**2023年全国职业院校技能大赛**

**赛题第05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 Hive、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 完全分布式安装配置

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-2.7.7.tar.gz、jdk-8u212-linux-x64.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JDK安装包解压到/opt/module路径中(若路径不存在，则需新建)，将JDK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. 修改容器中/etc/profile文件，设置JDK环境变量并使其生效，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. 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，用scp命令并使用绝对路径从master复制JDK解压后的安装文件到slave1、slave2节点（若路径不存在，则需新建），并配置slave1、slave2相关环境变量，将全部scp复制JDK的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；
4. 在master将Hadoop解压到/opt/module(若路径不存在，则需新建)目录下，并将解压包分发至slave1、slave2中，其中master、slave1、slave2节点均作为datanode，配置好相关环境，初始化Hadoop环境namenode，将初始化命令及初始化结果截图（截取初始化结果日志最后20行即可）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. 启动Hadoop集群（包括hdfs和yarn），使用jps命令查看master节点与slave1节点的Java进程，将jps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。

### 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、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：Kafka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zookeeper-3.4.6.tar.gz、kafka\_2.12-2.0.0.tgz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Zookeeper、kafka安装包解压到/opt/module目录下，将kafka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. 配置好zookeeper，其中zookeeper使用集群模式，分别将master、slave1、slave2作为其节点（若zookpeer已安装配置好，则无需再次配置），配置好Kafka的环境变量，使用kafka-server-start.sh --version查看Kafka的版本内容，并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；
3. 完善其他配置并分发Kafka文件到slave1、slave2中，并在每个节点启动Kafka，创建Topic，其中Topic名称为installtopic，分区数为2，副本数为2，将创建命令和创建成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）；  Hive的配置文件位于/opt/apache-hive-2.3.4-bin/conf/  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ChangeRecord，BaseMachine，MachineData， ProduceRecord全量抽取到Hive的ods库中对应表changerecord，basemachine， machinedata，producerecord中。

1. 抽取MySQL的shtd\_industry库中ChangeRecord表的全量数据进入Hive的ods库中表changerecord，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）。使用hive cli执行show partitions ods.changerecord命令，将hive cli的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；
2. 抽取MySQL的shtd\_industry库中BaseMachine表的全量数据进入Hive的ods库中表basemachine，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）。使用hive cli执行show partitions ods.basemachine命令，将hive cli的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；
3. 抽取MySQL的shtd\_industry库中ProduceRecord表的全量数据进入Hive的ods库中表producerecord，剔除ProducePrgCode字段，其余字段排序、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）。使用hive cli执行show partitions ods.producerecord命令，将hive cli的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；
4. 抽取MySQL的shtd\_industry库中MachineData表的全量数据进入Hive的ods库中表machinedata，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）。使用hive cli执行show partitions ods.machinedata命令，将hive cli的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。

### 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Hive的dwd库中对应表中。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ods库中changerecord的全量数据进入Hive的dwd库中表fact\_change\_record，抽取数据之前需要对数据根据changeid和changemachineid进行联合去重处理，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。使用hive cli按照change\_machine\_id、change\_id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；
2. 抽取ods库中basemachine的全量数据进入Hive的dwd库中表dim\_machine,抽取数据之前需要对数据根据basemachineid进行去重处理。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。使用hive 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；
3. 抽取ods库中producerecord的全量数据进入Hive的dwd库中表fact\_produce\_record,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。使用hive cli按照produce\_machine\_id、produce\_record\_id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；
4. 抽取ods库中machinedata的全量数据进入Hive的dwd库中表fact\_machine\_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。使用hive cli按照machine\_id、machine\_record\_id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题任务为并行任务且它们依赖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dwd层的fact\_change\_record表关联dim\_machine表统计每个车间中所有设备运行时长（即设备状态为“运行”）的中位数在哪个设备（为偶数时，两条数据原样保留输出），若某个设备运行状态当前未结束（即change\_end\_time值为空）则该状态不参与计算，计算结果存入MySQL数据库shtd\_industry的machine\_running\_median表中（表结构如下），然后在Linux的MySQL命令行中根据所属车间、设备id均为降序排序，查询出前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;

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 中文含义 | 备注 |
| machine\_id | int | 设备id |  |
| machine\_factory | int | 所属车间 |  |
| total\_running\_time | int | 运行总时长 | 结果以秒为单位 |

1. 编写Scala代码，使用Spark根据dwd层的fact\_change\_record表和dim\_machine表统计，计算每个车间设备的月平均运行时长与所有设备的月平均运行时长对比结果（即设备状态为“运行”，结果值为：高/低/相同），月份取值使用状态开始时间的月份，若某设备的运行状态当前未结束（即change\_end\_time值为空）则该状态不参与计算，计算结果存入MySQL数据库shtd\_industry的machine\_running\_compare表中（表结构如下），然后在Linux的MySQL命令行中根据车间号降序排序，查询出前2条，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;

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 中文含义 | 备注 |
| start\_month | varchar | 月份 | 如：2021-12 |
| machine\_factory | int | 车间号 |  |
| comparison | varchar | 比较结果 | 高/低/相同 |
| factory\_avg | varchar | 车间平均时长 |  |
| company\_avg | varchar | 所有设备平均时长 |  |

1. 编写Scala代码，使用Spark根据dwd层的fact\_change\_record表展示每一个设备最近第二次的状态（倒数第二次），时间字段选用change\_start\_time，如果设备仅有一种状态，返回该状态（一个设备不会同时拥有两种状态），存入MySQL数据库shtd\_industry的recent\_state表中（表结构如下），然后在Linux的MySQL命令行中根据设备id降序排序，查询出前5条，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；

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 中文含义 | 备注 |
| machine\_id | int | 设备id |  |
| record\_state | varchar | 状态信息 | 最近第二次的状态 |
| change\_start\_time | varchar | 状态开始时间 |  |
| change\_end\_time | varchar | 状态结束时间 |  |

## 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）；  Hive的配置文件位于/opt/apache-hive-2.3.4-bin/conf/  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dwd.fact\_machine\_learning\_data，使用hive 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hive中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语句与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HBase中的gyflinkresult:Produce5minAgg表，rowkey“设备id-系统时间”（如：123-2023-01-01 12:06:06.001），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，每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代表为检验；

时间语义使用Processing Time。

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 中文含义 |
| rowkey | String | 设备id-系统时间（如：123-2023-01-01 12:06:06.001） |
| machine\_id | String | 设备id |
| total\_produce | String | 最近5分钟生产总数 |

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。

## 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：用多个饼状图展示某设备每天各状态持续时间

编写Vue工程代码，根据接口，用多个饼状图展示OP170B该设备的每天各个状态的持续时长（秒）占比，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：用散点图展示设备运行时长

编写Vue工程代码，根据接口，用基础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。

### 子任务三：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。

### 子任务四：用折线图展示PM2.5浓度变化

编写Vue工程代码，根据接口，用折线图展示设备所处环境的PM2.5浓度的变化，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。

子任务五：用条形图展示每日各设备产量

编写Vue工程代码，根据接口，用条形图展示每日各设备产量，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：请简述维度建模和范式建模区别。

请简述你对维度建模和范式建模的理解以及他们的区别，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Flink资源管理中Task Slot的概念。

请简述你对Task Slot的理解，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：Flume采集数据会导致数据丢失吗？请简述其原理。

请简述Flume数据采集是否会导致数据丢失以及其原理，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。