**天枢数据爬虫业务架构**

1. 整体业务架构图：

**消息缓存系统**

**种子任务队列，缓存当天待采集任务**

**Dispatch**

**Dispatch**

分时

策略

调度

入列

**种子库**

1.请求当天任务

2.返回任务种子

**孵**

**化**

**种**

**子**

**更**

**新**

**种**

**子**

**用户/商品/广告**

**统计属性入库**

**天枢数据中心**

Flink Online

Process

**视频/商品/广告内容详情，评论等任务入二级采集任务队列**

**Spider Worker Spider Worker Spider Worker**

**IP代理 IP代理 IP代理**

**爬虫采集引擎**

**原始采集JSON数据缓存队列**

策略调度

系统

**Dispatch**

a

**消息缓存系统**

**消息缓存系统**

1. 爬虫业务架构模块说明
2. **种子库：**

保存着每个短视频/直播平台需要采集的所有任务种子，种子本身就是某个目标KOL的用户id等信息，以及相应的下一轮采集调度日期(next\_scheduling\_date)，前一次采集调度日期(pre\_scheduling\_date)，种子本身的采集频率通过动态更新next\_scheduling\_date控制，通过一定的粉丝数量区间映射策略，来确定该种子的采集调度日期设置。在每天凌晨开始，策略调度系统就会来种子库中取next\_scheduling\_date是当天的所有种子，且种子处于正常状态status=0

1. **策略调度系统**

这里主要有两层策略调度功能：

1. 采集频率动态配置策略：

主要是基于KOL 种子的粉丝数等关键性指标，根据区间映射等策略，来配置该种子的采集频率，动态更新种子库中的next\_scheduling\_date字段

1. 分时段种子投放策略：

调度系统获取到当日所有需要采集的种子集合，根据种子数量以及爬虫引擎实际的采集速度，来决定平均每个小时的种子投放速率，避免在种子任务队列中过于快速的投放种子任务，对爬虫引擎造成过大的采集压力，以及容易带来反爬机制等风险，尽量达到种子的投放速率与下游爬虫引擎的消耗速率基本能够达到动态平衡。

1. **爬虫采集引擎**

采用多台服务器分布式爬虫共享任务队列的机制，从而达到无重复消耗任务种子的

目标，也是整个爬虫架构中难度最大的核心部分，每台爬虫服务器配置相应的外部私密代理IP服务，这一部分涉及到的相关的反风控的相关机制，详细可以在README.MD文档中参考诗恒整理的方案

1. **多级缓存队列**

第一级任务队列topic中，会存放当天策略调度系统从种子库中投放的种子任务，由分布式爬虫集群来无重复消耗；在第二级任务队列topic中，flink平台解析了视频/商品/广告等原始json数据后，提取出视频，商品，广告的内容详情，评论等非结构数据采集任务，放入二级任务队列，由相应的爬虫来提取，采集。

1. **实时流分析平台**

以Flink流处理为核心，搭建数据实时分析平台，负责实时统计数据入天枢数据中台，实时更新种子库调度日期信息以及生成非结构化评论等二级爬取任务。