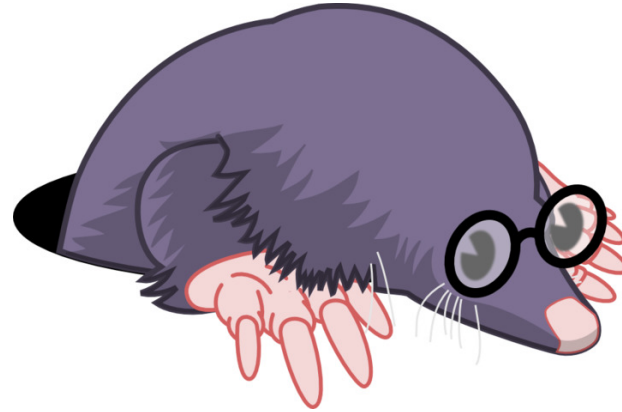
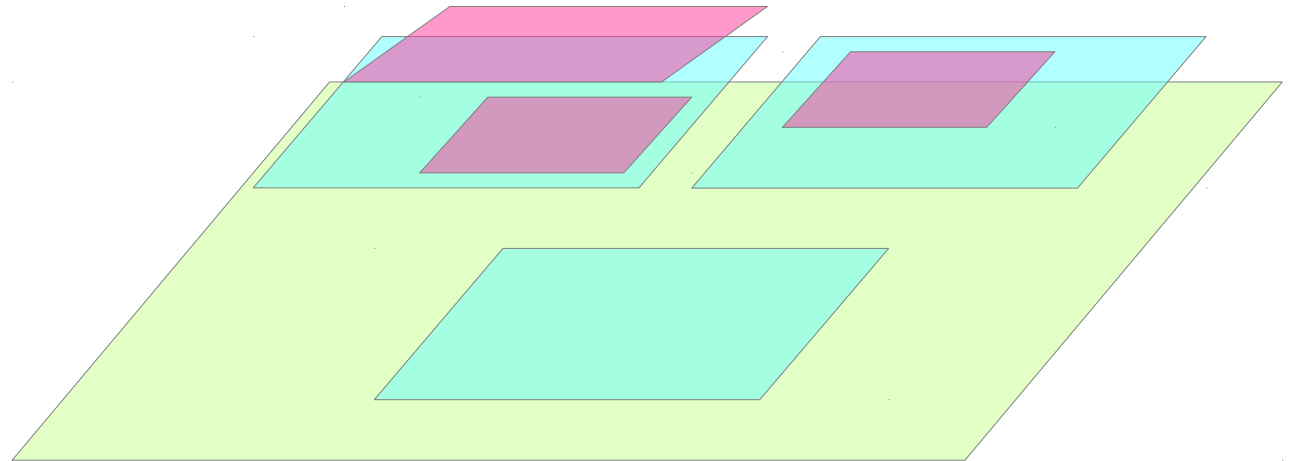


Virtual Machine Tracing for Everyone



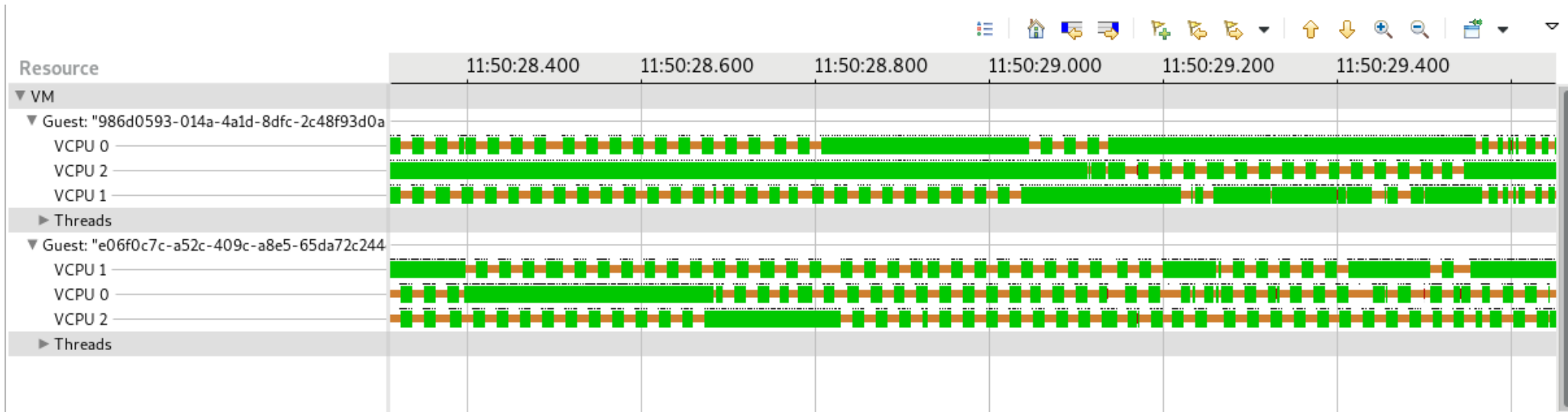
Progress Report Meeting,
École Polytechnique de Montréal
May 10, 2018

Geneviève Bastien
Research Associate



History of VM Tracing @Dorsal

2014: Analyze execution and preemption of virtual machines - Mohamad Gebai



History of VM Tracing @Dorsal

2016: Multilevel execution of VM, utilization of physical resources - Cédric Biancheri



History of VM Tracing @Dorsal

2017: Tracing Open Stack, from Very High to Very Low Level - Yves Junior Bationo



History of VM Tracing @Dorsal

Work in progress:

Host-assisted guest tracing - Abderrahmane BenBachir

Tracing and Analyzing Virtual Machines from Host – Hani Nemati

Tracing and Analyzing Virtual Resources other than CPU – Adel Belkhiri

Tracing and Analyzing Containers – Loïc Gelle

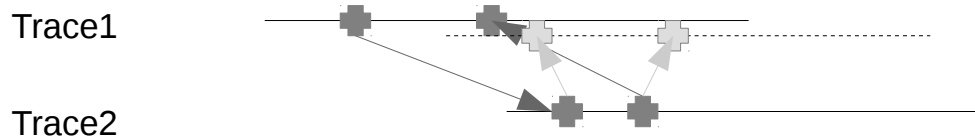


Method

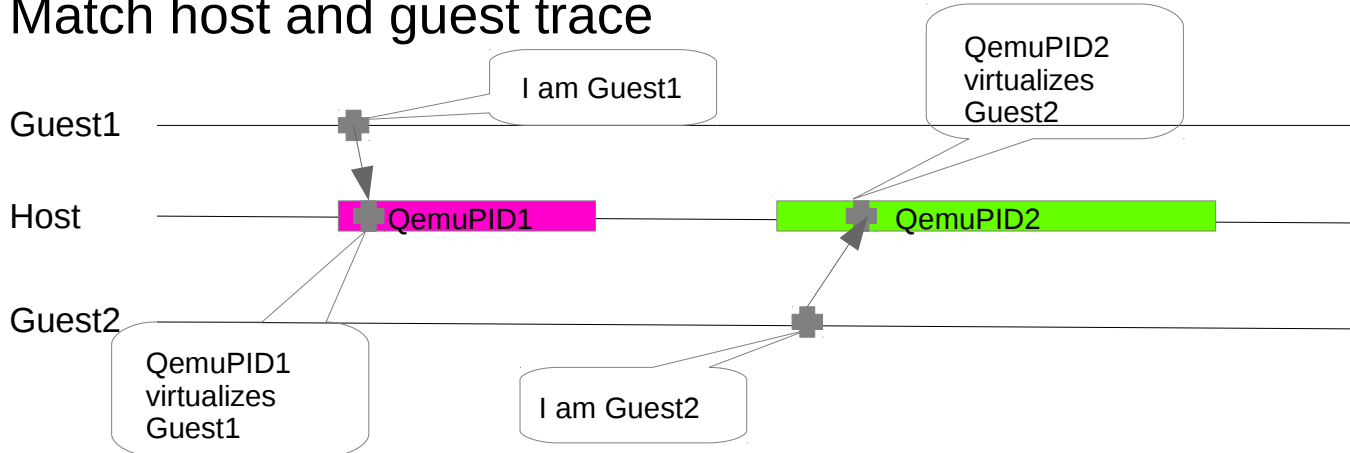
- Addons module in LTTng (by Mohamad and Francis) adds and traces a ping-pong hypercall between guest and host.
 - Hackish, but worked!

Why?

1- Synchronize the traces

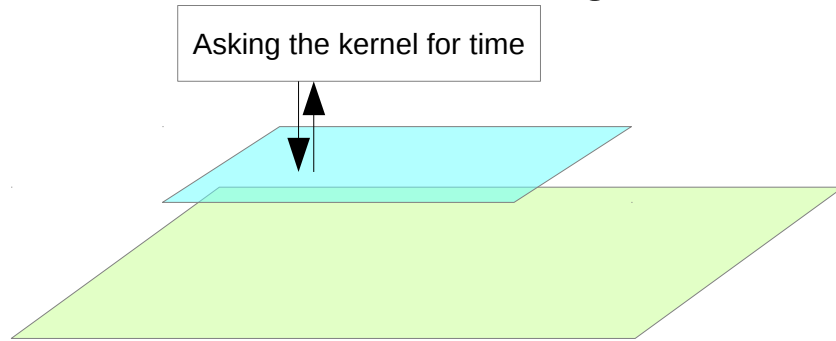


2- Match host and guest trace

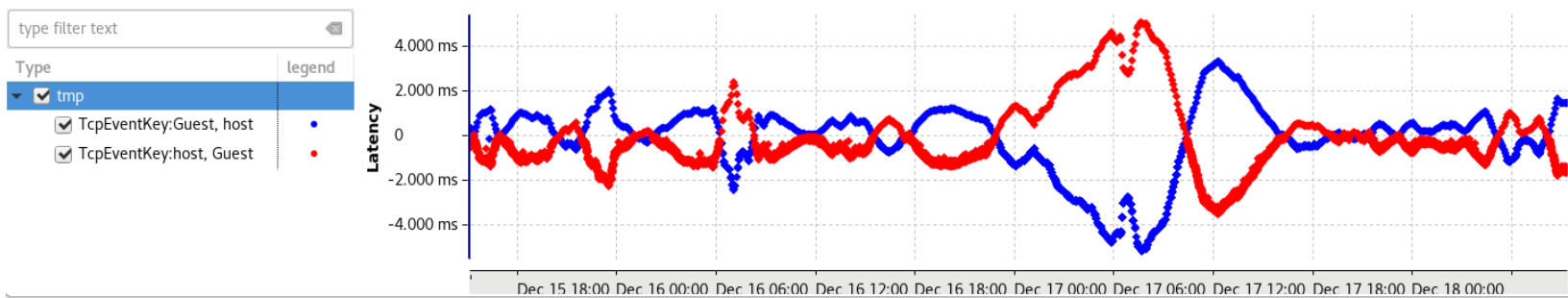


Problem 1: Trace synchronization

- Traces use clock monotonic: guest and host have their own!

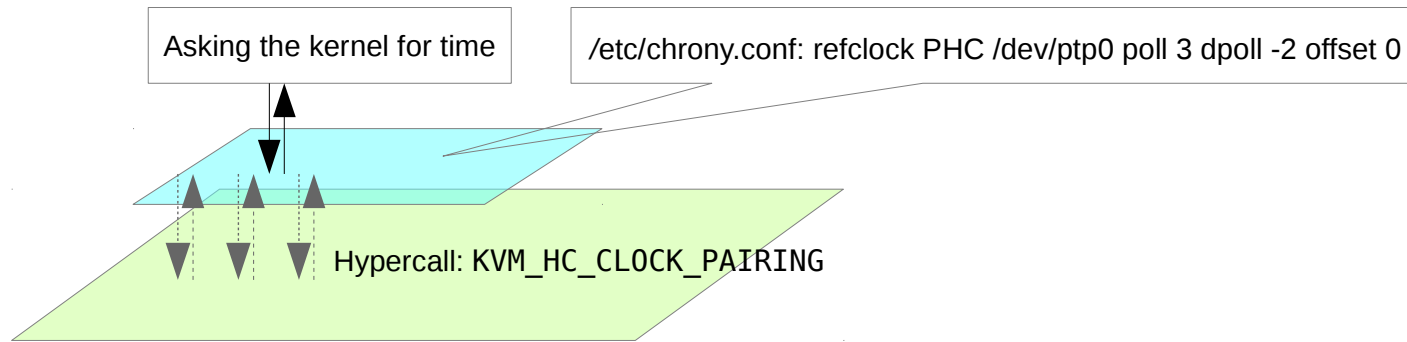


- TCP packets latency: NTP over 4 days. NOT synchronized!!!



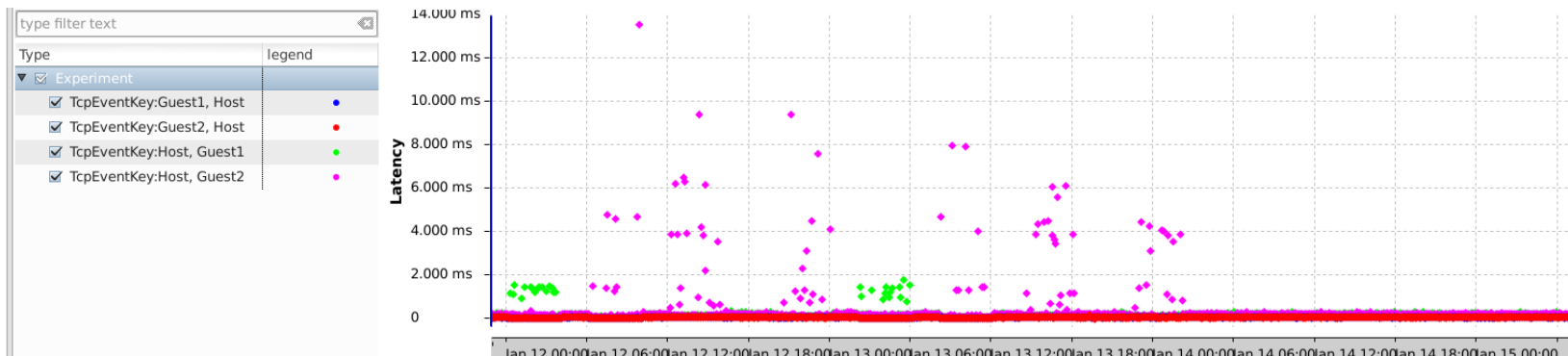
Problem 1: Trace synchronization

- As of kernel 4.11: ptp_kvm kernel module for guests + chronyd



- Over 4 days, the average TCP packet latencies: 200 μ s. Some outliers probably due to VM preemption.

By comparison convex hull gets an average latency of 100 μ s, but only for a few seconds at a time after which it drifts away.



Problem 2: Match guest/host trace

All it takes is 1 small piece of uniquely shared data!!

- Kvm stores data in the debugfs

```
[gbastien@wilbrod qemu]$ sudo ls /sys/kernel/debug/kvm
23461-18      halt_successful_poll  invlpg          max_mmu_page_hash_collisions  mmu_pte_write      pf_fixed        tlb_flush
exits        halt_wakeup          io_exits        mmio_exits                  mmu_recycled       pf_guest
fpu_reload   host_state_reload    irq_exits        mmu_cache_miss              mmu_shadow_zapped  remote_tlb_flush
halt_attempted_poll  hypercalls         irq_injections  mmu_flooded                  mmu_unsync         req_event
halt_exits    insn_emulation       irq_window      mmu_pde_zapped              nmi_injections     request_irq
halt_poll_invalid  insn_emulation_fail  largepages      mmu_pte_updated             nmi_window         signal_exits
[gbastien@wilbrod qemu]$ cat /proc/23461/cmdline
qemu-system-x86_64-enable-kvm-netnic-netbridge,br=qemubr0-drivefile=qemu1.cow,format=qcow2-m1024-smp2-uuid045a100-a110-1234-beef-c07fefe12345
[gbastien@wilbrod qemu]$
```

- What is this uuid?

```
QEMU
Machine View
lvirtual@gemultng0 ~]$ sudo dmidecode -s system-uuid
0045A100-A110-1234-BEEF-C07FEFE12345
lvirtual@gemultng0 ~]$
```

- Solution: Statedump the KVM process + UUID and add system UUID to trace's metadata!



Problem 2: Match guest/host trace

What about guests started after trace start

- KVM events in UDEV

```
[gbastien@wilbrod qemu]$ udevadm monitor --property
monitor will print the received events for:
UDEV - the event which udev sends out after rule processing
KERNEL - the kernel uevent

UDEV [272750.636607] change /devices/virtual/misc/kvm (misc)
ACTION=change
COUNT=2
CREATED=24
DEVNAME=/dev/kvm
DEVPATH=/devices/virtual/misc/kvm
EVENT=create
MAJOR=10
MINOR=232
PID=28261
SEQUENCE=4245
STATS_PATH=/kvm/28261-18
SUBSYSTEM=misc
USEC_INITIALIZED=272750633629

UDEV [272758.978251] change /devices/virtual/misc/kvm (misc)
ACTION=change
COUNT=1
CREATED=24
DEVNAME=/dev/kvm
DEVPATH=/devices/virtual/misc/kvm
EVENT=destroy
MAJOR=10
MINOR=232
PID=28261
SEQUENCE=4252
STATS_PATH=/kvm/28261-18
SUBSYSTEM=misc
USEC_INITIALIZED=272758976021
```

- Solution: Monitor UDEV for KVM creation/destruction and get the UUID from the process's command line

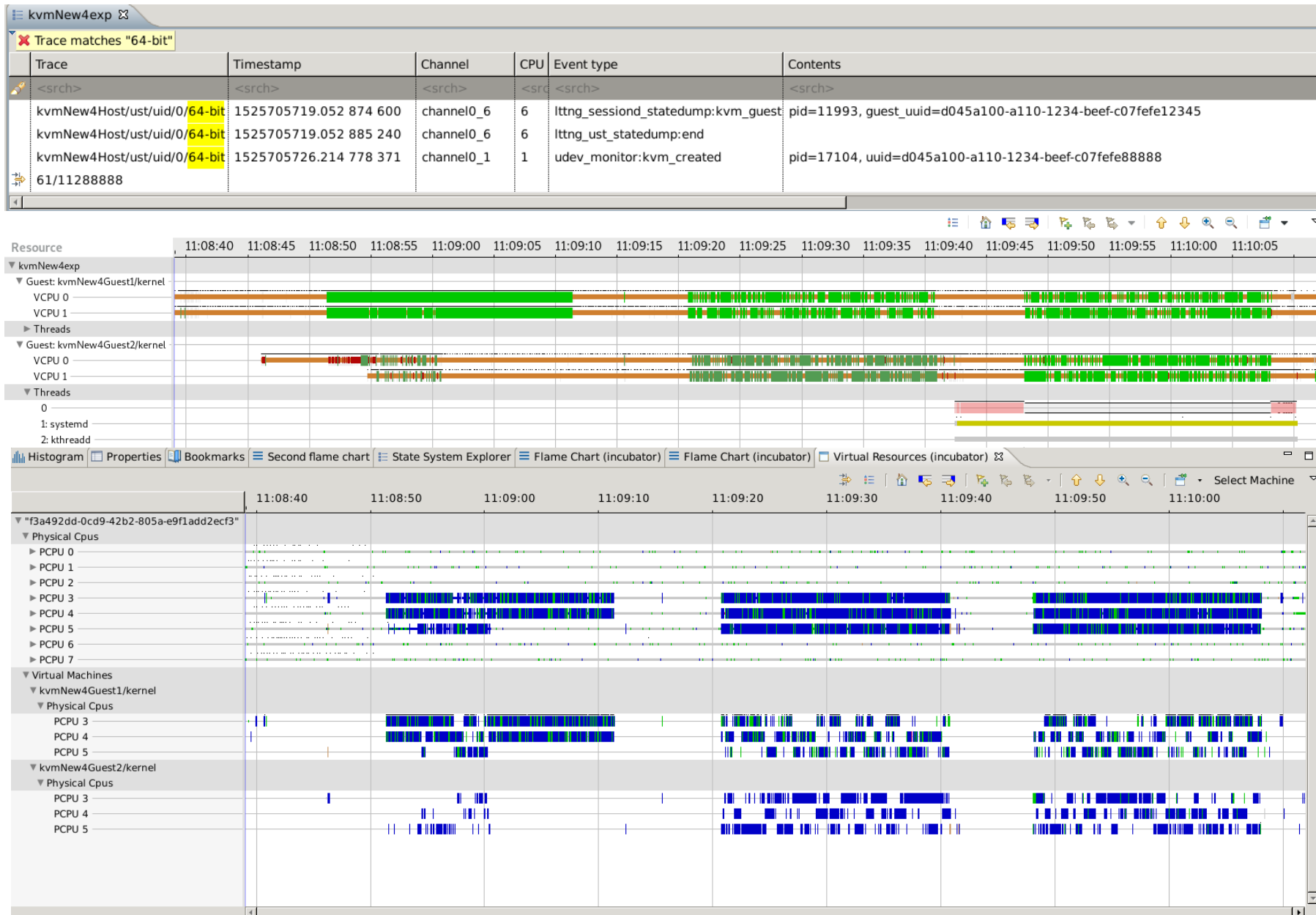


Summary of the new approach

- Use the `ptp_kvm` module + `chronyd` for trace synchronization
- Changes in LTTng:
 - 1- Add the `system-uuid` to the environment properties of the trace (kernel and ust)
 - 2- Statedump the active guests from the `debugfs` information
 - 3- Monitor `udev` for `kvm` guests creation/destruction (adds ~12 – 150 μ s per poll)



Results...



Future work

****Objective** As this research topic approaches maturity, get the word out, have people use it**

- Document! Promote!!! Blog posts, conferences
- Build a realistic cloud!
- Integrate with Hani's host-only approach, use snapshots
- Explore use cases for production environments: help analyze actual problems



Questions ?

Resources

- Sources:
 - <https://github.com/tahini/lttng-tools> branch kvm
 - <https://github.com/tahini/lttng-ust-1> branch statedump_notifier (feature by Mathieu Desnoyers)
 - <https://github.com/tahini/lttng-modules> branch product_uuid
- Blog post on VM synchronization:
<http://versatic.net/tracecompass/synchronization/2018/01/15/synchronization-and-ntp.html>
- Twitter: @genbastien

