

## Android性能调优之方法耗时

发件人：Steveyan<no-reply@yinxiang.com>

时 间：2016年7月25日(星期一) 中午1:42

收件人：中流击水<719905193@qq.com>

工具：traceview

### 1、选择跟踪范围

```
Debug.startMethodTracing("文件名");
```

```
Debug.stopMethodTracing();
```

### 2、添加SD卡访问权限

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

```
<uses-
```

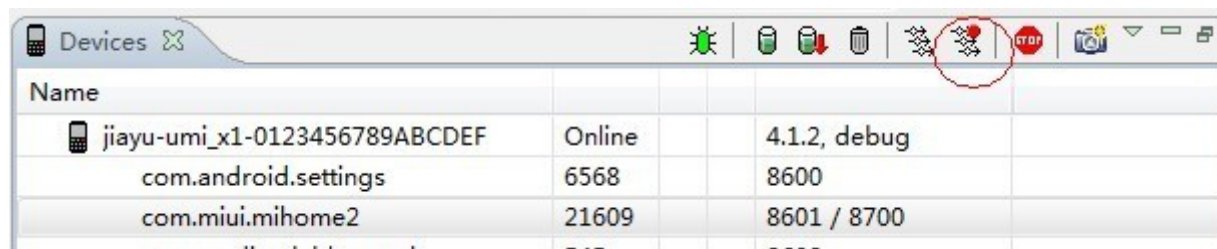
```
permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"/>
```

### 3、导出traceview文件

### 4、打开trace文件

## (2) 第二种使用方法演示

Eclipse -> DDMS -> Start Method Profiling



二者的区别，第一种方式更精确到方法，不方便的地方是自己需要添加方法并且要导出文件，第二种方式的优缺点刚好相反。

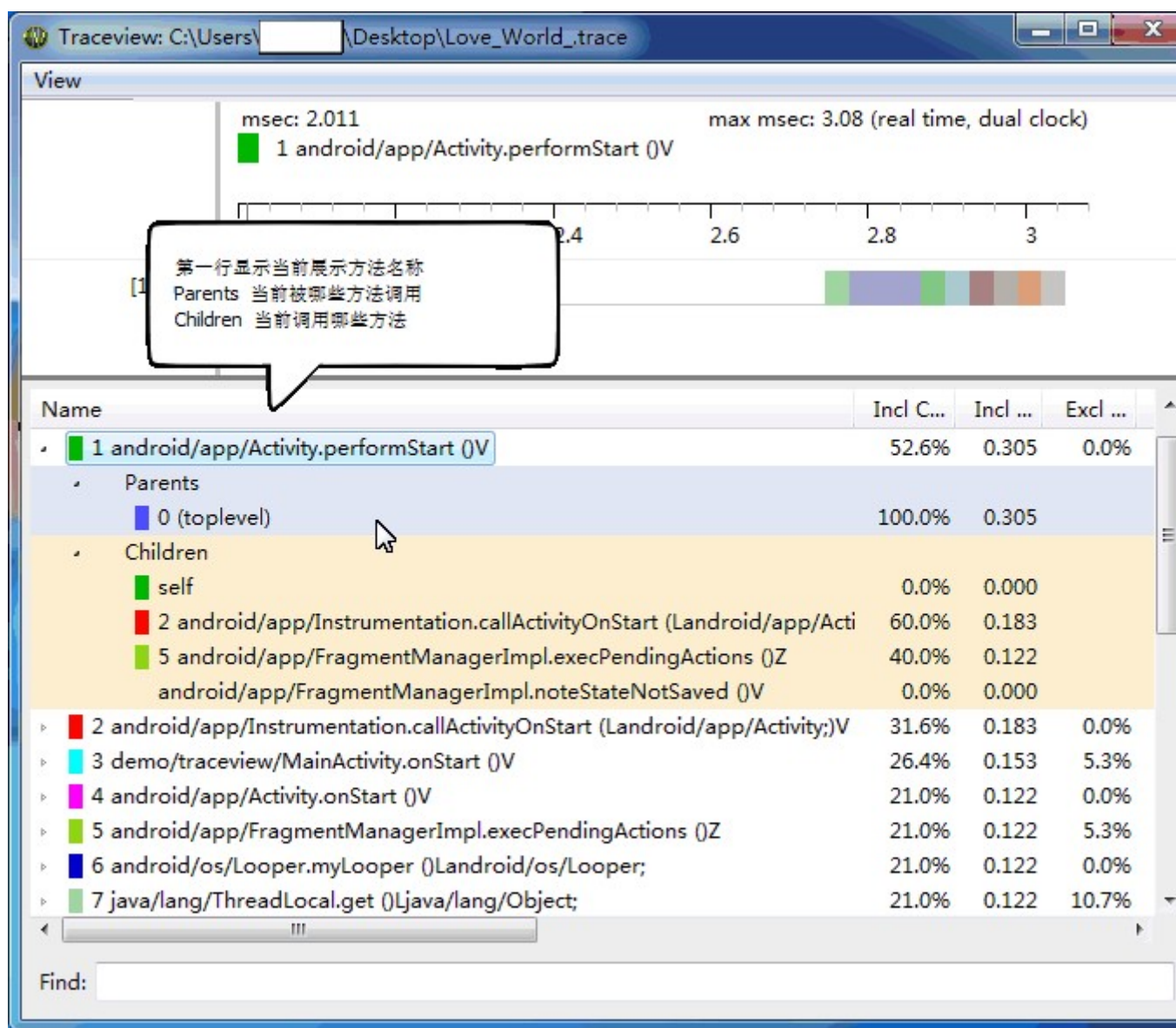
## 二、TraceView工具面板介绍

有两方面用途：

- 1 查看跟踪代码的执行时间，分析哪些是耗时操作
- 2 可以用于跟踪方法的调用，尤其是Android Framework层的方法调用关系

## 获取方法的调用顺序

1. 在traceview中搜索响应的方法名不能使用大写字母
2. 搜索出的方法会自动展开，其中包含Parents 和 Children 两组信息
3. 点击Parents下的方法名，直接跳转到调用当前的方法处。Children相反



Traceview 面板分上下两部分

### 上面是时间轴面板 (Timeline Panel)

左侧显示的是线程信息

右侧黑色部分是显示执行时间段、白色是线程暂停时间段，

右侧鼠标放在上面会出现时间线纵轴，在顶部会显示当前时间线所执行的具体函数信息

### 下面是分析面板(Profile Panel) - 每一列内容

Inclusive time - 函数本身运行花费时间 + 函数调用其他函数时间

Exclusive time - 函数本身运行花费时间。

Calls + RecurCall/Total 调用 + 重复调用次数 / 函数总调用次数

Cpu Time/Call 总的Cpu时间与总的调用次数之比

## 三、如何进行具体的分析

有两个问题需要解决：

### 1. 如何定位到所关心的地方？

上面只是介绍了如何使用TraceView且有两种用法，但是有时使用第一种方式范围又不太精确，使用第二种添加代码的方式，可能有些地方又监听不到。这种情况可以尝试把开始或者结束放到延迟线程中，延迟一段时间在执行开始或者结束。

### 2. 如何查找出哪些地方比较耗时？

TraceView罗列出了是所有监听到的方法，当然也包括Android系统很多方法的耗时，如何在这么多方法里面查找到自己关心的？可以通过TraceView 底部的find 来查找，通常Android app都是有包名的，可以先针对某些关心的列排序后，在通过包名进行一个个查找，这些就省去自己筛选出自己app 方法耗时排行的时间。

---