Using sensor hubs to support energy efficient sensor monitoring

Haichen Shen, Aruna Balasubramanian, Anthony Lamarca, David Wetherall
Continuous sensor monitoring apps are becoming popular

BeWell, Acoustic: Lifestyle monitoring

Ambulation: Healthcare monitoring

MobiPerf: Participatory sensing
...but are huge energy consumers...

Locale app, $10
Continuous sensing characteristics

Continuous sensing active times: MobiPerf, Ambulation, and Acoustics

→ minutes

Foreground active times: Angry birds

→ minutes
Large fraction of power consumption is overhead

Existing power saving techniques are ill-suited for sensing apps
Sensor hubs

- What should we do with the sensor hub?
- How can applications seamlessly leverage the sensor hub?
How do apps use sensor data?

• How often is a collected sensor data used?

• What is the time between when sensor data is collected and used?

• How is sensor data used by the application?

Answer those questions without knowing the source code.
Our approach: Taint tracking

**Source**

- READ SENSOR DATA
- COMPUTE
- SHOW USER

**Sink**

- How to profile both explicit and implicit information flow?
- \[ X = \text{Sensor.get}(); \]
- \[ Y = \text{Sensor.get}(); \]
- \[ \text{AVG} = \frac{X + Y}{2}; \]
Instrumenting android byte code

Diagram:
- Android Package (.apk)
  - .dex files
  - resources .arsc
  - uncompiled resources
  - AndroidManifest.xml

Process:
1. Disassemble
2. Byte code
3. Instrumented byte code
4. Assemble
5. Signing
6. ADB
7. Device or Emulator
Goal

Modified, energy-efficient Version of the same app

Instrument taint tracking

Learn sensor usage pattern

Instrument binary to leverage sensor hubs appropriately
Questions?

Contact:
haichen@cs.washington.edu
arunab@cs.washington.edu
anthony.lamarca@intel.com
djw@cs.washington.edu