Tencent Cloud

Increase KVM Performance/Density via performance tuning

CLK 2019

Wanpeng Li

wanpengli@tencent.com

About Myself

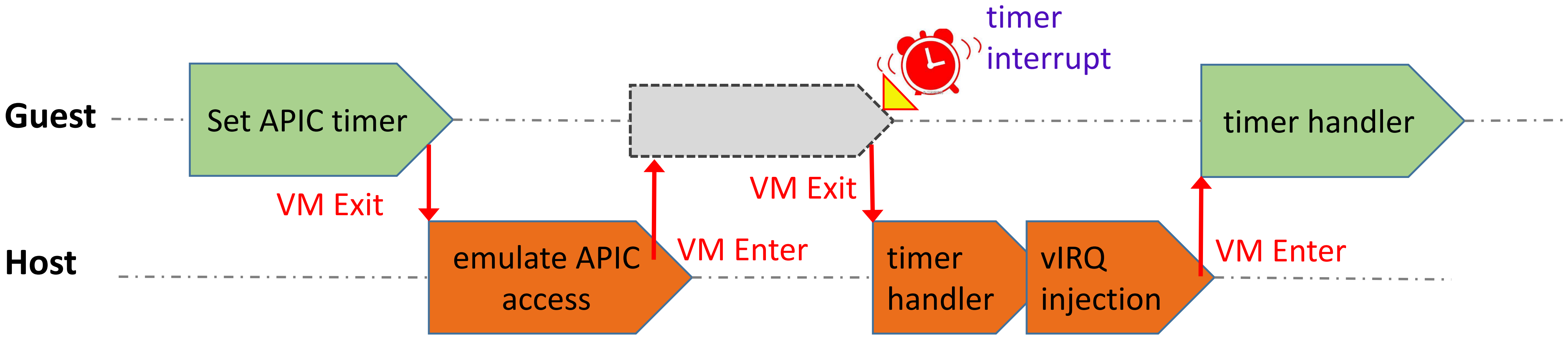
- Have name in Kernel MAINTAINERS file
- Contributing to Kernel Community since 2012
- Push cloud providers in global top contributing companies to KVM

Agenda

- Exitless Timer
- KVM_HINTS_DEDICATED performance hint
- Boost preempted vCPU
- Paravirtualized TLB shootdown

Exitless Timer

- Motivation
  - both arm timer and timer fire incur vmexits
  - dedicated instance encounter performance jitter



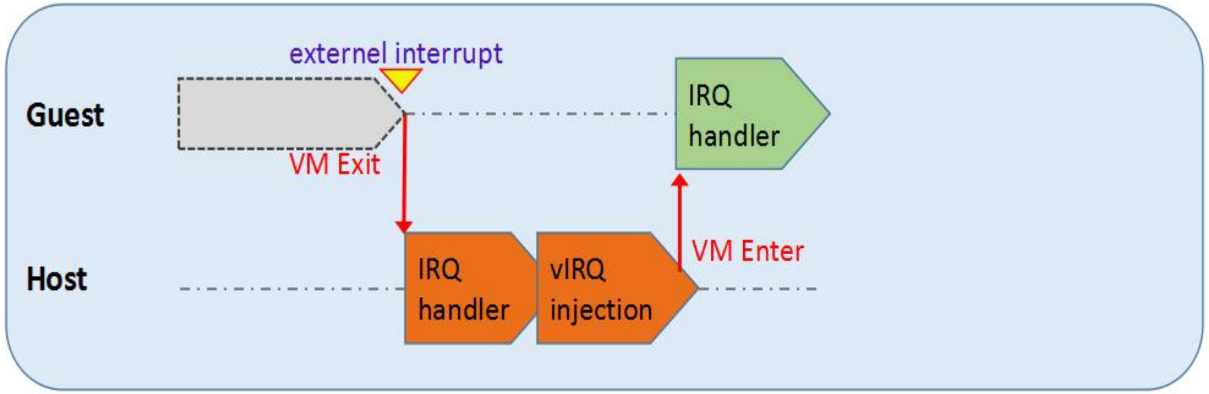
Exitless Timer

- Injection exitless
  - offload lapic timer to the housekeeping cpus
  - inject expired timer interrupt via posted interrupt
  - fine tuned host via enable nohz_full, disable mwait/pause/hlt vmexits etc

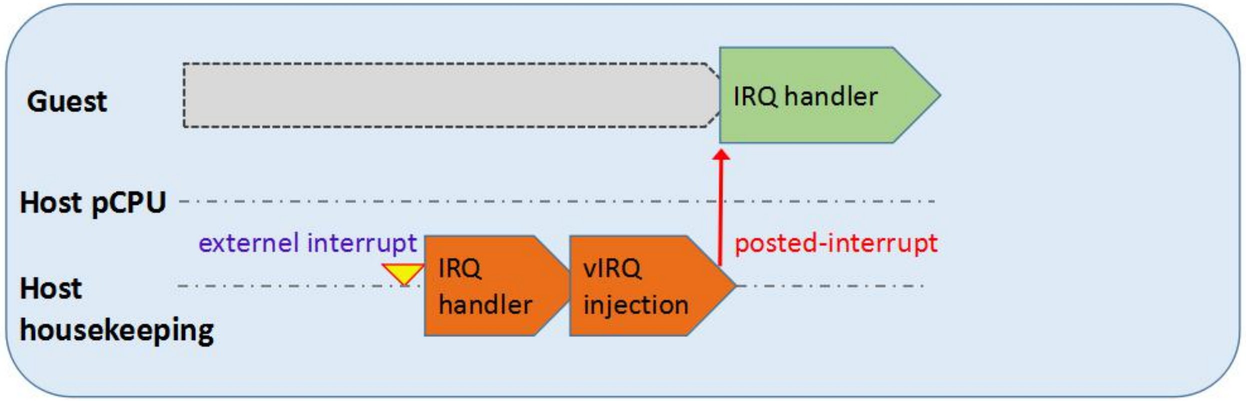
Tencent Cloud

Exitless Timer

- Normal KVM interrupt delivery

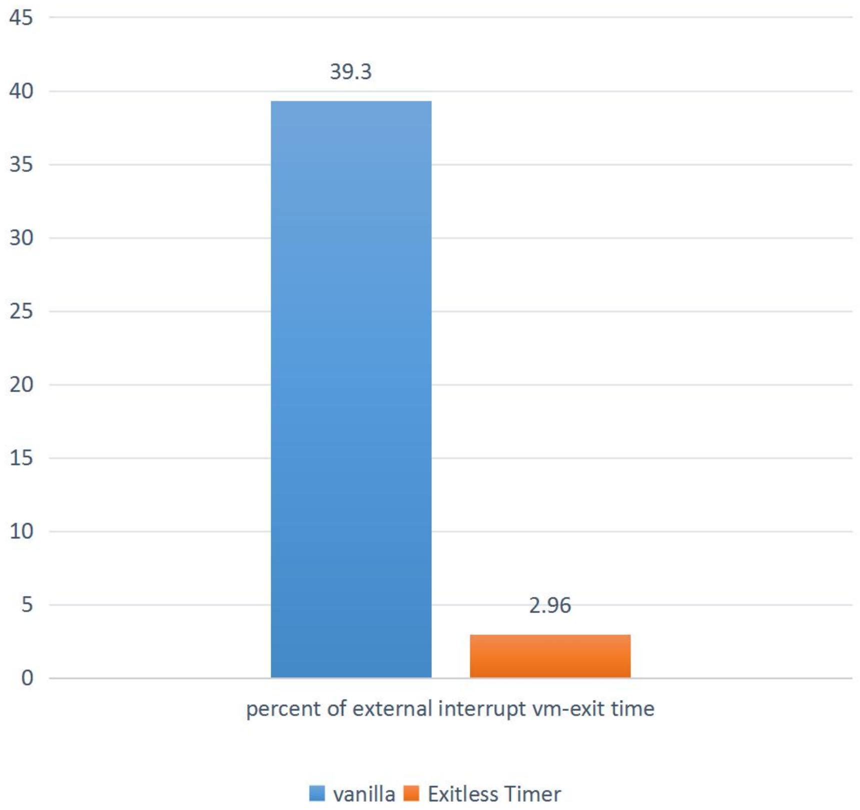


- Housekeeping cpus delivery interrupt via posted-interrupt



Exitless Timer

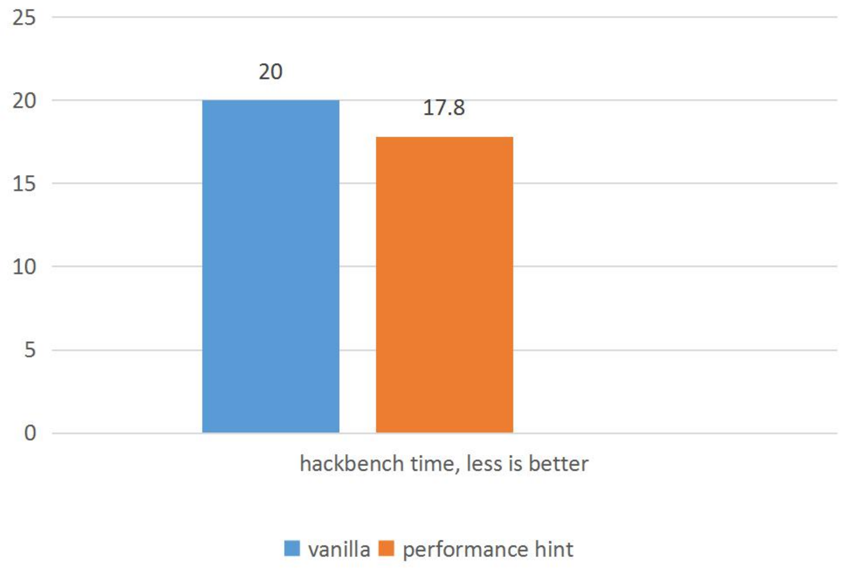
- Performance data



Tencent Cloud

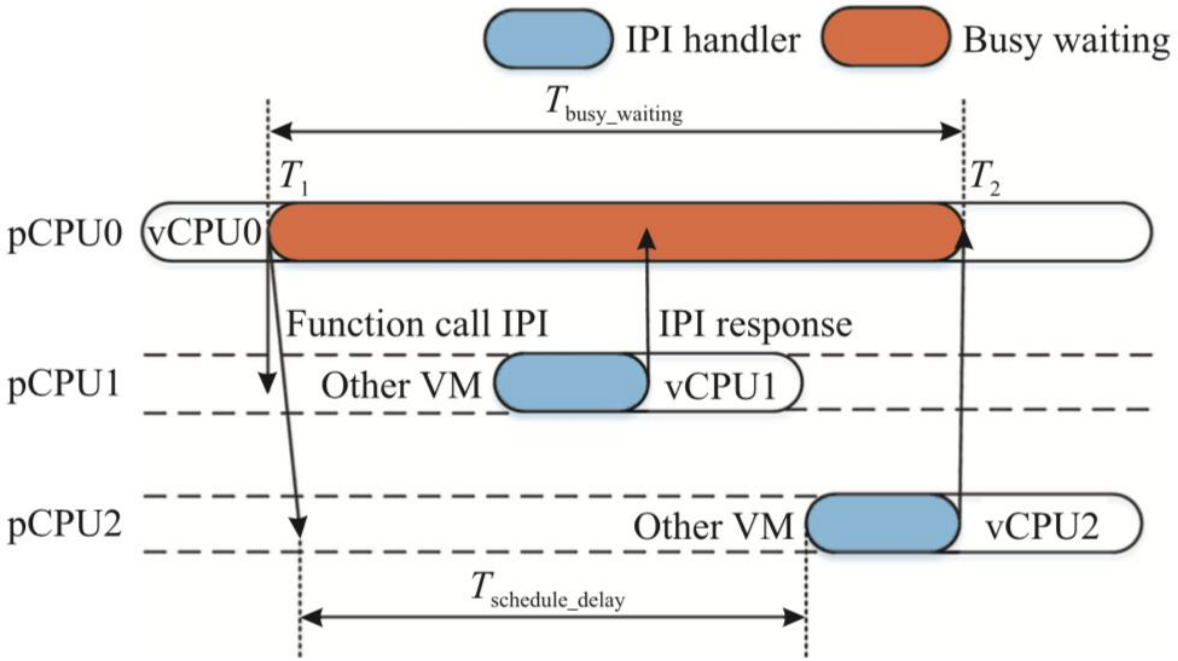
KVM_HINTS_DEDICATED performance hint

- Allows a guest to enable optimizations when running on dedicated pCPUs
  - choose qspinlock
  - native tlb shootdown
  - disable pv sched yield
  - enable guest halt-polling



Boost preempted vCPU

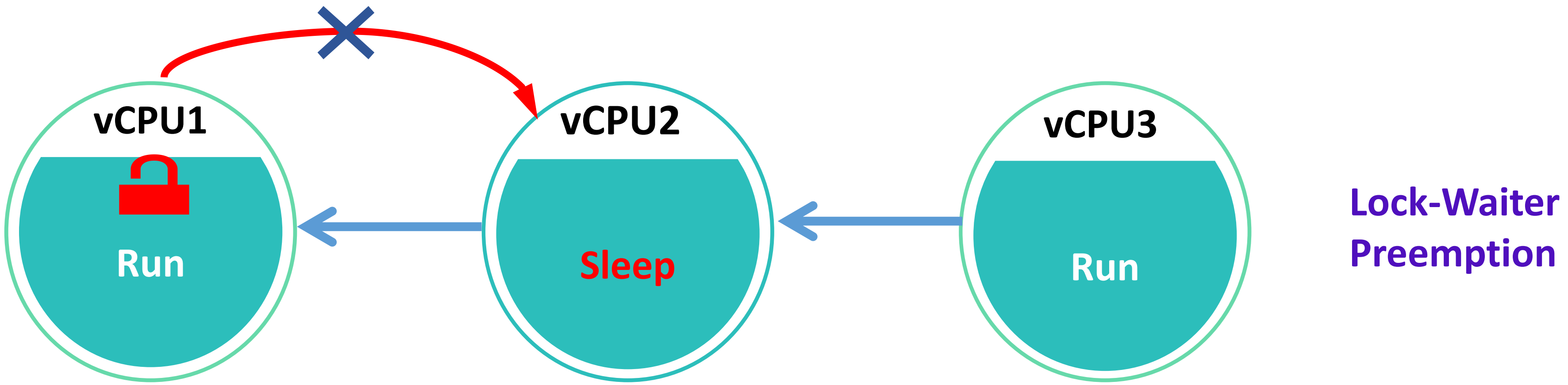
- Boost vCPUs that are ready to deliver interrupts
  - Most smp_call_function_many calls are synchronous, we want to boost not just lock holders but also vCPUs that are delivering interrupts. the IPI target vCPUs are also good yield candidates.



Tencent Cloud

Boost preempted vCPU

- Lock Waiter Preemption
  - Due to the FIFO-ordered spinlock algorithm whenever a hypervisor preeempts the next waiter that has not yet acquired the lock, even if the lock is released, no other thread is allowed to acquire it until the next waiter is allowed to run.



  - The lock holder vCPU yields to the queue head vCPU when unlock, to boost queue head vCPU.

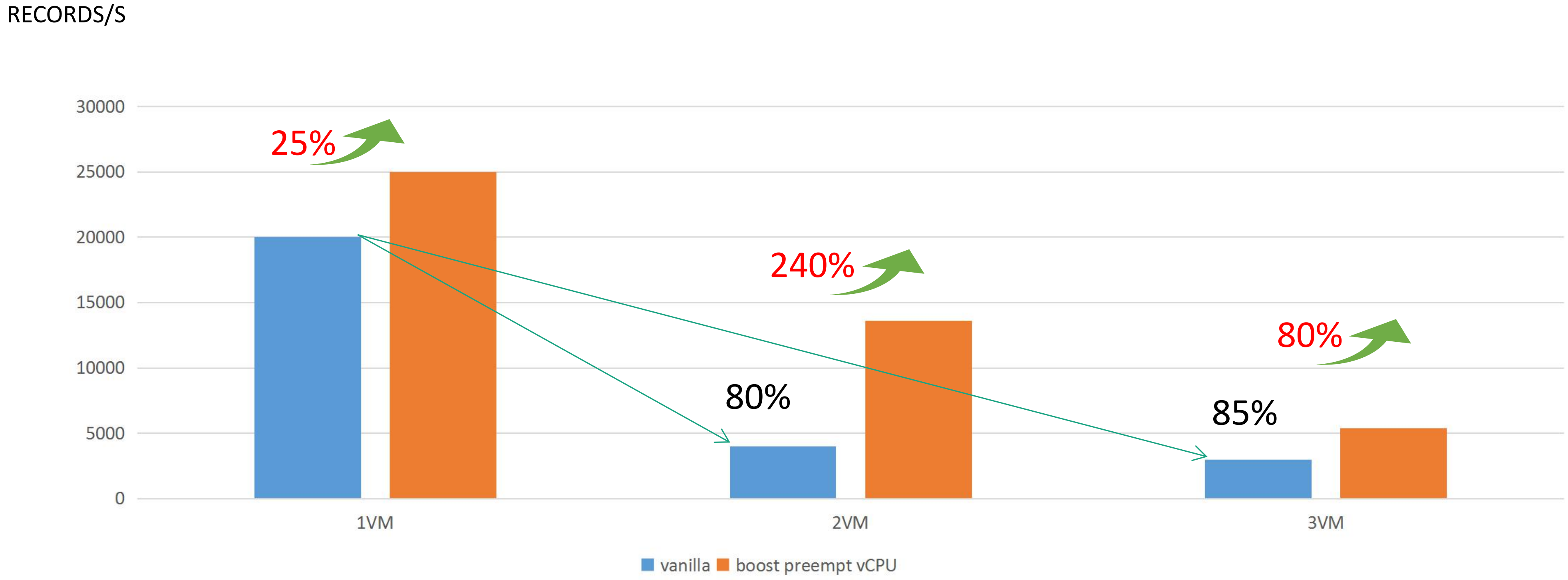
Boost preempted vCPU

- Evaluation Environment:
- Hardware: Xeon Skylake 2 sockets, 40 cores, 80 threads
- VM: each 80 vCPUs
- Test case: ebizzy -M
- Both guest and host are v5.3 kernel

Tencent Cloud

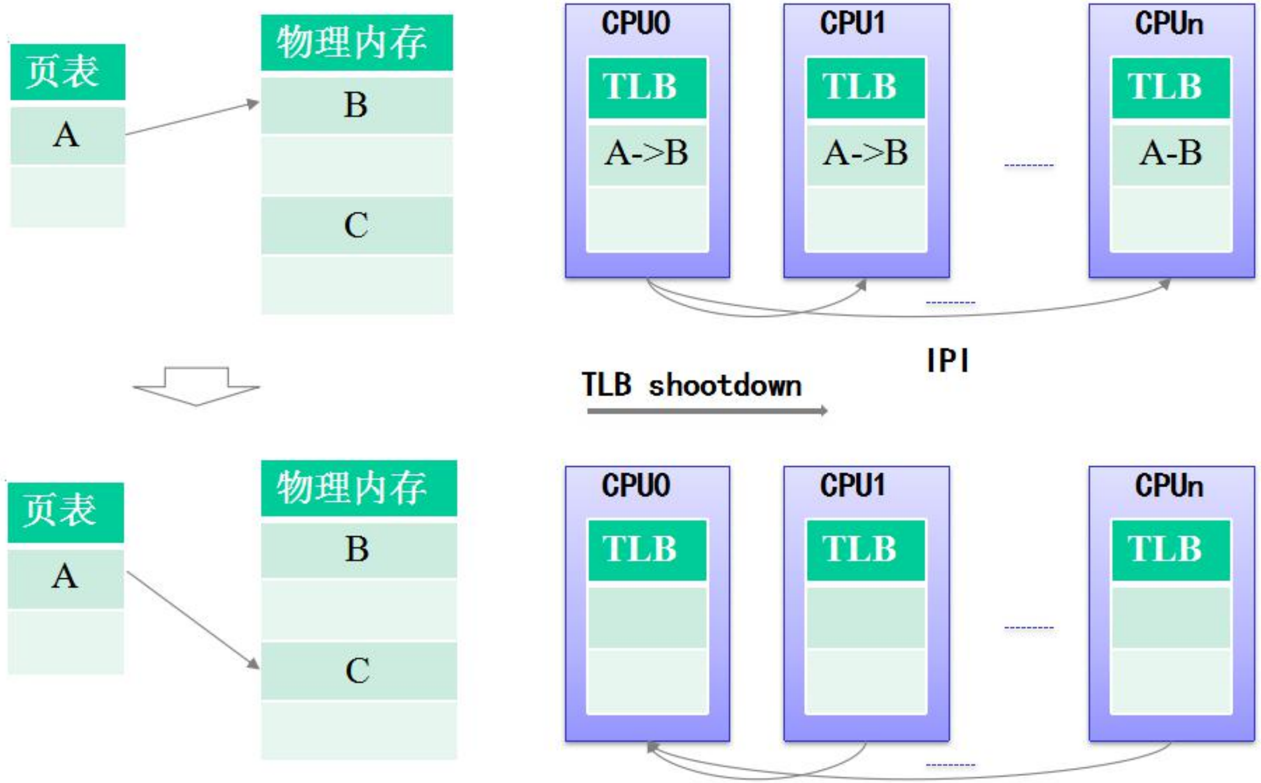
Boost preempted vCPU

- Performance data:



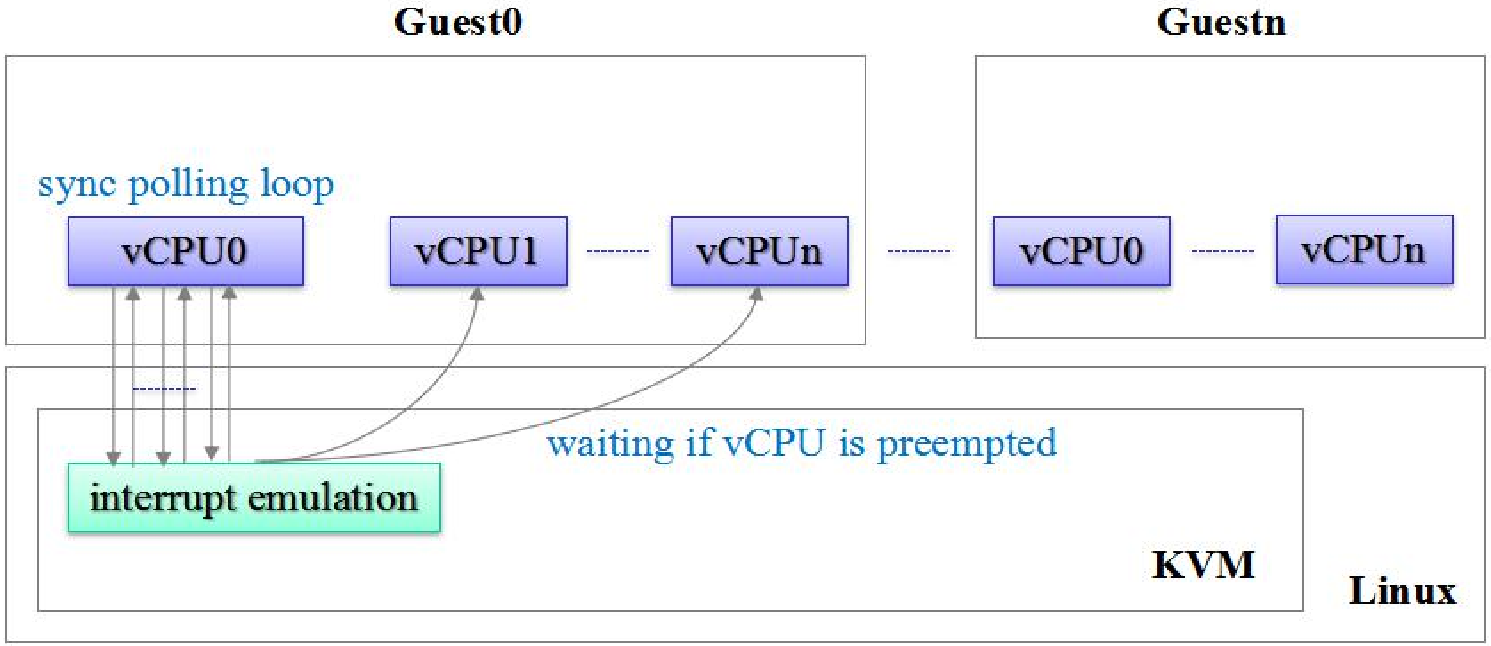
TLB Shootdown Preemption

- A TLB is a cache of translation from memory virtual address to physical address. When a CPU changes virtual to physical mapping of an address, it needs to invalidate other CPUs' stale mapping in their respective caches. This process is called TLB shootdown.



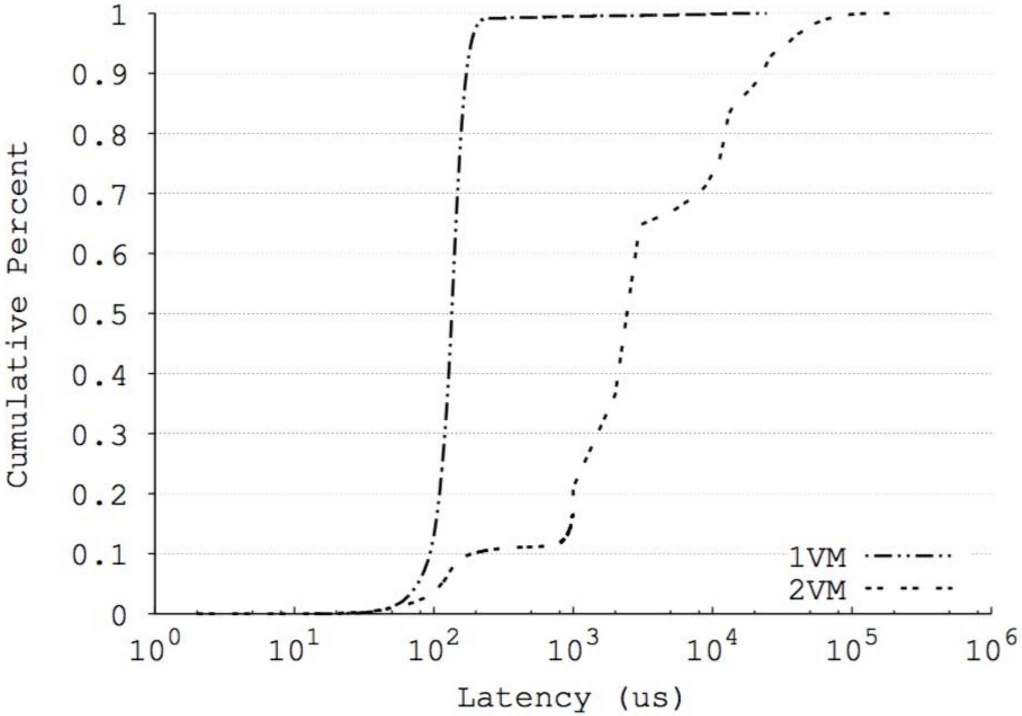
TLB Shootdown Preemption

- Remote TLB flush does a busy wait which is fine in bare-metal scenario. But with-in the guest, the vCPUs might have been pre-empted or blocked. In this scenario, the initiator vCPU would end up busy-waiting for a long amount of time; it also consumes CPU unnecessarily to wake up the target of the shootdown.



TLB Shootdown Preemption

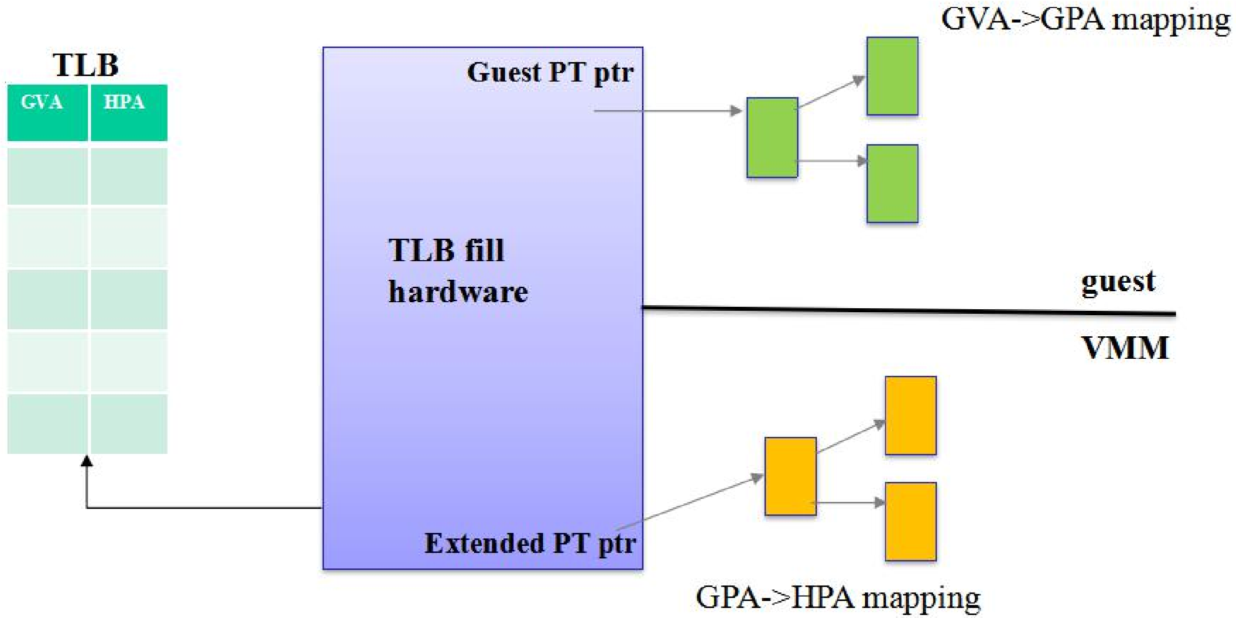
- The time between the preemption and rescheduling of a remote target vCPU is often orders of magnitude larger than the latency that TLB Shootdown operations were designed for.



Tencent Cloud

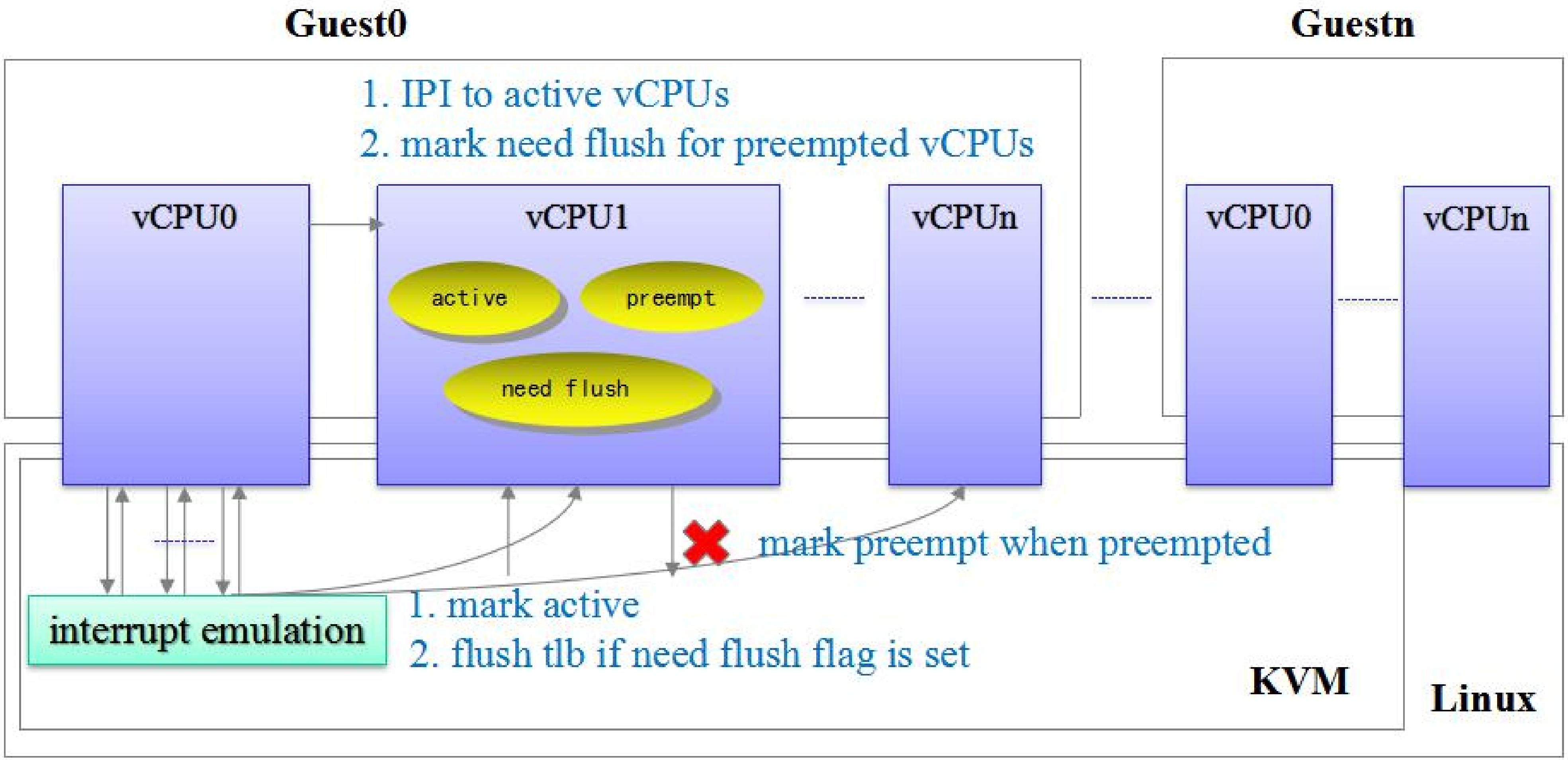
TLB Shootdown Preemption

- Operation that architecturally invalidate entries in the TLBs or paging-structure caches e.g. INVLPG will invalidate combined mapping while EPT is in use.
- INVVPID invalidates combined mapping while EPT is in use.



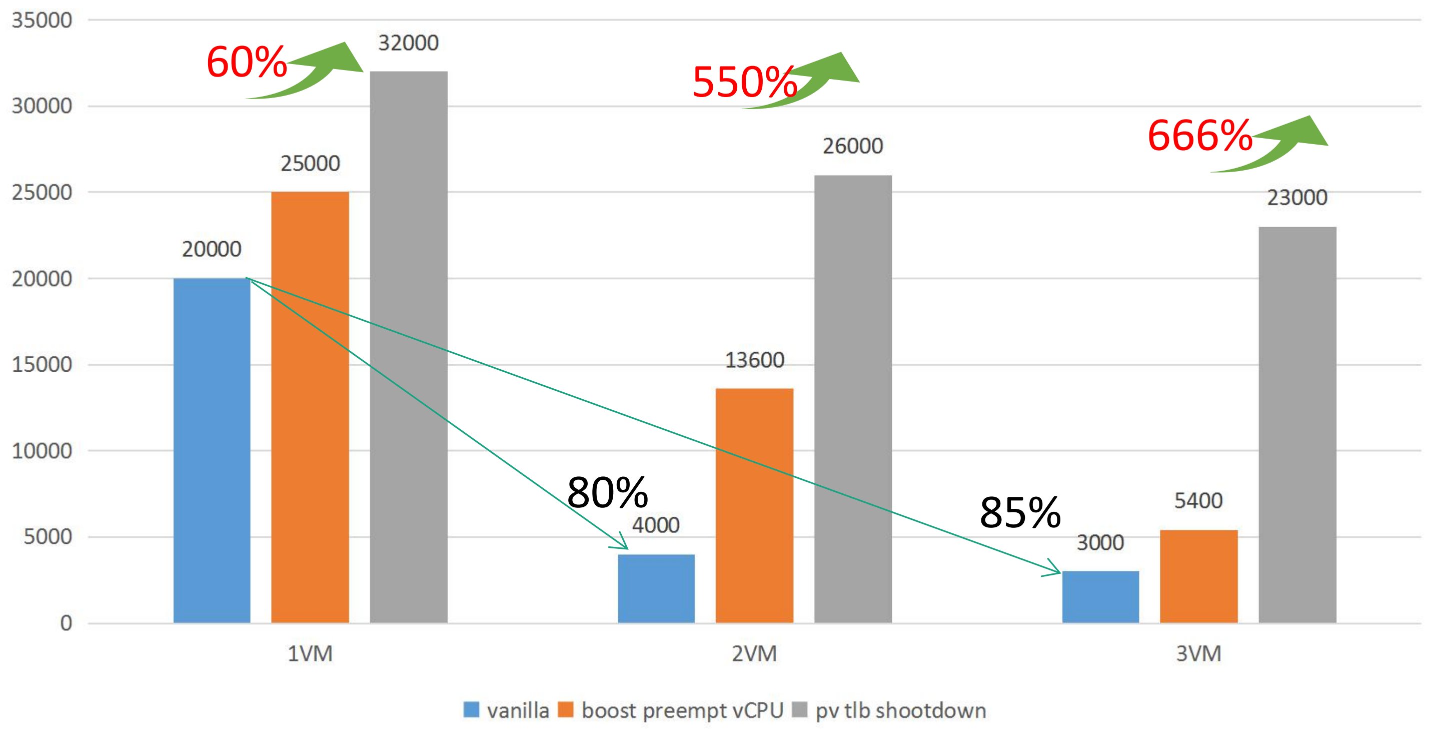
Tencent Cloud

TLB Shootdown Preemption Mitigation



TLB Shootdown Preemption Mitigation

- Performance data



Q/A？