

奨励金No.1482

# Mobile App コードの進化を包容する グリーンマイニング基盤の構築

亀井 靖高

九州大学大学院 システム情報科学研究院 准教授

## Development of Green Mining Infrastructure for Embracing the Evolution of Mobile App code

Yasutaka Kamei,

Kyushu University, Faculty of Information Science and Electrical Engineering, Associate Professor



本研究課題では、頻繁に更新される Mobile App コードの進化を包容する、ロバストかつライトウェイトなグリーンマイニング基盤を開発する。その実現に向け、バージョン間における同一機能特定方法を提案した。また、自動実行される Mobile App コードからログを計測して、その計測ログから近似的に消費電力を推定する方法を開発した。本マイニング基盤を用いることで、よりソフトウェア開発の初期段階における消費電力削減ノウハウの発見・普及、さらにはひとつひとつの Mobile App の消費電力の削減に繋がることが期待できる。

The goal of this research project is to develop a robust and lightweight green mining platform that encompasses the evolution of frequently updated mobile application code. To achieve this, we proposed an approach for identifying identical functionality between versions. Additionally, we proposed an approach for estimating power consumption by measuring logs of automatically executed mobile application code. By using this green mining platform, it is expected that knowledge about power reduction can be discovered and disseminated at an earlier stage of software development, leading to a reduction in power consumption for each mobile application.

### 1. 研究内容

#### 1.1 はじめに

昨今、Android や iOS の登場を皮切りに、モバイル端末が爆発的に普及し、今や我々の生活に欠かせないものとなっている。例えば、Android OS が搭載されたスマートフォンは、2021年5月の時点で、世界で30億人以上ものアクティブユーザーによって利用されている。そのため、モバイルアプリ（Mobile App: スマートフォン向けに開発されたアプリケーションソフトウェア）のひとつひとつが消費する電力は小さいものの、巨視的観点から見ると、電力をわずかにでも浪費する処理が世界全体へ及ぼす影響は無視できない大きさである。

本研究の目的は、頻繁に更新される Mobile App

コードの進化を包容する、ロバストかつライトウェイトなグリーンマイニング基盤の開発である。その実現に向け、大きく2つのサブテーマを実施した。サブテーマ1では、バージョン間における同一機能特定方法の開発（よりロバストに）を目指す。具体的には、ある機能を呼び出す手順（例：どこのボタンを押すか）を、リビジョンごとに自動で特定しトレースする方法を開発した。サブテーマ2の消費電力計測方法の開発（よりライトウェイトに）では、サブテーマ1で自動実行される Mobile App コードからログを計測し、その計測ログから近似的に消費電力を推定する方法を開発した。

## 1.2 要素技術

### 1.2.1 Android 関連のツール

Android Profiler [1] と Systrace [2]、Batterystats [3] を併用して、CPU、メモリ、ネットワーク、バッテリーリソースの使用状況に関するデータをリアルタイムで収集する。これらの Android ツールは、アプリケーションの電力消費量を直接測定するものではなく、サブテーマ2で行う消費電力を推定するモデルの入力として用いる。

### 1.2.2 Instrumented Tests

Instrumented Tests は、実行される Mobile App のコンテキストや Android フレームワークの API にアクセスできるテスト実行環境を利用して、実機上やエミュレータ上で実行できるテストである。開発者は、テストコードを記述することで、テスト対象の Mobile App に対して任意の操作を実行できる。

Instrumented Tests は操作をテストコードとして記述でき、常に同じイベントを発生させるため、Mobile App を実行した際の実行トレースを一貫して再現することが可能である。本研究課題では、Instrumented Tests を使用して、消費電力の測定対象となるメソッドを呼び出し、これ

らのメソッドの消費電力量を調査する。さらに、Instrumented Tests を経由して実行トレースとソースコードを関連づけることで、ユースケースごとの消費電力量を推定することも可能となる。

## 1.3 グリーンマイニング基盤

### 1.3.1 概要

本研究課題では、Instrumented Tests を用いて任意のコードを実行し、Android Profiler、Systrace、Batterystats を収集することで、サブテーマ1、及び、サブテーマ2の方法を実装したグリーンマイニング基盤を開発した。本グリーンマイニング基盤では、任意のコードを指定することや、メソッドレベルでの一貫した実行トレースの生成、さらには計測ログから近似的に消費電力を推定することで、Mobile App に対する消費電力の大規模分析を可能とする。

図1は、本研究課題で開発したグリーンマイニング基盤の全体図（アーキテクチャ）を示す。本基盤は、3つの主要なコンポーネントによって構成される。1つ目はAPK生成機構（APK：Android Application Package の略称で、Android 上で動作するソフトウェアパッケージの形式）で、2つ目は実行トレース等のログ生成機構で、3つ目は消

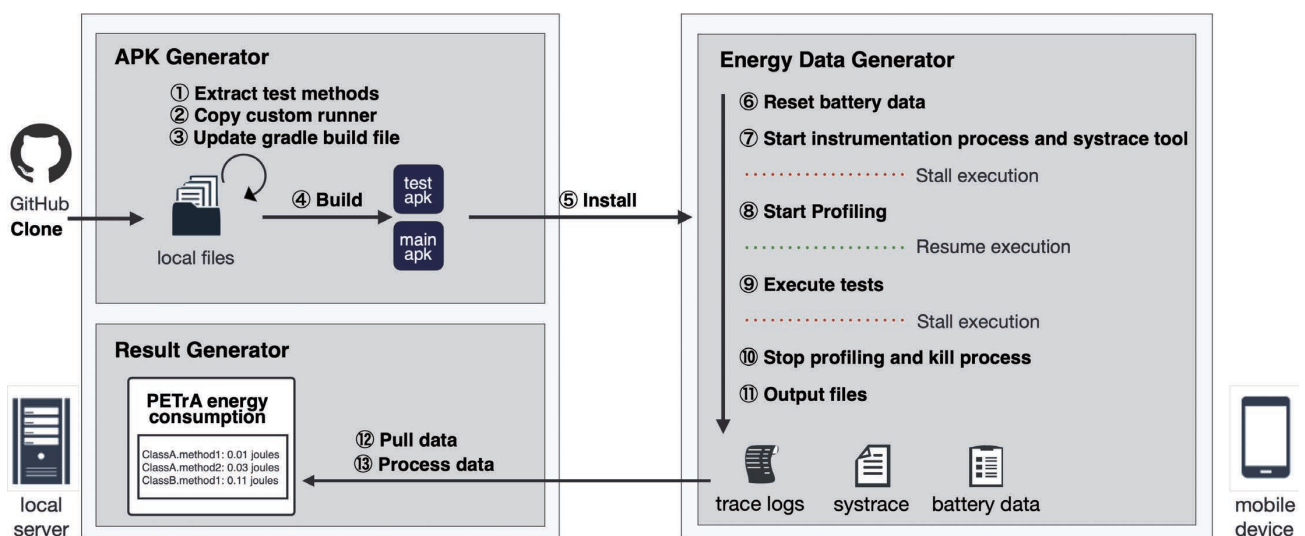


図1. グリーンマイニング基盤の全体図（Nourry2022ICSME より抜粋）

費電力量の推定機構である。なお、各機構の詳細な説明は、ICSME 2022 で発表した論文にて記載されている。

### 1.3.2 APK 生成機構

本グリーンマイニング基盤では、消費電力を推定するためのデータ（計測ログ）を取得するために Android Profiler を使用する。しかしながら、Android Profiler を Mobile App 上から直接実行することができない。そのため、プロファイルプロセスを開始／停止するためには、外部（ローカルサーバ）から開始／停止の操作を呼び出す必要がある。プロファイルプロセスの自動化を目指して、本研究課題では、APK（主にテスト用 APK）を改良することで、ローカルサーバに操作の呼び出しを送るタイミングを通知する。そして、改良されたテスト APK とメイン APK（Mobile App 本体）をモバイルデバイス等の実行環境に自動でインストールする。

### 1.3.3 消費電力推定のためのログ生成機構

androidTest フォルダ内の計測対象コードを含むソースコードファイルに対して AST パーサを実行し、すべての計測テストメソッドのシグニチャを抽出する。ユースケース、及び、実行トレースやソースコードの関連づけを可能にするためである。その後、trace logs、systrace、battery data (batterystats) を取得する。

### 1.3.4 消費電力の推定機構

batterystats には、計測中のモバイルデバイス上のバッテリー使用量に関連する情報が含まれている。Trace logs と systrace の出力ファイルを組み合わせ、PETrA [4] の消費電力量推定のアルゴリズムを用いて、メソッドレベルでの消費電力量の近似値を取得する。

## 1.4 実験結果

### 1.4.1 スケーラビリティ

スケーラビリティに関する評価を行うため、オープンソースの Mobile App である KISS プロジェクト [5] を対象に実験を行った。本プロジェクトを採用した理由は、ビルドシステムとして gradle を用いており、過去のリビジョンに遡ってビルドを容易にできたためである。401 リビジョンに対して本グリーンマイニング基盤を用いた結果、1つのリビジョンをプロファイルするために要した時間は平均で7分程度であった。また、再現性を評価するために、ある1つのリビジョンに対して、10回分のプロファイルを実施した。APK に含まれる16つのテストそれぞれについてプロファイルされたメソッドの数を調べたところ、いくつかの小さな不一致を伴うものの、実行全体としては非常に類似した結果を生成することができた。

### 1.4.2 従来手法との比較（ランダム実行との比較）

本グリーンマイニング基盤の特徴の1つが、特定の機能に狙いを絞った上での消費電力の測定である。従来の操作の呼び出し方法の1つである Monkey を用いることでも、特定の機能が実行されることもある可能性がある。そのため、どの程度の割合で Monkey によって特定の機能が実行されるかの実験を行った。本実験では、Mobile App として、AnyMemo [6] を用いた。本 Mobile App では、ユーザが単語などを記憶するためにカードを作成できる。本実験では、この MobileApp の最も一般的な使用例の1つである「新しいカードの作成」の消費電力量を測定する。Monkey によって、カード作成メニューにアクセスし、新しいカードを作成するために複数のユーザインタフェースをナビゲートできるかどうかを確認したところ、3万回以上のインタラクションを生成させたとしても、Monkey では1枚も新しいカードを作成できないこと（つまり、特定機能を絞ることができないこと）がわかった。

## 1.5 おわりに

近年、SDGs（持続可能な開発目標）が世界で謳われ、その目標の一つに地球温暖化を始めとする気候変動に対する行動が求められている。我が国でも2030年までの二酸化炭素排出量削減目標を2013年度比46%減とする目標を掲げており、省エネルギーによる二酸化炭素削減は急務な課題のひとつである。本研究が対象とするMobile Appは、一般家庭のみならず、オフィスでも広く利用される。ひとつひとつのアプリの電力消費は小さいものの、我が国および世界レベルで見た際に、消費電力は膨大な量となりうる。本研究課題で開発したグリーンマイニング基盤を活用することで、製品の内側（つまりソフトウェア）より社会課題の解決への貢献が期待できる。

## 参考文献

- [1] Android Profiler. <https://developer.android.com/studio/profile/android-profiler>.
- [2] Capture a system trace on the command line; android developers. [Online]. Available: <https://developer.android.com/topic/performance/tracing/command-line>
- [3] Profile battery usage with batterystats and battery historian, android developers. [Online]. Available: <https://developer.android.com/topic/performance/power/setup-battery-historian>
- [4] D. Di Nucci, F. Palomba, A. Prota, A. Panichella, A. Zaidman, and A. De Lucia, "Petra: A software-based tool for estimating the energy profile of android applications," in Proceedings of the 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C), 2017, pp. 3–6.
- [5] Kiss. [Online]. Available: <https://github.com/Neamar/KISS>
- [6] Anymemo. [Online]. Available: <https://github.com/helloworld1/AnyMemo>

## 2. 発表（研究成果の発表）

Olivier Nourry, Yutaro Kashiwa, Bin Lin, Michele Lanza, Gabriele Bavota, and Yasutaka Kamei "AIP: Scalable and Reproducible Execution Traces in Energy Studies on Mobile Devices," In Proceedings of In Proceedings of the International Conference on Software Maintenance and Evolution (ICSME), Tool Demo Track, pp. 449–453, October 2022. (Limassol, Cyprus).

招待講演、SES2022にて：Olivier Nourry, Yutaro Kashiwa, Yasutaka Kamei, and Naoyasu Ubayashi, "Does Shortening the Release Cycle Affect Refactoring Activities: A Case Study of the JDTCore, Platform SWT, and UI projects," Information and Software Technology, Vol. 139, Article number: 106623.