Customizable and Extensible Deployment of Mobile/Cloud Applications

Irene Zhang
Adriana Szekeres  Dana Van Aken  Isaac Ackerman,
Steven D. Gribble  Arvind Krishnamurthy  Henry M. Levy

University of Washington
Once upon a time, applications were single user, single platform, and single node.
Today, applications are ... multi-user, multi-platform and multi-node
Application programmers face new challenges.

How do I coordinate data and computation across nodes and platforms?

How do I hide performance limitations and failures?

How do I manage different programming environments and hardware resources?
Application programmers face new challenges.

How do I coordinate data and computation across nodes and platforms?

How do I hide performance limitations and failures?

How do I manage different programming environments and hardware resources?

These sound like distributed systems problems!
Application programmers face new challenges.

How do I coordinate data and computation across nodes and platforms?

How do I hide performance limitations and failures?

How do I manage different programming environments and hardware resources?

These sound like distributed systems problems!
Application programmers have different requirements.

- I need reliable RPC.
- No, I want caching!
- I need distributed objects.
- But, I want replication!
- I need code-offloading.
Application programmers have different requirements.

I need reliable RPC.

No, I want caching!

I need distributed objects.

But, I want replication!

I need code-offloading.

They are left implementing distributed deployment tasks!
Application programmers have different requirements.

I need reliable RPC.

No, I want caching!

But, I want replication!

I need code-offloading.

They are left implementing distributed deployment tasks!
A new programming system for deploying mobile/cloud applications.
Our Goals

1. Separate application logic from deployment code.

2. Allow programmers to easily choose and change application deployment.
Our Solution

A new system architecture that supports pluggable and extensible deployment managers.
Outline

1. Sapphire Architecture
2. Deployment Managers
3. Experience and Evaluation
Sapphire Architecture

Sapphire Application

Deployment Management Layer

Deployment Kernel

Android OS

iOS

DK Server

DK Server

DK Server
Partitioned into *Sapphire Objects*, which:

- Run in a single address space with RPC.
- Execute anywhere and move transparently.
- Provide a unit of distribution for deployment managers.
Sapphire Architecture

Deployment Management Layer

Deployment Kernel

Android OS

iOS

DK Server

Mobile App
Provides best-effort distribution services, including:

• Sapphire object tracking, mobility and replication.

• Making and routing RPC to Sapphire objects.

• Managing, distributing and running deployment managers.
Sapphire Architecture

Deployment Management Layer

DK Server

DK Server

OTS Server

Android OS

OS

OS
Consists of deployment managers, which:

• Extend the functions and guarantees of the deployment kernel.

• Interpose on Sapphire object events.

• Easy to choose and change without modifying the application.
Sapphire Architecture
Sapphire Deployment Manager Library

- **Primitives**
  - Immutable
  - AtLeastOnce RPC
  - Keep In Place
  - Keep On Device
  - Keep In Cloud

- **Caching**
  - Explicit Caching
  - Lease Caching
  - Writethrough Caching
  - Consistent Caching

- **Serializability**
  - Serializable RPC
  - Locking Transactions
  - Optimistic Transactions
  - Durable Transactions

- **Checkpoint**
  - Explicit Checkpoint
  - Periodic Checkpoint

- **Replication**
  - RSM-Cluster
  - RSM-Geo
  - RSM-P2P

- **Mobility**
  - Explicit Migration
  - Dynamic Migration
  - Explicit Code-offload

- **Scalability**
  - LoadBalanced Frontend
  - Scale-up Frontend
  - LB Master-slave
## Sapphire Deployment Manager Library

### Primitives
- Immutable
- AtLeastOnce RPC
- Keep In Place
- Keep On Device
- Keep In Cloud

### Caching
- Explicit Caching
- Lease Caching
- Writethrough Caching
- Consistent Caching

### Serializability
- Serializable RPC
- Locking Transactions
- Optimistic Transactions

### Checkpoint
- Explicit Checkpoint
- Periodic Checkpoint

### Replication
- RSM-Cluster
- RSM-Geo
- RSM-P2P

### Mobility
- Explicit Migration
- Dynamic Migration
- Explicit Code-offload

### Scalability
- LoadBalanced Frontend
- Scale-up Frontend
- LB Master-slave

### Extensible with the Sapphire Deployment Manager API!
Outline

1. Sapphire Architecture
2. Deployment Managers
3. Experience and Evaluation
Deployment Manager API

Deployment manager components, which the Sapphire kernel creates, deploys and invokes:

- **Instance Manager**: Co-located with the Sapphire Object.
- **Proxy**: Co-located with remote references to the Sapphire Object.
- **Coordinator**: Co-located with fault-tolerant Object Tracking Service (OTS).
Deployment Manager Architecture
Deployment Manager Architecture
Replicating a Sapphire Object
Replicating a Sapphire Object

Sapphire Object
Instance Manager
DK Server
OS
App
Leader
DK Server
OS
App
DK Server
OS
App
DK Server
OS
App
Replicating a Sapphire Object
Replicating a Sapphire Object
Replicating a Sapphire Object
Replicating a Sapphire Object
Replicating a Sapphire Object
Caching a Sapphire Object
Caching a Sapphire Object
Caching a Sapphire Object
Caching a Sapphire Object
Outline

1. Sapphire Architecture
2. Deployment Managers
3. Experience and Evaluation
## Experimental Setup

<table>
<thead>
<tr>
<th></th>
<th>Dell Server</th>
<th>Nexus 7</th>
<th>Nexus S</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Intel Xeon 8-core</td>
<td>ARM Cortex A9 4-core</td>
<td>ARM Cortex A8 1-core</td>
</tr>
<tr>
<td></td>
<td>2GHz</td>
<td>1.3GHz</td>
<td>1GHz</td>
</tr>
<tr>
<td>RAM</td>
<td>8GB</td>
<td>1GB</td>
<td>512MB</td>
</tr>
<tr>
<td>OS</td>
<td>Ubuntu</td>
<td>Android</td>
<td>Android</td>
</tr>
</tbody>
</table>
Peer-to-Peer Multiplayer Game

- Read
- Write

Time in milliseconds:
- Player 1: 0, 8, 15, 23, 30
- Player 2: 0, 8, 15, 23, 30

Options:
- Keep In Cloud
- Keep On Device
- RSM-P2P
Code-offloading for Physics Engine

- **Base**
  - WiFi: 0 milliseconds
  - 4G: 1190 milliseconds

- **Execution**
  - Network
  - Execution

- **Phone**
  - Base
  - WiFi
  - 4G

- **Tablet**
  - Base
  - WiFi
  - 4G
Modern applications implement difficult distributed deployment tasks.

Sapphire is a new programming system for deploying mobile/cloud applications.

Deployment managers makes it easy to choose, change and build deployment.