Building an efficient Mobile Cloud System for mobile cloud computing

Muhammad Haris
University of Helsinki, Finland
muhammad.haris@cs.helsinki.fi
Outline

• Why Cloudlets as a middle tier
• Reference Infrastructure and Architecture for MCC
• Abstract model For Mobile Data Offloading
• Security/Privacy and Access Control
• Conclusion
• References
Why Cloudlets as a middle tier

• Challenges in traditional cloud computing
  • High latency, less fault tolerant, less throughput

• A cloudlet is simply a trusted, resource rich computer or cluster of computers in the vicinity of mobile devices, that is well connected to the Internet for usage by mobile devices [1].
Properties of Cloudlet

- Cluster of Machines for Cyber Foraging
  - To offload part of data from poor machine to resource rich machine
- Low latency
  - because they are deployed in the vicinity of the mobile user
- Load balancing [2]
  - In case of load the data should be passed on to other cloudlet/cloud
- Data cache and storage
  - In case the cloudlet is not connected to cloud, the client still some data
- Fault tolerance
  - Backup for cloud, inform requester to arrange another cloudlet in case of failure
- Data foraging from cloud [2]
  - Cloudlet does pre filtering of the data from cloud, high throughput, low latency, high bandwidth
- Data source for the cloud/ mobile
Cloudlet Deployment

- Static Deployment
- Ad-hoc Deployment
- Hybrid Deployment
Static deployment
Ad-hoc deployment

Adhoc Cloudlet

Device 1 entering Adhoc cloudlet

Device 2

Device 3

Device 4

Only a stronger device will be a part of the cloudlet

(Stronger device: high processor, RAM, storage and battery life)
Hybrid deployment

Only a stronger device would be a part of cloudlet.
Reference Architecture of MCC

- Main components of architecture are Devices, Cloudlets, Cloud
- Architecture for Cloudlet Discovery and Formation
- Architecture for Mobile Data Offloading
Static Cloudlet Discovery and Formation
ZeroConf[2]

ZeroConf is a technique by which a client can consume services of the service provider without knowing its IP Address.
Ad-hoc Cloudlet Discovery and Formation
Hybrid Cloudlet Discovery and Formation

NM can be mobile device or resource rich machine

DS, CS will be present on all devices, while only DS will be present on resource rich device
Architecture for Mobile Data Offloading

• Existing Approaches for Mobile data Offloading
  • CloneCloud [3]
  • mClouds [4]
  • MAUI [5]
  • COMET [6]
  • ThinkAir [7]
  • Weblet [8]
  • Odessa [9]
  • Restful [10]
Existing Approaches for Mobile Data Offloading

• CloneCloud
  • As long as a thread can be executed faster in the cloud it should be migrated to cloud for processing
  • Static Analysis, Dynamic Profiling and Optimization solver
  • Offloading unit is thread

• mClouds
  • Forms an ad-hoc cloudlet with devices as mDevs
  • mDevs and Single master mDev
  • Offloading unit is component
Existing Approaches for Mobile Data Offloading

- MAUI
  - Annotated part of the code can be offloaded
  - Runtime profiler predicts the energy usage and decide offloading
  - Offloading unit is method

- COMET
  - Make use of Distributed Shared Memory (DSM) with as few interactions as possible
  - VM synchronization for restricting when migration occurs
  - Offloading unit multi-thread

- ThinkAir
  - Launches multiple VM’s for parallel processing or in case of overload
  - Offloading unit is method
Existing Approaches for Mobile Data Offloading

- Odessa
  - Applications should be structured as data-flow graphs, built on sprout: parallel processing framework
  - “Stages” are vertices
  - “Connectors” show data-dependencies between stages
  - Offloading unit is application task

- Weblet
  - Platform-independent entity represented as a stateless HTTP web service interface
  - Memory usage, battery level, processor decides whether to offload weblet or not.
  - Offloading unit is component(weblet)

- RESTful
  - Invokes a function with configuration on cloudlet through HTTP
  - Algorithms such as face recognition are packaged into cloud APIs
  - Offloading unit is cloud task
# Comparison of Mobile data offloading models

<table>
<thead>
<tr>
<th>Offloading model</th>
<th>Offloading unit</th>
<th>Scalability</th>
<th>Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>CloneCloud</td>
<td>thread</td>
<td>high</td>
<td>Yes</td>
</tr>
<tr>
<td>mClouds</td>
<td>component</td>
<td>high</td>
<td>No</td>
</tr>
<tr>
<td>MAUI</td>
<td>method</td>
<td>high</td>
<td>Yes</td>
</tr>
<tr>
<td>COMET</td>
<td>Multi-thread</td>
<td>high</td>
<td>No</td>
</tr>
<tr>
<td>ThinkAir</td>
<td>method</td>
<td>high</td>
<td>Yes</td>
</tr>
<tr>
<td>Weblet</td>
<td>component</td>
<td>high</td>
<td>No</td>
</tr>
<tr>
<td>Odessa</td>
<td>App task</td>
<td>low</td>
<td>Yes</td>
</tr>
<tr>
<td>Restful</td>
<td>Cloud task</td>
<td>low</td>
<td>No</td>
</tr>
</tbody>
</table>
Reference Mobile data offloading architecture

• What is missing in all theses models?
• Offloading unit can be anything
  • Component (method, thread, multi thread, task)
• Context
  • The context with which components are invoked [11]
• Our reference architecture will offload component along with their context to cloud/cloudlet
Framework For Context Aware Services in MCC [11]

Context Collector: collects the context from the device application

Adapter Determiner: analyzes the context and looks up candidate services and invoke adapted service

Generic Adapter: For making service personalized for a specific customer
Reference Mobile data offloading architecture

1. Parts of device side
Reference Mobile data offloading architecture

1. Parts of Cloudlet/Cloud
Scenarios in remote execution in cloudlets

Scenario 1: NM executing task on single VM

Node Manager (NM) → Virtual Machine (VM)

Scenario 2: NM executing task on two VMs in parallel

Virtual Machine (VM) → Virtual Machine (VM)

Scenario 3: NM executing task on Resource rich device

Virtual Machine (VM) → Resource Rich Device → Node Manager (NM)
Security and Privacy Challenges and Solutions

- Security and Privacy Challenges can be divided into three categories [12]
- Mobile Terminal
- Mobile Network
- Mobile Cloudlet/Cloudlet
Mobile Terminal

• Malware:
  • can be inserted to mobile with 3G/4G. USB, Bluetooth or MMS
  • Anti-viruses softwares were introduced on mobile
  • Move malware detection to cloud, CloudAV [14]

• Other issues
  • Software bugs may leak credit card information, user awareness should be increased
  • Operating System bugs, OS should regularly be updated
Mobile Network and Mobile Cloud/Cloudlet

• Information leaking, Stronger data Encryption, Security Protocol
• Data Protection, should have a proper authentication and authorization mechanism
• Access control [13]
  • Offload cryptographic algorithms to cloud to save battery
  • If less bandwidth is available offloading to cloudlet can be useful
  • Offloading decision (energy consumption, network status)
## Why access control to cloudlet?

<table>
<thead>
<tr>
<th></th>
<th>Mobile</th>
<th>Cloud</th>
<th>Cloud-Cloudlet</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input Data Size</strong></td>
<td>Response time Increased 275s – 2200s</td>
<td>Response time Increased 500s – 1600s</td>
<td>Response time Increased 200s – 1100s</td>
</tr>
<tr>
<td>Increased 6000 Bytes – 11000 Bytes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Bandwidth reduced</strong></td>
<td>Response time Increased 0.75 x 10^5 s – 9 x 10^5s</td>
<td>Remains constant 0.5 x 10^5 s – 0.5 x 10^5s</td>
<td></td>
</tr>
<tr>
<td>0.2Mbit/s – 3.4Mbit/s</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Input Data Size</strong></td>
<td>Encryption time Increased 0s – 55s</td>
<td>Encryption time Increased 0s – 44s</td>
<td>Encryption time Increased 0s – 17.5s</td>
</tr>
<tr>
<td>Increased 0 Mbits – 100 Mbits</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Input Data Size</strong></td>
<td>Decryption time Increased 0s – 147s</td>
<td>Decryption time Increased 0s – 81s</td>
<td></td>
</tr>
<tr>
<td>Increased 0 Mbits – 100 Mbits</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Adapting Reference Architecture for Access Control

1. Parts of device
Adapting Reference Architecture for Access Control

1. Parts of cloudlet/cloud
Conclusion

• Why Cloudlets are important in MCC, how they are formed and discovered
• What are the existing models for mobile data offloading
• How they can be improved to construct a reference MCC architecture
• What are the security and privacy issues in MCC
• How Access control architecture can be incorporated to the presented reference architecture
References


THANKS!

Questions?