

The background of the slide is a dark blue gradient. In the upper right, there is a faint, semi-transparent image of a laptop screen. The screen displays a line chart with a fluctuating line and a downward-sloping trend line. Below the chart, there is a pie chart with several segments. The overall aesthetic is professional and data-oriented.

APKBUS

深度挖掘APP质量大数据

夏鸣远

</> 现有性能优化方案

采集过CPU/内存/FPS , Activity onXXX

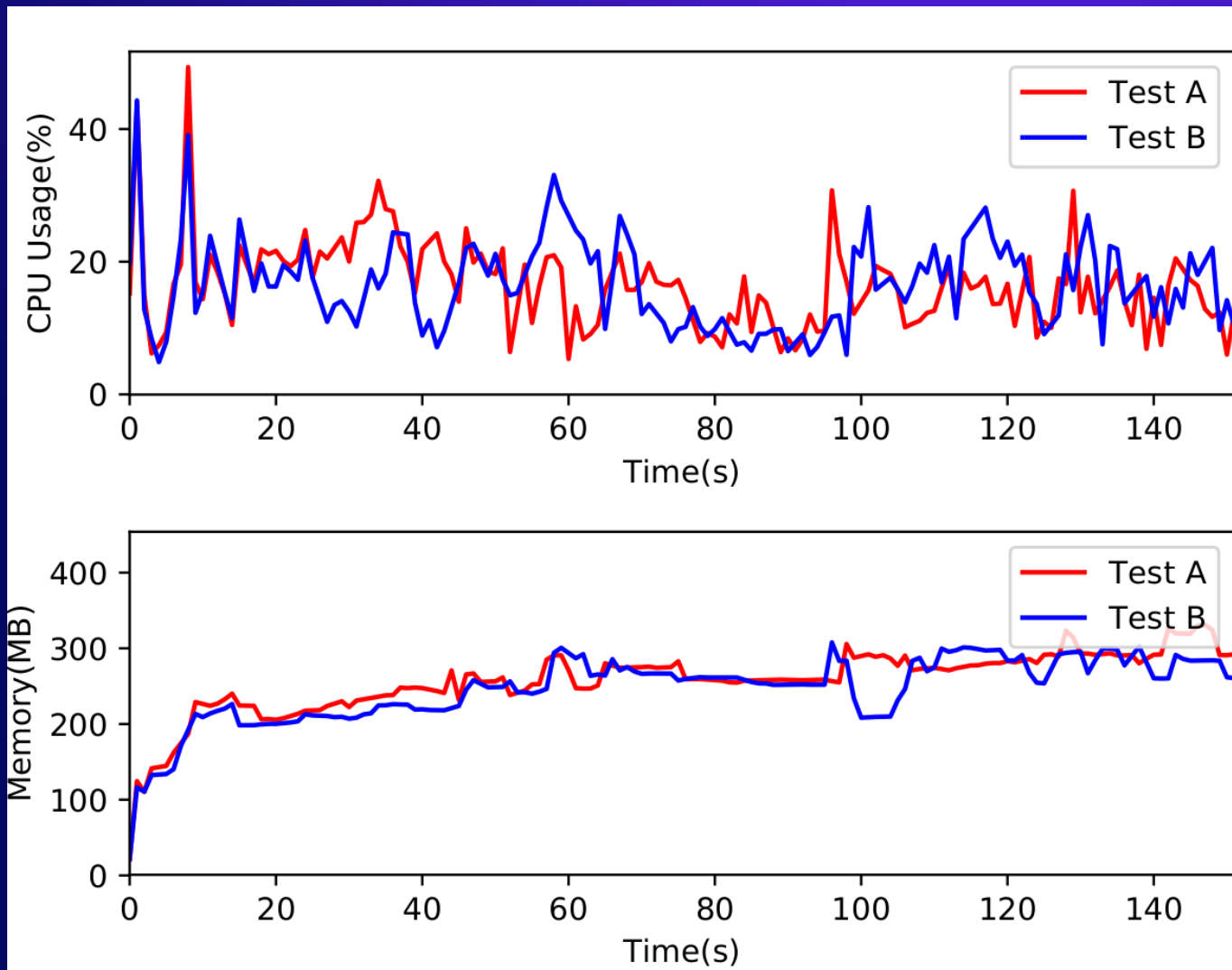
adb shell dumpsys xxx | findstr

“接入过一些SDK”

Excel可视化 对比

优化指导千篇一律：onXXX 少做点事情

</> 性能实验



同手机，APP冷启动，WiFi
录制重放屏幕操作
重放两次
指标的平均/最差并不稳定

平均值指导“性能总体情况”不科学
输入频率影响很大，本身不稳定
最差值指导“性能问题”
难以诊断问题原因
性能指标需要结合APP打点来看

</> 性能优化方案需求

基于打点数据的问题定位


问题原因

考虑多机型/运行环境

</> 数据驱动的APP性能优化



深度挖掘APP质量大数据

- 
- 1 • 数据驱动的APP性能优化
 - 2 • 从数据到问题到原因
 - 3 • 指标、回归和制度建设
 - 4 • DevOps: 线上性能Bug反馈的建设

</> 打点数据收集

- SDK二次开发性能打点
 - 腾讯GT
 - 滴滴DoraemonKit
 - AndroidGodEye
- DEX插桩方案自动打点
 - Appetizer

	SDK	Appetizer
ANR	✓	✓
CPU	✓	✓
内存	✓	✓
帧率	✓	✓
主线程卡顿	-	✓
耗时API调用	×	✓
Bitmap API	×	✓
Layout API	×	✓
网络请求基本	✓	✓
网络请求延迟细分		✓
界面路径	✓	✓
Fragment路径		✓
Jacoco覆盖率		✓
ADB控制		✓
数据形态	二次开发	统一JSON

</> 打点位置

- App内
 - ActivityLifecycleCallbacks
 - Fragment
 - Okhttp Interceptor或者配置抓包工具
 - 主线程(Looper.setMessageLogging)
 - ANR
- 其他数据
 - 定时截图
- logcat GC

</> 多设备/运行环境模拟



小米Mix2
2017年全面屏
骁龙835
6G内存
Android 8.0

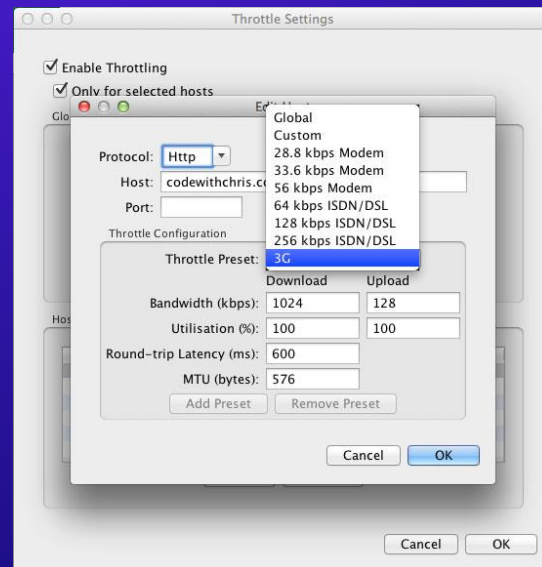


华为m3平板
2016年平板
麒麟950
4G内存
Android 7.0



三星Note4
2014年16:9
骁龙805
3G内存
Android 6.0

机型硬件



网络模拟（弱网）



后台进程

</> 性能数据管理

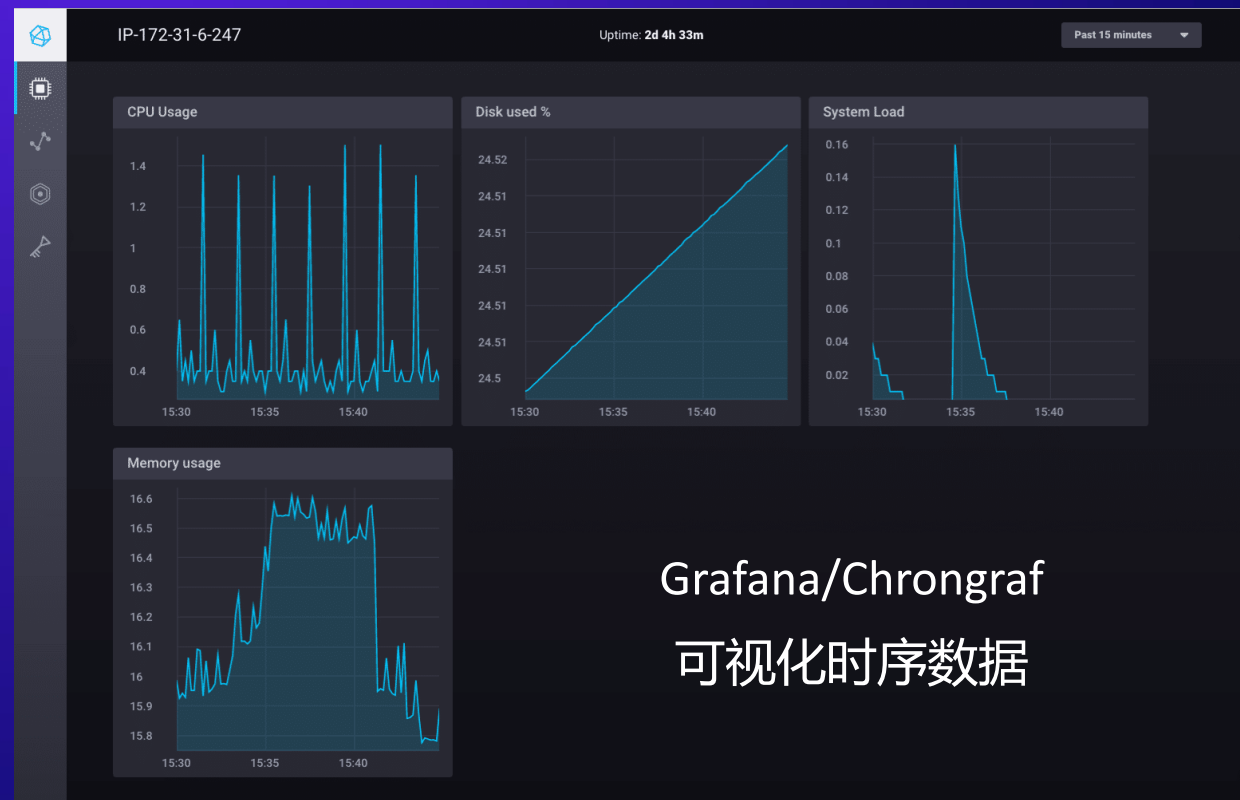
- InfluxDB
- 存取时序数据，提供类SQL查询接口

```
> INSERT cpu,host=serverA,region=us_west value=0.64
```

```
> SELECT "host", "region", "value" FROM "cpu"
```

name: cpu

time	host	region	value
2015-10-21T19:28:07.580664347Z	serverA	us_west	0.64



Grafana/Chronograf
可视化时序数据

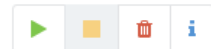


支持类SQL查询的时序数据库

<https://www.influxdata.com/>

开源 Docker部署 Windows Linux MacOS

了解一下



手工添加服务

? 有问题



漂亮的监控和时序数据图标

<https://grafana.com/>

开源 Docker部署 部分汉化 Windows Linux MacOS

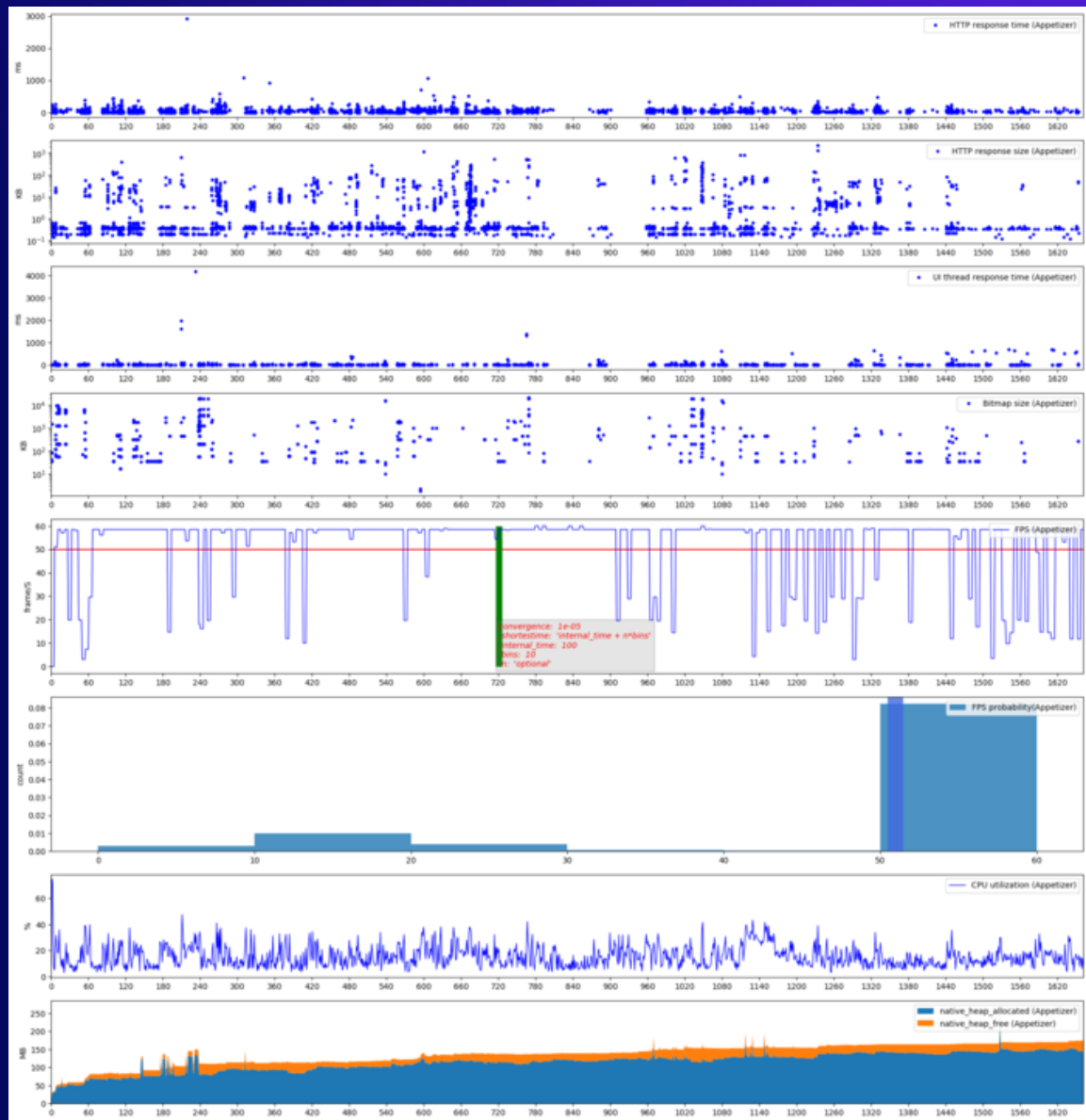
了解一下



手工添加服务

? 有问题

</> 插桩数据分析



<https://github.com/appetizerio/insights.py>

交互式的问题排查 + 排查面板 (演示视频)





选择模版

blank

a blank template

nova

nova, a powerful build-in template, provides CPU Usage charts, Memory charts, crash charts and just everything you need! We highly recommend it to all new starters, which would be very

本地模板



</> 如何去除重复问题

- FPS<30 , 倒推5秒 -> 一个性能问题
- 判定重复的标准
 - 同业务 (Activity/Fragment) +同前置条件 (某个控件点击)
 - 前置后端API请求 <http://xxx/v3/product/list>
- 自动化去重复后的形式
 - 哪个数据
 - timestamp

```
# ls -l
20190409-业务回归-MiMix2-3G-time-1554792023.json
20190411-业务回归-HuaweiP10-wifi-time-1554792023.json
20190411-业务回归-HuaweiP10-3G-time-1554792023.json
20190412-业务回归-MiMix2-3G-time-1554792023.json
...
```

深度挖掘APP质量大数据

1 • 数据驱动的APP性能优化

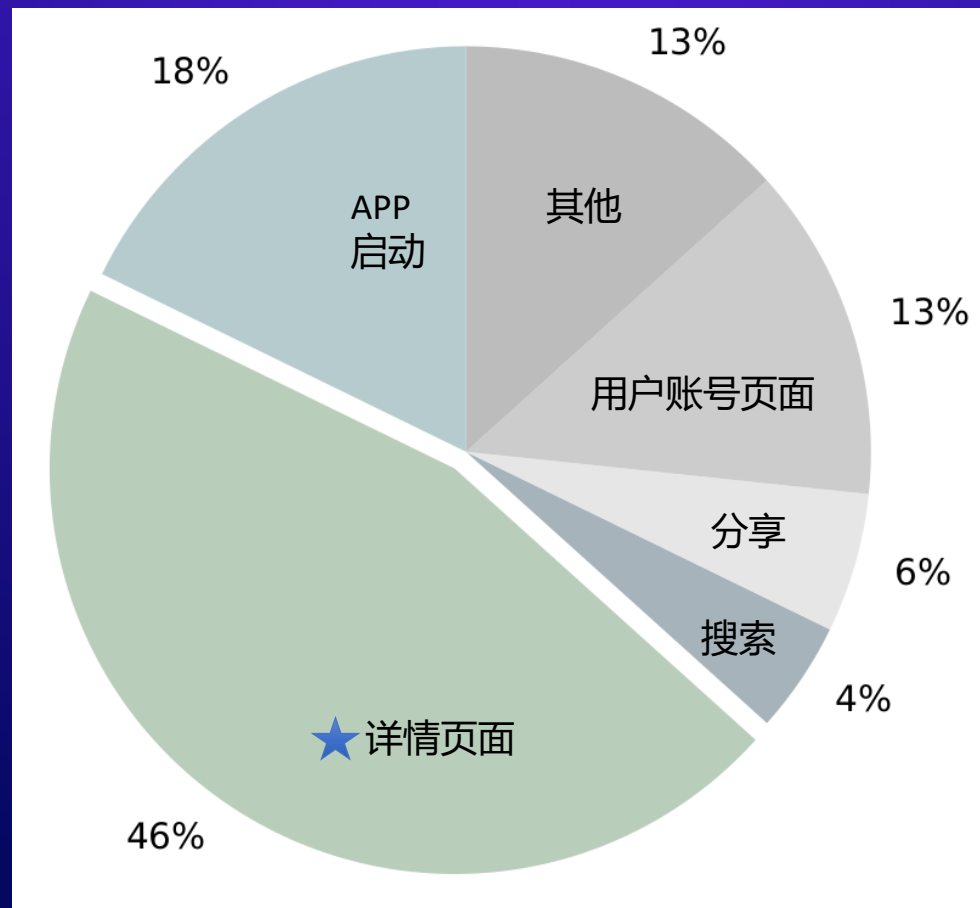


2 • 从数据到问题到原因

3 • 指标、回归和制度建设

4 • DevOps: 线上性能Bug反馈的建设

</> 性能问题频发的场景



</> 案例分析1：详情页过度预取

主页 文章 视频



文史茶馆 · LV1新生
2019/03/07 20:20:03

发布了新视频



@文史茶馆：主政这个省二十三年红旗不倒，这位将军为何没有军衔



赞



评论



转发



文史茶馆 · LV1新生
2019/03/07 13:47:42

发布了新文章



@文史茶馆：4位湖南籍蒋介石嫡系爱将，晚年都加入一个共同的机构



赞



评论



转发



文史茶馆 · LV1新生
2019/03/07 13:14:24

发布了新视频



@文史茶馆：给刘伯承当过政委，长征只走了一半，他后来成为正国级



赞



评论



转发

3 Views
Visible

8 Views
Invisible

ImageView

</> 案例分析1：详情页过度预取

用户只看见三项
却需要等待11项抓取



过度预取 = 将未来的卡提前到现在

ImageView

主页 文章 视频

文史茶馆 · LV1新生
2019/03/07 20:20:03

发布了新视频



@文史茶馆：主政这个省二十三年红旗不倒，这位将军为何没有军衔



文史茶馆 · LV1新生
2019/03/07 13:47:42

发布了新文章



@文史茶馆：4位湖南籍蒋介石嫡系爱将，晚年都加入一个共同的机构



文史茶馆 · LV1新生
2019/03/07 13:14:24

发布了新视频



@文史茶馆：给刘伯承当过政委，长征只走了一半，他后来成为正国级



3 Views
Visible

8 Views
Invisible

</> 案例分析1：详情页过度预取



用户只看见三项
却需要等待11项抓取



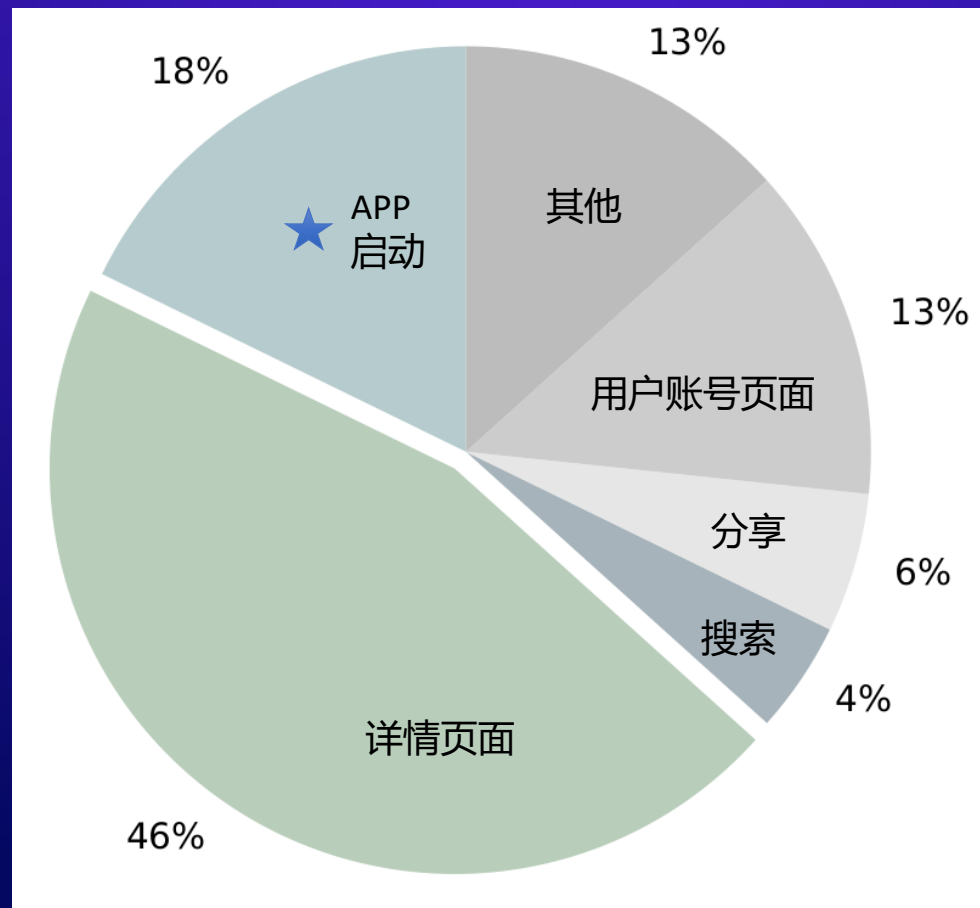
过度预取 = 将未来的卡提前到现在

尽快渲染用户
可见的内容



根据停留时间
渐进预取 设置预取buffer
上限

</> 性能问题频发的场景



</> 案例分析2：APP冷启动

统计信息

冷启动时间	6714ms	
CPU利用率	7.36% 平均	19.90% 最大
内存使用率	35 MB 平均	44 MB 最大
网络请求	32	1.93 MB
网络响应时间	184.66ms 平均	938ms 最大
UI卡顿	无	
主线程耗时调用	1	250ms 总计时间

最慢HTTP请求

文字内容

其他流量

图片流量

协议流量

详细情况	响应时间	URL
查看	445ms	https://img.alicdn.com/imgextra/i1/2206681511511/home_page_head
查看	415ms	https://img.alicdn.com/imgextra/i1/2206681511511/appservice
查看	390ms	https://img.alicdn.com/imgextra/i1/2206681511511/topgoods_tabs
查看	389ms	https://img.alicdn.com/imgextra/i1/2206681511511/index_appkey_and_title
查看	195ms	https://img.alicdn.com/imgextra/i1/2206681511511/msg_count?category=1

</> 案例分析2：APP冷启动

详细加载流程



三方库初始化 x 4次

重复的GET请求 x 4次

BI打点 x 4次

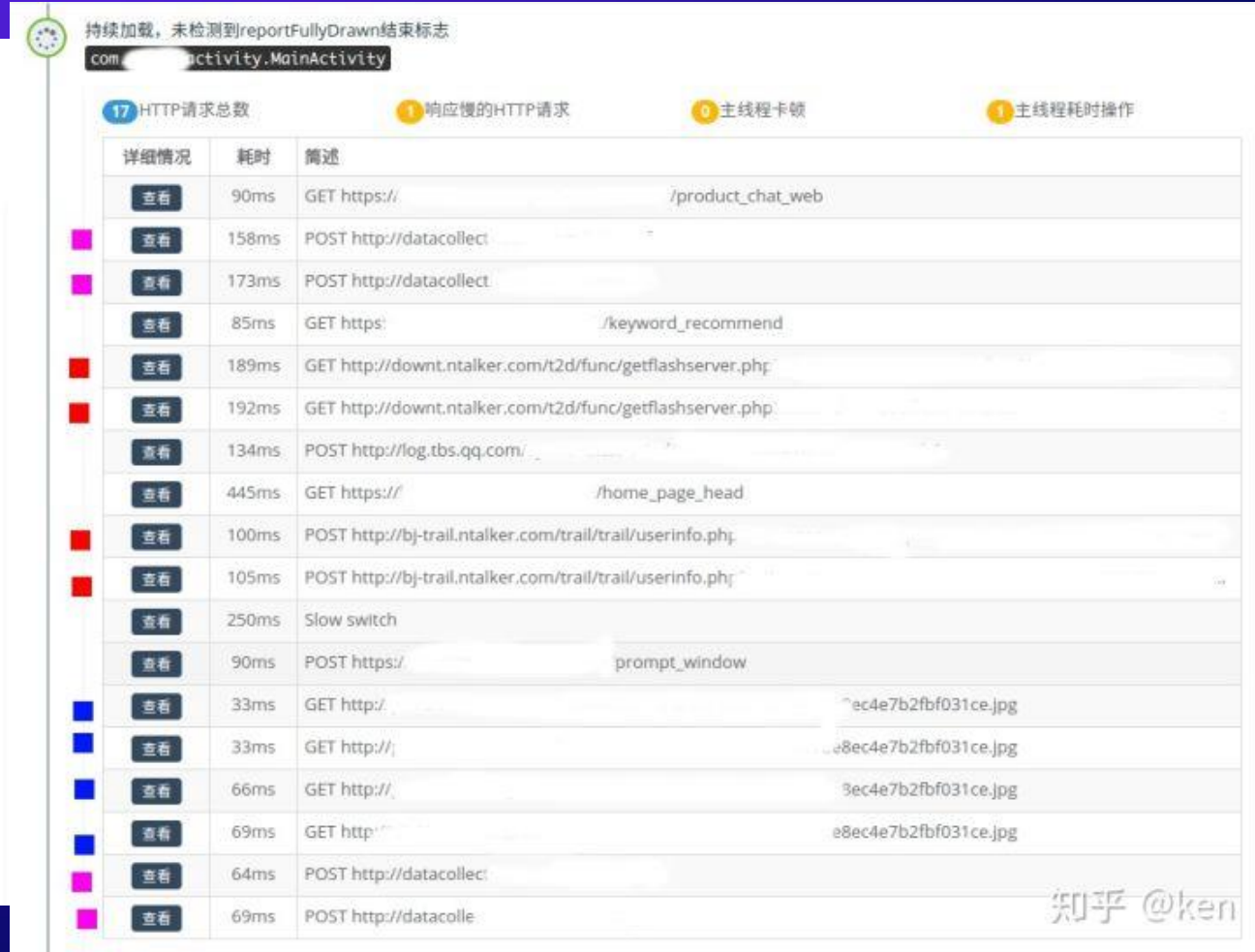
BI三方库 x 4次

三方库初始化后置

客户端缓存 x 1次

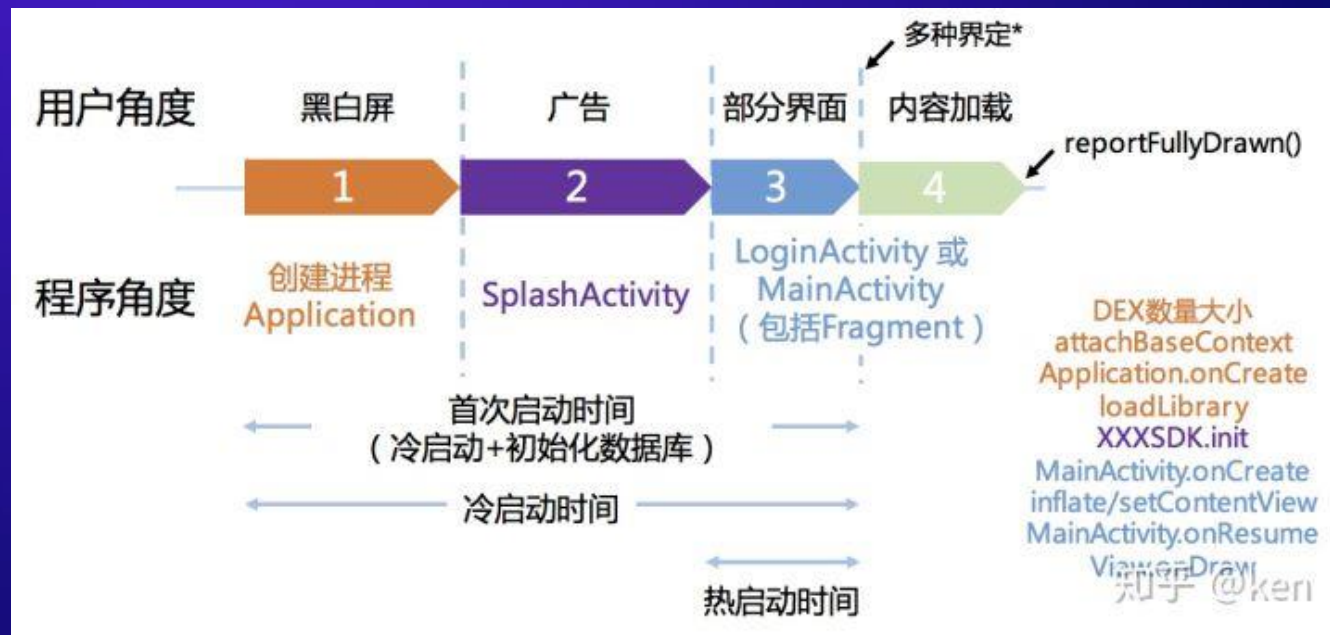
合并打点 x 1次

BI三方库后置

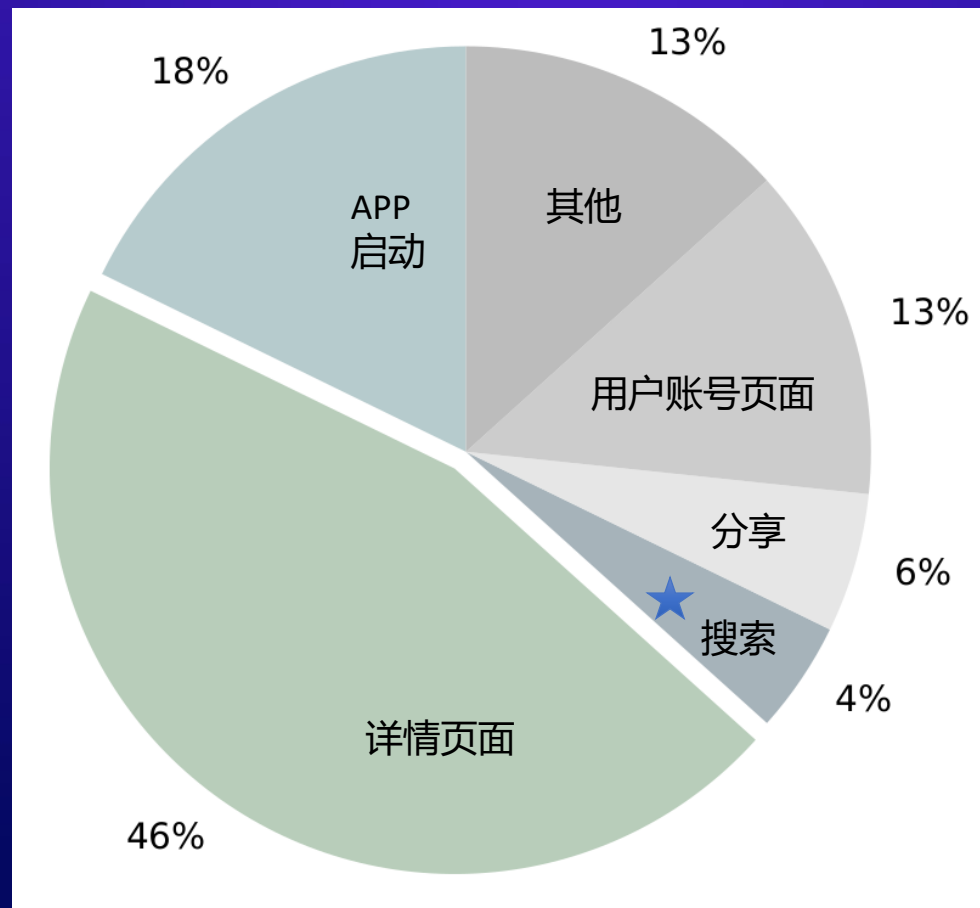


</> 案例分析2：APP冷启动

- 网络类问题
 - 三方库隐形成本，避免启动时不必要的初始化
 - 并发TCP连接限制
 - TCP建联成本
- 静态资源优化
 - content encoding: gzip
 - CDN自动转码：
xxx.png?width=400&height=600
 - 避免频发的小请求 -> 移动网络延迟高



</> 性能问题频发的场景

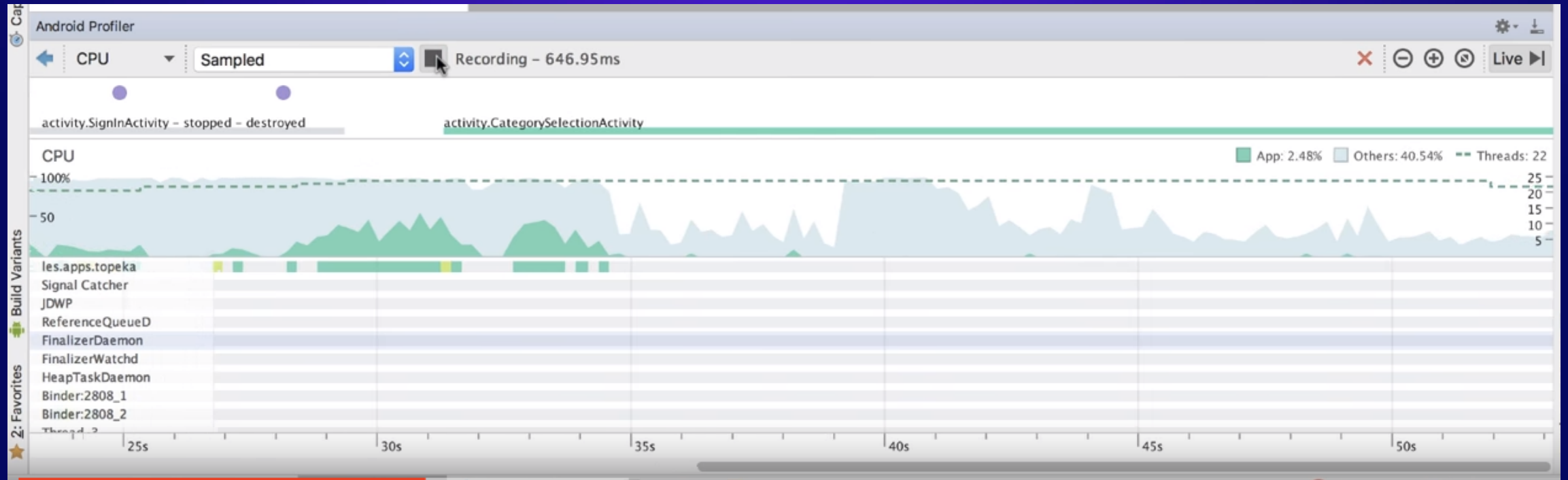


</> 搜索推荐卡顿

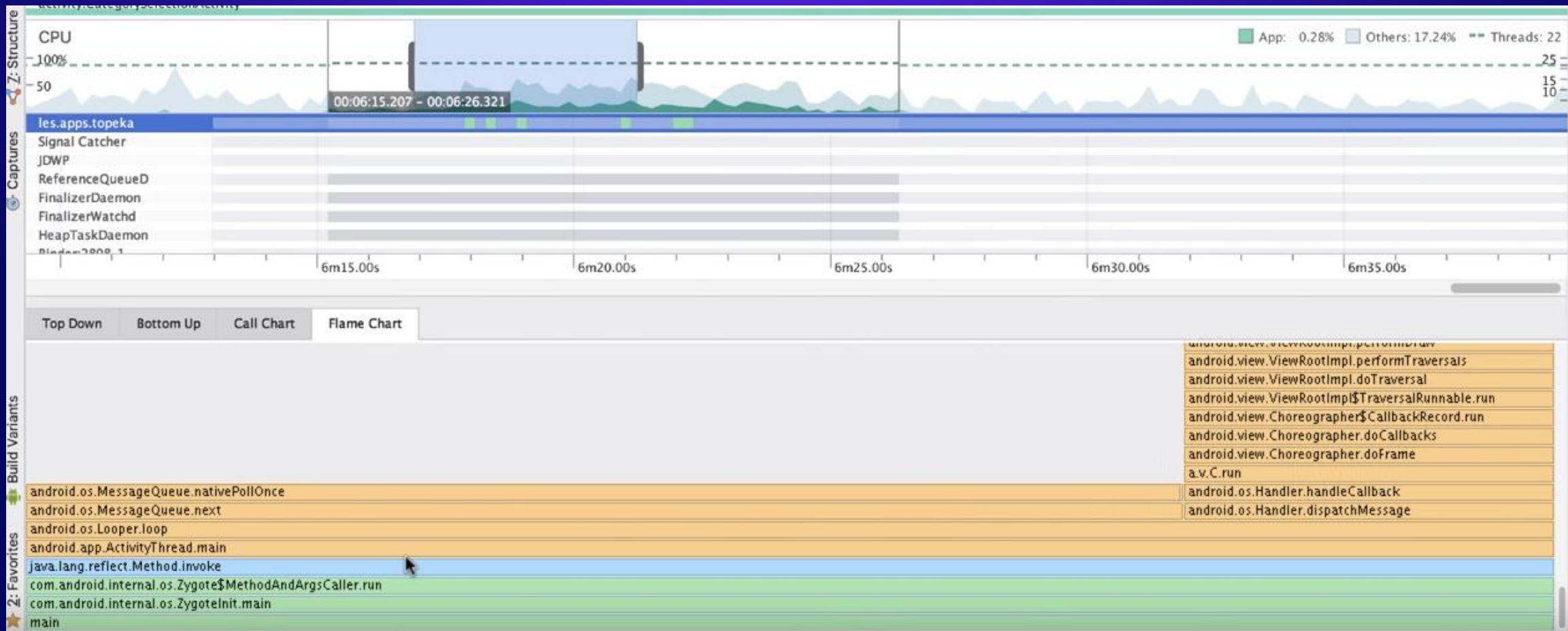


- 网络请求不多
- Bitmap也没有操作
- 主线程事件多累计耗时大

</> Android Profiler

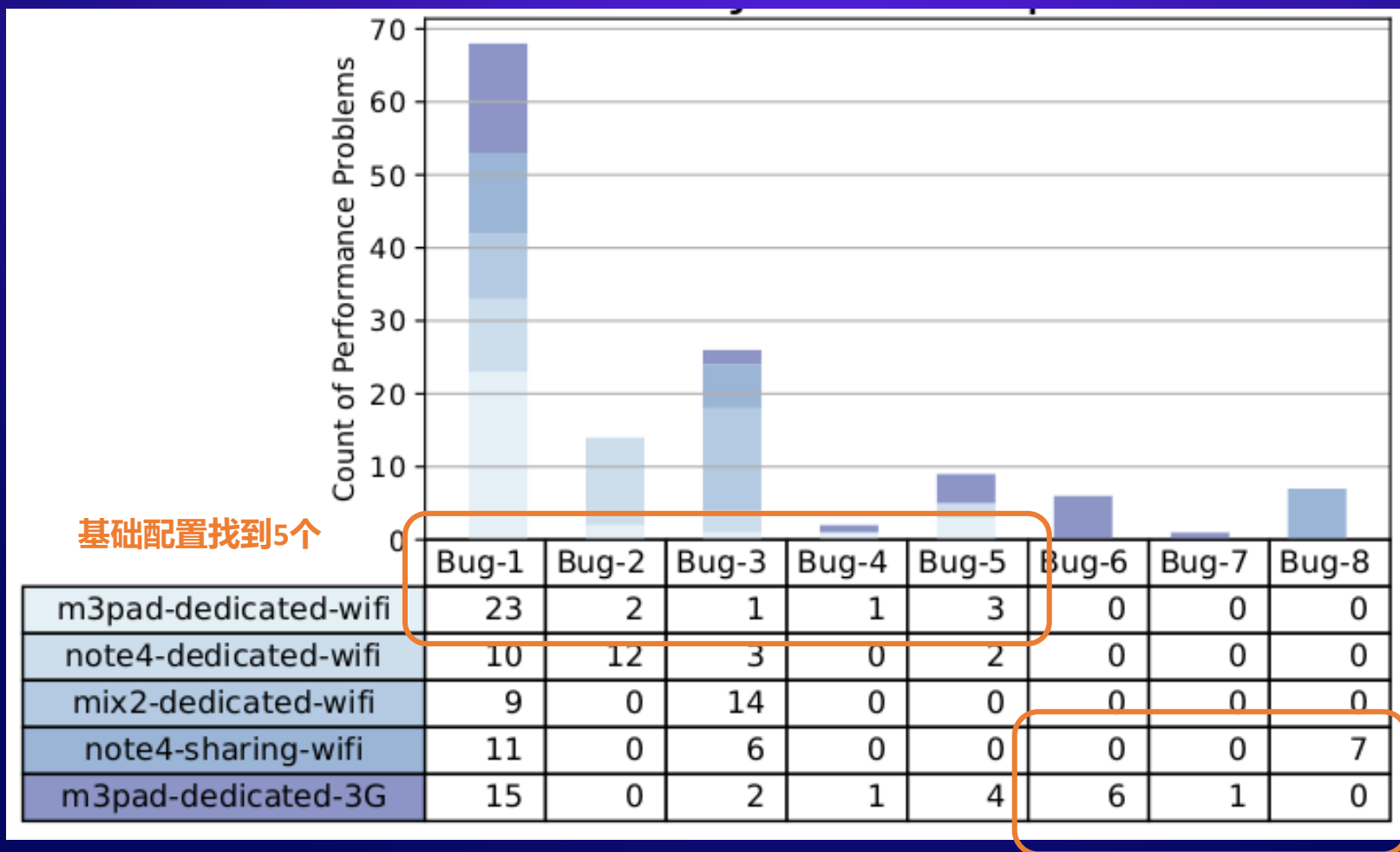


</> Android Profiler



性能开销较大
信息量大
仅适用排查已确定的CPU计算长耗时问题

</> 多设备测试结果

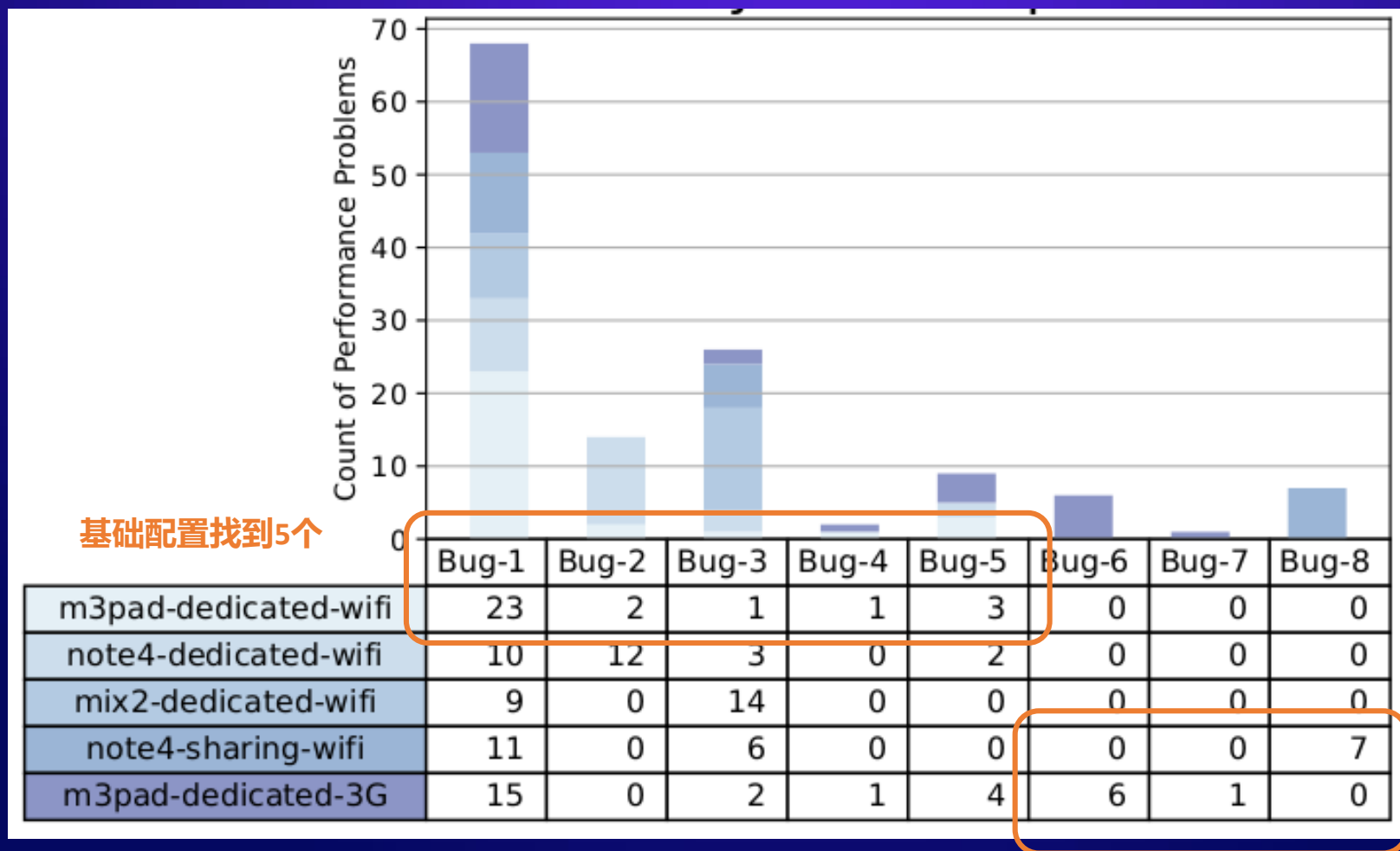


基础配置找到5个

弱网情况下找到2个新Bug

老手机情况下找到1个新Bug

</> 多设备测试结果



弱网情况下找到2个新Bug

老手机情况下找到1个新Bug

在时间有限的情况下：采用中等配置手机 + 3G + 开多后台进程

有条件的情况下：使用多安卓版本 + 自动化脚本多配置执行

深度挖掘APP质量大数据

1 • 数据驱动的APP性能优化

2 • 从数据到问题到原因

→ 3 • 指标、回归和制度建设

4 • DevOps: 线上性能Bug反馈的建设

</> 指标/回归/制度建设

- 硬指标：FPS多少算是卡？主线程延迟多少算是卡？
 - 回到了不稳定的平均/最差问题

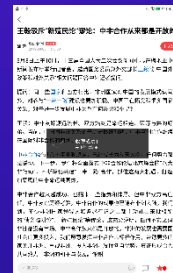
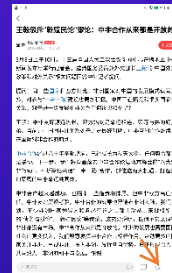
</> 指标/回归/制度建设

- 硬指标：FPS多少算是卡？主线程延迟多少算是卡？
 - 回到了不稳定的平均/最差问题
- 统筹优化最差：
 - 小目标：FPS指标上Worst 5%
 - 中期目标：N个业务场景无性能问题，2014-2019机型硬件跨度上的性能问题减半
 - 长期目标：捕获线上性能问题

</> 指标/回归/制度建设

- 硬指标：FPS多少算是卡？主线程延迟多少算是卡？
 - 回到了不稳定的平均/最差问题
- 统筹优化最差：
 - 小目标：FPS指标上Worst 5%
 - 中期目标：N个业务场景无性能问题，2014-2019机型硬件跨度上的性能问题减半
 - 长期目标：捕获线上性能问题
- 性能问题 -> 功能化 -> 加入发版看板
 - Issue正视性能Bug

- 复现步骤

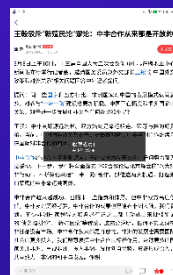
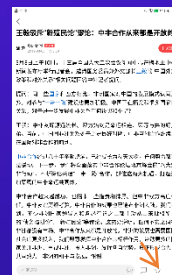
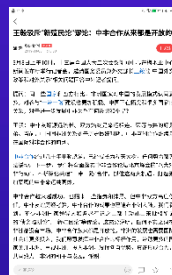


- 问题描述

- 数据Log

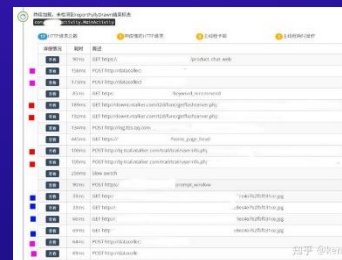
Issue

- 复现步骤



- 问题描述

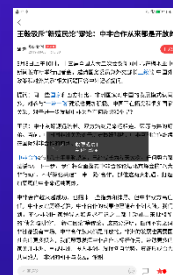
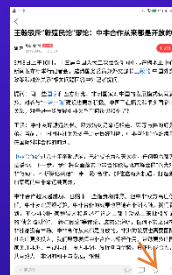
- 三方库初始化 x 4次
- 重复的GET请求 x 4次
- BI打点 x 4次
- BI三方库 x 4次



- 数据Log

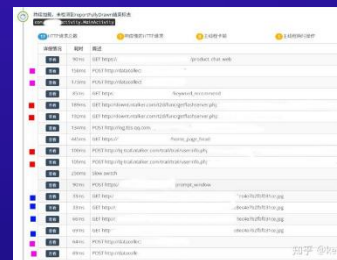


- 复现步骤



- 三方库初始化 x 4次
- 重复的GET请求 x 4次
- BI打点 x 4次
- BI三方库 x 4次

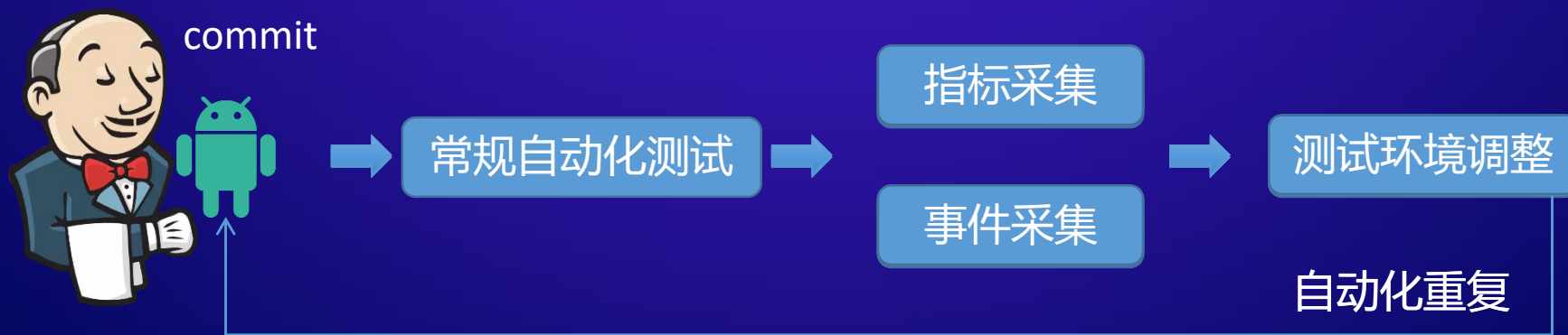
- 问题描述



- 数据Log



</> 性能进入回归



深度挖掘APP质量大数据

1 • 数据驱动的APP性能优化

2 • 从数据到问题到原因

3 • 指标、回归和制度建设

→ 4 • DevOps: 线上性能Bug反馈的建设

</> DevOps: 线上性能Bug反馈

- 线上才是主战场
 - 用户机型 (广度)
 - 用户网络 (移动网络/路边WiFi)
 - 用户系统环境 (各厂商定制版本/各路保活APP)

</> DevOps: 线上性能Bug反馈

- 线上才是主战场
 - 用户机型（广度）
 - 用户网络（移动网络/路边WiFi）
 - 用户系统环境（各厂商定制版本/各路保活APP）
- 如何把性能数据监控做到线上？
 - ANR
 - 线上主场景流畅度（响应时间）

</> DevOps: 线上性能Bug反馈

- 线上才是主战场
 - 用户机型 (广度)
 - 用户网络 (移动网络/路边WiFi)
 - 用户系统环境 (各厂商定制版本/各路保活APP)
- 如何把性能数据监控做到线上？
 - ANR
 - 线上主场景流畅度 (响应时间)
- 如何复现线上性能问题
 - 业务路径
 - 交互路径
 - 聚类，首先筛选环境

</> 总结

- 性能数据打点收集：SDK或者插桩
- 交互式诊断问题：搜索指标 -> 寻找热点 -> 排查
- 常见问题：详情页过度预取，冷启动网络，搜索推荐过量计算
- 采用中等配置手机 + 3G + 开多后台进程
- 多设备+网络环境模拟：问题数+60%~80%
- 统筹优化最差，Jenkins
- 线上ANR -> 线上流畅度 -> 以业务+交互路劲聚类

</> 总结

- 性能数据打点收集：SDK或者插桩
- 交互式诊断问题：搜索指标 -> 寻找热点 -> 排查
- 常见问题：详情页过度预取，冷启动网络，搜索推荐过量计算
- 采用中等配置手机 + 3G + 开多后台进程
- 多设备+网络环境模拟：问题数+60%~80%
- 统筹优化最差，Jenkins
- 线上ANR -> 线上流畅度 -> 以业务+交互路劲聚类

谢谢



</> 欢迎关注安卓巴士公众号



www.apkbus.com