

支付宝APP的性能稳定性 优化及运维实践

钟鹤 2016/06/25

InfoQ ueue

促进软件开发领域知识与创新的传播



关注InfoQ官方信息
及时获取移动大会演讲
视频信息

ArchSummit
全球架构师峰会 2016

[深圳站] 2016年07月15-16日
咨询热线: 010-89880682

QCon
全球软件开发大会

[上海站] 2016年10月20-22日
咨询热线: 010-64738142

支付宝介绍

产品和架构演进，1.0->2.0->3.0

性能及稳定性挑战、优化实践

超级APP的运维体系，监控->诊断->修复

架构上的容灾规划

丰富的场景，生活互动平台



日常生活
转账、缴费、手机
话费充值

O2O
美食、外卖、
超市、电影

社交
聊天、群、生活圈、
多媒体



移动金融
余额宝、理财、芝
麻信用、花呗、保
险

开放
支付、快捷登录、服务
窗、插件

APP的架构演进



- 分层、单体应用
- 模块、工具库



- 多应用的平台
- 服务化、模块化
- 工具组件化



- 多应用的生态
- 开放、动态化
- 高可用、高性能、高灵敏度

App

2013

平台型App

2015

超级App (3.0)

2016新春红包，混合架构验证



2016春晚这个超大规模的活动，充分验证了这种混搭架构

支付型架构

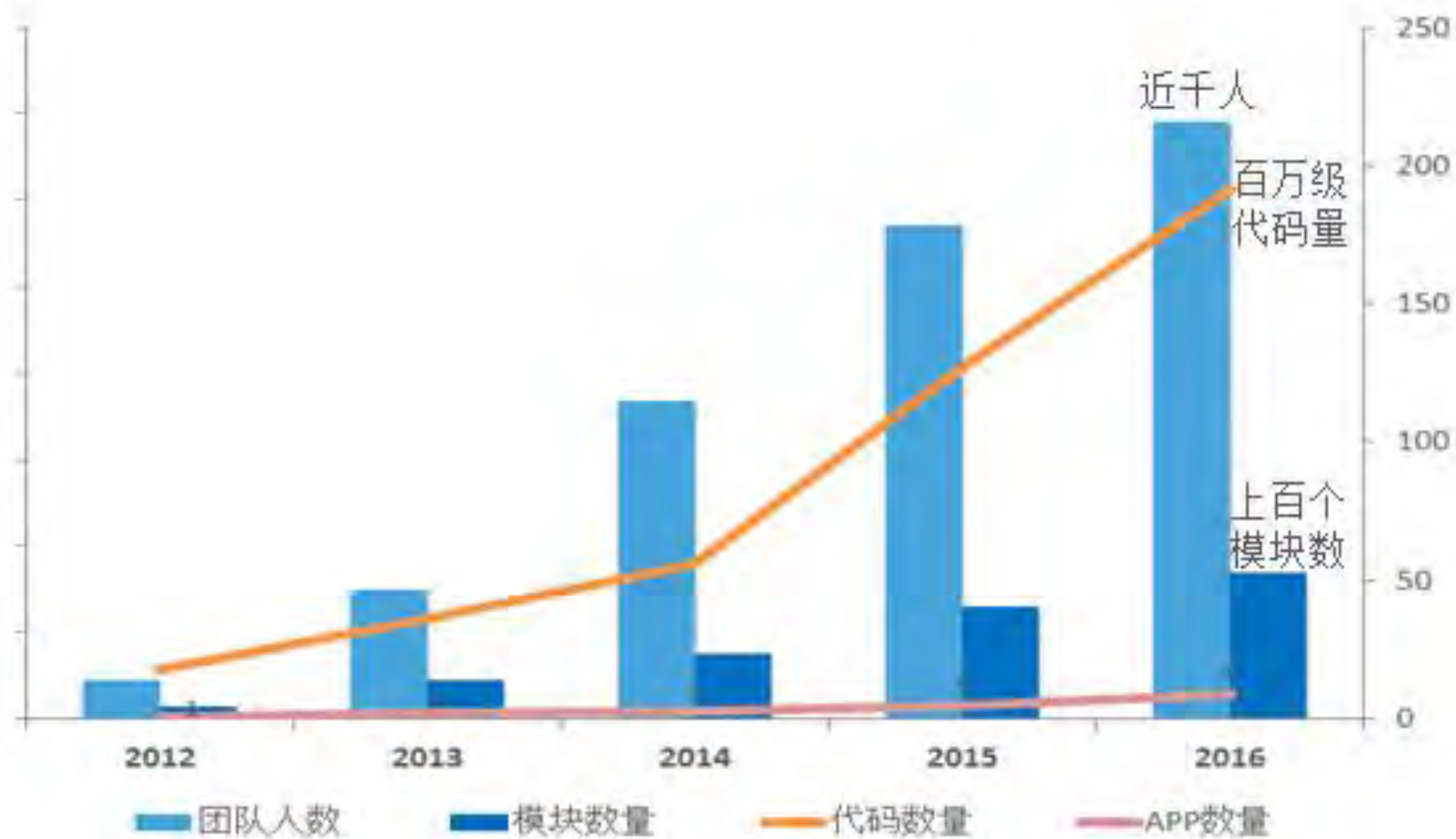


移动互联网金融型架构



生活互动型架构

大变化&大发展



业务复杂性带来的技术挑战



- 尾部用户绝对数增多
- 用户反馈增加
- 舆论传播影响大
- 345线城市 2/3空白
- 网络状况差
- 低端设备多，挑战大
- 团队人数增多
- 大团队协同开发
- 代码量迅速增长
- 启动尾部优化
- 外部h5电量流量治理
- 内存不断增大
- 用户对存储敏感

用户规模

农村&国际化

研发团队

性能

设备多样性带来的技术挑战

极端碎片化的机型分布



- COOLPAD 5890
- COOLPAD 5950
- COOLPAD 5951
- COOLPAD 7295C
- COOLPAD 7296
- COOLPAD 7620L
- COOLPAD 8670
- COOLPAD 8702
- COOLPAD 8702D
- COOLPAD 8705
- COOLPAD 8729
- GT-I8262D
- GT-I8268
- GT-I8558
- GT-I9082
- GT-I9100
- GT-I9100G

Top500

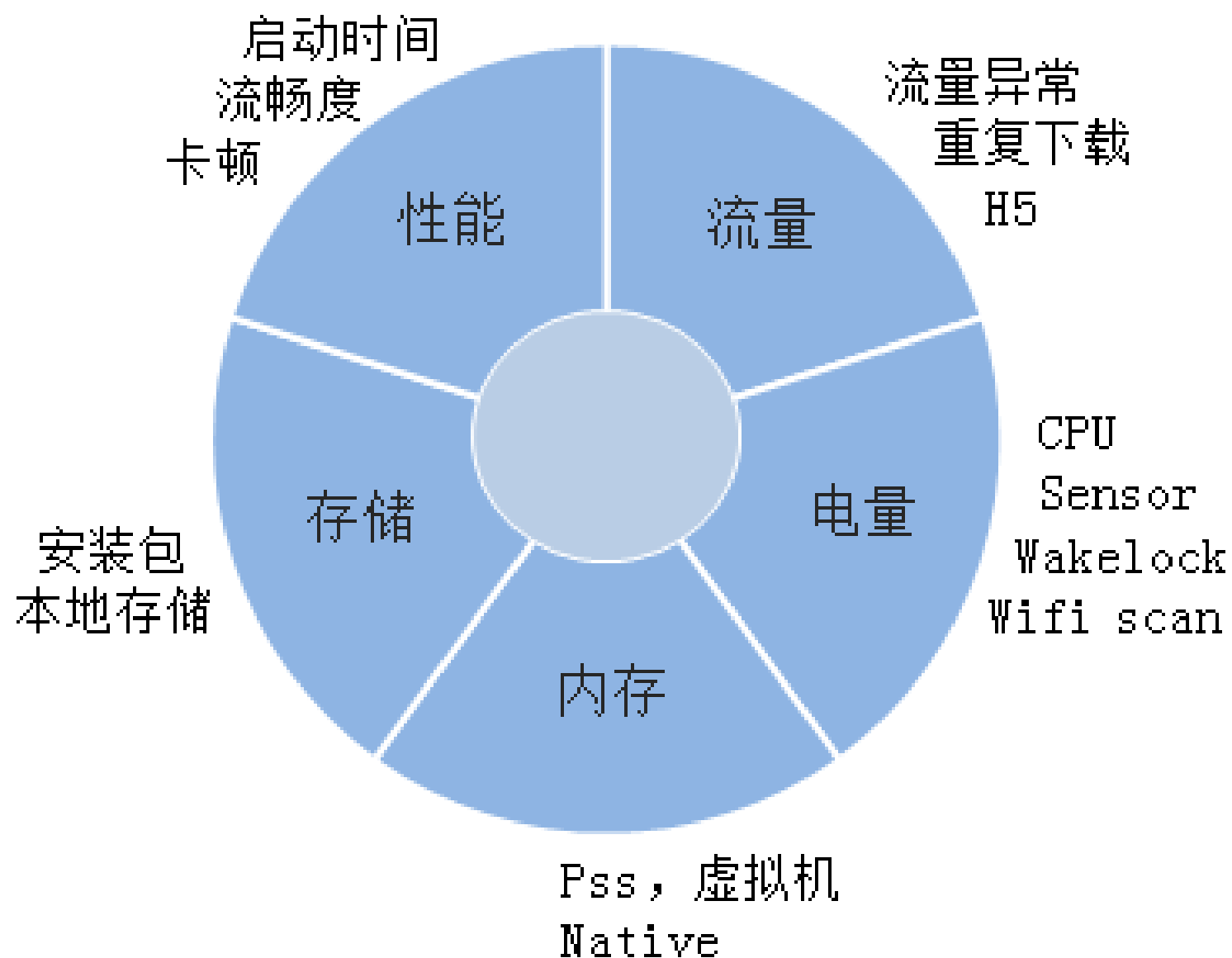
友盟
VS
支付宝

76款

R811	单核512M	2012年
R815T	4核1G	2013年
R821T	2核512M	2013年
R823T	4核1G	2013年
R830	2核512M	2013年
R833T	4核1G	2013年

V182	2核1G	金立	2013年07月
V183	4核1G	金立	
V188S	4核1G	金立	2014年10月
VIVO S7T	单核512M		2013年
VIVO Y13T	4核512M		2014年
VIVO Y913	4核1G		2015年
VIVO Y927	4核1G		2014年

性能问题的范畴



卓有成效的性能优化实践



- 1) 4.X图片解码转移Native, GC降50%
- 2) 清理内存泄漏、对象池
- 3) 内存大户专项优化, 启动引导图及日志、缓存等

- 1) quinox容器支持按需加载、性能调优
- 2) 线程治理, 80个减到40个, 耗时减少10%
- 3) 虚拟机dalvik调优、关闭jit
- 4) 主线程优先级调整、其它进程nice值调整
- 5) 启动流程重构, pipeline机制, 总共超过100项改进

内存

存储

流量

电量

性能

- 1) so编译使用STL共享库去重, 安装前小2M, 安装后小4-5M
- 2) Bundle内assets到主APK中、非必需lib转移到assets
- 3) 定向优化, 日志存储压缩、安装包7zip压缩、业务下线

- 1) 资源的增量/增量更新机制
- 2) 应用的按需下载
- 3) RPC、底层网络协议优化

- 1) 系统兼容优化, WifiScan
- 2) 业务优化, 完成及时释放
- 3) 技术基础优化, CPU、Wakelock唤醒, 心跳代理等

耗电原因


- . CPU 使用率
- . 各种sensor, gps
- . wakelock, 网络连接

耗电异常捕获

- . 接口获取
- . 遍历线程, 获取所有线程运行时间
- . 与主线程对比确定异常

耗电优化

- . 高性能dump线程调用栈
- . 通过线程映射调用方
- . 调用逻辑优化

- 
- . 半年累计优化问题 50+项
 - . iOS Android 电量异常率下降60%

耗流量原因


各种网络请求

流量异常捕获

- hook 所有网络请求
- 根据host 聚合流量
- 超过阈值确定异常

流量优化

- RPC, 底层协议优化
- 资源增量按需下载
- 通过切面信息映射调用方

- 
- 半年累计优化问题 50+ 项
 - iOS Android 电量异常率下降超过70%

内存分析

- memtrace/hprof 线下内存分析，遍历对象
- 根据生命周期标记内存泄漏
- 根据object创建引用确定业务归属

Native内存

- 图像库切换到native
- 4. x bitmap像素数据放到ashme共享内存
- GC 下降 50%

内存优化

- 对象池复用
- 减小bitmap对内存占用
- 使用更小的图，尤其注意H5
- 内存溢出率下降50%

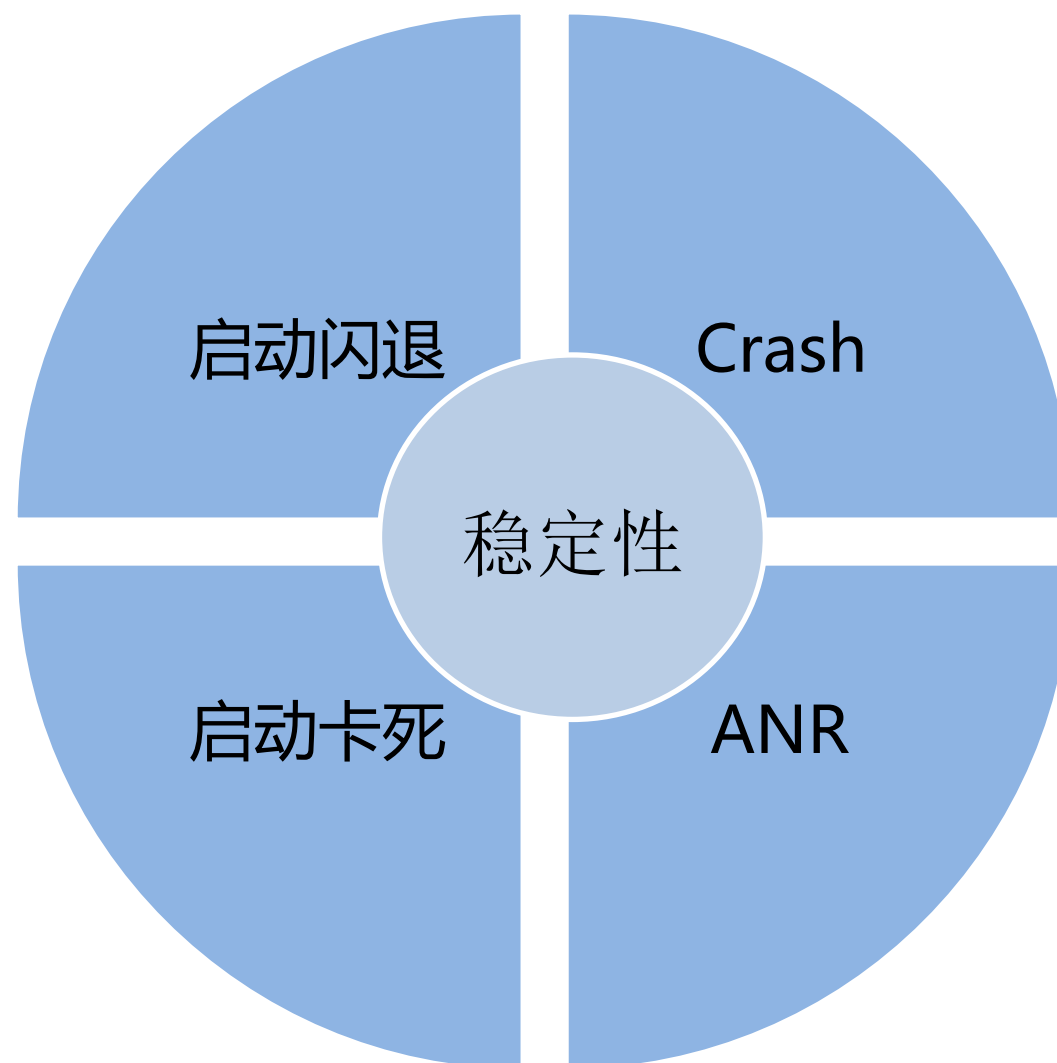
业务能正常运行吗？



点击业务崩溃

频繁重启、打不开

操作无响应



*crash*排名

系统栈分类

*crash*分级

*crash*上报

- . *crash* 率总体 0.1% 左右
- . 持久性闪退，比例<0.001%
- . 根据特征提取，对特殊栈*crash* 再分类，如webcore、系统相册等
- . 分前台闪退、后台闪退、持久性闪退、是否启动的占比
- . native闪退，捕获闪退100%上报

启动闪退

- 启动过程中监控
- 多次发生清洗数据
- 上报异常信息

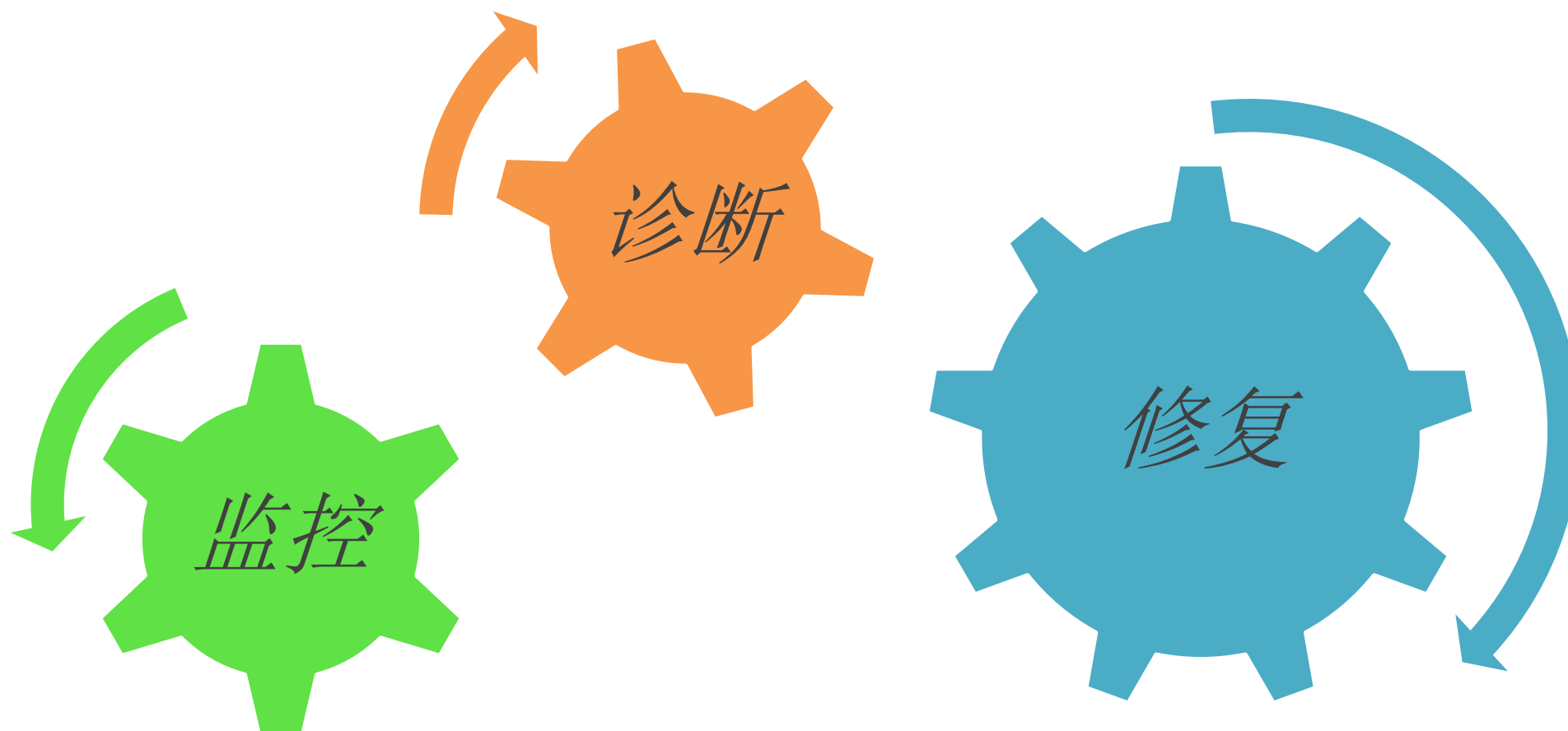
启动卡死

- 30s启动卡死监控
- 卡死发生上报异常
- 杀进程重启

ANR

- 主进程监控
- 5s 发生ANR上报异常

超级APP的运维体系



- ✓ 度量指标、全面（性能、电量、流量、内存、稳定性）
- ✓ 覆盖率99%，极端闪退、启动卡死、操作无响应
- ✓ 实时性、模块化、多样性展示（均值、分布）

- ✓ 诊断方式（推拉结合、白名单，1小时内必达）
- ✓ 诊断成功率（性能稳定性100%）
- ✓ 兜底线索（applog、logcat system trace）

- ✓ 多种手段（配置化、自动熔断&恢复、热修复）
- ✓ 修复本身的可靠性保证，可监控、可灰度、可回滚
- ✓ hotpatch成功率99%，sync pull结合，dalvik、art支持

客户端

- .量化、全面（性能、电量、流量、内存、稳定性）
- .覆盖率99%，极端闪退、启动卡死、操作无响应
- .业务侵入性小
- .按需配置

服务端

- .异常分析，提取特征
- .合并数量，预测风险
- .模块化，自动分配业务线
- .自动通知报警、状态跟踪

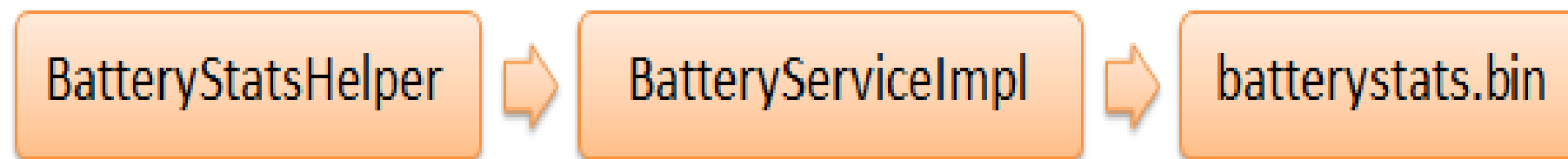
数据分析

- .实时数据解析
- .多种数据维度，均值，分布，尾部
- .版本同比对比持续提升

电量指数的计算



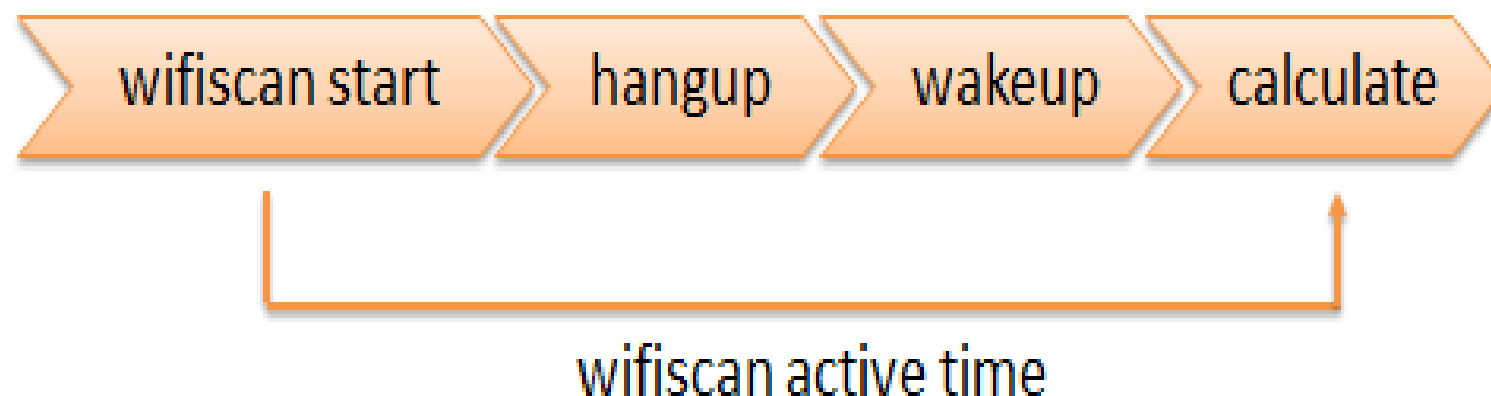
系统级数据采集



电量消耗模型

- 维度：cpu, wakeup, wakelock, wifi, tcp, wifiscan, sensor
- 权重：power_profile.xml @ framework-res.apk

关键的技术优化



快速定位与诊断



便捷的获取辅助信息

自动分配到人

快速定位

电量诊断

流量诊断

闪退诊断

...

- 自动收集cpu sensor wakelock等所有耗电数据
- 高性能 dump 线程栈
- 自动确定业务归属

- 自动收集高耗url
- 符号表偏移确定调用方
- 自动通过url映射业务归属

- 获取native闪退
- 获取辅助信息, 代码段附近内存, logcat, 调用栈
- 通过栈, logcat等信息自动划分业务归属

- 诊断方式 (推拉结合、白名单, 1小时内必达)
- 兜底线索 (trace logcat 其他日志)

多层次的动态化技术



配置同步 (*rsc*)

- . 100%到达率, sync + pull结合
- . 连接即触达

H5

- . 快速的web发布方式
- . 加载和渲染速度, 提升30%, prefetch + 离线包
- . fallback机制, 离线、在线无缝切换

跨平台框架 (*hcf*)

- . 实时发布, 当天日活覆盖率90%
- . 加载和渲染速度, 提升50%
- . 包大小, 仅仅是Native的一半

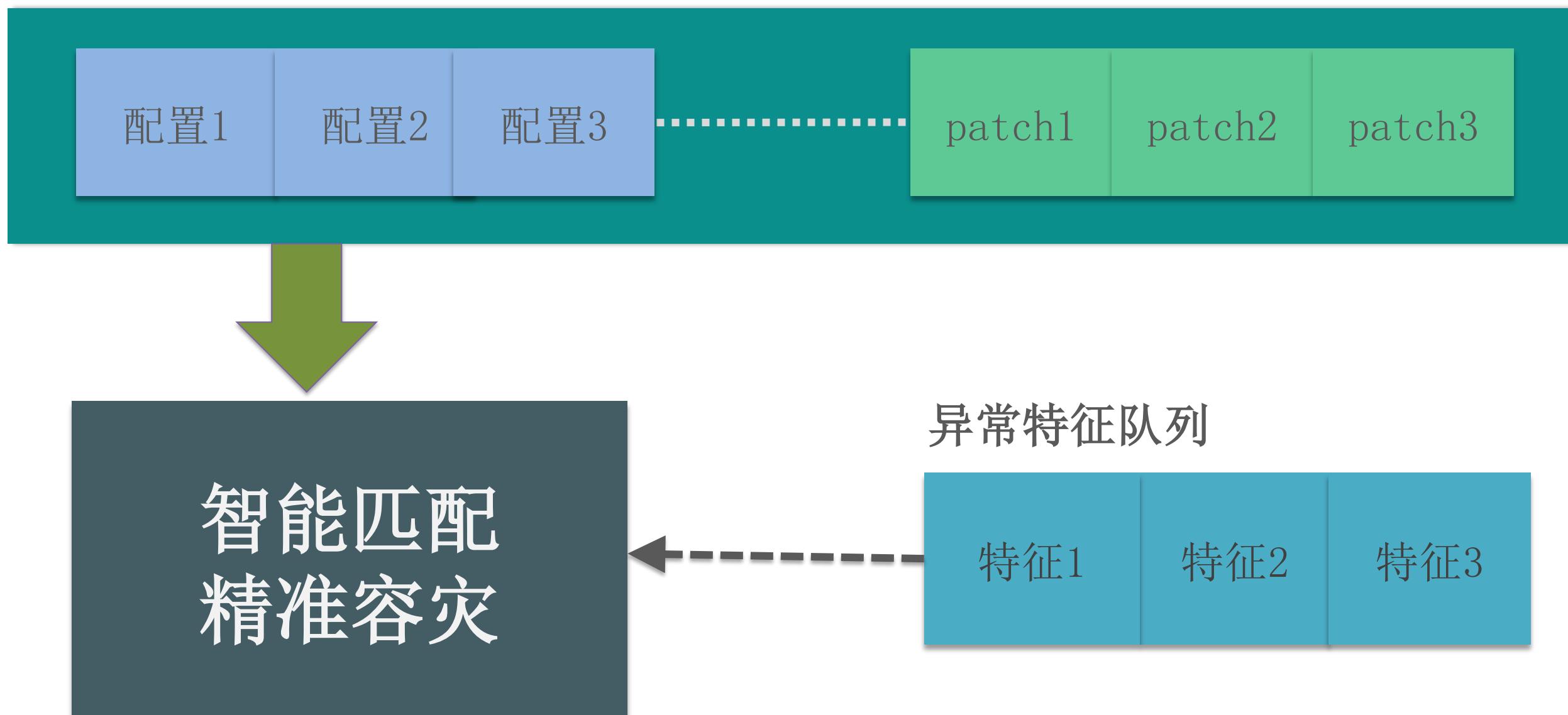
Hotpatch

- . 99%成功率, sync + pull结合
- . dalvik、art、yunos全支持
- . 仅仅2-3k, patch包

Native

- . 全功能的模块化发布, 周覆盖率90%+

架构的容灾能力



THANKS