Critical Path Analysis through Hierarchical Distributed Virtualized Environments using Host Kernel Tracing

Hani Nemati

May 10, 2018

Polytechnique Montréal

Laboratoire DORSAL

Introduction

Tracing

is

?

The figure retrieved from: https://mad-owl.deviantart.com/art/Chibi-Luke-found-something-193450025

Tracing

is

AWESOME

The figure retrieved from: https://mad-owl.deviantart.com/art/Chibi-Luke-found-something-193450025

Agenda

Introduction

- Research update and research motivation

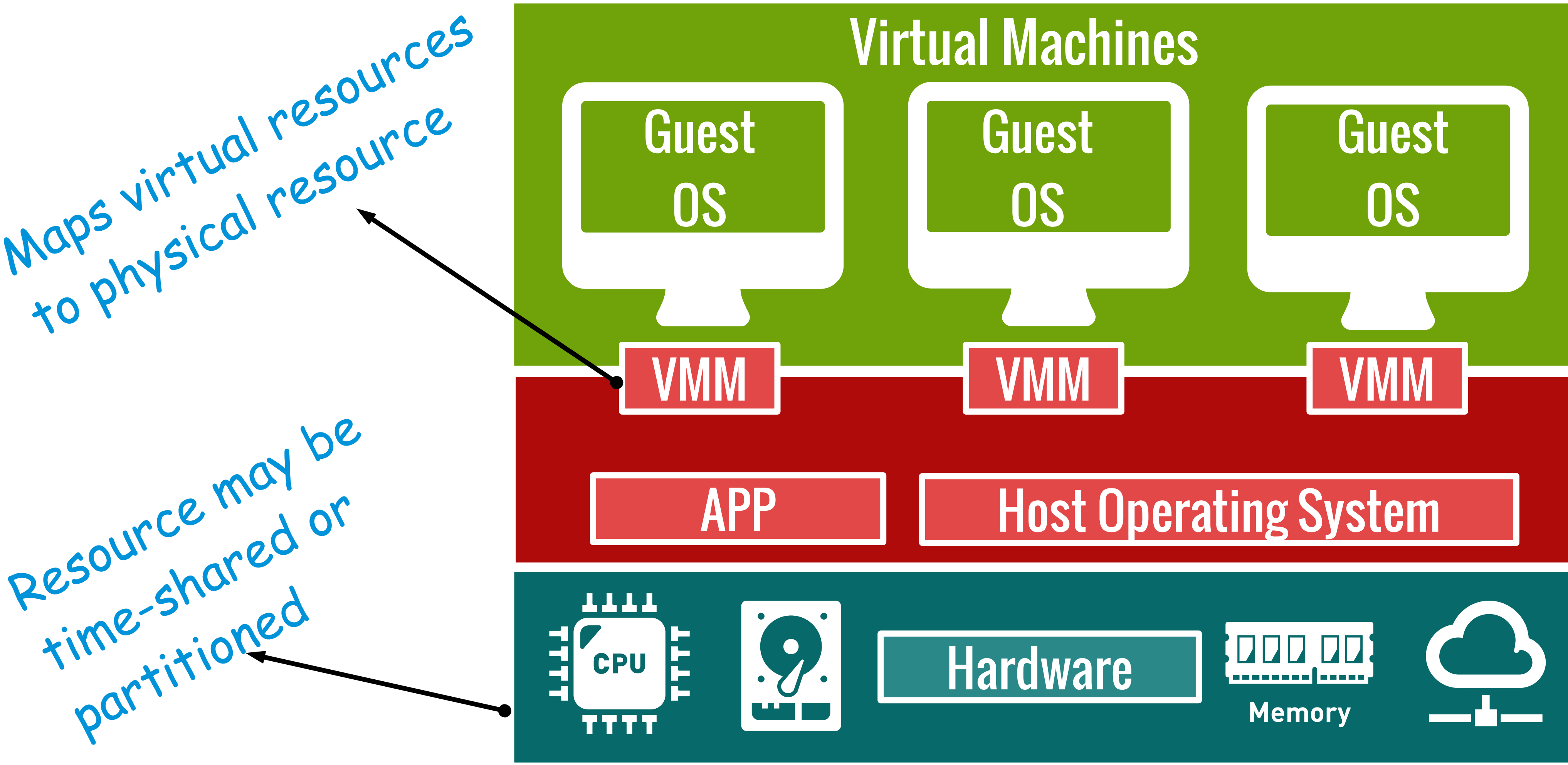
New Investigations

- Host-based Execution-graph Construction (HEC)
- Critical Path Analysis through hierarchical virtualized environments
  - Proposed Algorithm
  - Usecases
  - Demo
  - Overhead Analysis of HEC and existing critical path analysis

Conclusion and in-progress

Motivation

Virtual Machine Hierarchy



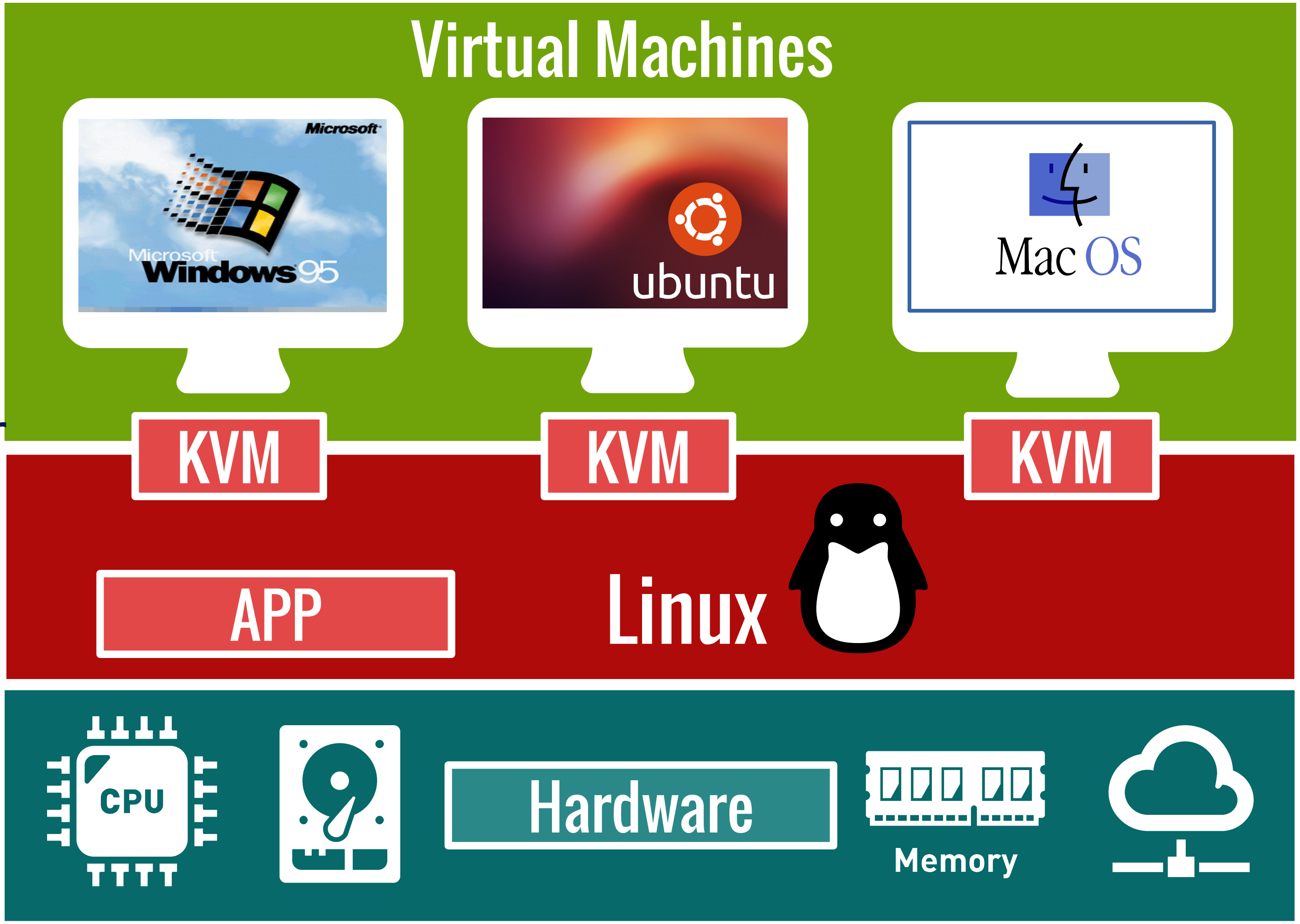
Motivation

Virtual Machine Hierarchy

Arbitrary Guest OS

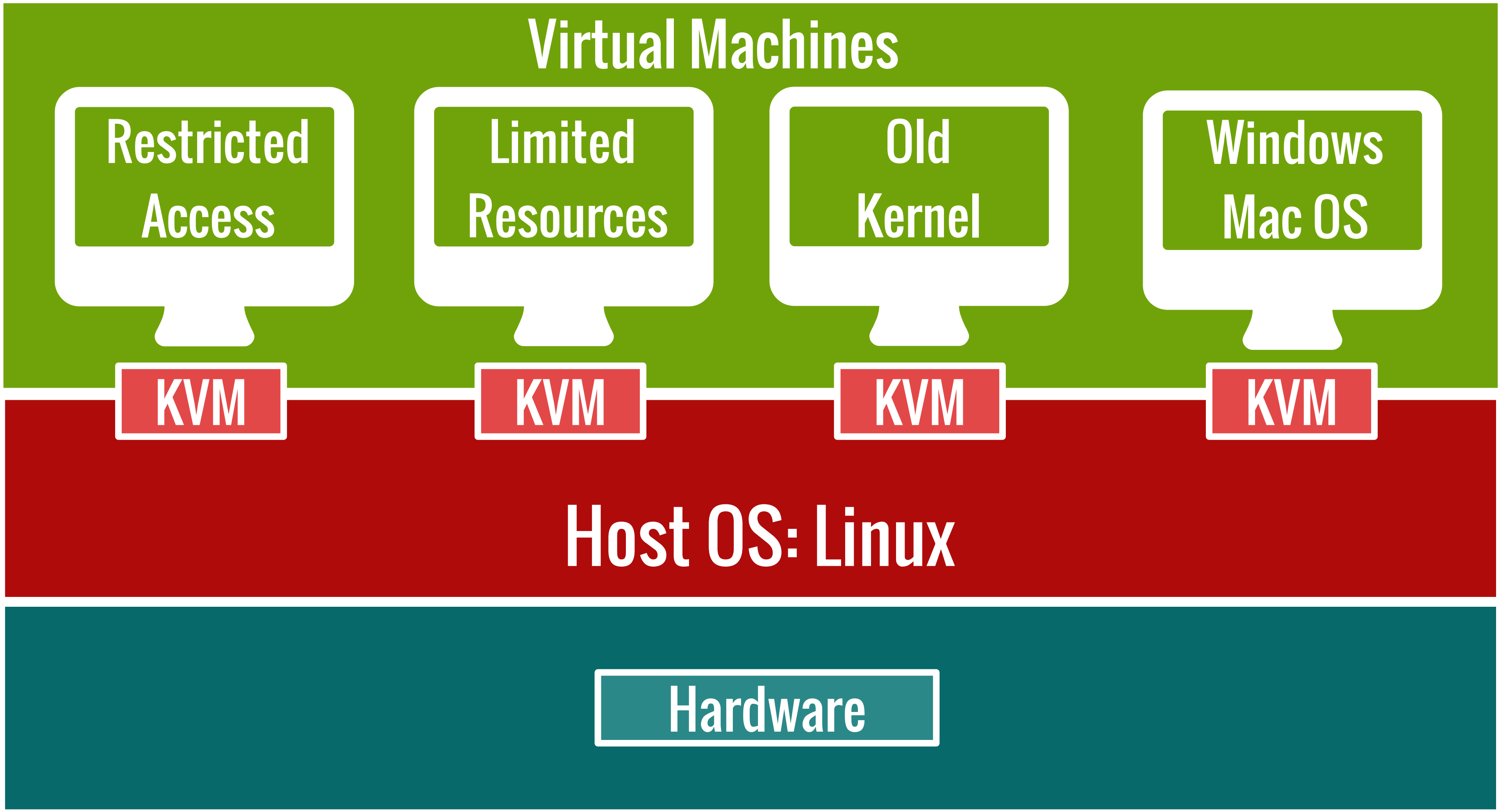
KVM as Hypervisor

Linux as Host OS

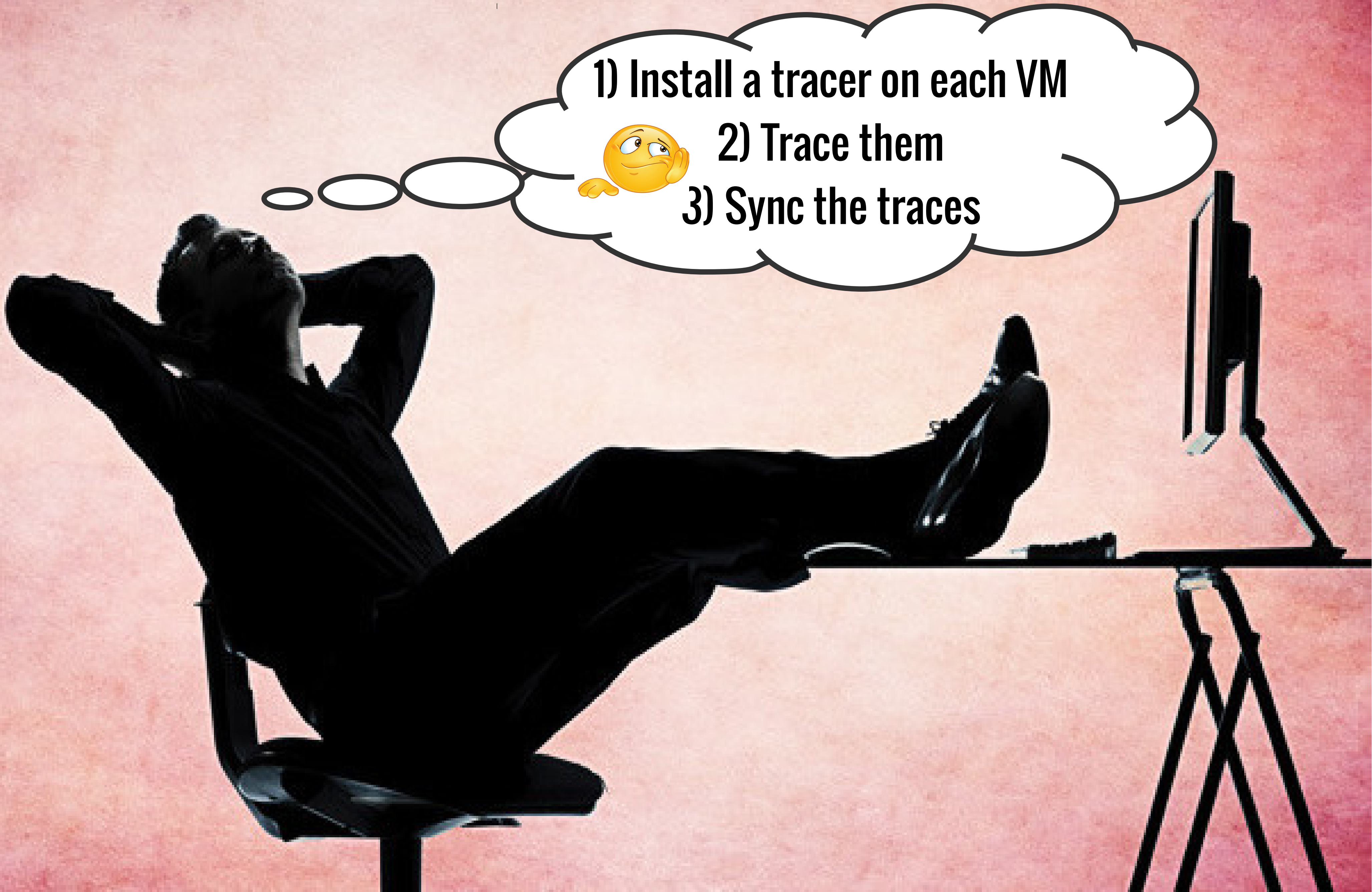


Motivation

Virtual Machine Hierarchy

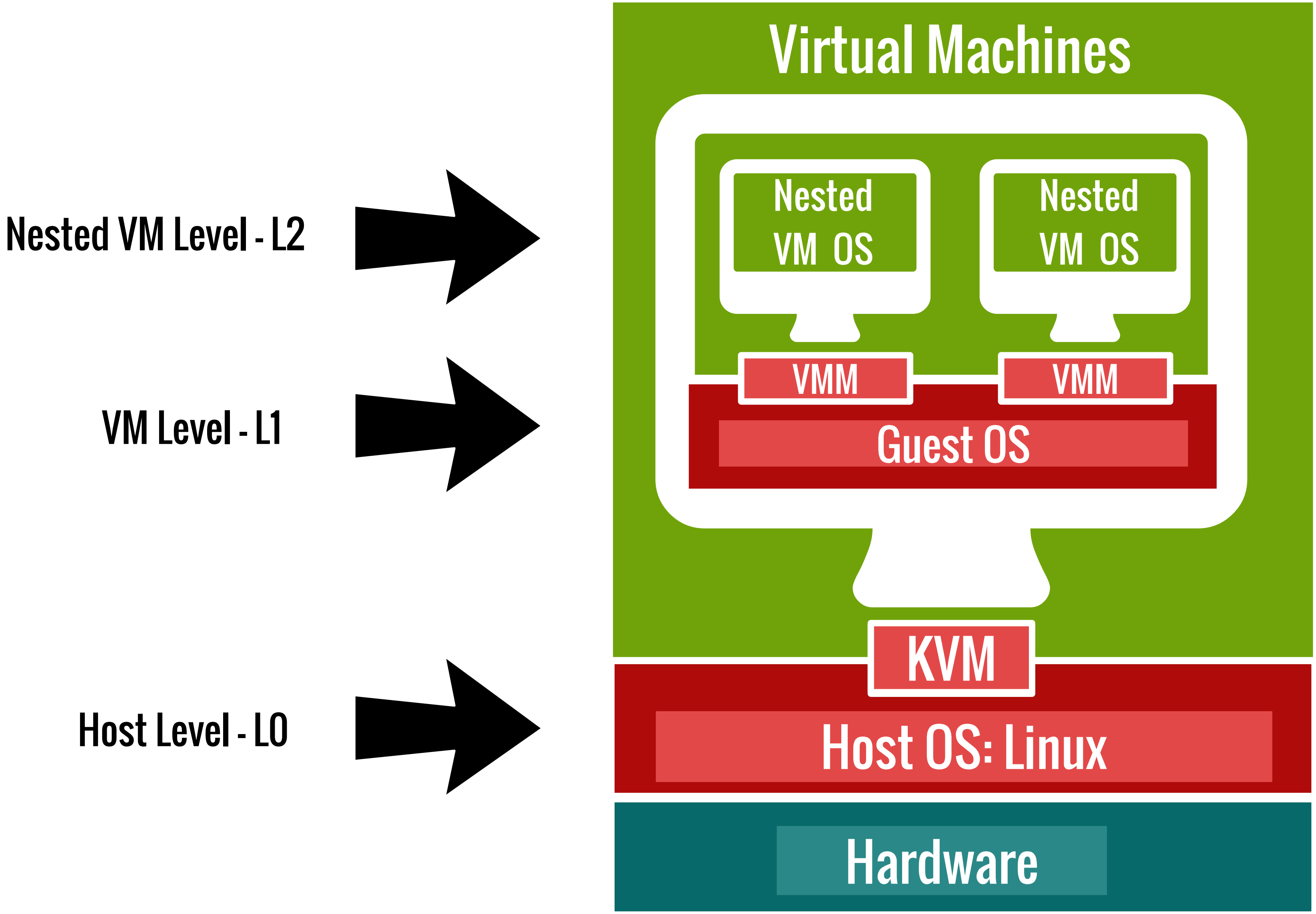


Motivation

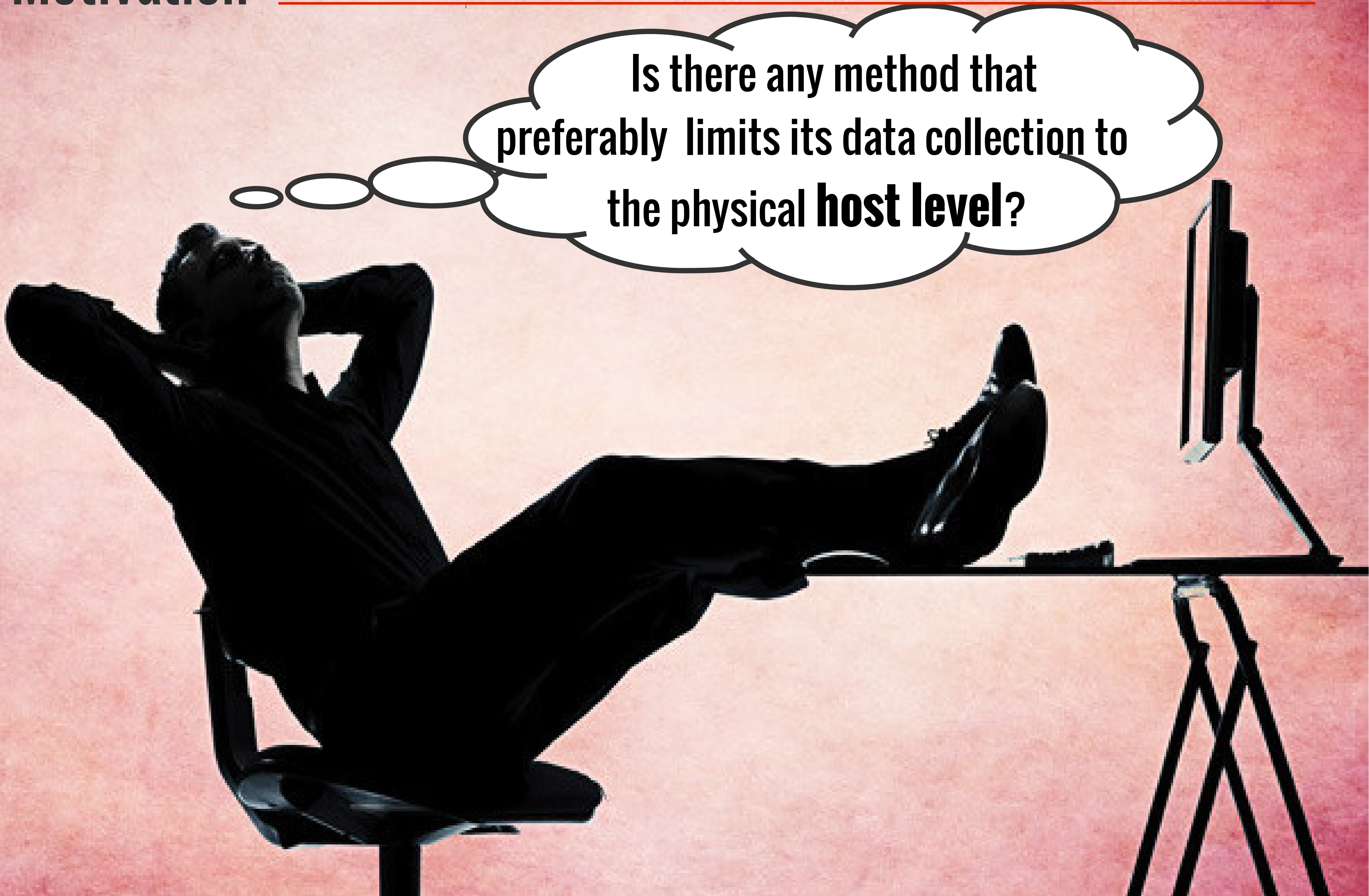


Investigation

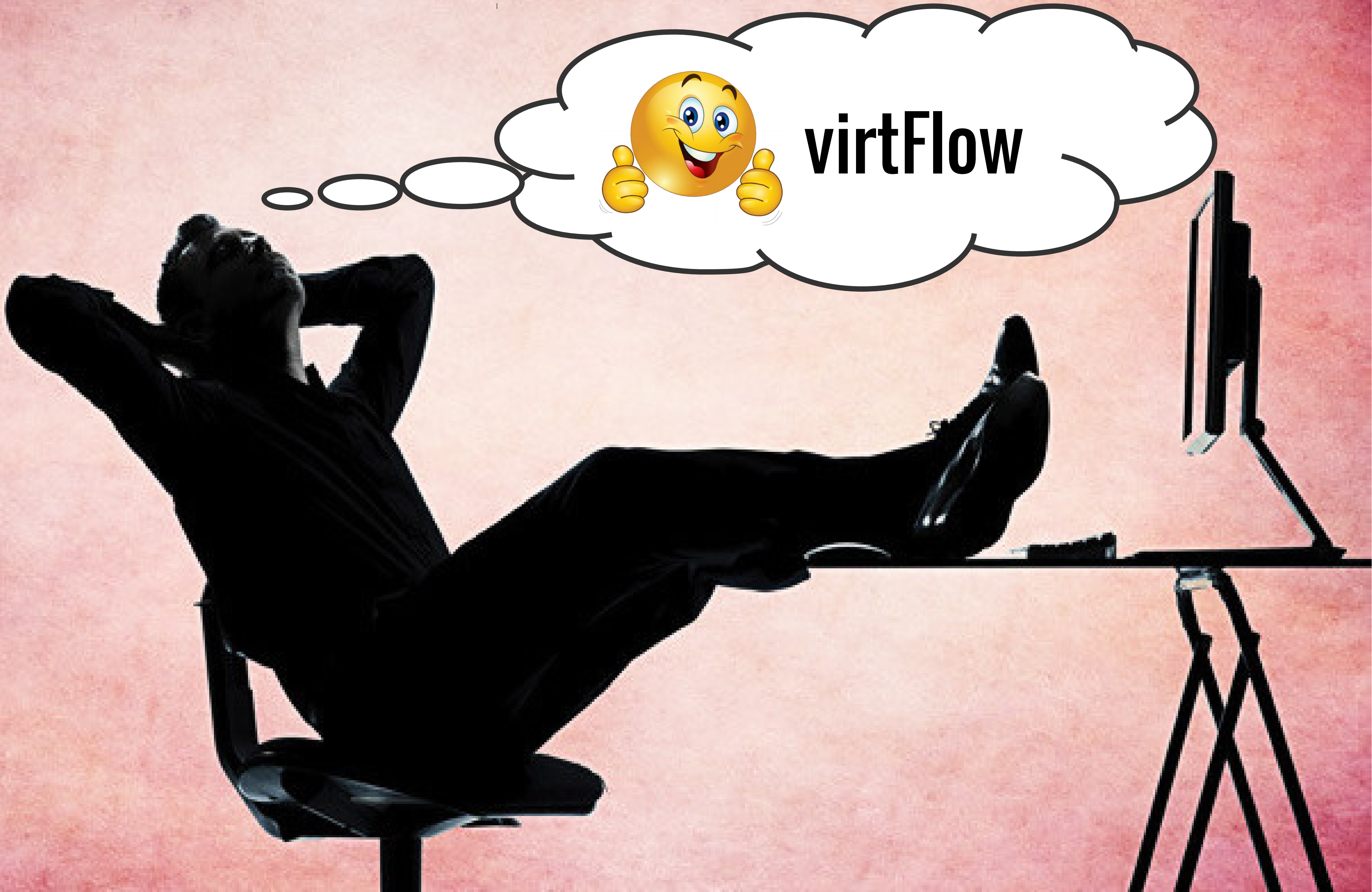
Hierarchical Virtualized Environments



Motivation


Is there any method that preferably limits its data collection to the physical **host level**?

Motivation

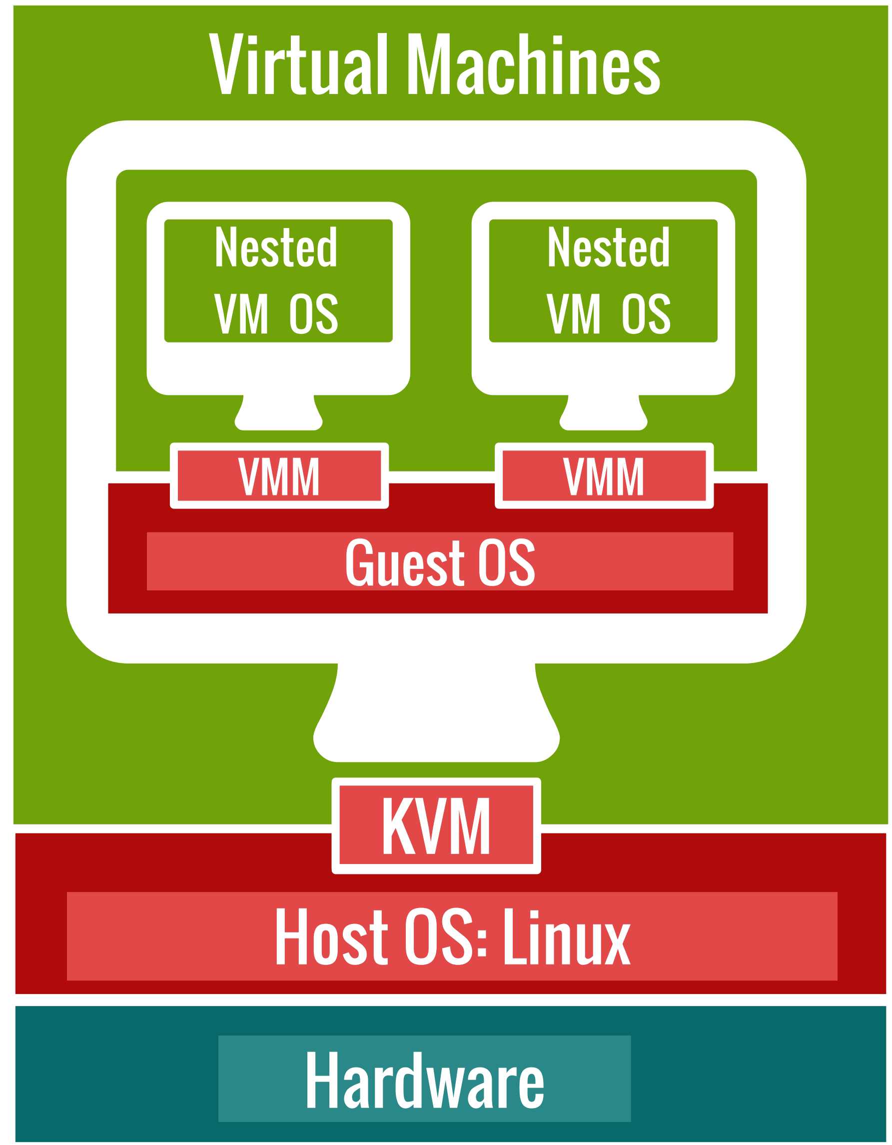


Investigation

virtFlow features

Hierarchal vCPU view for VM

- Running States
- Wait States

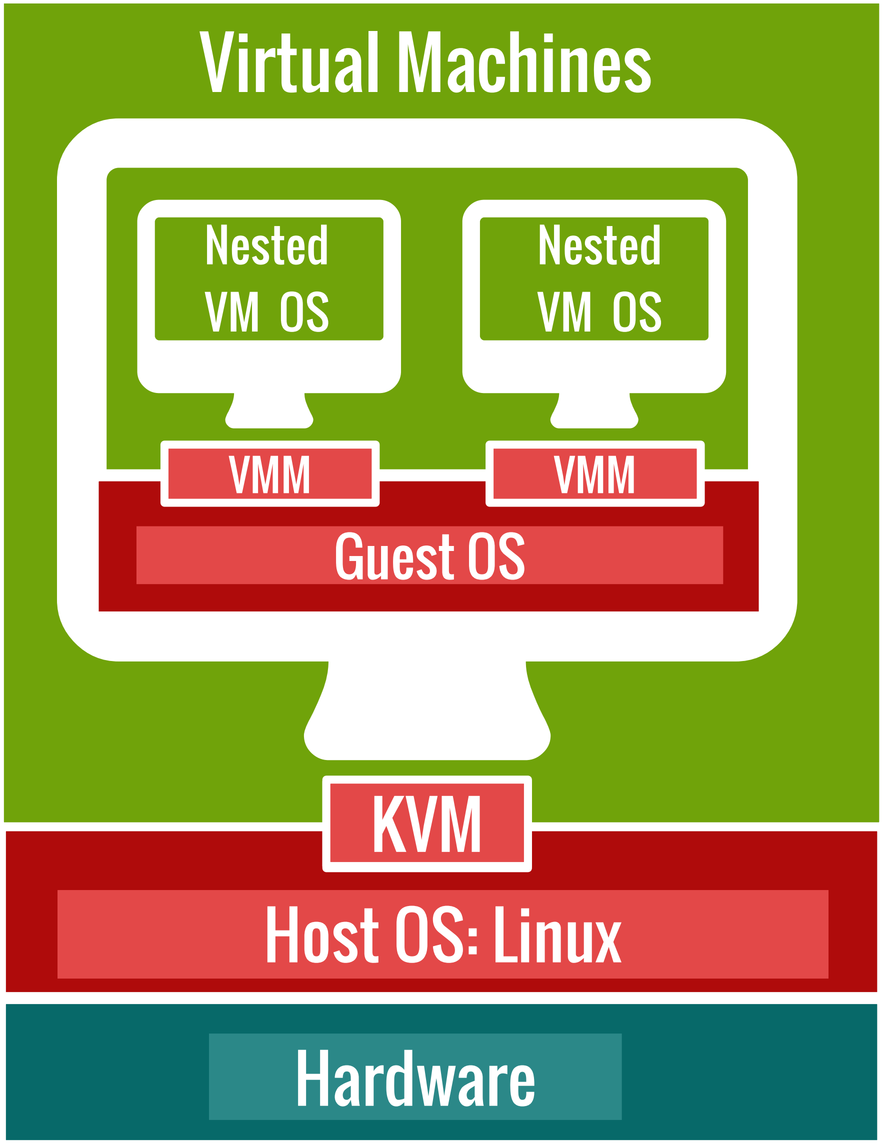


Investigation

virtFlow features

Hierarchal Process view for VM

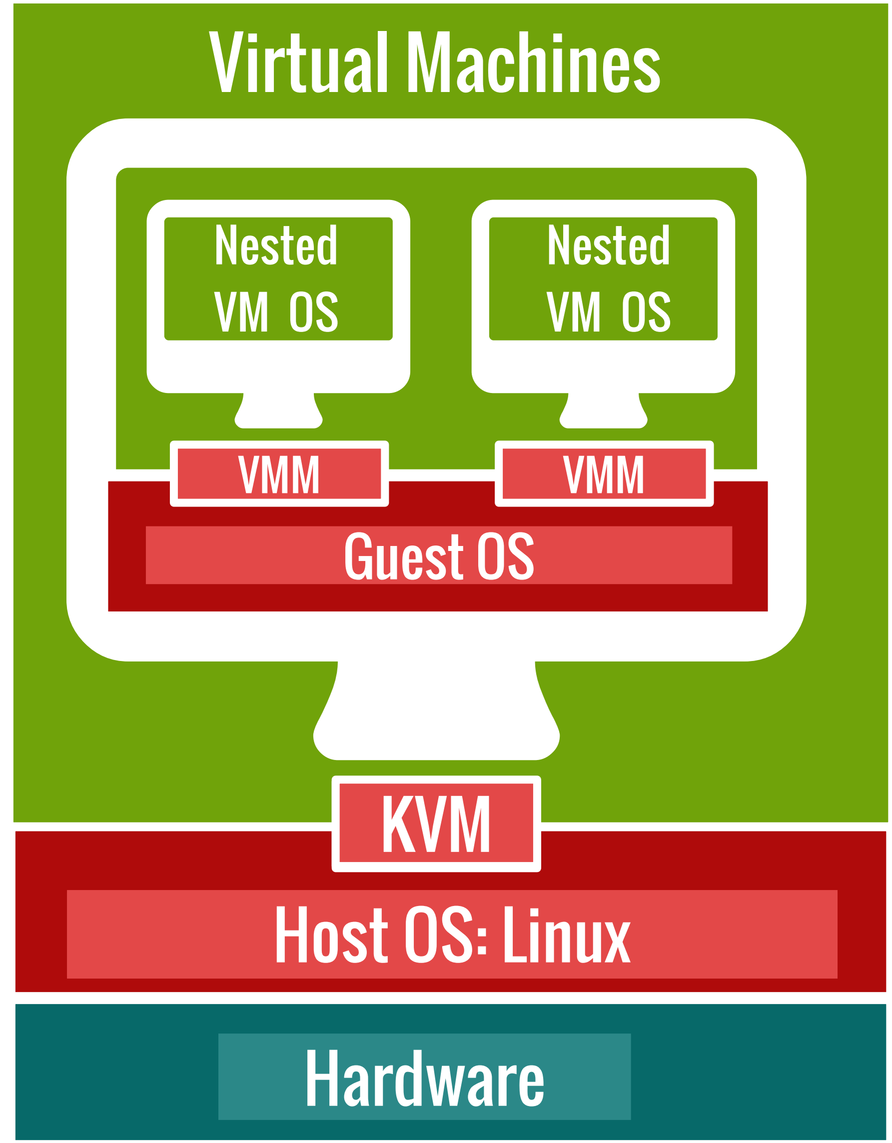
- Running States
- Wait States



Investigation

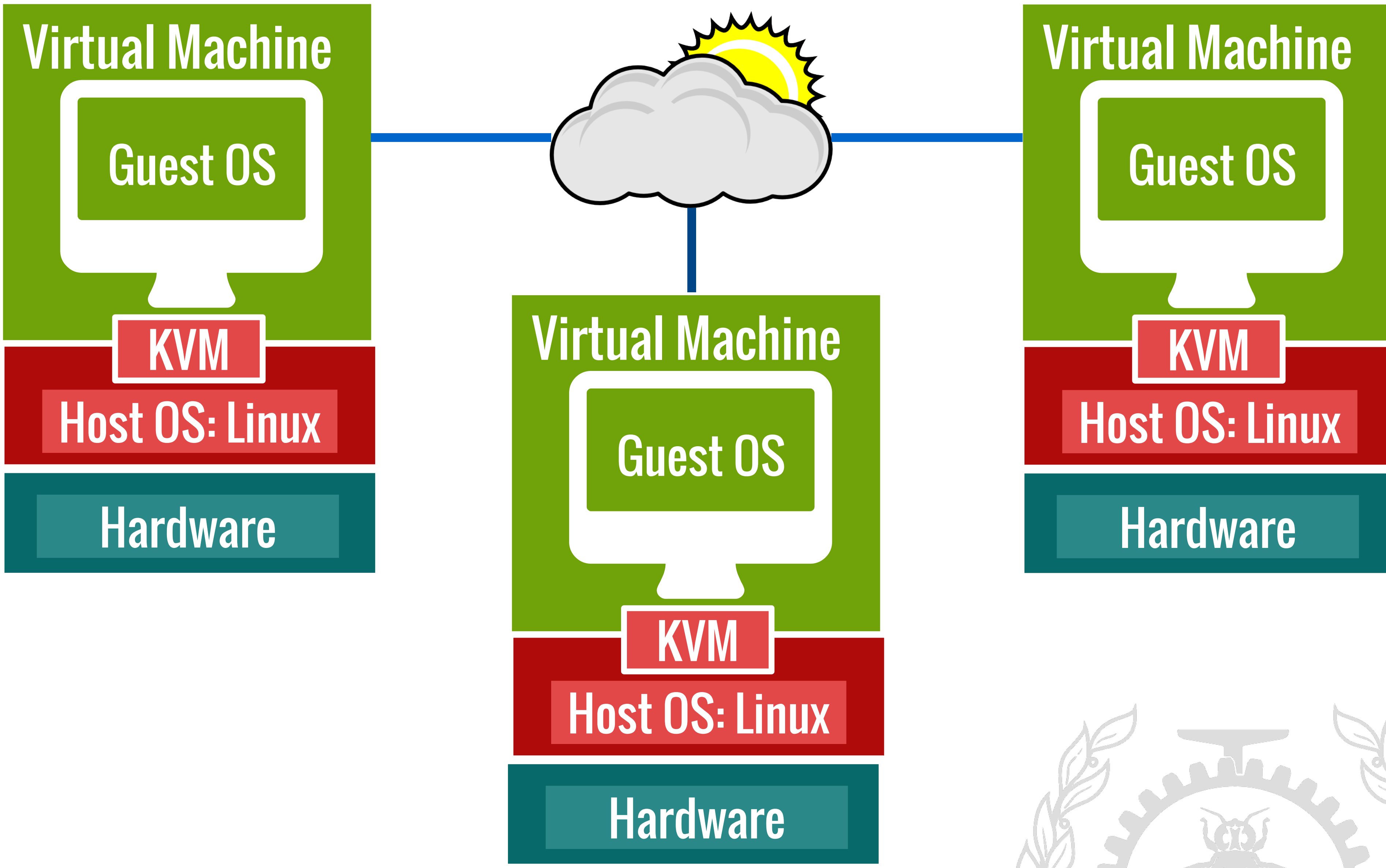
virtFlow features

- Critical Path Analysis through
- Hierarchical Virtualized Environments



Investigation

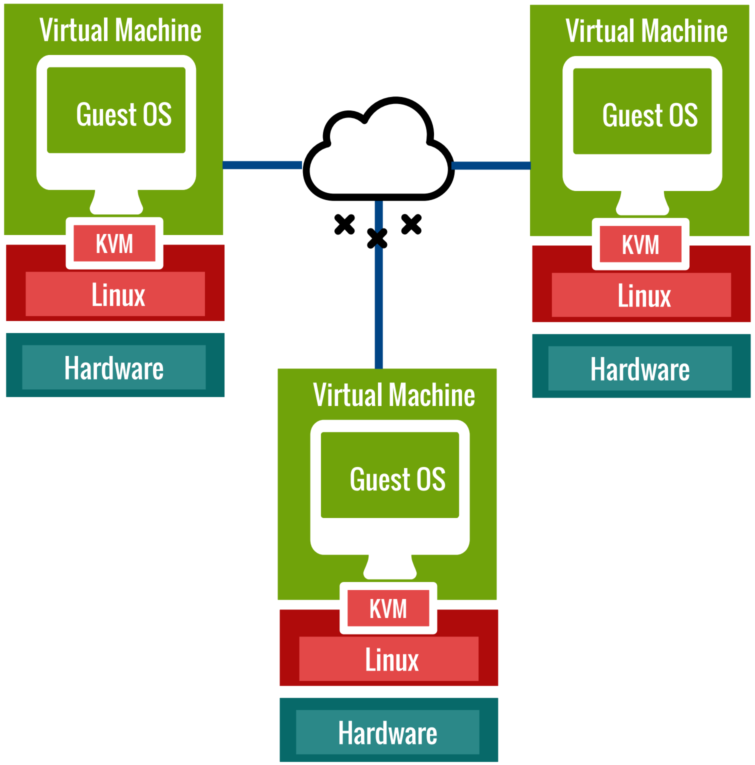
Distributed Virtualized Environments



Motivation

virtFlow features

Critical Path Analysis through Distributed Virtualized Environments



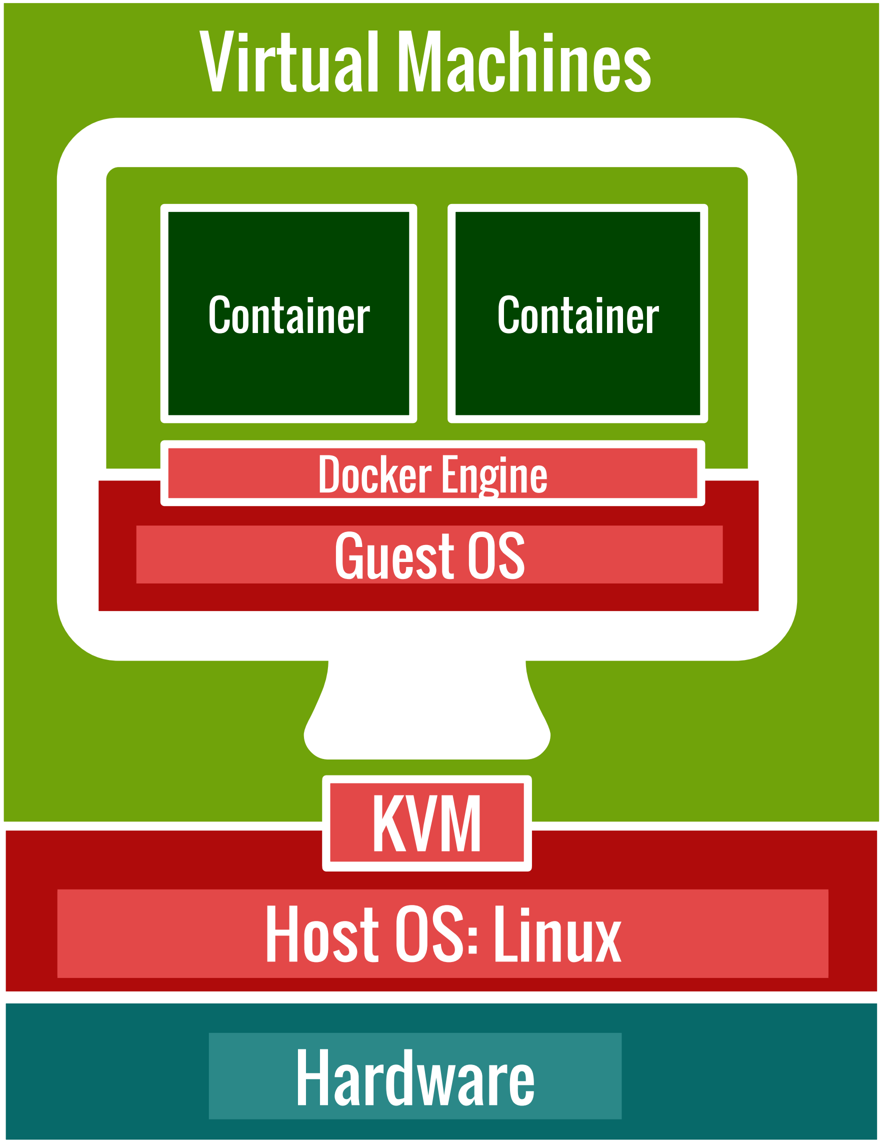
Motivation

virtFlow features

Critical Path Analysis through Distributed Virtualized Environments

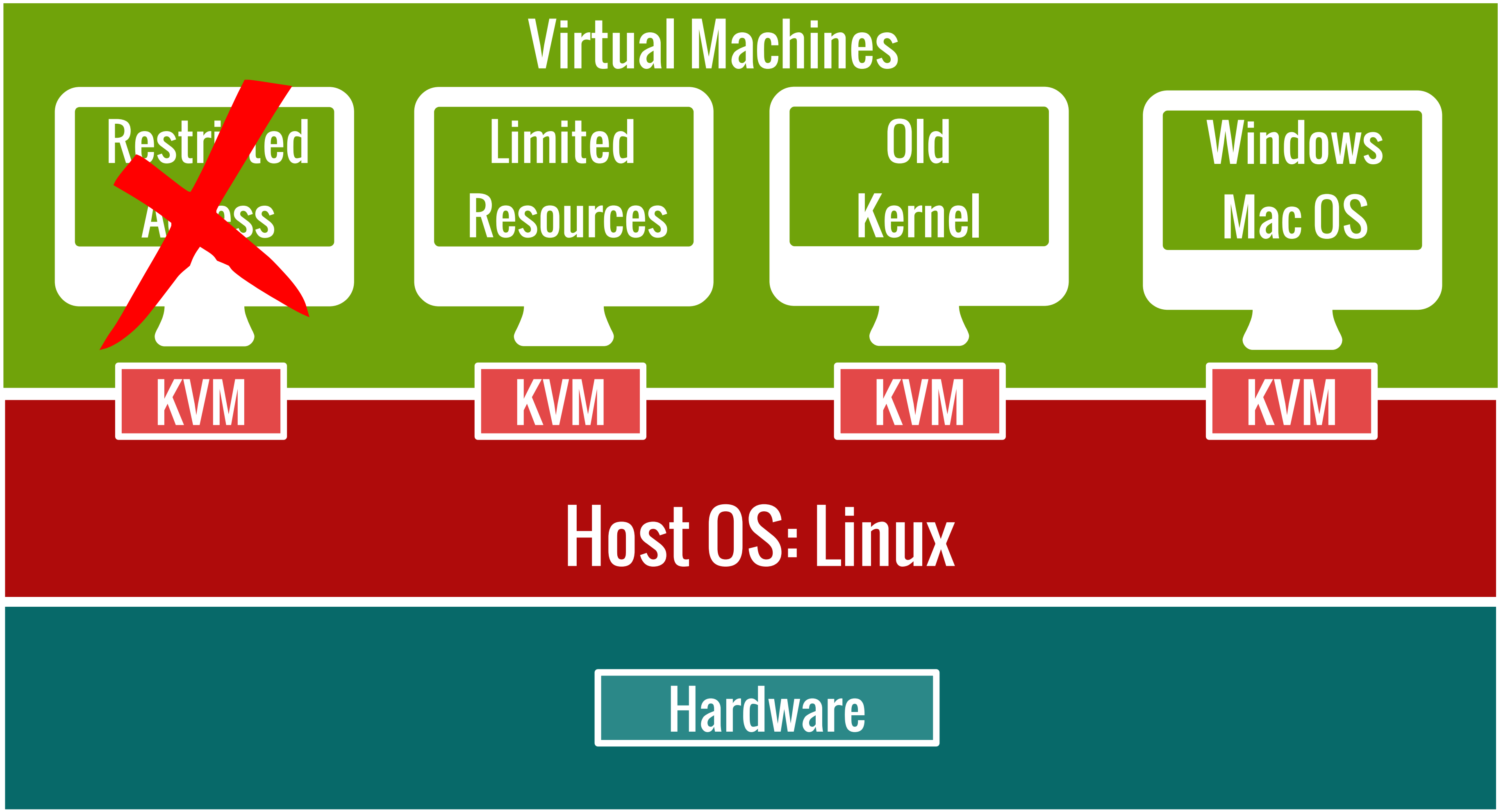
Investigation

Containers within Virtualized Environments



Investigation

Containers within Virtualized Environments



Investigation

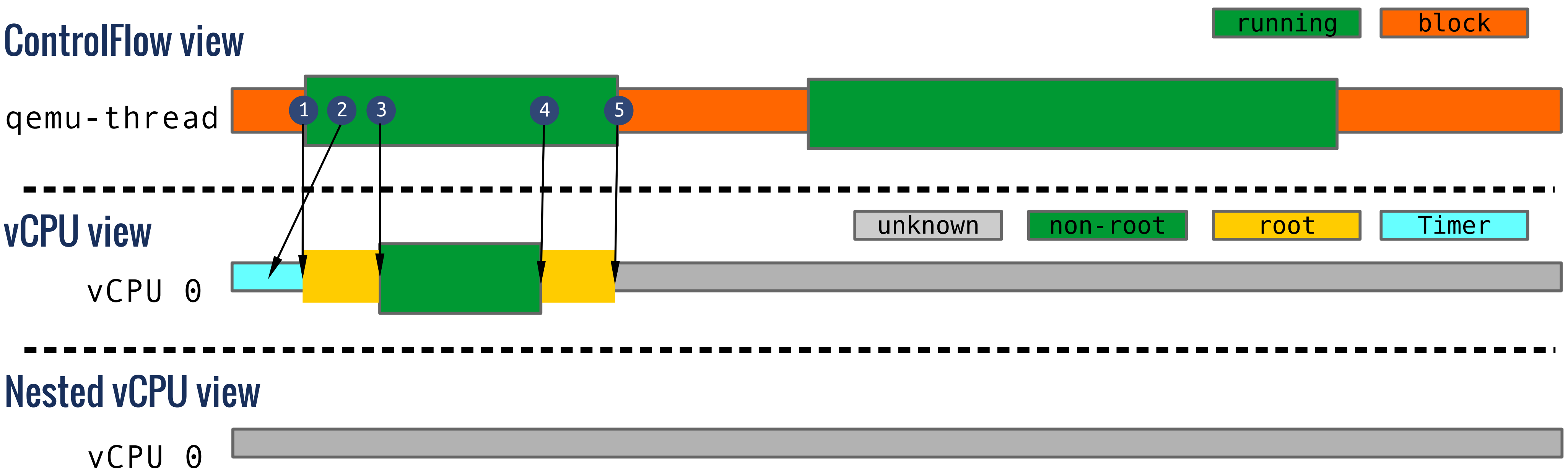
Containers within Virtualized Environments

Investigation

VM Analysis through Hierarchical Virtualized Environments

Methodology

Nested vCPU view



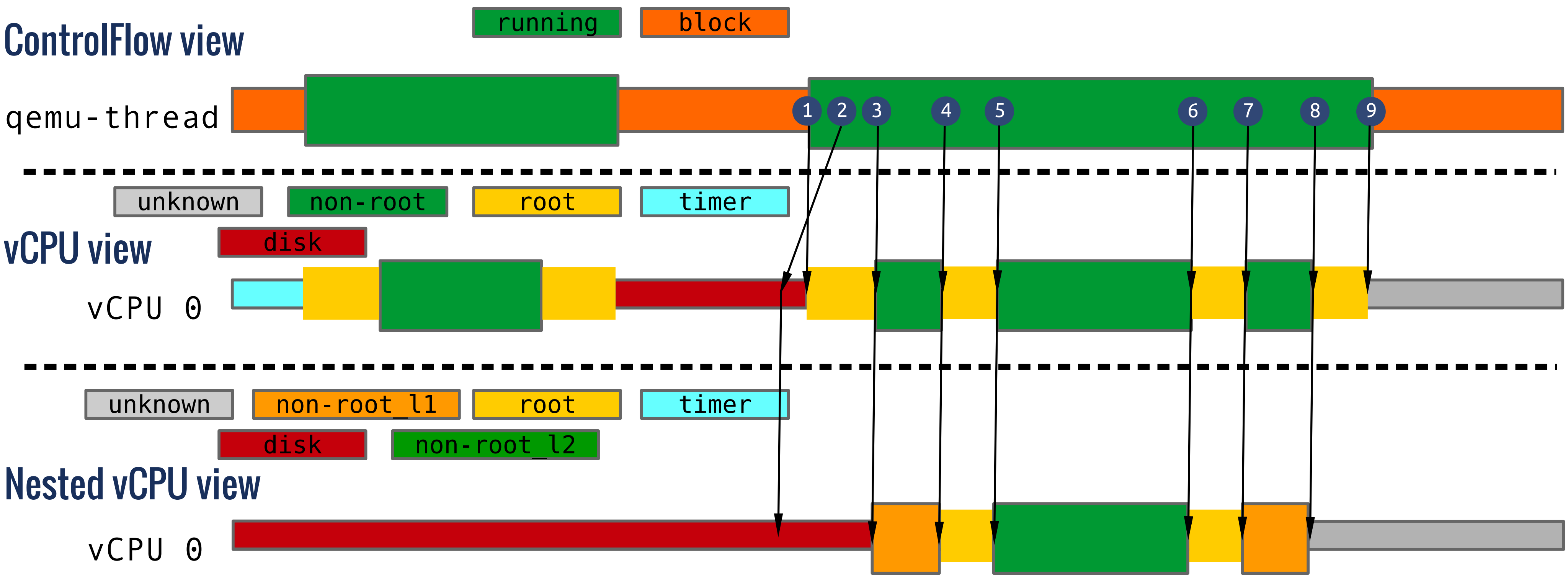
1. `sched_switch(in=qemu_thread)`
2. `inj_virq(vec=timer)`
3. `vm_entry(vcpu0, cr3#0)`
4. `vm_exit(reason=12)`
5. `sched_switch(out=qemu_thread)`

Investigation

VM Analysis through Hierarchical Virtualized Environments

Methodology

Nested vCPU view



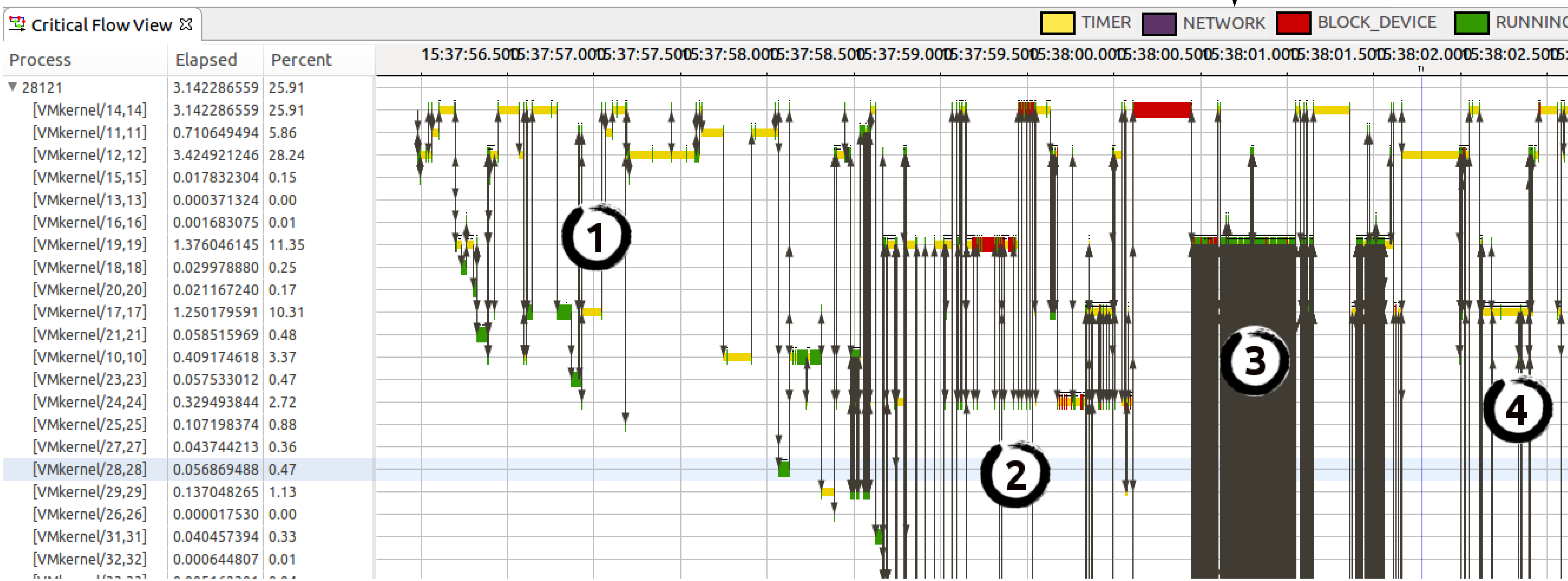
1. sched_switch(in=qemu_thread)
2. inj_virq(vec=disk)
3. vm_entry(vcpu0, cr3#1)
4. vm_exit(reason=24)
5. vm_entry(vcpu0, cr3#2)
6. vm_exit(reason=12)
7. vm_entry(vcpu0, cr3#1)
8. vm_exit(reason=12)
9. sched_switch(out=qemu_thread)

Investigations

Critical Path Analysis

Linux Advance Packaging Tool

What is going on here ?

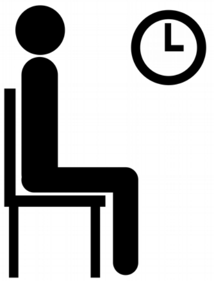


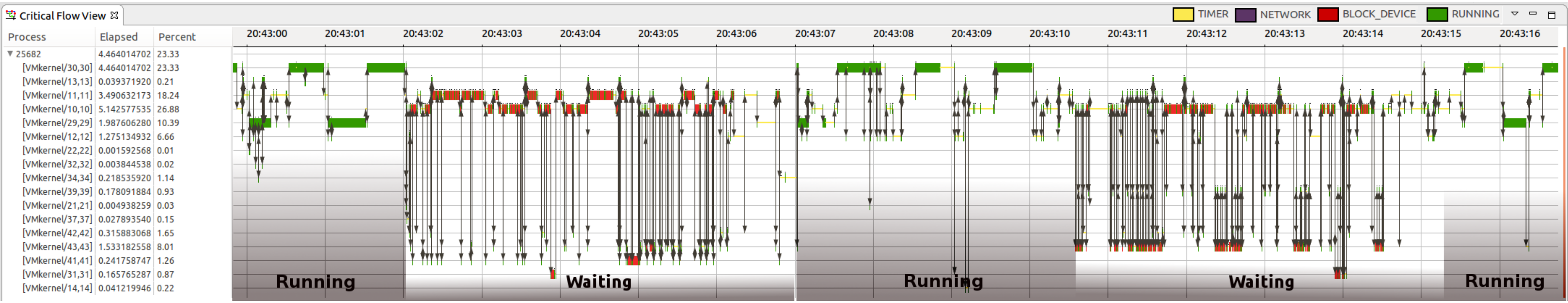

```
1) apt-get downloads and reads cached packages
2) apt-get installs the packages along with downloaded dependencies
3) The installation of man-pages
```

Investigations

Critical Path Analysis

Undesirable parallelism





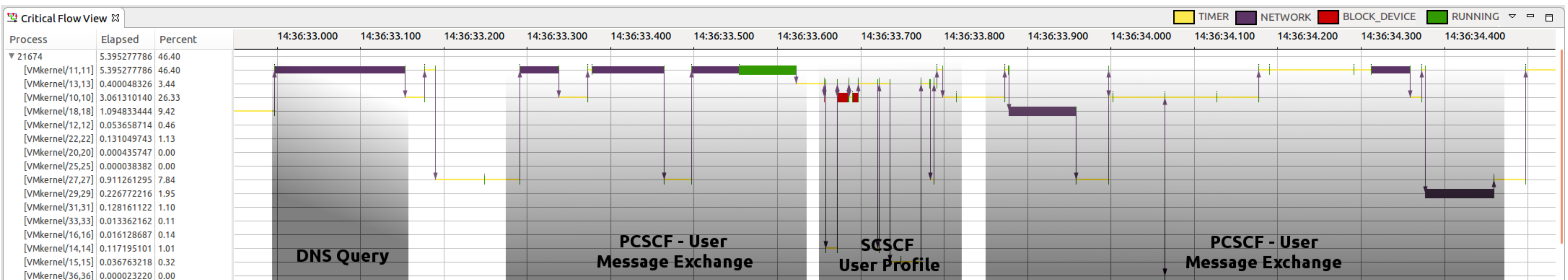
waits for disk

waits for another process

Investigations

Critical Path Analysis

Network Intensive VM - IMS Network



Waits for user to response

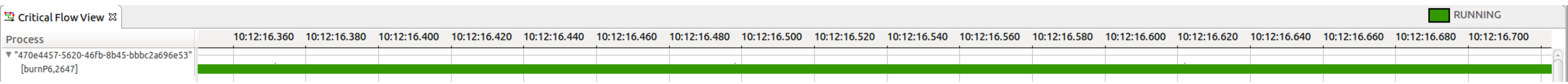
Waits for DNS server

Waits for user to response

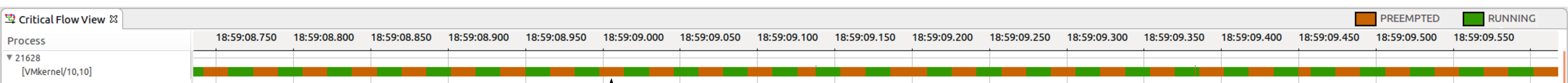
Investigation

Critical Path Analysis

Existing Critical Path Analysis in TraceCompass



Host-based Execution-graph Construction



Preemption State

Investigations

Overhead Analysis

CPA : Existing Critical Path Analysis in TraceCompass

HEC: Host-based Execution-graph Construction

Benchmark	Baseline	CPA	HEC	Overhead	
				CPA	HEC
File I/O (ms)	450.92	480.38	451.08	6.13%	0.03%
Memory (ms)	612.27	615.23	614.66	4.81%	0.01%
CPU (ms)	324.92	337.26	325.91	3.65%	0.30%

Investigations

Demo

Investigations

How to try these new features?

- Access to **Host** only
- Run **LTTng** on Host with my new added tracepoint (vcpu_enter_guest)
- Clone **TraceCompass** from github (incubator)
  - Open vCPU block View of TraceCompass (XML view)
  - Open vProcess block View of TraceCompass (XML view)
  - Open Nested VM vCPU Block View of TraceCompass (XML view)
  - Open Nested VM vProcess Block View of TraceCompass (XML view)
  - Use Execution Flow Analysis of TraceCompass

Conclusion

Inferences

- Wait Analysis of process inside VM and Nested VM
  - A process is waiting for
    - A Disk Block request to finish
    - A Network packet to receive
    - Another process
    - A Timer to fire
    - Other devices

- Critical Path Analysis of process inside VM and Nested VM

Questions?

Hani.nemati@polymtl.ca

https://github.com/Nemati