Executing mobile application on the cloud: Framework and issues

Shih-Hao Hung, Chi-Sheng Shih, Jeng-Peng Shieh, Chen-Pang Lee, Yi-Hsiang Huang

Computer and Mathematics with Application 63 (2012)

Ting-Yi Lin

Outline

- Introduction
- Related work
- A virtual environment for Android applications
- Probabilistically guaranteed communication
- Conclusions

Why design such system?

- Latest mobile devices are still constrained by power consumption, speed of computation, size of memory…

- Wireless network is more and more popular, mobile device can overcome the constraints by offloading to cloud server

Mobile cloud computing issues

- Application re-design and deployment
- Network condition and service availability
- Control of application
- Privacy of data
- Information security

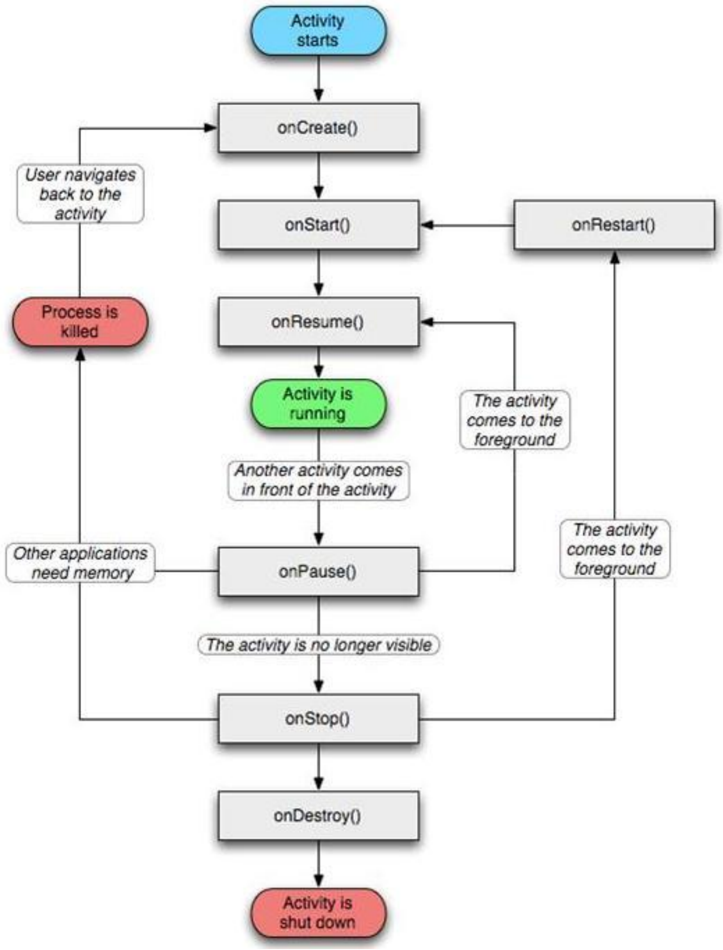
To address aforementioned issues

- Proposed a framework for a user to create a virtualized execution environment in the cloud for running mobile applications
- Use Android’s state saving mechanism and categorized the types of application data to decide on the necessity and the priority for data synchronization
- The communication framework creates several virtual network devices. Each device provides a specific QoS guaranteed communication channel.

Outline

- Introduction
- Related work
- A virtual environment for Android applications
- Probabilistically guaranteed communication
- Conclusions

Android framework



Activity starts
onCreate()
User navigates back to the activity
onStart()
onRestart()
Process is killed
onResume()
Activity is running
The activity comes to the foreground
Another activity comes in front of the activity
The activity comes to the foreground
Other applications need memory
onPause()
The activity is no longer visible
onStop()
onDestroy()
Activity is shut down

Other related works

- Replay system
  - deterministic replay

- Existing works does not addressed the need for offload with a limited network bandwidth and control / privacy / secure issues

- QoS guarantee
  - RSVP: reservation for data flows along a data path
  - IntServ: best-effort traffic model
  - de Niz and Rajkumar: resource reservation model to support real-time communication
  - Design the framework that provides QoS on system level

Outline

- Introduction
- Related work
- A virtual environment for Android applications
- Probabilistically guaranteed communication
- Conclusions

Creating a virtual environment

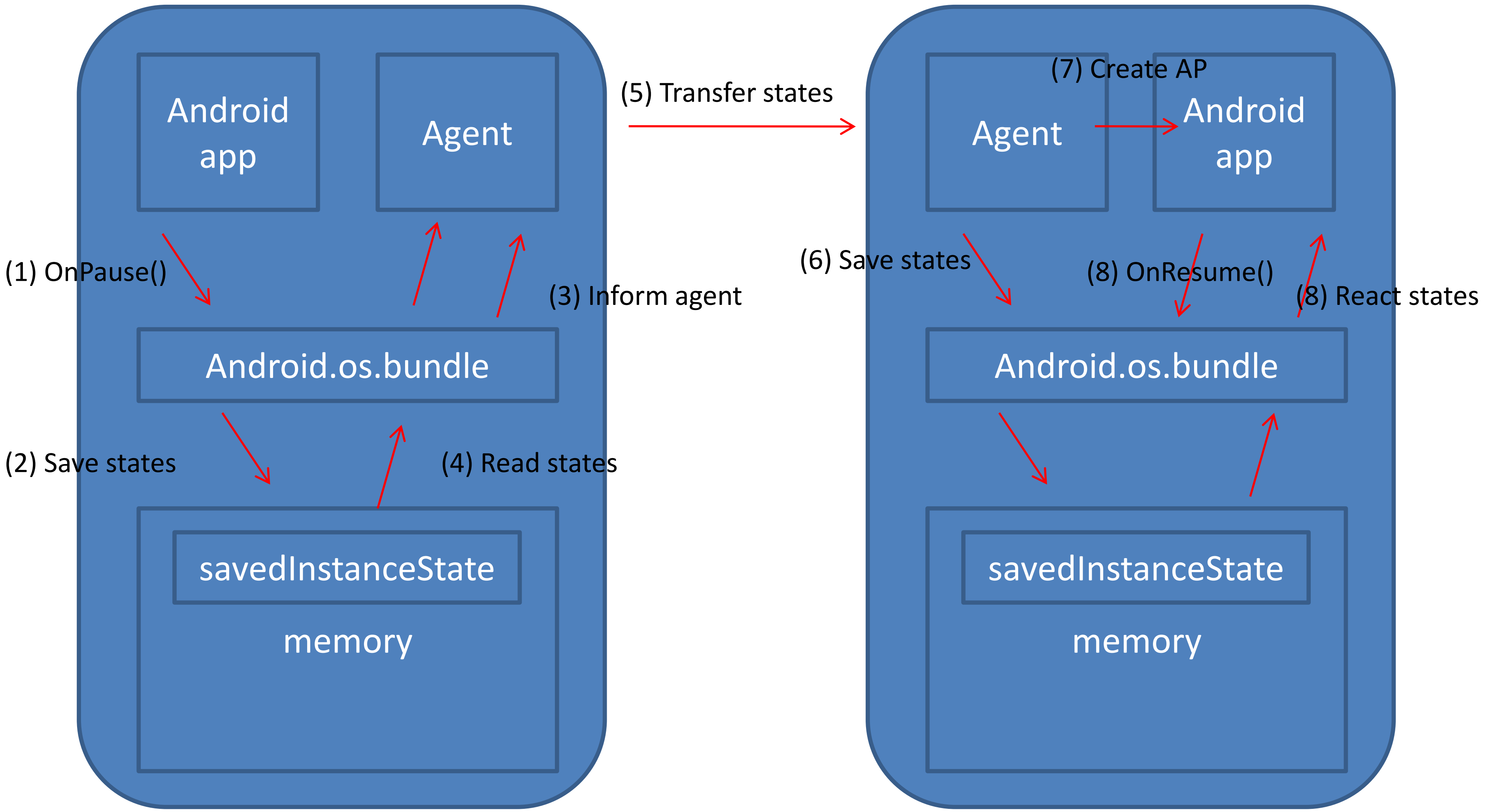
1. Installing the agent program
2. Allocation of a delegate system
3. Setting up a virtual environment
4. Cloning of the operating environment
5. Migration of applications
6. Synchronization of application and user data

Migrating an application

- Traditional virtual machine-based scheme needs to save and transfer the entire state of a virtual machine.

- Transfer the state saved explicitly by the OnPause() method and resume it on another device by Onresume() method

Procedure of migrating an application



Input event and application replay

- Many applications are organized in phases, and it would be wise for such an application to save its state in between the phases when the state is less.

- The work done by the user since the last checkpoint is lost. Record/replay mechanism is used to replay the input event from the checkpoint.

- Two types of event
  - Non-deterministic: location, input data, time stamp(e.g. keyboard input)
  - Deterministic: none (e.g. reading a file)

How replay scheme work

- The application has to explicitly notify the agent about a pseudo checkpoint.
- Pseudo checkpoint is simply a place holder which marks the location of resumption without actually saving the state.
- The state is saved while OnPause function is called and it will resume from the pseudo checkpoint.
- The agent suspends the application → save state → transfer state and recorded input events → remote agent resume application → replay input events → brings the application to the point of migration
- Developer should mark the pseudo checkpoints and identify the global state in the program by inserting function calls to the emphpseudo_checkpoint() function in the framework library.

Interactive applications

- Migrate the application back to the physical device to obtain the input event and then migrate the application to the virtualized environment.
  - Work best when state is small
- Send only the UI window back to the physical device to receive the input from the user via the agents.
  - Significantly reduce network traffic but highly dependent on display protocol and cannot be easily ported to another system.
- Display the UI window via an open remote display protocol such as VNC.

Native code and performance

- Some Android applications are linked to proprietary C or assembly functions for performance reasons.

- For those application, we may execute them via a processor emulator or find a server of the same instruction set to run on.

Synchronizing data

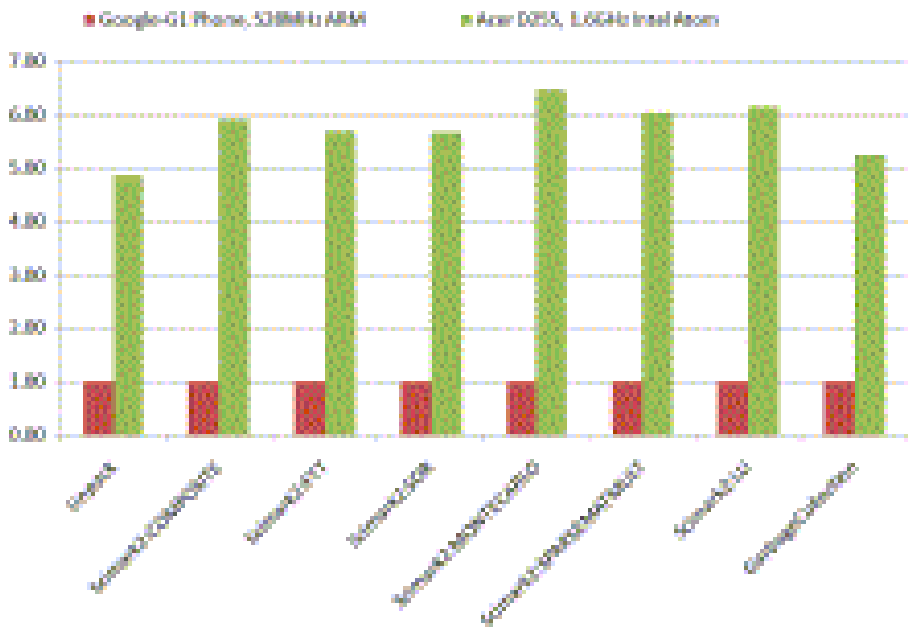
- System image: for initializing a virtual machine
- System-wide data: refers to those files that record system-wide information and/or would affect the operations of the system and applications. E.g. Libraries
- Application data: refers to the file owned by applications.
  - Apply lazy and on-demand policies to synchronize the data upon the request of an application without keeping applications data updated all the time

Security and privacy measures

- Use VPN to establish a private/secure tunnel
- Sensitive data can be encrypted

Performance evaluation

- Server CPU: Intel atom 1.6 GHz
- Mobile CPU: 528 MHz ARM

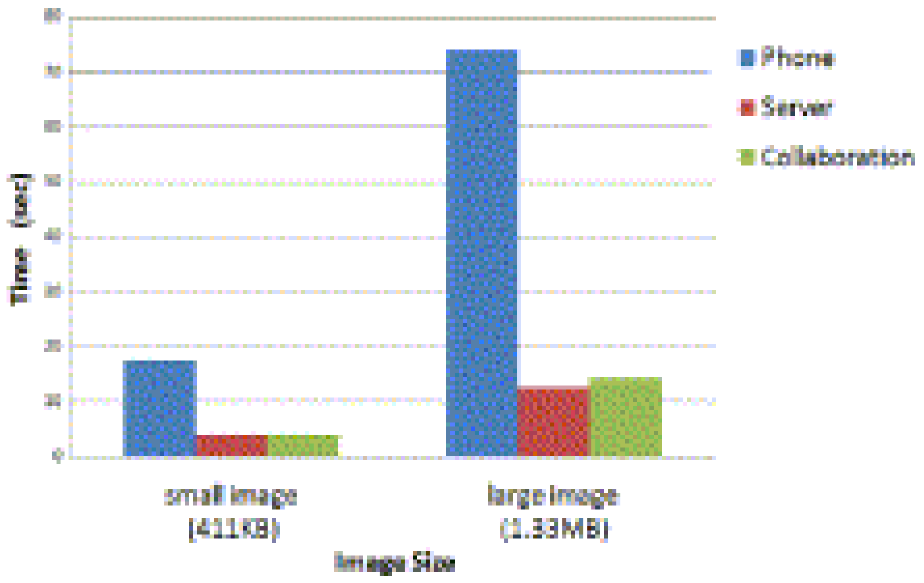


Case1: androidtorrent

- State size: 320 kB
- Take few seconds to send state via 3G network to the cloud
- Migrate and resume in 3.8 seconds (without temporal files)

Case2: face recognition

- Server CPU: i7-2600
- Mobile CPU: HTC desire
- WiFi

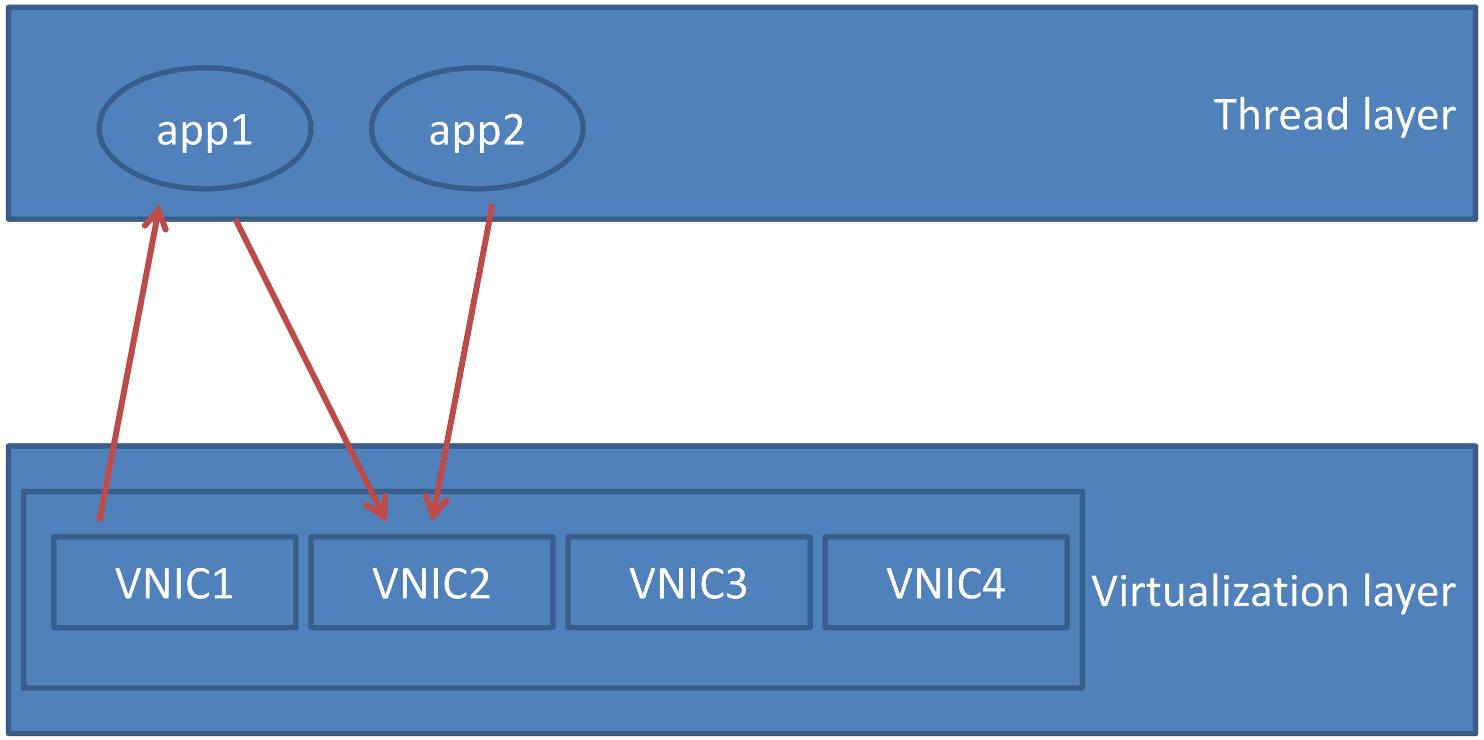


Outline

- Introduction
- Related work
- A virtual environment for Android applications
- Probabilistically guaranteed communication
- Conclusions

Virtual network device architecture

- Integrate different QoS services in a virtualized execution framework



Outline

- Introduction
- Related work
- A virtual environment for Android applications
- Probabilistically guaranteed communication
- Conclusions

Done and to do

- Proposed techniques to address the issues aforementioned
  - Migration, QoS, privacy…
- The framework is still in progress
  - Decide whether/when to offload