

# Performance Analysis of Mobile Applications

M. Mauro

Gran Sasso Science Institute (GSSI), L'Aquila, Italy  
`manuel.mauro@gssi.infn.it`



# Performance in Mobile Applications

## Characterization

Mobile applications greatly differs from desktop applications as far as performance are concerned

- Slow data connection
- Less powerful CPUs/GPUs
- Really limited battery life
- Faster user interaction

# Performance in Mobile Applications

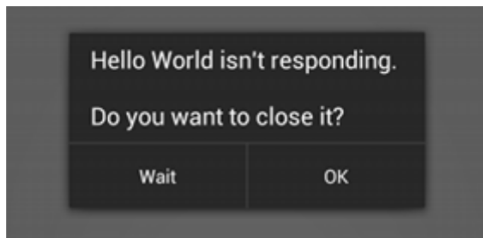
## Characterization

So in the context of mobile application performances are linked to

- UI Responsiveness
- Energy Consumption

# Performance in Mobile Applications

## UI Responsiveness



Usually caused by time consuming operations performed directly inside the graphical thread

# Performance in Mobile Applications

## Energy Consumption



Often associated to a sloppy management of device's sensors and peripherals

# Performance Bugs

## Problems

Performance bugs are particularly difficult to analyze

- Use of emulators and fast internet connections in developing environments
- Hidden behind a complex sequence of user's interactions

# Performance Bugs

## Research

Research has focused on the development of automated tools for

- Detecting
- Testing
- Debugging

performance bugs

# Performance Bugs

## Detection

- Guo, C., Zhang, J., Yan, J., Zhang, Z., and Zhang, Y. *Characterizing and detecting resource leaks in Android applications.*
- Pathak, A., Jindal, A., Hu, Y. C., and Midkiff, S. P. *What is keeping my phone awake? Characterizing and detecting no-sleep energy bugs in smartphone apps.*



# Performance Bugs

Characterizing and detecting resource leaks in Android applications.

The authors focus on three kind of resources

- Exclusive resources
- Memory-consuming resources
- Energy-consuming resources

Their common characteristic is that they should be released explicitly as specified in the Android API Reference.

# Performance Bugs

Characterizing and detecting resource leaks in Android applications.

## Contribution

They propose an automatic solution to detecting resource leaks based on a modified Function Call Graph analysis, which handles the features of event-driven mobile programming by analyzing the callbacks defined in Android framework

# Performance Bugs

What is keeping my phone awake? Characterizing and detecting no-sleep energy bugs in smartphone apps.

A “no-sleep bug” is a condition where at least one component of the phone is woken up and is not put to sleep due to a mistake in manipulating power control APIs in an app

# Performance Bugs

What is keeping my phone awake? Characterizing and detecting no-sleep energy bugs in smartphone apps.

## Contribution

Automatic solution to detect these bugs based on the classic reaching definitions dataflow analysis algorithm

# Performance Bugs

## Testing

- Hao, S., Li, D., Halfond, W.G.J., and Govindan, R. *Estimating mobile application energy consumption using program analysis.*
- Jensen, C. S., Prasad, M. R., and Mller, A. *Automated testing with targeted event sequence generation.*

# Performance Bugs

Estimating mobile application energy consumption using program analysis.

Research in estimating the energy usage of mobile devices explored different techniques

- Specialized hardware
- Cycle-accurate simulators and operating system level instrumentation
- Coarse-grained energy estimates

# Performance Bugs

Estimating mobile application energy consumption using program analysis.

Each of these approaches has one or more shortcomings

- Specialized hardware can be expensive
- Cycle-accurate simulators and operating system level instrumentation can slow down a mobile app beyond the point of usability
- Coarse-grained energy estimates may not be able to pinpoint hotspots within an app

# Performance Bugs

Estimating mobile application energy consumption using program analysis.

## Contribution

eLens, a tool for estimating energy consumption of applications written for Android mobile devices

- Without requiring power measurement hardware
- Level of granularity of the whole application, method, path, or source code line



# Performance Bugs

Automated testing with targeted event sequence generation.

Mobile applications are often structured as collections of screens where user interactions and other events trigger transitions from one screen to another and cause updates of the internal application state

- A key challenge in mobile application testing is managing the explosion in the number of possible event sequences.

# Performance Bugs

Automated testing with targeted event sequence generation.

## Contribution

An algorithm for automated testing of event-driven systems, in particular Android applications. The algorithm is tailored to targets that require long event sequences and reasoning about event parameters.

# Performance Bugs

## Debugging

- Zhang, L., Bild, D. R., Dick, R. P., Mao, Z. M., and Dinda, P. *Panappticon: event-based tracing to measure mobile application and platform performance.*

# Performance Bugs

Panappticon: event-based tracing to measure mobile application and platform performance.

Most mobile applications are interactive. Typically, an input from the user triggers a series of operations culminating in user-visible output, often an update to the display. User experience depends on perceived responsiveness

# Performance Bugs

Panappticon: event-based tracing to measure mobile application and platform performance.

## Contribution

Panappticon establishes causal relationships between user inputs and display updates by tracking execution flow among threads, through asynchronous calls, and across interprocess communication boundaries.