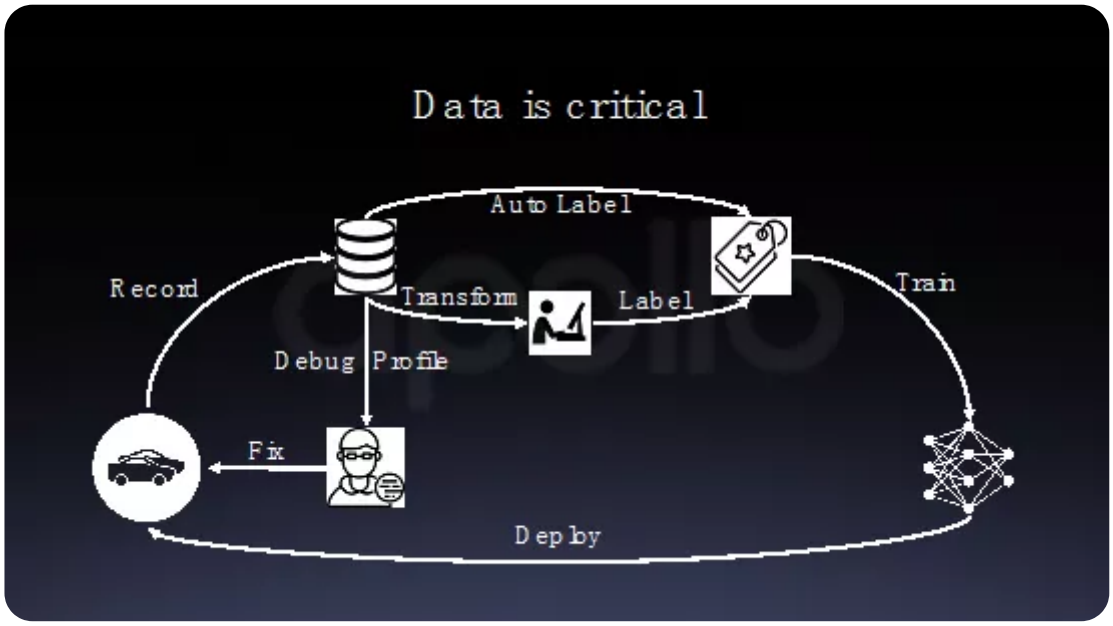
Apollo公开课丨Apollo Fuel 技术分享

我们都知道，数据在整个无人驾驶技术研发里处于非常重要的位置。它不是我们经常提到的定位、感知、预测、规划、控制、仿真等关键算法模块，但是它跟所有这些模块打交道。所以我们把这个项目叫做**Apollo Fuel**，因为**数据**对无人驾驶来说就像**燃料**对车一样驱动每个模块的发展。



▲数据闭环

这个是典型的**数据闭环**。最左边车子出去测试，录制数据回来，然后工程师对其中的问题进行**Debug**，**Profiling**，**修代码**，**部署**到车上继续测试。虽然无人驾驶是人工智能的集大成者，但是也免不了有很多这种Case by case的工作。

车子采回来的数据非常重要，它包含了非常多的量化信息。这应该是有史以来对现实世界最全面的数字化表示。物体变成了一个个的点，颜色，位置，速度等都数字化。从中可以得到大量的**标注数据**，供计算机学习。然后把模型部署到车上去，让它开得更好。

数据标注有两种，一种是**自动标注**，比如障碍车的轨迹预测，需要预测2秒后的状态，在录制的数据里可以确切地知道它2秒后的状态，这就是Ground truth。Apollo的数据管线可以做到全自动，从增量数据中抽取feature，训练，在仿真引擎里验证，然后部署。另一种是**人工标注**，需要把原始数据进行一些处理，得到标注公司接受的格式，付钱，然后拿到标注数据。比如图像、点云，很多需要人工标注。

数据模块面临的挑战

虽然数据很重要，但是做好数据并不容易，面临的挑战也很多。首先是**数据的录制**，如果用很多相机录制，每秒至少产生200M多的数据量，一小时下来就会产生接近1TB的数据量，录制占用太多的CPU资源。然后是**数据的传输**，数据量大之后，从车传到本地，或者本地传到云都会遇到各种各样问题。第三是**存储**，相对比较简单，因为现在有各种各样服务。最后是**数据处理**，数据量大之后，处理会有很大挑战。

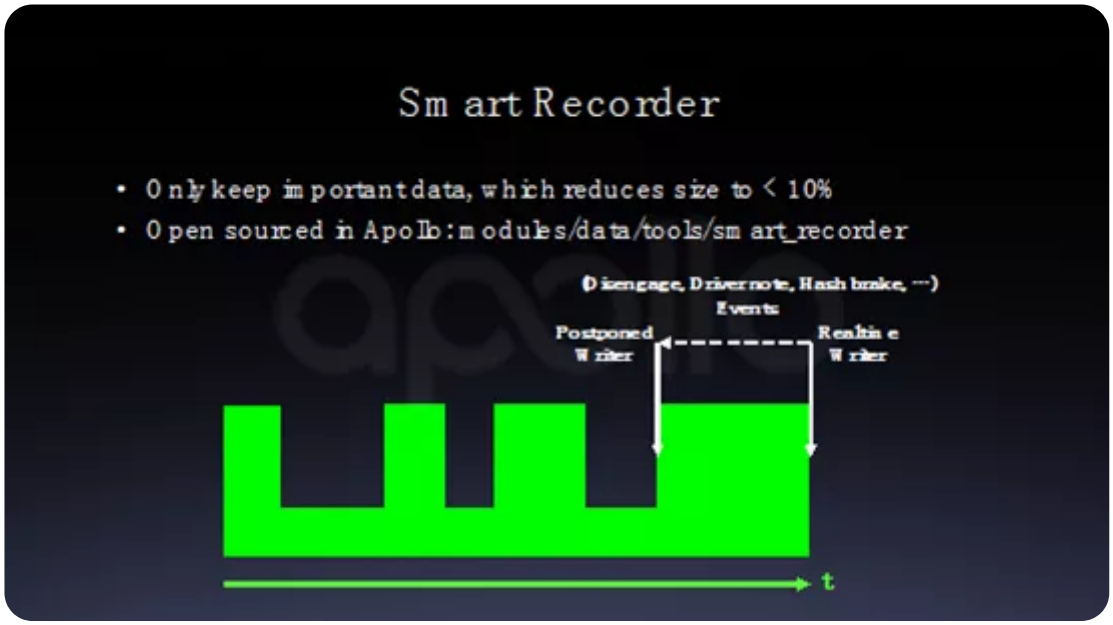
针对这些问题，我们提出了相应的解决方案，首先**Apollo5.0开源了一个叫Smart Recorder的录制模块**。在传送和存储方面，有**百度云**和其他一些云服务做支持，我们目前支持百度云。针对数据处理，我们提出了**Apollo Fuel**，各个算法模块对数据需求不一样，有的需要这几个topic，有的需要那几个topic，数据处理工具碎片化，Apollo Fuel把这些东西统一起来。

▲数据面临的挑战

Smart Recorder

Smart Recorder只保存最有价值的、最重要的数据，数据量可以减少到10%以下，究竟减少多少完全可以自己调节。其基本原理是**起两个写进程**，第一个写进程实时把所有数据录制下来，另外还有一

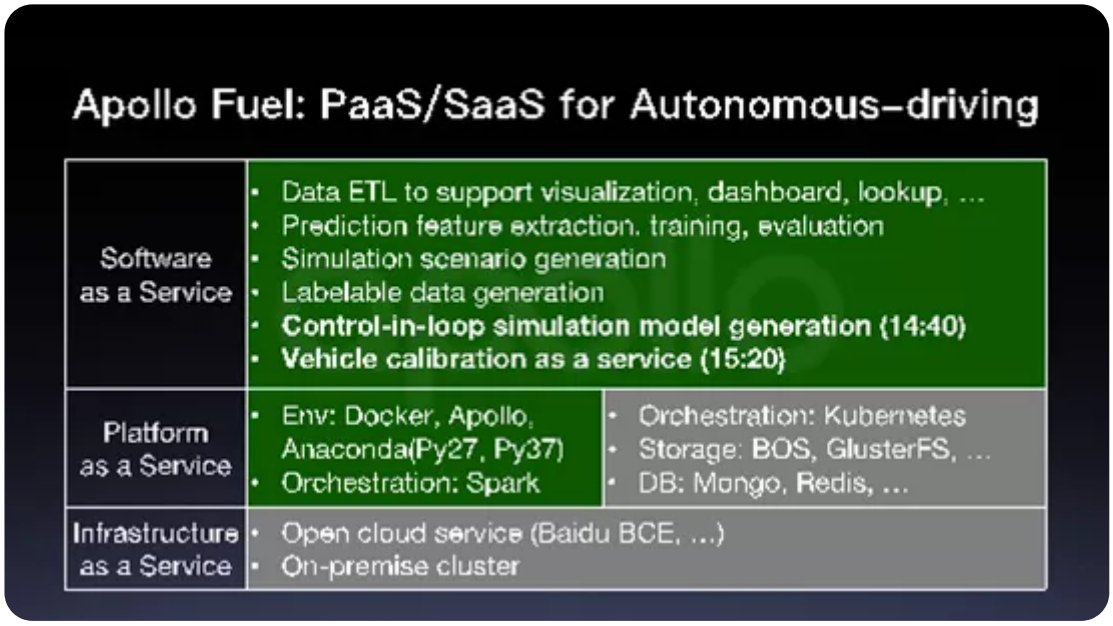
个后继的Writer，读取上一个Writer的数据，然后决定要全量数据还是只保留一些关键的小的数据，如图所示。



▲Smart Recorder

下面t表示时间轴，Real time Writer与Postponed Writer进程之间有一个时间间隔，比如30秒，后继写有30秒延迟，决定它之后的30秒数据要不要。当Realtime Writer发现一些感兴趣的事件，会通知后继Writer是否保存数据。以接管为例，Realtime writer发现一次接管，它会发一个消息给后继Writer，后继Writer知道30秒之后发生接管，将会完整记录接管发生时前面30秒以及后面10秒总共40秒的数据。如果没有感兴趣数据则只保留一些关键的小量数据。

此外，我们还支持别的情况，比如Apollo的**Driver note（驾驶员笔记）**，相当于司机通过界面发送消息，虽然没有发生任何问题，没有发生接管，但是司机觉得体感不好，用快捷键表示我现在体感不好，需要线下工程师看一下。这个note变成事件发给后继writer，把整个前30秒和后10秒数据记录下来，拿回来之后进行分析。



▲Apollo Fuel的全栈解决方案

上图为Apollo Fuel的全栈解决方案。**第一层是Iaas**，基础架构即服务，现在一般推荐用云服务，Iaas提供计算资源。当然数据量大之后为了节省开销以及减少运维工作，通常使用开放服务，例如百度云或者友商云。图中灰色部分是面向资源的，也是Apollo Fuel现在不做的。绿色部分面向任务，面向开发逻辑， Apollo Fuel更多关注绿色这一部分。

中间层是PaaS，平台即服务。我们并不会搭一个BOS或者mongo，而是直接由云服务提供。在环境这一端，我们有Docker，有Apollo镜像，Apollo镜像装了Anaconda，所以整个环境也是解决方案。有了PaaS之后，上一层算法团队需要知道PaaS这一层往上提供什么样保证才能实现什么样的业务。**第一是个提供存储级别服务**，我们直接借用云服务能力，比如说BOS。**第二是计算能力提供纵向可扩展性**，纵向可扩展性意思是说单个计算能力的扩展，比如计算密集型需要很多CPU，可以让这个机器非常强大，单个节点变得更加强大。现在百度云提供最强节点包含96个CPU，256GB内存，32TB磁盘。然后是横向可扩展能力，当任务很多，数据量非常大，需要做并发时，横向可扩展就决定并发度是多少，理论上横向是无限可扩展的，比如百度云可以增加100个节点， 100乘以上面纵向扩展节点，得到横向纵向可以扩展的资源。

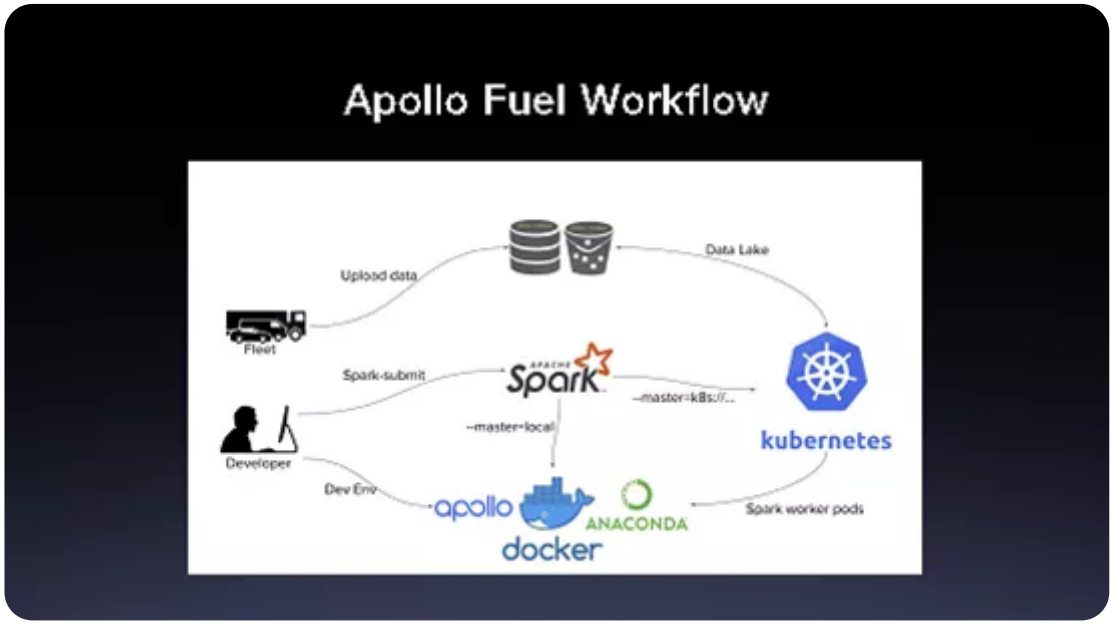
第三层是SaaS，软件即服务。各个工程和算法团队，有了下面的平台基础，就可以完全面向自己的业务来开发数据流水线。而且我们的代码都放在一起管理，有很多公共库，统一的代码评审、质量管理，尽量避免碎片化。比如举几个**应用例子**。

> - **数据团队**，会有一些ETL的工作，将录制的数据进行解析和索引，然后在我们的前端平台上提供可视化、统计、检索等功能。
> - **预测团队**，他们的feature是可以自动标注的，所以整个数据管线是可以无人值守的。我们每周末会自动运行，对新的数据进行特征抽取、训练、验证，下周就可以用到新的模型了。
> - **仿真团队**也会从新数据里发现有用的数据，自动制作成仿真场景。
> - **感知团队**会将原始数据进行转换，生成符合标注团队或者标注公司要求的数据集，然后交给他们标注。

- **控制团队**也会根据每一台特定车辆的数据，或者对它进行动力学模型建模，用在仿真引擎里面实现控制在环的仿真；或者对它进行标定，更新calibration table，使得将来的控制更加精准。

目前我们内部已经部署了9个数据流水线。那么各位开发者如何从Apollo Fuel当中受益呢？我们主要将在SaaS这一层为大家提供服务，其中**车辆标定**就是我们本次在Apollo 5.0里面首次对外发布的云端服务，未来我们也会将其他更多的应用进行孵化，凡是我们觉得稳定成熟、对外部开发者有用，将来都会一一开放。

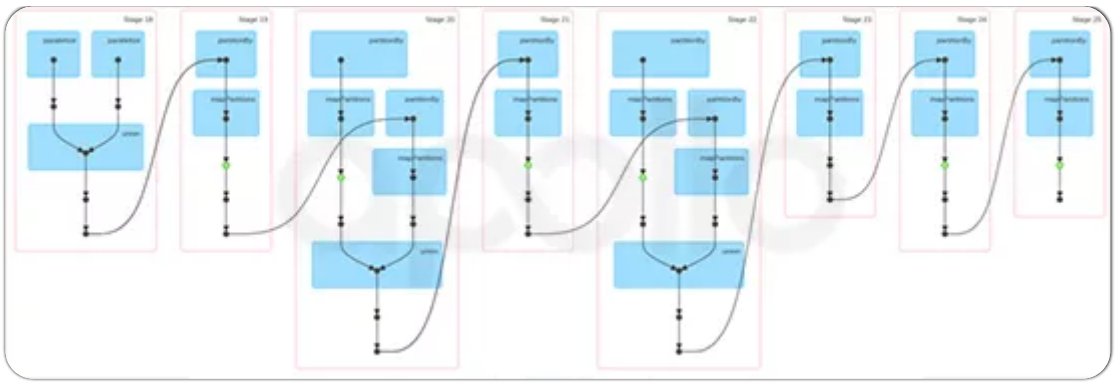
Apollo Fuel的工作流程如下：车队采集数据之后——上传到云服务——开发者提交spark作业对数据进行处理。作为Apollo Fuel算法开发者底层用Docker，里面装了spark技术组件。



▲Apollo Fuel的工作流程

在Apollo Fuel的底端，我们用到了Kubernetes，继承了一些很好的工具，比如它的Dashboard，可以满足我们在资源管理和监控上的很多需求。

而在任务管理上，我们使用Spark。下图是Spark的UI工具，可以很直观的理解为什么我们的应用叫数据流水线。不同的应用可以级连在一起，共享一些计算过程和中间结果，这也是我们高效率的来源之一。



▲Spark的UI工具