

# 第四章 离线数据开发

阿里数据计算层：

- 数据存储及计算平台
  - 离线计算平台MaxCompute
  - 实时计算平台StreamCompute
- 数据整合及管理体系： OneData

## 一、数据开发平台

阿里数据研发岗工作：了解需求 → 模型设计 → ETL 开发 → 测试 → 发布上线 → 日常运维 → 任务下线。

阿里数据研发岗与传统数仓开发（ETL）对比：

- 业务变更频繁
- 需要快速交付
- 频繁发布任务
- 运维任务多
- 系统环境复杂

对于上述痛点，阿里通过统一的计算平台（MaxCompute）、统一的开发平台（D2等相关平台和工具）、统一的数据模型和统一的数据研发规范来处理。

### 1.1 统一计算平台

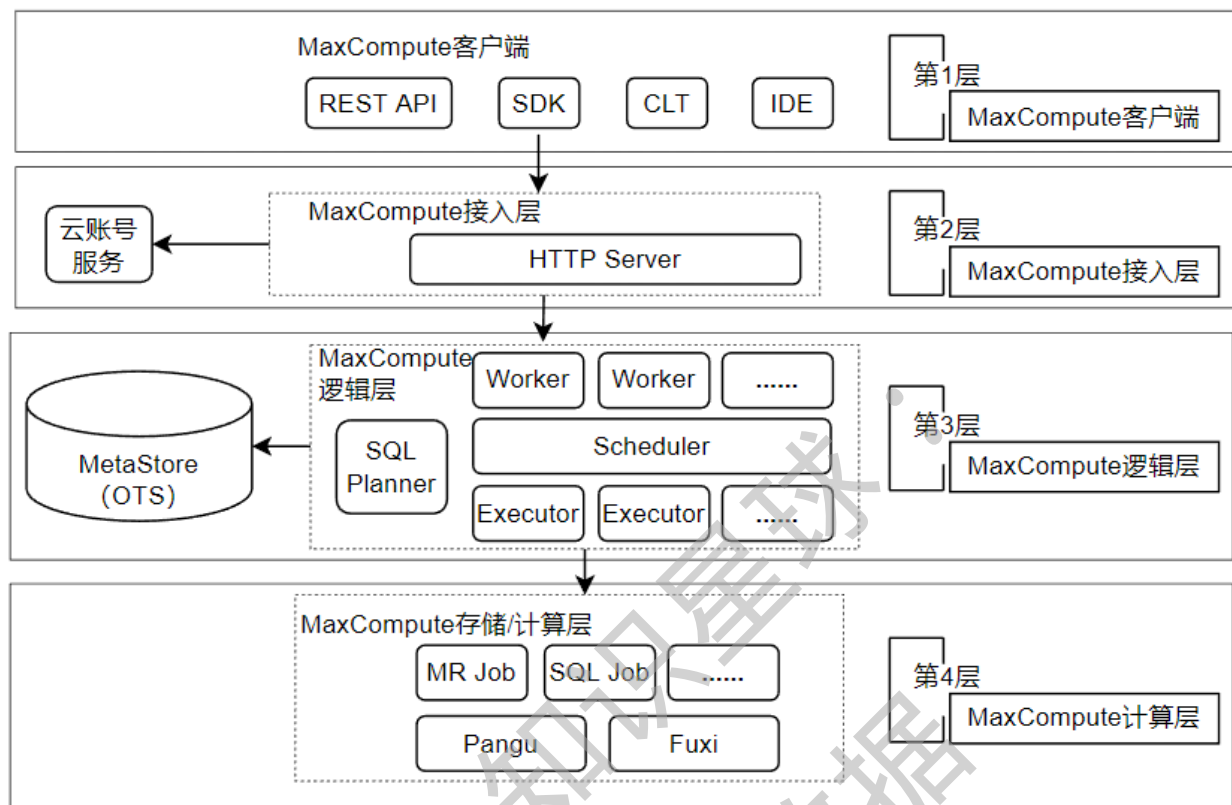
阿里离线数据仓库的存储和计算都是在**阿里云自主研发的海量数据处理平台MaxCompute**上完成的。

MaxCompute主要服务于海量数据的存储和计算，提供完善的数据导入方案，以及多种经典的分布式计算模型，提供海量数据仓库的解决方案，能够更快速地解决用户的海量数据计算问题，有效降低企业成本，并保障数据安全。

**MaxCompute工作原理：**

采用抽象的作业处理框架，将不同场景的各种计算任务统一在同一个平台上，共享安全、存储、数据管理和资源调度，为来自不同用户需求的各种数据处理任务提供统一的编程接口和界面。它能提供数据上传/下载通道、SQL、MR、机器学习算法、图编程模型和流式计算模型等多种计算分析服务，并提供完善的安全解决方案。

**MaxCompute体系架构**



### MaxCompute组成

- 客户端（MaxCompute Client）：提供RESTful API，命令行，可视化ETL/BI工具服务
- 接入层（MaxCompute Front End）：提供HTTP服务、Cache、负载均衡，实现用户认证和服务层面的访问控制。
- 逻辑层（MaxCompute Server）：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。
  - 逻辑层有Worker、Scheduler、Executor三个角色：Worker处理所有的RESTful请求，Scheduler负责Job的调度和拆解，Executor负责Job的执行。
- 计算层：飞天内核，运行在独立的计算集群上。

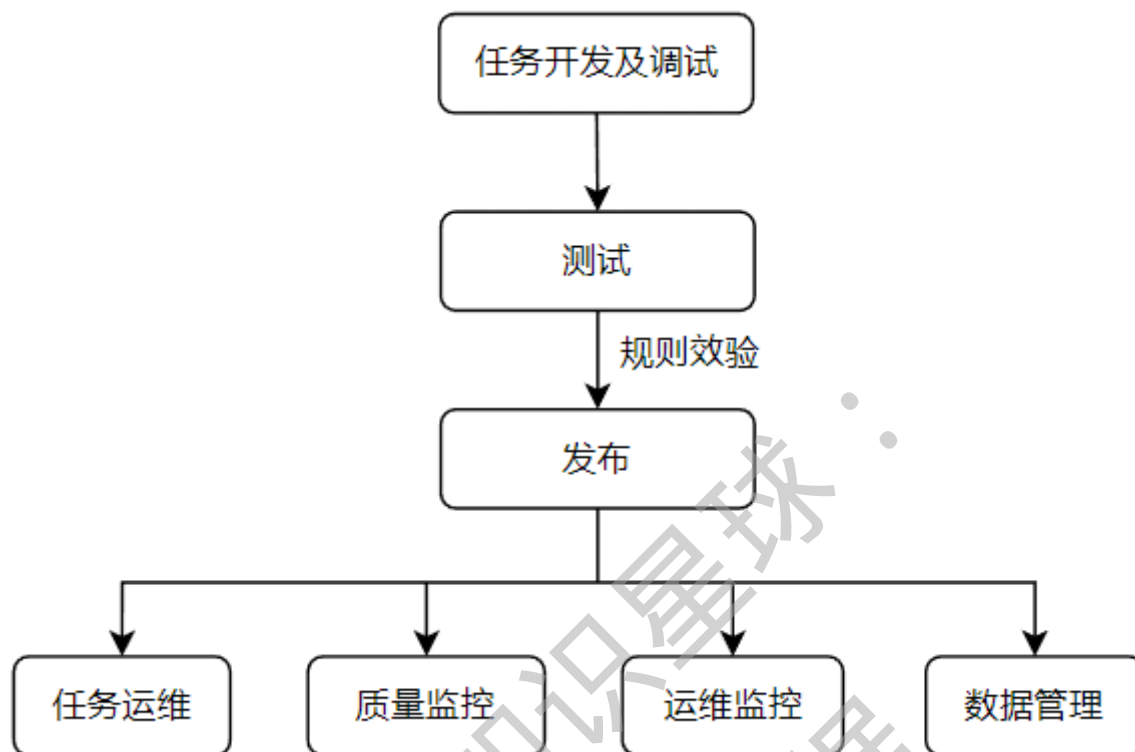
### MaxCompute特点

- 计算性能高且更加普惠
- 集群规模大且稳定性高
- 功能组件非常强大
- 安全性高

## 1.2 统一开发平台

阿里数据开发平台集成了多个子系统来解决实际生产中的各种痛点。围绕MaxCompute计算平台，从任务开发、调试、测试、发布、监控、到运维管理，形成了整套工具和产品，既提高了开发效率，又保证了数据质量，并且在确保数据产出时效的同时，能对数据进行有效管理。

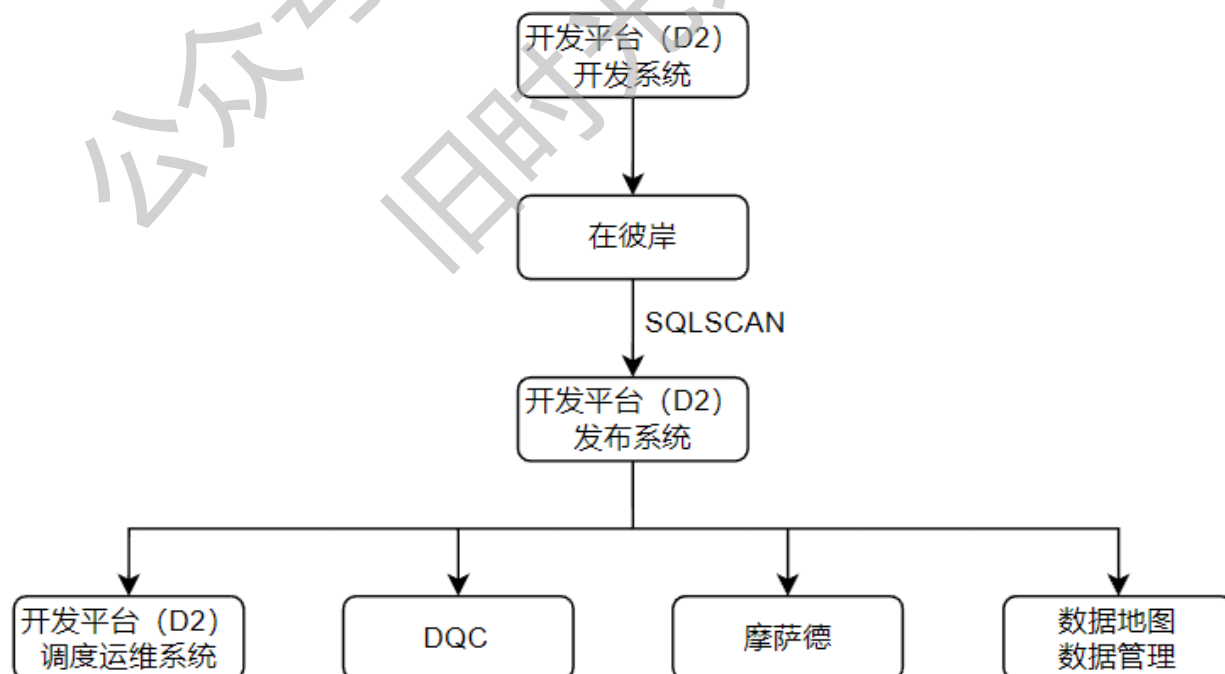
具体开发流程如下：



### 1、在云端 (D2)

D2是集成任务开发、调试及发布，生产任务调度及大数据运维数据权限申请及管理等功能的一站式数据开发平台 并能承担数据分析工作台的功能。

开发工作流对应的的产品和工具



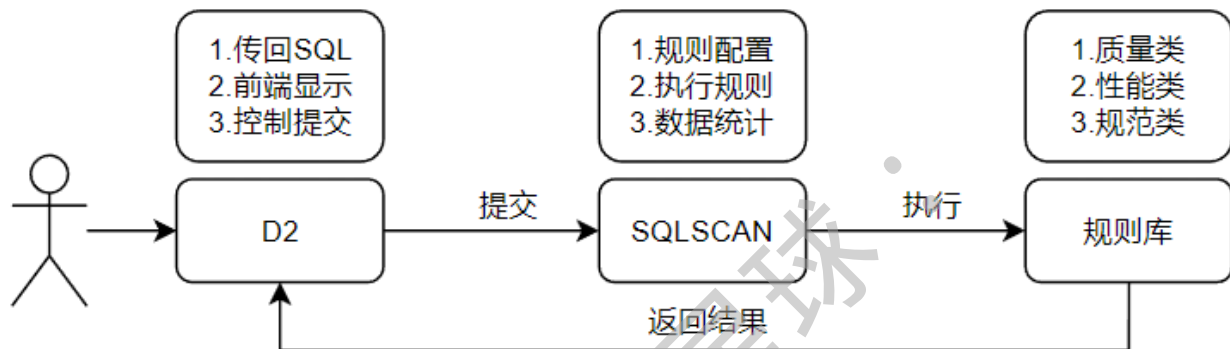
使用D2进行数据开发的基本流程：

- 用户通过IDE创建计算节点，SQL/MR任务，或者Shell、数据同步任务等
- 提交任务后，节点进入开发环境中，并成为某个工作流的其中一个节点
- 节点运行无误后，用户点击发布，将该节点正式提交到生产环境，上线

## 2、SQLSCAN

SQLSCAN将开发中遇到的问题（比如SQL质量差、性能低等）总结形成规则，通过系统及研发流程保障，任务执行前解决故障隐患。

SQLSCAN工作流程



SQLSCAN主要有三类规则效应：

- 代码规范类规则，如表命名规范、生命周期设置、表注释等
- 代码质量类规则，如调度参数使用检查、分母为0提醒、NULL值参与计算影响结果提醒、插入字段顺序错误等
- 代码性能类规则，如分区裁剪失效、扫描大表提醒、重复计算检测等

SQLSCAN规则：

- 强规则：触发后，任务提交会被阻断，必须修复代码后才能再次提交
- 弱规则：只会显示违反规则的提示，用户可以继续提交任务

## 3、DQC

DQC（Data Quality Center，数据质量中心）主要关注数据质量，通过配置数据质量校验规则，自动在数据处理任务过程中进行数据质量方面的监控。

DQC功能：

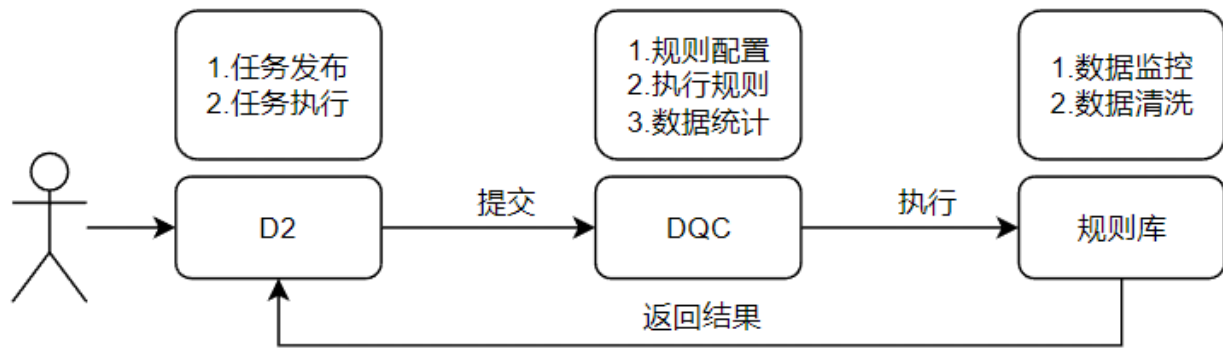
- 数据监控：只监控数据质量并报警，本身不处理数据
- 数据清洗：是将不符合既定规则的数据清洗掉，保障数据质量，不会触发报警

DQC数据监控规则有强规则和弱规则之分，详情见SQLSCAN规则

常见的DQC监控规则有：主键监控、表数据量及波动监控、重要字段的非空监控、重要枚举字段的离散值监控、指标值波动监控、业务规则监控等。

**阿里数据仓库的数据清洗采用非侵入式的清洗策略，在数据同步过程中不进行数据清洗，避免影响数据同步的效率，其过程在数据进入ODS层之后执行。**

DQC工作流程

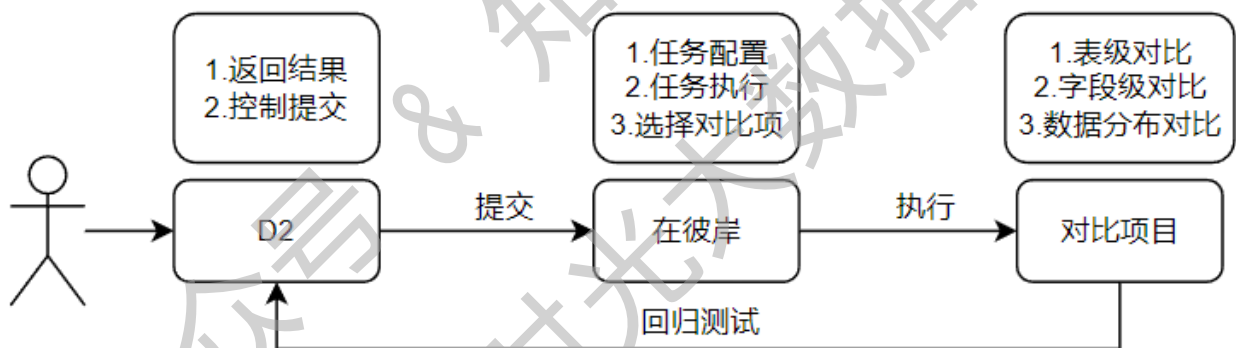


#### 4、在彼岸

在彼岸是自动测试平台，主要用于解决下列测试问题：

- 对**新增业务需求**，比如报表、应用、产品需求或ETL任务测试，其主要对目标数据和源数据进行对比，包括数据量、主键、字段空值、字段枚举值、复杂逻辑（如 UDF、多路分支）等的测试
- **数据迁移、重构和修改**测试，为了严格保证数据质量，对于优先级大于某个阈值的任务，强制要求必须使用在彼岸进行回归测试

在彼岸进行回归测试的流程



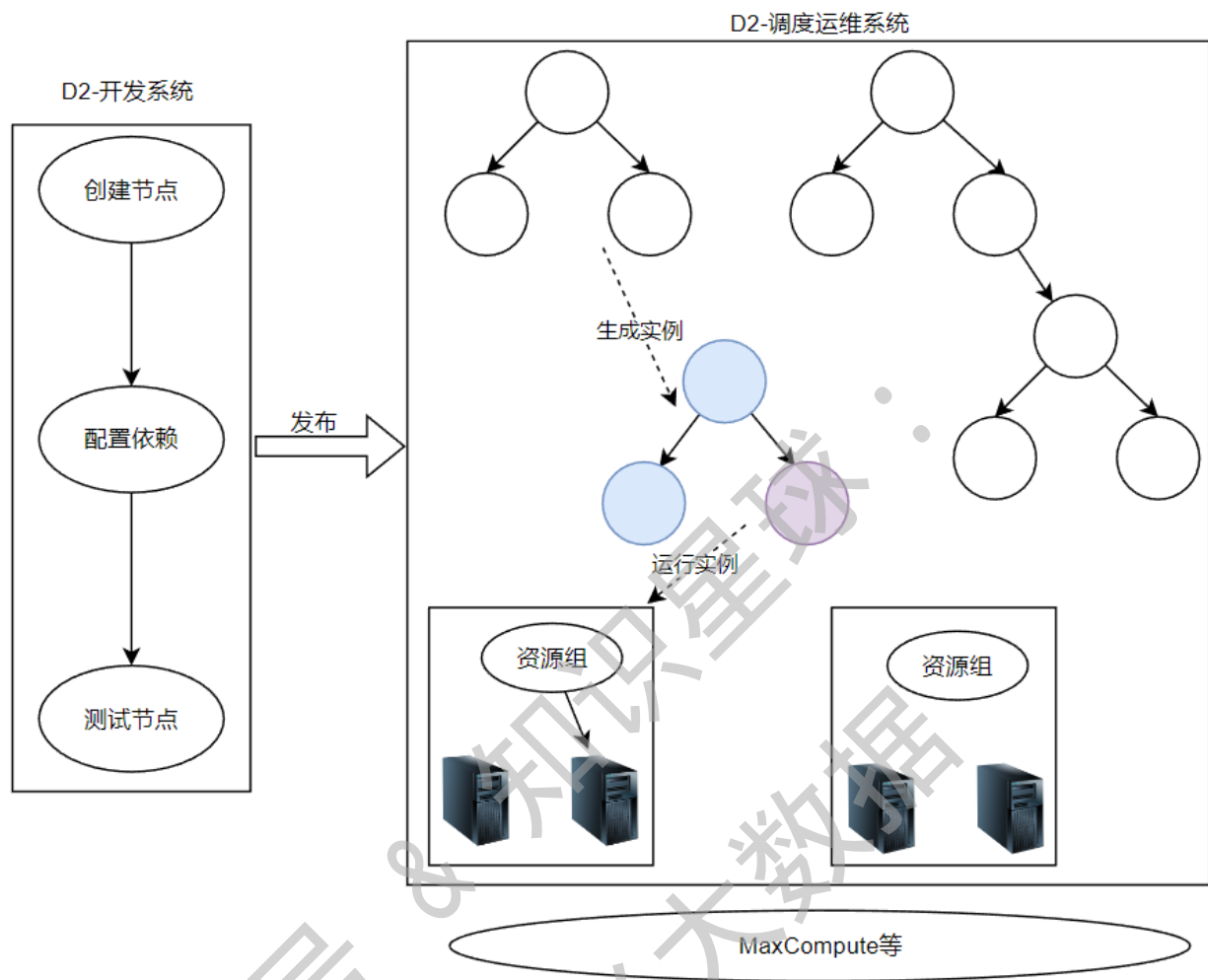
## 二、任务调度系统

云计算大数据时代，调度系统是整個大数据体系的指挥中枢。

为了处理海量、繁杂的任务，阿里的大数据调度系统应运而生。

### 2.1 D2调度系统的核心设计模型：调度引擎和执行引擎

#### 1、数据开发流程与调度系统的关系



用户通过D2平台提交、发布的任务节点，需要通过调度系统，按照任务的运行顺序调度运行。

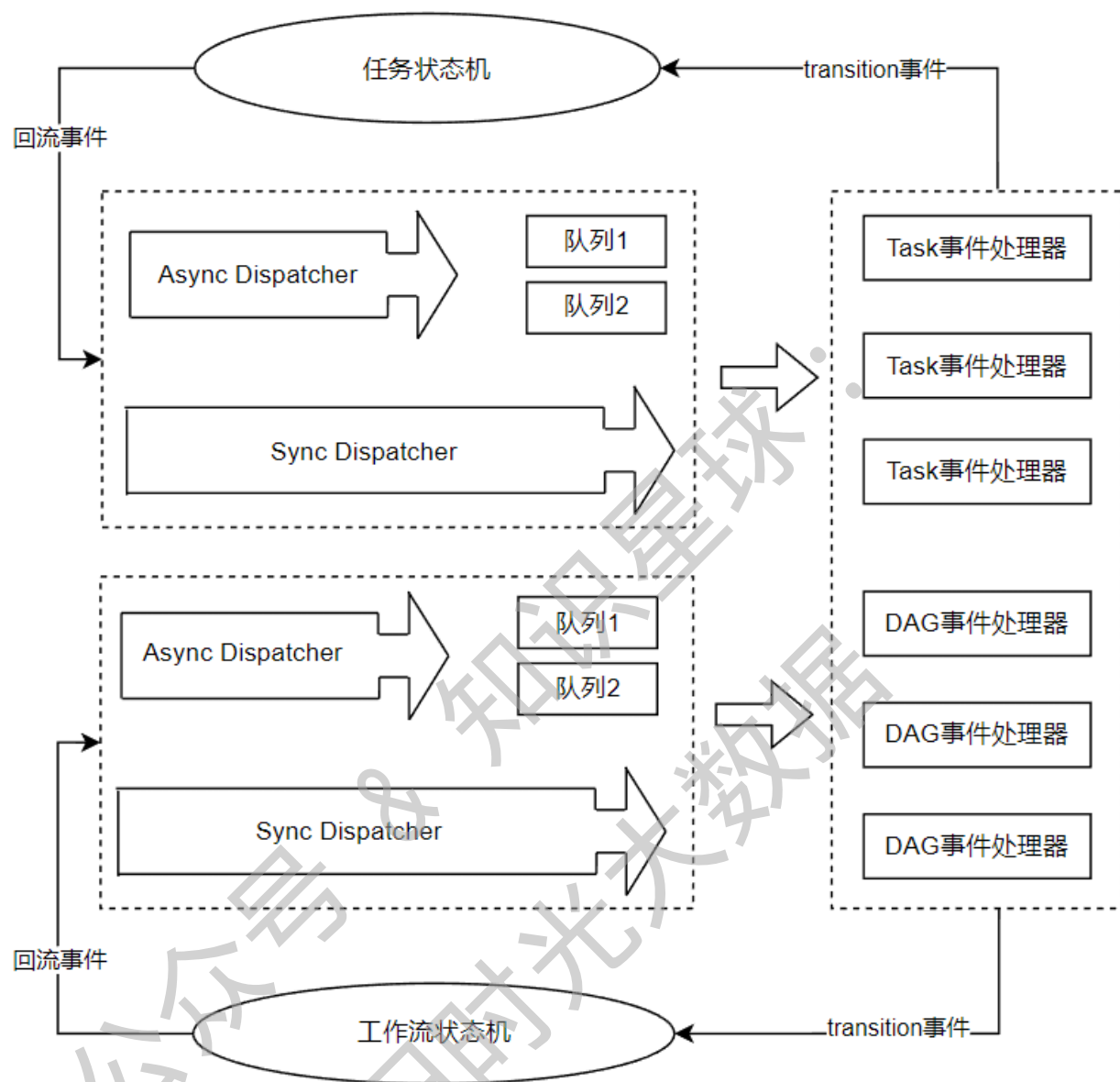
## 2、调度系统的核心设计模型

调度引擎（Phoenix Engine）：根据任务节点属性及依赖关系进行实例化，生成各类参数的实值，并生成调度树

执行引擎（Alisa）：根据调度引擎生成的具体任务实例和配置信息，分配CPU、内存、运行节点等资源，在任务对应的执行环境中运行节点代码。

## 3、调度引擎工作原理

调度引擎是基于任务状态机模型和工作流状态机模型原理，以事件驱动的方式进行，为数据任务节点生成实例，并在调度树中生成具体执行的工作流。调度引擎只涉及任务状态机的未运行和等待运行两种状态。



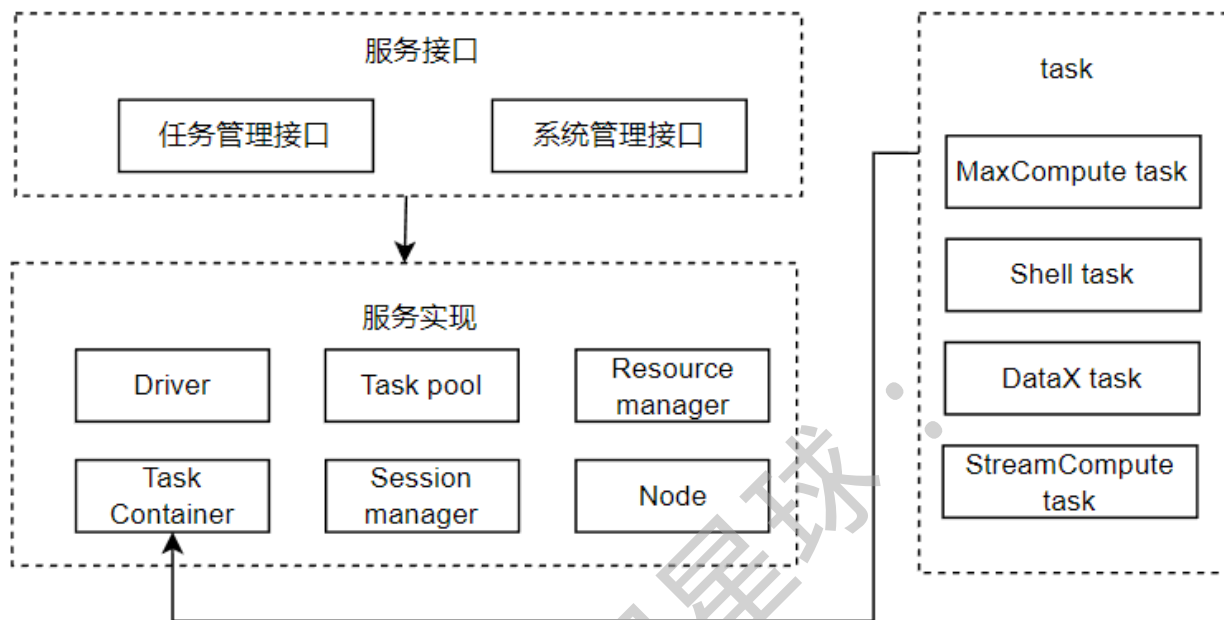
Async Dispatcher: 异步处理任务调度

Sync Dispatcher: 同步处理任务调度

Task事件处理器: 任务事件处理器, 与任务状态机交互

DAG事件处理器: 工作流事件处理器, 与工作流状态机交互, 一个DAG事件处理器包含若干个Task件处理器

#### 4、执行引擎工作原理



任务管理接口：供用户系统向 Alisa 中提交、查询和操作离线任务，并获得异步通知。

系统管理接口：供系统管理员进行后台管理。

Driver：Alisa的调度器，Driver中实现了任务管理接口和系统管理接口；负责任务的调度策略、集群容灾和伸缩、任务失效备援、负载均衡实现。

Task pool：采用Zookeeper实现，具备高可用能力，管理Driver提交的任务，直到任务完成，包括等待资源、数据质量检测、运行中、运行成功和失败的所有任务。

Resource manager：管理集群整体资源。

Task container：为Task提供运行的容器。

Session manager：管理Task session。

Node：代表Alisa集群中的一个节点。节点负责提供运行所需的物理资源。

## 5、执行引擎用法

提供数据处理任务的调度服务。

## 2.2 调度系统的特点及应用

### 1、调度配置

常见调度配置方式存在两个问题：

- 配置麻烦，需要知道上游依赖表的产出任务
- 上游任务修改不再产出依赖表或本身任务不再依赖某上游任务时，对调度依赖不做修改，导致依赖配置错误。

阿里的调度配置方式采用的是输入输出配置和自动识别相结合的方式。当任务提交时，SQL 解析引擎自动识别此任务的输入表和输出表，输入表自动关联产出此表的任务，输出表也是这样。通过这种方式，可以自动调整任务依赖，避免依赖配置错误。

### 2、定时调度



可以根据实际需要，设定任务的运行时间，共有5种时间类型：分钟、小时、日、周、月，具体可精确到秒。对于周任务和月任务，通常选择定时调度的方式。

### 3、周期调度

可按照小时、日等时间周期运行任务，与定时调度的区别是无须指定具体的开始运行时间。

任务根据依赖关系，按照调度树的顺序从上依次向下运行，每个周期的具体运行时间随着每天资源和上游依赖任务运行的总体情况会有所不同。

### 4、手动执行

当生产环境需要做一些数据修复或其他一次性的临时数据操作时，可以选择手动运行的任务类型。比如在开发环境写好脚本后发布到生产环境，再手动触发运行。

### 5、补数据

在完成数据开发的发布以后，有些表需要进行数据初始化。比如有些日增量表要补齐最近三年的历史数据。

可以设定需要补的时间区间，并圈定需要运行的任务节点，从而生成一个补数据的工作流，同时还能选择并行的运行方式以节约时间。

### 6、基线管理

调度系统引入了按优先级分类管理的方法，系统会先保障高优先级任务的运行资源。

对于同一类优先级的任务，放到同一条基线中，这样可以实现按优先级不同进行分层的统一管理，并可对基线的运行时间进行预测估计，以监控是否能在规定的时间内完成。

### 7、监控报警

调度系统有一套完整的监控报警体系，包括针对出错的节点、运行超时未完成的节点，以及可能超时的基线等，设置电话、短信、邮件等不同的告警方式，实现了日常数据运维的自动化。



知识星球

长按扫码领取优惠 ▶

