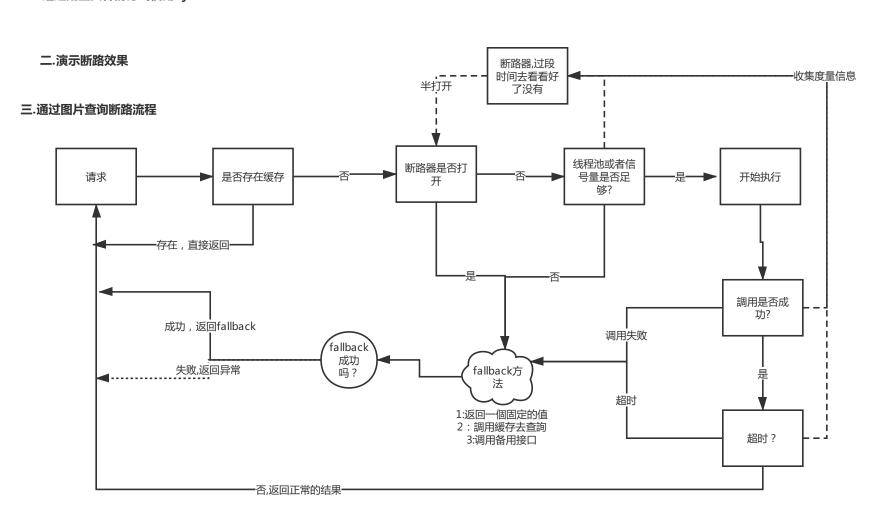
一.通过配置文件的方式使用hystrix



this.metrics = initMetrics(metrics, this.commandGroup, this.threadPoolKey, this.commandKey, this.properties); this.circuitBreaker = initCircuitBreaker(this.properties.circuitBreakerEnabled().get(), circuitBreaker, this.commandGroup, this.commandKey, this.properties, this.metrics); this.threadPool = initThreadPool(threadPool, this.threadPoolKey, threadPoolPropertiesDefaults);

四.注解的方式演示hystrix

五.了解原理

groupkey 我们通常用服务名来表示它是哪个服务 commandKey是服务中的哪个接口 properties去加载了我们自定义的配置,而在这里需要去 理解的hystrix事实上它会去加载Netflix自定义的配置中

度量器 即多长时间去收集一次数据 如我们5秒内访问了多 少次,以及我们的错误率达到了多少,都是由它来进行计 配置的init过 Abstract Command程 #去加载配置 熔断器 就是用来管理熔断的 allowRequest()方法表示是否允许指令执行,该方法在新版本中放弃

事实上,它会在初始化的过程中去

加载配置信息

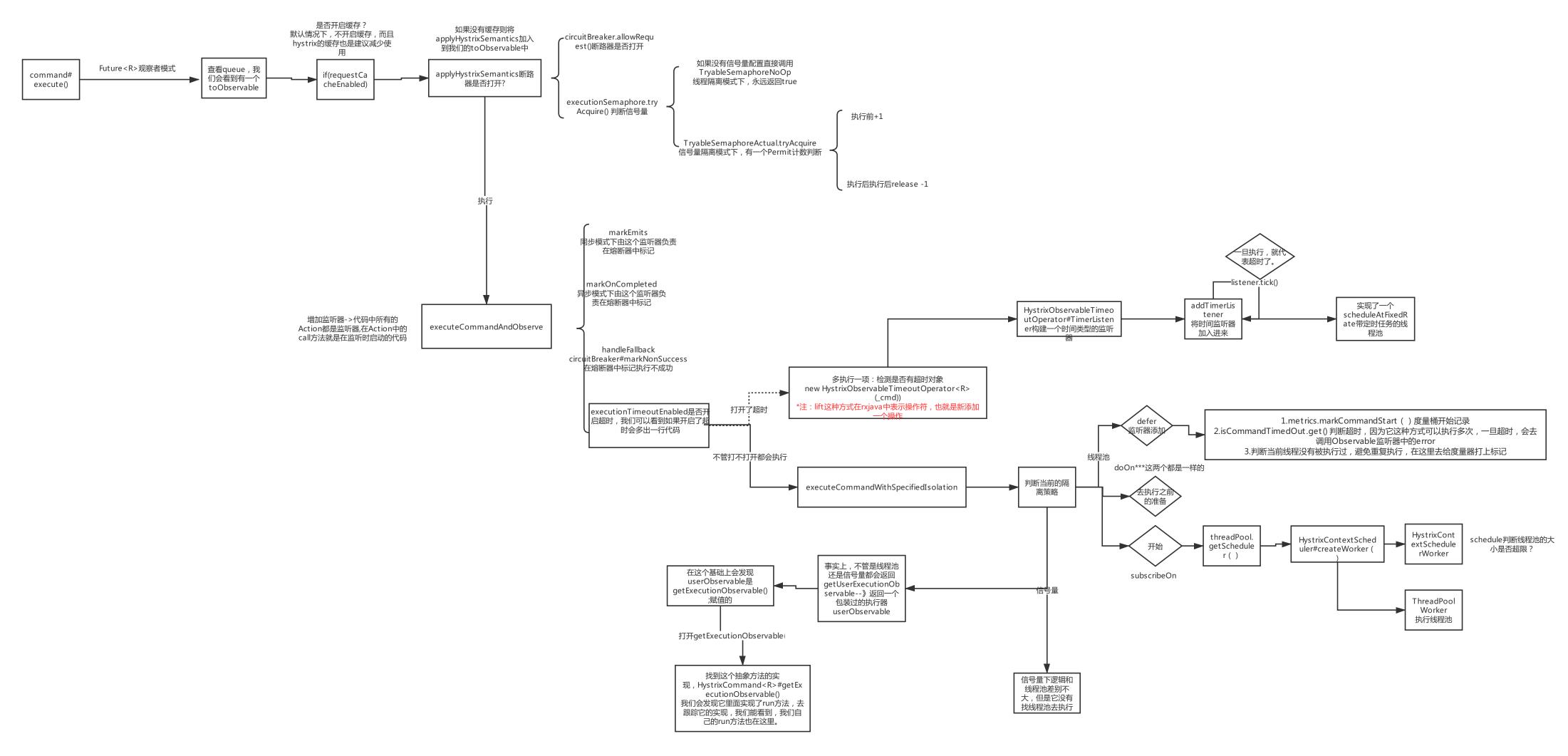
线程池:hystrix通过线程池去隔离的不同服务,所以,每一个groupkey都会有一个线程池。

isOpen()方法表示断路器是否为开启状态

markSuccess()处于半开状态时,如果尝试请求成功,就调用这个方法。用于将断路器关闭

默认是先开启再进入半开启。同时 还可以在配置文件进行强制开启状态。

注意:在执行流程中,实际上使用的是Rxjava的这个东西,这个东西确实很难懂,可能做安卓的接触比较 多,有兴趣的同学可以自己去研究一下。这个在我们这个专题不准备给大家讲懂,因为这个等于是一个新 领域了。所以这里我们只需要去理解它的思路,不要去纠结它的具体写法。 观察者模式下:会在代码的过程中不断的加入监听器,通过监听器来获得我们预期的结果。



六:降级的实现流程 handle Short Circuit Via Fallback () #handle Short Circuit Via Fallback所有的异常都是通过它 Exception shortCircuitException = 开启了熔断之后降级 也就是会返回一个异常的执行器 get Fallback Or Throw Exception new RuntimeException()

七:和springcloud的集成

