Constructing Services with Interposable Virtual Hardware

Andrew Whitaker, Richard S. Cox, Marianne Shaw, Steven D. Gribble
(andrew,rick,mar,gribble)@cs.washington.edu
VM Service Examples

- Migration
  - Collect device state to transport a VM
- Intrusion Detection
  - Interpose on the Ethernet device of a VM

Virtual Machine Services

- Services provided beneath guest OS by modifying virtual devices
- Devices have a relatively simple VM API
- Devices are implemented within VMM, hard to modify

How do I easily build virtual machine services?

The Missing Bus Interface

- Device implementations and inter-device links are not exposed
- There is no virtual bus
Virtual Bus

Physical Machine

CPU

NIC

Disk

Memory

Virtual Machine Monitor

VM 1

vCPU

vNIC

vDisk

vMemory

NetBSD

VM 2

vCPU

vNIC

vDisk

vMemory

NetBSD

Virtual Bus

Physical Machine

CPU

NIC

Disk

Memory

Virtual Machine Monitor

VM 2

vCPU

vNIC

vDisk

vMemory

NetBSD

VM 1

vCPU

vNIC

vDisk

vMemory

NetBSD

Virtual Device Monitors

- Instead of complete-VM interface, provide collection of devices
- Expose communication between devices: create a virtual bus

Mach Microkernel for VMs

- Mach provided a way to expose the communication within an operating system
- VDMs provide a way to expose the communication within a VMM
- Similar issues, different layer of abstraction

µDenali

- Converted the Denali VMM into the µDenali VDM
Apache*

- Continuous Rejuvenation
  - Repair leaks by rebooting
- On a cluster: roving reboots
- Apache*: Virtual cluster

Ease of Extension

- Built several services
  - Migration: 289 lines-of-code
  - Intrusion Detection: ~0 lines-of-code
  - Copy-on-Write disks: 675 lines-of-code
  - Apache*: 1131 lines-of-code
  - Time-travel disks: 877 lines-of-code

Summary

- Will see more VM services in the future
- VDMs answer the question: “How do I easily build VM services?”