IMCM: Actor-Based Adaptive Framework for Mobile Hybrid Cloud Applications

Reza Shiftehfar, Kirill Mechitov and Gul Agha
Outline

- Mobile hybrid cloud framework
- Lightweight system monitor
- Policy-based distribution
- ACC demo scenario
Mobile cloud computing (MCC)

- Mobile applications are growing increasingly complex and resource-heavy

- Want to make use of cloud resources to overcome limitations of mobile devices
  - Hardware
  - Network access
  - Energy

- How? Use code offloading
MCC issues

How to implement code offloading?
- Rewrite application code
  - For specific cloud provider?
- Transparent full VM emulation in the cloud
  - High overhead, data use
- Some combination
  - Fine-grained offloading
  - Separation of concerns

How to deal with multiple cloud providers, including private clouds?
- Code migration and data access restrictions
- Security and privacy considerations
<table>
<thead>
<tr>
<th>Year</th>
<th>System Name</th>
<th>Goal</th>
<th>Offloading Decision</th>
<th>Partition Level</th>
<th>Parallel</th>
<th>Policy-based Security/Privacy</th>
<th>Manual Work</th>
<th>No. Cloud spaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010</td>
<td>MAUI</td>
<td>Mobile Energy Saving</td>
<td>Dynamic</td>
<td>Method</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>1</td>
</tr>
<tr>
<td>2011</td>
<td>CloneCloud</td>
<td>Mobile Energy Saving = Performance Improvement</td>
<td>Static</td>
<td>Method</td>
<td>Pseudo</td>
<td>No</td>
<td>No</td>
<td>1</td>
</tr>
<tr>
<td>2012</td>
<td>ThinkAir</td>
<td>Mobile Energy Saving = Performance Improvement</td>
<td>Dynamic</td>
<td>Method</td>
<td>Pseudo</td>
<td>No</td>
<td>Yes</td>
<td>1</td>
</tr>
<tr>
<td>2012</td>
<td>Cloud OS (COS)</td>
<td>Load Balancing for Cloud space</td>
<td>Dynamic</td>
<td>Actor</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Can be Many</td>
</tr>
<tr>
<td>2015</td>
<td>IMCM</td>
<td>Mobile Energy Saving, Performance Improvement, Combination for Applications</td>
<td>Dynamic</td>
<td>Actor</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Many</td>
</tr>
</tbody>
</table>
IMCM: mobile hybrid cloud framework

• Bridge the gap between mobile application and cloud computing
• Separate application logic from component distribution
• Allow dynamic distribution of components based on run-time changes
• Support policy-based definition of customized constraints
IMCM design and implementation

- Actor programming model
  - Natural Concurrency
  - Decentralization
  - No data races
  - Elasticity
  - Ease of scaling-up or -out
  - Location transparency
    -> Transparent migration

- Implementation: SALSA
  - Full actor semantics
  - Lightweight actors
  - Migration support
  - Java-based (portable)
IMCM system overview

- Max app performance
- Min mobile energy consumption
- Min cloud cost
- Min network data usage

Application Target Goal

- Application Actions
- Network parameters
- User context
- Application profiling
- Energy estimator

System Properties

Elasticity Manager

- Application Policy
- Access Restrictions
- User preferences

Org/App/User Policy

Application Component Distribution

Decision Maker

Policy Manager

- Target goal
- Profiled exec
- Profiled comm

System Monitor
Offloading decision: bandwidth

\[
\frac{w}{S_m} > \frac{d_i}{B} + \frac{w}{S_s}
\]

\[
w * \left( \frac{1}{S_m} - \frac{1}{S_s} \right) > \frac{d_i}{B}
\]

Diagram:
- Never offload
- Offloading decision depends on Bandwidth (B)
- Always offload

Communicated data (d_i) vs. Computation size (w)
Offloading decision: parallelism
Some benchmark results

- Running the same code on a faster machine
- Running some components in parallel
- Keeping in mind the cost of offloading

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Application Characteristic</th>
<th>Raw Speedup</th>
<th>Offload Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Comp.</td>
<td>Comm.</td>
<td>I/O</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>read</td>
</tr>
<tr>
<td>NQueen</td>
<td>intensive</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Image</td>
<td>intensive</td>
<td>limited</td>
<td>limited</td>
</tr>
<tr>
<td>Trap</td>
<td>intensive</td>
<td>limited</td>
<td>-</td>
</tr>
<tr>
<td>Virus</td>
<td>-</td>
<td>-</td>
<td>intensive</td>
</tr>
<tr>
<td>Rotate</td>
<td>-</td>
<td>-</td>
<td>intensive</td>
</tr>
<tr>
<td>ExSort</td>
<td>intensive</td>
<td>-</td>
<td>intensive</td>
</tr>
<tr>
<td>Heat1</td>
<td>limited</td>
<td>medium</td>
<td>-</td>
</tr>
<tr>
<td>Heat2</td>
<td>limited</td>
<td>high</td>
<td>-</td>
</tr>
</tbody>
</table>
Example: face recognition app
Example: face recognition app

- Components with different computation, bandwidth, energy characteristics
- Some data inputs/outputs may be restricted to specific systems or users
IMCM system overview

- Max app performance
- Min mobile energy consumption
- Min cloud cost
- Min network data usage

Application Target Goal

Org/App/User Policy

System Monitor

Application actions
Network parameters
User context
Application profiling
Energy estimator

System Properties

Elasticity Manager

Policy Manager

- Application Policy
- Access Restrictions
- User preferences

Offloading Plan

Decision Maker

Target goal
Profiled exec
Profiled comm

Application Component Distribution

Org/App/User

Application Policy
Access Restrictions
User preferences

Target Goal
Profiled exec
Profiled comm

Application

Application actions
Network parameters
User context
Application profiling
Energy estimator

IMCM system overview
IMCM system monitor

- **What to monitor?**
  - Application actions
    - At the level of actor primitives (create, migrate, send/receive)
  - System actions
    - Offloading decisions, costs
  - System/environment conditions
    - Performance, energy, connection speed, resource availability

- **When to notify?**
  - Changes in monitored parameters
  - Changes in org/user/system policies
  - Violation of existing policies
Example: face recognition app

- Energy use estimation based on simple model

<table>
<thead>
<tr>
<th>Time (sec)</th>
<th>Voltage Drop (V)</th>
<th>Energy (mJ)</th>
<th>No. Face Detector</th>
<th>No. Feature Extractor</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>0.0324</td>
<td>58350</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>100</td>
<td>0.0313</td>
<td>56340</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>161</td>
<td>0.0187</td>
<td>33700</td>
<td>3</td>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Component</th>
<th>Energy Consumption (mJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Face Detector</td>
<td>2668</td>
</tr>
<tr>
<td>Feature extractor</td>
<td>2019</td>
</tr>
</tbody>
</table>

![Graph](image)

Remaining Battery and Estimated Energy Consumption vs. Remaining Battery Charge

- **Graph**: Remaining battery voltage (V) and estimated total mobile energy consumption (J) vs. remaining battery charge (%)
IMCM system overview

- Application Component Distribution
- Application Target Goal
- Org/App/User Policy

Application actions
Network parameters
User context
Application profiling
Energy estimator

System Monitor

Elasticity Manager

Max app performance
Min mobile energy consumption
Min cloud cost
Min network data usage

System Properties

Policy Manager

Application Target Goal

Org/App/User Policy

Application Policy
Access Restrictions
User preferences

Offloading Plan

Target goal
Profiled exec
Profiled comm

Decision Maker

Application Component Distribution
Flexible policy-based restrictions

- Complete security solution requires:
  - Authentication
  - Authorization (access control)
  - Auditing

- Privacy issues
  - Entities involved
    - Owner organization
    - Programmers/developers
    - End-user
  - NIST SP 800-144: Guidelines of Security & Privacy of Public Cloud
IMCM rule definition & enforcement

- Requirements
  - Fine granularity
  - Rules defined by organization, developer, end-user
  - Covers all possible actions

- Action control system based on:
  - Attributes of the requester
  - Attributes of the resource
  - The requested action
  - The specified policy
IMCM rule definition & enforcement

- Authorization policy
  - Hard
    - Organization-wide policy
  - Soft
    - User/developer policy (app-instance-specific policy)

- Every application-instance comes with:
  - An immutable hard policy
  - A customizable soft policy

- Grammar to define rules
  - Attribute based
  - Static & dynamic binding
Example: face recognition app

- **Hard policy**
  - Different users:
    - Security personnel
    - General public
Example: face recognition app

- **Soft policy**
  - Different user expectations:
    - Type 1: No additional privacy concerns
    - Type 2: Privacy concerned users
    - Type 3: Extremely cautious users
Example: face recognition app

- Sample private cloud rules for restricted code and data access
  - E.g., no send, migration from private to public cloud

```
1. ActorSystem: {Name:ActorSysPrivate1, Static (URL:174.123.78.456, Port:1362)}
2. Actor: {Name:ActorPrivateGateway, Static (Reference: akka.tcp://app@174.123.78.456/privateGateway, ActorSystem:"ActorSysPrivate1")}
3. Actor: {Name:ActorPrivateVisasDB, Static (Reference: akka.tcp://app@174.123.78.456/Actor1, ActorSystem:ActorSysPrivate1)}
4. Actor: {Name:ActorPrivateResidentDB, Static (Reference: akka.tcp://app@174.123.78.456/Actor2, ActorSystem:ActorSysPrivate1)}
5. Actor: {Name:ActorPrivateCitizenDB, Static (Reference: akka.tcp://app@174.123.78.456/Actor3, ActorSystem:ActorSysPrivate1)}
6. Actor: {Name:ActorPrivateVisaProcessor, Static (Reference: akka.tcp://app@174.123.78.456/Actor4, ActorSystem:ActorSysPrivate1)}
7. Actor: {Name:ActorPrivateResidentProcessor, Static (Reference: akka.tcp://app@174.123.78.456/Actor5, ActorSystem:ActorSysPrivate1)}
8. Actor: {Name:ActorPrivateCitizenProcessor, Static (Reference: akka.tcp://app@174.123.78.456/Actor6, ActorSystem:ActorSysPrivate1)}
10. AnonymousActorSystems: {Name:Other-ActorSys-Private, URL:174.123.78.456, Creation:FORBIDDEN}
13. Rule: {Name:Private-Rule-3, Subject (Actor:ActorPrivateGateway), Object (ALL), Actions: SEND-TO, RECEIVE-FROM, Permission: ALLOWED}
16. ActorSystem: {Name:ActorSysPrivate2, Static (URL:174.123.78.456, Port:1369)}
```
IMCM: Summary

- Provides an adaptive solution for mobile cloud application development
- Allows programmers to focus on application logic
- Requires minimal manual tuning
- Provides full parallelism
- Supports different target offloading goals
- Allows use of multiple hybrid cloud spaces
- Allows customizable privacy policies for users
- Allows mandatory security policies for organizations
ACC demo scenario
ACC demo scenario

- Often the first information from a disaster area comes from mobile photos and videos
- Some of this stream of data may be of use to first responders
  - Damage assessment
  - Location of downed power lines
  - Location of people to be evacuated
  - Etc.
ACC demo scenario

- Users can run an app that will automatically share some relevant information with the responders
  - User concerns:
    - Privacy
    - Energy use
  - Other concerns:
    - Bandwidth
    - Relevance
- Military may make some private cloud resources available for this to supplement/replace public clouds
  - Security concerns:
    - Data leaks
    - Attacks