## 模块D：数据采集与实时计算（20分）

**环境说明：**

|  |
| --- |
| **服务端登录地址详见各模块服务端说明。**  **补充说明：**各主机可通过Asbru工具或SSH客户端进行SSH访问；  请先检查ZooKeeper、Kafka、Redis端口是否已启动，若未启动则各启动命令如下：  ZK启动（netstat -ntlp查看2181端口是否打开）  /usr/zk/zookeeper-3.4.6/bin/zkServer.sh start  Redis启动（netstat -ntlp查看6379端口是否打开）  /usr/redis/bin/redis-server /usr/redis/bin/redis.conf  Kafka启动（netstat -ntlp查看9092端口是否打开）  /opt/kafka/kafka\_2.11-2.0.0/bin/kafka-server-start.sh -daemon（空格连接下一行）/opt/kafka/kafka\_2.11-2.0.0/config/server.properties  Flink任务在Yarn上用per job模式（即Job分离模式，不采用Session模式），方便Yarn回收资源。 |

### 任务一：实时数据采集

1. 在Master节点使用Flume采集/data\_log目录下实时日志文件中的数据，将数据存入到Kafka的Topic中（Topic名称分别为ChangeRecord、ProduceRecord和EnvironmentData，分区数为4），将Flume采集ChangeRecord主题的配置截图粘贴至对应报告中；
2. 编写新的Flume配置文件，将数据备份到HDFS目录/user/test/flumebackup下，要求所有主题的数据使用同一个Flume配置文件完成，将Flume的配置截图粘贴至对应报告中。

### 任务二：使用Flink处理Kafka中的数据

编写Scala工程代码，使用Flink消费Kafka中的数据并进行相应的数据统计计算。

1. 使用Flink消费Kafka中ProduceRecord主题的数据，统计在已经检验的产品中，各设备每5分钟生产产品总数，将结果存入Redis中，key值为“totalproduce”，value值为“设备id，最近5分钟生产总数”。使用redis cli以HGETALL key方式获取totalproduce值，将结果截图粘贴至对应报告中，需两次截图，第一次截图和第二次截图间隔5分钟以上，第一次截图放前面，第二次截图放后面；

注：ProduceRecord主题，生产一个产品产生一条数据；

change\_handle\_state字段为1代表已经检验，0代表未检验；

时间语义使用Processing Time。

1. 使用Flink消费Kafka中ChangeRecord主题的数据，当某设备30秒状态连续为“预警”，输出预警信息。当前预警信息输出后，最近30秒不再重复预警（即如果连续1分钟状态都为“预警”只输出两次预警信息），将结果存入Redis中，key值为“warning30sMachine”，value值为“设备id，预警信息”。使用redis cli以HGETALL key方式获取warning30sMachine值，将结果截图粘贴至对应报告中，需两次截图，第一次截图和第二次截图间隔1分钟以上，第一次截图放前面，第二次截图放后面；

注：时间使用change\_start\_time字段，忽略数据中的change\_end\_time不参与任何计算。忽略数据迟到问题。

Redis的value示例：115,2022-01-01 09:53:10:设备115 连续30秒为预警状态请尽快处理！

(2022-01-01 09:53:10 为change\_start\_time字段值，中文内容及格式必须为示例所示内容。)

1. 使用Flink消费Kafka中ChangeRecord主题的数据，统计每3分钟各设备状态为“预警”且未处理的数据总数，将结果存入MySQL数据库shtd\_industry的threemin\_warning\_state\_agg表中（追加写入，表结构如下）。请将任务启动命令截图，启动且数据进入后按照设备id降序排序查询threemin\_warning\_state\_agg表进行截图，第一次截图后等待3分钟再次查询并截图,将结果截图粘贴至对应报告中。

threemin\_warning\_state\_agg表：

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 中文含义 |
| change\_machine\_id | int | 设备id |
| totalwarning | int | 未被处理预警的数据总数 |
| window\_end\_time | varchar | 窗口结束时间（yyyy-MM-dd HH:mm:ss） |

注：时间语义使用Processing Time。

1. 使用Flink消费Kafka中ChangeRecord主题的数据，实时统计每个设备从其他状态转变为“运行”状态的总次数，将结果存入MySQL数据库shtd\_industry的change\_state\_other\_to\_run\_agg表中（表结构如下）。请将任务启动命令截图，启动1分钟后根据change\_machine\_id降序查询change\_state\_other\_to\_run\_agg表并截图，启动两分钟后根据change\_machine\_id降序查询change\_state\_other\_to\_run\_agg表并再次截图；

注：时间语义使用Processing Time。

change\_state\_other\_to\_run\_agg表：

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 中文含义 |
| change\_machine\_id | int | 设备id |
| last\_machine\_state | varchar | 上一状态。即触发本次统计的最近一次非运行状态 |
| total\_change\_torun | int | 从其他状态转为运行的总次数 |
| in\_time | varchar | flink计算完成时间（yyyy-MM-dd HH:mm:ss） |

1. 使用Flink消费Kafka中ChangeRecord主题的数据，每隔1分钟输出最近3分钟的预警次数最多的设备，将结果存入Redis中，key值为“warning\_last3min\_everymin\_out”，value值为“窗口结束时间，设备id”（窗口结束时间格式：yyyy-MM-dd HH:mm:ss）。使用redis cli以HGETALL key方式获取warning\_last3min\_everymin\_out值，将结果截图粘贴至对应报告中，需两次截图，第一次截图和第二次截图间隔1分钟以上，第一次截图放前面，第二次截图放后面。

注：时间语义使用Processing Time。

1. 使用Flink消费Kafka中EnvironmentData主题的数据,监控各环境检测设备数据，当温度（Temperature字段）持续3分钟高于38度时记录为预警数据。将结果存入Redis中，key值为“env\_temperature\_monitor”，value值为“设备id-预警信息生成时间，预警信息”（预警信息生成时间格式：yyyy-MM-dd HH:mm:ss）。使用redis cli以HGETALL key方式获取env\_temperature\_monitor值，将结果截图粘贴至对应报告中，需要两次截图间隔3分钟以上，第一次截图放前面，第二次截图放后面；

注：时间语义使用Processing Time。

value示例：114-2022-01-01 14:12:19，设备114连续三分钟温度高于38度请及时处理！

中文内容及格式必须为示例所示内容。

同一设备3分钟只预警一次。