VM Analysis – Episode 4

Wait analysis of virtualized environments using host kernel tracing

Hani Nemati

May 5, 2017

Polytechnique Montréal
Laboratoire DORSAL
Agenda

Introduction

- Research update and research motivation

New Investigations

- Wait analysis of virtualized environments using host kernel tracing
  - State of the art
  - Proposed Algorithm
  - Demo
  - KVM-Tool for eBPF

Conclusion and in-progress
Previously on “VM Analysis”

Available Trace-Points in different layers

- Guest Kernel Trace
- Qemu Trace
- KVM Trace
- Host Kernel Trace
- Hardware PMC
Previously on “VM Analysis”

Available Trace-Points in different layers
Previously on “VM Analysis”

Resource View for VM without tracing the VM
Previously on “VM Analysis”

VirtFlow: Execution Flow Analysis of Virtual Machine
Previously on “VM Analysis”

Two Nested VMs and One VM are preemption each other
Motivation

Why the VM is waiting?
Motivation

Let's use the Critical Flow view of Trace Compass?
### Investigations

#### Methodology

If \( \text{Vec} == \text{(Block I/O irq)} \) {
    Block State = Block I/O State
} else if \( \text{Vec} == \text{(network irq)} \) {
    Block State = Network State
}
If (Vec == 239) {
    Block State = Timer
} else if (Vec == 251) {
    Block State = Task
}
Demo
Investigations

What do you need to test this project?

- Access to **Host** only
- Run **LTTng** on Host with my new added tracepoint (vcpu_enter_guest)
- Clone **TraceCompass** from my github (virtFlow)
  - [https://github.com/Nemati](https://github.com/Nemati)
- Open Resource View of TraceCompass
One More Thing ...
KVM-Tools
For
eBPF
Conclusion and in-progress

Inferences

- Wait Analyzing of process inside VM
- A process is waiting for
  - A Block request to finish
  - A network packet to receive
  - Another process
  - A timer to fire

What you will see in Episode 5

- Wait Analyzing of process inside Nested VM
Questions?

Hani.nemati@polymtl.ca
https://github.com/Nemati