Minimizing VMExits in Private Cloud by Aggressive PV IPI and Passthrough Timer

Huaqiao & Yibo zhou

huaqiao@bytedance.com

zhouyibo@bytedance.com

ByteDance

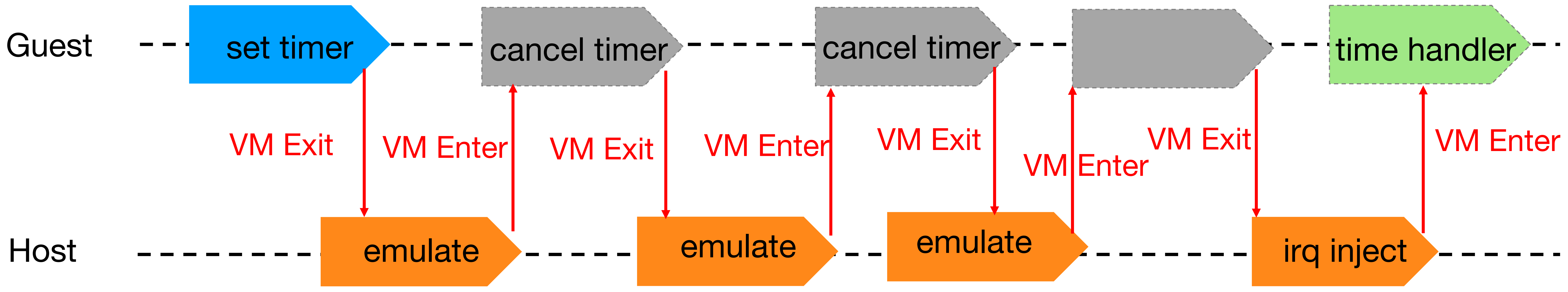
Agenda

- Background
  - problem
- The Solutions
  - Timer passthrough
  - NoExit PVIPI
- Future Work

Background

Problem 1:Timer Exits

- Arming/Disarming/Firing timers incur vmexits
- High frequency of timer reprogramming (arm/disarm) in our scenario

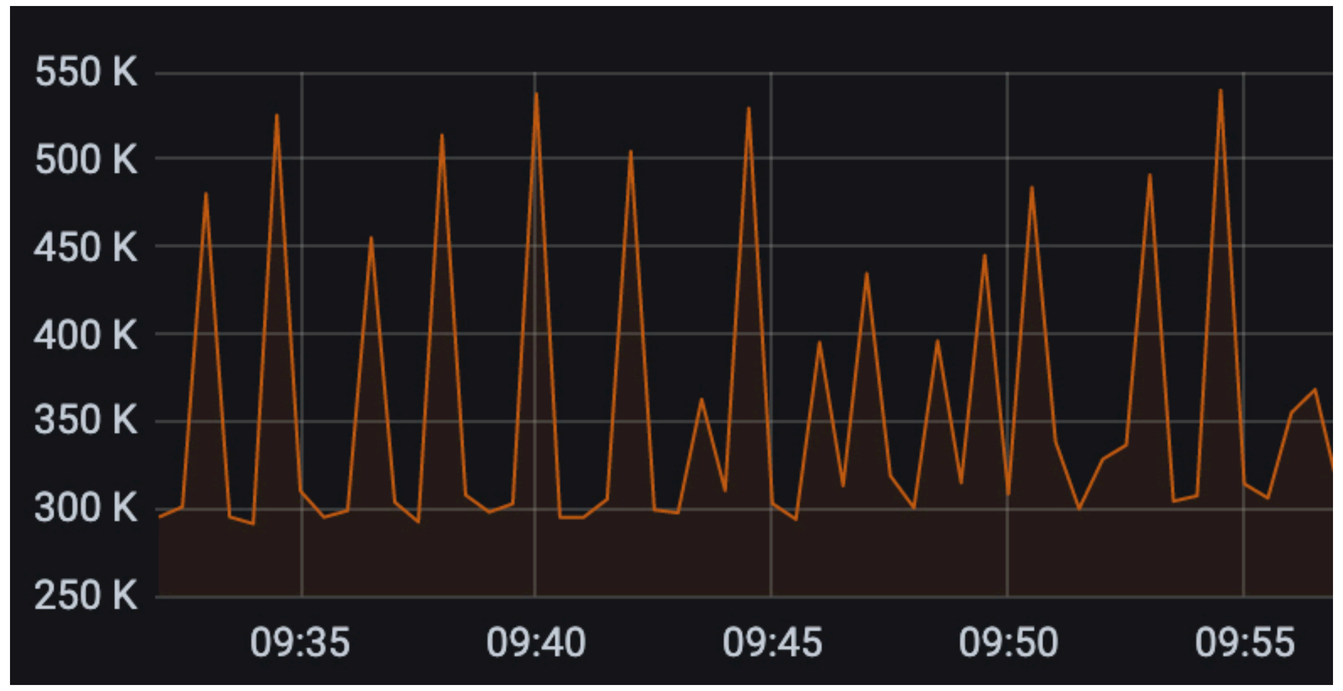


Problem 2: IPI Exits

- ➢ Large VMs are used widely in our scenario
  - High frequency of IPIs

- ➢ Not well addressed by existing PVIPI

Type	VCPUS	Memory(G)
T1	72	376
T2	104	187



The nums of kvmexit caused by ipi every 5m

Solutions

Existing Solution- Exitless timer

- ➢ Exitless timer by Wanpeng Li from Tencent cloud
  - Housekeeping cpus are needed
  - Inject expired timer interrupt via posted interrupt
- ➢ PV timer by Yang Zhang from Alibaba cloud
  - Guest kernel must be modified for the pv feature
  - Dedicate CPU must be reserved

Our solution-Exitless timer

- ➢ New Exitless Timer: passthrough timer
  - The vm access the physical lapic timer directly
  - Offload the host timer to the preemption timer when vmenter
  - Inject timer interrupt into vm when external interrupt exit happened

Our solution-Timer Passthrough

➢ The vm access physical lapic timer

- The lapic timer of vm should work in tsc deadline mode
- Disable intercept tsc deadline msr
- Adjust the host tsc value when vmenter for the vm can use the physical tsc successfully
    - □ vm tsc_value = host_tsc_value * (TSC_multiplier) + offset

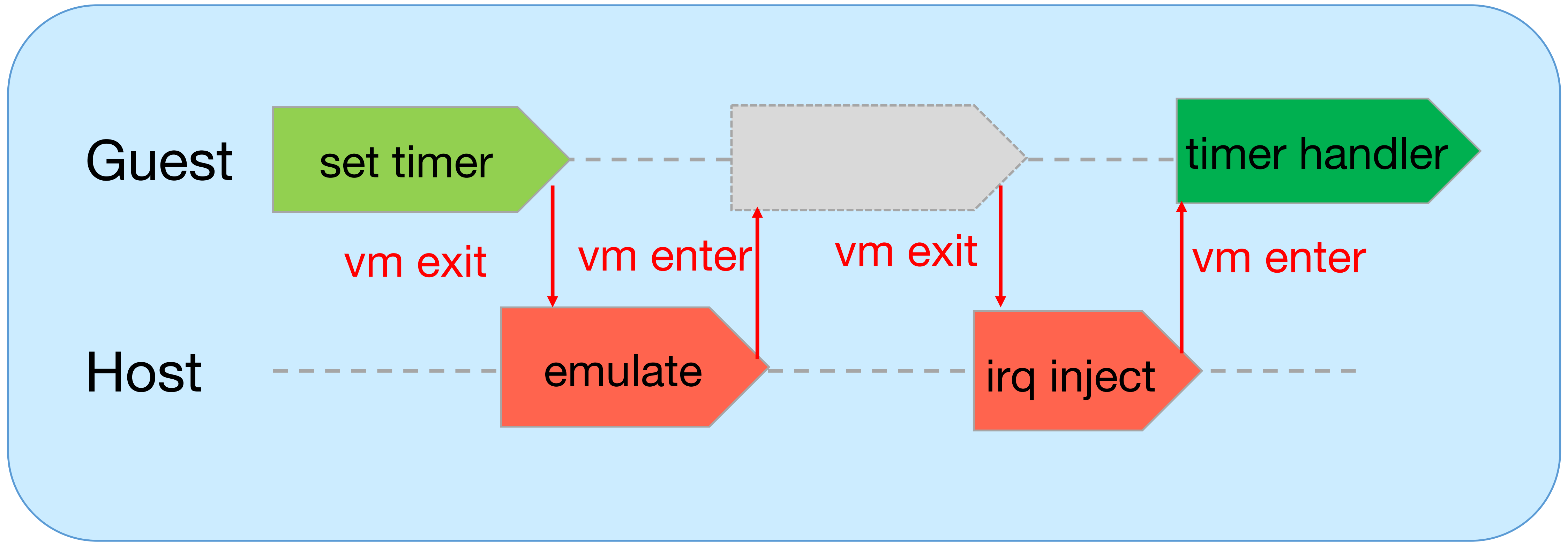
Our solution- Timer Passthrough

➢ Offload the host timer to preemption timer

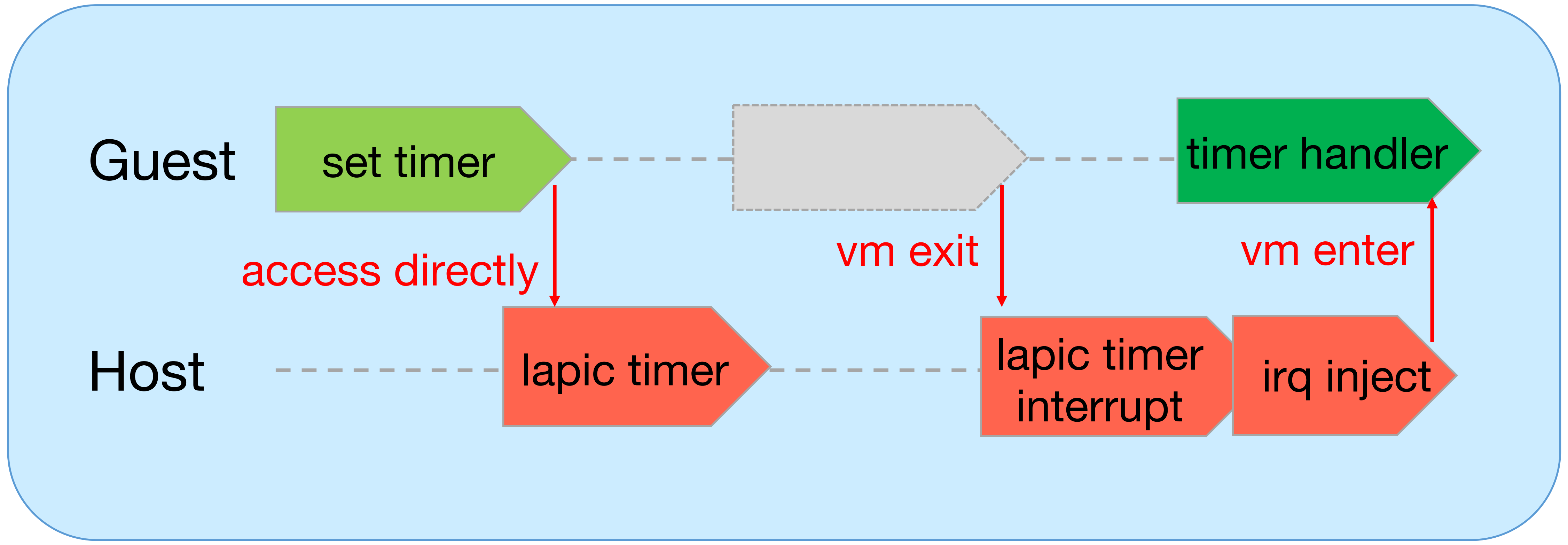
- On VM enter: get the latest timer which will expire and offload it to preemption timer
- On VM exit and vCPU preblock: restore the host timer to the physical timer and the vm timer to soft timer which is emulated by vmm.
- On preemption timer expire: call host clock event handler

Our solution-Timer Passthrough

- ➢ Normal VM lapic timer



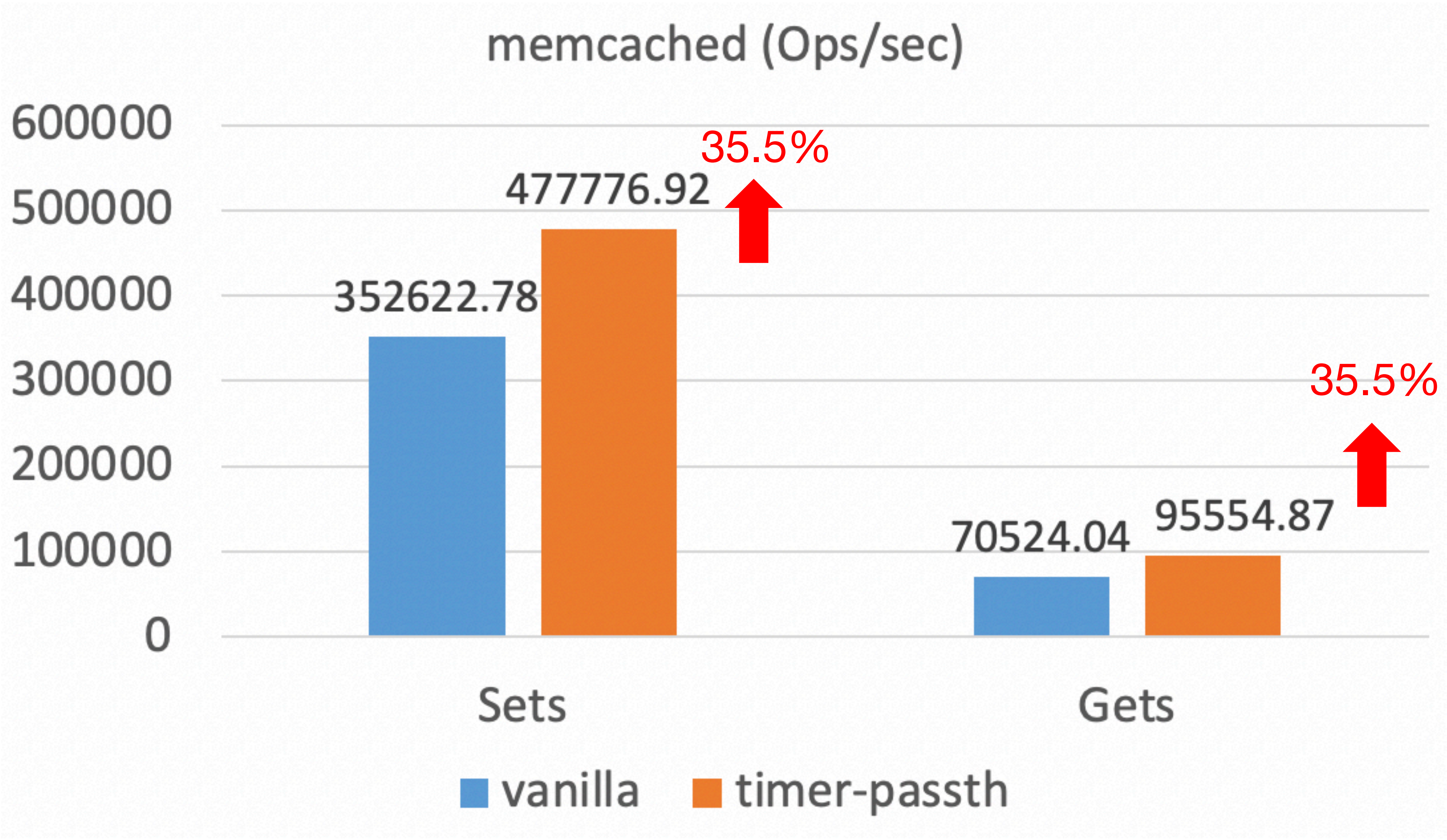
- ➢ Passthrough VM lapic timer

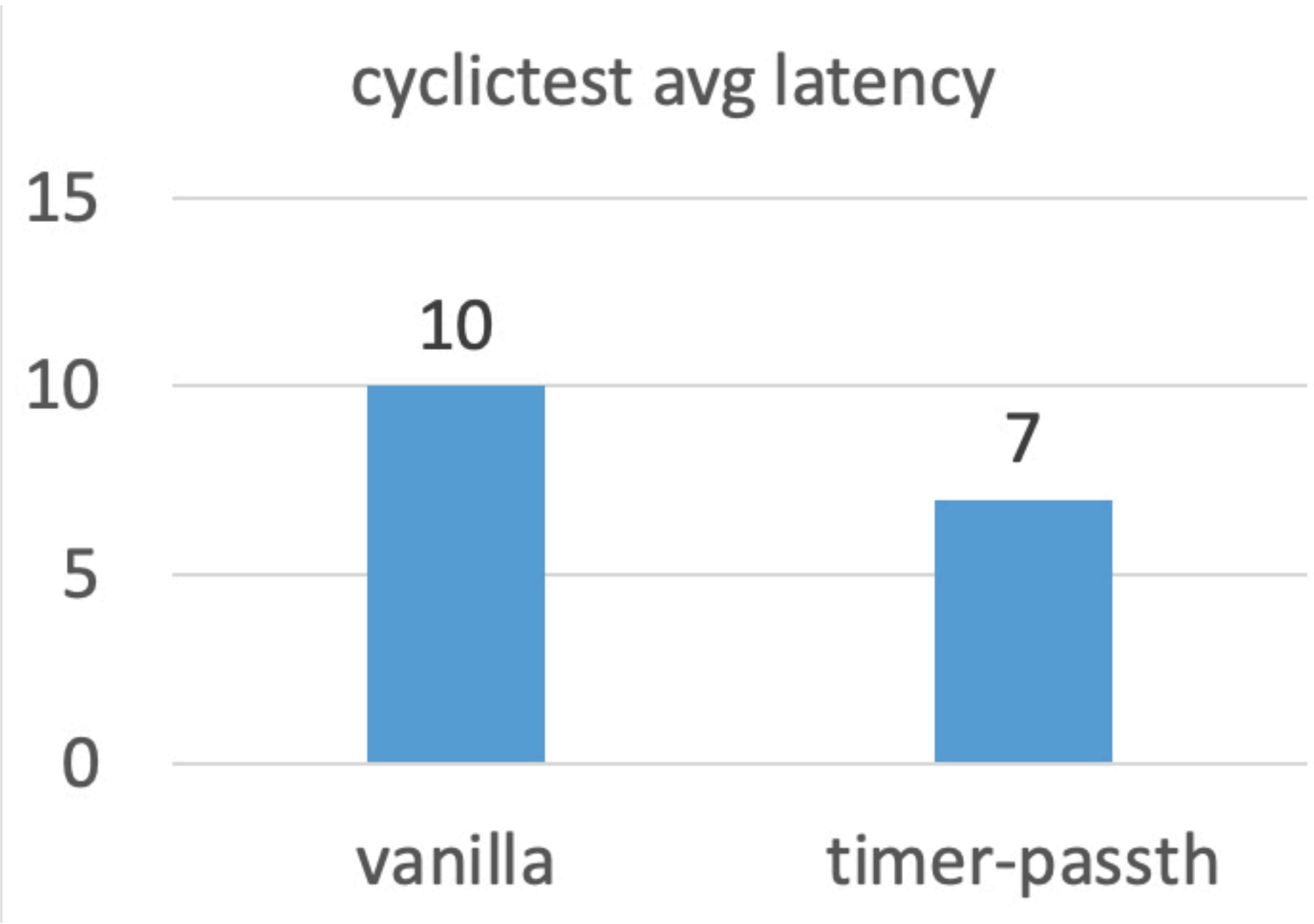


Our solution-Timer Passthrough

- Evaluation

Hardware: Intel(R) Xeon(R) Platinum 8260 CPU @ 2.40GHz





Existing solution – Exitless IPI

- Exitless IPI by Wanpeng Li from Tencent cloud
  - All the dst cpus are marked in a bitmap
  - Send ipis to all cpus together by one hypercall
  - VMM scans the bitmap and sends IPIs to vCPUs marked in bitmap one by one

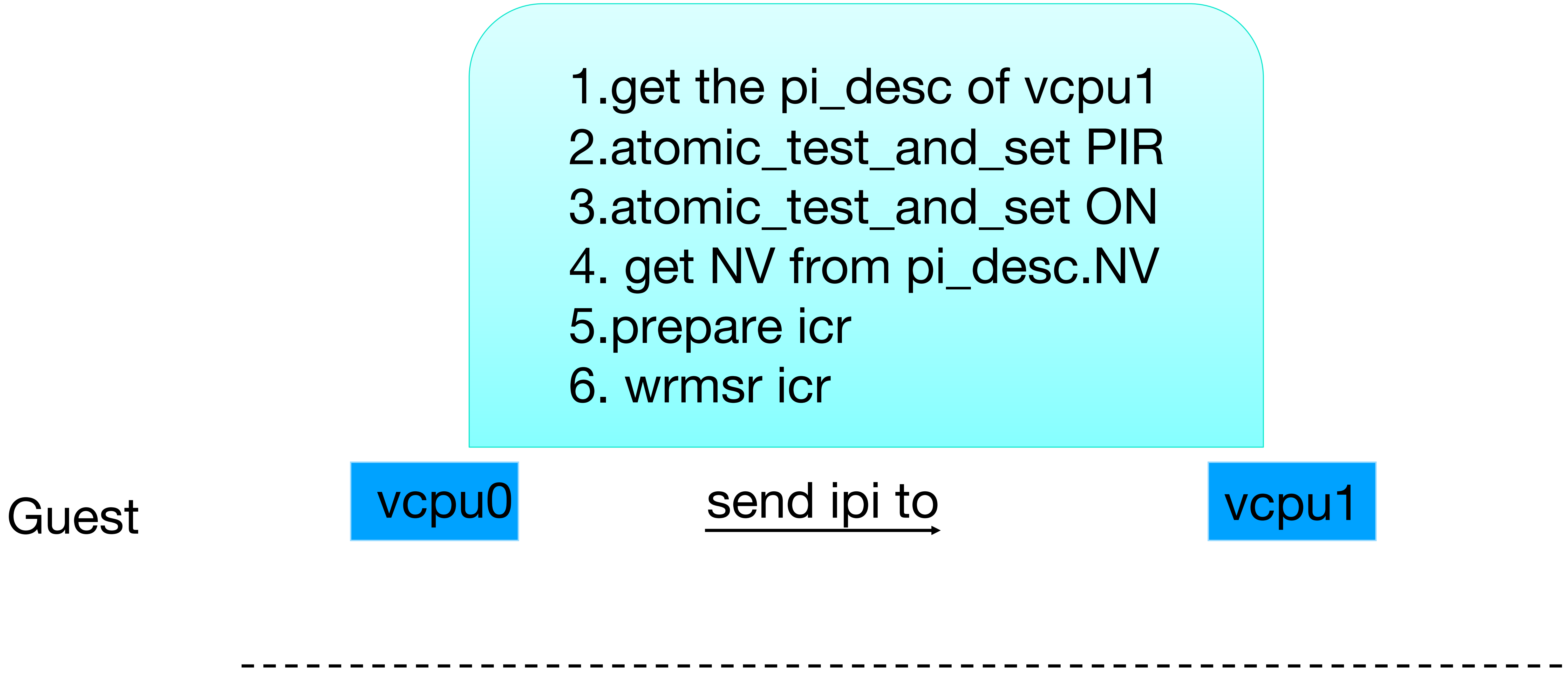
Our solution – NoExit PVIPI

➢ NoExit PVIPI

- Passthrough pi_desc to guest and do not intercept MSR.ICR
- Offer MSR_KVM_PV_ICR for guest to send special interrupt, e.g., SMI, NMI, SIPI, etc
- Send IPI directly by guest via posted interrupt without vmexit
- RFC: https://patchwork.kernel.org/patch/11759063/

Our solution – NoExit PVIPI

- NoExit PVIPI



1.get the pi_desc of vcpu1
2.atomic_test_and_set PIR
3.atomic_test_and_set ON
4. get NV from pi_desc.NV
5.prepare icr
6. wrmsr icr

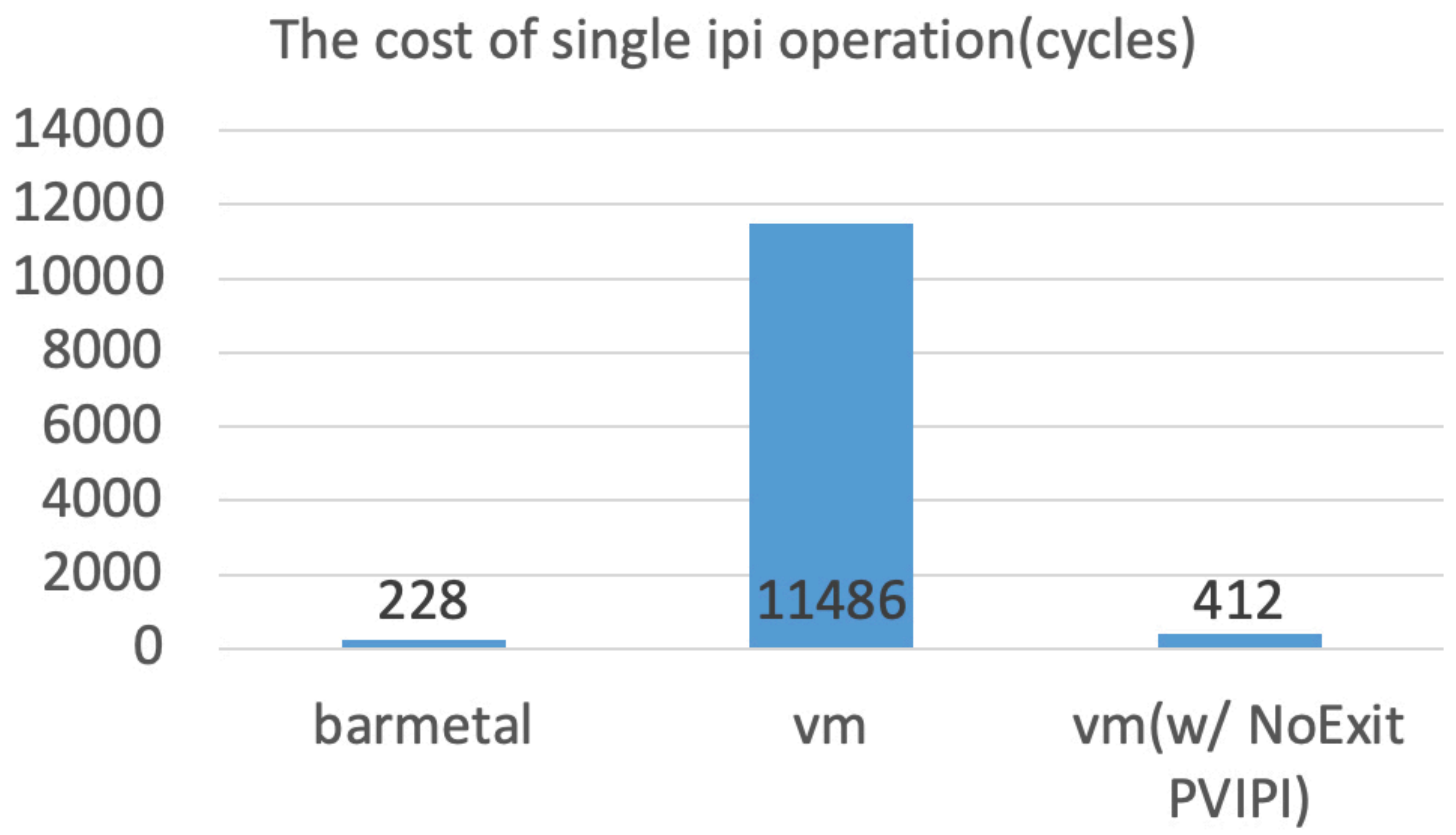
Host

- Disable the intercept of MSR.ICR when NoExit PVIPI is enabled in guest
- Passthrough pi_desc to vm

Our solution – NoExit PVIPI

- Evaluation

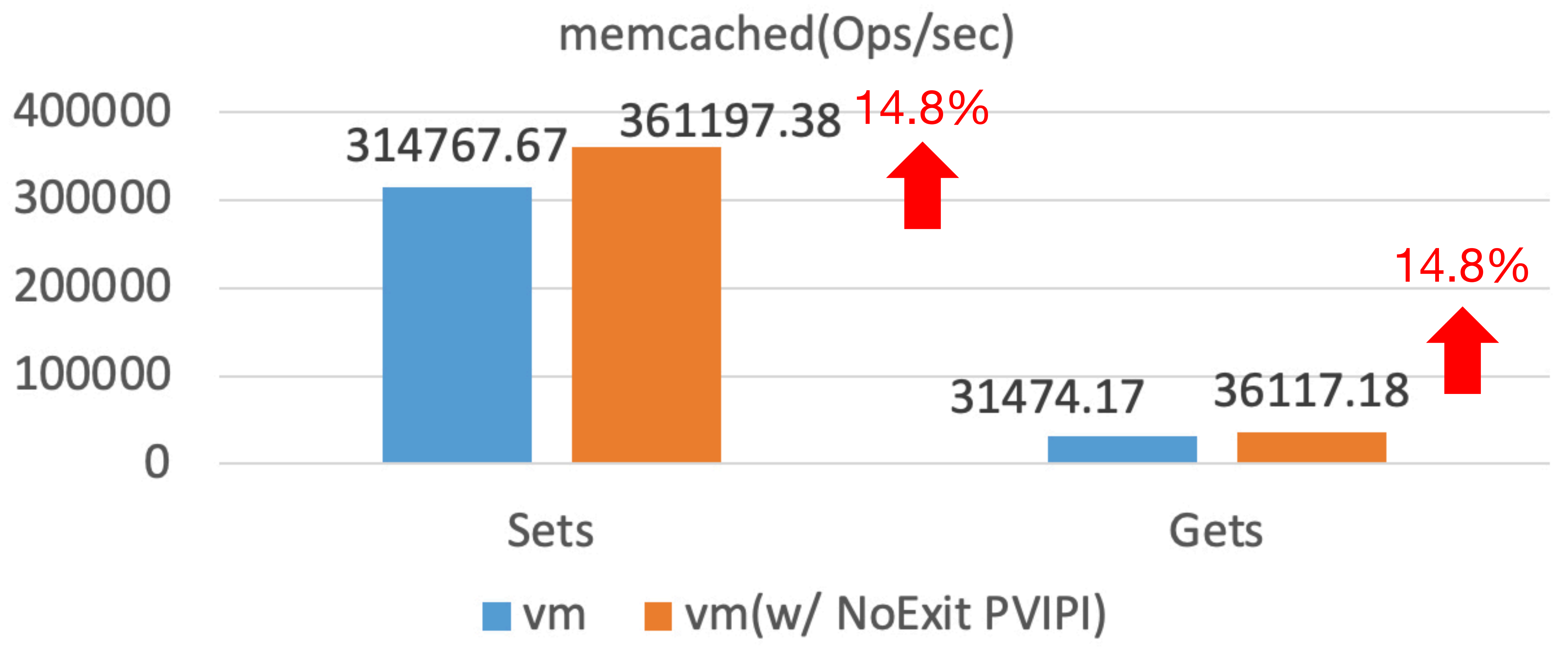
Hardware: Intel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz



Our solution – NoExit PVIPI

- Evaluation

Hardware: Intel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz



Future Work

Future Work

- ➢ NoExit PVIPI
  - Security harden, e.g., via EPTP Switch feature by VMFUNC
- ➢ Passthrough Timer
  - Support live migration
  - Dynamically turn on/off the feature

Thank You

ByteDance

ByteDance