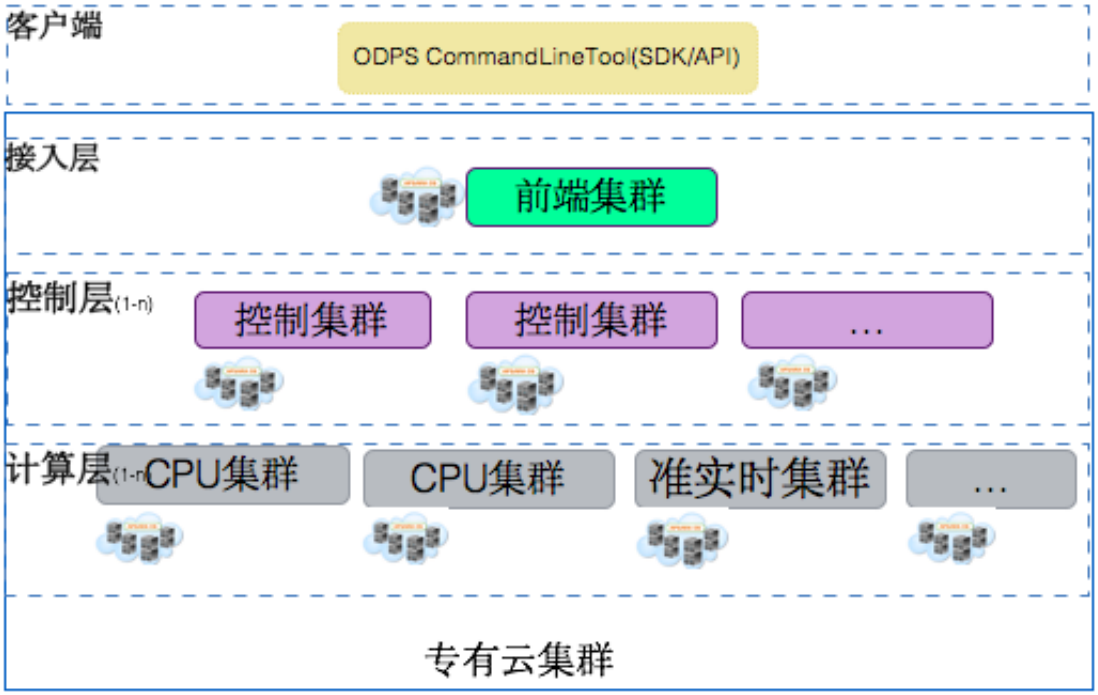
简单的说，当用户提交一个作业请求时，接入层的Web服务器查询获取已注册的Worker的IP地址，并随机选择选择某些Worker发送API请求。Worker将请求发送给Scheduler，由其负责调度和流控。Executor会主动轮询Scheduler的队列，若资源满足条件，则开始执行任务，并将任务执行状态反馈给Scheduler。

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。

6.2.2 系统架构

MaxCompute的系统架构如图 *6-2: MaxCompute系统架构图*所示：

图 6-2: MaxCompute系统架构图



6.3 功能特性

6.3.1 Tunnel

6.3.1.1 Tunnel特点

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

6.3.1.2 Tunnel数据上传下载

Tunnel命令

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

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

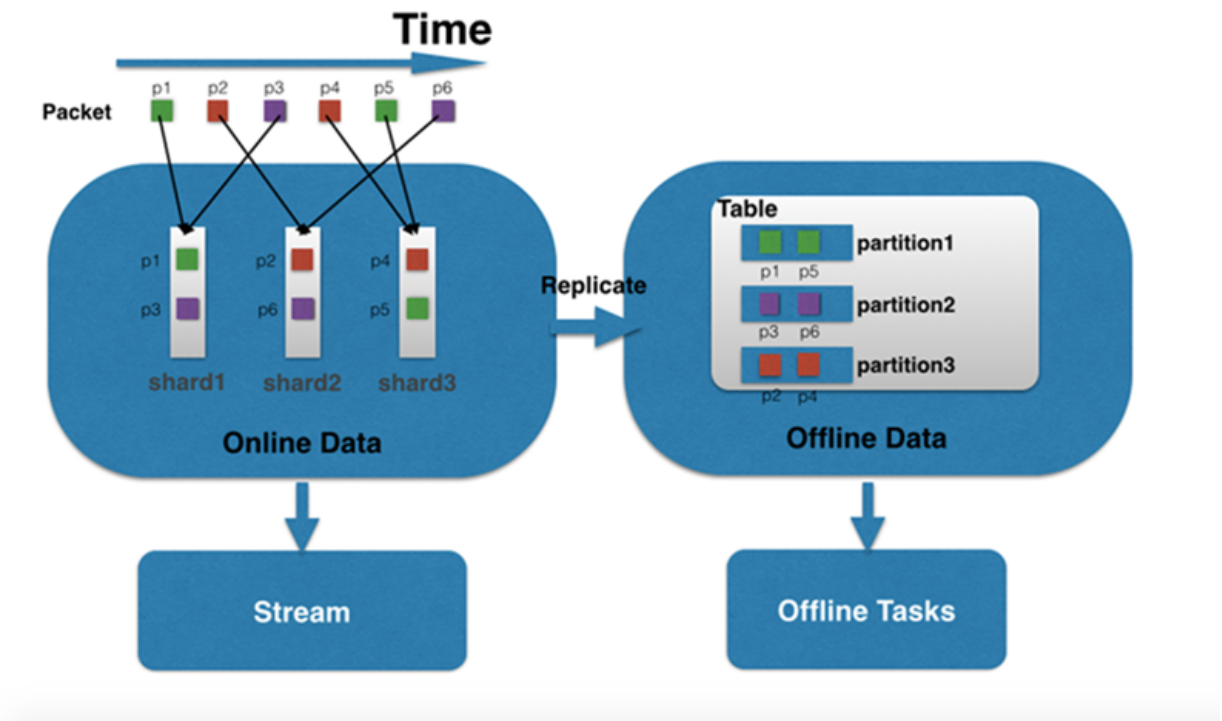
使用说明

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

实时上传

- 小batch上传。
- 高QPS。
- 毫秒级latency。
- 可订阅。

图 6-3: 实时上传



6.3.2 SQL

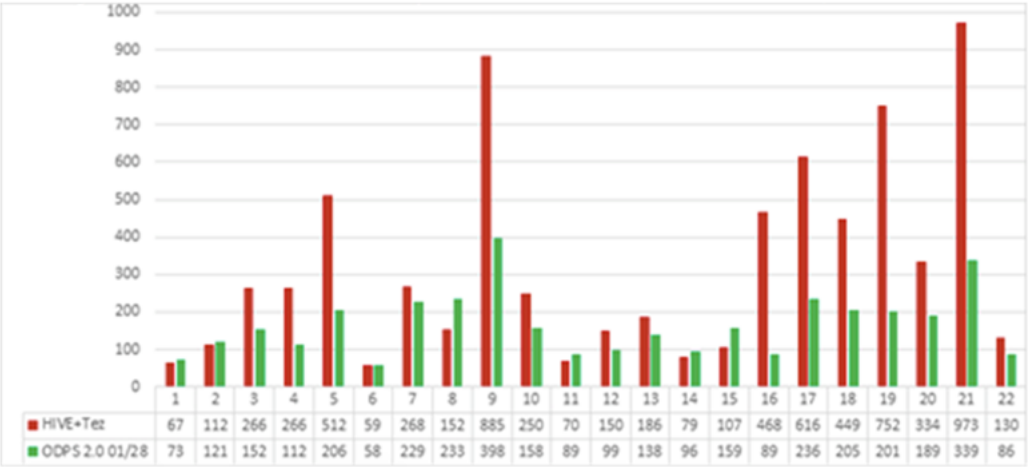
6.3.2.1 SQL特点

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

6.3.2.2 与开源对比

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

图 6-4: MaxCompute 2.0 VS Hive



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

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

图 6-5: MaxCompute 2.0 VS Spark SQL

标准 TPCH 450GB 数据，MaxCompute 2.0 VS Spark SQL (1.6.0 latest release),
MaxCompute 提升 17.8%

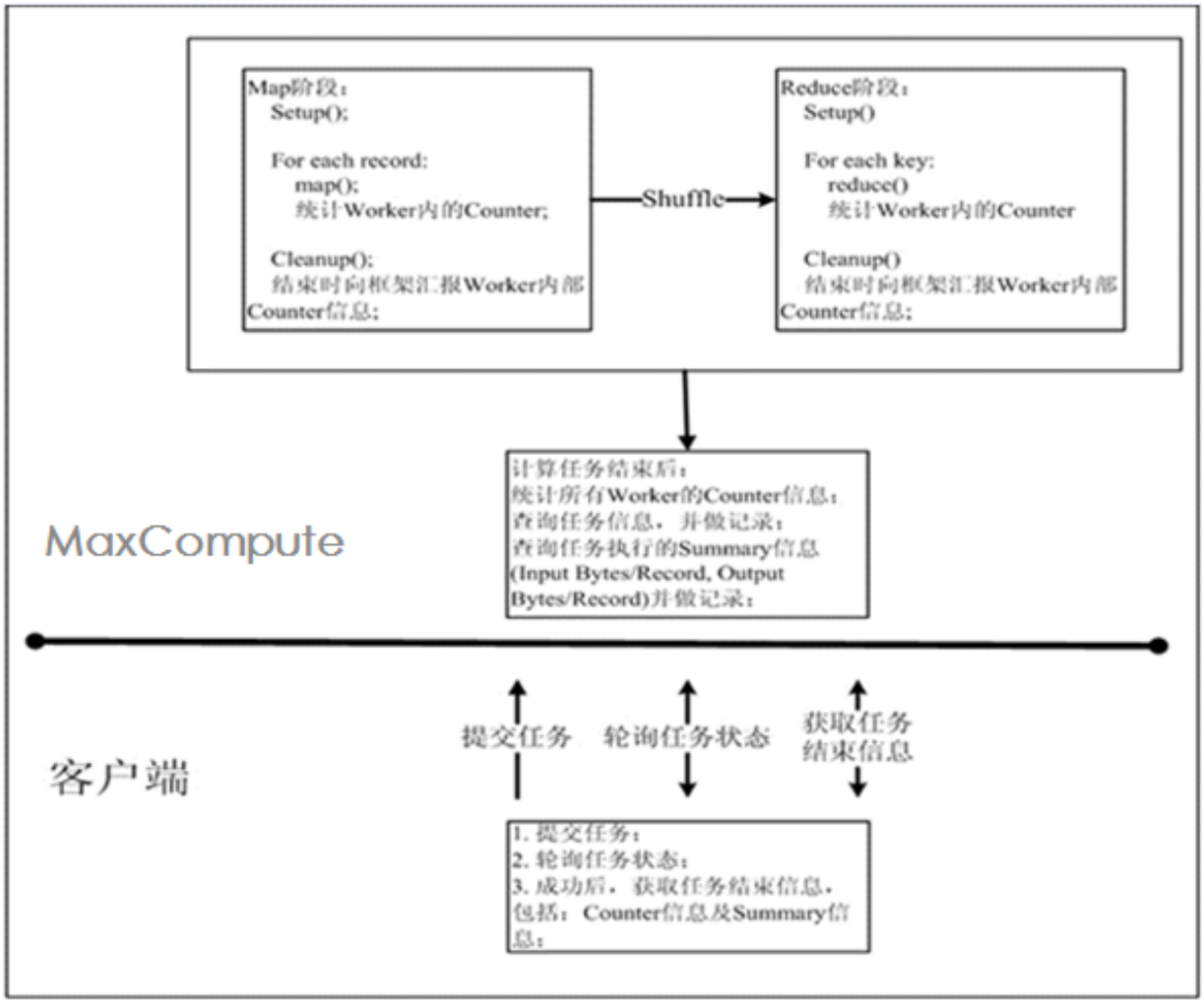
6.3.3 MapReduce

6.3.3.1 MapReduce特点

- 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。
- 编译MR程序需要JDK 1.8环境。
- 受限的沙箱安全环境。

6.3.3.2 MaxCompute MR过程

图 6-6: MR过程



6.3.3.3 Hadoop MR VS MaxCompute MR

表 6-1: Mapper/Reducer

Mapper/Reducer	
Hadoop MapReduce	MaxCompute MapReduce
map（InKey key,InputValue value,OutputCollector<OutKey,OutValue> output,Reporter reporter）	map（long key,Record record,TaskContext context）
Reduce（InKey key,Iterator<InValue> values ,OutputCollector<OutKey,OutValue> output, Reporter reporter）	reduce（IRecord key,Iterator<Record> values, TaskContext context）

图 6-7: MR

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

6.3.4 Graph

6.3.4.1 Graph特点

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

6.3.4.2 Graph关系网络模型

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

社区发现

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

半监督分类

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

孤立点检测

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

关键点挖掘

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

N度关系

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

6.3.5 系统安全

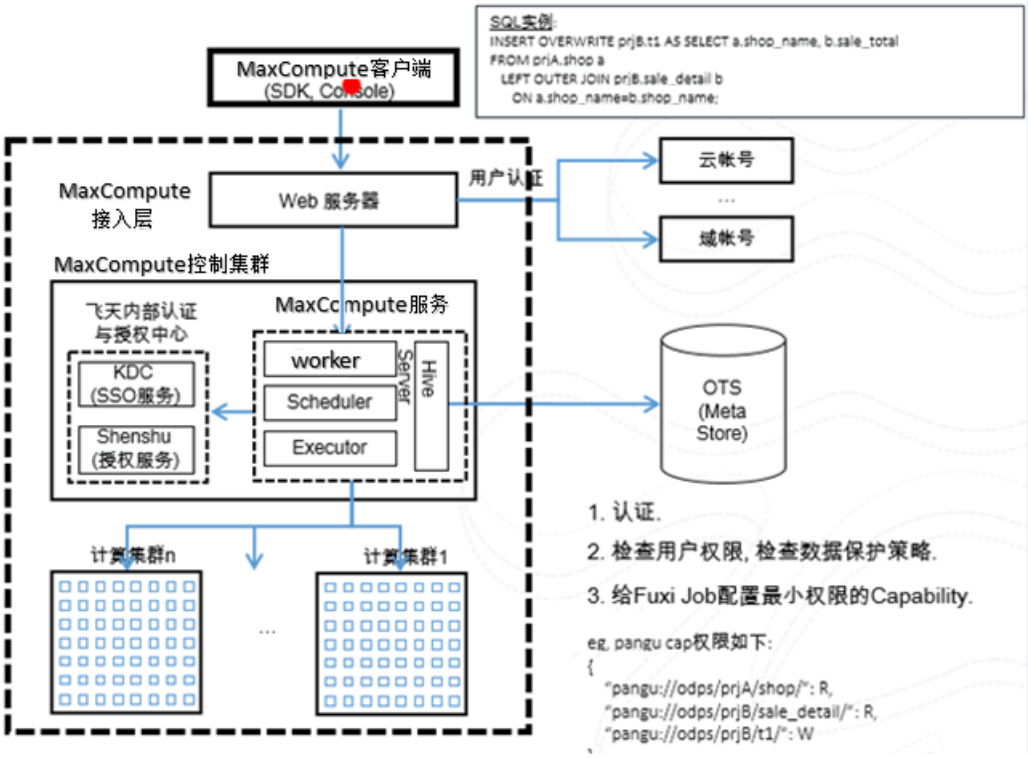
6.3.5.1 安全特点

- 支持多租户的使用场景，同时满足多用户协同、数据共享、数据保密和安全的需要。
- 用户访问需要认证，用户操作需要鉴权，所有操作记录审计日志。

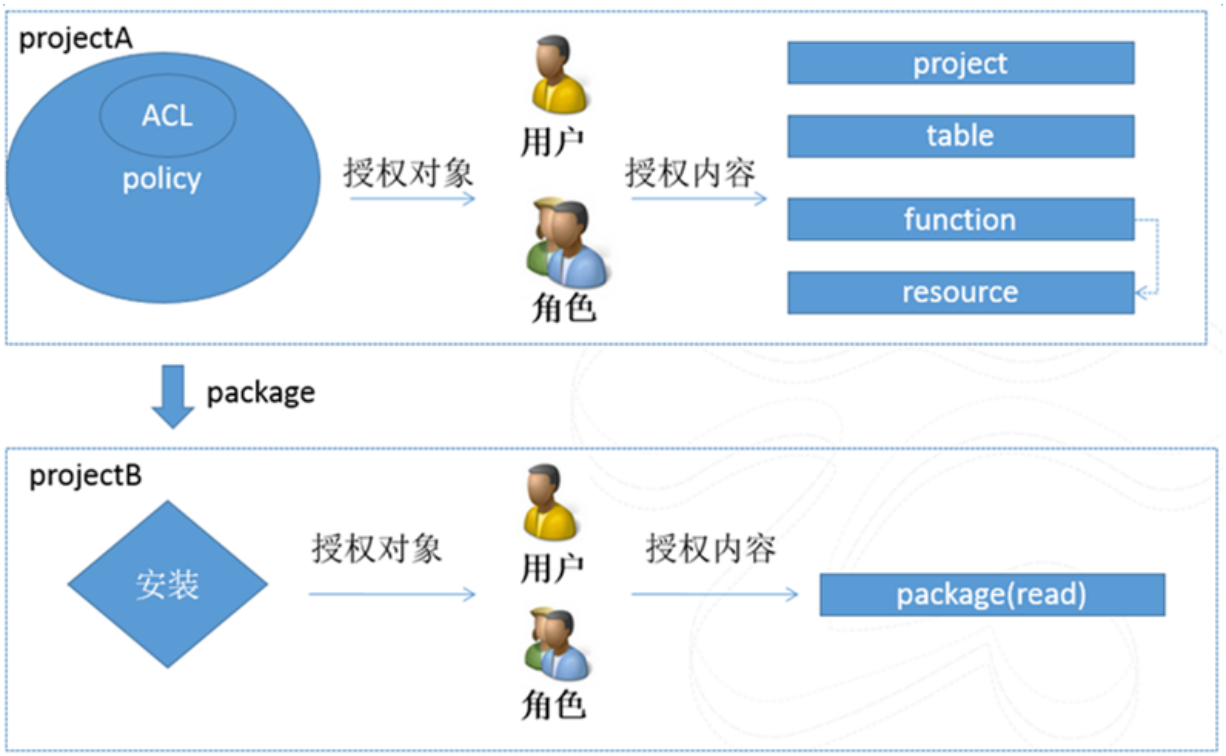
- 支持ACL授权、policy授权、角色授权、跨project授权多种权限管理方法，满足多种场景的需求。
- 同时提供DAC和MAC的安全管理方案，满足对于部分敏感数据的管理需求，可以提供精确到列级别的数据管理。
- 对于安全等级较高的数据，提供项目保护模式，防止数据泄露。
- 所有计算在受限的沙箱中运行，多层次的应用沙箱、系统沙箱配合请求鉴权管理机制，保证数据的安全。

6.3.5.2 安全架构

图 6-8: 安全架构



6.3.5.3 权限管理模型



6.3.5.4 ACL授权

- 对已存在的对象和用户进行授权管理。
- 用户必须存在于project中。
- 角色（role）是一组权限的集合。
- 系统预设admin角色。
- 对象删除时，所有相关的ACL也被删除。

```
grant CreateTable, CreateInstance, List on project myprj to user Alice;
-- 增加用户权限。
```

```
grant worker to aliyun$abc@aliyun.com;
-- 加用户进角色。
```

```
revoke CreateTable on project myprj from user Alice;
```

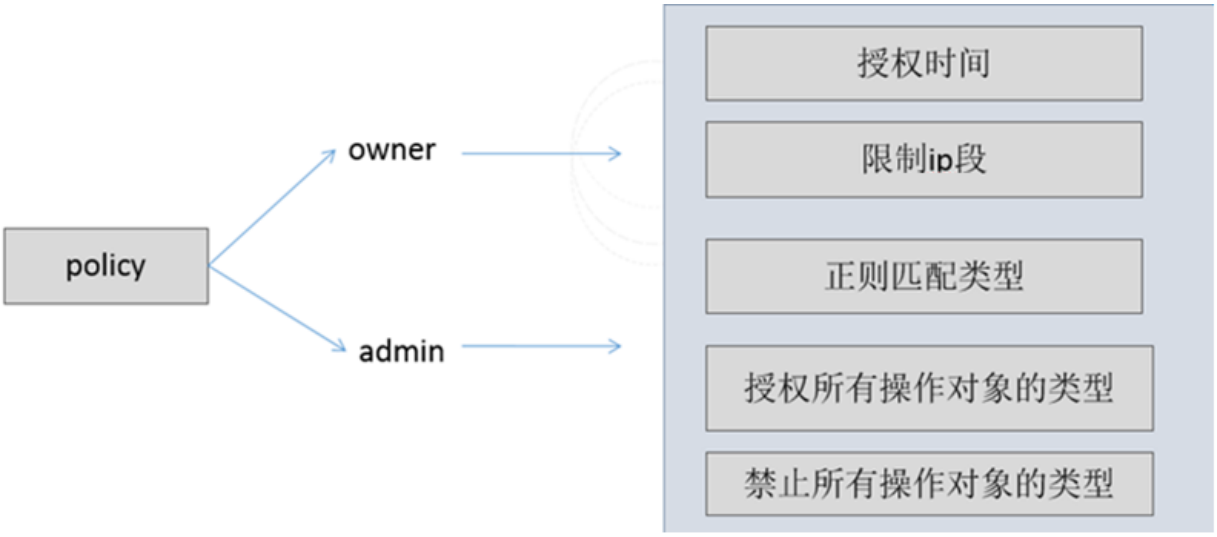
```
-- 删除用户权限。
```

6.3.5.5 Policy授权

Policy授权策略描述了授权的具体内容，它是一种规则，例如：允许所有用户读所有开发环境的表、允许A用户从某个IP访问数据；当管理的数据对象和用户数非常多时，可以极大的减轻管理负担；存在于project和角色上，分为project Policy与role Policy，不会随表的删除而消失。

```
GET POLICY
PUT POLICY <policyFile>;
GET POLICY ON ROLE <roleName>;
PUT POLICY <policyFile> ON ROLE <roleName>;
```

图 6-9: policy授权




```
{
"Version": "1",
"Statement":
 [{
 "Effect":"Allow",
 "Principal":"ALIYUN$alice@aliyun.com",  "Action":["odps:CreateTable","odps:CreateInstance","
odps:List"],  "Resource":"acs:odps:*:projects/test_project", "Condition":{    "DateLessThan": { "
acs:CurrentTime":"2013-11-11T23:59:59Z" },
  "IpAddress": { "acs:SourceIp":"10.32.180.0/23" }
 }
 },
{
 "Effect":"Deny",
 "Principal":"ALIYUN$alice@aliyun.com",
 "Action":"odps:Drop", "Resource":"acs:odps:*:projects/test_project/tables/*"
}
```

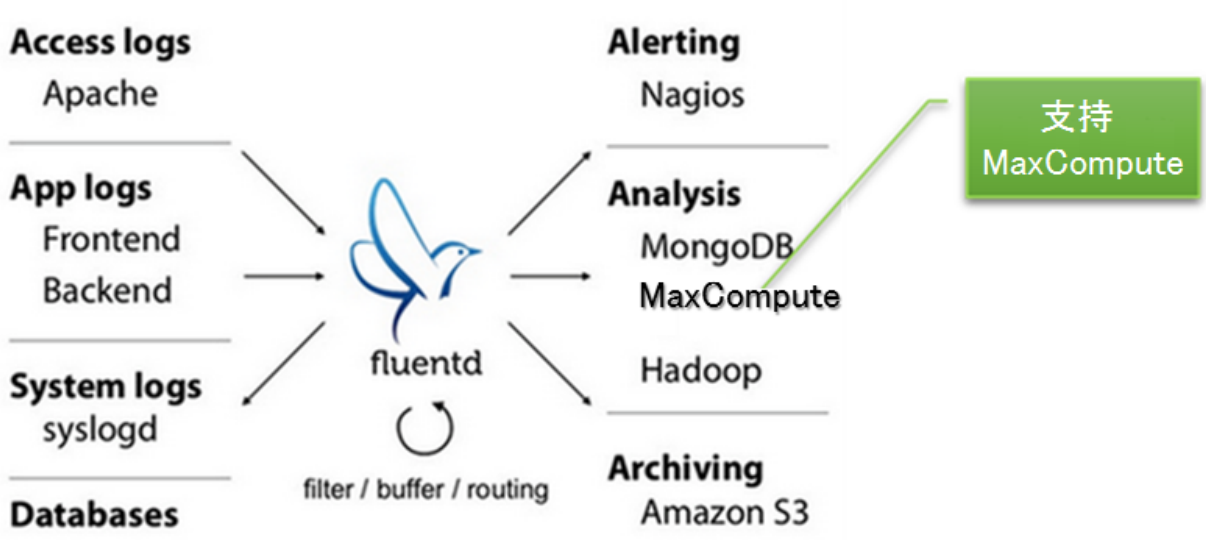
```
]}
```

6.3.6 开源生态

6.3.6.1 日志导入工具-Fluentd

官网：*http://www.fluentd.org*

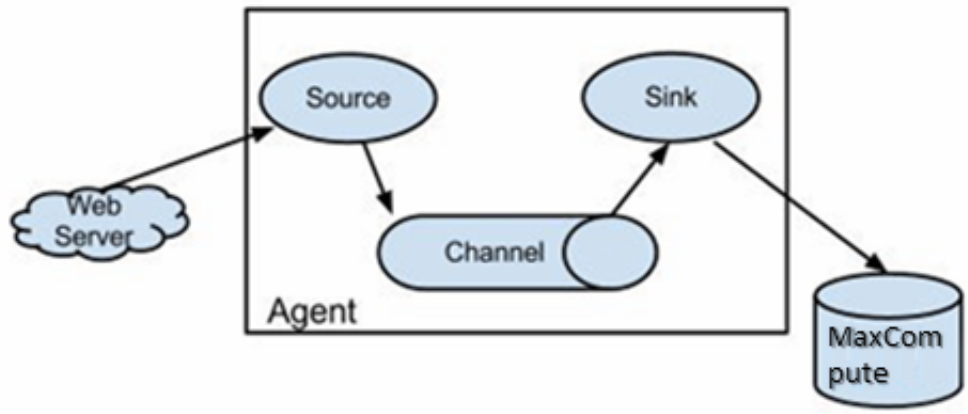
图 6-10: fluented



6.3.6.2 日志导入工具-Flume

官网：*http://flume.apache.org/*

图 6-11: flume



6.3.6.3 开源代码

- MaxCompute R插件：*https://github.com/aliyun/aliyun-odps-r-plugin*
- MaxCompute Sqoop：*https://github.com/aliyun/aliyun-odps-sqoop*
- MaxCompute ogg：*https://github.com/aliyun/aliyun-odps-ogg-plugin*
- MaxCompute eclipse插件：*https://github.com/aliyun/aliyun-odps-eclipse-plugin*
- MaxCompute JDBC Driver：*https://github.com/aliyun/aliyun-odps-jdbc*
- MaxCompute Python SDK：*https://github.com/aliyun/aliyun-odps-python-sdk*
- MaxCompute Java SDK：*https://github.com/aliyun/aliyun-odps-java-sdk*

6.4 产品优势

国内唯一的大数据云服务平台，真正的数据分享平台

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

MaxCompute规模

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

6.4.1 存储

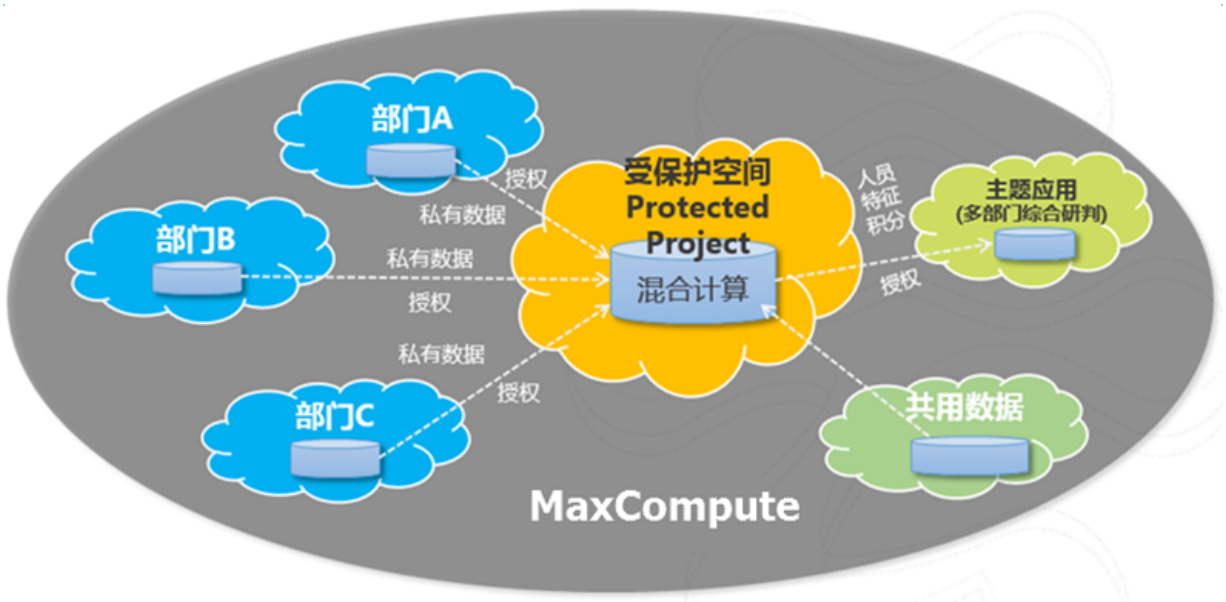
- MaxCompute处理数据大多数存储在结构化的二维表中（Table）。
  - 表隶属于project。
  - 表可以进行分区。
  - 表中的数据类型有bigint，double，boolean，datetime，decimal，string，tinyint，smallint，int，float，varchar，binary，timestamp，array，map以及struct。
- 数据由盘古存储系统管理，自动化的多副本存储策略提高数据可用性，屏蔽底层硬件故障。
- 列存储结构，压缩存储。
- 内置的数据生命周期管理策略。

- 基于存储配额的（Quota）多租户管理机制。

6.4.2 MaxCompute多租户机制

敏感数据共享式服务：数据不搬家，可用不可见。

图 6-12: 多租户机制

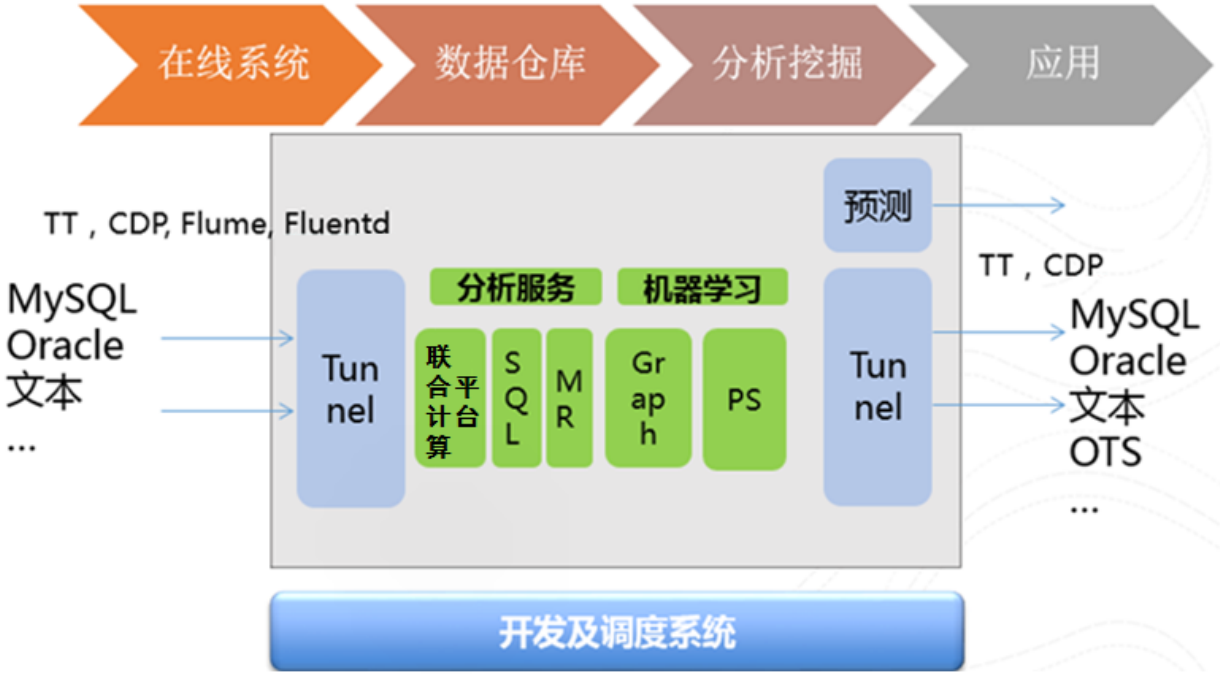


6.4.3 MaxCompute多集群支持

图 6-13: 多集群双活/灾备

6.4.4 数据处理流程

图 6-14: 数据处理流程



6.5 典型应用

海量数据的存储

只有大量的数据是不够的，还要将数据集中起来才能更好的发挥作用。

海量数据的计算

通过SQL，MR， Graph等方式，可以在一个任务中轻松处理TB级别的数据，并且有专门的分布式矩阵运算、数据挖掘算法（PAI）等对数据进行流式处理引擎及实时数据分析。

多组织间的数据交换

不仅能满足一个组织内部的数据管控，也能用于多个组织间的数据隔离和交换。

开箱即用的服务

用户不用关注基础设施管理而是关注自己业务内容，MaxCompute本身提供各服务的一致性与连续性。

说明：

MaxCompute并非在线OLTP关系数据库，它不支持事务，没有主键约束。

6.6 基本概念

项目空间

项目空间是MaxCompute的基本组织单元，它类似于传统数据库的Database或Schema的概念，是进行多用户隔离和访问控制的主要边界。一个用户可以同时拥有多个项目空间的权限。通过安全授权，可以在一个项目空间中访问另一个项目空间中的对象，例如：表（Table），资源（Resource），函数（Function），实例（Instance）。

用户可以通过Use Project命令进入一个项目空间，例如：

```
use my_project
-- 进入一个名为my_project的项目空间。
```

运行此命令后，用户会进入一个名为my_project的项目空间，从而可以操作该项目空间下的对象，例如：表（Table），资源（Resource），函数（Function），实例（Instance）等，而不需要关心操作对象所在的项目空间。Use Project是MaxCompute客户端提供的命令，相关命令的具体说明请参见《大数据计算服务*(MaxCompute)*用户指南》中的**常用命令**章节 。

表

表是MaxCompute的数据存储单元，它在逻辑上也是由行和列组成的二维结构，每行代表一条记录，每列表示相同数据类型的一个字段，一条记录可以包含一个或多个列，各个列的名称和类型构成这张表的Schema 。阿里云数加平台的数据管理模块可以对MaxCompute表进行新建、收藏、修改数据生命周期管理、修改表结构和数据表/资源/函数权限管理审批等操作，详情请参见《大数据计算服务*(MaxCompute)*用户指南》中的**基本介绍**章节，以对数据管理模块功能进行深入了解。

MaxCompute的表格分两种类型：外部表及内部表。

对于内部表，所有的数据都被存储在MaxCompute中，表中的列可以是MaxCompute支持的任意种数据类型。MaxCompute中的各种不同类型计算任务的操作对象（输入、输出）都是表 。用户可以创建表，删除表以及向表中导入数据。

分区

分区表指的是在创建表时指定分区空间，即指定表内的某几个字段作为分区列。在大多数情况下，用户可以将分区类比为文件系统下的目录。MaxCompute将分区列的每个值作为一个分区（目录）。用户可以指定多级分区，即将表的多个字段作为表的分区，分区之间类似多级目录的关系。在使用数据时如果指定了需要访问的分区名称，则只会读取相应的分区，避免全表扫描，提高处理效率，降低费用。

示例：

```
create table src (key string, value bigint) partitioned by (pt string);
-- 目前，MaxCompute仅承诺String类型分区。
select * from src where pt='20171201';
-- 正确使用方式。MaxCompute在生成查询计划时只会将'20171201'分区的数据纳入输入中。
select * from src where pt = 20171201;
-- 错误的使用方式。在这样的使用方式下，MaxCompute并不能保障分区过滤机制的有效性。pt是String类型，当String类型与Bigint(20171201)比较时，MaxCompute会将二者转换为Double类型，此时有可能会有精度损失。
```

数据类型

MaxCompute 表中的列必须是下列描述的任意一种类型，各种类型的描述及取值范围包括：

类型	常量定义	描述	取值范围
Tinyint	1Y,-127Y	8位有符号整形。	-128 ~ 127
Smallint	32767S,-100S	16位有符号整形。	-32768 ~ 32767
Int	1000,-15645787	32位有符号整形。	-2^{31} ~ $2^{31}-1$
Bigint	100000000000L,-1L	64位有符号整形。	$-2^{63}+1$ ~ $2^{63}-1$
String	"abc","bcd","alibaba", 'inc'	字符串，支持UTF-8编码。其他编码的字符行为未定义。	单个String列最长允许8MB
Float	无	32位二进制浮点型。	/
Boolean	True,False	布尔型。	True或False
Double	3.1415926 1E+7	64位二进制浮点型。	-1.0 10 308 ~ 1.0 10308
Datetime	Datetime '2017-11-11 00:00:00'	日期时间类型。使用东八区时间作为系统标准时间。	0001-01-01 00:00:00 000 ~ 9999- 12-31 23: 59:59 999
Decimal	3.5BD, 99999999999.9999999BD	10进制精确数字类型。	整形部分：$-10^{36}+1$ ~ $10^{36}-1$ 小数部分：精确到10^{-18}
Varchar	无	变长字符类型，n为长度。	1 ~ 65535
Binary	无	二进制数据类型。	单个Binary列最长允许8MB

类型	常量定义	描述	取值范围
Timestamp	Timestamp '2017-11-11 00:00:00.123456789'	与时区无关的时间戳类型。	0001-01-01 00:00:00 000000000 ~ 9999 - 12-31 23:59:59 999999999

说明：

- 上述的各种数据类型均可为NULL。
- 对于Int类型常量，如果超过Int类型取值范围，会转换为Bigint类型；如果超过Bigint类型取值范围，会转换为Double类型。MaxCompute在未设定**odps.sql.type.system.odps2**为**true**的情况下，保留此转换，但会报告一个警告，提示Int类型被当作Bigint类型处理了。如果用户的脚本有此种情况，建议全部改写为Bigint类型，避免混淆。
- Varchar类型常量可通过String类型常量的隐式转换表示。
- String类型常量支持连接，例如‘abc’ ‘xyz’会解析为‘abcxyz’，不同部分可以写在不同行上。

MaxCompute支持的复杂数据类型的定义及构造方法，如下表所示：

类型	定义方法	构造方法

Array	array< int >;
	array< struct< a:int, b:string >>
		array(1, 2, 3);
		array(array(1, 2);
		array(3, 4))

Map	map< string, string >; map< smallint, array< string>>	map("k1", "v1", "k2", "v2"); map(1S, array('a', 'b'), 2S, array('x', 'y'))
Struct	struct< x:int, y:int>; struct< field1:bigint, field2: array< int>, field3:map< int, int >>	named_struct('x', 1, 'y', 2); named_struct('field1', 100L, ' field2', array(1, 2), 'field3', map (1, 100, 2, 200)

资源

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

- SQL UDF：用户在编写UDF后，需要将编译好的jar包以资源的形式上传到MaxCompute。运行这个UDF时，MaxCompute会自动下载这个jar包，获取用户代码，运行UDF，无需用户干预。上传jar包的过程就是在MaxCompute上创建资源的过程，这个jar包是MaxCompute资源的一种。
- MapReduce：用户编写MapReduce程序后，将编译好的jar包作为一种资源上传到MaxCompute。运行MapReduce作业时，MapReduce框架会自动下载这个jar资源，获取用户代码。用户同样可以将文本文件以及MaxCompute中的表作为不同类型的资源上传到MaxCompute。用户可以在UDF及MapReduce的运行过程中读取、使用这些资源。MaxCompute提供了读取、使用资源的接口。详细示例请参见《大数据计算服务*(MaxCompute)*用户指南》中的**使用资源示例**章节以及《大数据计算服务*(MaxCompute)*用户指南》中的**UDTF使用说明**章节的相关描述。需要注意的是，MaxCompute的自定义函数（UDF）或MapReduce对资源的读取有一定的限制，请参见《大数据计算服务*(MaxCompute)*用户指南》中的**应用限制**章节。

MaxCompute资源的类型包括：

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

函数

MaxCompute为用户提供了SQL计算功能，用户可以在MaxCompute SQL中使用系统的内建函数完成一定的计算和计数功能。但当内建函数无法满足要求时，用户可以使用MaxCompute提供的Java编程接口开发自定义函数（User Defined Function，以下简称UDF）。自定义函数（UDF）又可以进一步分为标量值函数（UDF），自定义聚合函数（UDAF）和自定义表值函数（UDTF）三种。

用户在开发完成UDF代码后，需要将代码编译成jar包，并将此jar包以jar资源的形式上传到MaxCompute，最后在MaxCompute中注册此UDF。在使用UDF时，只需要在SQL中指明UDF的函数名及输入参数即可，使用方式与MaxCompute提供的内建函数相同。

任务

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

对于用户提交的大多数任务，特别是计算型任务，例如：DML语句、MapReduce等，MaxCompute会对其进行解析，得出任务的执行计划。执行计划是由具有依赖关系的多个执行阶段（Stage）构成的。目前，执行计划逻辑上可以被看做一个有向图，图中的点是执行阶段，各个执行阶段的依赖关系是图的边。MaxCompute会依照图（执行计划）中的依赖关系执行各个阶段。在同一个执行阶段内，会有多个进程，也称之为Worker，共同完成该执行阶段的计算工作。同一个执行阶段的不同Worker只是处理的数据不同，执行逻辑完全相同。计算型任务在执行时，会被实例化，用户可以操作这个实例（Instance）的信息，例如：获取实例状态（Status Instance）、终止实例运行（Kill Instance）等。

另一方面，部分MaxCompute任务并不是计算型的任务，例如：SQL中的DDL语句 ，这些任务本质上仅需要读取、修改MaxCompute中的元数据信息。因此，这些任务无法被解析出执行计划。

任务实例

在MaxCompute中，部分任务在执行时会被实例化，以MaxCompute实例（下文简称实例或Instance）的形式存在。实例会经历运行（Running）及结束（Terminated）两个阶段。运行阶段的状态为Running（运行中），而结束阶段的状态将会是Success（成功），Failed（失败）或Cancelled（被取消）。用户可以根据运行任务时MaxCompute给出的实例ID查询、改变任务的状态，例如：

```
status <instance_id>;
-- 查看某实例的状态。
```

```
kill <instance_id>;
-- 停止某实例，将其状态设置为Cancelled。
```

资源配额

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

6.7 高级应用

6.7.1 Spark on MaxCompute

6.7.1.1 什么是Spark on MaxCompute

Spark on Maxcompute是阿里云开发的基于Maxcompute平台无缝运行使用spark的方案，其很大程度上丰富了Maxcompute产品的功能体系。**Spark on MaxCompute**为用户提供原生Spark的使用

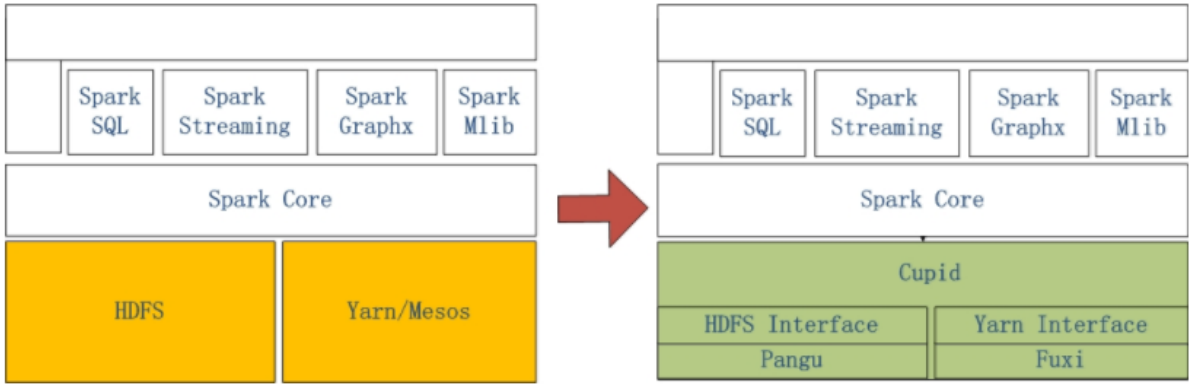
体验以及Spark原生的组件及API；提供存取MaxCompute数据源的能力；提供多租户场景更好的安全能力；提供管控平台，让Spark作业与MaxCompute作业共享资源、存储、用户体系，保证高性能和低成本。 Spark配合MaxCompute产品能够构建更加完善高效的数据处理解决方案，同时社区Spark的应用可以无缝的运行在**Spark on MaxCompute**之上。

6.7.1.2 产品架构

6.7.1.2.1 产品架构

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

图 6-15: Spark on MaxCompute与原生Spark架构比对



说明：

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

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

- MaxCompute控制集群使用CupidTask实例拉起Spark的Driver，后续Driver通过Yarn API的兼容层去向统一资源管理器FuxiMaster申请资源。
- MaxCompute控制集群管理Spark运行实例消耗的用户quota、Spark实例的生命周期、可访问数据源的权限管理。
- 在MaxCompute的计算集群通过父子进程的方式来拉起Spark的Driver以及Executor，并且让Spark的代码运行在MaxCompute的安全沙箱中，从而保证多租户场景的安全性。

- 系统通过MaxCompute的Proxyserver来提供Spark原生的UI界面，同时用户可以使用MaxCompute相关管控组件来管理作业信息。

6.7.1.2.2 产品特点

处理MaxCompute以及非结构化数据源

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

友好的使用体验及管控

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

扩展性

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

6.7.1.2.3 产品功能

Spark on MaxCompute的产品功能如下所示。

表 6-2: 产品功能列表

分类	特性	说明
分布式集群	集群部署 集群监控	提供运维平台页面监控集群、节点情况。
数据处理组件	可使用Spark Sql、Mllib、Graphx、Streaming组件	提供原生Spark的组件。
作业管控	资源统一管理、生命周期管理、权限验证	通过兼容Yarn API来实现。

分类	特性	说明
数据源	非结构化数据 MaxCompute的table数据源	具有MaxCompute上sql以及MapReduce同样的数据处理性能。
安全管理	用户身份认证、数据鉴权、多租户作业隔离	通过权限认证以及安全沙箱加强Spark的安全能力。

6.7.1.3 产品优势

Spark on MaxCompute的主要优势如下所示：

1. 支持Spark完整生态，使用体验跟开源保持一致。
2. 跟MaxCompute完全整合，做到资源统一、数据统一和安全统一。
3. 兼有Spark和飞天的优势，既有Spark的灵活性、易用性；又有飞天的高可用、高扩展性和稳定性。
4. 多租户，既保证物理机器的高性能，也能实现大集群的统一资源调度，从而降低成本。

6.8 典型应用场景

本章节将简要的介绍一些MaxCompute的典型实战场景。

6.8.1 典型实战场景1：使用成本低，数据上云周期短

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

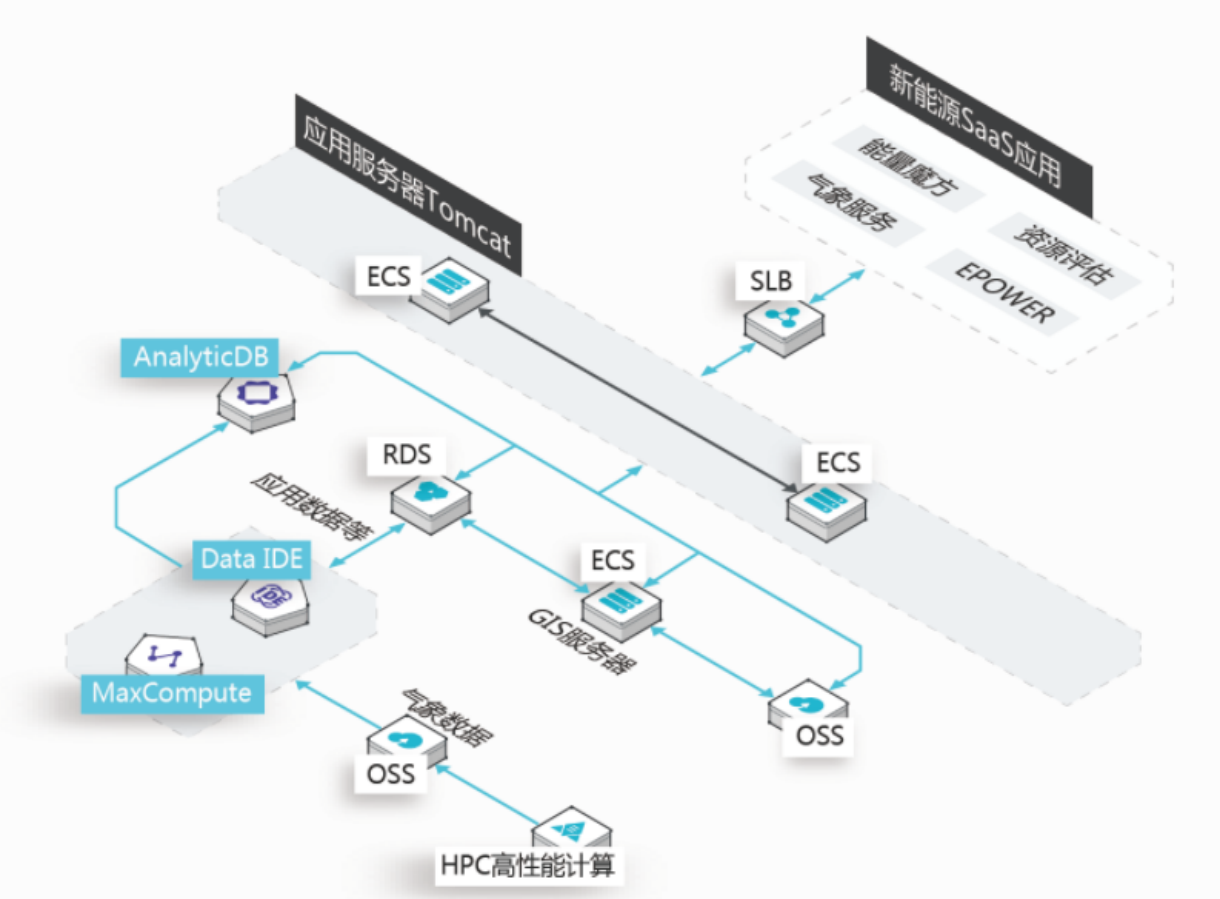
目标达成：3个月内业务全面交付云端，数据处理时间不到原来自建方式的1/3，并确保云上新能源电力数据安全无忧。

客户收益：

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

应用架构图：

图 6-16: 应用架构图



架构解读：

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

6.8.2 典型实战场景2：提升开发效率，降低存储和计算成本

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

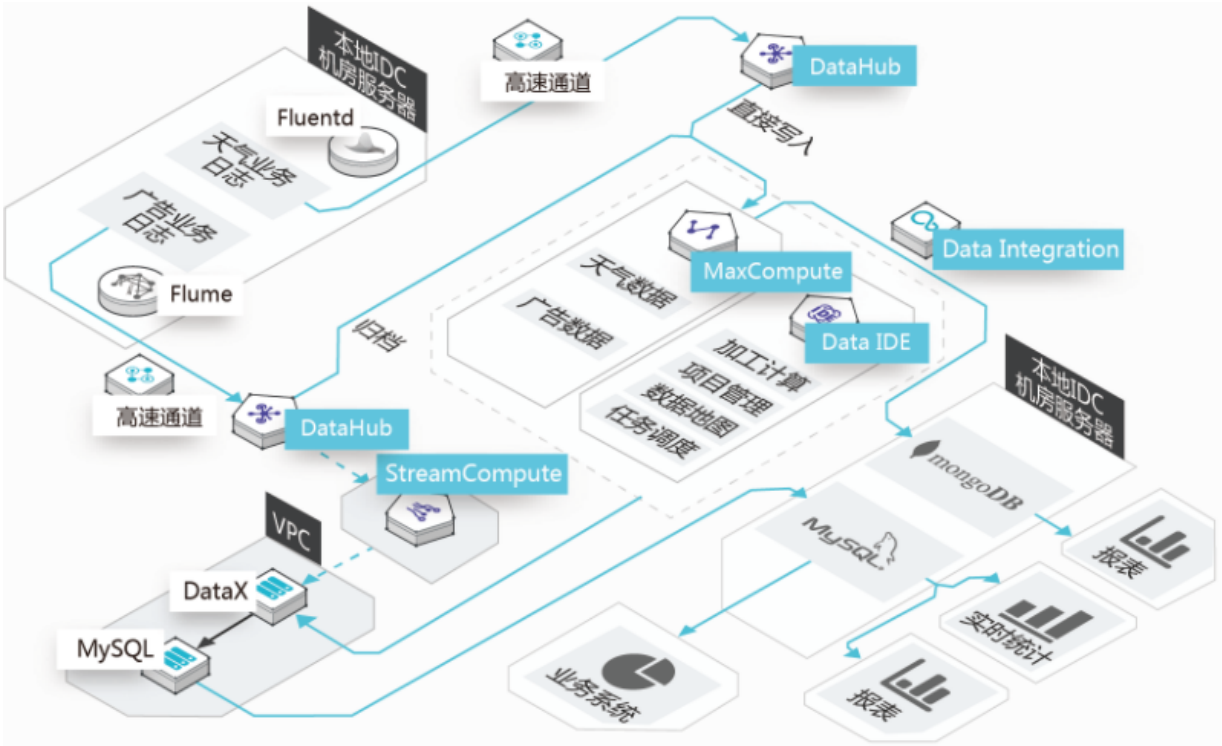
目标达成：该互联网公司日志分析业务迁移到MaxCompute后，开发效率提升了超过5倍，存储和计算费用节省了70%，每天处理分析2TB的日志数据，更高效的赋能其个性化运营策略。

客户收益：

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

应用架构图：

图 6-17: 应用架构图



架构解读：

1. 天气查询业务日志分析如架构图上半部分所示。
2. 广告业务日志分析如架构图下半部分所示。

6.8.3 典型实战场景3：盘活海量数据，实现百万用户精细化运营

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

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

客户收益：

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

应用架构图：

图 6-18: 应用架构图

架构解读：

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

6.8.4 典型实战场景4：大数据精准营销

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

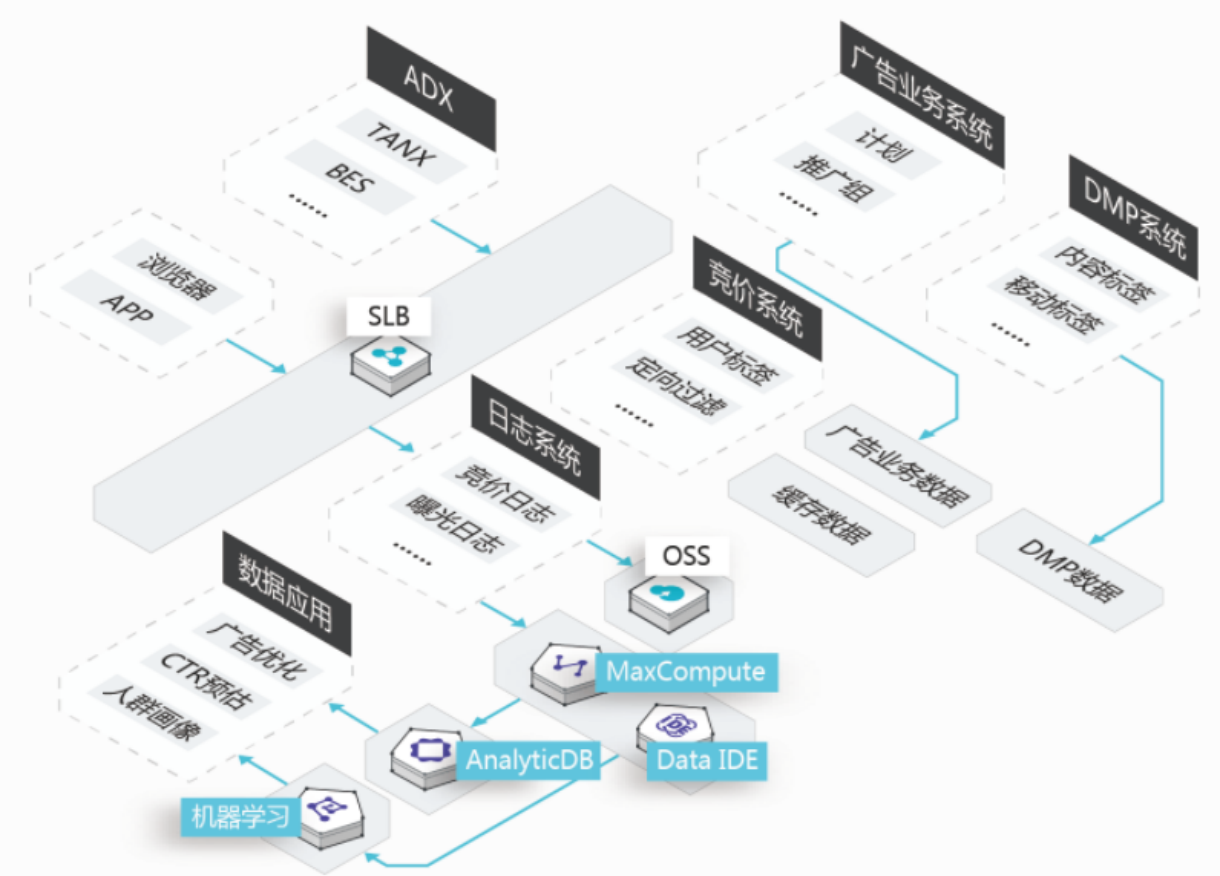
目标达成：该企业基于MaxCompute，搭建了核心的大数据精准营销平台，所有的日志数据存储在MaxCompute并通过Data IDE进行离线调度和分析。

客户收益：

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

应用架构图：

图 6-19: 应用架构图



架构解读：

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

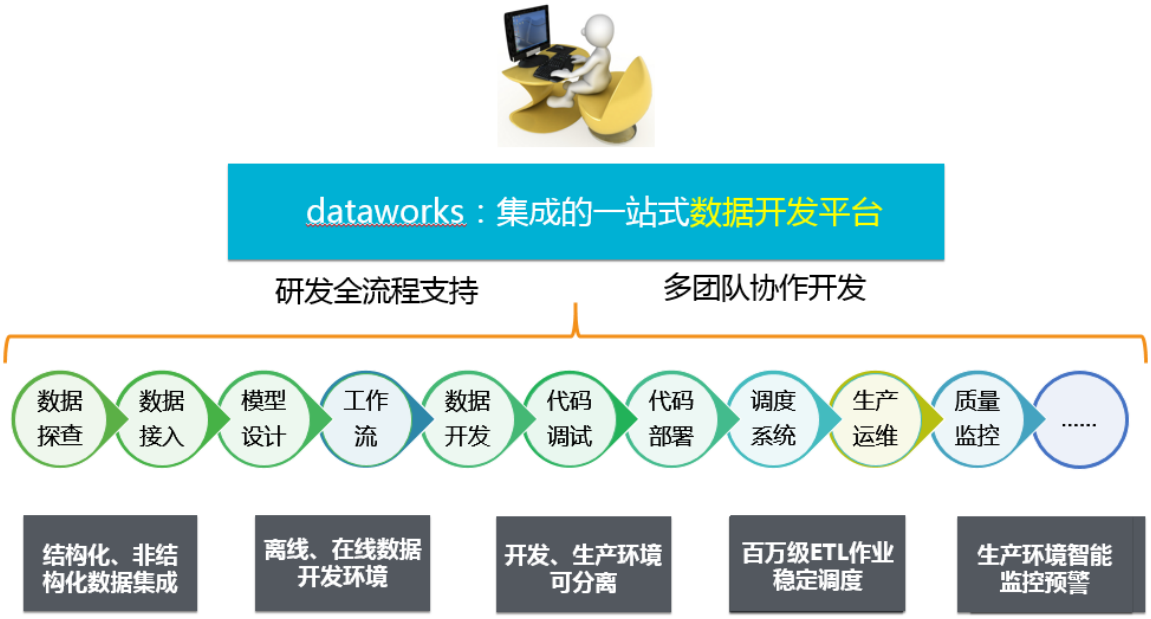
7 大数据开发套件

7.1 什么是DataWorks

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

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

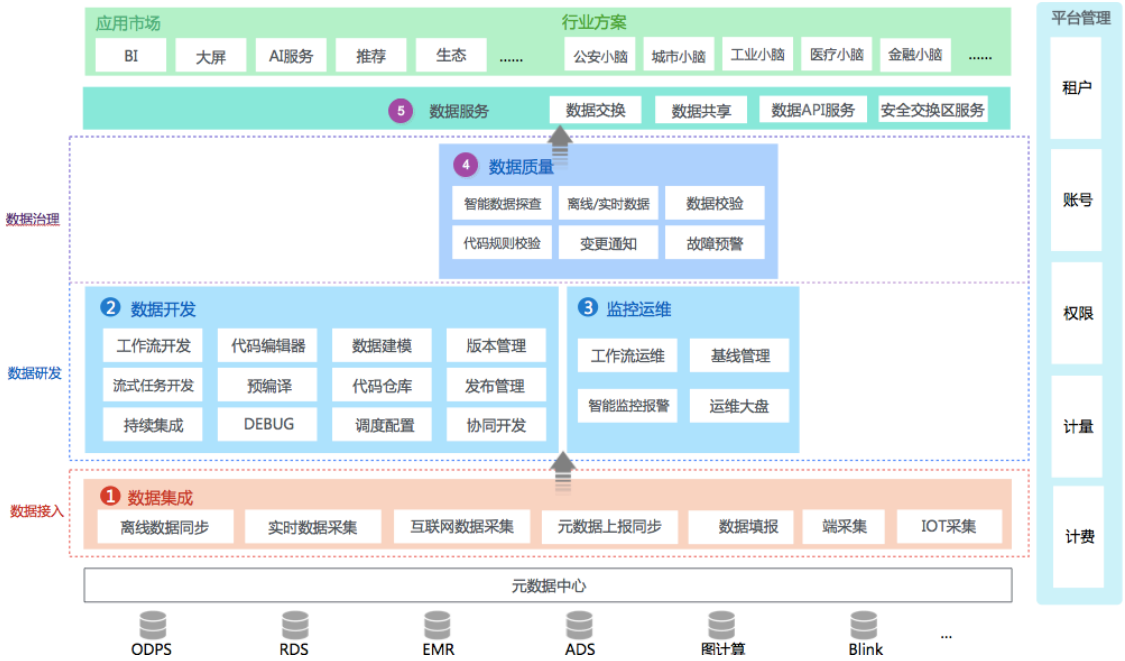
图 7-1: DataWorks概述



7.2 产品架构

7.2.1 功能架构

图 7-2: 功能架构图



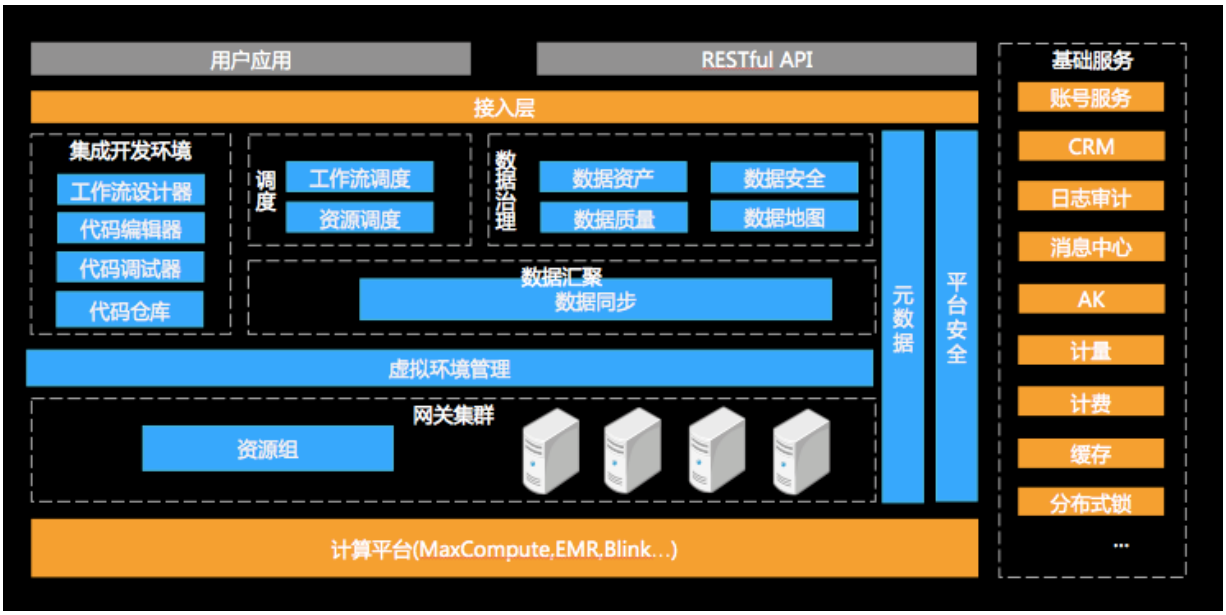
DataWorks提供以下四大服务模块：

- 数据集成：异构数据集成，将海量的数据从各种源系统汇集到大数据平台。
- 数据开发：数据仓库设计和ETL开发过程。
- 监控运维：ETL线上作业的运维监控。
- 数据质量：数据质量探查、监控、校验和评分体系。

7.2.2 系统架构

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

图 7-3: 数据资源平台架构图



7.2.3 安全架构

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

7.2.4 多租户模型

数据资源平台拥有自己的多租户权限模型。

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

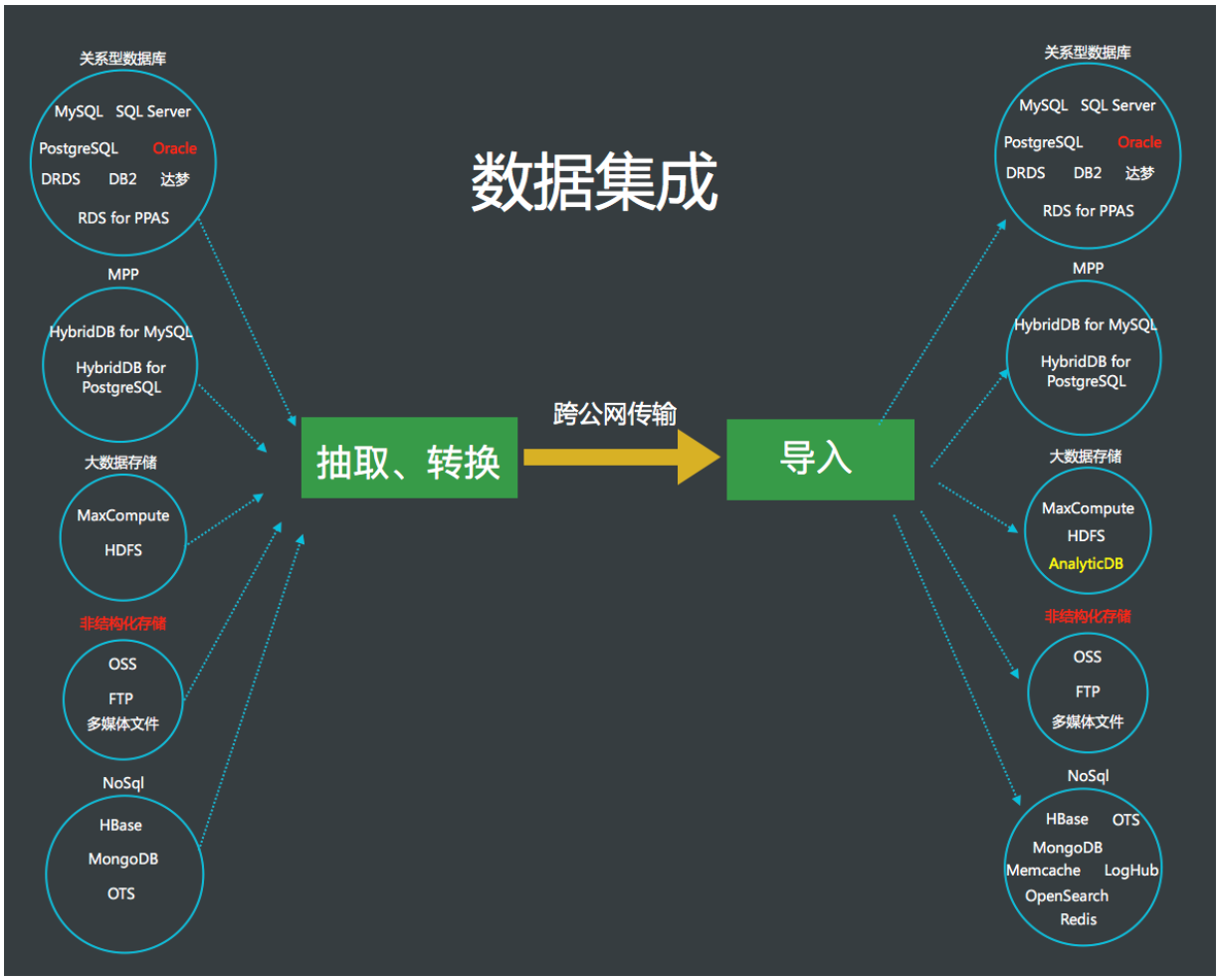
7.3 功能特性

7.3.1 数据集成

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

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

图 7-4: 数据集成



7.3.1.1 支持多种数据通道

7.3.1.1.1 元数据信息同步

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

7.3.1.1.2 关系型数据库同步服务

支持MySQL、SQL Server、Oracle、DRDS、postgresql、DB2、RDS for PPAS的读写操作索。

7.3.1.1.3 NoSQL数据库同步服务

支持HBase、MongoDB、Table Store的读写操作。

7.3.1.1.4 MPP数据库同步服务

支持HybridDB for MySQL、HybridDB for postgresql的读写操作。

7.3.1.1.5 大数据数据库同步服务

支持MaxCompute、HDFS的读写操作，以及ADS的写操作。

7.3.1.1.6 非结构化存储同步服务

支持OSS、FTP的读写操作。

说明：

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

7.3.1.2 数据流入管控

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

7.3.1.3 传输速度

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

7.3.1.4 控制友好

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

7.3.1.5 同步插件

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

7.3.1.6 跨网络传输

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

说明：

对长链路传输通过协议加速能更高效、稳定的传输大批量数据。

7.3.2 数据开发

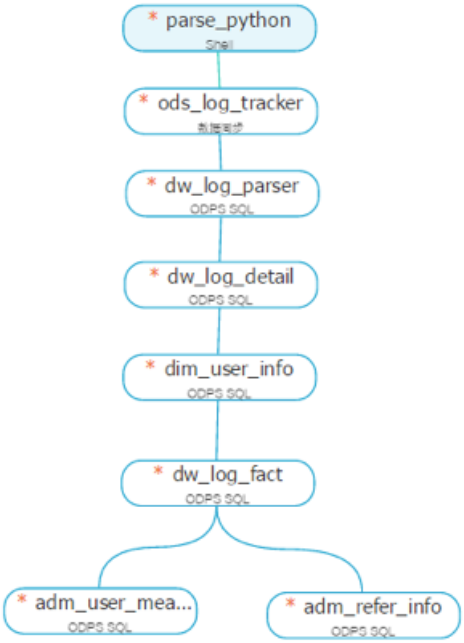
当底层数据进行聚合后，数据仍然出于零散的状态，数据是无法直接为上层智能算法和DI应用提供对应数据的，此时需要对数据进行汇聚加工。数据管理和开发人员需要在数据资源平台建立对应的数据中心，进行对应数据的加工。

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

7.3.2.1 工作流设计器

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

图 7-5: 工作流设计器



7.3.2.1.1 节点任务

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

7.3.2.1.2 任务属性配置

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

7.3.2.1.3 历史版本

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

7.3.2.1.4 任务管理

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

7.3.2.2 代码开发编辑器

代码开发编辑器为开发人员提供代码编辑界面，帮助完成数据开发任务，主要包含以下功能。

图 7-6: 代码开发编辑器

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

7.3.2.2.1 SQL编程

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

7.3.2.2.2 MR编程

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

7.3.2.2.3 Resource资源文件

支持将本地资源文件上传的类型，如下所示。

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

7.3.2.2.4 UDF函数注册

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

7.3.2.2.5 SHELL脚本编程

提供在线的shell脚本编程与调试环境。

7.3.2.3 代码管理与团队协作

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

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

7.3.3 监控运维

7.3.3.1 系统概述

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

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

7.3.3.2 运维概览

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

7.3.3.3 任务运维

可视化展示调度任务DAG图，极大地方便您对线上任务进行运维管理。

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

- 可以针对任务进行重跑、停止、查看运行日志、查看节点代码、查看节点属性。

7.3.3.4 监控告警

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

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

7.3.4 平台管理

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

> **说明**：
> 一个用户可以同时加入多个项目空间，在不同的项目空间中被授予不同的角色。

7.3.4.1 组织管理

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

7.3.4.2 项目管理

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

7.3.4.3 项目成员管理

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

- 支持模糊搜索项目成员并可将用户移出本项目空间的操作。
- 用户加入项目空间，只针对项目管理员才可新增项目成员，支持模糊匹配查找的方式用户添加至本项目空间。

> **说明**：
> 在添加用户到项目空间时必须指定至少一种角色。

用户退出项目空间，支持项目管理员主动清退用户。

说明：

用户移出本项目后失去之前在本项目分配的所有权限。

7.3.4.4 权限管理

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

表 7-1: 角色权限表

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

7.4 产品优势

7.4.1 超大规模计算处理能力

数据资源平台与底层计算平台天然集成，轻松处理海量数据。

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

7.4.2 一站式的数据工场

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

7.4.3 海量异构数据源快速集成能力

支持400对异构数据源的离线同步，支持分钟级、小时级、天级等的调度。

7.4.4 Web化的软件服务

可在互联网/内部网络环境下直接使用，无需安装部署，拎包入驻，开箱即用。

7.4.5 多租户权限模型

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

7.4.6 开放的平台

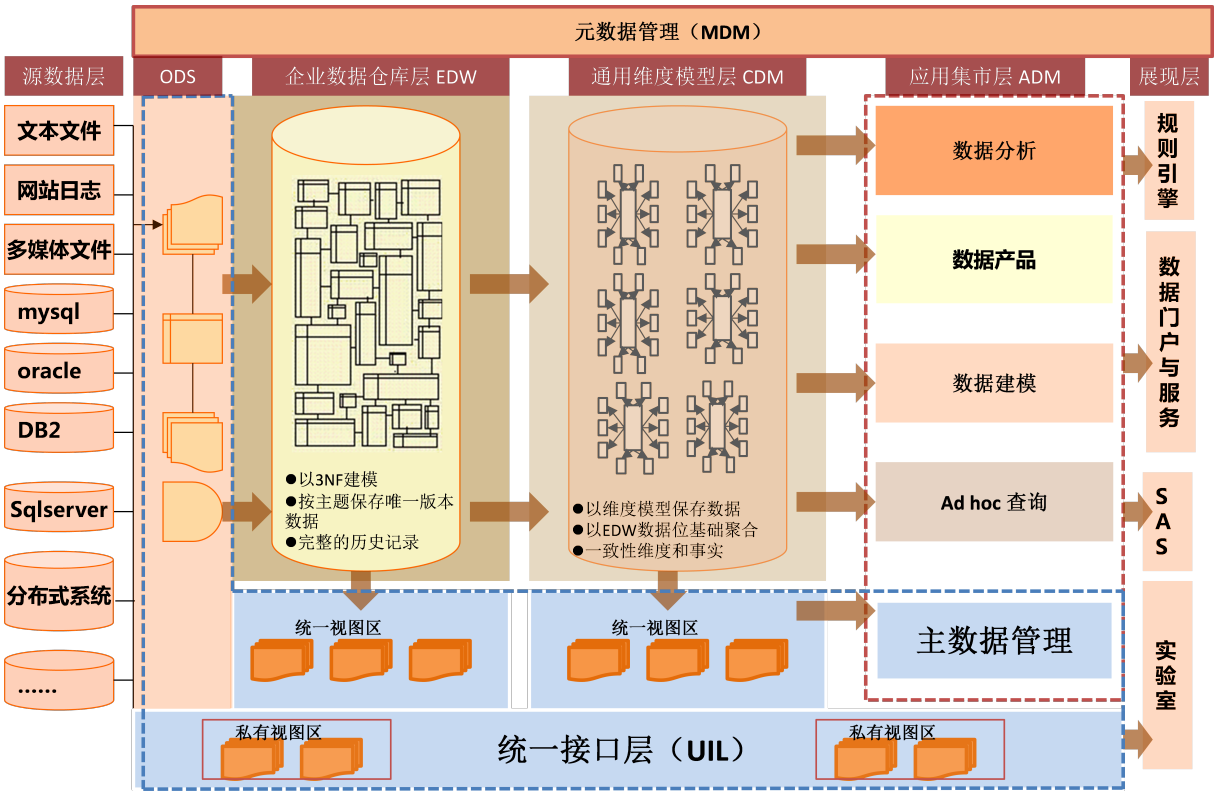
所有模块已实现组件化、服务化，您可基于DataWorks的Open API来定制开发扩展功能。

7.5 应用场景

7.5.1 云上数仓

大型企业可在专有云环境下使用DataWorks来构建超大型的数据仓库。

图 7-7: 数据采集到应用



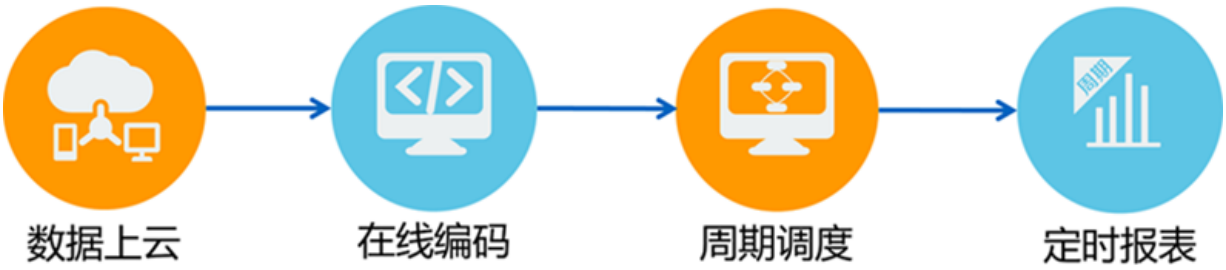
DataWorks为这类客户提供卓越的海量数据集成能力。

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

7.5.2 BI应用

本节将为您介绍如何使用DataWorks做报表。

图 7-8: BI应用流程



基于网络日志，完成如下分析需求：

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

截取一条真实的数据如下：

```
18.111.79.172 - - [12/Feb/2014:03:15:52 +0800] "GET /articles/4914.html HTTP/1.1" 200 37666 "http://coolshell.cn/articles/6043.html" "Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.1700.107 Safari/537.36" -
```

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

图 7-9: 建表

图 7-10: 各表依赖

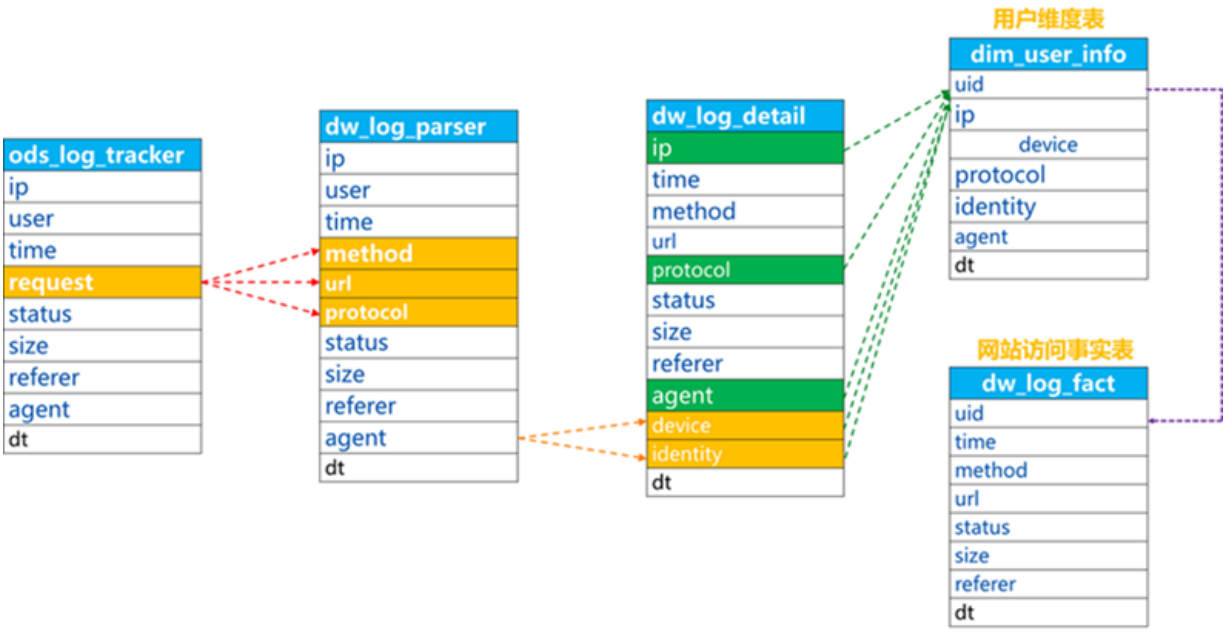


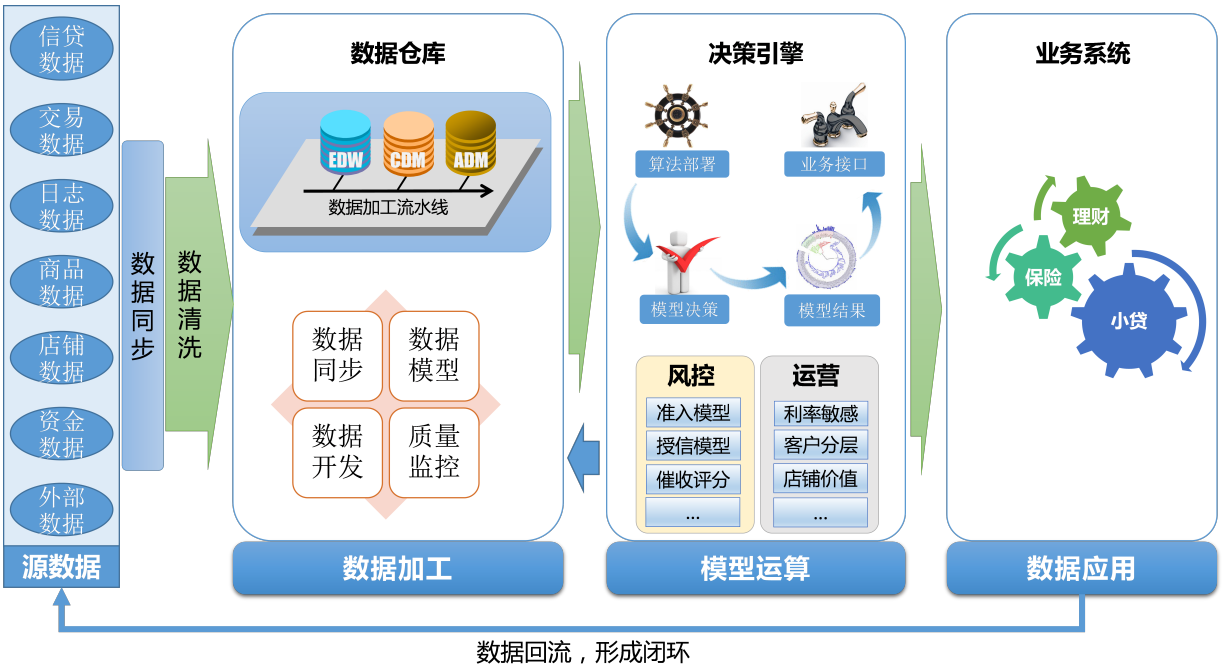
图 7-11: 新建任务

图 7-12: 开发过程

7.5.3 数据化运营

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

图 7-13: 阿里小贷的数据业务运营模式图



7.6 阿里云和DataWorks

阿里云创立于2009年，是全球领先的云计算及人工智能科技公司，为200多个国家和地区的企业、开发者和政府机构提供服务。2017年1月阿里云成为奥运会全球指定云服务商。截

至2017年3月，阿里云付费云计算用户达87.4万。阿里云致力于以在线公共服务的方式，提供安全、可靠的计算和数据处理能力，让计算和人工智能成为普惠科技。

阿里云在全球各地部署高效节能的绿色数据中心，利用清洁计算为万物互联的新世界提供源源不断的能源动力，目前已经在全球15个地域设立有数十个飞天数据中心，均部署阿里云自研的飞天操作系统，并提供中、英、日三种语言支持。

阿里云DataWorks产品经过阿里巴巴内部业务的7年沉淀，对内支撑了阿里全集团所有大数据业务，有超过1万名员工使用；目前产品技术成熟，已对外服务3年，服务于超过1万家企业客户，以及数十家政府机构。

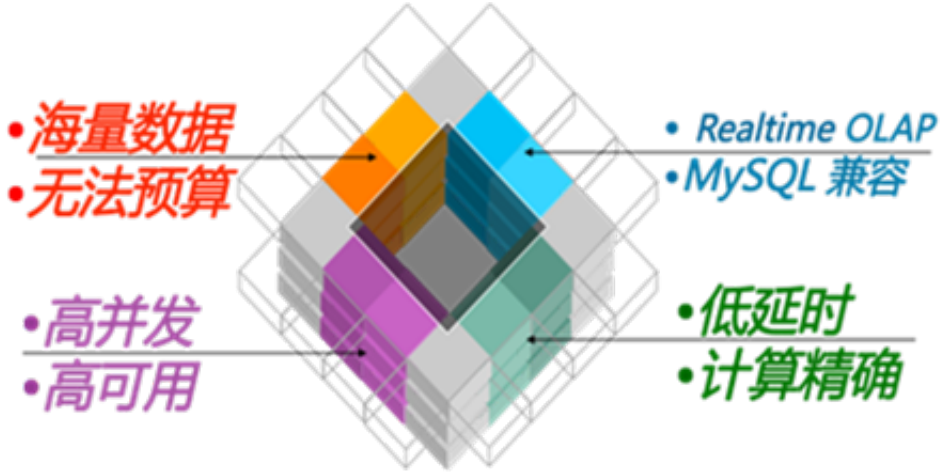
8 分析型数据库

8.1 什么是分析型数据库

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，企业待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，一旦待进行查询或计算的数据量过大，达到数千万甚至数十亿条，或待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 8-1: 数据库对比

图 8-2: 优势对比



分析型数据库（AnalyticDB，原ADS）和主流数据系统进行对比时，我们等待OLAP（On-Line Analytical Processing，联机分析处理）系统来进行处理。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。而在具体的产品中，我们通常将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，可以灵活地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，而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，分析型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。

另外，AnalyticDB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。

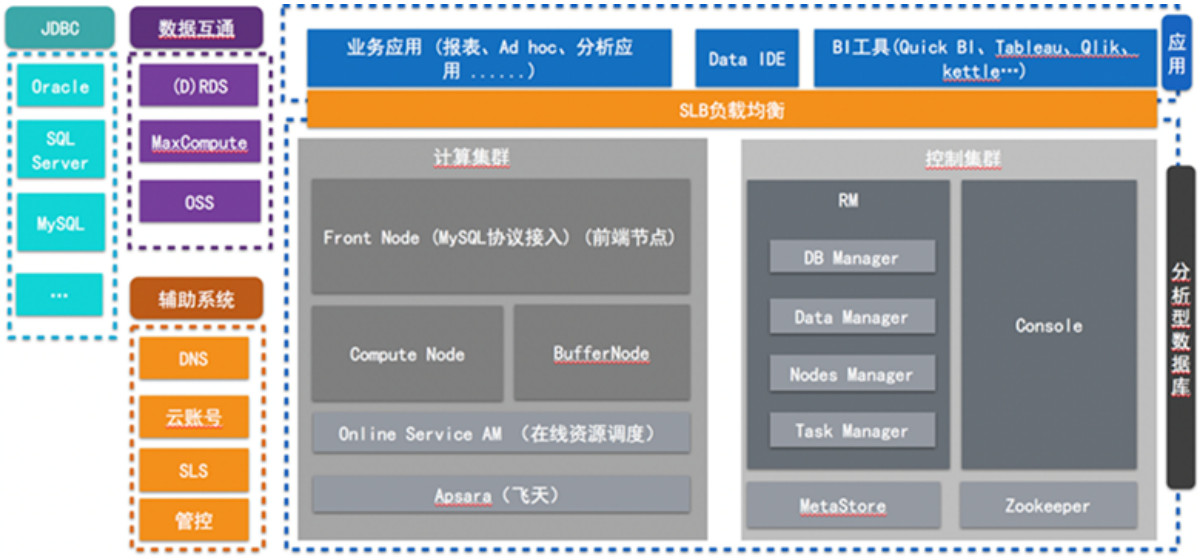
大数据领域，愈发向4M趋势发展。

图 8-3: 4M趋势

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。

8.2 产品架构

图 8-4: AnalyticDB架构



AnalyticDB（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）的阿里云分析型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FrontNode、用于进行实际的数据存储与计算的ComputeNode、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BufferNode。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。

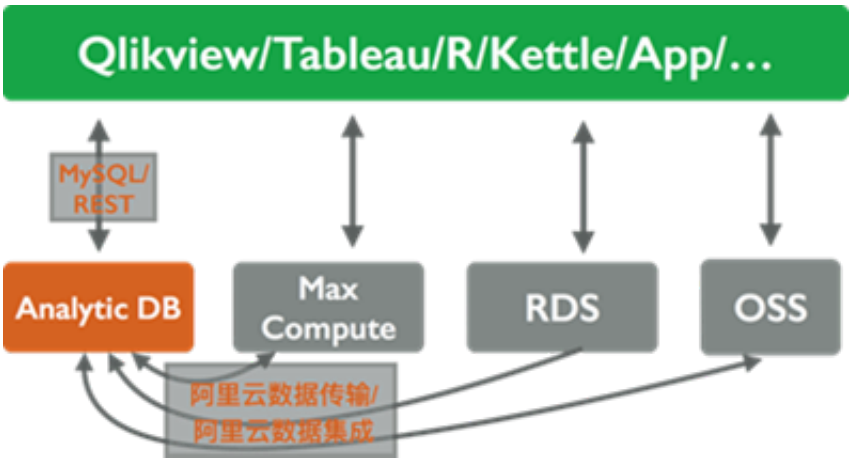
- 控制集群

  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 8-5: 外部系统交互



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(D)RDS的数据同步到分析型数据库中（须借助外部同步工具）。

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。

8.3 产品优势

海量数据计算能力

最高支持单表万亿记录、PB数据级别。

自由灵活的查询能力

通过SQL对海量数据灵活的进行多维分析、数据透视、数据筛选。

极速的查询响应时间

毫秒级的百亿级数据多维透视。

多通道并行数据导入

离线通道、在线通道双模式并行数据导入，导入性能随集群规模线性扩展。

精细化的安全机制

提供精确到列级别的权限管理和超细粒度用户操作审计，利用公私钥机制保护数据安全。

良好的兼容性

全面兼容MySQL协议（包括数据元信息），天生具备与商业分析工具、应用的兼容性，内置支持多种数据源数据快速接入，大幅度降低业务系统和商业软件的接入成本。

8.4 典型应用

电商行业

A-CRM、爆款选品、自动化运营、SKU组合分析等。

O2O

数据分析和CRM系统、地理围栏系统。

广告行业

数字营销，M-DMP系统。

金融行业

实时多维数据分析、交易流水查询系统、报表系统等。

大安全

人群透视分析，潜在关键元素挖掘，关系网络分析，明细查询等。

交通、交警

车辆卡口数据分析和研判。

物流和物联网业

车联网数据分析、企业安监数据分析、传感器数据存储和检索、物流实时数仓。

8.4.1 某银行

图 8-6: 银行应用

业务问题

通过分析型数据库，将原有的需要运行20分钟的报表加速到1秒钟，并且能够支撑冠字号查询等业务系统

已有方案

Oracle

数据规模

700G 30亿

优势

实时计算处理时间从20分钟提升到1秒

8.4.2 某交警

图 8-7: 交警应用

业务系统特点：

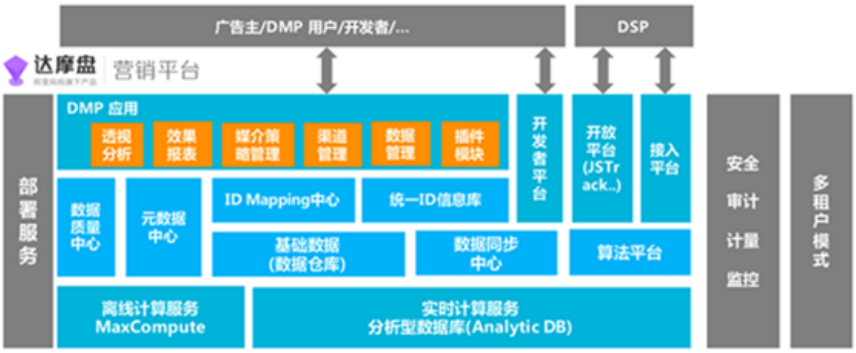
海量数据：一个市仅交通卡口过车纪录表达到300亿 - 500亿级别（保存半年），折合数据20-30T

飞速增长：市级系统每天数据增量1000万条

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

8.4.3 阿里妈妈DMP

图 8-8: DMP应用



广告DMP应用架构

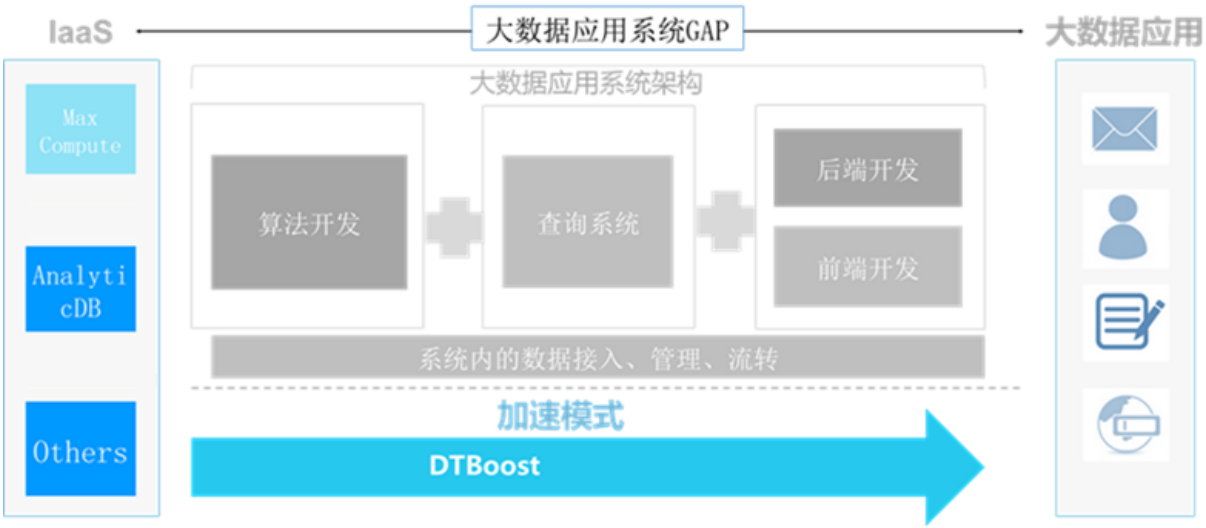
在DMP系统中，大数据处于整个系统的核心位置。MaxCompute进行用户数据清洗，标签的挖掘。AnalyticDB承接了广告主对大数据的透视和人群管理的计算工作。Analytic DB极速的海量数据DUMP功能可以将用户圈定完成的人群DUMP到查询更快速的KV存储系统中。定向引擎根据KV存储系统的数据，服务于DSP系统。

9 大数据应用加速器

9.1 什么是数据加速器

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地整合阿里云各个大数据产品，大大降低搭建大数据应用系统当中绝大部分的系统工程工作，在相应行业应用解决方案的结合下，能够让不是很熟悉大数据应用系统开发的程序员也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。

图 9-1: 大数据应用系统



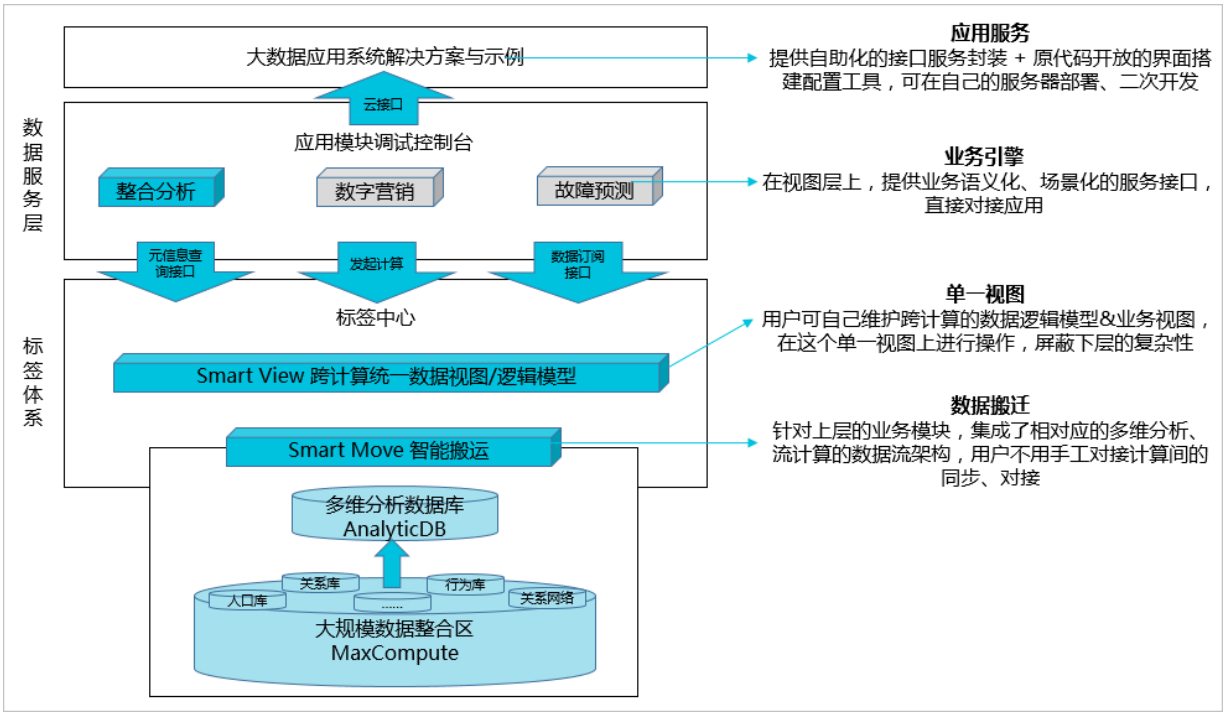
概括来讲，DTBoost是以标签中心为基础，建立跨多个云计算资源之上的统一逻辑模型，开发者可以在**标签**这种逻辑模型视图上结合画像分析、文本挖掘、个性化推荐、关系网络等多个业务场景的数据服务模块，通过接口的方式进行快速的应用搭建。这种方式有以下好处：

- 可屏蔽掉应用开发人员对于下层多个计算存储资源的深入理解与复杂的系统对接工作。
- 通过数据服务的形式透出也有助于 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。

9.2 产品架构

图 9-2: DTBoost架构图



9.3 功能特性

9.3.1 标签中心

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

功能简介

- 数据资源整合

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

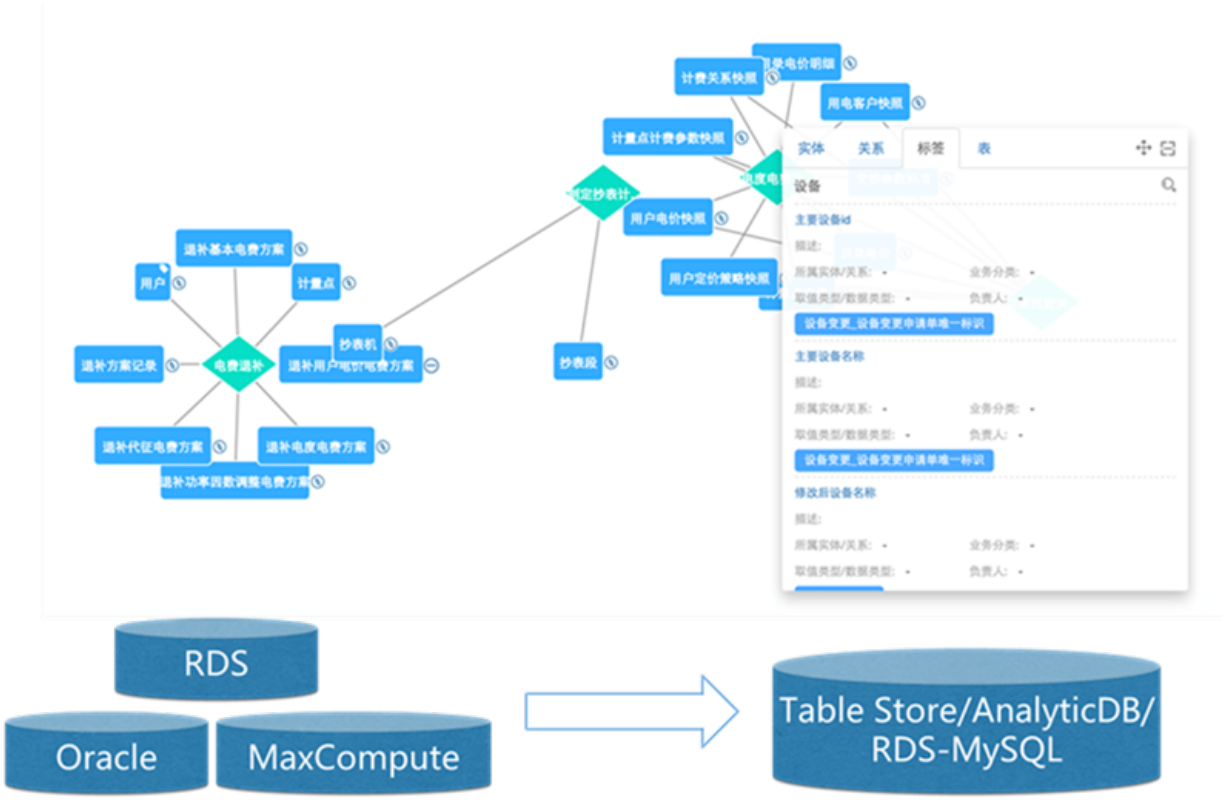
- 为数据服务提供视图支撑

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

- 数据访问管理

可以通过逻辑层对数据访问权限进行有效控制，比物理表的访问管理更加安全有效。

图 9-3: DTBoost功能



产品特性

- 业务视角管理

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

- 跨计算的统一逻辑模型

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

- 灵活拓展性

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

9.3.2 整合分析

在逻辑模型上像分析**大宽表**一样进行灵活自由的分析表达，通过快速自主封装可实时计算的分析接口头透出到前端，或是在线可视化界面配置生成可以独立部署交互式分析应用代码，实现敏捷的交互式分析应用搭建。

功能简介

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

图 9-4: 整合分析功能

图 9-5: 分析结果

产品特性

表 9-1: 一般 BI 工具和整合分析对比表

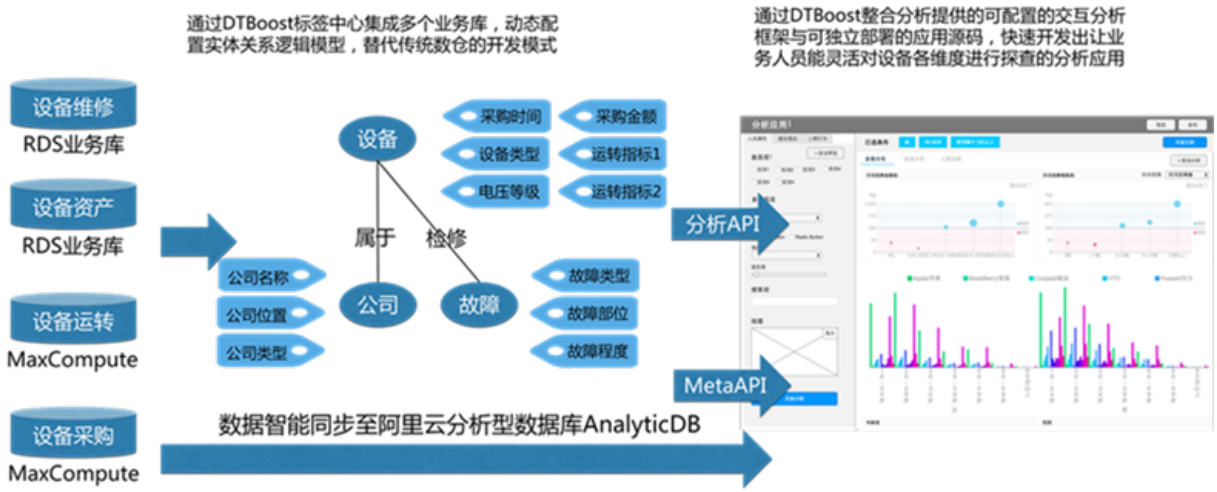
对比项	一般 BI 工具	整合分析
查询灵活度	一般来说以一层关联较多	与标签中心模型配合，能够自定义拖拽配置多层子查询
分析交互	以固定报表查看为主	强化交互式 SQL 模型不定由用户自己选择的分析
可拓展性	拓展性较低，以固定产品	开放应用层的源代码，可以自行进行集成拓展
系统集成性	一般作为独立分析工具，不与其他系统集成	能够方便的进行对下游系统的对接（如广告投放系统）

9.4 典型应用

- 结构化数据全量数据200G以上。
- 希望把多个业务系统/小数仓的数据进行整合打通。
- 分析场景围绕一个主体展开（人、设备、车辆.....）。

- 数据维度（包括衍生出来的维度）超过10个以上。
- 业务人员希望自己来进行分析，而不是只看固定报表。

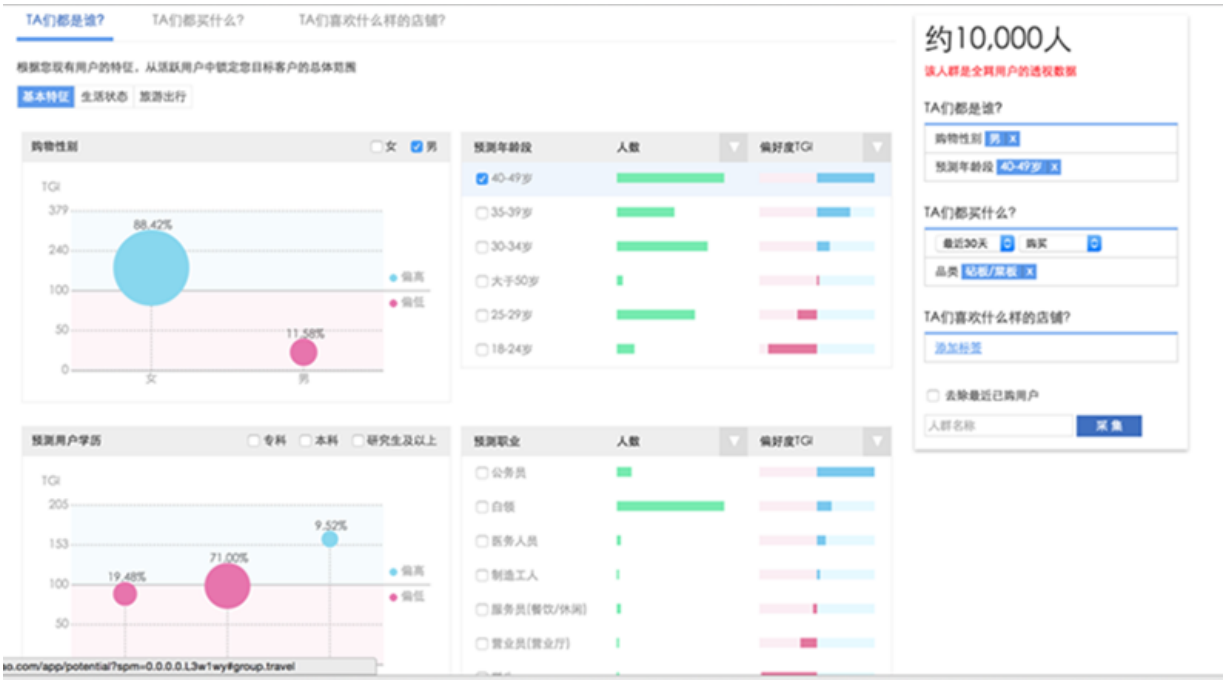
图 9-6: DTBoost应用



9.4.1 画像分析

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

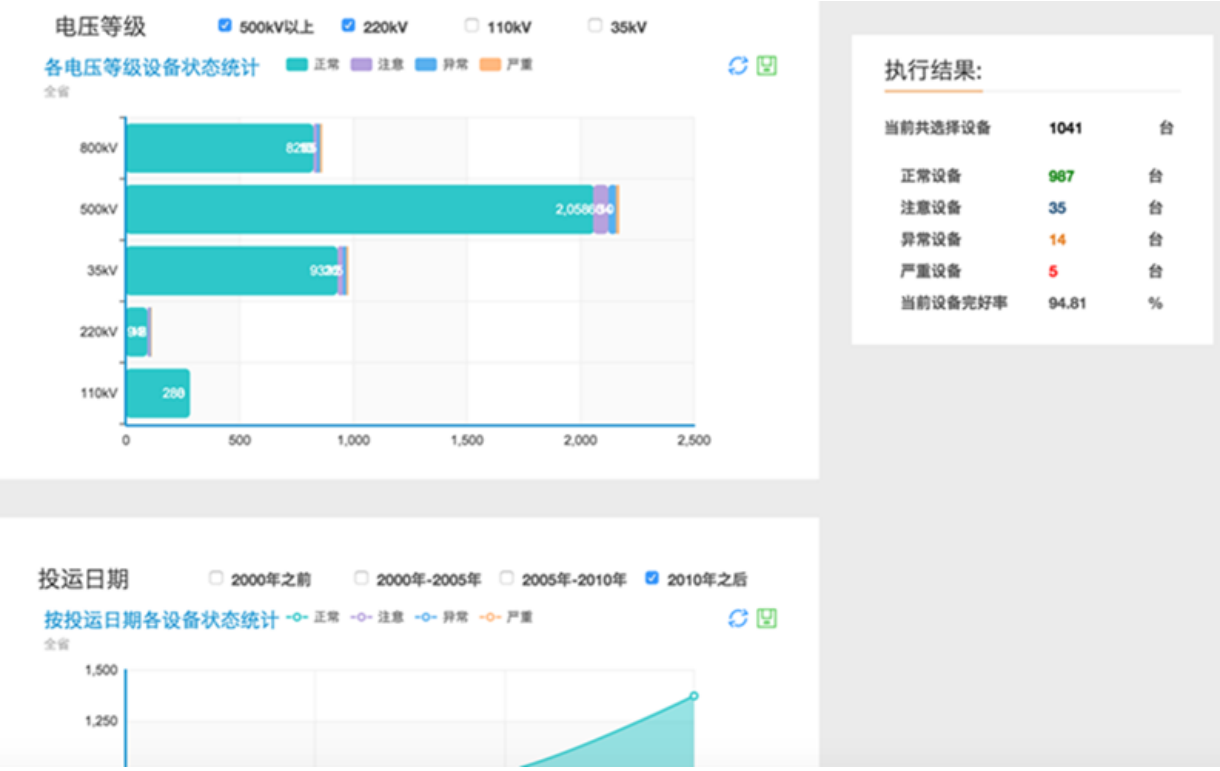
图 9-7: 画像分析



9.4.2 设备履历

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

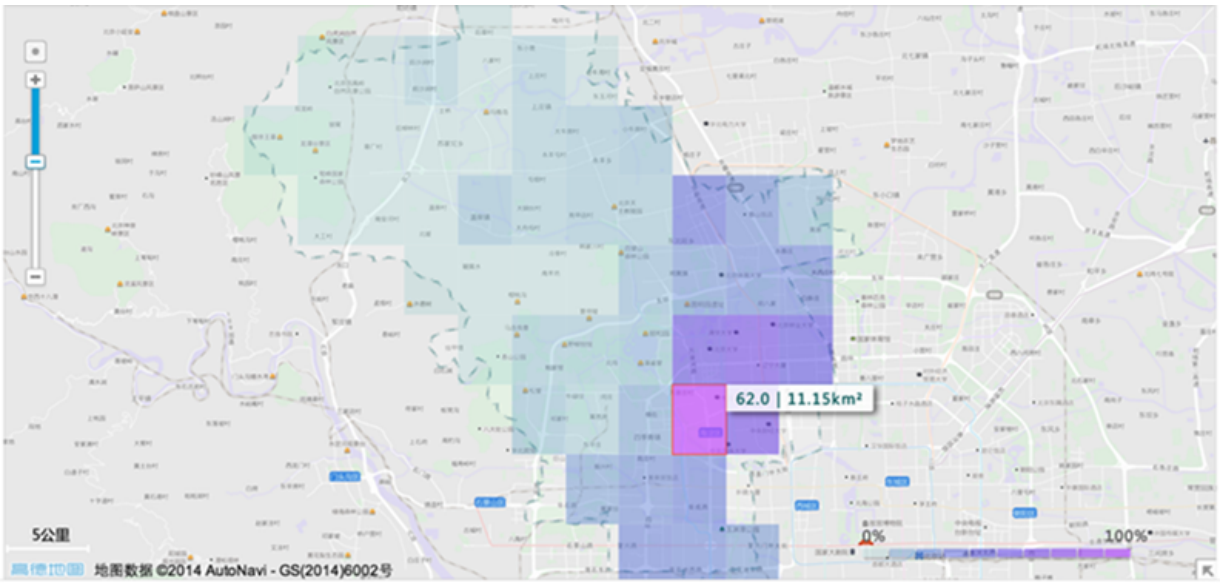
图 9-8: 设备履历



9.4.3 地理分析

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

图 9-9: 地理分析



10 大数据管家

10.1 什么是大数据管家

大数据管家（Big Data Manager，以下简称为BCC）是为阿里巴巴专有云的大数据产品定制的一款运维产品。通过大数据管家可以对MaxCompute、AnalyticDB、DataWorks等产品进行管理。

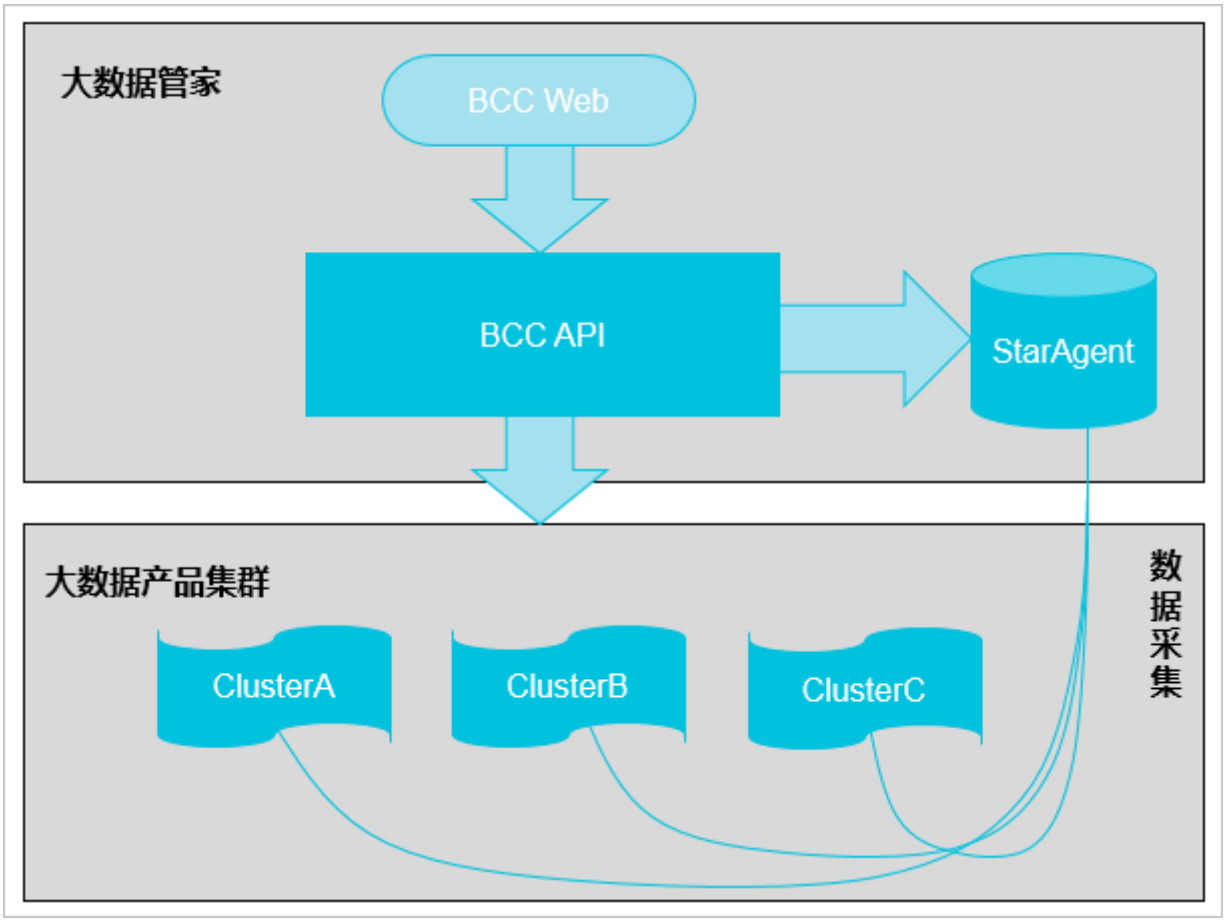
通过大数据管家的赋能，专有云驻场人员可以轻松地管理大数据产品，比如查看大数据产品的运行指标，修改大数据产品运行配置，对大数据产品进行自检，搜索日志等功能。

10.2 产品架构

大数据管家组成部分如下所示。

- BCC WEB（前端）
- BCC API（管控API）

图 10-1: BCC组成部分



10.3 功能特性

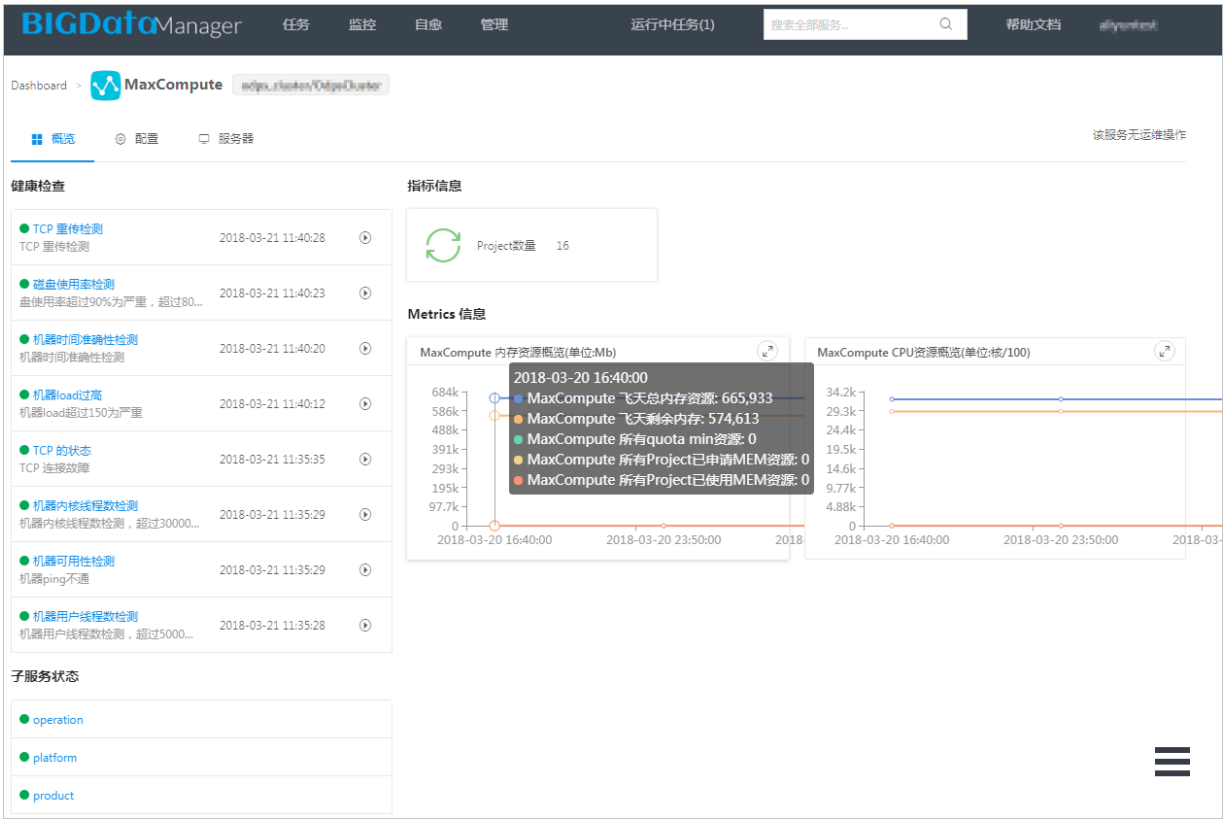
下面对大数据管家的主要功能做一些介绍，详细的功能使用可以参考《运维指南》中的**大数据管家**章节。

10.3.1 业务概览

概览页面显示集群的几项重要信息，包括计算配额的CPU利用率、操作系统平均负载、磁盘利用率、网络流量、集群内机器数量、盘古物理容量（已使用和总容量）、系统任务数量、盘古已创建文件数量及部分数据的一周内变化比例。

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

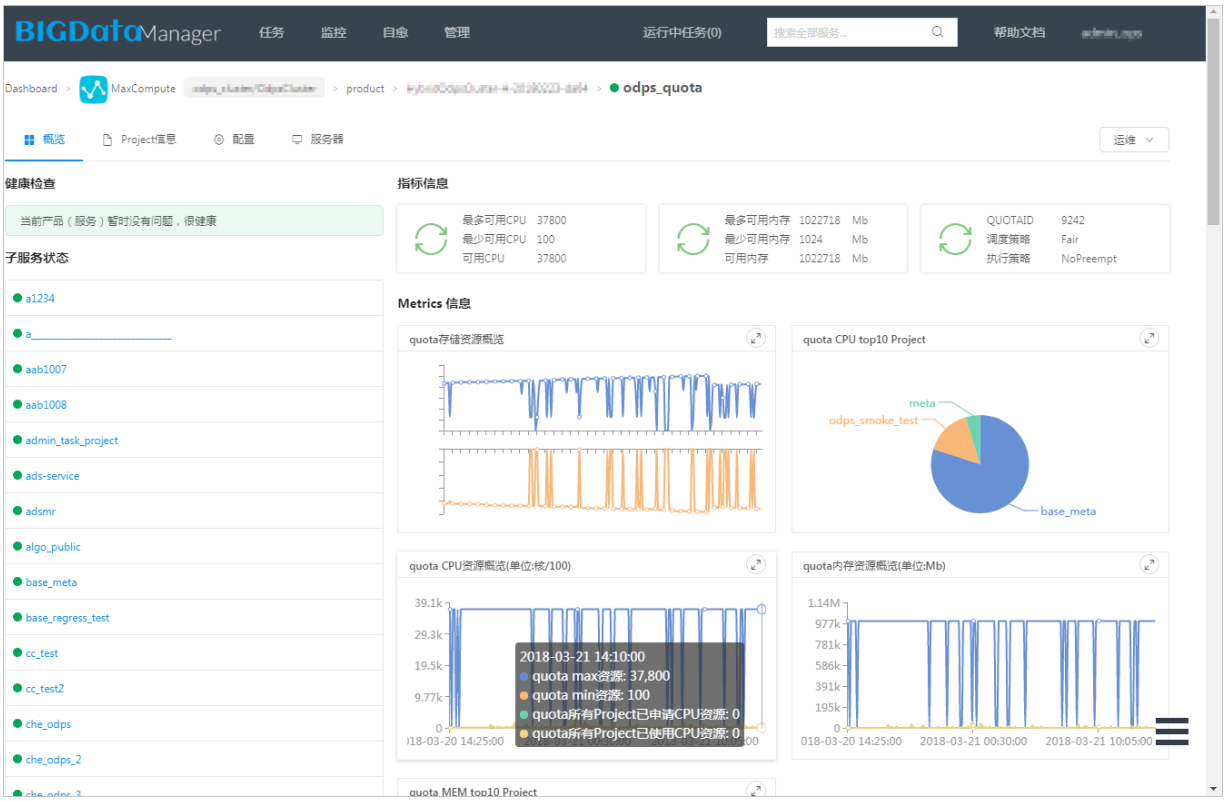
图 10-2: 业务概览图



10.3.2 业务管理

业务管理是项目管理、配合管理、SQL加速、多集群复制、集群容灾、计算集群管理、Tunnel集群endpoint管理、FuxiJob全局配置的总入口。

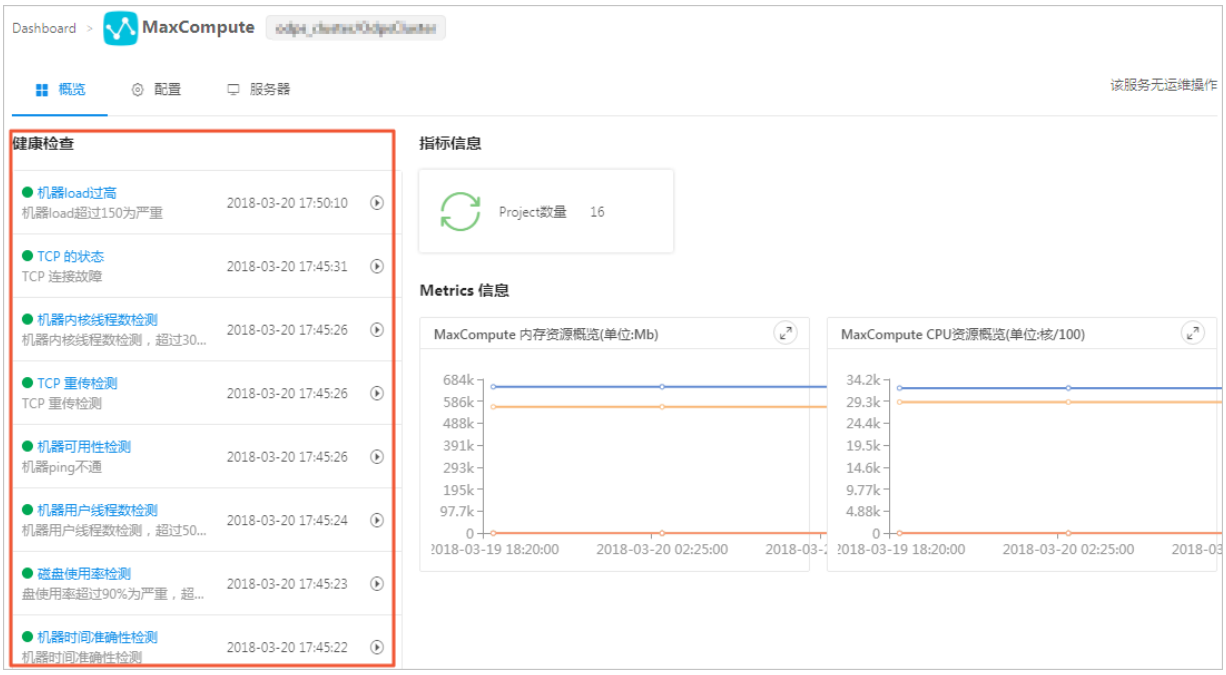
图 10-3: 业务管理图



10.3.3 业务自检

业务自检是指可以对飞天、AnalyticDB、MaxCompute等业务进行健康检查，并按分值（满分100）作出健康度的判断。

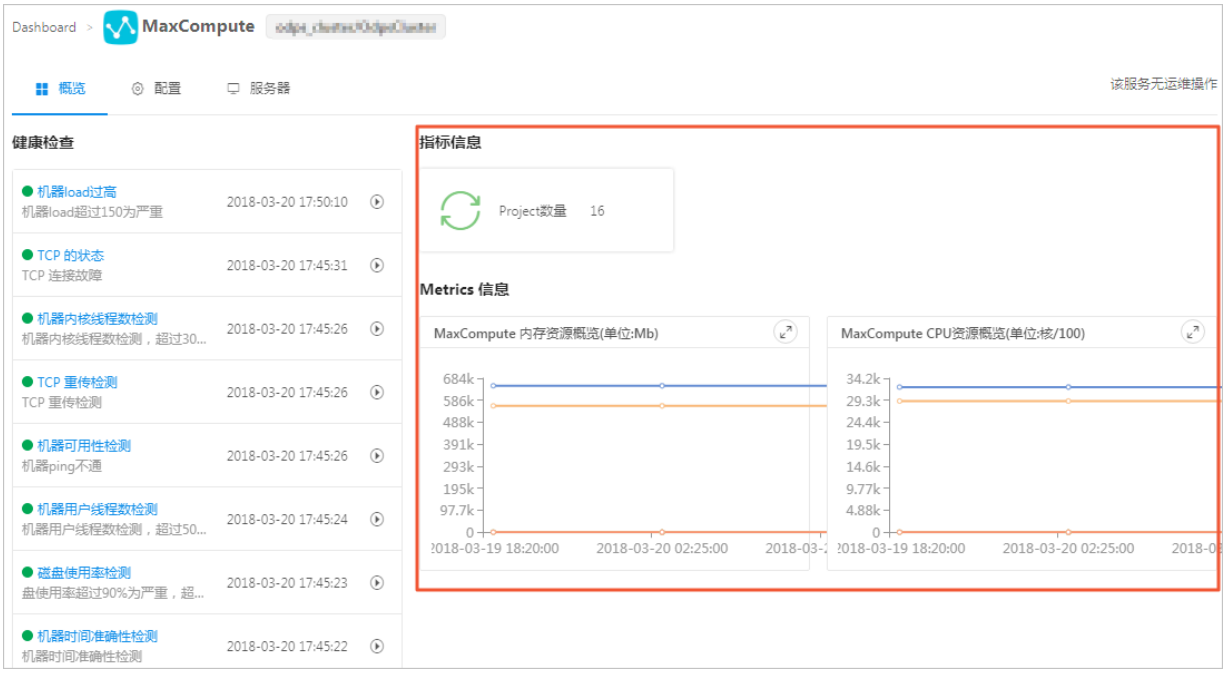
图 10-4: 业务自检图



10.3.4 系统监控

系统监控包括了集群CPU、内存、磁盘IO、网络等的使用量。可选择日期查看集群使用量趋势。对于不同的产品，大数据管家所监控的指标项不同。

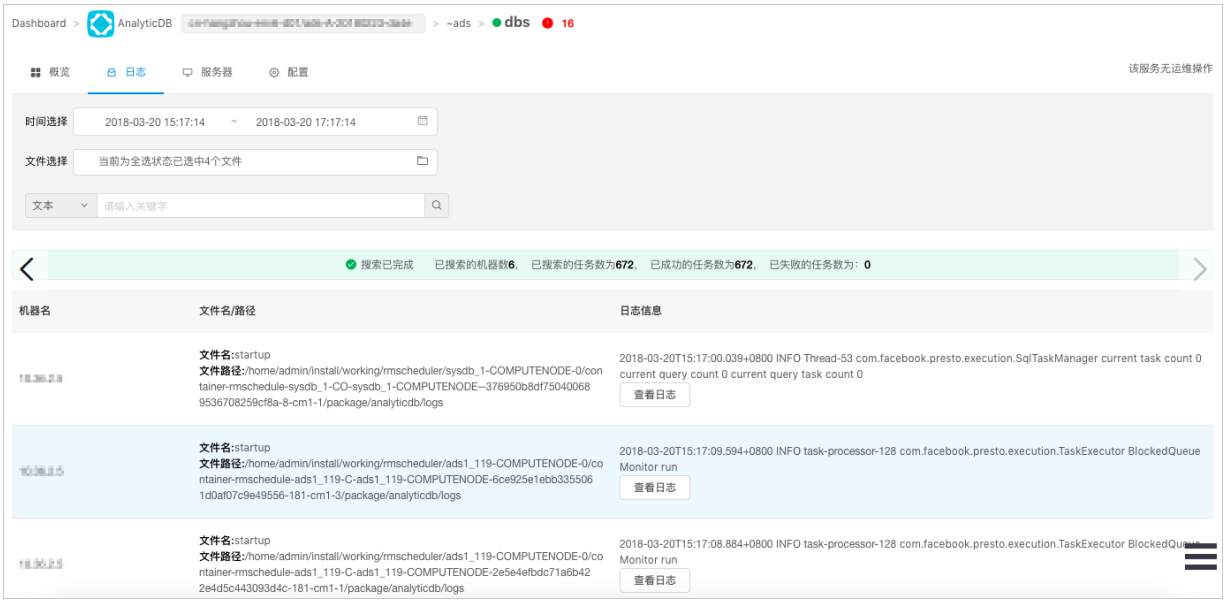
图 10-5: 系统监控图



10.3.5 日志管理

日志管理可以拿到各种service中的各个worker的日志，可以根据关键字筛选打包下载。

图 10-6: 日志管理图



时间选择 2018-03-20 15:17:14 ~ 2018-03-20 17:17:14

文件选择 当前为全选状态已选中4个文件

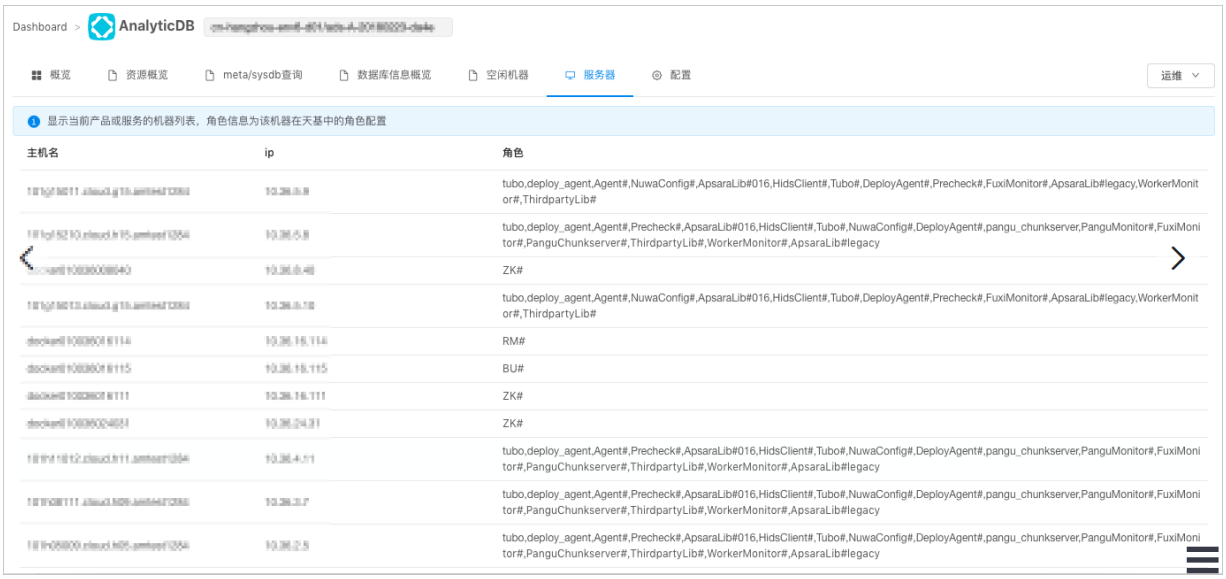
搜索已完成 已搜索的机器数**6**， 已搜索的任务数为**672**， 已成功的任务数为**672**， 已失败的任务数为：**0**

机器名	文件名/路径	日志信息
[redacted]	**文件名:**startup **文件路径:**/home/admin/install/working/rmscheduler/sysdb_1-COMPUTENODE-0/container-rmschedule-sysdb_1-CO-sysdb_1-COMPUTENODE--376950b8df750400689536708259cf8a-8-cm1-1/package/analyticdb/logs	2018-03-20T15:17:00.039+0800 INFO Thread-53 com.facebook.presto.execution.SqlTaskManager current task count 0 current query count 0 current query task count 0
[redacted]	**文件名:**startup **文件路径:**/home/admin/install/working/rmscheduler/ads1_119-COMPUTENODE-0/container-rmschedule-ads1_119-C-ads1_119-COMPUTENODE-6ce925e1ebb3355061d0af07c9e49556-181-cm1-3/package/analyticdb/logs	2018-03-20T15:17:09.594+0800 INFO task-processor-128 com.facebook.presto.execution.TaskExecutor BlockedQueue Monitor run
[redacted]	**文件名:**startup **文件路径:**/home/admin/install/working/rmscheduler/ads1_119-COMPUTENODE-0/container-rmschedule-ads1_119-C-ads1_119-COMPUTENODE-2e5e4efbdc71a6b422e4d5c443093d4c-181-cm1-1/package/analyticdb/logs	2018-03-20T15:17:08.884+0800 INFO task-processor-128 com.facebook.presto.execution.TaskExecutor BlockedQueue Monitor run

10.3.6 机器管理

后台管理是服务器与角色、全局配置、机器环境信息、管控API冒烟、飞天管理、操作审计、云账号管理、用户&角色的总入口。

图 10-7: 机器管理图



显示当前产品或服务的机器列表，角色信息为该机器在天基中的角色配置

主机名	ip	角色
[redacted]	[redacted]	tubo,deploy_agent,Agent#,NuwaConfig#,ApsaraLib#016,HidsClient#,Tubo#,DeployAgent#,Precheck#,FuxiMonitor#,ApsaraLib#legacy,WorkerMonitor#,ThirdpartyLib#
[redacted]	[redacted]	tubo,deploy_agent,Agent#,Precheck#,ApsaraLib#016,HidsClient#,Tubo#,NuwaConfig#,DeployAgent#,pangu_chunkserver,PanguMonitor#,FuxiMonitor#,PanguChunkserver#,ThirdpartyLib#,WorkerMonitor#,ApsaraLib#legacy
[redacted]	[redacted]	ZK#
[redacted]	[redacted]	tubo,deploy_agent,Agent#,NuwaConfig#,ApsaraLib#016,HidsClient#,Tubo#,DeployAgent#,Precheck#,FuxiMonitor#,ApsaraLib#legacy,WorkerMonitor#,ThirdpartyLib#
[redacted]	[redacted]	RM#
[redacted]	[redacted]	BU#
[redacted]	[redacted]	ZK#
[redacted]	[redacted]	ZK#
[redacted]	[redacted]	tubo,deploy_agent,Agent#,Precheck#,ApsaraLib#016,HidsClient#,Tubo#,NuwaConfig#,DeployAgent#,pangu_chunkserver,PanguMonitor#,FuxiMonitor#,PanguChunkserver#,ThirdpartyLib#,WorkerMonitor#,ApsaraLib#legacy
[redacted]	[redacted]	tubo,deploy_agent,Agent#,Precheck#,ApsaraLib#016,HidsClient#,Tubo#,NuwaConfig#,DeployAgent#,pangu_chunkserver,PanguMonitor#,FuxiMonitor#,PanguChunkserver#,ThirdpartyLib#,WorkerMonitor#,ApsaraLib#legacy
[redacted]	[redacted]	tubo,deploy_agent,Agent#,Precheck#,ApsaraLib#016,HidsClient#,Tubo#,NuwaConfig#,DeployAgent#,pangu_chunkserver,PanguMonitor#,FuxiMonitor#,PanguChunkserver#,ThirdpartyLib#,WorkerMonitor#,ApsaraLib#legacy

11 关系网络分析

11.1 什么是关系网络分析

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

产品围绕“大数据多源融合、计算应用、可视分析、业务智能”设计实现，结合关系网络、时空数据、地理制图学建立可视化表征，揭示对象间的关联和对象时空相关的模式及规律。产品提供关联网络、时空网络、即问网络、信息立方、智能研判、协作共享、动态建模等功能，以可视化的方式有效融合机器的计算能力和人的认知能力，获得对于海量数据的洞察力，帮助您更为直观、高效地获取信息和知识。

11.2 产品定位

I+= Information to Image to Intelligence to ...

解决的问题：**如何从海量数据中发现有价值的情报。**

基于阿里云的大数据平台，处理PB级别数据。

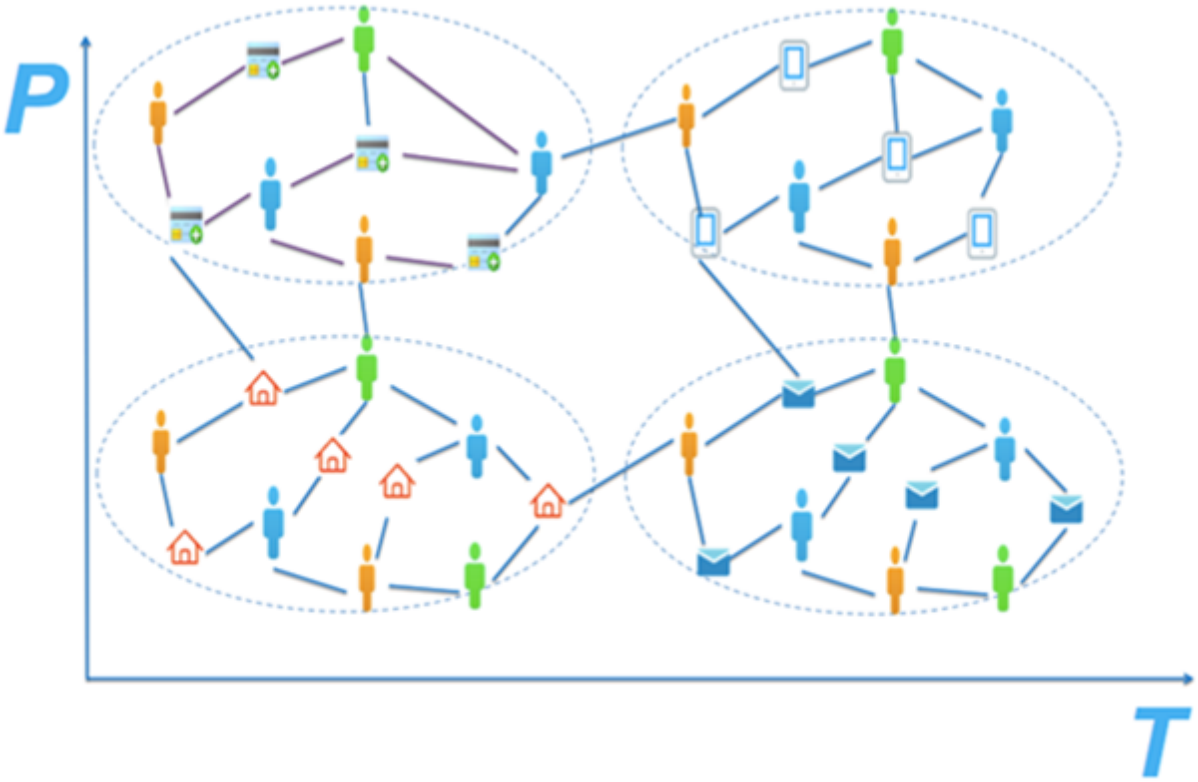
一个可视化的情报解决方案，不只是一个 BI 工具，也不只是可视化工具。

- 融合多张网，用户关系网络、资金关系网络、媒介关系网络、位置关系网络。
- 超过百亿对象，千亿关系。支付宝内部几亿用户，千亿关系对。
- 发现人和人，人和物，物和物等对象之间的关系。

11.2.1 风险防控价值

传统的风险管理中，基于单个设备、账户名单的账户黑白名单体系，存在被绕过的风险，所以需要融合风险事件中的位置、用户行为、关系等大数据元素，构建风险**可疑网络、可信网络、可视化分析体系**，并在此基础上预测潜在的风险、识别并监控团伙网络、还原风险事件现场、识别反洗钱团伙资金链路、预防营销活动反作弊。

图 11-1: 风险防控



11.2.2 公共安全价值

911 是发生在美国本土最为严重的**恐怖袭击**，遇难者总数高达**2996**人，对美经济损失达**2000**亿美元，相当于当年生产总值的**2%**。同时，此次事件对全球经济所造成的损害甚至达到**1万亿**美元左右。

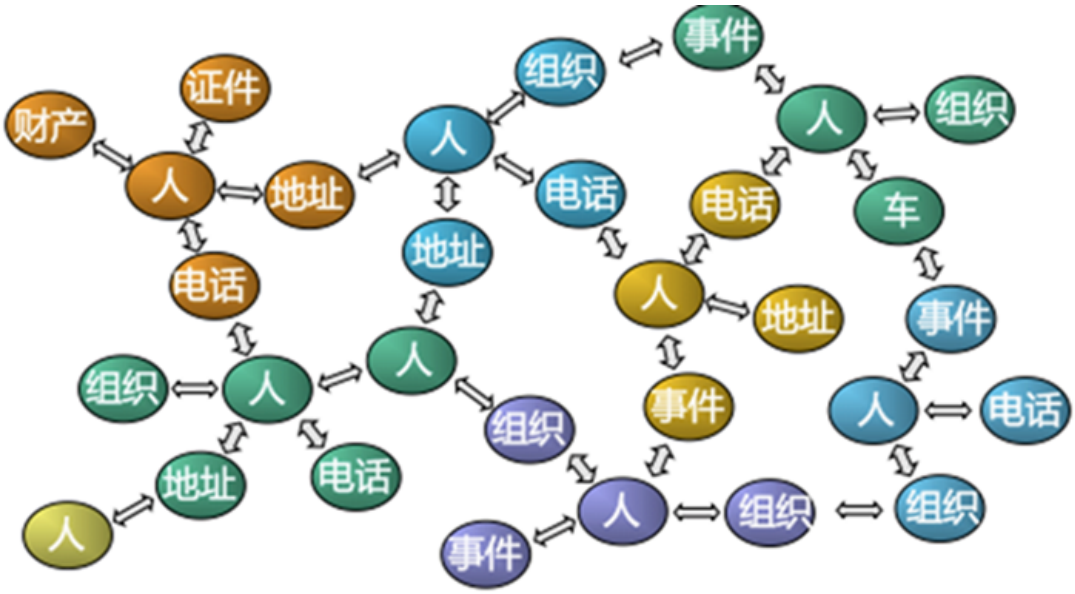
图 11-2: 恐怖袭击

图 11-3: 难民

11.2.3 世界万物相连

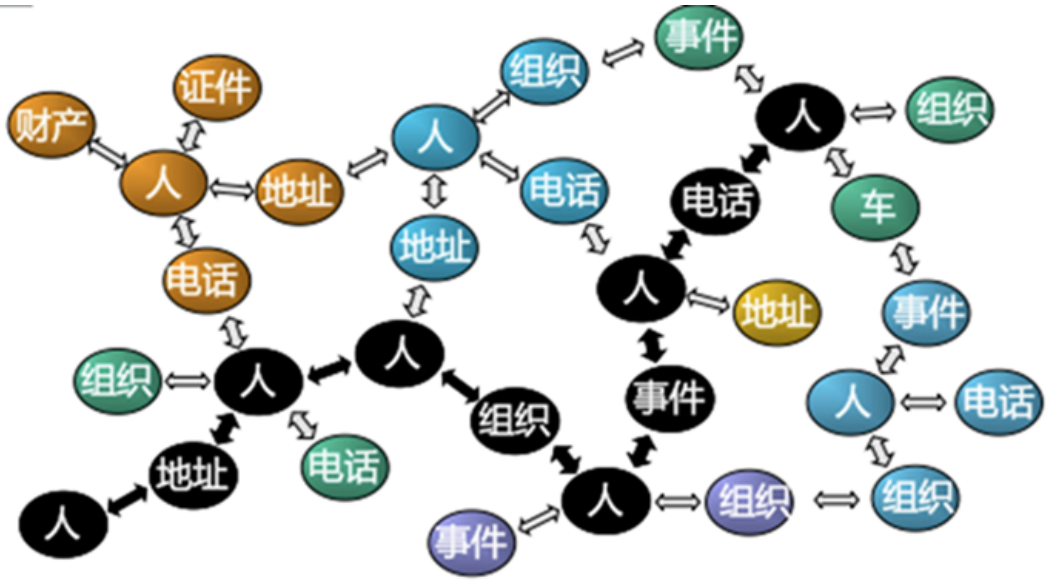
世界本就一张网，如下：

图 11-4: 世界本就一张网



任何两物之间都可以寻找关联，留下线索：

图 11-5: 两物关联

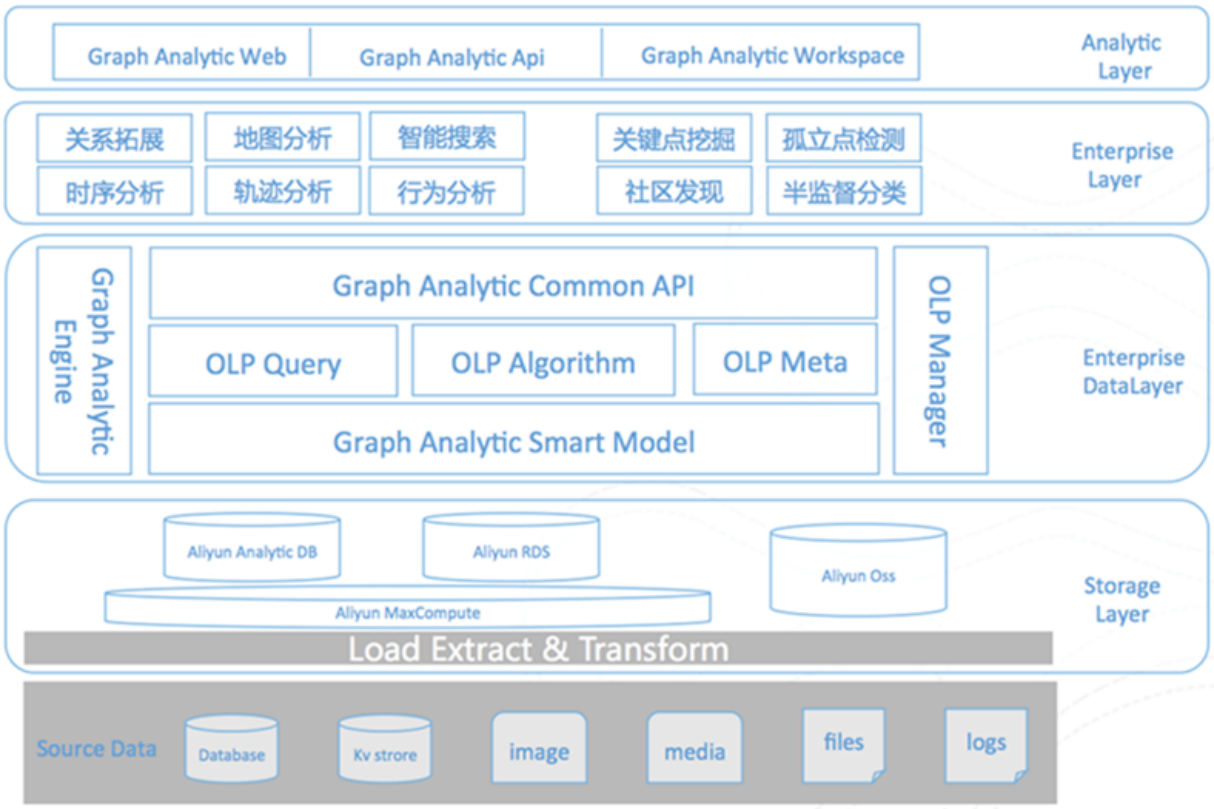


11.3 产品架构

11.3.1 系统架构

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

图 11-6: 系统架构



整个系统分为存储计算层、数据服务层、业务应用层、分析展现层。

- **存储计算层**

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

- **数据服务层**

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

- **业务应用层**

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

- **应用分析层**

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

11.3.2 OLP模型

图 11-7: OLP模型

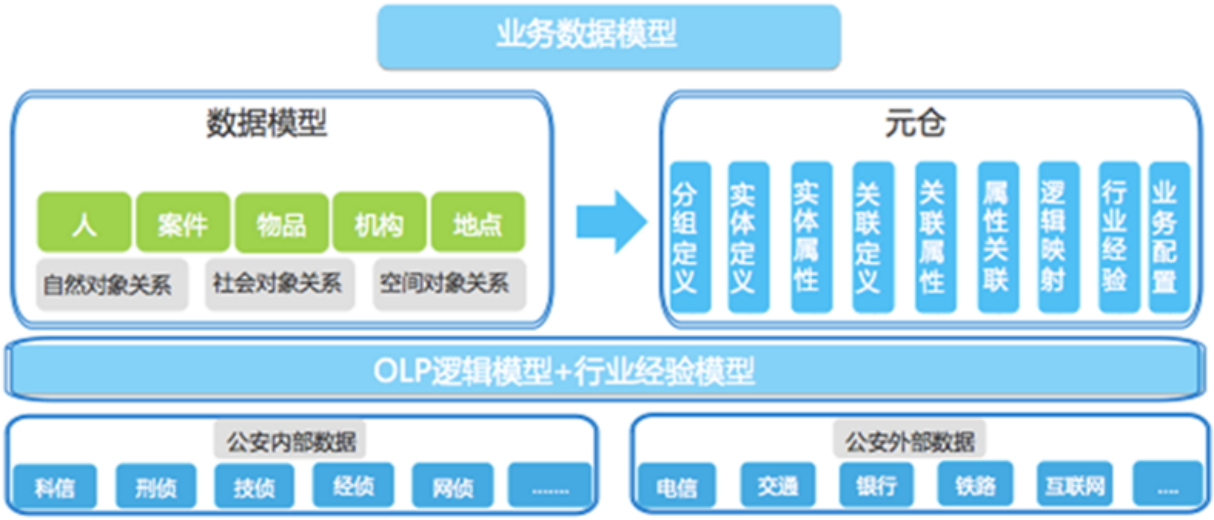


图 11-8: 以高铁为例

11.4 功能特性

三大功能模块

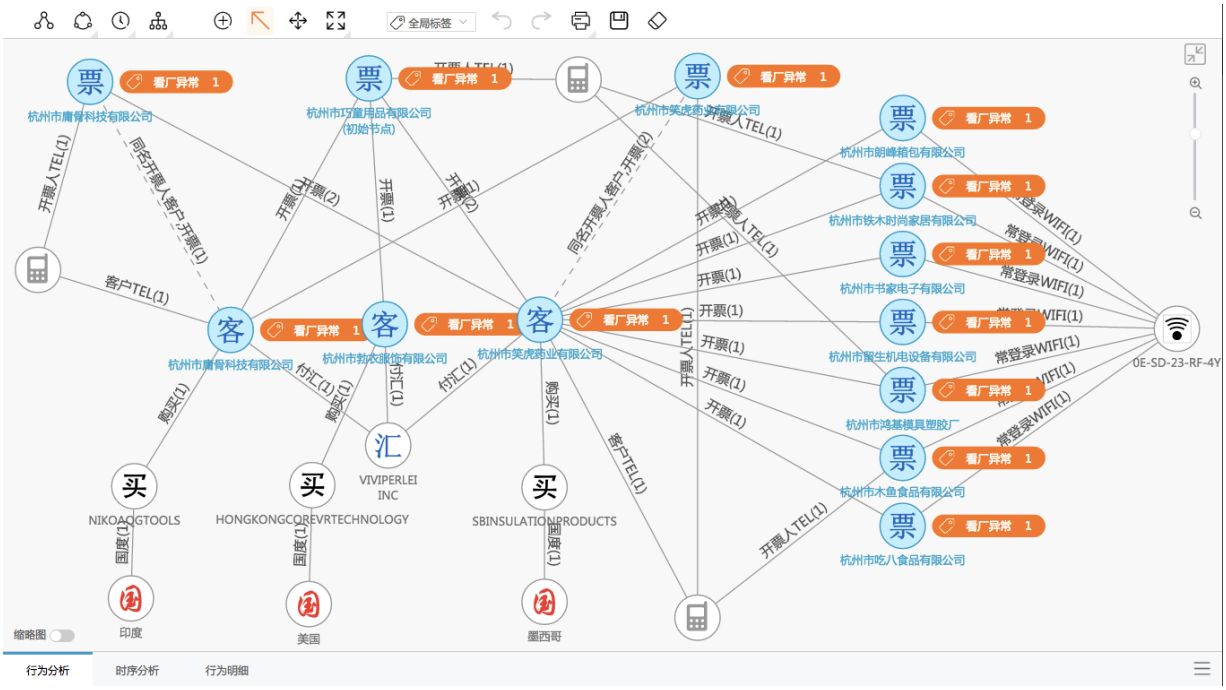
11.4.1 搜索

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

图 11-9: 搜索

11.4.2 关系网络

图 11-10: 关系网络



关联反查

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

群体分析

分析一批相同或不同类型的对象内部之间的关联关系，包含直接关系和间接关系。

共同邻居

分析一批相同或不同类型的对象共同联系对象。

路径分析

分析两个对象之间的关系路径。

骨干分析

针对团伙网络，通过智能业务算法，探索关系网络中核心骨干节点。

血缘分析

以家族户号为血缘脉络，展示所有人之间的血缘关联。

时序分析

在时间维度上详细展示每个事件发生细节。

信息立方

- 行为分析

  展示事件在时间维度上发生的分布频率情况。

- 行为明细

  将事件的明细信息（原数据记录按规则筛选）展示出来。

- 对象信息

  汇总关系网络中的实体，并按实体类型分类。

- 统计信息

  统计关系网络中的关系和实体，包含总体分布、对象熟悉和关系属性。

群体统计

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

标签统计

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

图区布局

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

右键操作

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

目录管理

目录管理可以对目录和分析管理操作。对于目录提供新建分析，删除目录，重命名，新建目录操作；对于分析提供打开，删除，重命名的功能操作。同时目录管理提供搜索对于内容检索定位和“分享”功能将个人的分析内容分享给他人延续分析。

协作共享

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

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

同时产品支持对协同分析的管理，包括删除、重命名，历史版本等管理。

11.4.3 地图分析

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

图 11-11: GIS模块的前端界面一

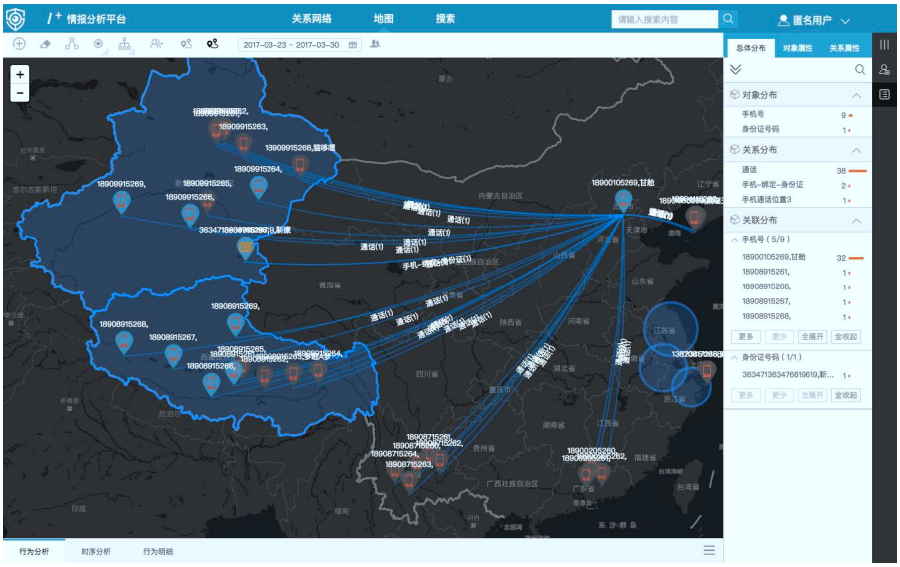
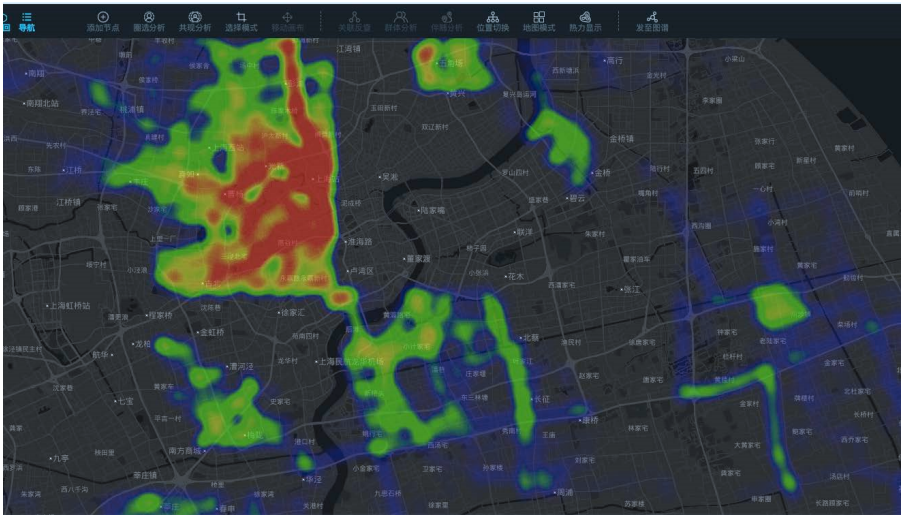


图 11-12: GIS模块的前端界面二



GIS模块的功能如下所示。

空间关系网络

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

动态轨迹

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

圈地选人

支持用户自定义地理位置和空间范围，搜寻时空交错中显现的对象。

伴随分析

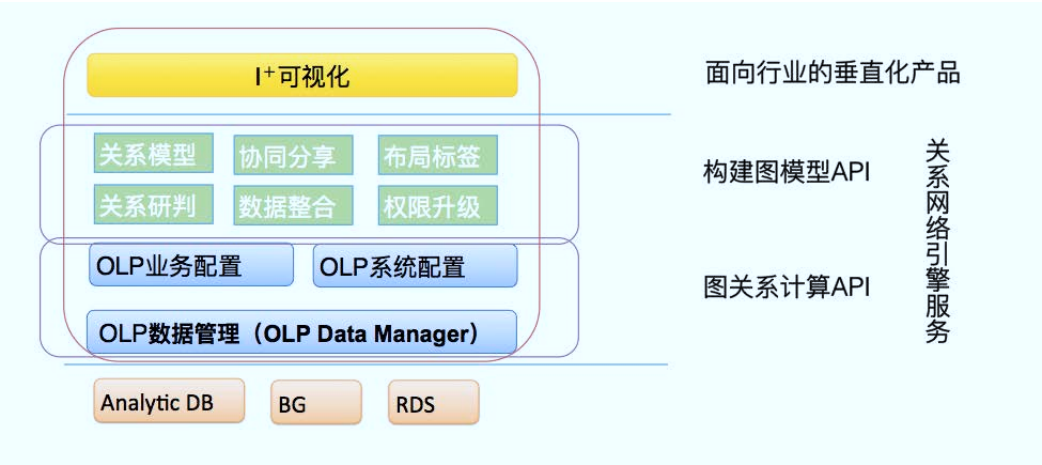
构建降纬hash空间，极速计算潜在伴随对象，支持业务时空自定义。

共现分析

以自定义时空为基础，海量数据中探查较高概率共同出现的对象。

11.4.4 关系引擎

图 11-13: 关系引擎



- 关系网络、搜索、地图模块所有的计算API开放。
- 提供基本的关系点和关系边的基本运算。
- 未来将支持用户自定义注册API。

11.5 产品优势

海量数据实时挖掘

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

模型认知万物相连

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

业务场景灵活赋能

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

可视分析高效体验

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

智能深入以人为本

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

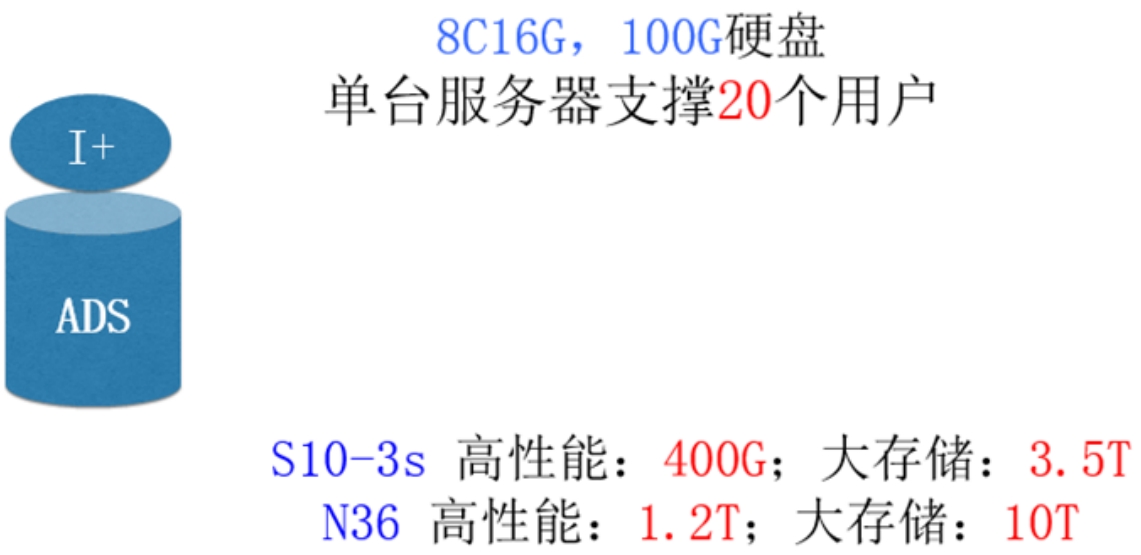
重大项目深度考验

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

11.6 性能指标

11.6.1 规格参考

图 11-14: 规格参考图



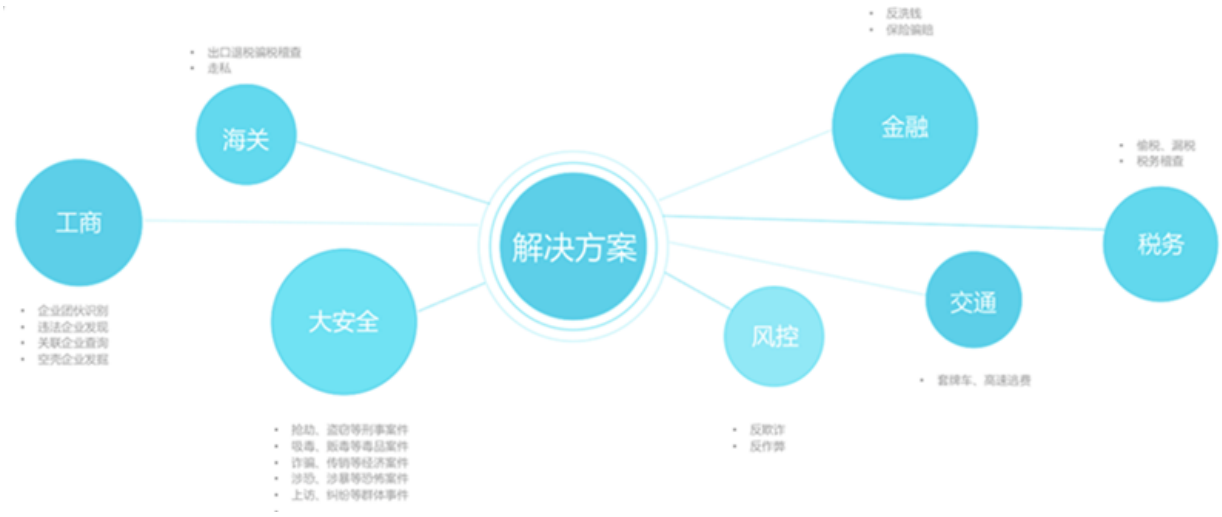
11.6.2 性能参考

压测数据#仅供参考。

图 11-15: 性能参考图

11.7 典型应用

图 11-16: 典型应用



11.7.1 典型案例

案例说明

本案例讲述如何通过I+平台侦破复杂BK案。

数据环境

本案例数据资源如下：

- 通讯工具、即时通讯通联

- 人员基本信息
- 手机轨迹、车轨迹
- 租车信息、住宿信息、火车信息
- 护照信息、出入境记录

案件背景

2015年底，某部门监测到部分小众通讯工具异常活跃，使用该工具的账号与境外某国家频繁联系，形迹可疑，按照工作要求，分析人员通过I+快速核实情况，甄别风险。

> **说明**：
> 本故事不代表真实案件，仅为 I+平台功能测试杜撰！

分析过程

分析人员筛选2015年12月份活跃用户。

图 11-17: 筛选活跃用户

结果显示，有一批账号在该时段频繁使用小众工具。

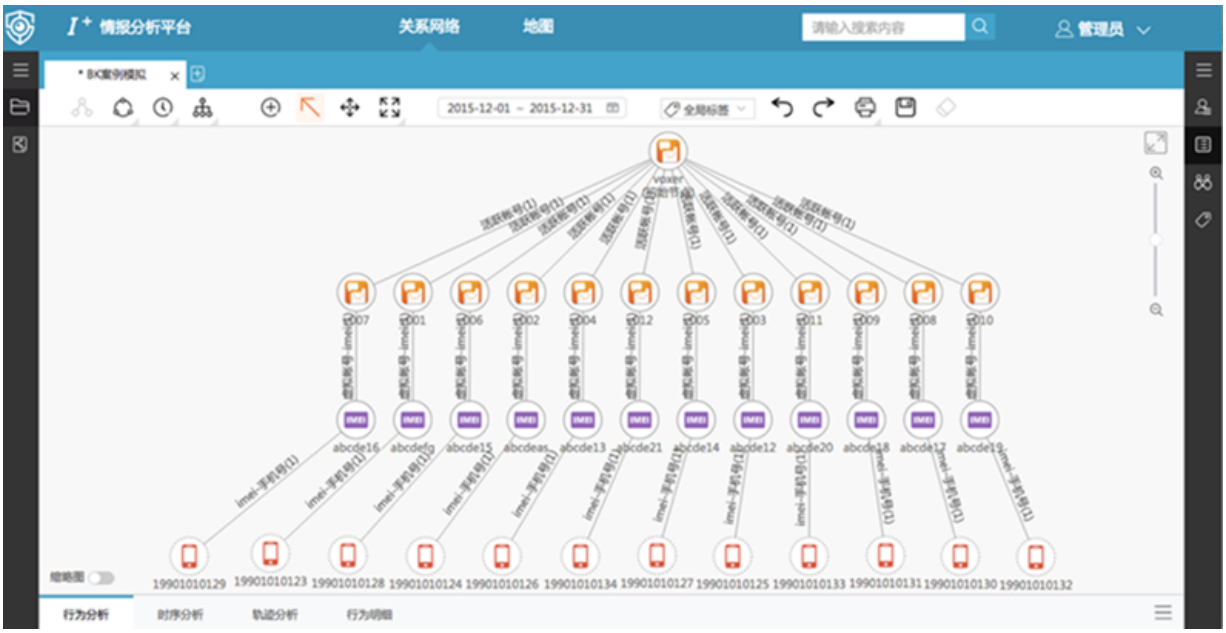
图 11-18: 关系网络图

进一步分析这批账号在这段时间的通联关系，结果显示，这批账号与境外某国的另外一批账号存在单线联系，分析人员尝试获取境外账号的情况，没有查到相关信息。

图 11-19: 单线联系

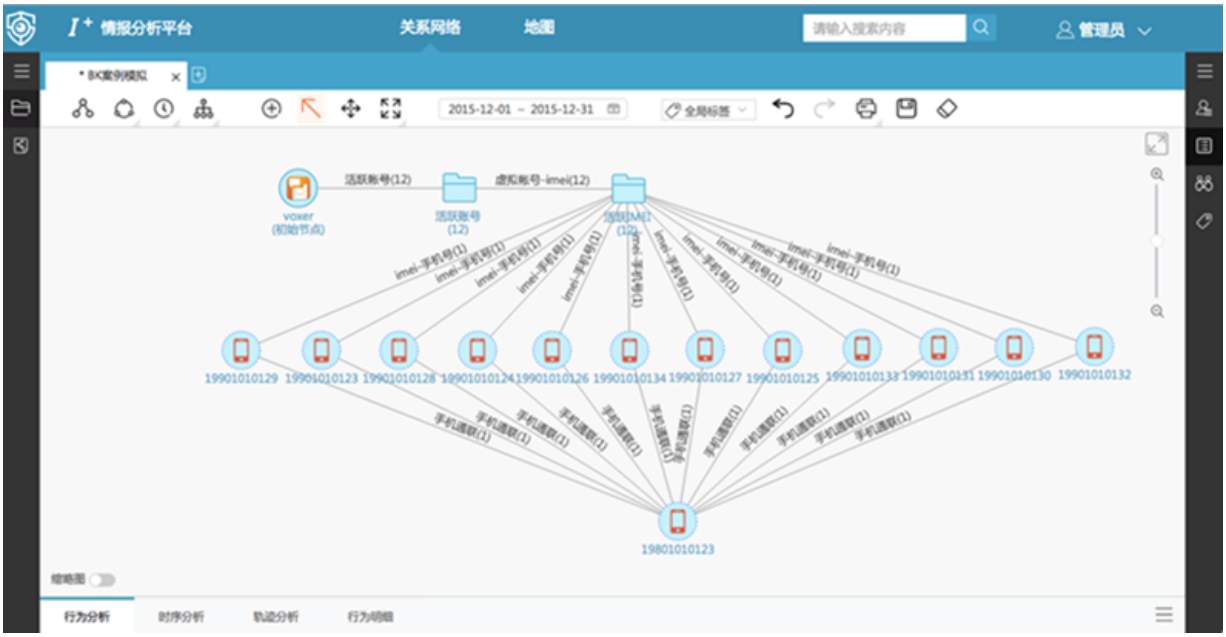
分析人员尝试获取到这批账号的实际使用人，进一步拓展分析发现，这批账号分别在不同的手机上使用，关联反查得到对应的手机号，结果显示，这批手机号只在单独的手机上使用过。

图 11-20: 单独手机使用



对这批手机进行分析，未获取到其开户信息，但发现与同一个手机号码存在联系。

图 11-21: 同一手机联系



进一步分析发现，此号码在多个地点出现。

图 11-22: 多地点出现

接下来，分析人员对该号码展开分析，尝试获取与其同行的其他号码。

图 11-23: 其他号码

结果显示，19701010123与该号码存在6次伴行关系。

图 11-24: 手机通联一

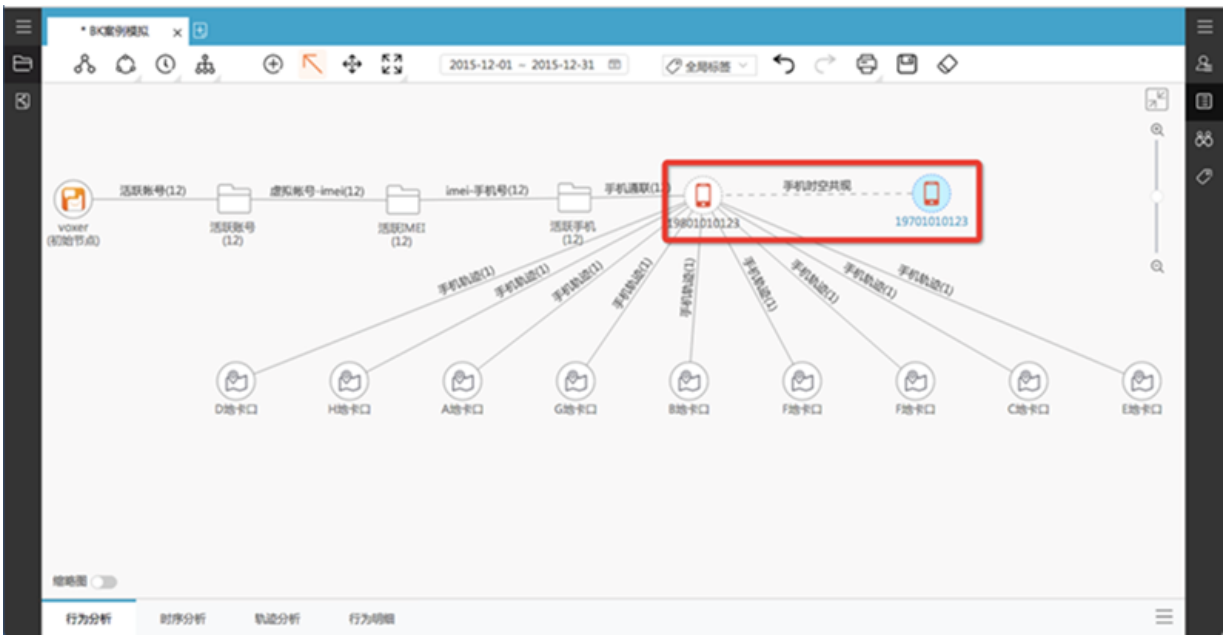
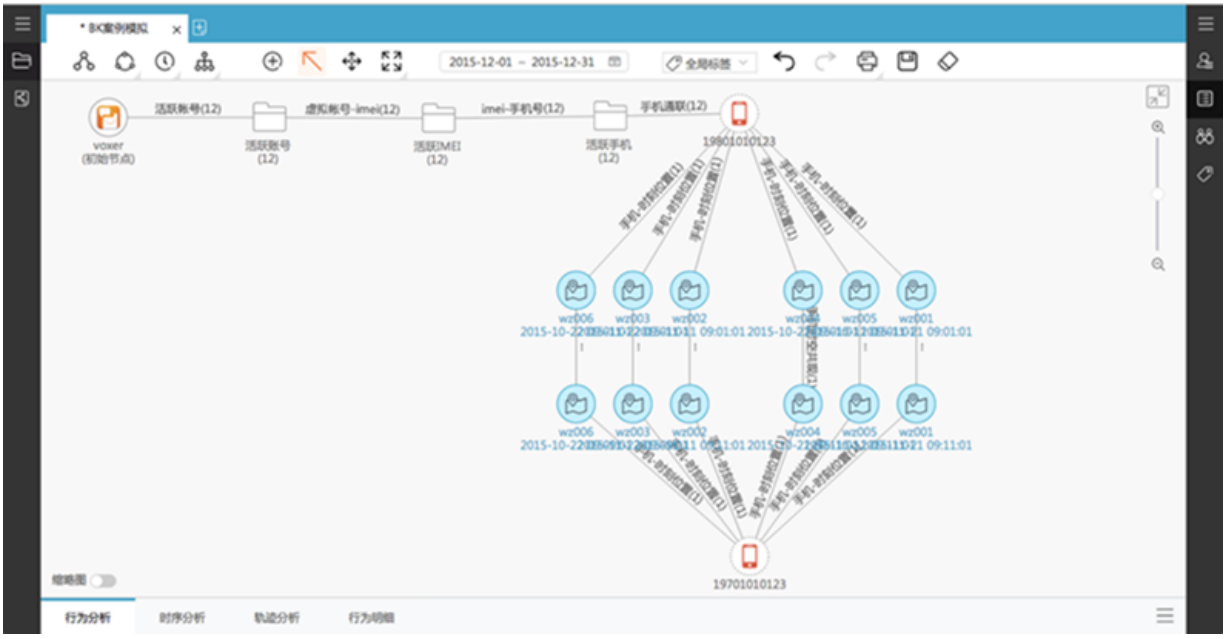
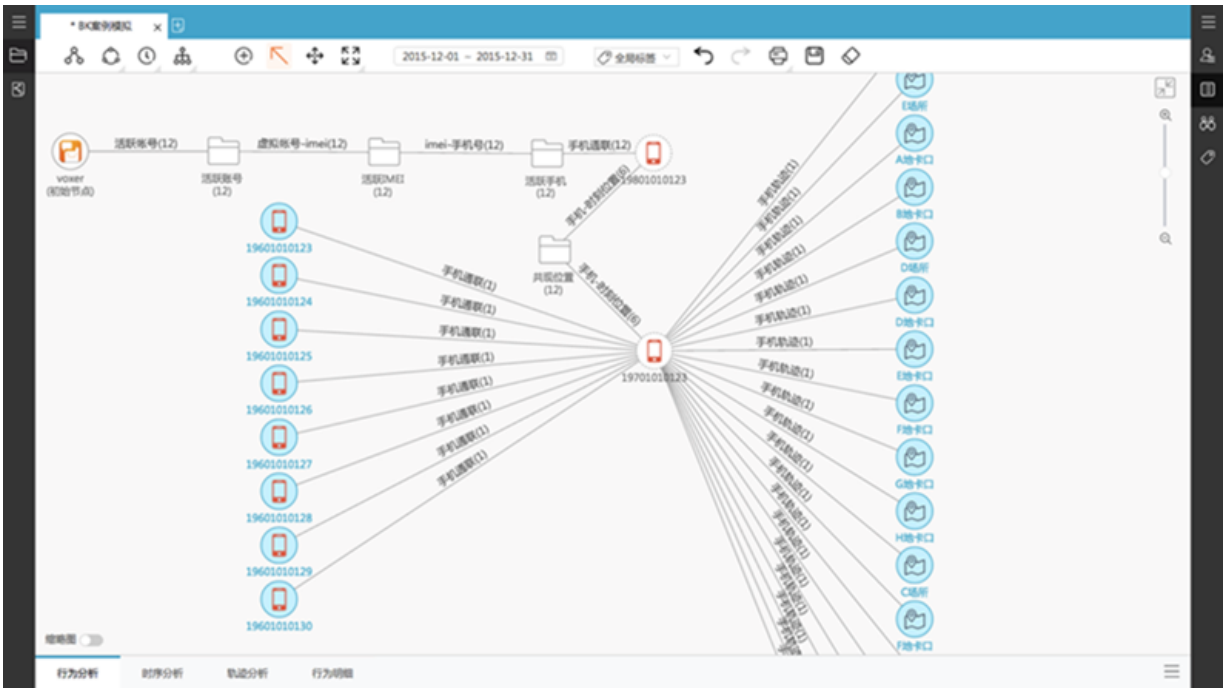


图 11-25: 手机通联二



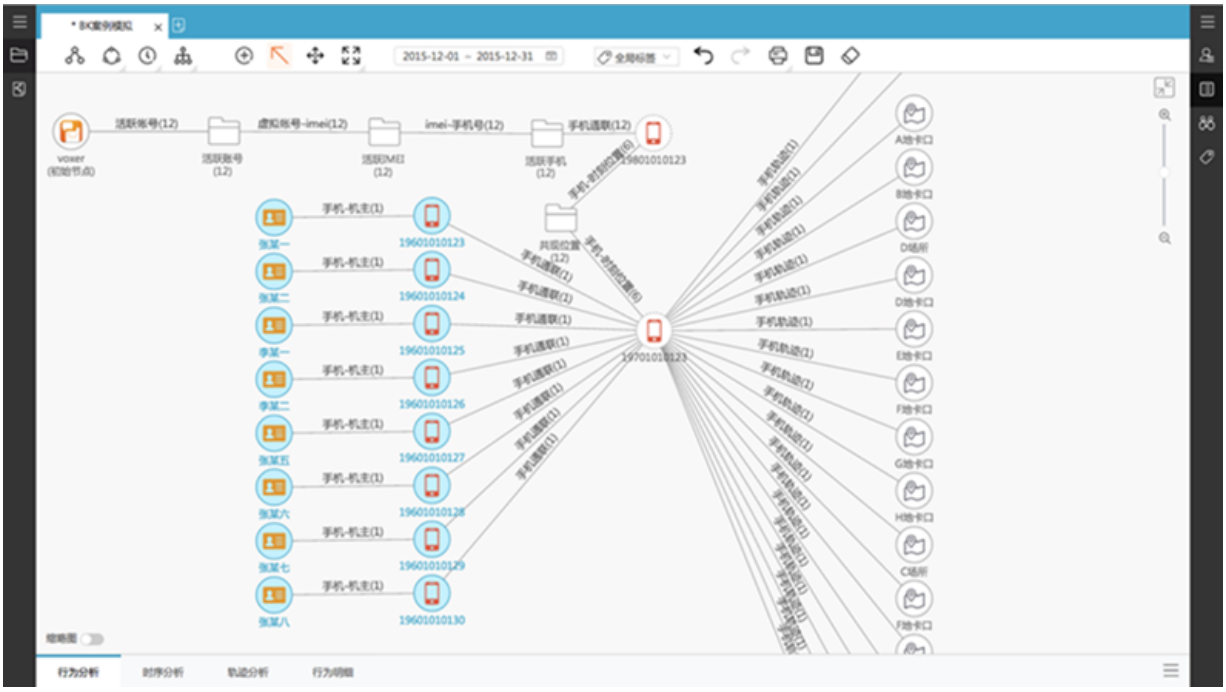
根据经验判断，19701010123很可能为19801010123持有的另一部手机。分析人员对19701010123开展分析，发现该部手机存在大量的通联信息、位置信息。

图 11-26: 手机信息



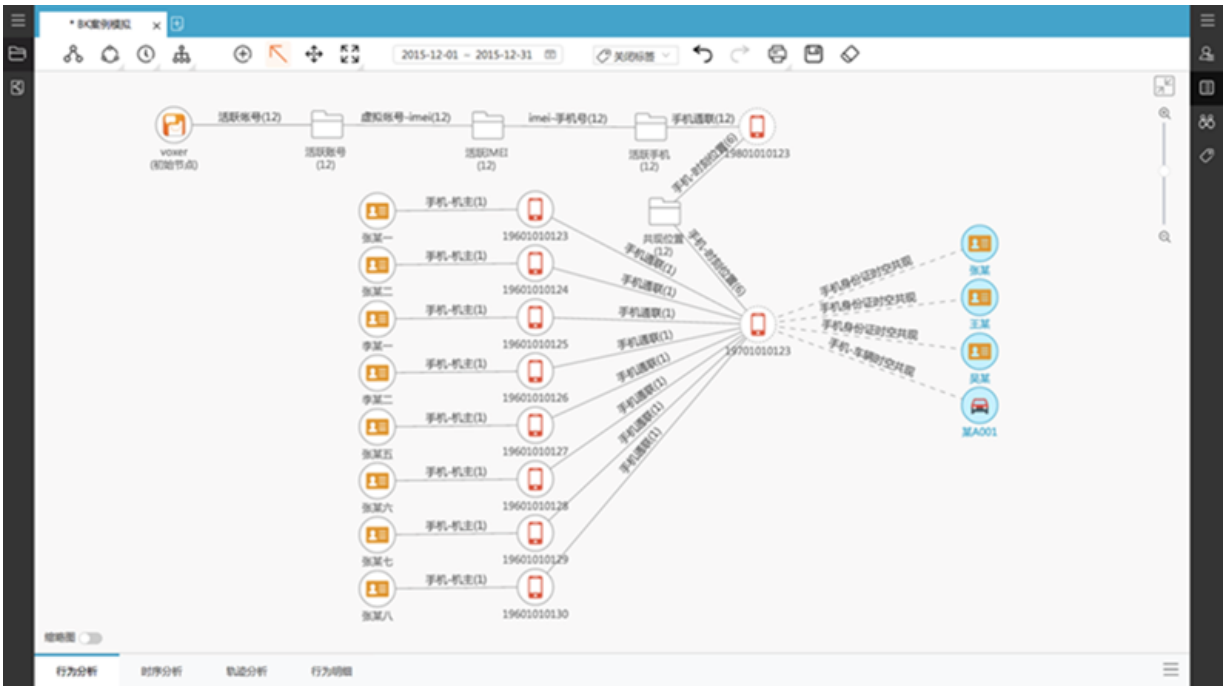
对该手机的通联对象展开分析，发现这批通联号码存在机主登记信息。

图 11-27: 机主登记信息



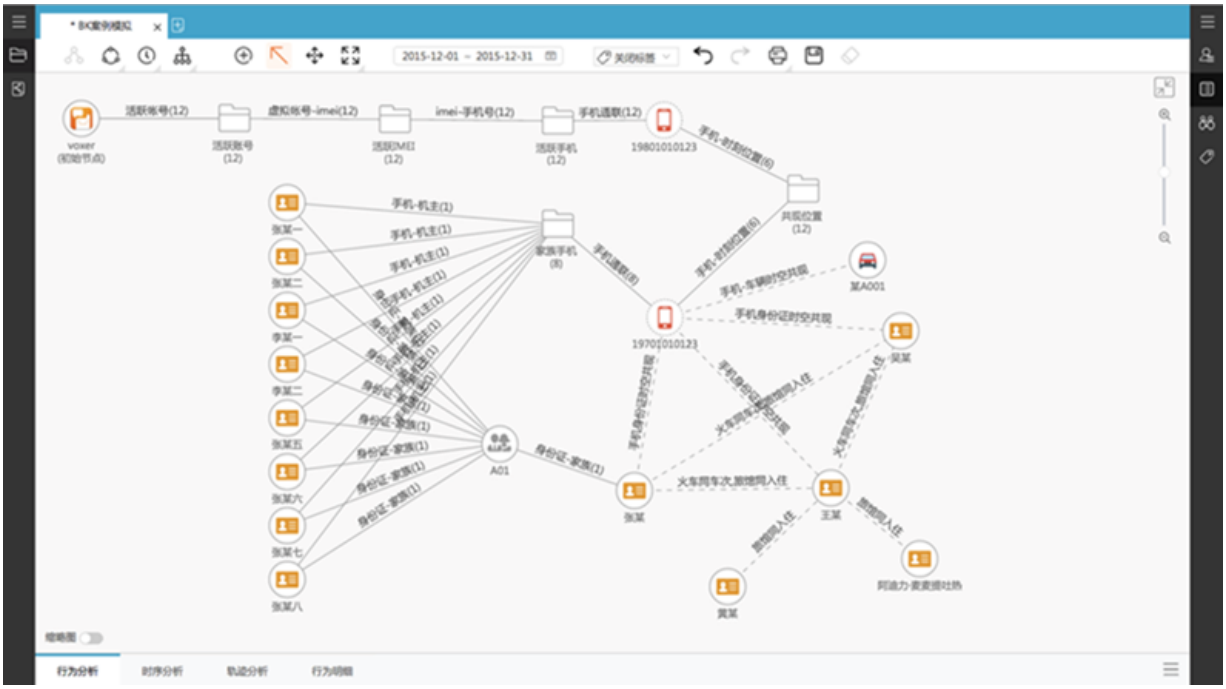
通过对19701010123的轨迹进行分析，发现该号码与张某等三人及某A001车存在伴行关系，此手机号很有可能为此三人所有。

图 11-28: 手机号持有人



进一步对上述发现的身份证进行分析，发现其中9人为同一家族成员，且张某与另外两人存在多次同行同住关系，同时，还挖掘到黄某、尚某与b12345123存在密切往来，三人曾经多次入住相同酒店。

图 11-29: 密切往来人员



对黄某、尚某等四人进行重点分析，发现此四人为SK前科人员，2011年由于涉嫌某BK案件被逮捕，于2015年释放出狱，结合上述异常行为，分析人员判断此四人很可能预谋作案。

图 11-30: 嫌疑人一

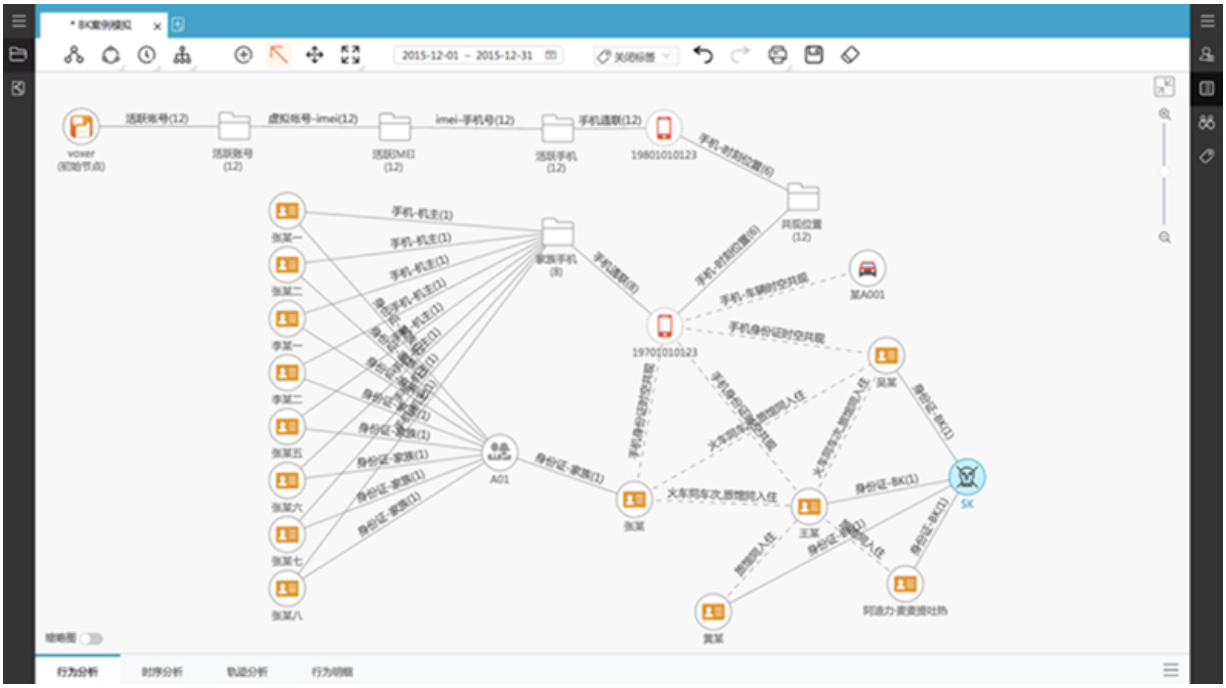
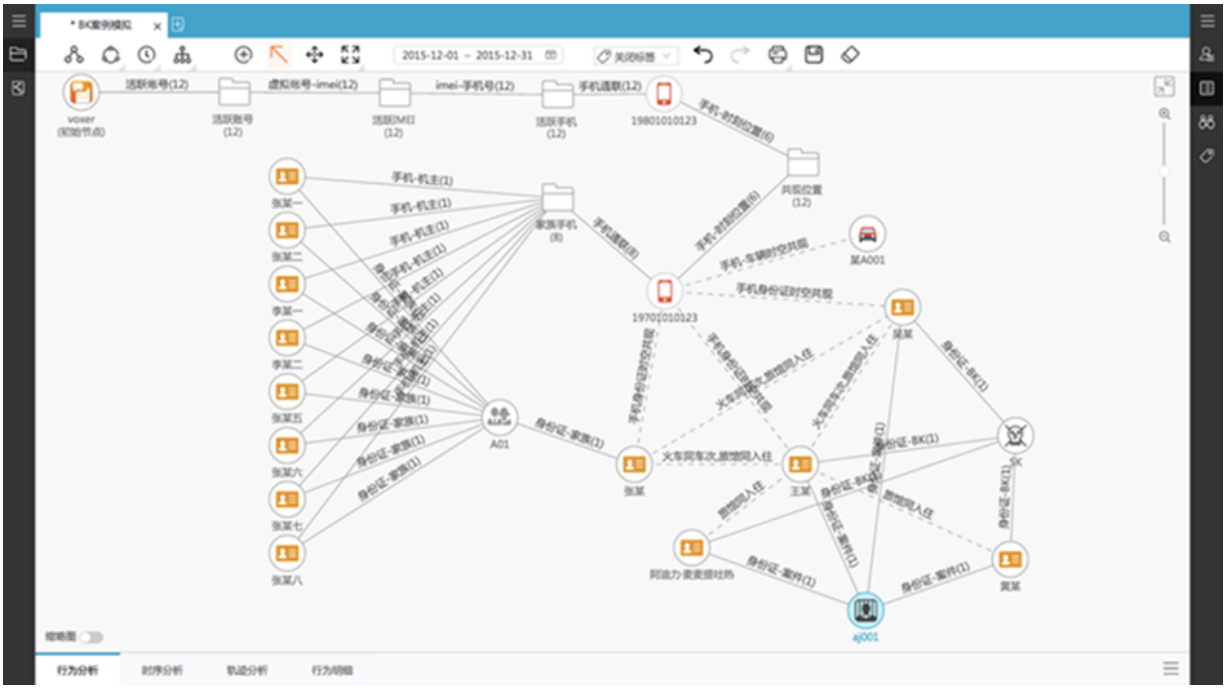


图 11-31: 嫌疑人二



再次拓展分析发现，黄某、尚某以及张某家族共有9人持有护照，并于2009年、2014年、2015年相继往返B国。

图 11-32: 拓展分析

B国为境外SK敏感国家，上述异常监测中所发现的小众通讯工具通联对象所在国家恰好也在B国，分析人员初步认为，张某、黄某、尚某等人很有可能预谋作案。

以上情报立即上报相关部分，经过民警实地排查，在某出租屋内发现张某、黄某、尚某等人正预谋在2016年元旦期间实施BK袭击，相关部门当即调派精干警力，摧毁了该团伙，避免了恶性案件的发生。

案情回顾如下：

图 11-33: 案情回顾

11.7.2 智能关系网络

图 11-34: 智能关系网络一

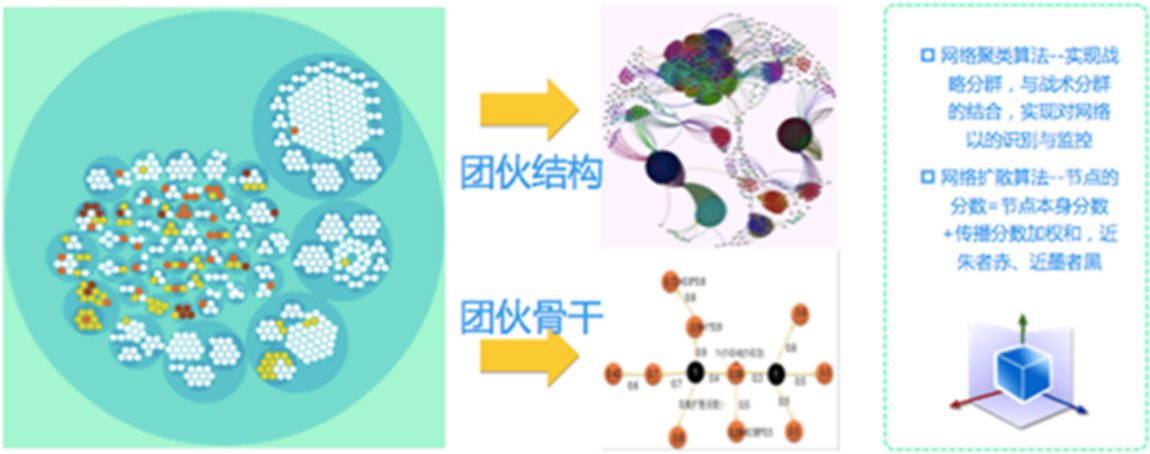


图 11-35: 智能关系网络二

11.7.3 行业风控

图 11-36: 行业风控图

11.7.4 公安安防

图 11-37: 公安安防图