

手机卫士性能演进历程



- 功能堆积—产品经理无尽需求
- 代码混乱—埋坑、填坑
- 论坛轰炸—内存大、卡、慢
- 响应速度—各种工具信息收集，日志包

- 防患于未然—预防性能问题
- 了然于胸—实时性能监控
- 深入群众—性能问题诊断
- 及时响应—动态热修复技术

代码检测工具

自动化分析

资源优化工具

APK构建检测

LINT

```
@Override
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    Point myPoint = new Point();
}
```

Correctness

Usability

Security

Accessibility

Name

MyView.java

Location

package com.example.jigang.myapplicationfile [MyView.java](#)

Problem synopsis

Avoid object allocations during draw/layout operations (preallocate and reuse instead) (at line [28](#))

FINDBUGS

警告类型

NP

OS

LSYC

AUC

SPP

FCBL

QIHOO_RL

RCN

```
Cursor cursor = ContextHelper.getMainContext().getContentResolver().query(
try {
    if (cursor != null && cursor.moveToFirst()) {
        do {
            } while (cursor.moveToNext());
        }
    } catch (Exception e) {
        LogUtils.logE(e);
    } finally {
        IoStreamUtils.closeQuietly(cursor); 原生findbugs认为有问题
    }
}
```

```
if (cursor != null) {
    try {
        cursor.close();
    } catch (Throwable e) {
        if (DEBUG) {
            Log.e(TAG, e.getMessage(), e);
        }
    }
}
```

代码红线和安全检测

无限团队开发过程中必须遵守的准则和不能碰触的红线

UI

UI线程的耗时操作

频繁查找资源

频繁创建对象

过度刷新

Adapter复用

布局扁平

直接使用大图片

Perf

Dex加载

解压

数据延迟加载

后台任务调度

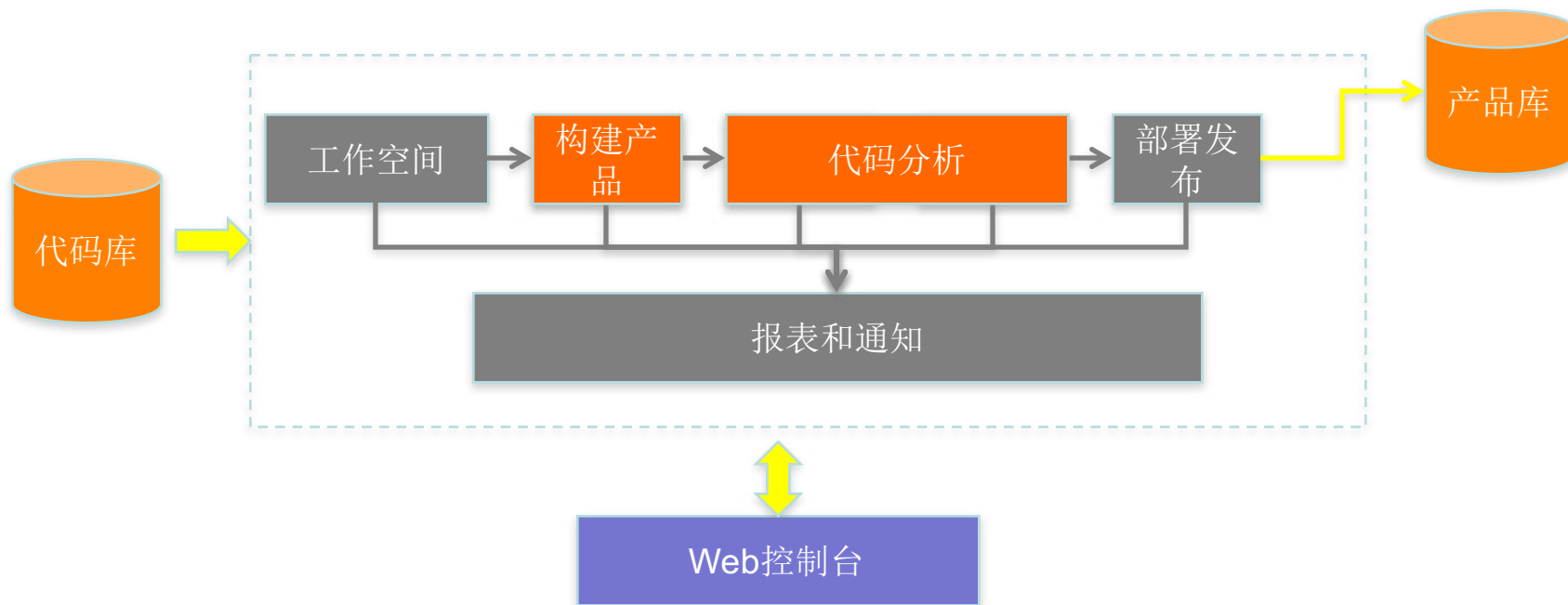
Webview

JENKINS平台

一个开源的持续集成平台

构建产品

定时任务



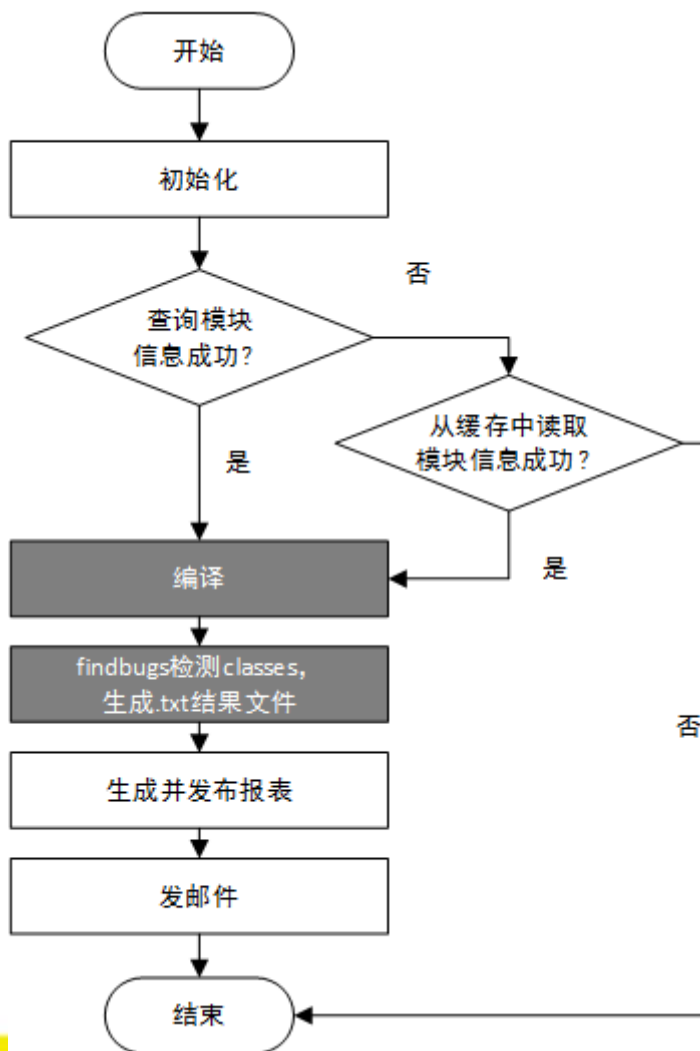
设计实现



已经实现

- Crash 报告自动分析
- Crash 问题的预警
- 代码静态检测
- 生成数据报表
-

举例



输出

Hi,

请检查下面的 **findBugs** 问题:

Time\Type	NP	OS	
2016/03/14	0	1	

DETAIL: <http://xxx.xxx.xxx/view/CodeStaticCheck/job/AutoFindbugs/205/label=>
 PS: 没有账号的可以通过公共账号登录: 用户名 xxx, 密码 xxx

警告类型	类型说明
NP	可能存在空指针
OS	IO流可能未能正常关闭
LSYC	可能在局部变量的中, 不必要的使用线程安全类, 如StringBuffer等
AUC	CURSOR 可能没有正常关闭
SPP	在使用"instanceof"之前使用"==null"判空, 没必要, instanceof本身就会判空
FCBL	定义了域, 实际只在一个方法里面使用, 请将该域对应的变量替换成方法里面的局部变量
QIHOO_RL	文件流或者指针关闭不正确
RCN	1. Nullcheck of... 表示在判断变量为空前, 使用了该变量 2. Redundant nullcheck of... 表示判空是多余的

2016/2/13 2016/2/14 2016/2/15 2016/2/16 2016/2/17 2016/2/18 2016/2/19 2016/2/20 2016/2/21

Strict Mode

Detect All

构建StrictMode的Apk

UI遍历自动化测试

自动化分析

基准对比

发布结果

increment	view	view	build	build	build	build
increment	logs	specific	debug	findbugs	strict debug	release

字段名称	字段说明
StrictModeNetworkViolation	问题详情里面的代码在UI线程访问了网络
StrictModeDiskReadViolation	问题详情里面的代码在UI线程读了磁盘
StrictModeDiskWriteViolation	问题详情里面的代码在UI线程写了磁盘
StrictModeCustomViolation	问题详情里面的代码位置所在的方法所花时间太长
A resource was acquired at attached stack trace but never released	资源未释放，可能是数据库cursor、文件指针等资源未释放。
Are you missing a call instances	注册了监听器或者serviceconnection在context消失的时候没有释放
	Activity实例被其他对象持有，在Activity实例应该消失的地方例如调整屏幕的时候该实例没有消失 or 超过了StrictMode里面设置的最大实例数

重复图片工具

图片压缩工具

资源混淆工具

核心法宝：制度

强大的构建分析

build	build	build	build	build	包大小 方法数	包大小查看 (debug)	包大小查看 (release)	查看 FindBugs
debug	findbugs	strict debug	release	status	detail	detail	detail	detail

版本,编译号	日期,时间	修订	包大小
6.5.1.1032	2016-03-14 20:02:46	1558442	10154328 -779 ↓
6.5.1.1031	2016-03-14 19:28:54	1558256	10155107 -4835 ↓
6.5.1.1031	2016-03-14 15:49:33	1557914	10159942 +60267 ↑

字符串数	类型数	原型数	字段数
18302	4089	5231	10019
18302 -42 ↓	4089	5231	10019 -44 ↓
18344 -1049 ↓	4089 -265 ↓	5231 -296 ↓	10063 -1567 ↓

方法数	类数	字节码大小
25163	3275	2631788
25163 +6 ↑	3275	2631788 -1020 ↓
25157 -1382 ↓	3275 -235 ↓	2632808 -141504 ↓

静态埋点

性能监控

观察者App

性能分析

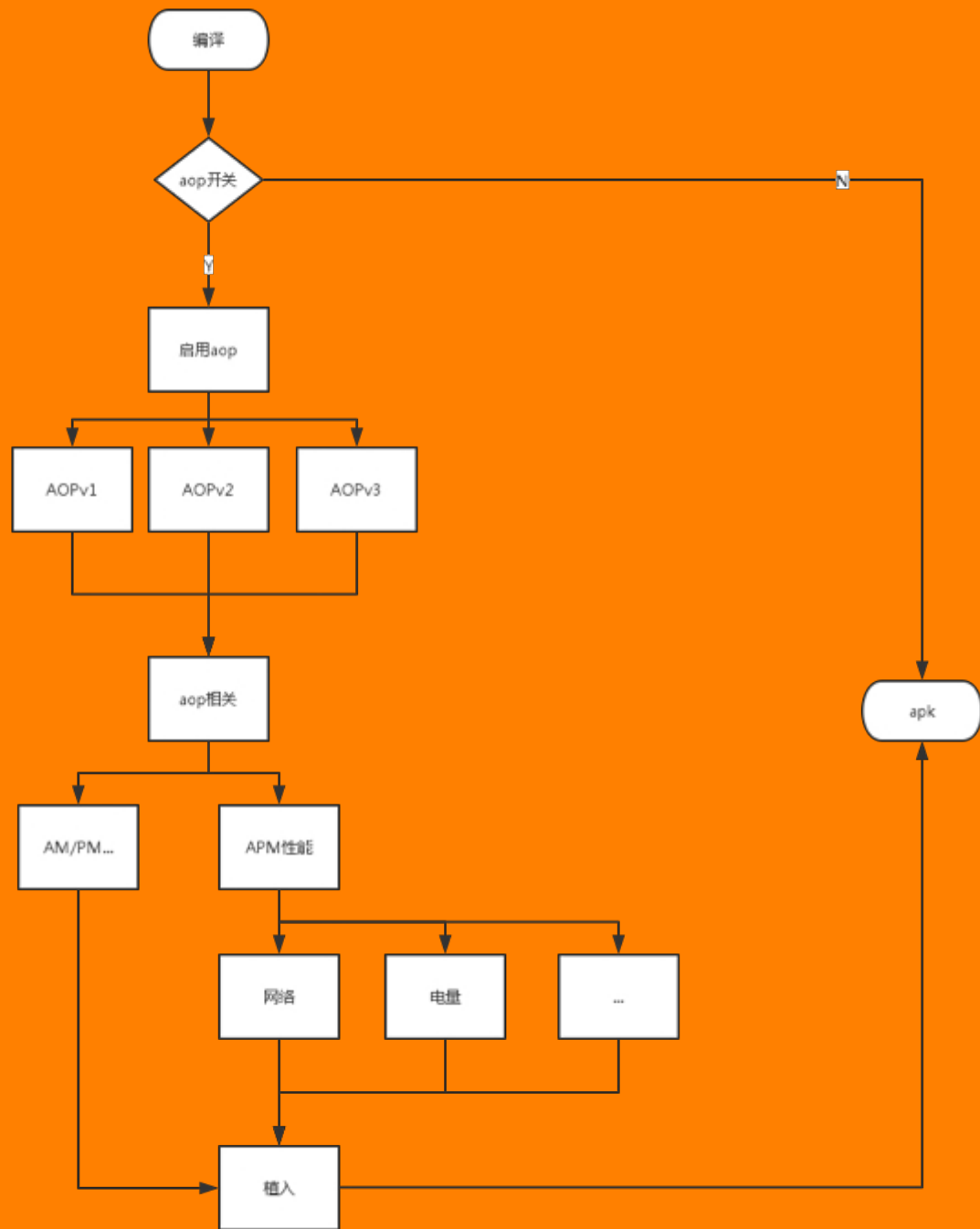
介绍及应用

基于java的面

静态埋点：编译时
文件中完成遥测

电量

流量



举例—网络请求静态埋点

对每一次网络请求数据进行记录，提供可评估的数据指标。

拦截对象：

org.apache.http.HttpHost
org.apache.http.HttpResponses
org.apache.http.HttpRequest
org.apache.http.client.ResponseHandler
org.apache.http.client.methods.HttpUriRequest
org.apache.http.protocol.HttpContext
java.net.URLConnection
org.apache.http.client.HttpClient

```
aspect TraceNetTrafficMonitor {  
  
    pointcut baseCondition():  
        !within(com.qihoo360.mobilesafe.aop.*) && !within(c.q.apm.*) && !within(com.qihoo360.mobilesafe.mms.transaction.MmsHttpClient) &&  
        !target(com.qihoo360.mobilesafe.mms.transaction.MmsHttpClient);  
  
    pointcut httpClientExecuteOne(HttpClient httpClient, HttpUriRequest request) :  
        call(HttpResponse HttpClient.execute(HttpUriRequest)) && target(httpClient) && args(request) && baseCondition();  
  
    HttpResponse around(HttpClient httpClient, HttpUriRequest request) throws IOException, ClientProtocolException : httpClientExecuteOne(  
        httpClient, request) {  
        return QHC.execute(httpClient, request);  
    }  
}
```


举例—网络请求静态埋点

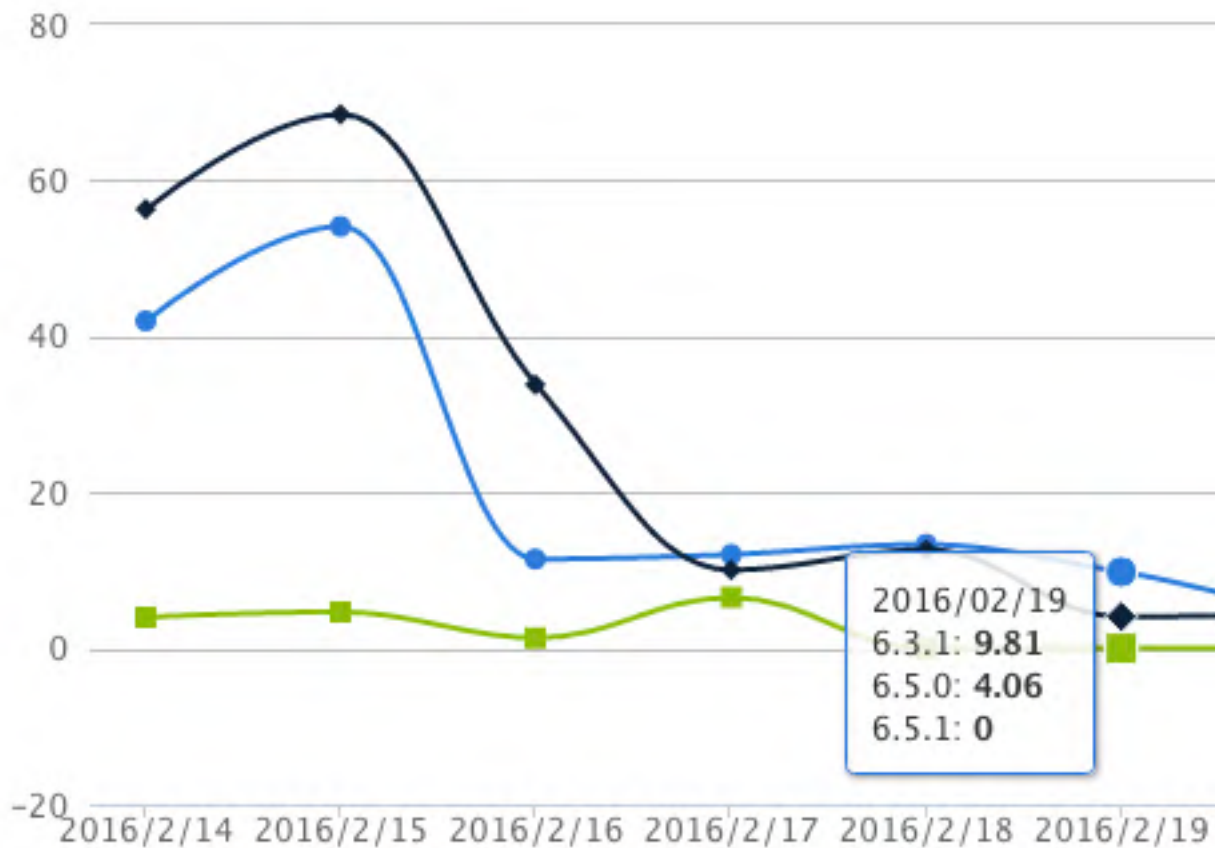
```

UsrManager : ===== 2G/
-----> IMEI :XXXXXXXXXX
UpLoad :18294K DownLoa

-----> IMEI : XXXXXXXXXXXX
UpLoad :33770K DownLoa
14K http://msoftdl.360.cn/

=====
-----> IMEI : XXXXXXXXXXXX
UpLoad :169465K DownLo
    
```

每人总请求网络次数



进程冷热启动

四大组件启动速度

各个线程卡慢监测

异步处理性能监测

进程冷热启动

关键字:

Context::startXXXX

Context::bindXXXX

Application::attach

Application::onCreate

Activity::onCreate

CustomInstrumentation::callApplicationOnCreate

service: exported?

activity: exported? UI?

provider: exported?

receiver: exported?

举例—Activity的加载时间

关键字:

ActivityThread

Instrumentation

CustomInstrumentation

```
public void callActivityOnCreate(Activity activity, Bundle icle) {  
    //...  
    //收集我们需要的activity信息。  
    long bTim = System.currentTimeMillis();  
    rTRunnable.setBeginTime(bTim, activity.getComponentName().getClassName());  
    super.callActivityOnCreate(activity, icle);  
    //activity绘制完成后, 执行回调函数, 用来记录时间  
    activity.getWindow().getDecorView().post(rTRunnable);  
    //...  
}
```

className	timeType	time	day	flag
com.qihoo360.mobilesafe.ui.index.AppEnterActivity	1	375	16794	0
com.qihoo360.mobilesafe.paysafe.nicaifu.NicaifuMain	1	373	16794	0
com.qihoo360.mobilesafe.block.ui.BlockRecordScreen	1	122	16794	0
com.qihoo360.mobilesafe.block.ui.BlockRecordScreen	1	252	16794	0
com.qihoo360.mobilesafe.main.ui.AppManagerActivity	1	150	16794	0

采样时间发送检测消息

Handle::post
Handle::dispatchMessage

CustomInstrumentation::waitForIdle
CustomInstrumentation:: waitForIdleSync

```
public void CustomInstrumentation:: waitForIdle() {  
    //...  
    mMessageQueue.addIdleHandler(new Idler(recipient));  
    mThread.getHandler().post(new EmptyRunnable(t));  
    //...  
}
```

异步处理静态埋点

拦截对象：

Handle::post(Runnable r)

Handle::postAtTime(Runnable r, long uptimeMillis)

Handle::postAtTime(Runnable r, Object token, long uptimeMillis)

Handle::postDelayed(Runnable r, long delayMillis)

Handle::postAtFrontOfQueue(Runnable r)

关键点：

- 1、加入上下文32位整数；
- 2、Runnable对象执行后，计算时间值；

- **Java反射**

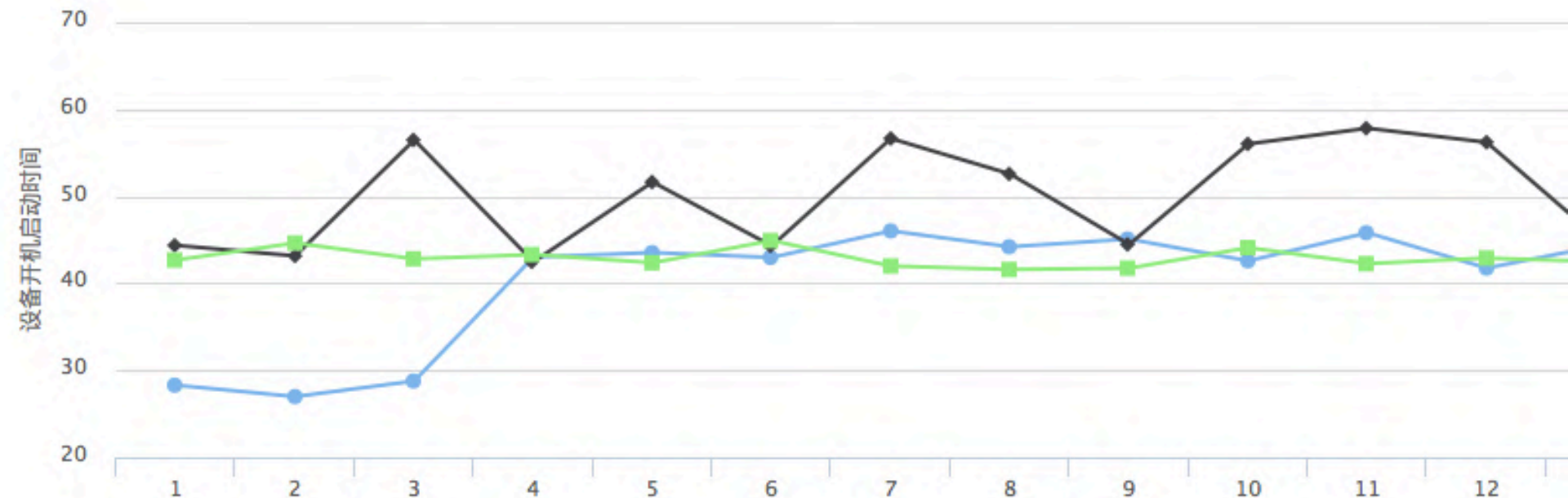
- **主线程卡慢 – 四大组件的OnCreate等方法**

- **ClassLoader.getResourceAsStream**

- **第三方SDK**

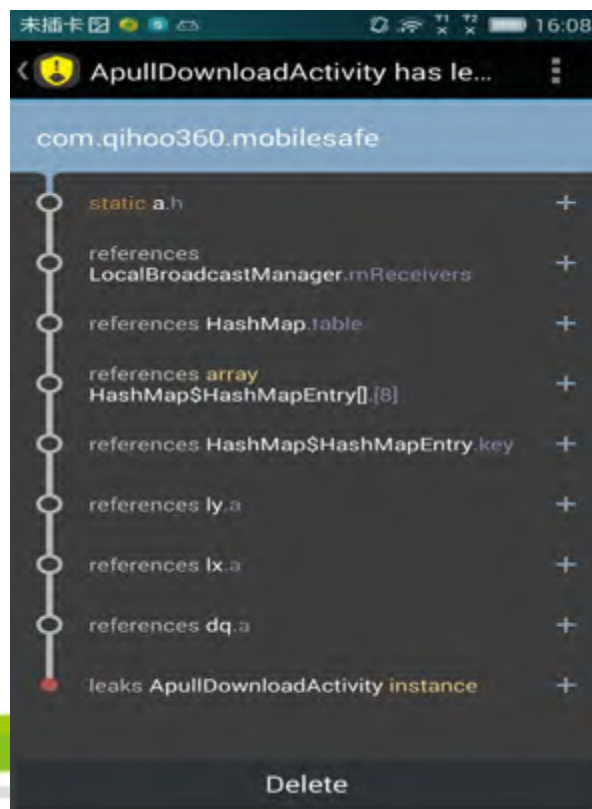
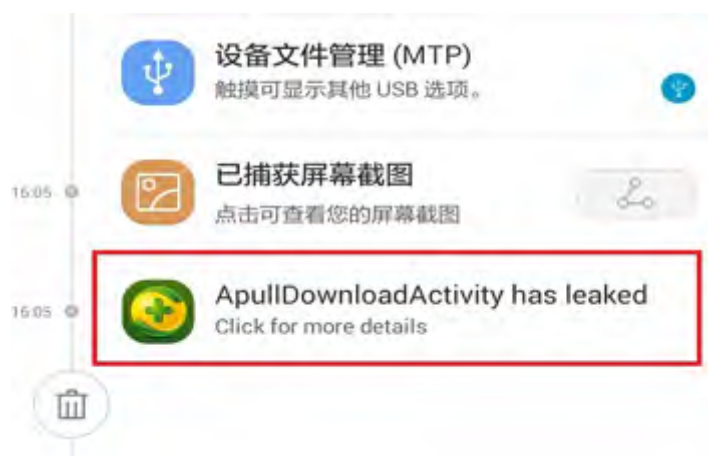
Stalker

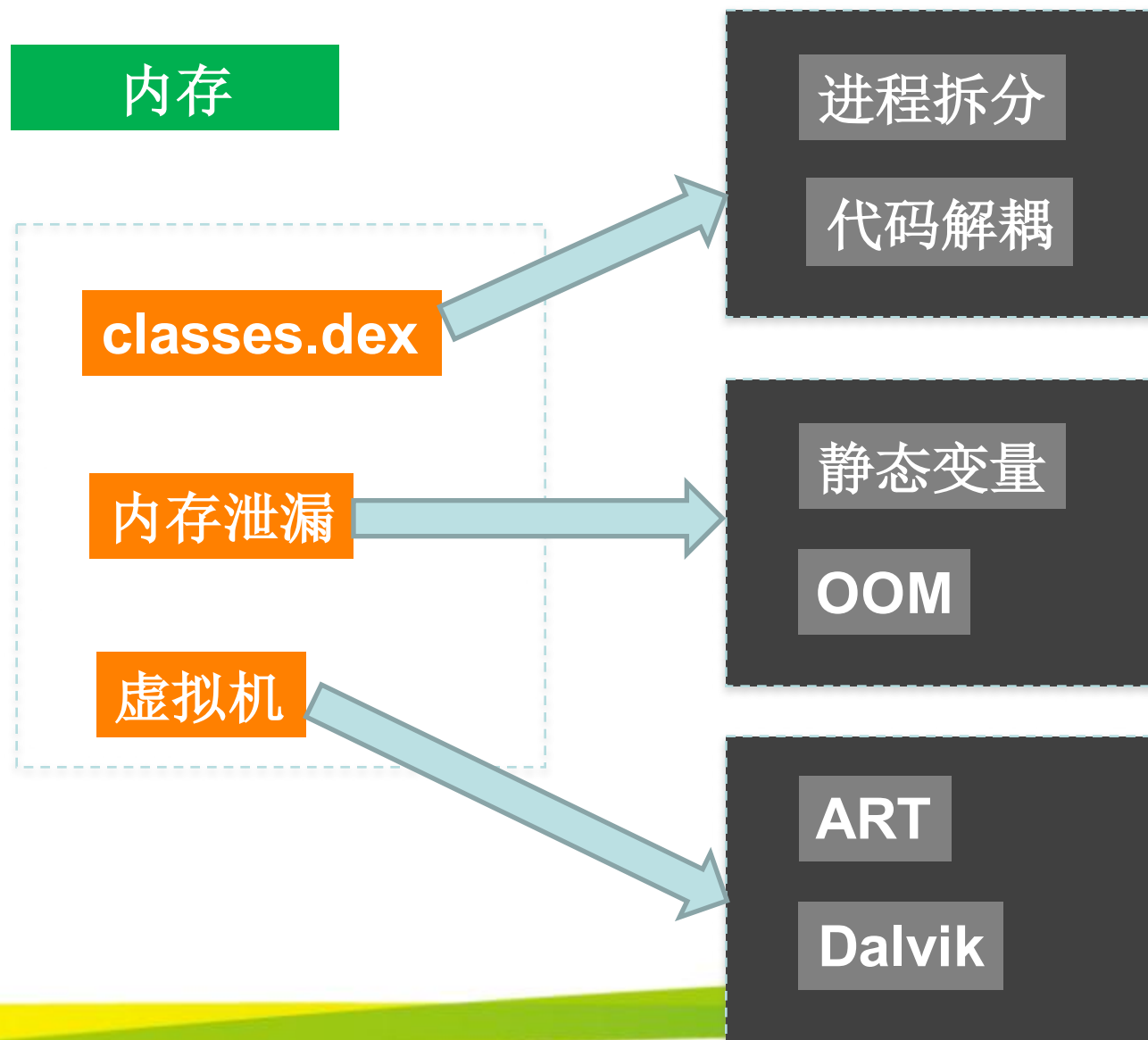
- 监控应用的多个进程
- 进程的cpu、内存、电量、流量信息
- 系统全局的cpu、内存、电量信息
- 界面的帧率信息



LeakCanary

检查Activity是否泄漏的问题，通过onDestroy方法检查，存在泄漏，则会dump内存镜像，获取栈信息





分析方法

Dalvik H

adb shell kill

Debug.dump

DDMS dump

hprof-conv

基本信息									
HPROF文件	对象数量	GC Roots数量	Class数量	ClassLoader数量	已用Dalvik堆内存 (byte)	卫士对象数量	卫士对象内存 (byte)	可回收对象数量	可回收内存 (byte)
init.hprof	70608	14623	4941	10	18039568	861	1253680	5676	208424
jiaozhun.hprof	74902(+4294)	15814(+1191)	5243(+302)	11(+1)	18286448(+246880)	988(+127)	1338840(+85160)	887(-4789)	49040(-159384)

- * 对象数量是指在该Hprof文件中产生的对象总数，代表了在dump hprof文件时该进程中的Java层对象总数，同理GC Roots数量、Class数量和ClassLoader数量
- * 已用Dalvik堆内存代表该进程在dump hprof时占用的Dalvik堆内存大小，已用Dalvik堆内存 = 卫士程序占用内存 + Android系统对象占用内存 + 可回收内存
- * 卫士对象数量和卫士对象内存代表由卫士程序定义的类而产生的对象数量和这些对象占用的Dalvik堆内存大小
- * 可回收对象数量和可回收对象内存代表在dump hprof时，该进程Dalvik堆内存中可以被回收的对象总数和内存大小

* 下表展示了Hprof文件中卫士程序自定义的类及其对象的内存占用情况

名称	init.hprof			jiaozhun.hprof		
	对象数量	自占内存 (byte)	持有内存 (byte)	对象数量	自占内存 (byte)	持有内存 (byte)
com.qihoo360.mobilesafe.notification.NotificationServiceHolder	1	48	896720	1	48	896720
com.qihoo360.mobilesafe.paysafe.modules.AppVerify	1	24	70808	1	24	70808(+208)
com.qihoo360.mobilesafe.paysafe.db.AppVerifyDatabase	1	16	70336	1	16	70336
com.qihoo360.mobilesafe.paysafe.db.AppVerifyDatabase\$PayRule	250	8000	64192	250	8000	64192
com.qihoo360.mobilesafe.block.data.YellowNumberManager	0	0	59552	0	0	59552
com.qihoo360.mobilesafe.share.NumberManager	0	0	42704	0	0	42856(+152)
com.qihoo.antivirus.shield.domain.ShieldDisguiseLocationInfo	20	1120	37136	20	1120	37136
com.qihoo360.common.utils.IniProperties	1	16	8448	2(+1)	32(+16)	27880(+19432)
com.qihoo360.loader2.Plugin	14	560	25960	14	560	27656(+1696)
com.qihoo.antivirus.shield.domain.PackageItem	79	9480	12040	79	9480	12336(+296)
com.qihoo360.mobilesafe.block.provider.SafeGuardProvider	1	48	10328	1	48	10328
com.qihoo360.mobilesafe.paysafe.payrecord.TriggerFactory	1	24	9048	1	24	9048
com.qihoo.antivirus.autostart.AutoStartedPackageItem	63	3024	8600	63	3024	8600
com.qihoo360.mobilesafe.paysafe.payrecord.EvidenceTriggerHelper	1	24	8560	1	24	8560
com.qihoo360.mobilesafe.utils2.OptConfig	1	16	8464	1	16	8464
android.support.v4.os.ParcelableCompat	1	8	7000	1	8	7000
com.qihoo360.mobilesafe.paysafe.modules.NetProtectDataManager	1	32	4240	1	32	4240
com.qihoo.antivirus.shield.domain.ShieldDisguisePhoneMode	18	432	4080	18	432	4080
com.qihoo360.loader2.PluginProcessMain	0	0	3544	0	0	3600(+56)
com.qihoo360.mobilesafe.report.ReportConfig	0	0	3264	0	0	3264
com.qihoo360.loader2.PluginManager	1	80	2840	1	80	2840
com.qihoo360.mobilesafe.loaded.AppJar\$FilterInstance	1	96	2680	1	96	2680
com.qihoo360.mobilesafe.ipc.pref.IpcPrefManagerImpl	1	8	2464	1	8	2464
com.qihoo360.loader2.PluginProcessMain\$ProcessPluginInfo	5	120	2208	5	120	2288(+80)
com.qihoo360.mobilesafe.block.i.BlockFactory	0	0	232	0	0	2216(+1984)
com.qihoo360.mobilesafe.loaded.client.in	0	0	2176	0	0	2176
android.support.v4.content.Loader\$OnLoadCompleteListener	1	24	752	1	24	2096(+1344)

分析方法

meminfo

mans

smans

mem

```

5badb000-5badc000 rw-s 00000000 b3:07 49846 /data/data/com.qihoo360.mobilesafe/shared_prefs/ipc_pref/pref_internal.shmflag
5badc000-5bae6000 r-xp 00000000 b3:07 57640 /data/data/com.qihoo360.mobilesafe/files/libmsb.so.4x.so (deleted)
5bae6000-5bae7000 rw-p 0000a000 b3:07 57640 /data/data/com.qihoo360.mobilesafe/files/libmsb.so.4x.so (deleted)
5bae7000-5bae8000 rw-p 00000000 00:00 0
5bae8000-5bae9000 ---p 00000000 00:00 0
5bae9000-5bbe8000 rw-p 00000000 00:00 0 [stack:23231]
5bbe8000-5bbf8000 rw-p 00000000 00:00 0
5bbf8000-5bbf9000 ---p 00000000 00:00 0
5bbf9000-5bcf8000 rw-p 00000000 00:00 0 [stack:23232]
5bcf8000-5bd18000 rw-p 00000000 00:00 0
5bd18000-5bd25000 r--s 007fa000 b3:07 32824 /data/app/com.qihoo360.mobilesafe-1.apk
5bd25000-5bd32000 r--s 007fa000 b3:07 32824 /data/app/com.qihoo360.mobilesafe-1.apk
  
```

	meminfo	mans	smans	mem	total
Ashmem	0	0	0		
Other dev	16	44	0		
.so mmap	1240	2736	940		
.jar mmap	4	0	0		
.apk mmap	71	0	0		
.ttf mmap	0	0	0		
.dex mmap	1652	0	24		
Other mmap	1129	336	244		
Unknown	2903	584	2896		
TOTAL	14443	9452	11436	11876	8441 3214

卫士 dex 名称	1058.init.smaps		1058.jiaozhun.smaps	
	PSS (kB)	RSS (kB)	PSS (kB)	RSS (kB)
/data/dalvik-cache/data@app@com.qihoo360.mobilesafe-1.apk@classes.dex	1680	3612	2400(+720)	2400(-1212)
/data/data/com.qihoo360.mobilesafe/app_plugins_v3_odex/shield-10-10-0.dex	396	396	408(+12)	408(+12)
/data/data/com.qihoo360.mobilesafe/app_plugins_v3_odex/root-10-10-0.dex	112	112	112	112
/data/dalvik-cache/data@data@com.qihoo360.mobilesafe@files@plugins_v3_data@shield@app_gabage@cache002.jar@classes.dex	33	216	35(+2)	224(+8)
/data/data/com.qihoo360.mobilesafe/files/com.qihoo360.mobilesafe_GuardService1113831752_apm.dex	32	32	32	32
/data/dalvik-cache/data@data@com.qihoo360.mobilesafe@files@plugins_v3_data@shield@app_gabage@cache001.jar@classes.dex	1	136	1	136
/data/data/com.qihoo360.mobilesafe/app_plugins_v3_odex/ntsvc-10-10-0.dex			320	320
Total	2254 percent:14.24%	4504 percent:6.01%	3308(+1054) percent:14.90%	3632(-872) percent:4.41%
卫士 so 名称	1058.init.smaps		1058.jiaozhun.smaps	
	PSS (kB)	RSS (kB)	PSS (kB)	RSS (kB)
/data/app-lib/com.qihoo360.mobilesafe-1/libmobilesafe360-jni-600.5.so	132	372	232(+100)	232(-140)
/data/app-lib/com.qihoo360.mobilesafe-1/libipc_pref.600.7.so	104	356	188(+84)	188(-168)
/data/app-lib/com.qihoo360.mobilesafe-1/libbreakpad-jni-1.3.so	40	112	64(+24)	64(-48)
/data/data/com.qihoo360.mobilesafe/files/plugins_v3_data/shield/files/libavb.so.4x.so	8	96	8	96
/data/data/com.qihoo360.mobilesafe/files/plugins_v3_data/shield/files/com.qihoo360.mobilesafe/libb.so.4x.so	4	12	4	12
Total	288 percent:1.82%	948 percent:1.27%	496(+208) percent:2.23%	592(-356) percent:0.72%
卫士 jar 名称	1058.init.smaps		1058.jiaozhun.smaps	
	PSS (kB)	RSS (kB)	PSS (kB)	RSS (kB)
/data/data/com.qihoo360.mobilesafe/app_plugins_v3/shield-10-10-0.jar	20	44	20	44
/data/data/com.qihoo360.mobilesafe/app_plugins_v3/root-10-10-0.jar	4	16	4	16
/data/data/com.qihoo360.mobilesafe/files/com.qihoo360.mobilesafe_GuardService1113831752_apm.jar	4	4	4	4
/data/data/com.qihoo360.mobilesafe/files/plugins_v3_data/shield/app_gabage/cache002.jar	0	8	0	8
/data/data/com.qihoo360.mobilesafe/files/plugins_v3_data/shield/app_gabage/cache001.jar	0	8	0	8
/data/data/com.qihoo360.mobilesafe/app_plugins_v3/ntsvc-10-10-0.jar			8	20
Total	28 percent:0.18%	80 percent:0.11%	36(+8) percent:0.16%	100(+20) percent:0.12%

- 静态Activity

- 内部类

- 匿名类(Runnable、AsyncTask)

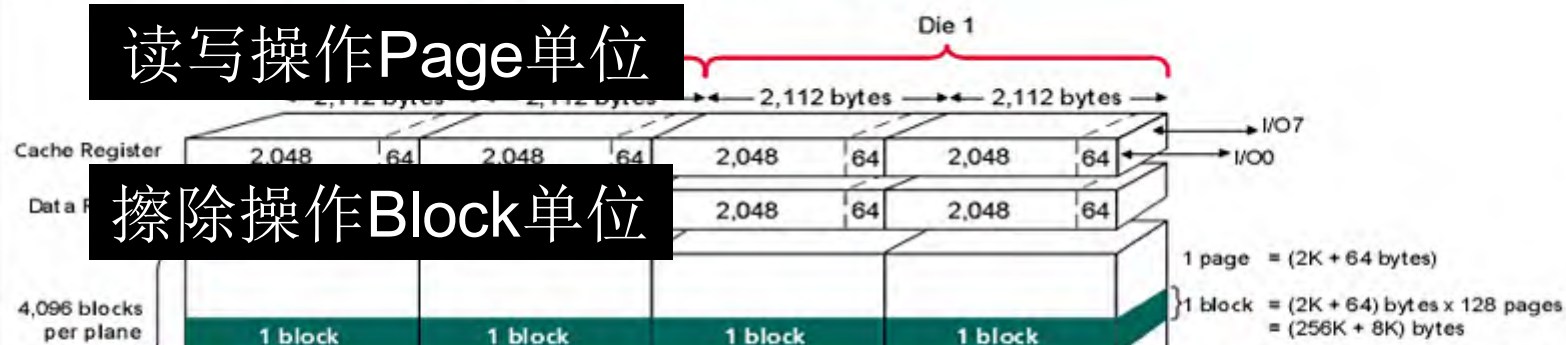
- 静态View

- Sensor Manager

磁盘对于性能的影响

NAND Flash

Figure 7: Array Organization for MT29F32G08TAA (x8)



具体规则:

- ✓ 删除数据时，芯片将标记这些 **Page** 为闲置状态，但并不会立马执行擦除操作；
- ✓ 写入数据时，如果目前磁盘剩余空间充足，则由芯片指定 **Block** 后直接按 **Page** 为单位进行写入即可；
- ✓ 写入数据时，如果目前磁盘剩余空间不足，为了获得足够的空间，磁盘先将某块 **Block** 的内容读至缓存，然后再在该 **Block** 上进行擦除操作，最后将新内容与原先内容一起写入至该 **Block**；

3. Die 1, Plane 0: BA19 = 1; BA7 = 0

4. Die 1, Plane 1: BA19 = 1; BA7 = 1

5. For the 32Gb MT29F32G08TAA, the 16Gb array organization shown applies to each chip enable (CE# and CE2#).

手机诊断检测用户设备性能

基本信息

双卡信息

ROOT信息

网络信息

自动分析耗电

分析屏幕密度

生成内存镜像文件

Method Tracing

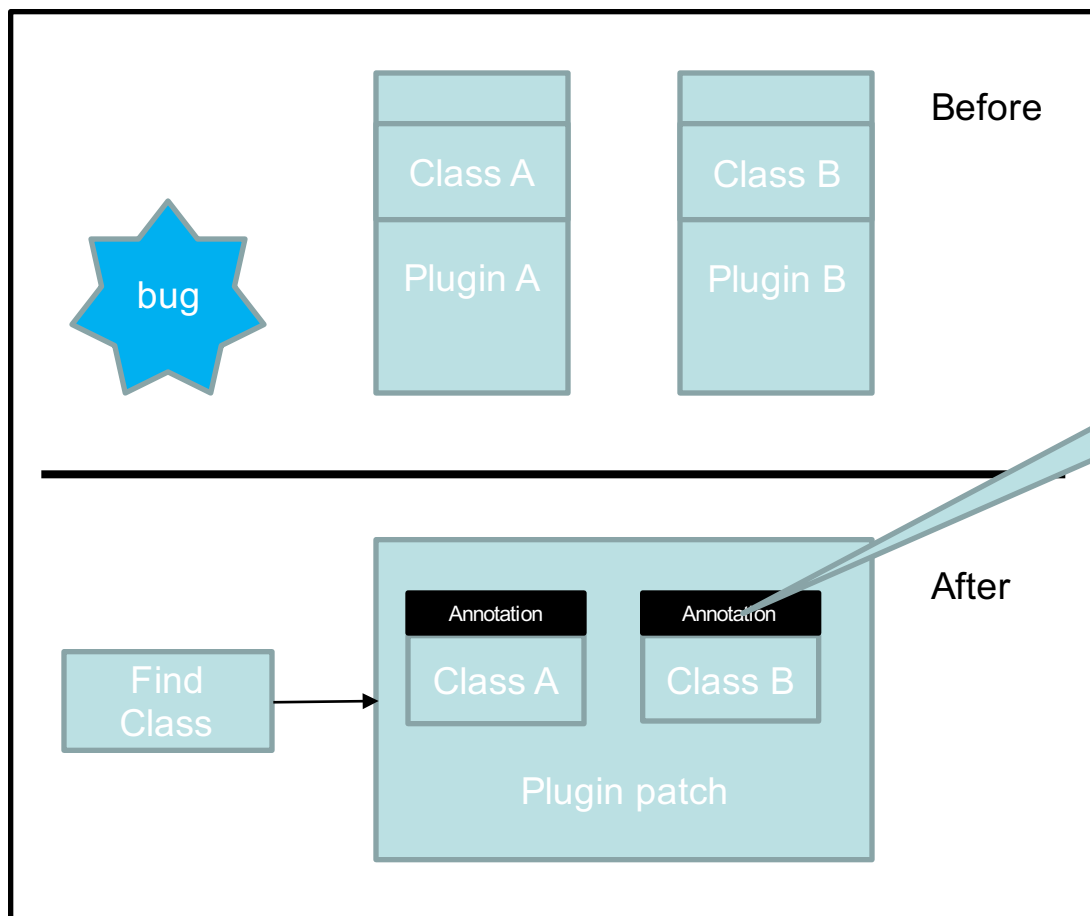
CPU/内存监控

.....

手机诊断检测用户设备性能 && 利用push自动获取




动态补丁技术



- 1、插件版本号
- 2、插件名
- 3、类名

欢迎交流



刘刚 

北京 石景山



扫一扫上面的二维码图案，加我微信

谢谢！

北京奇虎科技有限公司

