22 | 大数据平台设计：如何用数据为用户创造价值？

2022-04-08 李智慧

讲述：李智慧

时长 10:40 大小 9.78M

特别说明：本文相关技术仅用于技术展示，具体实践中，数据收集和算法应用需要遵循国家个人信息保护法与信息安全法等有关法律制度。

你好，我是李智慧。

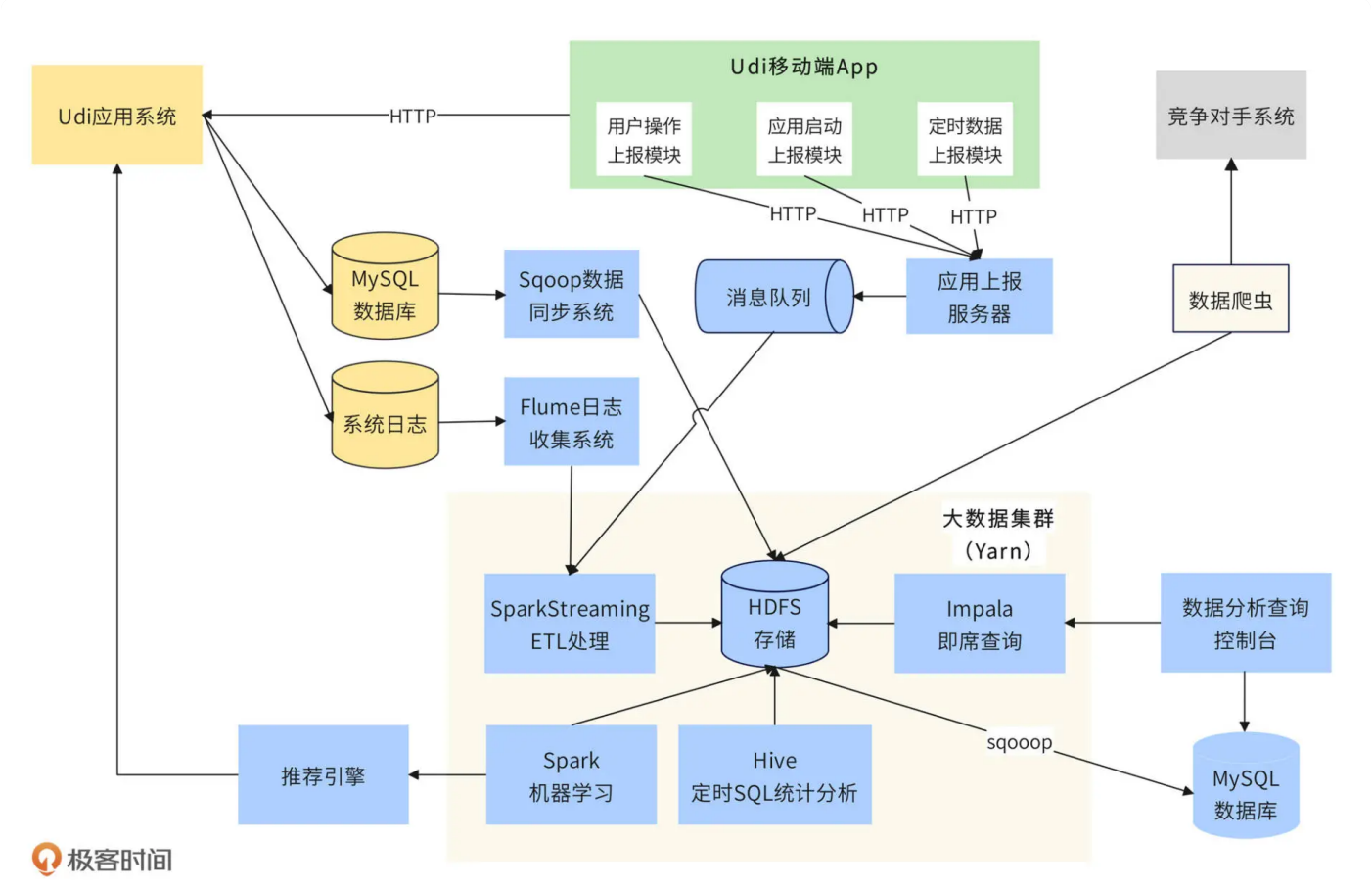
现在，业界普遍认为互联网创新已经进入下半场，依靠技术创新或者商业模式创新取得爆发性发展的机会越来越少。于是大家把目光转向精细化运营，主要手段就是依靠大数据技术，挖掘每个用户独特的商业价值，提供更具个性化的服务，以此来提升服务水平和营收能力，最终获得更强的市场竞争能力。

Udi 大数据平台的主要目标是根据用户的不同喜好，为其分配不同的车型，一方面改善用户体验，另一方面也增加平台营收。此外，如何为用户推荐最优的上车点和下车点，如何分析订单和营收波动，如何发现潜在的高风险用户等等，也需要依赖大数据平台。

大数据技术不同于我们前面设计的高并发案例，高并发案例虽然也要处理海量用户的请求，但是每个用户请求都是独立的，计算与存储也是每个用户独立进行的。而大数据技术则要将这些海量的用户数据进行关联计算，因此，适用于高并发架构的各种分布式技术并不能解决大数据的问题。

Udi 大数据平台设计

根据 Udi 大数据应用场景的需求，需要将手机 App 端数据、数据库订单和用户数据、操作日志数据、网络爬虫爬取的竞争对手数据统一存储到大数据平台，并支持数据分析师、算法工程师提交各种 SQL 语句、机器学习算法进行大数据计算，并将计算结果存储或返回。Udi 大数据平台架构如下图：



大数据采集与导入

Udi 大数据平台整体可分为三个部分，第一个部分是大数据采集与导入。这一部分又可以分为 4 小个部分，App 端数据采集、系统日志导入、数据库导入、爬虫数据导入。

App 端除了业务功能模块，还需要包含几个数据埋点上报模块。App 启动的时候，应用启动上报模块会收集用户手机信息，比如手机型号、系统版本、手机上安装的应用列表等数据；App

运行期间，也会通过定时数据上报模块，每 5 秒上报一次数据，主要是用户当前地理位置数据；用户点击操作的时候，一方面会发送请求到 Udi 后端应用系统，一方面也会通过用户操作上报模块将请求数据以及其他一些更详细的参数发送给后端的应用上报服务器。

后端的应用上报服务器收到前端采集的数据后，发送给消息队列，SparkStreamin 从消息队列中消费消息，对数据进行清洗、格式化等 ETL 处理，并将数据写入到 HDFS 存储中。

Udi 后端应用系统在处理用户请求的过程中，会产生大量日志和数据，这些存储在日志系统和 MySQL 数据库中的数据也需要导入到大数据平台。Flume 日志收集系统会将 Udi 后端分布式集群中的日志收集起来，发送给 SparkStreaming 进行 ETL 处理，最后写入到 HDFS 中。而 MySQL 的数据则通过 Sqoop 数据同步系统直接导入到 HDFS 中。

除了以上这些 Udi 系统自己产生的数据，为了更好地应对市场竞争，Udi 还会通过网络爬虫从竞争对手的系统中爬取数据。需要注意的是，这里的爬虫不同于04讲中的爬虫，因为竞争对手不可能将订单预估价等敏感数据公开。因此，爬虫需要模拟成普通用户爬取数据，这些爬来的数据也会存储在 HDFS 中，供数据分析师和产品经理在优化定价策略时分析使用。

大数据计算

Udi 大数据平台的第二个部分是大数据计算。写入到 HDFS 中的数据，一方面供数据分析师进行统计分析，一方面供算法工程师进行机器学习。

数据分析师会通过两种方式分析数据。一种是通过交互命令进行即席查询，通常是一些较为简单的 SQL。分析师提交 SQL 后，在一个准实时、可接受的时间内返回查询结果，这个功能是通过 Impala 完成的。另外一种是定时 SQL 统计分析，通常是一些报表类统计，这些 SQL 一般比较复杂，需要关联多张表进行查询，耗时较长，通过 Hive 完成，每天夜间服务器空闲的时候定时执行。

算法工程师则开发各种 Spark 程序，基于 HDFS 中的数据，进行各种机器学习。

以上这些大数据计算组件，Hive、Spark、SparkStreaming、Impala 都部署在同一个大数据集群中，通过 Yarn 进行资源管理和调度执行。每台服务器既是 HDFS 的 DataNode 数据存储服务器，也是 Yarn 的 NodeManager 节点管理服务器，还是 Impala 的 Impalad 执行服务器。通过 Yarn 的调度执行，这些服务器上既可以执行 SparkStreaming 的 ETL 任务，也可以执行 Spark 机器学习任务，而执行 Hive 命令的时候，这些机器上运行的是 MapReduce 任务。

数据导出与应用

Udi 大数据平台的第三个部分是数据导出与应用。Hive 命令执行完成后，将结果数据写入到 HDFS 中，这样并不方便数据分析师或者管理人员查看报表数据。因此还需要用 Sqoop 将 HDFS 中的数据导出到 MySQL 中，然后通过数据分析查询控制台，以图表的方式查看数据。

而机器学习的计算结果则是一些学习模型或者画像数据，将这些数据推送给推荐引擎，由推荐引擎实时响应 Udi 系统的推荐请求。

大数据平台一方面是一个独立的系统，数据的存储和计算都在其内部完成。一方面又和应用系统有很多关联，数据需要来自应用系统，而计算的结果也需要给应用系统使用。上面的架构图中，属于大数据平台的组件我用蓝色标出，其他颜色代表非大数据平台组件或者系统。

Udi 大数据派单引擎设计

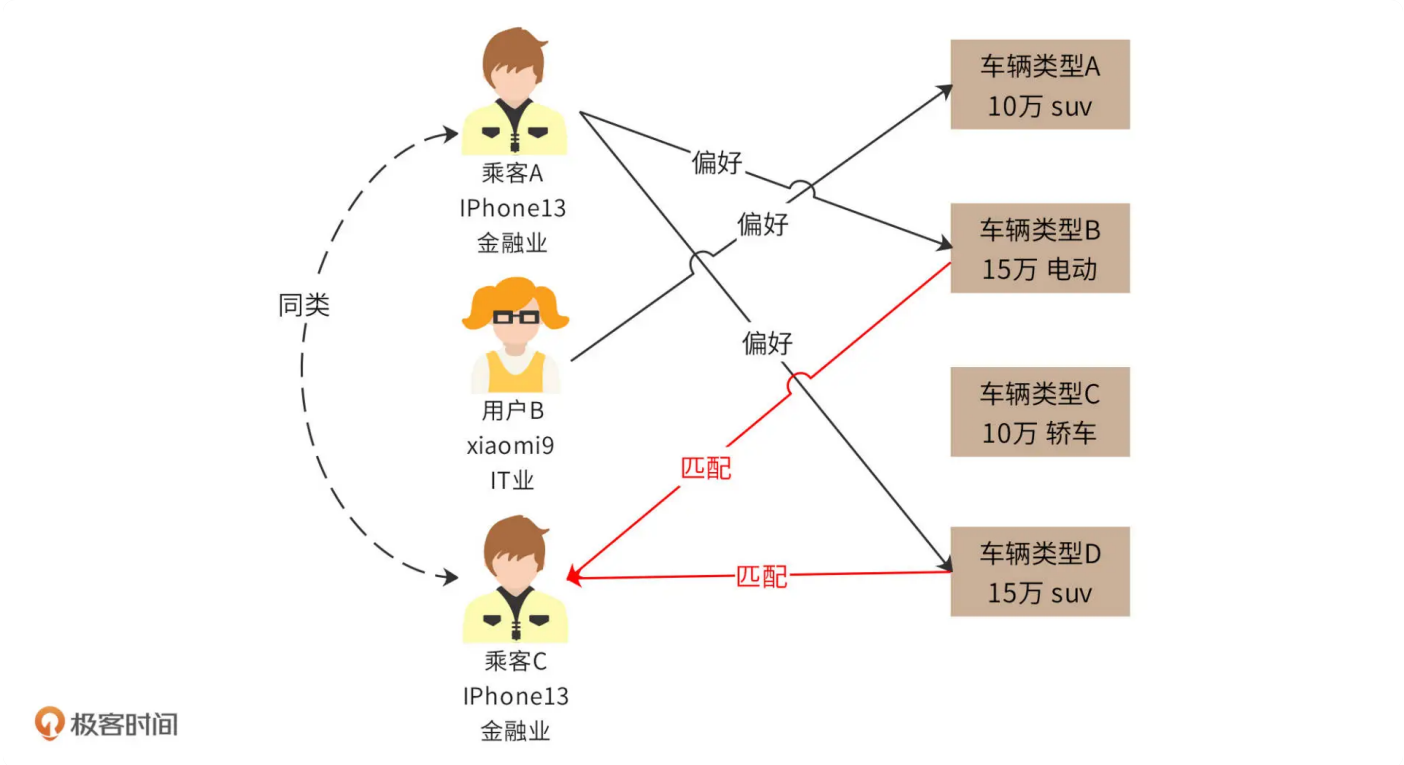
我们在第 20 讲讨论了 Udi 派单引擎，这个派单引擎并没有考虑乘客和车型的匹配关系。根据 Udi 的运营策略，车辆新旧程度、车辆等级与舒适程度、司机服务水平会影响到订单的价格。派单成功时，系统会根据不同车辆情况预估不同的订单价格并发送给乘客，但是有些乘客会因为预估价格太高而取消订单，而有些乘客则会因为车辆等级太低而取消订单，还有些乘客则会在上车后因为车辆太旧而给出差评。

Udi 需要利用大数据技术优化派单引擎，针对不同类别的乘客匹配尽可能合适的车辆。上面采集了乘客的手机型号及手机内安装应用列表，订单数据记录了乘客上下车地点，乘客评价以及订单取消原因记录了用户乘车偏好，车辆及司机数据记录了车辆级别和司机信息，这些数据最终都会同步到大数据平台。

我们将利用这些数据优化 Udi 派单引擎。根据用户画像、车辆画像、乘车偏好进行同类匹配。

基于乘客分类的匹配

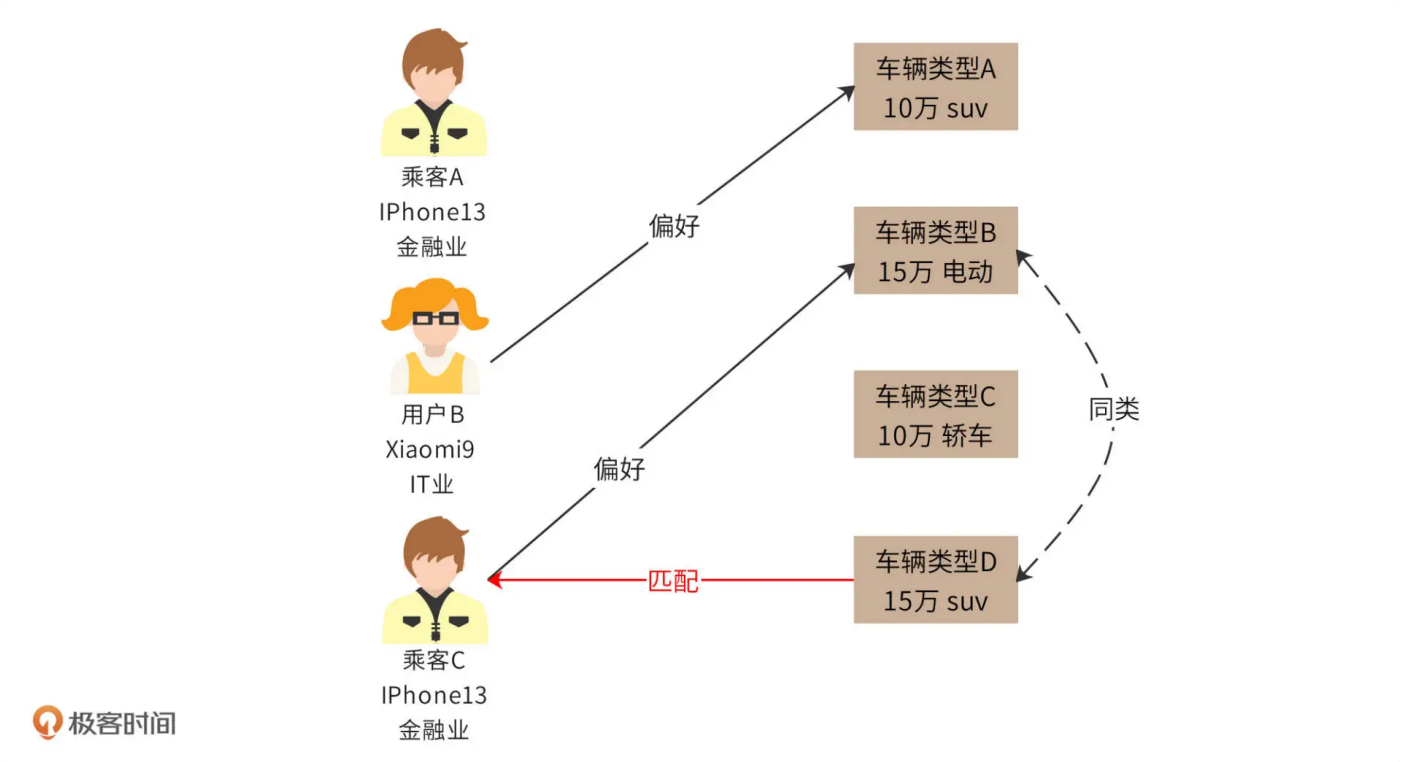
根据乘客的注册信息、App 端采集的乘客手机型号、手机内安装应用列表、常用上下车地点等，我们可以将乘客分类，然后根据同类乘客的乘车偏好，预测乘客的偏好并进行匹配。



比如根据数据分类，乘客 A 和乘客 C 是同类乘客，而乘客 A 偏好车辆类型 B 和 D。乘客 C 叫车的时候，那么派单系统会优先给他派单车辆类型 B 和 D。

基于车辆分类的匹配

事实上，我们可以直接根据车辆类型属性，对车辆类型进行再分类。比如通过机器学习统计分析，车辆类型 B 和 D 可以归为一类，那么如果乘客 C 偏好车辆类型 B，那么我们可以认为车辆类型 D 也匹配他。



使用推荐引擎对派单系统进行优化，为乘客分配更合适的车辆，前提是需要对用户和车辆进行分类与画像，想要完成这部分工作，我们可以在大数据平台的 Spark 机器学习模块通过聚类分析、分类算法、协同过滤算法，以及 Hive 统计分析模块进行数据处理，将分类后的数据推送给派单引擎去使用。

派单引擎在原有的最小化等车时间基础上，对派单进行调整，使车辆和乘客偏好更匹配，改善用户体验，也增加了平台营收。

小结

网约车是一个格外依赖大数据进行用户体验优化的应用。比如用户上车点，在一个几千平方米的 POI 区域内，乘客方便等车，司机不违章的地点可能只有一两个，这一两个点又可能在任何地图上都没有标示。这就意味着，司机和乘客需要通过电话沟通很久才知道对方说的上车点在哪里，然后要么乘客徒步几百米走过来，要么司机绕一大圈去接，给司机和乘客都造成很多麻烦，平台也会因此流失很多用户。

对于这种问题，电子地图应用的厂商需要派测绘人员现场标注这些点。而对于网约车平台，由于不停在上传司机的位置信息，只需要根据乘客最后的上车点进行聚类分析，就会发现该区域大部分乘客最后都是在某个点上车，这个点就是最佳上车点。也就是说，只需要最初的一批乘客忍受麻烦，他们的行为数据就可以被网约车平台用于机器学习和数据挖掘，并被用于优化用户体验。

网约车平台像这样依赖大数据的地方还有很多。所以，网约车平台需要尽可能获取、存储用户和司机的各种行为与业务数据，并基于这些数据不断进行分析、挖掘，寻找潜在的商业机会和用户体验优化。对于一个数亿用户规模的网约车平台，这些数据的规模是非常庞大的，因此需要一个强大、灵活的大数据平台才能完成数据的存储与计算。

思考题

在你的工作中，是否有涉及到大数据和机器学习，它们带来了哪些价值？

欢迎在评论区分享你的思考，我们共同进步。

> 【编辑温馨提示】4 月 10 日 12 点前，提交期中测试作业，有机会获得老师精心准备的奖励哦~

上一篇　21 | 网约车系统重构：如何用 DDD 重构网约车系统设计？

精选留言

写留言

> 由作者筛选后的优质留言将会公开显示，欢迎踊跃留言。