2022年全国职业院校技能大赛

高职组

“大数据技术与应用”

赛项赛卷（6卷）

任

务

书

参赛队编号：

**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。党中央、国务院高度重视工业互联网发展，习近平总书记连续四年对推动工业互联网发展做出重要指示。加快发展工业互联网产业，不仅是各国顺应产业发展大势，抢占产业未来制高点的战略选择，也是我国推动制造业质量变革、效率变革和动力变革，实现高质量发展的客观要求。

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。

## 模块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模块服务端说明。**  **补充说明：**宿主机可通过Asbru工具或SSH客户端进行SSH访问；  相关软件安装包在宿主机的/opt目录下，请选择对应的安装包进行安装，用不到的可忽略；  所有模块中应用命令必须采用绝对路径；  从本地仓库中拉取镜像，并启动3个容器  进入Master节点的方式为  docker exec –it master /bin/bash  进入Slave1节点的方式为  docker exec –it slave1 /bin/bash  进入Slave2节点的方式为  docker exec –it slave2 /bin/bash  同时将/opt目录下的所有安装包移动到3个容器节点中。 |

### 任务一：Hadoop 完全分布式安装配置

本环节需要使用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. 将Master节点JDK安装包解压并移动到/usr/java路径(若路径不存在，则需新建)，将命令复制并粘贴至对应报告中;
2. 修改/root/profile文件，设置JDK环境变量，配置完毕后在Master节点分别执行“java”和“javac”命令，将命令行执行结果分别截图并粘贴至对应报告中;
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，使用绝对路径从Master节点复制JDK解压后的安装文件到Slave1、Slave2节点，并配置相关环境变量，将全部复制命令复制并粘贴至对应报告中;
4. 在Master节点将Hadoop解压到/opt目录下，并将解压包分发至Slave1、Slave2节点中,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复制粘贴至对应报告中；
5. 启动Hadoop集群，查看Master节点jps进程，将查看结果复制粘贴至对应报告中。

### 任务二：Sqoop安装配置

本环节需要使用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. 将Master节点Sqoop安装包解压到路径/opt目录下，将完整命令复制粘贴至对应报告中；
2. 修改Sqoop环境变量，并使环境变量只对当前root用户生效,将完整命令复制粘贴至对应报告中；
3. 修改并配置sqoop-env.sh文件，将完整命令复制粘贴至对应报告中；
4. 测试Sqoop连接Master节点的MySQL数据库是否成功并展示所有的database，将完整命令复制粘贴至对应报告中。

### 任务三：Hive安装配置

本环节需要使用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. 将Master节点Hive安装包解压到/opt目录下，将完整命令复制粘贴至对应报告中；
2. 把解压后的apache-hive-1.2.2-bin文件夹更名为hive；进入hive文件夹使用ls进行查看，将完整命令复制粘贴至对应报告中；
3. 设置Hive环境变量，并使环境变量只对当前root用户生效；并将环境变量配置内容复制并粘贴至对应报告中；
4. 将Hive安装目录里hive-default.xml.template文件更名为hive-site.xml；将完整命令复制粘贴至对应报告中；
5. 修改hive-site.xml配置文件，将MySQL数据库作为Hive元数据库。将配置文件中配置Hive元存储的相关内容复制并粘贴至对应报告中；
6. 初始化Hive元数据，将MySQL数据库JDBC驱动拷贝到Hive安装目录的lib文件夹下；并通过schematool相关命令执行初始化，将初始化结果复制粘贴至对应报告中；
7. 完善其他配置并启动Hive，将命令行输出结果复制粘贴至对应报告中。

## 模块B：离线数据处理（25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；  Master节点MySQL数据库用户名/密码：root/123456（已配置远程连接）；  Hive的元数据启动命令为:  nohup hive --service metastore &  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。 |

### 任务一：数据抽取

编写Sqoop脚本，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\_store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）。并在hive cli执行show partitions ods.ChangeRecord命令，将Sqoop提交命令及hive cli的执行结果分别截图复制粘贴至对应报告中；
2. 抽取MySQL的shtd\_store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）。并在hive cli执行show partitions ods.Basemachine命令，将Sqoop提交命令及hive cli的执行结果分别截图复制粘贴至对应报告中；
3. 抽取MySQL的shtd\_store库中ProduceRecord表的全量数据进入Hive的ods库中表ProduceRecord，字段排序、类型不变，同时添加静态分区，分区字段类型为String，且值为当前比赛日的前一天日期（分区字段格式为yyyyMMdd）。并在hive cli执行show partitions ods.ProduceRecord命令，将Sqoop提交命令及hive cli的执行结果分别截图复制粘贴至对应报告中；
4. 抽取MySQL的shtd\_store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）。并在hive cli执行show partitions ods.ProduceRecord命令，将Sqoop提交命令及hive cli的执行结果分别截图复制粘贴至对应报告中；

### 任务二：数据清洗

编写Hive SQL代码，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，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均填写操作时间，并进行数据类型转换。并在hive cli执行desc dwd.fact\_change\_record命令，将结果内容复制粘贴至对应报告中；

|  |
| --- |
| Set hive.support.quoted.identifiers=none;  Insert into table dwd. fact\_change\_record partition(etldate) select `(etldate)?.+` ,”user1”,date\_format(current\_timestamp,”yyyy-MM-dd HH:mm:ss”), ”user2”,date\_format(current\_timestamp,”yyyy-MM-dd HH:mm:ss”), etldate, from ods. ChangeRecord |

1. 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均填写操作时间，并进行数据类型转换。在hive cli中按照Base\_machine顺序排序，查询dim\_machine前2条数据，将结果内容复制粘贴至对应报告中；

|  |
| --- |
| Set hive.support.quoted.identifiers=none;  Select `(etldate|pm)?.+`,user1,xxx,user2,xxx,etldate from (  Select \*,row\_number() over(partition by BasemachineId) as pm from Basemachine) as t where t.pm=1 |

1. 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均填写操作时间，并进行数据类型转换。在hive cli中按照produce\_machine\_id顺序排序，查询fact\_produce\_record前2条数据，将结果内容复制粘贴至对应报告中；
2. 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均填写操作时间，并进行数据类型转换。并在hive cli执行show partitions dwd. fact\_machine\_data命令，将结果内容复制粘贴至对应报告中；

### 任务三：指标计算

1. 编写Scala工程代码，根据dwd层表统计每个月、每个设备、每种状态的时长，存入MySQL数据库shtd\_store的表（表结构如下）中，然后在Linux的MySQL命令行中根据设备id、状态持续时长均为逆序排序，查询出前5条，将SQL语句与执行结果截图粘贴至对应报告中;

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 中文含义 | 备注 |
| MachineID | int | 设备id |  |
| ChangeRecordState | string | 状态 |  |
| duration\_time | string | 持续时长 | 当月该状态的时长和 |
| Year | int | 年 | 状态产生的年 |
| Month | int | 月 | 状态产生的月 |

1. 编写Hive SQL代码，根据dwd层MachineData表统计设备运行时各种状态（报警、离线、待机、运行）的连续日期。查询2021年10月各个设备连续同状态period\_state的起止日期（ChangeStartTime和ChangeEndTime）。即如果机器运行了，就是运行状态的起止日期，如果机器待机了，就是待机状态的起止日期。最后结果各个设备内部按照起始日期排序。然后在Linux的MySQL命令行中根据machineID倒序排序，查询出前5条，将SQL语句与执行结果截图粘贴至对应报告中。

下框中有点bug：

|  |
| --- |
| SELECT mn machineID, ms period\_state, from\_unixtime(mis) StartTime, c.endtime EndTime  FROM  (FROM  (FROM  (SELECT machine\_no mn, machine\_state ms, UNIX\_TIMESTAMP(starttime) st,  UNIX\_TIMESTAMP(endtime) et, unix\_timestamp(endtime) - unix\_timestamp(starttime) + 1 dif  FROM changerecord) t1  SELECT mn, ms, st, et, dif,  SUM(dif) over(PARTITION BY mn, ms ORDER BY st ROWS BETWEEN unbounded preceding AND 1 PRECEDING) dif2) t2  SELECT mn, ms, MIN(st) mis, MAX(st) mas GROUP BY (st - dif2), mn, ms ORDER BY mn, ms) t3  INNER JOIN changerecord c ON mn = c.machine\_no AND ms = c.machine\_state AND mas = c.starttime AND mis < mas  and mn='Equipment#9' and ms='waiting'  ORDER BY machineID, StartTime LIMIT 100; |

直接排序取前两个组合为一条记录，不使用排名函数：

|  |
| --- |
| select t3.riqi,collect\_list(t3.machine\_no)[0],collect\_list(t3.machine\_no)[1],collect\_list(t3.sc)[0],collect\_list(t3.sc)[1],t3.machine\_factoryId from (  select distinct machinedata.machine\_factoryId,machinedata.machine\_no,SUBSTR(startTime,1,10) as riqi,  sum(UNIX\_TIMESTAMP(endTime)-UNIX\_TIMESTAMP(startTime)) over(partition by SUBSTR(startTime,1,10),machinedata.machine\_factoryId,machinedata.machine\_no ) as sc from machinedata  inner join changerecord on machinedata.machine\_no = changerecord.machine\_no  where changerecord.machine\_state='running'  order by sc desc) t3 group by t3.riqi,t3.machine\_factoryId |

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 中文含义 | 备注 |
| machineID | int |  |  |
| period\_state | string |  | 如：“待机” “运行” |
| StartTime | timestamp |  |  |
| EndTime | timestamp |  |  |

## 模块C：数据挖掘（10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；  Master节点MySQL数据库用户名/密码：root/123456（已配置远程连接）；  Hive的元数据启动命令为:  nohup hive --service metastore &  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。  该模块均使用Scala编写，利用Spark相关库完成。 |

### 任务一：特征工程

1. 根据dwd库中fact\_machine\_data表，根据以下要求转换：获取最大分区的数据后，首先解析列get\_xmldata中的数据（数据格式为xml，采用dom4j解析，会给出解析demo），并获取主轴转速，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，同时转换machine\_record\_stat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，在hive cli中按照machine\_record\_id顺序排序，查询dwd.fact\_machine\_learning\_data前1条数据，将结果内容复制粘贴至对应报告中；

dwd.fact\_machine\_learning\_data表结构：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 中文含义 | 备注 |
| machine\_record\_id | int | 自增长id |  |
| machine\_id | double | 机器id |  |
| 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\_cricle\_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\_efficive\_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\_state | double | 机器状态 | 默认0.0 |

### 任务二：报警预测

1. 根据任务一的结果，建立随机森林（随机森林相关参数考生可自定义，不做限制），使用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（表结构与dwd.fact\_machine\_learning\_data一致，但machine\_record\_state列值为空）转成向量，预测其是否报警将结果输出到MySQL库shtd\_industry中的ml\_result表中。在Linux的MySQL命令行中查询出所有数据并按照machine\_record\_id顺序排序，将SQL语句与执行结果截图粘贴至对应报告中。

ml\_result表结构：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 中文含义 | 备注 |
| machine\_record\_id | int | 自增长id |  |
| machine\_record\_state | double | 机器状态 | 报警为1，其他状态则为0 |

## 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端口看是否已启动，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。  Flink run –m yarn-cluster –c com.xxx.xxx.XXX /opt/jars/xxx.jar |

### 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和ProduceRecord，分区数为4），将Flume的配置截图粘贴至对应报告中；
2. Flume接收数据注入kafka 的同时，将数据备份到HDFS目录/user/test/flumebackup下，将备份结果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Topic为ChangeRecord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，各设备每小时生产产品总数，将key设置成totalproduce存入Redis中（再使用hash数据格式，key存放为设备id，value存放为该设备生产总数），使用redis cli以get 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一分钟以上，第一次截图放前面，第二次放后面；
2. 使用Flink消费Kafka中ChangeRecord主题的数据，统计每个设备状态信息，当某设备的状态在24小时内连续三小时为“预警”状态没有改变，则将该设备的状态数据写入MySql表recordstatealarm中，然后在Linux的MySQL命令行中根据MachineID逆序排序，查询出前3条，将SQL语句与执行结果截图粘贴至对应报告中。

## 模块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模块服务端说明。** |

### 任务一：用柱状图展示设备各个状态运行时长（接口20）

编写Vue工程代码，根据接口，用柱状图展示某年(2018)某月（1月）各设备的各个状态的运行时长，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对应报告中。

### 任务二：用饼状图展示所有设备的各状态总时长（接口20）

编写Vue工程代码，根据接口，用饼状图展示某年（2018年）第一季度所有设备的各状态总时长占比，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对应报告中。

任务三：用折线图展示机器运行时长**（接口21）**  
编写Vue工程代码，根据接口，用折线图展示某机器（Equipment#26）在某一周（第1周传入数字1即可）每天的运行时长，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对应报告中。

任务四：用散点图展示PM2.5浓度变化**（接口22）设备ID传入：**（Equipment#26）  
编写Vue工程代码，根据接口，用散点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对应报告中。  
任务五：用条形图展示各机器每日产量（2014年1月各个设备的产量和Top8）**（接口23）**  
编写Vue工程代码，根据接口，用柱状图展示某机器在某天（2014年1月）机器的产量，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对应报告中。  
任务六：用折柱混合图展示机器日均产量和车间日均产量（2014年1月日均产量排名前6的设备和对应车间的产量）  
编写Vue工程代码，根据接口，用折柱混合图展示某机器在某月的日均产量和车间的日均产量，其中柱状图展示各机器的日均产量，折线图展示该机器所在车间的日均产量，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对应报告中。

## 模块F：综合分析（10分）

### 任务一：如何解决Job运行效率低的问题

在模块B中出现某些Job运行时间较长，你认为可能是哪些情况造成？有什么相关的方法吗？将内容编写至对应报告中。

### 任务二：如何解决数据倾斜问题

当使用spark产生了数据倾斜的情况时,请问有哪些方法可以解决这个问题呢？将内容编写至对应报告中。

### 任务三：简要描述任务过程中的问题并进行总结

将内容编写至对应报告中。