HumanSys 2017

CAMDroid

An Adaptation Framework for Android Context-Aware Multitasking

Kouemo Ngayo Anatoli Dimitrov[1],
Xiaolong Zheng[1], Fu Xiao[2]

[1]Tsinghua University
[2]Nanjing University of Posts and Telecommunications
P.R. China

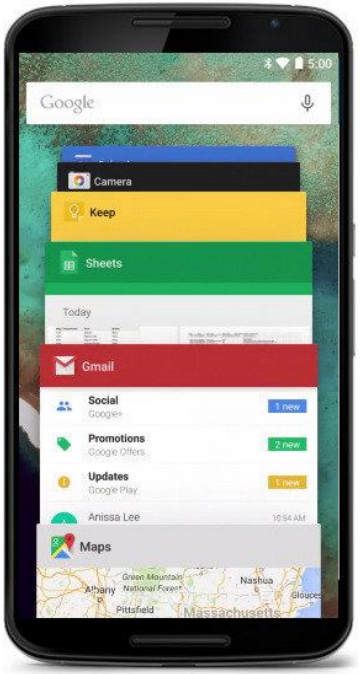
Multitasking

- Multitasking
  - Perform multiple tasks (also known as processes) over a certain period of time by executing them concurrently.

- Android supports multitasking
  - Starting from Android 4 in 2013

Is it satisfactory?

Android Multitasking



- Foreground
  - State: Running
  - Active and interactive
- Background
  - State: Sleeping/ Closed
  - Suspended to save energy
- Only interact with foreground App
  - Due to one small screen
  - not executing concurrently!

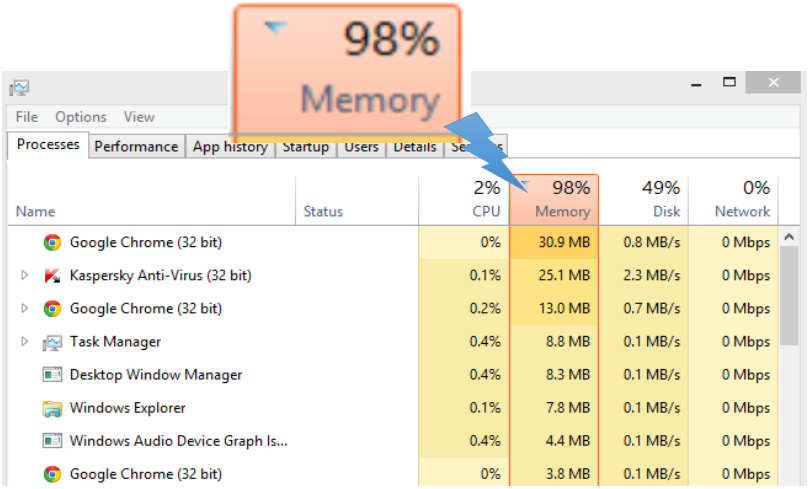
In current multitasking, Apps are sleeping instead of concurrently running in background!

Research Target

- Context-aware multitasking
  - Apps run in the background
  - “Real” concurrent execution
  - Enable users interact with background Apps
  - Dynamically preload/offload Apps to reduce the launch time/save the memory resource.

Challenges

- Background Apps are suspended and cannot access whole context information
- Keep all Apps running in background will lead to unacceptable energy consumption
- Use up the memory



Context Awareness

- Sense and react based on the physical conditions
- Context types:
  - location, identity, activity, time etc.

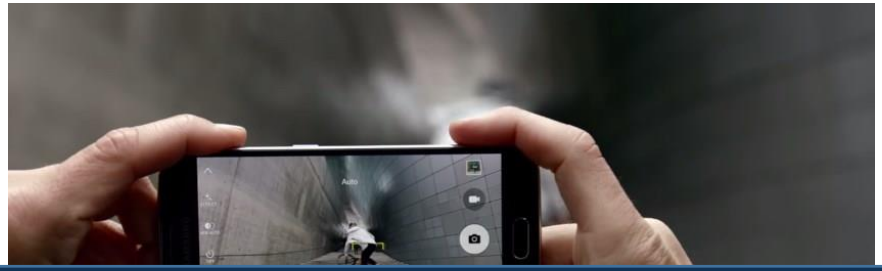
Context Awareness

- Widely used in mobile Apps
- Example: Location-based preloading

Apps maintain the context by themselves
Isolated adaption engine is used in own App

CAreDroid for Context-aware Apps

- External context (*outside OS*) only
  - Without internal context (App status *inside OS*)
- Foreground bias
  - Interact with foreground Apps only
- Static configuration written by App developers
  - Preform predefined actions in the given context

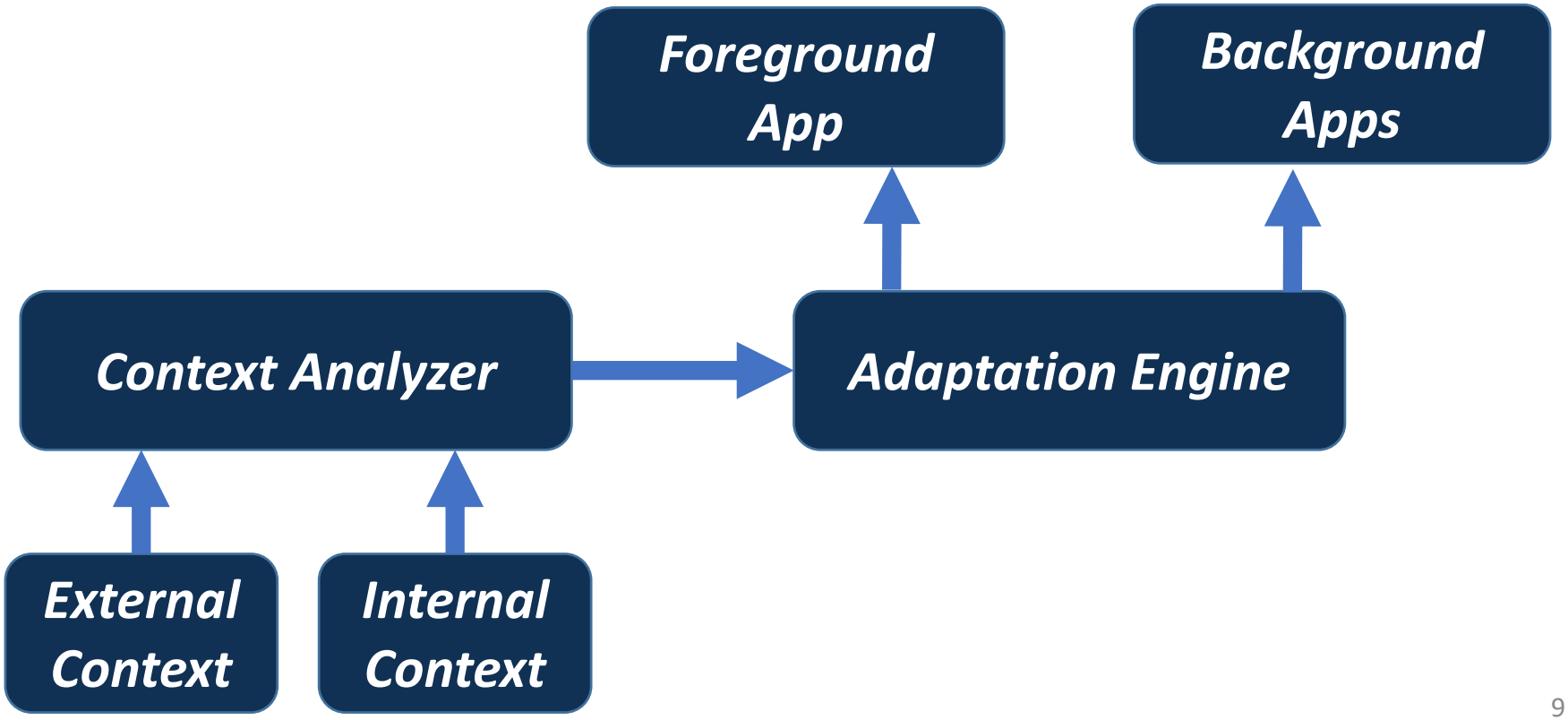


Context-aware multitasking demands dynamic control of background Apps based on both external and internal context

[1] Salma Elmalaki et.al, CAreDroid: Adaptation Framework for Android Context-Aware Applications, MobiCom 2015.

CAMDroid

- Context-aware multitasking
  - Dynamic control of background Apps
  - With both external and internal context

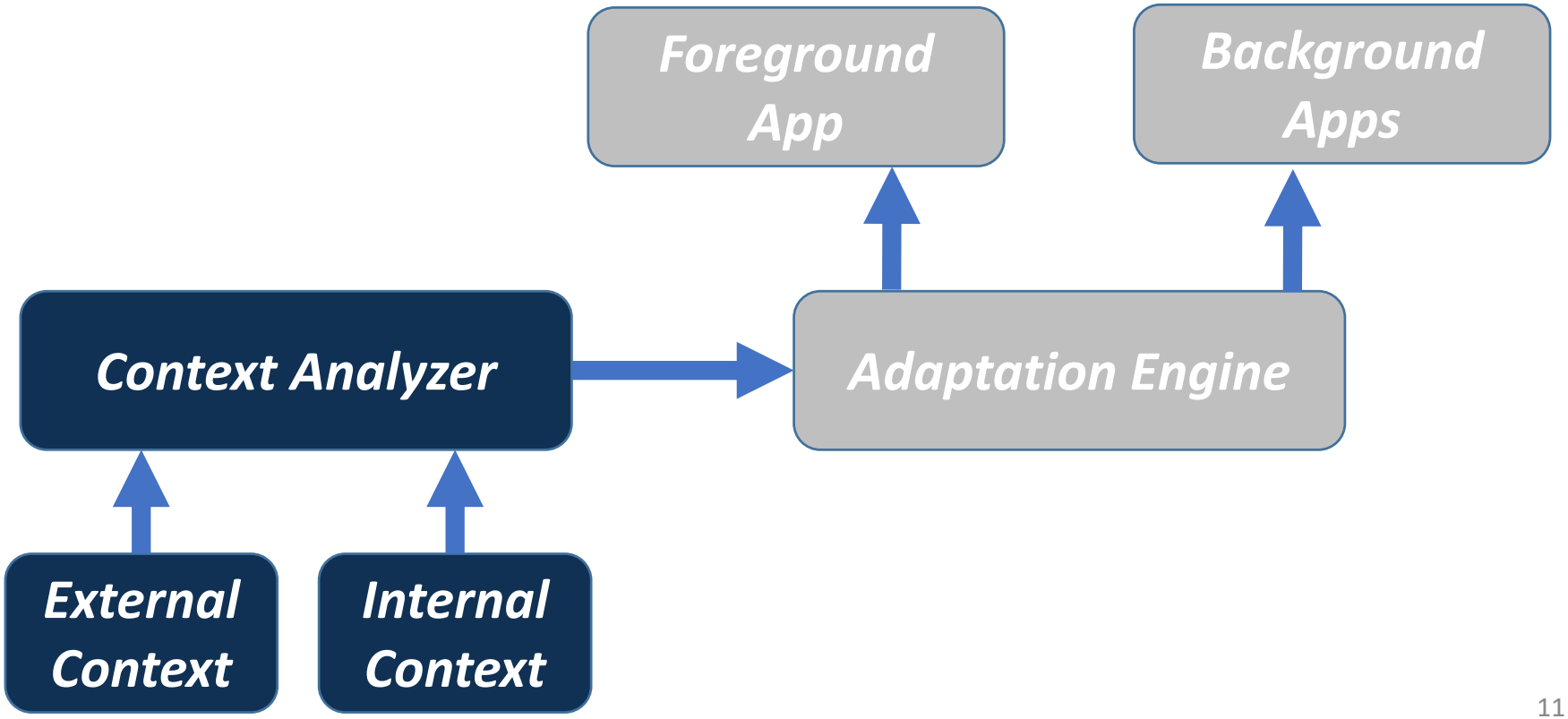


Our Solutions

- Background Apps are suspended and cannot access whole context information
  - Context analyzer inside OS to collect both external and internal context for all Apps
- Keep all Apps running in background will lead to unacceptable energy consumption
  - Adaptation engine that preloads or executes Apps that are frequently used in recent period, in current context
- use up the memory
  - Activate Apps with strict memory constraints

CAMDroid

- Context-aware multitasking
  - Dynamic control of background Apps
  - With both external and internal context

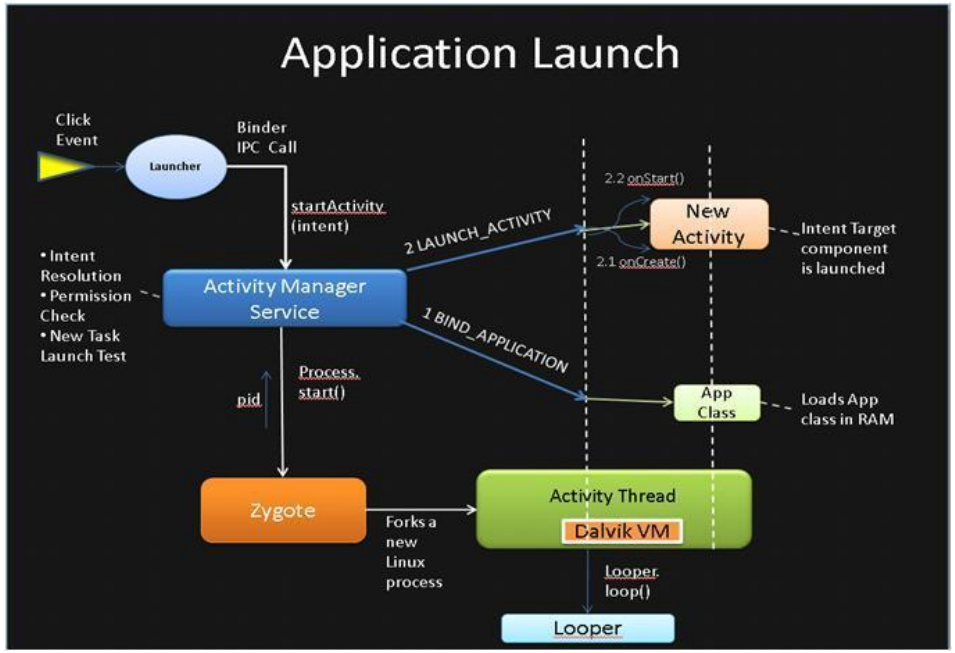


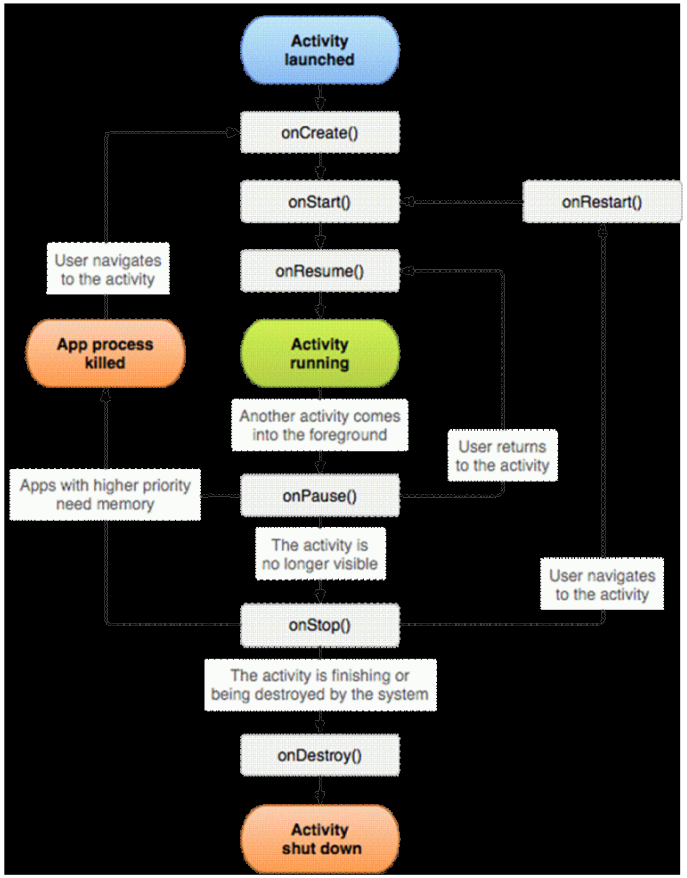
Context Analyzer

- External context
  - Analyze with sensor and sensorless sensing

Context Analyzer

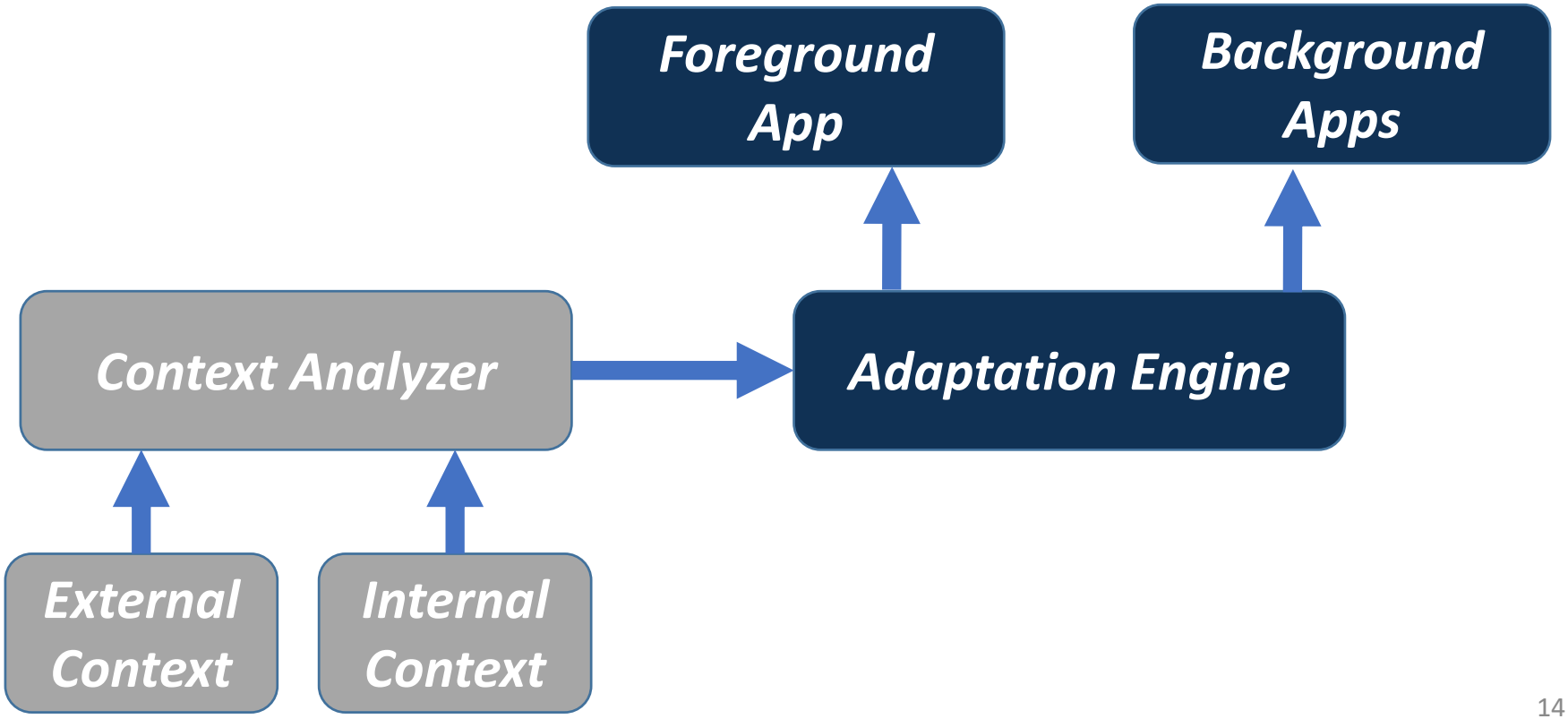
- Internal context
  - App status, number of use, service time, required memory size …
- Hook system calls





CAMDroid

- Context-aware multitasking
  - Dynamic control of background Apps
  - With both external and internal context



Adaptation Engine

- Real-time multitasking with context-awareness
  - Foreground/background Apps react accordingly
  - Preload/offload apps
- Current implementation
  - most frequently used in recent period

Metric App i will be used

$$v_i = \omega_1 t_i + \omega_2 h_i + \omega_3 c_i$$

Dynamic updating the score

Memory constraint

$$Subject\ to\ \sum_{i=1}^{n} mr_{yi} x_i \leq M_a$$

$$y = r, p, s, d$$

Algorithm 1 Algorithm for App Background List

Input: v_i, mr_{yi}, n, M_a
Output: PList

```
for J = 0 to MR do
    m[0, j] := 0
end for

for i = 1 to n do
    for j = 0 to MR do
        if mr[i] > j then
            m[i, j] := m[i − 1, j]
        else
            m[i, j] := max(m[i − 1, j], m[i − 1, j − mr_yi[i]] + v[i])
        end if
    end for
end for
return PList
```

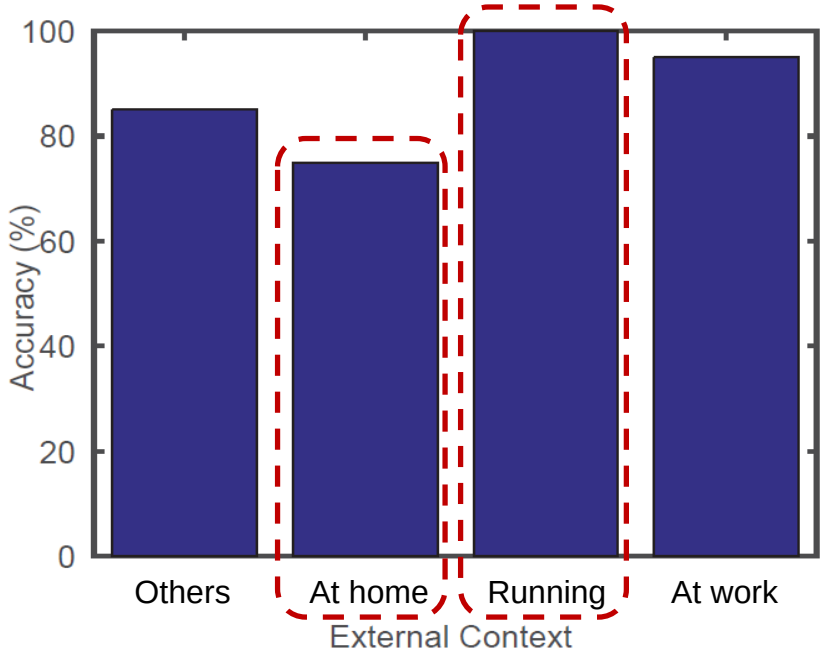
CAMDroid Implementation

- Device & Operating System
  - Android 5.1.1
  - Google LG Nexus 5 mobile phone

- System image size
  - Android: 358930 KB
  - CAMDroid: 380851 KB
  - Overhead: 21921 KB

Evaluation

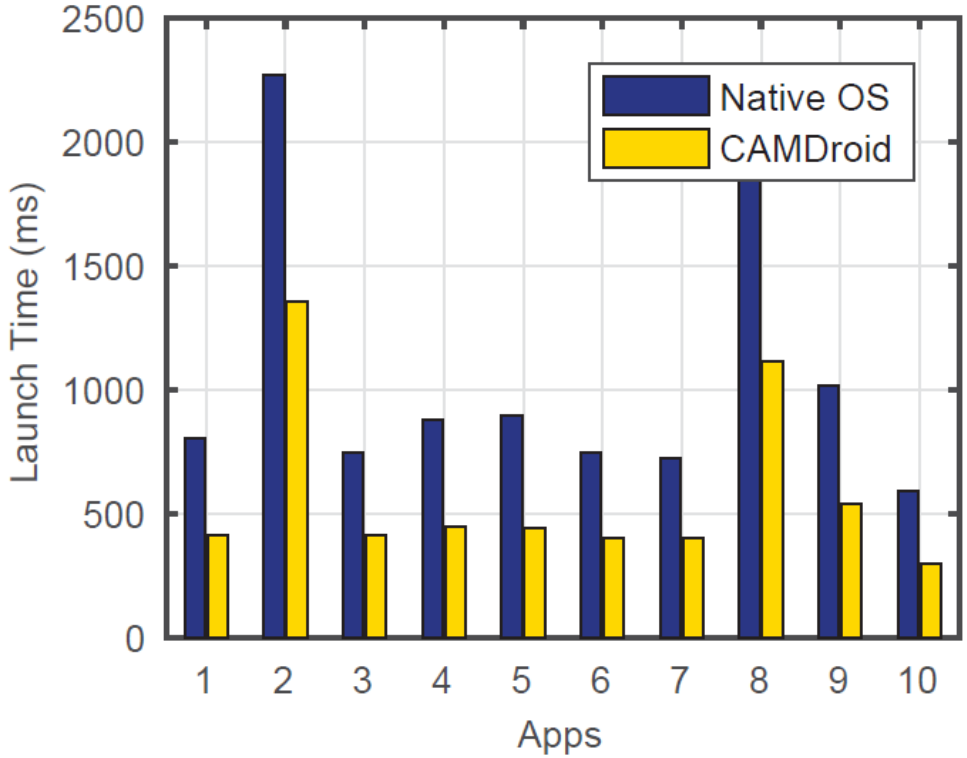
- Predicted task list
  - If the opened App is in the list, we regard CAMDroid accurately predicts once.
  - 100 trails under different external contexts



  - Our tested contexts are coarse-grained

Evaluation

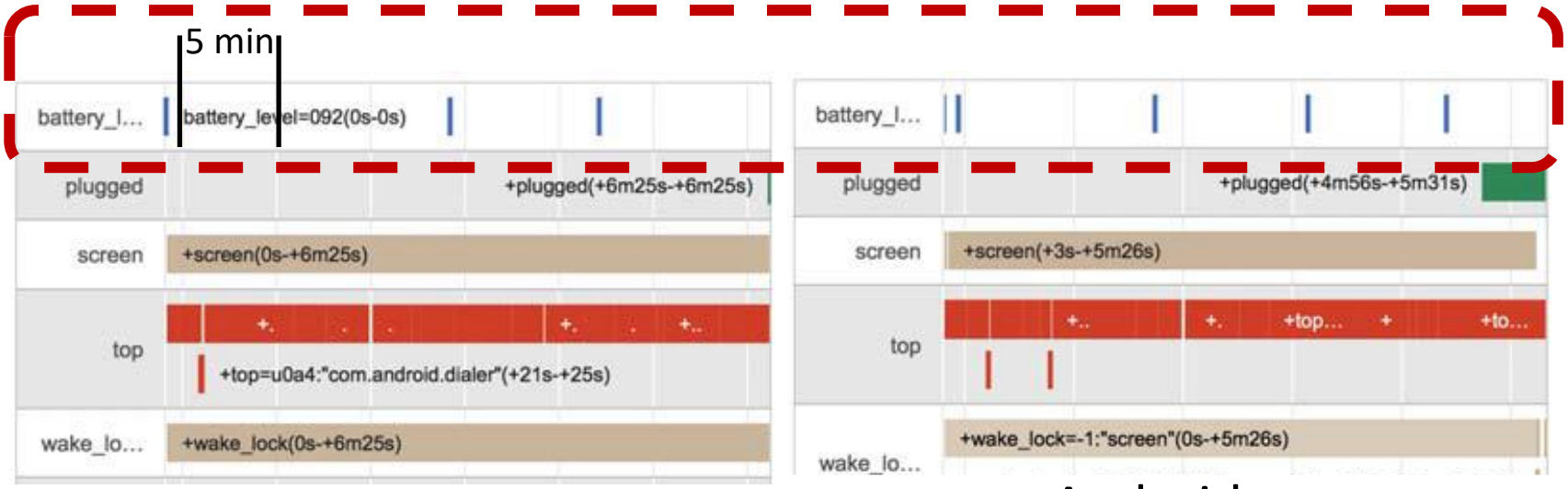
- Reduce the launch time
  - Due to the preloading, launch time is reduced
  - Reduced by 50% in average



Evaluation

- Off-loading saves energy
  - Close Apps unlikely used in current context

Battery level drops ***4%*** in native Android,
and ***3%*** in CAMDroid, during 30 minutes



CAMDroid

Android

Event Tracker

Demo

Conclusion

- CAMDroid -- Context-Aware Multitasking
  - Bring context-awareness into the operating system
  - Provide external and internal context to Apps
  - Enable the interaction between user/environment and background Apps
  - Save energy and launch time
- Future work
  - Improve prediction accuracy according to fine-grained correlation between context and App
  - Include personalized models

Thank you!

Xiaolong Zheng

http://www.greenorbs.org/people/xiaolong/

2017.11.05