**《Hadoop系统开发》实验报告**

**Hive实验**

**淘宝双11数据分析**

**成 绩：**

**学 号： 2104240713**

**姓 名： 江成**

**班 级： 大数据12101**

**指导教师： 梁晶**

**报告日期：2023 年11月20日**

**1、实验数据集说明**

数据集压缩包是淘宝2015年双11前6个月(包含双11)的交易数据user\_log.csv，约有5000万条日志，其中的字段定义如下：

1. user\_id | 买家id

2. item\_id | 商品id

3. cat\_id | 商品类别id

4. merchant\_id | 卖家id

5. brand\_id | 品牌id

6. month | 交易时间:月

7. day | 交易事件:日

8. action | 行为,取值范围{0,1,2,3},0表示点击，1表示加入购物车，2表示购买，3表示关注商品

9. age\_range | 买家年龄分段：1表示年龄<18,2表示年龄在[18,24]，3表示年龄在[25,29]，4表示年龄在[30,34]，5表示年龄在[35,39]，6表示年龄在[40,49]，7和8表示年龄>=50,0和NULL则表示未知

10. gender | 性别:0表示女性，1表示男性，2和NULL表示未知

11. province| 收货地址省份

**注意该数据集未经清洗，其中有可能有null字段，会导致查询失败。可自行编写Linux脚本或者Python程序删除有null字段的行。**

**2、使用Hive完成以下功能：**

1、计算出表内有多少条行数据；

2、查询双11那天有多少人购买了商品（要去掉同一个user\_id的重复购买）；

3、给定品牌brand\_id为2661，求双11当天此品牌商品的销售数量；

4、查询双11那天女性购买商品的数量；

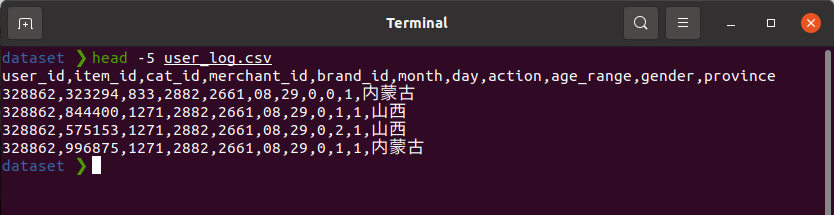
5、根据收货地址，统计各个省双11那天用户下单的总次数，并进行排序。

6、 思考题：对比在Hive和MySQL中以上查询语句的执行时间，分析Hive和MySQL查询效率差异产生的原因。

**3、Hive数据导入和数据查询的主要语句和输出结果示例；**

1. **准备数据集**
2. **下载并查看数据集**

执行 `head -5 user\_log.csv` 查看前五行。

****

1. **数据预处理**

user\_log.csv的第一行都是字段名称，我们在文件中的数据导入到数据仓库Hive中时，不需要第一行字段名称，因此，这里在做数据预处理时，删除第一行。

执行 `sed -i '1d' user\_log.csv` 命令删除第一行；

再次执行 ` head -5 user\_log.csv ` 查看前五行。

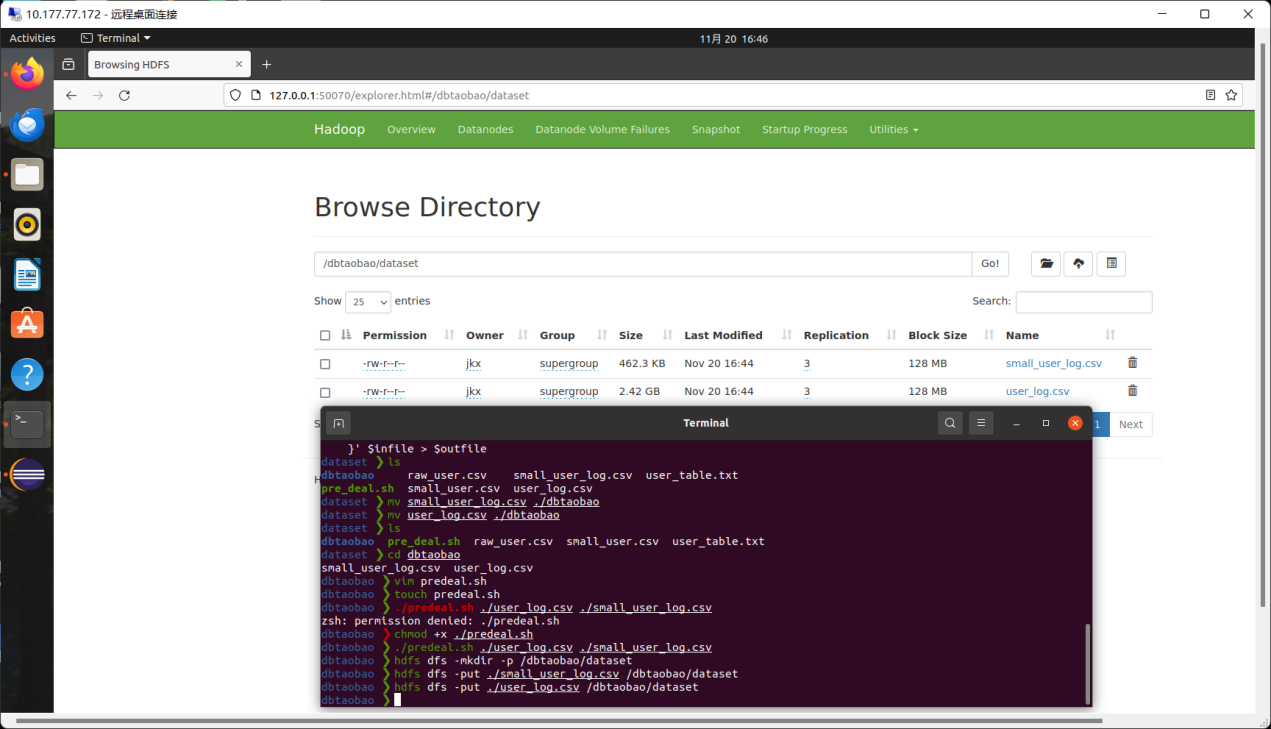
**文本

描述已自动生成**

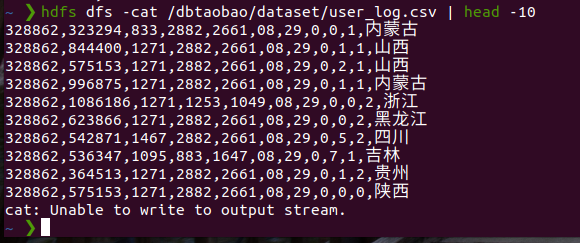
1. **将数据集上传至HDFS**

使用 `hdfs dfs -mkdir -p /dbtaobao/dataset` 在 HDFS 中创建目录 `dbtaobao` 及其子目录 `dataset` 。

使用 `hdfs dfs -put ./user\_log.csv /dbtaobao/dataset` 将本地数据集上传至 HDFS 中。

****

使用 `hdfs dfs -cat /dbtaobao/dataset/user\_log/small\_user\_log.csv | head -10` 查看HDFS中的user\_log.csv的前10条记录。

****

至此，数据集准备完成。

1. **在Hive上创建数据库**

Hive的元数据存储在MySQL数据库中，故先使用 `service mysql start` ，启动MySQL数据库。

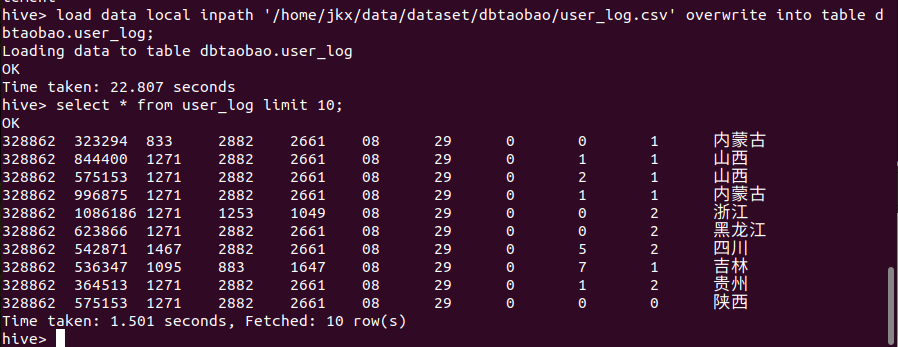
使用 `hive` 命令进入Hive。

`create database dbtaobao;` `use dbtaobao;` 使用这两个命令创建并使用 `dbtaobao` 数据库。

使用 `CREATE TABLE dbtaobao.user\_log(user\_id INT,item\_id INT,cat\_id INT,merchant\_id INT,brand\_id INT,month STRING,day STRING,action INT,age\_range INT,gender INT,province STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY ','` 在 `dbtaobao` 数据库下创建user\_log表。

使用 `load data local inpath ‘/home/jkx/data/dataset/dbtaobao/user\_log.csv’ overwrite into table dbtaobao.user\_log` 命令向user\_log表中导入数据。

使用 `select \* from user\_log limit 10;` 命令查询表中的数据。



使用 `show create table user\_log` 查看 `user\_log` 表中数据类型。

文本

描述已自动生成

使用 `desc user\_log` 命令查看简化后的`user\_log` 表中数据类型。

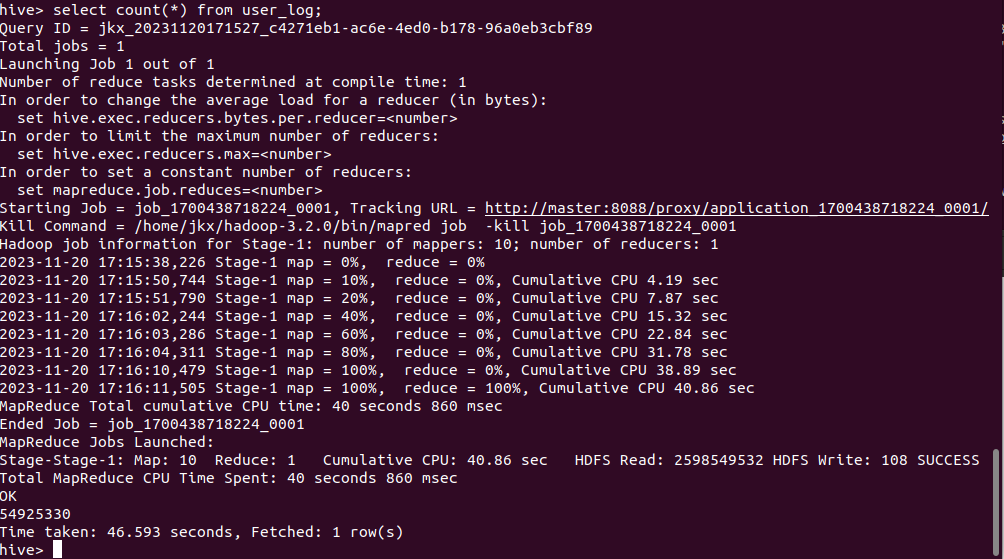
图片包含 文本

描述已自动生成

至此，在Hive上创建数据库完成。

1. **使用Hive完成以下功能**
2. 计算出表内有多少条行数据；

使用 `select count(\*) from user\_log` 命令查看 `user\_log` 表内有多少条行数据。



1. 查询双11那天有多少人购买了商品（要去掉同一个user\_id的重复购买）；

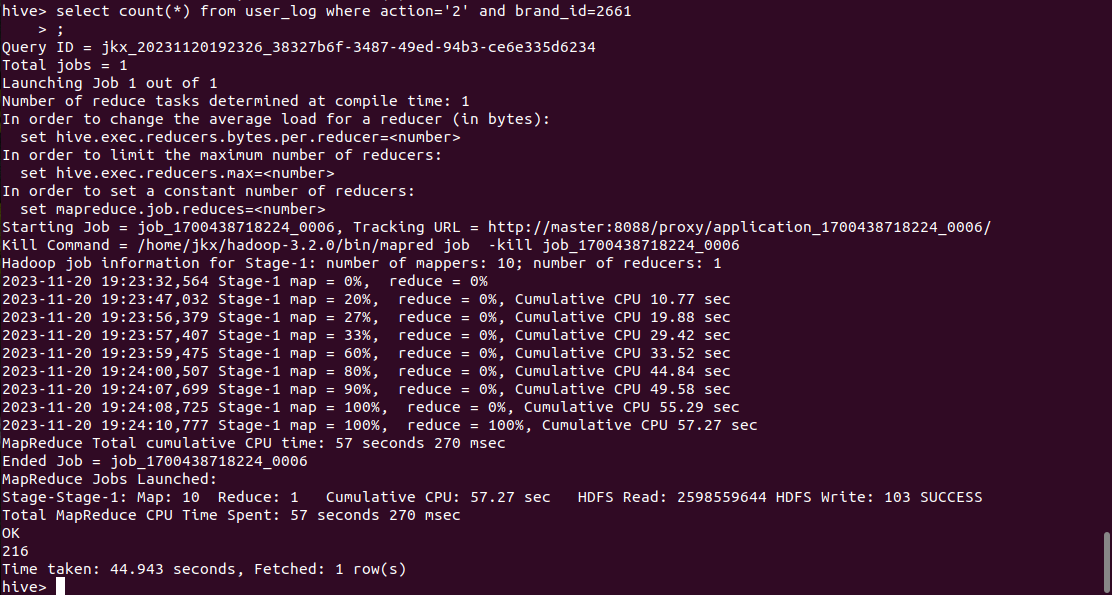
使用 `select count(distinct user\_id) from user\_log where action=’2’;` 查看双11那天有多少人购买了商品。

文本

描述已自动生成

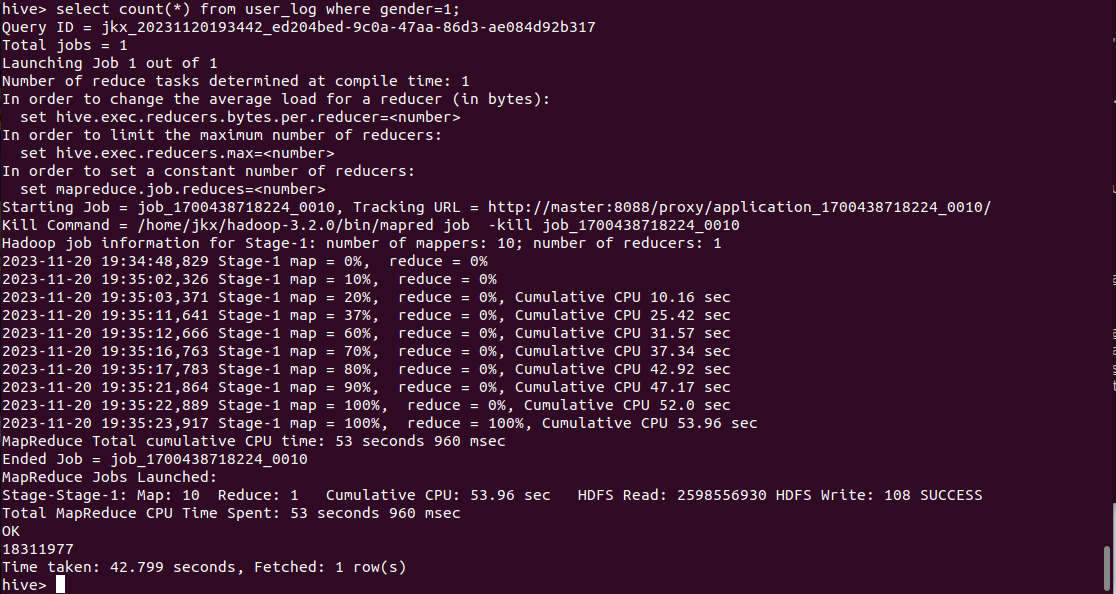
1. 给定品牌brand\_id为2661，求双11当天此品牌商品的销售数量；

使用 `select count(\*) from user\_log where action=’2’ and brand\_id=2661;`命令查看双11当天brand\_id为2661品牌商品的销售数量。



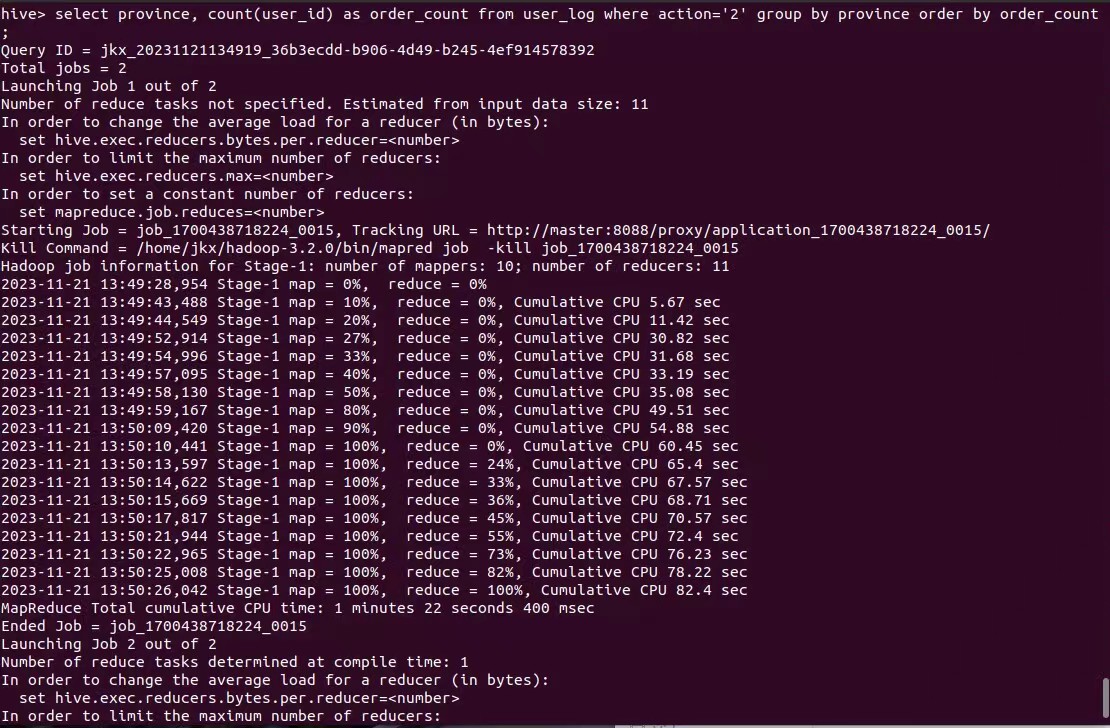
1. 查询双11那天女性购买商品的数量；

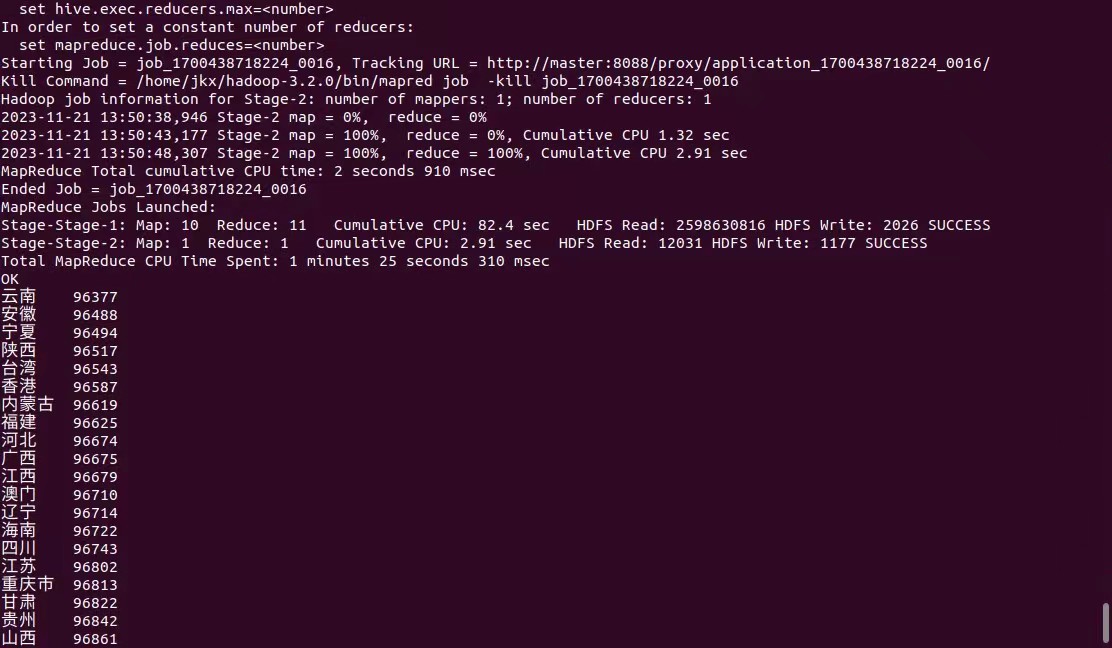
使用 `select count(\*) from user\_log where gender=1;` 命令查询双11那天女性购买商品的数量。

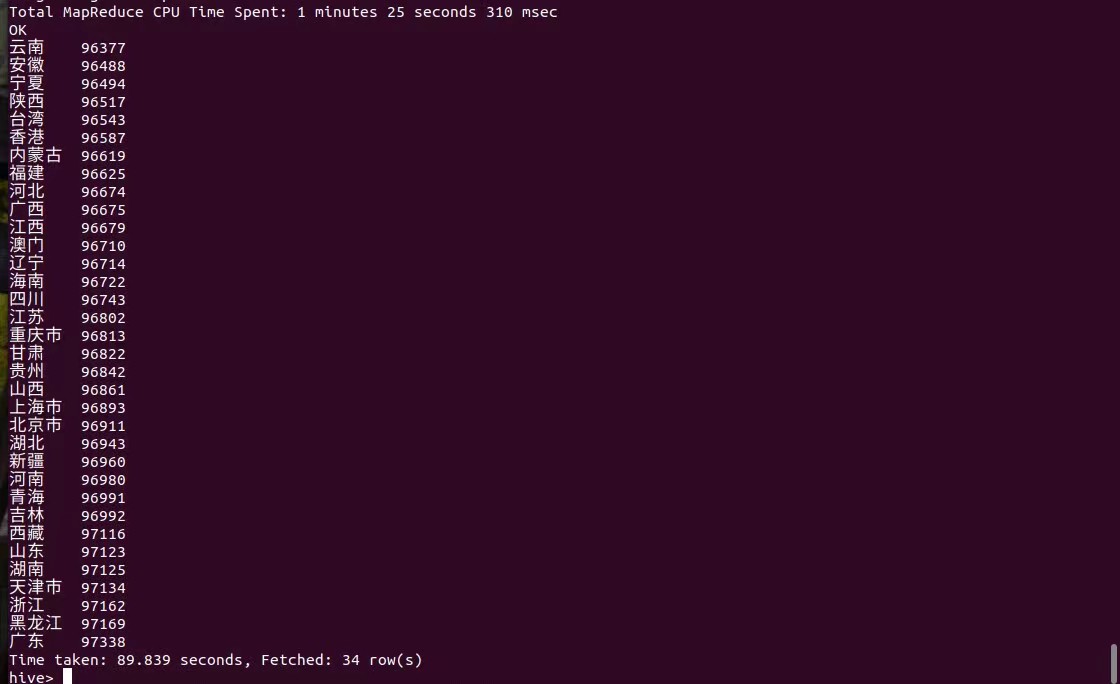


1. 根据收货地址，统计各个省双11那天用户下单的总次数，并进行排序。

使用 `select province, count(user\_id) as order\_count from user\_log where action=’2’ group by order\_count;` 命令统计各个省双11那天用户下单的总次数，并按照总次数升序排列。







6、 思考题：对比在Hive和MySQL中以上查询语句的执行时间，分析Hive和MySQL查询效率差异产生的原因。

在比较Hive和MySQL中相同查询语句的执行时间时，需要考虑它们的设计目标、架构、数据存储方式以及查询处理方式等因素。以下是一些可能导致查询效率差异的原因：

架构和设计目标：

Hive： Hive是一个基于Hadoop的数据仓库工具，设计用于处理大规模数据。它采用了类似MapReduce的批处理模型，适用于离线数据处理。Hive的目标是提供对大规模数据的高度容错性和可扩展性。

MySQL： MySQL是一个关系型数据库管理系统（RDBMS），设计用于在线事务处理（OLTP）和在线分析处理（OLAP）。它更适合处理较小规模的事务性数据。

数据存储方式：

Hive： Hive通常将数据存储在Hadoop分布式文件系统（HDFS）中，采用列式存储格式。这种存储方式适合大规模扫描操作，但可能导致随机访问的性能较差。

MySQL： MySQL使用行式存储，并且在传统文件系统上存储数据。这对于随机访问和小规模事务性操作更为高效。

查询处理方式：

Hive： Hive的查询处理通常通过将HiveQL查询转换为MapReduce作业来实现。这意味着在执行查询之前，可能需要启动一个MapReduce作业，这会引入一些开销。

MySQL： MySQL使用查询优化器来生成查询计划，尽量选择最有效的执行计划。通常，MySQL更适合处理较小规模的查询，因为它不需要启动像MapReduce这样的分布式计算框架。

索引和统计信息：

Hive： Hive在处理大规模数据时可能没有像MySQL那样灵活的索引支持。虽然可以使用分区和桶来提高性能，但并不像传统RDBMS那样具有细粒度的索引。

MySQL： MySQL支持细粒度的索引，这可以显著提高查询性能。优化器可以使用索引和统计信息来选择最优的执行计划。

数据规模：

Hive： Hive更适合处理大规模数据，因为它利用了Hadoop的分布式计算框架，适应于数据量较大的场景。

MySQL： MySQL更适合处理中小规模的数据，特别是在需要支持事务处理和实时查询的情况下。

在进行性能分析时，需要考虑具体的使用场景和需求。如果处理大规模数据，Hive可能更合适；而对于小规模数据和实时查询，MySQL可能更具优势。此外，通过调整配置、优化查询语句和使用适当的索引，可以改善两者的性能。