电量优化

提问：拍照和图片处理他们可以放在链接充电线后再处理？

答案：图片处理。

像这些不需要及时地和用户交互的操作可以放到后面处理。

比如：360手机助手，当充上电的时候，才会自动清理手机垃圾，自动备份上传图片、联系人等到云端。

# 通过监控电量状态来进行电量管理。

我们可以通过下面的代码来获取手机的当前充电状态：

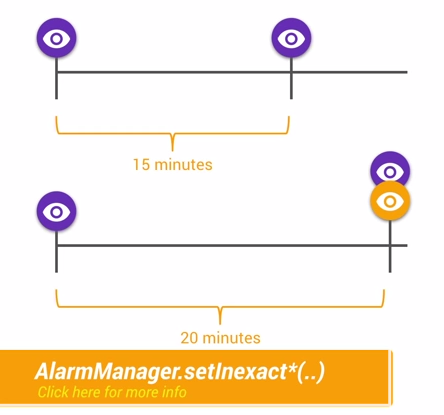
*// It is very easy to subscribe to changes to the battery state, but you can get the current  
// state by simply passing null in as your receiver. Nifty, isn't that?*IntentFilter **filter** = **new** IntentFilter(Intent.***ACTION\_BATTERY\_CHANGED***);  
Intent **batteryStatus** = **this**.registerReceiver(**null**, **filter**);  
**int chargePlug** = **batteryStatus**.getIntExtra(BatteryManager.EXTRA\_PLUGGED, -1);  
**boolean acCharge** = (**chargePlug** == BatteryManager.BATTERY\_PLUGGED\_AC);  
**if** (**acCharge**) {  
 Log.v(**LOG\_TAG**,“The phone is charging!”);  
}

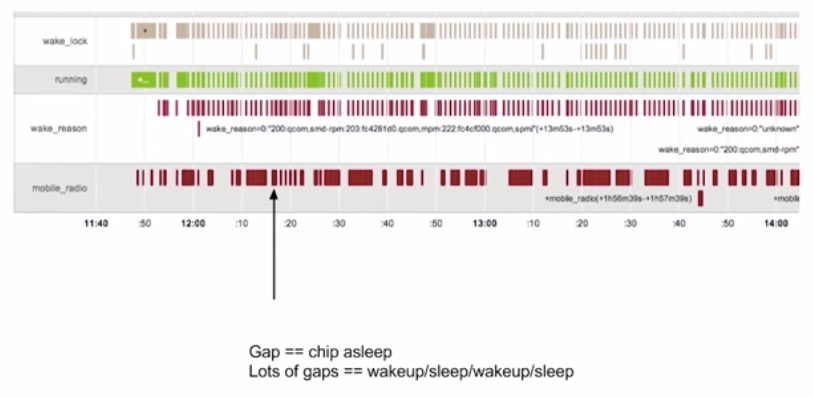
在上面的例子演示了如何立即获取到手机的充电状态，得到充电状态信息之后，我们可以有针对性的对部分代码做优化。比如我们可以判断只有当前手机为AC充电状态时 才去执行一些非常耗电的操作。

*/\*\*  
 \* This method checks for power by comparing the current battery state against all possible  
 \* plugged in states. In this case, a device may be considered plugged in either by USB, AC, or  
 \* wireless charge. (Wireless charge was introduced in API Level 17.)  
 \*/***private boolean** checkForPower() {  
 *// It is very easy to subscribe to changes to the battery state, but you can get the current  
 // state by simply passing null in as your receiver. Nifty, isn't that?* IntentFilter filter = **new** IntentFilter(Intent.***ACTION\_BATTERY\_CHANGED***);  
 Intent batteryStatus = **this**.registerReceiver(**null**, filter);  
  
 *// There are currently three ways a device can be plugged in. We should check them all.* **int** chargePlug = batteryStatus.getIntExtra(BatteryManager.EXTRA\_PLUGGED, -1);  
 **boolean** usbCharge = (chargePlug == BatteryManager.BATTERY\_PLUGGED\_USB);  
 **boolean** acCharge = (chargePlug == BatteryManager.BATTERY\_PLUGGED\_AC);  
 **boolean** wirelessCharge = **false**;  
 **if** (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.JELLY\_BEAN\_MR1) {  
 wirelessCharge = (chargePlug == BatteryManager.BATTERY\_PLUGGED\_WIRELESS);  
 }  
 **return** (usbCharge || acCharge || wirelessCharge);  
}

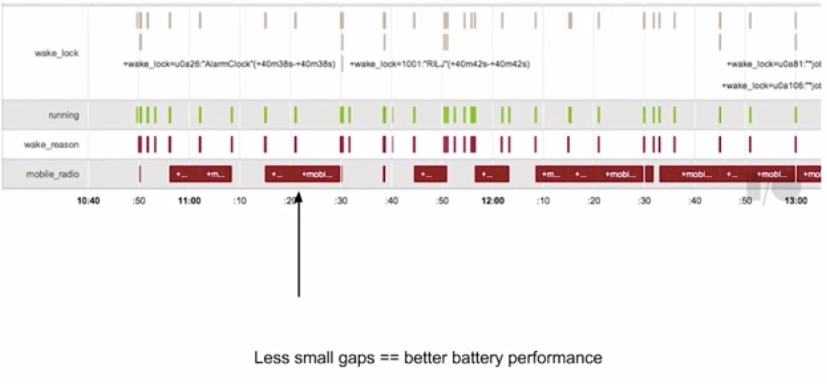
# 练习：Modify WaitForPowerActivity

提示用户插入手机充电，

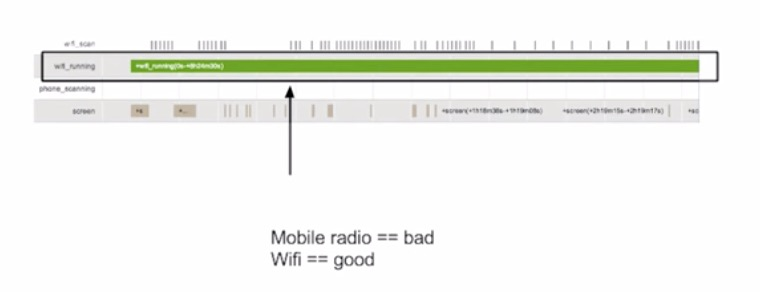




经过优化之后，如果呈现下面的图示，说明电量消耗的性能是良好的：



另外WiFi连接下，网络传输的电量消耗要比移动网络少很多，应该尽量减少移动网络下的数据传输，多在WiFi环境下传输数据。



那么如何才能够把任务缓存起来，做到批量化执行呢？下面就轮到Job Scheduler出场了。

使用[Job Scheduler](https://developer.android.com/reference/android/app/job/JobScheduler.html)，应用需要做的事情就是判断哪些任务是不紧急的，可以交给Job Scheduler来处理，Job Scheduler集中处理收到的任务，选择合适的时间，合适的网络，再一起进行执行。