 编号：

密级：

提醒彩印号码数据处理全流程联调

|  |  |  |
| --- | --- | --- |
|  | 部门: | 电信业务拓展部 |
|  | 项目: | 提醒彩印 |
|  | 编写: | 程依武 |
|  | 审核: |  |

东信北邮信息技术有限公司

2019年12月19日

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 文档名称 | 提醒彩印号码分析联调 | | | 密级 |  |
| 文档编号 |  | 版本号 | V1.00 | 总页数 |  |
| 部门名称 | 彩印产品部 | 编写人 | 程依武 | 编写日期 | 2019.12.19 |
| 项目名称 | 提醒彩印 | 审核人 |  | 审核日期 |  |

|  |  |
| --- | --- |
|  |  |
| 摘要 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本修订记录 | | | | |
| 编号 | 日期 | 版本 | 修订人 | 修订内容 |
| 1 |  | 1.00 |  | 初始版本 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 引言

## 编写目的

## 项目背景

## 术语定义

## 参考资料

# 任务概述

## 被测软件概述

## 测试人员

|  |  |
| --- | --- |
| 测试人员 | 负责测试项 |
| 程依武 | 提醒彩印分区号码联调 |
| 程依武 | 提醒彩印山东号码联调 |

## 测试进度

## 测试条件

# 测试项目

## 提醒彩印

### 文件下载

#### 分区号码下载

|  |
| --- |
| 测试编号：#1 |
| 测试项目：sftp 文件下载 |
| 测试目的：   1. 定时下载分区sftp 文件。并且拆分至flume处理文件夹下：   （包括山东、分区两处sftp文件下载） |
| 预置条件：   1. Linux 环境配置crontab 设置定时； 2. Linux 环境安装lftp 工具下载文件； 3. Linux环境安装sftp； 4. 下载脚本正确配置sftp 环境信息，远程路径、下载文件前缀；(两个脚本分别下载两处sftp ) 5. 下载脚本配置正确本地下载路径，flume 中分区处理（data\_fenqu）、移动12580处理（data\_12580）、山东标记处理(data\_sdbj)、山东诈骗处理(data\_sdzp)监听路径；日志文件路径； |
| 测试流程：  crontab中配置正确定时信息：  crontab -l  30 \*/1 \* \* \* /bin/sh /路径/cys/ftp\_getter\_fenqu.sh  30 \*/1 \* \* \* /bin/sh /路径/cys/ftp\_getter\_shandong.sh |
| 测试结果：   1. 查看日志信息：/路径/cys/log/txcy/txcy\_log.20191218 每天一个（配置路径） 2. 远程下载文件名前缀为配置中syncFuncBill前缀文件，仅下载比本地更新文件（新增或者修改文件生成时间文件）； 3. 下载文件至/路径/cys/txcy/download 下 4. 配置flume单个文件处理行数（默认1000000，不允许为空），若下载文件行数大于此配置，拆分文件为多个，放置flume 分区文件处理路径和移动12580 处理文件下，各一份； 5. 文件不大于此配置分区文件处理路径和移动12580 处理文件下，各一份； 6. 分区源文件中包含（号码$）去除($)，复制后更改文件名前缀为当前时间纳秒时间戳； 7. 山东源文件不做文件字符处理 |
| 建议： 1、远程文件包含多个文件，且行数大于小于配置行数均作详细测试； |

### 接入子系统calllog数据导出

|  |
| --- |
| 测试编号：#2 |
| 测试项目：calllog\_operator数据导出至分析目录 |
| 测试目的：  （1）calllog\_operator数据导出 |
| 预置条件：   1. 此数据为calllog 插入触发至calllog\_operator表中； 2. 导出数据脚本定时配置完成，定时导出数据； 3. 脚本正确配置数据库连接信息； |
| 测试流程：  定时启动脚本； |
| 测试结果：  1、导出表中数据，导出完成后删除表数据；  2、与分区文件下载处理流程一致，做文件数据拆分，然后存储至分区文件（data\_fenqu）、12580文件（data\_12580），flume监听此两个目录下； |
| 建议：1、大数据量测试，最大为一百万； |

### Flume+kafka集群配置

|  |
| --- |
| 测试编号：#3 |
| 预置条件：   1. 配置flume+zookeeper+kakfa集群，flume中分别监听分区、移动12580、山东标记、山东诈骗四个文件夹(data\_fenqu、data\_sdbj、data\_sdzp、data\_12580)，对应kafka中配置4个topic处理flume中数据（txcy-fenqu、txcy-sdbj、txcy-sdzp、txcy-12580）： 2. Java采用四个客户端分别监听上诉的4个kafka topic， |

### 分区号码处理

#### 分区号码处理

|  |
| --- |
| 测试编号：#4 |
| 测试项目：分区号码分析入库 |
| 测试目的：  （1）分区文件分析入库 |
| 预置条件：     1. 分区号码处理消费端配置topic 为txcy-fenqu; 2. Kafka 使用集群配置，即多个kafka broker 添加配置文件中； 3. 配置 黄页、第三方黄页、诈骗、标记服务信息； 4. 配置eureka config zipkin redis feign; 5. 其他redis 等配置正确，启动； |
| 测试流程：   1. 当flume 监听的文件夹下面新增文件后，即刻读取文件内容，传送至kafka 相应topic中； 2. Java 消费端，监听kafka 分区 topic：批量获取数据（配置批量获取条数）； |
| 测试结果：     1. 成功获取kafka集群数据    1. 当集群消费者（即java 程序）个数少于topic 中分区数，会存在一个服务消费多个分区；    2. 当集群消费者（即java程序）个数多余topic中分区数，会存在多余个数消费者无运行数据（即不会消费数据）； 2. 批量获取号码，去除单个号码前+86、0086、86； 3. 根据已经获取到白名单周表号码信息（redis中，查询范围配置），删除当前批次中存在号码； 4. 批量发送至多个查询渠道（目前为腾讯、360、百度、搜狗、阿里、电话邦），可配置，选择其中部分渠道； 5. 返回信息包含号码黄页、标记、非黄页非标记 数据（根据api返回字段获取信息）；   优先级比对、标记次数等处理详细逻辑：   1. 根据第三方接口规范，目前一共6个渠道（腾讯、百度、搜狗、阿里、360、电话邦），统一处理识别数据为黄页、标记、白名单数据（无标记、无黄页）； 2. 黄页： 3. 多个来源的认证黄页信息数据都会入认证黄页库，非认证的黄页信息数据入非认证黄页库； 4. 当某个来源没有某个号码的黄页信息时，则删除原有黄页库中的数据； 5. 当某个来源的某个号码的黄页信息变更时，则更新原有黄页库中的数据； 6. 标记：   第三方标记信息更新原则是：   1. 多个来源的信息，优先入360（可配）的标记数据； 2. 若无360来源信息其他来源的标记次数高的信息作为入标记库数据，更新标记库库里的数据； 3. 若多个来源标记次数相同，则采用轮询策略入标记库库； 4. 多个来源的数据都会入第三方标记原始库。   备注：第三方信息的策略可做成可配置；   1. 白名单：   当以上渠道均为白名单时，做白名单周表数据写入redis，设置有效期，下次号码分析直接排除此号码的查询；   1. 以上为号码处理逻辑，分析完成后，将认证黄页号码、非认证黄页号码、标记号码、其他标记号码，分别调用对应的基础服务，进行入库/删除； 2. 均为白名单数据写入redis 周白名单中，设置有效期（配置）； |
| 建议：   1. 其中配置较多，需要仔细做测试； 2. 需要测试出最恰当的批量处理数量，默认配置为批量处理10000； 3. 测试feign配置 http 超时时间，需要根据数据量测试出最佳时间； 4. 按照文件处理量，（每小时最大文件量为500W），测试出最佳kafka分区数，并配置对应的java 消费端； 5. 目前redis 存储数据为每天一条数据，每条数据有效期为配置时长，有效期时长不能小于查询范围时长； |

### 移动12580号码处理

#### 12580号码处理

|  |
| --- |
| 测试编号：#5 |
| 测试项目：12580号码分析 |
| 测试目的：  （1）12580分析分号码数据 |
| 预置条件：     1. 12580号码处理消费端配置topic 为txcy-12580; 2. Kafka 使用集群配置，即多个kafka broker 添加配置文件中； 3. 配置黄页、标记库服务信息； 4. 配置eureka config zipkin redis feign redis |
| 测试流程：   1. 当flume 监听的文件夹下面新增文件后，即刻读取文件内容，传送至kafka 相应topic中； 2. Java 消费端，监听kafka 12580 topic：批量获取数据（配置批量获取条数）； |
| 1. 成功获取kafka集群数据    1. 当集群消费者（即java 程序）个数少于topic 中分区数，会存在一个服务消费多个分区；    2. 当集群消费者（即java程序）个数多余topic中分区数，会存在多余个数消费者无运行数据（即不会消费数据）； 2. 批量获取号码，去除单个号码前+86、0086、86； 3. 根据已经获取到白名单月表号码信息（redis中，查询范围配置），删除当前批次中存在号码； 4. 批量发送号码至12580 基础服务； 5. 返回信息包含认证黄页号码、白名单号码；； 6. 将认证黄页入库至黄页表中（黄页基础服务）； 7. 白名单数据写入redis，月白名单中，设置有效期（配置）； |
| 建议：   1. 与分区号码需要测试结果一致； |

### 山东标记号码处理

#### 山东标记号码处理

|  |
| --- |
| 测试编号：#6 |
| 测试项目：山东标记号码分析 |
| 测试目的：  （1）山东标记号码入库 |
| 预置条件：     1. 山东号码处理消费端配置topic 为txcy-sdbj 2. Kafka 使用集群配置，即多个kafka broker 添加配置文件中； 3. 配置标记服务相关信息； 4. 配置eureka config zipkin redis feign; |
| 测试流程：   1. 当flume 监听的文件夹下面新增文件后，即刻读取文件内容，传送至kafka 相应topic中； 2. Java 消费端，监听kafka topic(txcy\_sdbj)：批量获取数据（配置批量获取条数）； |
| 1. 成功获取kafka集群数据    1. 当集群消费者（即java 程序）个数少于topic 中分区数，会存在一个服务消费多个分区；    2. 当集群消费者（即java程序）个数多余topic中分区数，会存在多余个数消费者无运行数据（即不会消费数据）； 2. 批量获取号码，每行号码按照 \t（制表符）拆分； 3. 按照第二个字段0为新增，1为删除，添加/删除标记数据、第三方标记源数据； 4. 直接调用标记、第三方标记库基础服务批量入库数据； |
| 建议：   1. 需要测试最佳批量获取kafka 数据量； 2. 测试最佳feign http调用超时时间； |

### 山东诈骗号码处理

#### 山东诈骗号码处理

|  |
| --- |
| 测试编号：#6 |
| 测试项目：山东标记号码分析 |
| 测试目的：    （1）山东标记号码入库 |
| 预置条件：   1. 山东号码处理消费端配置topic 为txcy-sdzp; 2. Kafka 使用集群配置，即多个kafka broker 添加配置文件中； 3. 配置诈骗基础服务相关信息； 4. 配置eureka config zipkin redis feign; |
| 测试流程：   1. 当flume 监听的文件夹下面新增文件后，即刻读取文件内容，传送至kafka 相应topic中； 2. Java 消费端，监听kafka topic(txcy\_sdbj)：批量获取数据（配置批量获取条数）； |
| 1. 成功获取kafka集群数据    1. 当集群消费者（即java 程序）个数少于topic 中分区数，会存在一个服务消费多个分区；    2. 当集群消费者（即java程序）个数多余topic中分区数，会存在多余个数消费者无运行数据（即不会消费数据）； 2. 批量获取号码，每行号码按照 \t（制表符）拆分； 3. 按照第二个字段0为新增，1为删除，添加/删除诈骗库数据； 4. 直接调用诈骗库基础服务批量入库数据； |
| 建议：   1. 需要测试最佳批量获取kafka 数据量； 2. 测试最佳feign http调用超时时间； |

# 测试总结

## 总体评价

## 最终结果

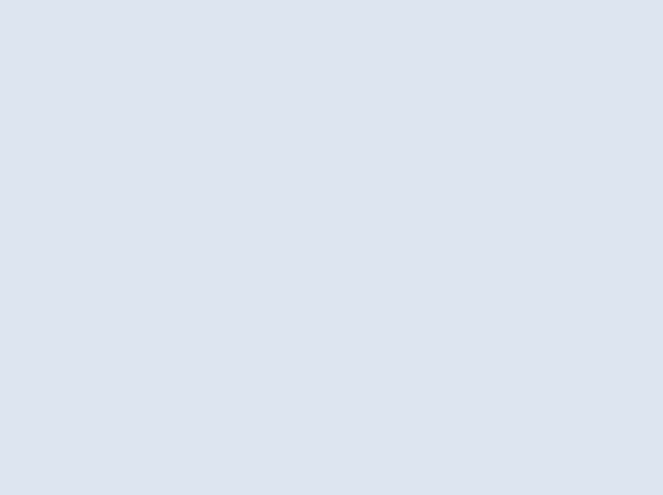
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **测试项目** | **用例总数** | **可测项目数** | **测试结果** | | | | **NRDY** |
| **OK** | **POK** | **NOK** | **通过率** |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **总体统计** |  |  |  |  |  |  |  |

## 问题总结

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **问题简要描述** | **解决情况** | **备注** |
|  |  |  |  |

## Bug总结

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Bug编号** | **对应的测试项目** | **问题简要描述** | **严重程度** | **Bug类型** | **发现人** | **发现时间** | **状态** | **责任人** | **备注** |
|  |  |  |  |  |  |  |  |  |  |

模板修订记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 日期 | 版本 | 修订人 | 修订内容 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |