关于项目必须知道的问题

1. 日活/月活：60w/180w
2. 注册总用户数:1000w
3. GMV：每天10个订单，一个订单是50-100，大概是500-1000w，利润大概是10%-20%
4. 1百万的商品种类
5. 有几个主题:5个主题：设备主题、会员主题、商品主题、地区主题、活动主题

统计：下单、支付、退款、评价、曝光次数

1. 统计ads层哪些指标，是怎么统计的，有什么价值

重要指标：留转G复活，清楚计算方式。

必须要知道：

1. 拉链表怎么做
2. 累积型快照事实表怎么做的
3. 金额平摊是实现方式
4. 数据量，日志数据大概70g，业务数据大概1g
5. 遇到的问题
6. Hive的数据倾斜
7. Spark的数据倾斜
8. Hbase和Phoenix
9. Es
10. Redis
11. Oom
12. Flink和spark的区别

计算模型：Spark是微批次准实时的计算框架，而flink是实时计算框架

框架的特点：

①Spark自身不支持精准一次性消费，而flink采用xxx方式实现了精准一次性消费

②flink有非常灵活的窗口：spark的窗口必须的采集周期的整数倍，但是flink是非常灵活的

Flink：低延迟、高吞吐

乱序、迟到和侧输出流

精准一次性消费

1. Spark 的shuffle和mr 的shuffle异同点
2. 订单数据重复可能是什么原因？
3. 需要增加字段怎么办？

预留字段

如果预留字段不够呢？我们还没有遇到够

如何修改字段名称呢？非常麻烦，脚本和以及统计累计值的时候。

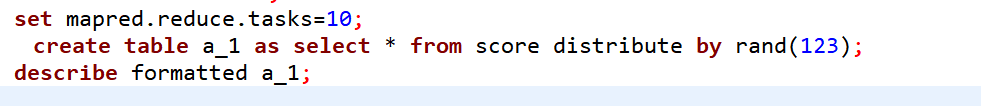
前端埋点的数据字段是谁给的？分析人员

1. 分析指标的作用：
2. 分地区 + 时间段 + 限制条件【最近7天连续3天等】流量统计和热门商品的统计
3. Hive数据倾斜的问题：

Hive底层是走mr：

Maptask数量：

和mr切片的方式一致，按文件切分，一个文件中128MB生成一个maptask。

通过如下方式，可以将一个文件分解成多个10 个文件。同时相同的key可以被打散。

Reducetask数量：

情况①：设置reduce的个数

默认情况= (mapTask输出文件的总大小大小）/ 256MB

参数设置：set mapred.reduce.tasks = 10

情况②：设置每个reduce处理数据的大小

set hive.exec.reducers.bytes.per.reducer=500000000; （500M）

常规操作：

①先过滤，先过滤不要的数据

②分区

③lzo + parquet

④开启mapjoin => 小表join大表时，如果小表的数据量比较小的时候，会将小表直接加载到内存中，避免了shuflle

⑤小文件：

a使用combinerhiveinputformat

set mapred.max.split.size=100000000; // 切片的100mb

set mapred.min.split.size.per.node=100000000; //

set mapred.min.split.size.per.rack=100000000; //

set hive.input.format=org.apache.hadoop.hive.ql.io.CombineHiveInputFormat

B 开启merge，将reduce端产生小于16m的小文件进行合并成一个256mb的大文件

情况1：小表join大表

①小表放在左边，大表放在后面。

情况2：一个任务中，某个key 的数量非常大，导致数据倾斜

①方式1：先groupby 分组 求count，确定key的分布情况，确定导致数据倾斜的key是否是需要计算你的值：

如果不需要，直接过滤，

如果需要：对key使用随机数打散，将数据发送到不同的reduce中，然后再起一个job，去掉随机数，再进行聚合操作，解决某个key特别多，导致的数据倾斜

②开启负载均衡功能

情况3：大表join大表

①增加一个字段 test= 关联字段%num。Num取决于你想启动几个job

然后（大表1 where test = 0 ）join （大表2 where test = 0）on 关联字段不变

（大表1 where test = 1 ）join （大表2 where test = 1）on 关联字段不变

（大表1 where test = 2 ）join （大表2 where test = 2）on 关联字段不变

（大表1 where test = 3 ）join （大表2 where test = 3）on 关联字段不变

（大表1 where test = 4 ）join （大表2 where test = 4）on 关联字段不变

......

出现一个reducetesk的情况

1. 没有group by的汇总，这点非常常见，希望大家尽量改写。

比如把select pt,count(1) from popt\_tbaccountcopy\_mes where pt = '2012-07-04' group by pt; 写成 select count(1) from popt\_tbaccountcopy\_mes where pt = '2012-07-04';

b) 用了Order by

c) 有笛卡尔积

21、 spark优化

问题1sparkStreaming处理后的数据保存到hdfs上有很多小文件怎么处理。

①方式1：采用sparkStreaming => kafka => flume =>hdfs

②方式2：对处理以后的结果

，如果

1. Topn这是一类指标

介绍指标

时间跨度（天、最近7天） + 过滤条件（地区）

实现方案

碰到什么问题

优化：使用flink优化

好在哪里？

低延迟和高吞吐？如何体现

1. 如何实现
2. 先将sparkStreaming实现报警数据：恶意登入监控和订单支付实时监控
3. 风控型的指标做的非常好
4. 然后先使用flink的cep实现

CEP：一个或多个由简单事件构成的事件流通过一定的规则匹配，然后输出用户想得到的数据，满足规则的复杂事件。

支持模式匹配，分为连续条件和不连续条件

什么是特点，模式序列、

1. Cep的特点
2. 不用考虑乱序、业务上的复程度
3. 时间上的连续
4. 使用cep实现，非常高

视频：2020-08-12 15:45分。第16个视频。

视频：17个，对账优化，项目技术亮点

Flink解决数据倾斜的问题

Spark vs Flink

1. Fink和spark的对比

①Flink是一个实时的流式处理框架，而spark是一个微批次的处理框架

②Flink通过。。。实现了exactly once ，但是spark未实现，需要开发人员自己实现

③Flink可以处理乱序和迟到的数据，但是Spark是不可以的

④Flink有灵活的窗口

⑤Flink有一个继承好的CEP，可以解决复杂的、时间周期的问题，降低了开发难度。

⑥资源的封装，spark提交资源只能被单个job使用，但是flink中，有session

1. Flink的提交流程

第一步：客户端提交任务到集群上

第二步：集群启动一个AM

第三步：在AM中创建一个调度器，客户端将job提交给这个调度器

第四步：调度器启动jobmanager

第五步：jobmanager向ResourceManager【Flink的RM资源管理】请求slot

第六步：RM向集群申请资源

第七步：创建Taskmanager

第八步：

26 .keyby不能设置并行度

1. 任务是什么？
2. 什么是watermark？

水位线、水位标记，水印

1.10 版本是2020.02份才有的

SparkStreaming也还在泡

上flink，用来做一些：

比如说：

风控、对账

相同的指标跑一次：日活

使用sparkStreaming的项目

TOpn：

SparkStreaming：存在数据倾斜的问题

Flink：也扛不住

后来使用预聚合

Filnk也可以认为不存在数据倾斜的问题

实时项目：

传统的sparkStreaming项目

Flink的项目：预演项目，大概4个月

Wagtermark + state + checkpoint

实操的思路

面试题

取topn：难题、优化、flink，必须讲完，背也要背下来。

在spark中：采用窗口的方式，一个窗口的数据处理一次，但是遇到数据量大的时候，容易出现oom

后来使用flink：开窗以后直接采用process，对一个窗口内的数据进行处理，但是也是容易出现oom，用时间窗口比较麻烦，当时跑618的时候，由于数据量很大，一个窗口的数据累积以后输出一次，那么就出现了oom

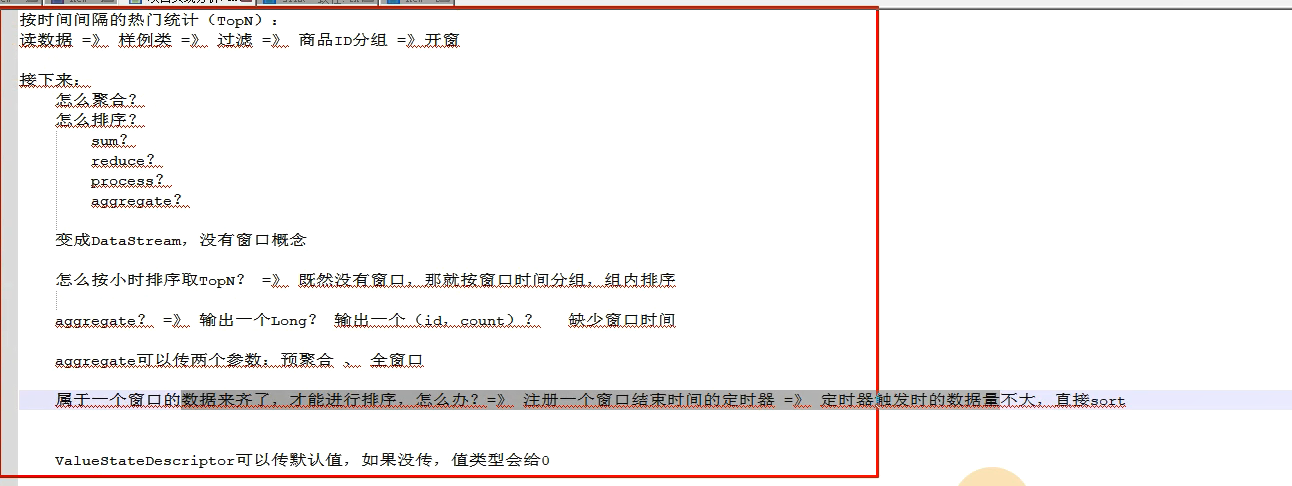
后面的优化：

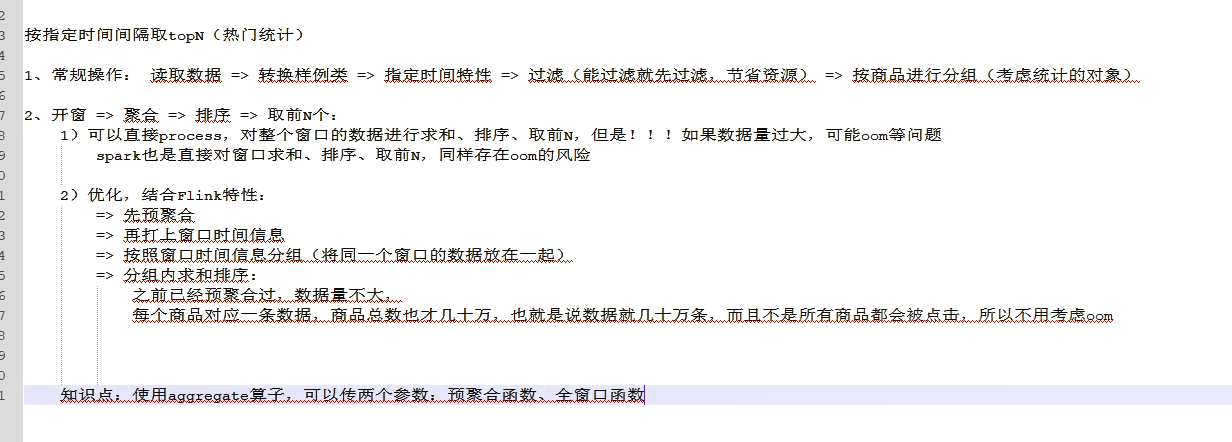
开窗 -> 使用预聚合 -> 然后通过全窗口函数给聚合后的数据打上窗口end数据

-> 然后再对windowend进行分组（为什么要分组？定时器只能在keyby以后使用，通过aggregate以后就没有了窗口的概念） -> 使用process算子进行处理

-> 在算子的内部：

1. 定义一个状态，每一个key共同享有这个状态
2. 创建一个定时器：用来触发计算，用来判断数据是否到其





处理乱序数据的时间：按照数据的最大乱序程度，我们通过采集到的数据进行分析，最大的乱序时间在3s左右，所以我们选择6s，

采集项目：6个月

搭建数仓：1-2年

实时项目：1年左右

每隔5分钟输出最近一小时内点击量最多的前N个商品

每隔5分钟输出最近一小时内点击量最多的前N个页面

亮点：

先预聚合，避免数据倾斜导致的oom

1. 设置时间语义，并将数据封装成样例类、然后抽取字段作为时间语义的时间
2. 过滤出点击行为
3. 按照itemId进行分组
4. 增加窗口函数：窗口大小为1H，滑动步长为5min
5. 使用aggregate函数，

有两个参数：一个aggregateFunciton，实现预聚合操作

第二个参数：全窗口函数，可以拿到窗口信息

预聚合以后，数据量变得非常小，聚合之后，数据量变得非常小，比如说：（100x，xx） 有100万条数据，聚合之后变成 （1001，100），并将聚合以后的结果传递给到第二个参数，传递的数据为：（item1,count1）,(item2,count2)...,将数据封装成样例类，有key、和count、窗口结束时间，由于数据是源源不断的产生，同一条数据会属于不同的窗口中，所以输出的结果集也是多个窗口的数据

窗口结束时间：是指当前窗口的结束时间，通过上下文就可以获取

1. 按照窗口结束时间进行分组，后续的操作都是针对某一个key而言的
2. 调用process方法，在方式的内部：自定义processFunction方法
3. 使用状态后端，对同一窗口的数据进行保存

定义两个状态，并在open方法中，状态进行初始化

1. 存储数据集，来一条数据存储一条
2. 存储定时器的值

（1001,100，windowEnd1）、（1002,102，windowEnd1）

问题1：我怎么知道数据来齐了，只有数据来齐了，才能进行排序

采用定时器。

来一条数据存储一条

注册一个定时器，定时器的时间是数据的窗口结束时间

并更新定时器的值

待定时器时间到了以后调用ontimer方法

将存储的数据转化转换成list集合

然后按照点击量进行倒排取前3条

最后将结果输出

恶意登入

如果同一用户（可以是不同IP）在2秒之内连续两次登录失败，就认为存在恶意登录的风险

实现逻辑：

对账信息

实现步骤：

1. 流1.IntervalJoin(流2)
2. Between(开始时间，结束时间)
3. Process(new ProcessJoinFunction（

重写process的方法，有左流和右流 ，ctx + out

相同数据的处理逻辑。

）)

实时统计每小时内的网站PV

实现逻辑：

1. 设定事件时间语义
2. 处理延迟数据为10s
3. 过滤出页面访问的数据
4. 对数据进行keyby，因为是统计每小时的pv，页面访问量->pv,1,按照pv进行分组
5. 开窗，使用timewindow，时长为1H
6. 使用sum求和

网站的每小时的独立访客数（Unique Visitor，UV）

涉及到去重的问题

常规操作：

1. 设置事件时间语义，并对数据进行转换，指定事件时间的字段
2. 过滤出pv行为
3. 转换数据格式（uv，用户id）
4. 按照uv进行keyby
5. 开窗：1H
6. 自定义process：创建一个集合，用来存储uid，遍历一个窗口的数据，将数据加入到uvset中
7. 然后一个窗口输出一次结果，并清空集合

页面广告点击量统计

1. 常规操作：设置事件时间语义 + 封装样例类 + 设定时间语义的字段
2. 按照省份和广告id进行分组
3. 开窗：10分钟，5秒钟
4. 使用aggregate函数：

有两个参数

参数1：统计相同key【uid + 广告点击量 】的点击量

参数2：是一个全窗口函数：将参数1的计算结果传递给到2，将数据打上windowEnd的标签

1. 按照窗口进行分组
2. 对窗口内的数据，直接打印即可。

黑名单过滤