Tencent Cloud

KVM Latency and Scalability Performance Tuning

KVM FORUM 2020

Wanpeng Li

wanpengli@tencent.com

Tencent Cloud

Agenda

- Virtual IPI Fastpath
- Virtual TSC-Deadline Timer Fastpath
- Boost Preempted vCPU
- Yield To IPI Target

Tencent Cloud

Generic Fastpath Handler Motivation

- ICR and TSCDEADLINE MSRs write cause the main MSRs write vmexits
- Multicast IPIs are not as common as unicast IPI like RESCHEDULE_VECTOR and CALL_FUNCTION_SINGLE_VECTOR

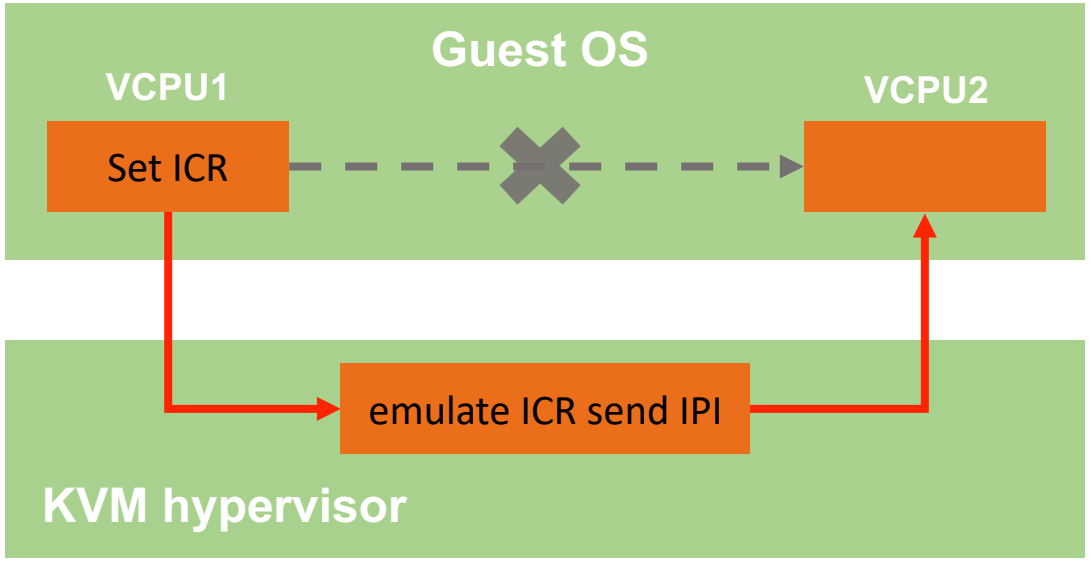
VM-EXIT	Samples	Samples%	Time%	Min Time	Max Time	Avg time
EXTERNAL_INTERRUPT	486083	52.91%	70.17%	2.09us	71.87us	6.17us (+- 0.07%)
MSR_WRITE	379929	41.36%	25.43%	1.90us	60.18us	2.86us (+- 0.03%)
EXCEPTION_NMI	45766	4.98%	3.25%	2.81us	58.02us	3.03us (+- 0.08%)
CPUID	3436	0.37%	0.13%	1.42us	10.70us	1.65us (+- 0.33%)
EPT_MISCONFIG	2084	0.23%	0.59%	2.12us	494.71us	12.08us (+- 2.22%)
IO_INSTRUCTION	1110	0.12%	0.29%	3.84us	50.04us	11.15us (+- 2.48%)
MSR_READ	222	0.02%	0.01%	1.52us	3.14us	1.84us (+- 1.49%)
EOI_INDUCED	30	0.00%	0.00%	2.87us	3.86us	3.35us (+- 1.36%)
EPT_VIOLATION	12	0.00%	0.13%	455.78us	474.64us	465.16us (+- 0.32%)

```
MSR_WRITE:
`6e0'(IA32_TSC_DEADLINE): 304839,
'830'(Interrupt Command Register, ICR): 67492
```

Tencent Cloud

Virtual IPI Fastpath

- Emulate IPI send after
  - various guest states save and host states load
  - various conditions checking
  - host interrupts and preemption enabled
  - expensive RCU operations



Tencent Cloud

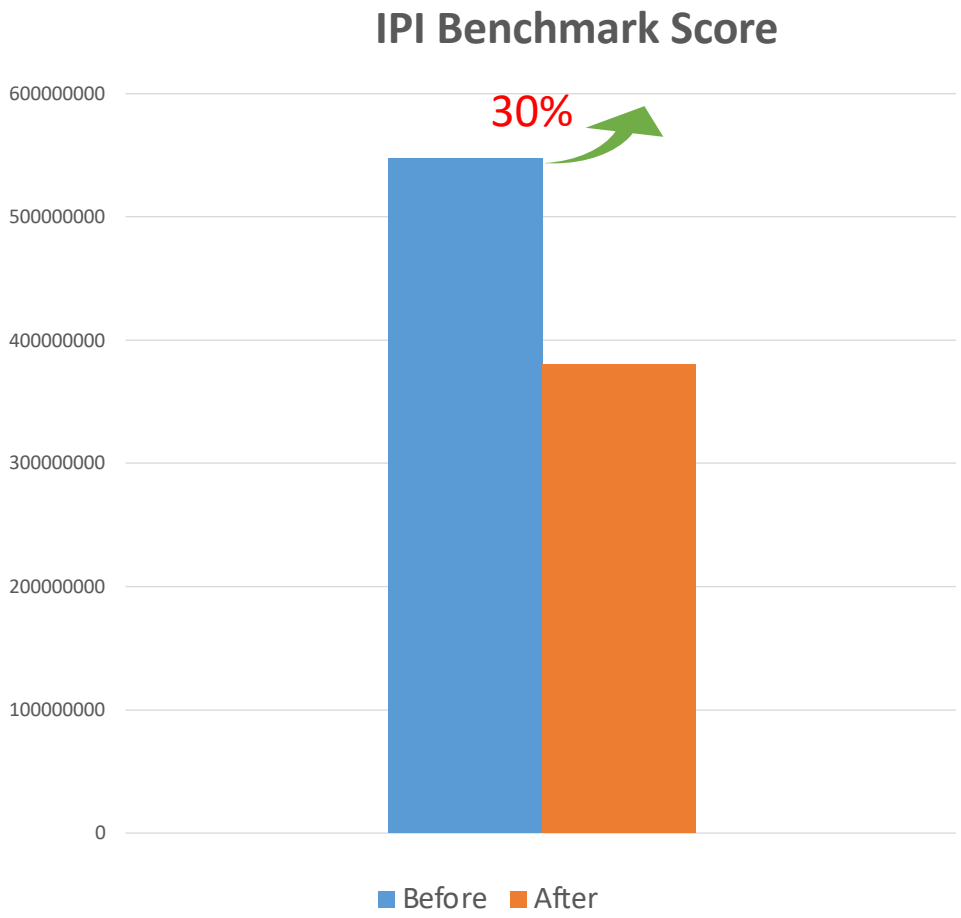
Virtual IPI Fastpath

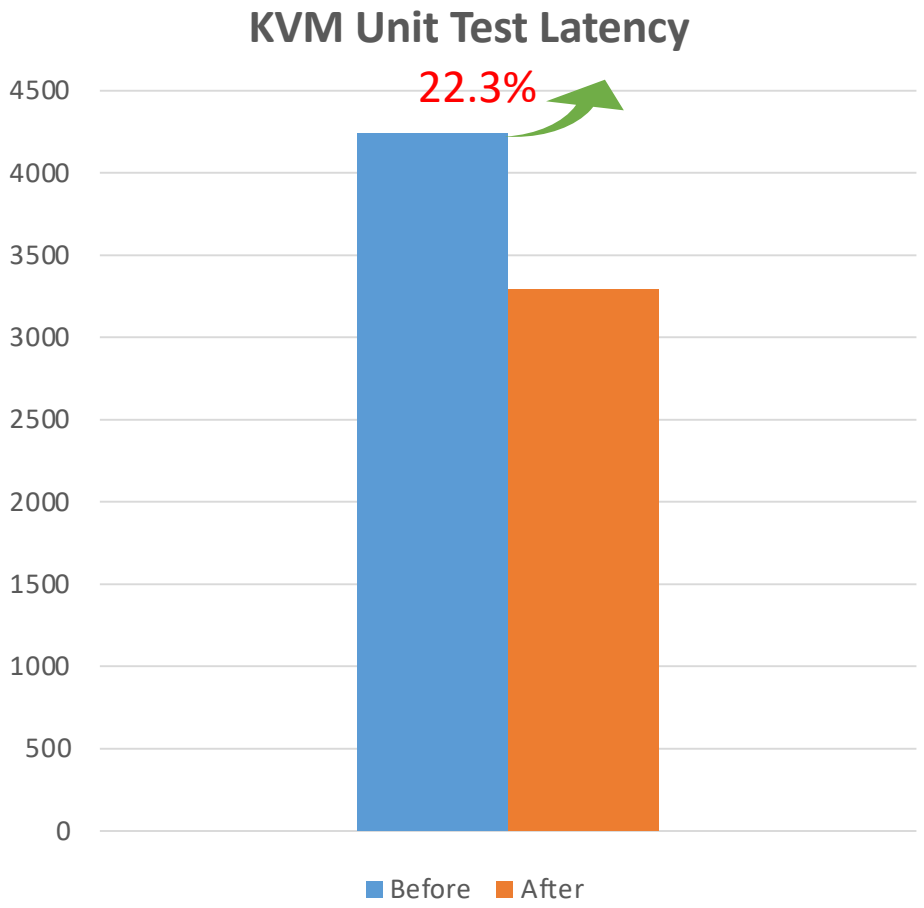
- Sending the virtual IPI to the target vCPU in a very early stage of KVM VMExit handler
  - Before host interrupts are enabled
  - Before expensive operations such as reacquiring KVM’s SRCU lock

Tencent Cloud

Virtual IPI Fastpath

- Performance data

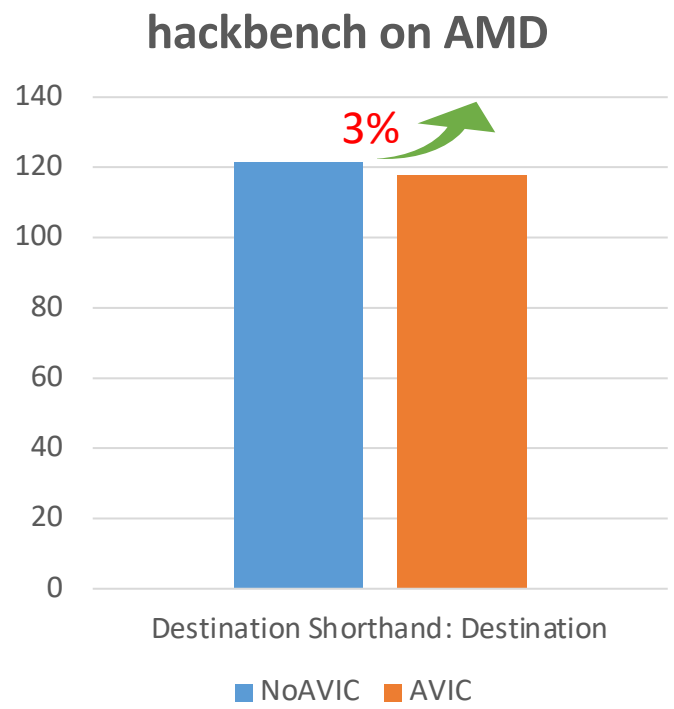


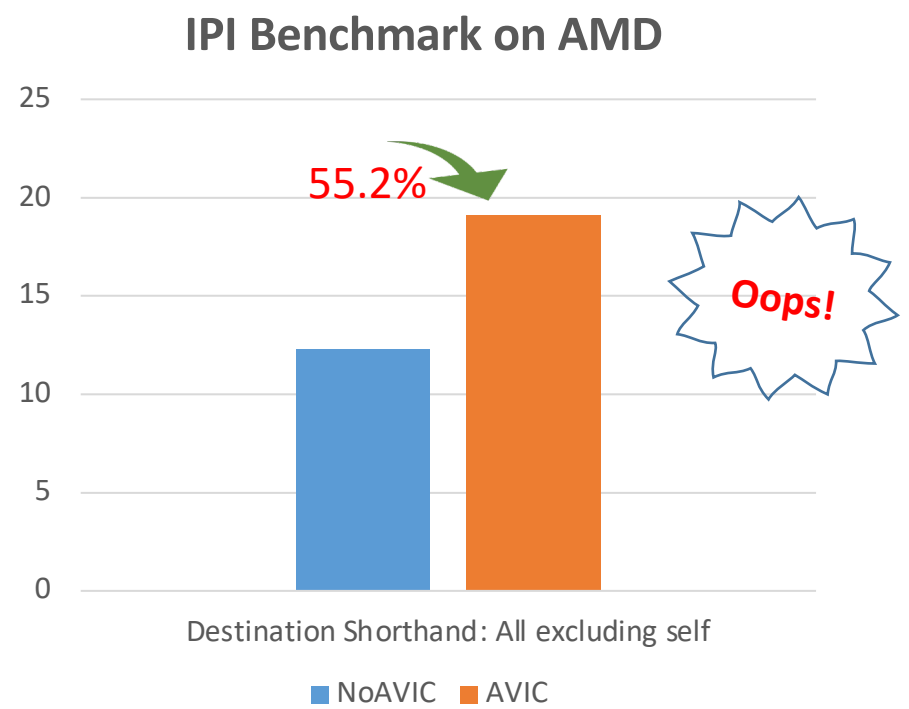


Tencent Cloud

IPI AMD hardware acceleration

- Evaluation Environment
  - Hardware: AMD ROME, 2 sockets, 96 cores, 192 threads
  - VM: 180 vCPUs, with xapic
  - Latency less is better

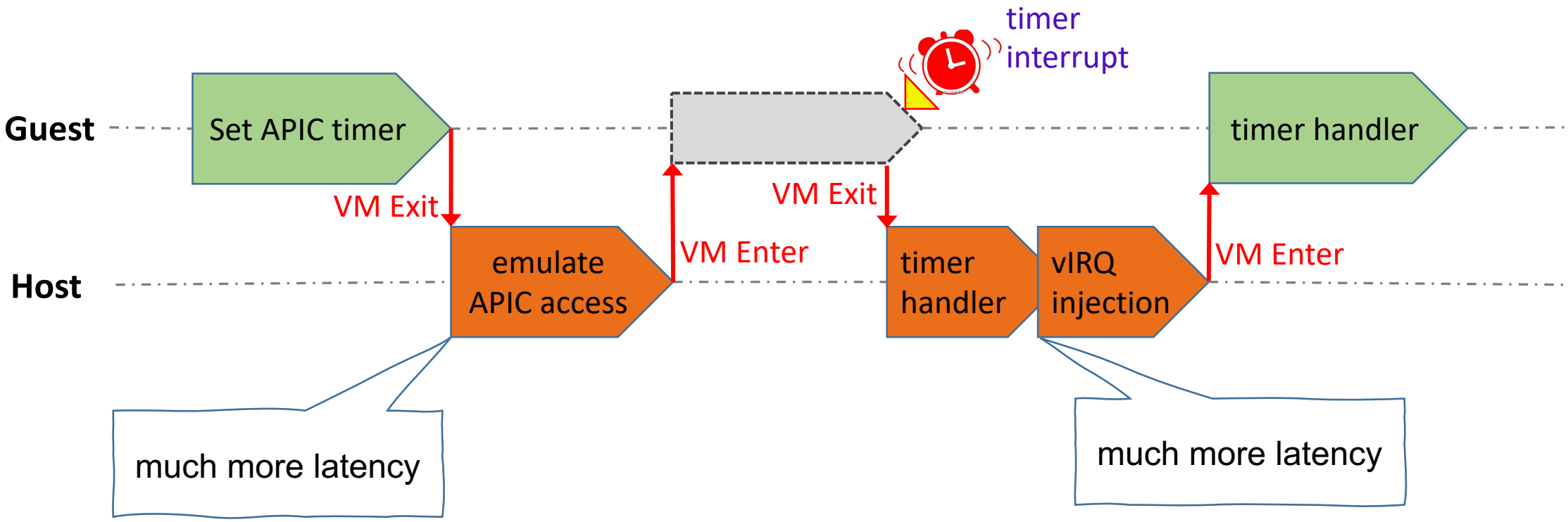




Tencent Cloud

Virtual TSC-Deadline Timer Fastpath

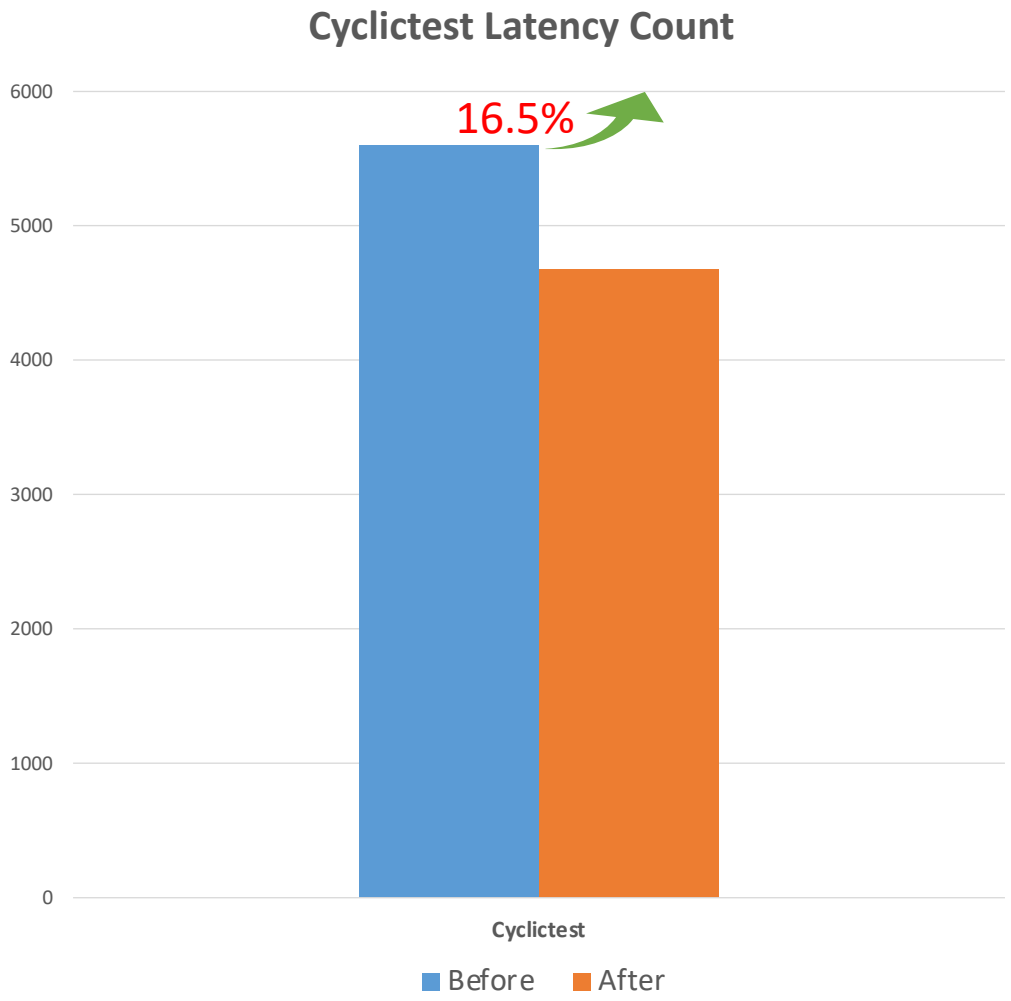
- Both arm timer and timer fire incur vmexits
- Various housekeeping tasks before emulation



Tencent Cloud

Virtual TSC-Deadline Timer Fastpath

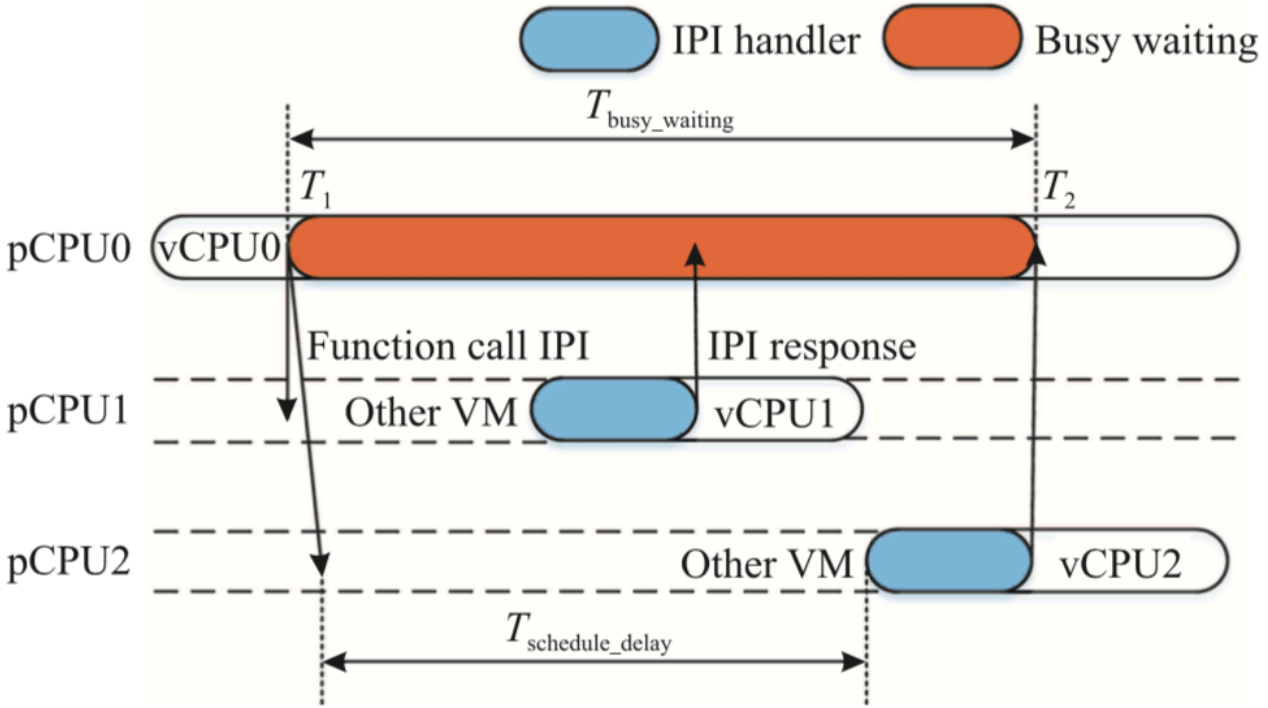
- Vmexit due to TSC deadline timer emulation
  - Shortcutting various housekeeping tasks in the vCPU loop
  - Handle it and vmentry immediately



Tencent Cloud

Boost preempted vCPU

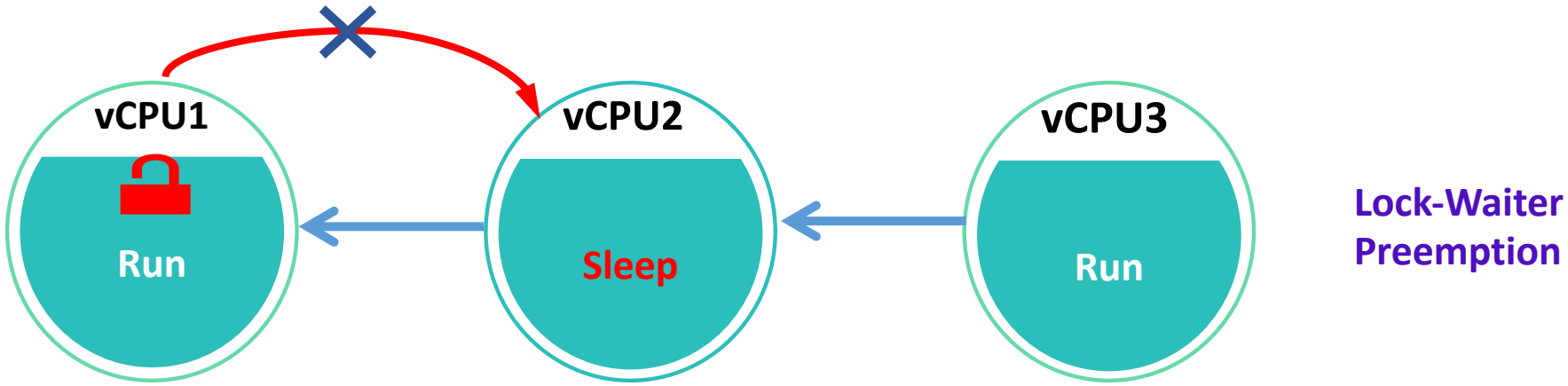
- Boost vCPUs that are ready to deliver interrupts
  - Most *smp_call_function_many* calls are synchronous, we want to boost not just lock holders but also vCPUs that are delivering interrupts. The IPI target vCPUs are also good yield candidates.



Tencent Cloud

Boost preempted vCPU

- Lock Waiter Preemption
  - Due to the FIFO-ordered spinlock algorithm whenever a hypervisor preempts 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.

Tencent Cloud

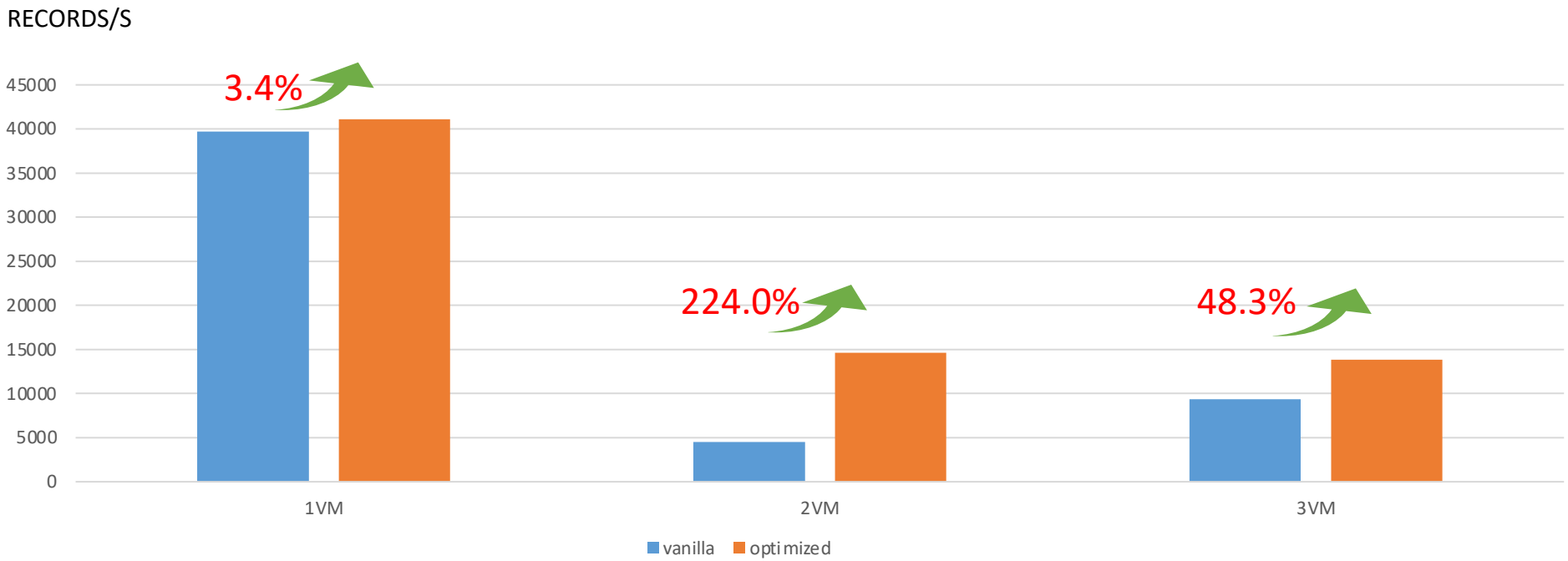
Yield To IPI Target

- When sending a call-function IPI-many to vCPUs
  - yield if any of the IPI target vCPUs was preempted
  - select the first preempted target vCPU which we found

Tencent Cloud

Boost preempted vCPU and Yield

- Evaluation Environment
  - Hardware: Xeon Cascade Lake 2 sockets, 48 cores, 96 threads.
  - VM: each 96 vCPUs
  - Test case: One is running *ebizzy -M*, others are running cpu-bound workloads



Tencent Cloud

Reference

- https://lkml.org/lkml/2019/11/20/1281
- https://lkml.org/lkml/2020/3/25/1221
- https://lkml.org/lkml/2020/5/6/881
- https://lkml.org/lkml/2019/7/18/385
- https://lkml.org/lkml/2019/6/11/469
- https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=89340d

Tencent Cloud

Q/A ?