Computer System Networking and Telecommunications, 计算机系统网络和电信 (2)2020,2
ISSN:2661-3719(Print)

基于 Spark 和 kafka 的视频网站用户需求大数据应用

胡云彰　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】视频网站流量的大量喷涌促进了资本的注入，也给投资者、运营者对于数据的强烈需求。怎么获取用户在视频网站上的行为点击就成了重中之重。Spark 是当前大数据领域优秀的计算框架，其中 Spark Streaming 组件是基于 Spark 的实时流处理技术，采用一系列短暂、无状态、确定性的批处理技术实现。kafka 是一个分布式平台，用于发布和订阅记录流且以容错方式永久粗存储记录流，本文将 Spark 和 Kafka 结合用于研究视频网站的用户需求，包括电脑的鼠标点击以及手机端的滑动交互。从而实现对视频网站流量的精准分析，通过可视化方式进行展示，为决策层提供数据保障。
【关键词】大数据；Spark；Kafka；Java Web；Spark Streaming；Python；Springboot；Flume

互联网视频用户数逐年递增，每年将产生大量的数据，商人们将自己的眼光对准了这些数据，亟待挖掘、理解和应用。用 Spark 等手段将这些数据做一个大数据分析，将得到的数据分析反馈给企业，从而找出企业在营销、推广、服务等方面的问题，再有管理层做出决策后，改变企业经营策略，进而提高企业收益。

1 技术选型

1.1 Scala 和 Spark

Scala 是一种计算机语言。Scala 提供了与 Java 的语言互操作性，因此可以直接在 Scala 或 Java 代码中引用以任何一种语言编写的库。与 Java 一样，Scala 也是面向对象的，并使用了使人联想起 C 编程语言的大括号语法。与 Java 不同，Scala 具有功能编程语言的许多功能，例如 Scheme，Standard ML 和 Haskell，包括 curring，不变性，惰性评估和模式匹配。相反，Scala 中不存在的 Java 功能就是检查异常，这已引起争议。

Apache Spark 是一个开放源代码集群计算机框架，最初是在加州大学伯克利分校 AMPLab 开发的。与 Hadoop 的基于磁盘的两阶段 MapReduce 相比，Spark 为具有内存原语的一些应用程序提供了高达 100 倍的性能提升。其内置模块如图 1 所示。

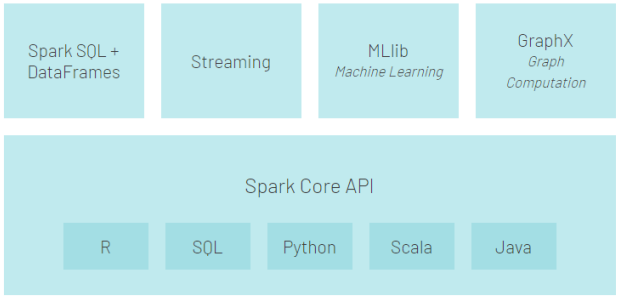

图 1 Spark 内置模块

Spark 的相关特点：快速、简洁容易使用、通用、多种运行模式。Spark 的框架分为 4 大模块：Spark SQL-RDD（数据执行的基本单元），MLlib（机器学习），Graphx（图计算）和 SparkStreaming（实时处理）[1]。Apache Spark 需要集群管理器和分布式存储系统。对于集群管理来说，Spark 支持独立的（Hadoop YARN）（本地 Spark 集群，可以在其中手动启动集群，也可以使用安装包提供的启动脚本，也可以在一台机器上运行这些守护程序进行测试）。Spark 可以整合各种各样的接口，包括 Alluxio，Hadoop 分布式文件系统（HDFS），MapR 文件系统（MapR-FS），Lustre 文件系统。Spark 还支持伪分布式本地模式，通常仅用于开发或测试目的，不需要分布式存储，而可以使用本地文件系统。在这种情况下，Spark 运行在单个计算机上，每个 CPU 内核有一个执行程序。

Spark Streaming 是 Spark 生态系统的重要组成部分，主要用于实时数据流的处理。其工作原理是将流式计算分解成一系列短小的批处理作业，本质上也是数据的批量处理，但却将时间跨度控制在数十毫秒到数秒之间。

1.2 Kafka

Kafka 是一个分布式的基于发布 / 订阅的消息队列（Message Queue），具有解耦，可恢复，缓冲的特点，并且灵活性和峰值处理能力都非常高 [2]，其发布 / 订阅模式如图 2 所示。

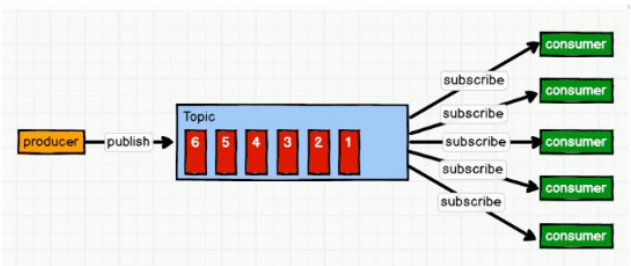

图 2 Kafka 发布 / 订阅模式

主要应用于大数据实时处理领域。随着通信行业的普及，以及人们对网络的需求越来越大，因此运营商的一些在线服务需求也来越大。虽然目前线下渠道占据主要地位，线上渠道作为业务办理的支撑和辅助。运营商主打便捷性卖点，向客户宣传推广电子渠道。对于客户体验来说，电子渠道提供了一个足不出户办理业务的便捷方式，对于运营商来说，电子渠道低成本分流了实体渠道的业务压力，将线下渠道的人力资源从低价值的业务办理中释放出来。对服务商给出的历史数据进行分析，以图形化的方式直观展现各服务区域服务指标达标情况，以及展现各服务区域的累积服务量。通过数据分析，对各类用户做多维画像，降低用户的投诉和流失。

1.3 Python

Python 是一种可用于爬取网页数据的计算机程序设计语言，是一种解释型的面向对象编程语言。易于阅读、使用和编写。本文将重点介绍 Python 爬虫程序，其目的是用于抓取网站上的数据信息资源按照不同的功能可分为：通用网络爬虫、垂直网络爬虫、增量网络爬虫、深层网络爬虫。

1.4 Flume

Apache Flume 是一种工具 / 服务 / 数据提取机制，用于收集各种流数据（例如日志文件，事件）的聚合并且将其从各种来源传输到集中式数据存储。

Flume 是一种高度可靠，分布式和可配置的工具。它主要用于将流数据（日志数据）从各种 Web 服务器复制到 HDFS。

Flume 是可靠的，可容错的，可扩展的，可管理的以及可定制的。

1.5 Spring Boot

Spring Boot 是由 Pivotal 公司维护的开源框架。它为 Java 开发人员提供了一个平台，可以开始使用可自动配置的生产级 Spring 应用程序。开发人员可以借由此快速入门，而不会浪费时间准备和配置自己的 Spring 应用程序。

1.6 Hbase

HBase 群集由很少的主服务器和很多 Region Server 组成。HBase 在 Hadoop（主要用于存储数据的 HDFS）和 Zookeeper 之上运行。而 Zookeeper 集群用于 HBase 节点的故障检测，并存储 HBase 集群的分布式配置。具有以下特性：①线性和模块化可扩展性；②严格一致的读写；③表的自动和可配置分片。

2 设计与实现

2.1 数据采集与预处理

使用 Python 脚本实时产生数据

```
import random
import time
url_paths = [
    “ www/2 ”,
```

直接分析日志以爱奇艺为例，www/2—代表电视剧，www/1—代表电影

```
def generate_log(count=10):
    while count >=1:
        query_log = “ {ip} ” .format(ip= ” 192.168.187.1 ” )
        print query_log
        count = count -1
url paths = [

import random
ip_slices=[132,156,124,10,29,167,143,187,30,100]
# 生成 i 地址
def sample_ip():
slice = random.sample(ip_slices,4)
return “ . ” .join

# 生成日志
def generate_log(count=10):
    while count >=1:
        query_log = “ {ip} ” .format(ip=sample_ip())
        print query_log
        count = count-1;
```

抓取到网页地 url 转换成信息，添加时间

```
Import time
Time_str=time.strftime(“ %Y-%m-%d %H:%M:%S ”,time.localtime())
{localtime}
```

然后把日志写入到文件，并通过调度器工具将每批数据按每分钟产生，然后自定义时间，

最后生成 sh 文件并上传到在虚拟机上启动 Flume。

```
bin/flume-ng agent –conf conf –conf-file conf/a1.conf –name a1
```

Flume 的配置文件：

```
# 定义 agent
a1.sources = src1
a1.channels = ch1
a1.sinks = k1
# 定义 sources
a1.sources.src1.type = exec
a1.sources.src1.command = tail-F /home/log
a1.source.src1.channels = ch1
# 定义 sink
a1.sinks.k1.type = org.apache.flume.sink.kafka.KafkaSink
a1.sinks.k1.topic = flumeTopic
```

```
a1.sinks.k1.brokerList = s201:9092
a1.sinks.k1.batchSize = 20
a1.sinks.k1.requiredAcks = 1
a1.sinks.k1.channel = ch1
# 定义 channels
a1.channels.ch1.type = memory
a1.channels.ch1.capacity = 100
```

在使用 kafka 之前必须先配置 zookeeper，安装之后通过 /zookeeper/bin/zkServer.sh 来启动。

然后启动在三台虚拟机上同时启动 kafka，命令如下 :

```
cd /usr/local/kafka
bin.kafka-server-start.sh config/server.properties
```

创建一个 topic 并创建一个消费者，然后启动 kafka consumer:

```
Bin/kafka-console-consumer.sh –zookeeper s201:2181 –topic flumeTopic –from-beginning
```

Python 里面生成的日志由 flume 收集，最终进入 kafka 里，kafka 作为一个消费者直接进行消费，再通过代码将其与 spark 对接。

2.2 数据分析

2.2.1 将 Flume 和 Kakfa 获取的数据传输给 SparkStreaming 进行处理

在 intellij idea 中创建一个 maven 工程，在 pom.xml 中导入所需的包，因为要引用 kafka，所以需要添加依赖包，包括 kafka、spark、hadoop、hbase、spark-streaming，配置 sparkstreaming，在虚拟机中运行 sh 文件并成功将数据传入到 idea：

```
import org.apache.kafka.clients.consumer.ConsumerRecord
import org.apache.kafka.common.serialization.StringDeserializer
import org.apache.spark.streaming.kafka010._
import org.apache.spark.streaming.kafka010.LocationStrategies.PreferConsistent
import org.apache.spark.streaming.kafka010.ConsumerStrategies.Subscribe

val kafkaParams = Map[String, Object](
 “ bootstrap.servers ”->“ localhost:9092,anotherhost:9092 ”,
  “ key.deserializer ” -> classOf[StringDeserializer],
  “ value.deserializer ” -> classOf[StringDeserializer],
  “ group.id ” -> “ use_a_separate_group_id_for_each_stream ”,
  “ auto.offset.reset ” -> “ latest ”,
  “ enable.auto.commit ” -> (false: java.lang.Boolean)
)

val topics = Array(“ topicA ”, “ topicB ”)
val stream = KafkaUtils.createDirectStream[String, String](
  streamingContext,
  PreferConsistent,
  Subscribe[String, String](topics, kafkaParams)
)

stream.map(record => (record.key, record.value))
```

视频网站上的点击量数据通过 sparkstreaming 计算出来了，获得点击日期，点击量，分批点击量。

例如：20200501 3 10.

20200502 5 20

获得的这些数据需要保存在数据库中

将 pom.xml 中的 mainclass 替换成 jar 包所在类

```
<mainClass>com.study.spark.project.StatSreamingApp</mainClass>
```

将 maven 项目进行打包，run building

脚本的运行命令 :

获取其中的命令：/soft/spark/bin/spark-submit \

```
--class com.study.spark.project.StatStreamingApp \
```

它里面的 jar 包：/home/centos/SparkTrain-1.0-jar-with-dependencies.jar \

2.2.2Hbase 数据存储

充分利用 Hbase 的高速加载技术以及基于 Hbase 的时序数据分析挖掘技术，写一个操作 Hbase 的工具类。HBase 作为 Bigtable[181 在 Hadoop 体系中的开源实现，为海量数据提供了面向列簇的存储支撑以及良好的在线应用支撑 [3]。

在 project 中创建一个 HbaseUtils 类，

```
public class HbaseUtils {
    HbaseAdmin admin = null;
    Configuration configuration = null
// 给一个私有的构造方法
private HbaseUtils(){
    configration = new Configuration();
    configration.set(“ hbase.zookeeper.quorum ”, ” h1m1 ” );
    configration.set(“ hbase.rootdir ”, ” hdfs://h1m1/habse ” );
    try{
        admin = new HbaseAdmin(configration);
    }catch(IOException e){
        e.printStackTrace();
```

```
        }
    }
    Public void put(String tableName,String rowkey,String cf,String column,String value){
        Htable table = getTable(tableName);
        Put put = new Put(Bytes.toBytes(rowkey));
        Put.add(Bytes.toBytes(cf),Bytes.toBytes(column),Bytes.toBytes(value));
                Try{
                        Table.put(put);
                } catch (IOException e){
                        e.printStackTrace();
                }
    }
```

添加数据到 Hbase 里面，

TableNme 表名

rowKey 对应 key 的值

cf hbase 列簇

colum hbase 对应的列

value hbase 对应的值

```
    public void put(String tableName,String rowKey,String cf,String colum,String value) {
        Htable table = getHtable(tableName);
        Put put = new Put(Bytes.toBytes(rowKey));
        Put.addImmutable()
        Table.put();
    }
```

测试数据

```
    Public static void main(String[] args) {
        String tableName = “ category_clickcount ” ;
        String rowKey= ” 20200516-1 ”
        String cf= ” info ”
        String colum = “ cagegory_click_count ” ;
        String value = “ 100 ” ;

        HbaseUtils.put(tableName);
```

2.3 数据可视化

2.3.1 可视化工具

可视化可以提高效率，如果没有可视化，我们的决策自然而然会降低，就会导致金钱和时间的不必要损失，甚至危及整个公司的利益，接下来创建可视化。

Springboot 构建 web 项目：在 idea 中新建一个 Spring Initializr 工程，创建一个 HelloBoot 并给他加一个注解

```
    @RestController
    Public class HelloWeb {
        @RequestMapping(value = “ hello ” ,method = RequestMethod.GET))
        Public String hello(){
                Return “ hellospoot ” ;
        }
    }
```

然后再配置文件 resources 里面将 application.properties 修改如下 :

```
        server.context-path=sparkweb
        server.port=9999
```

ECharts，这是一个开源的，基于 Web 的跨平台框架，它支持交互式可视化的快速构建提供直观，生动，可交互，可高度个性化定制的数据可视化图表。在我们 Echarts 的官网上可以找到它的相关信息，直接搭建一个 Web 应用复制官网的代码下来并引用，如图 3 所示。

在页面上显示出来数据：

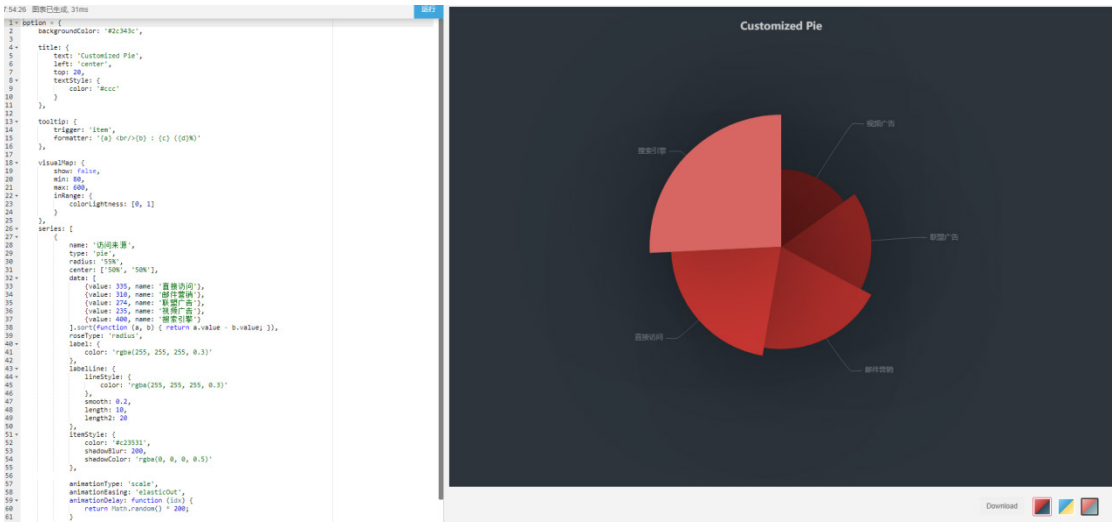

图 3 复制并引用代码

```
itemStyle:{
                normal:{
                        label:{
                                show:true,
formatter: ' {b}:{c}({d}%) '
                        }
                        labelLine:{show:true}
                        }
                }
```

使用 Echarts 构建静态的 HTML，再用 springboot 替换生成中的饼图

2.3.2DAO 数据访问层开发

在最终的结果中，每一个页面（扇区）的功能对应一个 DAO：

```
public class CategoryClickCountDAO {
    public List<CategoryClickCount> query(String day) throws IOException { List<CategoryClickCount> list = new Array<>();
    Map<String,Long>map = HbaseUtils.getInstance().query( " categoty_clickcount " ,day);
    For (Map.Entry<String,Long> entry:map.entrySet()) {
        CategoryClickCount categoryClickCount = new CategoryClickCount();
        categoryClickCount.setName(entry.getKey());
        categoryClickCount.setValue(entry.getKey());
        list.add(categoryClickCount);
        return list;
    }
    public static void main(String[] args){
        CategoryClickCountDAO dao = new CategoryClickCountDAO();
        List<CategoryClickCount> list = dao.query( " 2020 " );
        For (CategoryClickCount c : list) {
            System.out.println(c.getValue());
        }
    }
}
```

获取传入的数据并打印。

2.3.3 可视化展示

图 4 是一个类别的展示数据，在本例是一个爱奇艺类目，当鼠标悬停在某个扇区时，将显示该数据分区所对应的需求量。在这个页面可以获得想要了解的用户点击类别有偶像爱情，宫斗谋权，其他等等，从而帮助获得大的信息网储备进入数据库，并在后台对数据进行分析和视频网站维护的调度。

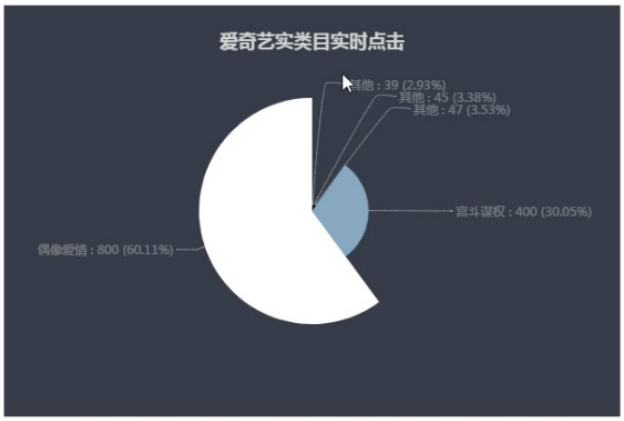

图 4 爱奇艺类目实时数据展示

3 结束语

本文通过现有的大数据集群和众多工具对视频网站做了一个简要的分析，对 Spark、Kafka 等工具做了一个简单整合。相信每个视频网站都有很重的用户分析需求，用以他们对公司未来的发展做出改变，进而合理地分配资金投入以最大化公司利益。我只是很简单地用基础知识实现这个应用，相信未来在大数据的加持下会更多的找到更多元化的数据提取方式。当大数据的宏利已经来临，把大数据应用的观念贯穿在各行各业将是未来不变的趋势，这就是所谓的科学化分析用户需求。

【参考文献】

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

[2] 王元卓，靳小龙，程学旗，等，网络大数据：现状与挑战 [J]. 计算机学报，2013，36(6)：1125-1138.

[3]Chang F，Dean J，Ghemawat S，et a1. Bigtable：ADistributed Storage System for Structured Data [J]. ACM Transactions on Computer Systems，2008，26（2）：1-26.