The traps of using Hyper V features in KVM environment

Liang Li  Aug 2021
Performance issues when using Hyper V features

Cause Analysis & Solutions

Conclusion
Background

- Application scenarios of Window guest
  - Cloud Desktop
  - Cloud Game

- Hyper V
  - Windows guest support is good

- KVM
  - Try to support Windows guest better by simulating Hyper V functions

- Hyper V related features
  - hv-relaxed, hv-time, hv-stimer, hv-stimer-direct, hv-vapic, hv-synic, hv-tlbflush, hv-mpi, hv-spinlocks …
  - Common usage: turn on all features
Workload characteristics of cloud gaming

• 3D rendering
  • High CPU & GPU usage
  • IPI intensive
    • 35000+ IPIs per second
    • 1200+ extra IPIs per second for when Microsoft Remote Desktop for Mac is used
      • IPI send to some of the VCPUs
      • Which can be accelerate by hv-mpi

• Performance drops significantly when running on a VM
  • Compared with running on a bare metal server
  • With Hyper V features enabled
  • Average FPS drops by 1
  • The proportion of [ >55] FPS decreased by 10%
Performance comparison with different config

None of Hyper V feature is set

All Hyper V features are set

Hyper V features can help to reduce the virtualization overhead a lot
Performance comparison with different config

Disable hypervisor CPUID

[Diagram showing VM Exit count and Virtualization cost (us) for different configurations]

- All hv features off
- All hv features on
- Disable hypervisor CPUID

Disable hypervisor CPUID has the lowest virtualization overhead
How Windows guest choose system timer

- **Expose hypervisor CPUID**
  - Priority: Stimer > HPET > RTC
  - Stimer is used when all the Hyper V features are turned on
  - HPET or RTC is used when the hyper v features are turned off

- **Hide hypervisor CPUID**
  - Priority: LAPIC timer > HPET > RTC
  - LAPIC timer is the default system timer
  - Hyper V related features are invalid
Virtualization efficiency of different system timers

- **RTC & HPET**
  - RTC trapped by PIO access
  - HPET trapped by MMIO access
  - Emulated in user space

- **Stimer**
  - Trapped by MSR access
  - Emulated in Kernel

- **LAPIC timer**
  - Trapped by APIC access
  - Emulated in Kernel

- **Virtualization overhead**
  - LAPIC timer == Stimer < HPET < RTC
Cause Analysis

• Why virtualization overhead is lower when Hyper V features are enabled
  • Stimer has lower virtualization overhead than HPET & RTC
• Why virtualization overhead is the lowest after hiding hypervisor CPUID
  • Stimer has side effects
Cause Analysis

• Some facts about Stimer
  • hv-stimer depends on hv-synic
  • The Auto EOI feature of hv-synic conflicts with APICv
    • APICv can reduce interrupt injection overhead
    • The hardware APICv feature is invalid when Stimer is on
• IPI virtualization for Intel CPU
  • Trapped by ICR access
  • Inject interrupts into the VCPU will cause vm exit if APICv is off
• Stimer will increase the IPI virtualization overhead
  • LAPIC timer does not have this problem
  • Turn off Stimer will increase overall virtualization overhead
Solutions

• Hide hypervisor CPUID for scenarios with intensive IPIs
  • Disable all Hyper V features at the same time
  • Can't enjoy the benefits of hv-tlbflush, hv-mpi, hv-spinlocks and hv-xxx

• Adjust the logic of Windows' selection of system timers
  • Decoupling hypervisor CPUID and LAPIC timer
  • Give priority to using LAPIC timer when Hypervisor CPUID is exposed
Solutions

• Resolve the conflict between hv-stimer and APICv
  • Disable the Auto EOI feature of hy-synic
    • Solved by expose HV_DEPRECATING_AEOI_RECOMMENDED
  • Recommend guests use hardware APICv MSR
    • Can be solved by clear HV_X64_APIC_ACCESS_RECOMMENDED
• Optimize the cost of EOI induced vm-exit
  • Avoid EOI induced vm-exit for Stimer
Effect of optimization

Set all HV features (with hv-stimer optimization)

• After optimization, turn on all Hyper V features perform best
Conclusion

• Hyper V features in KVM have room for improvement
  • Any feature should not cause performance degradation
  • Avoid the need for users to decide which feature to use according to workload
• Pay attention to the pitfalls when using the Hyper V features
  • Turning on all Hyper V features is not necessarily the best way
  • Pay attention to business scenarios with intensive IPIs
  • Before related problems are solved, performance tests are required