在分布式的场景下由于网络、时钟等原因，可能导致Zookeeper的数据与真实运行的作业产生不一致，这种不一致通过正向的校验无法完全避免。需要另外启动一个线程定时校验注册中心数据与真实作业状态的一致性，即维持Elastic-Job的最终一致性。

在2.0.6之前的版本中，网络不稳定的环境下Elastic-Job有可能有的作业分片并未执行，重启一下就能修复。在2.0.6，版本中Elastic-Job在提供reconcileIntervalMinutes设置修复状态服务执行间隔分钟数，用于修复作业服务器不一致状态，默认每10分钟检测并修复一次。

支持两种配置方式

* Spring方式

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:reg="http://www.dangdang.com/schema/ddframe/reg"

xmlns:job="http://www.dangdang.com/schema/ddframe/job"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.dangdang.com/schema/ddframe/reg

http://www.dangdang.com/schema/ddframe/reg/reg.xsd

http://www.dangdang.com/schema/ddframe/job

http://www.dangdang.com/schema/ddframe/job/job.xsd

">

<!--配置作业注册中心 -->

<reg:zookeeper id="regCenter" server-lists=" yourhost:2181" namespace="dd-job" base-sleep-time-milliseconds="1000" max-sleep-time-milliseconds="3000" max-retries="3" />

<!-- 配置作业-->

<job:simple id="oneOffElasticJob" reconcile-interval-minutes="10" class="xxx.MyElasticJob" registry-center-ref="regCenter" cron="0/10 \* \* \* \* ?" sharding-total-count="3" sharding-item-parameters="0=A,1=B,2=C" />

</beans>

* Java方式

public class JobMain {

public static void main(final String[] args) {

// ...

LiteJobConfiguration.newBuilder(simpleJobConfig).reconcileIntervalMinutes(10).build();

// ...

}

}