Calculating Source Line Level Energy Information

- Developers lack fine-grained energy feedback
- Other approaches have critical limitations
- We built a tool for fine-grained energy measurement
  - Co-authored by Shuai Hao, William G. J. Halfond and Ramesh Govindan
Direct Measurement

Measure with power meters

- Does not sample fast enough
  - 10 KHz vs. 1GHz
- Cannot detect runtime events
  - Thread Switching
  - Garbage Collection (GC)

Does not provide source line level information
Other Approaches

• Cycle-level simulators
  – Cannot run market apps realistically

• Model on OS features
  – Does not provide source line level granularity

• Estimate energy with models and runtime information, for example, eLens
  – Building a good model is expensive
Architecture of vLens

Runtime Measurement Phase

- App
- App Instrumenter
- Power Measurement Platform
- Insufficient data?
- App'

Offline Analysis Phase

- Analyzer
- Path Adjuster
- Energy

Test Cases

Annotated Code

Energy Report
1. App Instrumenter

- Add probes to record runtime information

- Record the execution path
  - Efficient Path Profiling [Ball & Larus]

- The entry and exit time of APIs
  - Instrument APIs
2. Power Measurement Platform

Run apps and collect energy and path information

- Monsoon power meter
- Samsung galaxy SII
3. Path Adjuster

Handle problematic energy costs prior to regression analysis

- Calculate energy of API calls
- Assign tail energy to corresponding API calls
  - Tail Energy: energy caused by an API but expended after return of the API
- Distribute energy among concurrent internal threads
3. Path Adjuster: API Energy Calculation

Calculate and remove non-linear API cost

Sum up energy between entry and exit time stamps

\[ t_2 - t_1 / t_4 - t_3 \ E_{t_3, t_4} \]

Minimal sampling interval
Assign tail energy to corresponding API

\[ TE_{API_1} = T_{s(API_2)} - T_{E(API_1)} / T_{tail} \]
3. Path Adjuster: Internal Multi-thread

Split energy among concurrent internal threads

\[ E_{\downarrow T1} = E_{\downarrow t1,t2} + \frac{1}{2} E_{\downarrow t2,t3} + \frac{1}{3} E_{\downarrow t3,t4} \]
4. Analyzer

Calculate source line level energy information

- Distribute energy to bytecodes
  - Use robust regression techniques
- Eliminate GC and external thread switching
  - Residual-based outlier detection
4. Analyzer: Robust Regression

Iterative

$$\sum_{i} \phi(y_{i} - \sum_{k} x_{ij} \theta_{ik})x_{ij} = 0$$

$$\phi_{k} = \begin{cases} 
\square (k \sigma - x_{i}^2)^2, & -k \sigma < x < k \sigma \\
0, & \text{otherwise}
\end{cases}$$
5. Annotator
Evaluation

RQ1: What is the time cost of our approach?
RQ2: How accurately can our approach calculate the energy consumption of apps?
# Apps for Evaluation

All of our apps are from the Google Play app market

<table>
<thead>
<tr>
<th>App</th>
<th>#Classes</th>
<th>#Methods</th>
<th>#Bytecodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBC Reader</td>
<td>590</td>
<td>4,923</td>
<td>293,910</td>
</tr>
<tr>
<td>Bubble Blaster II</td>
<td>932</td>
<td>6,060</td>
<td>398,437</td>
</tr>
<tr>
<td>Classic Alchemy</td>
<td>751</td>
<td>4,434</td>
<td>467,099</td>
</tr>
<tr>
<td>Skyfire</td>
<td>684</td>
<td>3,976</td>
<td>274,196</td>
</tr>
<tr>
<td>Textgram</td>
<td>632</td>
<td>5,315</td>
<td>244,940</td>
</tr>
</tbody>
</table>
RQ1: Analysis Time Cost

- Pre-execution: Efficient path profiling instrumentation time: 7.5 min
- In-execution: Runtime overhead: 4%
- Post-execution: Path & regression analysis time: 1.97 min
RQ2: Accuracy

vLens is accurate on both application level and path level

<table>
<thead>
<tr>
<th>App</th>
<th>Accuracy</th>
<th>R²</th>
<th>AEE (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBC Reader</td>
<td></td>
<td>0.94</td>
<td>6.5</td>
</tr>
<tr>
<td>Bubble Blaster II</td>
<td></td>
<td>0.90</td>
<td>8.6</td>
</tr>
<tr>
<td>Classic Alchemy</td>
<td></td>
<td>0.93</td>
<td>3.4</td>
</tr>
<tr>
<td>Skyfire</td>
<td></td>
<td>0.99</td>
<td>4.8</td>
</tr>
<tr>
<td>Textgram</td>
<td></td>
<td>0.92</td>
<td>6.3</td>
</tr>
</tbody>
</table>
An Empirical Study of the Energy Consumption of Android Applications

• Source line level measurement study on Android market apps

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

• This result is published in ICSME 2014
  – Co-authored by Shuai Hao, Jiaping Gui, and William G. J. Halfond
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%
- Photography (Ph): 8%
- Sports & News (SN): 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
Research Questions: How Energy Is Consumed in Apps

- RQ 1: App Energy
  - Idle State Energy
  - Non-Idle State Energy
- RQ 2: API Energy
- RQ 3: User Code Energy
- RQ 4: Component Level
- RQ 5: API Level
- RQ 6: Structure Level
- RQ 7: Bytecode Level
RQ 1: How Much Energy Is Consumed by Individual Applications?

- App Energy
  - Idle State Energy
  - Non-Idle State Energy
  - API Energy
    - Component Level
    - API Level
    - Structure Level
    - Bytecode Level
  - User Code Energy
RQ 1: How Much Energy Is Consumed by Individual Applications?

Energy consumption of different applications

[Box plot showing energy consumption for different applications]
RQ 1: How Much Energy Is Consumed by Individual Applications?

Energy consumption of different applications

- LP
- En
- TT
- MM
- HM
- SN
- Ph
- To
- Others
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

- LP
- En
- TT
- MM
- HM
- SN
- Ph
- To
- Others
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?

- App Energy
- Idle State Energy
- Non-Idle State Energy
- API Energy
- User Code Energy
- Component Level
- API Level
- Structure Level
- Bytecode Level
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?

- Code running, 38%
- Wait for input, 37%
- 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%
- Code running, 38%

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

62% energy
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]
  - dLens [Mian et al. ICST 2015]
RQ 3: Which Code Consumes More Energy: System APIs or Developer Written Code?

- App Energy
  - Idle State Energy
  - Non-Idle State Energy

- API Energy
  - Component Level
  - API Level
  - Structure Level
  - Bytecode Level

- User Code Energy
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

- 85% API
- 13% Outliers
- 2% Bytecode

System APIs from the Android SDK
RQ 3: Which Code Consumes More Energy: System APIs or Developer Written Code?

Breakdown of non-idle energy

- Normal user code: 85%
- API: 13%
- Bytecode: 2%
- Outliers: 2%

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

Breakdown of non-idle energy:

- 85% API
- 13% Garbage collection and thread switching
- 2% Bytecode
- Outliers

85%
RQ 3: Which Code Consumes More Energy: System APIs or Developer Written Code?

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?

**Diagram:**

- **App Energy**
  - **Idle State Energy**
  - **Non-Idle State Energy**
- **API Energy**
  - **User Code Energy**
  - **Component Level**
  - **API Level**
  - **Structure Level**
  - **Bytecode Level**
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
RQ 4: How Much Energy Is Consumed by the Different Components of A Smartphone?

Packages of APIs in SDK

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage of non-idle energy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI</td>
<td>6.0</td>
</tr>
<tr>
<td>Net</td>
<td>42.0</td>
</tr>
<tr>
<td>IO</td>
<td>3.0</td>
</tr>
<tr>
<td>Sqlite</td>
<td>7.0</td>
</tr>
<tr>
<td>Camera</td>
<td>17.0</td>
</tr>
<tr>
<td>Location</td>
<td>4.0</td>
</tr>
<tr>
<td>Sensor</td>
<td>0.1</td>
</tr>
<tr>
<td>Media</td>
<td>2.0</td>
</tr>
</tbody>
</table>

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?

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

- **Media**: Percentage of non-idle energy (%)
- **Sensor**: Percentage of non-idle energy (%)
- **Location**: Percentage of non-idle energy (%)
- **IO**: Percentage of non-idle energy (%)
- **Sqlite**: Percentage of non-idle energy (%)
- **Camera**: Percentage of non-idle energy (%)
- **Net**: Percentage of non-idle energy (%)
- **UI**: Percentage of non-idle energy (%)

The highest energy consumption is by **Net** with **android.net.* and java.net.***.
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?

- UI: 5%
- Net: 80%, HTTP requests
- IO: 5%
- Sqlite: 10%
- Camera: 15%
- Location: 5%
- Sensor: 0%
- Media: 0%

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?

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

- **UI**: 5%
- **Net**: 40%
- **IO**: 3%
- **Sqlite**: 10%
- **Camera**: 18%
- **Location**: 6%
- **Sensor**: 1%
- **Media**: 0%

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?

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
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.

One 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.

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

![Graph showing the frequency of API usage](image-url)
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.

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

- **App Energy**
  - **Idle State Energy**
  - **Non-Idle State Energy**
- **API Energy**
  - **Component Level**
  - **API Level**
  - **Structure Level**
  - **Bytecode Level**

RQ 6
RQ 6: How Much Energy Is Consumed by Code in Loops?

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
- User Code Energy
  - Structure Level
  - Bytecode Level

RQ 7
RQ 7: How Much Energy Is Consumed by the Different Types of Bytecodes

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

- **Others**
- **Data Manipulation**
- **Invocation**
- **Branch**

Most frequently used

Data manipulating instructions are the most energy consuming

Grouped based on functionality

Percentage of non-idle energy (%)
Summary

• We developed vLens, a source line level energy measurement tool
• We use vLens to conduct empirical studies
  – Idle state energy needs to be optimized
  – APIs dominate the energy consumption
  – Network is the most energy consuming component
  – Energy consumption is concentrated in a few APIs
Thank you