MAUI: Making Smartphones Last Longer With Code Offload

Slides based on a paper by:
Eduardo Cuervo (Duke University),
Aruna Balasubramanian (UMass. Amherst),
Dae-ki Cho (UCLA),
Alec Wolman, Stefan Saroiu, Ranveer Chandra,
Paramvir Bahl (MSR)

Paper presented in the The 8th International Conference on Mobile Systems, Applications, and Services (MobiSys '10)
June 2010, San Francisco, CA.

Adi Fuchs

COS518 Advanced Computer Systems, Fall 2015

Today's Topics

- The Problem
- Motivation
- MAUI
- Evaluation
- Summary

The Problem

- Mobile devices are ubiquitous
- Wider range of applications
- Mobile Computation gets more intense

- **Battery fails to keep up…**

The Problem

- Cloud services are also ubiquitous
- Possess high computation capabilities
- Not limited by battery!
- Idea: mobile computation offloading to the cloud!

Today's Topics

- The Problem
- Motivation
- MAUI
- Evaluation
- Summary

Motivation

- Three questions quantify the need for remote offloading:

1. How Severe is the Energy Problem in Today's Mobile Devices?

- Synthetic benchmark (bulk fetching+display) drained battery after 1.5 hours

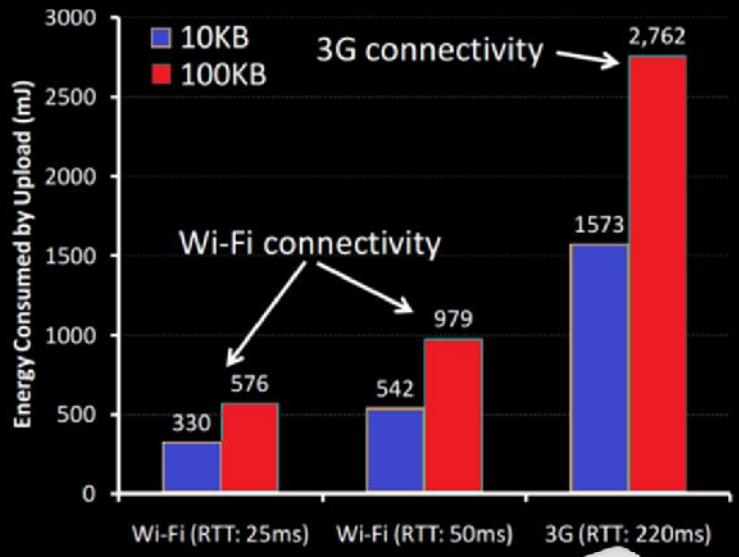
- Synthetic, yet realistic scenario (Video streaming)

Motivation

- Three questions quantify the need for remote offloading:

2. How Energy Efficient is 3G for Code offloading?

- Researchers tested the uploading and downloading of 10/100KBs of code
- Energy(3G) is roughly 5x Energy(Wi-Fi)
- Battery drained after 2 hours of extensive use
- 3G might be impractical to use



Motivation

❑ Three questions quantify the need for remote offloading:

3. How Sensitive is the energy consumed to the Wi-Fi RTT?

- 10/100KB offloading on Wi-Fi
- Near linear energy growth w.r.t. RTT
- Cloud should strive to minimize offloading RTT
- Energy saving is significant for nearby servers (RTT~10ms)

Today's Topics

- The Problem
- Motivation
- MAUI
- Evaluation
- Summary

MAUI

- Main Challenges:

- **Partitioning** - what is the granularity of the code that is offloaded?
- **Amortizing costs** – what is the minimal "state" for offloading?
- **Detection** - how to detect offload candidates "on-the-fly"?
- **Programmability** - do not over-burden the programmer

MAUI

- The MAUI programming model:
  - C# applications containing "remotable" methods (marked by the programmer)
  - Methods that do not implement UI
  - Methods that do not interact with mobile device's IO devices (GPS etc.)
  - Methods must be able to be re-executed (i.e. without irreversible side-effects)

MAUI

❑ The MAUI programming model:

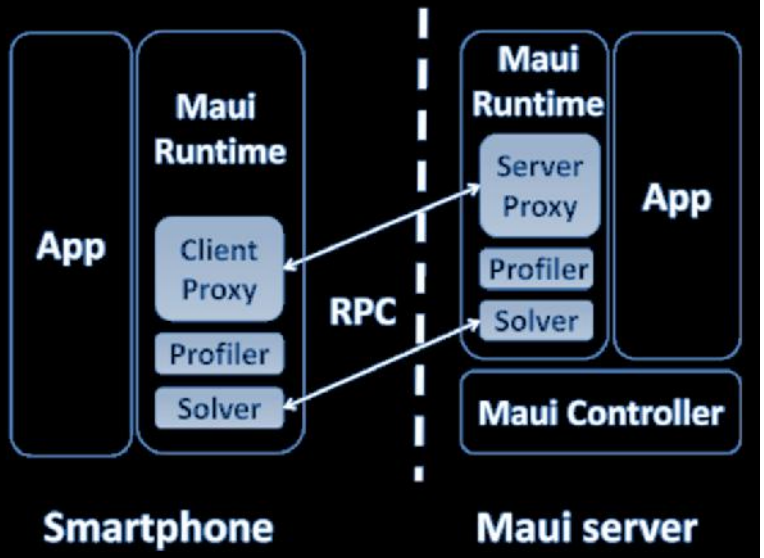
- **Methods are identified by attributes, server has matching messaging interface**

```
// original interface
public interface IEnemy {
[Remoteable] bool SelectEnemy(int x, int y);
[Remoteable] void ShowHistory();
void UpdateGUI(); }

// remote service interface
public interface IEnemyServer {
MAUIMessage <AppState,bool> SelectEnemy(int x, int y);
MAUIMessage <AppState,MauiVoid> ShowHistory();
}
```

MAUI

- The MAUI architecture:
  - **Proxy -** handles control + data transfers
  - **Profiler -** instruments the program
  - **Solver –** ILP solver (elaborated later)



- **MAUI coordinator –** handles incoming requests, creates a partitioned application

- both device and server hold copies of the application (using CLR)

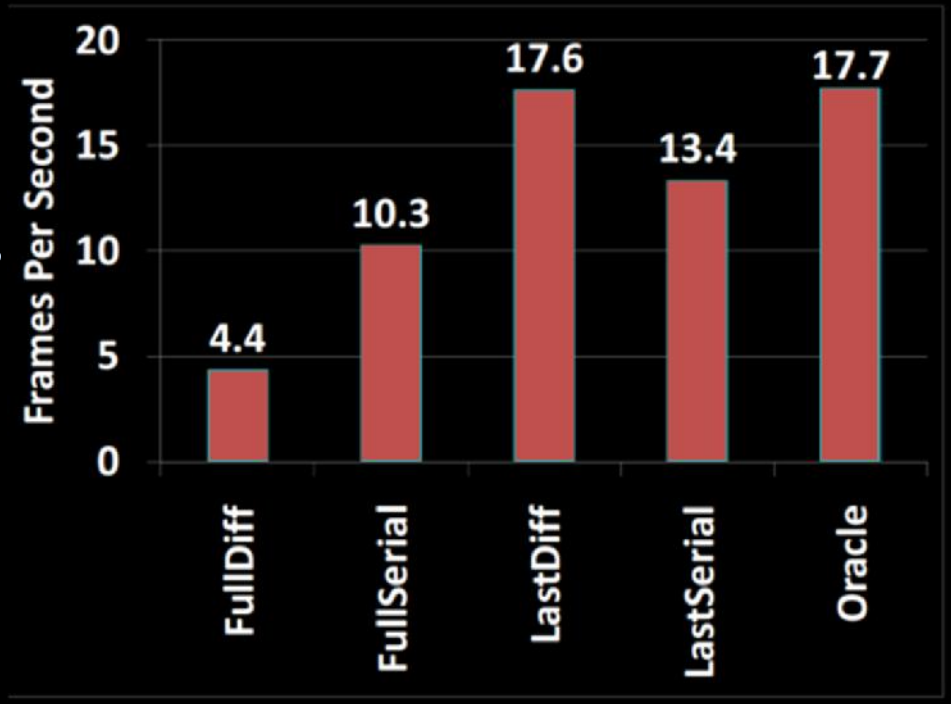
- Currently no support for multi-threaded applications ☹

MAUI

- The MAUI profiler
- Instruments methods to predict offload profitability,depending on:
  - The smartphone's energy consumption
  - Each method's characteristics (e.g. run-time and resources needed)
  - Network characteristics (RTT, BW latency, and packet loss rate)
- **Problem I**: serializing entire state is time-consuming
- **Problem II**: sending entire state wastes a lot of bandwidth
- **Heuristic Solution:** app-state delta calculation

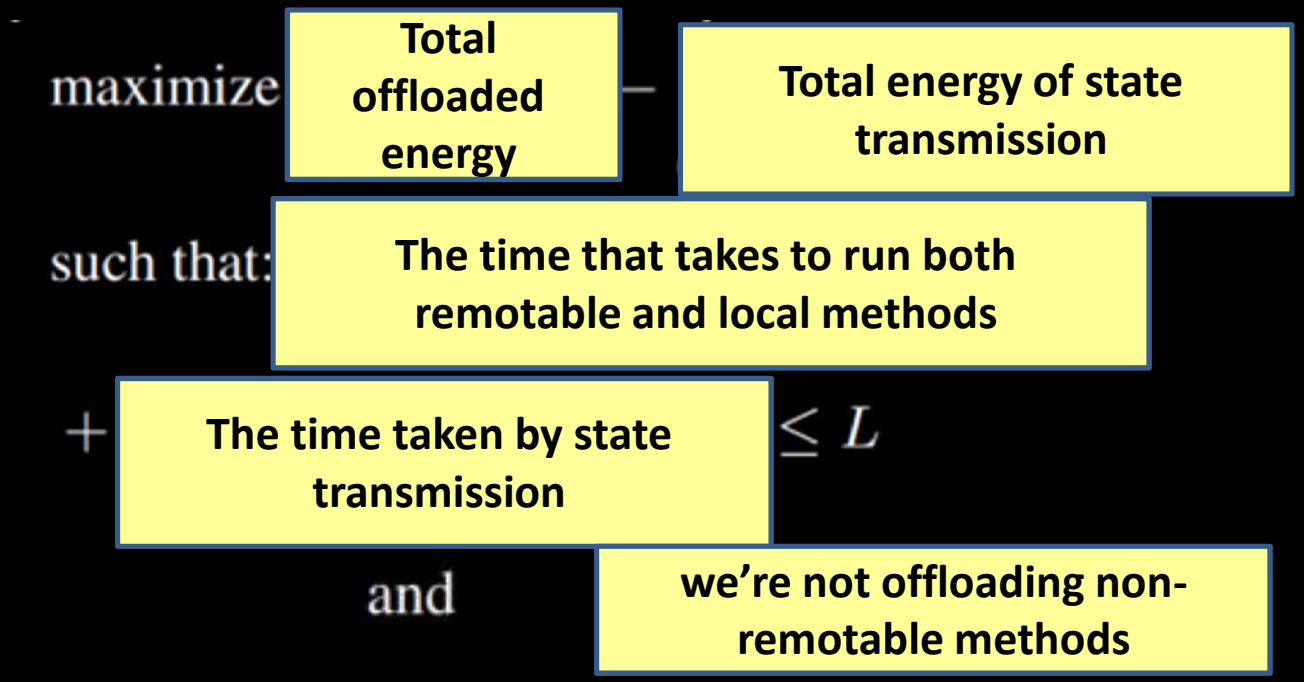
MAUI

- The MAUI profiler
- Profiling policies
  - FullDiff – serialize and calculate deltas on every call
  - FullSerial – serialize on every call
  - LastDiff – serialize on first call only, calculate deltas for each call
  - LastSerial – serialize first call only
  - Oracle – knows exactly which methods to offload without calculation



MAUI

- The MAUI solver: attempts to solve the offload decision problem
  - Reaching the optimal solution requires a global view of the program
  - Formal problem definition: G(V,E) v= call stack method e=(u,v)→ u invokes v

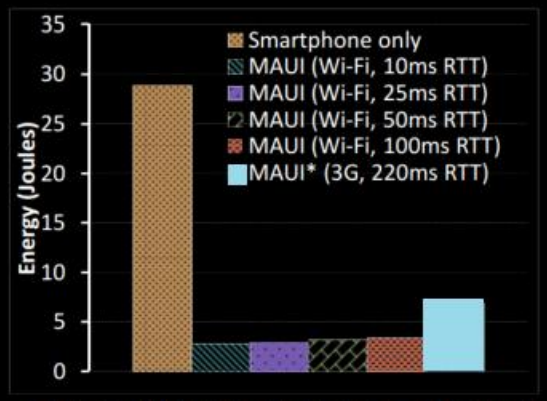


Today's Topics

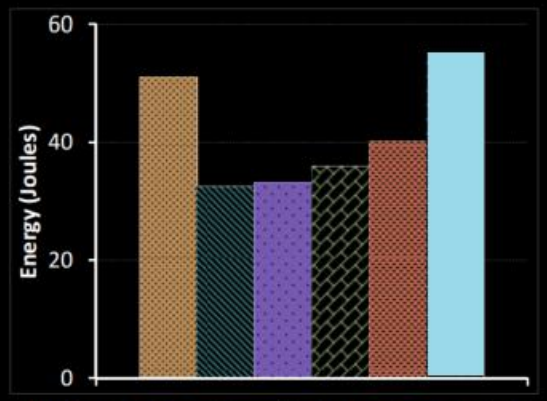
- The Problem
- Motivation
- MAUI
- Evaluation
- Summary

Evaluation

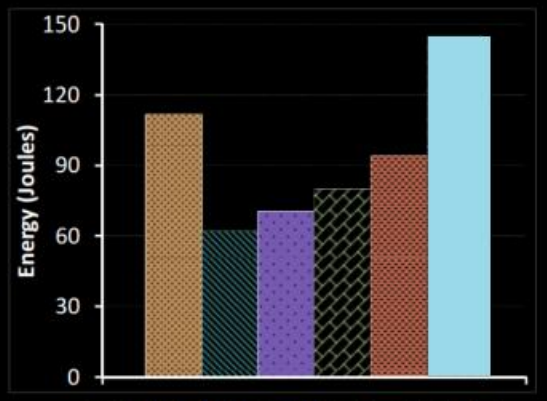
- **Methodology:**
  - 3 micro-benchmarks are evaluated (Face recognition, chess moves, video)
  - 6 configurations: smartphone only, MAUI + 4 WiFi RTTs, MAUI* + 3G



ONE RUN FACE RECOGNITION



400 FRAMES of VIDEO GAME

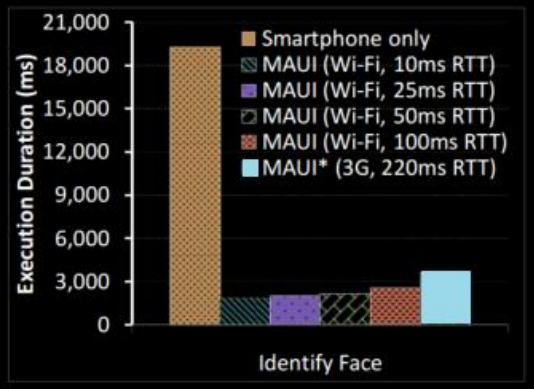


30 MOVE CHESS GAME

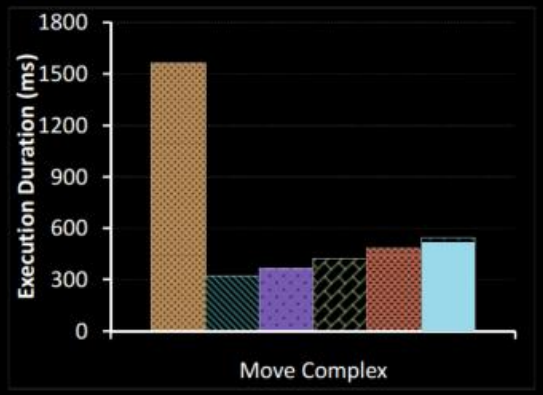
Evaluation

❑ Methodology:

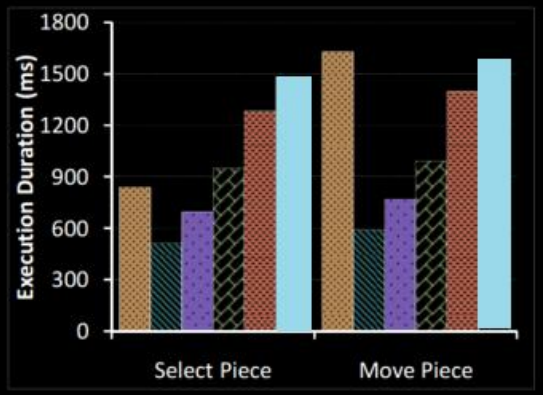
- 3 micro-benchmarks are evaluated (Face recognition, chess moves, video)
- 6 configurations: smartphone only, MAUI + 4 WiFi RTTs, MAUI* + 3G



ONE RUN FACE RECOGNITION



400 FRAMES of VIDEO GAME



30 MOVE CHESS GAME

Today's Topics

- The Problem
- Motivation
- MAUI
- Evaluation
- Summary

Summary

- Combines two approaches:
  - Fine-grained partitioning (offload strategies are defined by the programmer)
  - Process and VM migration (limited choice for offloading, all done by the OS)
- Use of CLR: same copy of the application on the device and server
  - Provides architecture-independent execution (translation overhead?)
  - Idea: maybe MAUI server should run a VM simulating mobile device?
- Might provide benefits beyond energy savings
  - Can offloading improve performance?
  - Applications that could not run on mobile devices run on the cloud

Summary

- Conservative approach: relying on entire objects as AppState
  - WIP: static analysis tool check which vars are referenced by remotable methods
- In the MAUI solver section they only formulate the problem…
  - How does it really solve the problem? Does it really solve an ILP?
- Tested on three micro-benchmarks
  - What about other applications?
  - How much of the presented gain came from programming effort?
- Does the fact that 3G is wasteful make MAUI impractical?

Summary

- Timeliness!= Performance

Summary