新思路打造移动端个案综合日志分析系统

——美团点评 资深移动架构师 周辉

大家好，非常欢迎大家来全球运维技术大会上听移动端的话题分享，移动端作为整个大会唯一的话题，这里不会对后端做过多的涉及，主要是谈一下对前端更深入的一些需求，以及在这个需求之上，美团点评自研的移动端“个案”综合日志分析系统。

我叫周辉，来自美团点评，有近13年的移动端开发经验。我是一个比较纯粹的前端开发者，对后端不是很了解，在美团点评，我主要负责移动端的底层，包括网络、日志、监控等基础设施。大家有兴趣的话，可以关注美团的技术博客，里面有很多我们团队分享的文章。

今天的分享主要分三大部分。首先是移动端日志的特点，然后对比日志处理专场重点涉及的后端日志，引出我们的 Logan 系统， Logan 是一个关于个案日志分析的系统。最后我会介绍一下整个 Logan 系统的开源计划。

移动端日志的特点

首先介绍一下，相对服务器端来说，移动端日志的特点。

第一个问题，移动端日志运行的环境差异非常大。这是相对于服务器端日志来说的。2017 年全年，国内市场发布的手机总共有 1997 款，这么多的机型要适配！雪上加霜的是，手机厂商都习惯自创一些“操作系统”，会对原生的基础组件进行改造。同时，网络运营商、系统版本、多 APP 版本，甚至一些本地的差异化环境，都会引起日志的不同。相对服务器端来说，移动端日志的运行环境是非常恶劣的。

第二个问题，移动端日志上报困难。服务器端的日志主要来源于整个内网空间，所以采集和存储都是比较方便的。移动端的日志要经移动网络进行采集，经常会有一些热点的切换，网络性能比较差，上报也存在一些问题。

第三个问题，移动生产终端种类众多、数量庞大，对后端固定的日志处理服务器来说，这存在一个重大的差异。如果把所有的日志都详细的采集下来，对后端的压力是非常

大的。虽然现在有非常多的日志后端搜集技术在解决这些问题，但从整个日志前端来说，问题更加严峻！我们现在主要是分析网络日志，但实际日志的种类很多，美团已超过了40种，这些日志如果全部上报上来，存储的压力是非常大的。

移动端日志面临的问题和挑战

从以上背景中可以看出，移动端的日志存在着一个矛盾点——就是在复杂多变的环境下，需要详细的记录环境信息和运行日志，这些日志如何准确的上报，上报以后又能存储，这是一个矛盾点。早期的时候，我们曾尝试解决这一矛盾。比如对我们的网络日志进行精简，只记录一些关键点，不把详细的日志上报上来，如丢弃请求的 header 信息等。同时，我们对频繁上报的接口进行了采样，就算采样接口设置 1% ，还是可以看出整个网络运行环境的一些问题的。但是，拿这个数据分析个案日志是存在很大问题的，因为被采样的关键点上的日志，可能有丢失。

第二个挑战，是日志种类的多样。美团点评现在的日志种类超过了 40 多种，如网络日志、用户行为日志、代码级日志和性能日志。大部分的日志都是采用网络采样上报的方式，从客户端采集之后，需要上报到后端进行记录，这样就会带来一些难点和缺陷。首先我们采集到的日志是不完整的，因为大部分的日志都经过了采样和精简；其次是后台查询不及时，由于后台的架构及性能设计，我们的日志同步存储会有1-3个小时的间隔时间，这就会造成查询日志的不及时。

第三个问题，是各日志体系的割裂。有些日志本地采集的时候，是加了时间戳的，这是一个很好的手段，但和其他的日志一起分析的时候，就会发现，有的日志加了时间戳，有的没有加。整个日志体系是混乱的。

最后一个问题，是大家过于关注后端的日志。我是这么觉得的。大家很少关注移动端的日志分析，即使有些团队做过一些移动端的日志分析工具，但大都比较简单的，而且缺乏后续的维护。

需要什么样的移动日志系统

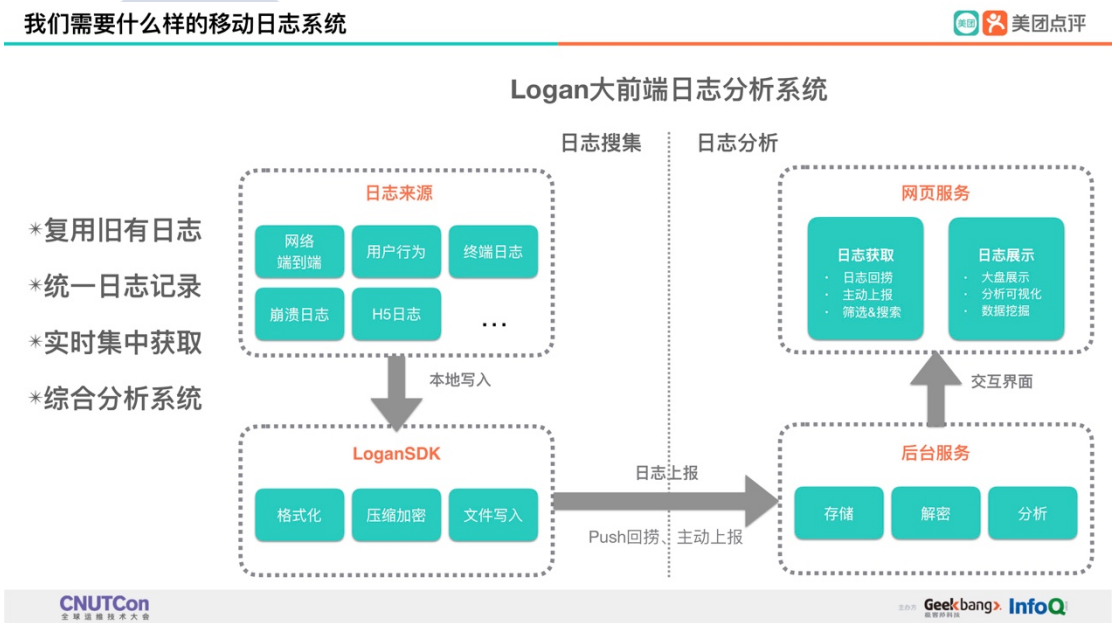
基于以上四点，我们打造了一个全新的移动日志系统。那我们到底需要一个怎么样的移动日志系统呢？主要是有几点。

第一是在原有的日志分析系统上开发新的功能。日志系统中基础的监控、报警甚至是生成报表等功能，需要保留下来。这些功能在移动端日志中的埋点，是很优秀的。

第二是前端统一记录。需要把移动端的日志全部记录在移动端，当然一些上报的功能是保留的。我们是希望把所有的日志集中保留在前端，这样子可以一次性把所有的日志打捞上来。

第三是需要所有日志的实时集中获取，也就是可以一次性把所有日志打捞上来。

第四是需要建立一个专注于前端日志的综合分析系统。



【图P7】

Logan 大前端日志分析系统

有了这些以后，我们打造了一个自研的 Logan 大前端日志分析系统，主要分日志搜集和日志分析二大块。

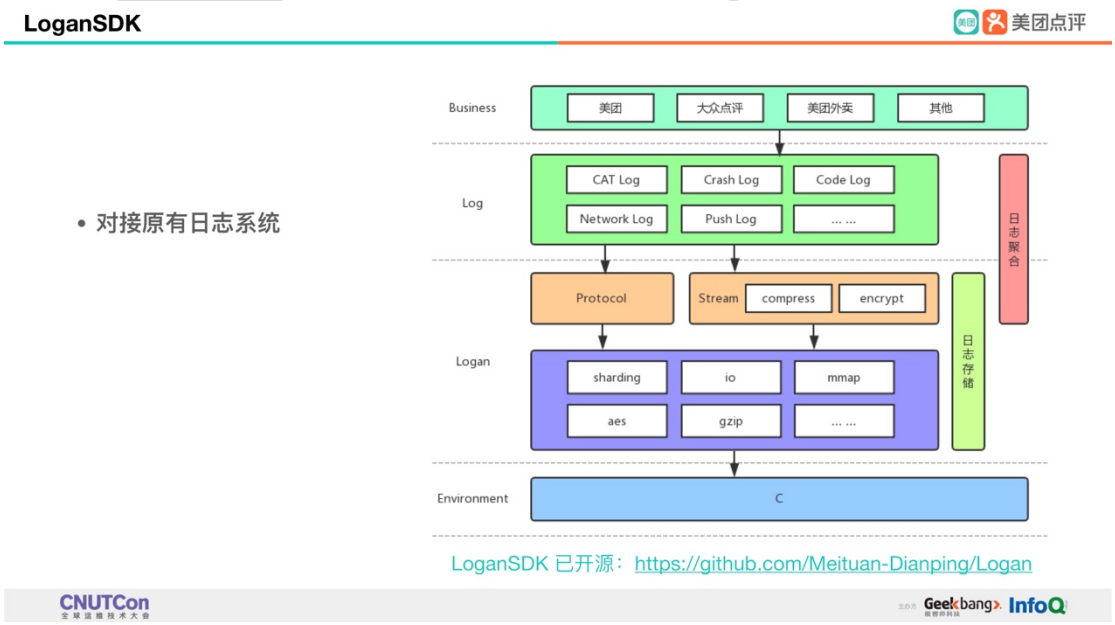
日志搜索方面，针对多日志来源，我们在原有的日志体系的基础上，为移动端量身打造了一款专注底层日志记录的 SDK 。这个 SDK 主要有二个功能，第一是进行日志的统一写入，会保证时间序是统一的，第二是在需要分析问题的时候，日志能够集中进行上报。

日志分析这块，包括发出日志回捞的需求、审核系统等。日志分析最终会以一个日志展示界面进行显示。

接下来我会重点从日志搜索和分析两大方面，详细的讲解Logan大前端日志分析系统。

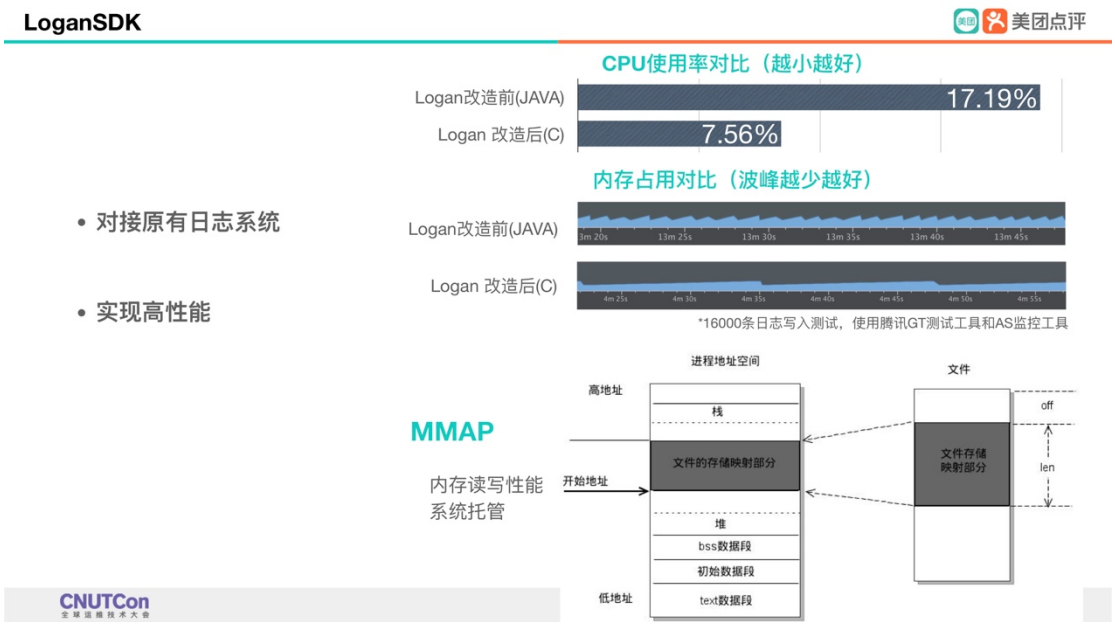
Logan 的日志采集

首先我们把 Logan 放置在 Business 和 Log 层之下，也就是保证在原有日志系统之上，使存储功能下沉到 Logan 系统中，Logan 能够将日志数据进行格式化加密并保存起来。



【P8图】

同时 Logan 会有一个上报的功能。出于性能考虑，我们的底层内核是用 C 语言来写。在日志的写入性能方面 ，我们有一些优化，最开始是用原生语言来写入的，安卓用 Java ， IOS 用的是 OC 。OC 语言性能还可以，Java 语言因为有 GC 的存在，性能和 CPU 的消耗是非常严重的，因此我们使用 C 语言进行了改造，改造后，在压力测试下 CPU 占用率减少了百分之六七十以上。且由于底层是用 C 语言写的，内存管理方面也更加可控，GC 上也更加舒缓，不像 Java 这么频繁。



【图9】

然后我们引入了一个老技术，叫 MMAP ，这是一种将文件从内存映射到文件的机制。MMAP 的作用有两个，它能够保证文件写入性能是内存级别的，相当于写入是在内存上进行操作。真实的文件写入是操作系统托管的，这在性能上没有问题。用了 MMAP ，当 APP 发生崩溃时，操作系统还能将崩溃时内存中的日志，写入真实的文件中。

Logan 的日志存储规则

Logan 在日志的存储方面有一些规则。

首先是 Logan 使用 Gzip 压缩，压缩率达到了 90% 以上。我们做过一些测试，在频繁操作的情况下，用户一到二个小时的操作，只会生成 2-3MB 的日志大小，占用空间相当于二三张图片的大小，这个是可以接受的。

第二个，我们的加密是从日期和设备的维度进行加盐，确保每一个用户加密的方式都是不一样的。

然后我们对所有的日志按日进行写入，单日的保存上限是 10MB 。如果超过 10MB 我们会进行丢弃。思路是这样的——如果日志写入过多，很有可能是发生了一些循环写入的 bug 。这时候我们应该是关注的是 bug 产生前的阶段，而不是循环写入的过程。当然会有一些特殊的情况，比如说美团点评的内部有旗手端和打车的司机端，他们会有一些全天候的需求，我们可以对这种需求进行线上配置及适度放宽，在日志的保存和用户的空间占用上有一些取舍。我们日志默认保存七天，这个也是可以调整的。

Logan 的日志上报规则

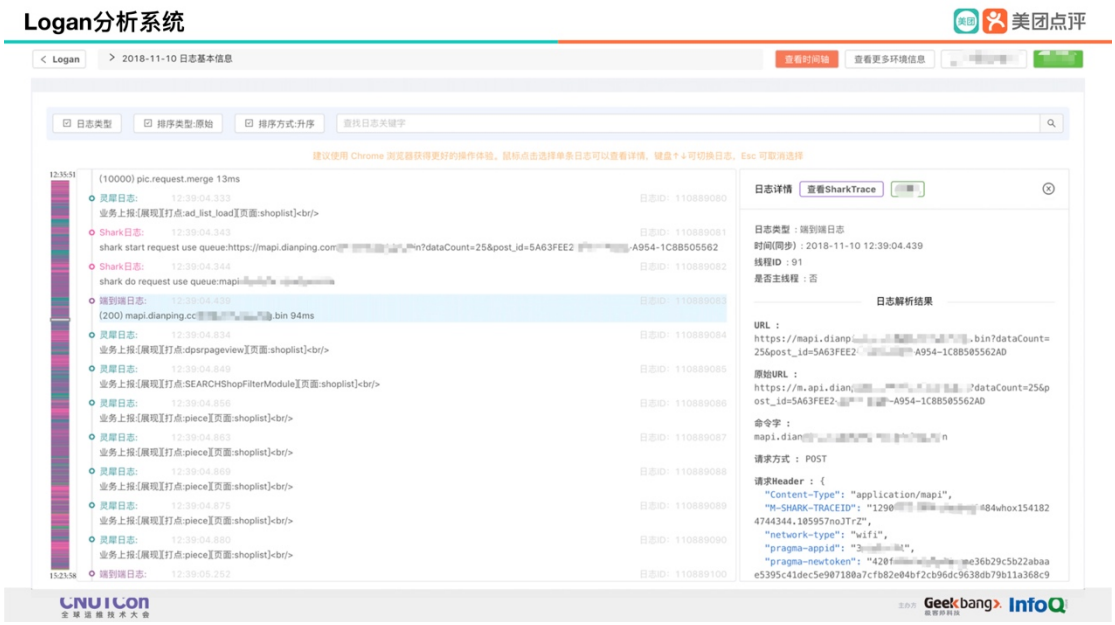
接下来我们说一下日志的上报。日志上报由于涉及到用户的隐私，这里不做过于详细的透露。Logan 日志上报主要是这么一个体系——用户通过客服进行问题的反馈，客服会帮助用户上报日志，我们把它简称为 Logan 日志（从客户端来的日志，我们都称之为 Logan 日志）。然后客服会把日志交给开发者进行分析，开发者分析后，会帮用户解决问题。我们的开发者，是不能直接和用户沟通的。

我们在用户隐私保护的方面做了很多的努力。比如说，我们日志的本地写入是有加密的，上报也是使用安全的通道，我们在后端写入日志是要经过部门领导审核的，而且日志系统对接了公司自研的权限管理系统，如果检测到一些开发者在异常使用用户的信息时，就会发出告警，及时的把这个人查出来。

Logan 移动端日志分析系统的能力

这一张图展示了我们 Logan 系统的分析界面。我们自己做的这一套系统，我也不敢说是不是业内比较好的，这里就是希望在这个大会上抛砖引玉，给大家带来一些新的思维。因为整套系统都是我们自研的，所有的细节都是我们自己亲自打磨的，这里有一些比较好的点可供大家借鉴。

整个日志系统，从上到下包括了很多的功能。最上方是日志的基本信息，提供了一些日志的展示界面，提供了数据挖掘，可以挖掘用户更多的环境信息，往下是一些日志的筛选、排序和搜索的功能，再往下是日志的列表，点击每一个日志都可以看到详情。



【P12/P14图】

首先说下日志分析系统的基础能力。

1.日志解析

在日志解析方面，将原生的日志直接展示在列表中，会使开发者在分析的时候造成一些误读。因此我们对日志进行了一些解析，可以将日志以更加通用的方式概括出来。当点击了某一个日志以后，右侧显示检测的结果，包括日志的类型等详细信息。

2.本地写入

Logan 日志主要的特性是本地写入。原来网络方面端到端的日志在上报的时候进行了精简，但是本地存储时可以更加详细，包括完整的 URL 、完整的 Head ，或 body 之内的详细内容，都可以在本地记录下来，然后由我们的日志系统进行分析。

3.日志定位

我们独创性的研发了日志定位的小控件。大家不要小看这个控件，这个控件在公司的内部是非常受欢迎的。在传统的日志体系里面，要定位一个日志是非常困难的，早期我们做过各种尝试，包括手动的输入日志的范围，但筛选之后只能看到一部分的日志。后来我们进行了改进，手动输入某一个时间点的日志，这样在使用中还需要键盘输入。于是，我们从用户的角度设计了这么一个控件，借用它可以看到全局日志的展示情况。图中左侧的柱状区域，每一个颜色代表不同的日志类型，鼠标滑动就可以查看日志类型，点击之后就可以定位到日志对应的位置。

【P13】

4.多类别筛选

同时我们提供了多类别的筛选，我们已经接入了 36 款公司内的日志。这样筛选的时候，就会有更多的选择。有一些业务团队已经有了自己的日志的记录体系，甚至是有自己的上报和展示体系，接入了 Logan 以后，可以直接复用 Logan 的能力，在 Logan 中查询他们的日志就可以了。

5.只查看某种日志类型

我们独创性的增加了一个“只选我”的功能，勾选可以选择只查看一种日志类型，还可按日志写入或生成的时间进行顺序。在搜索的方面我们也进行了一些尝试，包括一些快捷键等，这些在使用的时候都是非常方便的。

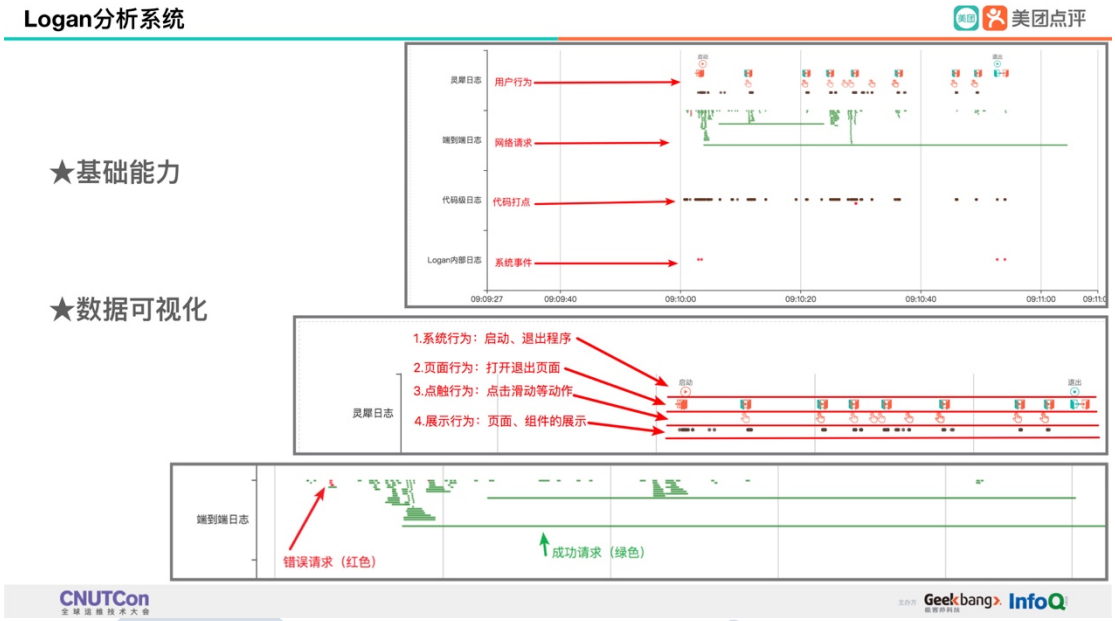
6.前后端打通

我们的日志系统在设计的时候是兼顾到了前后端打通的，我们是从网络请求这一块进行打通的，在后端的日志上会有一个统一的日志 ID ，所有的日志都会绑定该唯一 ID ，请求返回时会将 ID 记录在本地，这样日志打捞上来之后，就可以通过 ID 打通前后端日志。

7.时间轴控件

接下来是我们的一些小小的创新化的尝试，可以给大家借鉴一下。首先是我们采用了百度 Echarts 设计了一个时间轴控件，在这个控件之内，我们目前重点分析了四种日志：用户行为日志、网络日志、代码打点和系统事件。在使用了这个控件以后，我们发现日志从采集到图形化展示，有很多需要注意的点，比如在用户行为方面，如果把所有的日志打

点在一条线上，会混乱不堪，因此我们通过用户行为重要程度的不同，将用户行为日志分成了四个层次，包括系统级别的启动退出、页面级别的打开退出、点触行为和展示行动。



【P15图】

8.图形化展示

我们原来网络日志的点，都是会在发生错误的时候记录一下，这个实际上是没有时间维度的。在图形化展示的时候，可以将网络日志拓展出时间的概念，就可以看出有一些非常长的点，它有可能是程序在运行的过程进入了后台，下次打开的时候，才会进行响应或缓存失败，这些在我们没有做这个事之前都是不可想象的。同时，我们通过网络的状态码将这些日志进行了颜色的区分，比如说红色代表了错误，绿色代表了成功。我们还在图形化上将代码级日志用不同的颜色进行区分，代码级日志可以理解为控制台日志，因为控制台的日志是有一些日志级别的，这样查找的时候就会更加方便。

9.数据挖掘

接下来是我们数据挖掘中的应用，在数据挖掘上我们还有很大的提升空间。我们分析问题的时候，经常会需要得到用户的更加详细的信息，但是这些用户的详细信息，是分散在不同日志里面的：比如说网络请求中，User Agent 中会存放一些当前的环境信息，这些环境信息可能会发生一些改变；用户行为日志里面也有一些用户环境信息的搜集。我们把这些信息进行了汇总展示，在系统中点击一个按键，可以通过多个维度将信息进行汇总、分析、整理、展示，这包括网络请求、崩溃等。这是我们在数据挖掘方面进行的浅显尝试，当然以后还会有更多的尝试。

Logan 分析系统在美团内部的使用

我们在公司内部推广了 Logan 以后，接到了非常多的需求。比如有的团队想将他们的用户行为回放系统放进去。这个需求是很合理的，但这些需求对于 Logan 系统来说过于庞大。此外还有一些 DevOps 工具、网络性能分析工具，如网络用户画像分析，这些对 Logan 开发来说并不是最核心的点。

因此我们将所有的需求提出了一个新的概念，叫内部的数据开放平台。我们希望在保证安全的基础之上，将日志进行内部分享，各个团队对可以在 Logan 系统之上进行数据的分析、整理和展示。

这里简单的概括了一下，这些需求分为了很多种，包括业务上的分析、图形化展示的需求、开发工具，也有我们内部的大牛希望在大数据分析方面做的一些事，甚至还有机器学习方面、通过日志分析出用户的特征从而分析出整个日志的情况等等。

目前 Logan 在美团点评内部的使用中已经接入了 IOS ，安卓，Web 和小程序。Web 是分了二块的，一部分是在 APP 容器之内，我们通过桥接的方式将日志写入到 APP 内部，另一种是在浏览器或第三方的运行环境中，对日志进行搜索、整理、上报。

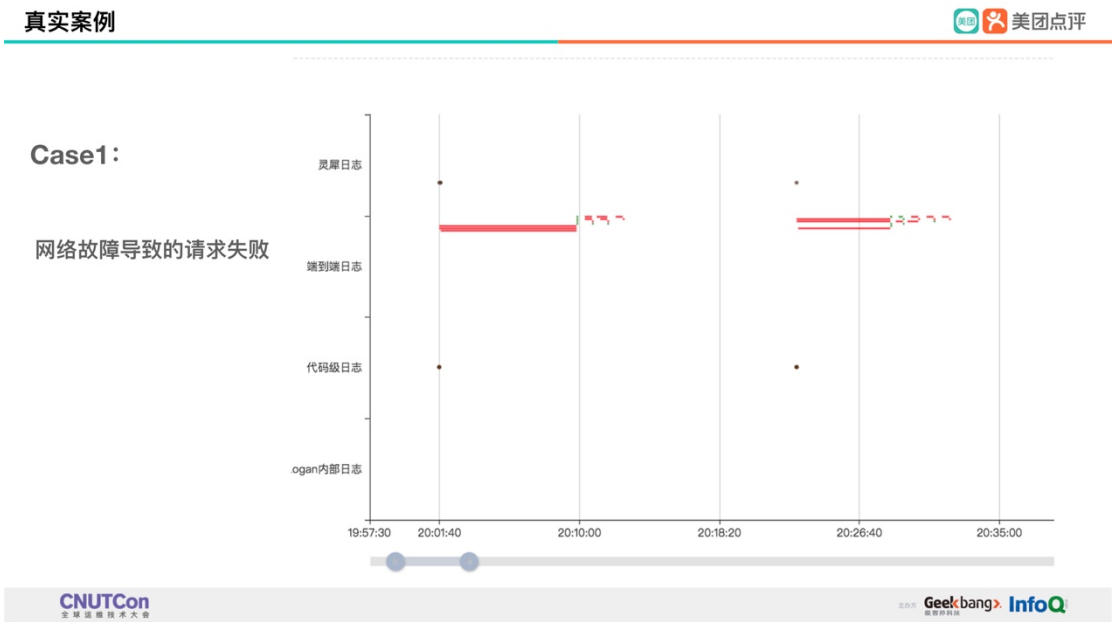
目前我们公司内部已经接入了 49 个 APP ，其实还可以推广的更多。美团点评的 APP 是非常多的，除了平台级别的，我们还有一些垂直业务线的，垂直业务线都会有销售端、后台运营端、商家端等等，所以日志的数量是非常庞大的。Logan 在美团内部接入了 26 种种类的日志，基本上美团所有的日志都接入了。

案例

接下来给大家分享二个真实的案例，第一个案例我称之为愤怒的老板。

-愤怒的老板

有次某个老板负责的服务页面打不开了。用传统的日志捞取方式，在后端是捞不到日志的。我们就向老板推荐用 Logan 系统进行回捞，就得到了这一张图。老板开始并不信服，他要我们证明，还说为什么你们捞的到而我捞不到，是我的姿势不对吗？



【图P19】

之后我们对他的日志进行了分析，发现日志里的绝大部分都是红色的，这说明网络方面出了错误，这也就是日志实时上报不上来的原因。而这从整体上是看不出问题的，从个案上就可以看出。我们把截图给老板看，并分析了理由给他，结果老板就不愤怒了，很愉快的就接受了。

从这里可以看出 Logan 系统的几大特性，第一个就是离线存储能力很有用，第二个是颜色可以帮助我们快速判断，第三是整个系统是一个多来源的综合日志系统，不管他提出什么需求，我们都可以帮他进行分析。

-焦虑的用户

第二个 case 是焦虑的用户。用户家住在七宝，有一天他到了公司以后，在大众点评公司总部，美团系统给他推荐了一个七宝的商户。这个情景感知功能，会根据用户当前的位置，进行一些附近的活动推荐。他觉得很奇怪，系统是不是出了问题呢，他要我们必须给个交代。我们开发接到了这个任务以后，就开始马不停蹄的找问题，但这很困难。

整个情景感知的功能很复杂，是在多维度、多角度的情况下，最终进行的一个信息决策。情景感知会涉及到用户画像、GPS 定位、网络最终反馈、用户行为等。我们需要从四个日志库拉日志，传统的方式要等一到三个小时，这个用户肯定是等不了的。有了 Logan 以后，我们很快定位出，是系统在取该用户位置的时候，错误的取到了缓存中的位置。

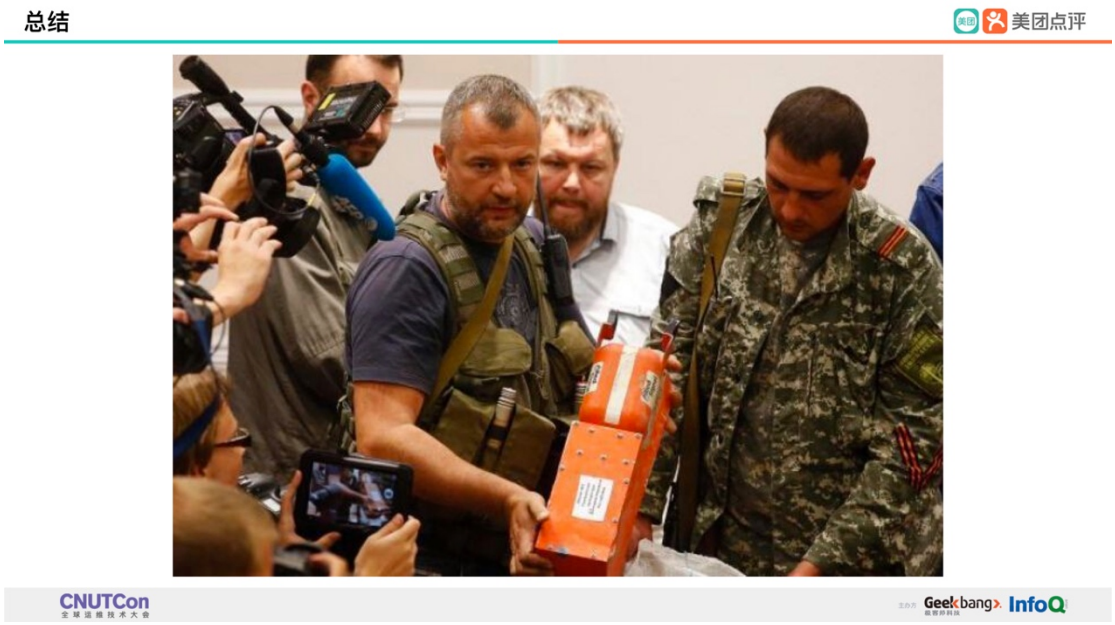
从这个例子中可以看到，多维度、多角度的综合日志是非常重要的。

最后总结一下 Logan 大前端的日志分析系统，一是提供了 Logan SDK，主要是提供统一存储和上报能力，二是日志分析系统，是我们自研的正准备开源的日志分析系统，未来我们准备将其打造成基于云端的日志分析平台，所有的日志分析功能都可以通过该平台进行分析。

从更高的层面来看，监控和是分析是分不开的。从原有的整个监控分析体系来看，监控的方面我们是 OK 的，在分析的方面，大都是从后端取日志，各个团队都有自己的日志记录，这一块是碎片化的。Logan 填补了整个版图中的重要的一环，能够从前端取到日志，它得到的日志是更加全面的。

当然使用 Logan 的时候，有一些注意点：第一个是我们的 Logan 系统是一个关注于个案分析的系统，并不是可以查到所有的信息，所有的信息可以从实时大盘上去查；第二 Logan 是一种前台集中式的获取，不需要从不同的日志平台上获取；第三个，Logan 系统是有缺陷的，它的日志回捞是有一点缺陷的，比如说用户把 APP 卸载了，所有的缓存都会被清掉，就只能借助原有的后台日志分析了。

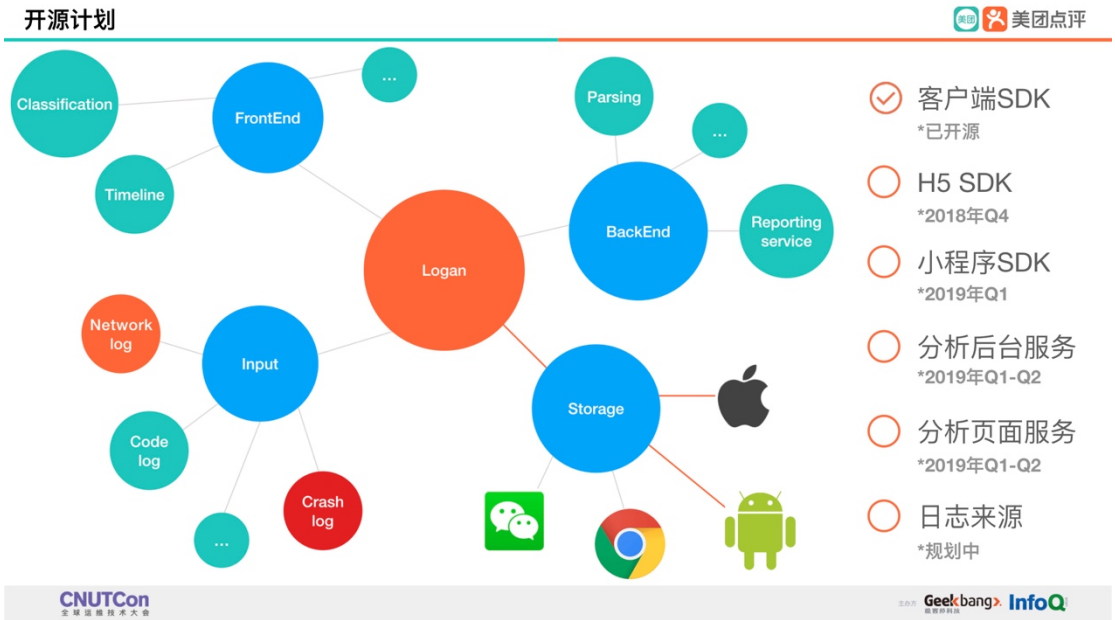
下面是一个故事。2014 年发生了很多的空难，有一个最重要的就是 MH17 航班。MH17 航班从乌克兰到俄罗斯的中间发生了坠毁，当时这两个地方正在发生冲突，乌克兰和俄罗斯都说不是我们干的。这起事件最终造成了298人遇难，不过还好当时的黑匣子被保存了下来。乌克兰的民间组织把黑匣子交给了相关的调查部门，2016年，调查证明是俄罗斯的导弹击中造成了这么一个事故。黑匣子在整个事件中，已经上升到国际化、全球性的高度。



【P24图】

从某种意义上来说，我们的 Logan 系统就相当于黑匣子的功能，会把犯罪现场所有的日志线索都汇总起来进行分析。

最后是我们的一个开源计划。我们的整个日志体系包括存储方面、输入方面、后端的分析方面，目前我们已经在 GitHub 上将客户端的 SDK 进行了开源，包含了日志的高效写入、日志回捞的支持，接下来我们会在 Q4 开源 H5 的 SDK，明年我们会持续开源小程序的 SDK，后续整个分析系统都会开源。这也是为整个开源社区贡献我们自己的力量。



【P23图】

我的分享就到这里，谢谢大家！

Q&A

提问：您说的前后端打通可能出现很复杂的情况，不知道究竟是前端的问题，还是后台的问题，谁去负责这个事情？

周辉：这个问题确实不好回答，从技术的角度来说，我们只提供详细的线索，但这一块我们是有一些案例的。比如说我们曾经发生过一件事，美团外卖所有的订单都是有满减规则的，有一个用户就遇到了前端显示可以满减，但是在付钱的时候发现满减规则不生效，多付钱了。客户端拿到详细的日志后，我们发现前端展示的是正确的，后端也是正确的，这时候就是扯皮的问题了。最终我们还是找到了中间的一些 Bug，因为展示和提交会有一些时间差。技术只是分析问题的，我们技术人员就是保留好线索，举出更充分的例子，最终谁来背锅，让更专业的人来评判。

提问：在移动端上传日志的时候，周期是多长时间？

周辉：这个地方有一个误解，我们的 Logan 系统并不会持续的进行上报，甚至不回捞的话完全不会上报，它只负责记录。分析问题的时候，我们可以采用一些综合的手段，比如用 Push 的方式进行上报，具体的细节不方便过多的透露。另外我们还有一些相应的手段，比如说我们发现很多用户投诉之前，很有可能点击了网络诊断，那我们就在网络诊断的内部埋点了上报的功能，我们的客服检索到上报的日志，就可以进行分析了。我们的日志并不是分散上报的，也没有周期的概念，而是一个集中回捞的方式。

提问：在浏览器方面是怎么上报的？

周辉：有两个方面。在 APP 容器之内的，我们是通过桥接的方式，把日志写入到了 APP 的内部；如果是在非容器之内的，我们有一个专门用来存储日志的数据库，只要接了 Logan 的 SDK，摇一摇就会出现一个日志上报界面，然后选择上报的日志就可以了。这是我们目前的一个比较简陋的解决方案，我们也在尝试通过一些其他方式解决，比如说预先为用户上线埋一个上报的 ID，提示用户上线后，我们会通过回捞系统，告知浏览器是不是要把日志上报上来，这一块还只是在尝试做，还没有最终的确定。谢谢。

提问：您好，你说 SDK 有可能在链路中丢失日志，这个丢失是要么取到，要么取不到。会不会出现丢失了一部分的情况呢？

周辉：有可能，比如说发生了崩溃的时候，程序已经是失去了响应了，有一些日志就丢了。当然我们也使用了 MMAP 的技术，可以系统托管，把内存中的日志写入到文件中。但是这个技术目前来看的话，成功率只有 50%，还是有很多日志丢失的情况。这一块我们暂时还没有找到解决的方案。

提问：日志已经落到文件中，在上传的过程中会不会丢？

周辉：上传是要么成功，要么失败。我们要保证日志的完整性和上捞的完整性，所以这一块暂时还没有遇到问题。

提问：日志回捞的时候，不管是前端还是后端都要关注日志的完整性，客户端上传了多少，服务端接收了多少，这个有没有调研？

周辉：我们的日志记录在本地的时候，可以确信是一定记录下来了，除非是发生了崩溃。当我们进行分析的时候，我们现在可以通过数据的开放平台，把数据交给后端的同学，后端的同学可以开发一个插件，来进行数据的比对。这块可以通过后端的数据分析来做！