bhyve memory overcommit

grehan@freebsd.org

EPT recap

- Page table structure in host phys mem
- VMCS pointed to this
- Entries (almost) identical to x86 PTEs
    - 2MB/1GB super pages, NX bit
- CPU TLB tagged with ‘vpid’
- Fault results in vm exit

bhyve mem recap

- bhyve requires extended page tables (EPT)
- 1st rev: partitioned mem, fixed EPTs
  - but could use 2MB/1GB EPT mappings
- 2nd rev: dynamic mem, fixed 4KB EPTs
  - still wired

Next step

- Guest mem allocated on-demand
    - Initially empty EPT table
- FreeBSD vmspace per guest addr space
    - backed by swap, zero-fill on fault

Integration with VM

- x86 pmap code already manages TLBs
- ... so, use that for EPTs
- run-time tests to handle EPT/TLB differences

Issues #1

- Code accessing guest addr space can't assume it is present
- In-kernel instruction emulation code
  - Used for APIC emulation
  - Shifted away from critical path
- PCI passthru - h/w requires wired.
  - So just wire at VM create

Issues #2

- No EPT accessed/dirty bits (pre-Haswell)
- Emulate with r/o mappings (ala MIPS)
- Test by using a/d emulation on host

Issues #3

- VT-x event priority
- Assumed interrupt injection always succeeded
- EPT violation has higher priority
- Combo results in missed interrupt
- VMCS already had info that interrupt wasn't injected - use that

Status

- code in proj/bhyve_npt_pmap
- Might/may go into 10.0
- Extensive review, feedback, ideas and support from kib@ and alc@