

アプリケーションの動作と消費電力を考慮したスマートフォン CPU クロック周波数制御

関屋拓司^{†1} 栗原駿^{†2} 福田翔貴^{†2} 濱中真太郎^{†2} 小口正人[‡] 山口実靖^{†1†2}

工学院大学 工学部 情報通信工学科^{†1} 工学院大学 工学研究科 電気・電子工学専攻^{†2}

お茶の水女子大学 理学部 情報科学科[‡]

1. はじめに

年々スマートフォンの普及率が増え、その中でも Android OS のシェアは世界でトップである。「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[1]、スマートフォンにおける CPU は主たる電力消費源の1つである。Android OS では CPU クロック周波数を制御できるが、サーバ等における使用を想定した Linux カーネルを用いており、スマートフォンにおいて必ずしも適切でないことも考えられる。

本稿では、1つのアプリケーションに着目し、既存の CPU クロック周波数制御手法の評価と、その改善に関する考察を行う。具体的には、アプリケーションの動作を考慮して CPU クロック周波数を制御し大きな性能劣化無く消費電力を減少させる手法を提案し、その性能評価を行う。

2. AndroidにおけるCPUクロック周波数制御

本章では、スケーリング governor と呼ばれる Android の CPU クロック周波数制御手法について述べる。ondemand は CPU の負担に応じて CPU クロック周波数を動的に切り替える governor であり、初期設定で選択されている governor である。powersave は省電力用の governor で、積極的に周波数を下げる governor である。

3. AndroidにおけるCPUクロック周波数制御の評価

本章では前章で説明した Android における CPU クロック周波数制御手法の評価を行う。評価で用いた実アプリケーションは 2016 年 10 月 15 日のトップセールスのアプリケーションランキング上位のゲームアプリケーションを 1 件用いた。測定は当該実アプリケーションを起動させ同一処理を完

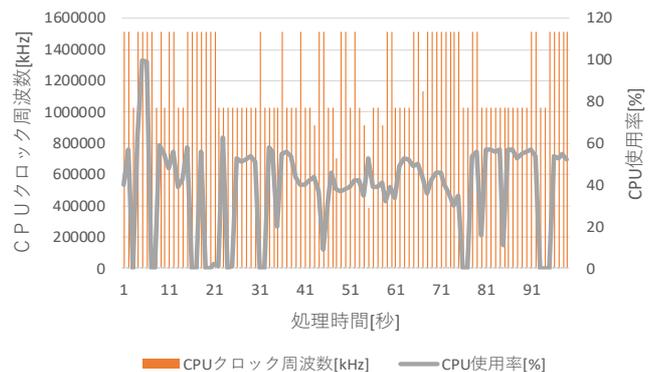


図 1 CPU クロック周波数と使用率 (ondemand)

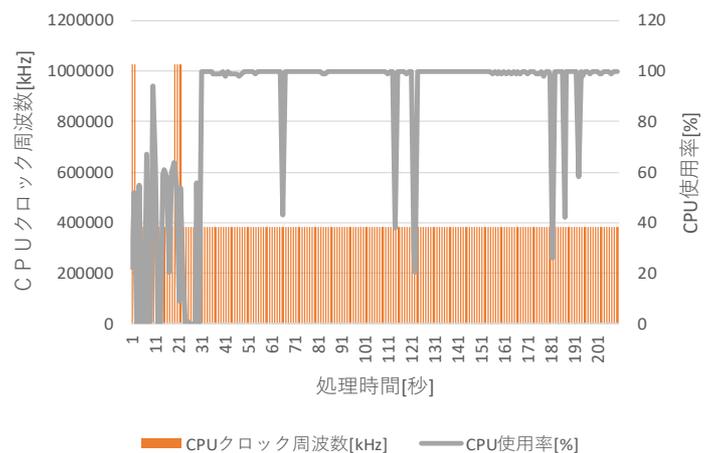


図 2 CPU クロック周波数と使用率 (powersave)

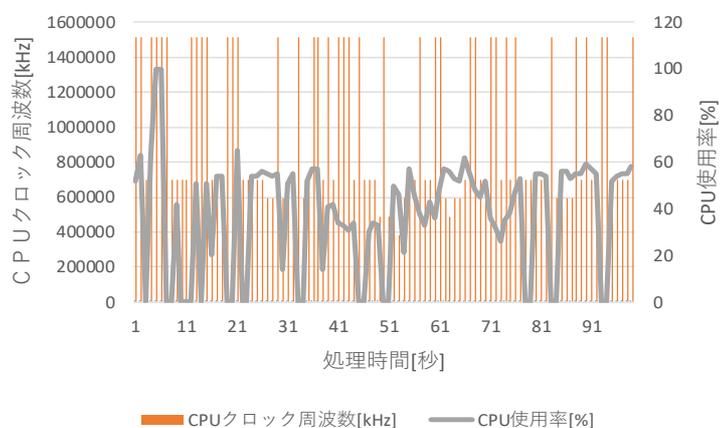


図 3 CPU クロック周波数と使用率 (改善案)

Smartphone CPU Clock Frequency Control with Consideration of Application Behavior and Power Consumption

^{†1}Hirokazu Sekiya, Department of Information and Communication Engineering Kogakuin University

^{†2}Shun Kurihara, Shoki Fukuda, Shintaro Hamanaka, Saneyasu Yamaguchi, Electrical Engineering and Electronics, Kogakuin University

[‡]Masato Oguchi, Department of Information Sciences, Ochanomizu University

了するまでの時間を5回測定する。評価に用いた端末は Nexus7 (2013), CPU Qualcomm Snapdragon S4 Pro 1.5GHz, メモリ 2GB, OS Android 6.0.0 である。ondeman における CPU クロック周波数と CPU 使用率の推移を図 1 に、消費電力と処理時間を図 4 と図 5 に ondemand として示す。powersave における消費電力と処理時間を、図 4 と図 5 の powersave として示す。図 1 は 5 回測定したうちの 1 回の結果であり、図 4, 5 は 5 回の平均である。

図 1 の 20 秒前後や 45 秒前後など、CPU 使用率が低い時に CPU クロック周波数が高い箇所があることがわかる。このことから、CPU クロック周波数を適切に下げていることが分かる。また、CPU クロック周波数の上昇に関しては、減少よりも積極的に上げていることがわかる。

4. CPU クロック周波数制御手法の改善

本章では、CPU クロック周波数制御手法の改善案を提案する。本改善案では、AndroidOS のカーネルの実装に対して、CPU クロック周波数が最大値から下げる場合は通常より大きく下げ、CPU クロック周波数を最大値以外に上げる場合は通常より小さく上げるように修正を行う。具体的には/arch/arm/mach-msm/cpufreq.c を修正し、最大値から下げる場合は選択される周波数テーブルの index を当初の実装の index の 1/4 とし(当初実装は table[index].frequency を選択し、修正版では table[index/4].frequency を選択)、最大値以外に上げる場合は当初実装の 1/2 の index 値を用いる様に修正した。

5. 性能評価

本章では前章の改善案の評価を行う。評価環境は 3 章と同様である。評価結果は図 2,3,4,5 に示す。

図 2 はスケーリング governor を powersave に設定した時の経過時間に対する CPU クロック周波数と CPU 使用率の推移である。図 3 はスケーリング governor を 4 章で提案した改善案を適用した ondemand にした場合の経過時間に対する CPU クロック周波数と CPU 使用率の推移である。図 2,3 は各 5 回測定した内の 1 回である。図 4, 5 の横軸は用いた governor である。

図 1 の ondemand と図 3 の改善 ondemand を比較すると、改善 ondemand は CPU クロック周波数が最大値から下げる場合は通常 ondemand より大きく下げていることがわかる。また、CPU クロック周波数を上げる場合は通常より小さく上げていることがわかる。図 2 の powersave は CPU クロック周波数を最低値に選択している事が多く、CPU 使用率が 100% である時間が多いことがわかる。ondemand と改善案を比較すると、改善案の方が消費電力量は 11.85% 少なく、処理時間は 0.84% しか長くなっていない。powersave と改善案を比較すると、改善案の方が消費電力量は 9.79% 減っており、処理時間は powersave の方が 112.96% と伸びている。以上のこと

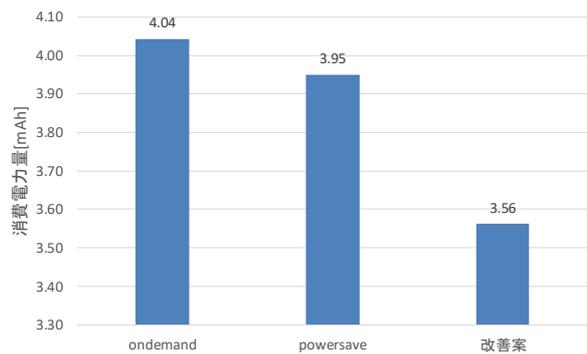


図 4 消費電力量の比較

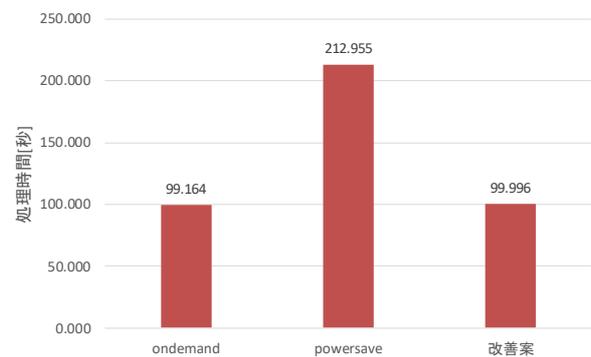


図 5 処理時間の比較

から既存の ondemand や powersave より改善案が優れ、本改善案の有効性が確認できた。

6. おわりに

本稿では、スマートフォンアプリケーション動作時における Android OS の CPU 周波数制御手法の評価と考察を示し、その改善に関する考察を行った。観察に用いたアプリケーションを用いた評価を行い、改善案により同等の処理時間で消費電力を削減できることが確認された。

今後はより多くのアプリケーションで評価を行う予定である。

謝辞

本研究はJSPS科研費25280022, 26730040, 15H02696の助成を受けたものである。

本研究は、JST、CRESTの支援を受けたものである。

参考文献

[1] 日本経済新聞 2013 年 4 月 1 日 http://www.nikkei.com/article/DGXNASFK2600W_W3A320C100000/