

< HDC.Together >

HUAWEI DEVELOPER CONFERENCE 2021

< HDC.Together >

华为开发者大会 2021

HarmonyOS 图形栈测试技术深度解析

1

HarmonyOS自研图形栈整体架构和测试能力

2

华为ArkUI开发框架和测试技术

- 实际案例—图库分析调优（现场演示）

3

华为自研3D图形栈和测试技术

- 实际案例—壁纸制作分析调优（现场演示）

4

Devco Testing图形测试能力介绍

HarmonyOS发布会上的自研图形栈



HarmonyOS自研图形栈背后的“黑科技”到底有什么？

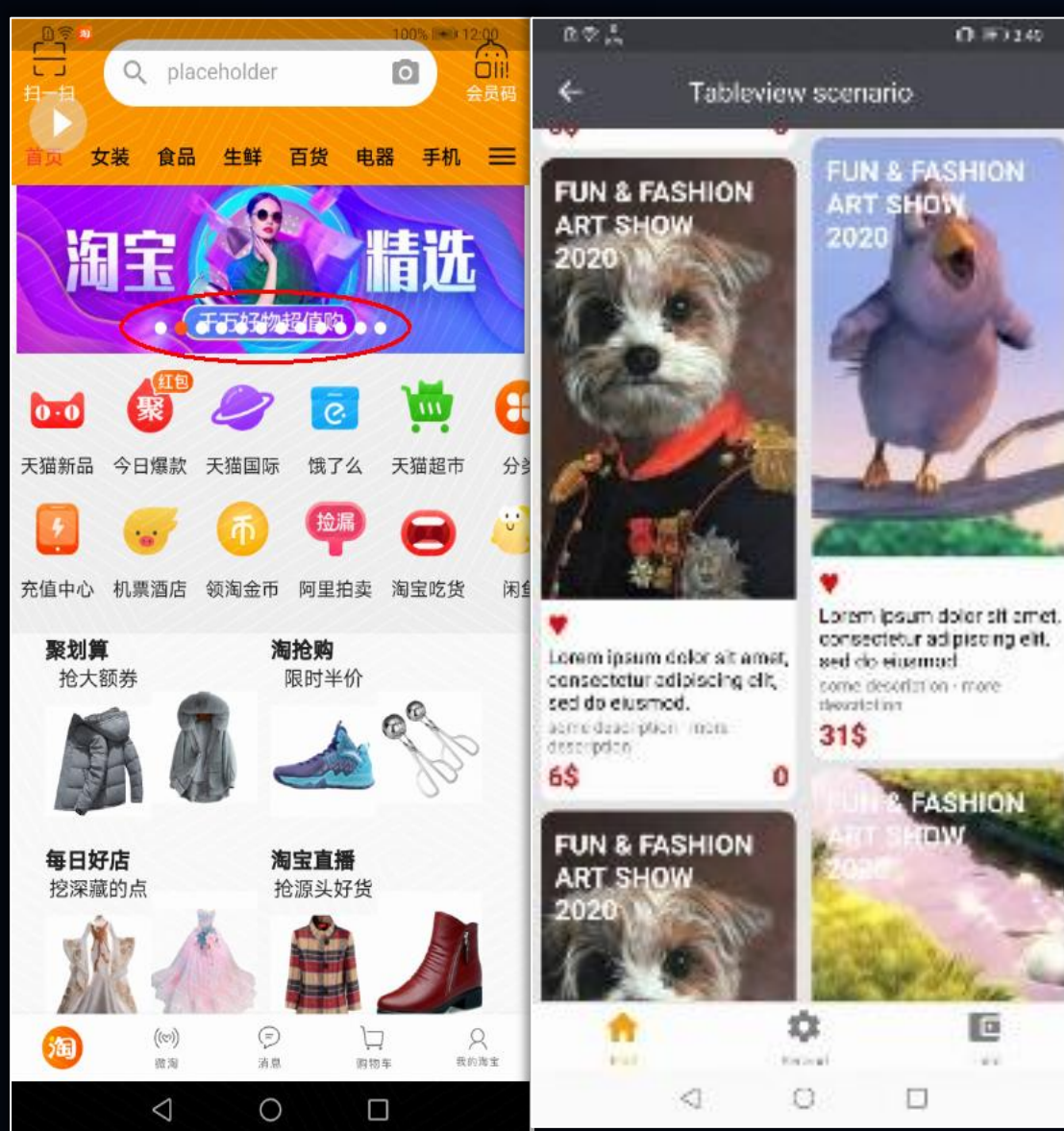
HarmonyOS 图形栈整体架构和测试能力



帮助开发者真正打通从“应用->系统->硬件”的全栈测试分析调优通道

HarmonyOS应用开发者在测试中遇到的痛点

HarmonyOS 2D应用



问题:

1. 开发完一个HarmonyOS应用, 怎么做性能分析?
2. HarmonyOS应用性能分析工具能不能告诉我优化建议?
3. 开发一个HarmonyOS ArkUI应用, 有参考示例吗?

HarmonyOS 3D应用



问题

1. 如何能够帮助我理解掉帧的背后根因?
2. 有没有工具链帮助我做SOC (CPU/GPU/DDR) 的细粒度分析?
3. HarmonyOS 3D图形栈SDK使用中, 是否有工具辅助调优?

测试不仅仅是数据搬运, 更是体现数据背后的深度分析和优化建议

1

HarmonyOS自研图形栈整体架构和测试能力

2

华为ArkUI开发框架和测试技术

- 实际案例—图库分析调优（现场演示）

3

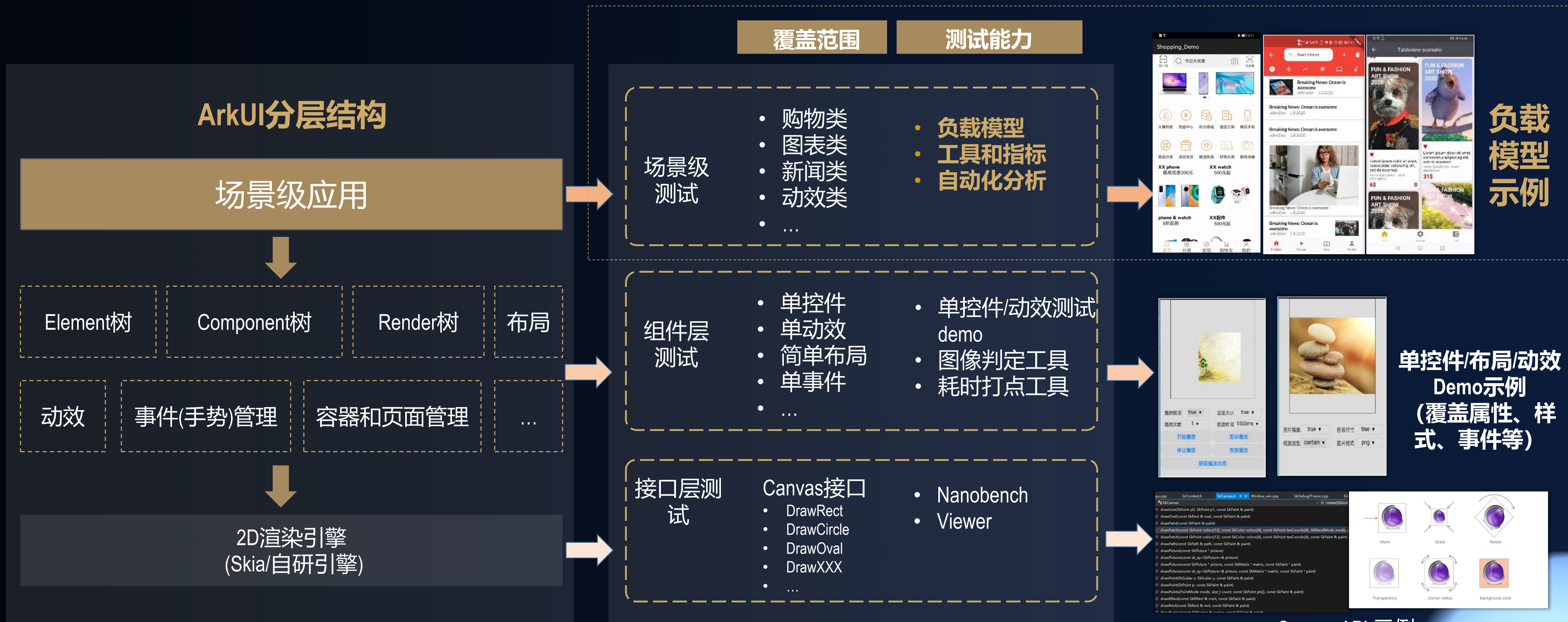
华为自研3D图形栈和测试技术

- 实际案例—壁纸制作分析调优（现场演示）

4

Devco Testing图形测试能力介绍

华为ArkUI开发框架和场景级测试



场景级别的测试和分析能力是开发者最为关注的

2D负载模型 (框架验证+Golden App)

| TOP200 | 场景分类(20类) | UI负载特征描述 | UI负载抽象 |
|--------|----------------------------|---------------------------------|---------|
| 微信 | 综合资讯 (如今日头条、腾讯新闻、网易新闻等) | 特征1 内外层tab嵌套, 内层多tab切换 | 组件、布局嵌套 |
| QQ | | 特征2 图片、视频、文字混合长列表 | 资源加载 |
| 今日头条 | | 特征3 单列多视频播放长列表 | 资源加载 |
| 抖音 | | 特征4 多列多视频播放长列表 | 资源加载+布局 |
| 网易新闻 | | 特征5 横向多视频播放长列表 (视频轮播, 可视区域播放) | 资源加载+布局 |
| 网易云音乐 | | 特征6 图片、文字混合长列表 | 组件、布局嵌套 |
| 新浪微博 | | 特征7 静态字体颜色、样式、大小切换 (阅读文本的显示) | 资源加载 |
| 高德地图 | 社交网络 (如新浪微博、百度贴吧等) | 特征1 应用内多图层叠加 (广告叠加到浏览页面上) | 资源加载 |
| QQ阅读 | | 特征2 页面切换动画 (阅读浏览页面翻页动画) | 图层叠加 |
| 百度地图 | | 特征3 动态字体颜色、样式、大小, 弹幕叠加文本、图片 | 动效绘制 |
| 腾讯新闻 | | 特征4 多图片拼接嵌入横向长列表或纵向长列表 | 图层叠加 |
| 哔哩哔哩 | | 特征5 图片和视频九宫格布局 (结合文字) 嵌入长列表 | 资源加载+布局 |
| 斗鱼直播 | | 特征6 高清图片切换浏览 | 组件、布局嵌套 |
| 快手 | | 特征7 应用内多图层叠加 (如探探卡片) 负载; 拖动单张负载 | 资源加载+布局 |
| 爱奇艺 | 电子商务 (如淘宝、拼多多、京东等) | 特征1 手势按压拖动图片资源 | 资源加载 |
| 百度 | | 特征2 多图片轮播 | 图层叠加 |
| 支付宝 | | 特征3 多图片、多视频组合成网格布局, 结合文本嵌长列表 | 手势事件 |
| UC浏览器 | | 特征4 视频、图片、文本混合瀑布流网格布局 | 组件、布局嵌套 |
| 拼多多 | | 特征5 多列多图片长列表 | 资源加载+布局 |
| | | | |

- ### 八类负载场景
- **购物类**: 资源复杂 (图片、视频、文字)、组件嵌套多样等。
 - **图库类**: 月视图、手势缩放。
 - **视频类**: 视频小窗口多图层、多视频。
 - **资讯类**: 字体库、歌词图层等。
 - **手势类**: 复杂操作。
 - **动效类**: 平移/旋转/缩放/透明度变化。
 - **地图类**: 2D绘制场景。
 - **图表类**: 复杂绘制场景。

- ### 六类负载类型
- **组件嵌套复杂**
如: 标签栏嵌套
列表嵌套
多容器嵌套
 - **资源复杂**
如: 文字 (字体库)
图片 (格式)
音视频 (格式)
 - **操作复杂**
如: 按下滑动
双指捏合
 - **图层叠加**
如: 路由多图层覆盖
窗口视频
窗口文字+图片
 - **绘制复杂**
如: 各类复杂动效
 - **后台多线程**
如: 重IO线程
重网络线程

TOP200构建

- 构建量大
- 重复性强
- 能力缺失

-->需要抽象

正向: TOP应用UI负载抽象

- 简要描述: 三方推动**百度贴吧**优化图层。下周五前解决, 不解决预警**应用**市场下架**百度**app。当前**百度贴吧**卡顿舆情41例, 为性能舆情TOP2。
- 简要描述: 【Merida-L11 10.1.0.64(SP10C900E22R1P1)_超级应用性能测试_必现】**淘宝**首页滑动**卡顿**丢帧明显, 测试平均值率45fps (com.taobao.taobao 9.3.0)
- 简要描述: 【应用/微博/性能 (卡顿/启动慢/延时等) /应用使用时卡顿/延时】**新浪****微博**滑动时界面不流畅
- 简要描述: 【第三方应用洞察】【RAI-AN00 11.0.0.102(C00E102R5P2log)】**爱奇艺**V11.9.0**爱奇艺**快进之后播放视频出现**卡顿**约1s, 对比竞品手机三星Note 10、iPhone XS不存在

逆向: 收集已知应用性能问题

通过已知的应用UI相关问题, 逆向印证问题属于以上六类负载类型

负载模型参数化适配多设备

The image displays eight different load configuration screens, each titled '负载配置' (Load Configuration). The settings vary by scenario:

- Top Left:** Subtitle '有弹幕' (With Subtitle) selected. Options for font (lato, roboto, montserrat), color (white, green, yellow), and transparency (1.0, 0.5, 0.2). Includes fields for '弹幕数量' (100) and '弹幕速度' (10).
- Top Middle-Left:** '原图' (Original Image) selected. Fields for '列数' (8) and '张数' (5000).
- Top Middle-Right:** '标记点是否显示' (true/false), '是否开启图表动画' (true/false), '图表数量' (5), '图表叠加的层数' (2), '标记点间隔' (25), '动画播放的速度' (3).
- Top Right:** '有阴影' (With Shadow) selected. '有圆角' (With Rounded Corners) selected. 'item总数' (30), '列数' (2), '视频数量' (4).
- Middle Right:** Longitude (28.99934 W), Latitude (41.02747 S), '初始放大倍数' (8.0), '定位标记数量' (2), '折线数量' (2), '多边形数量' (2), '圆形边界数量' (6).
- Bottom Left:** '有阴影' (With Shadow) selected. '有圆角' (With Rounded Corners) selected. '无旋转' (No Rotation) selected. '资源选择' (image/video/text), '手势选择' (drag/其它), '卡片总数量' (100).
- Bottom Middle-Left:** '有阴影' (With Shadow) selected. '有圆角' (With Rounded Corners) selected. '无旋转' (No Rotation) selected. '资源选择' (image/video/text), '手势选择' (drag/其它), '卡片总数量' (100).
- Bottom Middle-Right:** '有阴影' (With Shadow) selected. '有圆角' (With Rounded Corners) selected. '无旋转' (No Rotation) selected. '资源选择' (image/video/text), '手势选择' (drag/其它), '卡片总数量' (100).
- Bottom Right:** '原图' (Original Image) selected. '列数' (8), '张数' (128).

负载参数可调

- 资源种类 (图片/视频格式、字体库)
- 资源数量
- 资源大小
- 渲染参数(阴影、透明度等)
- 曲线参数
- ...

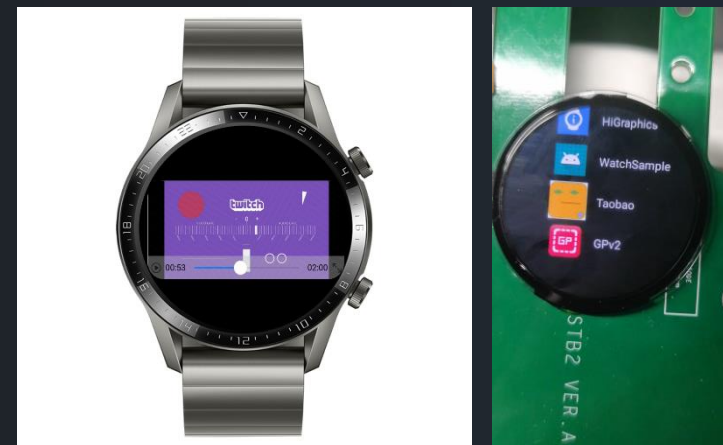
负载参数可调实现了设备“1+8+N”的覆盖测试

负载参数化, 实现资源、布局、渲染复杂度的负载调试, 可以支撑1+8+N设备

负载模型支持“1+8+N”设备



车机



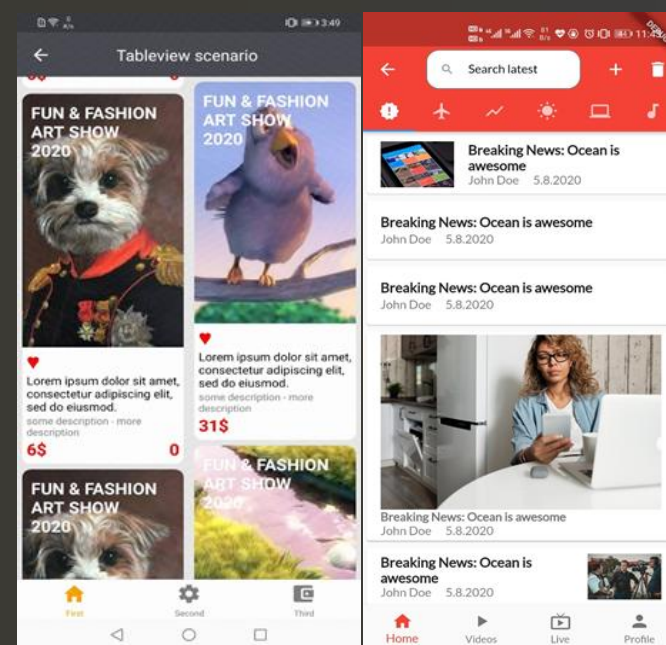
手表



智慧屏



IP Camera



手机

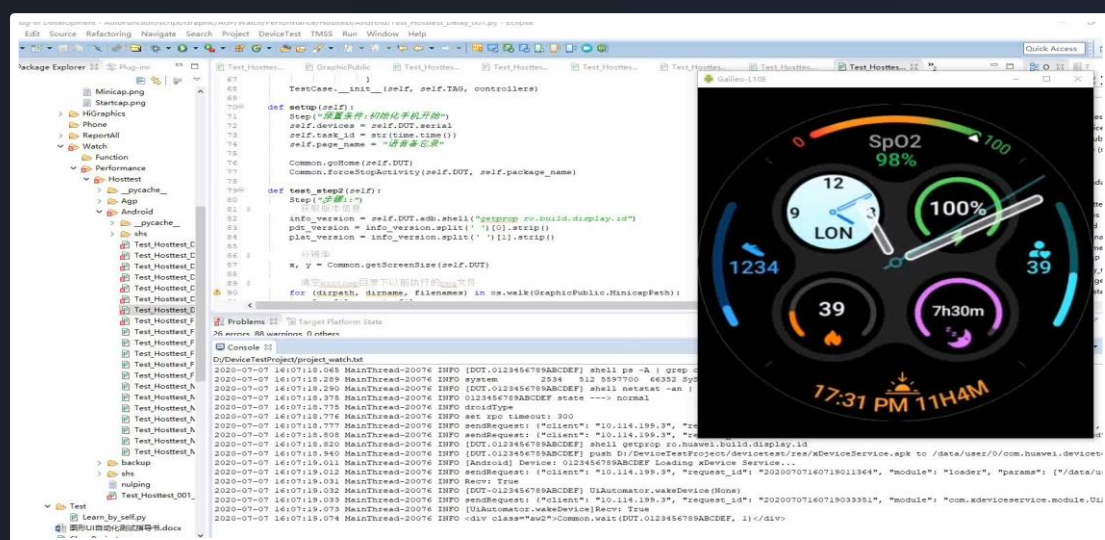


平板

负载模型+分析工具已经覆盖的设备，且L2设备已经开源

工具：支持“三个”自动化

终端设备



负载模型自动化测试

① 自动控制

界面实时展示测试数据

自动化测试

SP工具

被测负载应用

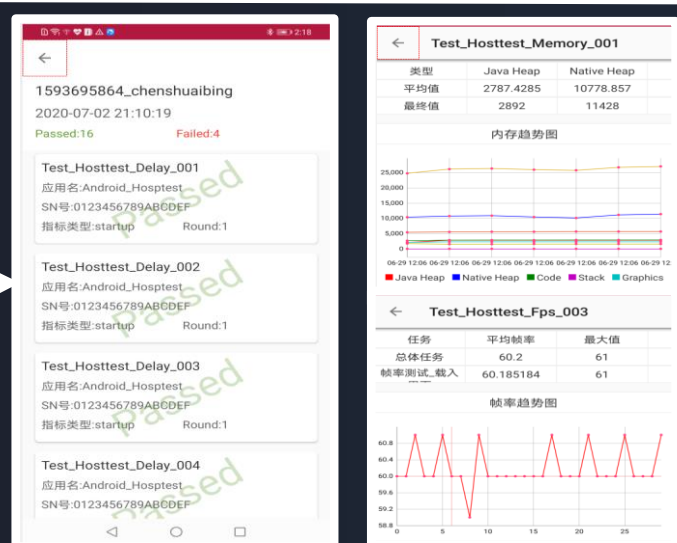
测试数据

② 自动采集

SP工具展示测试结果

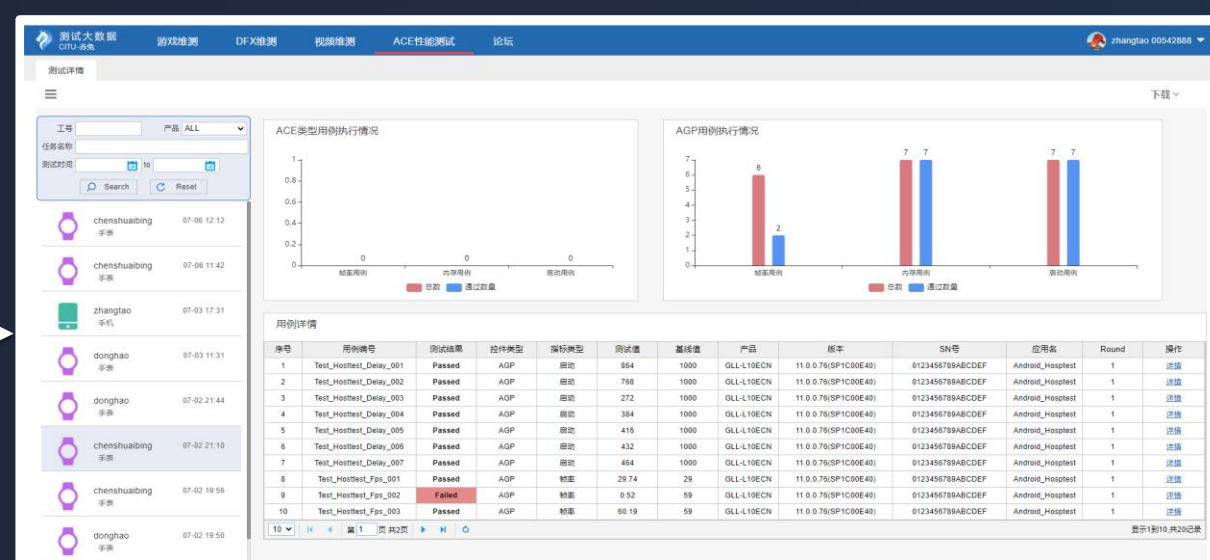


悬浮窗查看实时数据



设备端统计与分析

网站端



数据网页展示

数据对比分析

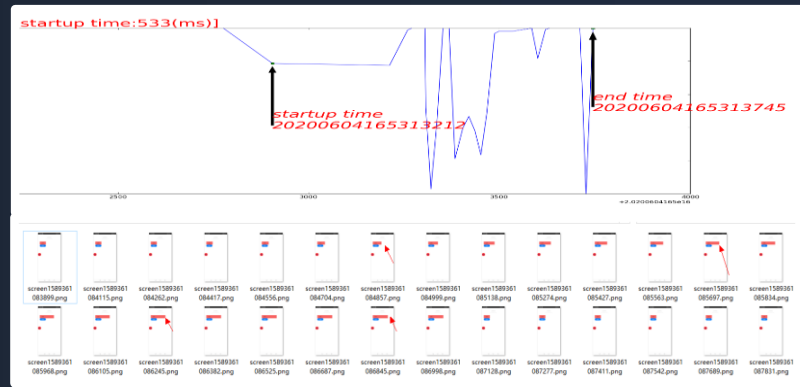
网站详情展示

③ 自动分析

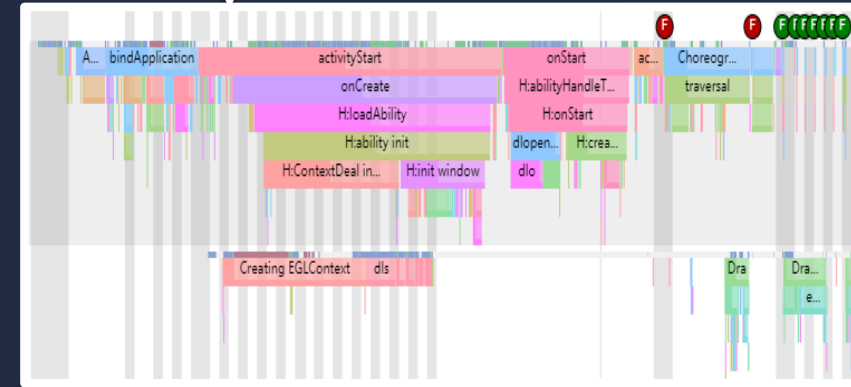
Server

数据分析

http



应用启动分析



函数耗时分析

自动化能力：

✓ 自动控制

- AccessibilityService 对接
- 支持滑动、点击

✓ 自动采集

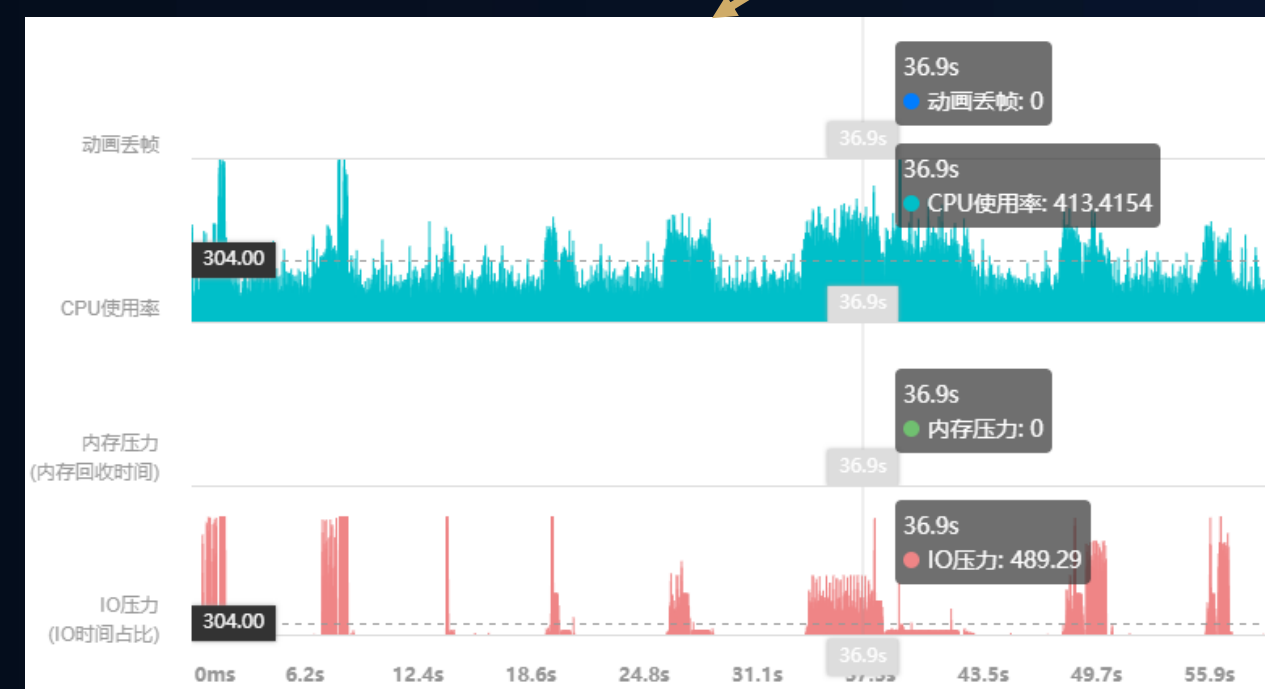
- 支持帧率、内存、功耗、时延、带宽、频点等，以及细粒度采集能力

✓ 自动分析

- 支持线程耗时分析
- 支持内存拆解分析

系统分析和优化建议

| 场景 | 指标类型 | 指标基线 | 数据 |
|------------|------|--------|-------|
| 多个应用连续页面切换 | 时延 | 1000ms | 500ms |
| | 丢帧 | 0.5% | 3% |



| 进程 | 线程id | 线程 | CPU使用率 | Cycles |
|-----------------------------|-------|-----------------|--------|------------|
| /system/bin/traced_probes | 902 | traced_probes | 20.15% | 1463220074 |
| com.ss.android.article.news | 10999 | id.article.news | 60.59% | 1580573167 |
| /system/bin/surfaceflinger | 757 | surfaceflinger | 12.32% | 1035917319 |
| com.ss.android.ugc.aweme | 5648 | RenderThread | 10.30% | 978452332 |

| 进程 | 读 (M/s) | 写 (M/s) | 操作 |
|-----------------|-----------|-----------|----|
| gecko-update-th | 780.35 | 1013.70 | 详情 |
| kworker/u16:1 | 7.63 | 1668.42 | 详情 |
| kworker/u16:3 | 0 | 1635.0098 | 详情 |
| drv_nv | 5.3418803 | 1549.5876 | 详情 |

在xx时刻，CPU负载达到85%，其中负载较高的线程如下所示，建议优化占用CPU较高的线程

IO压力较大，其中读写IO较多的线程是如上所示，建议优化相关读写

应用分析和优化建议

| | 场景 | 指标类型 | 指标基线 | 数据 | 问题分析 |
|------|------|------|-------|--------|------|
| 图库应用 | 应用启动 | 时延 | 550ms | 500ms | |
| | 大图展开 | 时延 | 200ms | 150ms | 点击弹框 |
| | | 丢帧率 | 1% | 15% | 点击弹框 |
| | 大图退出 | 丢帧率 | 0.5% | 0.4% | 点击弹框 |
| | 图片删除 | 时延 | 100ms | 1048ms | 点击弹框 |
| | 图片预览 | 内存 | 200M | 150M | |

| 线程 | UI | | | | | | |
|------|---------------------------|--------------------|------------------------|--------------------|---------------------|-----------------------|------------------------------|
| | H:ExecuteJS | | OnVsyncEvent | | | | |
| 函数 | MarkNeedUpdate 标记待刷新组件 | FlushBuild 组件创建 | FlushAnimation 动画刷新 | FlushBuild 组件创建 | FlushLayout 组件布局 | FlushRender 组件绘制指令 | FlushRenderFinish 指令提交后处理 |
| 调用次数 | 5000 | | | | | | |
| 耗时占比 | | 80% | | | 70% | | |
| 优化建议 | | 点击弹框 | | | 点击弹框 | | |

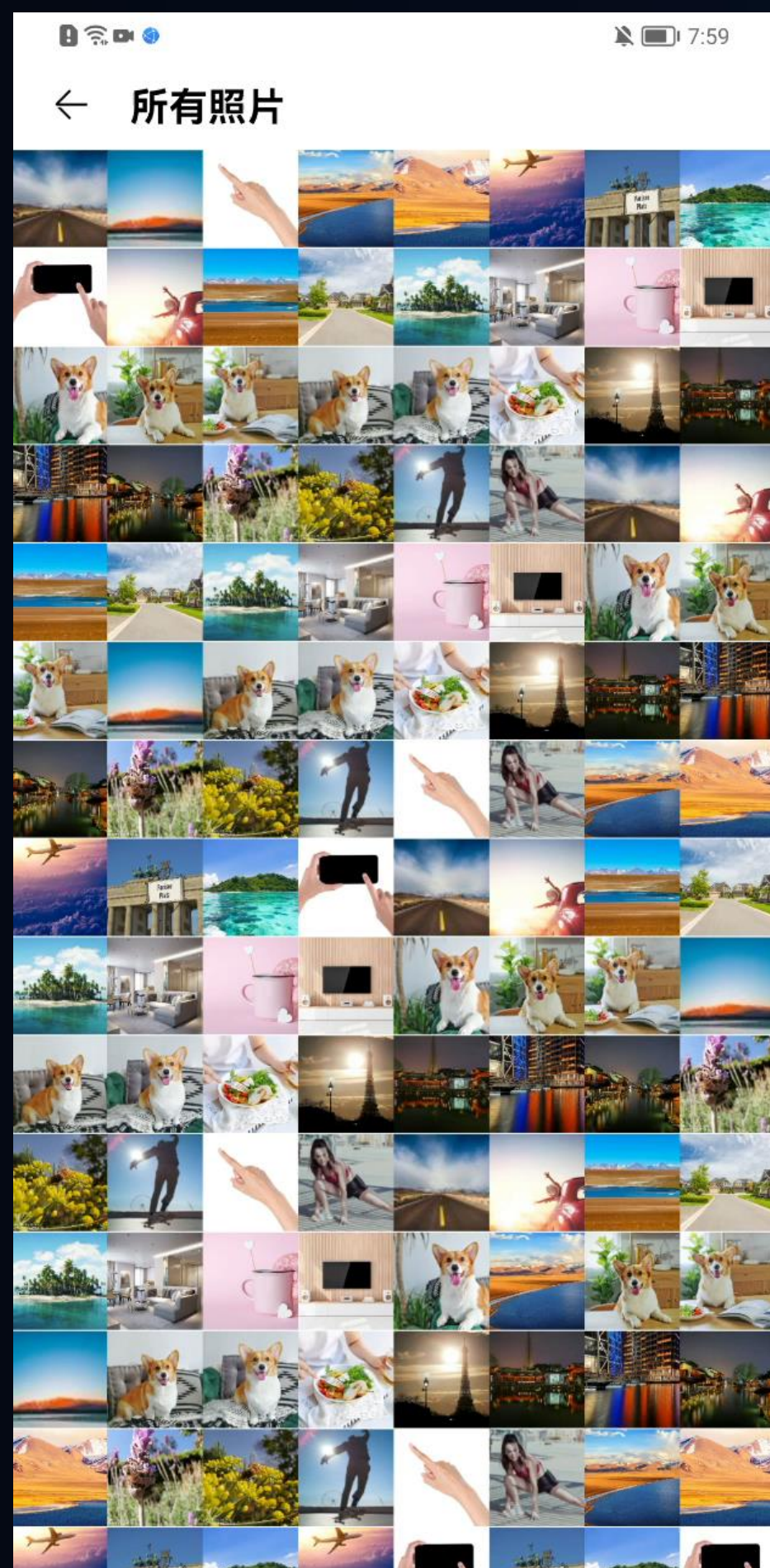
FlushBuild函数过程耗时过长，原因可能如下：
1、由于加载控件节点过多导致（MarkNeedUpdate为5000个），建议在容器组件中使用懒加载优化策略
2、通过show来进行显隐转换，创建过多dom节点，建议使用if方式进行显隐转换

FlushLayout函数过程耗时过长，原因可能如下：
1、控件嵌套层数过深
2、多层flex布局
建议优化布局策略

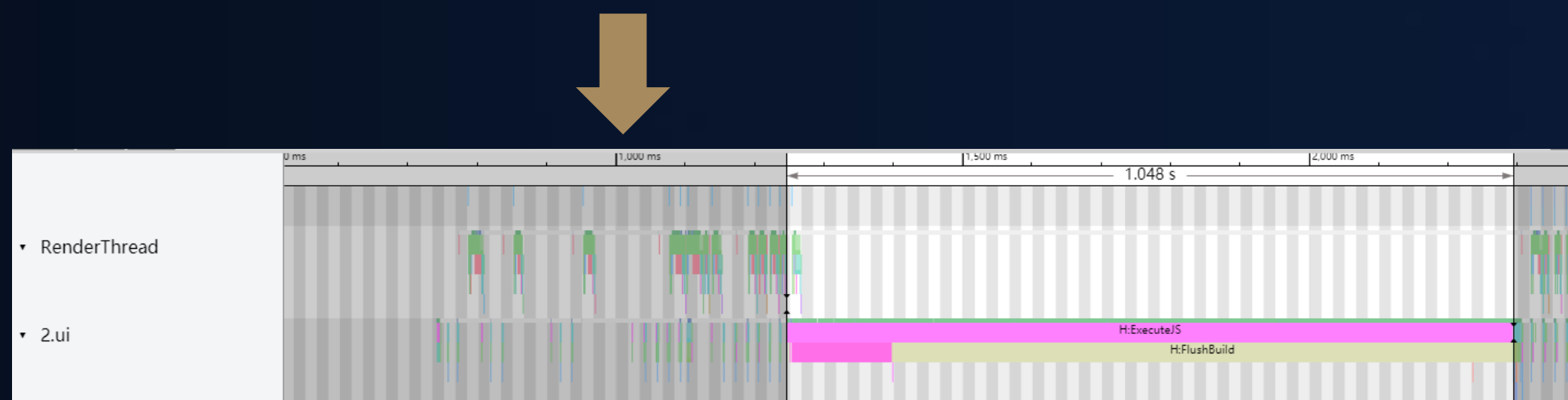


实际案例（图库）— 开发者代码没有使用懒加载

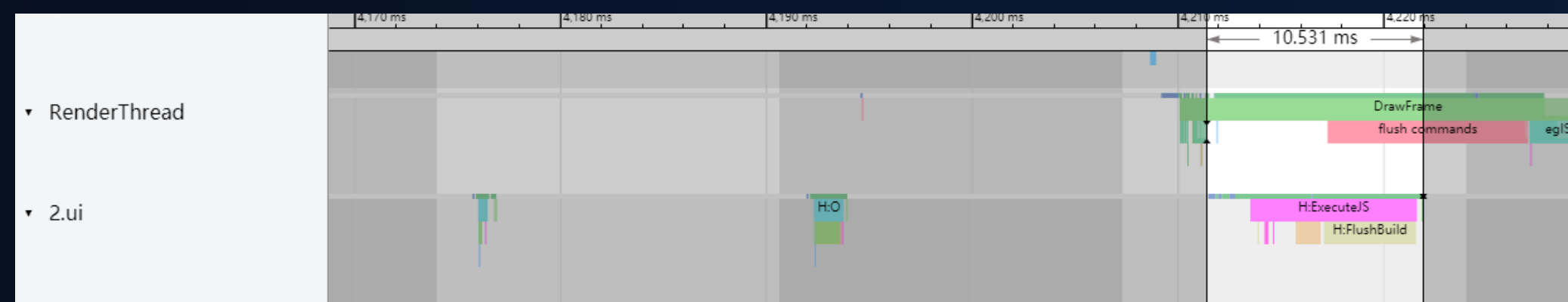
场景1：删除图片



| 应用 | 场景 | 指标类型 | 指标基线 | 数据 |
|------|------|------|-------|--------|
| 图库应用 | 图片删除 | 时延 | 100ms | 1048ms |



常规写法（非懒加载）：删除时，5000个image组件同时触发，导致FlushBuild () 函数耗时约800ms



优化写法（懒加载）：删除时，少量image组件（100-200）触发刷新，FlushBuild () 耗时仅为8ms，明显缩短

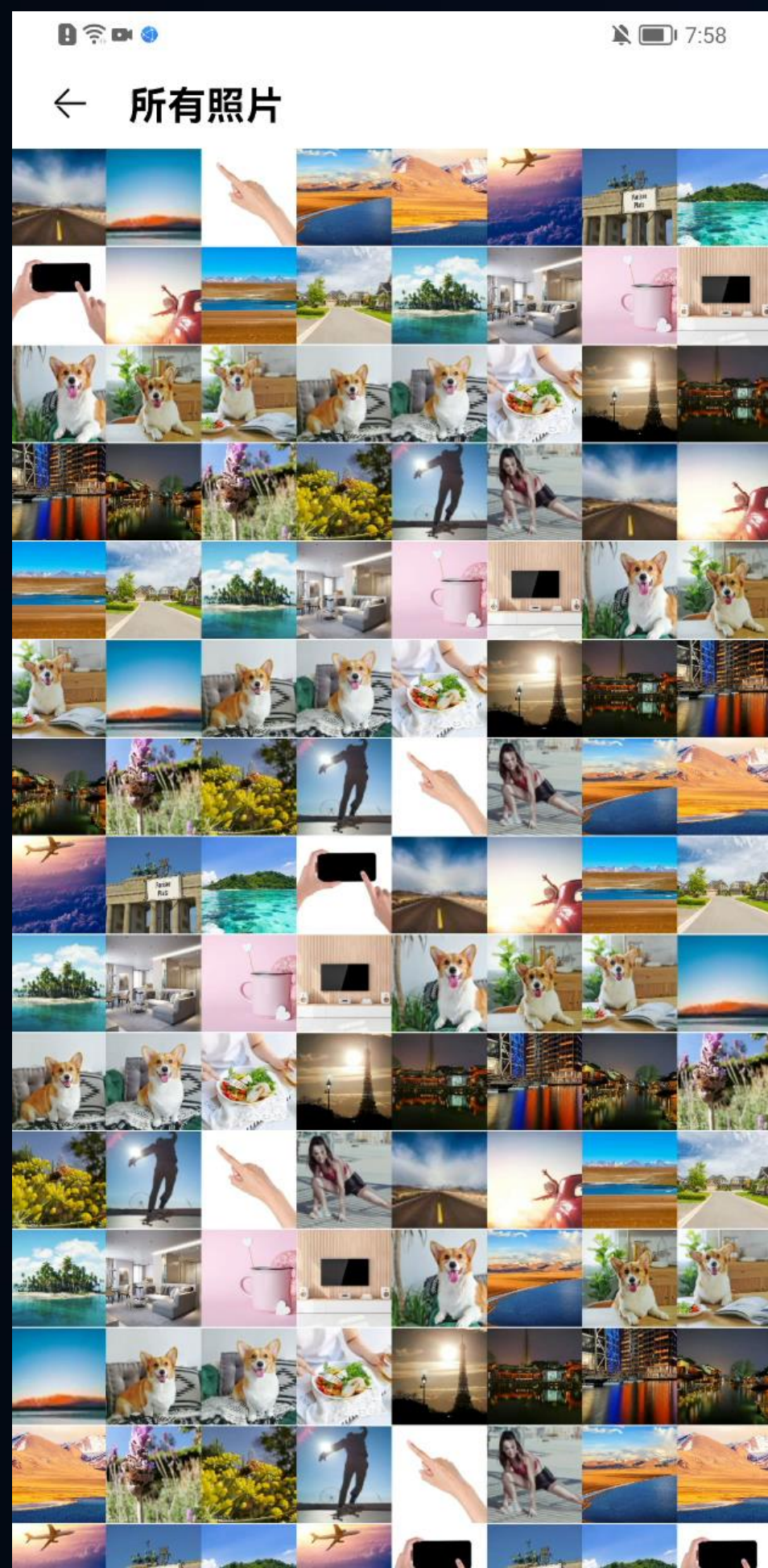
```

LazyForEach(this.data, (item: string) => {
  ListItem() {
    Row() {

```

实际案例（图库）—系统级优化（UI和渲染分离）

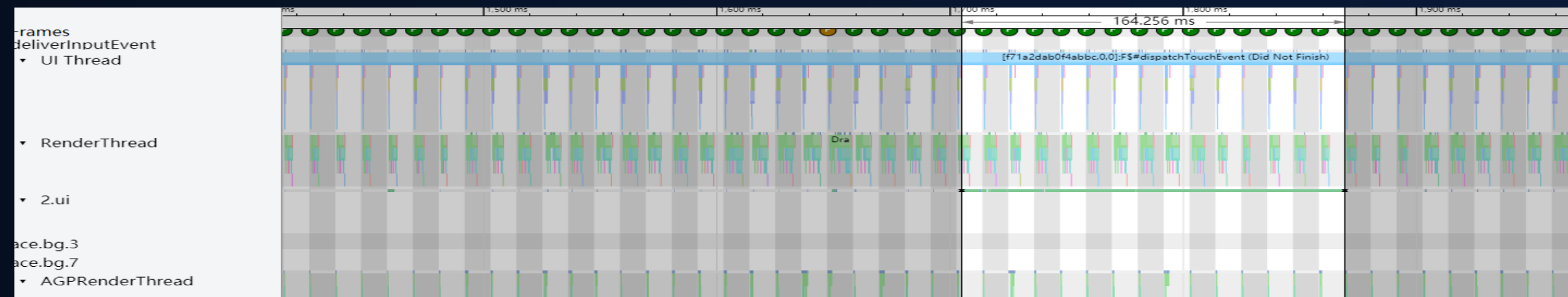
场景2：打开一张图片



| 应用 | 场景 | 指标类型 | 指标基线 | 数据 |
|------|------|------|------|-----|
| 图库应用 | 大图展开 | 丢帧率 | 1% | 15% |



优化前：打开大图过程中，出现一个新的复杂UI页面，UI布局耗时100+ms，UI Thread 阻塞渲染



优化后（UI与动效分离）：UI和动效（AGPRenderThread）分别在不同线程进行，UI不影响动效过程的连续性，UI Thread不阻塞渲染

1

HarmonyOS 自研图形栈整体架构和测试能力

2

华为 ArkUI 开发框架和测试技术

- 实际案例—图库分析调优（现场演示）

3

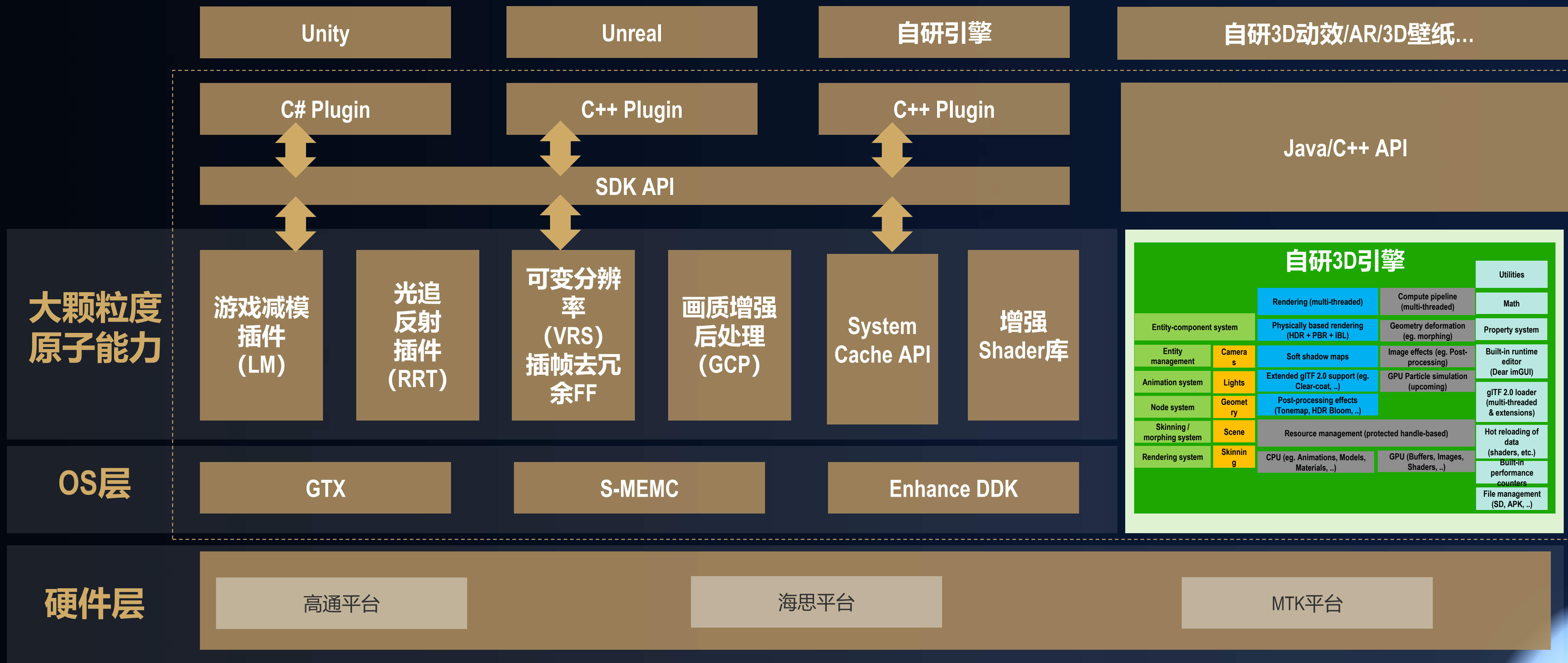
华为自研 3D 图形栈和测试技术

- 实际案例—壁纸制作分析调优（现场演示）

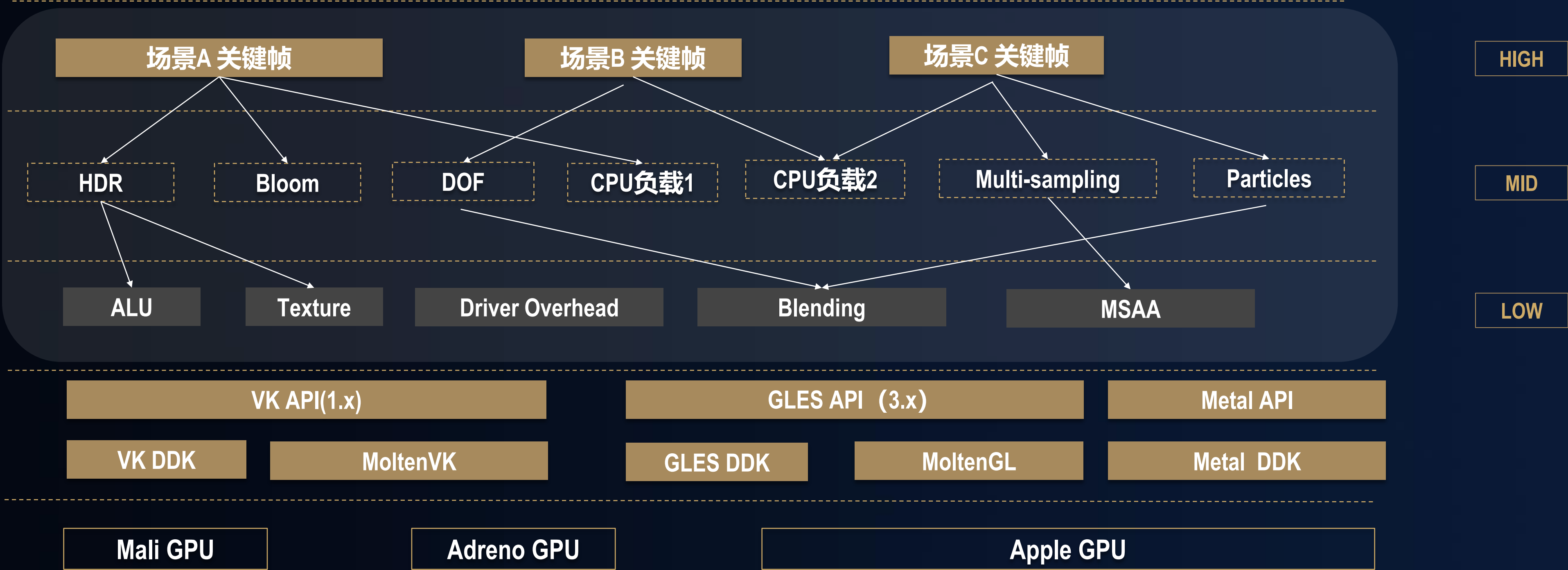
4

Devco Testing 图形测试能力介绍

华为3D图形栈

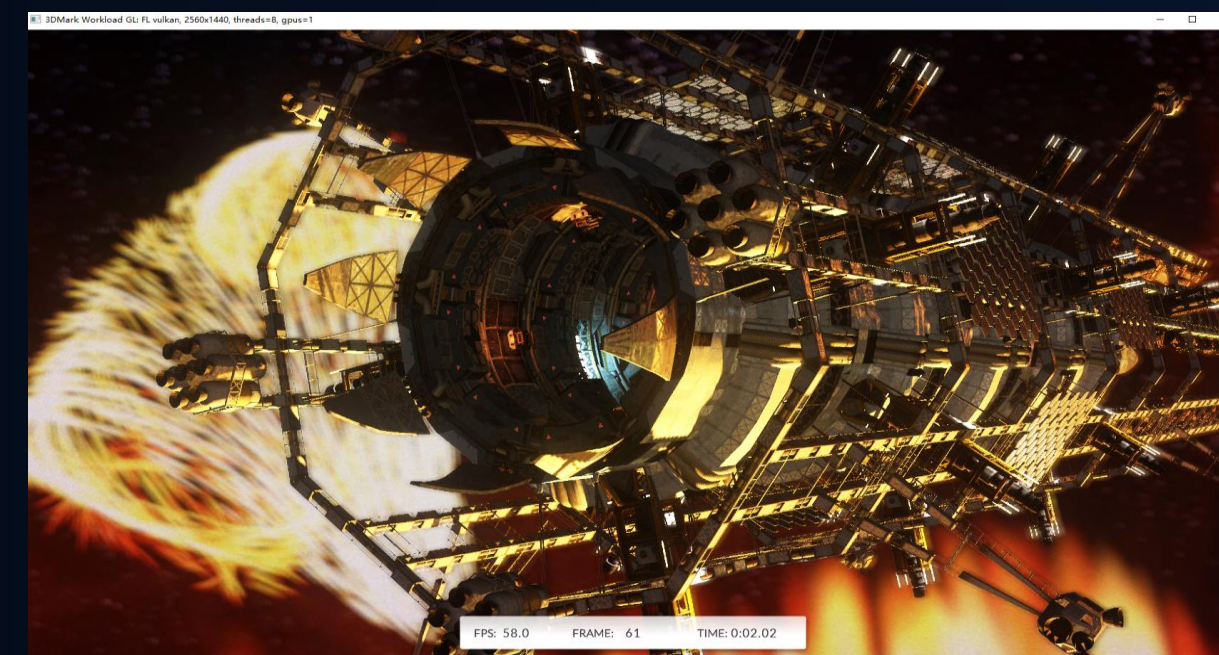


3D应用分析基础

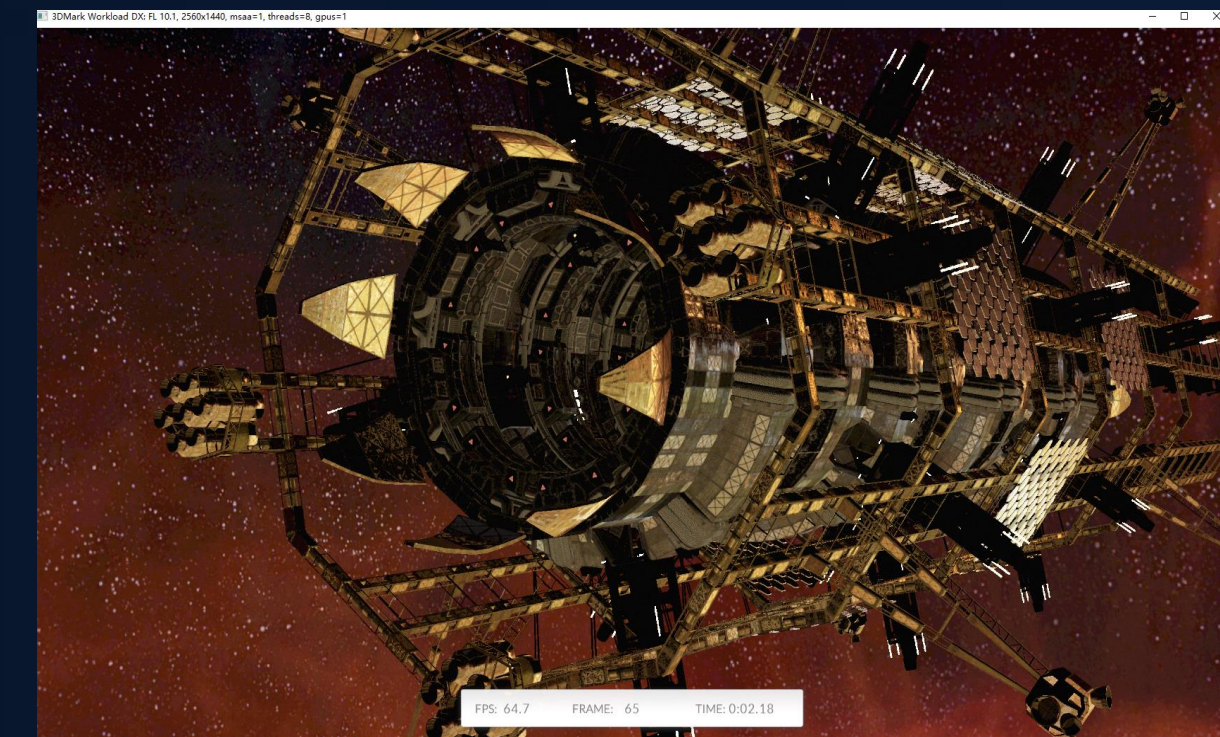


特性拆分和分析方法

| 序号 | 特效 |
|----|-----------------|
| 1 | Particle |
| 2 | Shadow map |
| 3 | Point light |
| 4 | Bloom |
| 5 | DOF |
| 6 | Cull |
| 7 | Volume Illum... |



特性关闭



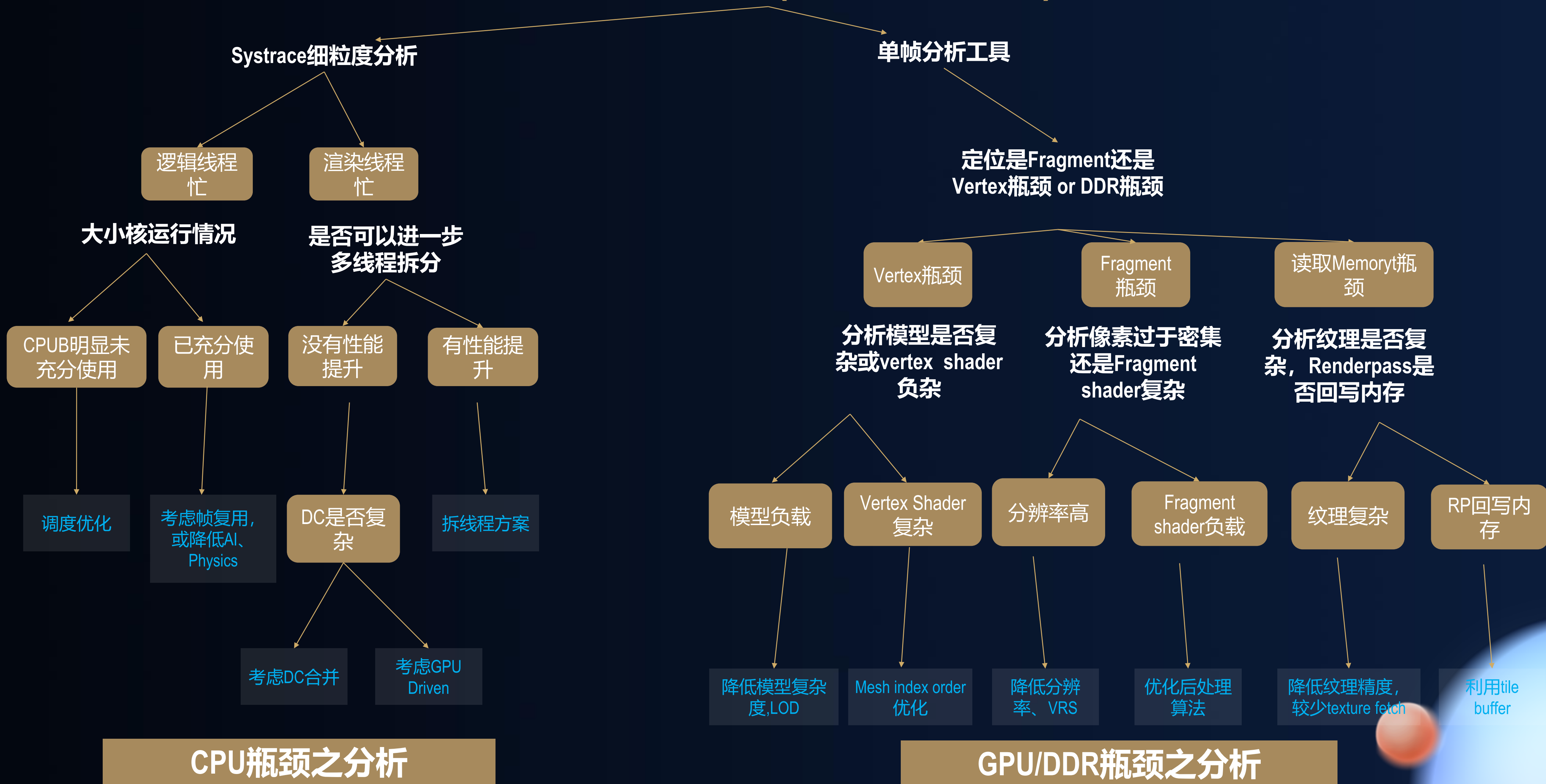
| | 场景A | 场景B | 场景C |
|--|---------|---------|---------|
| Average FPS | 35.61 | 35.85 | 57.92 |
| GPUCycles(Mcycles/s) | 599.94 | 396.37 | 430.01 |
| VertexComputeCycles(Mcycles/s) | 504.21 | 174.48 | 102.62 |
| FragmentCycles(Mcycles/s) | 591.82 | 284.40 | 414.01 |
| TilerCycles(Mcycles/s) | 461.54 | 141.39 | 84.04 |
| CacheReadLookups(Mlookups/s) | 889.59 | 395.48 | 193.21 |
| CacheWriteLookups(Mlookups/s) | 98.67 | 73.99 | 24.49 |
| ExternalMemoryReadAccesses(MAccesses/s) | 123.98 | 73.37 | 26.02 |
| ExternalMemoryWriteAccesses(MAccesses/s) | 59.18 | 48.43 | 12.88 |
| ExternalMemoryReadStalls(MStalls/s) | 108.67 | 132.64 | 40.26 |
| ExternalMemoryWriteStalls(MStalls/s) | 111.52 | 97.08 | 38.41 |
| ExternalMemoryReadBytes(MB/s) | 7934.38 | 4695.63 | 1665.01 |
| ExternalMemoryWriteBytes(MB/s) | 3686.89 | 3056.19 | 807.58 |

启发:

对单帧进行回放, 进行 Drawcall/Renderpass裁剪, 结合 GPU hw counters进行问题定位!

3D渲染领域瓶颈

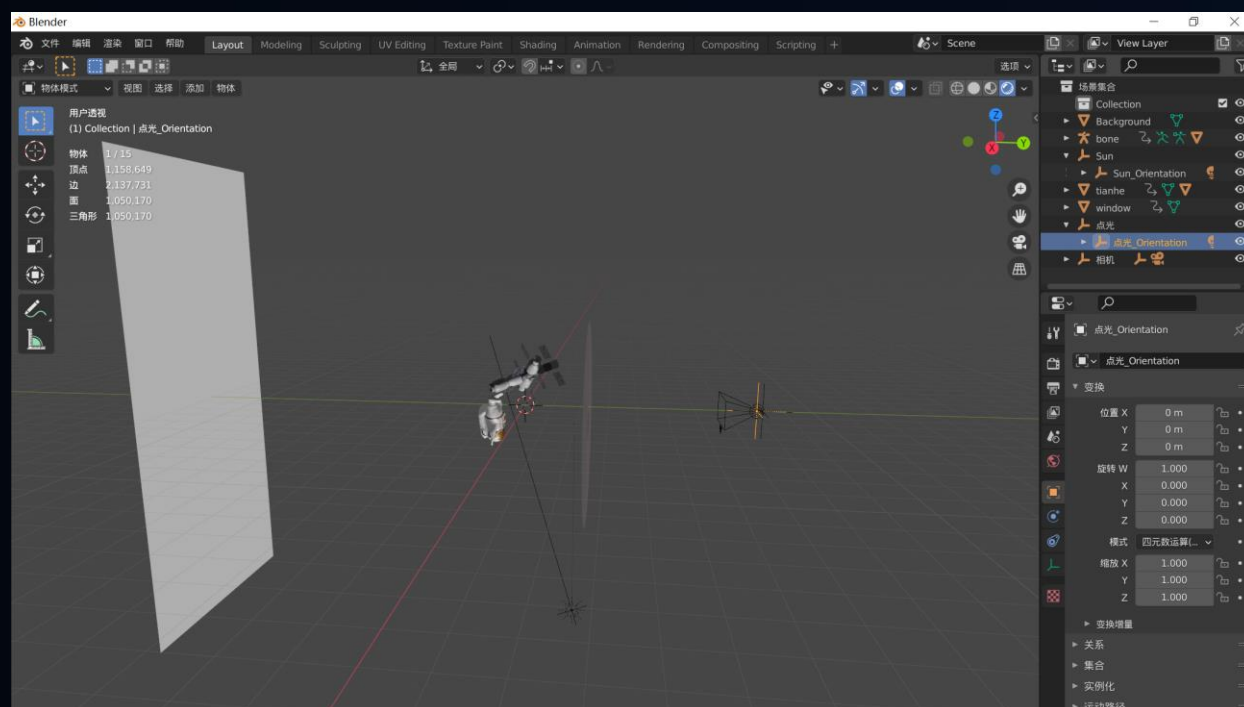
SP工具测试分解瓶颈 (CPU or GPU or DDR)



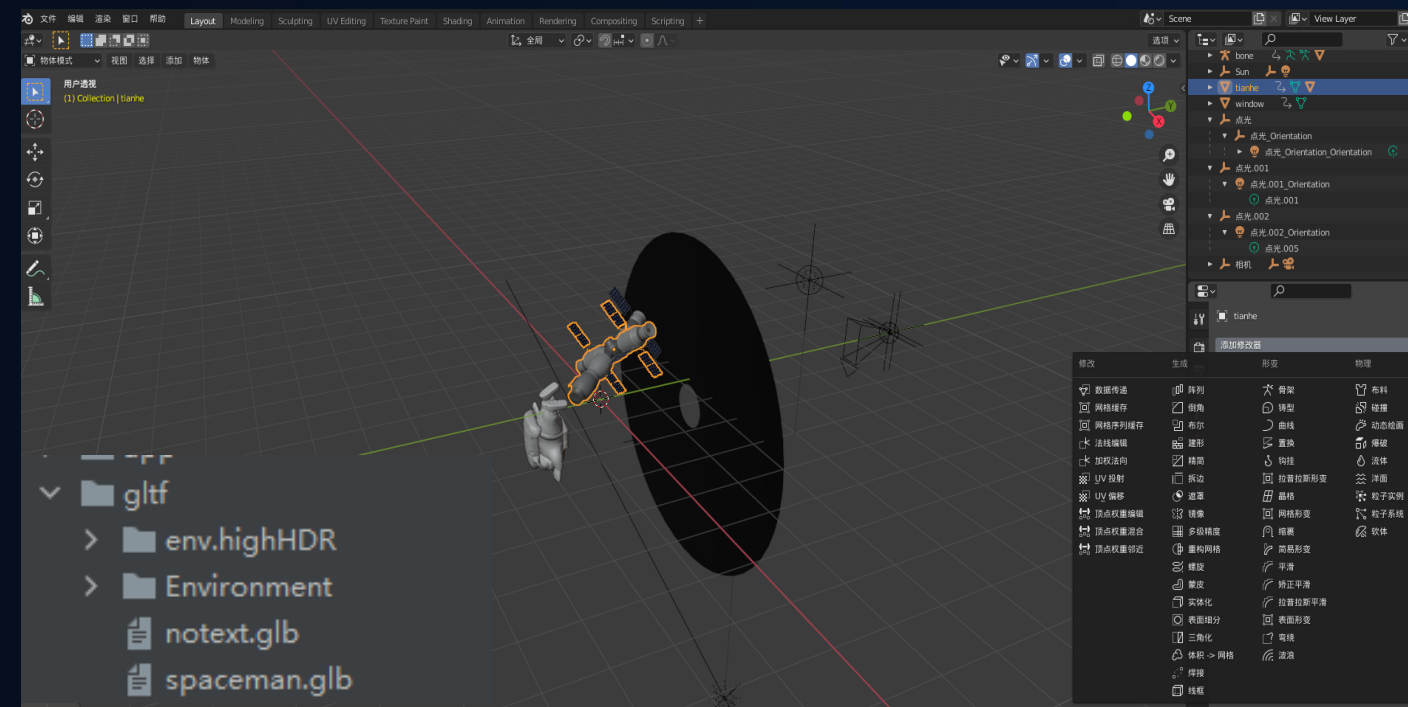
实际案例 (3D壁纸制作调优)

制作过程

Blender制作模型+PBR材质



表面细分+点光



反射面

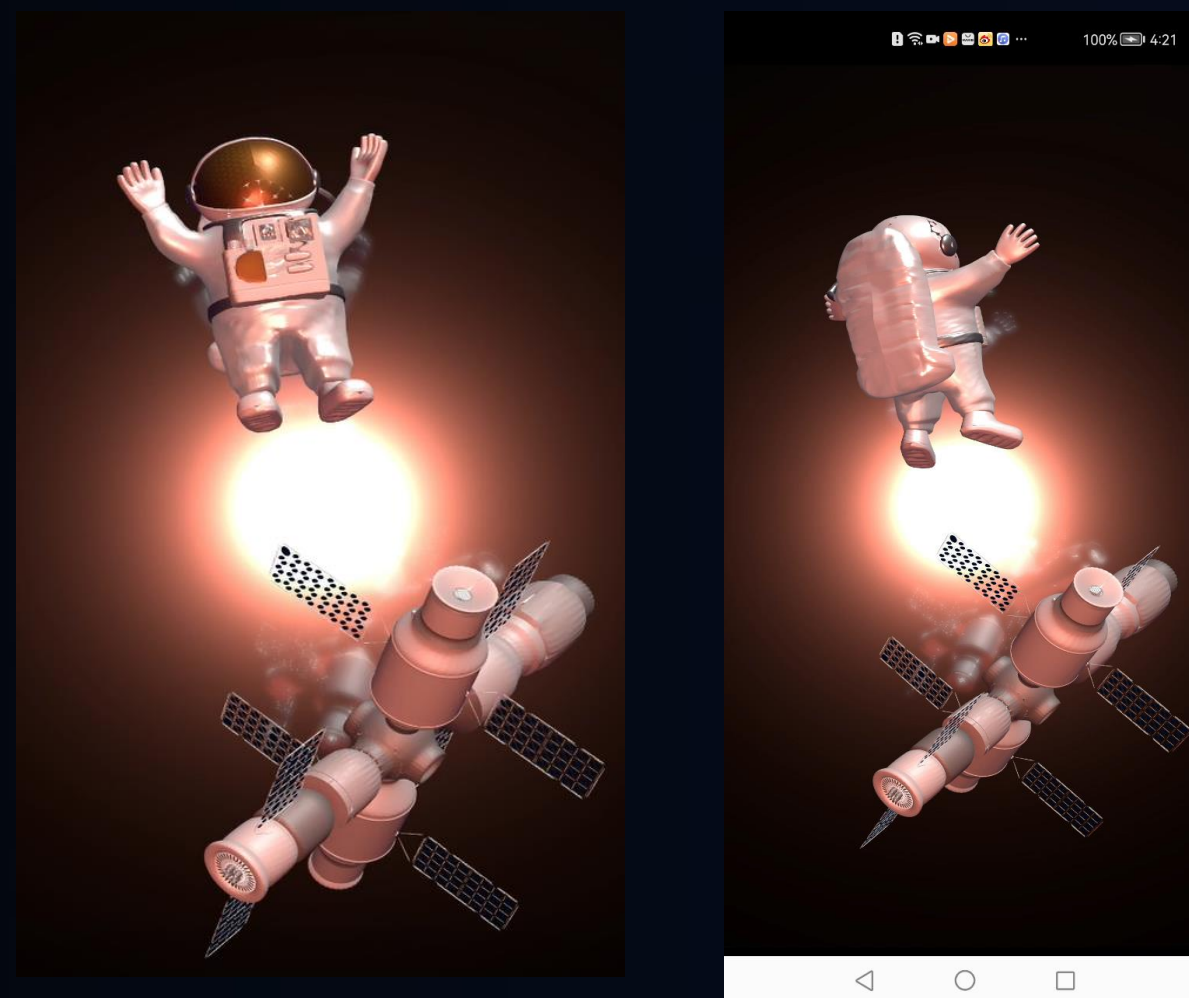
```
com.huawei.core.reflection 259  
AGPEngineAdapter 260  
BitmapHelper 261  
JavaSampleActivity 262
```

```
setNodeVisibility( nodeName: "Backgr  
createReflectionPlane();  
setReflectionPlaneVisible(false);
```

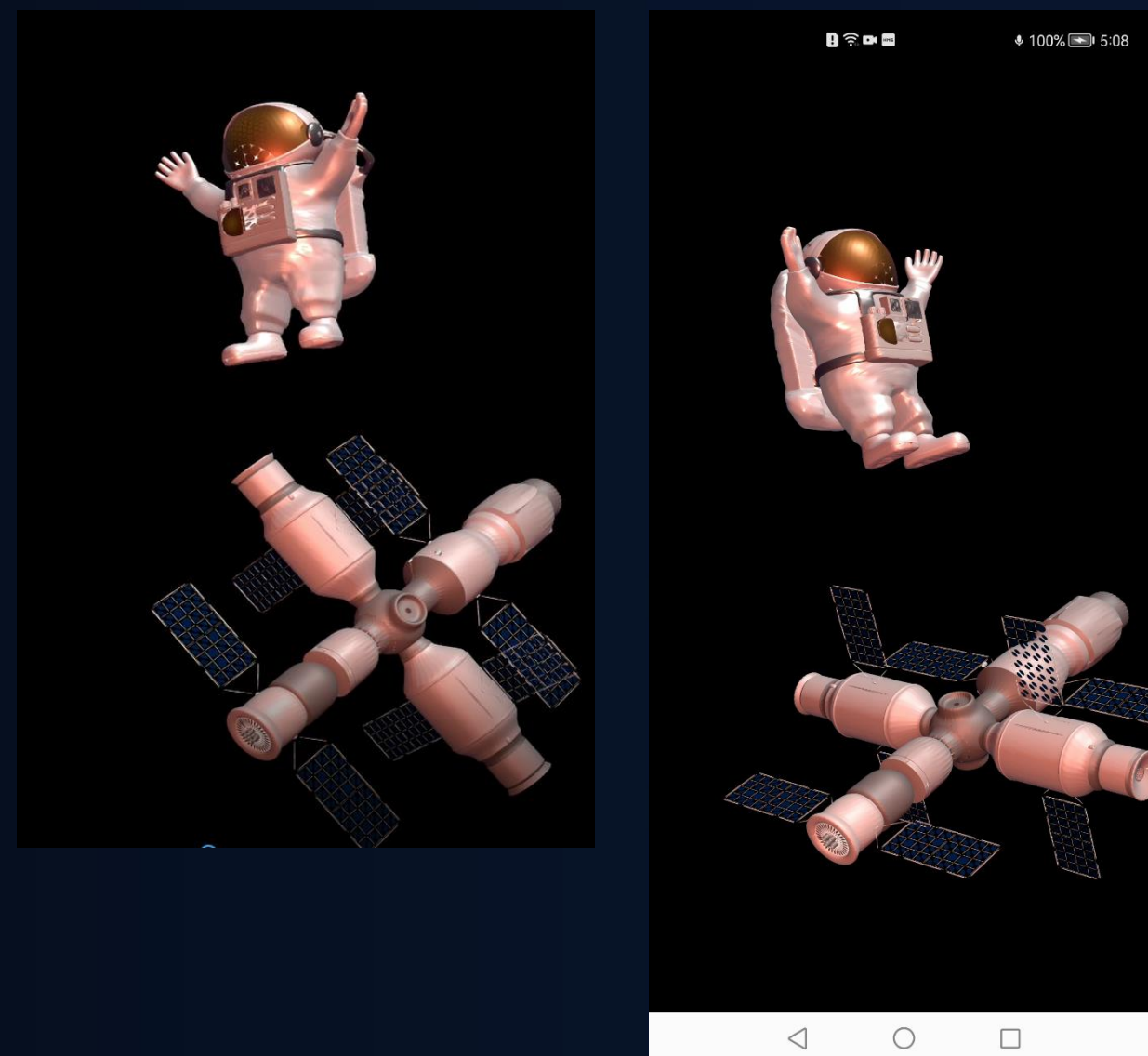
```
AGPEngineAdapter.java x GpuResourceHandle.class x SceneUtil.class x custom  
Decompiled .class file, bytecode version: 52.0 (Java 8)  
50  
57 SceneUtil.ReflectionPlane createReflectionPlane(String name,  
58  
59 void destroyReflectionPlane(SceneUtil.ReflectionPlane plane);
```

优化过程

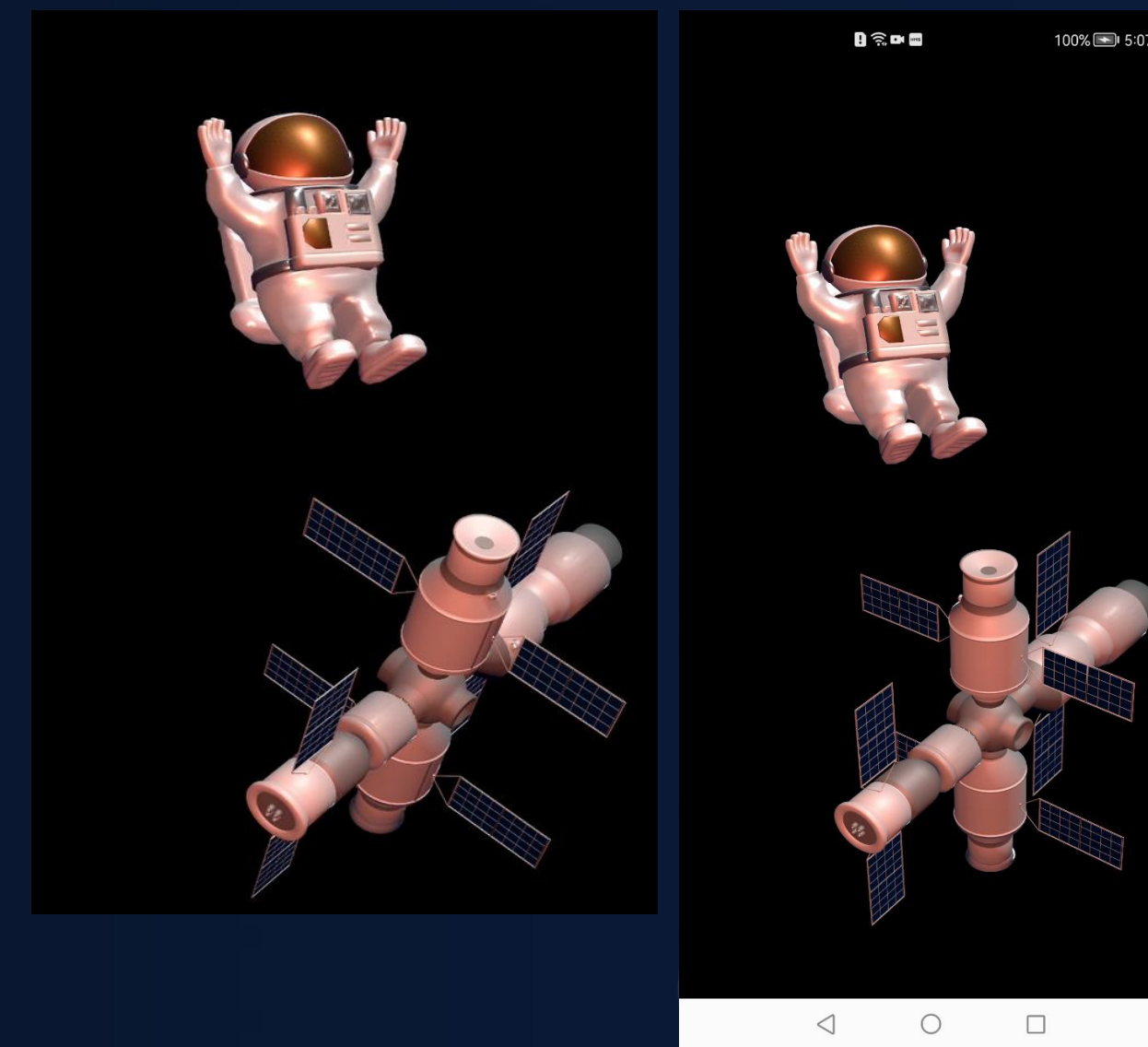
表面细分(顶点50w)+点光 (1束) +反射面



表面细分(顶点50w)+点光(1束)



表面细分(顶点5w)+点光(1束)



实际案例（性能、功耗结合GPU hw counters）

| 数据信息 | 壁纸名称 | 帧数 | 平均帧率/帧 | 低帧率 | 抖动率 | 每小时卡顿次数 | 最差丢帧 | 归一化电流 | shell最低温度 | shell最高温度 | 壳温温升 | 游戏时长(分钟) | CPU-C1(频率) | CPU-C2(频率) | CPU-C3(频率) | GPU(频率) | DDR(频率) |
|------------------------|-------------|----|--------|--------|-------|---------|------|---------|-----------|-----------|--------|----------|------------|------------|------------|---------|---------|
| 模型简化 | 3D_spaceman | 60 | 43.89 | 77.05% | 0.00% | 0 | 18 | 1921.33 | 27.367 | 37.56 | 10.193 | 1.02 | 1214.38 | 921.557 | 1434.74 | 743.721 | 2746.0 |
| 表面细分(顶点50w)+点光(1束)+反射面 | 3D_spaceman | 60 | 42.24 | 88.89% | 1.75% | 114 | 20 | 1303.99 | 27.365 | 33.798 | 6.433 | 1.05 | 999.587 | 925.381 | 1402.0 | 481.73 | 2524.81 |
| 表面细分(顶点5w)+点光(1束) | 3D_spaceman | 60 | 60.11 | 0.00% | 0.00% | 0 | 2 | 571.21 | 33.427 | 34.544 | 1.117 | 1.05 | 1145.49 | 928.429 | 1402.0 | 276.236 | 686.476 |

| HW Counters | Average FPS | GpuCycles | VertexComputeCycles | FragmentCycles | ExternalMemoryReadBytes | ExternalMemoryWriteBytes | ShaderCycles | ShaderTextureCycles | ShaderArithmeticCycles | ShaderLoadStoreCycles | Instructions |
|------------------------|-------------|-----------|---------------------|----------------|-------------------------|--------------------------|--------------|---------------------|------------------------|-----------------------|--------------|
| 模型简化 | Type | (Fps) | (Mcycles) | (Mcycles) | (Mcycles) | (MB/s) | (MB/s) | (MB/s) | (MB/s) | (MB/s) | (MB/s) |
| 表面细分(顶点50w)+点光(1束)+反射面 | 43.89 | 797 | 787 | 574 | 14400 | 8980 | 1910 | 69.3 | 676 | 253 | |
| 表面细分(顶点50w)+点光(1束) | 42.24 | 717 | 459 | 511 | 10500 | 6890 | 1190 | 49.6 | 480 | 171 | 2712076 |
| 表面细分(顶点5w)+点光(1束) | 60.11 | 297 | 93.2 | 290 | 821 | 712 | 493 | 48.9 | 335 | 61.7 | 2639863 |

结合性能、功耗、热数据以及GPU counters方便定位问题根因

1

HarmonyOS自研图形栈整体架构和测试能力

2

华为ArkUI开发框架和测试技术

- 实际案例—图库分析调优（现场演示）

3

华为自研3D图形栈和测试技术

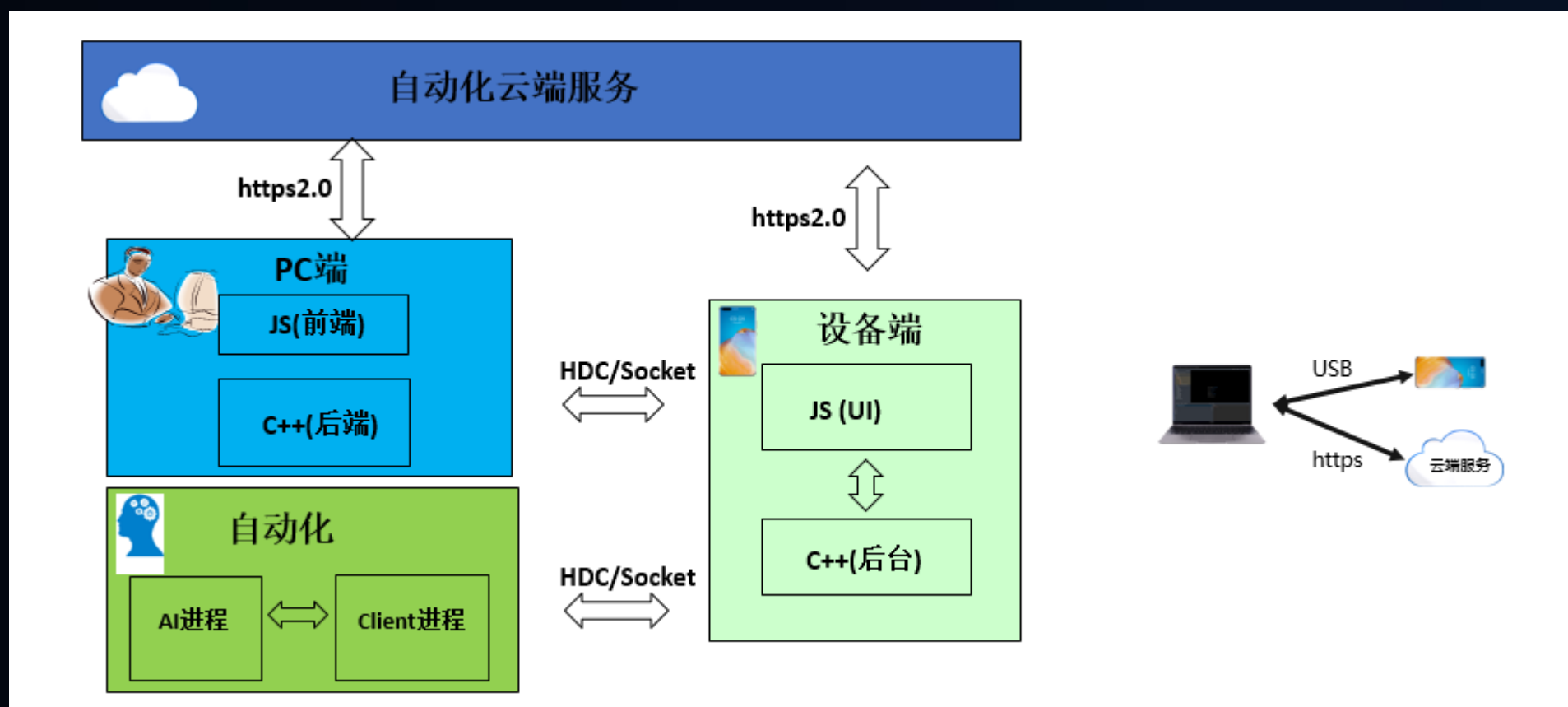
- 实际案例—壁纸制作分析调优（现场演示）

4

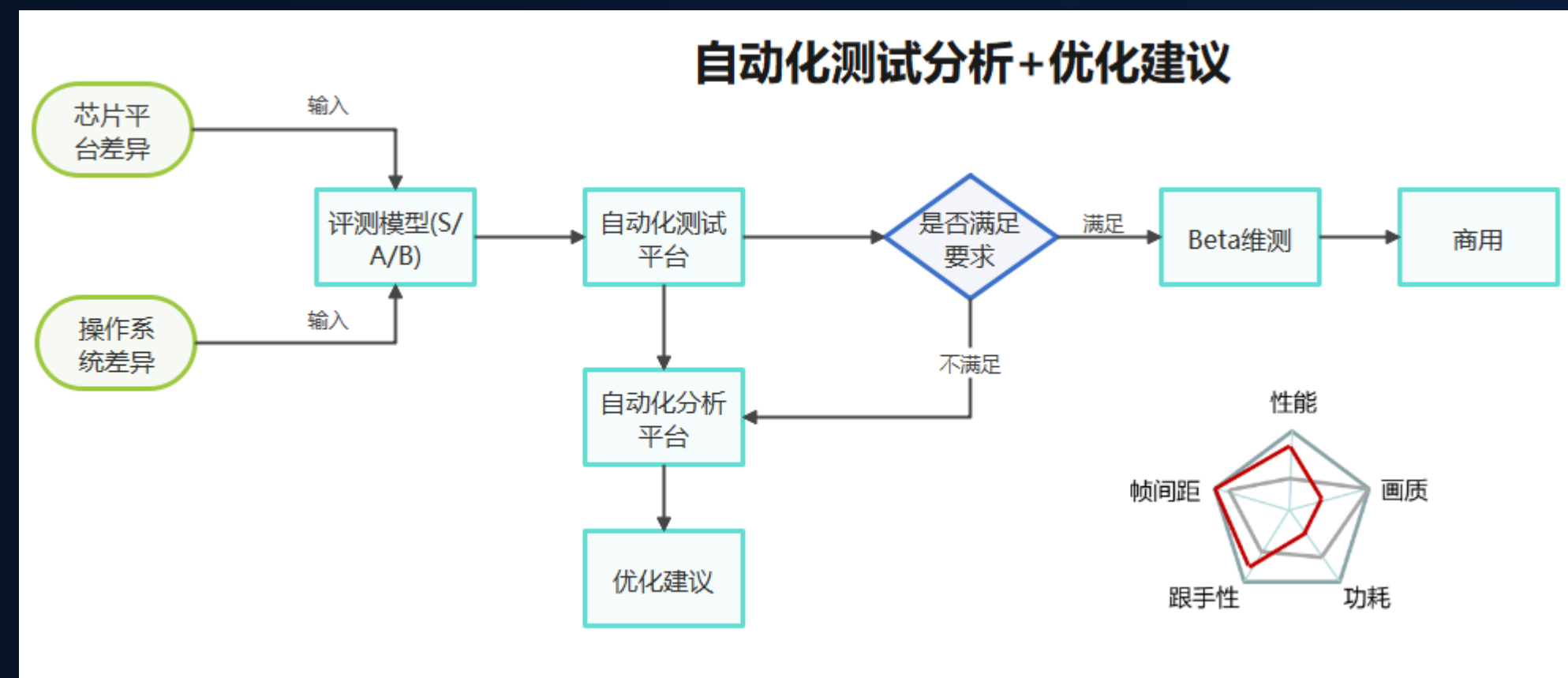
Deveco Testing图形测试能力介绍

Deveco Testing-图形栈测试分析能力一览

图形栈测试服务三端+自动化



自动化“测试+分析+优化建议”



性能采集和分析

- 支持底层CPU/GPU/DDR的频点和负载
- 支持平均帧率、抖动率和低帧率等指标的采集
- 支持GPU counters、CPU systrace细粒度分析、调度分析

功耗和热采集和分析

- 支持整机功耗采集和展示
- 支持功耗拆解采集和展示 (包括SOC、外设)
- 支持SOC温度采集
- 支持壳温采集
- 支持电池温度采集

自动化测试

- UI标注工具
- 地图定制工具
- 3D场景识别

大数据统计和分析

- 网站端数据统计和分析
- 手机端数据统计和实时展示
- 数据对比和Beta报告
- 用户画像

增强性服务

- 单帧采集和回放能力
- 画质检测能力
- 多路测试能力 (分布式)
- 关键耗时函数分析能力
- 方案对接
- ...

“三端”+自动化架构，支持基本性能功耗采集+增强型服务

特性1：灵活的独立APK测试

选择测试应用



实时数据展示



报告



特性2：分布式渲染多路测试

场景渲染增强



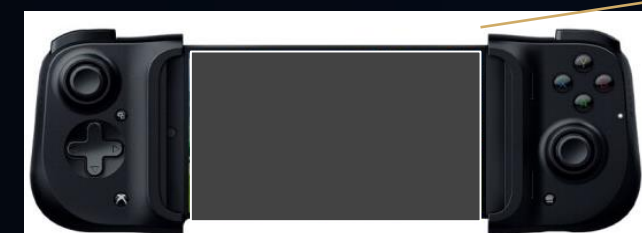
分布式渲染

2-B

华为智慧屏
(展示画面)

场景数据/指令RPC

2-A



分布式渲染

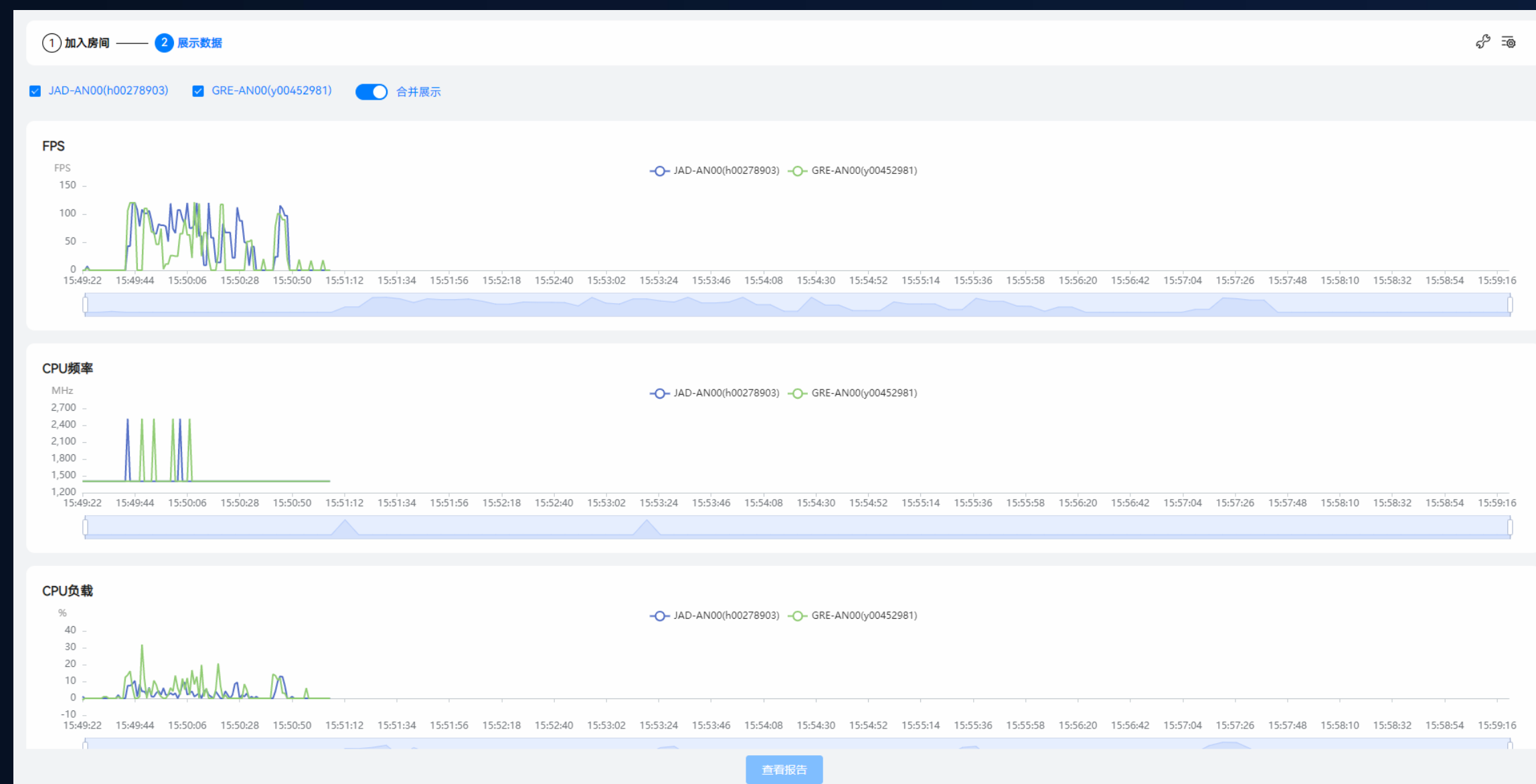


本地渲染

用户输入处理及游戏状态更新

1

分布式渲染和本地渲染



Devecu Testing工具

数据展示

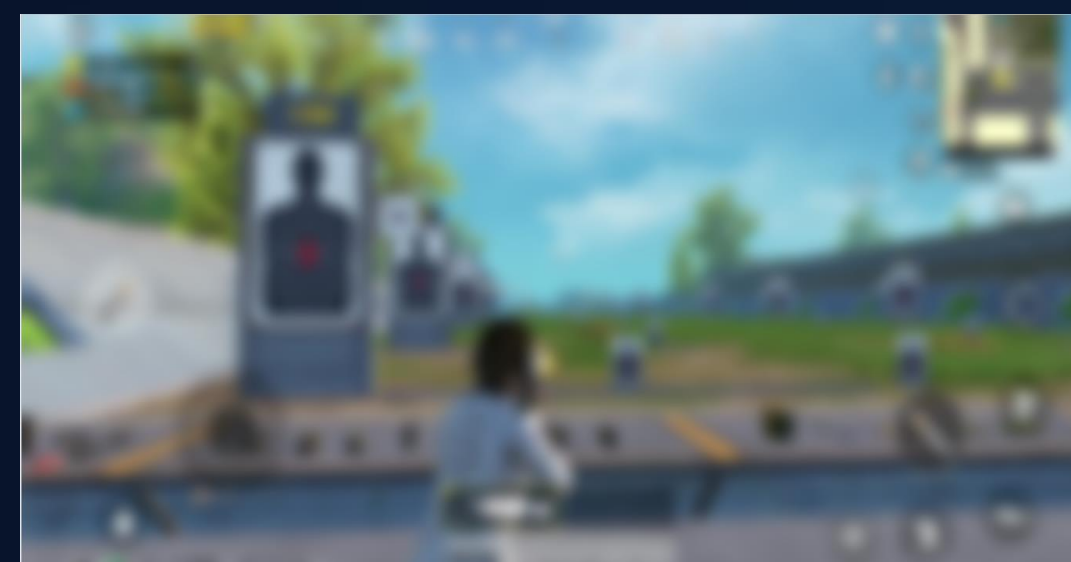
特性3：细粒度分析之systrace解析

Systrace抓取：

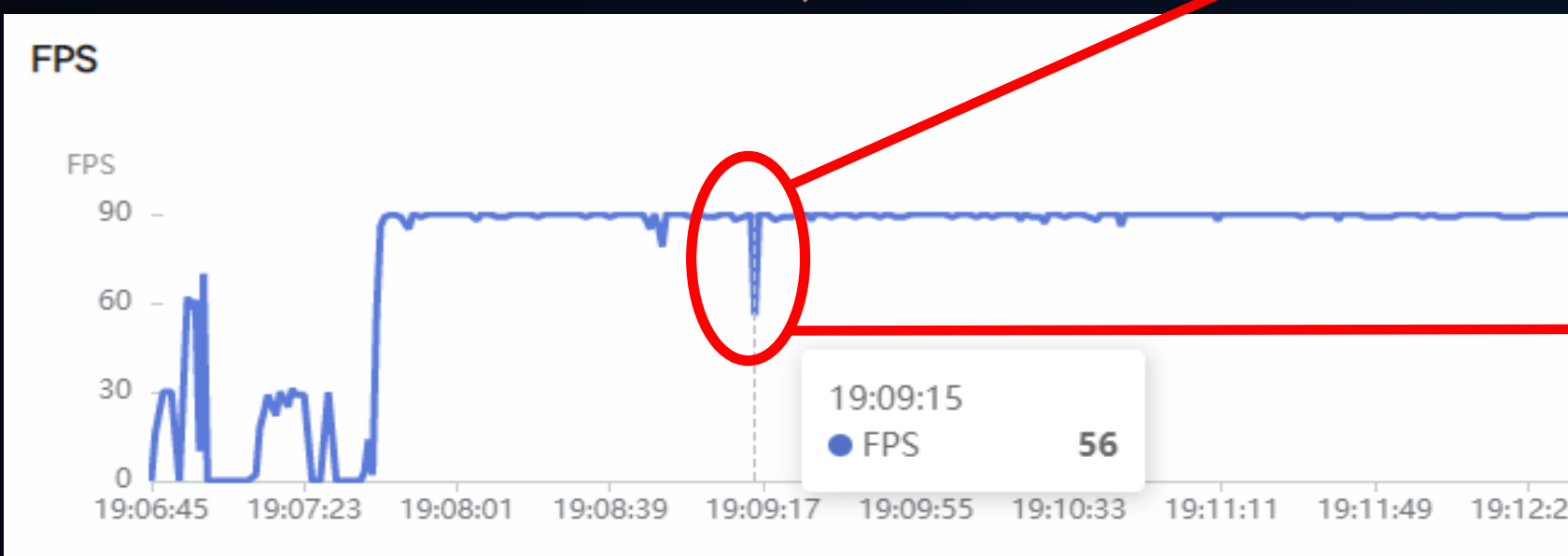
- 1、SP手动抓取，任意场景下随时触发采集，自动分析；
- 2、SP自动抓取，定制卡顿阈值，自动抓取卡顿时刻trace并自动分析。



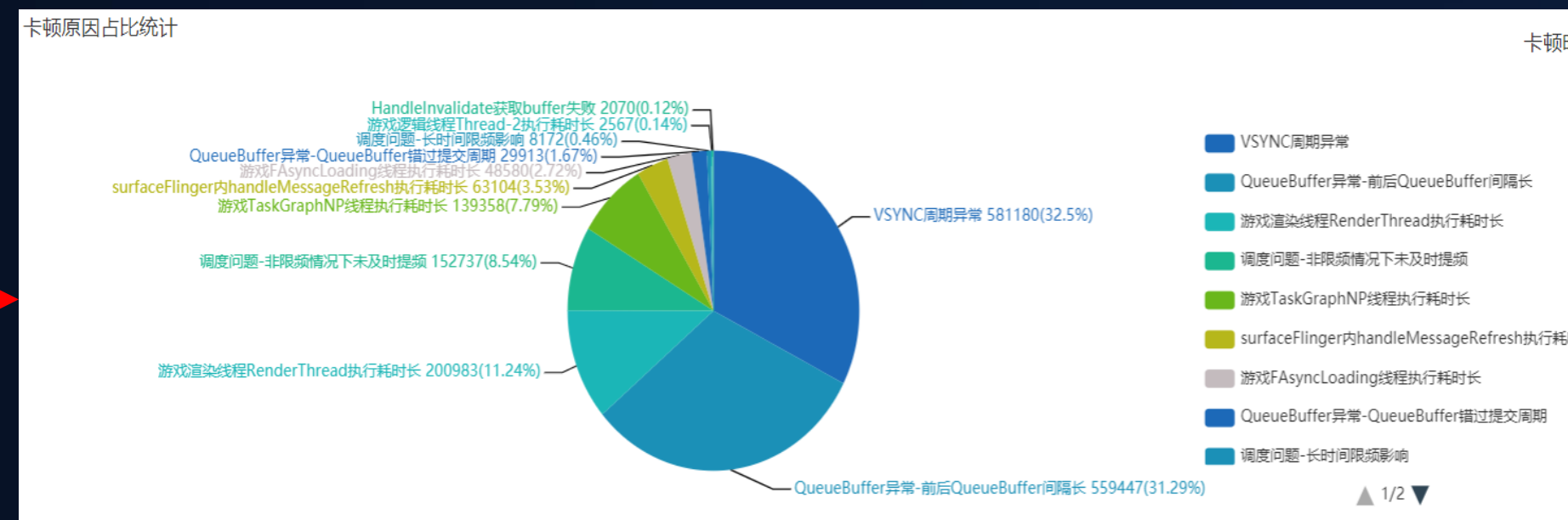
卡顿阈值/抓取次数设定



卡顿时刻截图



卡顿时刻自动触发systrace抓取



卡顿原因统计

特性4：细粒度分析之单/多帧采集回放

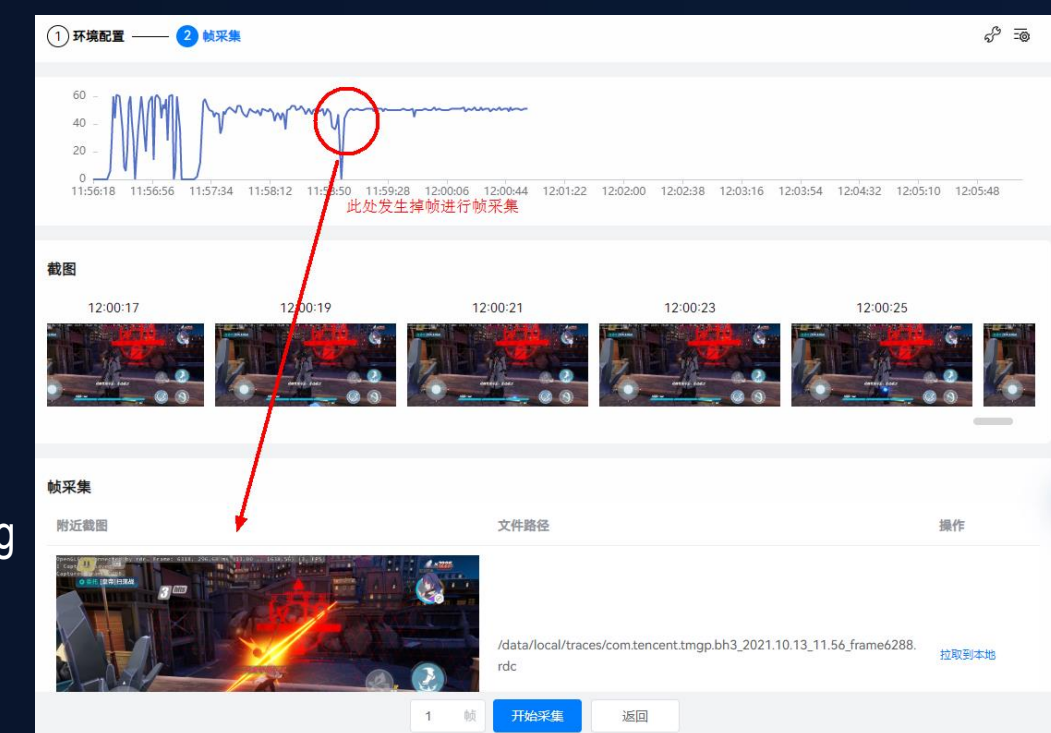


采集



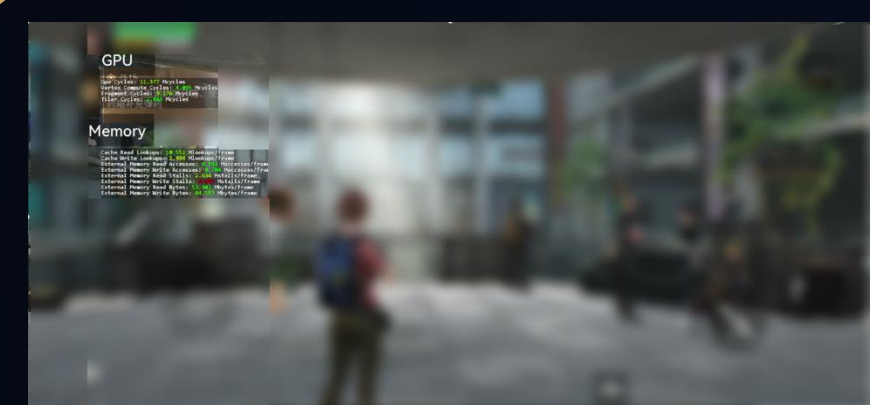
DevecoTesting抓取

Deveco Testing 中进行单多帧设置抓取



根据实时帧率采集关键帧 开发者高自由度配置单多帧

支持PC、Phone的文件回放



单帧回放

单多帧回放

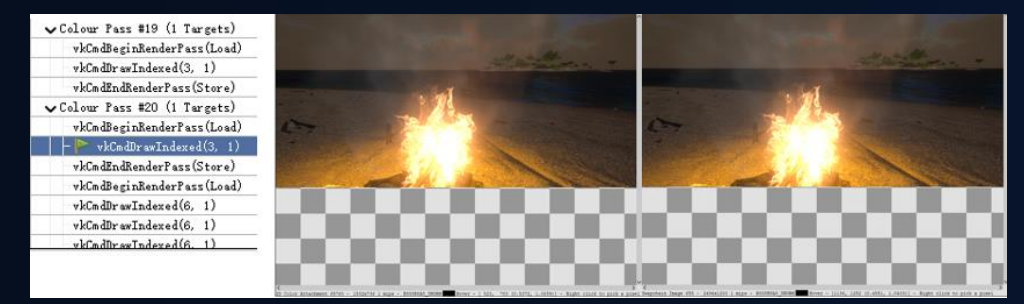


移动端支持单多帧抓取与回放

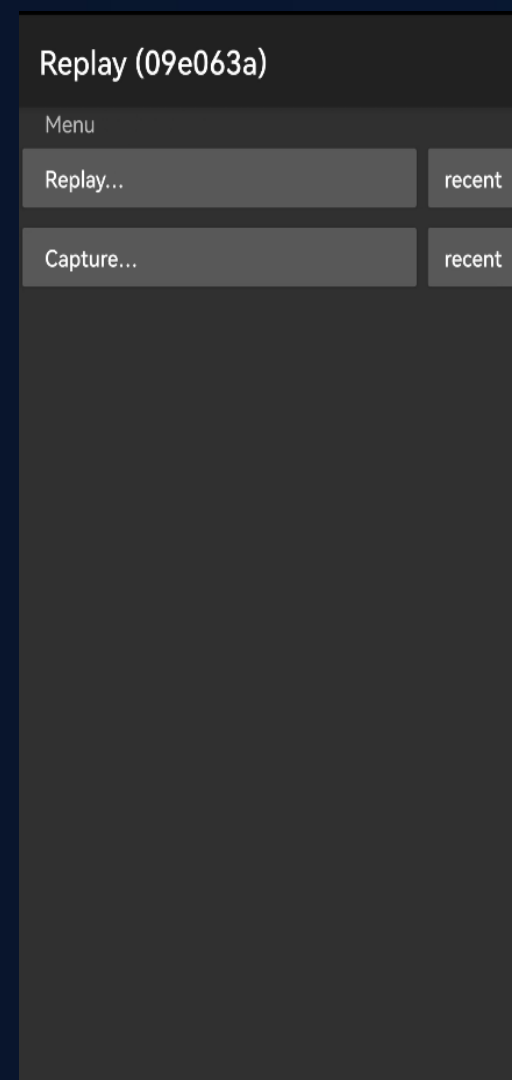


多帧回放

绘制优化专家建议 如：顶点优化、DC合并、texture压缩等



专家建议



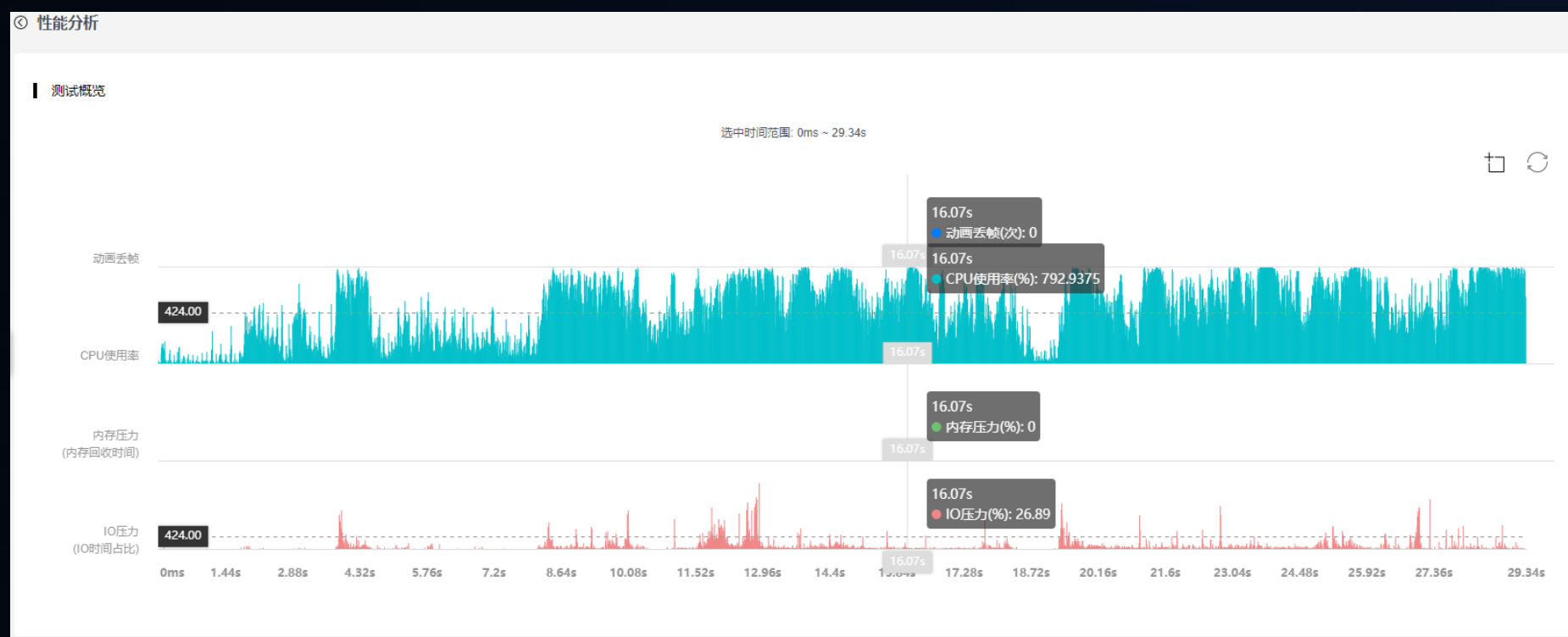
支持脱离PC抓取回放 (移动侧)

功能：

- 1、单多帧采集
- 2、单多帧回放
- 3、GPUCounter采集，并通过颜色来标示当前状态（红色代表重载，绿色代表轻载）

单帧采集/回放工具后续结合GPU hw counters可以抽丝剥茧的发现GPU瓶颈根因

特性5：细粒度分析之SOC统计



关键耗时段的CPU/内存/IO统计图



CPU IO

CPU TOP50进程

选中时间范围: 0ms ~ 29.34s

| 进程id | 进程 | 线程id | 线程 | CPU使用率(%) | Cycles |
|-------|---|-------|-----------------|-----------|----------------|
| 922 | /system/bin/traced_probes | 922 | traced_probes | 21.5422 | 1009634840.066 |
| 20544 | usap64 | 22601 | tp-io-10 | 12.6796 | 792187161.316 |
| 760 | /system/bin/surfaceflinger | 760 | surfaceflinger | 10.1864 | 560687491.256 |
| 2 | ktreadd | 1290 | hisi_hcc | 9.934 | 500115478.394 |
| 20544 | usap64 | 22517 | HeapTaskDaemon | 9.867 | 655429730.862 |
| 912 | /system/bin/adbd | 912 | adbd | 9.36 | 494233530.307 |
| 719 | /vendor/bin/hw/android.hardware.gnss.composer@2.2-service | 864 | HwBinder:719_2 | 8.105 | 366739602.257 |
| 2 | ktreadd | 1291 | hisi_pxdata | 8.046 | 416761440.425 |
| 2 | ktreadd | 351 | workingset_coll | 7.4509 | 435021103.048 |
| 20544 | usap64 | 22620 | APM_light-weigh | 7.139 | 439342192.505 |

进程id: 760

- CPU0: 1.52%
- CPU1: 2.0197%
- CPU2: 1.6541%
- CPU3: 2.0247%
- CPU4: 0.3176%
- CPU5: 0.3399%
- CPU6: 1.0685%
- CPU7: 1.2418%

关键耗时段TOP进程/线程的CPU占用统计



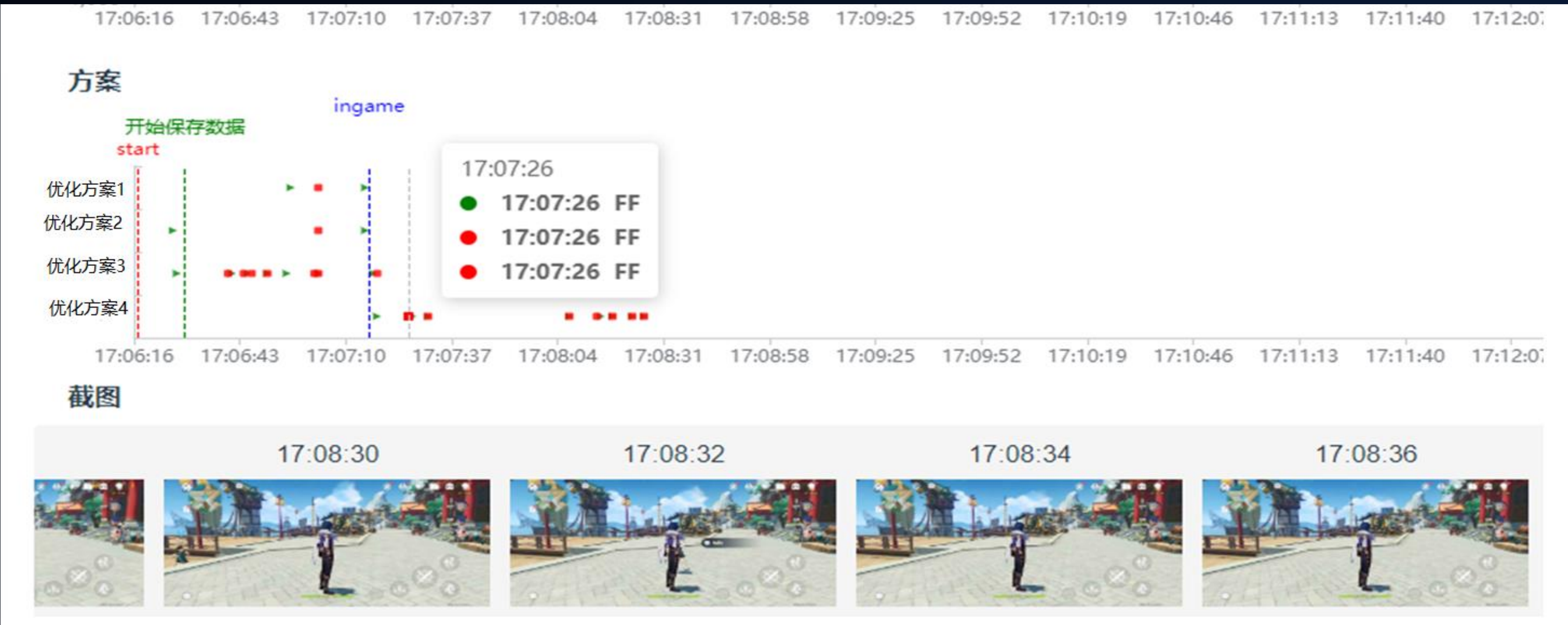
分析报告

动画丢帧分析 异常启动分析

| APP维度 | 发生时间 | 一级定界 | 二级定界 | 根因分析 | 总时长 | 优化建议 |
|-----------------------------|-----------------|------------|---------------|---------------------------|------|-----------------|
| com.huawei.android.launcher | 00:00:2.40599 | 关键线程被抢占 | 主线程被抢占 | Runnable运行时长:5ms | 17ms | 建议优化调度策略 |
| com.huawei.android.launcher | 00:00:2.40599 | 关键线程运行时... | 关键线程CPU频点供给不足 | 非小核运行时长占比: 100%,频点低于1.... | 17ms | 关键线程及时调频 |
| com.huawei.android.launcher | 00:00:2.40599 | 关键线程运行时... | IPC耗时: 8ms | sleep总时长: 7ms;TOP3 IPC 栈 | 17ms | 建议优化IPC或IPC耗时方法 |
| com.huawei.android.launcher | 00:00:10.429437 | 关键线程运行时... | 关键线程CPU频点供给不足 | 非小核运行时长占比: 100%,频点低于1.... | 26ms | 关键线程及时调频 |
| com.huawei.android.launcher | 00:00:10.429437 | 关键线程运行时... | | | 26ms | |

丢帧异常耗时的SOC跟因分析及优化建议

特性6：优化方案实时展示



方案

| 方案名称 | 生效时长占比 | 平均帧率 | 抖动率 | 最大壳温 | 归一化电流 |
|-------|--------|-------|-------|--------|---------|
| 优化方案1 | 38.3% | 49.02 | 1.85% | 47.265 | 1066.86 |
| 优化方案2 | 41.6% | 49.53 | 8.10% | 47.188 | 1355.72 |
| 优化方案3 | 90.0% | 46.67 | 7.17% | 47.265 | 1234 |
| 优化方案4 | 75.9% | 50.26 | 2.90% | 47.265 | 1196.61 |

总结

- HarmonyOS自研图形栈是OS的核心基座，包括ArkUI 2D和3D
- 图形栈测试分析有三层：用户体验->系统->硬件
- Devco Testing的图形测试服务化工具帮助开发者简化测试，并提供深度分析和优化建议

DevEco Testing获取方式



<https://devicepartner.huawei.com/cn/competency-test>

< HDC.Together >

华为开发者大会 2021

扫码参加1024程序员节

<解锁HarmonyOS核心技能，赢取限量好礼>

开发者训练营

Codelabs 挑战赛

HarmonyOS技术征文

HarmonyOS开发者创新大赛



扫码了解1024更多信息



报名参加HarmonyOS开
发者创新大赛

谢谢



欢迎访问HarmonyOS开发者官网



欢迎关注HarmonyOS开发者微信公众号