基于 Spark 与 Hive 的电商平台数据分析

许 晴 张桂花

四川大学锦城学院计算机与软件学院 四川 成都 611731

【摘 要】随着时代与计算机的发展，大数据席卷了全球，并为各大公司带来了惊人的收益。本次研究的目的是区分 Spark 与 Hive 的不同之处，便于选取合适的分析工具。本文针对某电商平台采集到的数据与提出的需求设计了两种方法，实现了电商平台对热门品类的统计与活跃的会话 ID 的统计。在实现方法的过程中对比 Spark 与 Hive 的具体实现方式，明确两种工具完成相同需求的不同之处。对比的结果证明 Spark 与 Hive 可以相互独立运行、Spark 实现需求的难度高于 Hive、Hive 环境搭建难度高于脱离 Hadoop 的 Spark。

【关键词】大数据；Hive; Spark; 电商平台; 数据分析

1 引言

大数据中蕴含的宝贵价值成为人们存储和处理大数据的驱动力 [1]“大数据”的关键是在种类繁多、数量庞大的数据中，快速获取信息。[2] 在获取信息的同时，不难注意到大数据的特点之一是数据类型繁多，结构各异。[3] 在大数据的时代背景之下，基于关系型数据库搭建的数据仓库系统，并不能够很好地满足时下的数据处理需求。[4] 为弥补查询方面的缺陷，目前提出了许多基于开源大数据处理平台的查询引擎，包括 Hive、Shark、SparkSQL 等。[5] 在用户使用电商平台的同时，平台会收集用户的访问数据，并且过滤出有意义的数据，根据数据来为用户进行精确推送。本文分别使用了 Hive 与 Spark 者两种查询引擎实现了对某电商平台具体数据的查询。

2 数据与需求分析

2.1 需求 1：TOP10 热门品类

通过数据中每个 Session 的操作来分品类统计其对应的点击、下单、支付的数量。再使用一定的计算公式来调整每个选项的权重，对每个品类分别进行计算。最后根据每个品类计算出的数值来排序，显示出前 10 热门品类。这个功能能够为我们生成 10 个用户间热门的品类。

2.2 需求 2：TOP10 热门品类中每个品类的 TOP10 活跃 Session 统计

获取 TOP10 品类中每个品类的前 10 活跃 SessionId。这个功能可以统计出某个用户群体最感兴趣的品类，各个品类最典型的用户的 SessionId。

2.3 数据分析

本次需要分析的数据文件类型为 txt，编码格式为常见的 UTF-8。文档中每一行数据代表某位用户在某时间进行了一次操作。一行数据中有十个字段，每个字段以下划线分隔开。每个字段代表的含义如图 1 所示。其中，点击、下单、支付可以再详细分解为 6 个字段，分别是：点击品类 id、点击品类 id、下单商品的品类 id、下单品类 id、支付商品的品类 id、支付商品 id。

图 1

若当前行的“搜索”字段的值不为″null″，则证明该用户本次行为是搜索；若“点击”字段不为“-1_-1”，则该用户本次行为是“点击”；用户的行为是“下单”时，可以下单多个单品；如果搜索关键字为 null，表示数据不是搜索数据。

3 Spark 实现

3.1 Spark 设计思路

本次分析主要会使用到 Spark 中的 RDD 与相关的算子。RDD 的转化与行动包含的算子能够实现题目的要求，将每行数据作为一个数据集，进行拆分、重组、聚合等操作，实现根据品类统计对应的数据的两个需求。

3.1.1 需求 1 设计思路

首先将电商网站后台得到的数据读取至 RDD 中，以便后续操作。

然后根据需求，将品类作为识别的每个数据的依据，分别统计每个品类的点击、下单、支付事件发生的数量，并生成对应事件的 RDD，其格式为（品类，点击数量），（品类，下单），（品类，支付）。将所有的统计结果合并，再次整理格式，将格式转换为（品类，（点击数量，下单，支付））。

根据预先设置的公式，对每个品类进行权值的计算。根据计算的结果对所有品类进行排序。因为 Spark

中排序方式默认为升序，所以排序的结果还需要进行反转，最后打印出反转后的前 10 位。

3.1.2 需求 2 设计思路

在实现需求 1 之后，可以得到按照权值排列的 TOP10 热门品类。在获得 TOP10 品类的基础之上重新读取文档，对文档中所有内容进行过滤，选取出所有 TOP10 品类相关的内容。完成数据的过滤后按照关键词和用户的 ID 再次进行统计。将所有的统计结果合并与整理，整理后每个算子具体的格式为（品类,(SessionID,count)）。然后以品类为主键，对统计的结果使用算子进行分组整合。

3.2 数据流分析

数据在 RDD 中整合的流程如图 2 所示。

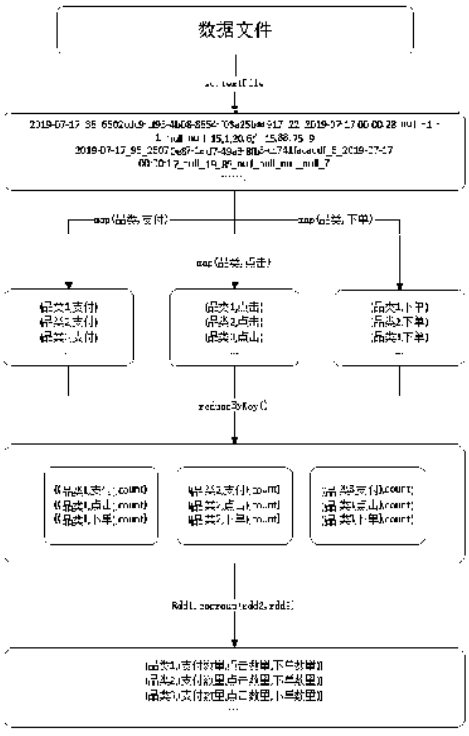

图 2

3.3 具体实现

3.3.1 实现代码

整合某品类 RDD 的代码（部分）：

```
val clickRDD = actionRDD.filter(actions => {val datas = actions.split('_') datas(6)!="-1"})
```

整合、排序、输出代码：

```
val chmodRDD = allRDD.mapValues(
x=>{for(i <- x._1)
 x1 +=i
for(i <-x._2)
 x2 += i
for(i <-x._3)
 x3 += i
x1*0.2+x2*0.3+x3*0.5}).sortBy(_._2,false).take(10).foreach(println)
```

过滤代码：

```
val filteredRDD=changeRDD.filter(action =>id.contains(action._1._1))
```

3.3.2 结果展示

Spark 实现需求 1 需求 2 的结果如图 3 所示。其中，如 (12,1860) 的键值对分别代表品类、聚合计算后该品类对应的权值；如 (20,List((199f8e1d…,8)…) 分别代表品类 id 与排名前十的 sessionId、点击次数。

图 3

4 Hive 实现

4.1 关于 Hive

Hive 具有延展性，支持用户自定义函数，可以根据实际的需求来实现自己的操作。因为 Hadoop 是面向批处理的系统，而 MapReduce 任务的启动过程需要消耗较长的时间，所以基于 Hadoop 的 Hive 延时严重，不能用于实时的数据处理。

4.2 Hive 设计思路

在 Hive 中生成新的数据库，并在此数据库中建表，表的列名与数据字段相对应，列之间的分隔符号为'_'。根据之前的数据分析，‘下单商品的品类’与‘支付商品的品类’字段中的包含多个品类，所以这两个字段的类型应该定义为 Hive 中的复杂类型。此时建立的表为空表，没有详细的数据内容，只有描述数据的字段，即该表的元数据。完成建表后，将位于本地的数据文件通过 Hive 语句读取至表中，此时就可以在 Hive 中得到包含数据内容的完整数据表。

完成表的创建与数据导入后，就可以使用 HQL 语句，根据需求对表中的数据进行查找。因为数据表中需要查询的下单品类 id 与支付品类 id 都已经定义为 array，如果直接对 array 进行统计的话只能的到 array 中第一个值的相关数据，所以需要将这两个列中的行全部转换为新列再进行统计。以品类字段进行分组，对点击、下单、支付的行数统计求和，完成求和之后再对计算出的值进行权值计算。计算后选取前 10 行，即可得到需求 1 的结果。

需求 2 的实现建立在需求 1 的基础之上。首先在点击列中过滤出排名前 10 品类的数据，然后按照 Session 字段的值来进行统计，计算每一个品类中每个 Session 进行了多少次点击操作，排序并返回具体的 Session。

4.3 具体实现

4.3.1 实现代码

需求 1 实现代码（部分）：

将复杂数据类型转换为独立的行：

```
select explode(split(order_category_ids,',' ))as category_id
```

需求 2（部分）：

```
select category_id,session_id from sessions_log where rown <=10
```

4.3.2 结果展示

Hive 实现需求的结果如图：

图 4

图中前十行内容为热门品类与该品类的具体权值，十行以后的数据为热门品类与其活跃的 sessionid。

5 Spark 实现与 Hive 实现的对比

在本次某电商平台数据分析项目中主要使用了两种实现方式，分别是 Spark 实现和 Hive 实现。当离线数据以本地文件存储在本地时，这种情况下 Spark 可以做到脱离 Hadoop 运行；而 Hive 依赖于 Hadoop 提供的 MapReduce，在使用 Hive 之前必须搭建 Hadoop 环境，所以 Hive 环境搭建相比起 Spark 环境搭建更加复杂。而在完成实现需求的具体代码时，Spark 对用户的要求比 Hive 对用户的要求更高，因为使用 Spark 需要用户掌握具体的算子与 Scala 语言，使用 Hive 只需要用户知道如何根据字段内容自定义数据表、加载数据至表中、如何将复杂类型的数据转换为行等操作，具体的查询语句与 SQL 没有区别；在代码执行速度方面，Hive 的数据处理依赖于 Hadoop 的 MapReduce，需要将 HQL 语句转换为对应的 SQL 语句再通过 MapReduce 执行，MapReduce 又取决于计算机的性能，所以 Spark 的运行速度远高于 Hive 的运行速度。

表 1 Spark 与 Hive 性能比较

	Spark	Hive
运行速度	快	慢
是否依赖 Hadoop	可以脱离 Hadoop 运行	必须在有 Hadoop 的环境下运行
对用户的要求	掌握 Spark 的算子与 Scala 语言	掌握 Hive 中一些复杂结构与操作，查询语句与普通 SQL 相同

结语

Spark 与 Hive 都是大数据数据处理中的重要工具。Spark 是一个分布式计算框架，Hive 是一个分布式数据仓库，两者可以结合在一起使用，在某些特殊情况下也可以单个独立使用。本文通过实现分析某电商网站数据的两个需求，对比了 Spark 与 Hive 在数据已经存在于本地时代码的运行速度、对环境的依赖程度、对用户的编程能力的不同。

本研究的意义在于放大 Spark 与 Hive 的区别，为准备使用 Spark 与 Hive 分析数据的用户提供参考，使用户能够在分析离线数据时根据自己所使用的计算机与其编译环境来选择使用更加高效的完成、实现离线数据的分析的数据处理工具。

【参考文献】

[1] 程学旗，靳小龙，王元卓，郭嘉丰，张铁赢，李国杰．大数据系统和分析技术综述 [J]．软件学报，2014,25(09):1889-1908.DOI:10.13328/j.cnki.jos.004674.

[2] 刘智慧，张泉灵．大数据技术研究综述 [J]．浙江大学学报（工学版），2014,48(06):957-972.

[3] 李广建，化柏林．大数据分析与情报分析关系辨析 [J]．中国图书馆学报，2014,40(05):14-22.DOI:10.13530/j.cnki.jlis.140020.

[4] 姜吉宁．基于 spark 和 hive 的新型种质资源数据仓库的设计和实现 [D]．中国科学技术大学，2018:

[5] 张玉杰，于双元．大数据查询综述 [J]．计算机与现代化，2017,(04):82-88.