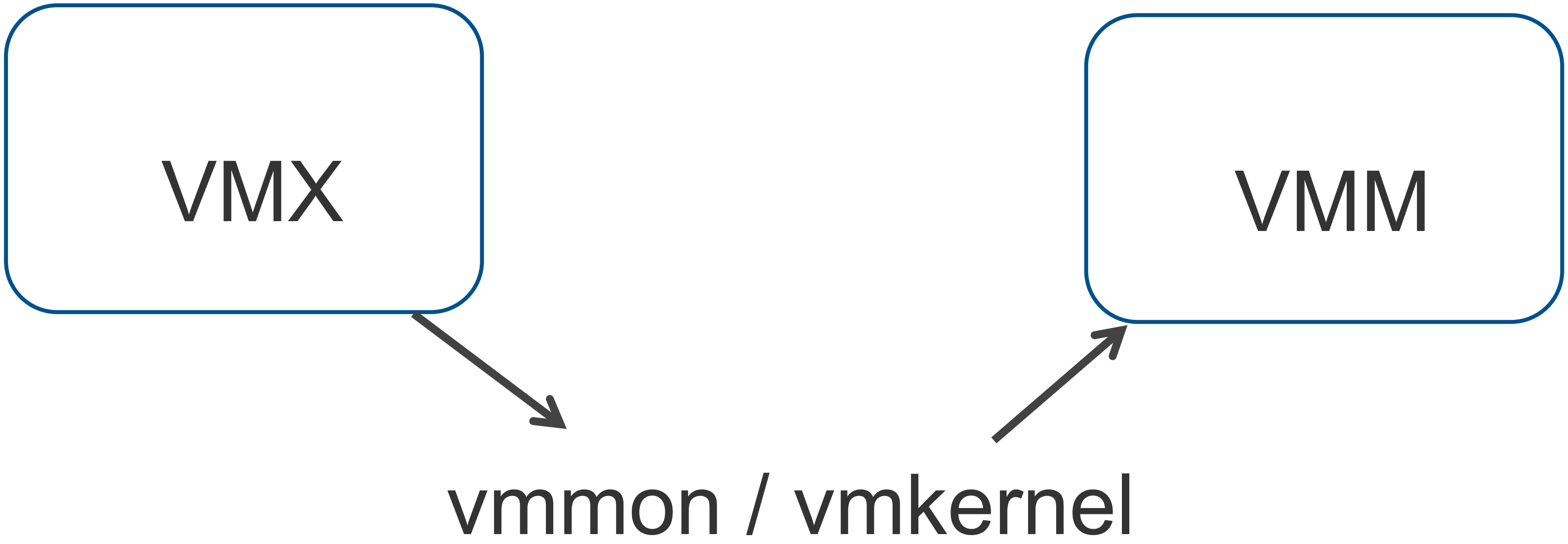
Monitor Composition

Carlos Robles

Confidential

- **From the first MonitorU talk:**

  - **VM Genesis**
    - Vmx launches, extracts vmm code/data.



VMX

VMM

vmmon / vmkernel

Outline

- Components of the monitor binary
- Linking/loading of the monitor in the VMX
- Coredumping
- Stats

What makes up the monitor?

- **Our monitor is modular.**
  - Has “extensions” or “modules”

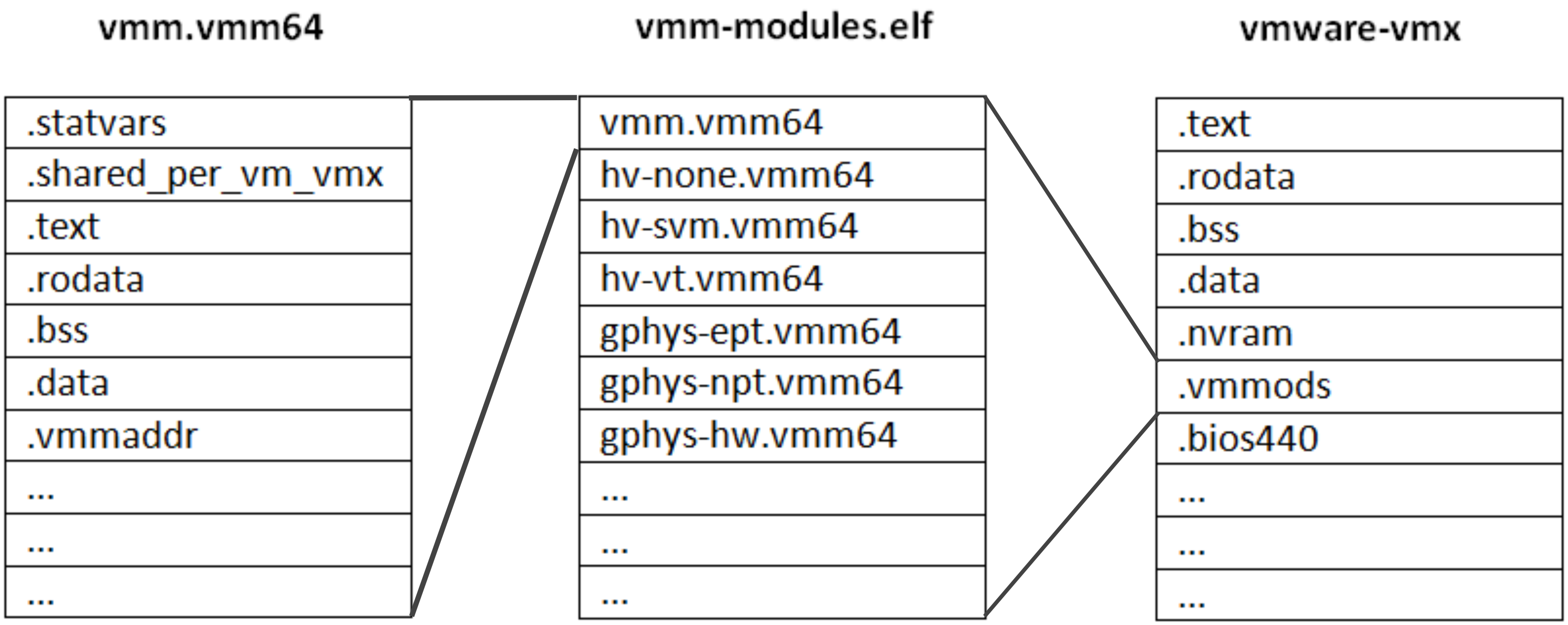
Module	Alternatives
vmm	
mmu	hwmmu, scratchas, nohv
hv	vt, svm, none
gphys	ept, npt, sw
vprobe	vprobe, none
callstack	callstack, none
(plus others)	

  - Found in the vmcore-exported directory

What’s inside a module?

- **Standard sections**
  - `.text`
  - `.rodata`
  - `.bss`
  - `.data`
- **monitor-specific sections**
  - `.shared_per_vm_vmx`
  - `.statvars`
  - `.vmmaddr`
  - `.wsbody32/64`

Where the binaries end up



vmm.vmm64
.statvars
.shared_per_vm_vmx
.text
.rodata
.bss
.data
.vmmaddr
…
…
…

vmm-modules.elf
vmm.vmm64
hv-none.vmm64
hv-svm.vmm64
hv-vt.vmm64
gphys-ept.vmm64
gphys-npt.vmm64
gphys-hw.vmm64
…
…
…

vmware-vmx
.text
.rodata
.bss
.data
.nvram
.vmmods
.bios440
…
…
…

On power-on

- **VMX fetches monitor modules from its own binary**
- **Select modules alternatives based on host/config file settings**
- **Linker links together all the modules**
  - “Monolithic” monitor binary

Monitor address space

- **Set up monitor page tables and loads monitor into memory**

- **Monitor occupies the last 64MB of the address space**
  - start address `0xfffffffffc000000`

- **Space reserved at the start of this address space for:**
  - stacks and guard pages, descriptor tables, etc.

- **After reserved area, we can map into the monitor:**
  - text and data
  - shared areas
  - statvars

Switching to the monitor

- **VMMon_RunVM() on hosted**
  - ioctl() into vmmon driver

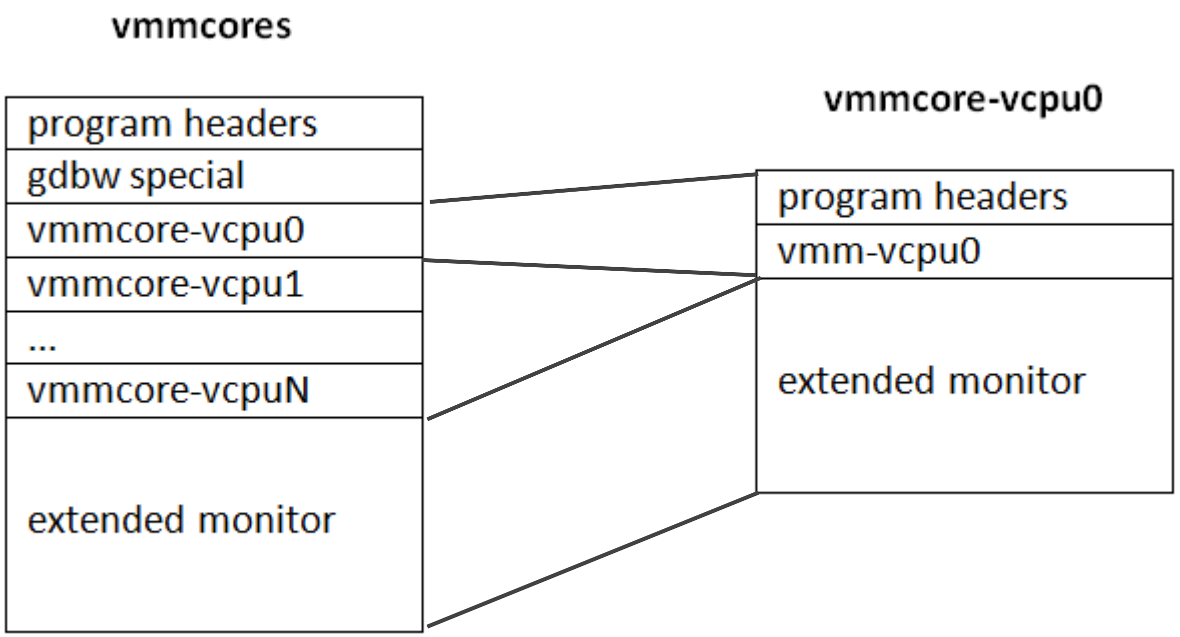
- **VMMon_SwitchToVMM() on esx**
  - syscall into vmkernel

Coredump

- One VMM thread enters Panic() and sets monPanicState
- First vcpu thread in the vmx that detects a monitor panic will dump monitor core for all vcpus.

Coredump file

- **Format: ELF within ELF!**
- **Single corefile format**



vmmcores

program headers

gdbw special

vmmcore-vcpu0

vmmcore-vcpu1

…

vmmcore-vcpuN

extended monitor

vmmcore-vcpu0

program headers

vmm-vcpu0

extended monitor

Coredump file

- **Special section to help gdbWrapper**
  - List of monitor modules that were loaded
  - Which vcpu panicked
  - ASLR info
  - build number

- **Extended monitor**
  - All anonymous pages

Corequery

- **Compiled with monitor headers to allow inspection of monitor data structures within the corefile.**

- **.vmmaddr from the monitor is embedded into the corefile**
  - Provides corequery the addresses of crucial data.
    - `&tc`
    - `&scratchASScratchCR3`
    - `mmuInfoPtr`
    - etc

- **Type “help user” into gdb (when using gdbWrapper.pl) to list available corequery commands.**

Stats

- **Old format**
  - Stats for all vcpus lived in shared area.
  - Lots of pointer arithmetic for every STAT_INC.
  - Only outputted cumulative statcounter values.
  - Used to be several hundred MBs worth of output to the log files.

- **Solution**
  - New ELF section called .statvars
  - Each vcpu maps only its own stat counters
  - No pointer arithmetic for each STAT_INC
  - Binary output

Q & A