

# AndroidにおけるLRUを用いたプロセスメモリ管理

野村 駿<sup>†</sup> 中村 優太<sup>†</sup> 坂本 寛和<sup>†</sup> 濱中 真太郎<sup>‡</sup> 山口 実靖<sup>‡</sup>

<sup>†</sup>工学院大学大学院 <sup>‡</sup>工学院大学

## 1. はじめに

Androidはスマートフォン、タブレットPC、音楽プレイヤーなど様々なデバイスのOSとして広く普及しており、重要性が高まっていると考えられる。

Androidには、low memory killerと呼ばれるプロセスメモリ管理システムが搭載されており、独自のルールに従ってメモリの管理を行なっている。low memory killerは空きメモリ量が少なくなると起動され、メモリの空き容量を確保するためにプロセスを強制終了する。このプロセスの強制終了により、ユーザが再度同じアプリケーションを使用する場合に、プロセスの再起動が必要となりユーザの待ち時間を増加させることがある。これに対して我々は、LRUを用いた強制終了プロセス選定手法を提案し、実機に実装し評価した[1]。

本稿では、ユーザのアプリケーションの起動の傾向や偏りに着目し、ユーザのアプリケーション起動履歴の解析を行う。そしてLRUを用いることの妥当性について考察を行う。

## 2. アプリケーションの起動手順

Androidにおけるアプリケーション起動の方法は「新規起動」と「再開」の2種類がある。前者はアプリケーションプロセスが存在しない状態からアプリケーションプロセスを生成する起動方法であり、後者はバックグラウンド状態のアプリケーションプロセスが存在する状態から既存プロセスを再利用して起動する方法である。ユーザが同じアプリケーションを再度使用しようとしたときの動作は、アプリケーションプロセスの状態により異なる。プロセスが無い状態であれば「新規起動」により起動され、バックグラウンド状態であれば「再開」により起動される。

Androidのアプリケーションを新規に起動する場合、次の手順に従って起動される。①ユーザがアプリケーションのアイコンをタッチするとホームアプリケーション(Launcher)がアプリケーション起動要

求のIntentをActivityManagerに送信する。②ActivityManagerがZygoteにプロセス生成要求を送信する。③Zygoteが自分自身をforkし、子プロセスを生成する。④新しいプロセスが初期化される。⑤アプリケーションのライフサイクルに従ってonCreate(), onStart(), onResume()が呼び出される。

また、起動済みであるがバックグラウンド状態にあるプロセスの再開は、次の手順に従って起動される。①ユーザがアプリケーションのアイコンにタッチするとホームアプリケーションが再開要求のIntentをActivityManagerに送信する。②ActivityManagerは対象のバックグラウンドアプリケーションプロセスに再開要求を出す。③バックグラウンドアプリケーションプロセスは再開要求を受けてアプリケーションプロセスとして起動しフォアグラウンド状態となる[2]。

起動済みプロセスがlow memory killerによって強制終了されることなく再利用された場合は、「再開」の手続きが行われるが、プロセスが強制終了された後に再利用された場合は「新規起動」の手続きが行われる。通常、「再開」よりも「新規起動」の方が要する時間が長いので、後に再利用するプロセスの強制終了はユーザの待ち時間を増加させてしまう。

## 3. アプリケーション起動履歴

4人の被験者の許可を得て、その被験者が起動したアプリケーションの履歴を入手した。

アプリケーションの起動履歴はlog1からlog7までであり、log1からlog7の各履歴の長さは順に436, 201, 95, 55, 43, 42, 30である。また、アプリケーション使用履歴は使用アプリケーションの変化時にのみ記録しており、1つのアプリケーションを連続して使用しても履歴においては1つとする。よって、同一アプリケーションが再度履歴に登場するまでに少なくとも1つの他のアプリケーションが履歴内に登場する必要がある。特定のアプリケーションの出現頻度は最高で50%となる。

## 4. 使用アプリケーションの種類

各履歴におけるアプリケーションごとの使用頻度(履歴における占有率)を図1に示す。

図1において、横軸x、縦軸yの点のプロットは、頻度順位が1位からx位のx個のアプリケーションの合計使用頻度がy%であることを意味している。

図より、多くのユーザにおいて、上位数個(4~7個)の使用頻度のアプリケーションにより全使用の

Improving Process Choice of Android Low Memory Killer using LRU

<sup>†</sup>Shun Nomura, Yuta Nakamura, Hirokazu Sakamoto, Kogakuin University Graduate School

<sup>‡</sup>Shintaro Hamanaka, Saneyasu Yamaguchi, Kogakuin University

80%を占めており、ユーザのアプリケーション起動には大きな偏りがあることがわかる。

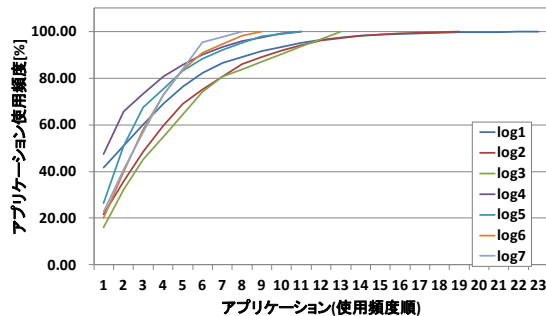


図 1 アプリケーションごとの使用頻度

### 5. アプリケーションの使用間隔

続いて、アプリケーション使用間隔に着目し、考察を行う。本稿では“アプリケーションの使用間隔”という語を「あるアプリケーションを使用して、次に同じアプリケーションを再度使用するまでに他のアプリケーションを何個使用しているか」の意味で用いる。アプリケーション起動履歴の  $x$  個目にアプリケーション A が登場し、次にアプリケーション A が登場するのが  $y$  個目であれば、今回のアプリケーション使用間隔は  $y-x-1$  となる。LRU は参照の時間的局所性を期待しており、最近使用されたアプリケーションが短い使用間隔で使用されると効果的に機能する。log1 におけるアプリケーション使用間隔の発生頻度とその積分値を図 2 に示す。

図より、アプリケーション使用間隔は短いことが多く、LRU が効果的に機能すると予想できる。また、累計(積分値)に着目すると、log1 においては、使用間隔 8 個で 80%を網羅しており、端末が過去 8 個分のプロセスを保持できればアプリケーション再開率 80%を達成できることが分かる。

### 6. アプリケーションの短期間内占有率

次に、特定のアプリケーションの短期間内占有率の時間変化について述べる。“短期間内占有率”とは、履歴内の連続する領域(10 アプリケーション)にて、あるアプリケーションが何回登場するかを表している。同一アプリケーションが 2 回連続して履歴に登場することがないため、登場回数は最高で 5 回であり、占有率は最高で 50%である。図 2 に log1 の①同履歴内で最も使用頻度の高いアプリケーション、②使用頻度の順位が 2 番であるアプリケーション、③使用頻度の順位が全体の中央であるアプリケーション、④使用頻度の順位が最下位であるアプリケーションの占有率の推移を示す。

図より、履歴全体で見ると使用頻度が高いアプリケーションであっても、占有率は時間とともに大きく変化し、占有率が 50%近くと非常に高くなる時間帯と 0%になる時間帯があることが分かる。すなわち、ユーザがあるアプリケーションを使用している時間帯と使用していない時間帯の二極化が生じてい

ることを意味しており、あるアプリケーションを使用している時間帯はそのアプリケーションを高確率で起動していることが分かる。これは、参照の時間的局所性(あるアプリケーションが使用されると、そのアプリケーションが近い将来に再度使用される確率が高い)に非常に近い現象であり、LRU が効果的に機能すると期待することができる。また時間帯ごとに使用される確率が高いアプリケーションが変化するため、LFU の様に重要視するアプリケーションが時間とともに変化しづらい手法は効果的に機能しないと予想される。

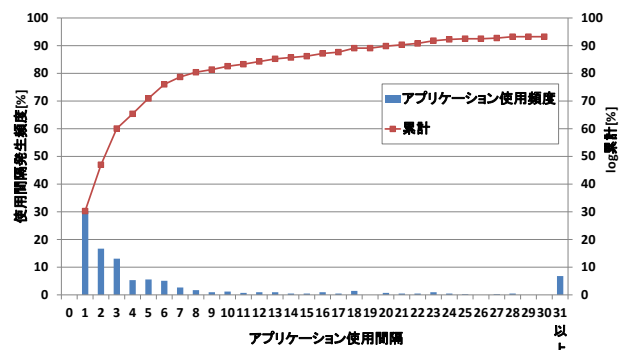


図 2 アプリケーション使用間隔の発生頻度(log1)

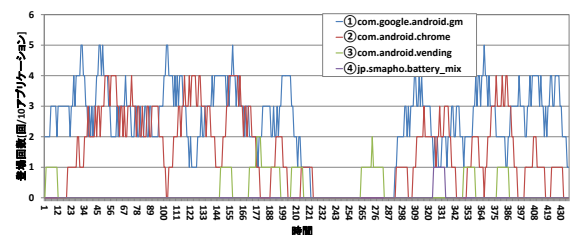


図 3 アプリケーションの占有率の時間変化(log1)

### 7. おわりに

本稿で我々は low memory killer における強制終了プロセスの選定手法に着目し、LRU を用いる手法の妥当性を実際のユーザのアプリケーション起動履歴を用いて考察した。考察の結果、実際のアプリケーションの起動には強い偏りがあることがわかり、LRU に適していることが分かった。

今後は、端末の種類、使用用途、職業などによるアプリケーション起動の傾向の違いについて考察をしていく予定である。

### 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。

- 1) 野村 駿, 中村 優太, 坂本 寛和, 山口 実靖 : Android における LRU を用いた終了メモリ選定, 第 10 回コンシューマ・デバイス & システム(CDS)研究発表会, CDS10-7
- 2) Kyosuke Nagata, Saneyasu Yamaguchi, :An Android Application Launch Analyzing System , 8th ICCM: 2012 International Conference on Computing Technology and Information Management