基于 SparkStreaming 在广告中的研究与应用

张 春 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】随着电商的兴起，许多商户会在网站中投放广告。这些广告在被点击时会产生利益，因此，带来“一些用户或者竞争对手恶意点击广告”的问题，所以要屏蔽这些用户。本文是基于 Spark Streaming 在广告中的研究与应用，意在解决上述问题，将恶意点击的用户加入黑名单。把黑名单存放在 Redis 中，并统计近一个小时广告的点击量，用 Echarts 动态展示。模拟广告点击日志，获取实时数据并写入 Kafka，由 SparkStreaming 从 Kafka 中消费。

【关键词】SparkStreaming; Kafka; Redis; Echarts; 广告

1 引言

随着各种电子商务、搜索引擎等网站的增加，“流式处理”已经成为了热门话题，是国内外研究的对象。流式处理现代数据的主要方案，比如：线上商城需要通过用户在网页上的点击和购买的物品来推断出用户的需要、和用户的兴趣爱好。然而，后台能否进行实时计算就显得尤为重要，向用户推荐最新的商品，最新的信息，向用户提供更好的服务。SparkStreaming 是 SparCore 上面的一个应用程序[3]。Spark Streaming 中的数据是不断的流进来，然后流进来的数据将会不断的生成对应的 Job，不断的提交到 HDFS 集群去处理。

然而，广告的计费系统则是电商必不可缺少的一个功能点。但是，甲、乙两个商户同时在某电商网站上做广告，甲和乙互为竞争对手，如果甲使用网站点击机器人对乙的广告进行恶意的点击，那么乙的广告费很快就用完为了，为了防止恶意的广告点击，必须对广告点击进行黑名单的过滤。将恶意点击的用户的 id 存入黑名单中，将其保存在 Redis 中，利用 SparkStreaming 的流处理特性，消费 kafka 中的数据，从 redis 数据库中保存的黑名单中获取黑名单数据，实时的动态展现近一小时广告的点击量，用于过滤掉哪些黑名单中用户的数据，实现实时黑名单的过滤实现。

2 关键技术

2.1SparkStreaming

SparkStreaming 用于处理流式数据。SparkStreaming 支持许多数据输入源，如：Kafka、Flume 和 TCP 套接字等。数据输入后使用 Spark 的算子如：map、reduce 和 join 等进行运算。其运算结果也可以保存在许多地方，如 Redis 数据库，HDFS 等。Spark Streaming 与 Spark 基于 RDD 的概念非常相似，它使用离散化流（DStream）作为抽象表示。[5] SparkStreaming 具有较高的容错性，比如：实时的流式处理系统是 7*24 运行的，还能够从各种系统错误中快速恢复过来，支持 worker 节点与 driver 节点的错误恢复。

2.2 Kafka

Kafka 是由 Linkedin 公司开发的一个分布式、多副本的、多订阅者的消息队列，它既支持离线的数据处理又实时的数据处理。它常用于访问日志、web 日志和消息服务等，具有很高的吞吐率，在廉价的、低性能的商用机器也能达到每秒钟 100K 条消息的传输速率。Kafka 不仅支持 Kafka Server 之间的分布式消费和消息分区，还保证了每个分区内的消息能够顺序传输。[1]

2.3 Redis

Redis 是一个开源的 key-value 存储系统，它将数据缓存在内存中，极大的提高了数据访问的效率。Redis 可以周期性的将更新的数据写入磁盘或者将修改操作写入追加的记录文件中，在此基础上实现主从的同步[6]。

2.4 Echarts

ECharts 是一个基于 JavaScript 语言开发的可视化库，它包含了各行各业的图表，极大的满足了人们的各种需求。ECharts 较好地遵守了 Apache 公司的开源协议，它兼容了目前绝大部分的浏览器，比如：Firefox，IE 8/9/10/11，Chrome，Safari 等，也兼容多种设备，可随时随地展示。

3 设计与实现

3.1 系统架构

本项目是基于 SparkStreaming 在广告中的研究与应用，主要有两个功能：统计广告黑名单，统计近一小时的广告点击量，其系统架构如下图 1：

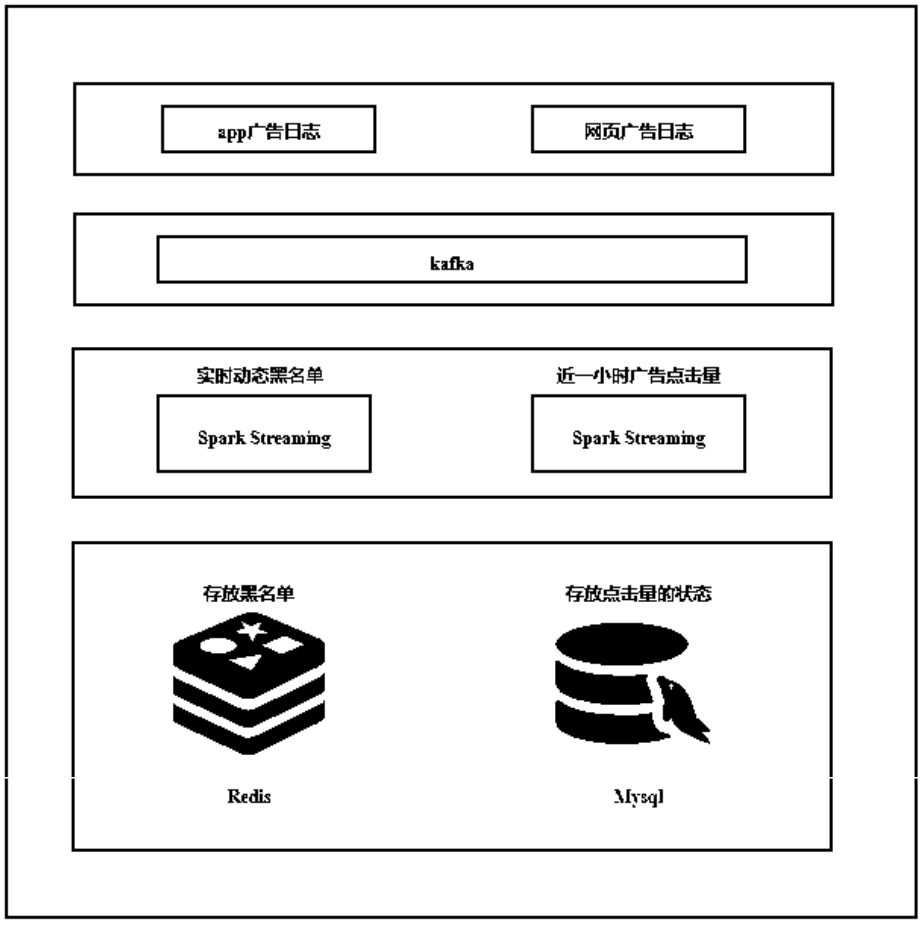

图 1 系统架构

Fig.1 System Structure

3.2 需求分析

3.2.1 广告黑名单

实现实时的动态黑名单机制：将每天对某个广告点击超过 10 次的用户拉黑，将其保存在 Redis 中。Spark Streaming 消费 kafka 中的数据，从 redis 保存的黑名单中获取黑名单数据，用于过滤掉哪些黑名单中用户的数据，按照用户每天对每个广告进行聚合统计，然后判断当前的聚合结果是否有超过阈值，若超过阈值，则加入 redis 中的黑名单，因为可能会存在机器恶意点击。Spark Streaming 实时统计，所以要统计出每天的点击量所以需要保存之前的状态，该需求中状态保存在 MySQL 中，所以当局和完成和我们要把本次聚合的数据与 MySQL 中之前的数据进行更新操作即 MySQL 中存在之前的值，则累加。不存在，则插入。更新操作后还需查询 Mysql 是否有超过阈值的，若超过那么则要保存到 redis 黑名单中。

3.2.2 近一小时广告点击量

实时的动态展现近一小时广告的点击量：Spark Streaming 实时的消费 kafka 中的数据，首先也要对该数据进行过滤，将属于 redis 黑名单中用户的数据过滤掉，若要统计近一小时数据的点击量，测试的时间成本较高，所以改为统计近一分钟的广告点击量，两则的思路都相同，首先对数据进行格式转换，将数据的时间戳转化为整十秒的时间戳，例如：12:01 转化为 12:00、12:17 转化为 12:10，转化为（时间戳，1）的二元组类型，reduceByKeyAndWindow 开窗聚合统计，将统计结果转化为 json 数组格式，导入到指定的 json 文件中，最后 Echarts 苏区 json 文件，将点击量实时动态的展示再页面上。

3.3 功能设计

3.3.1 广告黑名单

Spark Streaming 实时消费 kafka 中用户点击广告的日志，部分代码如下：

```
    val ad_logs: InputDStream[Consume
rRecord[String, String]] = KafkaUtils.
createDirectStream[String, String](
  ssc,
  LocationStrategies.PreferConsistent,
  ConsumerStrategies.Subscribe[String, String]
(Set(“ad_logs”), kafkaPara)
)
```

Map 类型转换：每一条广告日志转化成自定义的样

例类(ADLog)，方便后续操作数据，样例类包含时间戳、用户 id、用户 IP、广告 ID、用户性别

Filter 数据过滤：首先通过自定义 Jedis 连接池工具，读取存放在 redis 中的用户黑名单（数据类型 set，key 为：ad_black_list），根据黑名单的数据过滤掉恶意点击广告的用户数据，部分代码如下：

```
val balckList = JedisUtil.getBlackList()
val filter_rdd: RDD[ADLog] = rdd.filter(adlog => {
    !balckList.contains(adlog.user_id)
})
```

Map 类型转换：该用户的是否加入黑名单的依据，若用户每天对某一个广告的点击量超过了 10 次（该次数可根据公司的数据规模以及用户活跃度进行调整），则加入黑名单中使用 map 操作转换数据类型为((日期，用户 ID，广告 ID),1)，方便后续进行聚合统计

ReduceByKey 聚合操作：统计每个用户每天对某个广告的点击次数，得到结果为((日期，用户 ID，广告 ID),点击次数)

判断批次的统计结果是否超过阈值：Spark Streaming 可以看成一个批处理的框架（微批），即攒一定时间的数据再进行处理，时间通常为几秒或毫秒，但可能出现使用机器进行恶意点击广告，则在短时间内会超过阈值，将超过阈值的用户加入到 redis 黑名单 ad_black_list 中，部分代码如下：

```
case ((user_id, ad_id, date), count) => {
if(count>=10){
    JedisUtil.addBlackList(user_id)
}
```

Spark Streaming 是一个实时处理的框架，所以每次只能处理简短几秒或者毫秒的数据，但是我们需要根据用户每天对广告的点击次数，来判断当前用户是否有恶意点击广告的嫌疑，因此需要保存之前用户每天点击广告的状态，我们这里采用 MySQL 表 ad_count_status 保存状态，ad_count_status 表的字段有：日期，用户 ID，广告 ID，点击次数，我们把本批次 Spark Streaming 中处理的数据在 ad_count_status 表中进行更新操作，若该用户在本次之前有点击同一个广告的行为，那么就将本次统计到的次数与之前 ad_count_status 表中的次数做一个累加，否则插入本次统计到的次数，部分代码如下

```
val sql =
"""
  |insert into ad_count_status (date,user_id,ad_id,count) values (?,?,?,?)
  |on DUPLICATE KEY
  |UPDATE count = count + ?
  |""".stripMargin
conn.setAutoCommit(false)
val pst: PreparedStatement = conn.prepareStatement(sql)
pst.setString(1,date)
pst.setString(2,user_id)
pst.setString(3,ad_id)
pst.setInt(4,count)
pst.setInt(5,count)
```

更新完 Mysql 中的 ad_count_status 表后，需要判断当前用户表中点击次数是否超过阈值，因为 ad_count_status 可能存在，没有更新之前点击次数没有超过阈值，累计跟新后超出阈值，所以采用 sql 查询，where 条件 user_id= 当前更新次数的用户、date= 用户点击日期，ad_id= 用户广告、count>10，若查询出来的值为‘空’则表示当前用户没有超出阈值，若不为空则表示当前用户已超出阈值，则使用 Jedis 连接池操作 redis，将该用户加入黑名单 ad_black_list 中，部分代码如下：

```
val flag = JDBCUtil.get_grt_count(date, user_id, ad_id)
if(flag){
    JedisUtil.addBlackList(user_id)
}
```

3.3.2 近一小时广告点击量

因为上述黑名单的功能已经将原始的数据进行了类型转换，所以我们直接使用转换为 ADLog 样例类类型的数据进行后续的统计

Filter 过滤操作：需要对数据进行过滤，把黑名单中用户的数据过滤掉，这样保证其结果的准确性

Map 类型的转换：该功能是统计近一时广告的点击量，但为了方便测试，所以在代码中写成的是近一分钟广告的点击量（后续的操作都是统计近一分钟的广告点击量），在近一分钟里我们需要以十秒为一个间隔进行统计，以 12:00-13:00（12 分到 13 分）为例 12:00-12:10 的数据都归为是 12:00 的数据，12:10-12:20 之间的锁具都归为是 12:10 的数据 以此类推，所以我们需要把数据的时间戳进行处理 例如：12:03 的要转化为12:00,采用的公式 newts =(timestampe/(10*1000))*(10*1000)（timestampe：用户的时间戳，单位毫秒，newts：得到就是整十秒的时间戳）最后得到的数据类型为（newts ，1）部分代码如下：

```
val map_rdd: DStream[(Long, Int)] = filter_rdd.map(ad_log => {
    val timestamp = ad_log.timestamp
    val newTS = timestamp / 10000 * 10000
    (newTS, 1)
})
```

reduceByKeyAndWindow 开窗聚合统计：因为要统计近一分钟得数据，那么我们得保证能够得到近一分钟的数据，所以我们的就需要开窗操作，reduceByKeyAndWindow 需要三个参数：参数一 逻辑处理，该需求则是统计点击的次数那么只需要 t1+t2 即可；参数二 窗口的大小就为 1 分钟（根据业务需求进行改

动，若统计近一小时的广告点击量，那么窗口的大小则设置为 1 小时且该参数需要是 Spark Streaming 设置批处理大小的整数倍），参数三 滑动间隔 执行窗口操作的时间，以及每次滑动的大小（该参数需要是 Spark Streaming 设置批处理大小的整数倍）本需求中是按照 10 秒位一个间隔划分的，那么该参数设置为 10 秒，可根据自身业务灵活变动，部分代码如下：

```
val res: DStream[(Long, Int)] = map_rdd.
reduceByKeyAndWindow((c1: Int, c2: Int) => {
        c 1 + c 2
}, Seconds(60), Seconds(10))
```

将聚合后的结果转换为 json 数组类型并且写入到 adClick.json 文件中，部分代码如下：

```
case ( time, cnt ) => {
    val timeString = new SimpleDateFormat("mm:ss").format(new
        java.util.Date(time.toLong))
    list.append(s""" {"xtime":"${timeString}", "yval":"${cnt}"}""")
}
}
val out = new PrintWriter(new FileWriter(new File("D:\\CC_Project\\SparkStreaming_AD\\adBI\\adclick.json")))
out.println("[ "+list.mkString(",")+"]")
```

Echarts 从 adClick.json 文件中实时动态的显示近一分钟广告（这里为了方便测试选取时间为一分钟）点击量，如下图 2：

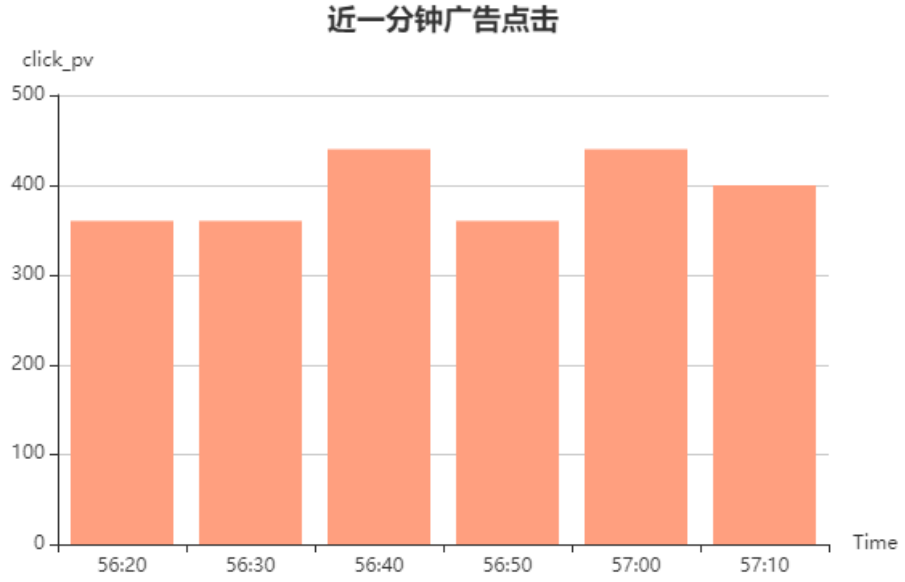

图 2 近一分钟广告点击量

Fig.2 Ad clicks in the last minute

总结

本文通过对 Spark Streaming、Kafka、Redis、Mysql 的整合，实现了实时处理广告日志，让我了解到大数据实时处理的魅力所在。随着 5G 时代的来临，数据量的剧增，对实时性更高的要求，这些都将给大数据行业带来新的机遇和挑战。本文目的屏蔽恶意点击广告的行为，可减少广告投放者的经济损失，为其带来最大的广告效益，同时营造一个良好的网络氛围；实时动态的显示近一小时广告的点击量，为广告投放者广告投放时间点提供了有力的数据支持。

【参考文献】

[1] 一种基于 Kafka 的可靠的 Consumer 的设计方案 [J]. 王岩，王纯． 软件． 2016(01).

[2] 陈虹君．基于 Hadoop 平台的 Spark 框架研究 [J]. 电脑知识与技术，2014（35）：84078408.

[3] 林子雨．Spark 编程基础（Scala 版）[M]. 北京：人民邮电出版社，2018.

[4] 肖力涛 ．Spark Streaming 实时流式大数据处理实战 [M]. 北京：机械工业出版社，2019

[5] 黄健宏．Redis 使用手册 [M]. 北京：机械工业出版社，2019