CSE 121: Homework 4

Due: Never

1. **Disco**

   (a) In a non-virtualized kernel, the OS is responsible for loading TLB entries on a TLB miss (when using a software-loaded TLB). However, with a VMM, the OS no longer has that privilege. Explain what occurs when a user-level process wants to issue an instruction that causes a TLB miss.

   The process gives a virtual address, which is not in the TLB. The OS converts that virtual address to a physical address and attempts to load the TLB. This will cause a trap to the VMM, which then puts a virtual to machine address mapping in the TLB.

   (b) One of the motivations of Disco was to allow commodity operating systems to take advantage of new hardware, particularly multiprocessor machines. The Disco authors attempted to sidestep the issue of modifying the operating system by introducing their own device drivers. How were they able to optimize performance by using device drivers?

   By writing a network device driver they could detect network communication to the same physical machine. Instead of sending it over the network, they could just remap memory. By writing a disk driver, they can detect when two operating systems are reading the same files and share the memory.

2. **Xen**

   (a) Paravirtualization provides a different abstraction from that of the underlying hardware. The idea is that better performance can be achieved at the expense of requiring modifications to the OS. The Exokernel also tries to provide an abstraction of the hardware with performance in mind. Which entries in table 1 are similar to ideas from the Exokernel?

   1. Paging: the Exokernel separates authorization from usage through secure bindings. In case you were wondering, a hardware page table is simply a memory data structure that the hadrware knows how to interpret. 2. Protection: libOS is the "guest" that runs in application space (with a lower privilege than the Exokernel). 3. System calls: Fast handlers are like application safe handlers (user downloaded, but checked).

   (b) Xen introduces an asynchronous I/O ring for communication between a guest OS and an I/O device, but there is no guarantee that they are serviced in order. What previous systems have we looked at that might run into trouble with this, and how might you modify the asynchronous I/O ring to support this?

   Journaling file systems need to flush the write-ahead log before any other operations. Similarly, Soft Updates requires flushing disk pages in a specific order to maintain file system consistency. Xen implements "reorder barriers", which they don't describe in detail. Because each request has a unique identifier, you could add an ordering field for each identifier.

3. **VMWare**

(a) ESX uses a balloon module device driver to punt the policy decision of deciding which VM and which page to page out. Louis Reasoner claims that he can create an application-level balloon that does the same thing by repeatedly calling malloc() when ESX needs more memory. Assuming his application still has a private communication channel with ESX, why will his idea still not work?

The balloon driver works by allocating physical memory (not virtual memory) and then pinning it. Only a kernel service or device driver can do this. This is necessary because it must communicate to ESX what the physical pages are, so that ESX can deallocate the correct machine pages.

(b) The idle memory tax in ESX allows VMs that are more memory intensive to borrow pages from VMs that aren't utilizing their pages. Suppose you are working on project 2 and the project is due in 6 hours. You've conveniently taken the OS mantra of "be lazy" to a literal extreme and have no draft or data. You think you can do both at once by running VMWare with two guest operating systems: Windows and Linux. Assume you are writing your paper in Word, and talking on Gmail in the Windows system, and assume you are writing some code, compiling, and running experiments in the Linux system. What kind of tax rate would you set for this kind of workload and why?

Because the project is due in a few hours, you may want the tax to be high. You'd like to have the experiments finish faster while you think about how to outline your paper. When not running experiments, you might want Excel to have more memory to plot figures. Having pure memory isolation does not seem too useful because none of the tasks require very low real-time latency.

4. **Mixed Bag**

(a) Mallory is still writing malware, but this time it's on your favorite conventional OS (e.g. Linux, Windows, BSD, etc). She knows that anti-virus vendors are testing malware inside a VM, and wants to disguise her behavior in these situations. How can Mallory detect she is running inside a VM?

Measure times to do certain operations. Probe for extension instructions. Other clever solutions also possible.

(b) Are virtual machine monitors microkernels done right?[1]

See paper.

(c) Intel-VT[2] adds hardware support for virtual machine monitors in the x86 architecture. The aim was to improve performance by reducing the amount of software emulation that had to be done in the VMM. What did Intel do to achieve this?

See paper.

[1] http://www.usenix.org/events/hotos05/final_papers/full_papers/hand/hand.pdf
[2] http://download.intel.com/technology/itj/2006/v10i3/v10-i3-art01.pdf