Partial Device Port Acceleration
About Me

- Alexander Graf
- KVM and Qemu developer
  - Server class PowerPC KVM port
  - S390x Qemu guest support
  - x86 Mac OS X in KVM
  - Nested SVM
  - Xenner
- ...

Montag, 19. November 12
Disk performance
Disk performance
Disk performance
Disk performance

- Overhead split into 2 categories
- Cost of parallel transfers
- Cost of single transfers
- Looking at single transfers today
Disk performance

- No prefetch
- No caching
- One request at a time
Disk performance

- No prefetch
- No caching
- One request at a time

$ dd if=/dev/xxx of=/dev/null bs=4k iflag=direct
Disk performance

Montag, 19. November 12
Disk performance

4k read MB/s

IDE

AHCI

Montag, 19. November 12
Disk performance

Montag, 19. November 12
Device Access

Host
- Qemu

Guest
- dd

Kernel
- kvm

Hardware

Montag, 19. November 12
Device Access

read(4096) from dd

Host

Qemu

Guest

dd

Kernel

kvm

Hardware

Montag, 19. November 12
Device Access

device port access

Host
- Qemu

Guest
- dd
- Kernel

Kernel
- kvm

Hardware
Device Access

device port trap

Host

Qemu

Guest

dd

Kernel

Kernel

kvm

Hardware

Montag, 19. November 12
Device Access

device port emulation

Host

Qemu

Guest

dd

Kernel

kvm

Hardware

Montag, 19. November 12
Device Access

Host
- Qemu

Guest
- dd
- Kernel

Kernel
- kvm

Hardware

return to guest

Montag, 19. November 12
Device Access

device port access done

Host

Qemu

Guest

dd

Kernel

kvm

Hardware
Device Access

read(4096) finished

Host

Qemu

Guest

dd

Kernel

Kernel

kvm

Hardware

Montag, 19. November 12
IDE
Exits per second

Montag, 19. November 12
Exits per second

IDE

AHCI

Montag, 19. November 12
Exits per second

IDE

AHCI

virtio-blk
Exits per second

- Device port emulation is slow
- How to speed it up?
Device Access

- Host
  - Qemu
- Guest
  - Programs
  - Kernel
- Kernel
  - kvm
- Hardware
Device Access

Host
- Qemu

Guest
- Programs

Kernel
- kvm

Hardware
Device Access

Host

Qemu

Guest

Programs

Kernel

Hardware
Device Access

Host

Qemu

Guest

Programs

Kernel

kvm

Hardware

Montag, 19. November 12
Device Access

What if we make this path shorter?
Device Access

Host

Qemu

Guest

Programs

Kernel

kvm

Hardware

Montag, 19. November 12
Device Access

- Host
  - Qemu

- Guest
  - Programs
  - Kernel

- Kernel
  - KVM

- Hardware
Device Access

Host
- Qemu

Guest
- Programs
- Kernel

Kernel
- kvm

Hardware
Device Access

Host
- Qemu
- Kernel
- Hardware

Guest
- Programs
- Kernel
- kvm
Exit Cost

Handle registers in KVM = speed?
Exit Cost

LAPIC ID

KVM

1500000
1125000
750000
375000
0
Exit Cost

- LAPIC ID
- PIO NOP

KVM

Montag, 19. November 12
Exit Cost

LAPIC ID  PIO NOP  HCALL

Montag, 19. November 12
Exit Cost

- LAPIC ID
- PIO NOP
- HCALL
Exit Cost

- Handling registers in QEMU doesn’t scale
- Handling registers in KVM is faster
Exit Cost

- Is it worth optimizing?
IDE Cost

Handling Port Exits

32%
IDE PIO Access

- cmd read: 17%
- status read (bus1): 4%
- BMDMA cmd read: 4%
- BMDMA cmd write: 4%
- select write: 4%
- BMDMA status read: 4%
- feature write: 13%
- lcyl write: 13%
- hcyl write: 13%
- cmd write: 17%
- BMDMA write: Rest

Montag, 19. November 12
IDE Cost

- 80% Handling Port Exits (QEMU)
- 11% Handling Port Exits (KVM)
- 9% Other

Montag, 19. November 12
IDE PIO Access

29% improvement
Disk performance

Montag, 19. November 12
Disk performance

- IDE
- AHCI
- virtio-blk
- IDE accelerated

4k read MB/s

Montag, 19. November 12
Problems

• Locking

• 2 code paths for the same logic
Conclusions

• Consider this when designing locks in QEMU

• Need a flexible framework

• Need to fix the other 80% in IDE

• Check for other devices that could benefit

• Idea: Implement PIO hcall for vhost
Questions
Exit Cost

- LAPIC ID
- PIO NOP
- HCALL

KVM
QEMU
KVM in-kernel