**模块A：大数据(30分)**

**竞赛题目：大数据运维与应用**

**背景简介**

Hadoop得以在大数据处理应用中广泛应用得益于其自身在数据提取、变形和加载(ETL)方面上的天然优势。Hadoop的分布式架构，将大数据处理引擎尽可能的靠近存储，对例如像ETL这样的批处理操作相对合适，因为类似这样操作的批处理结果可以直接走向存储。Hadoop的MapReduce功能实现了将单个任务打碎，并将碎片任务(Map)发送到多个节点上，之后再以单个数据集的形式加载(Reduce)到数据仓库里 。

Sqoop是一个分布式的数据迁移工具，可以将一个关系型数据库（例如 ：MySQL ,Oracle ,Postgres等）中的数据导进到Hadoop的HDFS中，也可以将HDFS的数据导进到关系型数据库中。

hive是基于Hadoop工具，用来进行数据提取、转化、加载，这是一种可以存储、查询和分析存储在Hadoop中的大规模数据的机制。hive数据仓库工具能将结构化的数据文件映射为一张数据库表，并提供SQL查询功能，能将SQL语句转变成MapReduce任务来执行。

Spark 是一种与 Hadoop 相似的开源集群计算环境，但是两者之间还存在一些不同之处，这些有用的不同之处使 Spark 在某些工作负载方面表现得更加优越，换句话说，Spark 启用了内存分布数据集，除了能够提供交互式查询外，它还可以优化迭代工作负载。

**环境介绍**

**远程环境：**

每位选手共三台aws云服务器，内部包含示与竞赛任务相关的Mysql数据库,以及大数据集群组件相关安装包。

**三台服务器环境信息如下：**

***master节点***

* 内网IP地址：用于内部大数据访问配置
* 操作系统： Red Hat 7.3.1-6
  + 用户名： ec2-user
* Hadoop 安装包目录~/hadoop搭建相关安装包
  + hadoop-2.9.2
  + sqoop-1.4.7
  + apache-hive-3.1.2
  + apache-flume-1.9.0

***node1节点***

* 内网IP地址：用于内部大数据访问配置
* 操作系统： Red Hat 7.3.1-6
  + 用户名： ec2-user
* Mysql 数据库
  + 用户名: root
  + 密码：123456
  + 数据库名：foshan\_competition
  + 连接 Mysql 数据库命令：mysql -uroot -p123456

***node2节点***

* 内网IP地址：用于内部大数据访问配置
* 操作系统： Red Hat 7.3.1-6
  + 用户名： ec2-user
* Mysql 数据库（hive数据仓库的元数据库）
  + 用户名: root
  + 密码：123456

**本地环境：**

WINDOWS 环境（带 chrome 或 firefox 浏览器）

**注意事项**

* 部分问题需要手动保存答案，注意按题目要求的格式保存
* 部分问题已给出解题步骤，可按给出步骤执行
* Hadoop 集群相关配置文件已配置完成，可根据任务要求启动
* Sqoop 和 Hive 大数据组件已配置完成，可直接应用，不需要选手再次安装配置
* Spark 集群已配置好，只需直接启动集群即可。
* 遇到linux权限相关问题时，可在命令前加sudo
* 比赛完成后**禁止**点击**退出模块**按钮，否则后果自负

**数据介绍**

在 Hive 数据仓库中已经存在一个名为 deflaut 的库，该库下面包含多个表，其中 ratings 是后面竞赛所需要的数据表。

ratings （商品评分）：

| **字段** | **数据类型** | **字段说明** |
| --- | --- | --- |
| posted | timestamp | 时间戳 |
| cust\_id | int | 客户 ID |
| prod\_id | int | 产品 ID |
| rating | tinyint | 商品评分 |
| message | string | 客户留言 |

在 spark 考核点中，我们将用到 ratings.csv 文件中的数据，数据字段的具体介绍如下所示。

ratings.csv（电影评分）：

| **字段** | **数据类型** | **字段说明** |
| --- | --- | --- |
| userId | string | 用户 ID |
| movieId | string | 电影名称 ID |
| rating | string | 电影评分 |
| timestamp | string | 评分时间戳 |

**任务列表**

1. Hadoop 大数据集群启动
2. HDFS 文件操作
3. Sqoop 相关操作
4. Hive 数据分析
5. Spark 分布式计算

**任务1：Hadoop 大数据集群启动（3分）**

1.1 配置hosts文件

1.2 修改 Hadoop 配置文件

1.3 启动 Hadoop 集群

**1.1 配置hosts文件**

分别获取三个节点中的内网IP地址与对应节点名称（可用ifconfig命令查询IP地址），并把三个IP地址以及对应节点名称一起分别写入到三个节点的/etc/hosts文件当中。（1分）

所示格式如下（如有多余master、node1、node2的配置，删除或者注释掉，否者会导致正确配置不起效）：

xxx.xx.xx.xx master  
xxx.xx.xx.xx node1  
xxx.xx.xx.xx node2

**1.2 修改 Hadoop 配置文件**

在每个节点上操作，修改HDFS数据自动备份数量，把原有的备份数量1，修改为2，并保存修改完成后的配置文件。（1分）

**1.3 启动Hadoop集群**

启动Hadoop集群，包含master节点上的NameNode、SecondaryNameNode、ResourceManager，以及node1、node2节点上的NodeManager、DataNode。（1分）

**任务2：HDFS文件操作（3分）**

2.1 HDFS 上新建目录

2.2 修改HDFS目录权限

2.3 上传本地文件到HDFS上

**2.1 HDFS上新建目录**

在master节点上操作，在HDFS上创建一个名为/dualcore的目录。（1分）

**2.2 修改HDFS目录权限**

在master节点上操作,HDFS上设置/dualcore目录权限为drwxrwxrwx。（1分）

**2.3 上传本地文件到HDFS上**

在master节点上操作,上传本地~/data/loyalty\_data.txt文件到HDFS的/dualcore目录下。（1分）

**任务3：Sqoop 相关操作（6分）**

3.1 Sqoop连接Mysql并查询

3.2 Sqoop导入

3.3 Sqoop导出

**3.1 Sqoop 连接 Mysql并查询**

在master节点上操作,运用sqoop，查询node1节点上foshan\_competition数据库中的order\_items表，查看order\_items中不重复的order\_item\_product\_id数量，并保存结果到~/results/task3-1.txt文件中。（2分）

**3.2 Sqoop 导入**

在master节点上操作，使用Sqoop将node1节点上Mysql里foshan\_competition数据库中products数据表导入到hive数据仓库中的default数据库里。（2分）

**3.3 Sqoop导出**

在master节点上操作，使用Sqoop将HDFS上的/sqoop/customers文件数据，导出到node1节点Mysql里foshan\_competition数据库中的customers表里。（2分）

**任务4：Hive 数据分析（7分）**

4.1 创建 Hive 外部表

4.2 Hive 查询语句

4.3 分析数值产品评分

**4.1 创建 Hive 外部表**

Dualcore 最近开始了一个忠诚度计划，以奖励我们最好的客户。同事已经提供了一个数据样本，其中包含有关注册该程序的客户的信息，包括他们的电话号码(作为 map)，以前的订单 ID(作为数组)和总结了过去订单的最小值、最大值、平均值和总值的结构。您将创建表，使用提供的数据填充表，然后运行一些查询来练习引用这些类型的字段。（2分）

依据以下给出的要求，在 Hive 中的default数据库里创建具有以下特征的表。

* **Name**: loyalty\_program
* **Type**: EXTERNAL
* **Columns**:

| **字段名称** | **字段类型** |
| --- | --- |
| cust\_id | STRING |
| fname | STRING |
| lname | STRING |
| email | STRING |
| level | STRING |
| phone | MAP |
| order\_ids | ARRAY |
| order\_value | STRUCT |

* **Field Terminator**: |(竖线)
* **Collection item terminator**: , (逗号)
* **Map Key Terminator**: : (冒号)
* **Location**:/dualcore/loyalty\_data.txt

**4.2 Hive 查询语句**

对上步任务创建的新表loyalty\_program实现以下要求的简单查询：

* 运行查询以选择客户ID1200866的HOME电话号码。并记录结果到 master 节点上 ec2-user 用户下的~/results/task4-2.txt文件中。（1分）
* 从order\_ids数组中为客户ID1200866选择第三个元素。并记录结果到 master 节点上 ec2-user 用户下的~/results/task4-2.txt文件中。（1分）
* 从order\_value结构中为客户ID1200866选择total属性。并记录结果到 master 节点上 ec2-user 用户下的~/results/task4-2.txt文件中。（1分）

**4.3 分析数值产品评分**

客户评价和反馈是客户和像 Dualcore 的零售商的重要信息来源。我们希望找到客户最喜欢的产品，但必须防止评级很少的产品误导。

探查 Hive 中的ratings表中数据，查询所有具有至少50个评级的产品中评分平均值最高的产品？并记录prod\_id(产品ID)结果到 master 节点上 ec2-user 用户下的~/results/task4-3.txt文件中。（2分）

**任务5：Flume 文件监控（6分）**

5.1 Flume 组件安装

5.2 Flume 文件监控

**5.1 Flume 组件安装**

在master节点上操作，安装 Flume 组件。flume 安装包路径为：~/hadoop搭建相关安装包/apache-flume-1.9.0-bin.tar.gz（2分）

**5.2 Flume 文件监控**

在master节点上操作，启动 flume 对 /home/ec2-user/flume文件夹进行监控，对于传入到/home/ec2-user/flume/文件夹下的文件，自动同步到 hdfs 上的/flume/路径下面。（4分）

**任务6：Spark 分布式计算（5分）**

6.1 计算不重复的电影数量

6.2 计算电影平均评分

6.3 计算用户在夜晚的评分总记录数

**6.1 计算不重复的电影数量**

在master节点上操作，读取 hdfs 上/spark/movie\_ratings.csv数据，启动 spark 计算不重复的电影（movieId）数量，并保存结果到~/results/task6-1.txt文件中。（1分）

**6.2 计算电影平均评分**

在master节点上操作，读取 hdfs 上/spark/movie\_ratings.csv数据，运用 spark 计算电影平均评分，保留两位小数，并保存结果到~/results/task6-2.txt文件中。**(平均评分=总评分数/总评分记录数）**（1分）

**6.3 计算用户在夜晚的评分总记录数**

在master节点上操作，读取 hdfs 上/spark/movie\_ratings.csv数据，按照以下要求把每天的评分时间字段划分为 morning、lunch、afternoon、evening、night 五个时间段，并运用 spark 计算用户在夜晚（evening）的评分总记录数，并保存结果到~/results/task6-3.txt文件中。（3分）

评分时间戳字段分类（以小时为单位）：

* morning：7~12（包含7点，不含12点）
* lunch：12~14（包含12点，不含14点）
* afternoon：14~18（包含14点，不含18点）
* evening：18~24（包含18点，不含24点）
* night：0~7（包含0点，不含7点）