基于 Spark 和 Hadoop 的电商网站用户大数据应用

王云兮 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】随着互联网高新技术的飞速发展，电商网站迎来了数据量爆炸式增长，巨大的数据量面临一个处理瓶颈，如何去充分有效的处理和利用这些后台数据成为目前亟待解决的难题。从该背景出发，本文设计了相应的技术解决方案，主要以 Hadoop、Spark 核心技术为手段开展相关的分析工作。首先针对后台相关日志，利用 SpringBoot 技术框架获取数据，之后将其存放于 Hadoop 框架内的 HDFS 之中，然后采用 Spark 技术针对需求进行相应的数据处理和分析工作，从而统计出页面跳转概率。页面跳转率是电商网站效益分析的一个至关重要的指标，可以为公司决策者提供有效的数据依据。

【关键词】Spark Hive HDFS 用户行为数据

1 前言

在任何大型电子商务网站上，用户都可以点击网络服务提供者的网站，点击网络运营商的网站，每天浏览、订购的日志数据将被综合处理、分析，针对电子商务相关的不一样的实际应用领域，相关的针对大数据信息的分析处理技术以完全渗透其中。选择合适和有效的数据处理引擎可以大大提高数据生产效率，然后间接或直接提高相关组的服务处理效率。

许多电商网站在以前处理数据都以 HiveSql 为主，底层计算采用擎 MapReduce，但是 HiveSql 不能处理的业务，都是交给了工程师去直接编写 MapReduce 程序实现。然而，对于相应领域的革新以及相应的需求愈发广泛的情况，一类简易的 MapReduce 应用程序以应付不及。首先，针对那些具有超大迭代量的相关工作而言，MapReduce 的简易计算模型已经不能够完成了，在每次的迭代计算任务中，都得将相应的数据信息落盘，这个缺点极大的影响了效率，在机器学习算法中，这个缺点被无限放大。另一方面，也是最重要的一个方面，我们从网页中收集到的数据通过前端返回来，对于原生态的相关日志而言，因为大量因素而导致它们的类型基本是结构与非结构化，所以，我们对其清洗的过程，需要结合 SQL 查询，以及很多复杂的过程式逻辑处理。这部分工作以前很多是由 Hive SQL 和 Python 脚本来完成的。这种方式存在很多的问题例如效率问题。如今一些具有影响力的电商网站的数据量都特别大，会直接影响整个流程的效率。

由此现在越来越多的电商网站采取了 Spark 进行数据分析和处理，但会继续沿用 Hadoop 集群作为基础平台使用。现在所利用相关模式则以“Spark on yarn”为核心，针对所涉及的各类程序而言，Yarn 会针对相关资源对程序执行统一调度操作。

2 技术选型

2.1Scala 和 Spark

Scala 作为一类编程性语言，它是属于多范式型的，将函数式以及面向对象两大类开发编程思想进行集成就是该语言设计之初的目标所在。该语言工作于 JAVA 环境的虚拟机之上，有效是适应于现存的 JAVA 程序。针对该语言源码执行编译操作之后，就会被直接转换成相应的 JAVA 字节码，所以 Scala 完全可以运行在 JVM 中，而且它可以使用 Java 的各个子类库。

Apache Spark 是一类实用高效行计算引擎，它在处理大规模信息数据中具有优势。Spark 是 UC Berkeley AMP lab （加州大学伯克利分校的 AMP 实验室）所开源的类 Hadoop MapReduce 的通用并行框架，Spark，拥有 Hadoop MapReduce 所具有的优点；相对于 MapReduce 而言，Spark 与其相异之处在于它针对中间的输出结果直接存放于相应的内存里，针对 HDFS 不会反复执行相应的读写操作。因此 Spark 能够完成迭代式开发作业，而 MapReduce 不可以执行迭代式开发作业。

针对 Hadoop 而言，Spark 与其的开源集群式的工作环境相类似，然而两者之间同时也存在着一些较为明显的不同之处，Spark 相比 Hadoop 有着更强大工作计算负载能力，更优的作业处理效率，也就是说，它可以在内存内完成数据的存取操作，不仅能够针对工作负载执行相应的优化操作，而且针对查询业务还能够提供交互式的操作方式。

Scala 语言去实现 Spark，Spark 把 Scala 当做它的应用程序的载体。与 Hadoop 相异的就是，Scala、Spark 二者密切联系，Scala 针对分布式存放的相关数据集，能够采取与针对本地集合相同的执行方式对其进行相应的操作处理。

Spark 以分布式的相关数据集为操作对象，对其执行相应迭代作业，然而在实际应用当中，可以把它视为是针对 Hadoop 的一类扩展，它能够有效的处于 Hadoop 文件系统中执行相应的并行式执行操作。该操作需要 Mesos 来协助完成。位于加州大学伯克利分校内的 AMP 实验室研发的 Spark，能够针对那些具有时延小以及大型等特性数据信息分析类程序开展有效的开发工作。然而与 MapReduce 不一样的一点在于，Spark 针对那些在 job 输出产生的中间结果能够实时的执行存储于内存中的措施[1]。

2.2 Flume

Flume 是一类涵盖了高可用，可靠，支持分布式等优质特性的系统，“Flume 是一种针对分布式数据采集、传输等可靠性能较高的工具，通过其能实现数据到数据流的控制管理”[2] 它能够针对那些海量型的日志数据执行有效的传送，聚合以及采集等操作。它还能够针对相关数据信息执行简单处理操作，并将其写入诸如 Hbase 和文本等数据信息的接受一方。flume 的数据流由事件 (Event) 贯穿始终。Flume 中最基本的数据单位就是事件，在一个事件内涵盖了相关的头数据以及相应的日志类信息，在 Agent 外的 Source 产生这些相关的事件。就在 Source 针对相应的事件完成获取之后，再对其执行相应的格式化处理，最后将处理事件传送至 Channel 内其中 Channel 可视为一类缓冲区，该缓冲区能够有效的对于相关事件进行存储，并且会存储至 Sink 事件完成相应的处理操作。Sink 所要完成的关键任务就是将相应事件传送至 Source 或针对相关的日志数据信息执行持久化存储。

2.3 SpringBoot

SpringBoot 可视为 Spring 的衍生技术，它相当于是 Spring 的一类子框架，而且在各自所执行的业务功能中，即为相近。然后相对 Spring 而言，它有着更为精简框架构造。真正开发中而言约定总是要高于配置，而它对突出的优势就在于无需在耗费精力执行与 Spring 相关的配置工作。只需要创建工程，就可以获得一个产评级的应用。

在 SpringBoot 进行实际应用时，仅仅针对那些该框架所必须的执行相应的配置操作，然后就能够有效利用 Spring 中的各类相关组件了。也就是说，它的内部有效集成了各类框架的优质特性，从而直接省去了手动配置的繁琐工作。但是其实本质上 Spring Boot 还是 Spring，它只是帮大家配置了 Spring Bean 配置。

2.4 Hadoop

Hadoop 的开发是由 Java 来完成的，其最为关键功能在于针对海量的数据信息执行相应的分布式存储以及数据的分析，是一类开源的架构，“Hadoop 可以看作是 MapReduce 处理引擎的处理框架，引擎和框架通常可以相互替换或同时使用，例如 :Spark 可以结合 Hadoop”[3] 它的最核心关键是 HDFS 和 MapRedeuce。“MapReduce 是一种分布式并行编程模型，是 Hadoop 生态系统中的最为核心和最早出现的计算模型”[4]HDFS 是一个分布式文件系统：其涵盖了两个关键的内容，其一为 Namenode 服务器能够针对原信息实施相应的存储操作，其二为能够针对实际信息执行存储操作的 Datanode，我们要存和取的数据其实就是从 Datanode 进行操作。此系统为什么如此火的原因是因为它能支持超大内容的文件，以及数据备份和心跳机制。能够很好的适应高低峰使用期中的各种情况。它通过有效利用负载均衡操作，从而完成针对各节点中负载的均衡化操作，该操作有效促进了资源利用率以及性能的提升。因负载均衡操作需要针对节点间存在数据信息执行相应的迁移操作，所以执行该操作就会产生一定程度的开销，对此就要求其应该具有高效率的算法 [5]。

MapReduce 是一个计算框架：MapReduce 中最关键的理论技术路线在于将计算任务直接分发于相应的集群服务器中进行相应的操作。它在执行分布式运算的操作中，首先针对计算任务完成相应的拆分操作，随后由 JobTracker（即任务调度器）完成最终的运算工作。能够有效的执行海量数据分析以及离线分析操作，就是 Hadoop 被广泛应用的理由所在，然而它针对几个记录执行随机读写操作的在线处理模式并不适用。在该系统集群完成部署操作之后，能够利用其相关指令完成传送相关文件至 HDFS 集群的操作，同时还可利用 Web 页面针对集群的状况执行实时查看的操作，利用相关指令完成目录的创建、文件的删除等相关操作。

3 设计与实现

3.1 系统架构图

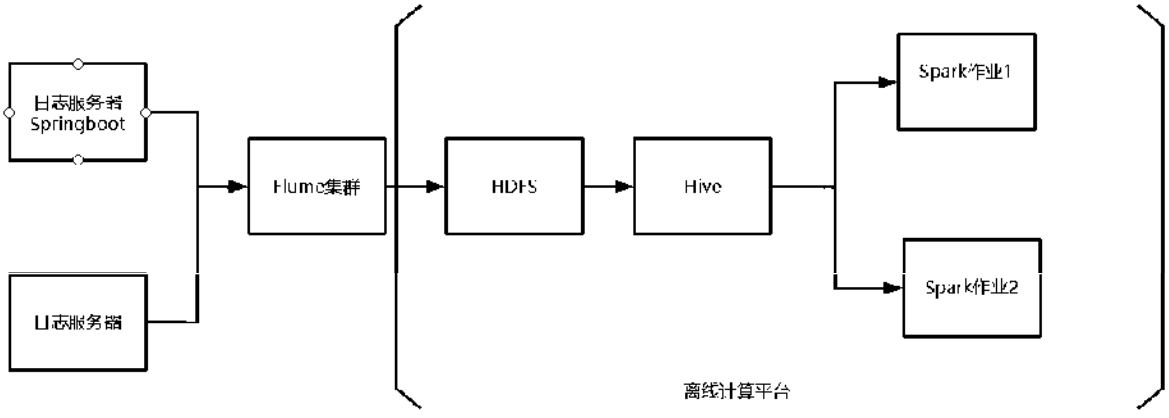

图 1 系统架构图

从上图可以得知，可以从 Springboot 中拿到需要的所有数据，并上传到 Flume 集群，再从 Flume 系统中上传到 HDFS 中，针对最终的数据仓库的构建工作，通过 Hive 的技术来完成，针对相关数据信息的处理分析

操作则利用 Spark 以相应需求为依据来完成，从而为企业中的相关决策工作提供相应的数据信息支持。

3.2 数据采集

针对数据的获取采集工作可有效利用 Springboot 来完成，在业务逻辑层（即 Service 层）内，针对相关用户操作产生的时间、品类等数据进行获取，并存至相应的日志文件内。随后 Flume 集群针对该日志实施相应的监听操作。

相关的数据信息的上传工作是由日志服务器至 Flume 集群内，之后传送至 HDFS 并实施保存。Flume 组件中的 source 针对获取的数据信息逐个实施封装至相应的事件内的操作，之后在传送至 channel 内该操作主要利用“commit”或“put”来完成。其中 channel 可视为队列，其相应的也就具有先进先出的特性，事件存至队列中，之后逐个出队，sink 主动获取数据并传送 HDFS 内。

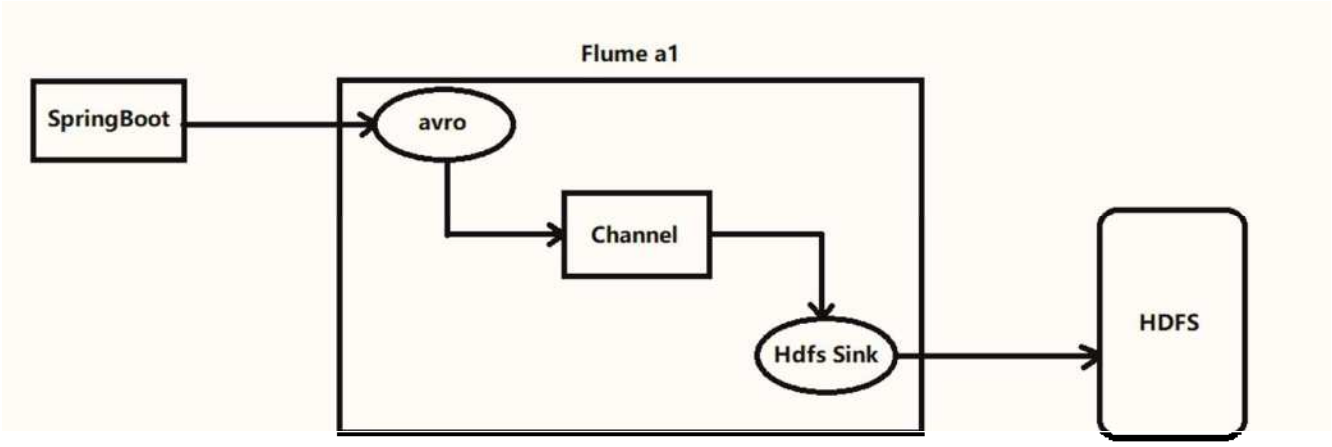

图 2 数据流

3.3 需求分析

所获取的数据信息就是由每个 Session 而生成的，每个 Session 会话都会生成相应海量数据信息，每次针对网页的执行点击操作后，其相应数据信息就会存放至相应的会话中。这可以让分析人员得到每一个用户在点击网页的时候的踪迹，这里可以按照时间顺序，判断用户点击的顺序，因为数据的量是庞大的，分析人员就可以按照分析的结果分析出跳转页面的概率为多大。

3.4 数据分析

数据从 HDFS 中获得，从而使用 Spark 进行分析。这里引入了一个案例，通过分析电商后台网页的 Session（会话）从而可以分析出来每一个用户再进行网页跳转的概率是多大的。此功能可以获得用户在电商网页的点击过程，从而调整网站的战略方向。第一步：首先要把数据读取出来。第二步，对指定的想知道的页面跳转进行统计。第三步，就是分别把所有跳转网页的数量求出来，第三步就是分别求得跳转网页的序列。第四步，针对网页跳转概率实施相应的计算，例如可以针对网页 A 转至网页 B 的概率进行一个计算操作。

部分代码展示：

```
val ids = List[Long](1,2,3,4,5,6,7)
val okflowIds: List[(Long, Long)] = ids.zip(ids.tail)
```

这一部分代码是可以随时更换的，明确了哪些页面需要计算跳转次数。这里的 ids.zip(ids.tail) 方法就是用到了一个拉链的操作，可以直接获得一组 List。

```
val pageidToCountMap: Map[Long, Long] = actionDataRDD.filter( action => { ids.init.contains(action.page_id) }).map( action => { (action.page_id, 1L) }).reduceByKey(_ + _).collect().toMap
```

这里可以求出起始跳转网页的总数，为计算概率得到分母

```
val sessionRDD: RDD[(String, Iterable[UserVisitAction])] = actionDataRDD.groupBy(_.session_id)
val mvRDD: RDD[(String, List[((Long, Long), Int)])] = sessionRDD.mapValues( iter => { val sortList: List[UserVisitAction] = iter.toList.sortBy(_.action_time)
```

对每一个 Session 分组，然后对分组后的结果进行时间排序，目的是为了得到一个正确的点击顺序。

```
val flowIds: List[Long] = sortList.map(_.page_id)
val pageflowIds: List[(Long, Long)] = flowIds.zip(flowIds.tail)
```

在此使用拉链操作，得到需要的跳转网页。

```
val flatRdd = Rddfenzi.map(_._2).flatMap(list => list).reduceByKey(_+_)
```

结果举例：a→b/a 可以这样直接得出 a 跳转 b 的概率。

图 3 页面跳转率

由结果图可以清晰的知道每个网页之间跳转的概率。对于产品经理和运营总监，他们可以根据这个指标，去尝试分析，整个网站，产品，各个页面表现得怎么样，是否要去分析产品的规划，吸引顾客可以最终付款下单。对数据分析师来说，可以由此数据做更深一步的计算和分析。对于企业管理层，可以看到整个公司的网站，各个页面之间的跳转的直观表现，对于不是特别懂技术的人来说，也是足够看懂的，这使得他们可以适当调整公司的经营战略或者策略。

总结

本文用到了 Spark，HDFS 这一系列的框架工具。以 Hadoop 作为基础平台，其涵盖了针对相关数据信息的分析以及存储等功能。该项目中针对海量网页中所涵盖的数据，利用 HDFS 完成相应的存储操作，之后通过 Spark 集群中在内存内实施计算工作的迭代式运算模型，可以帮助分析和计算用户的后台日志信息。在我国的几家龙头电商网站中，例如淘宝，京东，他们的用户数量都是以亿为单位计算的，每时每刻都会产生大量的数据。在 Spark 技术中，工程师可以计算大量用户产生的数据，这样就可以直接做到对用户的行为预判，从而使电商网站运营的更加的有针对性，更加促进电商网站的发展。

【参考文献】

[1] 王松 . 基于 spark 的会话语料库管理系统 [D]. 河北师范大学 ,2020:

[2] 张丽华 , 马家龙 , 程晓旭 , 邹雨轩 , 刘博宁 , 贾美娟 . 基于 hadoop 架构的电信离线数据综合处理的设计与实现 [J]. 智能计算机与应用 ,2020,10(12):160-163+169.

[3] 童莹 , 杨贞卓 .hadoop 和 spark 在 web 系统推荐功能中的应用 [J]. 现代信息科技 ,2020,4(19):87-89.DOI:10.19850/j.cnki.2096-4706.2020.19.022.

[4] 张国华 , 叶苗 , 王自然 , 周婷婷 . 大数据 hadoop 框架核心技术对比与实现 [J]. 实验室研究与探索 ,2021,40(02):145-148+176.DOI:10.19927/j.cnki.syyt.2021.02.028.

[5] 肖蓉 . 分布式文件系统负载均衡技术探讨 [J]. 电子世界，2020，(09)：51-52.DOI:10.19353/j.cnki.dzsj.2020.09.024