Questions

1. _____ When focusing on improving performance in a VMM, which optimization approach should be followed to achieve the largest performance improvement?

A. Prefer optimizations that reduce the total number of exits
B. Prefer optimizations that reduce the number of instructions spent processing each exit
C. Prefer optimizations that reduce the amount of actual time spent processing each exit
D. Prefer optimizations that improve nested paging performance

9

2. _____ Why do ASIDs / VPIDs improve nested paging performance in a hypervisor?

A. When used, ASIDs/VPIDs allow caching global page mappings across all VCPUs.
B. When used, they provide a hint to the processor when TLB flushes occur
C. When used, they avoid TLB flushes entirely
D. Use of ASIDs / VPIDs does not help improve nested paging performance.

3. _____ What would happen if "external interrupt exiting" was **disabled** on a VCPU and an external interrupt was received by the PCPU?

A. The interrupt would be dispatched to the currently running VM's clock interrupt handler
B. The interrupt would be dispatched to the VMM unconditionally
C. Nothing, if the interrupt was masked
D. The interrupt would be held, and delivered during the next guest VM entry

4. _____ What would happen first if "external interrupt exiting" was **enabled** on a VCPU and an interrupt was received by the PCPU?

A. Control would flow to the currently running VM's interrupt handler for that interrupt
B. The interrupt would always be dispatched to the VM's non maskable interrupt handler
C. The guest VM would exit to the VMM (if we are in VM)
D. The interrupt would be held, and delivered during the next guest VM entry
E. Not enough information is provided to answer the question accurately

5. _____ Which of the following statements regarding IOMMUs is true?

A. An IOMMU is required when emulating PCI devices such as disk controllers
B. An IOMMU is used to enforce memory isolation when DMA is used with certain devices
C. An IOMMU is used to enforce isolation to legacy devices like the keyboard controller
D. None of the above (A, B, C) is true
E. All of (A, B, C) are true

A. An IOMMU provides safety when dealing with emulated devices performing DMA
B. An IOMMU provides safety when dealing with pass through devices performing DMA
C. IOMMUs are used for port-based I/O using IN/OUT instructions
D. IOMMUs should always be used, if available on the host
E. None of the above

6. _____ Which of the following statements about namespace virtualization is most accurate?

A. Namespace virtualization is used in all application virtualization solutions
B. Namespace virtualization should never be used in application virtualization solutions
C. Namespace virtualization means altering the name of an intercepted resource
D. Namespace virtualization is only used for file-based resources

7. _____ True/False? All CPUs that support VT-x (Intel VMX) support VT-d?
8. _____ True/False? Type 1 hypervisors run natively on the host hardware.
9. _____ True/False? Type 2 hypervisors run directly on the host hardware, without a host OS.
10. _____ True/False? If VT-d is supported on a host, it must be enabled in order to use VT-x/VMX.
11. True or *False* – All VMEXITs require the VMM to emulate at least one guest VM instruction.
12. True or ***False*** – CPUs that support Shadow Paging also support Nested Paging.
13. **True** or False -CPUs that support Nested Paging also support Shadow Paging
14. *True* or False – A VMM reloads CR3 when a VMEXIT occurs.

15. All VMEXITs require the VMM to emulate at least one guest VM instruction. ***False***

16. Assuming you wanted to emulate a serial port (COM port) in your VMM, which is the best source of information you can look at in order to make the emulation successful?

➢ A. Intel SDM
B. COM port chip datasheet
C. Other example hypervisor code which already has COM port emulation
D. None of the above

17. Place the following activities that occur during VM live migration into order, from earliest to latest. Some activities may not be used, if they are not part of VM live migration. Some activities may be used more than once. Assume a single processor VM is being migrated.

A. System administrator configures shared storage for VMs
B. Migration is paused, final pages are transferred
C. VMCS content is transferred
D. Device context is transferred
E. VMCS content is deleted

If you have a multiprocessor VM being migrated, how does that change it? You'd have to transfer a VMCS for each CPU. And delete multiple VMCSs from the original machine.

18. For each of the following exit types, provide an instruction that would cause the exit, along with any assumptions or conditions required for the exit to happen.

A. Control Register Access (Intel VMX exit reason 28) **MOV to CR**
B. Exception or non-maskable interrupt (Intel VMX exit reason 0) No instructions
C. CPUID (Intel VMX exit reason 10) CPUID
D. I/O Instruction (Intel VMX exit reason 30) **IN/OUT**
E. Access to GDTR (Intel VMX exit reason 46) LGDT, LIDT, SGDT

19. What techniques should operating system authors use to improve performance when their Oss are used in virtualized environments? Be specific, more points will be awarded based on the correctness and completeness of your answer.

➢ Use Hypercall for complex operations to avoid lot of exits
➢ Use large pages
➢ Use Paravirtualized devices because they are architected to avoid lot of exits
➢ Don't use legacy features

20. What techniques can a hypervisor author employ to reduce TLB pressure when shadow paging is in use?

➢ Use ASIDs /VPIDs which are available in both modes (shadow and nested)
➢ The TLB is really big when you touch a lot of pages. So how do you touch fewer pages?
➢ Use large pages.
➢ Avoid excessive flushing,
➢ optimize your code to avoid flushing
➢ Coalesce lots of scattered pages into larger regions that can be mapped one entry
➢ Place your code that always gets executed on the same page, with compiler and linker tricks.
➢ You can use larger pages in your hypervisor, in your nested page table.
➢ With shadow paging, your additional pages would have to match the original TLB. But not with nested paging.
➢ Also, make sure you aren't doing a lot of flushing.
➢ Another good answer is to choose a VMCS to execute next based on how much TLB flushing it would cause, regardless of its priority. Plus, if you have larger pages, paging will be faster.

21. When optimizing hypervisor performance, it has been stated that optimizing to reduce the total

number of exits provides the largest amount of performance improvement, as compared with other optimization techniques. Why is this the case?

- Exits are expensive.
- Reading/writing to GSA/HSA etc. (waste of time/ overhead)
- By reducing VM exits all processors gets benefit
- This reduces time in CR3 switch, TLB pressure etc.
- If time spent in VM exit is reduced then total time spent in processer is reduced

22. Explain how ASIDs/VPIDs improve performance when used with nested paging.

- Virtual-processor identifiers (VPIDs) introduce to VMX operation a facility by which a logical processor may cache information for multiple linear-address spaces. When VPIDs are used, VMX transitions may retain cached information and the logical processor switches to a different linear-address space.
- Reduces the need to do page table walk, because entries are kept in TLB for longer time
- It is better to provide hint to the processor what to not flush from TLB
- Optimizes the processors ability to maintain full tlb with interesting entries.

23. What problems would a hypervisor author need to overcome when creating a VM livemigration algorithm that worked across different host CPU types?

- Backward compatibility (New instructions may not work on old machine)
- If old machine is migrated to new one
- Announce limited set of features

→ diff in supported instructions, CPU types,

memory pages
registers
device states
configurations.
all needs to be
taken care of

CMPE283 – Quiz 3 – Due next class meeting.

Answer this quiz and bring a printout to the next class meeting (01 or 02 May). No email submissions this time.

Name: ___ _____

Consider the following nested paging configuration. Assume the VMM has paged in all pages.

Guest VM Page Table		
VA	PA	Page Metadata Bits
0x1000	0x12000	P A_X_ D___
0x2000	0x14000	P A_X_ D___
0x4000	0x13000	P A_X_ D_X_
0x5000	0x9000	P A_X_ D_X_
0x6000	0x6000	P A_X_ D_X_
0x7000	0x7000	P A_X_ D___

Nested Page Table		
GPA	HPA	Page Metadata Bits
0x6000	0x40000	P A_X_ D_X_
0x7000	0x4A000	P A_X_ D___
0x9000	0x44000	P A_X_ D_X_
0x12000	0x45000	P A_X_ D___
0x13000	0x51000	P A_X_ D_X_
0x14000	0x55000	P A_X_ D___

All pages in the above two page tables are already marked 'present'. For a VM executing the following function in CPL0, place an X next to the proper A/D bits as changed by the processor. When finished, you should have a set of tables with all the appropriate A/D bits marked for the sequence of instructions. For each instruction, the first argument is the "source" and the second the "destination". Each correct line in the above two tables is worth 1 point (12 points total).

```
7020 <hamburger>:
    7020:   48 8b 04 25 18 16 00 00    mov    0x1618,%rax
    7028:   48 89 c3                   mov    %rax,%rbx
    702b:   48 89 04 25 28 41 00 00    mov    %rax,0x4128
    7033:   48 09 3c 25 50 51 00 00    or     %rdi,0x5150
    703b:   48 c7 c0 20 24 00 00       mov    $0x2420,%rax
    7042:   85 38                      test   %edi,(%rax)
    7044:   83 34 25 00 61 00 00       xorl   $0x0,0x6100
```

24. What is stored VMCS and how it is used?

processor inside. Yes, it is possible by setting affinity of the VM and restricting it to a specific processor and let other processor to run in non-VMX mode. It can even be done by Linux utility called as task set which allows setting the CPU affinity process which means bonding a process to specific set of CPU.

29. Listed below are actions that occur during the boot process of a standard x86 PC. Place the items in order of occurrence from earliest to latest. Some answers may be used more than once.

A. Power applied to CPU
F. IPI sent to start up other CPUs
B. Devices/buses are probed
G. Memory controller is configured
C. Second stage boot loader is loaded
H. Boot sector read from disk
D. OS idle loop is entered
I. Kernel is loaded
E. CPU registers initialized to default values
J. First process launched

➢ A,E,B/G,H,C,I,F(OPTIONAL),B/G,F,D,J

→ Power applied to CPU
→ CPU registers initialized to default values
→ Devices/buses are probed OR Memory controller is configured
→ Boot sector read from disk
→ Second stage boot loader is loaded
→ Kernel is loaded
→ IPI sent to start up other CPUs
→ Devices ... OR memory
→ IPI ...
→ OS idle loop is entered → First process launched

30. Describe how live migration of a virtual machine between hosts works. Be specific – more points will be awarded based on the completeness of your answer.

➢ Live migration is the movement of a virtual machine from a physical host to another host without downtime. It can keep the virtual machine is still running while switching all of the physical resources. It is very useful for maintenance and loading balance. In my experience on building and maintaining the website, I got a lot of trouble on the overflow of requests from the client sides. By applying the load balance, it shared the requests to multi computers to gain more resources such as CPUs. It can help multi works can be done at the same time and increase the bandwidth which a host can handle.

➢ Next, we consider what should we move when the live migration operate. There are two basic components create the VM: VM's storage and VM's configurations. There are a lot of problem when moving VM's storage so most modern data center using NAS/SAN device instead of local disk for each server. With NAS, servers can connect with different physical disk through a LAN network. It is valuable and flexible in maintenance.

➢ So the last thing we need to consider is how to migrate the VM's configuration or migrate memory. It can be divided into three phase: (Example we want to migrate host A to host B)

Push phase: The VM in host A is still running while certain pages are pushed to host B along the network.To ensure consistency, pages modified during this process must be re-sent
Stop-and-copy: the VM in host A is stopped. Based on copied pages, the host B starts the VM
Pull phase: when the new VM is executing, if any accessed page has not yet been copied, the page is faulted Of course, we can use some other ways such as pure stop-and-copy, pure demain-migration. In addition, we can have more details on each phase like below

Answer2:
Describe how live migration of a virtual machine between hosts works.
Answer:
Live migration refers to the process of moving a running virtual machine or application between different physical machines without disconnecting the client or application. Memory, storage, and network connectivity of the virtual machine are transferred from the original guest machine to the destination.
There are some prerequisites for Live Migration:
The source and destination host should both be members of the same cluster, ensuring CPU compatibility between them. (Live migrating virtual machines between different clusters is generally not recommended). The source and destination host must be up and running .The source and destination host must have access to the same virtual networks and VLANs. The source and

30. Describe how device I/O works in a virtual machine. Be specific and thorough.

- Some device example
- Serial Port
- Turn on exiting on serial port IO port
- When guest does In/OUT it wants to access serial port registers.
- If Hypervisor can not handle this exit then I transition to some userspace program like QEMU Which then does the emulation as per the data sheet of the serial port chip we are emulating
- Register get stuffed back into the guest VM
- We have now emulated the instruction, advance instruction pointer **range** to the guest having satisfied the IO

31.Describe how Shadow Paging works. Why would a VMM author want to support processors that provide this technology?

- Shadow page tables are used by the hypervisor to keep track of the state in which the guest VM thinks its page tables should be. The guest can't be allowed access to the hardware page tables because then it would essentially have control of the machine. So, the hypervisor keeps the real mappings i.e. guest virtual address to host physical address in the hardware when the relevant guest is executing and keeps a representation of the page tables that the guest thinks it's using in the back or shadows.

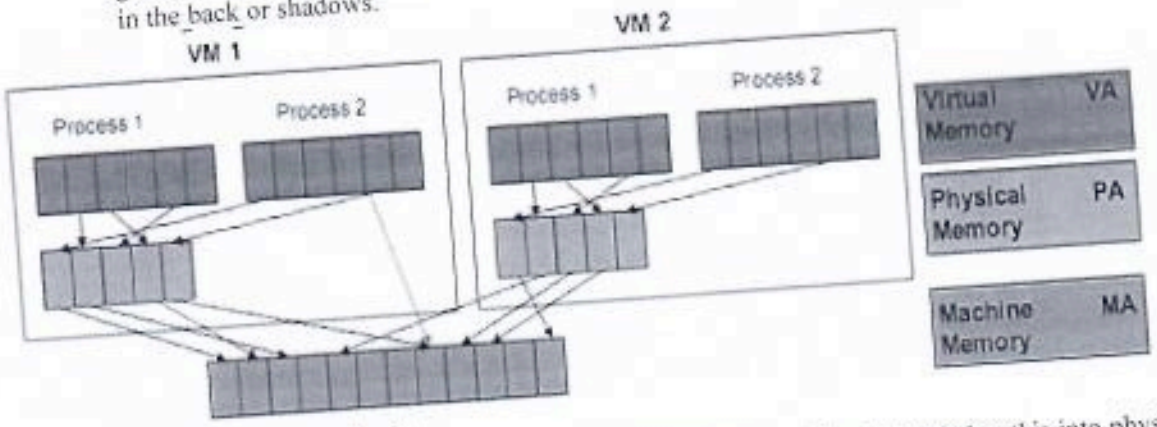

Figure 2: Shadow Paging Mechanism

- **Performance:** Managing the virtual memory of the different guest OS and translating this into physical pages can be extremely CPU intensive. Without shadow pages we would have to translate virtual memory (blue) into "guest OS physical memory" (gray) and then translate the latter into the real physical memory (green). Luckily, the shadow page table trick avoids the double bookkeeping by making the MMU work with a virtual memory (of the guest OS, blue) to real physical memory (green) page table, effectively skipping the intermediate "guest OS physical memory" step.

1. **Good:** If there is only a single CPU allocated to each VM, the Shadow paging mechanism works well as there is no contention for updating the page tables.
1. **Bad:** Each update of the guest OS page tables requires some shadow page table bookkeeping. This is rather bad for the performance of software-based virtualization solutions (BT and Para) but wreaks havoc on the performance of the early hardware virtualization solutions. The reason is that you get a lot of those ultra-heavy VMexit and VMentry calls. The performance penalty of shadow page tables gets worse as you use more (virtual) CPUs per VM.

32. Describe how Nested Paging (Intel EPT) works. Why would a VMM author embrace processors supporting this technology?

- Nested Paging: Second Level Address Translation (SLAT) or Nested paging implements memory management in hardware, which can greatly accelerate hardware virtualization since these tasks no longer need to be performed by the VMM.
- With nested paging, the hardware provides another level of indirection when translating linear to physical addresses. Page tables function as before, but linear addresses are now translated to guest physical addresses first and then to real or host physical addresses. A new set of paging registers now exists under the traditional paging mechanism and translates from guest physical addresses to host physical addresses, which are used to access memory. Nested paging eliminates the overhead caused by VM exits and page table accesses. In essence, with nested page tables the guest can handle paging without intervention from the hypervisor. Nested paging thus significantly improves virtualization

25. Describe the steps involved in handling a VM exit. (You may choose any exit type you wish in your description). Be sure to explain the various steps involved in as much detail as you can – points will be given for completeness and correctness of your answer.

- Give example of cupid

26. Describe how the TLB is used by the processor, VMM, and guest VMs in a multi-cpu host environment.

- TLB usage.
- TLB is cache that is used to store recently computed translations to avoid page table walks.
- How it is used with nested paging(same way with ASID/VPID)easy lookup
- Multi cpu when page table entry is deleted by the Vmm/guest OS, you must flush the TLB on the all CPUS that are using that page table.
- Example, quiz, example of instruction

27. Describe how LXC implements namespace virtualization, and what namespaces it provides.

- LXC is a namespace isolation / namespace virtualization technology
- LXC provides several default namespaces.
- LXC automatically manages the lifecycle of each namespace
- – When no more references to a namespace are in use, LXC destroys the namespace
- Eg, no more PIDs in a PID namespace ...
- PID Namespace
  Process ID namespace
  – Provides separate process ID trees
  Eg, "ps -ax" inside PID namespaces would only show PIDs from that namespace
- Network namespace
  – Provides separate views of underlying physical network interfaces
  – Provides separate iptables rules
  – Used to segregate network traffic between processes in one namespace from PIDs in other namespaces
  – Think – run two web servers running on the same host on the same port ...
- UTS namespace
  Hostname virtualization
- IPC namespace
  SysV UNIX interprocess communication virtualization
  Eg, "named pipes" or "shared memory"
- UID namespace
  – User ID / account virtualization
  – Think – same user ID in use across different processes
  – UIDs get mapped to nonconflicting UIDs on the Host
- Mount namespace
  – Used to change the view of mounted filesystems, or to mount new filesystems in the VFS tree
- Other namespaces
  – kmemcg – limit kernel memory use for specific processes
  – Etc..

28. In a dual or multi-CPU configuration, is it possible for one CPU to be operating in VMX mode (root or non-root) and another (or other) CPUs to be executing in legacy (non-VMX) mode? Explain, and cite your reference.

- This refers to multi-core technology that allows single processor to have more than one physical