Eliminating State Entanglement with Checkpoint-based Virtualization of Mobile OS Services
Kevin Boos & Lin Zhong

What is State Entanglement?
Application-relevant states are stored outside of the application's process memory.

Shared Library Model
- All states in one process

OS Service Model
- States spread across multiple processes

Motivation — Why do we care?
State entanglement prevents the following:
- Fault isolation
- Fault tolerance
- Application migration
- Live update (of both apps and services)
- Whole-application speculation

Solution: OS Service Virtualization
- Virtualize OS Service on a per-app basis
- Encapsulates only one app’s states in each service instance
- Disentangles states

CORSA: Checkpoint-based Virtualization
- Virtualizes OS Services via checkpoint/restore
- Intercepts app-service transactions
- Maintains a per-app checkpoint history
- Only one service instance is active at a time
- All other OS bodies see one service instance
- Satisfies legacy expectations and constraints

Ongoing Implementation
- Kernel-based C/R mechanism
  - Checkpoint: duplicates process structures, uses COW for speed
  - Restore: swaps process control block pointers to previous checkpoint
  - Triggered on Binder IPC transactions

Feasibility Measurement Study
- Checkpoint and Restore can be parallelized
  - Slow checkpoint, fast restore operation
  - Only restore is on the critical path

CORS Android Implementation is Feasible!
- Checkpoint latency \( t_C = 0.3 \text{ ms} \)
- Restore latency \( t_R = 4.4 \mu\text{s} \)
- Min. transaction interval: \( \theta = 1.07 \text{ ms} \)
- Max transaction frequency: \( f = 221 \text{ Hz} \)

An OS Service cannot be instantiated multiple times!
Each service must be a singleton instance to ensure compatibility with the global service directory and other legacy OS components.

www.owlnet.rice.edu/~kevinaboos