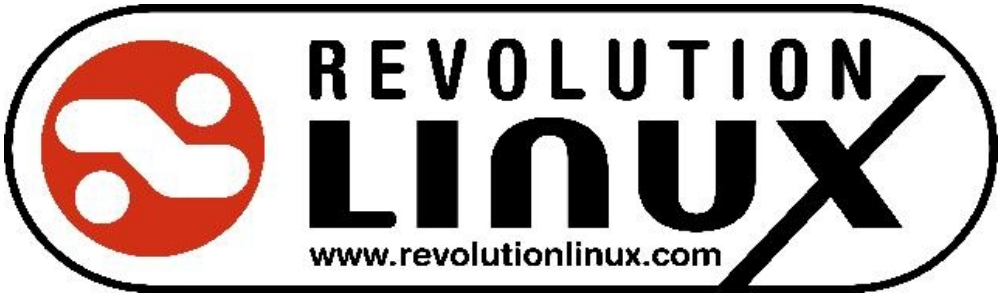
Virtual machines traces synchronization

Julien Desfossez <julien.desfossez@polymtl.ca>
Michel Dagenais <michel.dagenais@polymtl.ca>

December 8, 2010
Mid project meeting, École Polytechnique de Montréal

Content

1. Hypervisor infrastructure
2. Multi-level traces with KVM
3. Existing clock sources
4. A new trace clock
5. Future Work

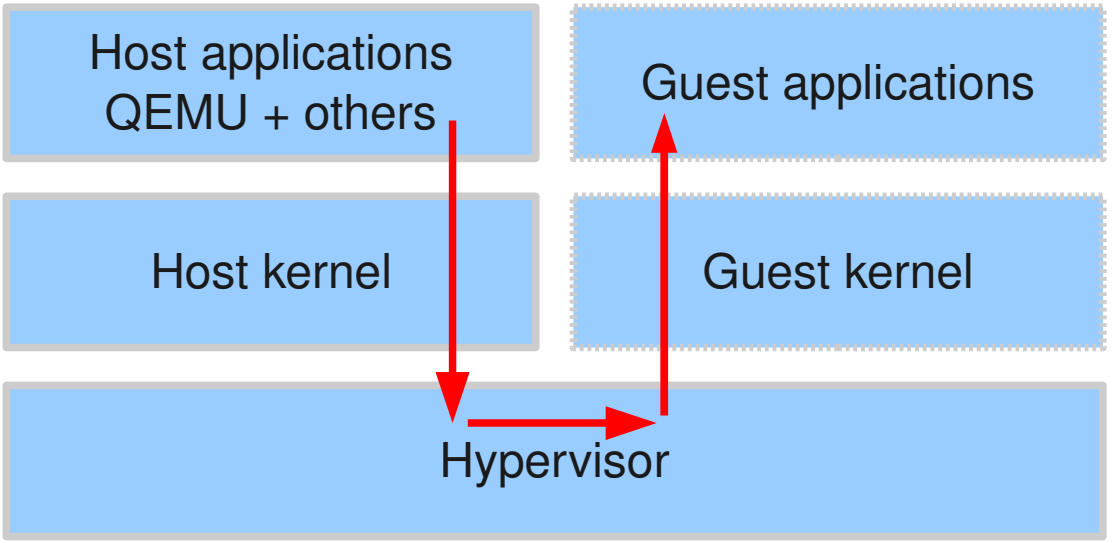


Services and Consulting
Large Scale Infrastructures
Thin Client / Applications Server
All open source software and Linux
Lots of virtualization

Hypervisor infrastructure

- Hardware assisted virtualization (Intel VT-x, AMD-V)
- Multiple virtual machines (VM) on a physical host
- Each VM has its own view of the system time (offset on the TSC)

Hypervisor infrastructure



Multi-level traces with KVM

- Physical host user-space (QEMU)
- Physical host kernel-space
- Virtual machine kernel-space
- Virtual machine user-space applications

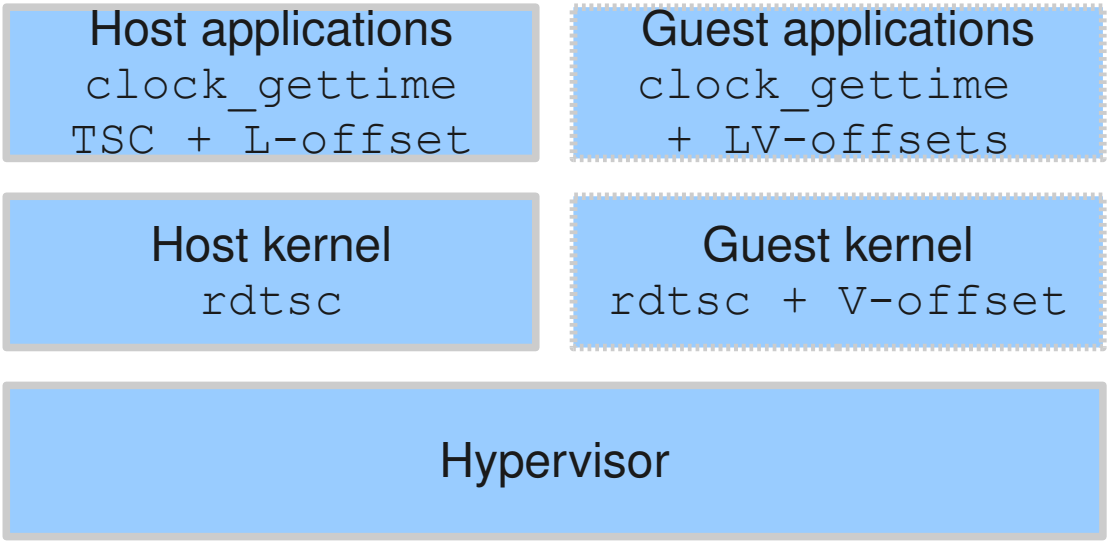
How can we record a consistent

trace across these layers efficiently ?

Existing clock sources

- TSC begins at machine boot time
- Linux time starts at Linux boot (offset on TSC)
- LTTng relies mostly on the TSC (cycles + freq)
- UST relies on clock_gettime vDSO (sec.nsec)
- Offset in the traces kernel/user-space

Time consistency



Efficient TSC based clock source

- Ensures that the TSC is synchronized across all cores
- Export cycles and TSC frequency to user-space (no timespec manipulation)
- clock_gettime vDSO : no system call if possible
- Activate if needed hardware debug clock on first use (ARM)
- Fallback on CLOCK_MONOTONIC in case of desynchronization

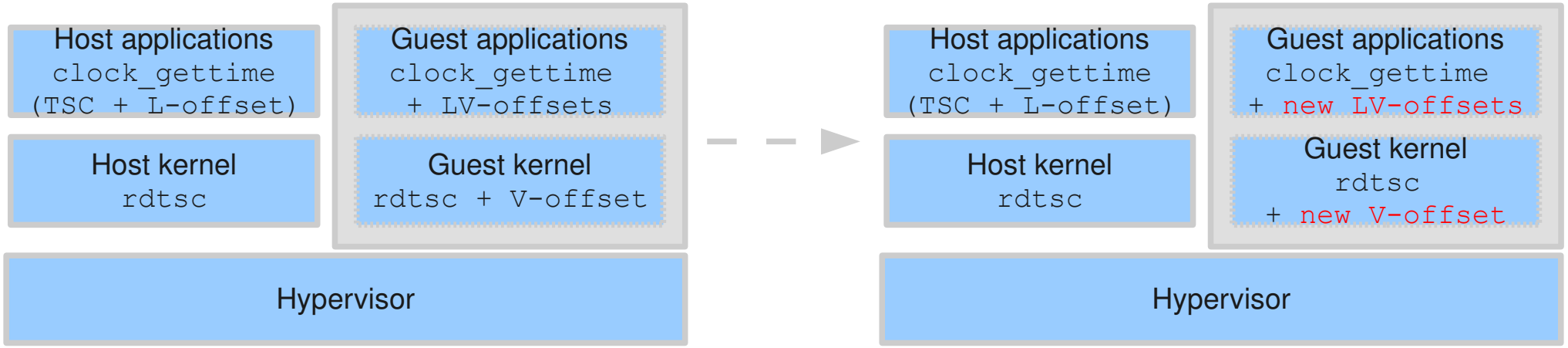
CLOCK_TRACE early results

- On 64 bits (clock_gettime vDSO available)
  - CLOCK_REALTIME : 101 cycles
  - CLOCK_MONOTONIC : 104 cycles
  - CLOCK_TRACE : 52 cycles
- On 32 bits (using a syscall)
  - CLOCK_REALTIME : 649 cycles
  - CLOCK_MONOTONIC : 661 cycles
  - CLOCK_TRACE : 616 cycles

Multi-level trace clock

- Export the TSC offset of each guest in the host trace
- Handle changing V-Offset in the trace analyzer (VM pause and migrations)
- Combine with the distributed traces synchronization algorithm to visualize migration

Future work : Migration



Conclusion

- CLOCK_TRACE : consistent and efficient clock source synchronized across
  - Cores
  - Host kernel/user space
  - Guest(s) kernel/user space