
公映许可证
电审故字[2018]第
网龙网龙公司
NETDRAGON WEBSOFT INC.

网龙网络公司
NETDRAGON WEBSOFT INC.

如何做有成效的性能测试？

CPA客户端性能测试保障体系

讲师：田志红

课程时间：20181216

CONTENTS

目 录

1 为什么要做性能测试

2 性能瓶颈是如何形成的

3 CPA性能测试保障体系

4 总结

网龙网络公司
NETDRAGON WEBSOFT INC.

01 什么要做性能测试

为什么要做性能测试

网龙网络公司
NETDRAGON WEBSOFT INC.

为什么要做性能测试

网龙网络公司
NETDRAGON WEBSOFT INC.

上工治未病

中工治欲病

下工治已病

为什么要做性能测试

网龙网络公司
NETDRAGON WEBSOFT INC.

竞争力

功 能 决 定 现 在 ， 性 能 决 定 未 来

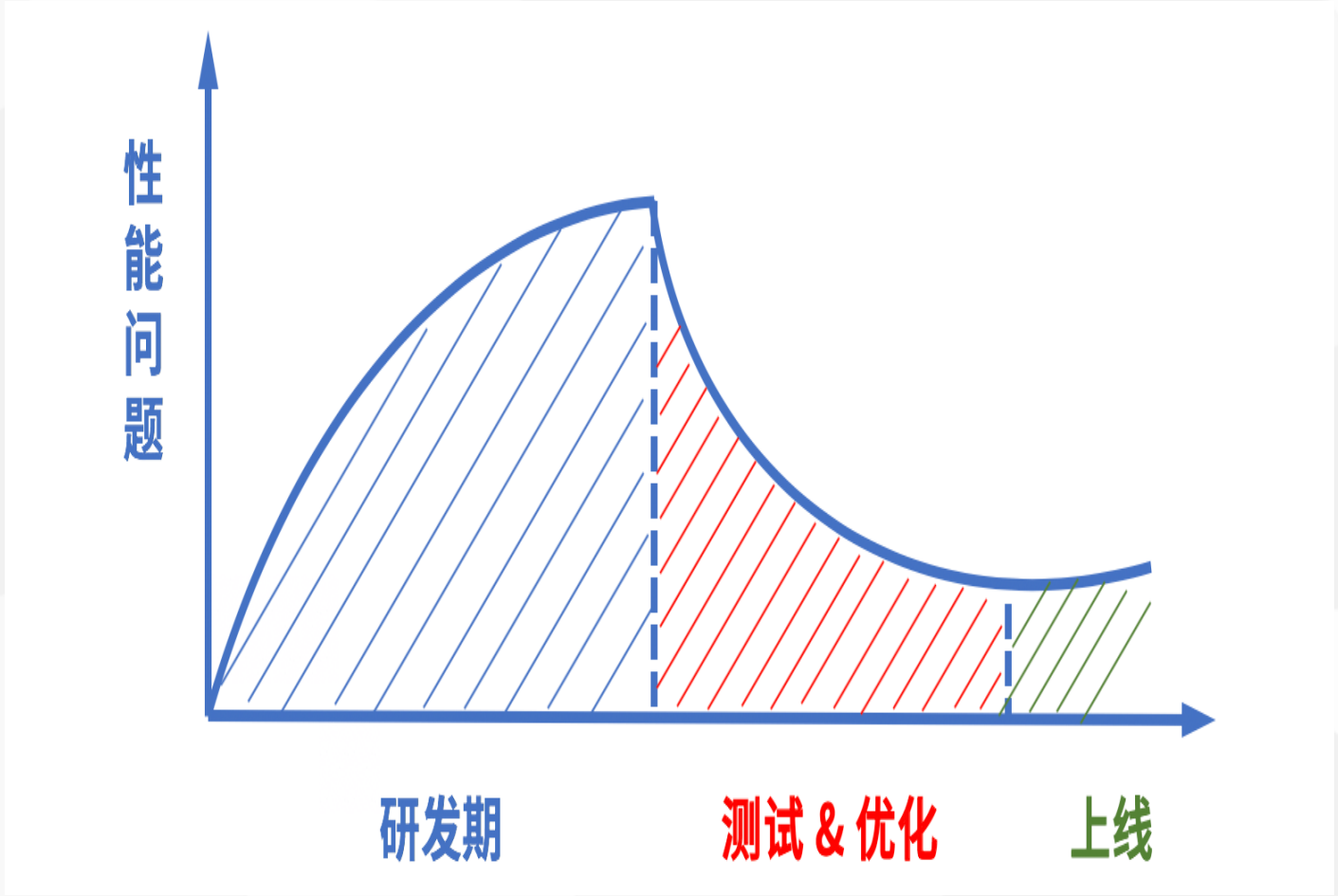
- 用户体验
- 产品空间
- 省钱

网龙网络公司
NETDRAGON WEBSOFT INC.

02 性能瓶颈是如何形成的

性能瓶颈是如何产生的

网龙网络公司
NETDRAGON WEBSOFT INC.



性能瓶颈是如何产生的

网龙网络公司
NETDRAGON WEBSOFT INC.

性能优化时间紧难度大

研发时间更长

消磨意志

网龙网络公司
NETDRAGON WEBSOFT INC.

03 CPA性能测试保障体系

CPA性能测试保障体系

网龙网络公司
NETDRAGON WEBSOFT INC.

CPA体系

网龙网络公司
NETDRAGON WEBSOFT INC.

CPA（Client Performance Analysis，客户端性能分析平台，包含MPA和PCPA）

简单易用

全面深入

多平台

多模式

CPA指标

网龙网络公司
NETDRAGON WEBSOFT INC.

CPA方案

网龙网络公司
NETDRAGON WEBSOFT INC.

通用技术

HOOK
PIPE
H5性能
深度流量

Android

资源监控
性能分析

PC

2D资源监控
U3D资源监控

iOS

资源监控
性能分析

通用技术

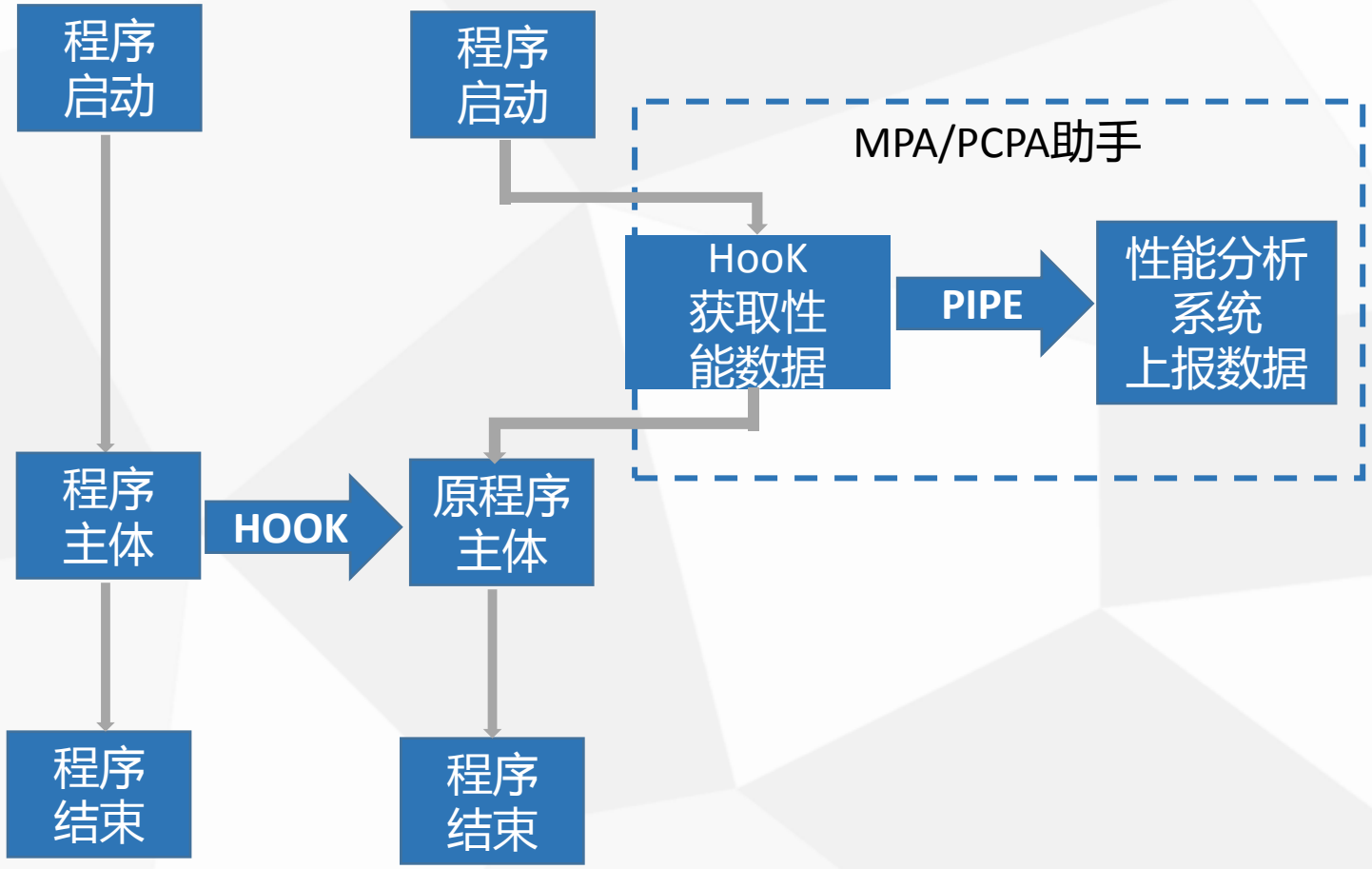
网龙网络公司
NETDRAGON WEBSOFT INC.

HOOK
PIPE
H5性能
深度流量

通用技术

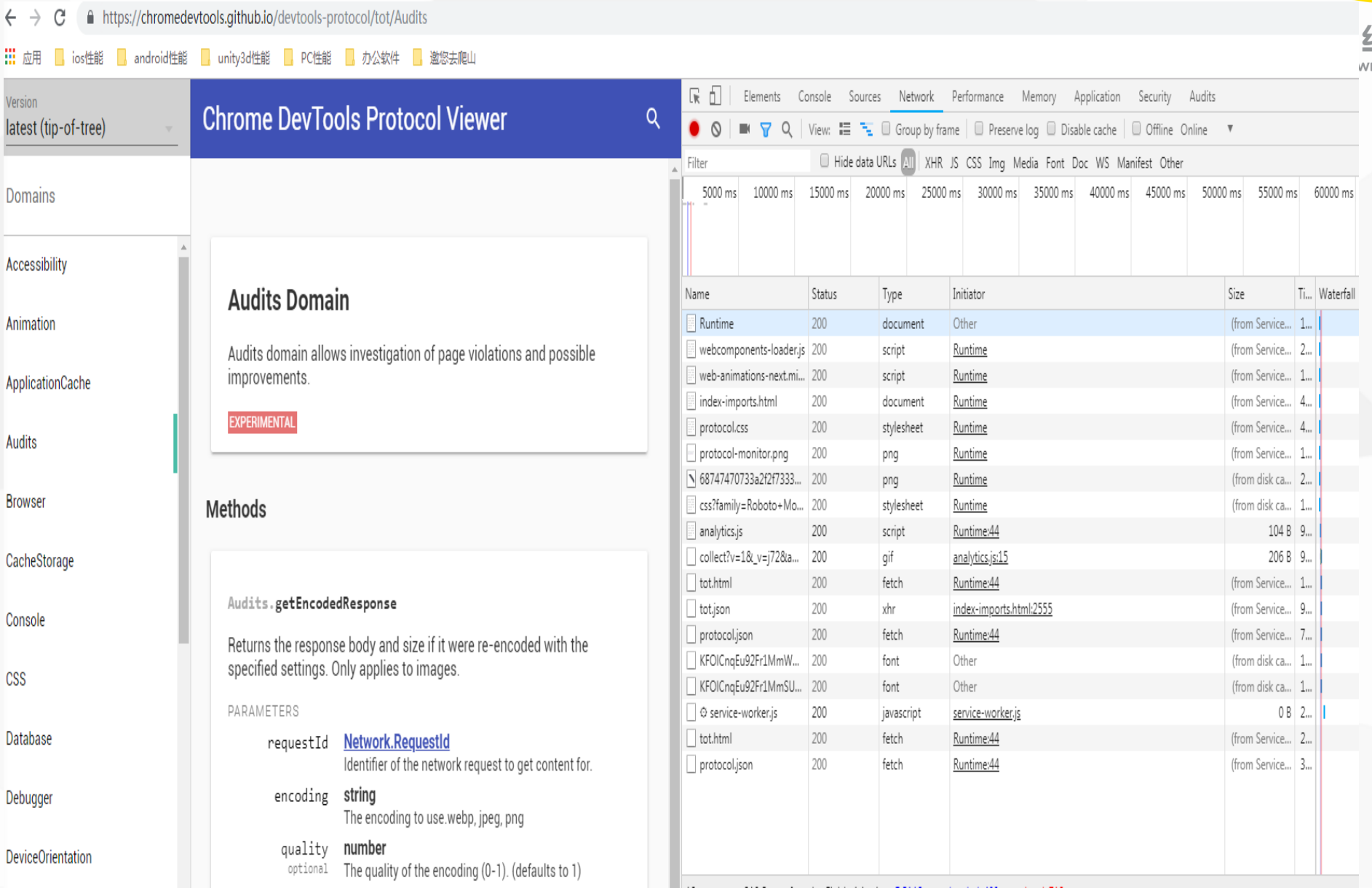
网龙网络公司
NETDRAGON WEBSOFT INC.

Hook
PIPE



通用技术

H5性能


https://chromedevtools.github.io/devtools-protocol/tot/Audits
应用
ios性能
android性能
unity3d性能
PC性能
办公软件
邀您去爬山
Version
latest (tip-of-tree)
Chrome DevTools Protocol Viewer
Domains
Accessibility
Animation
ApplicationCache
Audits
Browser
CacheStorage
Console
CSS
Database
Debugger
DeviceOrientation
Audits Domain
Audits domain allows investigation of page violations and possible improvements.
EXPERIMENTAL
Methods
Audits.getEncodedResponse
Returns the response body and size if it were re-encoded with the specified settings. Only applies to images.
PARAMETERS
requestId Network.RequestId
Identifier of the network request to get content for.
encoding string
The encoding to use.webp, jpeg, png
quality number optional
The quality of the encoding (0-1). (defaults to 1)
Elements
Console
Sources
Network
Performance
Memory
Application
Security
Audits
Group by frame
Preserve log
Disable cache
Offline
Online
Filter
Hide data URLs
All
XHR JS CSS Img Media Font Doc WS Manifest Other
5000 ms
10000 ms
15000 ms
20000 ms
25000 ms
30000 ms
35000 ms
40000 ms
45000 ms
50000 ms
55000 ms
60000 ms
Name
Status
Type
Initiator
Size
Waterfall
Runtime 200 document Other
webcomponents-loader.js 200 script Runtime
web-animations-next.mi... 200 script Runtime
index-imports.html 200 document Runtime
protocol.css 200 stylesheet Runtime
protocol-monitor.png 200 png Runtime
68747470733a2f2f7333... 200 png Runtime
css?family=Roboto+Mo... 200 stylesheet Runtime
analytics.js 200 script Runtime:44 104 B
collect?v=1&_v=j72&a... 200 gif analytics.js:15 206 B
tot.html 200 fetch Runtime:44
tot.json 200 xhr index-imports.html:2555
protocol.json 200 fetch Runtime:44
KFOlCnqEu92Fr1MmW... 200 font Other
KFOlCnqEu92Fr1MmSU... 200 font Other
service-worker.js 200 javascript service-worker.js 0 B
tot.html 200 fetch Runtime:44
protocol.json 200 fetch Runtime:44

通用技术

H5性能

```
# 封装上面4个事件对应的回调方法
class NetworkAPIImplemention(object):

    def __init__(self):
        self.request_dict = {}
        # 首个请求开始时间
        self.start = None

    def request_will_be_sent(self, **kwargs)
        if self.start is None:
            self.start = time.time()
        dict_http = {
            'url':kwargs.get('request').get(
            'start':kwargs.get('timestamp')
        }
        self.request_dict[kwargs.get('reques
        #print "loading:%s" % kwargs.get('re

    def loading_finished(self, **kwargs):
        # 服务器返回code 例如404也是finished
        self.request_dict[kwargs.get('reques
        self.request_dict[kwargs.get('reques

    def response_received(self, **kwargs):
        self.request_dict[kwargs.get('reques
        self.request_dict[kwargs.get('reques

    def loading_failed(self, **kwargs):
        self.request_dict[kwargs.get('reques
        self.request_dict[kwargs.get('requestId')]['error_text'] = kwargs.get('errorText')
```

```
browser = pychrome.Browser('http://127.0.0.1:%d' % 9222)
tab = browser.new_tab()
# 绑定回调函数
tab.Network.requestWillBeSent = network_api.request_will_be_sent
tab.Network.responseReceived = network_api.response_received
tab.Network.loadingFinished = network_api.loading_finished
tab.Network.loadingFailed = network_api.loading_failed
tab.start()
tab.Network.enable()
tab.Runtime.enable()
# 是否禁用缓存
if disable_cache:
tab.Network.setCacheDisabled(cacheDisabled=True)
tab.Page.navigate(url={你的页面地址})
tab.wait(10)
tab.stop()
self.browser.close_tab(tab)
# 获取的所有url详细信息
print network_api.request_dict
```

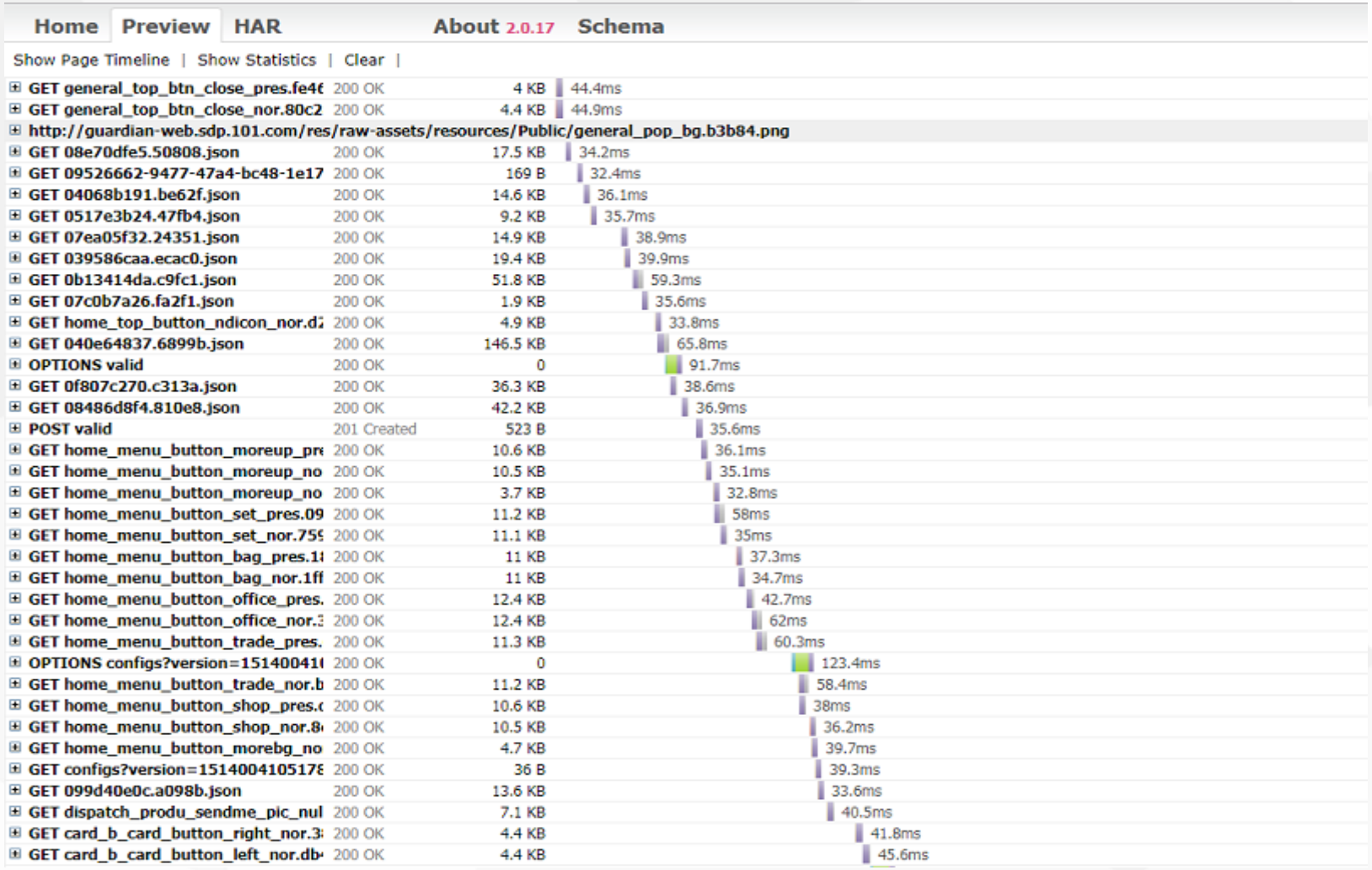
网龙网络公司

通用技术

网龙网络公司
NETDRAGON WEBSOFT INC.

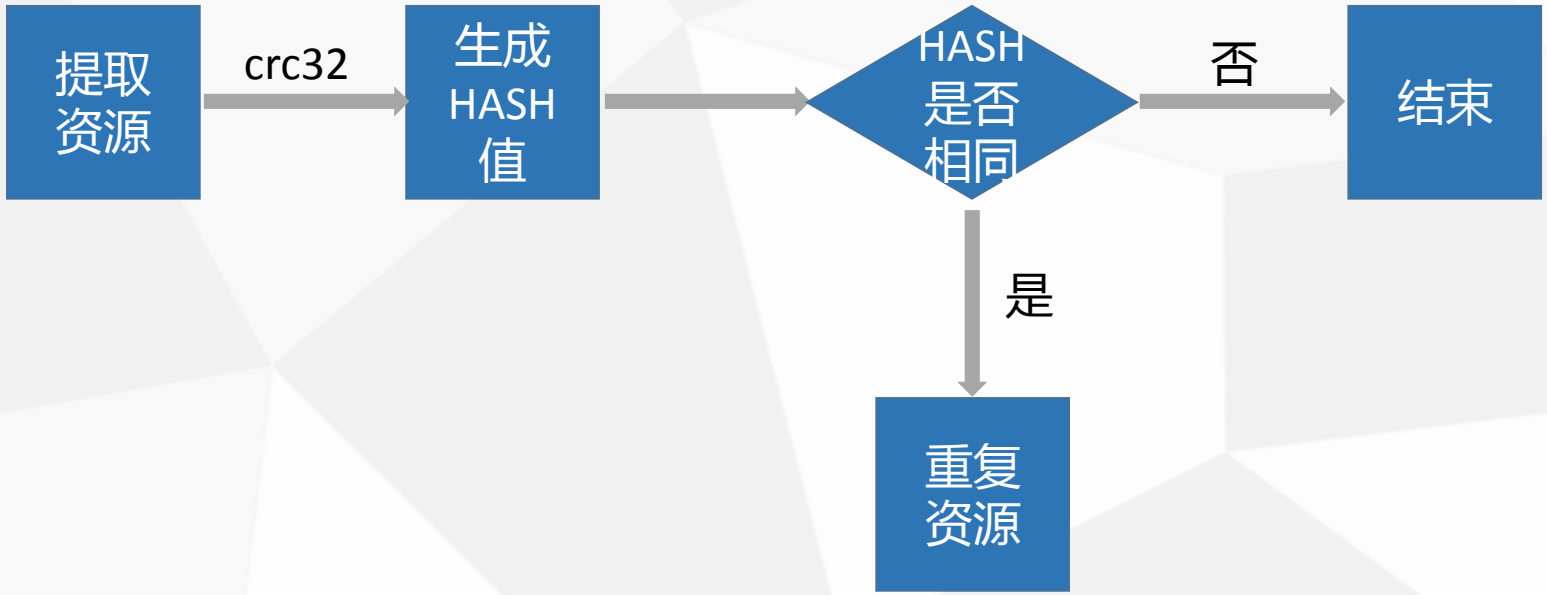
深度流量

流量:tcpdump+pcap2har+HarViewer，并根据雅虎军规，对重复请求、未合理使用缓存、大图片、未合理压缩等、400、500错误等问题进行分析



网龙网络公司
NETDRAGON WEBSOFT INC.

资源分析



Android实现方案

网龙网络公司
NETDRAGON WEBSOFT INC.

资源监控：GT 组件
主线程IO：严苛模式
内存抖动：监控日志，获取解析GC的日志
内存泄漏：
卡顿检测：
启动时间：
响应时间：
帧率：
流畅度：
电量：

Android实现方案

网龙网络公司
NETDRAGON WEBSOFT INC.

资源监控

《LeakCanary原理》 核心类分析

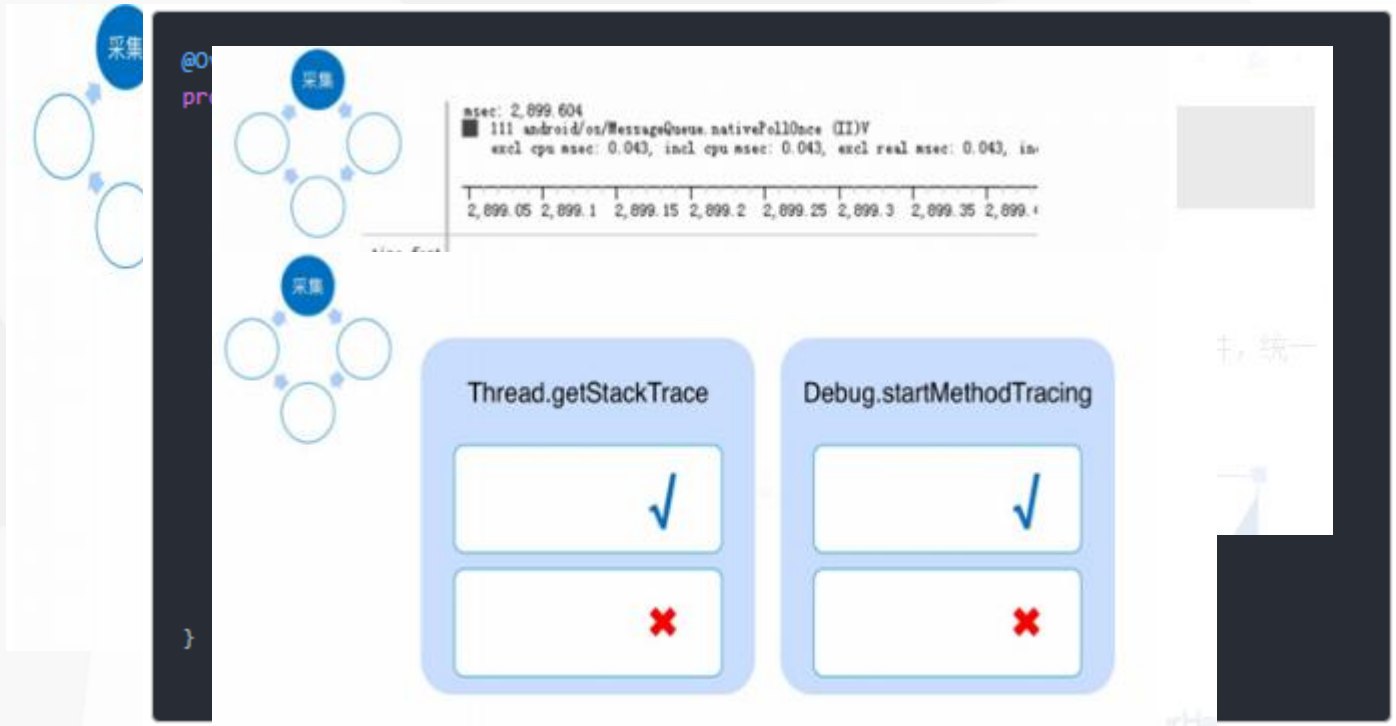
01	LeakCanary	源码解析
02	LeakCanary	SDK提供类
03	DisplayLeakActivity	内存泄漏的查看页面
04	HeapAnalyzerService	内存堆分析服务， 为了保证App进程不会因此受影响变慢&内存溢出，运行于独立的进程
05	HeapAnalyzer	分析由RefWatcher生成的堆转储信息， 验证内存泄漏是否真实存在
06	HeapDump	堆转储信息类，存储堆转储的相关信息
07	ServiceHeapDumpListener	一个监听，包含了开启分析的方法
08	RefWatcher	核心类， 翻译自官方： 检测不可达引用（可能地），当发现不可达引用时，它会触发HeapDumper(堆信息转储)
09	ActivityRefWatcher	Activity引用检测， 包含了Activity生命周期的监听执行与停止

ps:dump内存和分析内存会很消耗性能，测试进程和被测进程剥离

Android实现方案

网龙网络公司
NETDRAGON WEBSOFT INC.

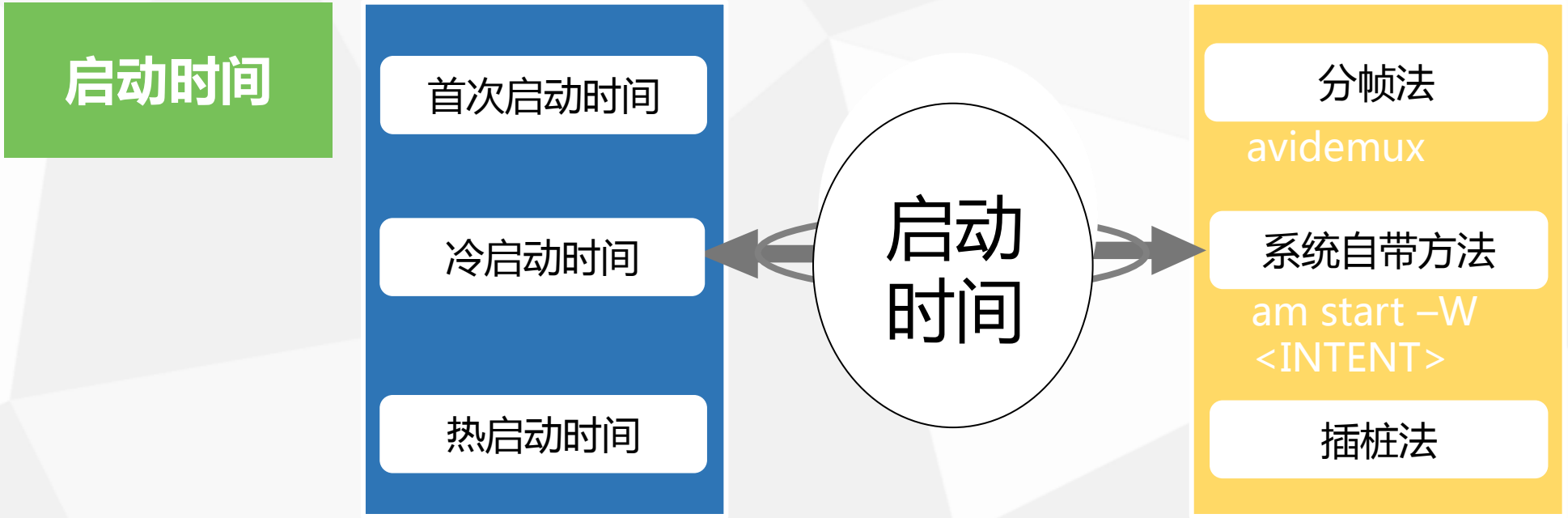
卡顿检测



Ps：超出标准时，增加trace获取一段事假的执行信息，一般时间断是2s，另外线上监控，卡顿标准可以提高一些。可以减少对性能的影响

Android实现方案

网龙网络公司
NETDRAGON WEBSOFT INC.



Android实现方案

网龙网络公司
NETDRAGON WEBSOFT INC.

响应时间

```
private void hookDispatchTouchEvent(XC_LoadPackage.LoadPackageParam lpparam){
    XposedHelpers.findAndHookMethod("android.view.View", lpparam.classLoader, "dispatchTouchEvent",MotionEvent.class,new XC_MethodHook(){
        @Override
        protected void beforeHookedMethod(MethodHookParam param) throws Throwable {
            View view = (View)param.thisObject;
            AccessibilityNodeInfo info;
            try{

//              LogUtil.d(TAG, "view name:"+view.getClass().getName());
                MotionEvent event = (MotionEvent)param.args[0];
                if(event.getAction()==MotionEvent.ACTION_UP){
                    String viewInfo = "click :"+view.getClass().getName()+"#"+view.getId()+"["+view.getHeight()+","+view.getWidth()+"]";
                    long currentTimeMillis = System.currentTimeMillis();
                    if(DEBUG){
                        LogUtil.d(TAG, viewInfo);
//                      mFrameStartTime = System.nanoTime();
                        LogUtil.d(TAG, "click start:"+currentTimeMillis);

                    }
                    info =view.createAccessibilityNodeInfo();
                    Rect rect = new Rect();
                    info.getBoundsInScreen(rect);
                    UiResponseTimeInfo data = new UiResponseTimeInfo();
                    data.setViewBottom(rect.bottom);
                    data.setViewClassName(view.getClass().getName());
                    data.setViewDescription(info.getContentDescription()!=null?info.getContentDescription(
                    data.setViewId(Integer.toHexString(view.getId()));
                    data.setViewInContext(view.getContext().getClass().getName());
                    data.setViewLeft(rect.left);
                    data.setViewRight(rect.right);
                    data.setViewText(info.getText()!=null?info.getText().toString():"");
                    data.setViewTop(rect.top);
                    mHookChoreographerListener.onTouchUp(data, currentTimeMillis);
```

```
@Override
public void onDoFrame(long frameTimeNanos,long doFrameStart,long doFrameEnd) {

    long jitterNanos = doFrameStart - frameTimeNanos;
    if(jitterNanos>mFrameIntervalNanos){
        mSkippedFrame_Per_Second += (jitterNanos/mFrameIntervalNanos);
    }
    //获得每帧结束时间doFrameEnd,计算点击到当前帧结束的时间来响应时间.
    mResponseTime = doFrameEnd - mResponseStartTime;

    //calculateFps(frameTimeNanos,doFrameStart);
}
```

Android实现方案

网龙网络公司
NETDRAGON WEBSOFT INC.

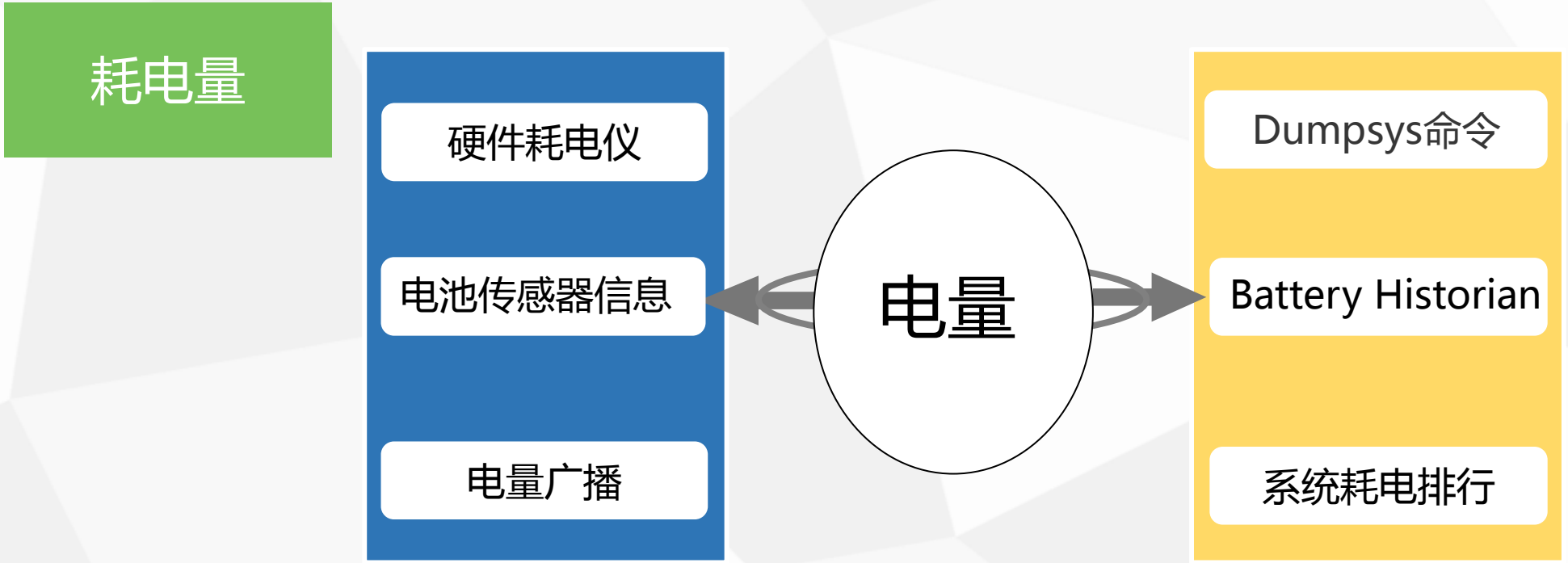
流畅度
帧率

```
try {
    //每隔1秒收集一次丢帧总数,并计算流畅度
    Thread.sleep(1000);
    if(DEBUG){
        Log.d(TAG,"frame["+(60-mSkippedFrame_Per_Second)+"],Skipped frames:"+mSkippedFrame_Per_Second);
    }

    UiFpsModel data = new UiFpsModel();
    data.setSkippedFps(mSkippedFrame_Per_Second);
    data.setSmoothness((60-mSkippedFrame_Per_Second)<0?0:(60-mSkippedFrame_Per_Second));
    data.setCreateTime(System.currentTimeMillis()/1000);
    data.setPreviousStack("");
    data.setStack("");
    data.setType(FPSType.DOFRAME.getValue());
    data.setUserId(mReportId);
```

Android实现方案

网龙网络公司
NETDRAGON WEBSOFT INC.



Android实现方案

网龙网络公司
NETDRAGON WEBSOFT INC.

耗电量

App电量统
Class : co
Method :

模块电流
Class : com.android.internal.os.PowerProfile
Method : getAveragePower()
XML : frameworks\base\core\res\res\xml\power_profiler.xml

- CP
- W
- Da
- W
- Se
- Bl
- Ca
- Fla

```
/**
 * Returns the average current in mA consumed by the subsystem for the given level.
 * @param type the subsystem type
 * @param level the level of power at which the subsystem is running. For instance, the
 *  signal strength of the cell network between 0 and 4 (if there are 4 bars max.)
 *  If there is no data for multiple levels, the level is ignored.
 * @return the average current in milliAmps.
 */
public double getAveragePower(String type, int level) {
    if (sPowerMap.containsKey(type)) {
        Object data = sPowerMap.get(type);
        if (data instanceof Double[]) {
            final Double[] values = (Double[]) data;
            if (values.length > level && level >= 0) {
                return values[level];
            } else if (level < 0 || values.length == 0) {
                return 0;
            } else {
                return values[values.length - 1];
            }
        } else {
            return (Double) data;
        }
    } else {
        return 0;
    }
}
```

PowerProfile	官方_N5(mA)
POWER_NONE	48.07
POWER_CPU_IDLE	221.9
POWER_CPU_AWAKE	3.5
POWER_CPU_ACTIVE	73.24
POWER_WIFI_SCAN	75.48
POWER_WIFI_ON	90.8
POWER_WIFI_ACTIVE	99.2
POWER_WIFI_CONTROLLER_IDLE	1.16
POWER_WIFI_CONTROLLER_RX	2.15
POWER_WIFI_CONTROLLER_TX	57.9|3MHz
POWER_WIFI_CONTROLLER_TX_LEVELS	88.2|4.224MHz
POWER_WIFI_CONTROLLER_OPERATING_VOLTAGE	99.6|6.528MHz
POWER_BLUETOOTH_CONTROLLER_IDLE	138.8|7.296MHz
POWER_BLUETOOTH_CONTROLLER_RX	149.6|8.832MHz
POWER_BLUETOOTH_CONTROLLER_TX	170.2|9.6MHz
POWER_BLUETOOTH_CONTROLLER_OPERATING_VOLTAGE	z 178.3|10.368MHz
POWER_MODEM_CONTROLLER_IDLE	z 189.1|11.904MHz
POWER_MODEM_CONTROLLER_RX	z 232.1|12.672MHz
POWER_MODEM_CONTROLLER_TX	z 256.5|14.976MHz
POWER_MODEM_CONTROLLER_OPERATING_VOLTAGE	266.4|15.74MHz
POWER_GPS_ON	287.7|17.28MHz
POWER_BLUETOOTH_ON	z 325.7|19.584MHz
POWER_BLUETOOTH_ACTIVE	z 386.2|22.656MHz
POWER_BLUETOOTH_AT_CMD	3.2
POWER_SCREEN_ON	17.4
POWER_RADIO_ON	
POWER_RADIO_SCANNING	
POWER_RADIO_ACTIVE	
POWER_SCREEN_FULL	
POWER_AUDIO	
POWER_VIDEO	
POWER_FLASHLIGHT	
POWER_CAMERA	
POWER_CPU_SPEEDS	
POWER_WIFI_BATCHED_SCAN	
POWER_BATTERY_CAPACITY	

iOS实现方案

网龙网络公司
NETDRAGON WEBSOFT INC.

资源监控：GT SDK
卡顿：BUGly
内存泄漏：
电量：

iOS实现方案

网龙网络公司
NETDRAGON WEBSOFT INC.

内存泄漏

内存泄漏：MLeaksFinder、OOMDetector

1.OOM监控：监控OOM，Dump引起爆内存的堆栈

2.大内存分配监控：监控单次大块内存分配，提供分配堆栈信息

3.内存泄漏检测：可检测OC对象、Malloc堆内存泄漏，提供泄漏堆栈信息

OOMDetector通过Hook系统底层的内存分配方法，能够记录到进程所有内存分配的堆栈信息，同时组件能够在对性能流畅度影响不大的情况下能够保证在App中独立运行

iOS实现方案

网龙网络公司
NETDRAGON WEBSOFT INC.

电量

表名内容

PLBatteryAgent_EventBackward_Battery 整台机器的电量数据，包含电流、电压、温度等，每 20 秒 左右一条数据

PLBatteryAgent_EventBackward_BatteryUI 电量百分比数据，每 20 秒一条数据

PLIOReportAgent_EventBackward_EneryModel 整机的详细电量数据。包含 CPUGPUDRAMISP 等关键信息。每半小时到一小时一条数据。

PLAccountingOperator_EventNone_NodesApp 结点信息，每个 APP 对应唯一的结点号。用来确定手机内具体哪个 App。

PLApplicationAgent_EventForward_ApplicationApp 运行信息，记录每个 App 在哪个时间段以什么状态运行

PLAppTimeService_Aggregate_AppRunTimeAPP 的运行时长统计，每个运行过的 APP，一小时一条数据

PLAccountingOperator_Aggregate_RootNodeEnergy APP 的电量详细数据，记录每个 APP 的CPU、GPU、DRAM、ISP 等的耗电信息。一小时更新一次数据。

量的值

PC实现方案

网龙网络公司
NETDRAGON WEBSOFT INC.

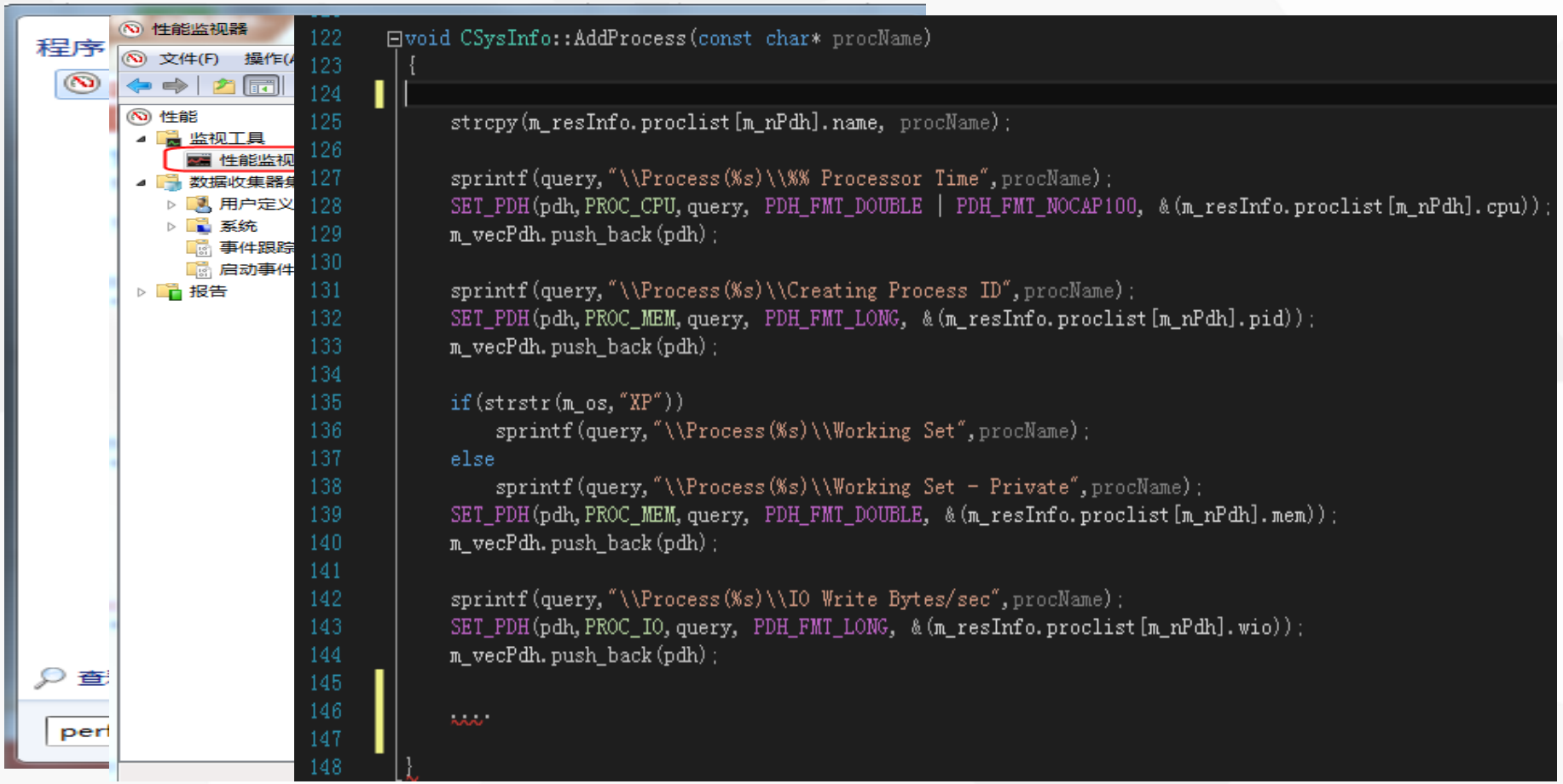
资源监控
响应时间
Unity mono内存
Unity drawcalls

PC实现方案

网龙网络公司
NETDRAGON WEBSOFT INC.

资源监控

资源监控：通过Performance Data Helper （PDH）

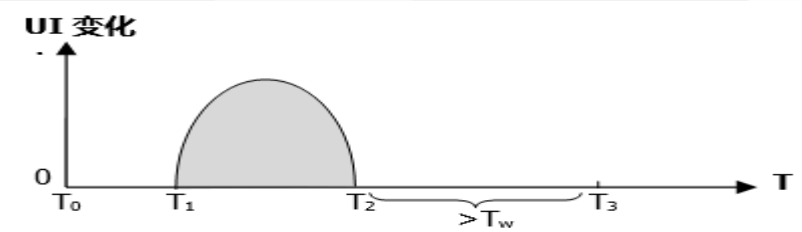



```
122  void CSysInfo::AddProcess(const char* procName)
123  {
124
125      strcpy(m_resInfo.proclist[m_nPdh].name, procName);
126
127      sprintf(query,"\\Process(%s)\\%% Processor Time",procName);
128      SET_PDH(pdh,PROC_CPU,query, PDH_FMT_DOUBLE | PDH_FMT_NOCAP100, &(m_resInfo.proclist[m_nPdh].cpu));
129      m_vecPdh.push_back(pdh);
130
131      sprintf(query,"\\Process(%s)\\Creating Process ID",procName);
132      SET_PDH(pdh,PROC_MEM,query, PDH_FMT_LONG, &(m_resInfo.proclist[m_nPdh].pid));
133      m_vecPdh.push_back(pdh);
134
135      if(strstr(m_os,"XP"))
136          sprintf(query,"\\Process(%s)\\Working Set",procName);
137      else
138          sprintf(query,"\\Process(%s)\\Working Set - Private",procName);
139      SET_PDH(pdh,PROC_MEM,query, PDH_FMT_DOUBLE, &(m_resInfo.proclist[m_nPdh].mem));
140      m_vecPdh.push_back(pdh);
141
142      sprintf(query,"\\Process(%s)\\IO Write Bytes/sec",procName);
143      SET_PDH(pdh,PROC_IO,query, PDH_FMT_LONG, &(m_resInfo.proclist[m_nPdh].wio));
144      m_vecPdh.push_back(pdh);
145
146      ....
147
148  }
```

网龙网络公司
NETDRAGON WEBSOFT INC.

响应时间

响应时间：通过监控鼠标最后一次操作时间作为响应时间的起点，通过观察界面静止时间（稳态时间）来判断响应时间的终点。



横轴为时间轴，T0时刻点击打开按钮，也就是起始时间点，T0~T1为系统响应时间，T1~T2为UI渲染时间，T2为时间终点。
代码中如何识别T2点？这里判断方法为UI在一段时间（Tw）内保持稳定,即T3-T2 > Tw即可认为T2（最后一次变化点）为终点时刻

遇到的问题：非静止的需要额外处理，使用了区域匹配、动态自动过滤

PC实现方案

网龙网络公司
NETDRAGON WEBSOFT INC.

MONO内存

MONO内存：通过HOOK PROFILER 接口。

```
long monoHeap = Profiler.GetMonoHeapSizeLong() / (1024 * 1024);
long monoUsedMemory = Profiler.GetMonoUsedSizeLong() / (1024 * 1024);

long totalReservedMemory = Profiler.GetTotalReservedMemoryLong() / (1024 * 1024);
long totalAllocatedMemory = Profiler.GetTotalAllocatedMemoryLong() / (1024 * 1024);
long totalUnusedReservedMemory = Profiler.GetTotalUnusedReservedMemoryLong() / (1024 * 1024);
```

PC实现方案

网龙网络公司
NETDRAGON WEBSOFT INC.

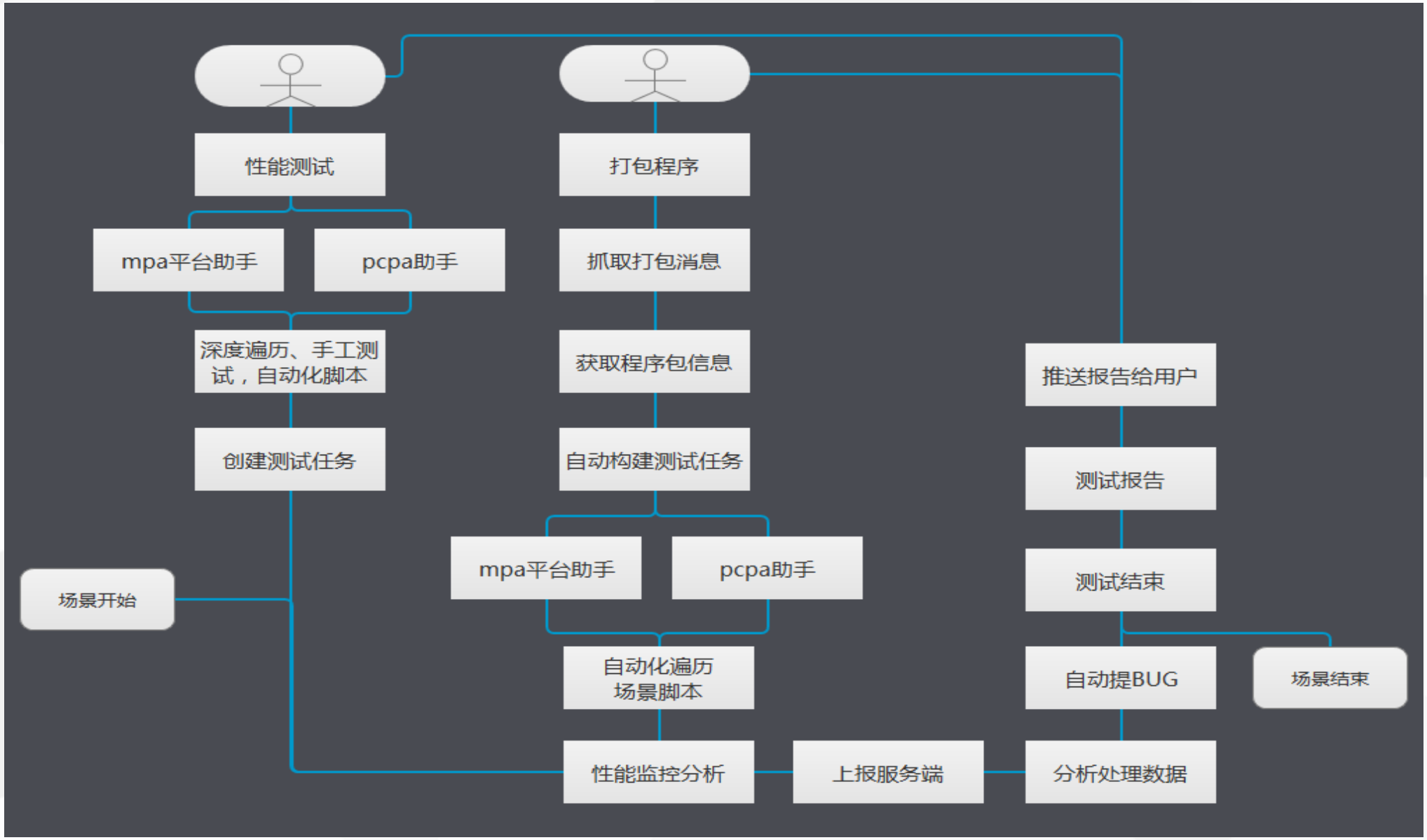
DrawCall

DrawCall：是CPU调用图形编程接口，比如DirectX或OpenGL，来命令GPU进行渲染的操作，Draw Call的次数是决定性能比较重要的指标。

```
22  void __stdcall CHookD3D11::hookD3D11DrawIndexed(ID3D11DeviceContext* pContext, UINT IndexCount, UINT StartIndexLocation, INT BaseVertexLocation)
23  {
24      pContext->DrawIndexed(IndexCount, StartIndexLocation, BaseVertexLocation);
25      nCurrDrawCall++;
26  }
27
28  void __stdcall CHookD3D11::hookD3D11Draw(ID3D11DeviceContext* pContext, UINT VertexCount, UINT StartVertexLocation)
29  {
30      nCurrDrawCall++;
31      pContext->Draw(VertexCount, StartVertexLocation);
32  }
33
34
35  HRESULT __stdcall CHookD3D11::hookD3D11Present(IDXGISwapChain* pSwapChain, UINT SyncInterval, UINT Flags)
36  {
37      if(Frame())
38      {
39          nDrawCall = nCurrDrawCall;
40          nFps = GetFPS();
41      }
42      nCurrDrawCall = 0;
43      return pSwapChain->Present(SyncInterval, Flags);
44  }
```

CPA流程

网龙网络公司
NETDRAGON WEBSOFT INC.



CPA标准

如何制定标准

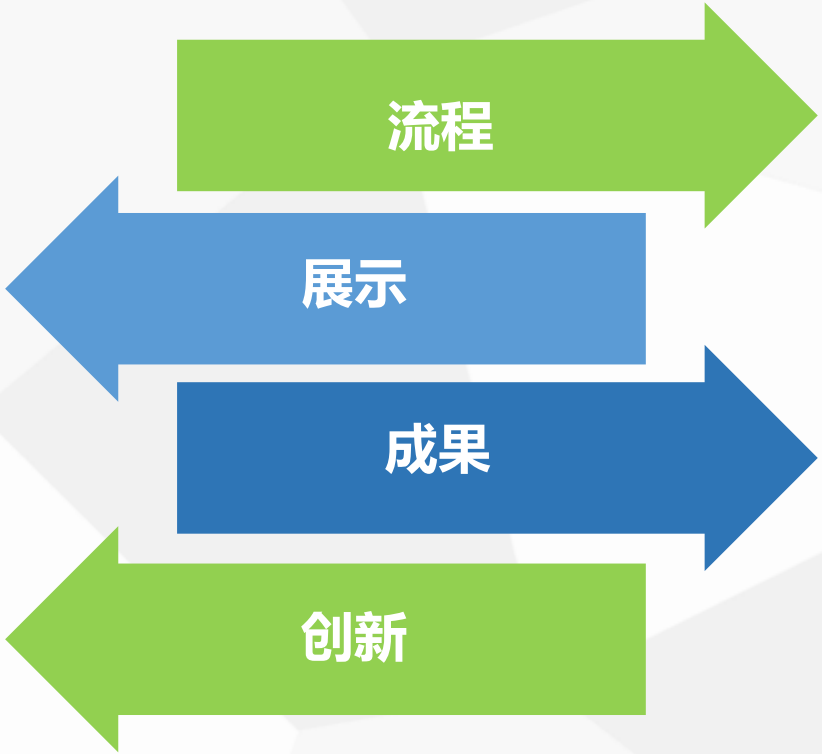
1、参考行业标准
2、参考历史数据
3、参考竞品数据
4、和开发一起立FLAG

A	B	C
性能测试项	标准	优先级
内存	避免Activity泄漏	P0
	运行过程中内存峰值<单个应用程序最大可能的内存的60%	P1
流量	每秒流量(KB)<5.12K	P1
	前台网络IO<60KB(要依赖网络下载完成才能展示的内容)	P1
	图片不超过60K	P1
CPU	前台cpu<40%	P1
	后台CPU<5%	P1
	启动时CPU<20%	P1
电量	后台运行2小时耗电不超过1mA	P1
	Alarm唤醒次数<20次/小时	P1
	WakeLock持有时间<5分钟/小时	P1
流畅度	1档机型的SM平均值>33，最小值>26	P0
	2档机型的SM平均值>30，最小值>24	P0
	3档机型的SM平均值>21，最小值>17	P0
	1、2档机的不允许出现连续丢帧>=6，连续丢帧>=3占比小于10%	P0
	3档机的连续丢帧>=6占比小于30%，连续丢帧>=3占比小于30%	P0
响应时间	1档机型的界面响应时间<2s	P0
	2档机型的界面响应时间<2s	P0
	3档机型的界面响应时间<3s	P0
	减少主线程IO	P1
	1档机型冷启动速度<2s	P0
	2档机型冷启动速度<2s	P0
	3档机型冷启动速度<3s	P0
	界面切换速度<500ms	P0
	避免黑屏	P0

注：
1. 标准参考腾讯、绿盟等性能测试标准
2. 出现P0则该项测试不通过，出现P1级该项测试为警告
Android平台：
1档机型的分类为：CPU-八核 1.7GHZ以上，RAM-4G。建议测试机型：OPPO R9, 华为P9
2档机型的分类为：CPU-四核 1.5GHZ以上，RAM-3G。建议测试机型：小米4LTE(3GB RAM高配版)、华为MATE8
3档机型的分类为：CPU-四核 1.2GHZ以上，RAM-2G。建议测试机型：VIVO Y51, HM NOTE 1S

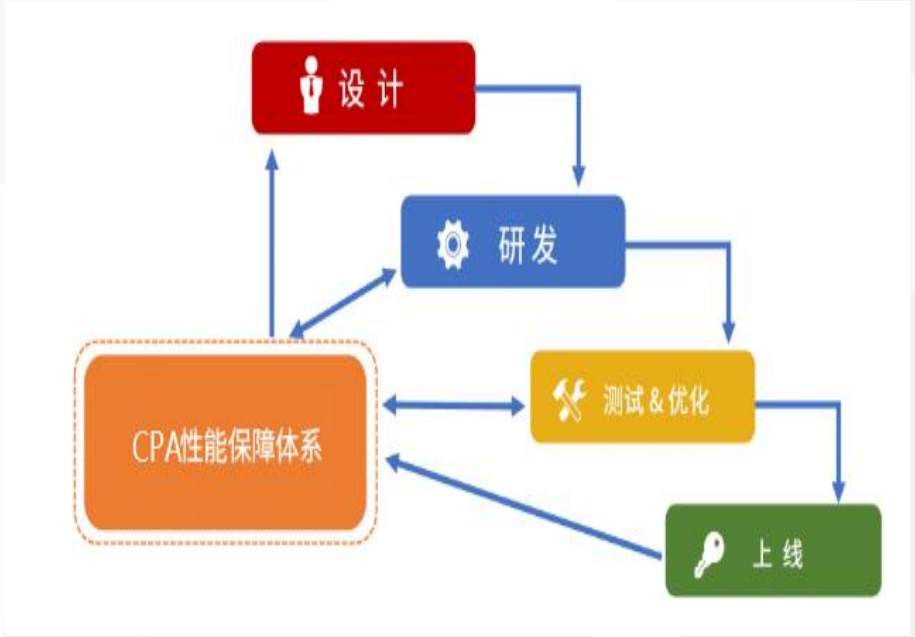
CPA成效

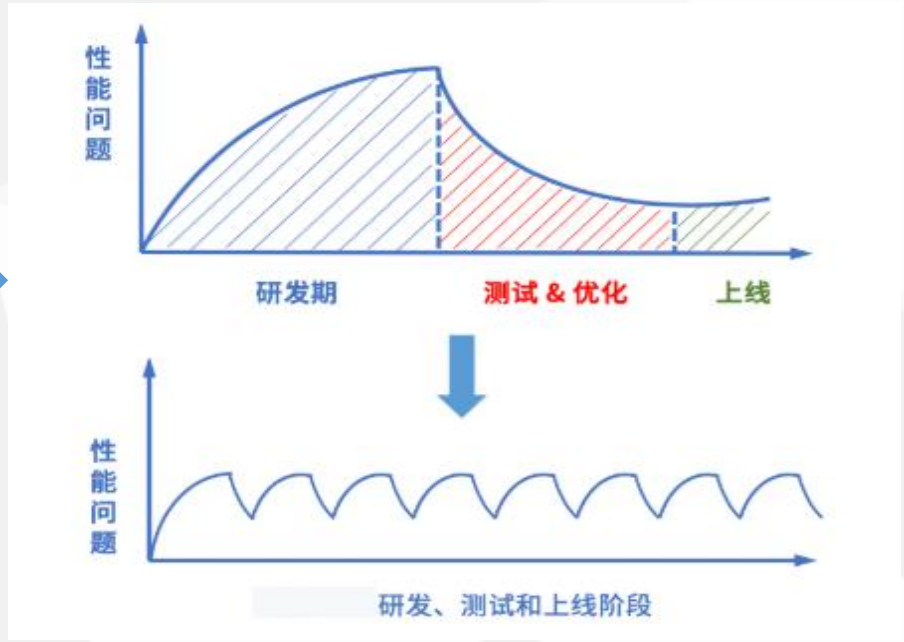
网龙网络公司
NETDRAGON WEBSOFT INC.



改进测试流程

网龙网络公司
NETDRAGON WEBSOFT INC.

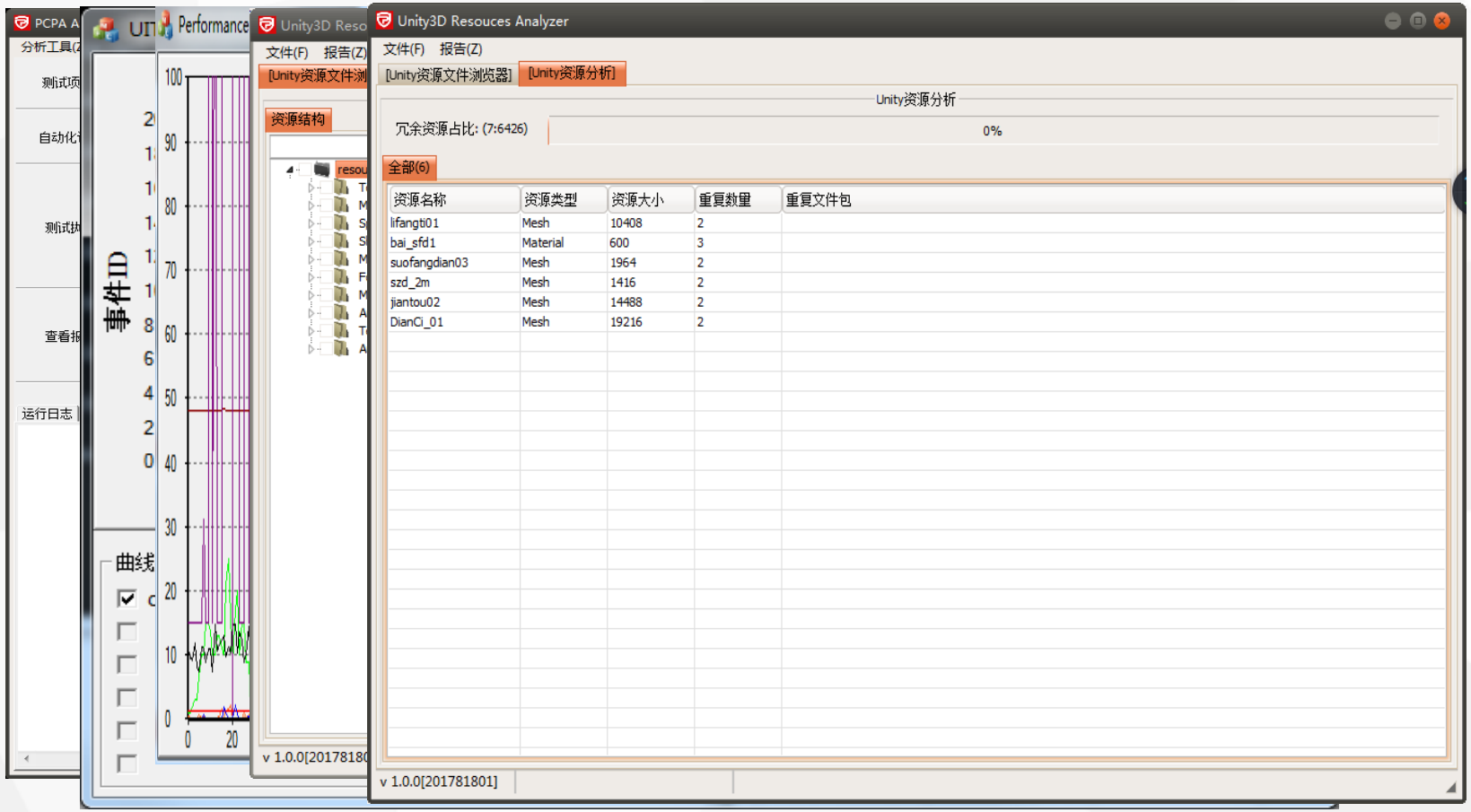




1、对于QA团队，无论是研发、测试还是上线阶段，都可以针对性能进行及时监控；
2、对于技术团队，性能问题及时发现、及时修复，具体问题具体解决；
3、对于策划团队，性能问题背后的设计需求可及时调整和变更。

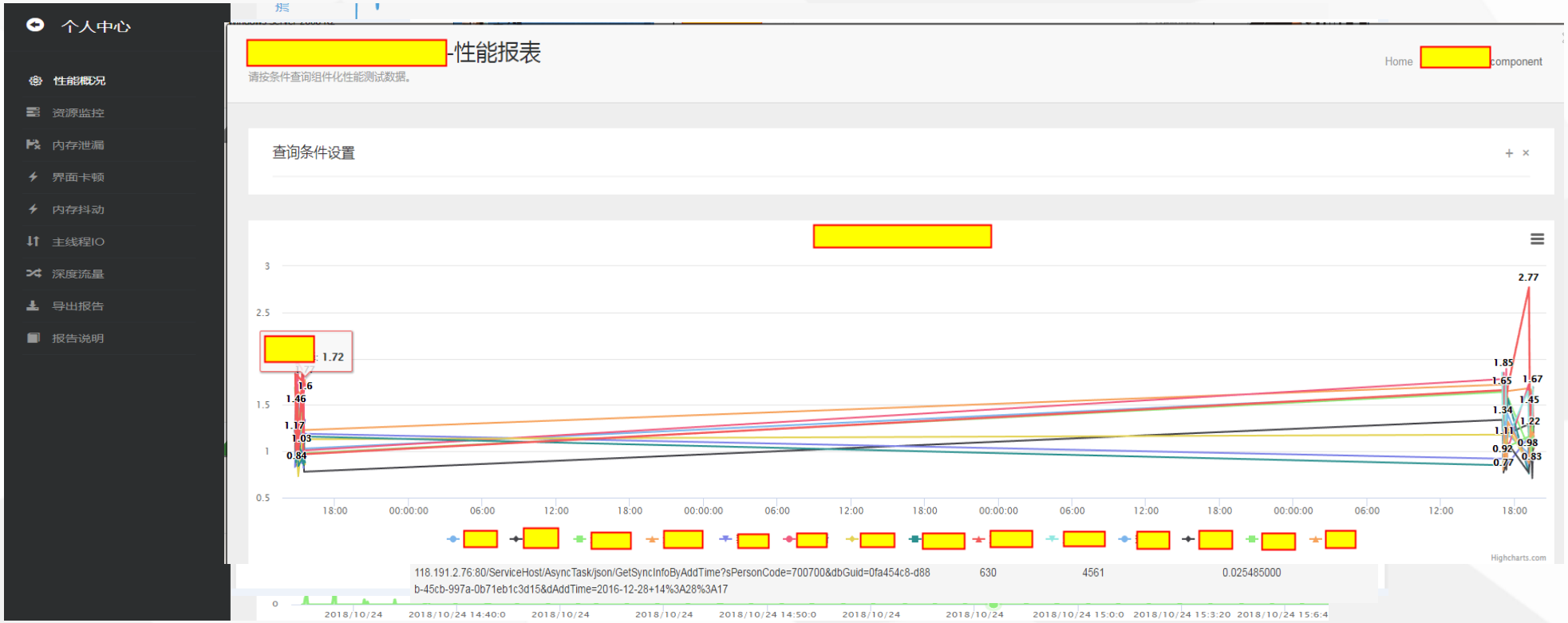
平台展示

网龙网络公司
NETDRAGON WEBSOFT INC.



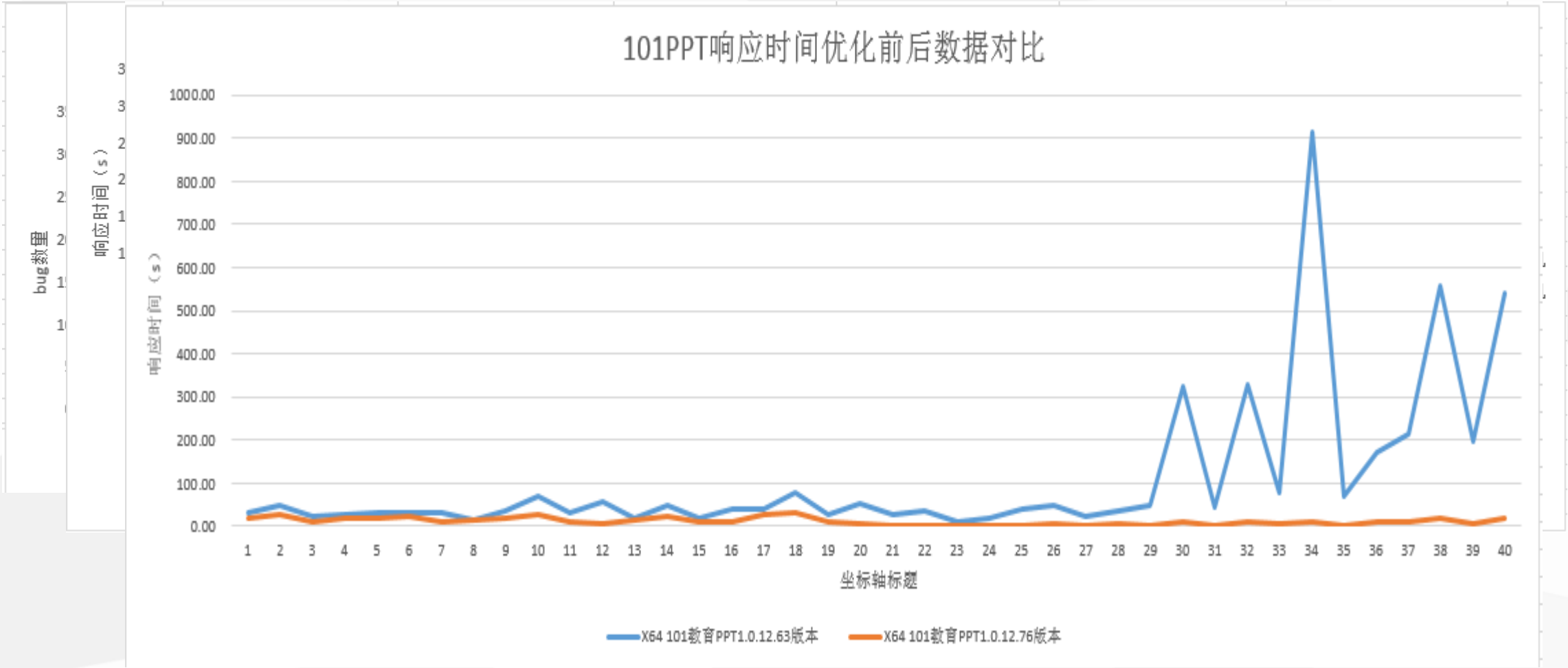
平台展示

网龙网络公司
NETDRAGON WEBSOFT INC.



成效

网龙网络公司
NETDRAGON WEBSOFT INC.



专利

网龙网络公司
NETDRAGON WEBSOFT INC.

一种基于神经元模型的自动遍历控件方法及终端 【公开】

申请号：CN201710697128

申请日：2017.08.15

公开（公告）号：CN107562619A

公开（公告）日：2018.01.09

IPC分类号：G06F11/36；G06N3/06；

申请（专利权）人：福建天晴数码有限公司；

一种测量响应时间的方法及终端 【公开】

申请号：CN201710815643

申请日：2017.09.12

公开（公告）号：CN107797904A

公开（公告）日：2018.03.13

IPC分类号：G06F11/34；

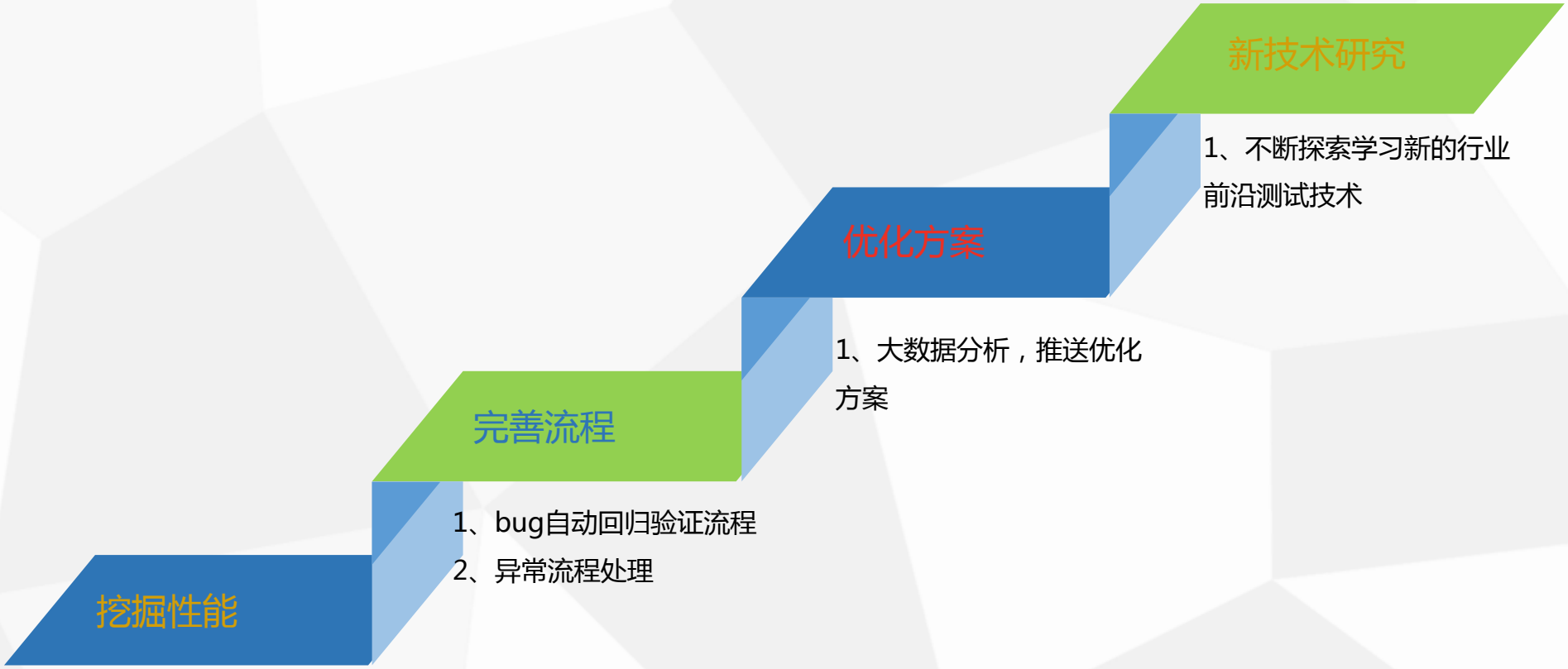
申请（专利权）人：福建天晴数码有限公司；

网龙网络公司
NETDRAGON WEBSOFT INC.

04 总结

持续研发

网龙网络公司
NETDRAGON WEBSOFT INC.



性能之路漫漫而修远兮，吾们将上下而求索

网龙网络公司
NETDRAGON WEBSOFT INC.

感谢聆听

网龙网络公司
NETDRAGON WEBSOFT INC.