Multilayer traces analysis with Kernel Tracing

Cédric Biancheri
Michel Dagenais

10 December, 2015
École Polytechnique de Montréal
Content

- General objectives
- Previous approach
  - Virtual Machine Cpu Analysis
  - Virtual Machine View
- Fused Virtual Machine Analysis
- Fused Virtual Machine View
Objectives

- Represent multilayer traces (Virtual machines, Containers, JVM…)
- Bring out indirect interactions between machines
- Track a virtual CPU and processes
- Find possible sources of preemption
Virtual Machine Analysis

Based on Kernel tracing

Required events:
- sched_switch
- sched_process_fork
- sched_process_exit
- kvm_entry
- kvm_exit

State History Tree:

|--Virtual Machines
  |--Ubuntu
    |--CPUs
      |--CPU 0
      |       |--Current Status

M. Gebai: “Fine grained preemption analysis across virtual machines”
Virtual Machine Analysis

Virtual Cpu states transitions:

1. `kvm_entry`
2. `kvm_exit`
3. `sched_switch`
4. `sched_switch [cpu is idle]`
5. `sched_switch`
6. `sched_switch`
Virtual Machine Cpu View

Resources View:
Virtual Machine Cpu View

Resources View:

Virtual Machine View:
Virtual Machine Cpu View

Resources View:

Virtual Machine View:

Cédric Biancheri 10/12/2015
Fused Virtual Machine Analysis

- Similar to the Kernel Analysis
- Centered on the host
- Requires traces from the host and guests

Objective:
Analyse the events of guests as if they were from the host
And remember where they are really from
Fused Virtual Machine Analysis

- IDLE
  - (3) \texttt{sched\_switch}
  - (4) \texttt{sched\_switch} [cpu is idle]
- PREEMPTED
  - (5) \texttt{sched\_switch}
  - (6) \texttt{sched\_switch}
- VMM
- RUNNING
  - (1) \texttt{kvm\_entry}
  - (2) \texttt{kvm\_exit}

Events from guest
Fused Virtual Machine Analysis

Kernel Analysis:
|--Host
 | |--CPUs
 | | |--CPU 0
 | | | |--Status
 | | | |--CurrentThread
 | |--Threads
 | | |--1461

Fused Virtual Machine Analysis

Kernel Analysis:
|--Host
| |--CPUs
| | |--CPU 0
| | |--Status
| | |--CurrentThread
| |--Threads
| | |--1461

Fused Virtual Machine Analysis:
|--Host
| |--CPUs
| | |--CPU 0
| | |--Condition
| | |--Status
| | |--CurrentThread
| | |--MachineName
| | |--vCPU
| |--Threads
| | |--Host
| | | |--1523
| | |--VM1
| | | |--2401
Fused Virtual Machine View

Demo:

• Host with 8 pCPUs
• One task switching between pCPUs 0 & 1

• Guest with 2 vCPUs on pCPUs 0 & 1
• One task on each vCPU
Conclusion

- Multilevel traces aggregated in one level
- Highlight a virtual machine
- Track a vCPU on the host
- Observe directly the cause of a preemption

email: cdc.biancheri@gmail.com       Github: 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

Select machine

Machines

- 192.168.0.2/host
- 192.168.0.2/server1

Display:

- 0
- 1

OK
In case of Murphy's Law
In case of Murphy's Law
In case of Murphy's Law