A Virtual Cloud Computing Provider for Mobile Devices

Gonzalo Huerta-Canepa, Dongman Lee
1st ACM Workshop on Mobile Cloud Computing & Services

Presented by: Daniel Ogleja
Vrije Universiteit Amsterdam
Faculty of Sciences
March 21.03.2012
Problem

• Smart phones
  – are still resource constrained
  – are becoming pervasive
  – battery life still a problem

• Applications
  – demand more resources

• Cloud computing platforms
  – access is not guaranteed
  – could be too expensive
Outline

• Solutions
  – Existing cloud computing platforms
  – New framework
• Related work
• Mobile Devices - a virtual cloud computing provider
  – Motivation & Scenario
  – Design consideration
  – Architecture
• Implementation
• Evaluation
• Conclusions
One way...

• Cloud computing platforms
  – Amazon EC2, Microsoft Azure, Google Appengine

• Mobile device
  – Client or/and resource provider
  – Need a coordinator to manage the mobile devices and jobs

• Problem
  – Connection to the cloud
The other way...

- A framework to create virtual mobile cloud computing providers
- Detect nearby nodes
  - Stable nodes
- Create on the fly connection
  - Avoid a connection to infrastructure-based cloud
  - Maintain the main benefits of offloading
Motivation - on economic basis

• 2 costs for cloud computing providers
  – Networking cost
    • More than 200 USD for 1 GB downloaded
    • 3G connection consumes battery
    • 3G slower than Wi-Fi
  – Providers resources cost
    • 5 USD/month for small on-demand server for 2 hours/day
Motivation - on technical basis

• Benefits:
  – preserve conventional offloading benefits
  – increase performance by increasing the level of parallelism
  – communication overhead must not affect the overall performance
  – save energy
Scenario

• A group of people visiting South Korea
• Jim need to translate a text on a sign or from a museum
• Take a picture with the text
  1) Connect to internet – roaming costs
  2) Find other users interested in that text
  3) Create an ad-hoc network and process the images
Design – features

– Resource monitoring and management
  – Ex. A task can be executed locally?
– Seamless integration with the existing cloud APIs
  – mimic the same API on top of the ad hoc mobile P2P cloud
– A partition and offloading scheme suitable for mobile devices
  – Job splitting
– Activity detection to find users of the same or similar goals
  – Minimize potential disconnections
– Spontaneous interaction network support
  – discovery and selection of mobile devices
– A memory cache scheme to save intermediate results
– Lightweight and resource friendly architecture
Architecture

• The process for the creation and usage of a virtual cloud provider:

  – User must be at a stable place
  – Need more resources for a task
  – The system listens for nodes in vicinity
  – If available, the system intercepts the application loading and modifies the application in order to use the virtual cloud
Architecture – 5 main feature

- Application Manager
- Resource Manager
- Context Manager
- P2P Component
- Offloading Manager
Application Manager

• Launching and intercepting an application at loading time

• Modify the application to add features required for offloading
  • Proxy creation, RPC support

• modifying the reference to that provider with a reference to the virtual provider
Resource Manager

• In charge of application profiling and resource monitoring on a local device

• Creates a profile
  – Number of remote devices
  – Sensibility to privacy
  – Amount of resources needed for the migration
Context manager

- wields and synchronizes contextual information from context widgets

- Subcomponents
  - context widgets
  - context manager
  - social manager
P2P Component

• it sends events to the context manager in case a new device enters the space
• Ad hoc discovery mechanism
• groups the nodes using a P2P scheme
Offloading manager

• in charge of sending and managing jobs
• receiving and processing jobs
• in charge of detecting failures in the execution and to re-emit them
General architecture for the ad hoc mobile cloud

24/3/2012

A Virtual Cloud Computing Provider for Mobile Devices
Current Implementation

- Two sub-implimentations:
  - Cloud computing provider client
  - Ad Hoc mobile cloud framework
- Retroweaver – port Hadoop client to Java 1.4
- PhoneME, Mysafu, JamVM
- the map/reduce framework calls were replaced to RPC methods implemented using the Jabber RPC
Current Implementation

• Communication
  – Extensible Messaging and Presence Protocol (XMPP)

• Modified Yaja! (java XMPP client)
  – Serverless Messaging
  – Jabber RPC9
  – discovery and messaging among devices
Evaluation settings

• Input data - less than 100kb
• mobile devices with PhoneME and Mysaifu
  – lack of needed APIs
• jailbroken Ipod Touch with JamVM as the Java VM
• Hadoop 1.8 – four servers (OpenJDK VM 6)
• Communication - Ad Hoc WIFI, Access Point
• Korean OCR – for tests
Results

• Execution of tasks – slower (less 1%)
• offloading preparation and waiting time  
  – 44% of the execution time
• Processing time  
  – is approx. 56%
• Good performance with small files
• Saving processing time = saving energy
Problems

• Hadoop
  – low performance small files
  – mapred.job.reuse.jvm.num.tasks – modified (infinite number of reuse)
  – 2-3% improvement
  – Concatanation of input files
    • Not always possible – pictures
Performance of Mobile Offloading compared to local execution. Results are normalized to local execution (value 1).
Conclusion

- Mobile devices can be a virtual cloud computing provider