

Android OS におけるアプリケーションの起動による電力消費に関する一考察

中村 優太^{†1} 早川 愛^{†2} 小柳 文乃^{†2} 半井 明大^{†3} 竹森 敬祐^{†3} 小口 正人^{†2} 山口 実靖^{†1}
 工学院大学^{†1} お茶の水女子大学^{†2} 株式会社 KDDI 研究所^{†3}

1. はじめに

近年、スマートフォンやタブレット PC が普及し、それらの携帯端末で動作するソフトウェアプラットフォームとして Android OS が注目されている。Android OS の世界市場における 2014 年第 3 四半期のシェアは 84.4% に達しており [1], Android OS は非常に重要なプラットフォームとなっている。また、スマートフォンの最大の課題は「バッテリーの持続時間である」との報告があり [2], Android OS における消費電力の低減は非常に重要な課題であると考えられる。

Android OS は PC 用 OS である Linux を元に作成されており、Linux OS に類似した動作をする。すなわち、Android OS はマルチタスク OS であり常に複数のプロセスが常駐、動作している。また、指定時刻やイベント発生時にアプリケーションを起動させる仕組みが用意されており、多くのアプリケーションにおいてユーザが直接操作を行わなくても動作する機能が備わっている。これらのユーザの直接的操作を伴わないアプリケーションの起動により端末のバッテリーが消費されるが、ユーザが直接操作をしていない状態における端末やアプリケーションの動作の把握はユーザには困難であると予想できる。

2. WakeLock

Android OS には端末が Sleep していない状態を保持するために使用される WakeLock という仕組みが用意されている。これは、センサで情報を取得し続ける、画面を点灯し続けるなどの目的で使用される。

WakeLock が行われずにディスプレイが消えた状態になると、Android 端末は Sleep 状態に入りバッテリーの消費を抑える。Sleep 状態では、バッテリーの主な消費原因である CPU 稼働、ディスプレイ点灯、通信が抑制される。WakeLock と

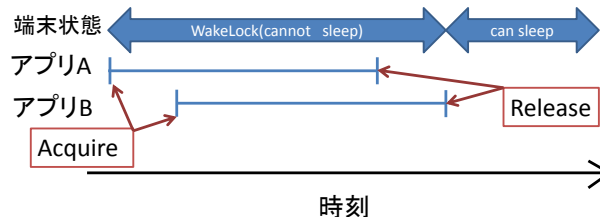


図1 WakeLock と Sleep の関係

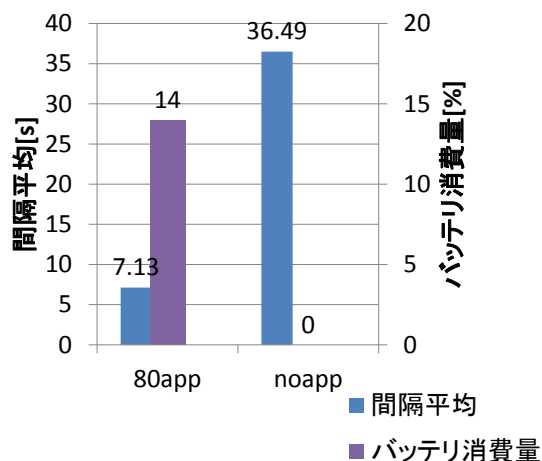


図2 WakeLock とバッテリー消費量の関係

Sleep 状態の関係を図 1 に示す。図は端末内で WakeLock を行うアプリケーション A および B が実行されている状態を示しており、横軸は時刻となっている。アプリケーションはそれぞれ独自のタイミングで WakeLock の取得と開放を行うことができ、取得には `acquired()` を、開放には `released()` を使用する。すべてのアプリケーションが WakeLock を開放すると端末は Sleep 状態に移行することが可能となる。

3. WakeLock による電力消費

本章にて WakeLock による電力消費について考察する。

3.1. 測定方法

Android OS を改変し、無操作状態の Android 端末のバッテリー残量の推移と、アプリケーションが WakeLock の Release を行った時刻と、Release を行ったアプリケーションを取得した。

A Study on Power Consumption of Android Devices Considering Waking-up by Applications

^{†1} Kogakuin University

^{†2} Ochanomizu University

^{†3} KDDI R&D Laboratories, Inc.

使用したアプリケーション数は 80 で、計測は 24 時間(1440 分間)行った。ただし、実験開始後 30 分はディスプレイが点灯する設定とした。

実験に用いたアプリケーションは 2014 年 5 月 19 日の Google Play Store 無料アプリケーションランキング[3]の上位 80 件のアプリケーションである。

3.2. 計測結果

図 2 に測定結果を示す。“80app”はすべてのアプリケーションを端末に導入した場合を示し，“noapp”は端末にアプリケーションを追加でインストールしていない状況（Android Open Source Project[4]にて配布されている OS に添付されているアプリケーションのみインストールされている状況）を示す。図の“間隔平均”は WakeLock の Release が行われた間隔の平均である。間隔が短いほど頻繁に WakeLock が行われており、端末の Sleep が妨げられていることを意味する。図内のバッテリー消費量は /sys/class/power_supply/battery/capacity より得られるバッテリー残量の減少量である。両実験ともバッテリー残量が 100%の状態から実験を開始している。

測定結果より、アプリケーションを 80 個インストールし WakeLock の Release が行われる間隔が短い環境ではバッテリーが 14%消費されたのに対し、アプリケーションをインストールせず WakeLock の Release が行われる間隔が長い環境ではバッテリーが減少していないことがわかる。これより、WakeLock 回数と電力消費には相関があると予想することができる。

3.3. WakeLock を行うアプリケーション削除時の電力

本節にて、WakeLock を行うアプリケーションのアンインストールによる消費電力削減効果について考察する。

すべてのアプリケーションを入れた“80app”，WakeLock を行わないアプリケーションから無作為に 6 つ選びアンインストールした“without_nowake6”，WakeLock を行う 6 件のアプリケーションを端末からアンインストールした“without_top6”，追加でインストールしたアプリケーションが 0 個の標準状態(Android Open Source Project にて配布されている OS に添付のアプリケーションのみがインストールされている状態)である“noapp”の消費電力比較を行った。計測は 24 時間無操作でバッテリー消費量と WakeLock を Release した時刻の調査を行った。結果を図 3 に示す。図中の間隔平均は、WakeLock の Release を行った時刻とその次の

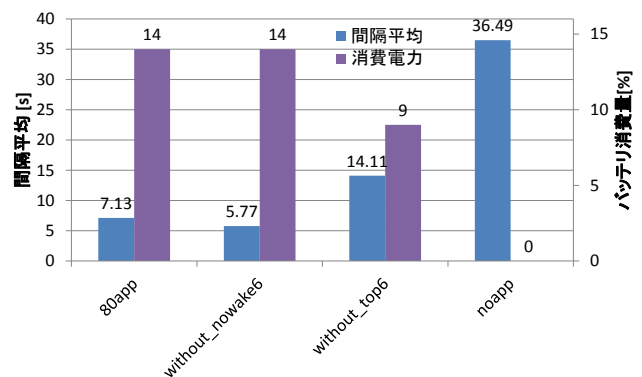


図 3 Release 回数が多いアプリケーション削除時の電力

WakeLock の Release を行った時刻の差の平均を示している。図より、WakeLock を行わないアプリケーションをアンインストールしても無操作時の消費電力の低減にはほとんど効果が無いが、WakeLock を行うアプリケーションのアンインストールは無操作時の消費電力の低減に大きな効果があることがわかる。すなわち、WakeLock の Release の観察により、無操作時の消費電力が多いアプリケーションの特定に成功できたと考えることができる。

4. おわりに

本稿では、無操作状態の Android 端末における消費電力に着目し、WakeLock の実行回数に基づく無操作時電力消費の原因となるアプリケーションの特定手法についての考察を行った。また、本手法により特定したアプリケーションを削除したところ消費電力の低減が確認され、本手法に有効性があることが確認された。

今後は、異なるアプリケーションを用いての評価を行っていく予定である。

謝辞

本研究は JSPS 科研費 24300034, 25280022, 26730040 の助成を受けたものである。

参考文献

- 1) Smartphone OS Market Share, Q3 2014.
<http://www.idc.com/prodserv/smartphone-os-market-share.jsp>
- 2) BCN
http://www.nikkei.com/article/DGXNASFK2600W_W3A320C1000000/
- 3) 無料トップ Android アプリ
https://play.google.com/store/apps/collection/topselling_free
- 4) Android Open Source Project
<http://source.android.com/>