



ThinkAir: Dynamic Resource Allocation and Parallel Execution in Cloud for Mobile Code Offloading

Sokol Kosta, Pan Hui

Deutsche Telekom Labs, Berlin, Germany

Andrius Aucinas, Richard Mortier

University of Cambridge, Cambridge, UK

Xinwen Zhang

Huawei Research Center, Santa Clara, CA, USA

IEEE INFOCOM 2012

Prominent Related Works

- MAUI(2010)
 - Provides method level code offloading based on .NET framework.
 - Does not address the scaling of execution in cloud.
- CloneCloud(2011)
 - Provides offline static analysis of different running condition of the process binary, and build a database of pre-computed partitions.
 - Limited input/environment conditions, and needs to be bootstrapped for every new apps.

ThinkAir

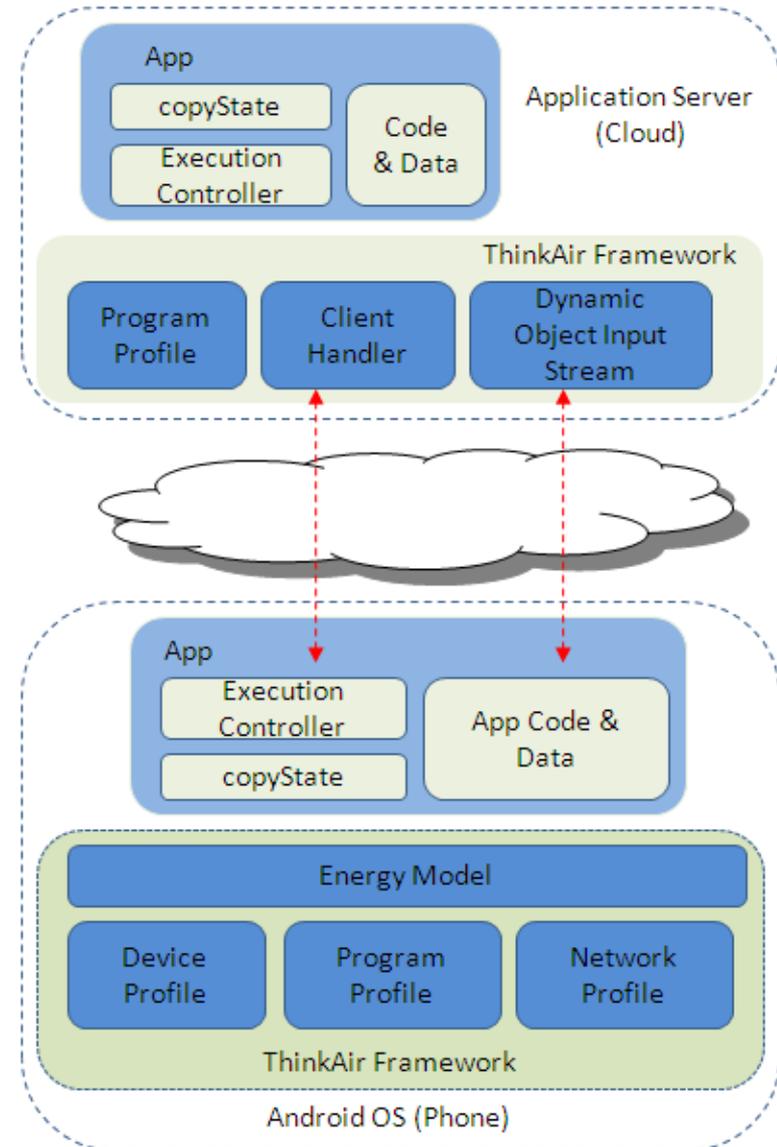
- A framework that exploits the concept of smart phone virtualization in the cloud, and provides method-level computation offloading.
 - Parallelizing method execution using multiple VM images.
 - On-demand resource allocation.
 - Online method-level offloading.

Design Goals

- Dynamic adaptation to changing environment.
- Ease of use for developers.
- Performance improvement through cloud computing.
- Dynamic scaling of computation power.

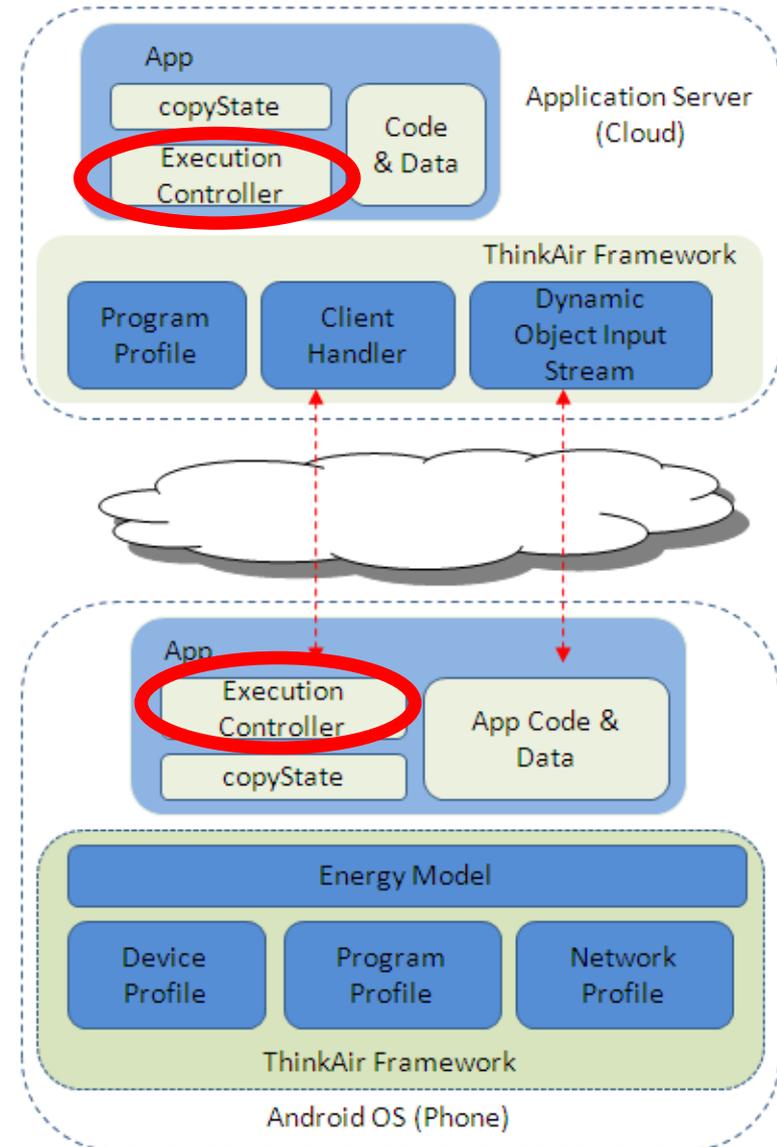
Overview

- Annotate methods with `@Remote`.



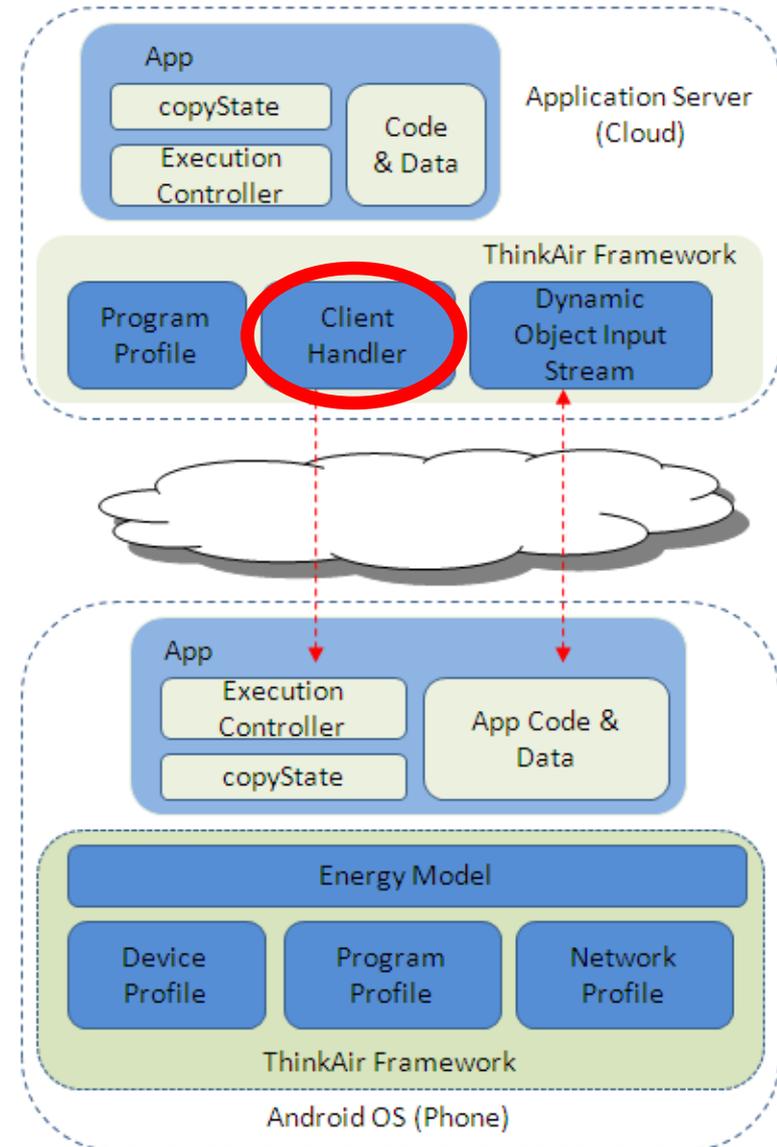
Execution Controller

- Make offloading decisions.
- Four policies:
 - Execution time
 - Energy
 - Execution time and energy
 - Execution time, energy, and cost.



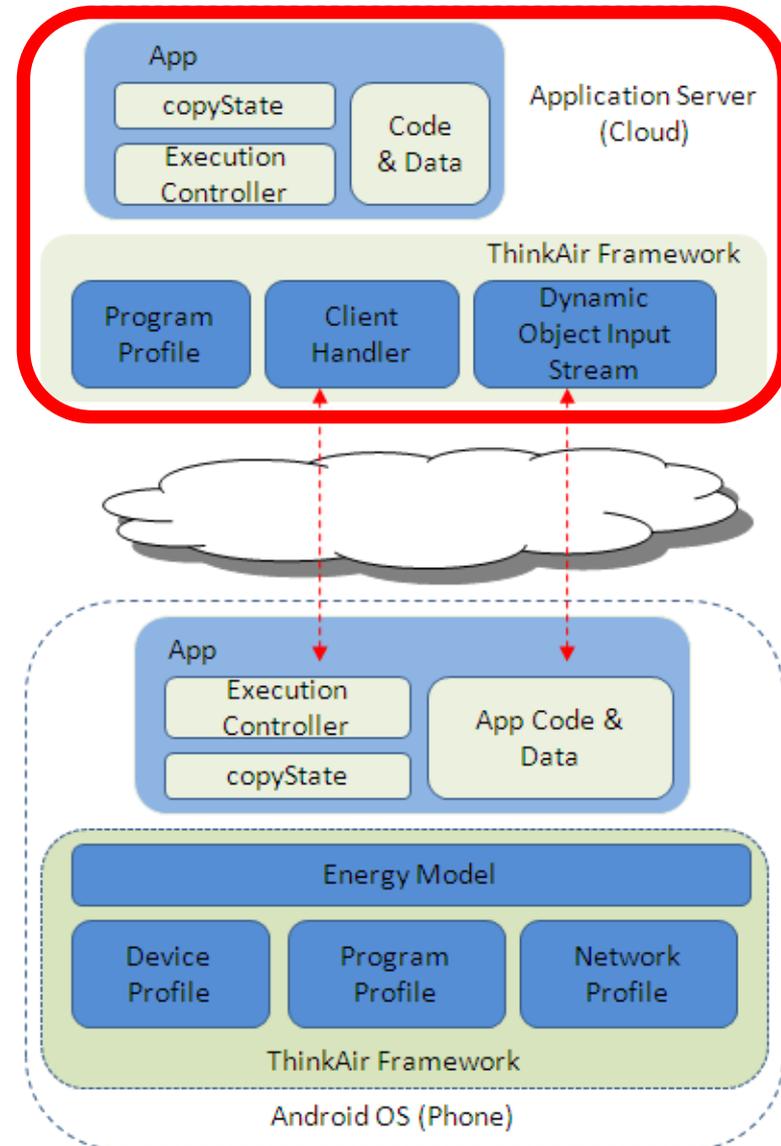
Client Handler

- Code execution
 - Manage connection.
 - Execute code.
 - Return results.
- VM management
 - Add VM with more computing power or resources.
 - Distributes task among VMs, and collects results.



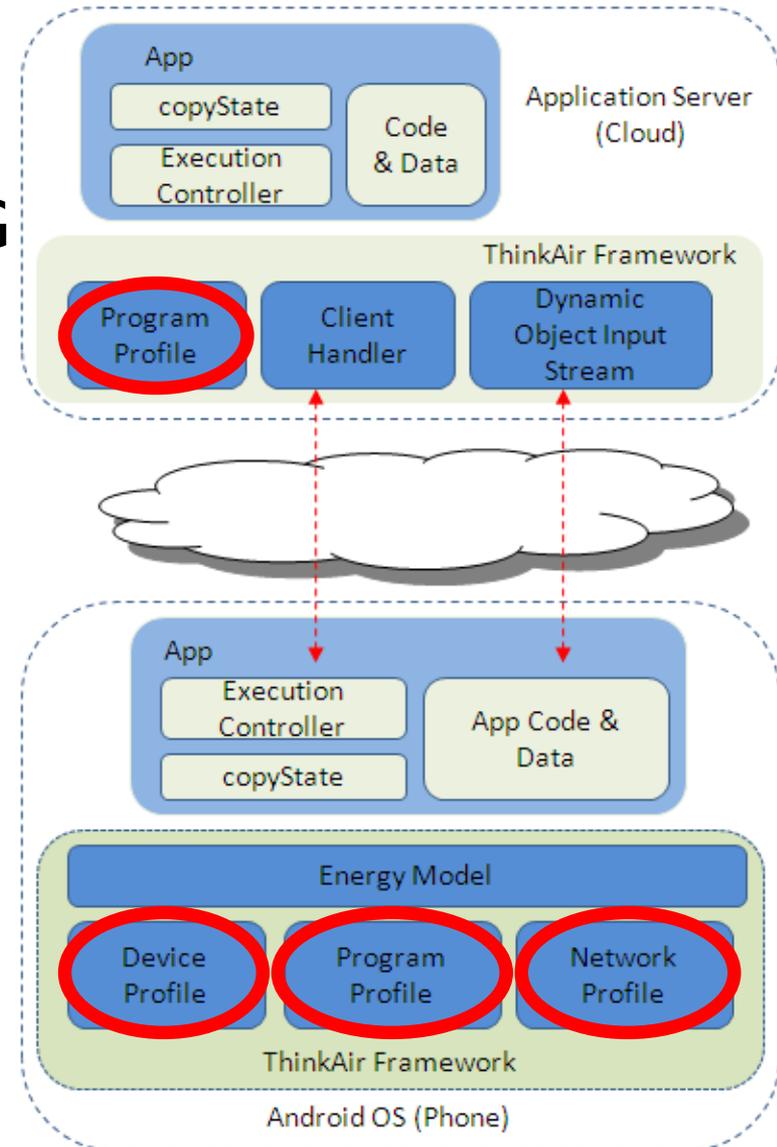
Cloud Infrastructure

- OS: customized version of Android x86.
- 6 types of VM.
- VM Resume latency:
 - *Paused*: 300ms
 - Up to 7s if too many VMs are resumed simultaneously.
 - *Powered-off*: 32s



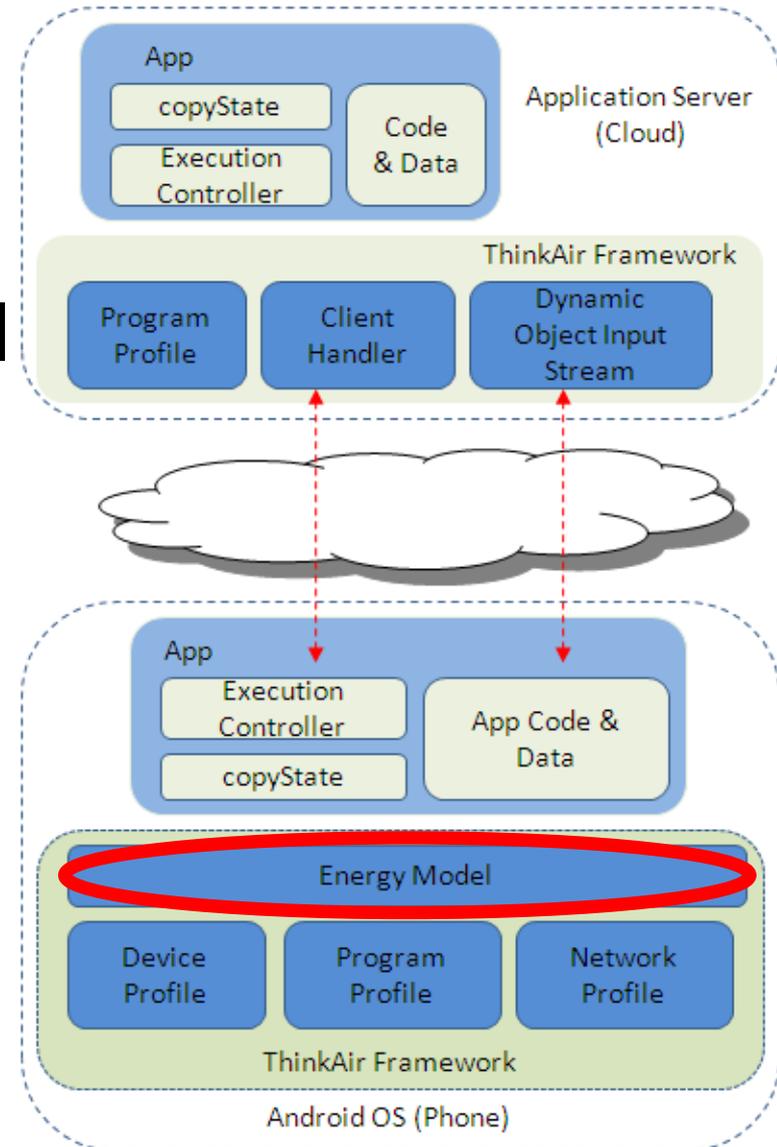
Profilers

- Hardware profiler
 - CPU, Screen, WiFi, 3G
- Software profiler
 - Use Android Debug API to record information.
- Network profiler



Energy Estimation Model

- Modify the original PowerTutor model.
- PowerTutor^[1] model
 - CPU, LCD screen, GPS, WiFi, 3G, and audio interface.
 - HTC Dream phone.



[1] Accurate online power estimation and automatic battery behavior based power model generation for smartphones, CODES/ISSS '10

Experiment Setup

- BIV(boundary input value)
 - The minimum value of the input parameter for which offloading would give a benefit.
- Offloading policy: *execution time*.
- Different Scenarios:
 - Phone
 - WiFi-Local (RTT 5ms)
 - router attached to cloud server.
 - WiFi-Internet (RTT 50ms)
 - 3G (RTT 100ms)

Micro-Benchmark^[2] Results

TABLE II

BOUNDARY INPUT VALUES OF BENCHMARK APPLICATIONS, WITH WiFi AND 3G CONNECTIVITY, AND THE COMPLEXITY OF ALGORITHMS.

Benchmark	BIV		Complexity	Data (bytes)	
	WiFi	3G		Tx	Rx
Fibonacci	18	19	$O(2^n)$	392	307
Hash	550	600	$O(n^2 \log(n))$	383	293
Methcall	2500	3100	$O(n)$	338	297
Nestedloop	7	8	$O(n^6)$	349	305

- Network latency clearly affects the BIV.

N-Queen Results

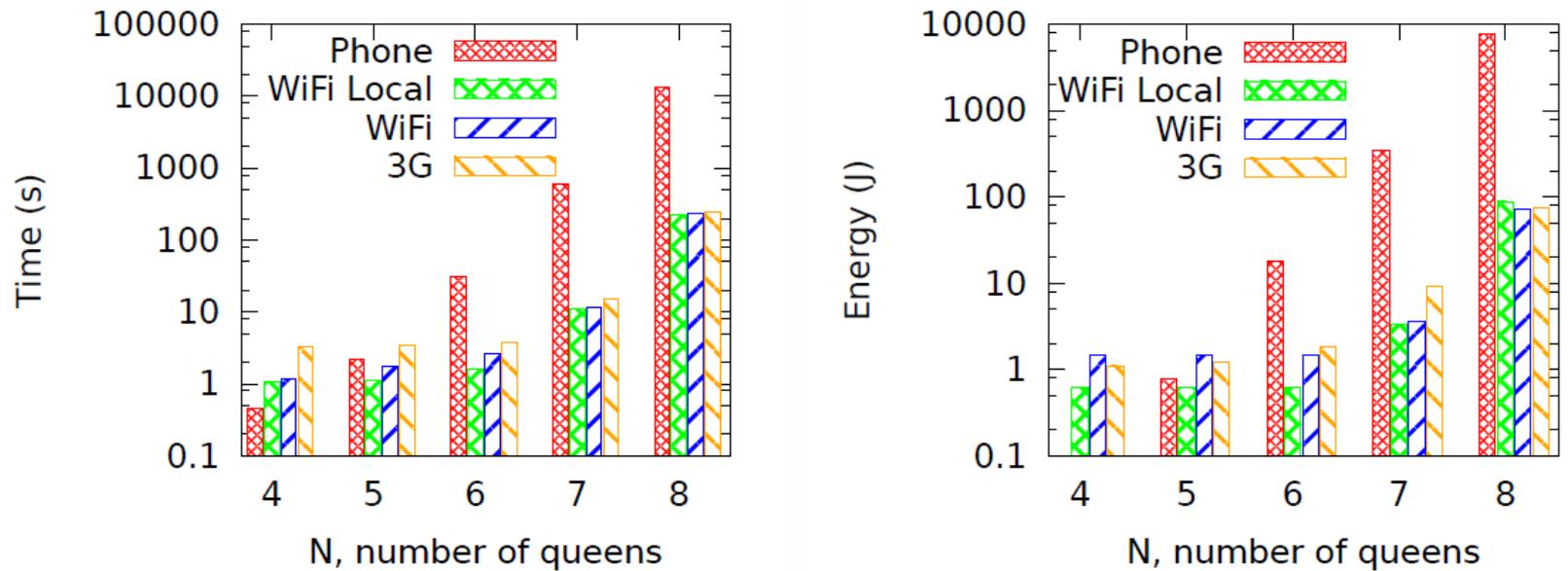


Fig. 2. Execution time and energy consumption of the N -queens puzzle, $N = \{4, 5, 6, 7, 8\}$.

- **BIV = 5**

N-Queen Results(Cont.)

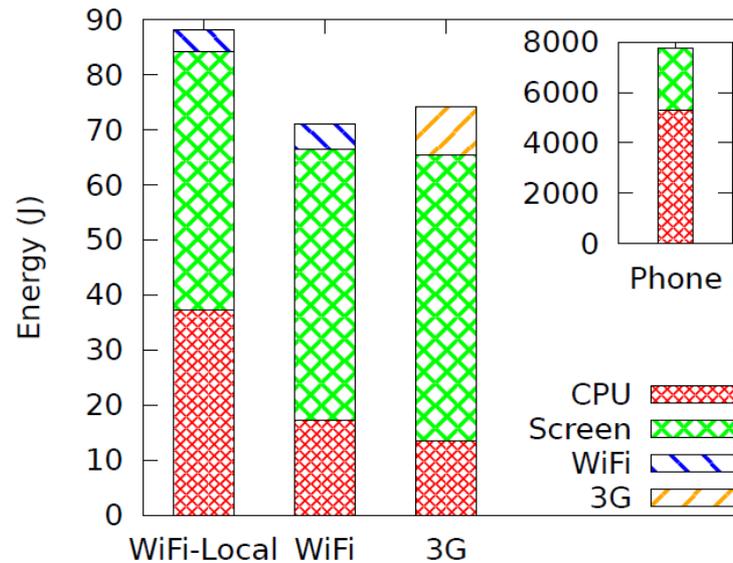


Fig. 3. Energy consumed by each component when solving 8-queens puzzle in different scenarios.

- $N = 8$
- Different CPU energy consumed
 - Due to bandwidth and latency of the links, and subsequently affected the time spent waiting for results and in transmission.

Face Detection Results

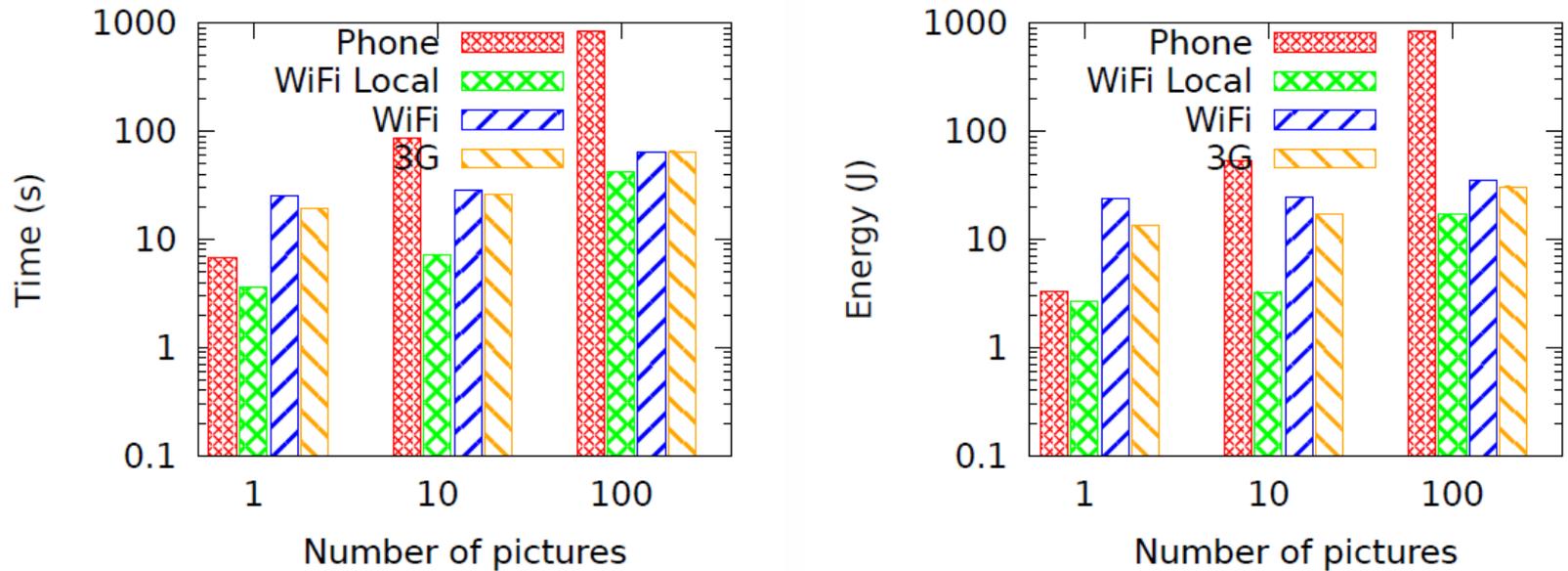


Fig. 4. Execution time and energy consumed for the face detection experiments.

- Counts the number of faces in a picture.
 - Photos are loaded in both device and cloud.

Face Detection Results(Cont.)

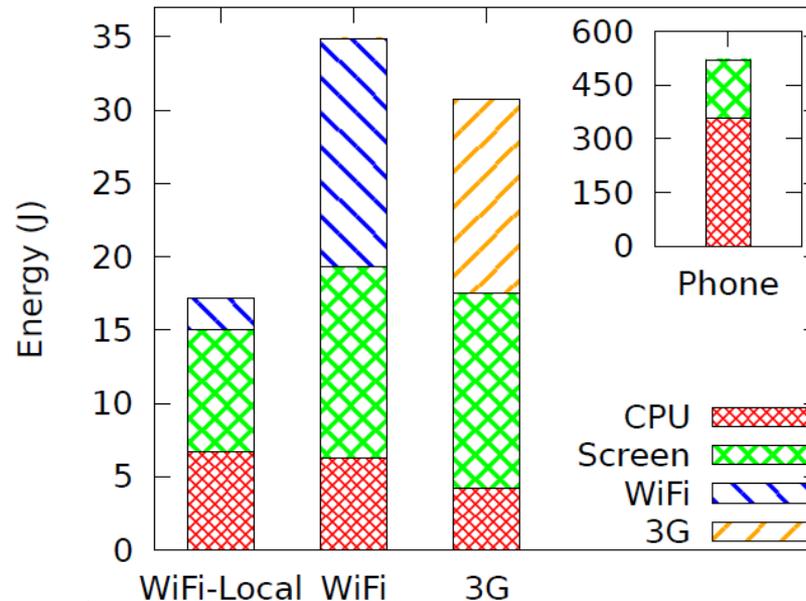


Fig. 5. Energy consumed by each component for face detection with 100 pictures in different scenarios.

- 100 pictures

Virus Scanning Results

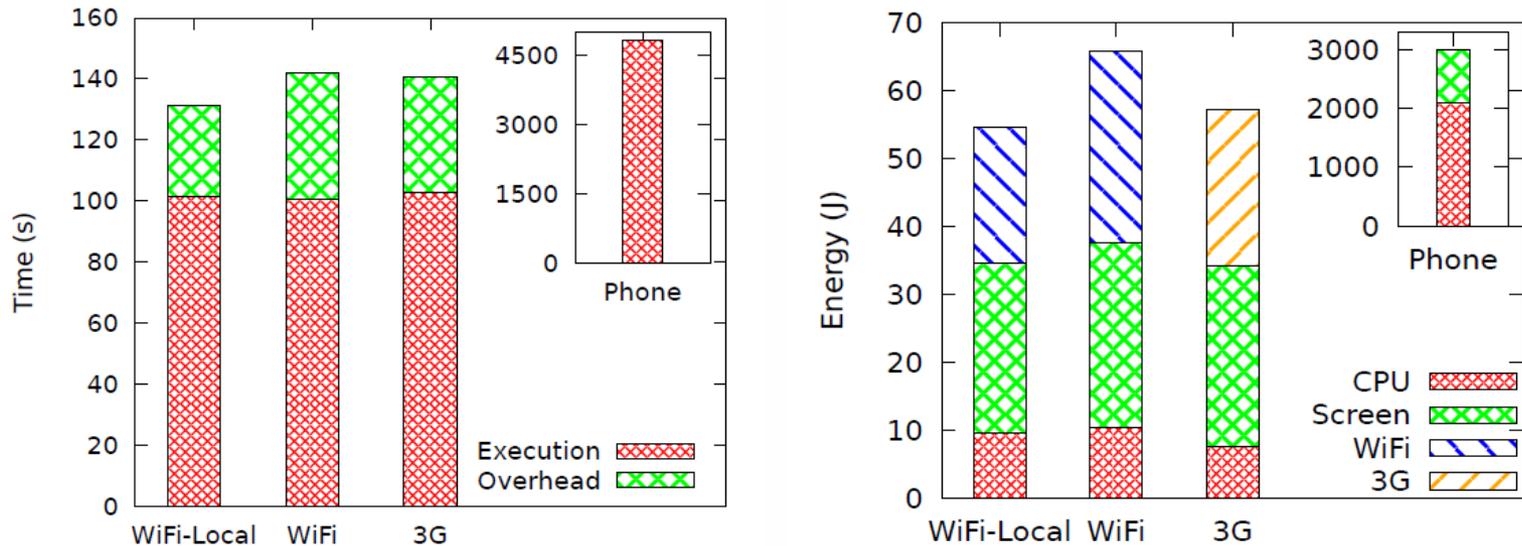


Fig. 6. Execution time and energy consumption of the virus scanning in different scenarios.

- Total size of files: 10MB
- Number of files: ~3,500
- CPU energy consumption is lower when offloading using 3G.

Parallelization with Multiple VM Clones

Clones

- Workloads are evenly distributed among VMs.
- Clones are resumed from *pause* state.

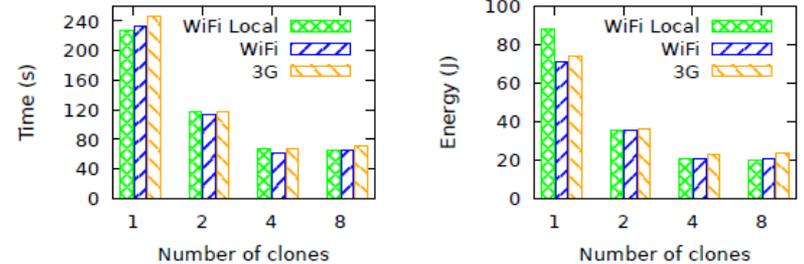


Fig. 7. Time taken and energy consumed on the phone executing 8-queens puzzle using $N = \{1, 2, 4, 8\}$ servers.

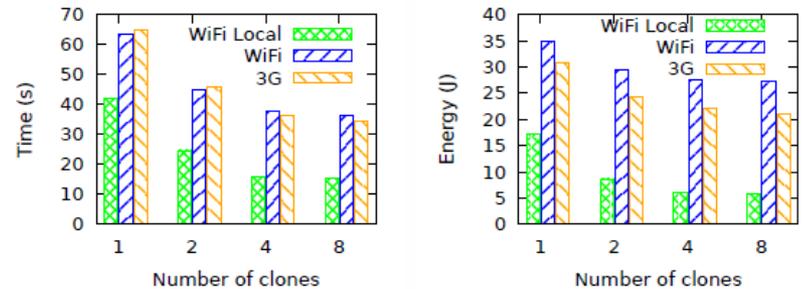


Fig. 8. Time taken and energy consumed for face detection on 100 pictures using $N = \{1, 2, 4, 8\}$ servers.

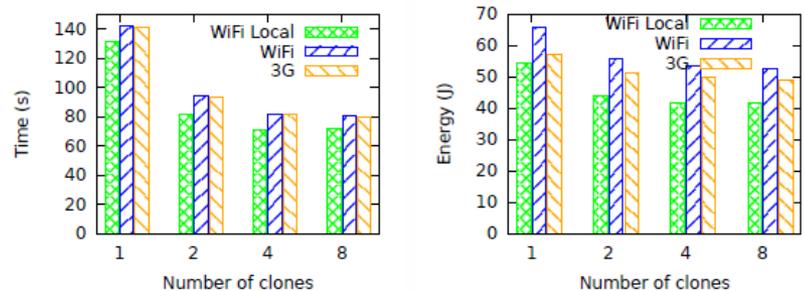


Fig. 9. Time taken and energy consumed for virus scanning using $N = \{1, 2, 4, 8\}$ servers.

Conclusion

- ThinkAir is a framework for offloading mobile computation to the cloud, with the ability of on-demand VM resource scaling.
- The authors will continue to work on improving programmer support for parallelizable applications, since they think it a key direction to use the capabilities of distributed computing of the cloud.