SMR in bhyve

Mark Johnston
markj@FreeBSD.org

FreeBSD

bhyve Office Hours
December 16, 2020
Page table virtualization

- Component of hardware virtualization technologies
- Virtualize hardware MMU
- Otherwise VMMs must intercept guest PT updates - tricky!
- AMD: RVI/NPT, Intel: EPT
- Adds second translation table, GVA $\rightarrow$ GPA, GPA $\rightarrow$ HPA
- TLB misses become more expensive
- Introduces new microarchitectural state
TLBs

- EPT translations are cached by a TLB
- Tagged with a VPID/ASID assigned by the VMM
- VMM may need to invalidate EPT translations
  - When migrating between host CPUs
  - To track page references (pagedaemon, migration)
  - To relocate passthrough device mappings
void
invalidate_ept_translations(void)
{
    atomic_add_64(&g_eptgen, 1);
    ipi(guest_cpus);
    /* XXX guest might still be executing */
}

void
vmrun(void)
{
    eptgen = g_eptgen;
    do {
        intr_disable();
        if (g_eptgen != eptgen) {
            eptgen = g_eptgen;
            flush_tlb();
        }
    } while (handle_vmexit());
}
Safe Memory Reclamation

Refers to a family of synchronization algorithms

Implementation in FreeBSD designed for the kernel slab allocator

Readers: \texttt{smr\_enter()}, \texttt{smr\_exit()}

Writers: \texttt{goal = smr\_advance()}, \texttt{smr\_wait(goal)}
void invalidate_ept_translations(void)
{
    atomic_add_64(&g_eptgen, 1);
    goal = smr_advance();
    ipi_all();
    smr_wait(goal);
}

void vmrun(void)
{
    eptgen = g_eptgen;
    do {
        intr_disable();
        smr_enter();
        if (g_eptgen != eptgen) {
            eptgen = g_eptgen;
            flush_tlb();
        }
        VMENTER
        smr_exit();
        intr_enable();
    } while (handle_vmexit());
}