

モバイルデバイス間における省電力を考慮したファイル同期方式の実装

高見澤 拓郎[†] 鶴岡 行雄[‡] 佐藤 喬[†] 多田 好克[†]

[†]電気通信大学 大学院情報システム学研究科

〒182-8585 東京都調布市調布ヶ丘 1-5-1

[‡]NTT 情報流通プラットフォーム研究所

〒180-8585 東京都武蔵野市緑町 3-9-11

1. はじめに

現在スマートフォンやノート PC 等のモバイルデバイスが普及しており、1人のユーザが複数台所有することも増えている。複数台のモバイルデバイスを利用する場合、デバイス間でのデータの共有が必要となる。例えば電車の中においてスマートフォンでダウンロードしたコンテンツを、喫茶店でノート PC で視聴する場合である。

モバイルデバイスとしてスマートフォンとノート PC からなる2つのデバイス間の同期について考える。ノート PC はバッテリー持続時間が短いため、通常は停止しているものとする。一方、スマートフォンは通話待ちのため常時起動しているものとする。ユーザが利用時にノート PC を起動した後に同期を行った場合、同期が完了してユーザがデータにアクセスできるまでには待ち時間が生じる。このような待ち時間はサービスの利便性を損ねるため問題となる。

同期に伴う待ち時間を軽減するためには、事前にノート PC を起動し同期を行う方法が考えられる。この場合、ノート PC の起動と停止に伴う消費電力がオーバヘッドとなる。もし更新が起る度に頻繁に同期を行った場合、ユーザの待ち時間は軽減される半面、上記のオーバヘッドの蓄積が問題となる。このように、ユーザの待ち時間と消費電力はトレードオフの関係にある。

同期ソフトウェアとしては Dropbox[1] や iCloud[2] などがあるが、これらについては停止状態のデバイスへの同期について考慮されていなかった。したがって、更新量が多い場合にはユーザの待ち時間が問題となる。

本研究ではモバイルデバイス間のファイル同期、特に停止デバイスへの同期においてユーザの待ち時間を一定範囲に抑えつつ、消費電力を低減する同期方法を提案する。提案法では同

期を行うタイミングを制御し、かつ単位時間あたりの更新量に基づき電源状態を切り替える。この提案法をプロトタイプ実装し、動作を確認した。

2. モバイルデバイス間のファイル同期

本稿ではスマートフォンからノート PC への同期に着目する。逆方向の同期の場合、両デバイスは起動中であるため、更新が蓄積することによる待ち時間の増加はない。また、スマートフォン上でのファイルの変更(生成, 修正, 削除)を更新と呼び、ノート PC へ更新を反映させるための、スマートフォンからノート PC へのデータの送信を転送と呼ぶ。ここでの同期の要求条件はユーザの待ち時間を一定の許容範囲内に収めつつ消費電力を最小限に抑えることである。

同期に必要な消費電力量とは、ノート PC の起動と停止に必要な消費電力量と、スマートフォンがノート PC へデータを転送する間に各々のモバイルデバイスが消費する電力量の和である。ただし、ノート PC と比べスマートフォンは消費電力量が少ないため、ここではノート PC の消費電力量のみを考える。

ユーザが許容できる待ち時間を T_w [秒]とする。 T_w は起動時間と転送にかかる時間に分けられる。起動と停止を伴う1回の同期において許容できる最大の転送データ量を D [byte]とすると、 $D = (T_w - \text{起動時間})$ [秒] \times 転送速度[byte/秒]である。スマートフォン上において、更新の蓄積が D バイトを超えた時、ノート PC を起動して転送を行い、再び停止させる。これによって待ち時間を許容範囲内に抑えつつ、更新をまとめて転送することで起動と停止に伴う消費電力量を低減できる。

デバイスの電源状態として休止, スリープ, オン状態の3つを考える。デバイスの停止状態とは休止かスリープのいずれかを指すものとする。

Energy-efficient Synchronization between Mobile Devices

[†] Takuro Takamizawa(takamizawa@spa.is.uec.ac.jp)

[‡] Yukio Tsuruoka(tsuruoka.yukio@lab.ntt.co.jp)

[†] Takashi Sato(satou-t@spa.is.uec.ac.jp)

[†] Yoshikatsu Tada(tada@is.uec.ac.jp)

る。スリープ中の消費電力は休止中の消費電力より大きい。一方、起動と停止にかかる時間および消費電力量はスリープのほうが小さい。また、起動時間がより短いことから、最大転送データ量 D はスリープの方が大きい。したがって、同じ量の更新を行う場合、スリープは休止に比べ起動と停止の回数が少なくなる。

スマートフォンでの更新量が多い場合、データ転送に伴う起動と停止が頻繁に必要となる。このため、起動と停止にかかる消費電力量がより少なく、かつ起動と停止の回数がより少ないスリープが休止より省電力となる。逆に、ほとんど更新がない場合、停止中の消費電力が少ない休止がスリープより省電力となる。図1はスマートフォン上での単位時間あたりの更新量(更新速度)とそれによって必要となるノート PC の消費電力の関係を示している。更新速度がしきい値 V_{th} より大きい時、スリープの方が省電力となり、小さい時、休止の方が省電力となる。

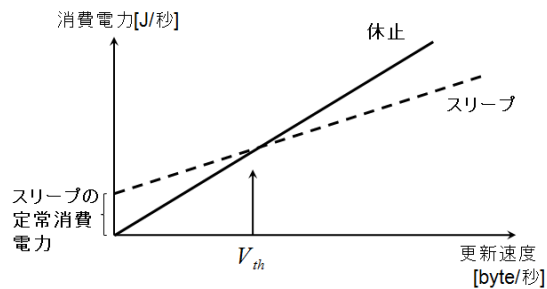


図1: 更新速度と消費電力の関係

3. 提案法の実装

提案法では、現在の更新速度 V_u を計測し、 $V_u < V_{th}$ なら休止へ、 $V_u > V_{th}$ ならスリープに切り替えることで消費電力を削減する。

プロトタイプ実装ではスマートフォンの代わりに Linux が動作する PC を用いた。この上で動く同期プログラムは C 言語で約 600 行であった。ノート PC 上で動く同期プログラムは C 言語で約 200 行であった。

2つのデバイスが起動中の場合、更新があり次第、転送を行う。それ以外の場合、すなわち本稿で着目する、停止状態のノート PC への同期の動作について以下に示す。スマートフォン(実際は PC)上ではスレッド A1~A4 が動作し、ノート PC 上ではスレッド B1 が動作する。

スレッド A1: 更新が発生したら、ファイル名等の更新情報を更新ファイルリストへ保存し、更新の蓄積が最大転送データ量 D を超えた時、ノート PC へ起動要求を送信する。

スレッド A2: 一定時間毎に更新速度 V_u を計算し、予め記憶しておいたノート PC の電源状態を読み出す。 $V_u < V_{th}$ かつ読み出した電源状態がスリープの場合、ノート PC へ起動要求を送信する。

スレッド A3: ノート PC から転送要求を受信したら、更新ファイルリストを参照し、未転送のファイルを転送する。

スレッド A4: ノート PC から停止指示要求を受信したら、更新速度 V_u を計算し、 $V_u < V_{th}$ なら休止を、 $V_u > V_{th}$ ならスリープを、ノート PC の電源状態として記録し、これを停止指示としてノート PC に送信する。

スレッド B1 (A1 または A2 による起動後に実行する): A3 へ転送を要求する。転送後、A4 に停止指示を要求し、受信した停止指示に基づきノート PC を停止させる。

更新の検知には inotify 関数[3]、転送には rcp コマンド、デバイス間の無線通信には IEEE 802.11g をそれぞれ用いた。また、ノート PC の起動要求は WoWLAN[4]によりスマートフォンから無線経由で行えるが、対応機種が限られるため、本実装では有線経由の WOL[4]で代用した。

4. おわりに

モバイルデバイス間のファイル同期において待ち時間を一定範囲に抑えつつ、消費電力を低減する同期方法を提案し、プロトタイプを実装した。提案法では同期を行うタイミングを制御し、かつ単位時間あたりの更新量に基づき電源状態を切り替える。実装したシステムを動作させることにより、電源状態の切り替えを行わない場合と比較して消費電力を削減できることを確認した。

今後の課題として、ファイルの差分の転送やデータ圧縮による転送量の削減が挙げられる。また、実用的なシステムとするためにスマートフォン上で実装する必要がある。

参考文献

- [1] Dropbox. <https://www.dropbox.com>.
- [2] iCloud. <http://www.apple.com/icloud/>.
- [3] inotify. Linux programmers manual.
- [4] Wake-on-Wireless-LAN Intel Centrino Mobile Technology Wake on Wireless LAN Feature technical brief.
- [5] 高見澤拓郎, 鶴岡行雄, 佐藤喬, 多田好克, “モバイルデバイス間における省電力を考慮した同期について”, 信学技報, vol. 110, no. 376, pp. 43-48, 2011 年.