安排

流畅度-度量

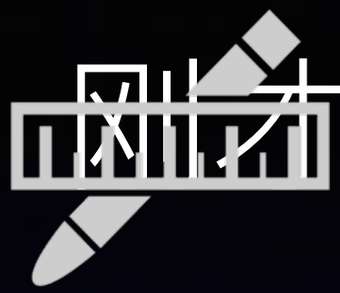
响应时延

动手实验

度量：度量价值

“If You Can't Measure It, You Can't Manage It”

彼得·德鲁克



卡顿发现手段中哪个方案适合度量

- 开发者选项(gfxinfo)
- SurfaceFlinger
- 日志里面的SkipFrame
- Looper监控
- DoFrame监控
- 用户反馈Keyword统计

基于DoFrame的流畅度度量

分组	预定义		即通应用部	社交平台部	即通产品部				
版本	标准值	所有	群会话	好友动态	消息列表	C2C会话列表	联系人列表(分组)	讨论组会话列表	合计
7.1.0.692	**98.0%**	98.4%(7817K)	98.4%(2414K)	98.3%(2359K)	98.4%(1037K)	98.2%(1062K)	98.6%(339K)	98.6%(103K)	98.3%
7.0.0.676	**98.0%**	97.6%(614K)	97.6%(206K)	97.7%(163K)	97.2%(91K)	97.3%(77K)	97.6%(28K)	98.0%(7K)	97.3%

卡顿时间占比 = 总卡顿时间/总使用时长

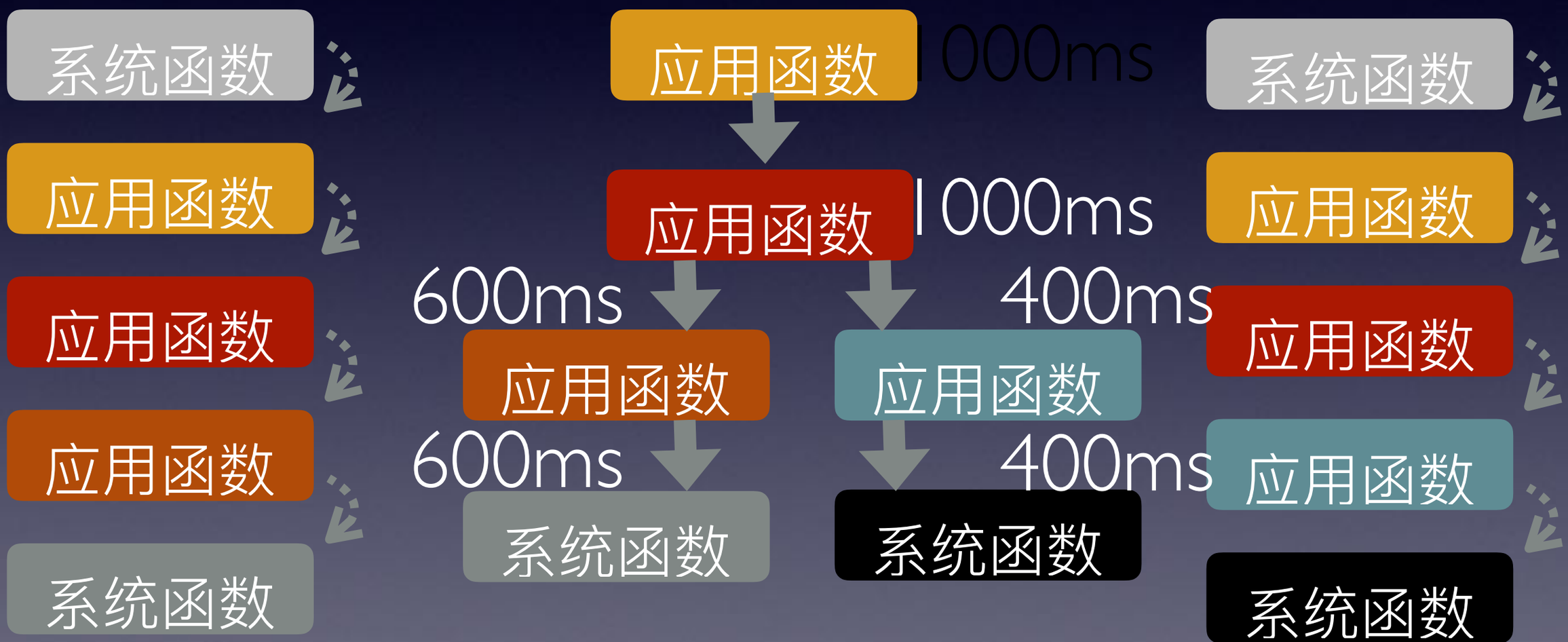
人均卡顿次数=卡顿总次数/总人数(去重)

卡顿人数占比=卡顿人数（去重）/总使用人数

这些指标有什么优劣势?

借助我们上报的堆栈信息还可以有什么度量的方式

卡顿堆栈聚合

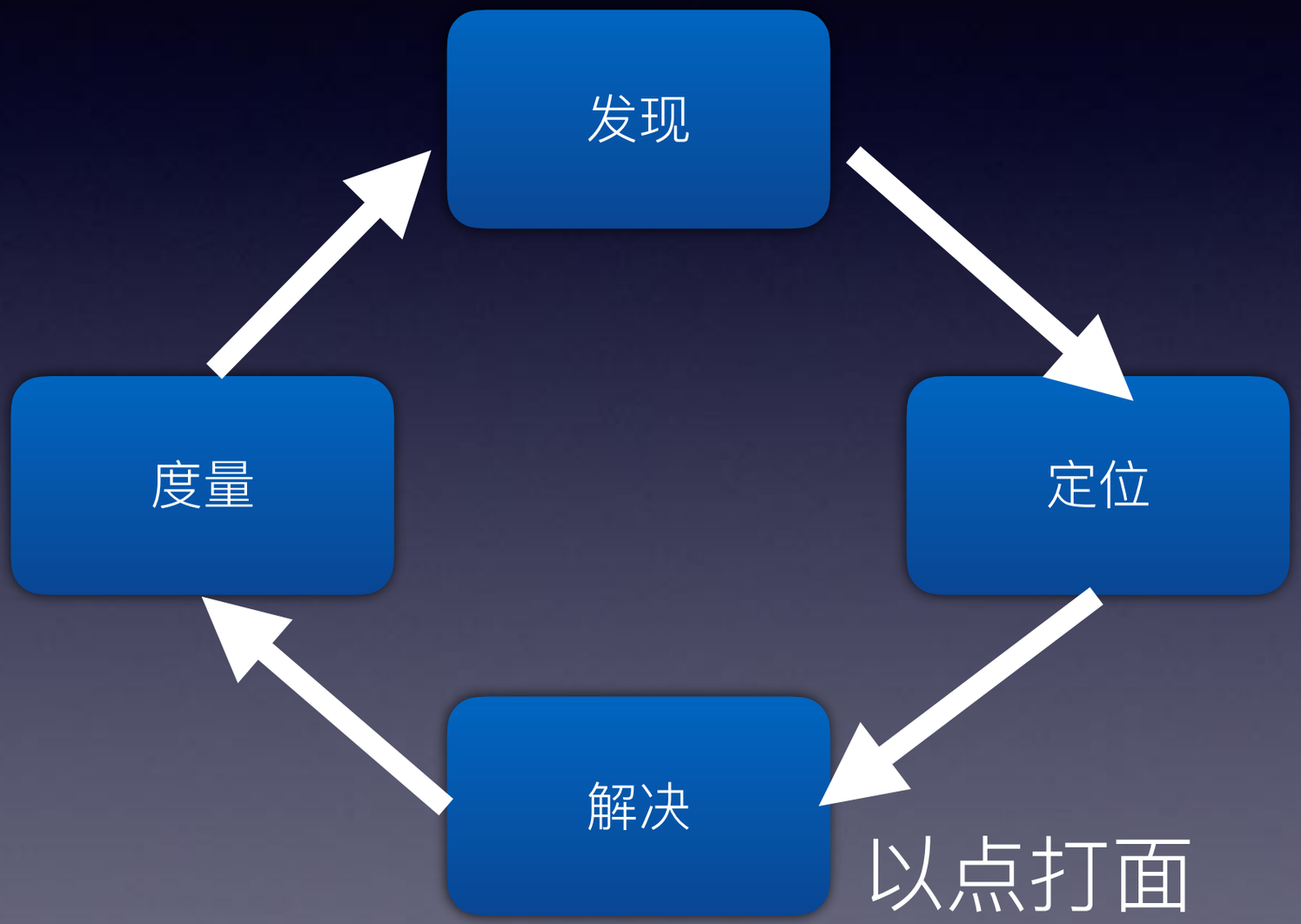

600ms
系统函数
应用函数
应用函数
应用函数
系统函数
000ms
应用函数
1000ms
应用函数
600ms
应用函数
600ms
系统函数
400ms
应用函数
400ms
系统函数
400ms
应用函数
应用函数
应用函数
应用函数
系统函数

另外一种度量

节点	辅助功能	版本占%	基准版本占%	版本占比变化	耗时(秒)
com.tencent.wi	展开红色	14.606	11.032	32.4 %	147
com.tencent .getView	展开红色	9.637	7.150	34.8 %	97
com.tenc dapter1.getView	展开红色	9.637	7.140	35.0 %	97
com.tencent ItemLayout.onMeasure	展开红色	2.422	1.384	75.0 %	24
com.tencent motionAdapter.getView	展开红色	0.780	0.215	262.1 %	8
com.tencent.mobil elViewPagerAdapter.instantiateItem	展开红色	0.436	0.128	239.8 %	4
com.tencent el.AudioPanel.onMeasure	展开红色	0.298	0.109	174.4 %	3
com.tencent .getItemViewType	展开红色	0.177	0.105	69.2 %	2
com.tencent ayout	展开红色	0.108	0.058	86.7 %	1
com.tencent .onLayout	展开红色	0.070	0.000	19645.1 %	1
mqq.app.AppAc	展开红色	13.609	9.663	40.8 %	137

小结

完整且量化



贴近用户

至上而下
至下而上

以点打面

从用户出发的交互类性能
响应延时

腾讯 victorhuang

有界面的才有的交互类性能

	最佳	好	可接受
响应延迟	≦100 ms	≦200 ms	≦500 ms
图形动画	≧120 fps	≧60 fps	≧30 fps
视频回放	≧60 fps	≧30 fps	≧20 fps

表 1：用户体验的行业经验值的示例（来源：英特尔公司，2011 年）

启动速度、界面切换速度、图片加载速度、视频首帧

流畅度

从点按钮到界面渲染出
来最核心的有哪几步?

看一下App启动
第一步main，创建looper

```
public final class ActivityThread {

    ......

    public static void main(String[] args) {
        .....
        Looper.prepareMainLooper();        //1

        ActivityThread thread = new ActivityThread();       //2
        thread.attach(false);

        if (sMainThreadHandler == null) {
            sMainThreadHandler = thread.getHandler();         //3
        }
      ......
    }

     ......

    final ApplicationThread mAppThread = new ApplicationThread();
    final Looper mLooper = Looper.myLooper();
    final H mH = new H()
}
```

看一下App启动
开始打开activity的旅程

```
c final class ActivityThread {

.....

ublic static void main(String[] args) {
  .....
  Looper.prepareMainLooper();       //1

  ActivityThread thread = new ActivityThread();      //2
  thread.attach(false);

  if (sMainThreadHandler == null) {
      sMainThreadHandler = thread.getHandler();      //3
  }
 ......


......

inal ApplicationThread mAppThread = new ApplicationThread();
inal Looper mLooper = Looper.myLooper();
inal H mH = new H()
```

```
public final class ActivityThread {

    ......

    public void handleMessage(Message msg) {
            switch (msg.what) {
                case LAUNCH_ACTIVITY: {
                    .....
                    handleLaunchActivity(r, nu
                    .......
                    break;
                    }
            }
        }
    }
```

打开activity的关键步骤

ActivityThread.handleLaunchActivty

ActivityThread.performLaunchActivity

Activity.onCreate()

Activity.setContentView()

ActivityThread.handleResumeActivity()

ViewRootImpl.performTraverals()

ViewRootImpl.performTraverals 之后

- measure
- layout
- draw

想起了什么?

复习一下流畅度渲染分步骤

chTouchEvent()
ouchEvnent()

getView()

nMeasure()

onLayout()

draw()
patchDraw()
onDraw

Event

(AdapterView)
Invalidate

Adapter getView()

Adapter getViewForPosition()

Measure

Layout

Draw

getDisplayList ->DrawDisplaylist->SwapBuffers

StrictMode

Lint

HierarchyViewer

dumpsys gfxinfo

GLTrace/GLTrace跟踪

Systrace

TraceView

为什么说是相通的?

Android

looper

tchTouchEvent()
ouchEvnent()

getView()

onMeasure()

onLayout()

draw()
spatchDraw()
onDraw

渲染分步骤

触发事件

doFrame

Event

(AdapterView)
Invalidate

Adapter getView()

Measurement

Layout

Draw

getDisplayList
->DrawDisplaylist
->SwapBuffers

iOS

MainLoop

Event

布局

[CALayer layoutSublayers]

[UIView layoutSubviews]

绘制

[CALayer display]

[UIView drawRect]

案例+实操

复习一下


main
启动app
prepareMainLooper
流畅度
handleLaunchActivty
onCreate
setContentView
Inflat Layout
启动界面
handleResumeActivity
doFrame
Inflat Layout
getView
getViewForPosition
ViewRootImpl.performTraverals()
5.0
Measure
measureHierarchy
Layout
dispatchOnPreDraw
updatePostion
Draw
performDraw

Main之前是什么?

作业

- 埋入流畅度，启动性能相关的缺陷
- 代码中添加注释，指明埋入位置
  （例如/*此处埋入流畅度缺陷*/）
- 提交代码到给TA