**2023年全国职业院校技能大赛**

**赛题第10套**

**赛项名称： 大数据应用开发**

**英文名称：** Big Data Application Development

**赛项组别： 高等职业教育组**

**赛项编号： GZ033**

**背景描述**

工业互联网是工业全要素、全产业链、全价值链的全面连接，是人、机、物、工厂互联互通的新型工业生产制造服务体系，是互联网从消费领域向生产领域、从虚拟经济向实体经济拓展的核心载体，是建设现代化经济体系、实现高质量发展和塑造全球产业竞争力的关键支撑，工业大数据则是工业互联网实现工业要素互联之后的核心价值创造者｡随着大数据行业的发展,工业数据收集呈现时间维度不断延长、数据范围不断扩大、数据粒度不断细化的趋势｡以上三个维度的变化使得企业所积累的数据量以加速度的方式在增加,最终构成了工业大数据的集合｡

为完成工业大数据分析工作，你所在的小组将应用大数据技术，以Scala作为整个项目的基础开发语言，基于大数据平台综合利用

Hudi、Spark、Flink、Vue.js等技术，对数据进行处理、分析及可视化呈现，你们作为该小组的技术人员，请按照下面任务完成本次工作。

## 任务A：大数据平台搭建（容器环境）（15分）

**环境说明：**

|  |
| --- |
| **服务端登录地址详见各任务服务端说明。**  **补充说明：**宿主机及各容器节点可通过Asbru工具或SSH客户端进行SSH访问。 |

### 子任务一：Hadoop HA安装配置

本任务需要使用root用户完成相关配置，安装Hadoop需要配置前置环境。命令中要求使用绝对路径，具体要求如下:

1. 从宿主机/opt目录下将文件hadoop-3.1.3.tar.gz、jdk-8u212-linux-x64.tar.gz，apache-zookeeper-3.5.7-bin.tar.gz复制到容器Master中的/opt/software路径中（若路径不存在，则需新建），分别将Master节点Hadoop、ZooKeeper、JDK安装包解压到/opt/module路径中(若路径不存在，则需新建)，其中将JDK、Hadoop解压命令复制并粘贴至客户端桌面【Release\任务A提交结果.docx】中对应的任务序号下；
2. 请完成host相关配置，将三个节点分别命名为master、slave1、slave2并做免密登录，修改容器中/etc/profile文件，设置JDK环境变量并使其生效，分发jdk至slave1、slave2中，均配置完毕后在Master节点分别执行“java -version”和“javac”命令，将命令行执行结果分别截图并粘贴至客户端桌面【Release\任务A提交结果.docx】中对应的任务序号下；
3. 配置好zookeeper，其中zookeeper使用集群模式，分别在master、slave1、slave2作为其集群的节点，使用zkServer.sh status获取zookeeper服务端状态，将命令和结果截图粘贴至客户端桌面【Release\任务A提交结果.docx】中对应的任务序号下（注：只截取三个节点中zookeeper server角色模式为leader的节点）；
4. 配置好Hadoop HA，请将dfs.ha.namenodes.hadoopcluster设置为nn1、nn2，同时yarn.resourcemanager.ha.rm-ids设置为rm1、rm2，并在Master启动nn1与rm1，在slave1启动nn2与rm2，将master、slave1、slave2均作为datanode，分发hadoop至slave1、slave2中，启动yarn与hdfs的HA集群（Hadoop HA集群），并在Master节点上使用命令分别查看服务nn2与rm2进程状态，并将查看命令及结果截图粘贴至客户端桌面【Release\任务A提交结果.docx】中对应的任务序号下；
5. Hadoop HA配置并启动完毕后，使用jps在slave1节点查看服务进程，将查看命令及结果截图粘贴至客户端桌面【Release\任务A提交结果.docx】中对应的任务序号下。

### 子任务二：Flume安装配置

本任务需要使用root用户完成相关配置，已安装Hadoop及需要配置前置环境，具体要求如下：

1. 从宿主机/opt目录下将文件apache-flume-1.7.0-bin.tar.gz复制到容器master中的/opt/software路径中（若路径不存在，则需新建），将Master节点Flume安装包解压到/opt/module目录下，将解压命令复制并粘贴至客户端桌面【Release\任务A提交结果.docx】中对应的任务序号下；
2. 完善相关配置，配置Flume环境变量，并使环境变量生效，执行命令flume-ng version并将命令与结果截图粘贴至客户端桌面【Release\任务A提交结果.docx】中对应的任务序号下；
3. 启动Flume传输Hadoop日志（namenode或datanode日志），查看HDFS中/tmp/flume目录下生成的内容，将查看命令及结果（至少5条结果）截图并粘贴至客户端桌面【Release\任务A提交结果.docx】中对应的任务序号下。

### 子任务三：ClickHouse单节点安装配置

本任务需要使用root用户完成相关配置，具体要求如下：

1. 从宿主机/opt目录下将clickhouse开头的相关文件复制到容器Master中的/opt/module/clickhouse路径中（若路径不存在，则需新建），将全部解压命令复制并粘贴至客户端桌面【Release\任务A提交结果.docx】中对应的任务序号下；
2. 执行启动各个相关脚本，将全部启动命令复制并将执行结果（截取结果最后倒数15行即可）截图粘贴至客户端桌面【Release\任务A提交结果.docx】中对应的任务序号下；
3. 设置远程访问并移除默认监听文件（listen.xml），同时由于9000端口被Hadoop占用，需要将clickhouse的端口更改为9001，并启动clickhouse，启动后查看clickhouse运行状态，并将启动命令复制、查看运行状态命令复制并将执行结果截图粘贴至客户端桌面【Release\任务A提交结果.docx】中对应的任务序号下。

## 任务B：离线数据处理（25分）

**环境说明：**

|  |
| --- |
| **服务端登录地址详见各任务服务端说明。**  **补充说明：**各节点可通过Asbru工具或SSH客户端进行SSH访问；  主节点MySQL数据库用户名/密码：root/123456（已配置远程连接）；  Spark任务在Yarn上用Client运行，方便观察日志。 |

### 子任务一：数据抽取

编写Scala代码，使用Spark将MySQL库中表EnvironmentData，ChangeRecord，BaseMachine，MachineData,ProduceRecord全量抽取到**Hudi**的hudi\_gy\_ods库（路径为/user/hive/warehouse/hudi\_gy\_ods.db）中对应表environmentdata，changerecord，basemachine， machinedata， producerecord中。

1. 抽取MySQL的shtd\_industry库中EnvironmentData表的全量数据进入Hudi的hudi\_gy\_ods库中表environmentdata，字段排序、类型不变，同时添加静态分区，分区字段为etldate，类型为String，且值为当前比赛日的前一天日期（分区字段格式为yyyyMMdd）。PRECOMBINE\_FIELD使用InPutTime，EnvoId作为主键。使用spark-sql的cli执行show partitions ods.environmentdata命令，将结果截图粘贴至客户端桌面【Release\任务B提交结果.docx】中对应的任务序号下；
2. 抽取MySQL的shtd\_industry库中ChangeRecord表的全量数据进入Hudi的hudi\_gy\_ods库中表changerecord，字段排序、类型不变，分区字段为etldate，类型为String，且值为当前比赛日的前一天日期（分区字段格式为yyyyMMdd）。PRECOMBINE\_FIELD使用ChangeEndTime，ChangeID和ChangeMachineID作为联合主键。使用spark-sql的cli执行select count（\*） from ods.changerecord命令，将cli的执行结果分别截图粘贴至客户端桌面【Release\任务B提交结果.docx】中对应的任务序号下；
3. 抽取MySQL的shtd\_industry库中BaseMachine表的全量数据进入Hudi的hudi\_gy\_ods库中表basemachine，字段排序、类型不变，分区字段为etldate，类型为String，且值为当前比赛日的前一天日期（分区字段格式为yyyyMMdd）。PRECOMBINE\_FIELD使用MachineAddDate，BaseMachineID为主键。使用spark-sql的cli执行show partitions ods.basemachine命令，将spark-sql的cli的执行结果分别截图粘贴至客户端桌面【Release\任务B提交结果.docx】中对应的任务序号下；
4. 抽取MySQL的shtd\_industry库中ProduceRecord表的全量数据进入Hudi的hudi\_gy\_ods库中表producerecord，剔除ProducePrgCode字段，其余字段排序、类型不变，分区字段为etldate，类型为String，且值为当前比赛日的前一天日期（分区字段格式为yyyyMMdd）。PRECOMBINE\_FIELD使用ProduceCodeEndTime，ProduceRecordID和ProduceMachineID为联合主键。使用spark-sql的cli执行show partitions ods.producerecord命令，将spark-sql的cli的执行结果分别截图粘贴至客户端桌面【Release\任务B提交结果.docx】中对应的任务序号下；
5. 抽取MySQL的shtd\_industry库中MachineData表的全量数据进入Hudi的hudi\_gy\_ods库中表machinedata，字段排序、类型不变，分区字段为etldate，类型为String，且值为当前比赛日的前一天日期（分区字段格式为yyyyMMdd）。PRECOMBINE\_FIELD使用MachineRecordDate，MachineRecordID为主键。使用spark-sql的cli执行show partitions ods.machinedata命令，将cli的执行结果分别截图粘贴至客户端桌面【Release\任务B提交结果.docx】中对应的任务序号下。

### 子任务二：数据清洗

编写Scala代码，使用Spark将ods库中相应表数据全量抽取到**Hudi**的hudi\_gy\_dwd库（路径为/user/hive/warehouse/hudi\_gy\_dwd.db）中对应表中。表中有涉及到timestamp类型的，均要求按照yyyy-MM-dd HH:mm:ss，不记录毫秒数，若原数据中只有年月日，则在时分秒的位置添加00:00:00，添加之后使其符合yyyy-MM-dd HH:mm:ss。

1. 抽取hudi\_gy\_ods库中environmentdata的全量数据进入Hudi的hudi\_gy\_dwd库中表fact\_environment\_data，分区字段为etldate且值与ods库的相对应表该值相等，并添加dwd\_insert\_user、dwd\_insert\_time、dwd\_modify\_user、dwd\_modify\_time四列,其中dwd\_insert\_user、dwd\_modify\_user均填写“user1”，dwd\_insert\_time、dwd\_modify\_time均填写当前操作时间，并进行数据类型转换。dwd\_modify\_time作为preCombineField，EnvoId作为primaryKey。使用spark-sql的cli按照envoid降序排序，查询前5条数据，将结果截图粘贴至客户端桌面【Release\任务B提交结果.docx】中对应的任务序号下；
2. 抽取hudi\_gy\_ods库中changerecord的全量数据进入Hudi的hudi\_gy\_dwd库中表fact\_change\_record，分区字段为etldate且值与hudi\_gy\_ods库的相对应表该值相等，并添加dwd\_insert\_user、dwd\_insert\_time、dwd\_modify\_user、dwd\_modify\_time四列,其中dwd\_insert\_user、dwd\_modify\_user均填写“user1”，dwd\_insert\_time、dwd\_modify\_time均填写当前操作时间，并进行数据类型转换。dwd\_modify\_time作为preCombineField，change\_id和change\_machine\_id作为联合primaryKey。使用spark-sql的cli按照change\_machine\_id、change\_id均为降序排序，查询前1条数据，将结果截图粘贴至客户端桌面【Release\任务B提交结果.docx】中对应的任务序号下；
3. 抽取hudi\_gy\_ods库中basemachine的全量数据进入Hudi的hudi\_gy\_dwd库中表dim\_machine。分区字段为etldate且值与hudi\_gy\_ods库的相对应表该值相等，并添加dwd\_insert\_user、dwd\_insert\_time、dwd\_modify\_user、dwd\_modify\_time四列,其中dwd\_insert\_user、dwd\_modify\_user均填写“user1”，dwd\_insert\_time、dwd\_modify\_time均填写当前操作时间，并进行数据类型转换。dwd\_modify\_time作为preCombineField，Base\_machine\_id作为primaryKey。使用spark-sql的cli按照Base\_machine\_id升序排序，查询dim\_machine前2条数据，将结果截图粘贴至客户端桌面【Release\任务B提交结果.docx】中对应的任务序号下；
4. 抽取hudi\_gy\_ods库中producerecord的全量数据进入Hudi的hudi\_gy\_dwd库中表fact\_produce\_record,分区字段为etldate且值与hudi\_gy\_ods库的相对应表该值相等，并添加dwd\_insert\_user、dwd\_insert\_time、dwd\_modify\_user、dwd\_modify\_time四列,其中dwd\_insert\_user、dwd\_modify\_user均填写“user1”，dwd\_insert\_time、dwd\_modify\_time均填写当前操作时间，并进行数据类型转换。dwd\_modify\_time作为preCombineField，produce\_record\_id和produce\_machine\_id作为联合primaryKey。使用spark-sql的cli按照produce\_machine\_id、produce\_record\_id均为升序排序，查询fact\_produce\_record前1条数据，将结果截图粘贴至客户端桌面【Release\任务B提交结果.docx】中对应的任务序号下；
5. 抽取hudi\_gy\_ods库中machinedata的全量数据进入Hudi的hudi\_gy\_dwd库中表fact\_machine\_data。分区字段为etldate且值与hudi\_gy\_ods库的相对应表该值相等，并添加dwd\_insert\_user、dwd\_insert\_time、dwd\_modify\_user、dwd\_modify\_time四列,其中dwd\_insert\_user、dwd\_modify\_user均填写“user1”，dwd\_insert\_time、dwd\_modify\_time均填写当前操作时间，并进行数据类型转换。dwd\_modify\_time作为preCombineField，machine\_record\_id作为primaryKey。使用spark-sql的cli按照machine\_id、machine\_record\_id均为降序排序，查询前1条数据，将结果截图粘贴至客户端桌面【Release\任务B提交结果.docx】中对应的任务序号下。

### 子任务三：指标计算

1. 本任务基于以下2、3、4小题完成，使用DolphinScheduler完成第2、3、4题任务代码的调度。工作流要求，使用shell输出“开始”作为工作流的第一个job（job1），2、3、4题任务为串行任务且它们依赖job1的完成（命名为job2、job3、job4），job2、job3、job4完成之后使用shell输出“结束”作为工作流的最后一个job（endjob），endjob依赖job2、job3、job4，并将最终任务调度完成后的工作流截图，将截图粘贴至客户端桌面【Release\任务B提交结果.docx】中对应的任务序号下；
2. 编写scala代码，使用Spark根据hudi\_gy\_dwd层的fact\_machine\_data表统计出每日每台设备，状态为“运行”的时长（若运行无结束时间，则需根据时间判断这个设备的运行状态的下一个状态是哪条数据，将下一个状态的数据的时间置为这个设备运行状态的结束时间,如果设备数据的运行状态不存在下一个状态，则该设备这个阶段数据的运行状态不参与计算，即该设备的这个阶段数据的运行状态时长按0计算），将结果数据写入hudi\_gy\_dws层的表machine\_data\_total\_time中，分区字段为etldate且值与fact\_machine\_data表该值相等，etldate作为preCombineField，machine\_id、machine\_record\_date和total\_time作为联合primaryKey。然后使用spark-sql的cli根据machine\_id降序和machine\_record\_date升序排序查询前5条数据，将SQL语句复制粘贴至客户端桌面【Release\任务B提交结果.docx】中对应的任务序号下，将执行结果截图粘贴至客户端桌面【Release\任务B提交结果.docx】中对应的任务序号下；

dws.machine\_data\_total\_time：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 中文含义 | 备注 |
| machine\_id | int | 设备id |  |
| machine\_record\_date | string | 状态日期 | 如：2021-10-01 |
| total\_time | int | 一天运行总时长 | 秒 |

1. 编写scala代码，使用Spark根据hudi\_gy\_dws层表machine\_data\_total\_time，计算每日运行时长前三的设备（若存在运行时长相同的数据时应全部输出，例如有两条并列第二，则第三名次不变，总共输出四条数据）。将计算结果写入ClickHouse数据库shtd\_industry的machine\_data\_total\_time\_top3表中（表结构如下），然后在Linux的ClickHouse命令行中根据查询所有数据，将SQL语句复制粘贴至客户端桌面【Release\任务B提交结果.docx】中对应的任务序号下，将执行结果截图粘贴至客户端桌面【Release\任务B提交结果.docx】中对应的任务序号下；

machine\_data\_total\_time\_top3：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 中文含义 | 备注 |
| date\_day | varchar | 日期 | 如：2021-10-01 |
| first\_id | int | 第一 | 一天运行总时长第一 |
| second\_id | int | 第二 | 一天运行总时长第二 |
| tertiary\_id | int | 第三 | 一天运行总时长第二 |
| first\_time | int | 第一的时长 | 秒 |
| second\_time | int | 第二的时长 | 秒 |
| tertiary\_time | int | 第三的时长 | 秒 |

1. 编写scala代码，使用Spark根据hudi\_gy\_dwd层的fact\_produce\_record表，基于全量历史数据计算各设备生产一个产品的平均耗时，produce\_code\_end\_time值为1900-01-01 00:00:00的数据为脏数据，需要剔除（注：fact\_produce\_record表中，一条数据代表加工一个产品，produce\_code\_start\_time字段为开始加工时间，produce\_code\_end\_time字段为完成加工时间），将设备每个产品的耗时与该设备平均耗时作比较，保留耗时高于平均值的产品数据，将得到的数据写入ClickHouse数据库shtd\_industry的machine\_produce\_per\_avgtime表中（表结构如下），然后在Linux的ClickHouse命令行中根据设备id降序排序查询前3条数据，将SQL语句复制粘贴至客户端桌面【Release\任务B提交结果.docx】中对应的任务序号下，将执行结果截图粘贴至客户端桌面【Release\任务B提交结果.docx】中对应的任务序号下；

machine\_produce\_per\_avgtime：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 中文含义 | 备注 |
| produce\_record\_id | int | 生产记录id | 每生产一件产品产生一条数据 |
| produce\_machine\_id | int | 设备id |  |
| producetime | int | 该产品耗时 |  |
| produce\_per\_avgtime | int | 设备生产一个产品平均耗时 | 单位：秒 |

## 任务C：数据挖掘（10分）

**环境说明：**

|  |
| --- |
| **服务端登录地址详见各任务服务端说明。**  **补充说明：**各节点可通过Asbru工具或SSH客户端进行SSH访问；  主节点MySQL数据库用户名/密码：root/123456（已配置远程连接）；  Spark任务在Yarn上用Client运行，方便观察日志。  该任务均使用Scala编写，利用Spark相关库完成。 |

### 子任务一：特征工程

1. 根据hudi\_gy\_dwd库中fact\_machine\_data表（或MySQL的shtd\_industry库中MachineData表），根据以下要求转换：获取最大分区（MySQL不用考虑）的数据后，首先解析列machine\_record\_data（MySQL中为MachineRecordData）的数据（数据格式为xml，采用dom4j解析，解析demo在客户端/home/ubuntu/Documents目录下），并获取每条数据的主轴转速，主轴倍率，主轴负载，进给倍率，进给速度，PMC程序号，循环时间，运行时间，有效轴数，总加工个数，已使用内存，未使用内存，可用程序量，注册程序量等相关的值（若该条数据没有相关值，则按下表设置默认值），同时转换machine\_record\_state字段的值，若值为报警，则填写1，否则填写0，以下为表结构，将数据保存在Hive的dwd.fact\_machine\_learning\_data，使用cli按照machine\_record\_id升序排序，查询dwd.fact\_machine\_learning\_data前1条数据，将结果截图粘贴至客户端桌面【Release\任务C提交结果.docx】中对应的任务序号下。

dwd.fact\_machine\_learning\_data表结构：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 中文含义 | 备注 |
| machine\_record\_id | int | 主键 |  |
| machine\_id | double | 设备id |  |
| machine\_record\_state | double | 设备状态 | 默认0.0 |
| machine\_record\_mainshaft\_speed | double | 主轴转速 | 默认0.0 |
| machine\_record\_mainshaft\_multiplerate | double | 主轴倍率 | 默认0.0 |
| machine\_record\_mainshaft\_load | double | 主轴负载 | 默认0.0 |
| machine\_record\_feed\_speed | double | 进给倍率 | 默认0.0 |
| machine\_record\_feed\_multiplerate | double | 进给速度 | 默认0.0 |
| machine\_record\_pmc\_code | double | PMC程序号 | 默认0.0 |
| machine\_record\_circle\_time | double | 循环时间 | 默认0.0 |
| machine\_record\_run\_time | double | 运行时间 | 默认0.0 |
| machine\_record\_effective\_shaft | double | 有效轴数 | 默认0.0 |
| machine\_record\_amount\_process | double | 总加工个数 | 默认0.0 |
| machine\_record\_use\_memory | double | 已使用内存 | 默认0.0 |
| machine\_record\_free\_memory | double | 未使用内存 | 默认0.0 |
| machine\_record\_amount\_use\_code | double | 可用程序量 | 默认0.0 |
| machine\_record\_amount\_free\_code | double | 注册程序量 | 默认0.0 |
| machine\_record\_date | timestamp | 记录日期 |  |
| dwd\_insert\_user | string |  |  |
| dwd\_insert\_time | timestamp |  |  |
| dwd\_modify\_user | string |  |  |
| dwd\_modify\_time | timestamp |  |  |

### 子任务二：报警预测

1. 根据子任务一的结果，建立随机森林（随机森林相关参数可自定义，不做限制），使用子任务一的结果训练随机森林模型，然后再将dwd.fact\_machine\_learning\_data\_test（该表字段含义与dwd.fact\_machine\_learning\_data表相同，machine\_record\_state列值为空，表结构自行查看）转成向量，预测其是否报警将结果输出到MySQL数据库shtd\_industry中的ml\_result表中（表结构如下）。在Linux的MySQL命令行中查询出machine\_record\_id为1、8、20、28和36的5条数据，将SQL语句复制并粘贴至客户端桌面【Release\任务C提交结果.docx】中对应的任务序号下，将执行结果截图粘贴至客户端桌面【Release\任务C提交结果.docx】中对应的任务序号下。

ml\_result表结构：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 中文含义 | 备注 |
| machine\_record\_id | int | 主键 |  |
| machine\_record\_state | double | 设备状态 | 报警为1，其他状态则为0 |

## 任务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任务服务端说明。**  **补充说明：**各节点可通过Asbru工具或SSH客户端进行SSH访问；  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。 |

### 子任务一：实时数据采集

1. 在主节点使用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客户端桌面【Release\任务D提交结果.docx】中对应的任务序号下；
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的配置截图粘贴至客户端桌面【Release\任务D提交结果.docx】中对应的任务序号下。

### 子任务二：使用Flink处理Kafka中的数据

编写Scala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）。请将任务启动命令复制粘贴至客户端桌面【Release\任务D提交结果.docx】中对应的任务序号下，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2分钟后根据change\_machine\_id降序查询change\_state\_other\_to\_run\_agg表并再次截图，将两次截图粘贴至客户端桌面【Release\任务D提交结果.docx】中对应的任务序号下；

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，将结果截图粘贴至客户端桌面【Release\任务D提交结果.docx】中对应的任务序号下，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，将结果截图粘贴至客户端桌面【Release\任务D提交结果.docx】中对应的任务序号下，需要Flink启动运行6分钟以后再截图。

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。

## 任务E：数据可视化（15分）

**环境说明：**

|  |
| --- |
| **数据接口地址及接口描述详见各任务服务端说明。**  **注：所有数据排序按照接口返回数据顺序处理即可，不用特意排序。** |

### 子任务一：用折线图展示PM2.5浓度变化

编写Vue工程代码，根据接口，用折线图展示PM2.5浓度变化，同时将用于图表展示的数据结构在浏览器的console中进行打印输出，将图表可视化结果和浏览器console打印结果分别截图并粘贴至客户端桌面【Release\任务E提交结果.docx】中对应的任务序号下。

### 子任务二：用饼状图展示每日各状态总时长

编写Vue工程代码，根据接口，用饼状图展示每日各状态总时长（秒），同时将用于图表展示的数据结构在浏览器的console中进行打印输出，将图表可视化结果和浏览器console打印结果分别截图并粘贴至客户端桌面【Release\任务E提交结果.docx】中对应的任务序号下。

### 子任务三：用柱状图展示每日所有车间各设备平均运行时长

编写Vue工程代码，根据接口，用柱状图展示每日所有车间各设备平均运行时长（秒，四舍五入保留两位小数），同时将用于图表展示的数据结构在浏览器的console中进行打印输出，将图表可视化结果和浏览器console打印结果分别截图并粘贴至客户端桌面【Release\任务E提交结果.docx】中对应的任务序号下。

### 子任务四：用单轴散点图展示设备运行时长

编写Vue工程代码，根据接口，用单轴散点图展示设备运行时长（秒），同时将用于图表展示的数据结构在浏览器的console中进行打印输出，将图表可视化结果和浏览器console打印结果分别截图并粘贴至客户端桌面【Release\任务E提交结果.docx】中对应的任务序号下。

### 子任务五：用单轴散点图展示各设备加工每件产品所需时长

编写Vue工程代码，根据接口，用单轴散点图展示各设备加工每件产品所需时长（秒），同时将用于图表展示的数据结构在浏览器的console中进行打印输出，将图表可视化结果和浏览器console打印结果分别截图并粘贴至客户端桌面【Release\任务E提交结果.docx】中对应的任务序号下。

## 任务F：综合分析（10分）

### 子任务一：Kafka中的数据如何保证不丢失？

在任务D中使用到了Kafka，将内容编写至客户端桌面【Release\任务F提交结果.docx】中对应的任务序号下。

### 子任务二：请简述HBase的rowkey设计原则。

请简要概述HBase的rowkey的重要性并说明在设计rowkey时应遵循哪些原则，将内容编写至客户端桌面【Release\任务F提交结果.docx】中对应的任务序号下。

### 子任务三：数据仓库中怎么处理缓慢变化维，有哪几种方式？

在任务B的数据仓库的设计过程中，会出现缓慢变化维的问题，缓慢变化维是指一些维度表的数据不是静态的，而是会随着时间而缓慢地变化，在数仓设计中有哪些方式应对这些问题？将内容编写至客户端桌面【Release\任务F提交结果.docx】中对应的任务序号下。