Multilayer virtualized systems analysis with Kernel Tracing

Cédric Biancheri
Michel Dagenais

5 May, 2016
École Polytechnique de Montréal

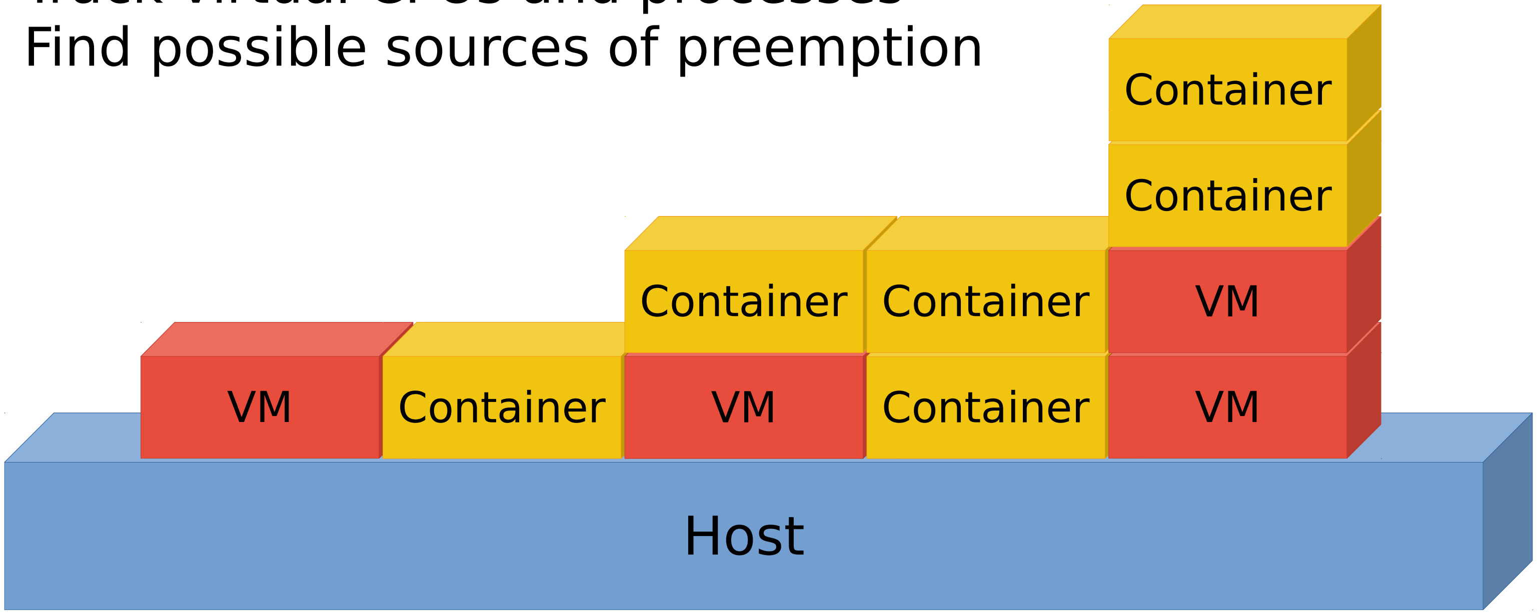
POLYTECHNIQUE
MONTRÉAL

Content

- General objectives
- Fused Virtual Machine Analysis
- Fused Virtual Machine View

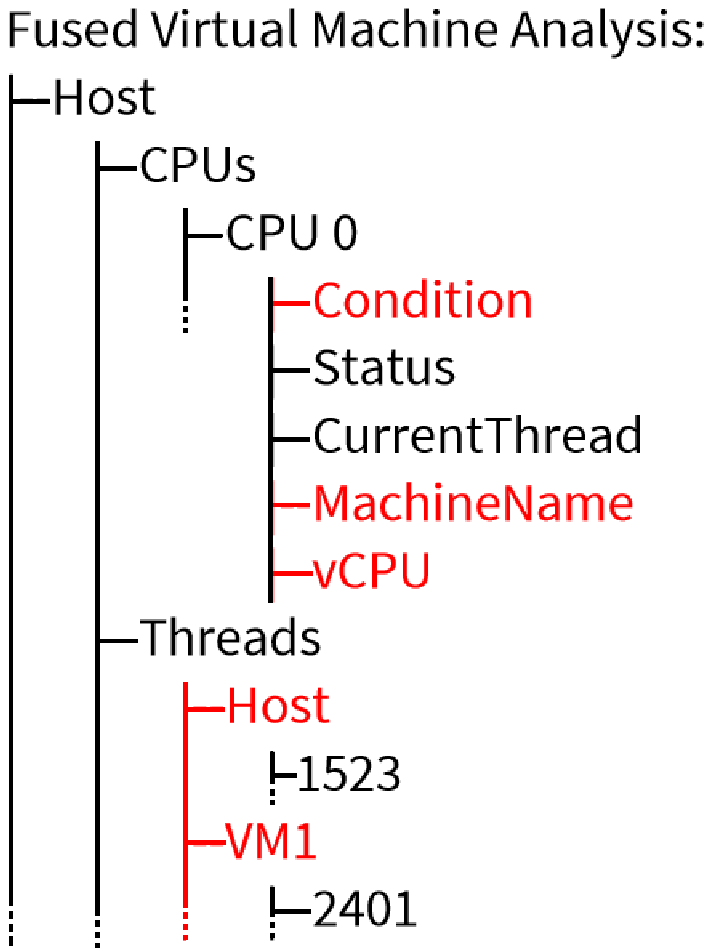
Objectives

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



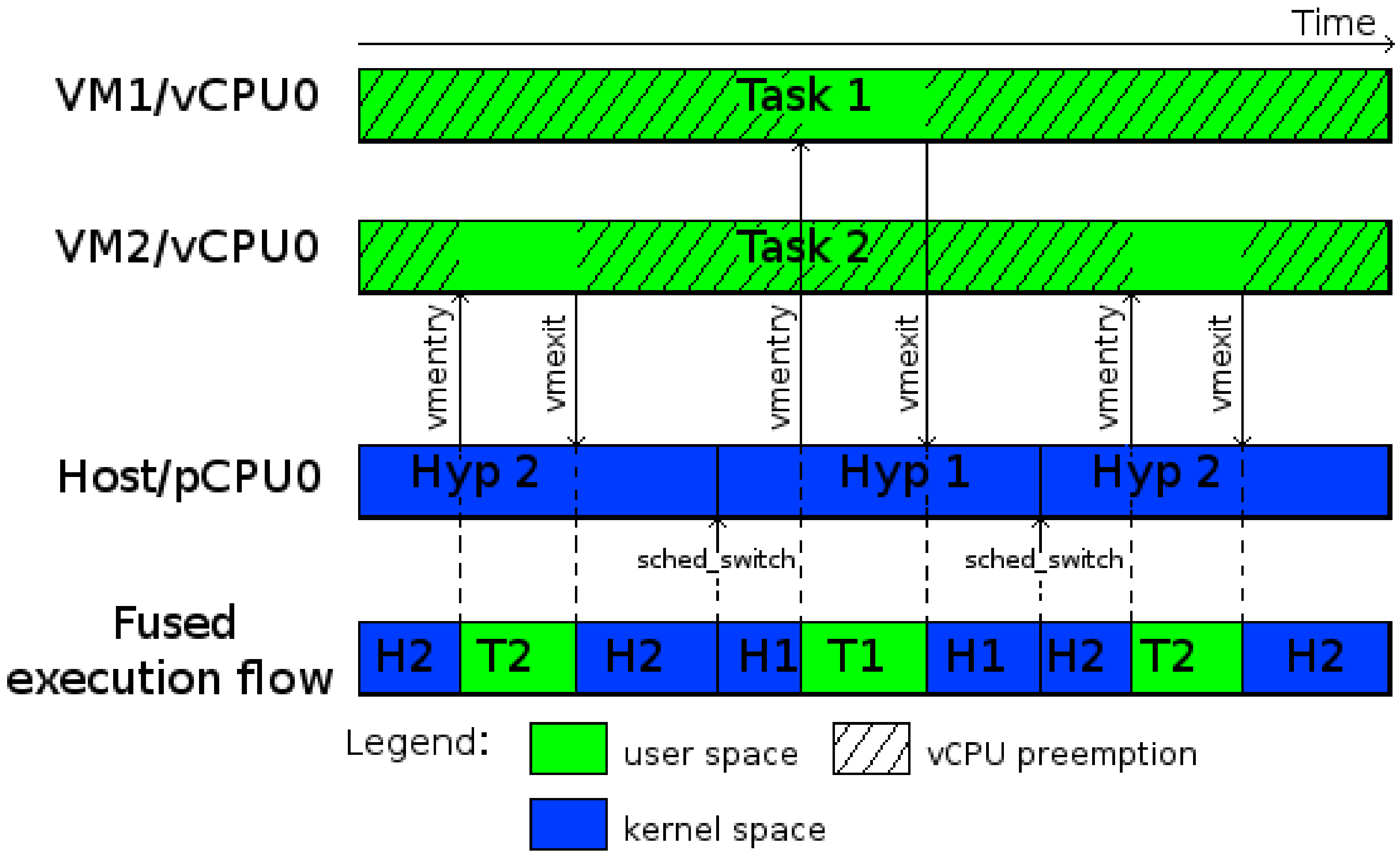
Fused Virtual Machine Analysis

- 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:
├─Host
    ├─CPUs
    │   ├─CPU 0
    │   ┆   ├─Condition
    │   ┆   ├─Status
    │   ┆   ├─CurrentThread
    │   ┆   ├─MachineName
    │   ┆   ├─vCPU
    ├─Threads
        ├─Host
        │   ├─1523
        ├─VM1
        ┆   ├─2401
```

Fused Virtual Machine Analysis



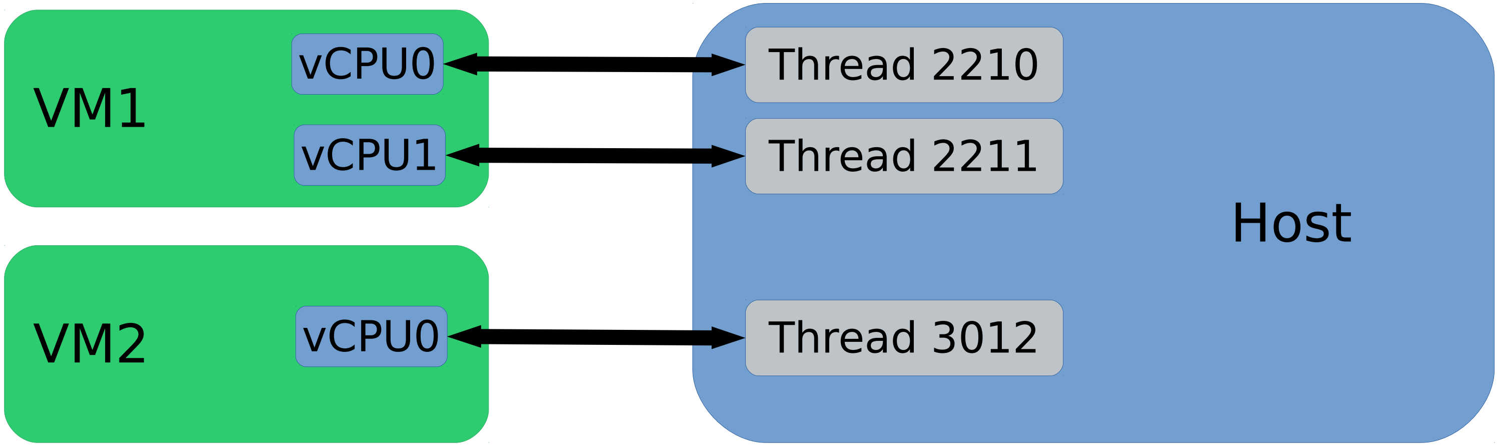
Fused Virtual Machine Analysis

Automatic recognition of the machine's role

Synchronization event for host → Host → Associate thread to VM

Synchronization event for VM → Virtual Machine

kvm_entry → Host → Associate thread to vCPU



Fused Virtual Machine Analysis

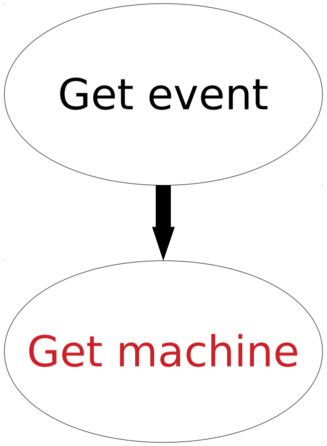
Handling events from all layers

Fused Virtual Machine Analysis

Handling events from all layers

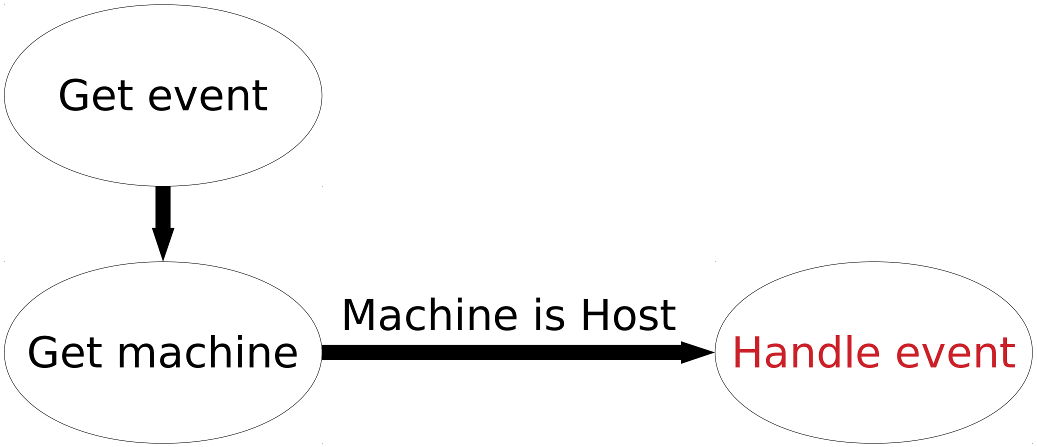
Fused Virtual Machine Analysis

Handling events from all layers



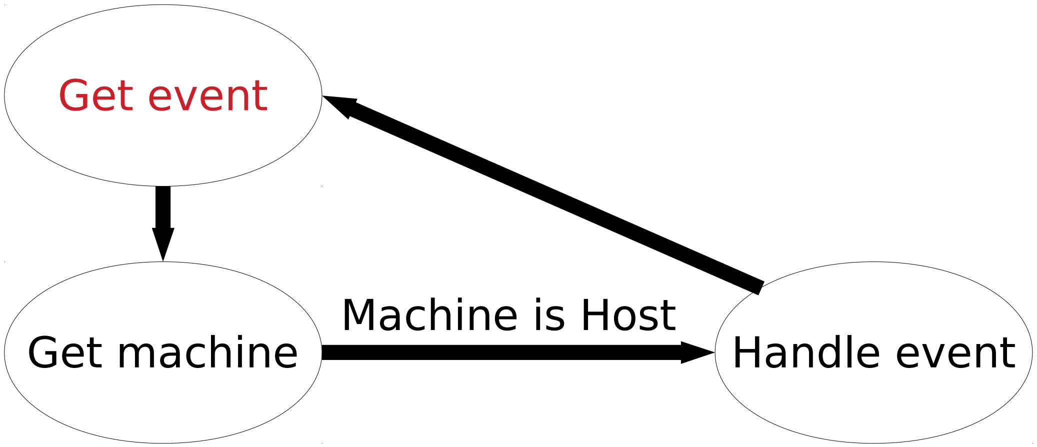
Fused Virtual Machine Analysis

Handling events from all layers



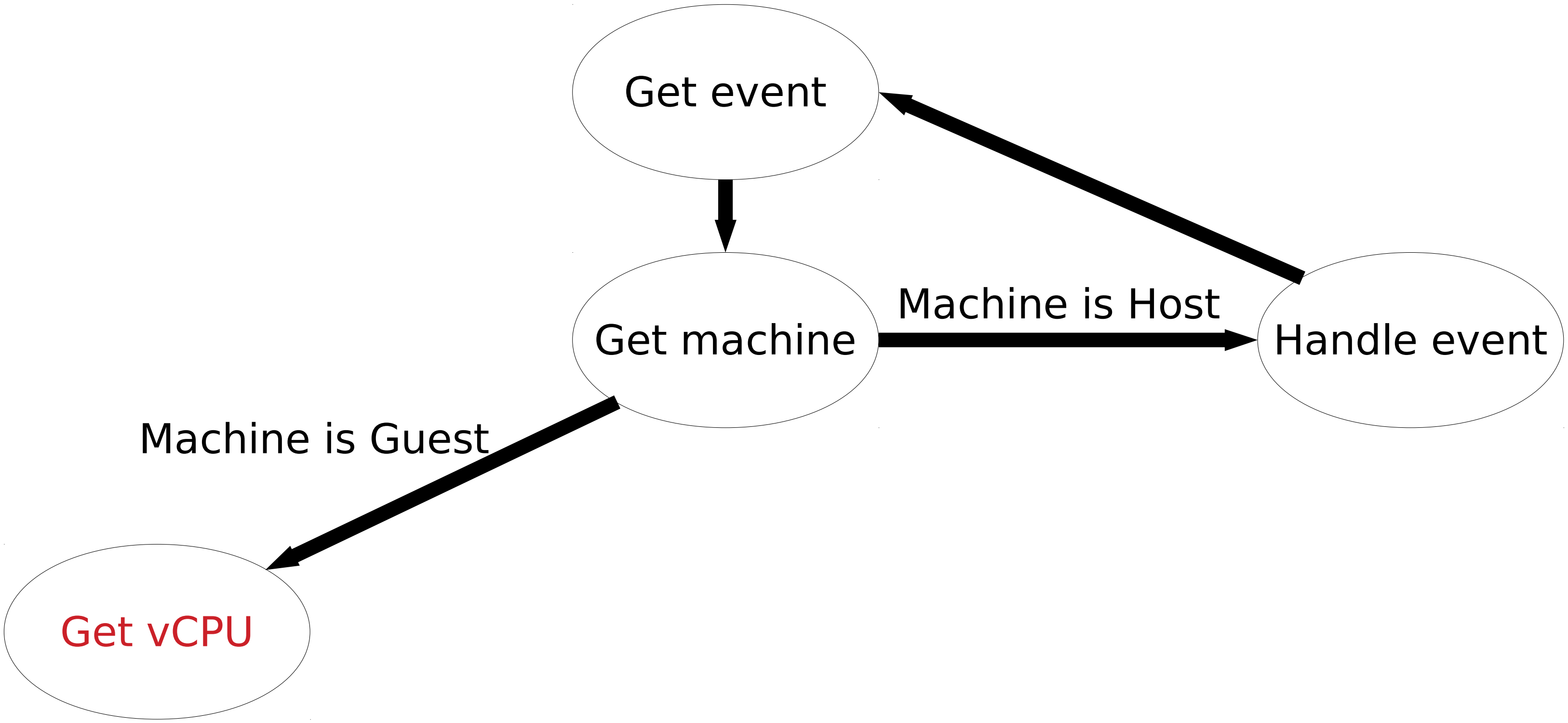
Fused Virtual Machine Analysis

Handling events from all layers



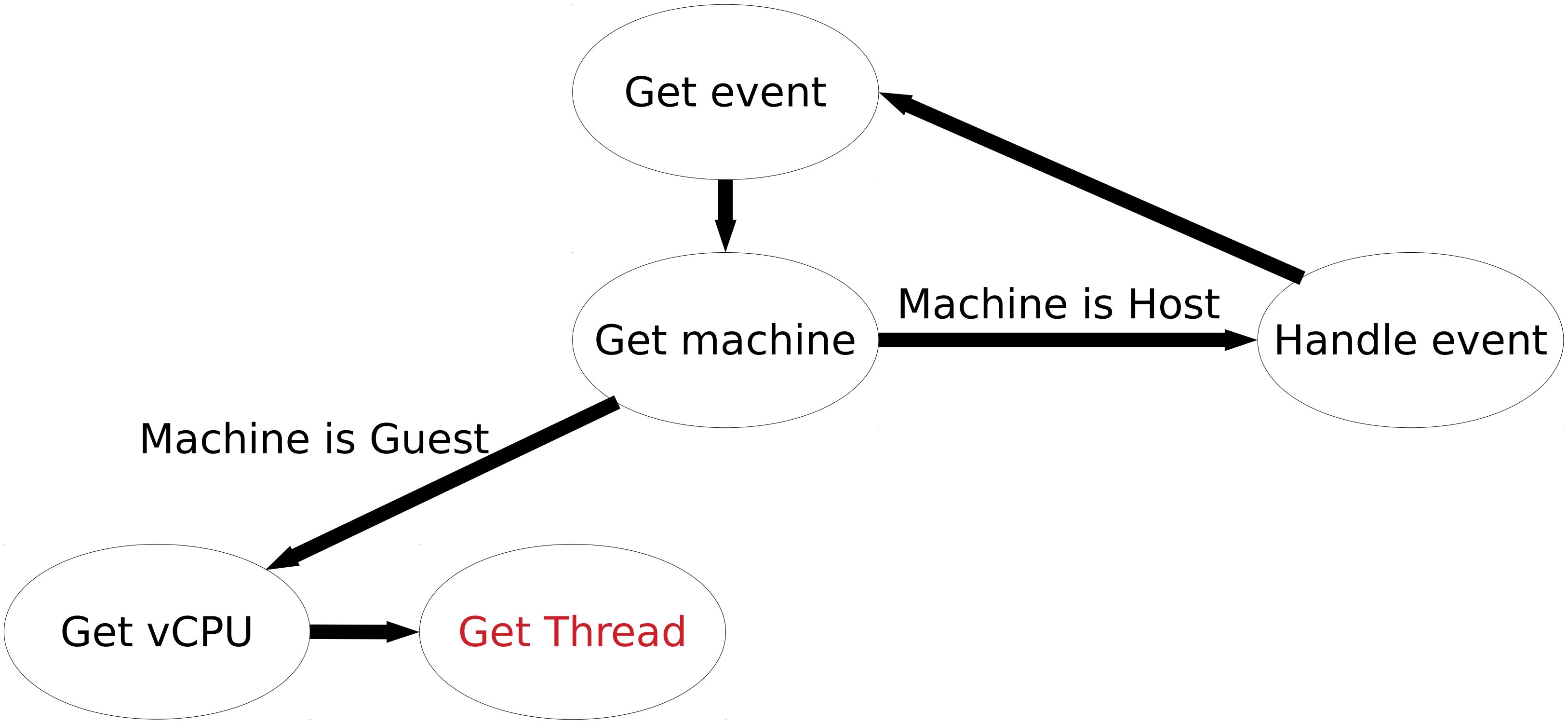
Fused Virtual Machine Analysis

Handling events from all layers



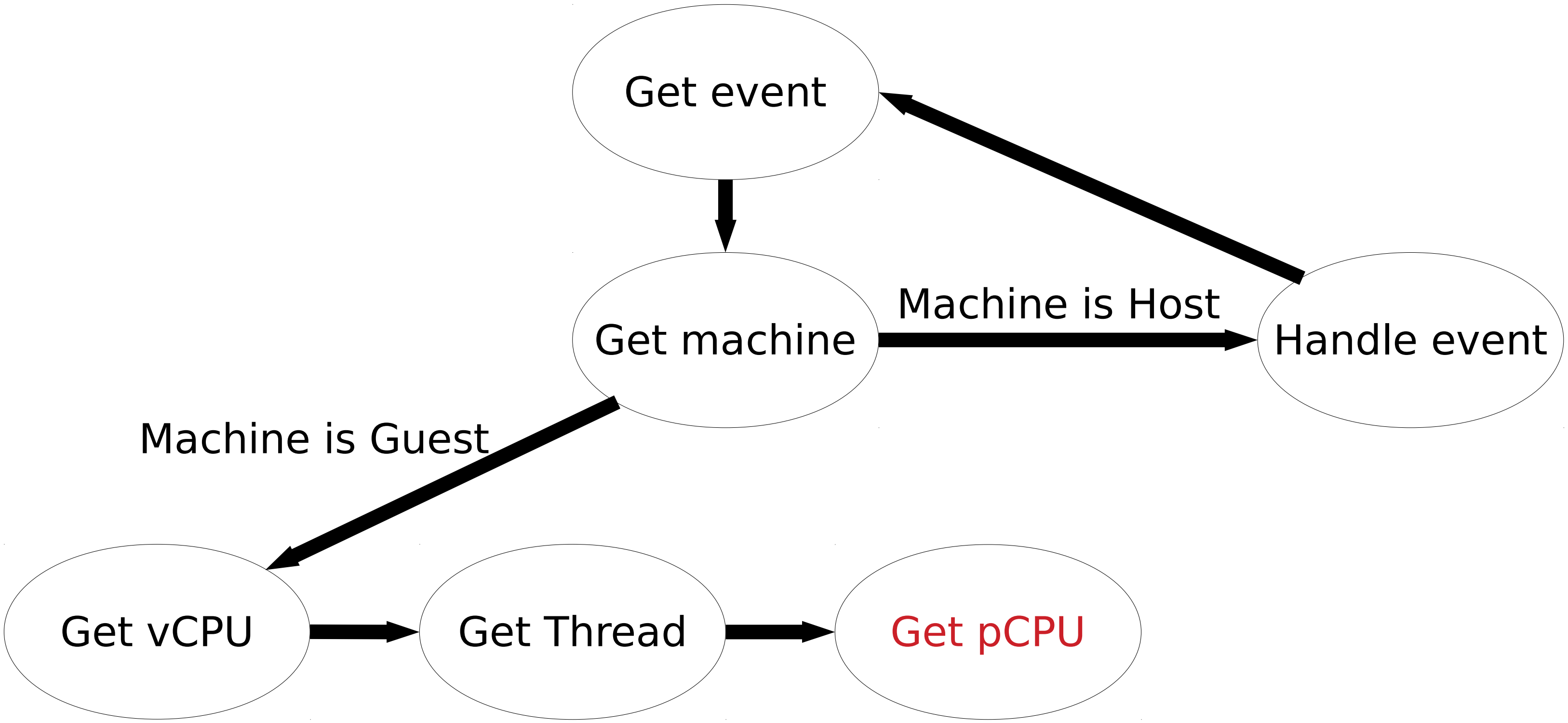
Fused Virtual Machine Analysis

Handling events from all layers



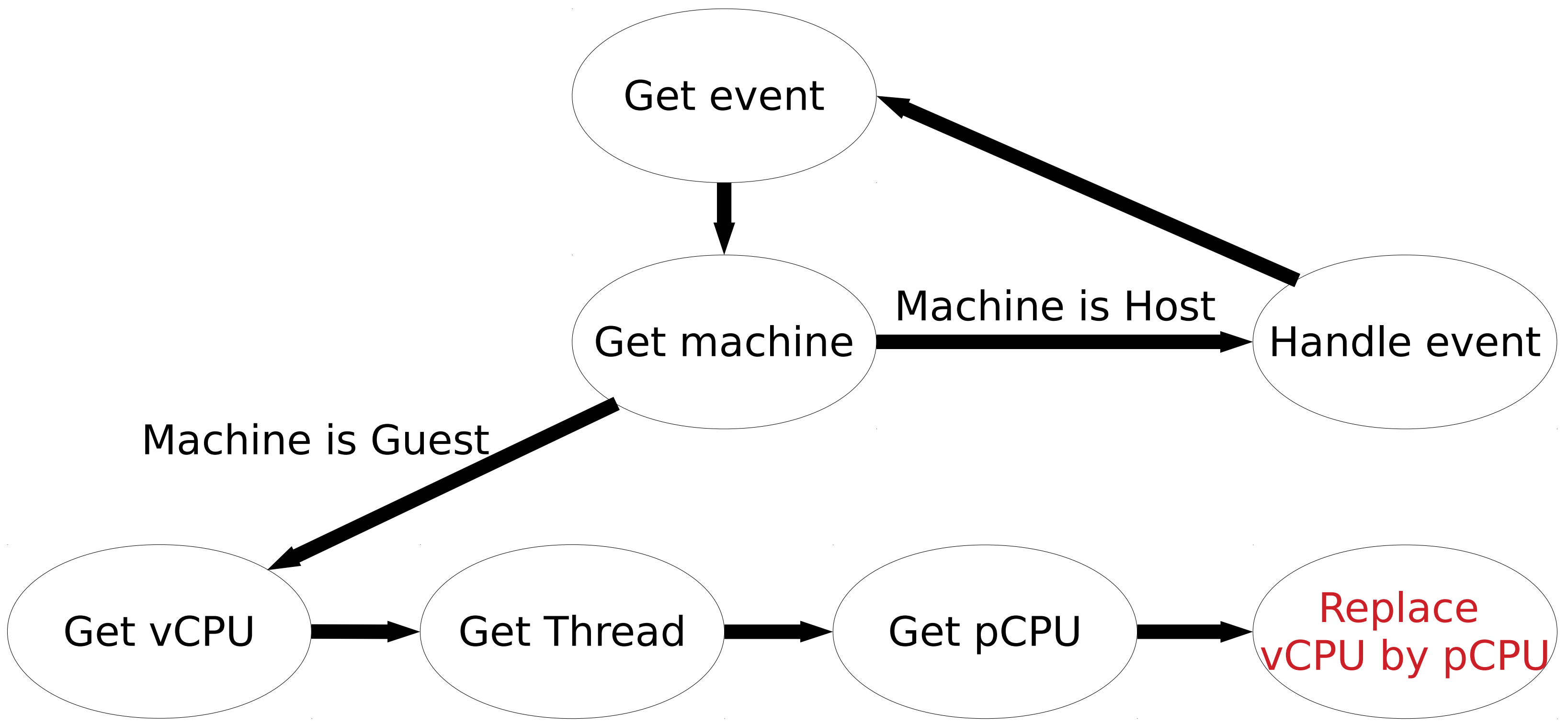
Fused Virtual Machine Analysis

Handling events from all layers



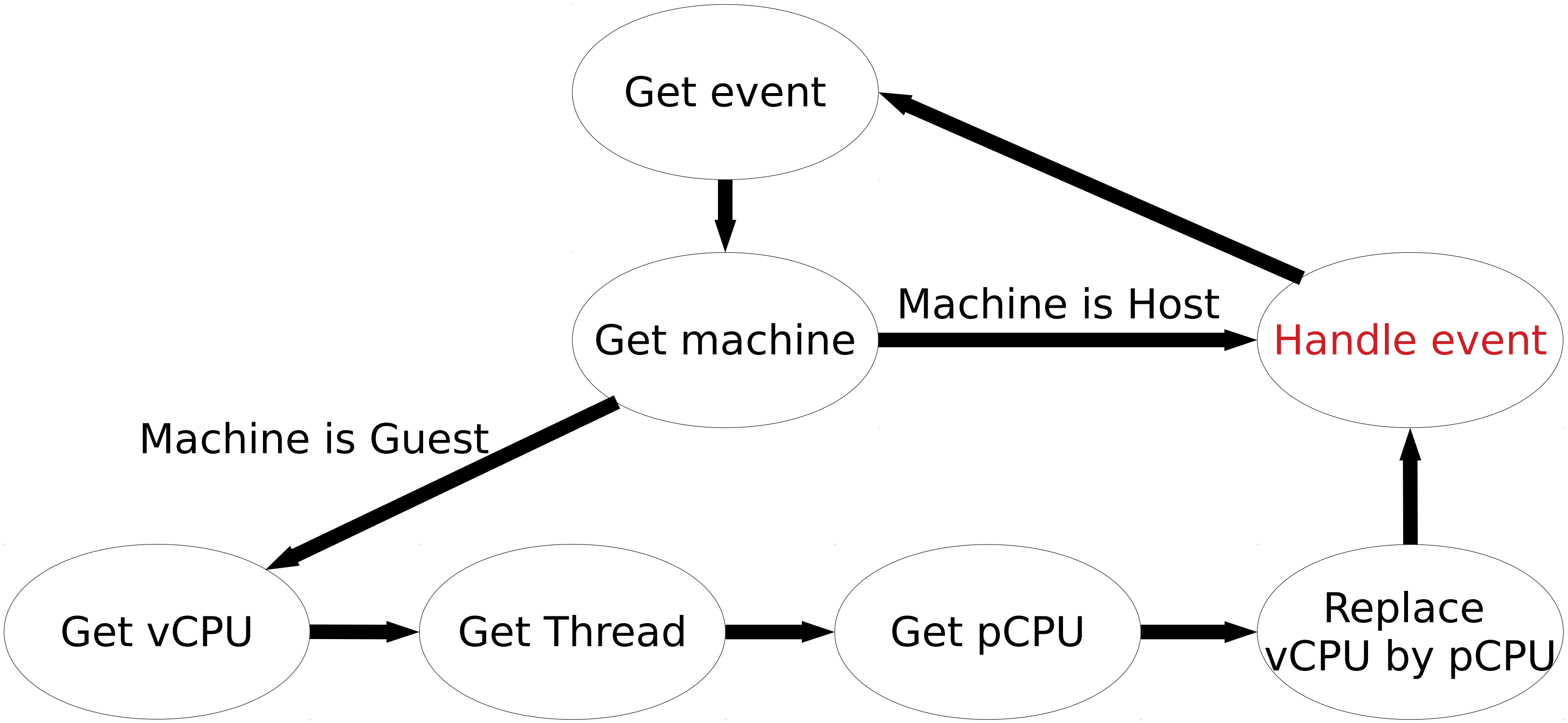
Fused Virtual Machine Analysis

Handling events from all layers



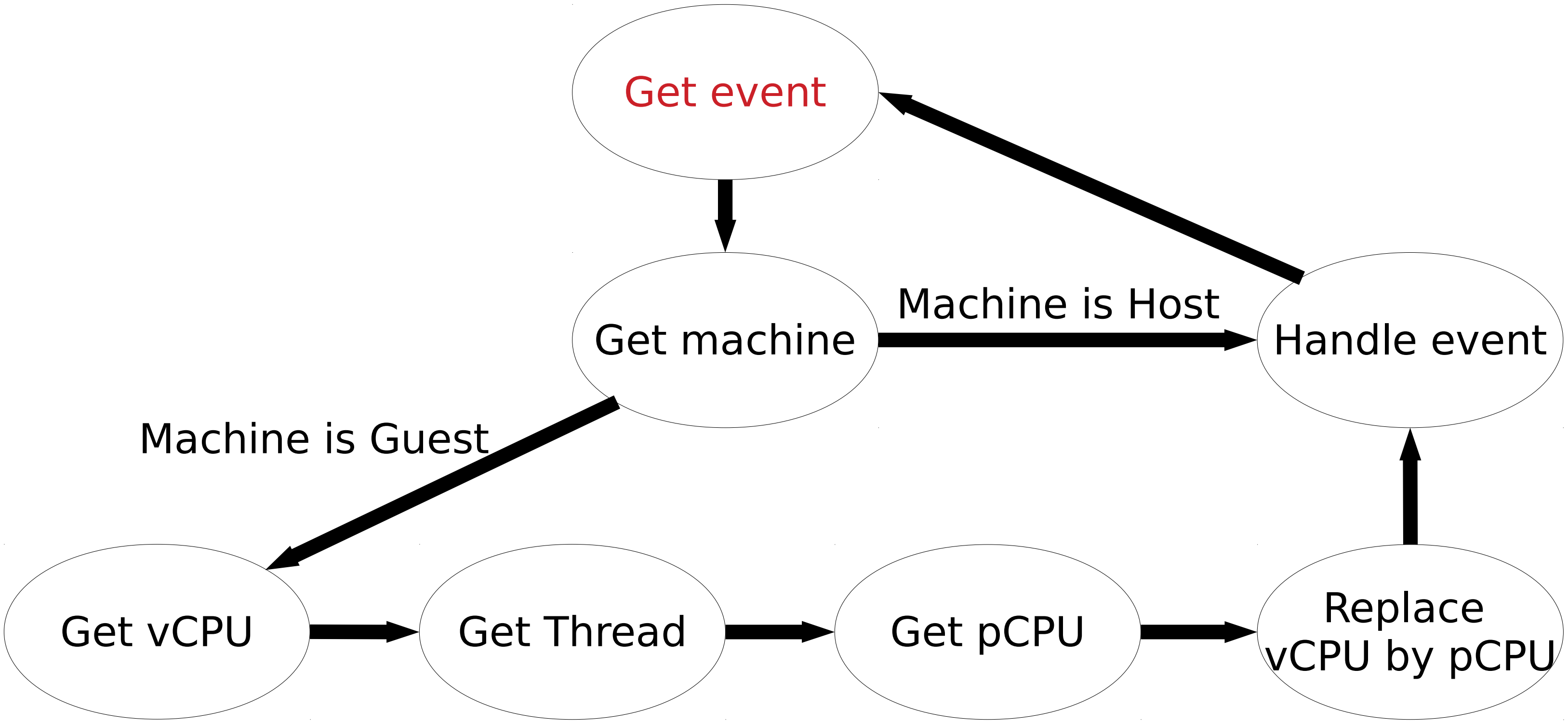
Fused Virtual Machine Analysis

Handling events from all layers



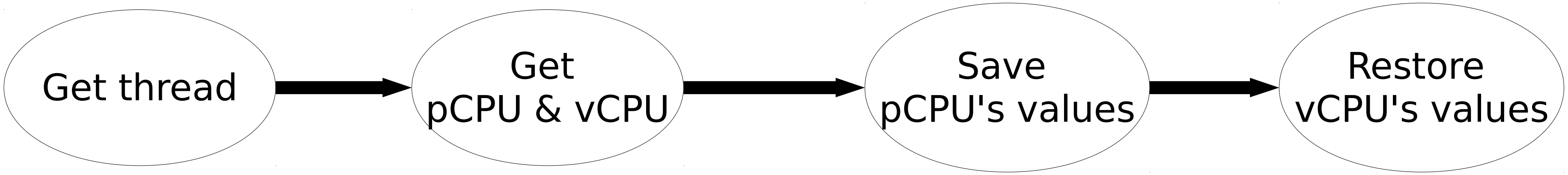
Fused Virtual Machine Analysis

Handling events from all layers



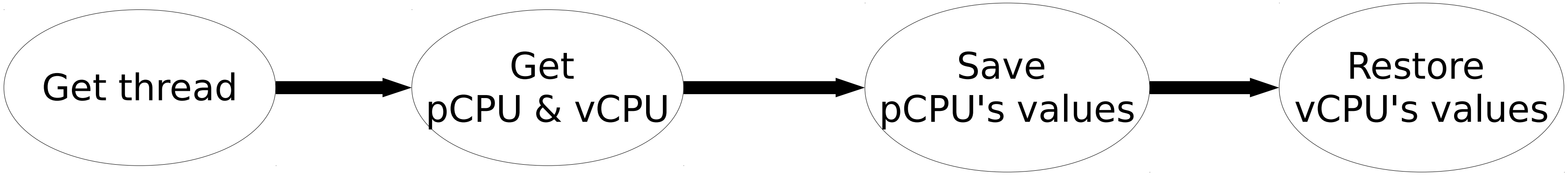
Fused Virtual Machine Analysis

The event is a VM Entry



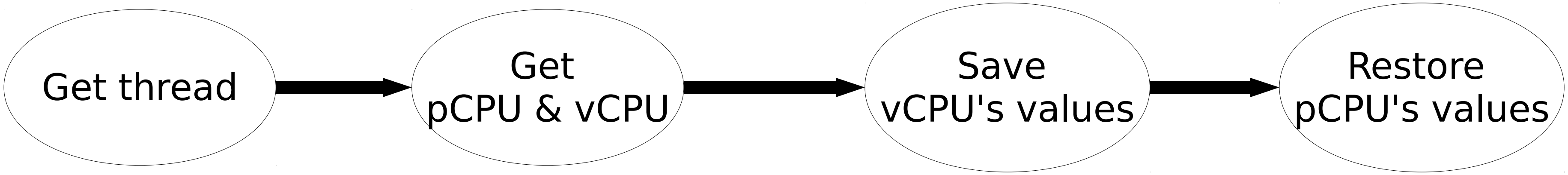
Fused Virtual Machine Analysis

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

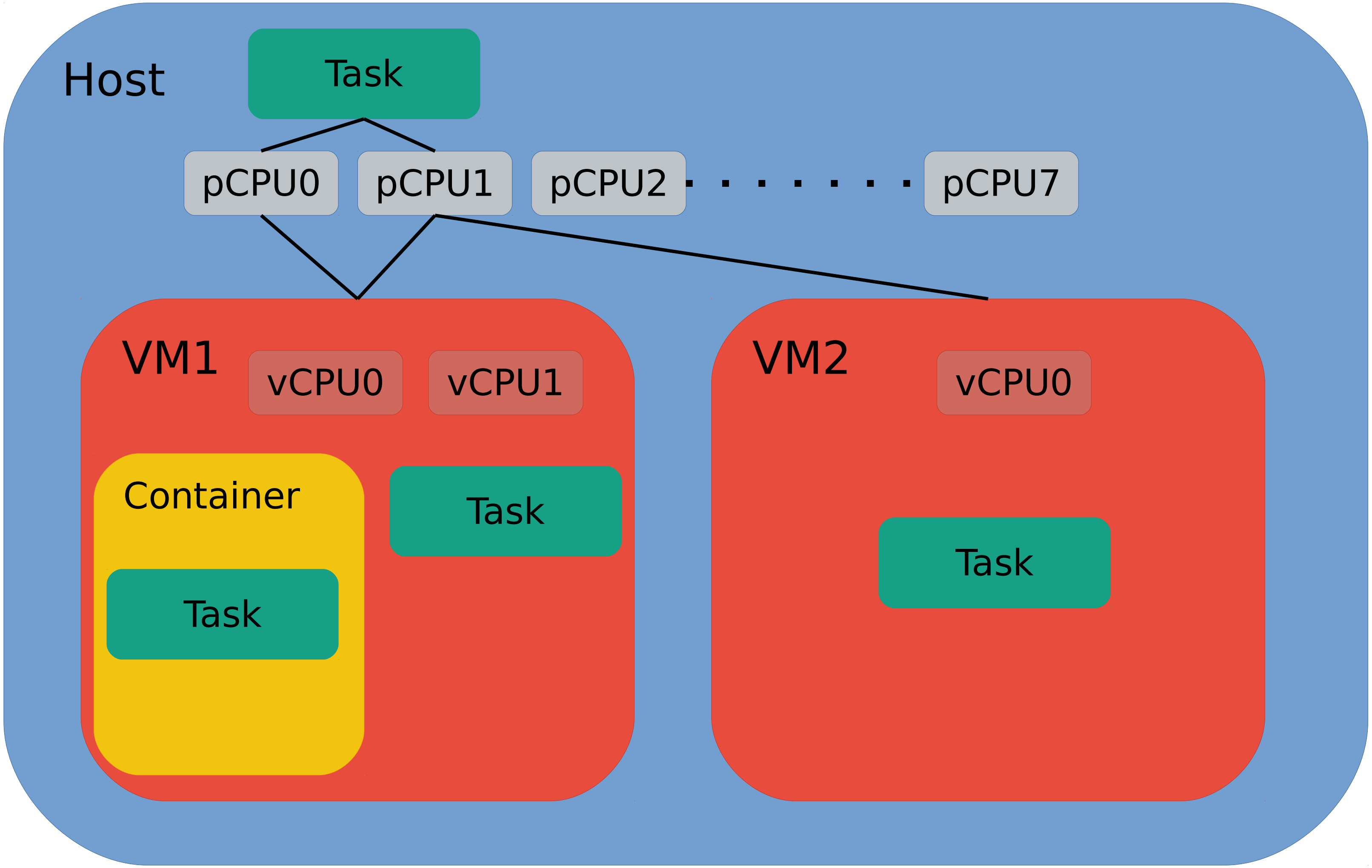
Demo:

- 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:



Conclusion

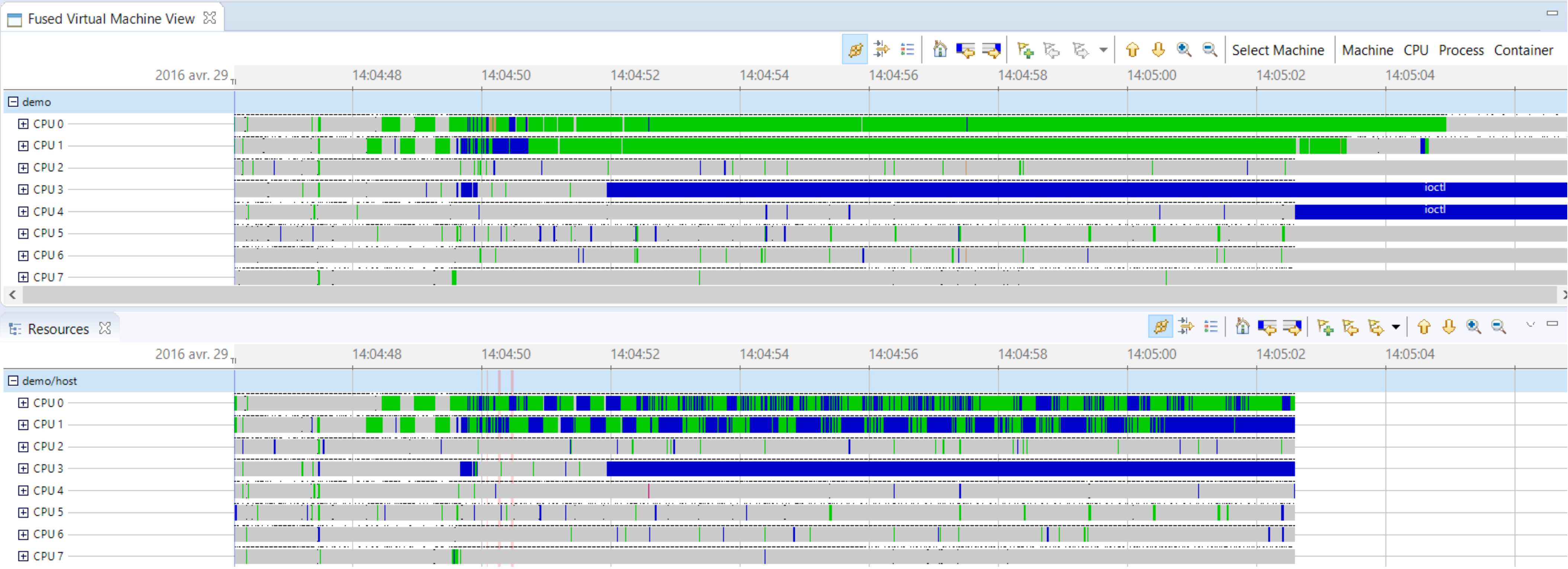
- Multilevel traces aggregated in one level
- Highlight a virtual machine or a container
- Track a vCPU on the host
- Track any thread on the host
- Observe the cause of a preemption

Conclusion

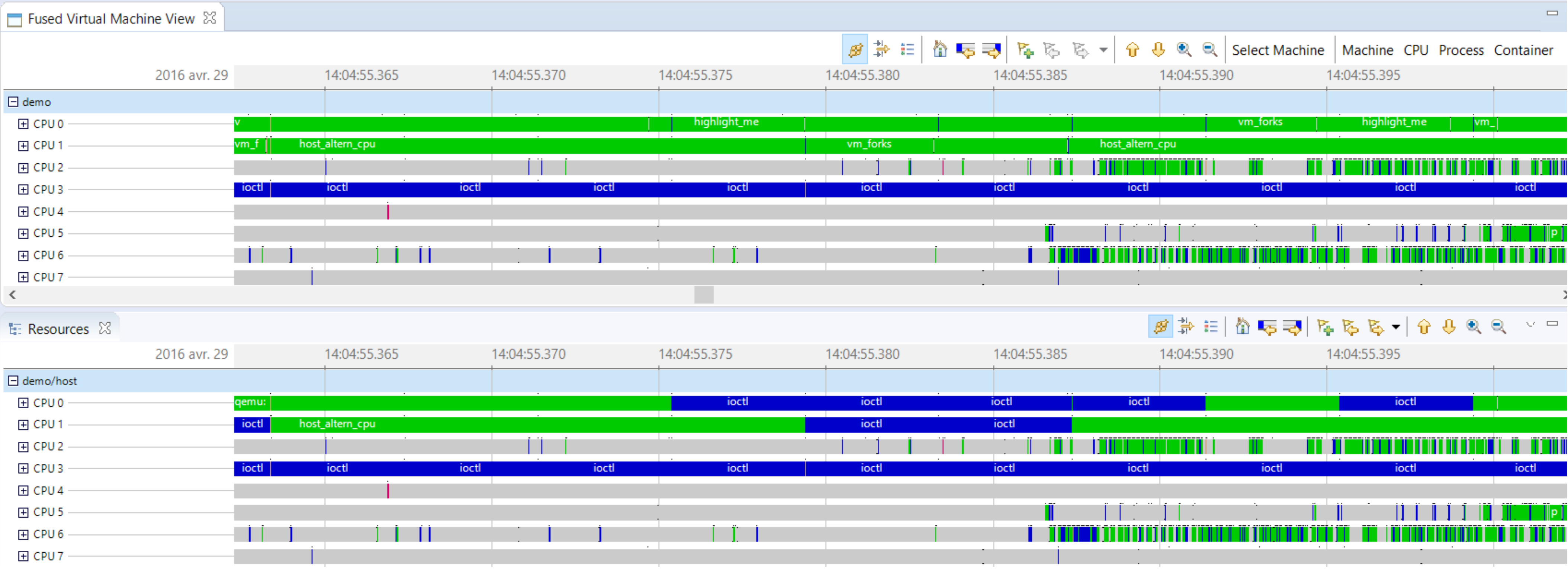
Questions?

cdc.biancheri@gmail.com
https://github.com/cbiancheri/

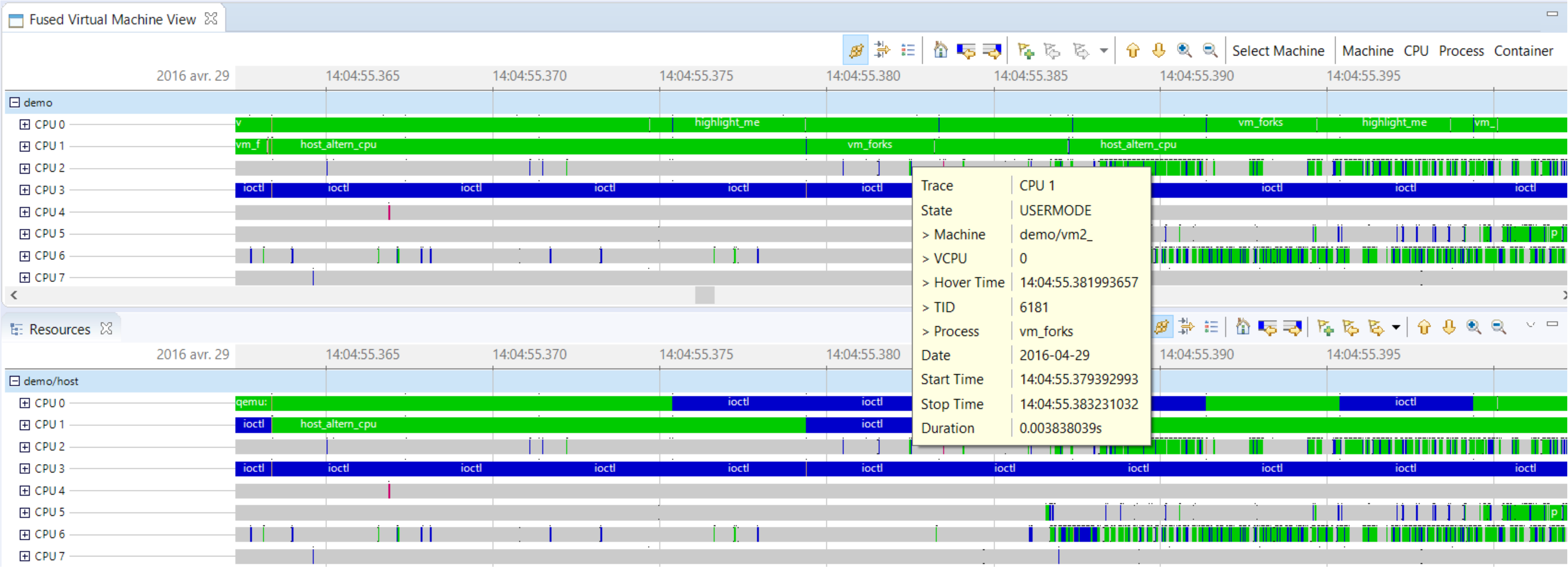
In case of Murphy's law



In case of Murphy's law

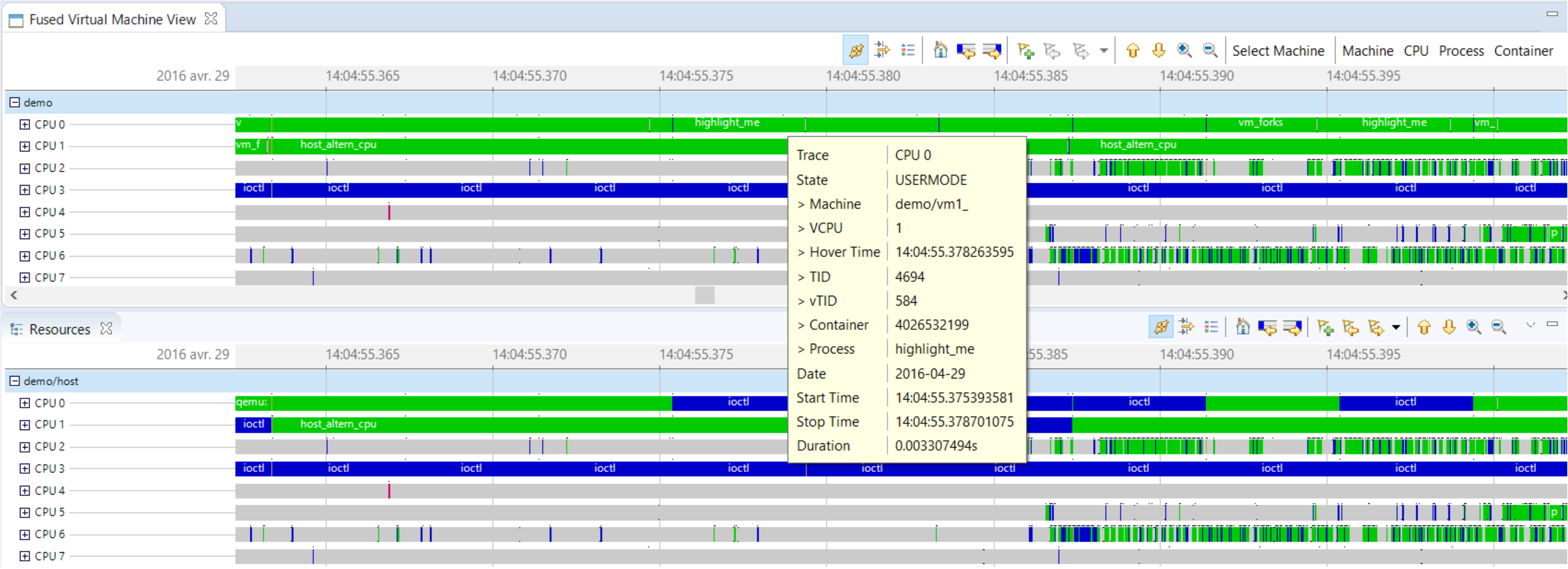


In case of Murphy's law



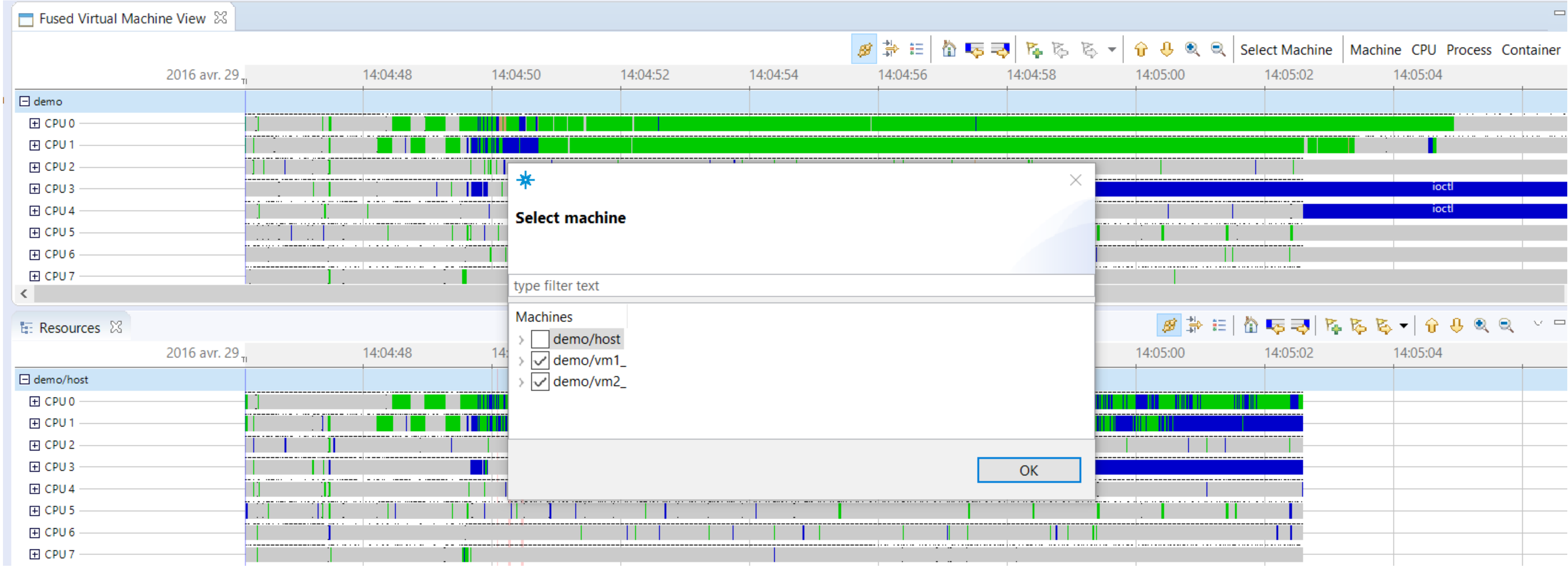
Trace	CPU 1
State	USERMODE
> Machine	demo/vm2_
> VCPU	0
> Hover Time	14:04:55.381993657
> TID	6181
> Process	vm_forks
Date	2016-04-29
Start Time	14:04:55.379392993
Stop Time	14:04:55.383231032
Duration	0.003838039s

In case of Murphy's law



Trace	CPU 0
State	USERMODE
> Machine	demo/vm1_
> VCPU	1
> Hover Time	14:04:55.378263595
> TID	4694
> vTID	584
> Container	4026532199
> Process	highlight_me
Date	2016-04-29
Start Time	14:04:55.375393581
Stop Time	14:04:55.378701075
Duration	0.003307494s

In case of Murphy's law



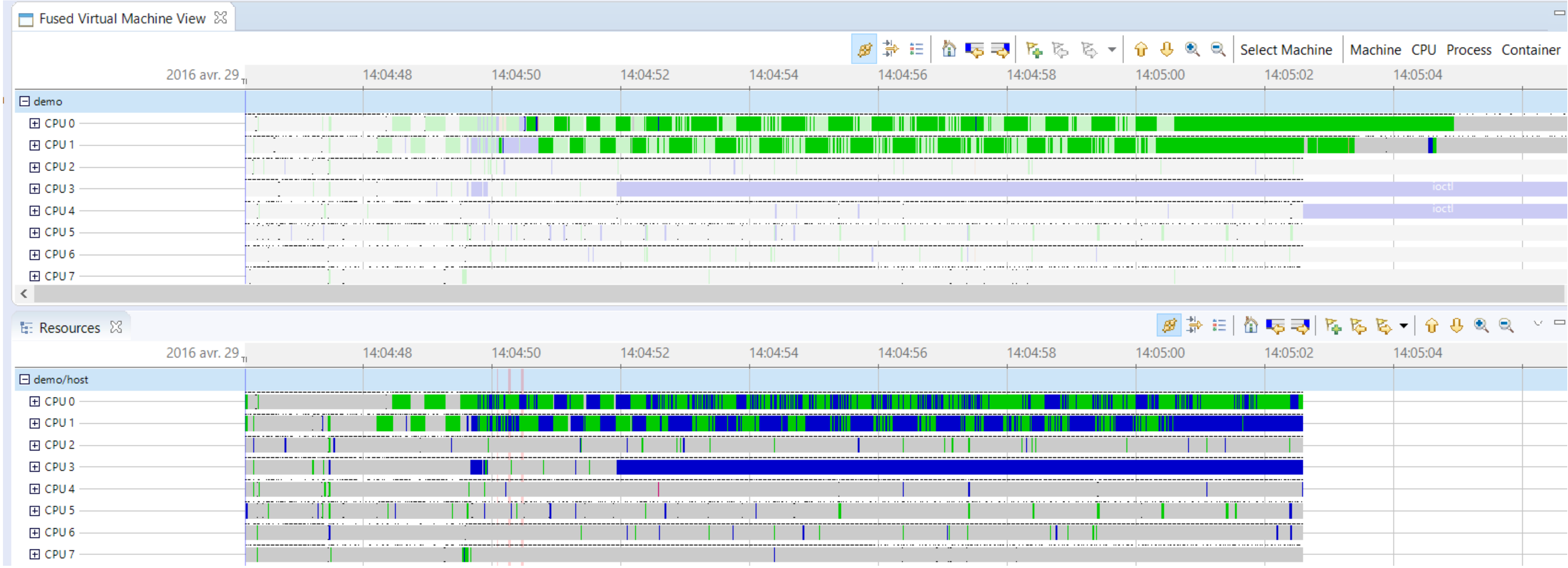
Select machine

type filter text

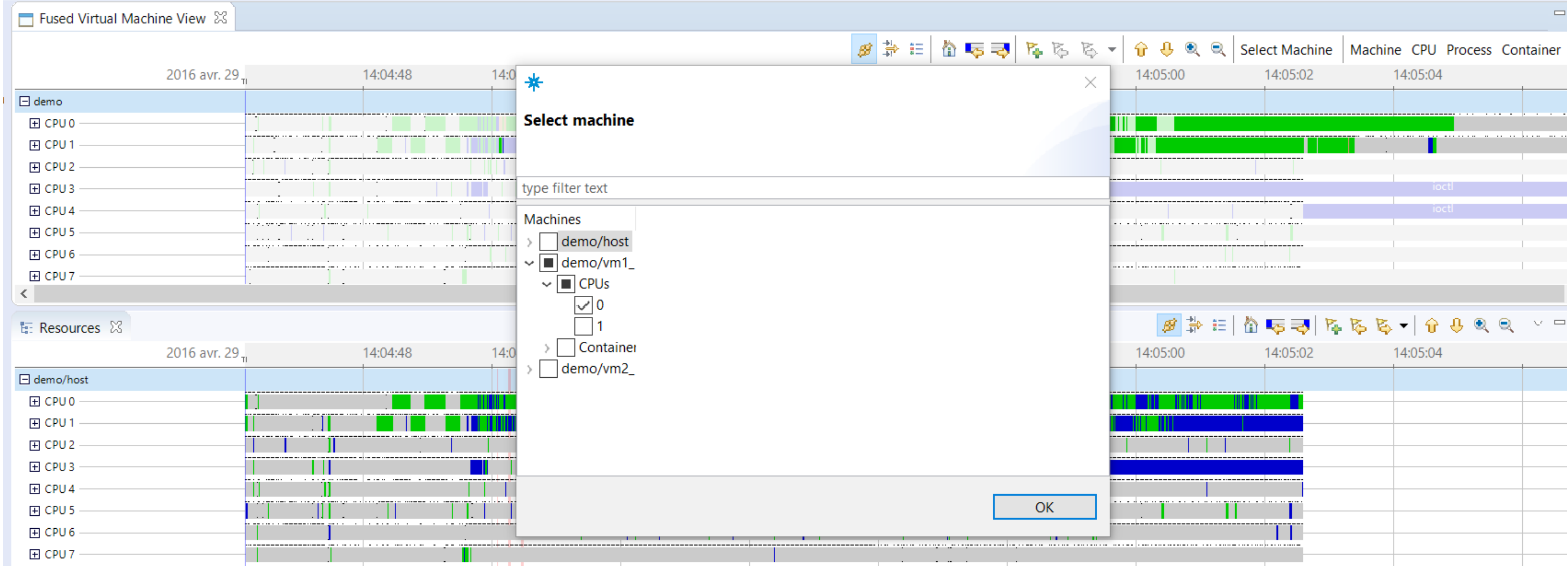
Machines
- □ demo/host
- ☑ demo/vm1_
- ☑ demo/vm2_

OK

In case of Murphy's law



In case of Murphy's law



Select machine

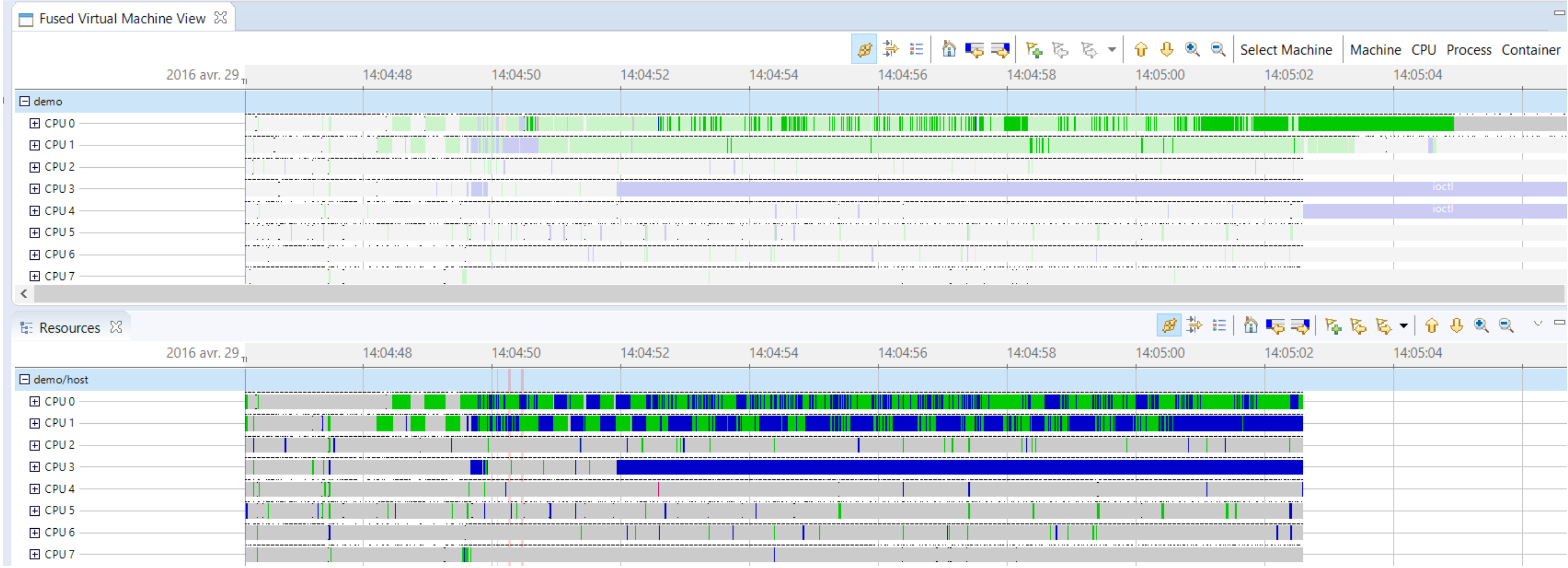
type filter text

Machines

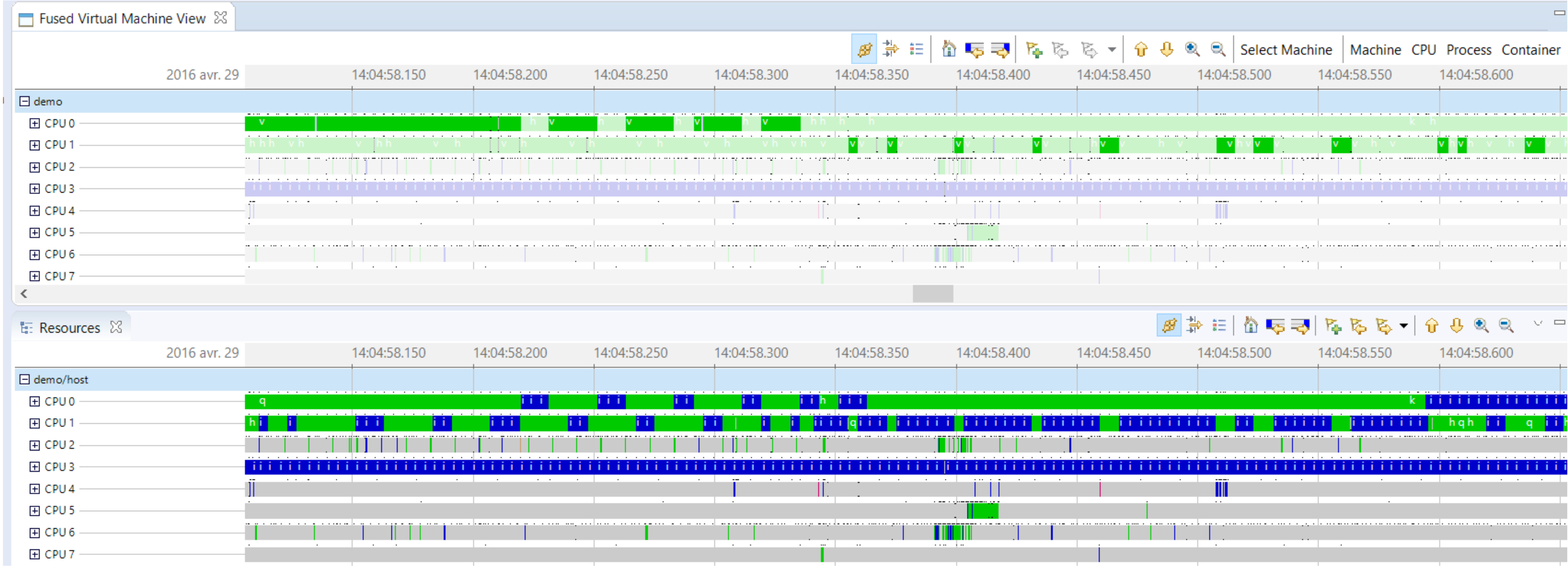
- □ demo/host
- ■ demo/vm1_
  - ■ CPUs
    - ☑ 0
    - □ 1
  - □ Container
- □ demo/vm2_

OK

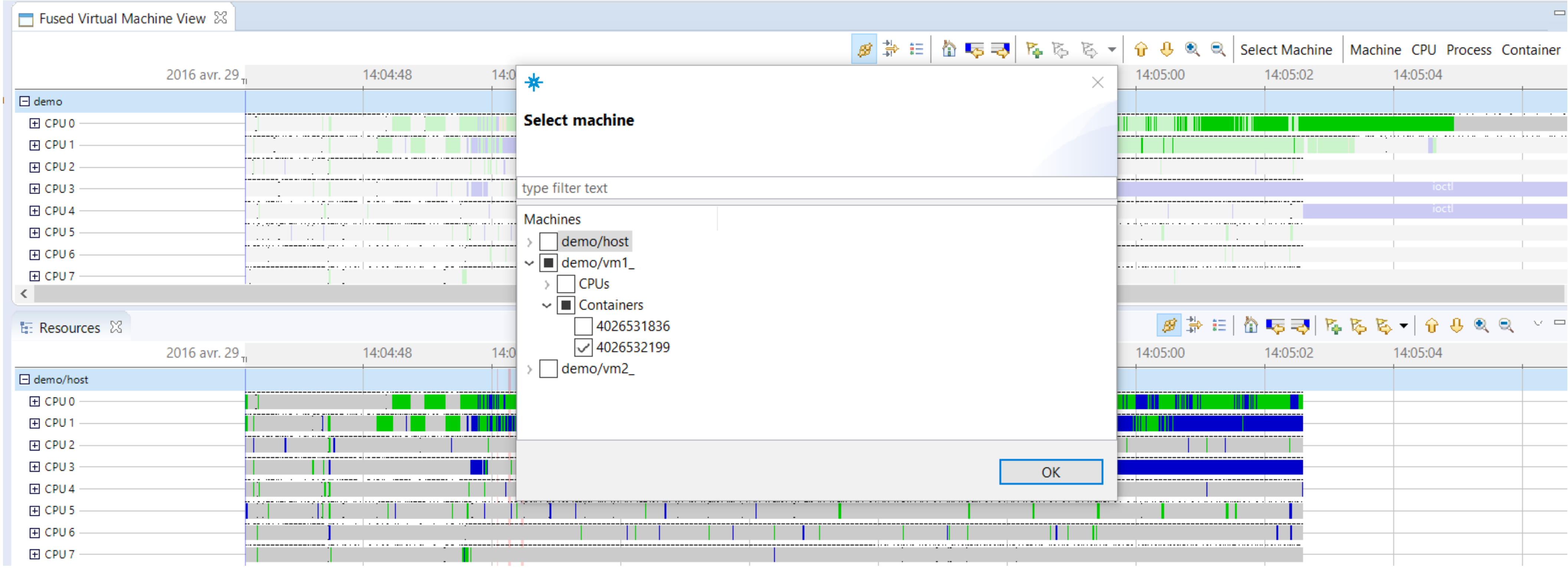
In case of Murphy's law



In case of Murphy's law



In case of Murphy's law



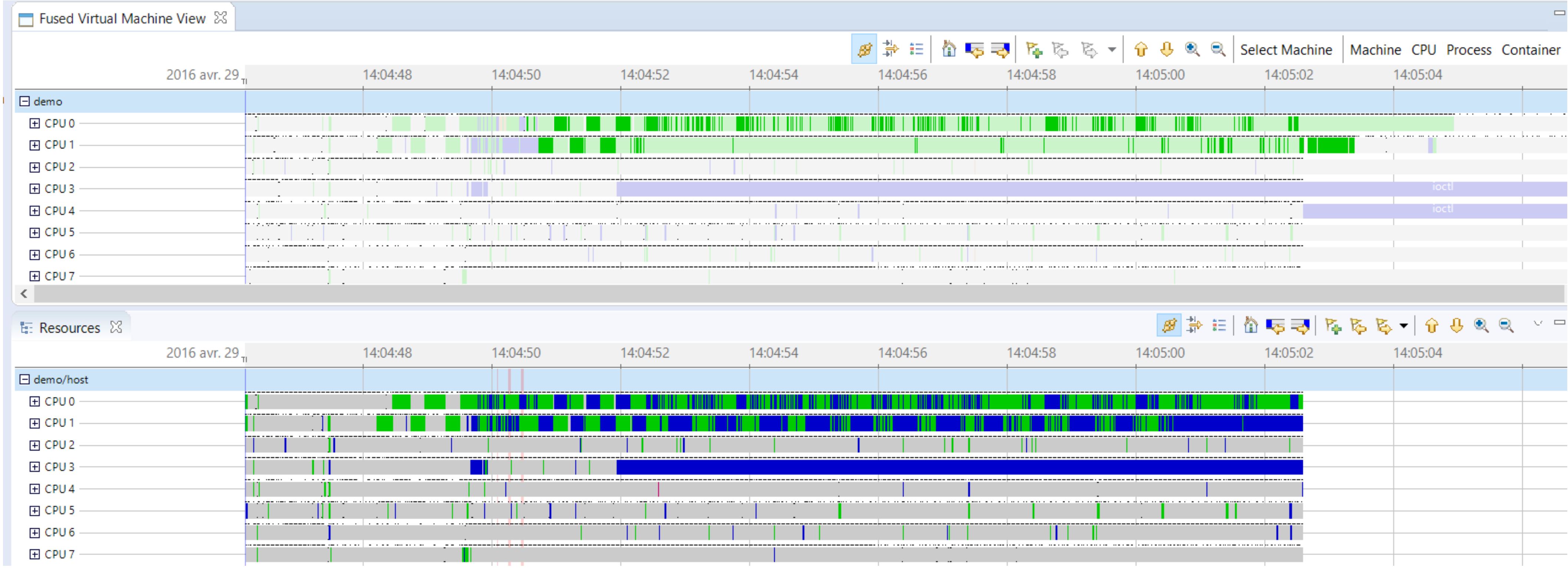
Select machine

type filter text

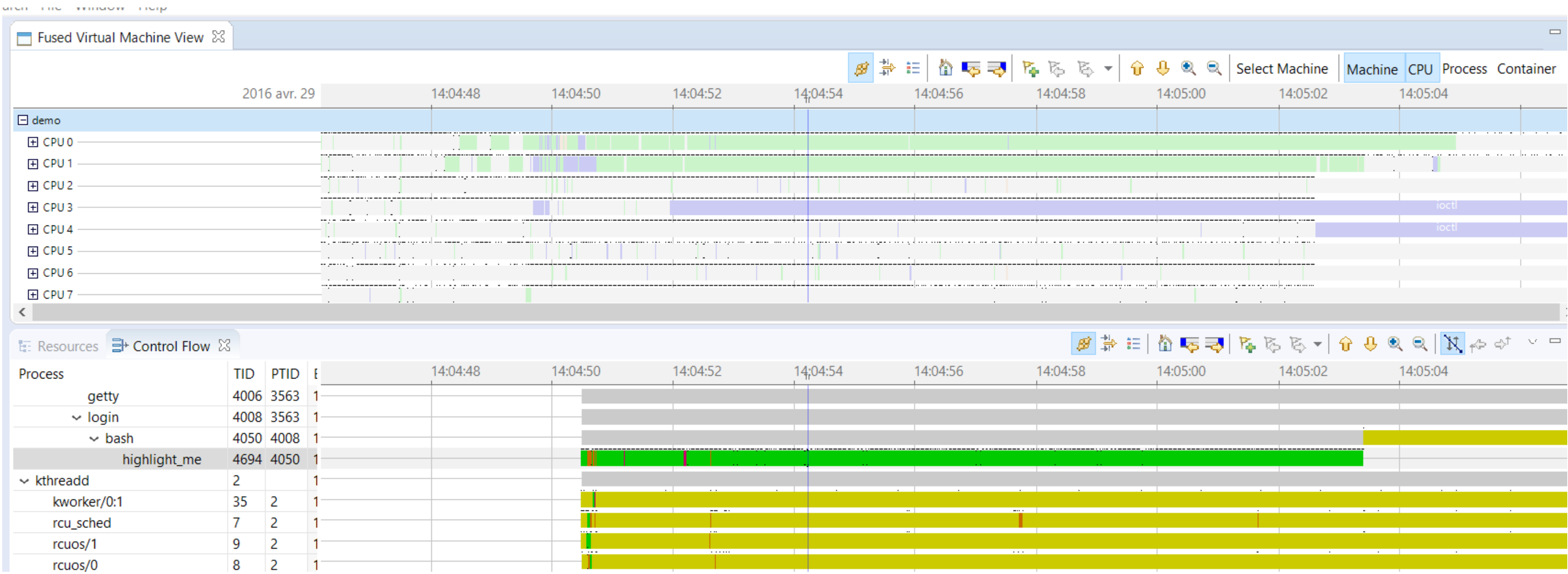
Machines
- › □ demo/host
- ⌄ ■ demo/vm1_
  - › □ CPUs
  - ⌄ ■ Containers
    - □ 4026531836
    - ☑ 4026532199
- › □ demo/vm2_

OK

In case of Murphy's law

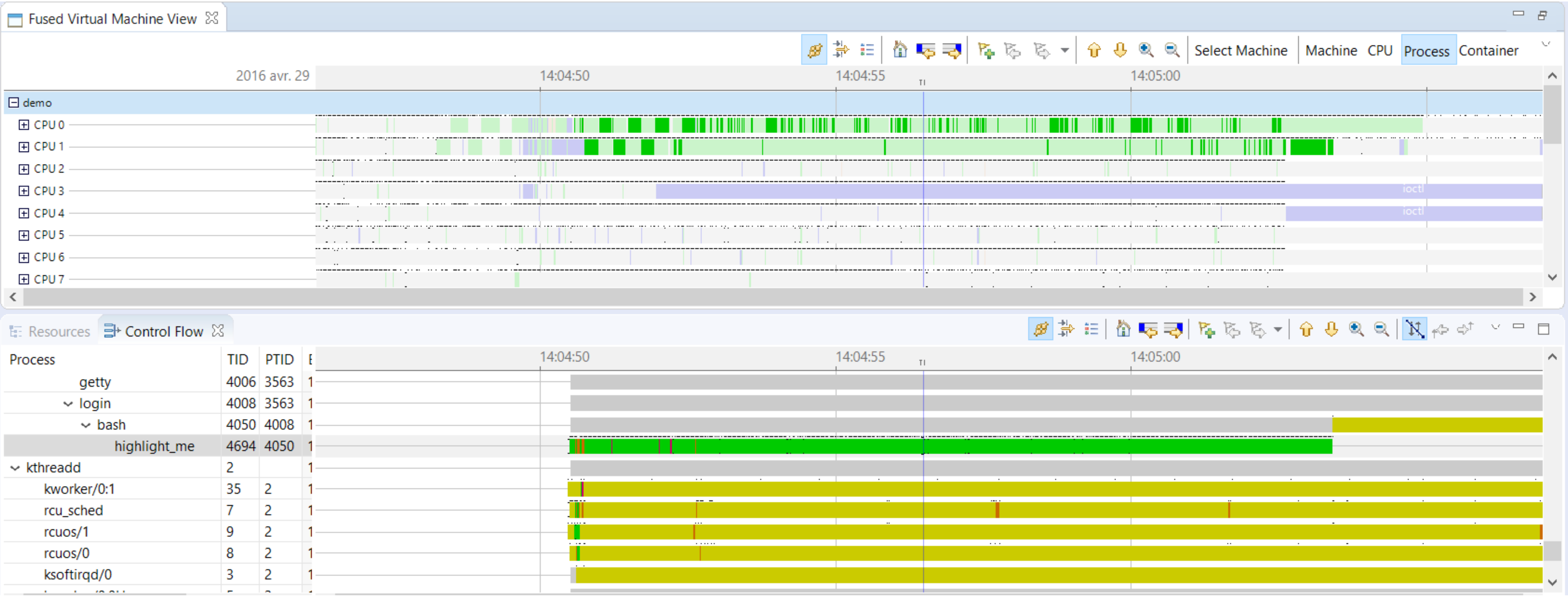


In case of Murphy's law



Process	TID	PTID	E
getty	4006	3563	1
login	4008	3563	1
bash	4050	4008	1
highlight_me	4694	4050	1
kthreadd	2		1
kworker/0:1	35	2	1
rcu_sched	7	2	1
rcuos/1	9	2	1
rcuos/0	8	2	1

In case of Murphy's law



Process	TID	PTID	E
getty	4006	3563	1
login	4008	3563	1
bash	4050	4008	1
highlight_me	4694	4050	1
kthreadd	2		1
kworker/0:1	35	2	1
rcu_sched	7	2	1
rcuos/1	9	2	1
rcuos/0	8	2	1
ksoftirqd/0	3	2	1