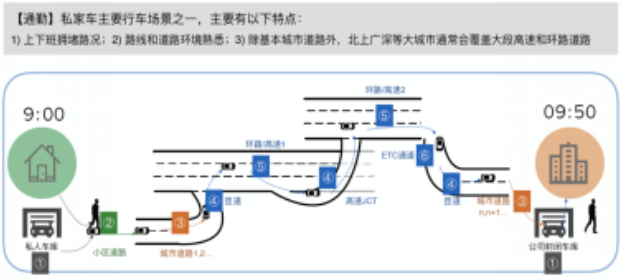
ANP3系统故障排查&&春季大扫除发布

ANP SYS | 辛建康

ANP3行业现状 – 快速过一下ANP3产品



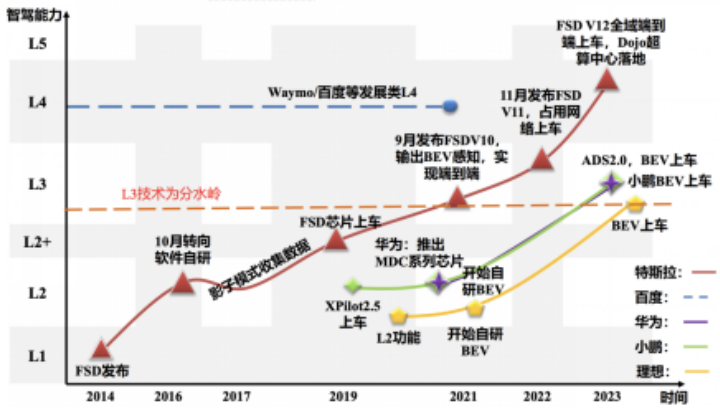
功能核心优势

- 将传统的辅助驾驶升级为具备A到B能力的高级辅助驾驶；
- 通过人机界面，降低用户的认知难度和使用门槛；

产品形态

- 车机地图发起导航
- 车辆按照导航路径自动驾驶（人机共驾）
- 在能力不足的地方提醒接管

ANP3行业现状 – 大家都在”卷”，超级”卷”



- **国内某证券公司行业调研**：2022 年乘用车 NOA 标配前装搭载交付量为 21.2 万 辆，今年 1-6 月交付 20.9 万辆，已接近去年全年水平，1-9 月交付量已达 37.7 万辆，同比增 长 **151.2%**，渗透率接近 2.5%。
- **ANP3三域融通**：快速集结，”开箱即用”，22年10月正式开始融通，500W+行代码，数百个模块，快速进入第一梯队；我们很”**难**”，也很”**强**”

ANP3系统架构 – 大型迭代不低于10轮，小型迭代无数，如何应对？

23年H1		23年Q3	23年Q4	24年
BEV上线	DriveOS 6050	纯视觉感知	OCC感知	轻图
ANP3三域融通	ANP3感知复用	AVP+ANP感知融通	Cyber迁移	感知能力提升
预处理异构计算解耦	…	SOP交付	预测、规划升级	…

➤ 系统的典型问题

- **硬件向**：授时跳变、相机掉线、BGM联网、底盘丢心跳、底盘控制延时、网卡UDP流量上限等…
- **OS和CUDA**：CPU10卡死、CUDA Context竞争、系统卡顿等
- **中间件**：跨中间件通讯、多运行时竞争、自研中间件稳定性等
- **业务层**：随机越界、算法退出、临界区竞争卡顿、死锁、死循环、不合理计算量等
- **基础库/系统运维**：故障检测、深度检测、容灾、故障恢复等

应对

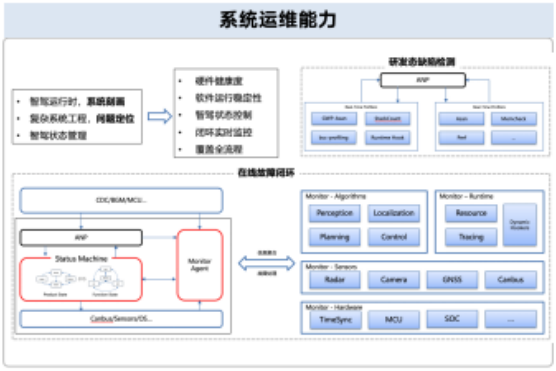
- “工具”远比”人”靠谱
- 懂E2E、懂大数据、会巧用

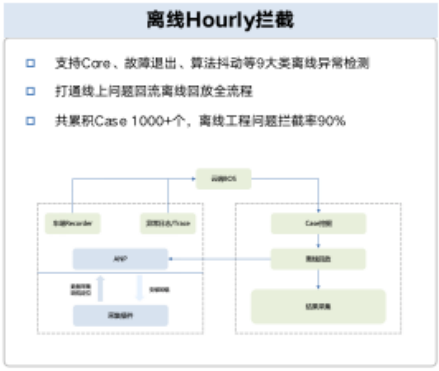
ANP3系统故障Tips – 相信AI的力量

题外话：AI在知识垂向和广度上做的很好，但在知识关联和逻辑上稍差，这也是现在唯一人类还有点用处的地方…

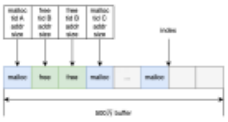
ChatGPT

ANP3系统故障排查 – 我们有什么

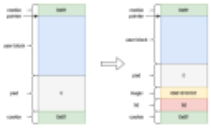




内存泄露排查 – 线程监控、大内存函数级实时监控

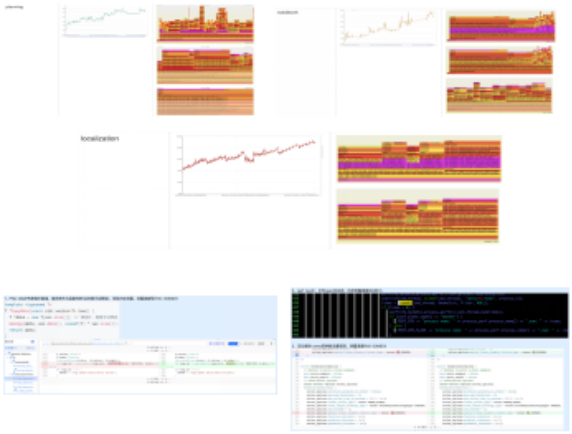


关键技术1：Jemalloc Hook



关键技术2：内存染色

关键技术3：Stackcount Hook mmap

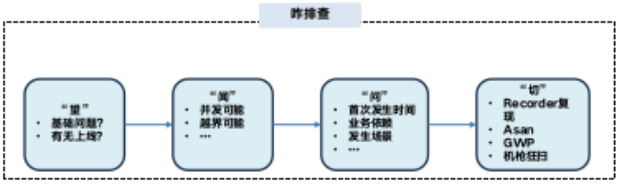


解决效率0.5月->5天

工程可用性 – Core、越界、踩踏等

难在哪

- "万恶的" Jemalloc：很多越界问题，不能第一时间报出
- 诡异的内存踩踏：往往犯罪者并不在第一现场
- 低概率问题：比如V1.4感知模块800小时core 1次
- 小心SIGBUS……



Tips

- ✔ 大部分问题，基本在望的阶段，就被拦住了，比如…空指针、abort
- ✔ 少量的问题，需要再进一步看一下，现在依赖人工，但未来希望能AI替代
- ✔ 需要相信大数据的力量，从Sugar、实时回传推算，首次问题发生版本、场景、模块可能运行到哪等
- ✔ 95%以上问题，离线Asan都能扫出来；5%问题，需要依赖GWP-Asan(比如800小时复现1次的诡异问题)

Core、越界、踩踏 – Why GWP

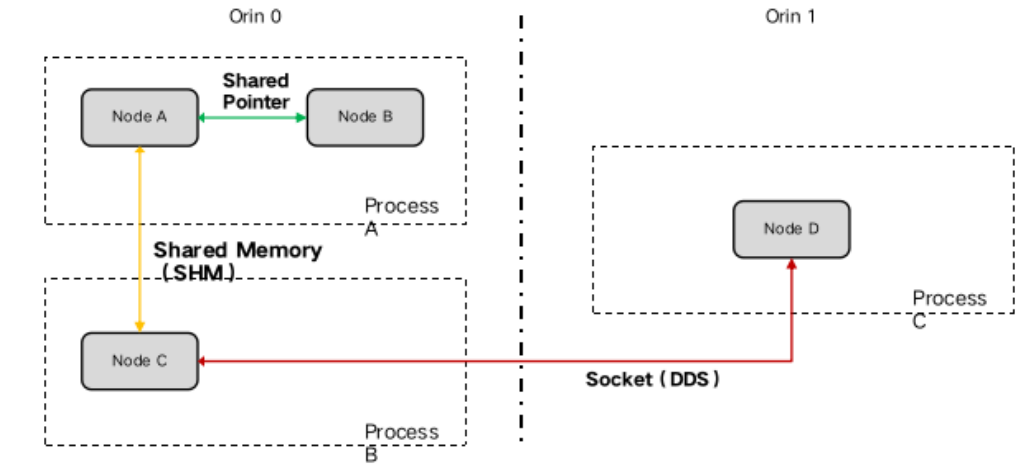
- `Use-after-free`
- `Buffer-underflow`
- `Buffer-overflow`
- `Double-free`
- `free-invalid-address`

原线上core：只知道哪个函数挂了，但不确定是那段内存飞了

GWP-Asan：明确给出写飞内存；正常路测就可以打开

参考：https://mp.weixin.qq.com/s/xipHtjHPVlyFQ6W-1HfUQQ

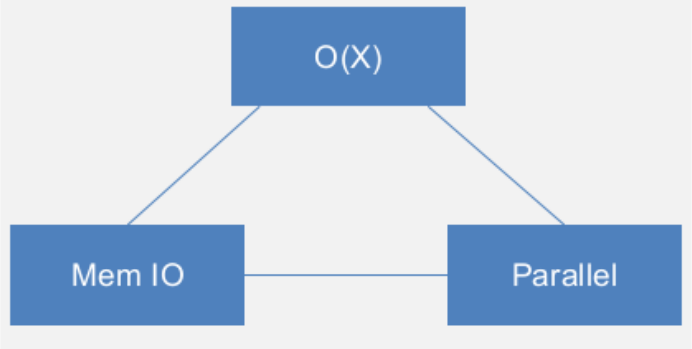
工程可用性 – 卡死、消息堵塞等



卡死排查思路

- 排core：/opt/log/anp/stdout
- 排自身：工程，有无锁、网络等待、超长sleep、IO等待等；业务，是否满足触发条件、上下游依赖等
- 排系统：CPU、内存、网络、IO、系统异常进程五要素
- 排对端：是否收到对端消息；对端是否发送了消息；
- 排中间件：线程池打满、拓扑掉线、消息阻塞、抖动丢帧等
- 排外部依赖：BGM(授时、CDC转发)；MCU(底盘、radar、gnss)等
- 系统：未来会为全域增加ANR检测能力

性能问题排查 – 启动延时、端到端延时



排查思路：资源观测->找工具->Profile->分析->优化->…

启动延时、端到端延时 – 常用工具

- strace：**启动延时**排查利器，可以深入到每一个SO加载耗时
- sys ebpf profiling：**CPU性能分析**，函数级火焰图，实车闭环采集
- jemalloc：**内存Profile**，支持实车闭环采集
- 系统线程内存、CPU分析
- 进程级**流量**采集
- …

其它

- 活用JIDU VP群公告文档，大部分路测问题(联网、授时、地图问题)等，都可以找到答案
- 多报明确问题，少报一些现象(比如DV黑屏了，谁来看下)，提高问题排查效率
- 相信工具、相信大数据
- …

推荐知识库

- ANP工程Best Practice
- ANP典型Core分析
- 定位问题分析宝典
- ANP出车手册(强烈建议多看JIDU VP群)
- 系统性能指标观测
- 线程级内存分析
- 线程级CPU分析
- ...

24年春季大扫除 – 预告

- Rule1：重点清理可用性类问题
- Rule2：对潜在问题坚决清理
- Rule3：部分引入AI扫描能力(对接中)

- Q1 清理规则(初步)
  - for 循环中不允许出现 size – 1
  - 指针都要检查有效性
  - 不要在一个函数中返回地址
  - 不要相信 vector 中元素的指针
  - 递归函数一定要加层深限制
  - if 语句中，右值放在 == 的左边
  - if 语句中，即便只有一行，也要加 {}
  - 浮点数比较，要考虑精度
  - 即使再不合理的代码，也不要 CHECK
  - 默认使用 std::abs
  - non-void 函数先写返回值
  - 不要在头文件里写 using namespace xxx 或 using xxx (别名除外);
  - 当一个类必需持有自身类的时候，禁止使用 std::shared_ptr;
  - 提前申请 vector 的空间
  - 对于 vector，能用 emplace_back, 就用 emplace_back;
  - 能用乘法，就不要用除法。
  - 类内提前开辟内存
  - 禁用Malloc、New
  - 禁用CHECK
  - 日志、write消息削峰

感谢~