VM Analysis

Execution Path Analysis of Virtualized Environments using Host Kernel Tracing

Hani Nemati

Dec 7, 2017

Polytechnique Montréal

Laboratoire **DORSAL**

Agenda

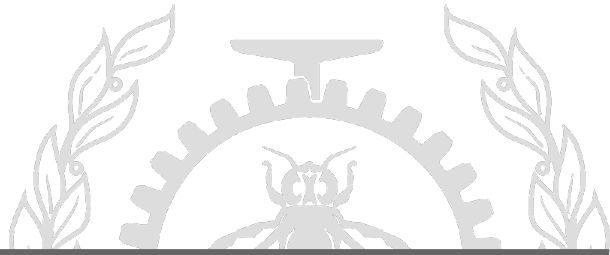
Introduction

- Research update and research motivation

New Investigations

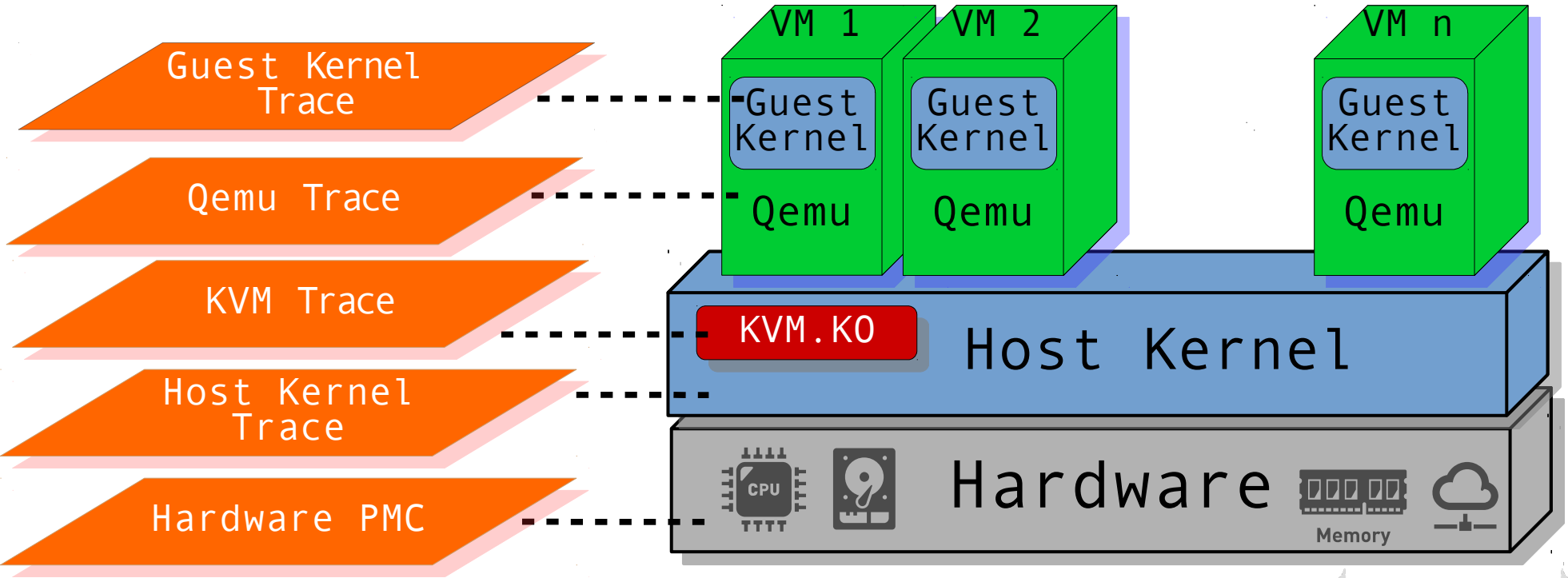
- Execution Path Analysis of virtualized environments using host kernel tracing
  - Sate of the art
  - Proposed Algorithm
  - Demo

Conclusion and in-progress



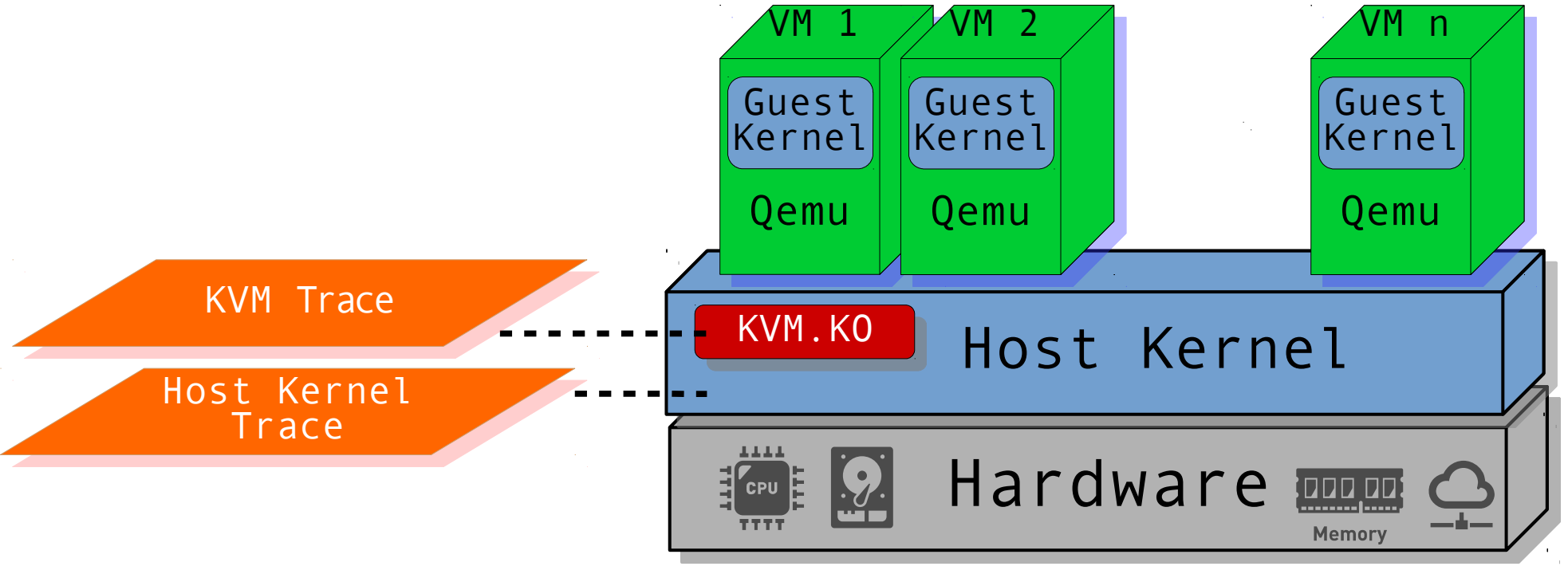
Previously on “VM Analysis”

Available Trace-Points in different layers



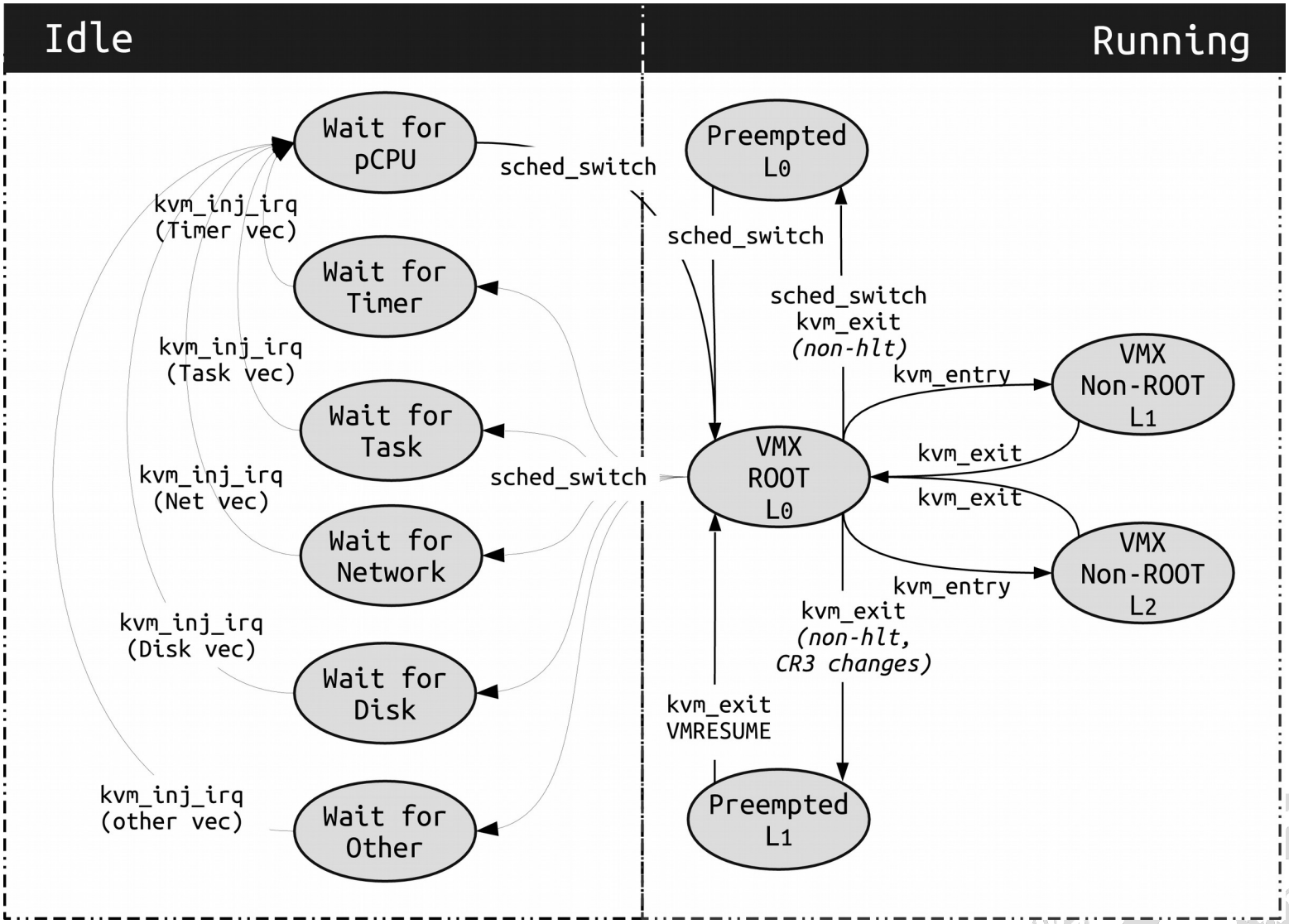
Previously on “VM Analysis”

Used Trace-Points in our approaches



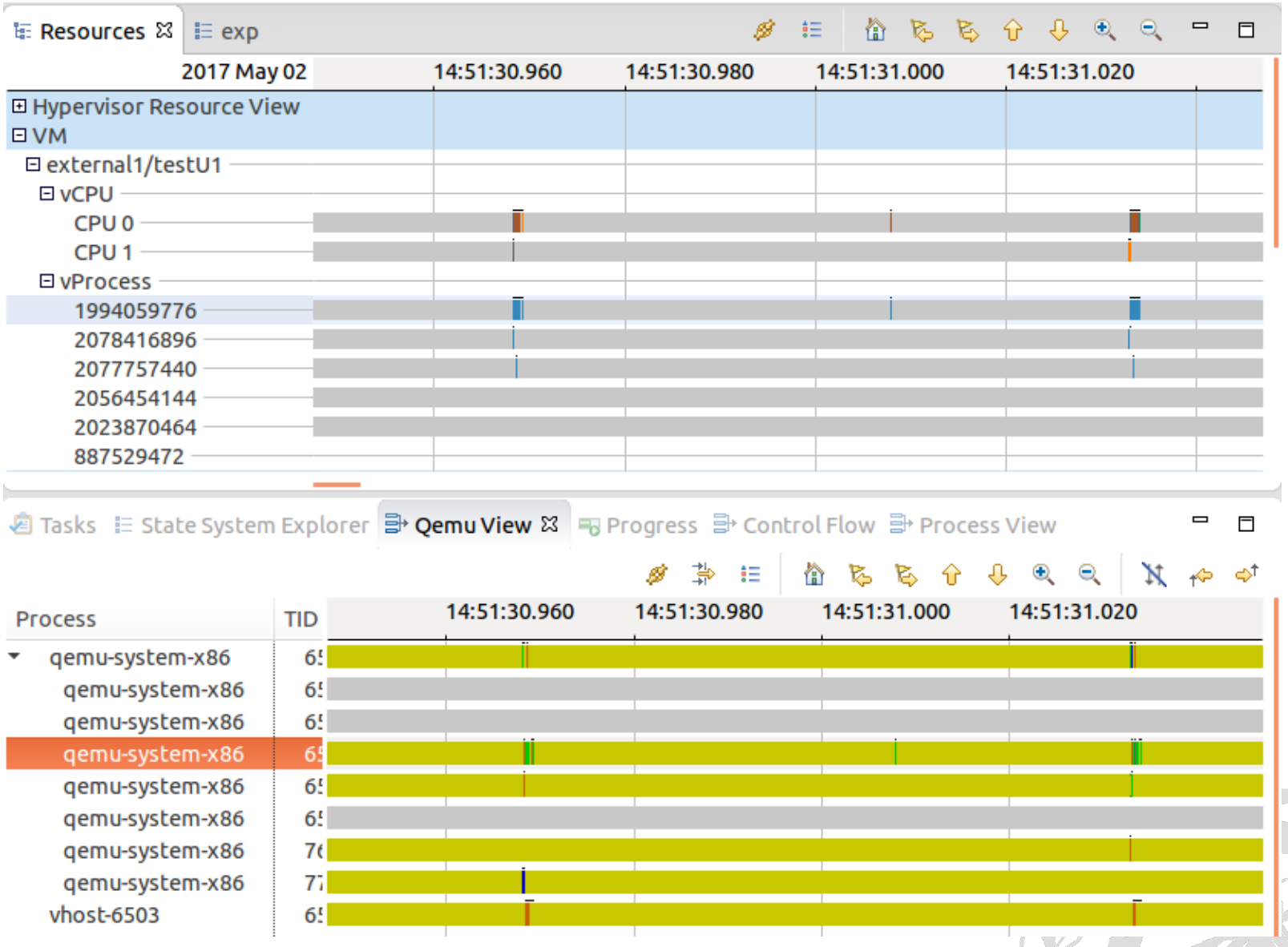
Previously on "VM Analysis"

States for a Process inside VM



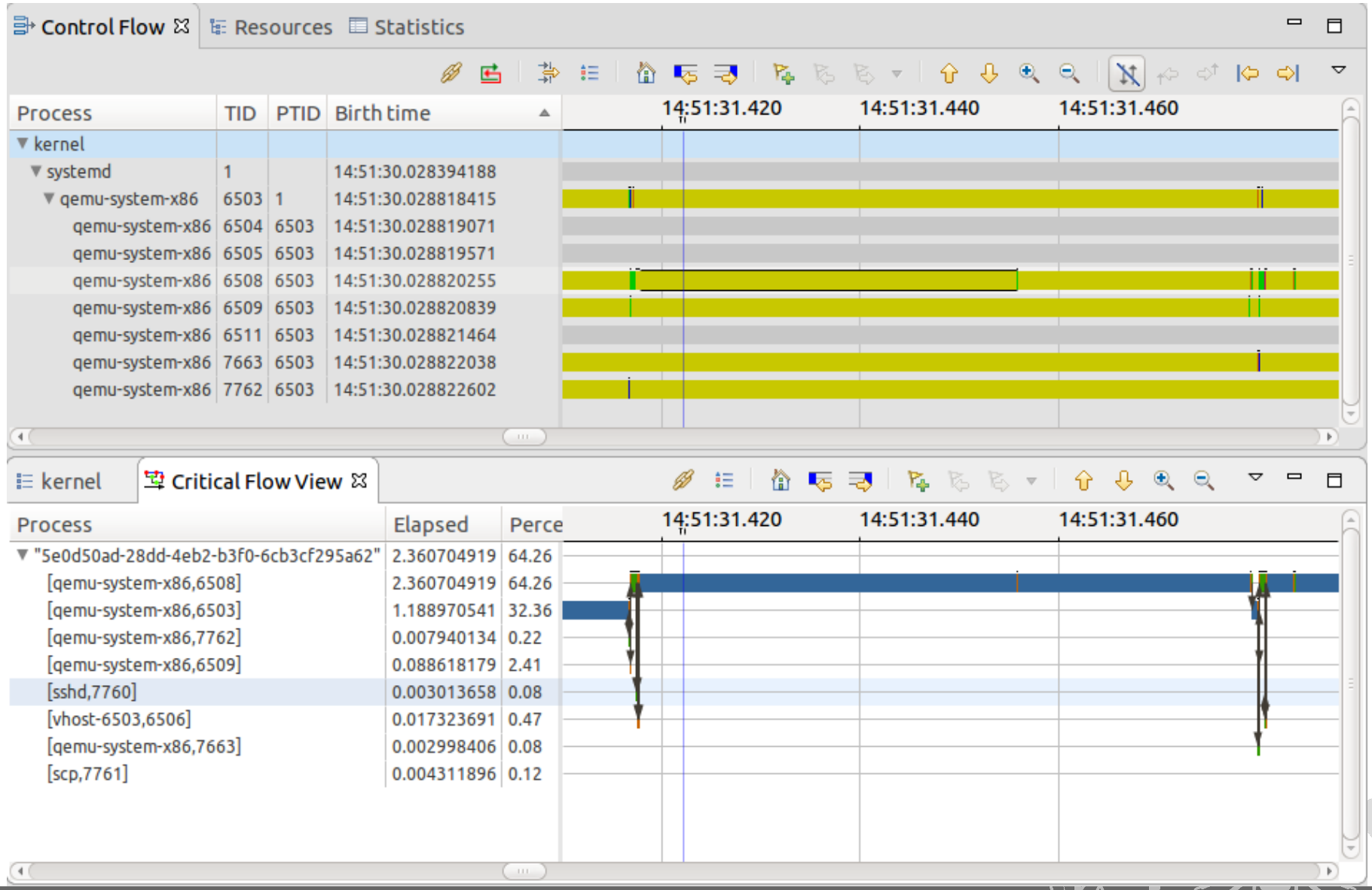
Motivation

Why is the VM waiting?



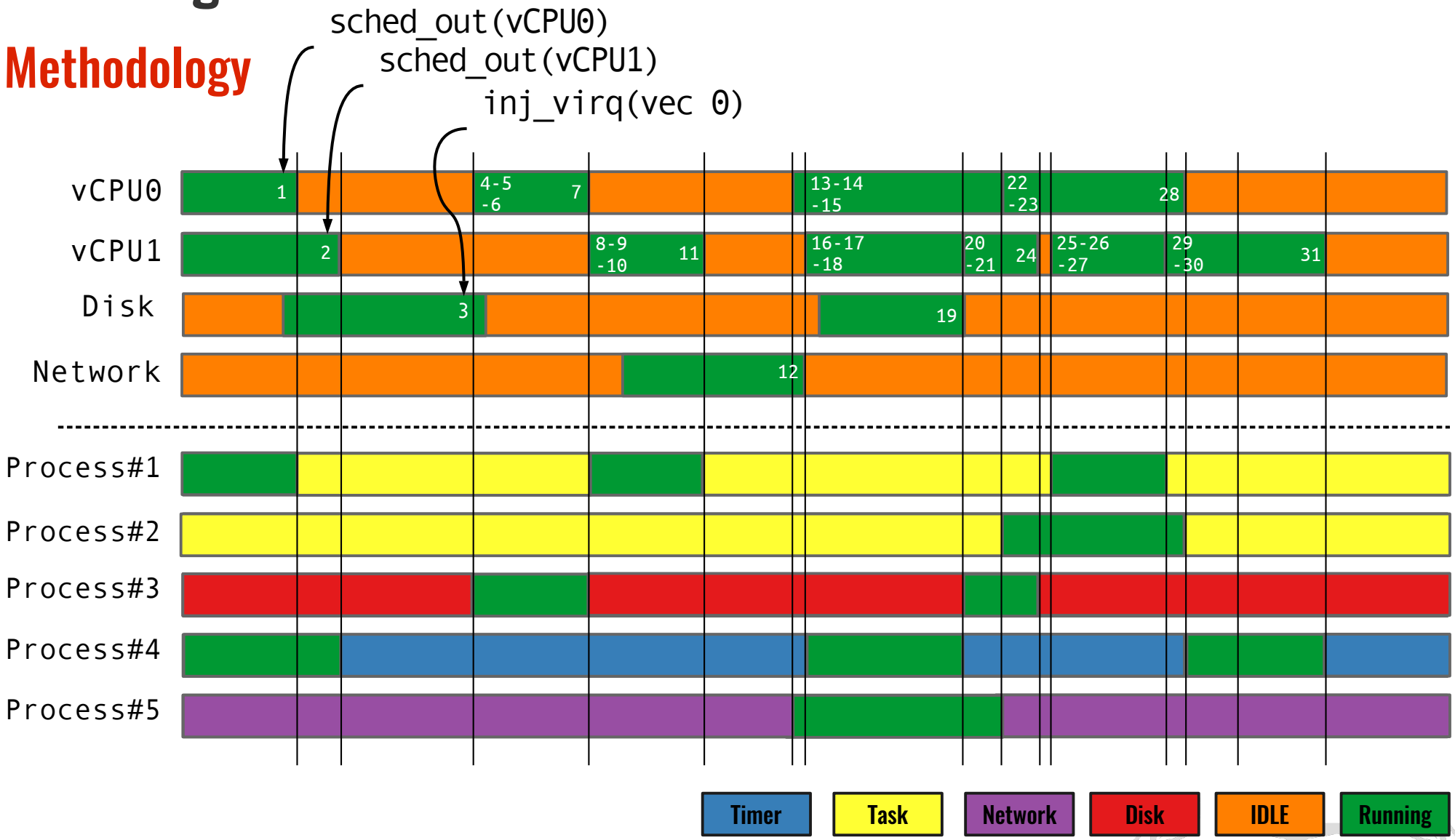
Motivation

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



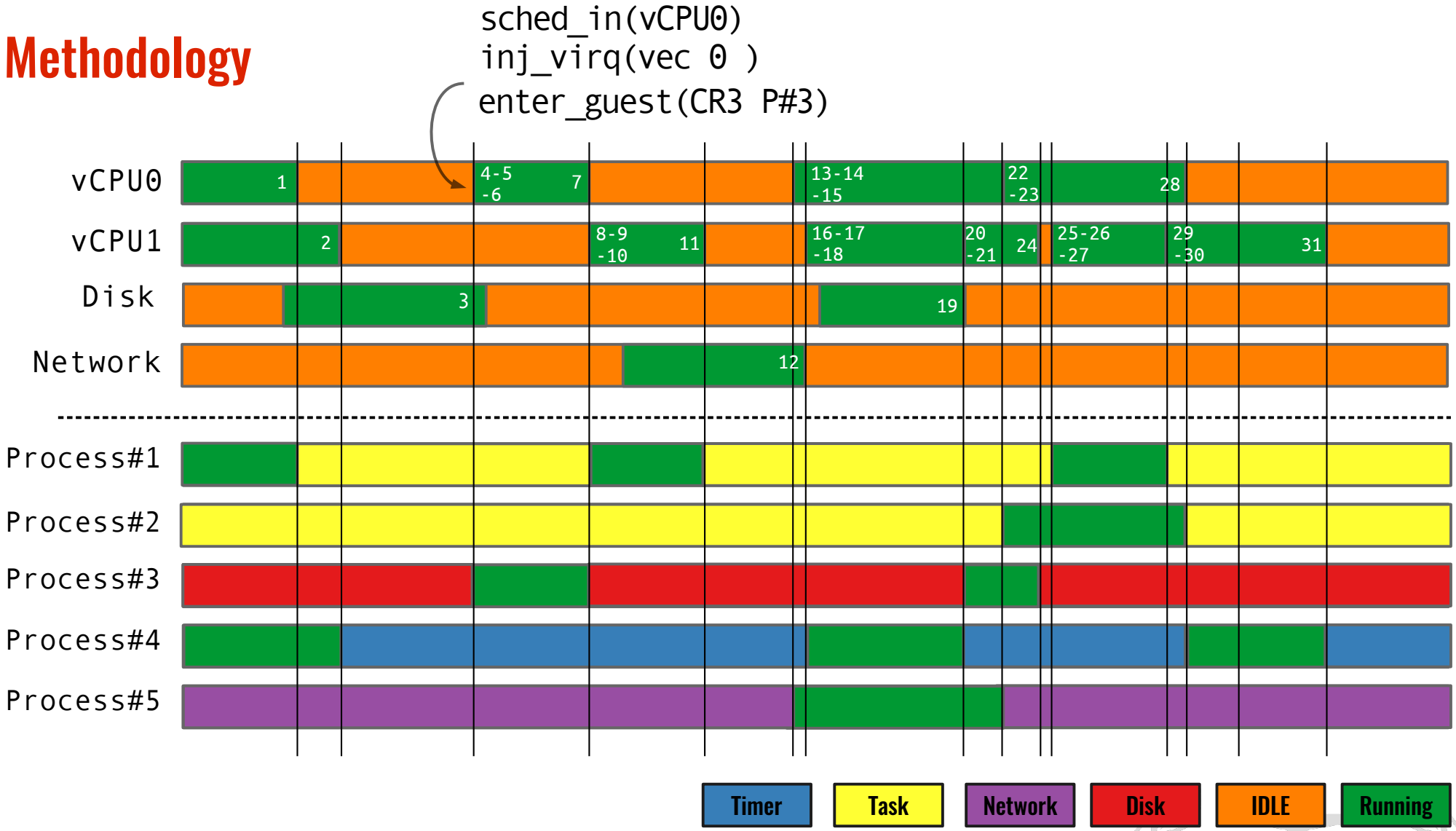
Investigations

Methodology



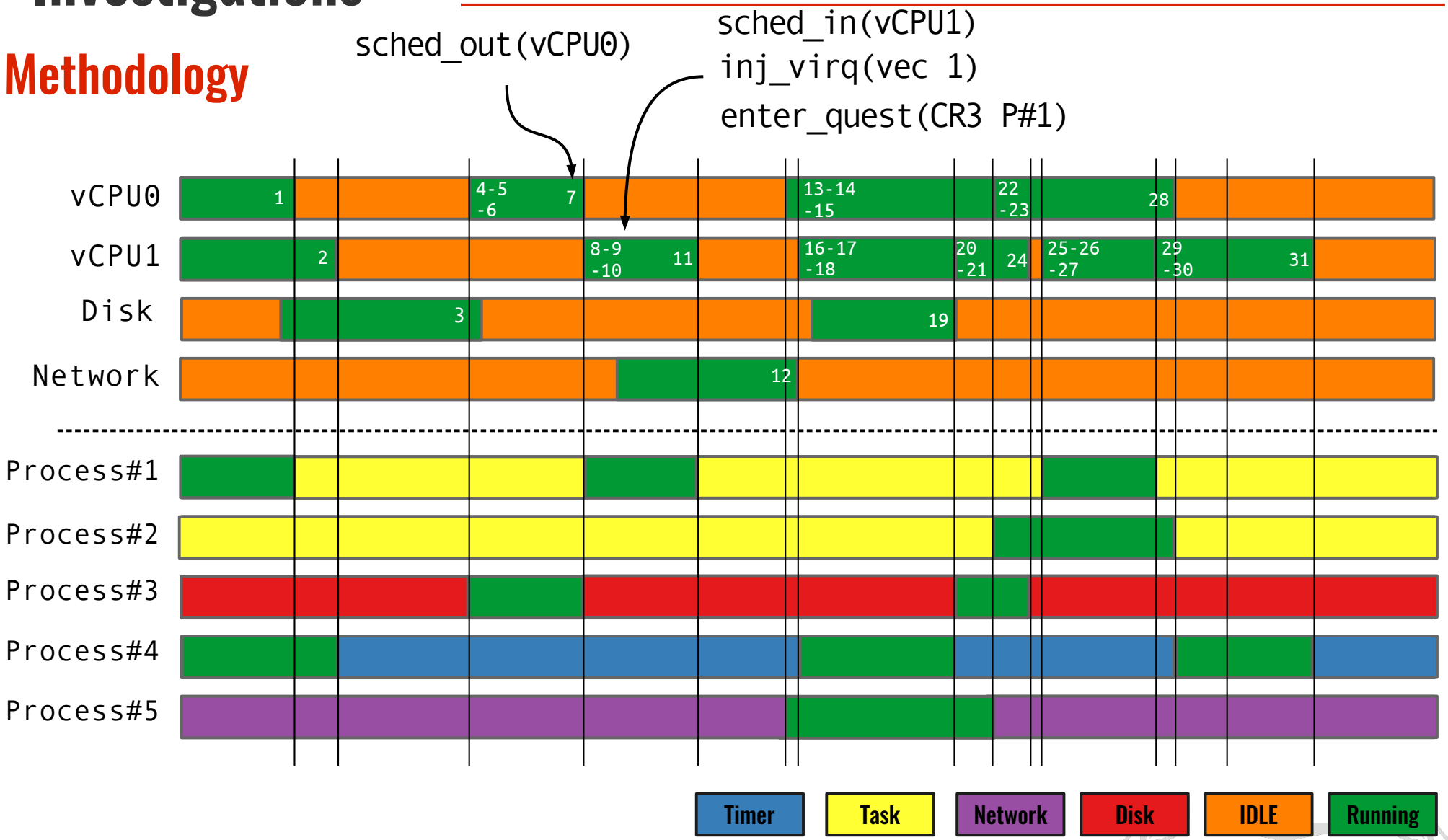
Investigations

Methodology



Investigations

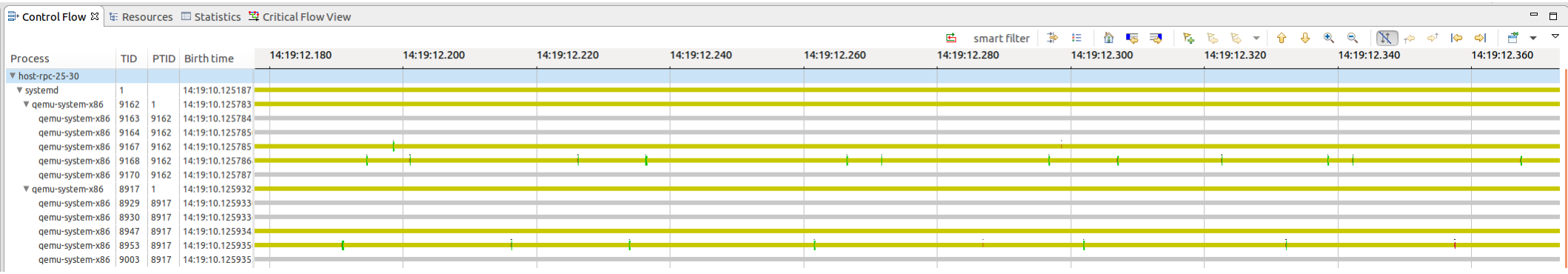
Methodology



Investigations

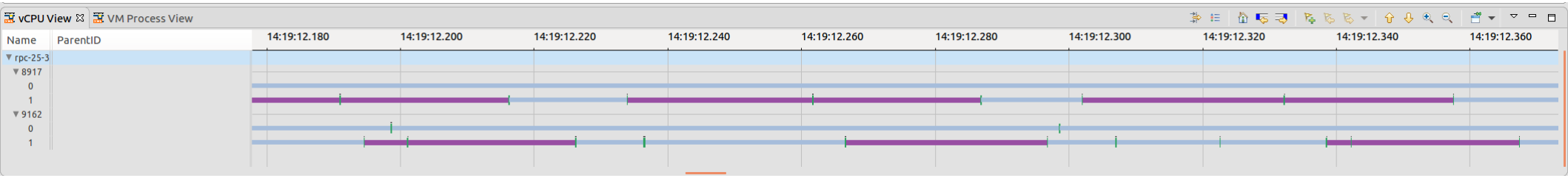
Control Flow View

Block Running



vCPU View

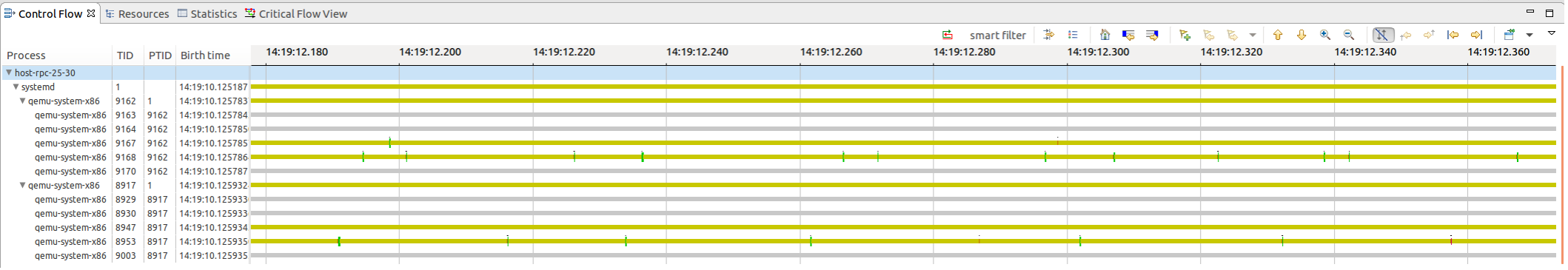
Timer Network Running



Investigations

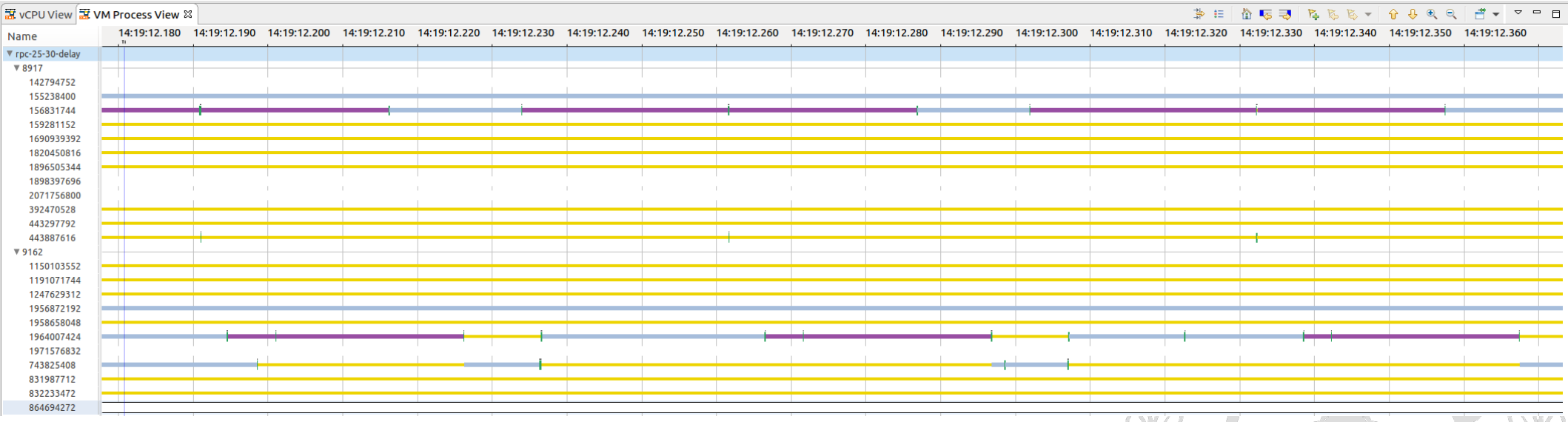
Control Flow View

Block Running



vProcess View

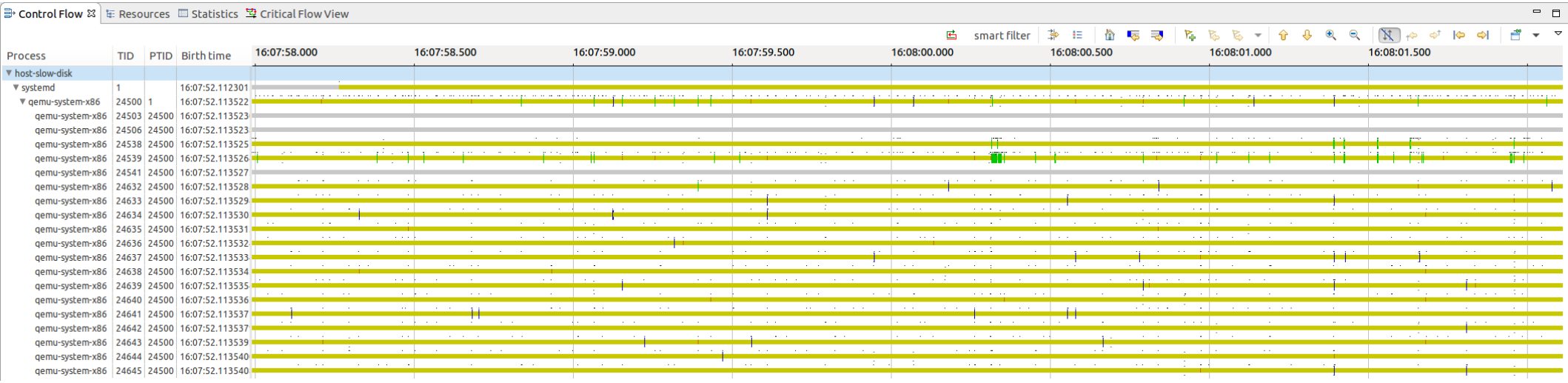
Task Timer Network Running



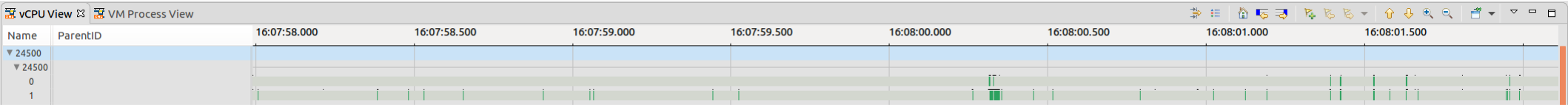
Investigations

Execution Flow Analysis View

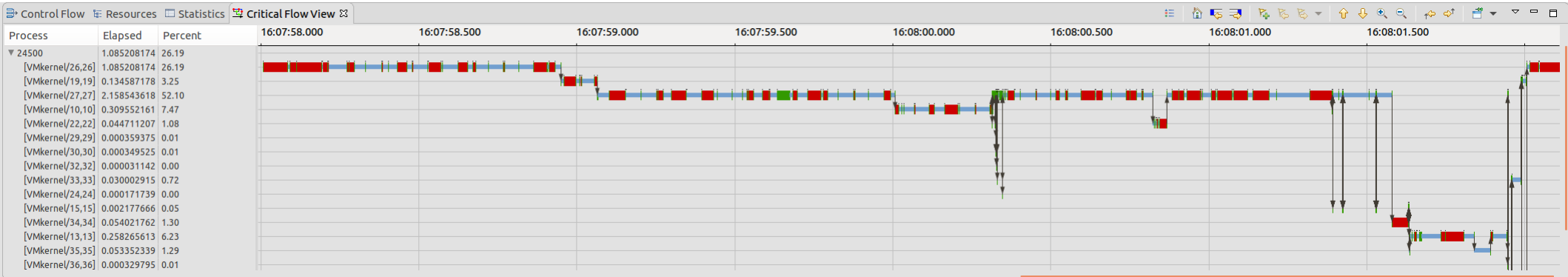
Running Block



Running Idle-unknown



Task Timer Disk Running



Investigations

Slow Hadoop Slave

Processes	Root	non-Root	Task	Timer	Disk	Net
2039963648	0.004	1.499	54.040	43.512	0.198	0.217
869769216	0.001	0.979	39.938	56.978	1.251	0.700
2046287872	0.004	3.125	57.357	38.022	0.766	0.283
2029756416	0.002	1.898	36.508	60.883	0.098	0.340
877412352	0.001	0.970	24.947	72.767	0.029	1.130
886243328	0.005	8.350	85.240	5.588	0.003	0.258

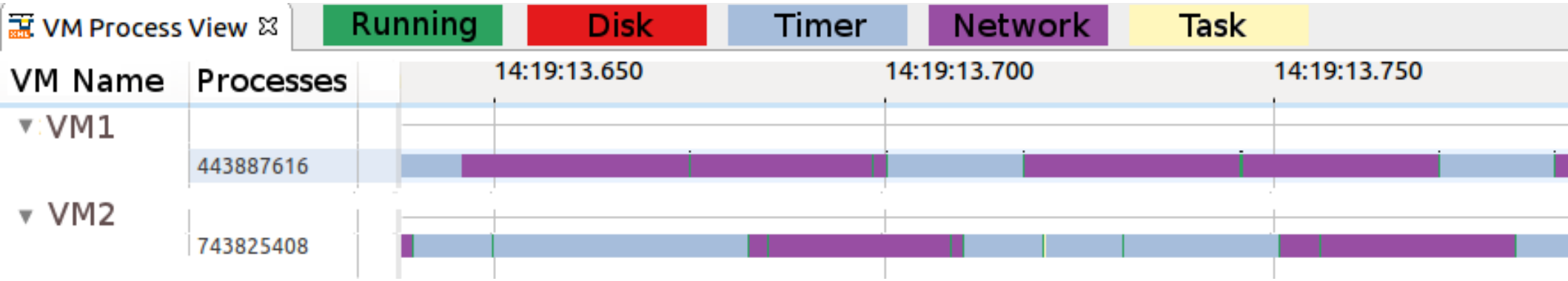
Investigations

Slow Hadoop Slave

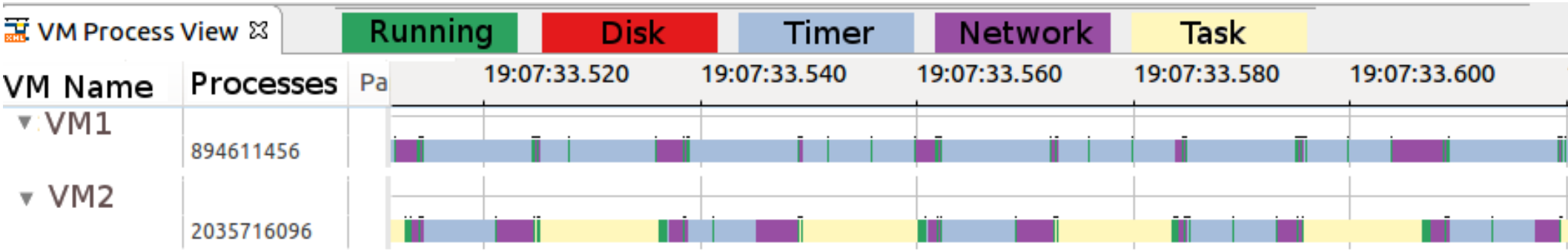
Processes	Root	non-Root	Task	Timer	Disk	Net
2039963648	0.004	1.499	54.040	43.512	0.198	0.217
869769216	0.001	0.979	39.938	56.978	1.251	0.700
2046287872	0.004	3.125	57.357	38.022	0.766	0.283
2029756416	0.002	1.898	36.508	60.883	0.098	0.340
877412352	0.001	0.970	24.947	72.767	0.029	1.130
886243328	0.005	8.350	85.240	5.588	0.003	0.258

Investigations

Cap on Network

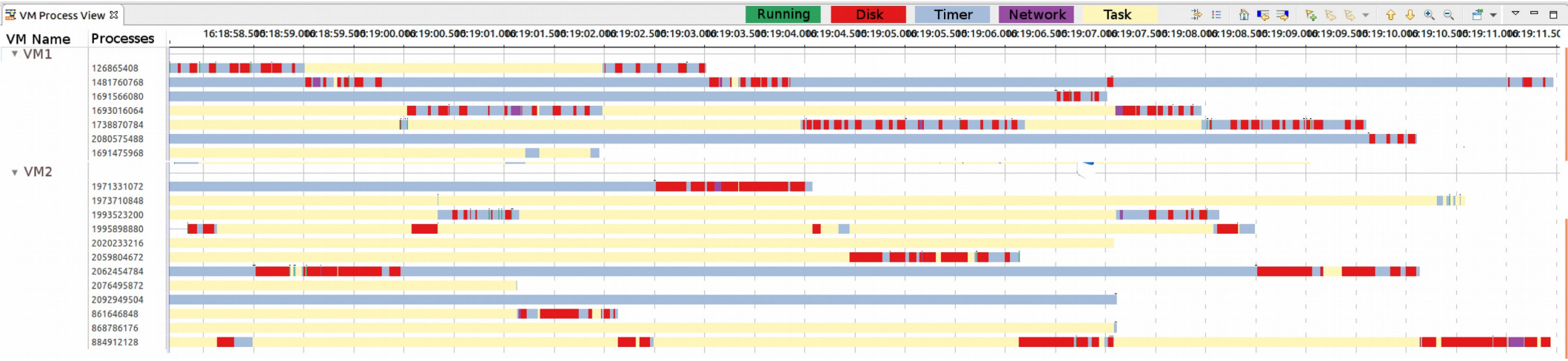


Without Cap on Network



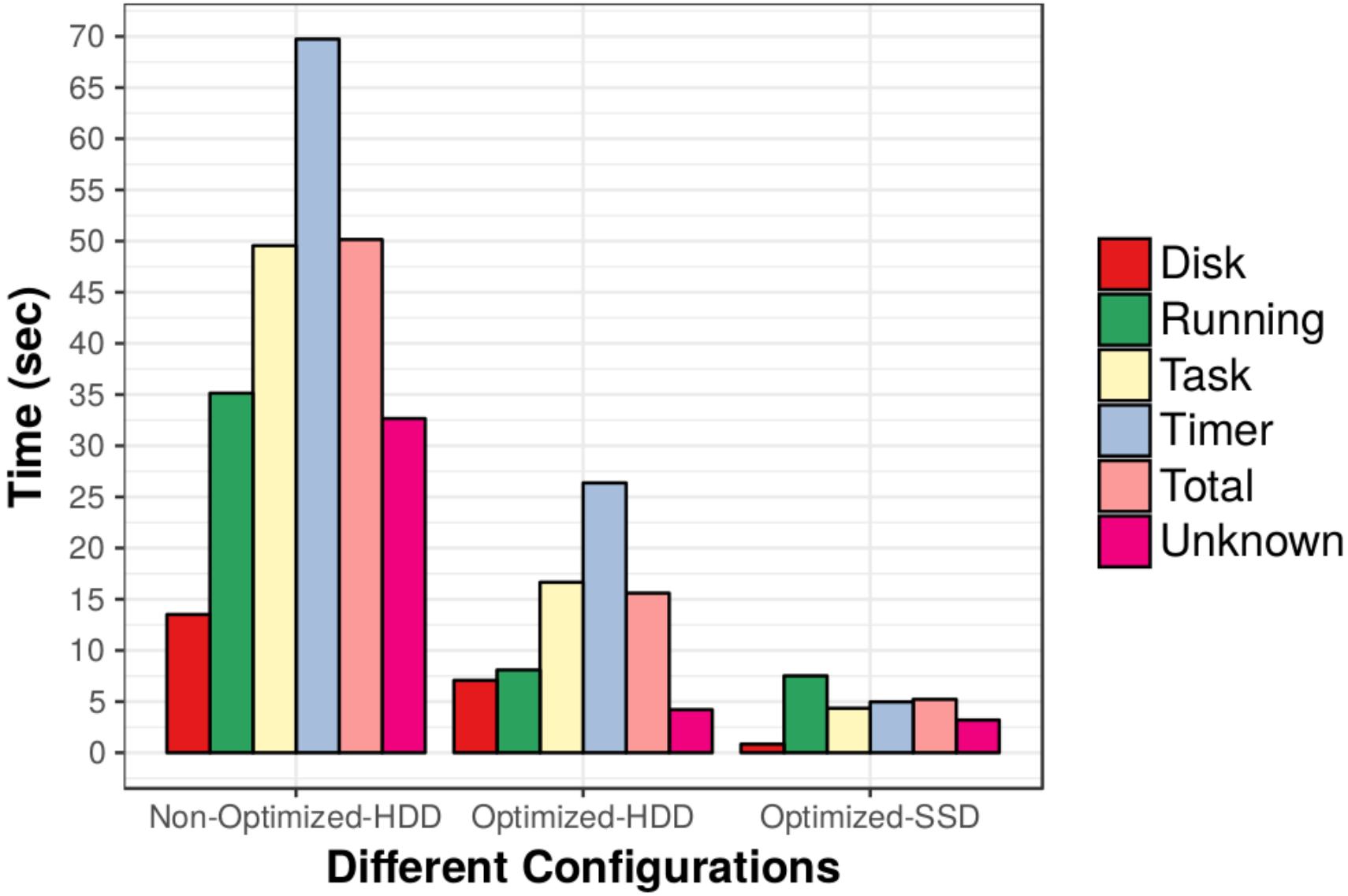
Investigations

Contention on Block Disk

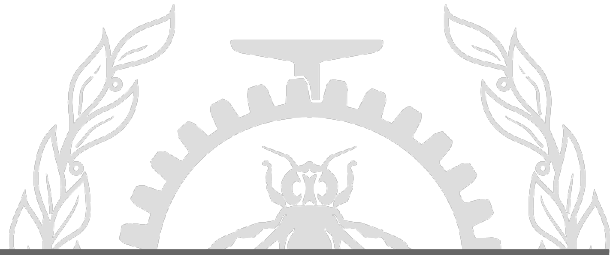


Investigations

VM Boot-up comparison



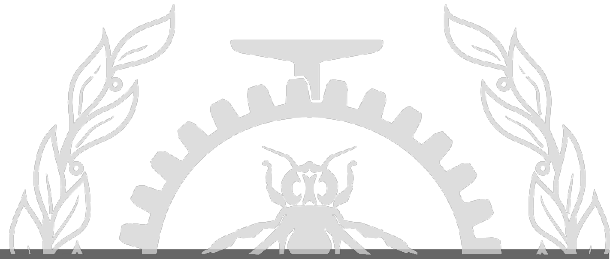
Demo



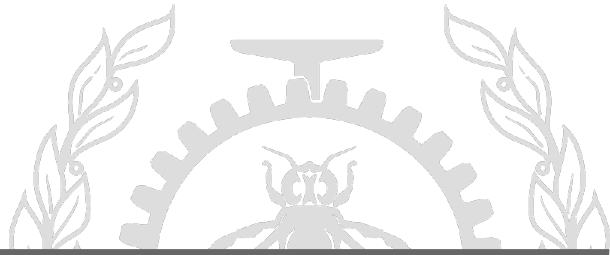
Investigations

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)
  - Use Execution Flow Analysis of TraceCompass

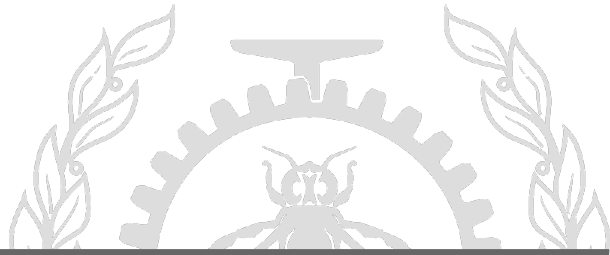


One More Thing ...



Let’s make TraceCompass Intelligent

Demo



Conclusion and in-progress

Inferences

- Wait Analysis of process inside 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

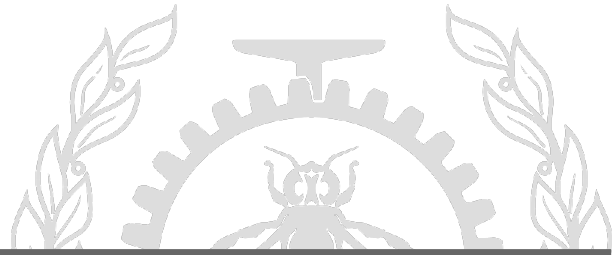
In progress

- VM contention avoidance based on VM classification

Outcome of this project

[1] Hani Nemati, Michel R. Dagenais, “ VM Processes State Detection by Hypervisor Tracing. “ Submitted to IEEE System Conference 2018

[2] Hani Nemati, Genevieve Bastien, Michel Dagenais “Wait analysis of Virtual Machines using host kernel tracing”, Accepted at IEEE Cloud Summit 2018



Questions?

Hani.nemati@polymtl.ca

https://github.com/Nemati