Multilayer virtualized systems analysis with Kernel Tracing

Cédric Biancheri
Michel Dagenais

5 May, 2016
École Polytechnique de Montréal
General objectives
Fused Virtual Machine Analysis
Fused Virtual Machine View
Objectives

- Represent multilayer systems (Virtual execution environments)
- Bring out indirect interactions between layers
- Track virtual CPUs and processes
- Find possible sources of preemption
Fused Virtual Machine Analysis

- Similar to the Kernel Analysis
- Traces from Host and Virtual Machines
- Analyze the events of guests as if they were from the host
- Erase the bounds between Virtual Machines and their host
Fused Virtual Machine Analysis

![Diagram of virtual machine and host execution flow with time annotations and color-coded execution states for user space, kernel space, and vCPU preemption.]

Legend:
- Green: User space
- Blue: Kernel space
- VCPU Preemption

Cédric Biancheri 05/05/2016
Fused Virtual Machine Analysis

Automatic recognition of the machine's role

- Synchronization event for host → Host
- Synchronization event for VM → Virtual Machine

kvm_entry → Host

- Associate thread to VM
- Associate thread to vCPU

VM1
- vCPU0
- vCPU1

VM2
- vCPU0

Host
- Thread 2210
- Thread 2211
- Thread 3012
Fused Virtual Machine Analysis

Handling events from all layers
Handling events from all layers

Get event
Fused Virtual Machine Analysis

Handling events from all layers

Get event

Get machine
Fused Virtual Machine Analysis

Handling events from all layers

Get event

Get machine

Machine is Host

Handle event
Handling events from all layers

Get event

Get machine

Machine is Host

Handle event
Handling events from all layers

Get event

Get machine

Handle event

Machine is Host

Machine is Guest

Get vCPU
Handling events from all layers

- **Get event**
- **Get machine**
  - Machine is Host: **Handle event**
  - Machine is Guest:
    - **Get vCPU**
    - **Get Thread**

Handling events from all layers

- Get event
- Get machine
- Handle event

Machine is Host

- Get vCPU
- Get Thread
- Get pCPU

Machine is Guest
Fused Virtual Machine Analysis

Handling events from all layers

Get event

Get machine

Machine is Host

Handle event

Machine is Guest

Get vCPU

Get Thread

Get pCPU

Replace vCPU by pCPU
Fused Virtual Machine Analysis

Handling events from all layers

Get event

Get machine

Machine is Host

Handle event

Machine is Guest

Get vCPU

Get Thread

Get pCPU

Replace vCPU by pCPU
Fused Virtual Machine Analysis

Handling events from all layers

Get event

Get machine

Handle event

Machine is Host

Machine is Guest

Get vCPU

Get Thread

Get pCPU

Replace vCPU by pCPU
The event is a VM Entry

Get thread → Get pCPU & vCPU → Save pCPU's values → Restore vCPU's values
Fused Virtual Machine Analysis

The event is a VM Entry

Get thread  →  Get pCPU & vCPU  →  Save pCPU's values  →  Restore vCPU's values

The event is a VM Exit

Get thread  →  Get pCPU & vCPU  →  Save vCPU's values  →  Restore pCPU's values
**Fused Virtual Machine View**

**Demo:**

- **Host:** 8 pCPUs
  - One task switching between pCPUs 0 and 1

- **VM1:** 2 vCPUs on pCPUs 0 and 1
  - One container
  - One task in the VM
  - One task in the container

- **VM2:** 1 vCPU on pCPU 1
  - One task
Fused Virtual Machine View

Demo:

Host

- Task
- pCPU0
- pCPU1
- pCPU2
- ... pCPU7

VM1
- vCPU0
- vCPU1
- Container
- Task
- Task

VM2
- vCPU0
- Task
Conclusion

- Multilevel traces aggregated in one level
- Highlight a virtual machine or a container
- Track a vCPU on the host
- Track any thread on the host
- Observe the cause of a preemption
Questions?

cdc.biancheri@gmail.com
https://github.com/cbiancheri/
In case of Murphy's law
In case of Murphy's law
In case of Murphy's law
In case of Murphy's law
In case of Murphy's law
In case of Murphy's law
In case of Murphy's law
In case of Murphy's law
In case of Murphy's law
In case of Murphy's law
In case of Murphy's law
In case of Murphy's law
In case of Murphy's law