Peter Grehan: Graphics support in bhyve

What Is bhyve
- BSD native hypervisor

What's bhyve graphics
- Provide workstation UX

Serial port Isn't enough?
- Some users just won't use bhyve without a GUI

Windows support
- Actually works with only a serial port since Vista
- Server versions support tmux-like interface, SAC (System Administration Console)
- If ACPI SPCR table is present, Windows will do the right thing
- Unattended install required
- Needs custom install media for virtio drivers to boot on bhyve

w2k12r2 install experience

(Installer Text UI)

The UEFI Frame Buffer
- Provided linear frame buffer, worked even with no PCI adapter for it

Getting pixels to users
- bhyve is a FreeBSD base-system component, can't link with Xorg/SDL
- VNC: IETF spec

Prototyping VNC
- Started with the Cairo library rendering chars to a bitmap
- Used GPL libvnc for prototyping
  - tcpdump to see what's going on
- Implemented a VNC server from scratch
  - coredump'ed a lot of VNC viewers in the process

"fbuf" device emulation
- proprietary PCI frame buffer device
  - opposite of bhyve's "only emulate well supported devices" philosophy
- UEFI provides standard interface (GOP protocol) to guest OSes
  - bhyve already provide a UEFI implementation
- 8MB frame buffer with 32-bit pixels. Resolution can change on the fly.
- Guest frame buffer accesses are NOT emulated, but passed through
  - No instruction emulation required (would require emulating all sorts of instruction that access memory, including SIMD, otherwise)
- Host memory inserted into guest EPT map for frame buffer
  - marked as non-executable
- Guest rendering runs at memory bandwidth
  - Not through vesa/scfb bus-attached mem
  - But not-accelerated. Software rendering only.

fbuf meet VNC
- fbuf memory passed to user
- used "screen scraping" technique
  - screen is sampled every 1/30 secs (downside: tearing)
- Sending entire frame buffer in this interval requires a lot of band width
  - Run CRC over each 32x32 pixel "tile", don't send if CRC equal
    - If more than X% changed, send whole screen instead of small rects
  - Works well, doesn't burn CPU when idle
  - Some older VNC clients don't like it

    - Optionally use zlib compression if clients advertise support for it
- No CPU is used when VNC is not connected

The Mouse

- VNC provides absolute mouse coordinates
- bhyve's PS2 keyboard/mouse hooked up to VNC server
- Problem: PS2 mouse only supports relative coordinates
- guest mouse acceleration and software rendering cause weird "cursor running away from mouse" effect

Fixing the mouse

- Emulate a tablet: good match for VNC
- Hard to find generic tablet device -> had to emulate USB tablet
    - Solution: implement XHCI USB controller
        - Provides a future path for USB emulation/passthru
    - Downside: XHCI not supported in older guests like Windows 7
    - Bigger downside: XHCI still not supported by FreeBSD!

The keyboard

- VNC provides Xorg scan-codes: reasonable
- Unfortunately: mix of VNC clients and non-US keyboards creates a torrid mix

What about VGA?

- Complicated to support all modes, limited resolution
- An emulation was written but not fully enabled
    - Renders to a linear frame buffer to support output to VNC
    - Requires trapping all accesses for fidelity e.g. planar modes: slow
    - UEFI CSM has BIOS INT10 interface to support this

The Collision of VGA and UEFI

- Windows Vista, 7 and Server 2k8 requires *both* UEFI GOP *and* BIOS INT10h
    - VGA registers are accessed
- Fix: an INT10h 16-bit asm stub in non-CSM UEFI that reports required VESA BIOS info
    - partial VGA register implementation
- Unfortunately breaks OpenBSD UEFI, thinks system is VGA. (had to add option "vga=off")
- Currently forces resolution to 1024x768 regardless of config. Fixable.

Futures

- PCI passthrough of graphics adapters
    - Full-speed 3D rendering
- USB keyboard support
- Expand VNC client support
    - Fix non-US keyboard language issues
- External API for non-VNC viewers (FreeRDP, Spice)
- Worth supporting virtio graphics emulation?
    - Has retrace interrupt to avoid tearing

Acknowledgements

(Peter's slides contained a lot of screenshots specifically created for a Taiwanese audience, including installing the localized version of Windows 10, which is greatly appreciated 😀)