

Flywheel & Coco 技术优势

1. Flywheel

对比：百度移动统计，腾讯bugly，友盟，网易云捕

1.1. 崩溃

1. Native崩溃抓取支持同时获取Java层堆栈。

其它业界方案	百度移动统计	腾讯bugly	友盟	网易云捕
	不支持	支持	不支持	不支持

1.2. 卡顿

1. 改进的Watchdog机制，在保持性能损耗最低的情况下可以统计到卡顿时长
2. 适配输入法特性，在用户静置不使用情况下性能损耗为0

其它业界方案	百度移动统计	腾讯bugly	友盟	网易云捕
	不支持	不支持	不支持	不支持

1.3. ANR

1. 基于Watchdog + processesInErrorState方案，ANR捕获率远高于Fileobserver方案。
2. 适配输入法特性，在用户静置不使用情况下性能损耗为0

其它业界方案	百度移动统计	腾讯bugly	友盟	网易云捕
	不支持	FileObserver方案，6.0以上机器基本收集不到	不支持	同时存在FileObserver方案和Watchdog+读取anr/trace文件方案，前者6.0以上机器基本收集不到，后者anr/trace文件权限不能保证一定能获取到，捕获率不高

1.4. 其它

- 1. 所有监控的用户流量消耗小于其它方案：端上去重与过滤

注：以一次崩溃消耗流量为例对比

Flywheel	百度移动统计	腾讯bugly	友盟	网易云捕

- 2. 基于Kotlin实现，上线以来框架自身未出现过问题
- 3. 链式数据流处理，可以随时对数据增加预处理流程

2. Coco

- 1. bpi接口设计解决了传统SPI架构的组件化框架需要人工下沉大量接口到基础库的问题，避免繁琐的维护成本。
- 2. 改进的ServiceLoader将加载时间从100-200ms降低为1ms，内存消耗从2mb降低为1kb以内。
- 3. Coco整体库大小11k，核心代码行531行（剔除编译期运行代码），远小于其它方案，但是支持完整的接口/路由两种机制