Estimating Mobile Application Energy Consumption Using Program Analysis

Shuai Hao, Ding Li, William G.J. Halfond, and Ramesh Govindan

University of Southern California
Motivation

• Smartphones are popular

• Batteries don’t last that long

• Many user complaints

Help app developers understand energy implications of their implementation choices
Related Approaches

1. Underlying hardware/OS improvements
2. Cycle-accurate simulators
3. Field measurements
4. Whole program/method level feedback
eLens – Our Approach

Combine program analysis and per instruction cost modeling

1. Lightweight → no OS changes or specialized hardware required
2. Fine-grained → feedback at the source line level
3. Accurate → within 10% of ground truth
4. Fast → estimates within minutes
Overview of eLens

1. Generate workload
2. Identify corresponding executed paths
3. Compute power values for paths
4. Annotate source lines
Generating the Workload

Convert use cases to paths

• Use cases represent scenarios of interest to the developers
• Specified informally or formally
• Our approach: run instrumented version of the app and record runtime information
Estimating a Path’s Energy

\[
\text{Energy} = \sum_{h \in \text{Hardware}} \sum_{i \in \text{path}} C_h(i)
\]

- Cost functions \((C_h)\) for each component \((h)\)
- Instruction’s energy cost is either:
  - Path-independent: “fixed-cost” energy
  - Path-dependent: varies based on path

- Cost functions provided by a Software Environment Energy Profile (SEEP)
Software Environment Energy Profile

- Enables rapid analysis for multiple platforms
- LEAP based profiling
  - Runs Android 3.2
  - Samples at 10KHz
  - Synchronization pulses
  - Multiple hardware components
Instructions: Path-Independent

“Fixed-cost” instructions

- Energy varies by hardware component and power state
- Profiled on LEAP node
  1. Invokes/Returns
  2. Load/Stores
  3. Arithmetic/Logic
  4. Stack management
  5. Jumps/Branches
  6. Fixed-cost APIs
Instructions: Path-dependent

Based on information from other instructions in the path

- Four general categories:
  1. Array allocation
  2. Argument data
  3. Implementing class
  4. External data
- Propagate certain types of information along paths
### Example Energy Calculation

<table>
<thead>
<tr>
<th>ID</th>
<th>Instruction</th>
<th>Cost Functions (nJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>CPU</td>
</tr>
<tr>
<td>0</td>
<td>aload_0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>arraylength</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>icnst_2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>if_icmpeq 14</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>getstatic &quot;Network.out&quot;</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>ldc &quot;Usage…”</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>invokevirtual &quot;println&quot;</td>
<td>20</td>
</tr>
<tr>
<td>14</td>
<td>return</td>
<td>1</td>
</tr>
</tbody>
</table>
Visualization
Evaluation

**RQ1**: What is the accuracy of the energy estimates?

**RQ2**: How much time is needed to estimate the energy consumption?

**RQ3**: Is time profiling equivalent to energy estimation?
Challenges to Obtain Ground Truth

1. Apps compatible with LEAP node
2. LEAP sampling interval
3. Idle time dominates execution time
4. Isolation of application energy
# Subject Applications

<table>
<thead>
<tr>
<th>App</th>
<th>Classes</th>
<th>Methods</th>
<th>Bytecodes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBC Reader</td>
<td>590</td>
<td>4,923</td>
<td>293,910</td>
<td>RSS Reader for BBC News</td>
</tr>
<tr>
<td>Bubble Blaster II</td>
<td>932</td>
<td>6,060</td>
<td>398,437</td>
<td>Bubble blasting game</td>
</tr>
<tr>
<td>Classic Alchemy</td>
<td>751</td>
<td>4,434</td>
<td>467,099</td>
<td>Science game</td>
</tr>
<tr>
<td>Location</td>
<td>428</td>
<td>3,179</td>
<td>232,898</td>
<td>Provide location</td>
</tr>
<tr>
<td>Skyfire</td>
<td>684</td>
<td>3,976</td>
<td>274,196</td>
<td>Web browser</td>
</tr>
<tr>
<td>Textgram</td>
<td>632</td>
<td>5,315</td>
<td>244,940</td>
<td>Text editor</td>
</tr>
</tbody>
</table>
Accuracy: App Level

eLens differs from Ground Truth, on average, 8.8%
Accuracy: Method Level

eLens differs from Ground Truth, on average, 7.1%
## Accuracy: Hardware Components

No more than 12% difference from Ground Truth

<table>
<thead>
<tr>
<th>Application</th>
<th>Error Rate (%)</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CPU</td>
<td>RAM</td>
<td>WiFi</td>
<td>GPS</td>
</tr>
<tr>
<td>BBC Reader</td>
<td>-6.2</td>
<td>5.9</td>
<td>-6.8</td>
<td>-</td>
</tr>
<tr>
<td>Bubble Blaster II</td>
<td>-11.5</td>
<td>3.5</td>
<td>-11.6</td>
<td>-</td>
</tr>
<tr>
<td>Classic Alchemy</td>
<td>-7.9</td>
<td>-6.9</td>
<td>-4.4</td>
<td>-</td>
</tr>
<tr>
<td>Location</td>
<td>-7.8</td>
<td>-8.4</td>
<td>-</td>
<td>8.1</td>
</tr>
<tr>
<td>Skyfire</td>
<td>-7.9</td>
<td>0.9</td>
<td>-8.4</td>
<td>-</td>
</tr>
<tr>
<td>Textgram</td>
<td>5.2</td>
<td>4.6</td>
<td>4.6</td>
<td>-</td>
</tr>
</tbody>
</table>
Runtime of eLens

<table>
<thead>
<tr>
<th>Application</th>
<th>Runtime (s)</th>
<th>Instr.</th>
<th>Est.</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBC Reader</td>
<td>344</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>Bubble Blaster II</td>
<td>450</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>Classic Alchemy</td>
<td>886</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td>274</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Skyfire</td>
<td>258</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Textgram</td>
<td>269</td>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>

Runtime ranged from 5 to 15 minutes
Is Time Equal to Energy?

Compare methods’ time vs. energy

- Linear correlation?
  → Pearson == 0

- Ranking similarity?
  → Cosine similarity == .21

Strong indication of no linear relationship or ranking similarity.
Conclusions

• eLens → estimate energy consumption
  – Uses program analysis and per-instruction cost modeling
  – Imposes minimal requirements on developer

• Evaluation
  – Accuracy was 8.8% at whole program level
Thank you
Case Study of Usefulness

Energy Consumption

- Textgram
- Bubble
- Alchemy
- Skyfire
- BBC

University of Southern California
Working with Market Apps

- APK
- manifest.xml
- classes.dex
- resources
- libs
- JVM Bytecode
- instrumentation
- eLens Lib
- classes.dex
- resources
- libs
- APK
- manifest.xml
- classes.dex
- resources
- libs
Bytecode Costs