Slingshot: Deploying Stateful Services in Wireless Hotspots

Ya-Yunn Su
Jason Flinn
University of Michigan
Motivation

- Portable: take everywhere
  - Easy to carry and less obtrusive
  - Limited in resources

- Performance: run demanding applications
  - More processing power and storage capacity
  - Bulkier and heavier
Remote Execution

Wireless Hotspot

Limited bandwidth

AP

VNC client

Internet

VNC Server

High latency
Cyber Foraging

Internet

Wireless Hotspot

AP

Surrogate

VNC client

Home Server

VNC VM
Migrating Remote Services

Problems:
1. No service during migration
2. Loss state on surrogate failure
Slingshot: Replicate Services

- Home replica always available
- No state loss on surrogate failure
Ease of Management

• Surrogates should be appliances

• Slingshot
  – Minimizes the surrogate computing base
  – Uses a heavyweight virtual machine
  – Places no hard state on surrogates
Outline

• Motivation
• Implementation
• Evaluation
• Related Work
• Conclusion
Slingshot Overview

- Slingshot applications
  - Remote desktop: VNC
  - Speech recognition: IBM ViaVoice
State of a Remote Service

• The virtual machine state contains:

  **Volatile state:** memory image and registers
  – Unique to each service
  – Compressed and stored as individual files

  **Persistent state:** virtual disk image
  – Large: ex. 4 GB for our VNC service
  – Stored in content addressable database
Content Addressable Database

Disk Blocks

<table>
<thead>
<tr>
<th>Block #</th>
<th>SHA-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>ABC</td>
</tr>
</tbody>
</table>

Chunk Table for VNC

<table>
<thead>
<tr>
<th>Block #</th>
<th>SHA-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>ABC</td>
</tr>
</tbody>
</table>

Hash Table

<table>
<thead>
<tr>
<th>SHA-1</th>
<th>Ref count</th>
<th>offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC</td>
<td>2</td>
<td>18</td>
</tr>
</tbody>
</table>

Chunk Database

- This runs on the home server
- Built upon Internet Suspend Resume

[1Kozuch02]
Persistent state can be
- Fetched on demand
- Shared between applications
Creating a Replica

• Big idea: recreate current state from
  – **Checkpoint** on the home server
  – **Event log** on the client
• Application level determinism
• Example:

  Keystroke
  Keystroke
  Keystroke
  Mouse movement

**Checkpoint** + **Event Log** = **Current State**
Instantiate a Replica

1. Checkpoint

2. Transfer volatile state and chunk table

SLOW!

3. Replay event log

VNC proxy

VNC client

VNC VM

Surrogate

Home server

2. Log requests
Leveraging Portable Storage

1. Log requests
   - Portable storage carries:
     - Volatile state
     - Content addressable database

2. Replay event log

1. Transfer volatile state, chunk table and hash table

Surrogate

Home server
Outline

• Motivation
• Design Principles
• Implementation
• Evaluation
• Related Work
• Conclusion
Network Topology

- Workload: open Word, insert text, save document and close Word

Surrogate

1.5 Mbps

30 ms latency

Internet

1.5 Mbps

256 Kbps

1.5 Mbps

Home Server
Benefit of Slingshot

- Slingshot: 2.6 times faster than remote execution
Network Topology

Surrogate Home Server

- 30 ms latency
- 1.5 Mbps

AP

Handheld

- 256 Kbps
- 1.5 Mbps

VNC

VM

Internet
Instantiating the First Replica

- Slingshot executes 2.6 times faster than remote execution
Network Topology

Nearby Surrogate

Distant Surrogate

Home Server

Handheld device

AP

VM

VNC

1.5 Mbps

latency

15 ms

Internet

1.5 Mbps

latency

15 ms

256 Kbps

1.5 Mbps
Instantiating Another Replica

- Remote execution
- Without microdrive
- With microdrive

Response Time (seconds) vs. Elapsed Time (minutes)
Related Work

• Cyber foraging [Balan03, Goyal04]
  – Support user mobility and stateful services

• Virtual machine/process migration
  [Sapuntzakis02, Kozuch02, Tolia03]
  – Apply the same optimization techniques
  – Replicate VM to service mobile computers

• Replay at different level
  [Dunlap02, Bressoud95, Rodrigues01, Brown02]
  – Enforce determinism at the application level
Conclusion

• Slingshot
  – Is 2.6x faster than remote execution
  – Hides surrogate failure
  – Minimizes surrogate maintenance cost

• Questions?