3W-7

Androidにおけるアプリケーション起動時間の短縮に関する一考察

永田　恭輔†　　服部　拓也†　　中村　優太‡　　野村　駿‡　　山口　実靖‡

†工学院大学大学院工学研究科電気・電子工学専攻　　‡工学院大学工学部情報通信工学科

1. はじめに

Android は登場以来の世界シェアを増加させ続けており，2012 年には 68 パーセントを超えている[1]．また、スマートフォン，タブレット PC，音楽プレイヤーなど，様々なデバイスで採用されており，その重要性が年々増加している．Android において，アプリケーションの起動時間はユーザーにとって重要な性能の一つである．Android には Zygote と呼ばれる特殊なプロセスがあり，アプリケーション起動時間の短縮のための工夫がなされている．

本稿では，Android アプリケーションの起動時間短縮のための方法として Zygote による事前読み込みクラス数の増加を紹介し，その効果について考察する．

2. Zygote

アプリケーション起動のオーバーヘッドを減少させるために，新しいアプリケーションプロセスは Zygote と呼ばれる特殊なプロセスから fork される．いくつかの重要なクラスファイルは，多くのアプリケーションから読み込まれるため，それぞれのアプリケーションが個別に読み込みを行っていると効率が悪いと言える．これを改善するために Zygote プロセスは重要なクラスファイルを既にロードしており（preload しており），新しいアプリケーションプロセスはこの Zygote からフォークされる．よって，アプリケーションプロセスは重要クラスファイルをロード済みの状態で起動される．これにより，アプリケーション起動時間の短縮が実現されている．

また，アプリケーションの fork は CoW (Copy on Write)で行われるため，プロセスが生成された時点では親子のプロセスでメモリを共有しており，消費メモリ量が増加しない．通常は読み込まれたクラスファイルへの書き込みは行われないため，CoW でコピーされた領域はプロセスの終了までコピーされることはなく，メモリ使用量の削減も実現されている．

Zygote が preload するクラスファイルはフレームワーク内の preloaded-classes にて規定されており，Android 4.1.1 では 2303 個のクラスが Zygote により preload されている．

3. アプリケーション起動方法

アプリケーションの起動方法には 2 通りあり，どちらの方法が適用されるかは，プロセスの状態により決まる[2]．一つ目の方法は起動対象のプロセスが存在しない場合に，新しくプロセスを生成する方法である．2 つ目の方法は起動対象のアプリケーションプロセスがバックグラウンドプロセスとして既に存在している場合に，フォアグラウンドプロセスとして再開する方法である．本稿では，前者を「新規起動」，後者を「再開」と呼ぶ．

4. アプリケーション起動時間の短縮

4.1. 標準クラス先読み機能の拡張

第 2 章で述べたように，Android OS には Zygote プロセスが存在し，このプロセスが多くのクラスファイルを読み込み済みの状態で待機している．この Zygote プロセスが preload するクラスの数を増やすことにより，プロセス起動時間のさらなる短縮が実現できると考えられる．本節では preload されるクラスの数とアプリケーション起動時間の関係について考察する．

図 1 に，Zygote が preload するクラスの量と起動時間の関係を示す．図内の“without_preload”は preload されるクラスを無くした場合，“normal_preload”は Android 4.1.1 の標準状態(2303 個)の場合，“+51_preload”は標準状態に加え dalvik 以下にあるすべてのクラス(51 個)を preload した場合，“+120_ preload”は 51_preload に加えさらに java.net 以下にあるすべてのクラス(合計 120 個)を preload した場合，“+281_preload”はさらに java.security 以下のすべてのクラス(合計 281 個)を preload した場合の測定である．測定は「新規起動」により行い，キャッシュが機能している状態でアプリケーションの起動を行った．

“A Study on Improving Application Launching Time in Android”

†Kyosuke Nagata, Takuya Hattori, Electrical Engineering and Electronics, Kogakuin University Graduate School

‡Yuta Nakamura, Shun Nomura, Saneyasu Yamaguchi, Department of information and Communications Engineering, Kogakuin University

図内の起動時間は，ユーザーによるアプリケーションアイコンのタッチからアプリケーションライフサイクルの onResume()の呼び出しまでの時間である．

図より，Zygote が preload を全く行わないとアプリケーション起動時間が大幅に増加してしまうことが確認できる．よって，Zygote による preload はアプリケーション起動時間の短縮に効果があると言える．また，標準より多くのクラスを preload させることにより起動時間のさらなる短縮が可能であることがわかる．

次に，我々が個別に選択した 100 個のクラスと，我々が選択した 300 個のクラスを preload させた Zygote による実験を行った．結果を図 2 に示す．この図からも，Zygote に preload させるクラスの数の増加により，アプリケーションの起動時間のさらなる短縮が可能であることが分かる．

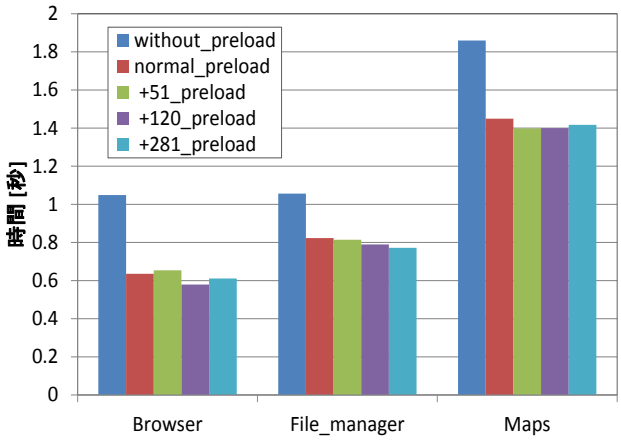

図 1. preload クラス数と起動時間の関係(A)

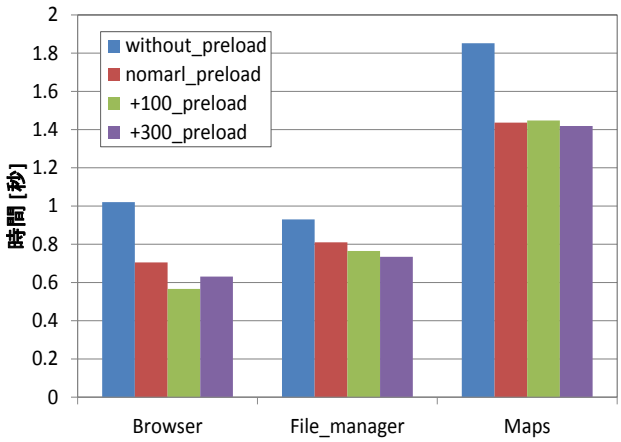

図 2. preload クラス数と起動時間の関係(B)

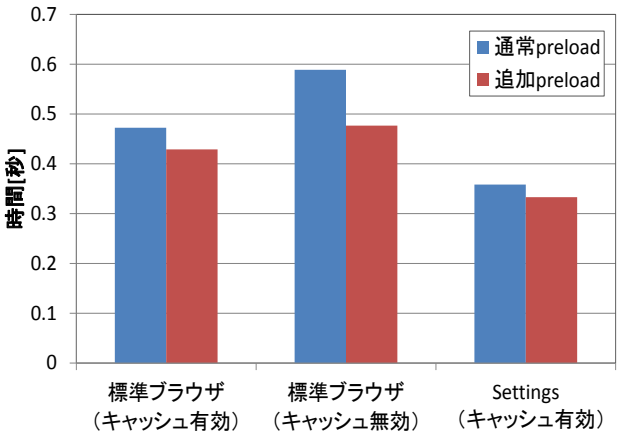

図 3. アプリケーション固有クラスの preload

4.2. アプリケーション固有クラスの先読み

次に，アプリケーション固有のクラスを Android の framework に組み込み，Zygote の preload クラスに追加することによる起動時間を短縮させる手法について考察する．標準ブラウザと Settings の 2 種類のアプリケーションについて，それらのアプリケーションの固有クラスを Zygote に preload させ起動時間を評価した．標準ブラウザに関してはキャッシュ有効時とキャッシュ無効時で比較を行い，Setting に関してはキャッシュ有効の状態で評価を行った．結果を図 3 に示す．図より，アプリケーション固有のクラスを Zygote に組み込むことにより起動時間のさらなる短縮が可能であることが確認され，当該クラスがキャッシュ内に格納されていないときに特に効果が大きいことが確認された．

5. おわりに

本稿で我々は，アプリケーション起動時間の短縮方法として，Zygote による読み込みクラス数を増加させる手法を紹介した．そして実験により，この手法に効果があることを示した．

今後は，Zygote が preload するクラスの選定に関する考察をしていく予定である．

謝辞

本研究は科研費(22700039，24300034)の助成を受けたものである．

参考文献

1) Android and iOS Surge to New Smartphone OS Record in Second Quarter, According to IDC : http://www.idc.com/getdoc.jsp?containerId=prUS23638712

2) Kyosuke Nagata, Saneyasu Yamaguchi, "An Android Application Launch Analyzing System ," 8th ICCM: 2012 International Conference on Computing Technology and Information Management, 2012