**LogNorma2.0说明**

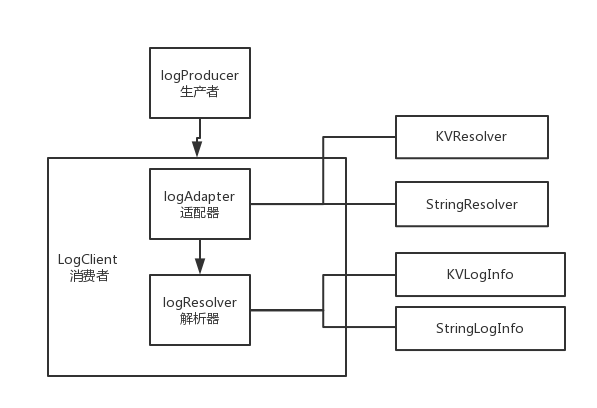
1. **需求概述**

LogNorma设计的初衷是为了解决多种日志格式的归一化问题。如果网络环境中存在多种日志设备，且各设备的日志格式也不尽相同，这时候想要从众多日志中获得有价值的消息，则需要对日志进行归一化处理。也就是将多种不同类型的日志解析成最终一致的日志格式。

1. **设计**

**2.1整体设计思路**

在对日志进行归一化处理的工程中，不可避免的要经过四个过程，采集，解析，处理。结果可以抽象成如下模式：



其中生产者主要负责日志的采集和推送。

适配器负责为当前日志匹配合适的解析器解析日志。适配器封装有多种解析器。

解析器负责使用特定的解析规则来生成最终格式日志。解析器封装了多种日志模板类型。

Client是日志的消费者，在client完成日志的解析以及归一化的日志最终输出。

**2.2开闭性设计**

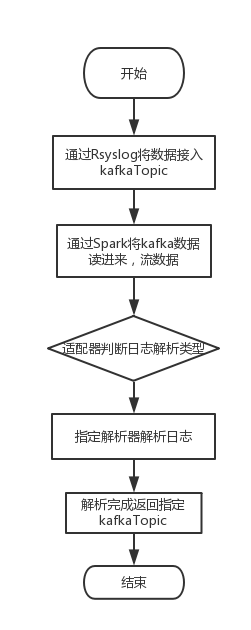
日志归一化的一大难题是程序的可扩展和可复用性。这样当环境中加入了新的设备或者老设备的日志格式发生变更时就可以通过简单的调整来完成这些变更，无须重新开发代码。所以日志归一化的设计必须符合开闭性原则。

通过对logproducer,logadapter,logresolver,logclient四个角色的多层抽象来保证代码的可扩展，和可复用。对外提供统一的接口，有新的设备加入时通过实现特定接口来完成扩展，而不用修改已有的代码。另外，提供了多种方法，可以灵活的对适配器，解析器，日志模板进行修改和配置。

**2.3接口设计**

系统现有接口有7个，分别是LogClient，LogAdapter，LogInfo，LogType，ConfigArgs，KVLogResolver，StringLogResolver。

LogClient：是一个空的超类，实现了序列化接口，针对不同的日志生成模式需要自己来定义处理逻辑。目前系统实现了Rsyslog\_Kafka的客户端处理流程。流程如下：



LogAdapter，对应于LogClient，没有固定的输入和输出，需要结合实际处理逻辑来完成编码。系统提供了对应Rsyslog\_Kafka的适配器接口和实现类。

LogInfo是日志的模板接口，MOULD\_FLAG:日志的特征字段，JOUNAL\_DELIMIT:日志的分隔符，MOULD\_SIZE:日志大小，Log\_type:日志所属的事件类型。MOULD\_MAP：日志原始字段名与归一化日志字段名的映射表。

LogType：日志事件类型。主要包括了事件名，和归一化日志分隔符。

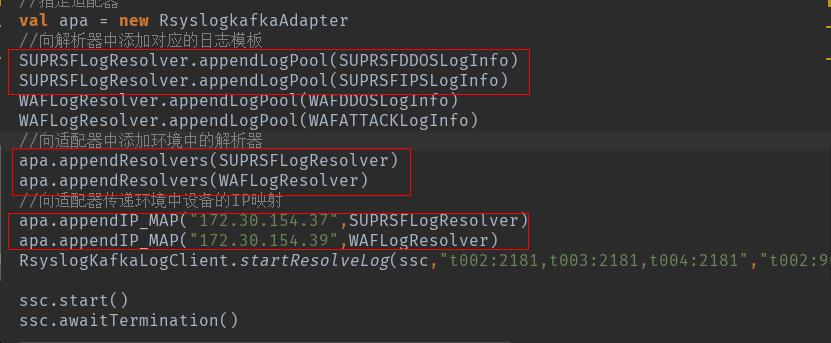
ConfigArgs：是系统的环境变量接口，用来封装可能要用到的一些参数。

KVLogResolver：解析器接口，封装了一个解析方法和对应的日志模板数组。以及添加修改模板数组的方法。

1. **结语**

**3.1使用说明**

日志归一化启动的步骤主要有三步。1.选择要使用的适配器并且按照适配器定义的规则传参，2.向适配器中添加需要的解析器，3.向解析器中添加需要的日志模板。



**3.2添加新的设备**

当有新的设备加入到环境时，需要根据新设备的日志特点来定义解析规则实现LogResolver接口并且实现resolverLog方法。其中在LogUtil中包含了一些通用的日志解析的手段可以使用，也可以根据自己的需求来编写新的解析方法。设备日志按照模板格式编写实现类。最后将新设备的解析器和日志模板告知适配器和解析器即可。另外归一化的日志格式可以再log.properties文件中进行配置修改。

**3.3性能测试**

**集群测试：**

日志数量 :50万条 平均耗时：1s