今天我们来讲讲整个的架构入手，先知道每一个工具是处于一个什么样子的角色，然后接下来我们再去做实践会有更好的理解。

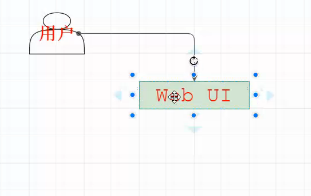
我们一起来看一下，对于一个完整的线上架构是一个怎么弄出来的，那么一个好的系统好的产品它必然是要打造一个生态，然后让生态里面的各个模块能够彼此之间有效的运转是吧？能够健康的存活下去，大概是这么一个思路，所以我们先从这个角度来看一下大致一个的系统它是应该存在哪个必要的环节，那么首先你得有用户是吧？（如下图1所示）

图1：



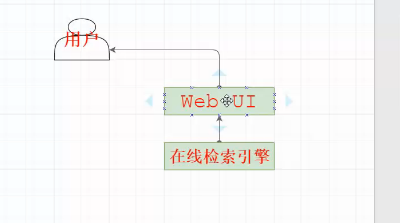
那么有了用户之后，相当于我们请求的流量就已经进来了对不对？那比如说我们想象我们大家都是一个普通的网名，然后大家来上网，大家上网首先面对的是什么呢？面对的是一个浏览器或者app等等一些应用，那么这些东西我们统一把它叫为一个web ui，这个UI页面相当于就是前端应用，你会发现因为每一种平台它的实现方式不一样，那比如你去开发一个安卓系统的app，它会用的语言和用的思想而且它的呈现方式可能都不太一样是不是？那相当于有些平台你直接写完一套代码之后可以进行统一部署，但是对于有一些UI页面或者有一些应用的话你想实现统一的服务，那你必须要对不同流量上或者不同设备上背后的一个用户你进行区别开发，但目的就是要把不同端的流量统一起来。

所以第一步这个网名得先访问到你这个UI页面对吧？（如下图2）

图2：  


那谁给这个前端UI做支撑呢？那么就是需要后端在线检索引擎来进行支撑（如下图3所示）

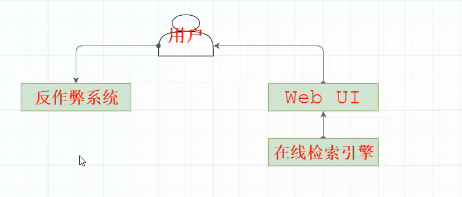
图3：



那什么是在线检索引擎呢？也就是一个在线服务，相当于就是借助一个快速的框架然后做一个类似于小网页的形式，然后一个可以给大家展示的这个一个HTML这么一个网页或者一个接口是吧？然后比如说我们中文分词的时候我们输入一个句子，然后它会把这个句子切开不同的一个词语然后做一个展现是不是？那为了一个大家都有的一个用户交互的话，那我希望通过一个URL把这个句子的输入，然后切完词之后，把这个词然后在页面中能够返回回来，这样的话直观感受能够更直接对不对？如果webUI是一个页面的话，那么在线引擎是它的一个后台。

那在线引擎通常是什么呢？通常都是一个webserver对不对？那这个时候用户访问你的网站，其实在你这个后端的检索引擎这个webserver会留一定的log数据，这个log数据是一个用户访问网站的时间等等的数据，用户点击一次页面，那么后台log就会记录一条记录，那接下来假设说我这个网民访问了这个页面了，那还需要做哪些事情？对于网站来说是不是所有人都应该把这个网页都暴露给它对不对？那么对谁不暴露我这个网页功能呢？那肯定是对一些恶意攻击等等，所以这里面会涉及到一个很重要的反作弊系统。（如下图4所示）

图4：



这个反作弊系统比如说对于一个视频或者音乐的网站都会对这个网站本身做一个数据加密，就为了防止一些恶意下载和一些恶意攻击，你比如说短时间内大量的并发量的请求，类似于这种恶意攻击，如果你做的这个平台防御不好的话会很容易的瞬时就垮掉，那么这些突增量很大的这些特征是通常借助一些系统或者整个平台的一些漏洞，那么高级的话就会制造一些不存在的用户或者是假的用户然后在短时间内会在同一个ip段同一个地址等等这些数据比较集中的一些倾向的表现，那你就可以通过它请求的数据就有一定的作弊特征，如果你能拿到这些有作弊的特征的话，你就可以进行一个反作弊的识别。

那这个反作弊对于这种视频网站或者应用网站是很有必要的，但是对于这种很高系统的来说，这反作弊就更为重要了，因为你这个平台会牵扯了多方利益，特别是一些广告主的一些资产。

那比如说什么是作弊流量呢？就是说来自同行的恶意点击，比如说你一个卖苹果手机的，那如果我也是一个卖苹果手机的，然后我就为了这个转化降低，然后让你这个付出成本提高，那我就不断的去影响你这个服务的时间，而且你还拿不到任何的有效转化。所以同行的恶意点击也有一些个性特征，然后也可以去提取出来。

当然也有一些水军作弊对不对？就是说可以花很少的成本去雇一些水军然后在短时间内制造一些舆论或者是去进行一些个恶意破坏对吧？这也是一种作弊方式

当然还有一种形式就是第三方的寄生依赖，比如说你是淘宝网站，然后我是一个很小的一个电商平台，但是我自己是不卖任何的货物的，我也没有商品描述没有那么多的商家等等对不对？那怎办呢？虽然我有自己的好像跟淘宝没有任何关系的一个独立的一个电商网站，其实这个网站表现出的页面数据其实都是抓取了淘宝的一些数据，相当于是淘宝的数据为这些第三方公司进行一些数据的提供，对于这种其实都是有破坏作用的，那本身自己是拿不到收益的，然后第三方平台还要恶意的承载那么多的流量，而且你也无法控制这种第三方平台是不是有这种瞬时间的一些刷量的一些操作对不对？所以这也可以通过反作弊去屏蔽一部分，当然这反作弊不可能做的很完美，它总会遗漏一些东西，所以这种系统也是不断的升级的。

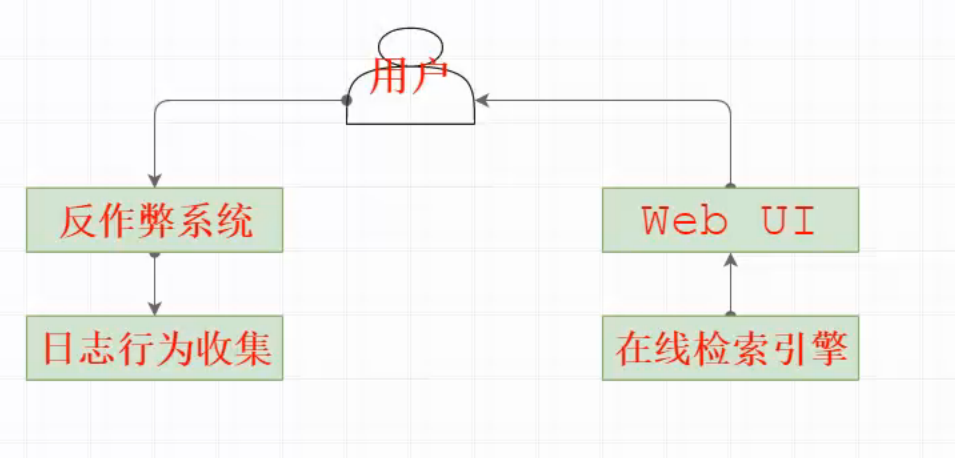
然后还有一种另外一些高级的东西，反作弊有几种方式，第一种就是基于规则，统计一些共性特征，进行恶意流量识别，进行屏蔽。第二种就是基于模型，模型就是机器学习，它的目的就是你能不能在这个当前的流量下，你能够明确的找到哪些流量是一个恶意的行为。

比如说对恶意行为打一个1，然后打0这种都是一些正常行为，那恶意行为你是不是可以从大批量的数据里面能不能得到一些特征，这些特征主要是集中在用户角度，比如说地理位置，ip信息，行为时长等等，当然正常行为也要抽取这些特征，那这些特征纬度特别高，然后你仅从这个数据的背后我们通过人眼是很难辨别出哪些流量是恶意的对吧？那你需要对这些数据进行一个模型寻列，然后得到一个模型，然后在线识别反作弊，大概是这么一个思路。

所以目前反作弊基于两种规则，一个是基于规则，一个是基于模型这两个。

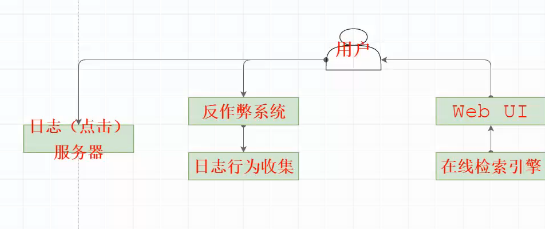
假如这个时候经过反作弊后的流量都是一些比较真实的流量了，那接下来就需要对这批流量进行一个日志行为收集，那么日志行为收集这一块是对用户的行为进行收集（如下图5所示）

图5：



就是比如说用户打开了一个视频页面，然后这个视频相当于是一个物品然后呈现给了当前这个用户对不对？所以相当于这个物品对用户来说它是做了一个展现，如果物品展现不符合当前用户的品味，那可能马上就把这个页面给关了对不对？但是不管怎么样呢，起码这个物品还是在用户面前它有了一次展现，那既然它打开了这个页面，它必然在这个在线检索引擎里面会留下一条展现日志，那么这个页面给用户展现的这个物品他很喜欢，于是这个用户就点击了这个物品，那他点击了这个物品之后，这个用户会往远程的一个日志点击服务器上做一次请求（如下图6所示）

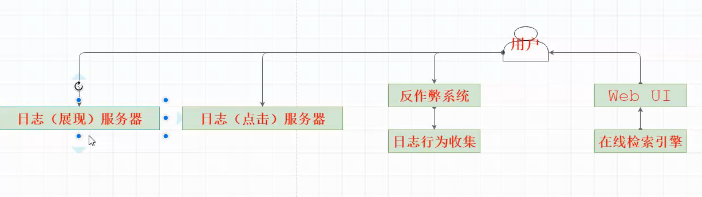
图6：



只不过是这个服务器不会给你用户提供服务，它只是用户请求一下而已，那用户也不需要你这个服务器给我带来什么信息，那它目的也就是说我就想在你这个服务器上打一次log日志，然后打这个日志的目的是为了记录我是哪天几分几秒我点了这个物品，然后这个物品的名字是什么等等还有一些其他的信息。

日志（展现）服务器和日志（点击）服务器是对等关系，这个服务跟在线引擎应该是部署到不同的集群上去，只不过是这个用户打开这个页面会在展现服务器上会留下展现日志，如果这个用户在点击服务器上就会留下点击日志（如下图7所示）

图7：



如果说这个用户在去点了这个物品之后，这个时间和物品给用户展示的这个时间，这两个时间一定是一样的吗？肯定不一样对不对？那不可能我这个物品马上展示出来，然后用户就马上点了是不是？那这个用户神了，哪能反应那么快

26分钟