XX电商用户行为分析系统

1. 项目介绍
2. 项目背景

当今社会有很多的电商网站，这就存在着一些竞争关系，为了更好的设计一个网站，让一个 电商网站浏览的人数更多，从而增加点击量和订阅量的，这样就需要我们对这个电商网站进 行分析和数据挖掘，我们可以根据每天浏览某电商网站的人数和访客量来判断一个网站的好坏和受欢迎程度，同时也可以根据外链的跳转率和访客或会员所用的浏览器等工具的分析来 进行精准的广告推广，我们也可以根据地区的点击量和访客或是会员访问的时间的分析来进 行合理的商品推广，精准的推荐等操作。同时每一个电商网站，可以根据这个网站的支付订单数以及成功支付订单数来进行业务的分析，这些对于提高一个网站的点击量、浏览量、以 及成功支付订单量都是必不可少的。

即指用户访问网站时的所有访问、浏览、点击行为数据。比如点击了哪一个链接，在哪个网页停留时间最多，采用了哪个搜索项、总体会话时间等。而所有这些信息都可被保存在网站 日志中。通过分析这些数据，可以获知许多对网站运营至关重要的信息。采集的数据越全面， 分析就能越精准。

总之，一个电商网站就应该设计出一款产品能让用户的体验好，能让用户精准的寻找想要购买的商品，能提高用户的转化率，能提广告的转化率。

1. 点击流数据分析
2. 日志
   1. **日志的生成渠道**：

|  |
| --- |
| ①是网站的 web 服务器软件（httpd、nginx、tomcat）所记录的 web 访问日志；这里是NGINX  ②是通过在页面嵌入自定义的 JS 代码来获取用户的所有访问行为（比如鼠标悬停的位置，点击的页面组件等），然后通过 AJAX 请求到后台记录日志；这种方式所能采集的信息最全面；  ③通过在页面上埋点 1 像素的图片，将相关页面访问信息请求到后台记录日志； |

* 1. **日志数据内容详述**：

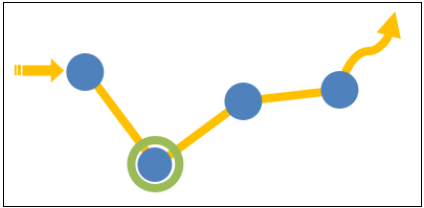
|  |
| --- |
| 在实际操作中，有以下几个方面的数据可以被采集：   1. 访客的系统属性特征。比如所采用的操作系统、浏览器、域名和访问速度等。 2. 访问特征。包括停留时间、点击的 URL、所点击的“页面标签”及标签的一些属性 （比如业务 entity的名称）等。 3. 来源特征。包括来访 URL，来访 IP 等。 4. 产品特征。包括所访问的产品编号、产品类别、产品颜色、产品价格、产品利润、产 品数量和特价等级等。 |

* 1. **日志数据举例**：

|  |
| --- |
| GET  /log.gif?t=item.010001&m=UA-J2011-1&pin=- &uid=1679790178&sid=1679790178|12&v=je=1$sc=24-bit$sr=1600x900$ul=zhcn$cs=GBK$dt=xxxx$hn=item.jd.com$fl=16.0r0$os=win$br=chrome$bv=39.0.2171.95$wb=14 37269412$xb=1449548587$yb=1456186252$zb=12$cb=4$usc=direct$ucp=- $umd=none$uct=-$ct=1456186505411$lt=0$tad=- $sku=1326523$cid1=1316$cid2=1384$cid3=1405$brand=20583$pinid=- &ref=&rm=1456186505411 HTTP/1.1 |

1. 点击流概念

**概念介绍**：点击流这个概念更注重用户浏览网站的整个流程，网站日志中记录的用户点击就像是图上的 “点”，而点击流更像是将这些“点”串起来形成的“线”。也可以把“点”认为是网站的 Page， 而“线”则是访问网站的 Session。所以点击流数据是由网站日志中整理得到的，它可以比网 站日志包含更多的信息，从而使基于点击流数据统计得到的结果更加丰富和高效。



* 1. 点击流数据模型

**点击流数据**在具体操作上是由散点状的点击日志数据梳理所得，从而，点击数据在数据建模时应该存在两张模型表（**pageviews** 和 **visits**）：

* 用于生成点击流的**原始访问日志表**:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **时间戳** | **IP地址** | **请求URL** | **Referal** | **响应码** | **流量** | **br信息** |
| 2012-01-01 12:31:12 | 1 101.0.0.1 | 2012-01-01 12:31:12 | /a/.... | 200 |  |  |
| 2012-01-01 12:31:12 | 1 101.0.0.1 | 2018-01-01 12:31:12 | /a/.... | 302 |  |  |
| 2018-01-01 12:31:12 | 1 101.0.0.2 | 2018-01-01 12:31:12 | /c/.... | 500 |  |  |
| 2018-01-01 12:31:12 | 1 101.0.0.3 | 2019-01-01 12:31:12 | /d/.... | 503 |  |  |

* 页面点击流模型 **pageviews** 表:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Session** | **IP 地址** | **时间** | **访问页面 URL** | **停留时长** | **第几步** |
| S001 | 101.0.0.1 | 2012-01-01 12:31:12 | /a/.... | 30 | 1 |
| S002 | 101.0.0.2 | 2012-01-01 12:31:12 | /b/... | 20 | 1 |
| S003 | 101.0.0.3 | 2012-01-01 12:31:12 | /c/... | 10 | 3 |
| S004 | 101.0.0.4 | 2012-01-01 12:31:12 | /d/... | 100 | 2 |

* 点击流模型 **visits 表**(按 **session** 聚集的页面访问信息)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Session** | **起始时间** | **结束时间** | **进 入 页面** | **离 开 页面** | **访 问 页 面数** | **IP** | **cookie** | **referal** |
| S001 | 2012-01-01 12:31:12 | 2012-01-01 12:31:12 | /a/.... | /a/.... | 1 | 101.0.0.1 | User01 | somesite. com |
| S002 | 2012-01-01 12:31:12 | 2012-01-01 12:31:12 | /b/... | /b/... | 2 | 101.0.0.1 | User02 | - |
| S003 | 2012-01-01 12:31:12 | 2012-01-01 12:31:12 | /c/... | /c/... | 15 | 101.0.0.1 | User03 | baidu.co m |
| S004 | 2012-01-01 12:31:12 | 2012-01-01 12:31:12 | /d/... | /d/... | 5 | 101.0.0.1 | User04 | google.co m |

这就是点击流模型。当 WEB 日志转化成点击流数据的时候，很多网站分析度量的计算变得 简单了，这就是点击流的“魔力”所在。基于点击流数据我们可以统计出许多常见的网站分 析度量指标

1. 网站流量数据分析的意义

网站流量统计分析，可以帮助网站管理员、运营人员、推广人员等实时获取网站流量信息， 并从流量来源、网站内容、网站访客特性等多方面提供网站分析的数据依据。从而帮助提高 网站流量，提升网站用户体验，让访客更多的沉淀下来变成会员或客户，通过更少的投入获 取最大化的收入。

|  |  |  |
| --- | --- | --- |
| **网站的眼睛** | **网站的神经** | **网站的大脑** |
| 访问者来自哪里？  访问者在寻找什么？  哪些页面最受欢迎？  访问者从哪里进入？  访问者从哪里跳出？ | 网页布局合理吗？  网站导航清晰吗？  哪些功能存在问题？  网站内容有效吗？  转化路径靠谱吗？ | 如何分解目标？  如何分配广告预算？  如何衡量产品表现？  哪些产品需要优化？  哪些指标需要关注？ |

1. 点击流分析的意义
   1. 技术上

|  |
| --- |
| 可以合理修改网站结构及适度分配资源，构建后台服务器群组，比如：   * 辅助改进网络的拓扑设计，提高性能 * 在有高度相关性的节点之间安排快速有效的访问路径 * 帮助企业更好地设计网站主页和安排网页内容 |

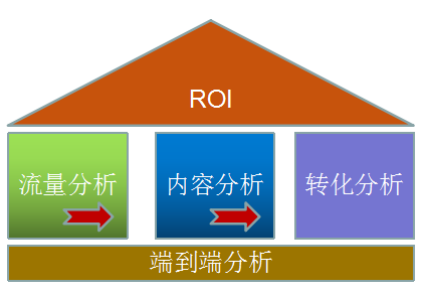
* 1. 业务上

|  |
| --- |
| * 帮助企业改善市场营销决策，如把广告放在适当的 Web 页面上。 * 优化页面及业务流程设计，提高流量转化率。 * 帮助企业更好地根据客户的兴趣来安排内容。 * 帮助企业对客户群进行细分，针对不同客户制定个性化的促销策略等。 |

终极的目标是：改善网站(电商、社交、电影、小说)的运营，获取更高投资回报率（ROI）

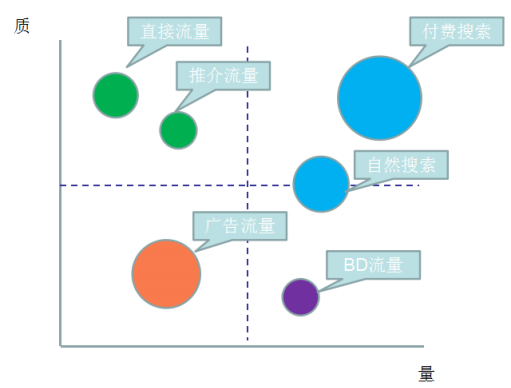
1. 网站流量分析

流量分析整体来说是一个内涵非常丰富的体系，其整体过程是一个金字塔结构：



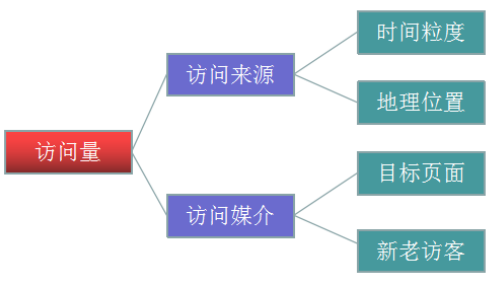
1. 网站流量质量分析

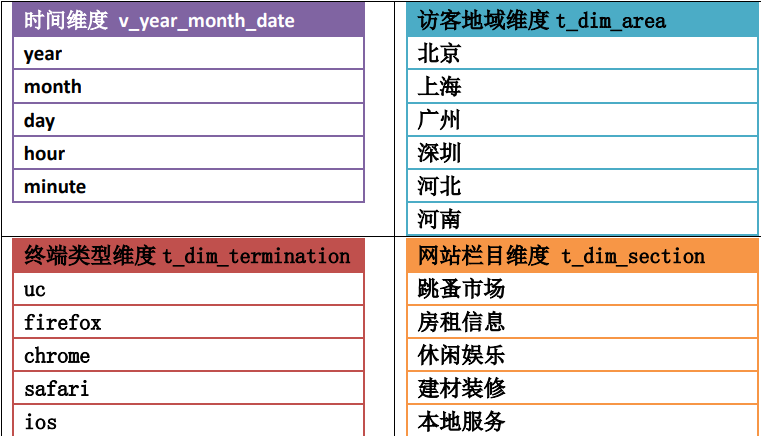
流量对于每个网站来说都是很重要，但流量并不是越多越好，应该更加看重流量的**质量**，换 句话来说就是流量可以为我们带来多少收入。



1. 网站流量多维度细分

细分是指通过不同维度对指标进行分割，查看同一个指标在不同维度下的表现，进而找出有 问题的那部分指标，对这部分指标进行优化。

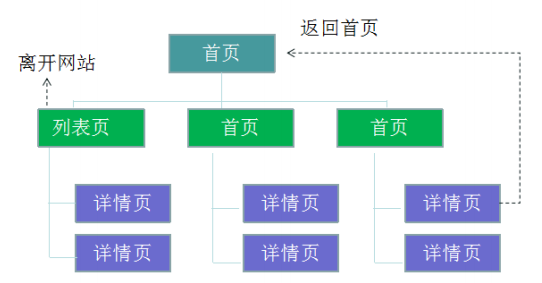




1. 网站内容及导航分析

对于所有网站来说，页面都可以被划分为三个类别：

* **导航页** -- 首页和列表页都是典型的导航页
* **功能页** -- 站内搜索页面、注册表单页面和购物车页面都是典型的功能页
* **内容页** -- 比如从内容导航分析中，以下两类行为就是网站运营者不希望看到的行为：



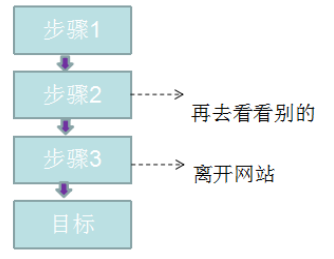
问题一：访问者从导航页进入，在还没有看到内容页面之前就从导航页离开网站，需要 分析导航页造成访问者中途离开的原因。

问题二：访问者从导航页进入内容页后，又返回到导航页，说明需要分析内容页的最初 设计，并考虑中内容页提供交叉的信息推荐。

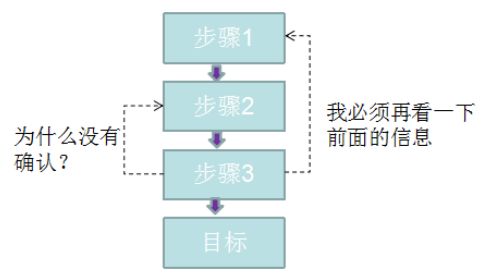
1. 网站转化及漏斗分析

所谓转化，即网站业务流程中的一个封闭渠道，引导用户按照流程最终实现业务目标（比如 商品成交）；而漏斗模型则是指进入渠道的用户在各环节递进过程中逐渐流失的形象描述； 对于转化渠道，主要进行两部分的分析：访问者的流失和迷失。

* 阻力和流失



* 迷失



1. 流量分析常见指标
   * 1. 基础分析

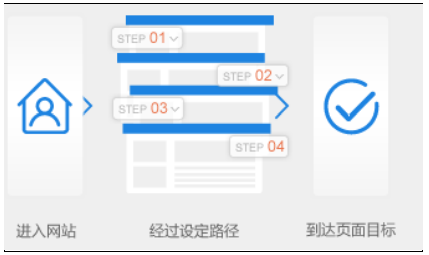
比如一些非常核心的指标：PV，IP，UV，独立访客。

* **趋势分析**：根据选定的时段，提供网站流量数据，通过流量趋势变化形态，为您分析网 站访客的访问规律、网站发展状况提供参考。
* **对比分析**：根据选定的两个对比时段，提供网站流量在时间上的纵向对比报表，帮您发 现网站发展状况、发展规律、流量变化率等。
* **当前在线**：提供当前时刻站点上的访客量，以及最近 15 分钟流量、来源、受访、访客变 化情况等，方便用户及时了解当前网站流量状况。
* **访问明细**：提供最近 7 日的访客访问记录，可按每个 PV 或每次访问行为（访客的每次会 话）显示，并可按照来源、搜索词等条件进行筛选。 通过访问明细，用户可以详细了解网 站流量的累计过程，从而为用户快速找出流量变动原因提供最原始、最准确的依据。
  + 1. 来源分析
* **来源分类**：提供不同来源形式（直接输入、搜索引擎、其他外部链接、站内来源）、不同 来源项引入流量的比例情况。通过精确的量化数据，帮助用户分析什么类型的来路产生的流 量多、效果好，进而合理优化推广方案。
* **搜索引擎**：提供各搜索引擎以及搜索引擎子产品引入流量的比例情况。从搜索引擎引入 流量的角度，帮助用户了解网站的 SEO、SEM 效果，从而为制定下一步 SEO、SEM 计划提 供依据。
* **搜索词**：提供访客通过搜索引擎进入网站所使用的搜索词，以及各搜索词引入流量的特 征和分布。帮助用户了解各搜索词引入流量的质量，进而了解访客的兴趣关注点、网站与访 客兴趣点的匹配度，为优化 SEO 方案及 SEM 提词方案提供详细依据。
* **最近 7 日的访客搜索记录**：可按每个 PV 或每次访问行为（访客的每次会话）显示，并可 按照访客类型、地区等条件进行筛选。为您搜索引擎优化提供最详细的原始数据。
* **来路域名**：提供具体来路域名引入流量的分布情况，并可按“社会化媒体”、“搜索引擎”、 “邮箱”等网站类型对来源域名进行分类。帮助用户了解哪类推广渠道产生的流量多、效果 好，进而合理优化网站推广方案。
* **来路页面**：提供具体来路页面引入流量的分布情况。尤其对于通过流量置换、包广告位 等方式从其他网站引入流量的用户，该功能可以方便、清晰地展现广告引入的流量及效果， 为优化推广方案提供依据。
* **来源升降榜**：提供开通统计后任意两日的 TOP100 搜索词、来路域名引入流量的对比情 况，并按照变化的剧烈程度提供排行榜。用户可通过此功能快速找到哪些来路对网站流量的 影响比较大，从而及时排查相应来路问题。
  + 1. 受访分析
* **受访域名**：提供访客对网站中各个域名的访问情况。一般情况下，网站不同域名提供的 产品、内容各有差异，通过此功能用户可以了解不同内容的受欢迎程度以及网站运营成效。
* **受访页面**：提供访客对网站中各个页面的访问情况。 站内入口页面为访客进入网站时浏 览的第一个页面，如果入口页面的跳出率较高则需要关注并优化；站内出口页面为访客访问 网站的最后一个页面，对于离开率较高的页面需要关注并优化。
* **受访升降榜**：提供开通统计后任意两日的 TOP100 受访页面的浏览情况对比，并按照变 化的剧烈程度提供排行榜。可通过此功能验证经过改版的页面是否有流量提升或哪些页面有 巨大流量波动，从而及时排查相应问题。
* **热点图**：记录访客在页面上的鼠标点击行为，通过颜色区分不同区域的点击热度；支持 将一组页面设置为"关注范围"，并可按来路细分点击热度。通过访客在页面上的点击量统计， 可以了解页面设计是否合理、广告位的安排能否获取更多佣金等。
* **用户视点**：提供受访页面对页面上链接的其他站内页面的输出流量，并通过输出流量的 高低绘制热度图，与热点图不同的是，所有记录都是实际打开了下一页面产生了浏览次数（PV）的数据，而不仅仅是拥有鼠标点击行为。
* **访问轨迹**：提供观察焦点页面的上下游页面，了解访客从哪些途径进入页面，又流向了 哪里。通过上游页面列表比较出不同流量引入渠道的效果；通过下游页面列表了解用户的浏览习惯，哪些页面元素、内容更吸引访客点击。
  + 1. 访客分析
* **地区运营商**：提供各地区访客、各网络运营商访客的访问情况分布。地方网站、下载站 等与地域性、网络链路等结合较为紧密的网站，可以参考此功能数据，合理优化推广运营方 案。
* **终端详情**：提供网站访客所使用的浏览终端的配置情况。参考此数据进行网页设计、开 发，可更好地提高网站兼容性，以达到良好的用户交互体验。
* **新老访客**：当日访客中，历史上第一次访问该网站的访客记为当日新访客；历史上已经 访问过该网站的访客记为老访客。新访客与老访客进入网站的途径和浏览行为往往存在差 异。该功能可以辅助分析不同访客的行为习惯，针对不同访客优化网站，例如为制作新手导航提供数据支持等。
* **忠诚度**：从访客一天内回访网站的次数（日访问频度）与访客上次访问网站的时间两个 角度，分析访客对网站的访问粘性、忠诚度、吸引程度。由于提升网站内容的更新频率、增 强用户体验与用户价值可以有更高的忠诚度，因此该功能在网站内容更新及用户体验方面提 供了重要参考。
* **活跃度**：从访客单次访问浏览网站的时间与网页数两个角度，分析访客在网站上的活跃 程度。由于提升网站内容的质量与数量可以获得更高的活跃度，因此该功能是网站内容分析 的关键指标之一。
  + 1. 转化路径分析

1. 转化定义：访客在您的网站完成了某项您期望的活动，记为一次转化，如注册或下载。
2. 目标示例
   1. 获得用户目标：在线注册、创建账号等。
   2. 咨询目标：咨询、留言、电话等。
   3. 互动目标：视频播放、加入购物车、分享等。
   4. 收入目标：在线订单、付款等。
3. 转化数据的应用
   1. 在报告的自定义指标中勾选转化指标，实时掌握网站的推广及运营情况。
   2. 结合“全部来源”、“转化路径”、“页面上下游”等报告分析访问漏斗，提高转化率。
   3. 对“转化目标”设置价值，预估转化收益，衡量 ROI。

路径分析：根据设置的特定路线，监测某一流程的完成转化情况，算出每步的转换率和流失 率数据，如注册流程，购买流程等。

1. 转化类型：
   1. 页面



b）事件



* + 1. 用户分析模块

主要分析**新增会员**（如果按天分析，就是昨天注册的会员）、**活跃会员**（只要今天登陆的就 是活跃会员，或者按照一定的周期去计算，只要在这个周期内都是活跃会员）以及**总会员**相 关信息

1. 访客：主要分析新增用户、活跃用户以及总用户的相关信息
2. 会员：主要分析新增会员、活跃会员以及总会员的相关信息
3. 会话：主要分析会话个数，会话长度和平均会话长度相关的信息
4. 主要分析每天每小时的用户、会话个数以及会话长度的相关信息
5. 项目技术架构
6. 数据处理流程

该项目是一个纯粹的数据分析项目，其整体流程基本上就是依据数据的处理流程进行，依此 有以下几个大的步骤：

* **数据采集**：首先，通过页面嵌入 JS 代码的方式获取用户访问行为，并发送到 web 服务的后台记录日志 然后，将各服务器上生成的点击流日志通过实时或批量的方式汇聚到 HDFS 文件系统中。当 然，一个综合分析系统，数据源可能不仅包含点击流数据，还有数据库中的业务数据（如用 户信息、商品信息、订单信息等）及对分析有益的外部数据。
* **数据预处理**：通过 mapreduce 程序对采集到的点击流数据进行预处理，比如清洗，格式整理，滤除脏数据等
* **数据入库**：将预处理之后的数据导入到 HIVE 仓库中相应的库和表中
* **数据分析**：项目的核心内容，即根据需求开发 ETL 分析语句，得出各种统计结果
* **数据展现**：将分析所得数据进行可视化

1. 技术选型
2. **数据收集层**：

A：前台数据埋点：使用 JavaScript 去写

B：后台数据埋点：使用 **Java** 去写、**Flume**：收集日志（如果实时的还需要 Kafka）、**MapReduce**：对数据进行预处理

1. **数据分析层**：

MapReduce、Hive、SparkSQL、azkaban、Hive + HBase(SQL)。

1. **数据展示层**：

Sqoop 导入到 MySQL/HBase：

结合三大框架：

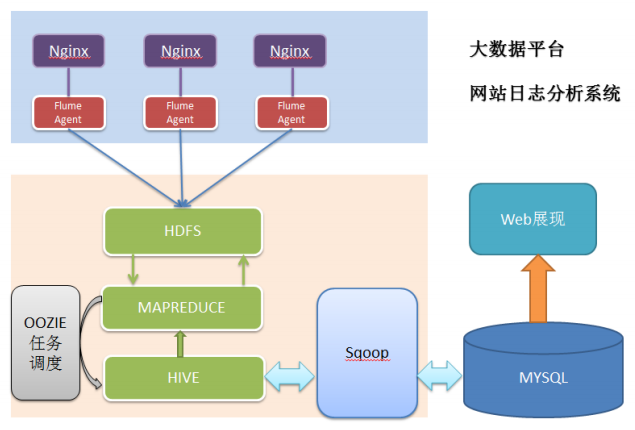
Spring + SpringMVC + MyBatis

Echarts/Highcharts

Tableau、Bootstrap、EasyUI、D3.js

1. 项目结构

本项目是一个纯粹数据分析项目，其整体结构亦跟分析流程匹配，并没有特别复杂的结 构，如下图：



强调的是：

系统的数据分析不是一次性的，而是按照一定的时间频率反复计算，因而整个处理链条中的 各个环节需要按照一定的先后依赖关系紧密衔接，即涉及到大量任务单元的管理调度，所以， 项目中需要添加一个任务调度模块

1. 项目实现
2. 模块开发—数据生成
3. 方式介绍：

* 使用 JS 进行埋点生成到服务器记录到日志中
  + Google 的 ga.js 或者新版本的 analatics.js 或者自定义的 JS 进行埋点生成和收集 数据
* 利用 Nginx 自生成日志
  + Nginx 安装和使用基本手册

1. 模块开发--数据采集
2. 需求：

数据采集的需求是广义来说分为**2**个部分：

**第一**：是在页面采集用户的访问行为，具体开发工作（**web团队**）：

* 开发页面埋点 JS，采集用户访问行为
* 后台接受页面 JS 请求记录日志

**第二**：从web服务器汇聚日志到hdfs，是数据分析系统的数据采集，（**数据分析平台建设团队**）具体实现有：

* **Shell脚本**：
  + 优点：轻量级，开发简单
  + 缺点：对日志采集过程中的容错处理不便控制
* **Java程序采集**：
  + 优点：可对采集过程实现精细控制
  + 缺点：开发工作量大
* Flume 日志采集框架
  + 成熟的开源日志采集系统，且本身就是 Hadoop 生态体系中的一员，与 Hadoop 体系中的各 种框架组件具有天生的亲和力，可扩展性强。

1. 数据收集实现：

在点击流日志分析这种场景中，对数据采集部分的可靠性、容错能力要求通常不会非常严苛， 因此使用通用的 flume 日志采集框架完全可以满足需求。**所以本项目使用flume实现数据的采集**。

1. 数据源信息：

本项目分析的数据**用 nginx 服务器所生成的流量日志**，存放在各台 nginx 服务器上，如： /var/log/nginx/access.log。

1. 数据内容样例：

|  |
| --- |
| 58.215.204.118 - - [18/Sep/2013:06:51:35 +0000] "GET /wpincludes/js/jquery/jquery.js?ver=1.10.2 HTTP/1.1" 304 0 "http://blog.fens.me/nodejs-socketiochat/" "Mozilla/5.0 (Windows NT 5.1; rv:23.0) Gecko/20100101 Firefox/23.0"  字段解析：   * + - 1. 访客ip地址： ----58.215.204.118       2. 访客用户信息： - -       3. 请求时间：----[18/Sep/2013:06:51:35 +0000]       4. 请求方式：---- GET       5. 请求的url：----/wpincludes/js/jquery/jquery.js?ver=1.10.2       6. 请求所用的协议：----HTTP/1.1       7. 响应码：----304       8. 返回的数据流量：----0       9. 访客的来源url：---- <http://blog.fens.me/nodejs-socketiochat/>       10. 访客所用的浏览器：Mozilla/5.0 (Windows NT 5.1; rv:23.0) Gecko/20100101 Firefox/23.0 |

1. 日志文件生成规律：

基本规律：

当前正在写的文件为access\_log

文件体积达到 256M，或时间间隔达到 60 分钟，即滚动重命名切换成历史日志文件： 形如： access\_log.2018-11-10-13-00.log。

1. 具体实现

|  |
| --- |
| 1. 在web服务器中部署agent点，修改配置文件。 2. 启动agent节点，将采集的数据汇聚到指定的hdfs目录中 3. 原理图：     版本选择：**flume-1.8.0**。  **采集源**：Nginx 服务器日志目录。  **存放地**：hdfs目录/weblogs。   1. **采集方案配置详情**：nginx.properties：   agent1.sources = source1  agent1.sinks = sink1  agent1.channels = channel1  # 使用spooldir（NGINX的日志目录）作为source1  #agent1.sources.source1.type = spooldir  #agent1.sources.source1.spoolDir = /var/log/nginx/  #agent1.sources.source1.fileHeader = false  #这里使用的是 tail -F 命令监控access\_log文件作为：source1  #使用 exec 作为数据源 source 组件  agent1.sources.source1.type = exec  #使用 tail -F 命令实时收集新产生的日志数据  agent1.sources.source1.command = tail -F /var/log/nginx/access\_log  agent1.sources.source1.channels = channel1  #configure host for source  #配置一个拦截器插件  agent1.sources.source1.interceptors = i1  agent1.sources.source1.interceptors.i1.type = hadoop03  #使用拦截器插件获取 agent 所在服务器的主机名  agent1.sources.source1.interceptors.i1.hostHeader = hadoop03  #配置 sink 组件为 hdfs  agent1.sinks.sink1.type = hdfs  #a1.sinks.k1.channel = c1  #agent1.sinks.sink1.hdfs.path=hdfs://myha01/weblog/nginx-access/%y-%m-%d/%H%M%S  #指定文件 sink 到 hdfs 上的路径  agent1.sinks.sink1.hdfs.path=  hdfs://myha01/weblog/nginx-access/%y-%m-%d/%H-%M\_%hostname  #指定文件名前缀  agent1.sinks.sink1.hdfs.filePrefix = access\_log  agent1.sinks.sink1.hdfs.maxOpenFiles = 5000  #指定每批下沉数据的记录条数  agent1.sinks.sink1.hdfs.batchSize= 100  agent1.sinks.sink1.hdfs.fileType = DataStream  agent1.sinks.sink1.hdfs.writeFormat =Text  #指定下沉文件按 1G 大小滚动  agent1.sinks.sink1.hdfs.rollSize = 1024\*1024\*1024  #指定下沉文件按 1000000 条数滚动  agent1.sinks.sink1.hdfs.rollCount = 1000000  #指定下沉文件按 30 分钟滚动  agent1.sinks.sink1.hdfs.rollInterval = 30  #agent1.sinks.sink1.hdfs.round = true  #agent1.sinks.sink1.hdfs.roundValue = 10  #agent1.sinks.sink1.hdfs.roundUnit = minute  agent1.sinks.sink1.hdfs.useLocalTimeStamp = true  # Use a channel which buffers events in memory  #使用 memory 类型 channel  agent1.channels.channel1.type = memory  agent1.channels.channel1.keep-alive = 120  agent1.channels.channel1.capacity = 500000  agent1.channels.channel1.transactionCapacity = 600  # Bind the source and sink to the channel  agent1.sources.source1.channels = channel1  agent1.sinks.sink1.channel = channel1 |

1. 启动采集

在部署了 flume 的 nginx 服务器上，启动 flume 的 agent，命令如下：

**flume-ng agent --conf $FLUME\_HOME/conf -f $FLUME\_HOME/conf/nginx.properties -n agent**

**注意**：

* 启动命令中的-n 参数要给配置文件中配置的 agent 名称。
* Flume采集数据的时候配置的大小不要是128M，因为真实的文件夹有差距：在128M左右。

1. 模块开发--数据预处理
2. 需求：

过滤“不合规”数据，格式转换和规整，根据后续的统计需求，过滤分离出各种不同主题的基础数据。（这里使用的MapReduce程序进行数据清洗）。

将access.log原始数据清洗出3个表：

表1（**access\_log**）：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **时间戳** | **IP地址** | **请求URL** | **Referal** | **浏览器属性** | **........** |
| 2012-01-01 12:31:12 | 101.0.0.1 | /a/... | somesite.com |  |  |
| 2012-01-01 12:31:16 | 201.0.0.2 | /a/... | aura.cn |  |  |
| 2012-01-01 12:33:06 | 101.0.0.2 | /b/... | baidu.com |  |  |
| 2012-01-01 15:16:39 | 234.0.0.3 | /c/... | google.com |  |  |
| 2012-01-01 15:17:11 | 101.0.0.1 | /d/... | /c/... |  |  |
| 2012-01-01 15:19:23 | 101.0.0.1 | /e/... | /d/.... |  |  |

表2（**Pageviews**）：由表1的数据得出。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SessionID** | **userid<ip>** | **时间** | **访问页面URL** | **页面停留时长** | **第几步** |
| S001 | User01 | 2012-01-01 12:31:12 | /a/.... | 30 | 1 |
| S002 | User02 | 2012-01-01 12:31:16 | /a/.... | 10 | 1 |
| S002 | User02 | 2012-01-01 12:33:06 | /b/.... | 110 | 2 |
| S002 | User02 | 2012-01-01 12:35:06 | /e/.... | 30 | 3 |

表3（**Visits**）：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Session** | **起始时间** | **结束时间** | **进入页面** | **离开页面** | **访问页面数** | **IP** | **cookie** | **referal** |
| S001 | 2012-01-01 12:31:12 | 2012-01-01 12:31:12 | /a/... | /a/... | 1 | 101.0.0.1 | User01 | somesite.com |
| S002 | 2012-01-01 12:31:16 | 2012-01-01 12:35:06 | /a/... | /e/... | 3 | 201.0.0.2 | User02 | - |
| S003 | 2012-01-01 12:35:42 | 2012-01-01 12:35:42 | /c/... | /c/... | 1 | 234.0.0.3 | User03 | baidu.com |
| S003 | 2012-01-01 15:16:39 | 2012-01-01 15:19:23 | /c/... | /e/... | 3 | 101.0.0.1 | User01 | google.com |
| …… | …… | …… | …… | …… | …… | …… | …… | …… |

1. 实现方式：

使用MapReduce程序将数据清洗出来，然后对相应的数据，在hive中建立上述的三张表：

1. 清洗表1 的数据（使用原始的access\_log数据作为输输入）。
2. 使用第一步清洗出来数据作为输入清洗出第二张表的数据。
3. 使用第二步清洗出来的数作为输入清洗出第三张表的数据。
4. 预处理任务调度

将以上各预处理 MapReduce 程序写成 shell 脚本，配置成 azkaban 的 job，并上传到 azkaban 服务器进行定时调度。

1. 模块开发--数据仓库设计

**注意**：

* 数据仓库设计中的 **星型模型** 和 **雪花型模**
* 事实表 和 维度表

1. 事实表

|  |  |  |
| --- | --- | --- |
| **基础数据表access\_log**： | | |
| vaild | boolean | 是否有效 |
| ip\_addr | string | 访客 ip |
| remote\_user | string, | 访客用户信息 |
| time | string | 请求时间 |
| request | string | 请求 ur |
| status\_code | string | 响应码 |
| body\_bytes\_sent | string | 响应字节数 |
| http\_referer | string | 来源 url |
| user\_agent | string | 访客终端信息（浏览器相关信息） |

Request 进行了拆分：

|  |  |  |
| --- | --- | --- |
| **ETL 中间表：t\_etl\_referurl：** | | |
| vaild | boolean | 是否有效 |
| ip\_addr | string | 访客 ip |
| remote\_user | string, | 访客用户信息 |
| time | string | 请求时间 |
| request | string | 请求 ur |
| request\_host | String | 请求的域名 |
| status\_code | string | 响应码 |
| body\_bytes\_sent | string | 响应字节数 |
| http\_referer | string | 来源 url |
| user\_agent | string | 访客终端信息（浏览器相关信息） |
| host | String | 外链 url 的路径 |
| Query | String | 外链 url 的参数 |
| query\_id | String | 外链 url 的参数值 |

|  |  |  |
| --- | --- | --- |
| **访问日志明细宽表：t\_ods\_access\_detail：** | | |
| vaild | boolean | 是否有效 |
| ip\_addr | string | 访客 ip |
| remote\_user | string, | 访客用户信息 |
| time | string | 请求时间 |
| daystr | string | 访问日期 |
| timestr | string | 访问时间 |
| Month | string | 访问月 |
| day | string | 访问日 |
| hour | string | 访问时 |
| request | string | 请求 url |
| status\_code | string | 响应码 |
| body\_bytes\_sent | string | 响应字节数 |
| http\_referer | string | 来源 url |
| ref\_host | String | 来源的 host |
| ref\_path | String | 来源的路径 |
| ref\_query | string | 来源参数 query |
| ref\_query\_id | string | 来源参数 query 的值 |
| http\_user\_agent | String | 客户终端标识 |

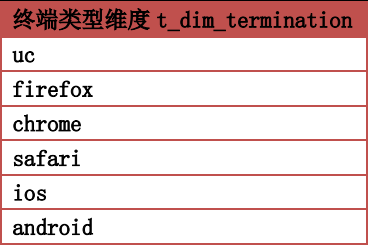
1. 维度表
2. 时间维度：



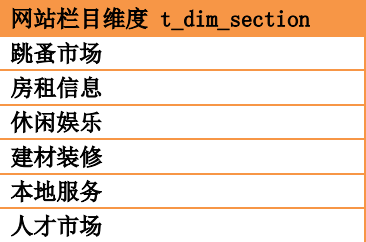
1. 访客地域维度



1. 终端类型维度



1. 网站栏目维度



1. 模块开发--ETL

该项目的数据分析过程在 Hadoop 集群上实现，主要应用 hive 数据仓库工具，因此，采集并 经过预处理后的数据，需要加载到 **hive** 数据仓库中，以进行后续的挖掘分析。

1. 建立**access\_log**表：

|  |
| --- |
| 根据上面的步骤1：建立出原始数据表：  create database if not exists weblog;  use weblog;  drop table if exists access\_log ;  create table access\_log(  vaild boolean ,  ip\_addr string ,  remote\_user string,  time string ,  request string ,  status\_code string ,  body\_bytes\_sent string ,  http\_referer string ,  user\_agent string  ) row format delimited fields terminated by '\001';    **导入数据**：  load data inpath '/user/hadoop/output/setup\_1' into table access\_log  **导入检测**：  select \* from access\_log limit 10; |

1. 建立**Pageviews**表：

|  |
| --- |
| 点击流模型 **pageviews** 表 **ods\_click\_pageviews** 建表语句：  create database if not exists weblog;  use weblog;  drop table if exists ods\_click\_pageviews ;  create table ods\_click\_pageviews(  sessionid string,  ip\_addr string,  remote\_user string,  time string,  request string,  step string ,  staylong string ,  referal string ,  useragent string ,  bytes\_send string ,  status string  ) row format delimited fields terminated by '\001' ;    **导入数据**：  load data inpath '/user/hadoop/output/setup\_2' into table ods\_click\_pageviews  **导入检测**：  select \* from ods\_click\_pageviews limit 10; |

1. 建立**Visits**表：

|  |
| --- |
| 在 hive 仓库中建点击流 visit 模型表：  create database if not exists weblog;  use weblog;  drop table if exists click\_stream\_visit ;  create table click\_stream\_visit(  sessionid string ,  ip\_addr string ,  inTime string,  outTime string ,  inPage string ,  outPage string ,  referal string ,  pageVisits int  ) row format delimited fields terminated by '\001';    **导入数据**：  load data inpath '/user/hadoop/output/setup\_3' into table click\_stream\_visit;  **导入检测**：  select \* from click\_stream\_visit limit 10; |

1. 生成**ods\_weblog\_detail**表：

|  |
| --- |
| 1. 解析URL中的信息   抽取http\_referer到中间表："t\_ods\_tmp\_referurl"  将来访 url 分离出 host path query query id  create table weblog.t\_ods\_tmp\_referurl as  SELECT a.\*, b.\*  FROM access\_log a  LATERAL VIEW parse\_url\_tuple(regexp\_replace(http\_referer, "\"", ""), 'HOST', 'PATH', 'QUERY',  'QUERY:id') b  as host, path, query, query\_id;     1. 解析时间字符串字段：   抽取转换 time 字段到中间表明细表：t\_ods\_detail  drop table if exists weblog.t\_ods\_tmp\_detail;  create table weblog.t\_ods\_tmp\_detail as  select b.\*,substring(time,0,10) as daystr,  substring(time,11) as tmstr,  month(time) as month,  day(time) as day,  hour(time) as hour  from t\_ods\_tmp\_referurl b;     1. 将数据导入到ods\_weblog\_detail表中：   insert into table ods\_weblog\_detail  select  vaild,  ip\_addr,  remote\_user, time,  daystr, tmstr,  month, day,  hour, request,  status\_code, body\_bytes\_sent,  http\_referer, host,  path, query,  query\_id, user\_agent  from t\_ods\_tmp\_detail; |

1. 模块开发--统计分析

**注意**：每一种统计指标都可以跟各维度表进行钻取, 为了在前端展示时速度更快，可以把每一个指标都事先算出各维度结果存入 mysql。

1. 多维度流量统计

|  |
| --- |
| 1. 按时间维度统计 2. 计算指定的某个小时 pvs   select count(\*) ,month,day,hour from ods\_weblog\_detail group by month,day,hour;   1. 计算该处理批次（一天）中的**各小时 pvs**   drop table dw\_pvs\_hour;  create table dw\_pvs\_hour(month string,day string,hour string,pvs bigint) partitioned by(datestr  string);  insert into table weblog.dw\_pvs\_hour partition(datestr='2013-09-18')  select a.month as month,a.day as day,a.hour as hour,count(1) as pvs from ods\_weblog\_detail a  where a.datestr='2016-06-10' group by a.month,a.day,a.hour;   1. 按天统计   drop table dw\_pvs\_day;  create table if not exists dw\_pvs\_day(  month string,day string,pvs string ) ;  insert into table dw\_pvs\_day  select month as month,day as day,count(\*) as pvs from ods\_weblog\_detail  group by month,day;   1. 按月统计   drop table dw\_pvs\_month;  create table if not exists dw\_pvs\_month(  month string,pvs string ) ;  insert into table dw\_pvs\_month  select month as month,count(\*) as pvs from ods\_weblog\_detail  group by month;   1. 按终端维度统计   对ods\_weblog\_de表中的http\_user\_agent字段进行分析：  Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/33.0.1750.29 Safari/537.36  Mozilla/5.0：浏览器公司的标识。  Windows NT 6.3：Windows 8.1的标识符  WOW64 ：32位的Windows系统运行在64位的处理器上  AppleWebKit/537.36：苹果公司开发的呈现引擎  KHTML：是Linux平台中Konqueror浏览器的呈现引擎KHTML  Gecko：呈现引擎  Chrome：浏览器类型，谷歌浏览器。  探索数据中的终端类型：  方法一：对http\_user\_agent字段进行分组聚合统计。  方法二：对http\_user\_agent抽样：  select distinct(http\_user\_agent) from ods\_weblog\_detail where http\_user\_agent like '%Mozilla%' limit 200;   1. 终端维度：firefox   drop table if exists t\_display\_pv\_terminal\_firefox;  create table t\_display\_pv\_terminal\_firefox(  pvs bigint,month string, day string,hour string  );  insert into table t\_display\_pv\_terminal\_firefox  select  count(\*) pvs,month,day,hour  from ods\_weblog\_detail  where lower(http\_user\_agent) like '%firefox%'  group by month,day,hour;   1. 终端维度：chrome   drop table if exists t\_display\_pv\_terminal\_chrome;  create table t\_display\_pv\_terminal\_chrome(  pvs bigint,month string, day string,hour string  );  insert into table t\_display\_pv\_terminal\_chrome  select  count(\*) pvs,month,day,hour  from ods\_weblog\_detail  where lower(http\_user\_agent) like '%chrome%'  group by month,day,hour;   1. 终端维度：safari   drop table if exists t\_display\_pv\_terminal\_safari;  create table t\_display\_pv\_terminal\_safari(  pvs bigint,month string, day string,hour string  );  insert into table t\_display\_pv\_terminal\_safari  select  count(\*) pvs,month,day,hour  from ods\_weblog\_detail  where lower(http\_user\_agent) like '%safari%'  group by month,day,hour;   1. 按栏目维度统计 2. 栏目维度：job   drop table t\_display\_pv\_dimension\_job ;  create table t\_display\_pv\_dimension\_job(month string,day string,hour string,pvs bigint) ;  insert into table t\_display\_pv\_dimension\_job  select month,day,hour,count(\*) pvs  from ods\_weblog\_detail  where ref\_path regexp '/job.\*' and ref\_path is not null  group by month,day,hour;   1. 栏目维度：news   drop table t\_display\_pv\_dimension\_news ;  create table t\_display\_pv\_dimension\_news(month string,day string,hour string,pvs bigint) ;  insert into table t\_display\_pv\_dimension\_news  select month,day,hour,count(\*) pvs  from ods\_weblog\_detail  where ref\_path regexp '/news.\*' and ref\_path is not null  group by month,day,hour;   1. 栏目维度：bargin   drop table t\_display\_pv\_dimension\_bargin ;  create table t\_display\_pv\_dimension\_news(month string,day string,hour string,pvs bigint) ;  insert into table t\_display\_pv\_dimension\_bargin  select month,day,hour,count(\*) pvs  from ods\_weblog\_detail  where ref\_path regexp '/bargin.\*' and ref\_path is not null  group by month,day,hour;   1. 栏目维度：lane   drop table t\_display\_pv\_dimension\_lane ;  create table t\_display\_pv\_dimension\_lane(month string,day string,hour string,pvs bigint) ;  insert into table t\_display\_pv\_dimension\_lane  select month,day,hour,count(\*) pvs  from ods\_weblog\_detail  where ref\_path regexp '/lane.\*' and ref\_path is not null  group by month,day,hour; |

1. 按 referer 维度统计 pv 总量

**需求**：按照来源及时间维度统计 pv，并按照 pv 大小倒序排序。

|  |
| --- |
| 1. 按照小时粒度统计，查询结果存入：weblog.dw\_pvs\_referer\_h   drop table if exists dw\_pvs\_referer\_h;  create table dw\_pvs\_referer\_h(  referer\_url string,referer\_host string,month string,day string,hour string ,pvs bigint  );  insert into table dw\_pvs\_referer\_h  select http\_referer as referer\_url, ref\_host as referer\_host,month,day,hour,count(\*) pvs  from ods\_weblog\_detail  group by http\_referer,ref\_host,month,day,hour  having ref\_host is not null  order by hour asc,day asc,month asc,pvs desc ;   1. 按小时粒度统计各来访域名的产生的 pv 数并排序   drop table if exists dw\_ref\_host\_visit\_cnts\_h;  create table dw\_ref\_host\_visit\_cnts\_h(  referer\_host string,month string,day string,hour string ,pvs bigint  );  insert into table dw\_ref\_host\_visit\_cnts\_h  select ref\_host as referer\_host,month,day,hour,count(\*) pvs  from ods\_weblog\_detail  group by ref\_host,month,day,hour  having ref\_host is not null  order by hour asc,day asc,month asc,pvs desc ; |

1. 统计 PV 总量最大的来源 TopN

需求描述：按照时间维度，比如，统计一天内各小时产生最多 pv 的来源 TopN。

|  |
| --- |
| select a.referer\_host,a.time,a.pvs,a.top  from (select referer\_host ,concat(month,hour,day) as time,pvs ,  row\_number() over(partition by concat(month,hour,day) order by pvs desc ) as top  from dw\_ref\_host\_visit\_cnts\_h ) a  where a.top=1     1. 根据上述功能编写 HQL 取各小时的 ref\_host 访问次数 TopN：   drop table if exists weblog.dw\_pvs\_refhost\_topn\_h;  create table if not exists weblog.dw\_pvs\_refhost\_topn\_h(  referer\_host string,  time string,  pvs string,  top string  ) partitioned by(datestr string);  insert into table dw\_pvs\_refhost\_topn\_h  select a.referer\_host,a.time,a.pvs,a.top  from (select referer\_host ,concat(month,hour,day) as time,pvs ,  row\_number() over(partition by concat(month,hour,day) order by pvs desc ) as top  from dw\_ref\_host\_visit\_cnts\_h ) a  where a.top<=3 |

1. 人均浏览页面数/人均流量

**需求**：比如，今日所有来访者，平均请求的页面数。

计算方式：总页面请求数/去重总人数

|  |
| --- |
| drop table if exists dw\_avgpv\_user\_d;  create table if not exists dw\_avgpv\_user\_d(  day string,  avgpv string  ) ;  insert into table dw\_avgpv\_user\_d  select a.daystr as daystr,sum(a.visit\_count)/count(a.ip\_addr) as avgpv from (  select ip\_addr,daystr, count(1) visit\_count  from ods\_weblog\_detail  group by ip\_addr ,daystr) a  group by a.daystr; |

1. 其他模块的分析（此时未实现）

|  |
| --- |
| 1. 受访分析 2. 各页面统计 3. 热门页面分析 4. 访客分析 5. 独立访客 6. 每日新访客 7. 访客visit分析 8. 回头客/单次访客统计 9. 人均访问频次 10. 访客忠诚度分析 11. 访客活跃度分析 12. 关键路径转化率分析 |

1. 模块开发--结果导出

报表统计结果，由 Sqoop 从 Hive 表中导出到 MySQL

Sqoop 对 Hive 与 MySQL 之间数据互导的主要命令如下：

|  |
| --- |
| 1. 列出MySQL数据库中的所有表：   sqoop list-tables \  --connect jdbc:mysql://hadoop01:3306/weblog \  --username hadoop \  --password 123456 \   1. 将关系型数据的表结构复制到 hive 中   sqoop create-hive-table \  --connect jdbc:mysql://:hadoop02:3306/weblog \  --table access\_log \  --username hadoop \  --password 123456 \  --hive-table access\_log \   1. 从关系数据库导入文件到 hive 中   sqoop import \  --connect jdbc:mysql://:hadoop02:3306/weblog \  --username hadoop \  --password 123456 \  --table access\_log \  --hive-import \ #默认的会在hive中创建一张与MySQL数据库中表名相同的表   1. 将 Hive 表的数据导出到 MySQL   #注意如果从hive中导出到MySQL 必须在之前MySQL有这张表  sqoop export \  --connect jdbc:mysql://:hadoop02:3306/weblog \  --username hadoop \  --password 123456 \  --table access\_log \  --export-dir /user/hadoop/warehouse/weblog/access\_log \  --input-fields-terminated-by '\001' |

1. 模块开发--工作流调度

将整个项目的数据处理过程，从数据采集到数据分析，再到结果数据的导出，一系列的任务 分割成若干个 azkaban 的 job 单元，然后调度执行。

1. 数据收集
2. 数据预处理
3. 数据分析
4. 数据导出
5. 数据展示

在企业的数据分析系统中，前端展现工具有很多：

1. 独立部署专门系统的方式：以 Business Objects(BO,Crystal Report),Heperion(Brio),Cognos 等 国外产品为代表的，它们的服务器是单独部署的，与应用程序之间通过某种协议沟通信息。
2. 有 WEB 程序展现方式：通过独立的或者嵌入式的 java web 系统来读取报表统计结果，以 网页的形式对结果进行展现，如，100%纯 Java 的润乾报表

**本项目用户行为日志分析项目采用自己开发 web 程序展现的方式**：

* Web 展现程序采用的技术框架：Jquery + Echarts + SpringMVC + Spring + MyBatis + MySQL
* 展现流程：
  + 使用 SSM 从 MySQL 中读取要展现的数据。
  + 使用 JSON 格式将读取到的数据返回给页面。
  + 在页面上用 Echarts 对 JSON 解析并形成图表。

1. 结果描述
2. 项目后的分析和期望