Towards An Elastic Application Model for Augmenting the Computing Capabilities of Mobile Devices with Cloud Computing

Presenter: Shravya Rukmannagari

Xinwen Zhang, Anugeetha Kunjithapatham, Sangoh Jeong, Simon Gibbs, Samsung Information Systems

Some material from Ruisheng Shi
Motivation

**Limited resources**
- Low frequency
- Small memory
- Battery

Cloud Computing

**SaaS, PaaS, IaaS**
Motivation (cont.)

IT View of Cloud Computing

Cloud = Web Service Platform

- Cloud is a platform for service delivery
- Push from services into devices

Service Provider Perspective
Motivation (cont.)

Proposed View of Cloud Computing

Cloud = Data/core/network center + API

- Cloud is a platform
- New applications that run across the cloud and device (“elastic applications”)
- API exposes cloud to device apps
- Expand the device into the cloud

User’s Perspective
Ongoing Approaches for Mobile + Cloud

- CloneCloud (HotCloud’09)
  - Clone of phone at cloud
- Splittig applications between device and cloud
  - Dynamic partitioning of applications (MCS’10)
  - Dynamic remote method invocation with managed code (Mobsys’10)
- Dynamic Composable Computing (HotMobile’08)
  - Dynamic composition of functions with mobile devices and surrogates
- Cloudlet (PVC’09)
  - Offloading VM to proximate infrastructure
  - 60-90 sen on VM synthesis
- HW-supported VM migration (Atom) (MobiCase’09)
  - Focus on mobility app
- Elastic Device/Application
  - On application level
  - Dynamic execution configuration
  - More flexible and easy for parallel
Motivation (cont.)

One approach to realize user’s perspective: Duplicate runtime environment of device in cloud (surrogate or clone or cloudlet) – Amazon EC2

- Clone requires physical hardware, I/O between device and clone
- Fails to take advantage of the cloud compute resources
- Increases complexity of device management – Security Protection and data privacy control.

**Soln:** Develop Elastic Application
The goal of the Elastic Device is to enable development of cross device/cloud applications. The advantages are:

- Remove device constraints, create new classes of powerful applications
- Help realize a new business model for device applications
- Provide developers a transition path to multi/many core
Elastic Device Concept

When device resources are not sufficient

New!

When device resources are sufficient
Elastic Applications (EA)

- EA are **cloud aware** applications

- **Weblets**
  - Define discrete application components
  - Communicate using REST interface
  - Run on device or cloud
  - Can be replicated to handle loads

- **Application GUI**
  - Launches the program
  - Directs the creation of new weblets

- **Manifest**
  - Meta-data of EA
  - Dynamic configuration info
  - Integrity of weblets
  - Policies for each weblet
    - E.g. JVM, network, access control, location
Elastic Devices (ED)

- ED support EAs
  - Enable seamless migration of weblets
  - Manage resources to optimize costs
  - Interface with cloud providers

- Elastic Manager
  - Spawns weblets in demand
  - Migrates weblets to/from cloud
  - Senses resource availability

- Cloud Fabric Interface
  - Exposes cloud services to devices
  - Controls weblets on behalf of EM
    - Start / Stop / Create / Destroy
  - Can provide PaaS or IaaS model
Application Model

- UI may be device dependent, possibilities:
  - Native code
  - HTML+CSS+JavaScript
  - Flash or Silverlight ...
- Weblets are device independent, possibilities:
  - Java bytecode
  - CLR bytecode
  - Python bytecode ...

- Autonomous
- Communicate via HTTP
- Long-living requests
- Dedicated (to a client)
- Persistent data
- Migratable
- Synchronizable

'h' - weblet request handler
'w' - weblet worker
Advantages of EA

• Many-to-one virtualization
  - Seamlessly expand and shrinks of platform capability
• Dynamic user experience
  - User control of expanding/shrinking based on factors such as battery consuming, monetary cost, latency/throughput, etc.
• Device flexibility
  - CE device computation and storage capabilities need not be designed to satisfy the most demanding applications.
• Dependindability
  - Migrating applications to cloud when device is low in battery/ weak signal
• Future proof:
  - Move app from cloud to device, extend app lifetime, reduce development cost
# Weblets vs Web Services

<table>
<thead>
<tr>
<th>Weblet</th>
<th>Web Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP (REST interface)</td>
<td>HTTP (REST or SOAP interface)</td>
</tr>
<tr>
<td>single client</td>
<td>many clients</td>
</tr>
<tr>
<td>short-lived &amp; long-lived requests</td>
<td>short-lived requests</td>
</tr>
<tr>
<td>dynamic endpoints (may migrate)</td>
<td>fixed endpoints (eg, <a href="http://www.google.com">http://www.google.com</a>)</td>
</tr>
<tr>
<td>lifetime is client dependent</td>
<td>lifetime is client independent</td>
</tr>
<tr>
<td>runs on servers or client (cloud or device)</td>
<td>runs on servers</td>
</tr>
<tr>
<td>push to client</td>
<td>NA / non-standard</td>
</tr>
</tbody>
</table>
Elasticity Patterns
Cost Model

Sensing (Inputs)
- battery level
- network quality
- device loads
- cloud loads
- aggregated performance data

Goal (examples)
- minimize costs
- maximize performance
- minimize power
- maximize robustness
- maximize security

Constraints
- resources
- cost model
- application requirements

Actions (Outputs)
1) allocation & migration (at launch & run-time)
   - weblet
   - cloud
   - device
   - WiFi
   - 3G

2) connection selection & switching
   - UI
   - w

3) replication
   - wi
   - (weblet pool)

4) shadowing
   - UI
   - w
   - w'
Cost-Optimal Execution Configuration

< Training System >

X (Status Vector)  Y (SWA configuration)

Log data
Run the training system for all configurations and collect related status vectors

Training data
$(x, y)_1$, $(x, y)_2$, $\ldots$, $(x, y)_n$

Cost Prediction

Offline computation

< User Device >

CPU Usage
Processing time
Memory Usage
Cloud usage cost

$p_1$, $p_2$, $p_3$, $p_4$, $p_M$

$p$ (User preference vector)

Preferred cost-optimal SWA configuration decision

$y$ (Suggested SWA configuration for the same-type devices)

(# of device SWA’s, # of Cloud SWA’s)
Reference Architecture

- Elastic application package including UI and weblets.
- Cloud nodes running on Amazon EC2 instances
- Web Service –based CFI
- Application installation on both cloud and device sides
SDK Development

- C# binding only so far
- A weblet is an independent functional unit of an application
- A weblet resembles an embedded or dedicated web server
  - presents a web service interface accessed via HTTP
- AppRoot is root UI of an application
- Actions are HTTP requests
Elastic Image Processing

on device: image processing
on cloud: image processing
Elastic Augmented Video

Samsung Q1

ElasticAV Application
(identify, track & replace “target” images)

ElasticAV App
Tracker Compositor
Camera

Splitter
Matcher 1
Matcher 2
Matcher 3

planar object recognition and replacement

on device: feature point extraction from video, tracking, compositing
on cloud: matching live features against library of target images
Elastic Augmented Reality

on device: using compass and GPS to align POI markers with live video from camera
on cloud: POI service and crowd simulator (gives # people in proximity to POI's)
Experimental Validation

- Elastic Image Processing application
  - Running on device only and on lab cloud cluster
  - LAMP as web service stack on each cloud node
  - Measure upload/ download bandwidth, workload, cpu usage, and available memory.
Throughput vs. Configurations
CPU usage rate vs. Configurations
Ongoing and Future Work

• Implementation of more general cost optimization
  - With more sensing data from both device and cloud
  - Cost model as a service

• Migration and replicate of code and data
  - Some synchronization protocols
  - Allow offline

• Security and privacy
  - Mutual authentication between weblets on device and cloud
  - Authorization delegation to weblets running on public cloud
Thank You!!