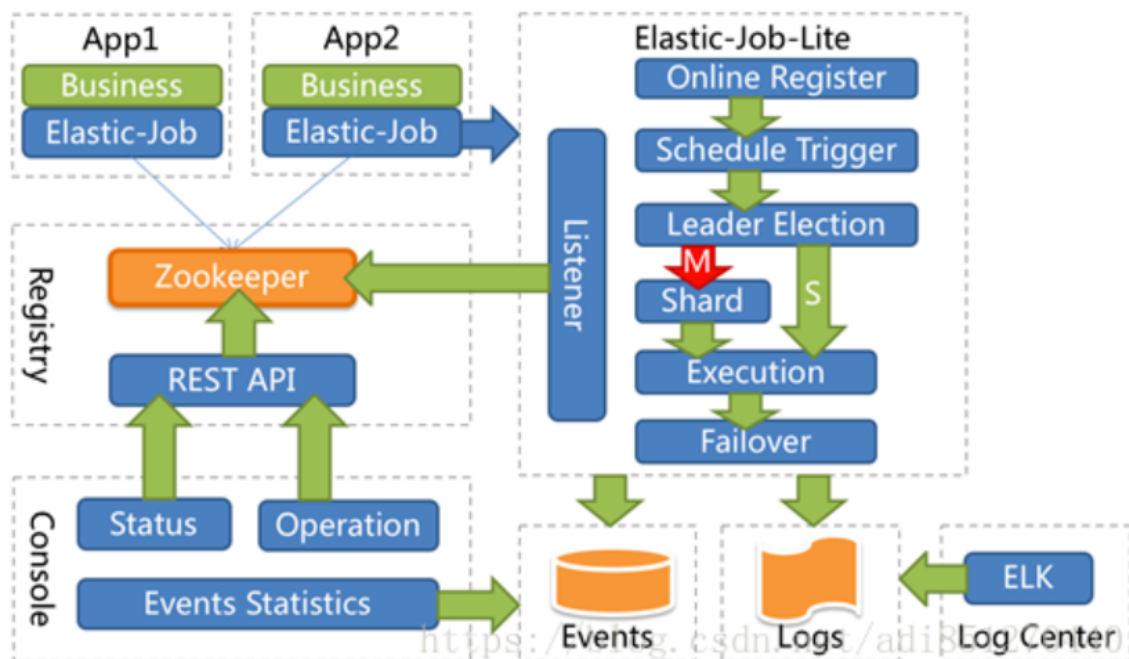


分布式调度的理解

调度—>定时任务，分布式调度—>在分布式集群环境下定时任务这件事，分布式调度有以下两层含义：

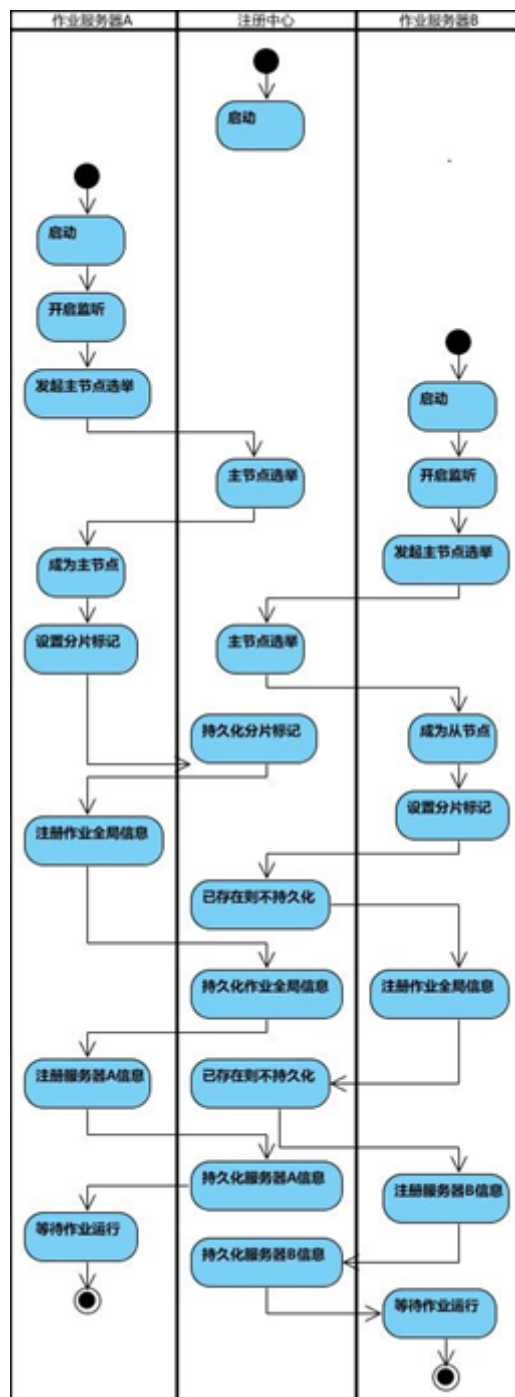
- 1) 运行在分布式集群环境下的调度任务（同一个定时任务程序部署多份，只应该有一个定时任务在执行）
- 2) 分布式调度—>定时任务的分布式—>定时任务的拆分（即为把一个大的作业任务拆分为多个小的作业任务，同时执行）

Elastic-Job是当当网在2015年开源的一个分布式调度解决方案，由两个相互独立的子项目Elastic-Job-Lite和Elastic-Job-Cloud组成。

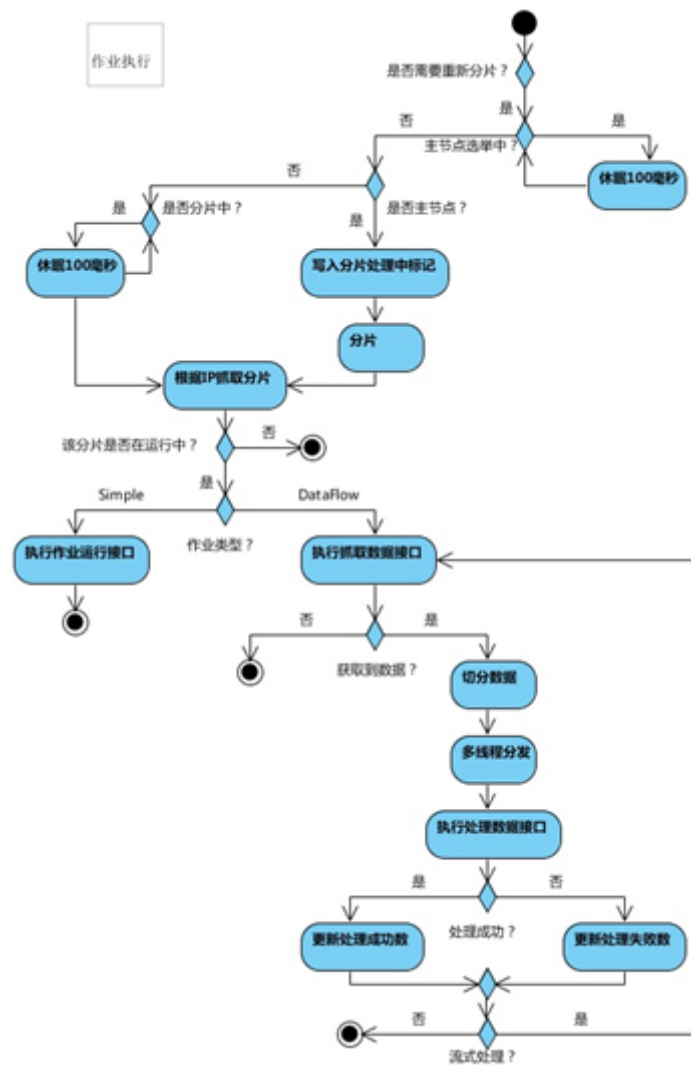


Elastic-Job-Lite框架使用zookeeper作为注册中心，从架构图可以看出，Elastic-Job-Lite框架通过监听器感知zookeeper数据的变化，并做相应的处理；运维平台也仅是通过读取zk数据展现作业状态，或更新zk数据修改全局配置。运维平台和Elastic-Job-Lite没有直接关系，完全解耦合。Elastic-Job-Lite并不直接提供数据处理的功能，框架只会将分片项分配至各个运行中的作业服务器，分片项与真实数据的对应关系需要开发者在应用程序中自行处理

Elastic-Job-Lite源码由两个入口，一个是JobSchedule类的init方法，当启动服务时，程序会进入该方法，进行任务的初始化等操作；另一个是LiteJob类，它实现了Quartz框架的Job接口，当定时任务启动时，会进入该实现类，完成失效转移项执行、重新分片、获取并执行本机任务项、错过任务重触发等操作。



Elastic-Job-Lite初始化的入口是JobSchedule，应用服务器启动时，会调用JobSchedule的init方法，开启作业启动流程。首先添加或更新作业配置信息，并将配置信息持久化到zk上；接着创建quartz调度器，作业的调度执行依赖quartz技术；然后启动所有的监听器，包括leader选举监听、失效转移监听、分片监听等，并发起主节点选举，将leader节点信息set到leader/election/instance节点下；然后将服务器信息、实例信息注册到zk上，并且在leader/sharding下创建necessary节点，作为重新分片的标记；最后由quartz调度器根据cron表达式调度执行。



Elastic-Job-Lite执行器的入口是实现了Job接口的LiteJob类，当任务调度执行时，进入LiteJob类的execute方法。在这里完成一系列的操作，包括获取失效转移分片项，如果没有分配的失效转移项，则判断是否需要重新分片，然后获取分配给自己的分片项，然后判断当前分片项是否正在running，如果否，则执行任务项；如果是，则在sharding/[item]下添加misfire节点，标示该分片项错过执行，等待分片项执行结束后，再触发misfire的分片项执行。