What Makes Mobile Apps Tick?
Lessons Learned In Collaborative App Analysis

These slides: is.gd/caratmmw

Eemil Lagerspetz, Ella Peltonen, Sasu Tarkoma,
Hien Truong, N. Asokan
NODES Group
University of Helsinki

Adam Oliner
AmpLab, UC Berkeley
Smartphone Energy Consumption

- Why is my battery empty?

- Where is the energy going?

- What can I do about this?
  - As a user
  - As a developer
  - As a manufacturer
Battery Sizes

Battery size (mAh)

<table>
<thead>
<tr>
<th>Device</th>
<th>Battery Size (mAh)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple iPhone 3G</td>
<td>1300</td>
</tr>
<tr>
<td>Nokia N97</td>
<td>1450</td>
</tr>
<tr>
<td>Nokia N900</td>
<td>1300</td>
</tr>
<tr>
<td>Apple iPhone 4</td>
<td>1500</td>
</tr>
<tr>
<td>Samsung Galaxy Nexus</td>
<td>1600</td>
</tr>
<tr>
<td>Apple iPhone 5</td>
<td>1500</td>
</tr>
<tr>
<td>Nokia Lumia 920</td>
<td>2300</td>
</tr>
<tr>
<td>Samsung Galaxy S4</td>
<td>2500</td>
</tr>
</tbody>
</table>
## Power Profilers

<table>
<thead>
<tr>
<th>Name / Authors</th>
<th>Year</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>PowerScope</td>
<td>1999</td>
<td>Energy profiling of device and processes</td>
</tr>
<tr>
<td>Joule Watcher</td>
<td>2000</td>
<td>Fine-grained thread-level profiling</td>
</tr>
<tr>
<td>Nokia Energy Profiler</td>
<td>2006-2007</td>
<td>On-device power profiler</td>
</tr>
<tr>
<td>Shye et al.</td>
<td>2009</td>
<td>Energy profiling of device and components</td>
</tr>
<tr>
<td>PowerBooeter</td>
<td>2010</td>
<td>Short-term power model for components</td>
</tr>
<tr>
<td>PowerTutor</td>
<td>2011</td>
<td>Hybrid profiler based on PowerBooeter</td>
</tr>
<tr>
<td>battOr</td>
<td>2011</td>
<td>Portable power monitor</td>
</tr>
<tr>
<td>Sesame</td>
<td>2011</td>
<td>On-device power model for device and components</td>
</tr>
<tr>
<td>PowerProf</td>
<td>2011</td>
<td>Application-level power profile</td>
</tr>
<tr>
<td>MobiBug</td>
<td>2011</td>
<td>Automatic diagnosis of application crashes</td>
</tr>
<tr>
<td>eProf</td>
<td>2012</td>
<td>Power model for device, components, and applications</td>
</tr>
<tr>
<td>DevScope</td>
<td>2012</td>
<td>On-device power model for device and components</td>
</tr>
<tr>
<td>eDoctor</td>
<td>2013</td>
<td>Automatic diagnosis of battery drain problems</td>
</tr>
<tr>
<td>V-Edge</td>
<td>2013</td>
<td>On-device power model for device and components</td>
</tr>
</tbody>
</table>
Introducing Carat

Carat is the **first system** to use the mobile device community to detect and correct energy problems.

Our method for **diagnosing** energy anomalies uses the community to infer a specification (expected energy use), and we call deviation from that inferred specification an anomaly.
Outline

What Makes Mobile Apps Tick?

- What is Carat?
- Data and Analysis
- Project Infrastructure
- Carat and Human Behavior
- Detecting Malware Using Carat
- Android extensions: Beyond Apps
- Research Directions
- MSc Thesis Topics

What Makes Mobile Apps Tick?
I. Carat

- Originated in UC Berkeley, in collaboration with University of Helsinki
- Mobile app for Android and iOS
- Currently 700,000 users
- Research project with many directions
What is Carat? 1 / 2

- Mobile app shows users advice: “Kill Facebook for 16m ± 41s battery life”
- Tracks user's battery life average since installation
- Places users within community with a ranking called J-Score
What is Carat? 2/2

- Users see Hogs, high energy use apps
- And Bugs that use energy faster on THEIR device than on others
- Users with these issues quickly see battery life benefits once they are taken care of
• The data we get from users:

  - Battery level (%)
  - timestamp (1s gran.)
  - process list (name only)
  - network status (WIFI/Mobile/off)
  - distance moved

<table>
<thead>
<tr>
<th></th>
<th>95%</th>
<th>90%</th>
</tr>
</thead>
<tbody>
<tr>
<td>timestamp</td>
<td>9:05am</td>
<td>10:08am</td>
</tr>
<tr>
<td>process</td>
<td>Facebook</td>
<td>Facebook</td>
</tr>
<tr>
<td>network</td>
<td>Twitter</td>
<td>Safari</td>
</tr>
<tr>
<td>status</td>
<td>WIFI</td>
<td>off</td>
</tr>
<tr>
<td>distance</td>
<td>0m</td>
<td>100m</td>
</tr>
</tbody>
</table>

The Data & Analysis 2/6

• We pair samples to obtain %/s drain rates
  – The 5% granularity on iOS is a problem, see the Carat SenSys 2013 paper for details
• Associated with a list of processes
  (union of the two samples)
• And a list of features
  – Network status
  – Distance
  – More on Android

\[
\frac{\Delta b}{\Delta t} = \frac{b_1 - b_2}{t_2 - t_1}
\]

95% − 90%

63min

\[\approx 0.0013\%/s\]
The Data & Analysis 3/6

- Energy drain probability distributions are created for
  - Users
  - Apps
  - App and User pairs
  - OS versions
  - Device models
Distributions are compared using the distance between their 95% confidence interval error bars.

- If a distribution has a positive distance from another and a higher mean, it is a Hog (for an app vs the distribution for other apps).
- Hog (for app & user combination vs other users of the same app).
- Bug (for app & user combination vs other users of the same app).
The Data & Analysis 5/6

Facebook on other devices

Facebook bug on your device

What Makes Mobile Apps Tick?
The Data & Analysis 6/6

- A J-Score of i is simply:
- The $i^{th}$ percentile of the average battery life in the community
- Calculated from the ranked list of users' average battery lives
- When recommending OS upgrade, we compare the current OS to the latest OS version
  - Distributions for OS versions are used here
What Makes Mobile Apps Tick?

• Background processes use energy too
• Obvious hogs were found quickly
  – Skype, Pandora Radio, Facebook Messenger, Whatsapp
• Not so obvious Bugs:
  – Twitter with older Android versions
  – SwiftKey on newer Android versions
  – Kindle with 3G (The WhisperSync bug)
The Kindle WhisperSync bug

- With Kindle: 8.4 h
- Without Kindle: 7.7 h
  - Network Off: 12.5 h
  - 3G: 6.3 h
  - Wi-Fi: 6.9 h
Project Infrastructure 1/2

• Data Analysis: Amazon EC2
  - 10 x X-Large VM (4 cores, 15G memory)
• Server facing mobile devices: Amazon EC2
  - 4 x medium VM (1 core, 4G memory)
  - Load balancer, independent DNS name for easy changing of infrastructure when required
• Amazon S3
  - Storage of data (incoming 0.5-1.0 GB / week)
Project Infrastructure 2/2

- Mobile app, open-source on GitHub
  - https://github.com/amplab/carat/
- Deployed to the AppStore and Google Play
- The Carat infrastructure runs on Spark
  - Can use Amazon, real hardware, KVM, ...
- Ukko Cluster @ University of Helsinki
  - For research tasks of Carat
II: Research Directions

- Carat research is currently done mostly @UH
- Three main directions
- Carat and Human Behavior
- Detecting Malware Using Carat
- Android extensions: Beyond Apps
Carat and Human Behavior 1/2

• We conducted a questionnaire answered by over a thousand Carat users
  • We asked why they open Carat, why they kill/do not kill bugs/hogs, has Carat changed the way they use the device
• We combined the questionnaire answers with the users' Carat data
  • Battery life average, bugs/hogs found, which apps were running on the device over time, ...
Carat and Human Behavior 2/2

- We found that if you use Carat for longer than 3 months you become an advanced user
- You gain significantly higher battery life
- Reduce use of significantly more hogs
- Start to use the Carat way of diagnosing battery life problems on your own
  - Advanced users reduced use of other apps also
How Carat Changes User Behavior

Changes in Behavior

- Stopped using applications, Replaced with similar ones
- Kill running applications, More often
- Use hogs and bugs less
- Stopped using applications, Did not replace functionality
- Restart applications, More often
- Did not change behavior

Beginners
Advanced users

What Makes Mobile Apps Tick?
Android Extensions: Beyond Apps

• Use the Android extended features to find feature combinations with distinctive energy patterns

  Upgrade OS +30 ± 2 min
  No Movement +10 ± 3 min
  Use WIFI +30 ± 5 min

  with SwiftKey

  Downgrade OS -14 ± 2 min
  Move around -24 ± 4 min
  Disable WIFI -14 ± 4 min

• Two Master's theses: Ella Peltonen & Cenyu Shen

• Huge dataset, still a lot more to explore
Detecting Malware using Carat 1/2

• We made Carat collect the public key used to sign applications
• We got thousands of application, signature, version records
• We compared them with blacklists from multiple anti-malware vendors and projects
  • McAfee, Mobile Sandbox, MalGenome, ...
Detecting Malware using Carat 2/2

- We found that malware infection rates are higher than you might think (0.26-0.28%)
  - Google says 0.12% of manually installed packages are malware
MDoctor: Malware Prognosis

- ID: 821140c7db2439c4
- Status: At Risk

- Flashlight
- Google Maps
- Guncrafter

Please remove applications listed here if you use them rarely.

ID: 821140c7db2439c4
Status: Infected
Please remove the applications listed in the Prognosis tab or contact your network operator for help.
MSc thesis topics 1/2

• System settings’ impact to energy consumption in mobile devices
  • Turn on automatic brightness + switch to WiFi +
    close Facebook for 2h more battery life

• Carat Gamification: A Facebook game with integrated Carat monitoring
  – Fine-grained energy report of how the game functions use your battery
MSc thesis topics 2/2

• MDoctor and malware prognosis
  • How to analyze noisy time-series data to detect rare events
  • Malware infection, application uninstallation

• Distributed Machine Learning research
  • Topics TBD, contact us if you are interested
Thanks

• The Carat Project
  – Our group: www.cs.helsinki.fi/group/carat
  – Original website: carat.cs.berkeley.edu
• Eemil Lagerspetz, Ella Peltonen and prof. Sasu Tarkoma
  – first.last@cs.helsinki.fi
• We are looking for students interested in mobile development and big data analysis
  – email us