

崩溃

腾讯 victorhuang



用户遇到的三个可感知的专项问题

卡慢

崩溃

发热



稳定性问题的重要性？

难点与挑战

发现

定位

解决

度量

重点讲，然后案例

大家认为最难是哪个部分？



什么样的crash难发现？

通过crash的分类来想办法，发现问题

1. 发起HTTP请求，返回内容为NULL，导致空指针错误

2. 读取磁盘的一张超大的照片，导致OOM

3. 网络拥塞，导致依赖网络回包才初始化的变量未被初始化就被使用，导致空指针错误

4. 多次打开关闭聊天窗口，界面对象重复被生成且没释放，最终导致OOM

5. 调用系统接口获取GPS定位信息，结果返回为空，导致空指针错误

案例

 【394359595】 【49次】 【6.3.0.338】 【RDM异常上报】 【java.lang.IllegalStateException】 ID: 52431631 状态:

详细... 需求 (0) 代码评审 (0) SVN提交 (3) 变更历史 (9) 更多

Crash编号: 394359595([查看RDM异常详情](#))

Crash编号: 394359595

软件版本: 6.3.0.338

产品名称: com.tencent.mobileqq

错误级别: 严重错误

错误类型: java.lang.IllegalStateException

Crash详情:

java.lang.IllegalStateException: Cannot execute task: the task is already running.

android.os.AsyncTask.executeOnExecutor(AsyncTask.java:576)

android.os.AsyncTask.execute(AsyncTask.java:535)

com.tencent.mobileqq.activity.aio.anim.VipPendantDrawable\$2.decodeBigImage0(ProGuard:877)

com.tencent.mobileqq.activity.aio.anim.VipPendantDrawable\$2.setBigImageDrawable(com.tencent.mobileqq.activity.aio.anim.VipPendantDrawable\$2:126)

com.tencent.mobileqq.activity.aio.anim.VipPendantDrawable\$2.run0(ProGuard:366)

com.tencent.mobileqq.app.Job.run0(ProGuard:126)

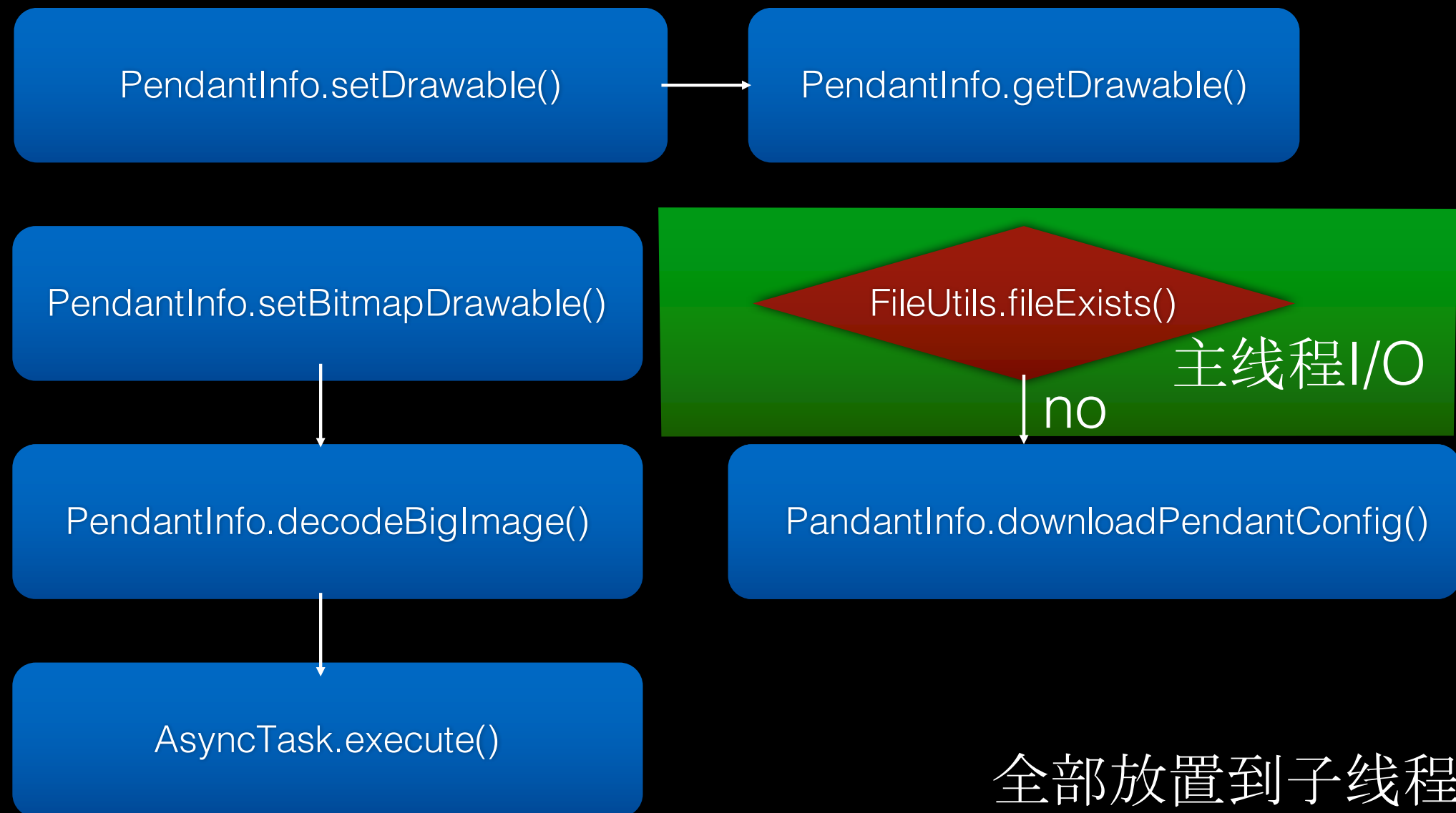
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1112)

java.util.concurrent.ThreadPoolExecutor\$Worker.run(ThreadPoolExecutor.java:587)

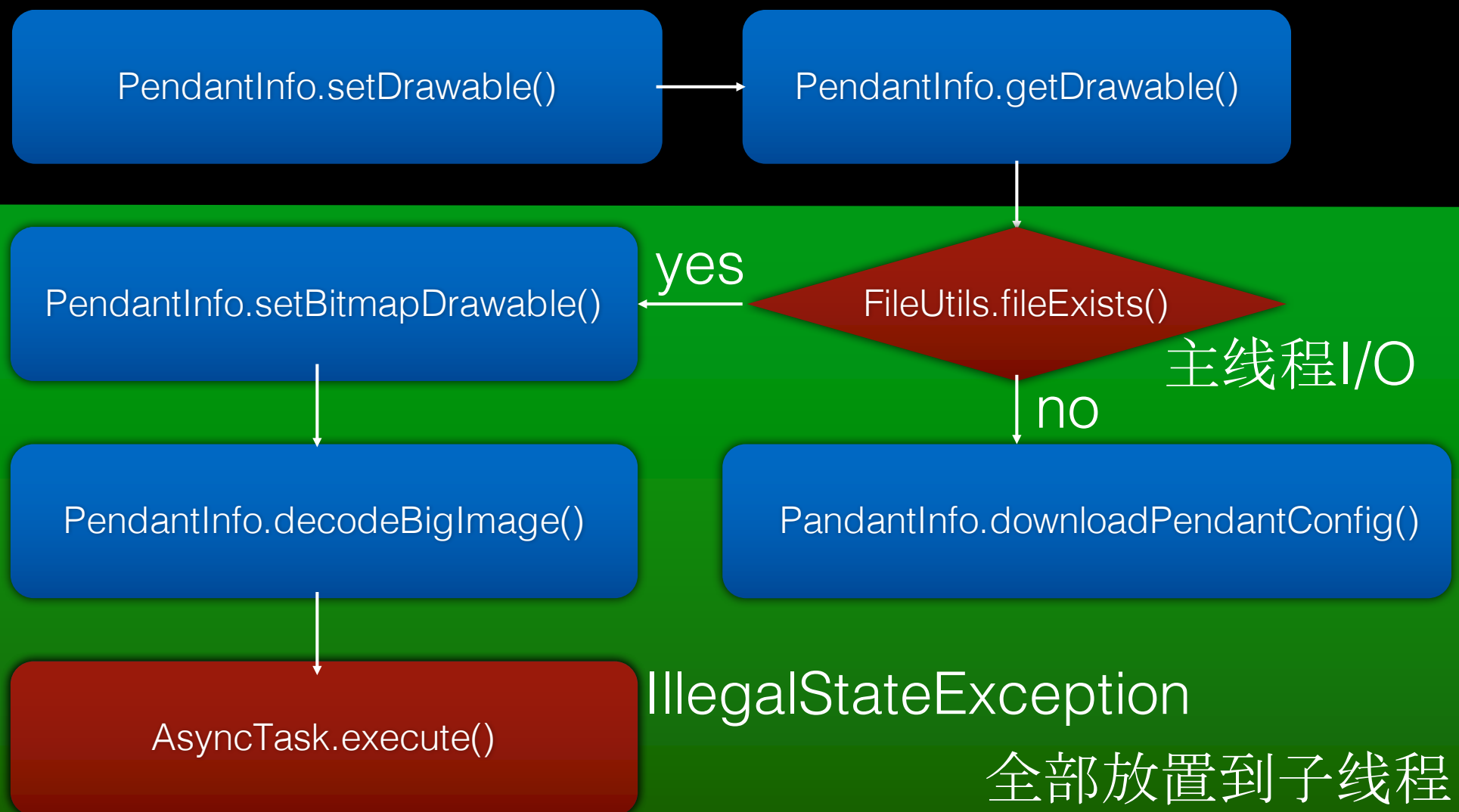
java.lang.Thread.run(Thread.java:818)



闪退怪：案例



闪退怪：案例



进一步分析：手工极难重现

```
} else if (mTask != null) {//如果已经在解析了，就不要再解析了，只负责刷新好了  
    Message msg = mHandler.obtainMessage(REFRESH_DRAWABLE);  
    Bundle data = msg.getData();  
    if (data == null) {  
        data = new Bundle();  
        data.putLong("targetId", PENDANT_TARGET_ID_ALL);  
        msg.setData(data);  
    } else {  
        data.putLong("targetId", PENDANT_TARGET_ID_ALL);  
    }  
    mHandler.sendMessage(msg);  
    return;  
}
```

1. 如果mTask被初始化，新的线程进来也会

2. 如果mTask在没有被覆盖前先execute了，也不会执行同一个mTask

```
mSource = new SourcesBigPng(mBitImage  
mRepeatCount = 0;  
mTask = new DecodeNextFrameTask(mSource, PENDANT_TARGET_ID_ALL);  
long nextFrameTime = mTask.execute(mSource, mRepeatCount, mIntervalTime);  
mTask.execute(nextFrameTime, (long) mCurFrameIndex, (long) mPendantResType);
```

mTask.execute 同一个mTask对象execute第二次



难定位

什么问题难定位？

OOM

内存溢出

DeadLock



解决思路

OOM

内存溢出

DeadLock

什么信息可以帮助定位OOM与DeadLock

取内存快照

Address
Sanitizer

取堆栈

QAPM

QAPM



解决与度量

解决

度量

VS

native hook
修改系统行为

crash率
人均crash次数
crash覆盖人数

