POMAC: **P**roperly **O**ffloading **M**obile **A**pplications to **C**louds

Mohammed Anowarul Hassan
George Mason University
Kshitiz Bhattarai
SAP Lab Palo Alto
Qi Wei and Songqing Chen
George Mason University

Outline

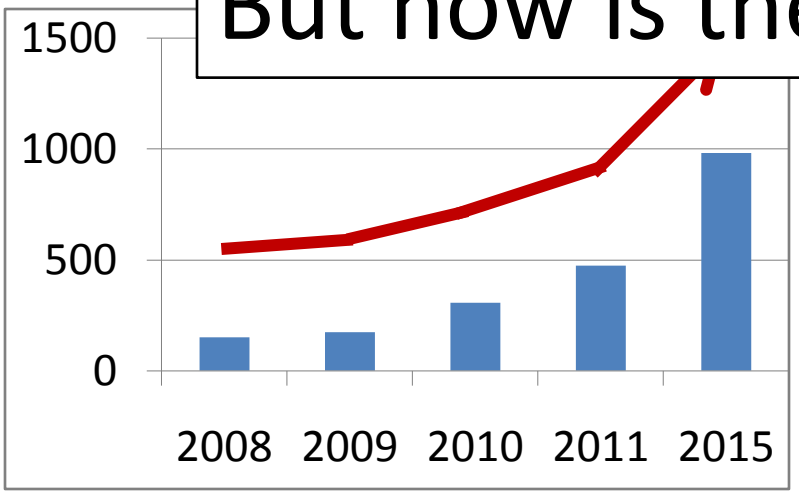
- Introduction
- Motivation
- Preliminary solution and evaluation
- Conclusion

Mobile devices replacing counterpart





But how is the performance?



Number of smartphone sold (*millions*)

International data corporation

More and more **resource-intensive applications for** mobile devices

Comparing smartphone and laptop

- **Lagging** behind modern **desktop** computers
- **iPhone5s** vs **MacBook Pro**

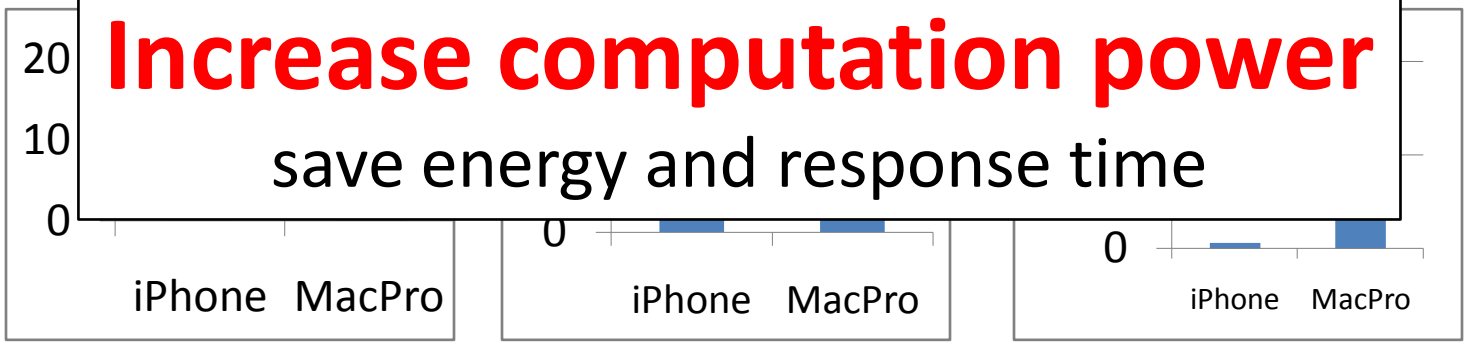




How to address?

Increase computation power

save energy and response time



RAM (*GB*) **CPU** (*GHz*) **Storage** (*GB*)

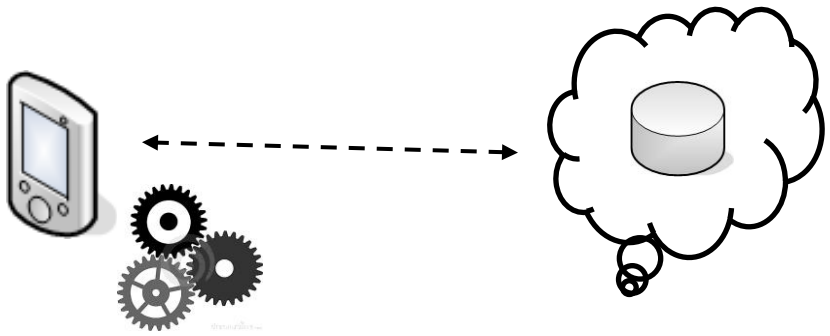




exhaust battery in **5 hours**

Optimize energy consumption

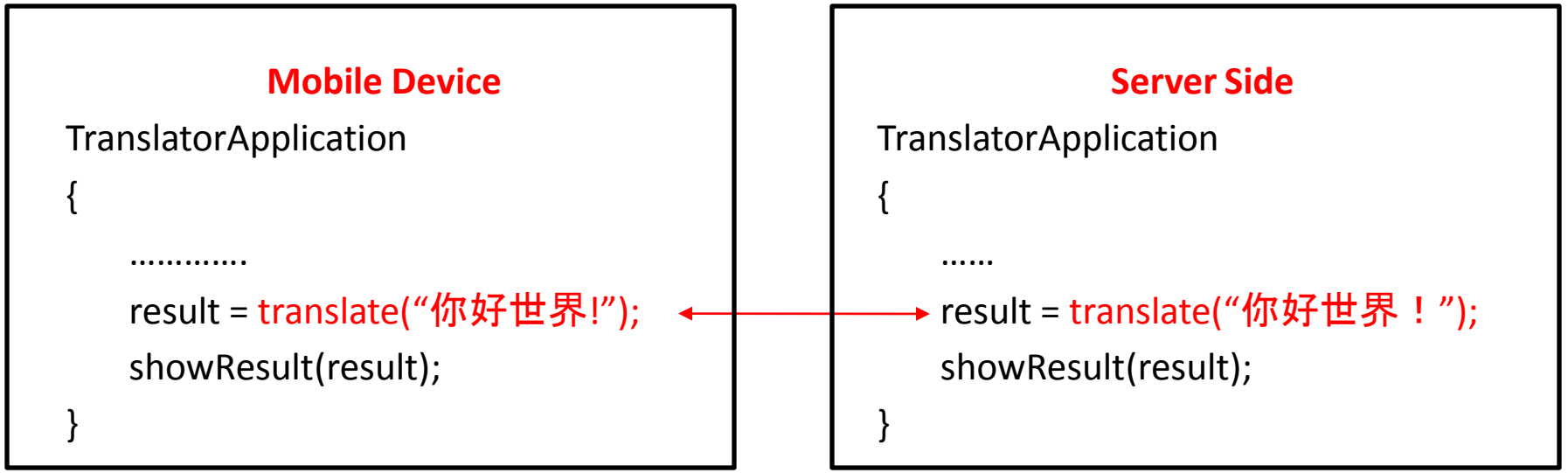
- **Hardware based solutions**
  - Dynamic Voltage Scaling : Mobile Device and Laptop
  - Dynamic Frequency Scaling: Microprocessor
- **Application specific solutions**
  - Use less battery-consuming devices whenever possible
  - Burst traffic pattern
- **Computation Augmentation : Cloud based solution**
  - VM based Clone
  - Task Partitioning



VM based cloning

Clone the entire phone image to the cloud:

- **VM Based Cloud-lets** [M. Satyanarayanan et al. *PerCom 2009*]
- **CloneCloud** [B. G. Chun et al. *HotCloud 2009, EuroSys 2011*]
- **Comet** [M. Gordon et al. *OSDI 2012*]



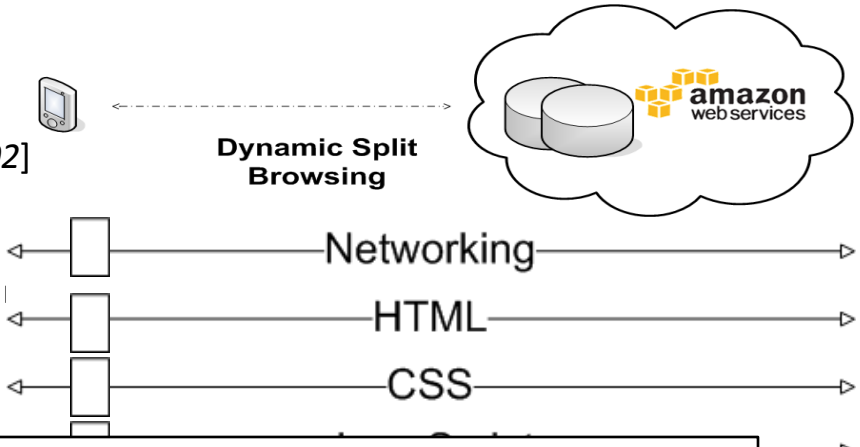
- High overhead for image synchronization
- ***Cloud-lets:*** ~100 MB

Task partitioning

Kindle: Silk

Partitioning the job and outsource to powerful machines:

- **Cyber Foraging** [Rajesh Balan et al. *ACM SigOPS 2002*]
- **MAUI** [Eduardo Cuervo et al. *MobiSys 2010*]
- **Odessa:** [M. Ra et al. MobiSys 2011]
- **Dynamic Deployment:** [I. Giurgiu et al. Middlewa 2012]
- **Power-efficient:** [Y. W. Kwon et al. ICDCS 2012]



Transparent to the Applications

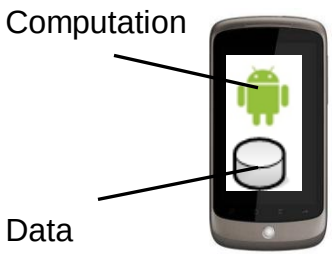
```
TranslatorApplication
{
    ……
    result = translate(“你好世界！”);
    showResult(result);
}
```

```
{
    …..
    sendToServer(“你好世界！”);
    showResult(result);
}
```

Decision making

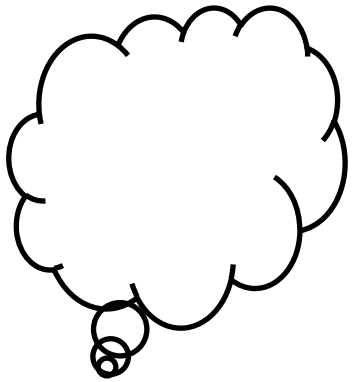
OnDevice execution > OnServer execution

```
TranslatorApplication
{
    result = translate("你好世界！");
    showResult(result);
}
```



Transfer time

Computation time



OnServer execution = **Transfer**+**Computation**

Decision making

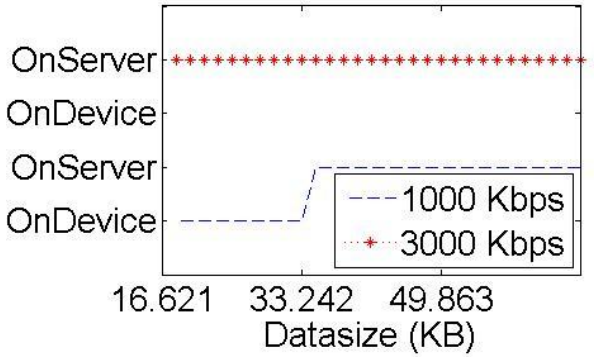
- Threshold Based:
  - **Power-efficient:** [Y. W. Kwon et al. ICDCS 2012]
- Data size > 6 MB

Single threshold value doesn't work

Mobile device

```
TranslatorApplication
{
    void translate(String data)
    {
        if(data.size > 6 MB)
            sendToServer(Data);
        else
            localExecution(Data);
    }
}
```

- Face recognition application
- Send an image to recognize face

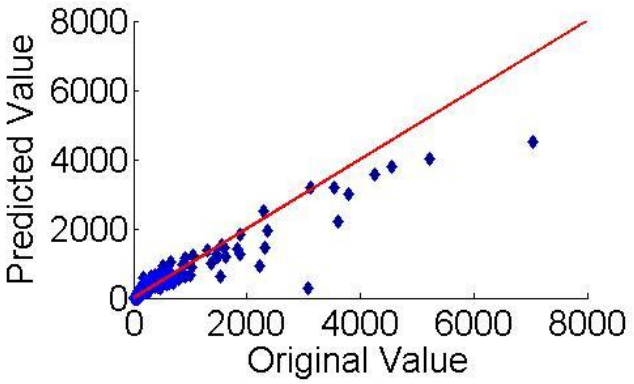


Decision making

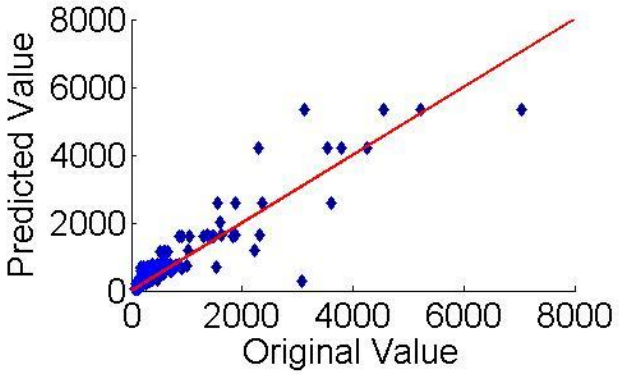
- **MAUI** [Eduardo Cuervo et al. *MobiSys 2010*]
  - Linear regression model
  - Predict and compare the OnDevice and OnServer execution time
  - Considered different factors:
    - Bandwidth
    - Latency

- Linear regression model too simplified
  - May predict wrong values

- Compare with better classifier: SVM



Linear



SVM

Decision making

- Intelligent classifier
- Consider many factors
  - Bandwidth
  - Latency
  - Data size
  - Server memory
  - Server CPU availability

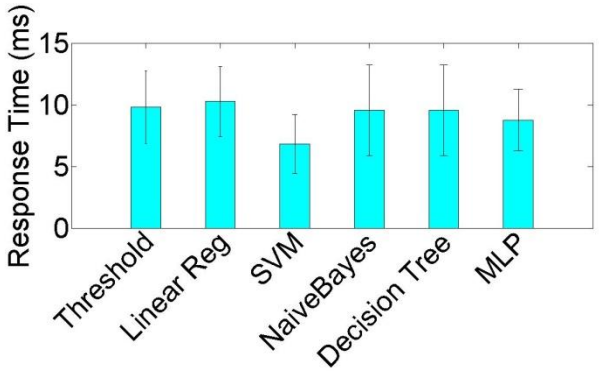
POMAC

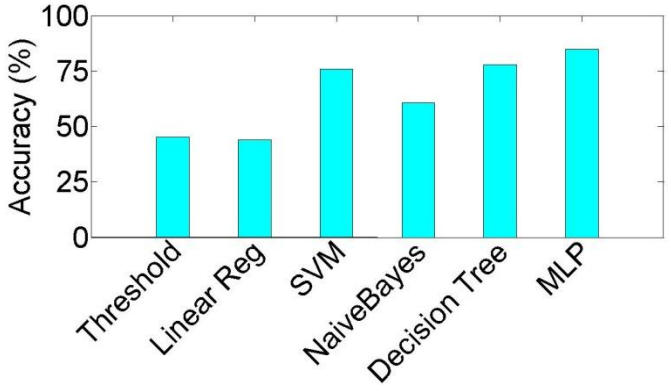
Properly **O**ffloading **M**obile **A**pplications to **C**louds

- Decision maker : Should know when to offload
- Offloading mechanism : transparent to the applications

Decision maker

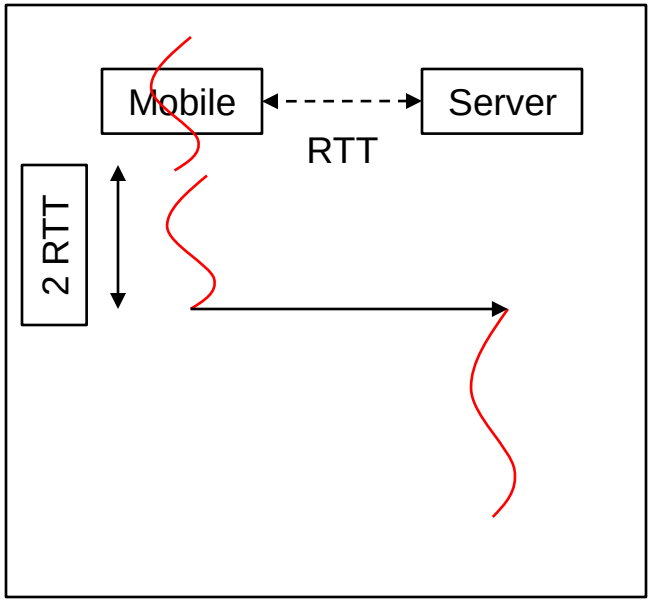
- Threshold
- Linear Regression
- Support Vector Machine
- Naïve Bayes
- Decision Tree
- Multi-Layer Perceptron



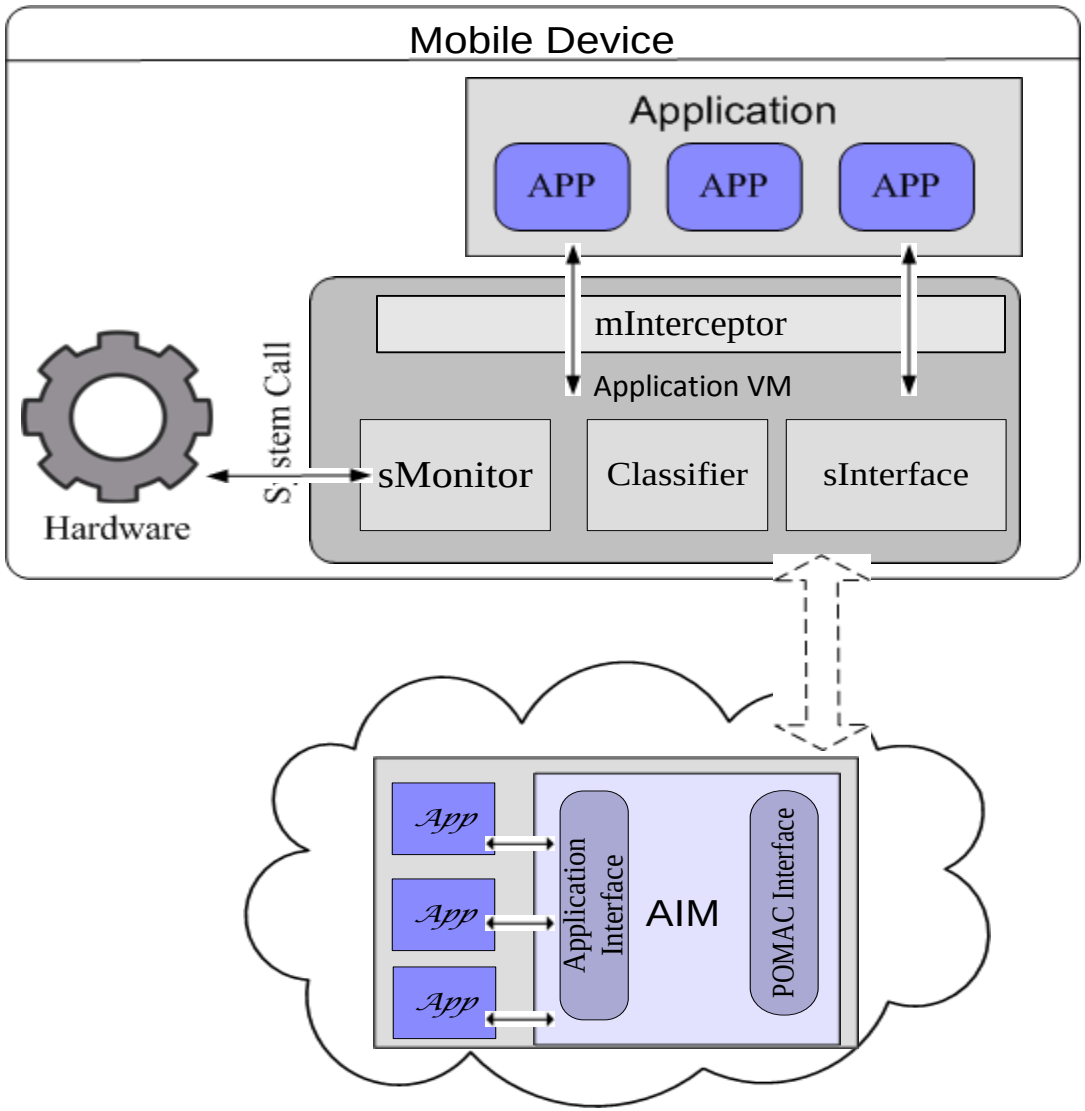


Offloading mechanism

- Modify application:
  - Modify existing source code [Simplifying Cyber Foraging Rajesh Balan et al. *Mobisys 2007*]
  - Modify binary [Y. W. Kwon et al. ICDCS 2012]
- Keep application unchanged - modify application VM:
  - Thread migration [Comet: Gordon et al. OSDI 2012]:
    - Synchronizing images : more overhead
    - Simple decision making : offloading when the so far thread execution time > 2 × RTT
  - Method interception: RPC in application VM !
    - Less overhead
    - Allows application profiling
    - Can make offloading decision efficiently



Implementation



Evaluation

Application	Offloading Candidate	Description
DroidSlator	translate	Translation app
Zxing	decodeWithState	Barcode reader
Mezzofanti	ImgOCRAndFilter	Optical character recognition
Picaso	project_and_compare	Face recognition app
MathDroid	computeAnswer	Calculator app

Evaluation

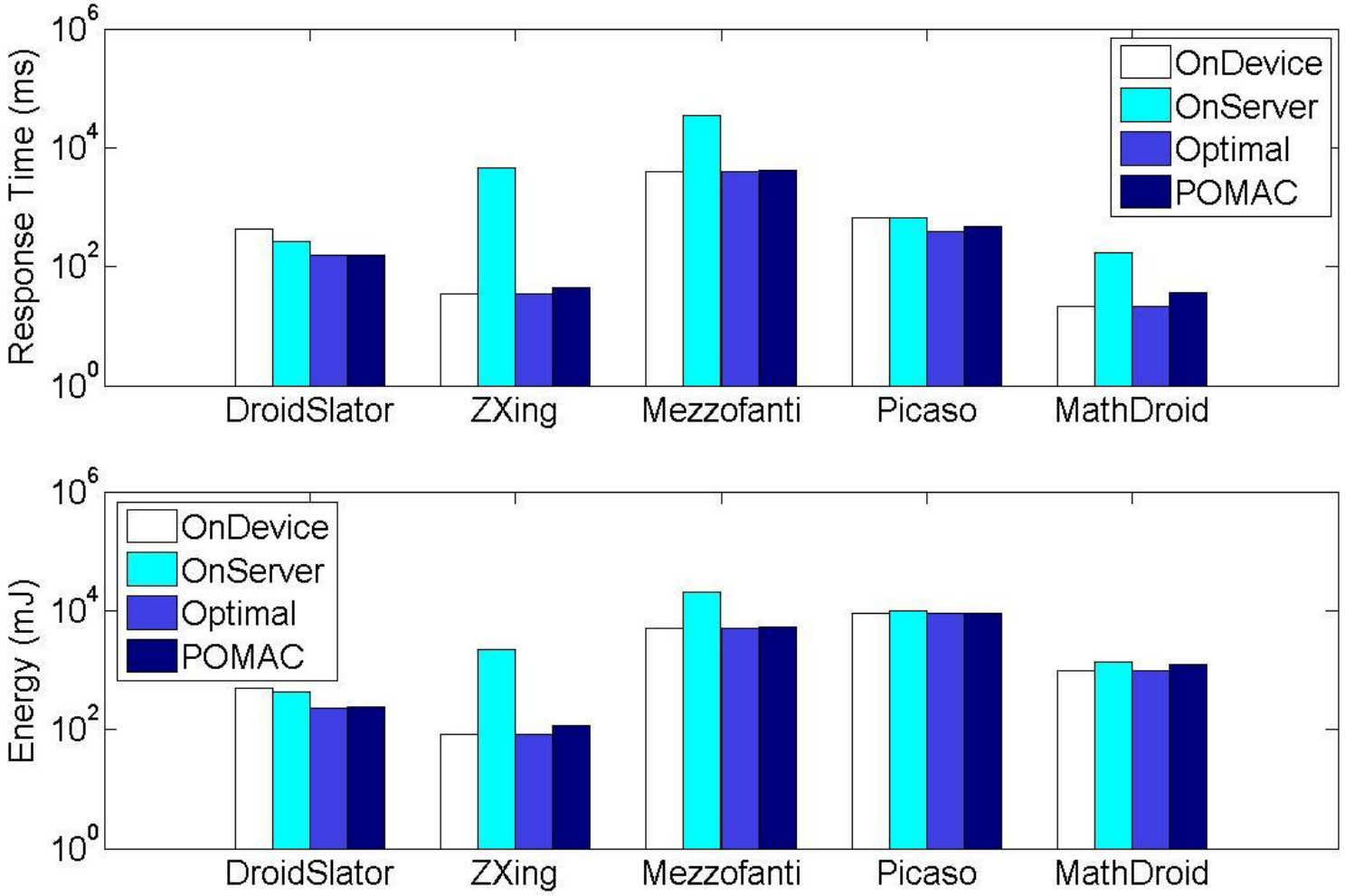
- Android nexus one
  - 1 GHz CPU
  - 512 MB memory
  - Cyanogenmod Passion distribution

- 5 Configurations:

	CPU	Memory	Latency	BW
LAN	2 GHz	1 GB	20 ms	100 Mbps
WLAN	1 GHz	2 GB	20 ms	30 Mbps
802.11	2 GHz	2 GB	50 ms	25 Mbps
4G	2 GHz	2 GB	75 ms	5 Mbps
3G	2 GHz	2 GB	200 ms	500 Kbps

Evaluation

MLP



Conclusion

- Mobile device constrained by limited power supply, CPU, and memory
- Computation can be augmented with cloud
- Proposed a framework for mobile applications to augment transparently with cloud

Thank You!

Question?