一、 课程计划

目录

二、 网站流量日志数据分析系统

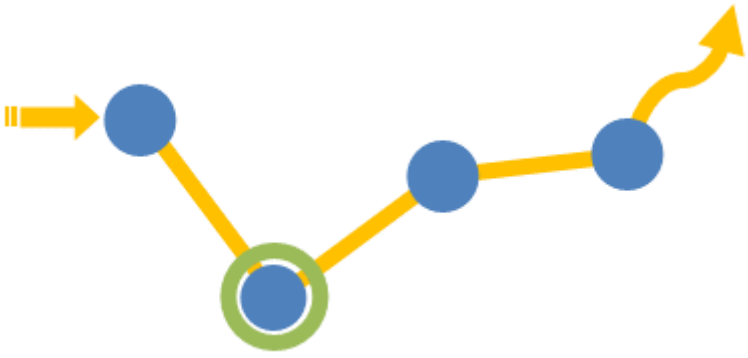
1. 点击流数据模型

1.1. 点击流概念

点击流（Click Stream）是指用户在网站上持续访问的轨迹。这个概念更注重用户浏览网站的整个流程。用户对网站的每次访问包含了一系列的点击动作行为，这些点击行为数据就构成了点击流数据（Click Stream Data），它代表了用户浏览网站的整个流程。

点击流和网站日志是两个不同的概念，点击流是从用户的角度出发，注重用户浏览网站的整个流程；而网站日志是面向整个站点，它包含了用户行为数据、服务器响应数据等众多日志信息，我们通过对网站日志的分析可以获得用户的点击流数据。

网站是由多个网页（Page）构成，当用户在访问多个网页时，网页与网页之间是靠 Referrers 参数来标识上级网页来源。由此，可以确定网页被依次访问的顺序，当然也可以通过时间来标识访问的次序。其次，用户对网站的每次访问，可视作是一次会话（Session），在网站日志中将会用不同的 Sessionid 来唯一标识每次会话。如果把 Page 视为“点”的话，那么我们可以很容易的把 Session 描绘成一条“线”，也就是用户的点击流数据轨迹曲线。



图：点击流概念模型

1.2. 点击流模型生成

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。点击数据在数据建模时存在两张模型表 Pageviews 和 visits，例如：

原始访问日志表

时间戳	IP 地址	请求 URL	Referal	响应吗	……
2012-01-01 12:31:12	101.0.0.1	/a/…	somesite.com	200	
2012-01-01 12:31:16	201.0.0.2	/a/…	-	200	
2012-01-01 12:33:06	101.0.0.2	/b/…	baidu.com	200	
2012-01-01 15:16:39	234.0.0.3	/c/…	google.com	304	
2012-01-01 15:17:11	101.0.0.1	/d/…	/c/…	404	

页面点击流模型 Pageviews 表

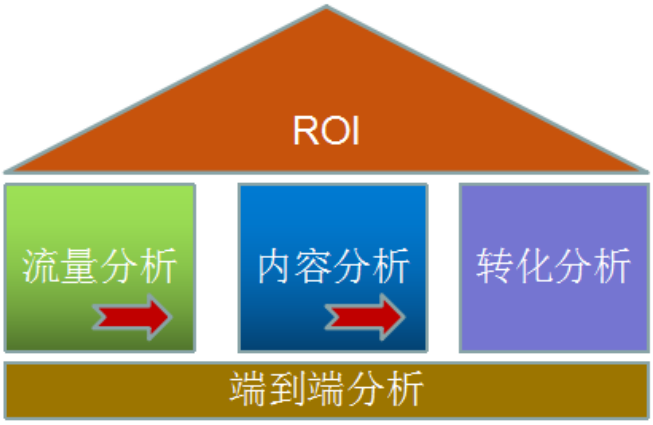
Session	IP 地址	时间	访问页面 URL	停留时长	第几步
S001	101.0.0.1	2012-01-01 12:31:12	/a/….	30	1
S002	201.0.0.2	2012-01-01 12:31:16	/a/….	10	1
S002	201.0.0.2	2012-01-01 12:33:06	/b/….	110	2
S002	201.0.0.2	2012-01-01 12:35:06	/e/….	30	3
S003	201.0.0.2	2012-01-01 15:35:06	/a/….	20	1

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	referal
S001	2012-01-01 12:31:12	2012-01-01 12:31:12	/a/…	/a/…	1	101.0.0.1	somesite.com
S002	2012-01-01 12:31:16	2012-01-01 12:35:06	/a/…	/e/…	3	201.0.0.2	-
S003	2012-01-01 12:35:42	2012-01-01 12:35:42	/c/…	/c/…	1	234.0.0.3	baidu.com
S003	2012-01-01 15:16:39	2012-01-01 15:19:23	/c/…	/e/…	3	101.0.0.1	google.com
……	……	……	……	……	……	……	……

2． 如何进行网站流量分析

流量分析整体来说是一个内涵非常丰富的体系，整体过程是一个金字塔结构：

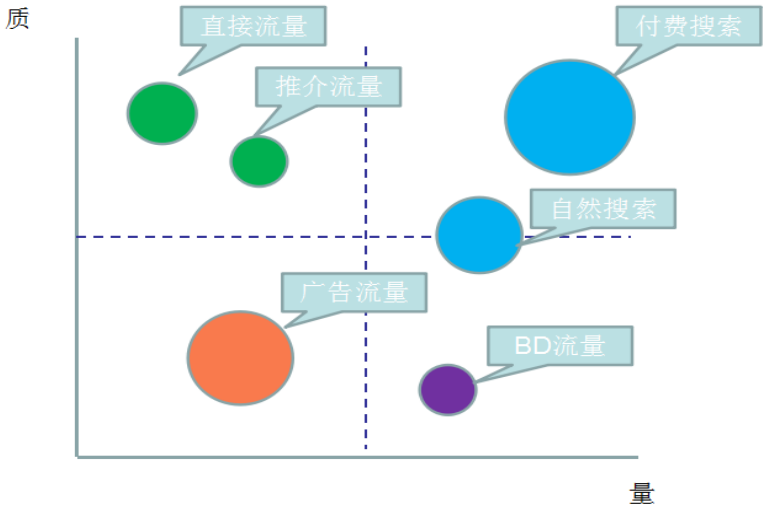


金字塔的顶部是网站的目标：投资回报率（ROI）。

2.1. 网站流量分析模型举例

网站流量质量分析（流量分析）

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。

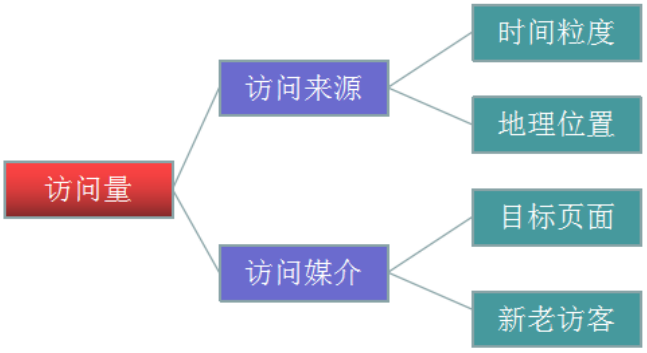


X 轴代表量，指网站获得的访问量。Y 轴代表质，指可以促进网站目标的事件次数（比如商品浏览、注册、购买等行为）。圆圈大小表示获得流量的成本。

BD 流量是指商务拓展流量。一般指的是互联网经过运营或者竞价排名等方式，从外部拉来的流量。比如电商网站在百度上花钱来竞价排名，产生的流量就是 BD 流量的一部分。

网站流量多维度细分（流量分析）

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。



网站内容及导航分析（内容分析）

对于所有网站来说，页面都可以被划分为三个类别：

导航页、功能页、内容页

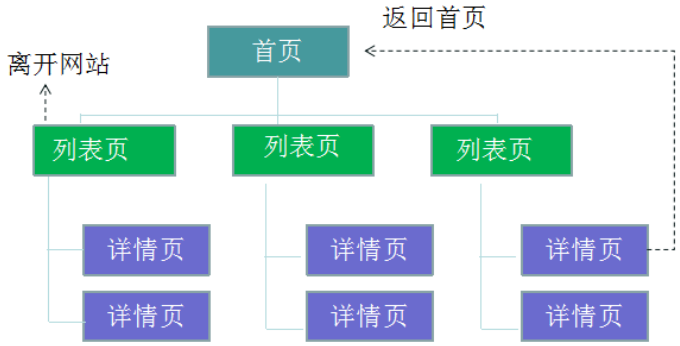
导航页的目的是引导访问者找到信息，功能页的目的是帮助访问者完成特定任务，内容页的目的是向访问者展示信息并帮助访问者进行 决策。

首页和列表页都是典型的导航页；

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，

而产品详情页、新闻和文章页都是典型的内容页。

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：



第一个问题：访问者从导航页（首页）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。

第二个问题：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。

网站转化以及漏斗分析（转化分析）

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：

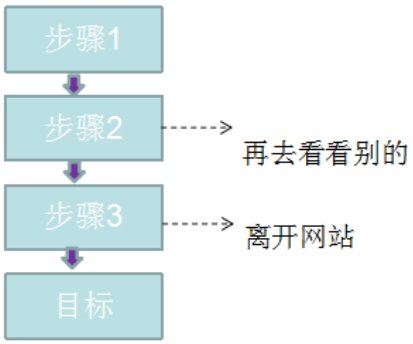
访问者的流失和迷失

- 阻力的流失

造成流失的原因很多，如：

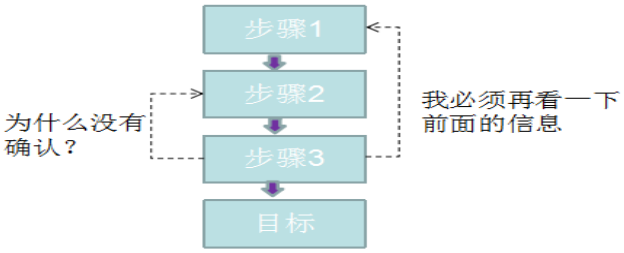
不恰当的商品或活动推荐

对支付环节中专业名词的解释、帮助信息等内容不当



- 迷失

造成迷失的主要原因是转化流量设计不合理，访问者在特定阶段得不到需要的信息，并且不能根据现有的信息作出决策，比如在线购买演唱会门票，直到支付也没看到在线选座的提示，这时候就很可能会产生迷失，返回查看。



总之，网站数据分析是一门内容非常丰富的学科，本课程中主要关注网站流量分析过程中的技术运用，更多关于网站数据分析的业务知识可学习文档首页推荐的资料。

2.2. 流量分析常见分类

指标是网站分析的基础，用来记录和衡量访问者在网站自的各种行为。比如我们经常说的流量就是一个网站指标，它是用来衡量网站获得的访问量。在进行流量分析之前，我们先来了解一些常见的指标。

骨灰级指标

IP：1 天之内，访问网站的不重复 IP 数。一天内相同 IP 地址多次访问网站只被计算 1 次。曾经 IP 指标可以用来表示用户访问身份，目前则更多的用来获取访问者的地理位置信息。

PageView 浏览量: 即通常说的 PV 值，用户每打开 1 个网站页面，记录 1 个 PV。用户多次打开同一页面 PV 累计多次。通俗解释就是页面被加载的总次数。

Unique PageView: 1 天之内，访问网站的不重复用户数（以浏览器 cookie 为依据），一天内同一访客多次访问网站只被计算 1 次。

基础级指标

访问次数：访客从进入网站到离开网站的一系列活动记为一次访问，也称会话(session),1 次访问(会话)可能包含多个 PV。

网站停留时间：访问者在网站上花费的时间。

页面停留时间：访问者在某个特定页面或某组网页上所花费的时间。

复合级指标

人均浏览页数：平均每个独立访客产生的 PV。人均浏览页数=浏览次数/独立访客。体现网站对访客的吸引程度。

跳出率:指某一范围内单页访问次数或访问者与总访问次数的百分比。其中跳出指单页访问或访问者的次数，即在一次访问中访问者进入网站后只访问了一个页面就离开的数量。

退出率:指某一范围内退出的访问者与综合访问量的百分比。其中退出指访问者离开网站的次数，通常是基于某个范围的。

有了上述这些指标之后，就能结合业务进行各种不同角度的分类分析，主要是以下几大方面：

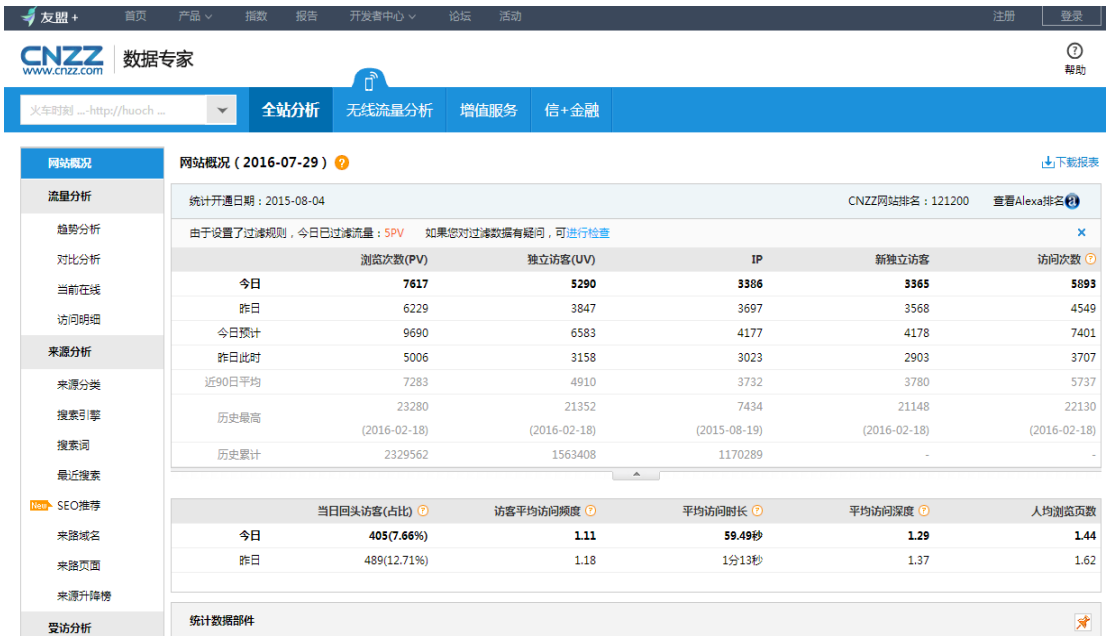
基础分析（PV, IP, UV）

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。



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。

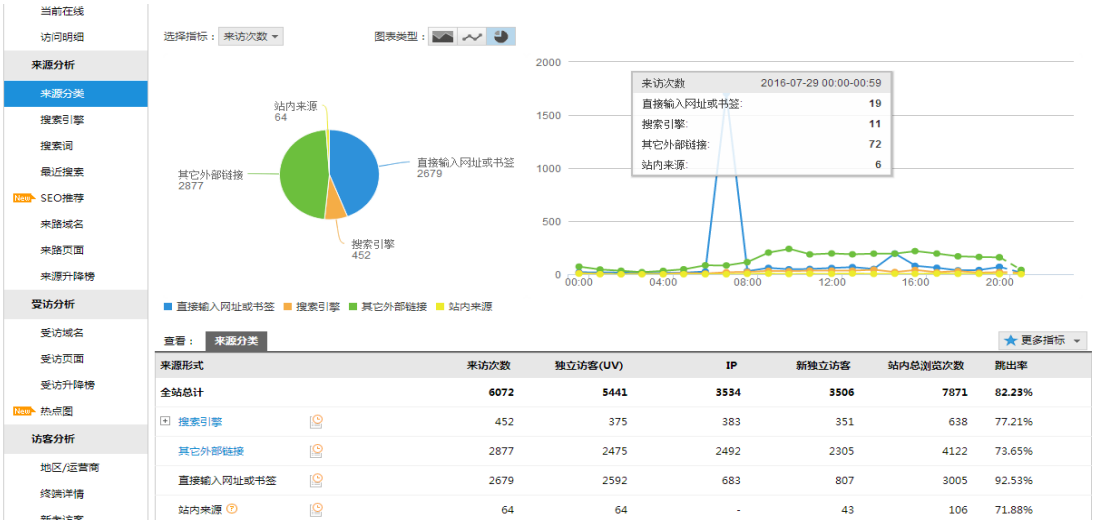
搜索词：提供访客通过搜索引擎进入网站所使用的搜索词，以及各搜索词引入流量的特征和分布。帮助用户了解各搜索词引入流量的质量，进而了解访客的兴趣关注点、网站与访客兴趣点的匹配度，为优化 SEO（搜索引擎优化）方案及 SEM（搜索引擎营销）提词方案提供详细依据。

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 TOP10000 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。



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。

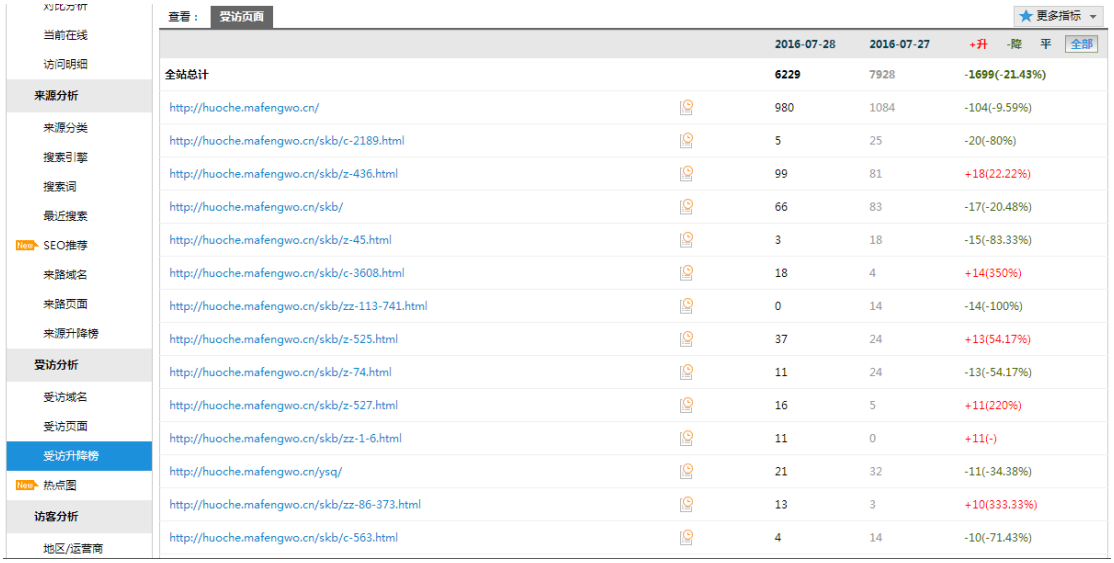
受访页面：提供访客对网站中各个页面的访问情况。 站内入口页面为访客进入网站时浏览的第一个页面，如果入口页面的跳出率较高则需要关注并优化；站内出口页面为访客访问网站的最后一个页面，对于离开率较高的页面需要关注并优化。

受访升降榜：提供开通统计后任意两日的 TOP10000 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。



访客分析

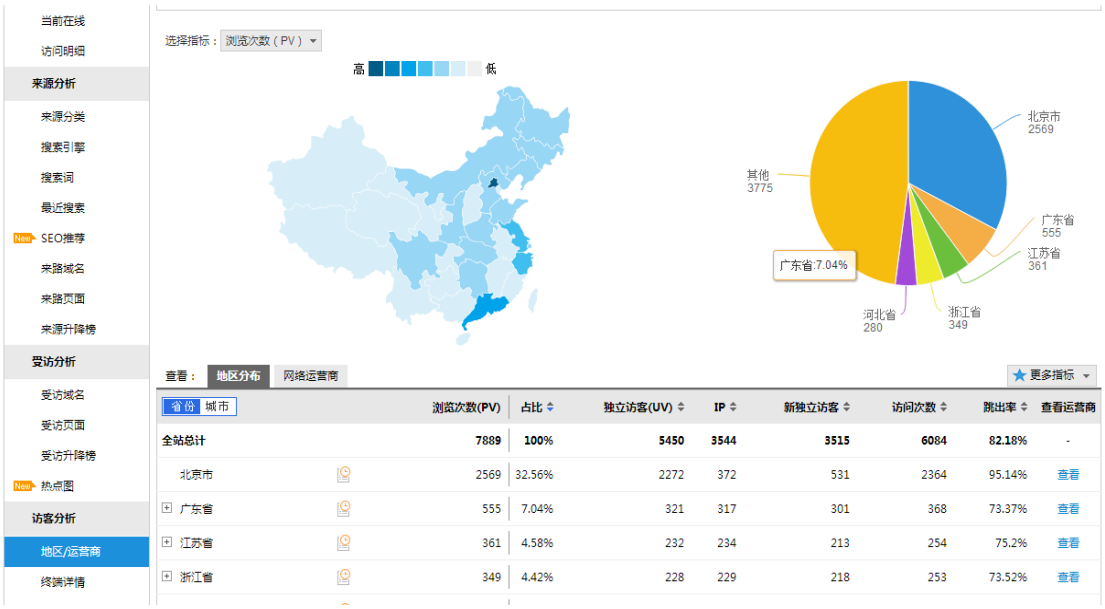
地区运营商：提供各地区访客、各网络运营商访客的访问情况分布。 地方网站、下载站等与地域性、网络链路等结合较为紧密的网站，可以参考此功能数据，合理优化推广运营方案。

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。



转化路径分析

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、下载、购买。

目标示例：

- 获得用户目标：在线注册、创建账号等。
- 咨询目标：咨询、留言、电话等。
- 互动目标：视频播放、加入购物车、分享等。
- 收入目标：在线订单、付款等。

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。

转化类型：

- 页面

- 事件

三、 整体技术流程及架构

1. 数据处理流程

网站流量日志数据分析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。有以下几个大的步骤：

- **数据采集**

数据采集概念，目前行业会有两种解释：一是数据从无到有的过程（web 服务器打印的日志、自定义采集的日志等）叫做数据采集；另一方面也有把通过使用 Flume 等工具把数据采集到指定位置的这个过程叫做数据采集。

关于具体含义要结合语境具体分析，明白语境中具体含义即可。

- **数据预处理**

通过 mapreduce 程序对采集到的原始日志数据进行预处理，比如清洗，格式整理，滤除脏数据等，并且梳理成点击流模型数据。

- **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。

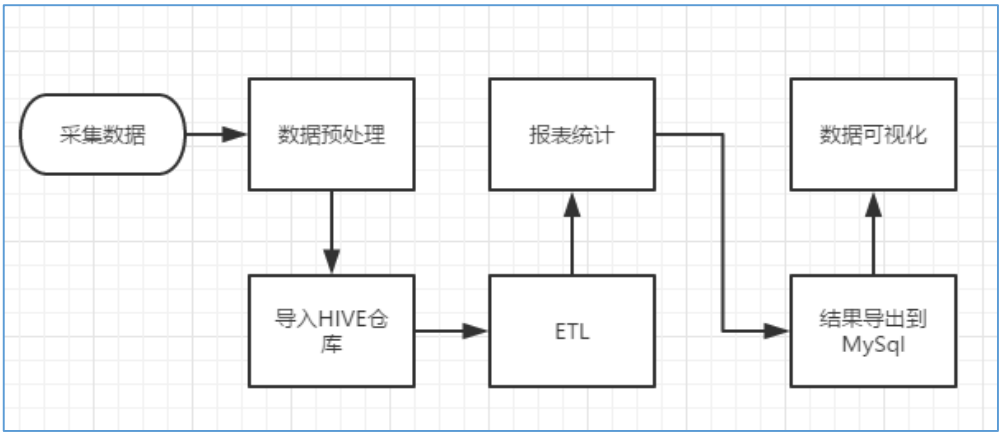
- **数据分析**

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数据可视化，一般通过图表进行展示。

2. 系统的架构



相对于传统的 BI 数据处理，流程几乎差不多，但是因为是处理大数据，所以流程中各环节所使用的技术则跟传统 BI 完全不同：

数据采集：定制开发采集程序，或使用开源框架 Flume

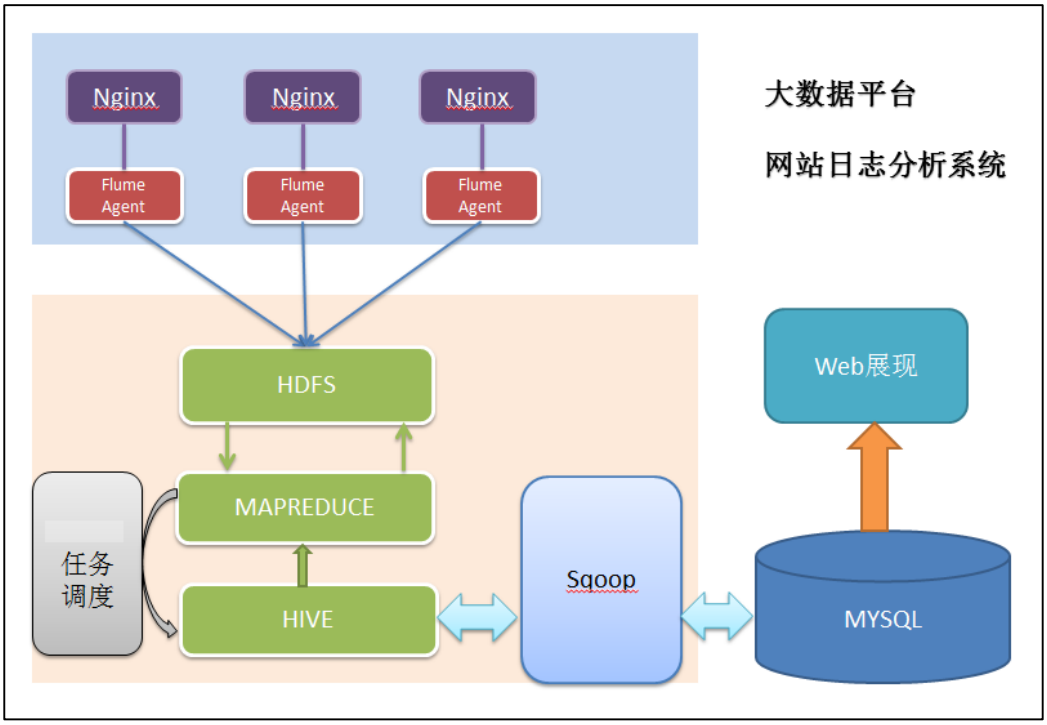
数据预处理：定制开发 mapreduce 程序运行于 hadoop 集群

数据仓库技术：基于 hadoop 之上的 Hive

数据导出：基于 hadoop 的 sqoop 数据导入导出工具

数据可视化：定制开发 web 程序(echarts)

整个过程的流程调度：hadoop 生态圈中的 azkaban 工具



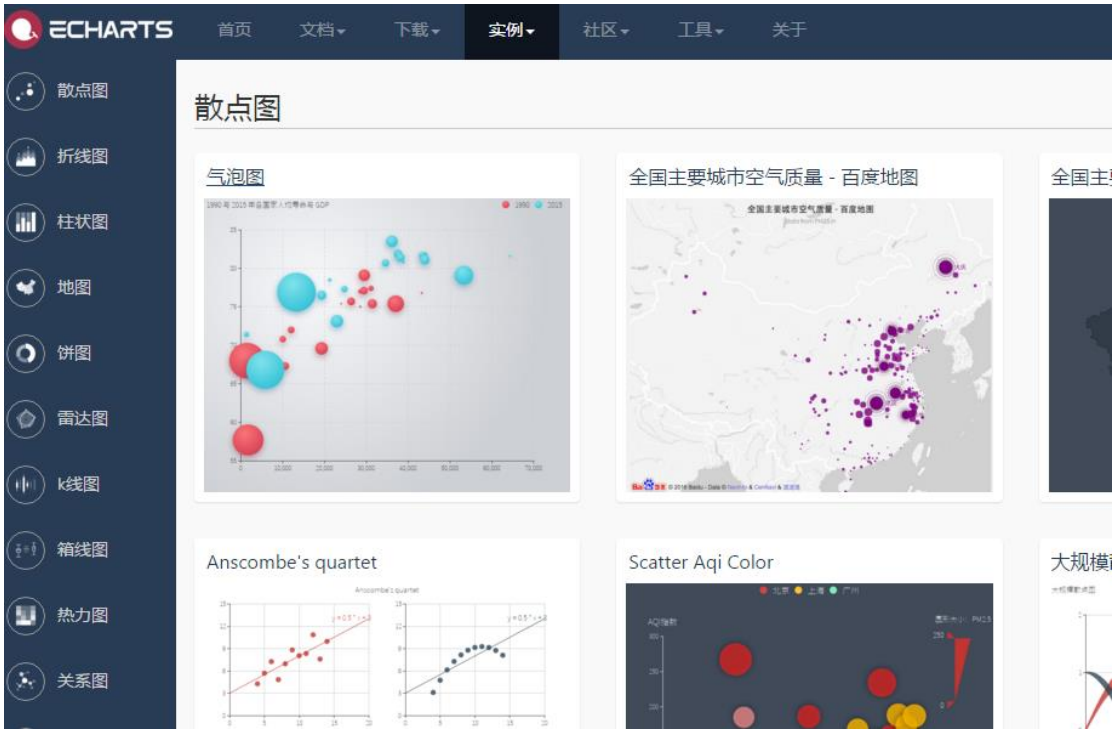
其中，需要强调的是：

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。

3. 数据展现

数据展现的目的是将分析所得的数据进行可视化，以便运营决策人员能更方便地获取数据，更快更简单地理解数据。

市面上有许多开源的数据可视化软件、工具。比如 Echarts.



四、 模块开发----数据采集

1. 需求

在网站 web 流量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。

2. Flume 日志采集系统

2.1. Flume 采集

Flume 采集系统的搭建相对简单：

1、在服务器上部署 agent 节点，修改配置文件

2、启动 agent 节点，将采集到的数据汇聚到指定的 HDFS 目录中

针对 nginx 日志生成场景，如果通过 flume（1.6）收集，无论是 Spooling Directory Source 和 Exec Source 均不能满足动态实时收集的需求，在当前 flume1.7 稳定版本中，提供了一个非常好用的 TaildirSource，使用这个 source，可以监控一个目录，并且使用正则表达式匹配该目录中的文件名进行实时收集。

核心配置如下：

```
a1.sources = r1
a1.sources.r1.type = TAILDIR
a1.sources.r1.channels = c1
a1.sources.r1.positionFile = /var/log/flume/taildir_position.json
a1.sources.r1.filegroups = f1 f2
a1.sources.r1.filegroups.f1 = /var/log/test1/example.log
a1.sources.r1.filegroups.f2 = /var/log/test2/.*log.*
```

filegroups:指定 filegroups，可以有多个，以空格分隔；（TailSource 可以同时监控 tail 多个目录中的文件）

positionFile:配置检查点文件的路径，检查点文件会以 json 格式保存已经 tail 文件

的位置，解决了断点不能续传的缺陷。

filegroups.<filegroupName>：配置每个 filegroup 的文件绝对路径，文件名可以用正则表达式匹配

通过以上配置，就可以监控文件内容的增加和文件的增加。产生和所配置的文件名正则表达式不匹配的文件，则不会被 tail。

2.2. 数据内容样例

```
58.215.204.118 - - [18/Sep/2013:06:51:35 +0000] "GET /wp-includes/js/jquery/jquery.js?ver=1.10.2 HTTP/1.1" 304 0 "http://blog.fens.me/nodejs-socketio-chat/" "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：/wp-includes/js/jquery/jquery.js?ver=1.10.2

6、请求所用协议：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-socketio-chat/

10、访客所用浏览器：Mozilla/5.0 (Windows NT 5.1; rv:23.0) Gecko/20100101 Firefox/23.0

五、 模块开发----数据预处理

1. 主要目的

过滤“不合规”数据，清洗无意义的数据

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(不同栏目 path)的基础数据。

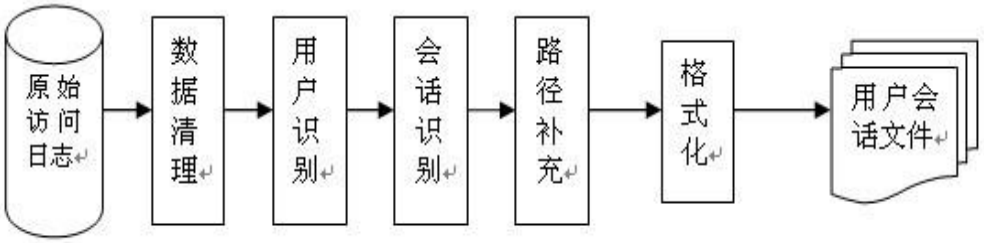


图 2.4 数据预处理过程

2. 实现方式

开发一个 mr 程序 WeblogPreProcess(*内容太长，见工程代码*)

```
public class WeblogPreProcess {
    static class WeblogPreProcessMapper extends Mapper<LongWritable, Text, Text, NullWritable> {
        Text k = new Text();
        NullWritable v = NullWritable.get();
        @Override
        protected void map(LongWritable key, Text value, Context context) throws IOException,
InterruptedException {
            String line = value.toString();
            WebLogBean webLogBean = WebLogParser.parser(line);
//          WebLogBean productWebLog = WebLogParser.parser2(line);
//          WebLogBean bbsWebLog = WebLogParser.parser3(line);
//          WebLogBean cuxiaoBean = WebLogParser.parser4(line);
            if (!webLogBean.isValid())
                return;
            k.set(webLogBean.toString());
            context.write(k, v);
//          k.set(productWebLog);
//          context.write(k, v);
```

```
        }
    }
    public static void main(String[] args) throws Exception {

        Configuration conf = new Configuration();
        Job job = Job.getInstance(conf);
        job.setJarByClass(WeblogPreProcess.class);
        job.setMapperClass(WeblogPreProcessMapper.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(NullWritable.class);
        FileInputFormat.setInputPaths(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        job.waitForCompletion(true);

    }
}
```

- 运行 mr 对数据进行预处理

```
hadoop jar weblog.jar    cn.itcast.bigdata.hive.mr.WeblogPreProcess /weblog/input /weblog/preout
```

3. 点击流模型数据梳理

由于大量的指标统计从点击流模型中更容易得出，所以在预处理阶段，可以使用 mr 程序来生成点击流模型的数据。

3.1. 点击流模型 pageviews 表

Pageviews 表模型数据生成，详细见：`ClickStreamPageView.java`

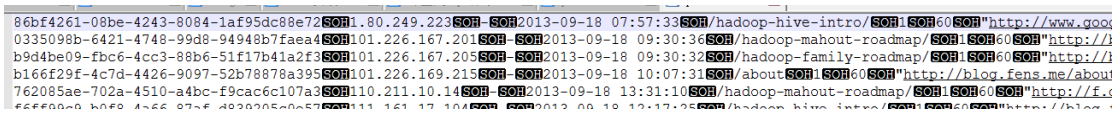
```
    });

    /**
     * 以下逻辑为：从有序bean中分辨出各次visit，并对一次visit中所访问的page按顺序标号step
     * 核心思想：
     * 就是比较相邻两条记录中的时间差，如果时间差<30分钟，则该两条记录属于同一个session
     * 否则，就属于不同的session
     *
     */

    int step = 1;
    String session = UUID.randomUUID().toString();
    for (int i = 0; i < beans.size(); i++) {
            WebLogBean bean = beans.get(i);
            // 如果仅有1条数据，则直接输出
            if (1 == beans.size()) {

                    // 设置默认停留时长为60s
```

此时程序的输入数据源就是上一步骤我们预处理完的数据。经过此不处理完成之后的数据格式为：



3.2. 点击流模型 visit 信息表

注：“一次访问”=“N 次连续请求”

直接从原始数据中用 hql 语法得出每个人的“次”访问信息比较困难，可先用 mapreduce 程序分析原始数据得出“次”信息数据，然后再用 hql 进行更多维度统计

用 MR 程序从 pageviews 数据中，梳理出每一次 visit 的起止时间、页面信息
详细代码见工程：`ClickStreamVisit.java`

```
/**
 * 输入数据：pageviews模型结果数据
 * 从pageviews模型结果数据中进一步梳理出visit模型
 * sessionid  start-time   out-time   start-page   out-page   pagecounts  ......
 *
 * @author
 *
 */
public class ClickStreamVisit {

        // 以session作为key，发送数据到reducer
        static class ClickStreamVisitMapper extends Mapper<LongWritable, Text, Text, PageV

                PageViewsBean pvBean = new PageViewsBean();
                Text k = new Text();

                @Override
```

六、 工作流调度器

1. 工作流调度系统产生背景

一个完整的数据分析系统通常都是由大量任务单元组成：

shell 脚本程序，java 程序，mapreduce 程序、hive 脚本等。

各个任务单元之间存在时间先后依赖关系。

为了很好地组织起这样的复杂执行计划，需要一个工作流调度系统来调度执行；

2. 工作流调度实现方式

简单的任务调度：

直接使用 linux 的 crontab 来定义,但是缺点也是比较明显，无法设置依赖。

复杂的任务调度：

自主开发调度平台

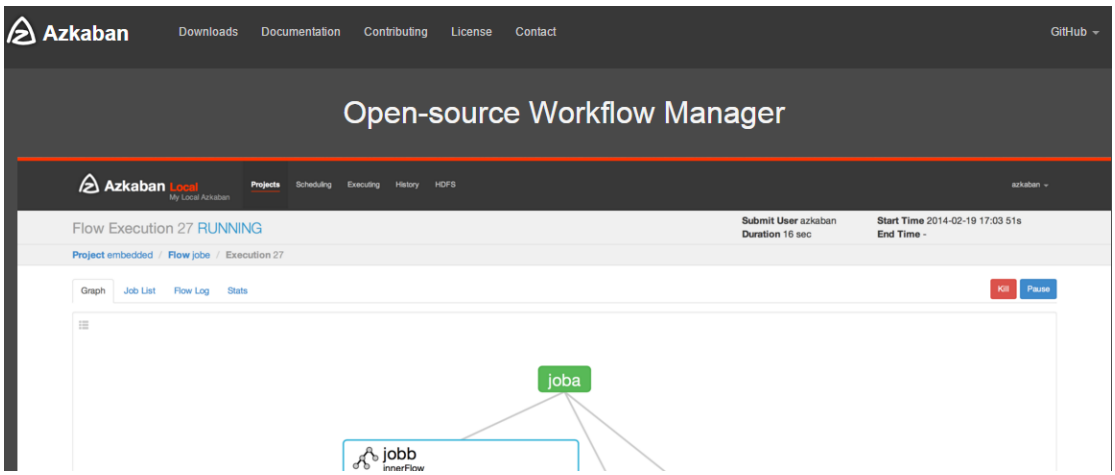
使用开源调度系统，比如 azkaban、ooize、Zeus 等。

其中知名度比较高的是 Apache Oozie，但是其配置工作流的过程是编写大量的 XML 配置，而且代码复杂度比较高，不易于二次开发。

3. Azkaban 调度器

3.1. Azkaban 介绍

Azkaban 是由 Linkedin 公司推出的一个批量工作流任务调度器，用于在一个工作流内以一个特定的顺序运行一组工作和流程。Azkaban 使用 job 配置文件建立任务之间的依赖关系，并提供一个易于使用的 web 用户界面维护和跟踪你的工作流。

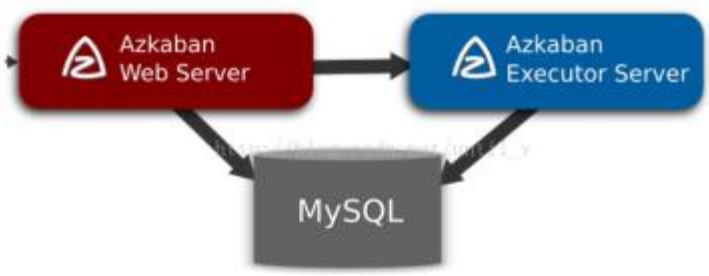


Azkaban 功能特点：

- 提供功能清晰，简单易用的 Web UI 界面
- 提供 job 配置文件快速建立任务和任务之间的依赖关系
- 提供模块化和可插拔的插件机制，原生支持 command、Java、Hive、Pig、Hadoop
- 基于 Java 开发，代码结构清晰，易于二次开发

3.2. Azkaban 安装部署

Azkaban 的组成如下：



mysql 服务器:用于存储项目、日志或者执行计划之类的信息

web 服务器:使用 Jetty 对外提供 web 服务，使用户可以通过 web 页面方便管理

executor 服务器:负责具体的工作流的提交、执行。

Azkaban 有两种部署方式：solo server mode 和 cluster server mode。

- solo server mode(单机模式)：该模式中 webServer 和 executorServer 运行在同一个进程中，进程名是 AzkabanSingleServer。可以使用自带的 H2 数据库或者配置 mysql 数据。该模式适用于小规模的使用。
- cluster server mode(集群模式)：该模式使用 MySQL 数据库，webServer 和 executorServer 运行在不同进程中，该模式适用于大规模应用。

其实在单机模式中，AzkabanSingleServer 进程只是把 AzkabanWebServer 和 AzkabanExecutorServer 合到一起启动而已。

准备工作

Azkaban Web 服务器

azkaban-web-server-2.5.0.tar.gz

Azkaban 执行服务器

azkaban-executor-server-2.5.0.tar.gz

MySQL

本文档中默认已安装好 mysql 服务器。

下载地址:http://azkaban.github.io/downloads.html

上传安装包

将安装包上传到集群,最好上传到安装 hive、sqoop 的机器上,方便命令的执行。

新建 azkaban 目录,用于存放 azkaban 运行程序。

azkaban web 服务器安装

解压 azkaban-web-server-2.5.0.tar.gz

命令: tar -zxvf azkaban-web-server-2.5.0.tar.gz

将解压后的 azkaban-web-server-2.5.0 移动到 azkaban 目录中,并重新命

名 webserver

命令: mv azkaban-web-server-2.5.0 ../azkaban

cd ../azkaban

mv azkaban-web-server-2.5.0 server

azkaban 执行服器安装

解压 azkaban-executor-server-2.5.0.tar.gz

命令:tar -zxvf azkaban-executor-server-2.5.0.tar.gz

将解压后的 azkaban-executor-server-2.5.0 移动到 azkaban 目录中,并重新命名 executor

命令:mv azkaban-executor-server-2.5.0 ../azkaban

cd ../azkaban

mv azkaban-executor-server-2.5.0 executor

azkaban 脚本导入

解压: azkaban-sql-script-2.5.0.tar.gz

命令:tar -zxvf azkaban-sql-script-2.5.0.tar.gz

将解压后的 mysql 脚本,导入到 mysql 中:

进入 mysql

mysql> create database azkaban;

mysql> use azkaban;

Database changed

mysql> source /home/hadoop/azkaban-2.5.0/create-all-sql-2.5.0.sql;

创建 SSL 配置（https）

命令: keytool -keystore keystore -alias jetty -genkey -keyalg RSA

运行此命令后,会提示输入当前生成 keystor 的密码及相应信息,输入的密码请劳记,信息如下:

输入 keystore 密码:

再次输入新密码:

您的名字与姓氏是什么?

[Unknown]:

您的组织单位名称是什么?

[Unknown]：

您的组织名称是什么？

[Unknown]：

您所在的城市或区域名称是什么？

[Unknown]：

您所在的州或省份名称是什么？

[Unknown]：

该单位的两字母国家代码是什么

[Unknown]： CN

CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=CN 正确吗？

[否]： y

输入<jetty>的主密码

（如果和 keystore 密码相同，按回车）：

再次输入新密码:

完成上述工作后，将在当前目录生成 keystore 证书文件，将 keystore 拷贝到 azkaban web 服务器根目录中.如:cp keystore azkaban/webserver

配置文件

注：先配置好服务器节点上的时区

先生成时区配置文件 Asia/Shanghai，用交互式命令 tzselect 即可

拷贝该时区文件，覆盖系统本地时区配置

cp /usr/share/zoneinfo/Asia/Shanghai /etc/localtime

- azkaban web 服务器配置

进入 azkaban web 服务器安装目录 conf 目录

修改 azkaban.properties 文件

```
#Azkaban Personalization Settings
azkaban.name=Test                    #服务器 UI 名称,用于服务器上方显示的名字
azkaban.label=My Local Azkaban       #描述
azkaban.color=#FF3601                #UI 颜色
azkaban.default.servlet.path=/index  #
web.resource.dir=web/                #默认根 web 目录
default.timezone.id=Asia/Shanghai    #默认时区,已改为亚洲/上海 默认为美国
```

```
#Azkaban UserManager class
user.manager.class=azkaban.user.XmlUserManager   #用户权限管理默认类
user.manager.xml.file=conf/azkaban-users.xml      #用户配置,具体配置参加下文

#Loader for projects
executor.global.properties=conf/global.properties   # global 配置文件所在位置
azkaban.project.dir=projects                         #

database.type=mysql                                  #数据库类型
mysql.port=3306                                      #端口号
mysql.host=hadoop03                                  #数据库连接 IP
mysql.database=azkaban                               #数据库实例名
mysql.user=root                                      #数据库用户名
mysql.password=root                                  #数据库密码
mysql.numconnections=100                             #最大连接数

# Velocity dev mode
velocity.dev.mode=false
# Jetty 服务器属性.
jetty.maxThreads=25                                  #最大线程数
jetty.ssl.port=8443                                  #Jetty SSL 端口
jetty.port=8081                                      #Jetty 端口
jetty.keystore=keystore                              #SSL 文件名
jetty.password=123456                                #SSL 文件密码
jetty.keypassword=123456                             #Jetty 主密码 与 keystore 文件相同
jetty.truststore=keystore                            #SSL 文件名
jetty.trustpassword=123456                           # SSL 文件密码

# 执行服务器属性
executor.port=12321                                  #执行服务器端口

# 邮件设置
mail.sender=xxxxxxxx@163.com                         #发送邮箱
mail.host=smtp.163.com                               #发送邮箱 smtp 地址
mail.user=xxxxxxxx                                   #发送邮件时显示的名称
mail.password=**********                             #邮箱密码
job.failure.email=xxxxxxxx@163.com                   #任务失败时发送邮件的地址
```

```
job.success.email=xxxxxxxx@163.com                    #任务成功时发送邮件的地址
lockdown.create.projects=false                         #
cache.directory=cache                                  #缓存目录
```

- azkaban 执行服务器配置

进入执行服务器安装目录 conf，修改 azkaban.properties

vi azkaban.properties

```
#Azkaban
default.timezone.id=Asia/Shanghai                      #时区

# Azkaban JobTypes 插件配置
azkaban.jobtype.plugin.dir=plugins/jobtypes            #jobtype 插件所在位置

#Loader for projects
executor.global.properties=conf/global.properties
azkaban.project.dir=projects

#数据库设置
database.type=mysql                                    #数据库类型(目前只支持 mysql)
mysql.port=3306                                        #数据库端口号
mysql.host=192.168.20.200                              #数据库 IP 地址
mysql.database=azkaban                                 #数据库实例名
mysql.user=azkaban                                     #数据库用户名
mysql.password=oracle                                  #数据库密码
mysql.numconnections=100                               #最大连接数

# 执行服务器配置
executor.maxThreads=50                                 #最大线程数
executor.port=12321                                    #端口号(如修改,请与 web 服务中一致)
executor.flow.threads=30                               #线程数
```

- 用户配置

进入 azkaban web 服务器 conf 目录,修改 azkaban-users.xml

vi azkaban-users.xml 增加 管理员用户

```
<azkaban-users>
      <user username="azkaban" password="azkaban" roles="admin" groups="azkaban" />
      <user username="metrics" password="metrics" roles="metrics"/>
     <user username="admin" password="admin" roles="admin,metrics" />
      <role name="admin" permissions="ADMIN" />
      <role name="metrics" permissions="METRICS"/>
</azkaban-users>
```

启动

web 服务器

在 azkaban web 服务器目录下执行启动命令

bin/azkaban-web-start.sh

注:在 web 服务器根目录运行

执行服务器

在执行服务器目录下执行启动命令

bin/azkaban-executor-start.sh ./

注:只能要执行服务器根目录运行

启动完成后,在浏览器(建议使用谷歌浏览器)中输入 `https://服务器 IP 地址:8443` ,即可访问 azkaban 服务了.在登录中输入刚才新的户用名及密码,点击 login.

3.3. Azkaban 实战

Command 类型单一 job 示例

- 创建 job 描述文件

  vi command.job

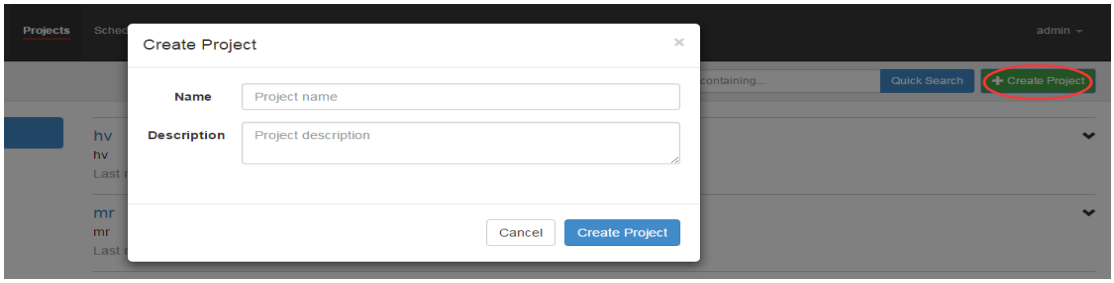
```
#command.job
type=command
command=echo 'hello'
```

- 将 job 资源文件打包成 zip 文件

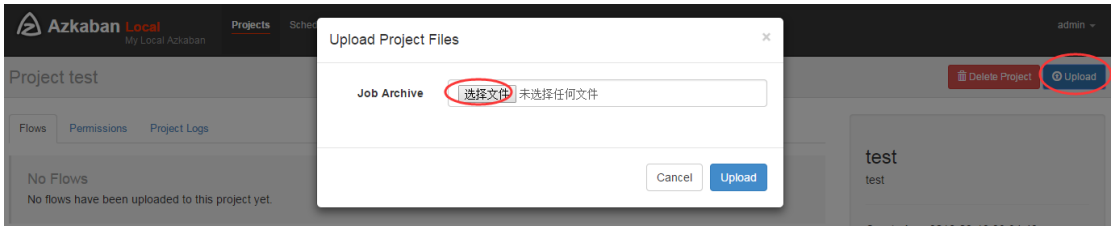
  zip command.job

- 通过 azkaban 的 web 管理平台创建 project 并上传 job 压缩包

  首先创建 Project



上传 zip 包



- 启动执行该 job

Command 类型多 job 工作流 flow

- 创建有依赖关系的多个 job 描述

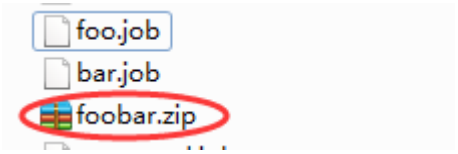
第一个 job：foo.job

```
# foo.job
type=command
command=echo foo
```

第二个 job：bar.job 依赖 foo.job

```
# bar.job
type=command
dependencies=foo
command=echo bar
```

- 将所有 job 资源文件打到一个 zip 包中



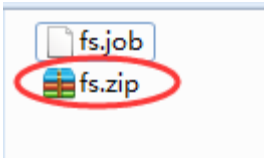
- 在 azkaban 的 web 管理界面创建工程并上传 zip 包
- 启动工作流 flow

HDFS 操作任务

- 创建 job 描述文件

```
# fs.job
type=command
command=/home/hadoop/apps/hadoop-2.6.1/bin/hadoop fs -mkdir /azaz
```

- 将 job 资源文件打包成 zip 文件



- 通过 azkaban 的 web 管理平台创建 project 并上传 job 压缩包
- 启动执行该 job

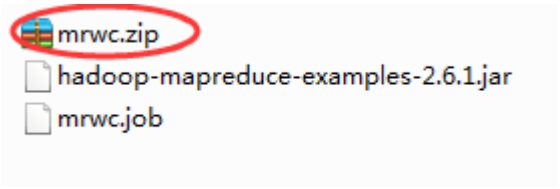
MAPREDUCE 任务

mr 任务依然可以使用 command 的 job 类型来执行

- 创建 job 描述文件，及 mr 程序 jar 包（示例中直接使用 hadoop 自带的 example jar）

```
# mrwc.job
type=command
command=/home/hadoop/apps/hadoop-2.6.1/bin/hadoop jar hadoop-mapreduce-examples-2.6.1.jar wordcount /wordcount/input /wordcount/azout
```

- 将所有 job 资源文件打到一个 zip 包中



- 在 azkaban 的 web 管理界面创建工程并上传 zip 包
- 启动 job

HIVE 脚本任务

- 创建 job 描述文件和 hive 脚本

Hive 脚本： test.sql

```
use default;
drop table aztest;
create table aztest(id int,name string) row format delimited fields terminated by ',';
load data inpath '/aztest/hiveinput' into table aztest;
create table azres as select * from aztest;
insert overwrite directory '/aztest/hiveoutput' select count(1) from aztest;
```

Job 描述文件：hivef.job

```
# hivef.job
type=command
command=/home/hadoop/apps/hive/bin/hive -f 'test.sql'
```

- 将所有 job 资源文件打到一个 zip 包中创建工程并上传 zip 包,启动 job