bhyve VM_MAXCPU cleanup

Rod Grimes
(Janitor)
rgrimes@FreeBSD.org
bhyvecon Ottawa, Ontario, CA 2019
CPU Topology and VM_MAXCPU

Simple number of vCPU mapped to sockets

sysctl could change Sockets and Cores

Added Sockets, Cores, and Thread
  -c 8 becomes -c sockets=2,cores=4
VM_MAXCPU

- Compile time constant
- Long standing value of 16
CPU Topology added maxcpu

- Forth, presently hidden, topology value
- Planned for future hot plugging of cpus
Completed code and reviews

- bhyve cpu topology control
- https://reviews.freebsd.org/D9930
- Add accessor for vm->maxcpus in preparation for runtime maxcpu setting
- https://reviews.freebsd.org/D18755
- bhyve acpi MADT table correction for VM_MAXCPU > 21
- https://reviews.freebsd.org/D18755
ACPI Tables

- Still statically compiled, now they are properly sized and follow the compiler constant
- Tested up to 254 vCPU
- Need to write a table builder
- Need to investigate UEFI related tables
VMM statistics tables

- Statically sized at 64 entries
- But there is a per vCPUipi statistic
- First attempt to fix would break ABI compatibility
- Punt for now, the code simply issues a warning
- New ABI in the future
Some fun locks

- Resource locks that use the last vCPU as the locking point
- Work by Joyent/Pmooney on better resource locking makes this go away later
- Continue to use the old method until then
vmm.c: struct vcpu * vcpu[VM_MAXCPU];  /* (i) guest vcpus */
amd/svm_softc.h: uint8_t apic_page[VM_MAXCPU][PAGE_SIZE];
amd/svm_softc.h: struct svm_vcpu vcpu[VM_MAXCPU];
intel/vmx.c: uint16_t vpid[VM_MAXCPU];
intel/vmx.h: struct vmcs * vmcs[VM_MAXCPU];  /* one vmcs per virtual cpu */
intel/vmx.h: struct apic_page apic_page[VM_MAXCPU];  /* one apic page per vcpu */
intel/vmx.h: struct pir_desc pir_desc[VM_MAXCPU];
intel/vmx.h: uint64_t guest_msrs[VM_MAXCPU][GUEST_MSR_NUM];
intel/vmx.h: struct vmxctx * ctx[VM_MAXCPU];
intel/vmx.h: struct vmxcap * cap[VM_MAXCPU];
intel/vmx.h: struct vmxstate * state[VM_MAXCPU];