Selectively Taming Background Android Apps to Improve Battery Lifetime

Marcelo Martins  
Brown University

Justin Cappos  
New York University

Rodrigo Fonseca  
Brown University
Connect charger

The battery is getting low.
1% remaining

Battery use OK
Your Battery and You

Active device → high battery drainage

Matches our battery life expectation
Your Battery and You

Idle device $\rightarrow$ expect low battery drain (?)

Background activities break low-power harmony

Energy bugs keep the device awake
Hidden Energy Drain

Pure AOSP

+ Google Mobile Services
Battery Level (%)

Hours

Samsung Galaxy Nexus

~1.8 days

 ASUS MeMO Pad 7

~8.3 days
Hidden Energy Drain

Power (W)
Timeline (s)

AlarmManager
NetworkLocationService
GCoreFlp
GeofencerStateMachine
NlpCollectorWakeLock
NlpWakeLock
UlrdDispatchingService
PendingIntentCallbackService
NlpLocationReceiverService

Pure AOSP

+ Google Mobile Services
What Can We Do Today?

Reconfigure apps

Uninstall/Switch apps

Kill apps

Change system settings
What Can We Do Today?

- Reconfigure apps
- Uninstall/Switch apps
- Kill apps
- Change system settings
What Can We Do Today?

- Reconfigure apps
- Uninstall/Switch apps
- **Kill apps**
- Change system settings
What Can We Do Today?

Reconfigure apps

Uninstall/Switch apps

Kill apps

Change system settings
Runtime mechanism to rate-limit background events
Backgrounding Sources?

**Services**

- Alarms
- Intent Receivers
- Wakelocks

Tomorrow’s temperature?

25° C
Backgrounding Sources?

Services

Alarms

Intent Receivers

Wakelocks

Sync files every 15 min
Backgrounding Sources?

Services

Alarms

**Intent Receivers**

Wakelocks

Wake me up on new msg
Backgrounding Sources?

Services

Alarms

Intent Receivers

Wakelocks

Keep screen awake
Backgrounding Sources?

- Services
- Alarms
- Intent Receivers
- Wakelocks

Asynchronous
Symmetric
class fetchTemperature extends IntentService {
    @Override
    public void onHandleIntent(...) {...}
};

1. Event definition (what)
Prelude to Blocking: Event Dispatching

1. Event definition (what)
2. Event request

```java
class fetchTemperature extends IntentService {
	...

startService(new Intent(..., fetchTemperature.class));
```
Prelude to Blocking: Event Dispatching

1. Event definition (what)
2. Event request
3. Event callback (where)

```java
class fetchTemperature extends IntentService {
    ...
    startService(new Intent(..., fetchTemperature.class));
    ...
    acquireWakelock(...)
    ...
    startServiceLocked(...)
    ...
    startRunningLocked(...)
}
```
Blocking the Event Dispatch

1. Event definition (what)
2. Event request
3. Event callback (where)

```java
class fetchTemperature extends IntentService {
    ...
}

startService(new Intent(…, fetchTemperature.class));
```

Call order:
- startService(…)
- acquireWakelock(…)
- startServiceLocked(…)
- startRunningLocked(…)

Tomorrow’s temperature?
Throttling Abstraction

Controller

Observer → Arbiter → Actuator
Observer
Arbiter
Actuator

Event rate

Observer Arbiter Actuator

App

TAMER

App Name Type #Calls
fetchTemp Service 1231+1
syncPhotos Service 342
scanSystem Service 3

Tomorrow's temperature?

Controller

Event

Event rate
if (monitoredRate > policyThreshold) → should block
if (monitoredRate > policyThreshold) \(\rightarrow\) should block
Implementation

Java-based runtime code injection via Xposed

Interposes on event callbacks to control their execution

Policies in JSON format
Installation

Installs like any other app (APK file)

Runtime modifications

- No need for app recompilation/reinstallation

Android compatibility

- Different vendors (Samsung, HTC, ASUS, CyanogenMod, etc.)
- Different versions (Ice Cream to KitKat)
- Requires rooting
Evaluation

Four devices

Android KitKat

Experiments

General lifetime

Energy Bugs

GPS throttling

Galaxy Nexus (2011)
Kindle Fire2 (2012)
Galaxy S4 (2013)
MeMOPad 7 (2014)
Evaluation

Four devices

Android KitKat

Experiments

General lifetime

Energy Bugs

GPS throttling
General Lifetime

Pure AOSP + Google Mobile Services
Battery Level (%) vs. Hours for Samsung Galaxy Nexus and ASUS MeMO Pad 7.
Exploring Taming Possibilities

New Policies

**Tamer-15:** Events allowed every 15 minutes

**Tamer-45:** Events allowed every 45 minutes

<table>
<thead>
<tr>
<th>Event Name</th>
<th>Type</th>
<th>Count</th>
<th>Duration (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NlpWakelock</td>
<td>W</td>
<td>5963</td>
<td>1662.71</td>
</tr>
<tr>
<td>NlpCollectorWakelock</td>
<td>W</td>
<td>2121</td>
<td>3926.63</td>
</tr>
<tr>
<td>LocationManagerService</td>
<td>W</td>
<td>2030</td>
<td>67.12</td>
</tr>
<tr>
<td>NlpLocationReceiverService</td>
<td>S</td>
<td>1159</td>
<td>-</td>
</tr>
<tr>
<td>NetworkLocationService</td>
<td>S</td>
<td>579</td>
<td>-</td>
</tr>
</tbody>
</table>

Monitoring time: 100 hours

One call per min
Sys awake 11hs
Battery Level (%) vs. Hours for
- Samsung Galaxy Nexus
- ASUS MeMO Pad 7

Comparison between Pure AOSP and w/ Google M. Services.
Battery Level (%)

- Pure AOSP
- w/ Google M. Services
- w/ GMS + Tamer-15
- w/ GMS + Tamer-45

Hours

**Samsung Galaxy Nexus**

- +23 hours
- +32 hours

**ASUS MeMO Pad 7**

- +4.6 days
- +5.8 days
Chasing Energy Bugs – Bejeweled Blitz

10M+ downloads on Play Store
Top 20 on Puzzle Category
Chasing Energy Bugs – Bejeweled Blitz
Bejeweled Blitz – Effects on Battery Lifetime

Battery Level (%)

- Candy Crush Saga
- Bejeweled (Untamed)

Hours

CPU Residency (%)

CPU Frequency

Samsung Galaxy S4
Chasing Energy Bugs – Bejeweled Blitz

```
{
   "AppName": "com.ea.BejeweledBlitz_na",
   "EventList": {
      "Entry": {
         "Type": "wakelock",
         "Name": "AudioIn",
         "AllowEvery": "0",
         "BackgroundOnly": "true"
      }
   }
}
```
After Taming

- Battery Level (%)
  - Candy Crush Saga: Blue line
  - Bejeweled (Untamed): Red line
  - Bejeweled (Tamed): Green line

- CPU Residency (%)
  - Untamed: Red bars
  - Tamed: Green bars

- CPU Frequency
  - 3.84 GHz
  - 4.66 GHz
  - 5.94 GHz
  - 7.02 GHz
  - 8.10 GHz
  - 9.18 GHz
  - 1.03 GHz
  - 1.13 GHz
  - 1.24 GHz
  - 1.35 GHz
  - 1.46 GHz
  - 1.57 GHz
  - 1.67 GHz
  - 1.78 GHz
  - 1.89 GHz

Samsung Galaxy S4
Limitations and Extensions

How to define good policies?

- Avoid breaking functionality
- Dynamic policies
- Context awareness
Limitations and Extensions

How to define good policies?

How to engage users?
Limitations and Extensions

How to define good policies?

How to engage users?

Event batching instead of blocking
Wrap Up

Mobile apps can silently kill battery

**TAMER:** OS-level mechanism to **throttle** background events

Separation of control and policy

Extends **battery** life / Selectively keeps **functionality**

Effectively mitigates energy bugs