bhyve

past, present, future

grehan@freebsd.org

bhyvecon Tokyo 2014
2010

- Proprietary hypervisors > 1
- GPL-licensed hypervisors >= 2
- BSD-licensed hypervisors 0*

(* possibly research ones)
MeetBSD Nov 2010

- Short presentation on what NetApp would like to see in FreeBSD.
- Amongst other things: “anyone working on a type-2 hypervisor?”
  - “No”
Feasible?

• Had done extensive low-level FreeBSD x64 work with co-author for a number of years
• KVM was done with a small dev team
  • but using GPL’d Qemu as the base
  • used VT-x gen I
• Let’s give it a try
Timing wrt Intel

- Nehalem arch introduced EPT to VT-x
  - took away complexity of shadow-paging
- Real-mode support introduced in Westmere
- Available across entire product line with SandyBridge
Reducing the problem

- Limit to logical partitioning
  - Static memory assignment, hw.physmem
- Simple virtio device models for net/block
- Paravirtualized console
- x2apic, MSR access only
  - avoided instruction emulation
BSDCan 2011

- NetApp allowed code to be released
- 2-clause BSD license
- loader missing: new one written at conference (thanks to Doug Rabson)
- initial code 8.2, jhb@ ported to 9
Getting into FreeBSD

- Long period of inactivity
- Branch merged to CURRENT Jan 2013
- Flurry of activity; made it into 10.0
- Activity unabated in CURRENT
  - MFC work is time-consuming :(
What’s Ahead

• Will discuss:
  • Boot
  • Storage
  • Networking
  • Other I/O
  • “Expected” features
Boot

• Currently use user-space loaders
  • bhyveload, grub-bhyve
• Requires work for each guest o/s
  • Not scalable
• Difficult to track resource usage
Boot - UEFI

- Solution is to use Intel UEFI (aka EDKII)
  - OVMF target for Qemu
  - Modify for bhyve
- Non UEFI-capable o/s? (FreeBSD)
  - Implement “CSM” BIOS compatibility
- Allows single-process model for bhyve
Storage

• Currently support virtio-block, AHCI device models

• Only support file backend

• Futures:
  • virtio-scsi
  • “sparse” filters e.g. VMDK/Qcow2/VHD
Networking

- Currently support virtio-net
  - Single queue, no stateless offload
- Futures:
  - 82580 and e1000 device models
  - Flexible backend support: netmap, wanproxy, vhost-net like in-kernel.
Other I/O

- Video: ancient Cirrus ala Qemu.
  - Indirection needed to stay in base
- Keyboard/mouse - USB ?
- Sound ?
“Expected” features

• Suspend/resume
  • Need to serdes device and CPU state
• Live Migration
  • Live serdes of device, CPU and RAM
  • Hard :)  
• Nested operation: also hard.
Other Arch Support

- ARM
  - Cortex A7/15 have excellent h/w assist
  - Would like to get started
- MIPS/PowerPC
  - Some models have assist
  - Won’t stop anyone volunteering :)

Thursday, March 13, 2014