袋鼠云
DTSTACK

分布式调度引擎——DAGScheduleX

分享者—偷天

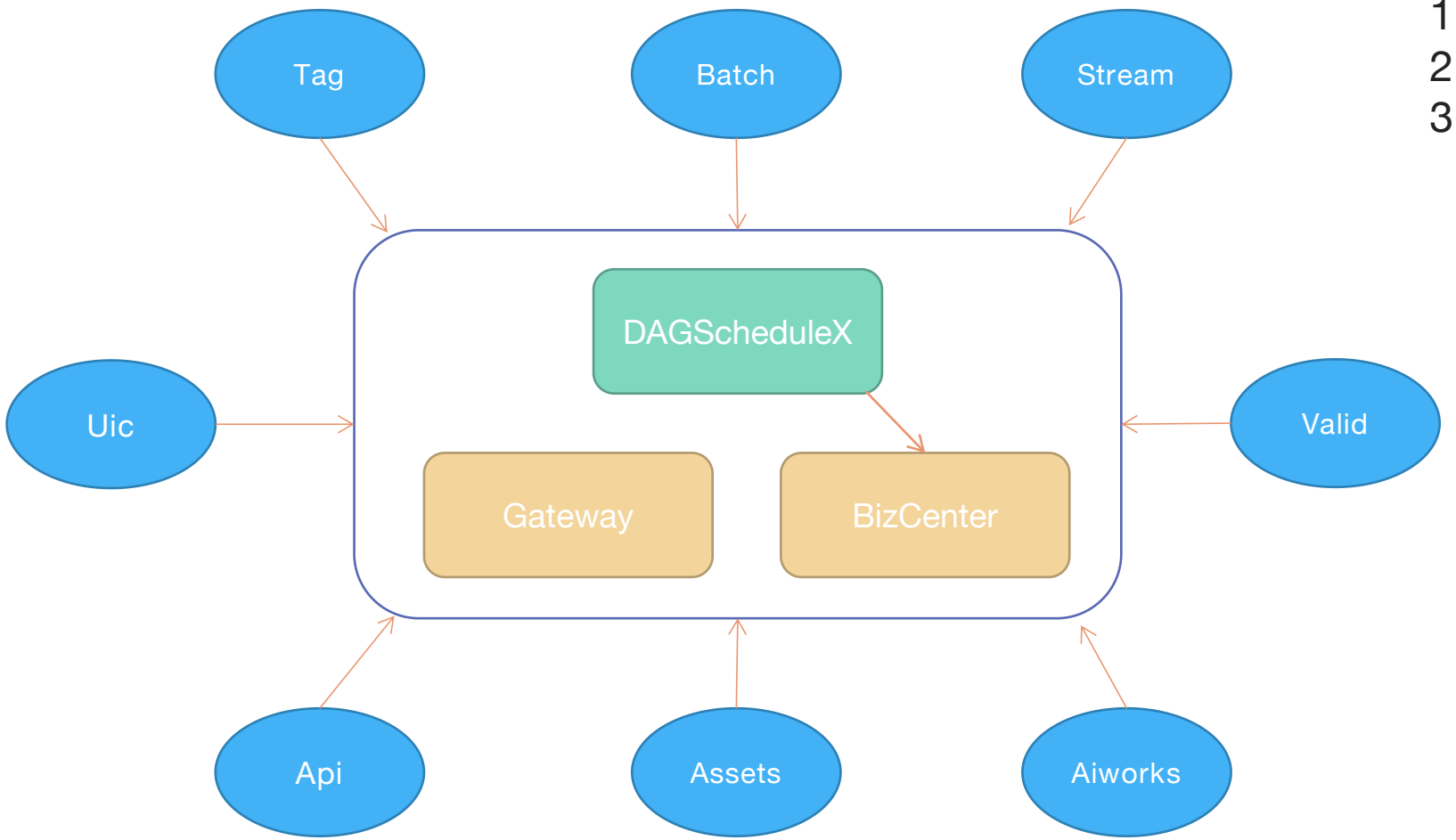
数连万物，栈通中台

目录

CONTENTS

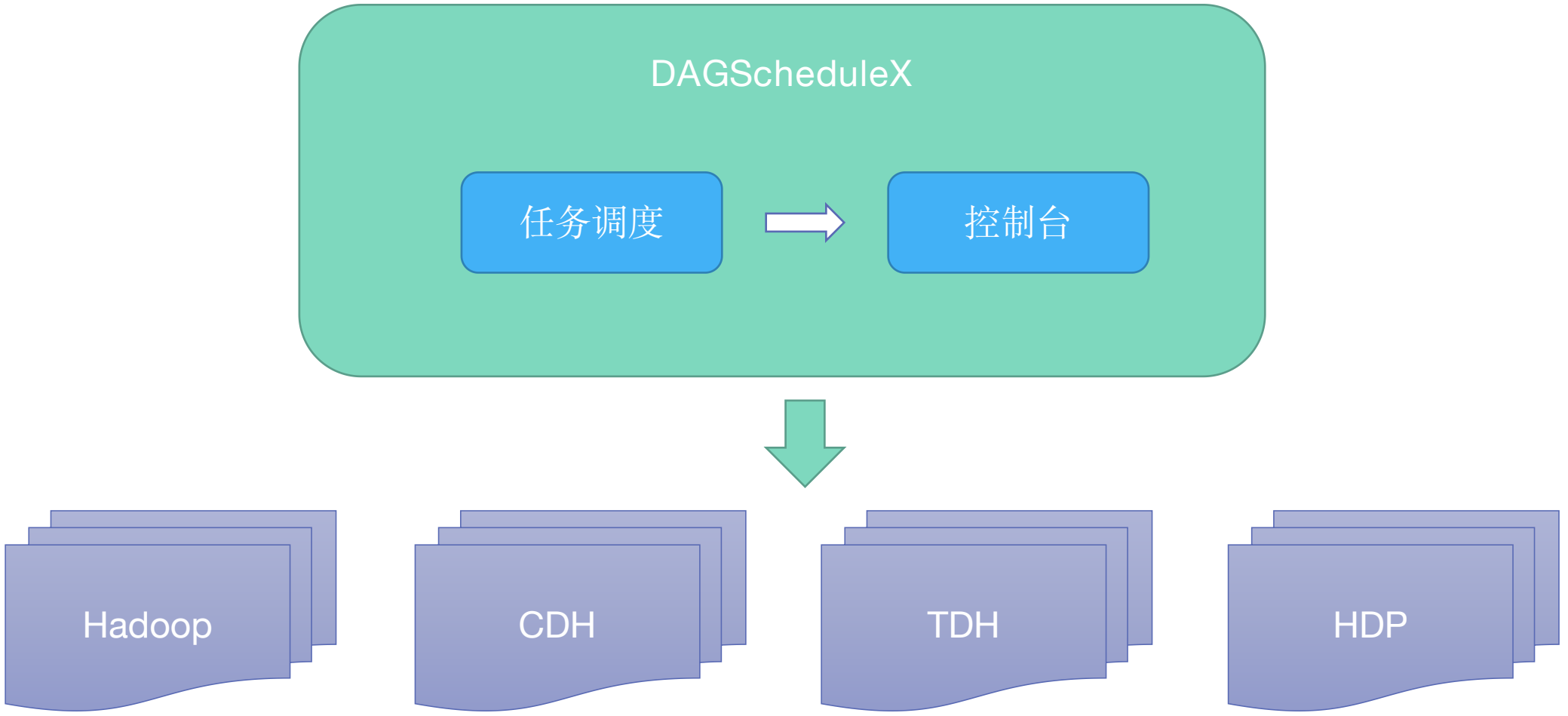
DAGScheduleX 是数栈基础组件之一。DAGScheduleX 向上对接各个上层应用（离线开发、实时开发、算法开发、标签引擎、数据服务、数据质量、数据资产），向下兼容多集群多版本（Hadoop、CDH、TDH、HDP）、SQL引擎（LibrA、Oracle、TiDB、Greenplum），从而实现任务实例的分布式调度与运行。

应用架构

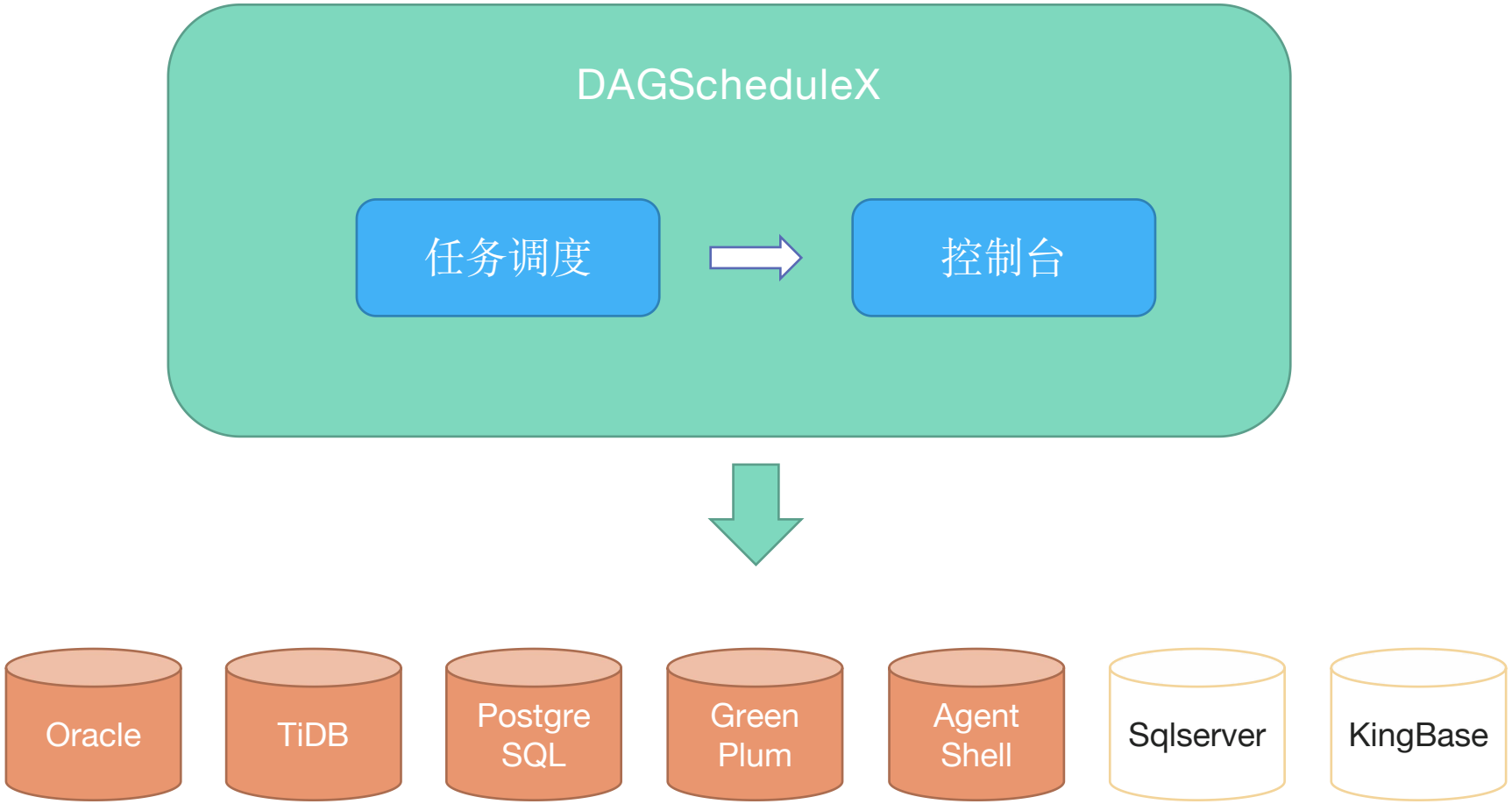


1. Weak Dependence
2. DB Isolation
3. Cache Sharing

兼容多集群多版本



SQL引擎（轻量级）



shell 任务压测，单实例2G内存：1w/15min，日增150w+周期实例

任务实例分布式调度

袋鼠云 DTSTACK

01

实例构建

解析任务DAG图，根据周期性生成周期实例，负载均衡到各个执行节点；

02

调度执行

分布式节点同时负责实例的调度执行；以上下游关系调度实例；

03

状态管理

实例的状态对应当前任务的各个执行阶段，状态有标准的流转；

04

任务类型支持

插件化加载，以兼容不同计算组件不同版本；可扩展性高；

目录

CONTENTS

实例特点

周期性调度

支持5种（分钟/小时/天/周/月）不同周期粒度.

DAG依赖

实例存在上下游依赖关系，保障上下游的实例
按顺序、时间要求进行实例调度

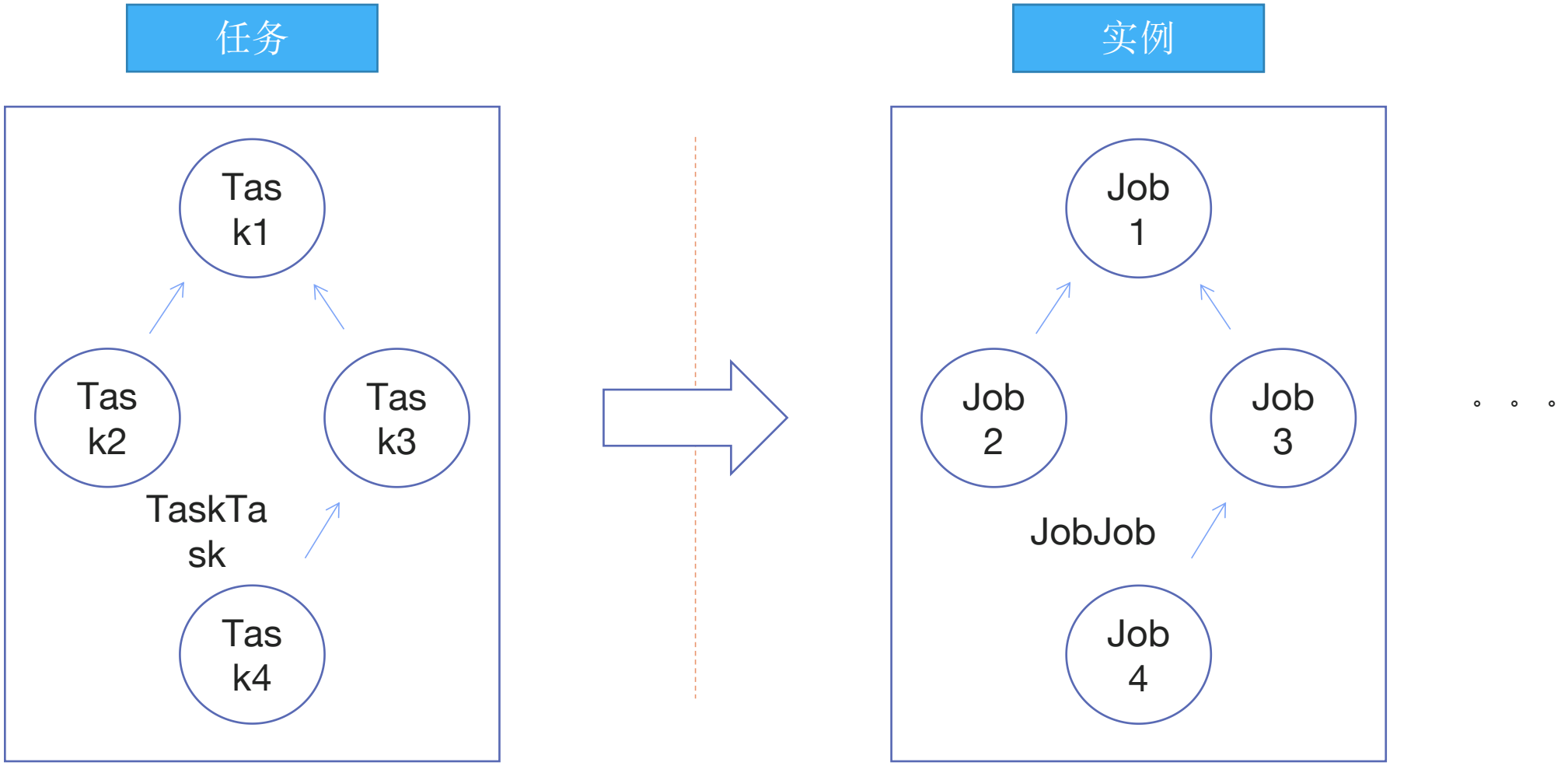
实例 T+1 生成

周期实例T+1的方式生成，可配置修改生成时间
前一天会生成当天的所有实例数据

实例优先级

实例通过优先级队列处理
优先级保证弱一致，提高并发性能

实例构建（1）

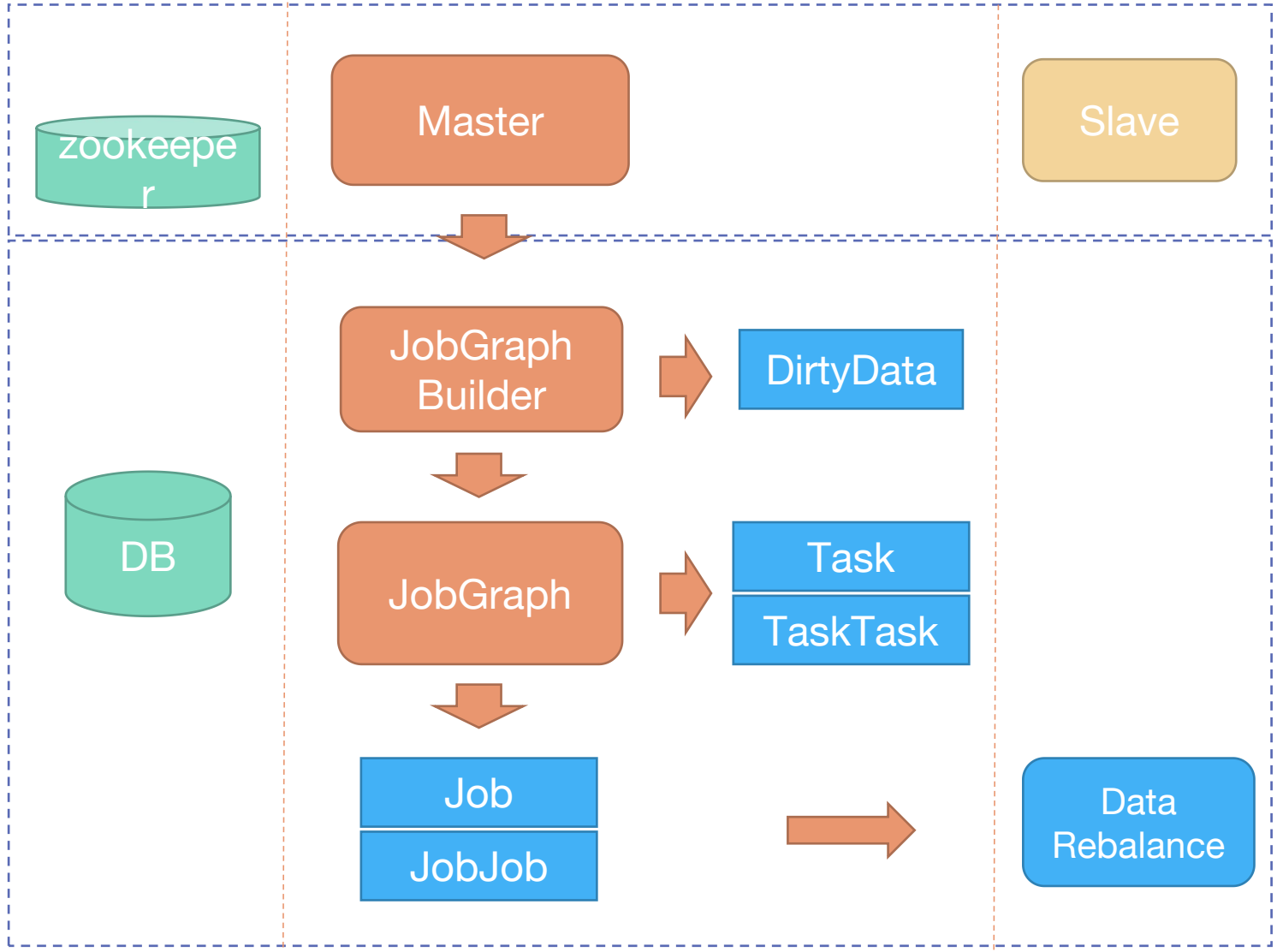


DAG有向无环图

Job 数量与周期类型紧密相关

实例构建（2）

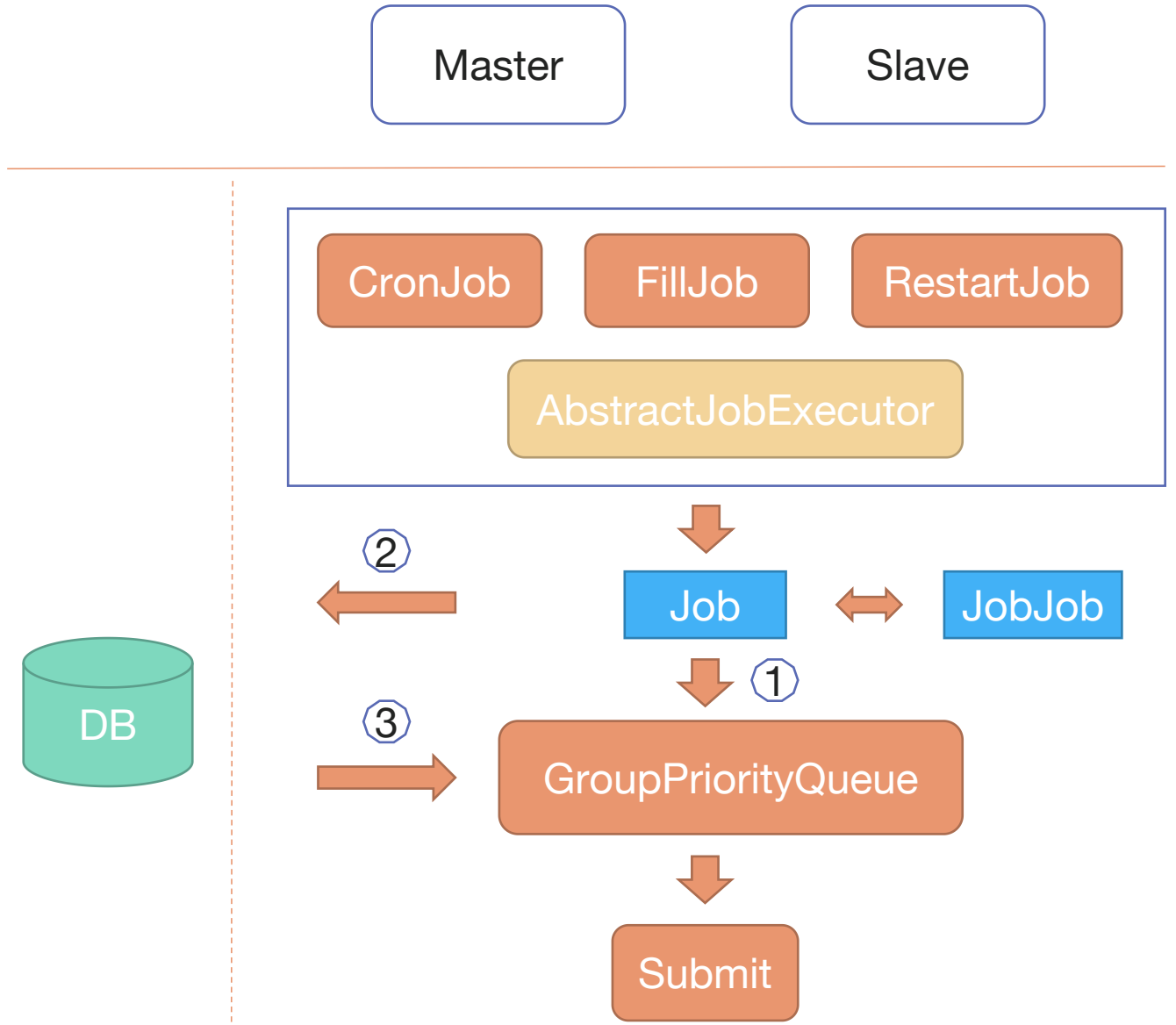
1. 基于 Zookeeper 选举 Master 节点参与 Job 实例构建
2. JobGraph 构建前 check & clean DirtyData
3. 依据 Task、TaskTask 的数据（JobGraph）生成Job 、JobJob实例数据
4. Master 节点控制实例数据的负载均衡持久化入数据库



实例调度执行

1. 所有节点（Master/Slave）都负责实例的调度执行；
2. 三种任务类型：周期任务、补数据任务、重跑任务，统一调度方式
3. Job 优先入队列（①），队列容量不足入DB（②）；
4. 当队列容量空余时，异步线程从DB加载数据入队列（③）
5. Job出队列后进行任务提交；

说明：(3)、(4) 解决 Job 数据超大时内存存储问题



实例调度策略

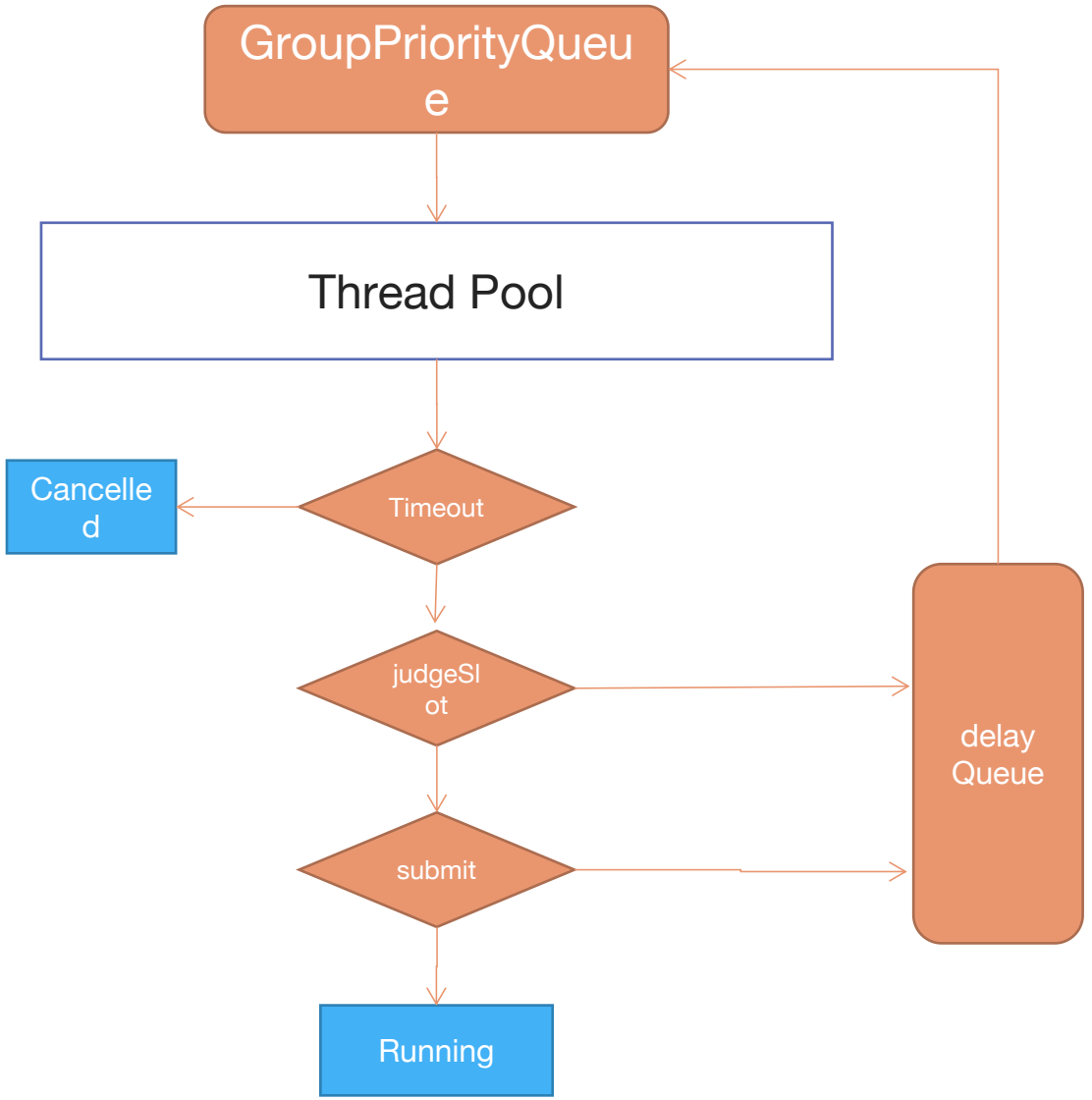
- **轮询（当前支持）**
  1. 从 DB 查询可执行的任务实例（满足执行时间：cyctime < now()）
  2. 判断实例是否存在上游实例，上游实例是否运行结束；
  3. 当前实例存在上游实例且为“失败状态”，设置下游实例为“上游运行失败”
  4. 出错容易排查，稳定性较高；空间复杂度、时间复杂度随 Job 数量增多而正比增长；

PK

- **事件通知**
  1. Job 运行结束后，设置下游实例为可执行状态；
  2. 从 DB 查询可执行的状态任务直接执行；
  3. 出错不易排查，容易上游实例事件通知中断导致下游无法调度；空间复杂度 、时间复杂度O(1)；

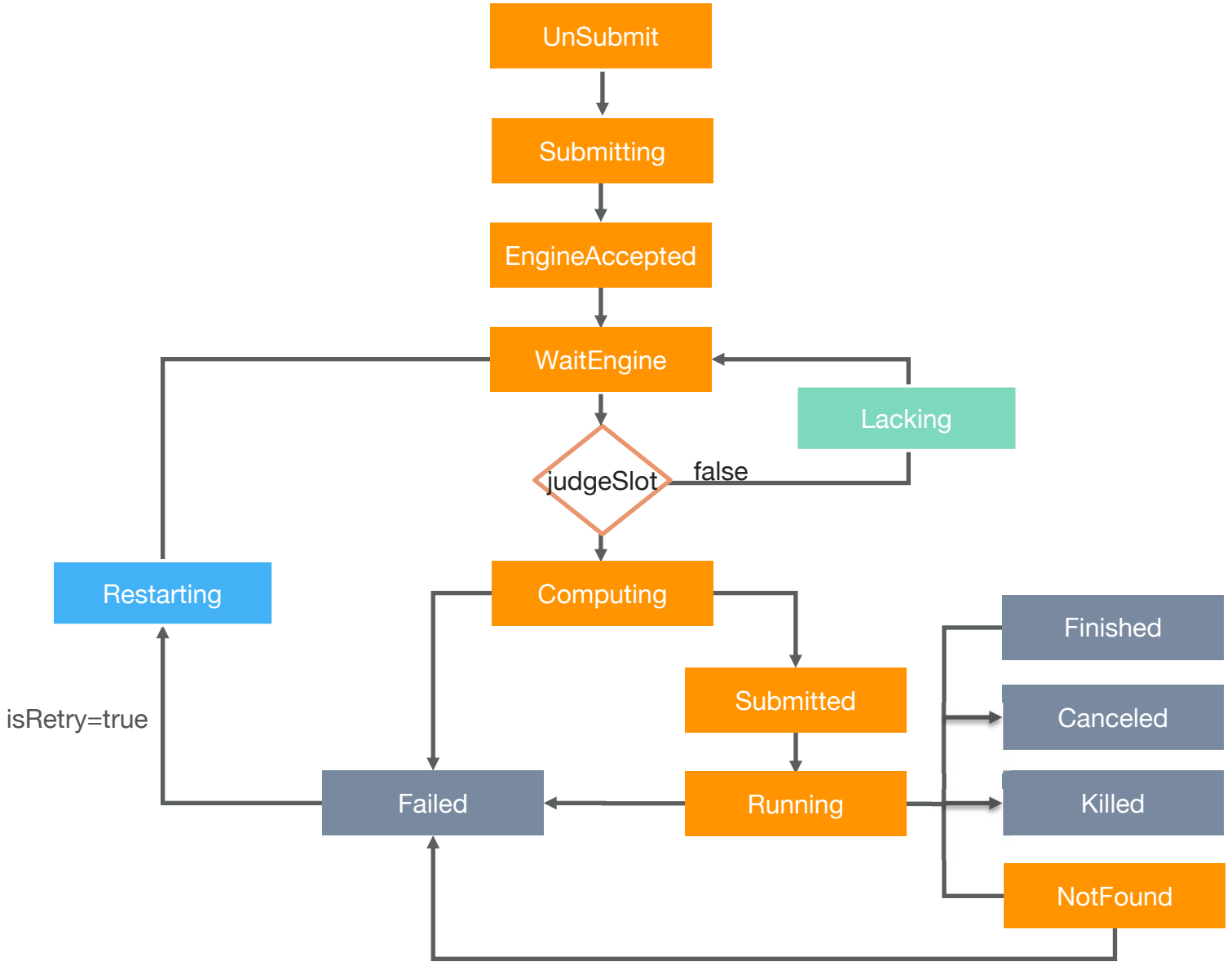
优先级队列&并发提交

1. 内存优先级队列，控制队列内 Job 有序执行;
2. 多线程并发提交（可配置）
3. Job 执行超时判断（可配置）；
4. Job 资源不足/失败重试进入延迟队列（可配置）；避免长时间占用提交权;



任务状态流转

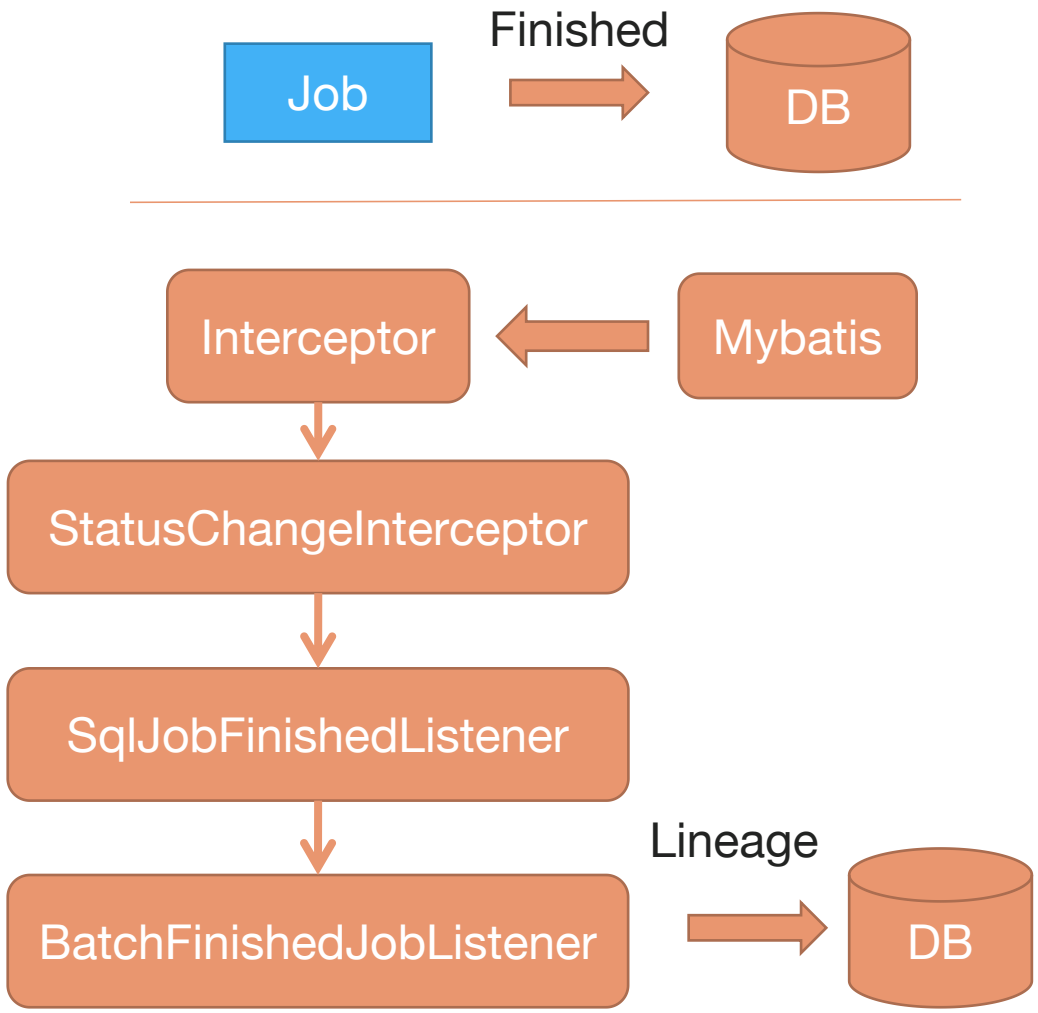
1. WaitEngine：内存队列中的Job、内存容量不足存储在DB中的Job（默认500）
2. Lacking：资源不足暂时等待的Job（默认2min）
3. Restarting：失败重试的Job（默认2min）
4. Finshed、Failed、Canceled、Killed：结束状态



血缘解析

➢ 解析时机&过程

1. 实例运行结束后，更新状态为成功（ Finished ）；
2. StatusChangeInterceptor 监听到（1）状态更新，发送状态变更事件；
3. SqlJobFinishedListener 接收到变更事件，获得 SQL 文本，触发解析SQL解析操作；
4. BatchFinishedJobListener（离线） 调用 SQLParser 插件进行 SQL 解析；
5. SQL 得到的表级、字段级血缘关系持久化到数据库；



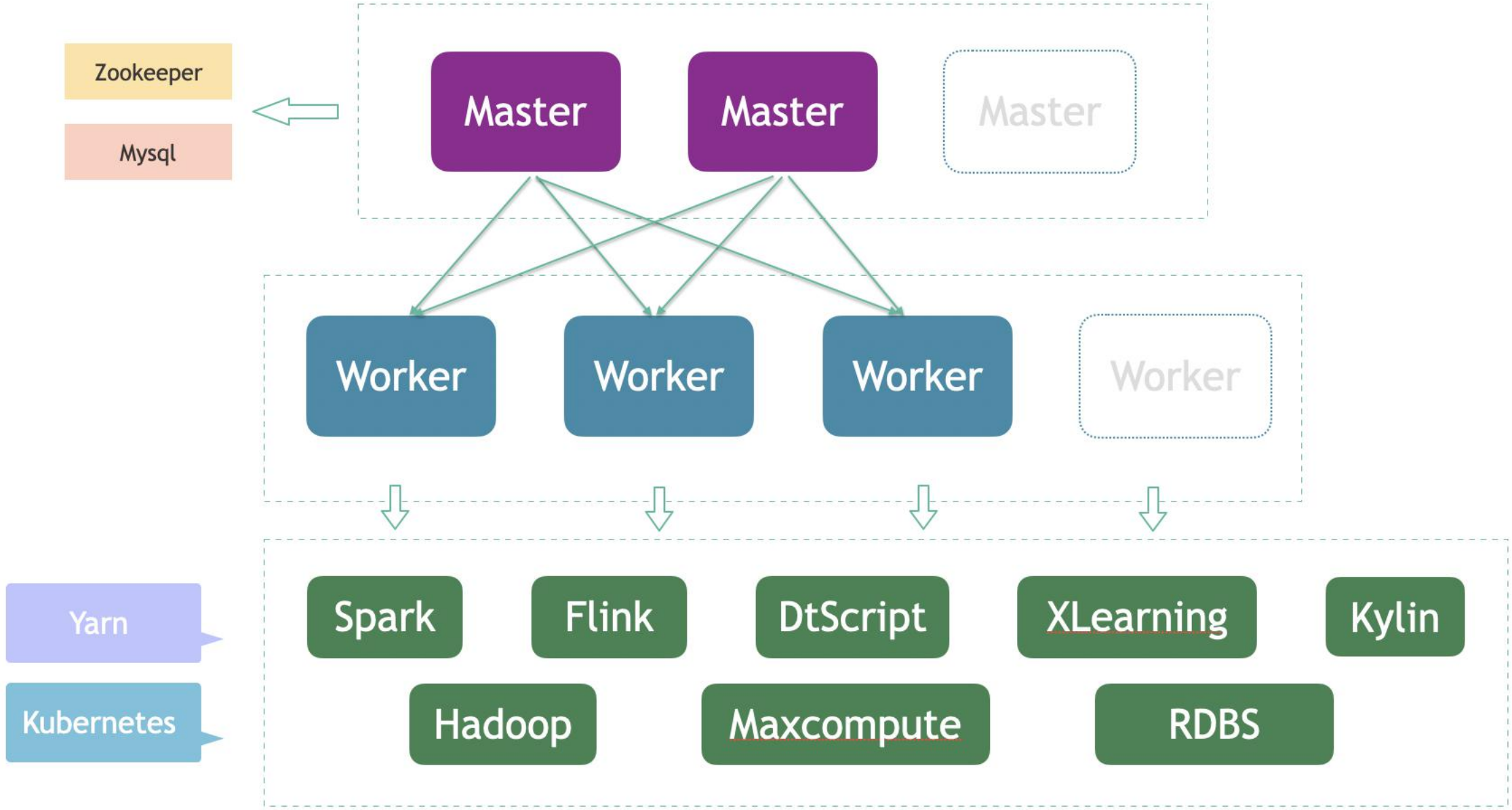
任务类型支持

袋鼠云 DTSTACK

- 插件化加载

1. 一种任务类型对应一个插件，即一个jar包
2. 自定义类加载器（Classloader）破坏双亲委派优先加载（Child-First）插件
3. 每个插件各自实现 IClient 接口
4. SPI：在classpath下的 META-INF/services/ 目录下，创建以接口 IClient 全限定名命名的文件，内容是上一步中实现类的全限定名

DAGScheduleX 架构



袋鼠云
DTSTACK

数据智能
让未来变成现在

400 002 1024

www.dtstack.com

袋鼠云公众号

资料获取