An Empirical Study of the Energy Consumption of Android Applications

Ding Li, Shuai Hao, Jiaping Gui, William G.J. Halfond
Department of Computer Science
University of Southern California

This work was supported in part by the National Science Foundation under Grant No. CCF-1321141 to the University of Southern California
Battery Life Is a Critical Problem

Battery life is important to the user experience
Battery Life Is a Critical Problem

Battery life is important to the user experience

Many apps drain the battery quickly
Battery Life Is a Critical Problem

Battery life is important to the user experience

Many apps drain the battery quickly

How to create energy efficient apps?
How to Create Energy Efficient Apps?
Software Engineering Oriented Information

Which part is most energy consuming?
Which part should be optimized?
How to Create Energy Efficient Apps?

Measurement of Energy Consumption

- Granularity
- Noise
- Isolation
Other Works on Energy

• Energy estimation and measurement
  – E.g., eLens [Hao et al. ICSE 2013]
  – Do not directly provide information
• Techniques for energy optimization
  – E.g., No-sleep bugs [Pathak et al. MobiSys 2012]
  – Focus only on a particular problem

There are no empirical studies on large numbers of apps
Our Study

• Source line level measurement study on 405 Android market apps

• 7 research questions on how energy is consumed in apps
  – To provide software engineering practitioner oriented energy information

• 3 research questions on how to do energy measurement
Research Questions: How Energy Is Consumed in Apps

- RQ 1: App Energy
  - Idle State Energy
  - Non-Idle State Energy

- RQ 2: Component Level
  - API Energy
  - User Code Energy

- RQ 3: API Level
- RQ 4: Structure Level
- RQ 5: Bytecode Level
Research Questions: 
How to Measure Energy

- RQ 8: Proxy Measurement
- RQ 9: Measurement Granularity
- RQ 10: Handling Idle State Energy
Experiment Protocol

• Hardware
  – Samsung Galaxy SII smart phone
  – With Android 4.3

• Energy measurement tool
  – Monsoon

• Source line level measurement
  – vLens [Li et al. ISSTA 2013]

• Automate the UI interaction
  – Monkey
  – 5 random events per second, 500 in total

• 405/412 apps with code coverage higher than 50%
  – No game apps
Distribution of App Types

- Lifestyle & Productivity (LP): 19%
- Entertainment (En): 19%
- Travel & Transportation (TT): 12%
- Music & Media (MM): 11%
- Health & Medical (HM): 10%
- Sports & News (SN): 8%
- Photography (Ph): 7%
- Utilities & Tools (To): 7%
- Others: 7%

Legend:
- Lifestyle & Productivity (LP)
- Entertainment (En)
- Travel & Transportation (TT)
- Music & Media (MM)
- Health & Medical (HM)
- Sports & News (SN)
- Photography (Ph)
- Utilities & Tools (To)
- Others
RQ 1: How Much Energy Is Consumed by Individual Applications?

App Energy

Idle State Energy

Non-Idle State Energy

API Energy

User Code Energy

Component Level

API Level

Structure Level

Bytecode Level
RQ 1: How Much Energy Is Consumed by Individual Applications?

Energy consumption of different applications
RQ 1: How Much Energy Is Consumed by Individual Applications?

Energy consumption of different applications
RQ 1: How Much Energy Is Consumed by Individual Applications?

Energy consumption of different applications

Average = 57,977 mJ,
Standard deviation = 62,416 mJ
RQ 1: How Much Energy Is Consumed by Individual Applications?

Energy consumption of different applications

Average energy 30% differ
RQ 1: How Much Energy Is Consumed by Individual Applications?

Energy consumption of different applications

Energy differs for more than 100 times
RQ 1: How Much Energy Is Consumed by Individual Applications?

Energy consumption of different applications

Variance within category is larger than across categories
RQ 2: How Much Energy Is Consumed by the Idle State of An Application?
RQ 2: How Much Energy Is Consumed by the Idle State of An Application?

Breakdown of app energy:
- Code running, 38%
- Wait for input, 37%
- Sleep, 25%
RQ 2: How Much Energy Is Consumed by the Idle State of An Application?

Breakdown of app energy:
- Wait for input: 37%
- Code running: 38%
- Sleep: 25%

Idle-states, no code is running.
RQ 2: How Much Energy Is Consumed by the Idle State of An Application?

Breakdown of app energy

- Code running, 38%
- Wait for input, 37%
- Sleep, 25%
- 37% energy

Only optimizing code is insufficient, idle-state energy also needs to be optimized

62% energy
An Example of How to Reduce Idle State Energy

• Display energy could be saved in idle states
• Using energy efficient color designs
  – Nyx [Li et al. ICSE 2014]
  – Chameleon [Dong et al. Mobisys 2011]
An Example of How to Reduce Idle State Energy

Save 40% energy [Li et al. ICSE 2014]
RQ 3: Which Code Consumes More Energy: System APIs or Developer Written Code?

Diagram:
- App Energy
  - Idle State Energy
  - Non-Idle State Energy
- API Energy
  - Component Level
  - API Level
- User Code Energy
  - Structure Level
  - Bytecode Level
RQ 3: Which Code Consumes More Energy: System APIs or Developer Written Code?

Breakdown of non-idle energy:
- API: 85%
- Bytecode: 2%
- Outliers: 13%
RQ 3: Which Code Consumes More Energy: System APIs or Developer Written Code?

Breakdown of non-idle energy

- System APIs from the Android SDK: 85%
- Outliers: 13%
- Bytecode: 2%

System APIs from
the Android SDK

API
Bytecode
Outliers
RQ 3: Which Code Consumes More Energy: System APIs or Developer Written Code?

Breakdown of non-idle energy:
- 85% for API
- 13% for Outliers
- 2% for Bytecode

Normal user code
RQ 3: Which Code Consumes More Energy: System APIs or Developer Written Code?

Breakdown of non-idle energy:
- API: 85%
- Bytecode: 2%
- Outliers: 13%

Garbage collection and thread switching:

School of Engineering
RQ 3: Which Code Consumes More Energy: System APIs or Developer Written Code?

Breakdown of non-idle energy

Developer written code does not consume a significant amount of energy.
RQ 4: How Much Energy Is Consumed by the Different Components of A Smartphone?
RQ 4: How Much Energy Is Consumed by the Different Components of A Smartphone?

![Bar chart showing the percentage of non-idle energy consumed by different components.]

- **UI**: Around 5%
- **Net**: Around 42%
- **IO**: Around 3%
- **Sqlite**: Around 9%
- **Camera**: Around 17%
- **Location**: Around 4%
- **Sensor**: Around 1%
- **Media**: Around 4%

**Average ratio of the energy consumption of a component to the non-idle energy of apps**
RQ 4: How Much Energy Is Consumed by the Different Components of A Smartphone?

Packages of APIs in SDK

- UI
- Net
- IO
- Sqlite
- Camera
- Location
- Sensor
- Media

Percentage of non-idle energy (%)

Average ratio of the energy consumption of a component to the non-idle energy of apps
RQ 4: How Much Energy Is Consumed by the Different Components of A Smartphone?

Percentage of non-idle energy (%)

- UI
- Net
- IO
- Sqlite
- Camera
- Location
- Sensor
- Media

Average ratio of the energy consumption of a component to the non-idle energy of apps.
RQ 4: How Much Energy Is Consumed by the Different Components of A Smartphone?

Calculated over the apps that used the network

Average ratio of the energy consumption of a component to the non-idle energy of apps

Percentage of non-idle energy (%)
RQ 4: How Much Energy Is Consumed by the Different Components of a Smartphone?

Average ratio of the energy consumption of a component to the non-idle energy of apps

- Net: 80%, HTTP requests
- Camera: 15%
- UI: 5%
- IO: 5%
- Sqlite: 10%
- Location: 5%
- Sensor: 1%
- Media: 1%

Percentage of non-idle energy (%)

37
RQ 4: How Much Energy Is Consumed by the Different Components of A Smartphone?

These components may still dominate the energy of a particular app.

Average ratio of the energy consumption of a component to the non-idle energy of apps.

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage of non-idle energy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI</td>
<td>5</td>
</tr>
<tr>
<td>Net</td>
<td>35</td>
</tr>
<tr>
<td>IO</td>
<td>3</td>
</tr>
<tr>
<td>Sqlite</td>
<td>8</td>
</tr>
<tr>
<td>Camera</td>
<td>20</td>
</tr>
<tr>
<td>Location</td>
<td>5</td>
</tr>
<tr>
<td>Sensor</td>
<td>1</td>
</tr>
<tr>
<td>Media</td>
<td>2</td>
</tr>
</tbody>
</table>
RQ 4: How Much Energy Is Consumed by the Different Components of A Smartphone?

Network is generally the most energy consuming component, but other components may also dominate the energy consumption of an app.
RQ 5: Which APIs Are Significant in Terms of Energy Consumption?
RQ 5: Which APIs Are Significant in Terms of Energy Consumption?

• Distribution of energy consumption over APIs
  – Across apps
  – Within an app
• Similarity across apps of top 10 most energy consuming APIs
• Frequency of APIs being in the top 10 most energy consuming APIs
RQ 5: Which APIs Are Significant in Terms of Energy Consumption?

• Distribution of energy consumption over APIs
  – Across apps
RQ 5: Which APIs Are Significant in Terms of Energy Consumption?

Average ratio of energy consumed by an API to the non-idle state energy across apps
RQ 5: Which APIs Are Significant in Terms of Energy Consumption?

Average ratio of energy consumed by an API to the non-idle state energy across apps

98% of APIs
RQ 5: Which APIs Are Significant in Terms of Energy Consumption?

Average ratio of energy consumed by an API to the non-idle state energy across apps

HttpClient.execute
RQ 5: Which APIs Are Significant in Terms of Energy Consumption?

Average ratio of energy consumed by an API to the non-idle state energy across apps

Most APIs are not significant in energy consumption.
RQ 5: Which APIs Are Significant in Terms of Energy Consumption?

• Distribution of energy consumption over APIs
  – Across apps
  – Within an app
RQ 5: Which APIs Are Significant in Terms of Energy Consumption?

The ratio of the energy consumption of an app’s top 10 most energy consuming APIs to its total API energy consumption.
RQ 5: Which APIs Are Significant in Terms of Energy Consumption?

The ratio of the energy consumption of an app’s top 10 most energy consuming APIs to its total API energy consumption.

For 91% of apps, the top 10 APIs consume more energy than all other APIs.

Energy is concentrated in the top 10 most energy consuming APIs.
RQ 5: Which APIs Are Significant in Terms of Energy Consumption?

• Distribution of energy consumption over APIs
  – Across apps
  – Within an app

• Similarity across apps of top 10 most energy consuming APIs
RQ 5: Which APIs Are Significant in Terms of Energy Consumption?

Top 10 most energy consuming APIs of app A

Top 10 most energy consuming APIs of app B
RQ 5: Which APIs Are Significant in Terms of Energy Consumption?

In general, the overlap is 1

Top 10 most energy consuming APIs of app A

Top 10 most energy consuming APIs of app B

There is very little similarity among the top 10 most energy consuming APIs
RQ 5: Which APIs Are Significant in Terms of Energy Consumption?

• Distribution of energy consumption over APIs
  – Across apps
  – Within an app

• Similarity across apps of top 10 most energy consuming APIs

• Frequency of APIs being in the top 10 most energy consuming APIs
RQ 5: Which APIs Are Significant in Terms of Energy Consumption?

Number of times each API is among the top 10 most energy consuming APIs for an app
RQ 5: Which APIs Are Significant in Terms of Energy Consumption?

Number of times each API is among the top 10 most energy consuming APIs for an app
RQ 5: Which APIs Are Significant in Terms of Energy Consumption?

Number of times each API is among the top 10 most energy consuming APIs for an app.
RQ 5: Which APIs Are Significant in Terms of Energy Consumption?

HTTP requests are most likely to be most energy consuming.
RQ 6: How Much Energy Is Consumed by Code in Loops?
RQ 6: How Much Energy Is Consumed by Code in Loops?

Average ratio of energy consumption of loops to non-idle energy

On average, loops consume 41% of non-idle state energy
RQ 7: How Much Energy Is Consumed by the Different Types of Bytecodes

Diagram:

- App Energy
  - Idle State Energy
  - Non-Idle State Energy
  - API Energy
    - Component Level
    - API Level
    - Structure Level
    - Bytecode Level
  - User Code Energy
RQ 7: How Much Energy Is Consumed by the Different Types of Bytecodes

Data manipulating instructions are the most energy consuming.
How Energy Is Consumed in Apps

• RQ 1: App energy varies significantly
• RQ 2: Idle states consume more energy than code
• RQ 3: APIs dominate the non-idle state energy
• RQ 4: Network is the most energy consuming component
• RQ 5: Only a few APIs are significant in energy consumption
• RQ 6: Loops may consume a lot of energy
• RQ 7: Data manipulation instructions are more energy consuming than other bytecodes
How to Measure the Energy

• RQ 8: Is time equal to energy?
  – On average, for 4.6 of the top 10 most energy consuming APIs ranking by time is correct

• RQ 9: What granularity of measurement is sufficient?
  – Using millisecond level instead of nanosecond level measurements can introduce a 64% error, on average

• RQ 10: Is it necessary to account for idle state energy?
  – Not accounting for idle state energy introduces a 36% measurement error, on average
Threats to Validity

• Generalizability (External Validity)
  – 405 apps from 23 categories
  – All have code coverage higher than 50%

• Accuracy of measurement (Internal Validity)
  – We have 19% estimation error for bytecode energy
  – Not large enough to affect our conclusions

• Bytecode mismatch (Construct Validity)
  – We measured the energy of JVM bytecodes
  – JVM bytecodes are matched to DVM bytecodes
Summary

• A field study of 405 Android market apps
  – With source line level measurement
  – Apps from 23 categories

• We answer 10 research questions and provide actionable information to developers
An Empirical Study of the Energy Consumption of Android Applications

Presented by
Ding Li
Department of Computer Science
University of Southern California
public class App {

    public static void main(String[] args) throws IOException, DbxException {
        final String APP_KEY = "xxxxx"; // change to your own appkey
        final String APP_SECRET = "xxxxx"; // change to your own secret
        DbxAppInfo appInfo = new DbxAppInfo(APP_KEY, APP_SECRET);

        DbxRequestConfig config = new DbxRequestConfig("JavaTutorial/1.0",
                Locale.getDefault().toLanguageTag());
        DbxWebAuthNoRedirect webAuth = new DbxWebAuthNoRedirect(config, appInfo);

        // Have the user sign in and authorize your app.
        String authorizeUrl = webAuth.start();
        System.out.println("1. Go to: " + authorizeUrl);
        System.out.println("2. Click Allow (you might have to log in first)\n); System.out.println("3. Copy the authorization code.");

        String code = new BufferedReader(new InputStreamReader(System.in)).readLine().trim();

        // This will fail if the user enters an invalid authorization code.
        DbxAuthFinish authFinish = webAuth.finish(code);
        System.out.println(authFinish.accessToken);
        DbxClient client = new DbxClient(config, authFinish.accessToken);

        System.out.println("Linked account: " + client.getAccountInfo().displayName);
    }
}
public class App {

    public static void main(String[] args) throws IOException, DbxException {
        final String APP_KEY = "xxxxx"; // change to your own appkey
        final String APP_SECRET = "xxxxx"; // change to your own secret
        DbxAppInfo appInfo = new DbxAppInfo(APP_KEY, APP_SECRET);

        DbxRequestConfig config = new DbxRequestConfig("JavaTutorial/1.0",
                Locale.getDefault().toString());
        DbxWebAuthNoRedirect webAuth = new DbxWebAuthNoRedirect(config, appInfo);

        // Have the user sign in and authorize your app.
        String authorizeUrl = webAuth.start();
        System.out.println("1. Go to: " + authorizeUrl);
        System.out.println("2. Click Allow (you might have to log in first)");
        System.out.println("3. Copy the authorization code.");

        String code = new BufferedReader(new InputStreamReader(System.in)).readLine().trim();

        // This will fail if the user enters an invalid authorization code.
        DbxAuthFinish authFinish = webAuth.finish(code);
        System.out.println(authFinish.accessToken);
        DbxClient client = new DbxClient(config, authFinish.accessToken);

        System.out.println("Linked account: " + client.getAccountInfo().displayName);
    }
}
Software Components and Hardware Components

Software components (APIs)

- UI
- Net
- Camera

Energy from all hardware components

- Screen
- CPU
- 4G net card
- GPS sensors
- Camera