Tencent Cloud

Torwards a more Scalable KVM Hypervisor

KVM FORUM 2018

Wanpeng Li

wanpengli@tencent.com

Tencent Cloud

Agenda

- Paravirtualized TLB shootdown
- Exitless IPIs
- Disable mwait/hlt/pause vmexits to improve latency

Tencent Cloud

TLB Shootdown Preemption

- When executing inside a virtual machine environment, OS level synchronization primitives such as locking, tlb shootdown and RCU are faced with significant challenges due to the scheduling behavior of the underlying host scheduler. Operations that are ensured to last only a short amount of time on bare-metal hardware, are capable of taking considerably longer when running virtualized.

Tencent Cloud

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.
- Modern operating systems consider TLB shootdown operations to be performance critical and so optimize them to exhibit very low latency. The implementation of these operations is therefore architected to ensure that shootdowns can be completed with very low latencies through the use of IPI based signalling.

Tencent Cloud

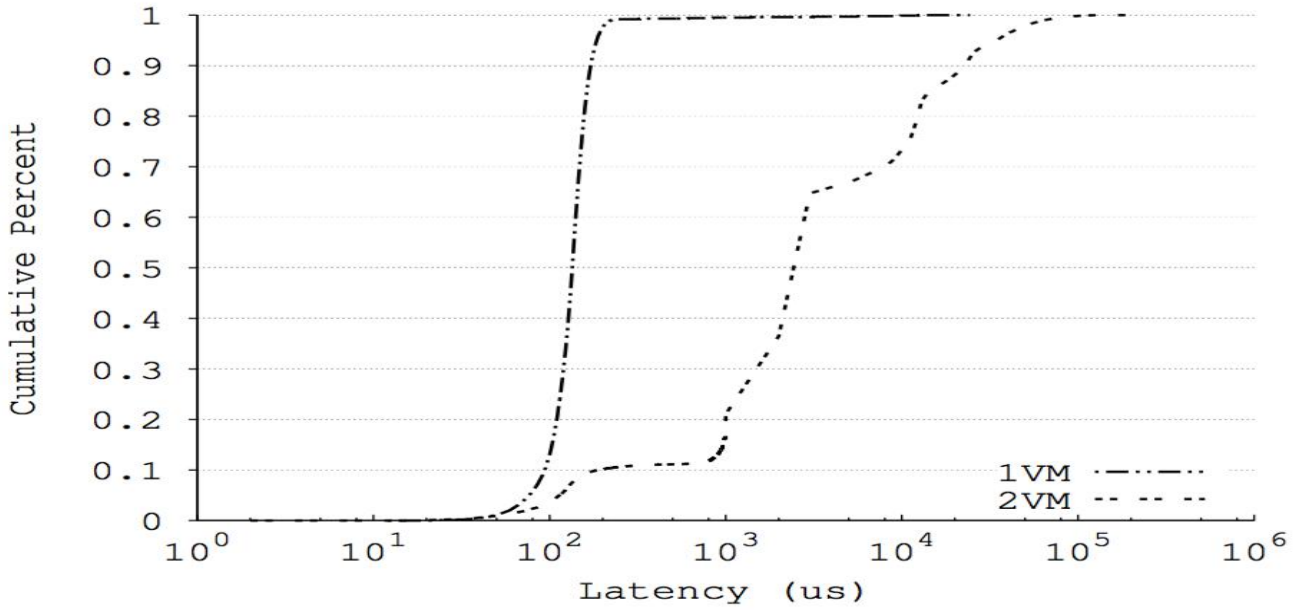
TLB Shootdown Preemption

- 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.

Tencent Cloud

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.



CDF of dedup benchmark TLB Shootdown Latency

Tencent Cloud

TLB Shootdown Preemption Mitigation

- The paravirtualized TLB shootdown does not wait for vCPUs that are sleeping, instead KVM will flush the TLB as soon as the vCPU starts running again. The improvement is clearly visible when the host is overcommitted; in this case, the PV TLB flush (in addition to avoiding the wait on the main CPU) prevents preempted vCPUs from stealing precious execution time from the running ones.

Tencent Cloud

TLB Shootdown Preemption Mitigation

- The caching translation information while EPT is in use:
  - guest-physical mapping = guest physical to host physical, page-structure-cache entries
  - combined mapping = guest virtual to host physical, page-structure-cache entries
- Operation that architecturally invalidate entries in the TLBs or paging-structure caches e.g. INVLPG will invalidate combined mapping while EPT is in use.
- INVEPT invalidates guest-physical and combined mappings.
- INVVPID invalidates combined mapping while EPT is in use.

Tencent Cloud

TLB Shootdown Preemption Mitigation

- KVM_VCPU_PREEMPED flag
  - The flag is recorded to the memory which is shared between the guest and the host when preemption notifier on the host notifies the vCPU is preempted and be scheduled out.
  - The flag will be cleared before next vmentry.
- pv_mmu_ops.flush_tlb_others
  - call IPIs for active vCPUs and record another KVM_VCPU_FLUSH_TLB flag for pre-empted vCPUs.
  - KVM will just do the flush via INVVPID on the guest's behalf the next time the vCPU is scheduled if KVM_VCPU_FLUSH_TLB flag is set

Tencent Cloud

TLB Shootdown Preemption Mitigation

Evaluation Environment:

Hardware: Xeon Gold 6142 2.6GHz 2 sockets, 32 cores, 64 threads

VM : each 64 vCPUs

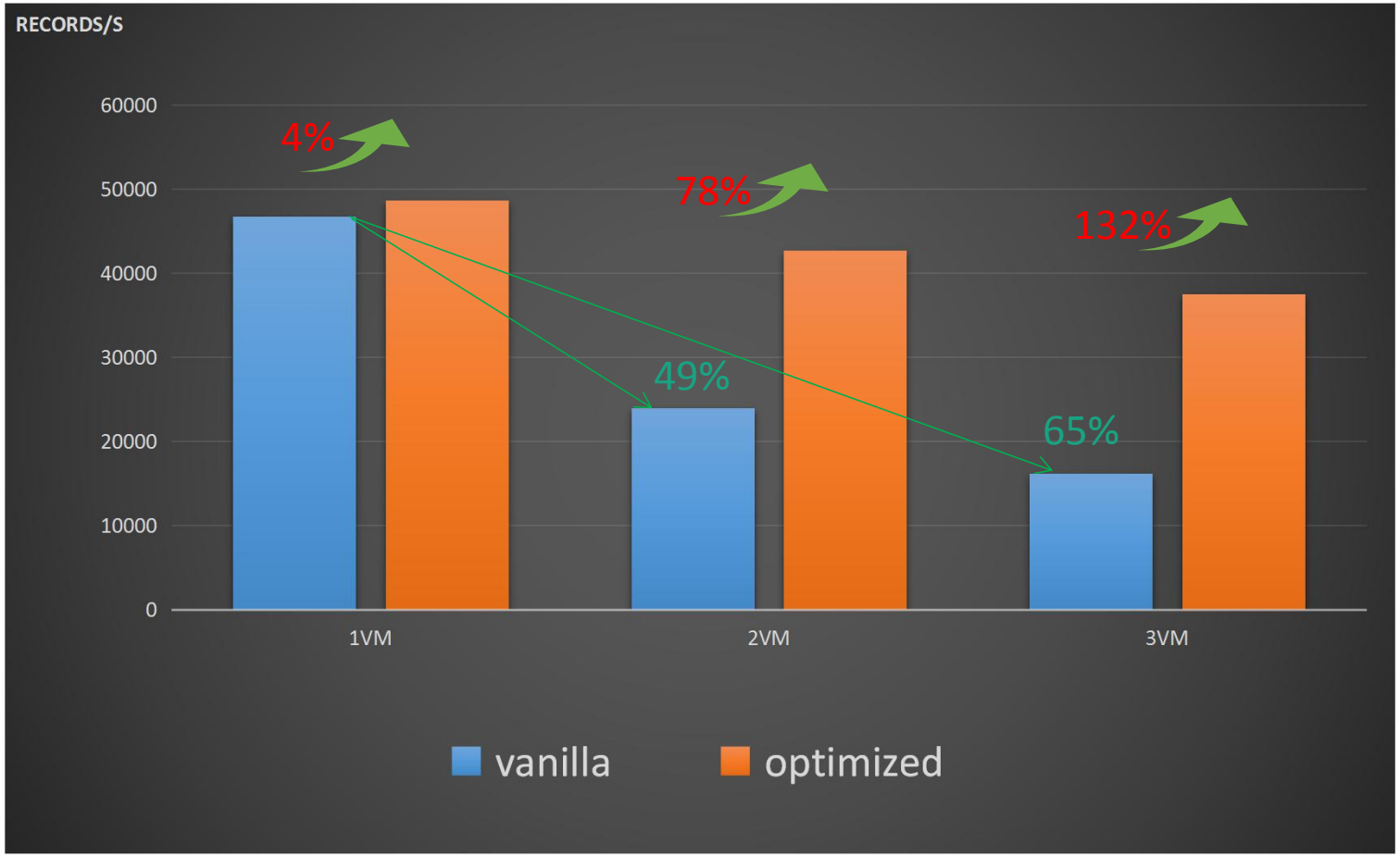
Test case : ebizzy -M

(ebizzy is designed to generate a workload resembling common web application server workloads. It is highly threaded, has a large in-memory working set, and allocates and deallocates memory frequently.)

Tencent Cloud

TLB Shootdown Preemption Mitigation

Evaluation Result:



Tencent Cloud

Exitless IPIs

- x2APIC in linux kernel:
  - Cluster mode: send IPI per Cluster(max 16 logical cpus per Cluster)
  - Phyiscal mode: send IPI one by one
- Each writes to ICR register will cause a vmexit, multicast IPIs and “Function Call interrupts” make it worse when scaling to large VMs.

Tencent Cloud

Exitless IPIs

- Set the destination to bitmap in the guest.
- Use a hypercall to send IPIs to multiple vCPUs. The hypercall lets a guest send multicast IPIs, with at most 128 destinations per hypercall in 64-bit mode and 64 vCPUs per hypercall in 32-bit mode.
- The destinations are represented by a bitmap contained in the first two arguments. The bitmap will be scaned to send IPIs to the target vCPUs.

Tencent Cloud

Exitless IPIs

Evaluation Environment:

Hardware: Xeon Skylake 2.5GHz, 2 sockets, 40 cores, 80 threads

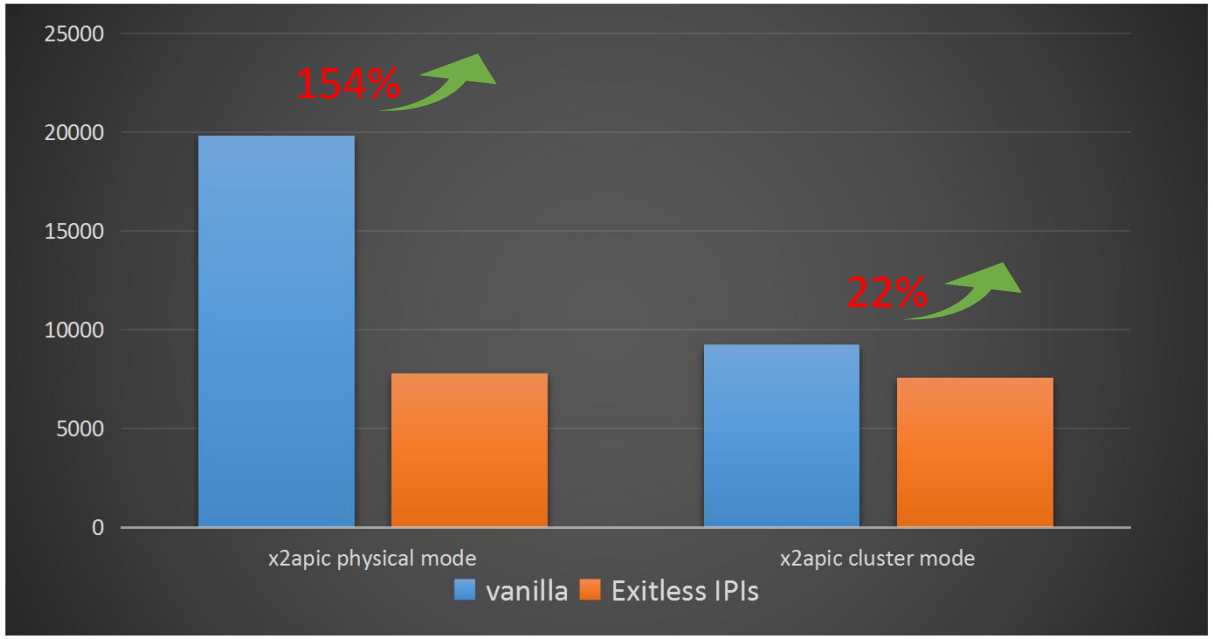
VM : 80 vCPUs

Test case : IPI microbenchmark

Evaluation Result:

time-consuming

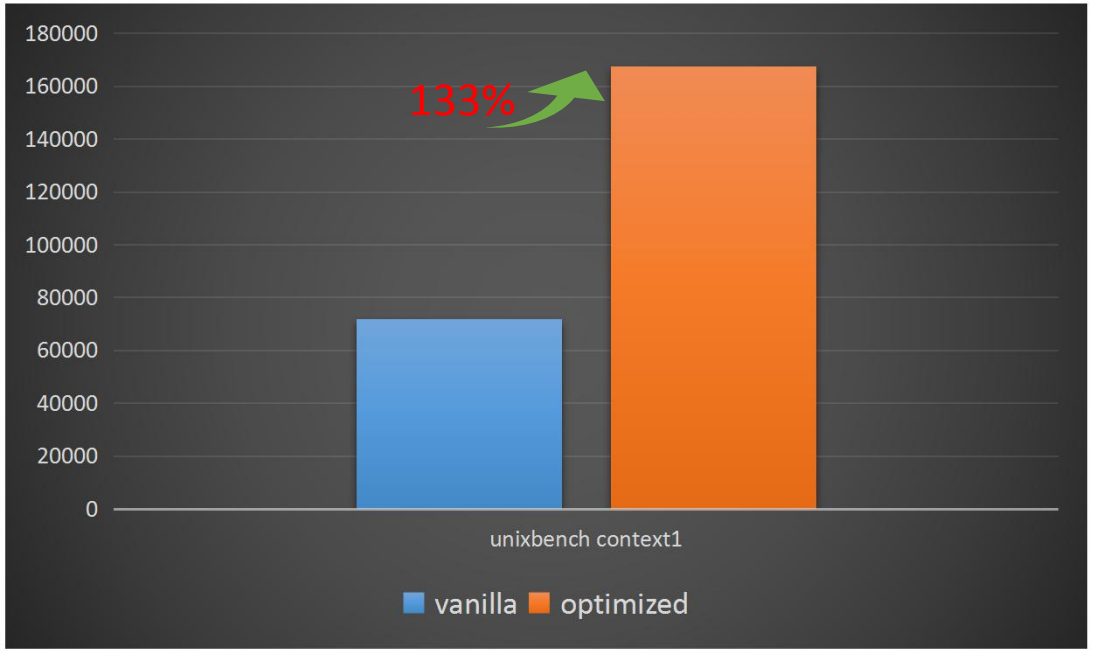
less is better



Tencent Cloud

Disable mwait/hlt/pause vmexits to improve latency

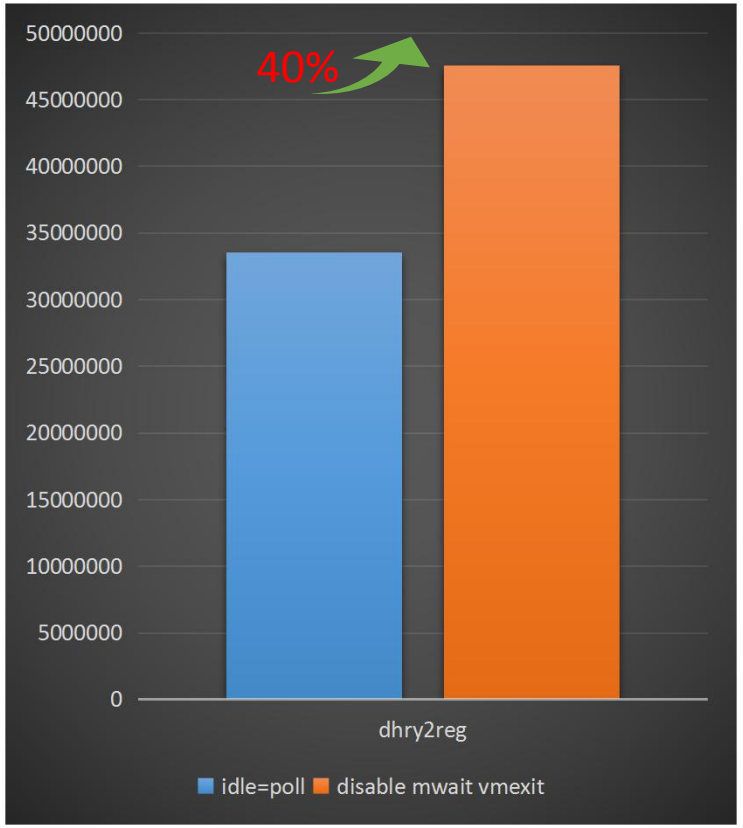
Allow userspace to disable MWAIT/HLT/PAUSE vmexits, a guest can put a (physical) CPU into a power saving state. The other CPUs in the same package can get turbo boost on demand when comparing to idle=poll.

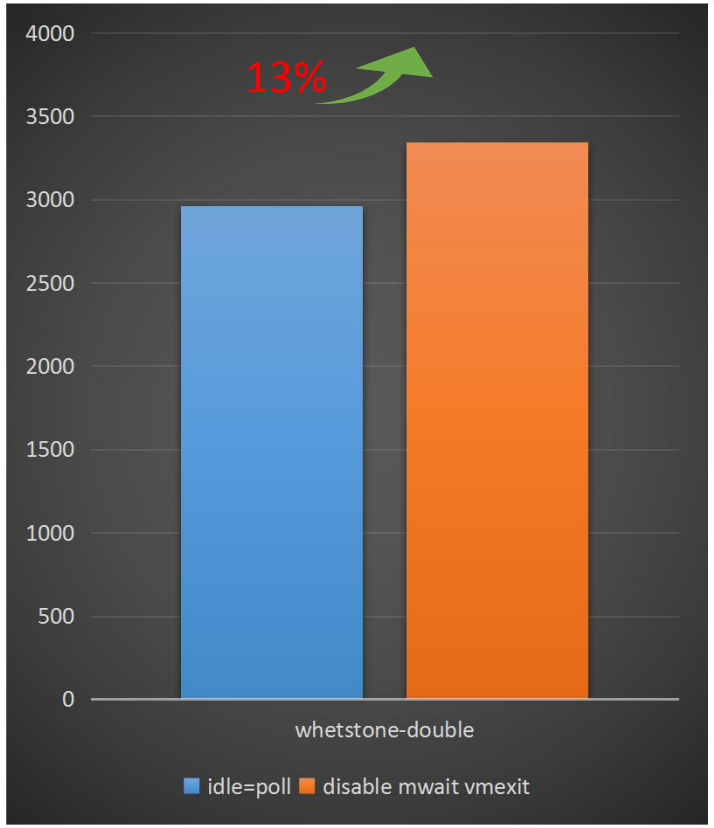


Tencent Cloud

Disable mwait/hlt/pause vmexits to improve latency

Evaluation Environment:
test case 1: two VMs, each 4 vCPUs, both idle=poll, one running Unixbench
test case 2: two VMs, each 4 vCPUs, both disable mwait vmexit, one running Unixbench





Tencent Cloud

Disable mwait/hlt/pause vmexits to improve latency

```
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
 2438 kernel    20   0 10.227g 3.378g  32320 S 398.7 21.8  12:20.38 qemu-system-x86
 2771 kernel    20   0 5168008 385536  30032 S 396.7  2.4   0:32.86 qemu-system-x86
```

idle=poll in guest

```
        Package   |             Core    |           CPU 0       CPU 4
                  |                     | C0 active 105.4%      105.4%
                  |                     | POLL        0.0%    0.0 ms  0.0%    0.0 ms
                  |                     | C1E         0.0%    0.0 ms  0.0%    0.0 ms
C2 (pc2)    0.0%  |                     |
C3 (pc3)    0.0%  | C3 (cc3)    0.0%    | C3          0.0%    0.0 ms  0.0%    0.0 ms
C6 (pc6)    0.0%  | C6 (cc6)    0.0%    | C6          0.0%    0.0 ms  0.0%    0.0 ms
C7 (pc7)    0.0%  | C7 (cc7)    0.0%    | C7s         0.0%    0.5 ms  0.0%    0.1 ms

                  |             Core    |           CPU 1       CPU 5
                  |                     | C0 active 105.3%      105.4%
                  |                     | POLL        0.0%    0.0 ms  0.0%    0.0 ms
                  |                     | C1E         0.0%    1.0 ms  0.0%    0.0 ms
                  |                     |
                  | C3 (cc3)    0.0%    | C3          0.0%    0.4 ms  0.0%    0.0 ms
                  | C6 (cc6)    0.0%    | C6          0.0%    1.3 ms  0.0%    0.0 ms
                  | C7 (cc7)    0.0%    | C7s         0.0%    0.8 ms  0.0%    0.2 ms

                  |             Core    |           CPU 2       CPU 6
                  |                     | C0 active 105.4%      105.4%
                  |                     | POLL        0.0%    0.0 ms  0.0%    0.0 ms
                  |                     | C1E         0.0%    0.0 ms  0.0%    0.2 ms
                  |                     |
                  | C3 (cc3)    0.0%    | C3          0.0%    0.0 ms  0.0%    0.3 ms
                  | C6 (cc6)    0.0%    | C6          0.0%    0.0 ms  0.0%    0.0 ms
                  | C7 (cc7)    0.0%    | C7s         0.0%    0.0 ms  0.0%    0.0 ms

                  |             Core    |           CPU 3       CPU 7
                  |                     | C0 active 105.4%      105.3%
                  |                     | POLL        0.0%    0.0 ms  0.0%    0.0 ms
                  |                     | C1E         0.0%    0.0 ms  0.0%    0.1 ms
                  |                     |
                  | C3 (cc3)    0.0%    | C3          0.0%    0.0 ms  0.0%    0.3 ms
                  | C6 (cc6)    0.0%    | C6          0.0%    0.0 ms  0.0%    0.5 ms
                  | C7 (cc7)    0.0%    | C7s         0.0%    0.0 ms  0.0%    1.3 ms
```

disable mwait vmexit

```
        Package   |             Core    |           CPU 0       CPU 4
                  |                     | C0 active   1.3%       12.6%
                  |                     | POLL        0.0%    0.0 ms  0.0%    0.0 ms
                  |                     | C1E         0.0%    0.0 ms  0.0%    0.0 ms
C2 (pc2)    9.8%  |                     |
C3 (pc3)    0.0%  | C3 (cc3)   60.4%    | C3          0.0%    0.0 ms  0.0%    0.0 ms
C6 (pc6)    0.0%  | C6 (cc6)    2.9%    | C6          0.0%    0.0 ms  0.0%    0.0 ms
C7 (pc7)    0.0%  | C7 (cc7)   21.1%    | C7s         0.0%    0.0 ms  0.0%    0.0 ms

                  |             Core    |           CPU 1       CPU 5
                  |                     | C0 active   3.9%        2.2%
                  |                     | POLL        0.0%    0.0 ms  0.0%    0.0 ms
                  |                     | C1E         0.0%    0.0 ms  0.0%    0.0 ms
                  |                     |
                  | C3 (cc3)   18.3%    | C3          0.0%    0.0 ms  0.0%    0.0 ms
                  | C6 (cc6)    0.9%    | C6          0.0%    0.0 ms  0.0%    0.0 ms
                  | C7 (cc7)   74.6%    | C7s         0.0%    0.0 ms  0.0%    0.0 ms

                  |             Core    |           CPU 2       CPU 6
                  |                     | C0 active   4.7%        2.9%
                  |                     | POLL        0.0%    0.0 ms  0.0%    0.0 ms
                  |                     | C1E         0.0%    0.0 ms  0.0%    0.0 ms
                  |                     |
                  | C3 (cc3)    0.1%    | C3          0.0%    0.0 ms  0.0%    0.0 ms
                  | C6 (cc6)    0.3%    | C6          0.0%    0.0 ms  0.0%    0.0 ms
                  | C7 (cc7)   92.9%    | C7s         0.0%    0.0 ms  0.0%    0.3 ms

                  |             Core    |           CPU 3       CPU 7
                  |                     | C0 active   4.8%       93.7%
                  |                     | POLL        0.0%    0.0 ms  0.0%    0.0 ms
                  |                     | C1E         0.0%    0.0 ms  0.0%    0.0 ms
                  |                     |
                  | C3 (cc3)    0.4%    | C3          0.0%    0.0 ms  0.0%    0.2 ms
                  | C6 (cc6)    0.0%    | C6          0.0%    0.0 ms  0.0%    0.0 ms
                  | C7 (cc7)   13.3%    | C7s         0.0%    0.0 ms  0.0%    0.0 ms

                  |             GPU     |
```

Tencent Cloud

Reference

- https://lkml.org/lkml/2017/12/12/1300
- https://git.qemu.org/?p=qemu.git;a=commitdiff;h=6976af663d3a19d1
- https://lkml.org/lkml/2018/7/23/108
- https://git.qemu.org/?p=qemu.git;a=commit;h=7f710c32bb893c68b931c68265f0427c032eb7f4
- https://lkml.org/lkml/2018/3/12/359
- https://lists.gnu.org/archive/html/qemu-devel/2018-06/msg06794.html

Tencent Cloud

Q/A?