基于大数据的哔哩哔哩网站统计分析研究

涂文豪 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】随着智能时代的到来，视频网站哔哩哔哩应运而生，越来越多的用户加入，哔哩哔哩每天产生的数据是成倍增长的。哔哩哔哩用户对自己的排名等，需要个性化了解。本文利用大数据生态系统中的 Hadoop-Spark-Flume-Kafka 构成分析处理框架，并采用 python 对网站数据进行爬取，最终构建 springmvc+ECharts 解决方案将数据在网页上进行可视化展示。最终获取用户喜好等相关重要分析结果，通过前端可视化展示，可以让用户可直观地了解自己在哔哩哔哩网站的排名等重要信息，用户体验良好。

【关键词】Spark；Hadoop；Flume Kafka；哔哩哔哩

随着哔哩哔哩网站用户的增加，每天会产生大量的数据，发视频、评论、弹幕，这也对应了这个时代的特征 – 数据爆炸增长。面对大量的而复杂的数据，我们就可以利用大数据框架对其进行挖掘，大数据的特征可总结为 4V，即数据规模、处理速度快、结构复杂以及数据真实有效性 [1]。因此使用大数据框架来处理这些繁杂的数据，并从中发现有用的信息，通过 flume，kafka 监听，过滤，最终将信息进行可视化展示。

1 整体框架介绍

框架软件使用的是 hadoop+kafka+flume+spark+ spring，整套的大数据生态软件，为本文分析处理数据提供了保障。

整体框架具体使用的是分别是 Hadoop（分布式系统的基础架构）、Flume（一个高可用的，高可靠的，分布式的海量日志采集、聚合和传输的系统）、Spark（基于内存计算的开源的集群计算系统，为速度和通用目标所设计的集群计算平台）、Kafka（可持久化的分布式的消息队列）、Python。整体流程：首先我们需要获取数据，使用 python 爬虫，从哔哩哔哩网站爬取数据，然后对数据进行处理，利用 Hadoop 分布式处理，Flume+Kafka 监听过滤数据，Sparkstreaming 分析处理数据，最终通过 Springmvc 嵌套 Echars 框架展示。整体数据流程如图 1 所示：

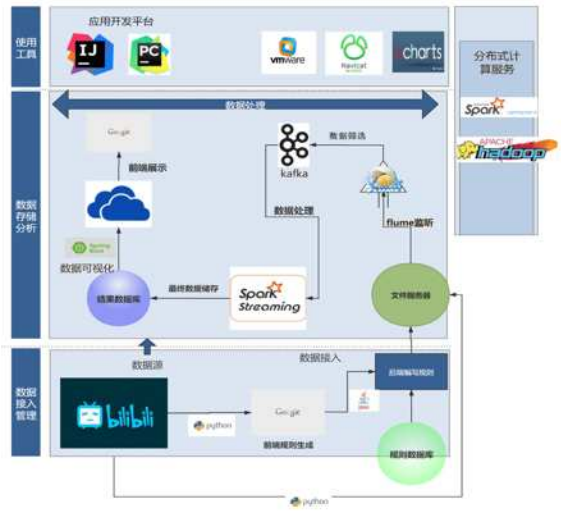

图 1 数据流程图

Fig.1 Data flow chart

Hadoop 是一个分布式的结构，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地靠近存储，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。利用这分布式的框架，加上大数据生态的 flume+kafka，对 B 站数据进行分布式处理。

2 哔哩哔哩视频网站数据获取 + 处理

2.1 背景

在大数据的时代，我们每天面对大量的数据，数据成倍地增长，视频网站的发展迅速，也让数据变得庞大，哔哩哔哩作为年轻人喜爱的视频网站，每天会产生许多

有趣的数据，因此可以通过大数据框架对其进行分析，获取我们想要的数据，例如博主的发视频习惯，热门词汇，受欢迎博主排名等。

2.2 获取数据

本文采用 python 爬虫获取数据，哔哩哔哩网站的数据繁多，我们主要获取 up 主，uid，关注数，黑名单，粉丝数，视频名，aid 播放量弹幕量，收藏数，硬币数，转发数，点赞数，up 主视频播放数，up 主文章总浏览数。使用 Pycharm 编写 py 文件爬虫，通过 JavaApi 直接获取所需信息。代码如图 2 所示：

图 2 爬虫代码图

Fig.2 Crawler code diagram

爬虫数据是通过 uid（视频博主 id）来获取的，获取数据如下：

up 主	记录我抽卡的日常
uid	37663924
关注数	48
黑名单	0
粉丝数	3922569
视频名	《食物语》蟹酿橙池子实录＋百日抽卡能得到什么呢?
aid	625718222
播放量	39
弹幕量	5434347
收藏数	1279
硬币数	97522
转发数	7636

2.3 监听过滤数据

使用 Flume+Kafka 处理数据。由于获取的数据错综复杂，存在我们不想要的数据，因此使用 Flume 监听数据的读入，kafka 启动消费者，kafka 和 flume 都是日志系统，kafka 是分布式消息中间件，自带存储，提供 push 和 pull 存取数据功能。flume 分为 agent（数据采集器）,collector（数据简单处理和写入）,storage（存储器）三部分，每一部分都是可以定制的。kafka 做日志缓存应该是更为合适的，但是 flume 的数据采集部分做的很好，可以定制很多数据源，减少开发量。将 python 爬下来的数据放入指定目录后，对此目录进行监听，过滤，筛选。数据进行存取后，直接用 flume 监听数据存放的位置，里面可以进行一个正则拦截 / 转换，然后初步筛选后，自己选择再过 kafka 数据分析处理。主要设置的参数有 source，channel，sink，这里我分别使用的是 taildir，memory 对数据进行分布式处理，属性参数如下：

```
a1.sources.r1.type=TAILDIR
a1.sources.r1.positionFile=/root/software/Data/taildir_position.json
a1.sources.r1.filegroups=f1
a1.sources.r1.filegroups.f1=/root/software/Test/shuju.txt
a1.sources.r1.interceptors.i1.type=timestamp
a1.channels.c1.type=memory
a1.sinks.k1.type=org.apache.flume.sink.kafka.KafkaSink
a1.sinks.k1.kafka.topic=Test2
```

一个完整的 Agent 中包含了三个组件 Source、Channel 和 Sink，Source 是指数据的来源和方式，Channel 是一个数据的缓冲池，Sink 定义了数据输出的方式和目的地 [2]。

2.4 数据处理

通过 python 拿过来数据，再通过 flume+kafka 过滤监听数据，这一步能将我们的脏数据变为干净数据，方便我们接下来的数据处理，但数据是一个文本文件，无法进行直接分析处理，因此我们可以通过数据连接池，将 kafka 获取的数据写入数据库，最终得以保存，这需要使用 SparkStreaming 来对数据进行入库，然后使用 Scala 语句（Scala 是一门多范式的编程语言，一种类似 java 的编程语言，设计初衷是实现可伸缩的语言、并集成面向对象编程和函数式编程的各种特性）分析处理。

首先需要建立连接池，然后使用连接池的 getConnection 方法可以获取数据库连接。

连接池需要设置最大并发数，还有当数据库初始化时，创建的连接个数，最小空闲连接数，数据库最大连接数，空闲释放时间。将 url，username，password 设置好，基本的配置就完成了，代码如图 4 所示：

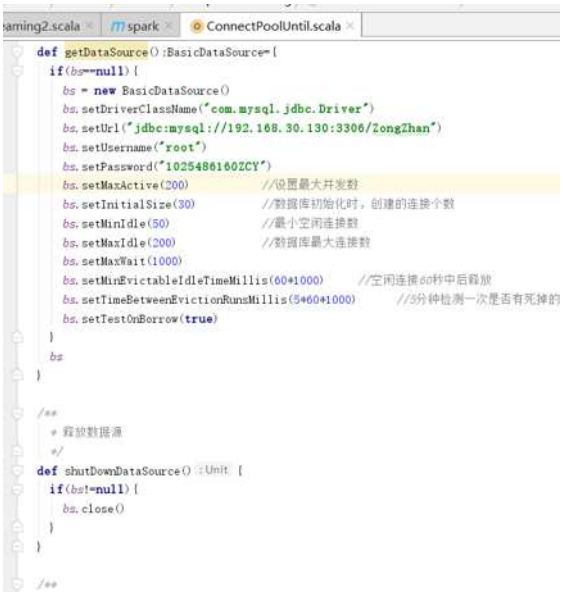

图 3 数据连接池图

Fig.3 Data connection pool diagram

完成数据池以后用数据池，将我们清洗以后的干净数据，使用 sql 语句 insert 将数据存入数据库，并使用 foreach 循环将多个数据放进去，最终能在数据库的表中查询需要的数据。关键代码如下：

```
windowtopUp.foreachRDD(line =>{
    line.foreachPartition(rdd =>{
val conn=ConnectPoolUntil.getConnection
conn.setAutoCommit(false);// 设为手动提交
val stmt=conn.createStatement()
rdd.foreach(word=>{
val sql=(“insert into countPlay(play,count) values(&apos;”+word._1+”&apos;,&apos;”+word._2+”&apos;)ON DUPLICATE KEYUPDATE count=count+values(count)”)
 println(“sql:”+sql)
stmt.addBatch(sql)
  })
```

这一步将数据存到了数据库中，但是没有对其进行处理，还只是简单的原始数据，接着，采用 scala 语言编程，对原始数据进行预处理，筛选出需要的数据字段，其关键代码如下：

```
val FansSort:DStream[(String,Int)] = lines.transform((rdd) =>{
      rdd.map{ x=>val fields:Array[String]=x.value().split(“\\s”)
      (fields(0),fields(1))
      }
      .map(x=>(x._1,x._2))
      .sortBy(x=>x._2,false).take(10)
    })
```

以上代码处理流程是：拆分数据，将拆分后的数据存入数组，使用数组将数据进行求和之类的处理，最终得到需要的数据。

数据存储的另一种方式可以使用缓存，可以使用 Docker 容器 +redis 镜像，docker 容器可以加快部署生产环境，可以将一个任务加快数十倍的速度，还可以将应用打包到容器中，可以节省很多的计算时间，提高计算性能。搭配 Docker 容器，redis 可以让数据缓存，方便使用，redis 支持数据的持久化，还支持 k–v 模式，备份等等，比直接写入库大大提高了性能。

3 前后端交互实现

当数据处理完成以后，用户无法直接简单快捷的查看使用，因此本文提供一个清晰的前端页面，利用 spring 的兼容性，嵌套 css 框架，可以实现我们的前端展示，一个优秀的前端页面可以大大地提高用户的工作效率和使用体验，在前端点击相应位置，在后端可以自动运行爬虫文件，通过输入数据规则可以获取所需的目标数据。然后将目标数据传入后端，经过相应的处理以后在前端进行展示。

Spring 框架是一个开放源代码的 J2EE 应用程序框架，由 Rod Johnson 发起，是针对 bean 的生命周期进行管理的轻量级容器 [4]。Springmvc 是 spring 框架的一个模块，springmvc 和 spring 无需通过中间整合层进行整合。spring 有多个模块，可以很好地将整个框架进行融合，分包，分类，使用 web 功能，嵌套框架，结合 tomcat 服务，可以将我们的数据使用网页展示。Springmvc 可以将我们的整体框架进行整合，最终实现 web 前端点击，后端运行，结果展示。

3.1 前端页面展示

通过 Spring+kafka+flume+speak 整体框架，将数据走一遍流程，最终展示在网页上，启动。通过最终的框架整合，完成了项目的要求，效果如下图所示：

图 4 前端首页

Fig.4 The front page

图 5 可视化展示

Fig.5 Visual presentation

4 结束语

大数据时代的到来，为我们带来了很多的便捷，哔哩哔哩网站的用户数量的日益增多，数据爆炸式的增长，用户对哔哩哔哩网站的排行，最受欢迎的up主等很好奇，用户也可以通过本网站查询自己的排名，本网站为哔哩哔哩用户带来了极大的帮助。在这个以数据说话的时代，一份可靠、稳定的数据分析，为我们的生活会带来极大的便捷。本文通过大数据生态圈里的几个经典软件，根据我们的需求和选择，搭配了一整套框架。Hadoop，spark，flume，kafka 的灵活运用，为我们处理海量的数据带来了便利且可靠的处理方式，实时监听，过滤，分布式处理，加上 spark 的计算框架，利用了 spark 的很多优点，取代了 Hadoop 的一些东西，Hadoop 有自己的优点，但存在与一些问题，性能低等等，通过 spark 的高兼容性，完美与 Hadoop 结合，最终实现高效快捷的处理方式。

前端与后端的结合，在 idea 上编写 springmvc 框架，spring 可以很很好的结合一些框架，具有很高的兼容性，嵌套 echars 图表 (ECharts[5]，一个使用 JavaScript 实现的开源可视化库，可以流畅的运行在 PC 和移动设备上，兼容当前绝大部分浏览器)，搭配 jsp，实现了前端展示。

本文通过大数据的几个经典软件，完成了对数据的采集 – 处理 – 展示这一系列的操作，最终展示，做出了一个网站，但因为没有购买域名，只能在本地运行，用户可以通过本网站，很好地获得自己想要的数据结果。

【参考文献】

[1] Tom White. 华东师范大学数据科学与工程学院译 .Hadoop 权威指南 (第 3 版) (修订版)[M]. 北京: 清华大学出版社 ,2015.

[2] 张光前 , 白雪 . 基于消费性格的新商品推荐方法 [J]. 张光前 , 白雪 . 管理科学 ,2015(02).

[3] 王伟 , 王洪伟 , 孟园 . 协同过滤推荐算法研究 : 考虑在线评论情感倾向 [J]. 系统工程理论与实践 ,2014(12).