MAUI: Making Smartphones Last Longer With Code Offload

Eduardo Cuervo - Duke
Aruna Balasubramanian - U Mass Amherst
Dae-ki Cho - UCLA
Alec Wolman, Stefan Saroiu, Ranveer Chandra, Paramvir Bahl – Microsoft Research

Presented by Mingzhou Zhou
Outline

1. Motivation
2. MAUI Overview
3. MAUI Architecture
4. Implementation & Evaluation
5. Summary & Discussions
Motivation

Battery is a scarce resource
One Solution: Remote Execution (Offload)
Challenges of offload on energy consumption:
1. Via 3G or Wi-Fi?
Pervasive Vs. Energy saving
2. What code should be offloaded?
3. How to minimize the burden of programmer?
Challenge: Via 3G or Wi-Fi?

- **3G connectivity**
  - 10KB: 576
  - 100KB: 979
  - Total: 2,762

- **Wi-Fi connectivity**
  - 10KB: 330
  - 100KB: 542
  - Total: 1,573

Bar chart showing energy consumption by upload (mJ) for different conditions and upload sizes.
MAUI Overview

**MAUI**: Mobile Assistance Using Infrastructure

Enables fine-grained energy-aware offload of mobile code to the infrastructure

1. Architecture
2. Prerequisite
3. Making Decision of offloading
MAUI Architecture

Handle control and data transfer for offloaded methods

Application

Maui Runtime
Client Proxy
Profiler
Solver

RPC

Maui Runtime
Server Proxy
Profiler
Solver

RPC

Maui Controller

Smartphone

Maui server
2. Prerequisite

I. Two versions of application
   Smartphone & Remote Server

II. Methods that can be executed remotely must be marked in the source code

III. Typical Linkage
MAUI Overview: Making Decision

3. Making Decision

Profiler:
1. Estimate energy consumption if running the program on the smartphone
2. Estimate energy consumption and delay on network transfer

Solver:
According to the estimations, decides which methods should be offloaded that minimizes the smartphone’s energy consumption, subjects to latency constraints
MAUI Architecture

Handle control and data transfer for offloaded methods

- **Maui Runtime**
  - Client Proxy
  - Profiler
  - Solver

- **Server Proxy**
  - Profiler
  - Solver

**RPC**

**Smartphone**

**Maui server**

- **Maui Controller**

Application

- **Handle control and data transfer for offloaded methods**
MAUI Architecture: Profiler

1. Device Profiling
2. Program Profiling
3. Network Profiling
MAUI Architecture: Profiler

1. Device Profiling
   Why?
   Fine-grained energy measurements are not available on today’s smartphones
   Solution?
   Building two model:
   Energy consumption = f (Total No. of CPU cycles)
   Energy consumption model for Wi-Fi & 3G

2. Program Profiling
3. Network Profiling
MAUI Architecture: Profiler

1. Device Profiling

2. Program Profiling
   I. Measure runtime duration
   II. Measure the CPU state (reason: dynamic voltage scaling, high voltage corresponds to higher CPU speed)

→ Energy Consumption = f(I \times II)!

3. Network Profiling
MAUI Architecture: Profiler

1. Device Profiling
2. Program Profiling
3. Network Profiling

MAUI uses TCP: Send **10 KB** of data over TCP to the MAUI server and measure the transfer duration to obtain the average throughput, latency and bandwidth.

Representative of typical transfers
MAUI Architecture: Solver

1. Global optimal
2. Solving 0-1 integer linear programming (ILP) problem
MAUI Architecture: Solver

1. Global optimal

Callgraph of Face Recognition App

- User Interface: 1006 mJ
- FindMatch: 872 mJ
- InitializeFace Recognizer: 4703 mJ
- DetectAndExtract Faces: 13030 mJ

2. Solving 0-1 integer linear programming (ILP) problem

Cheaper to do local
1. Global optimal

2. Solving 0-1 integer linear programming (ILP) problem

MAUI Architecture: Solver

User Interface 1006 mJ
FindMatch 872 mJ
InitializeFaceRecognizer 4703 mJ
DetectAndExtractFaces 13030 mJ

Cheaper to offload
MAUI Architecture: Solver

1. Global optimal
2. Solving 0-1 integer linear programming (ILP) problem

\[
\text{maximize } \sum_{v \in V} I_v \times E_v^L - \sum_{(u,v) \in E} |I_u - I_v| \times C_{u,v} \\
\text{such that: } \sum_{v \in V} \left( (1 - I_v) \times T_v^L + (I_v \times T_v^R) \right) \\
+ \sum_{(u,v) \in E} \left( |I_u - I_v| \times B_{u,v} \right) \leq L \\
\text{and } I_v \leq r_v, \forall v \in V
\]

- Total energy saved by executing methods remotely
- Energy cost of data transfer
- Execution time constraint
Implementation & Evaluation

1. MAUI Implementation
2. Evaluation: Answer to three questions:
   I. How much can MAUI reduce energy consumption?
   II. How much can MAUI improve performance?
   III. Does the MAUI solver adapt to changing network conditions?
Implementation

1. Platform
   Windows Mobile 6.5
   .NET Framework 3.5
   HTC Fuze Smartphone

2. Applications
   Chess
   Face Recognition
   Video Game
   Voice-based translator
Evaluation: I. Reduce energy consumption?
Evaluation: II. Improve performance?

ONE RUN FACE RECOGNITION

400 FRAMES of VIDEO GAME
Evaluation: III. Adapt to changing network?

Video game in changing network? Yes. (Refer to 7.3.4 for more details)

Why?
Profiler continuously monitors the program and network characteristics.
Summary & Discussions

1. MAUI enables developers to:
   Bypass the resource limitations of handheld devices
   Low barrier entry: simple program annotations

2. For a resource-intensive application:
   MAUI reduced energy consumed by an order of magnitude
   MAUI improved application performance similarly

3. MAUI adapts to:
   Changing network conditions
Summary & Discussions

1. How popular is the Windows Mobile OS?

2. Is a MAUI server always available?

3. Programmers need to reconstruct and mark methods that can be executed remotely, their workload reduced?

4. Support multi-threaded application?
Questions?