DISCO
(a) A major part of the Disco paper was hiding the NUMA-ness of FLASH from the commodity operating system. How does Disco achieve this?

Cache misses should be satisfied by local memory instead of remote to avoid extra latency. Disco does this by watching the cache miss counters to identify pages that are being accessed remotely, and acting accordingly. The appropriate response may be migration or replication depending on the type of sharing involved, or even doing nothing if there is extensive write sharing.

(b) Disco uses direct execution (runs on the hardware without modification) for most operations to achieve reasonable performance. Unfortunately this cannot be done to some services. For example, to virtualize memory, Disco maintains a set of physical to machine address mappings and performs the necessary translation by changing the MIP's software TLB. Two problems that occur from this approach are:

- Some kernel segments on MIPS operating systems are traditionally direct mapped like KSEG0.
  Adjust the source and recompile the kernel so that all memory accesses are virtual, even within the kernel.
- TLB misses are now both more frequent and more expensive since the TLB must be flushed on virtual CPU switches.
  They used a second level software TLB for recent virtual to machine translation. Now on a TLB miss, Disco consults the second level software TLB first before forwarding the TLB miss exception to the client operating system.

XEN
(a) Louis Reasoner believes that he can use a lottery scheduler for CPU scheduling in XEN because he can implement the BVT with tickets. Louis believes he can simulate what BVT does when a process is unfairly scheduled, by temporarily removing tickets from that process. Is Louis correct? Explain.

Louis is incorrect because removing tickets from the process just gives it a lower chance to run again. In BVT after the process is unfairly scheduled, it should have the time it uses unfairly subtracted to its interval the next time it runs. The lottery tickets have no way to control the time interval of the process.

(b) DISCO uses shadow page tables because of the software loaded TLB in MIPS. The shadow page tables kept track of physical addresses to machine addresses. Does XEN need a shadow page table to keep track of physical addresses to machine addresses like DISCO? Explain.

XEN ports the guest operating system to deal with machine addresses directly. It does not need to deal with physical addresses to machine addresses because it did not move memory around like DISCO did to deal with its NUMA-ness. Xen also needs to make sure the page table entries being inserted by the guest OSes are valid, otherwise they could violate the protection between VMs.

VMWare
(a) ESX goal is to run unmodified operating systems workloads that over-commit their memory. One way that they accomplish this is by using a idle memory tax to efficiently utilize memory. Does this remind you of another paper that we read? (HINT: Think about a
GMS had similar goals in utilizing idle memory, in their case in a workstation cluster. Like the idle memory tax, GMS determines idleness by adding more weight to global pages in workstations because they represent the workstations idleness (they are not using their memory which is why other workstations are hosting their pages on them). Another similarity is checking for idleness. ESX samples pages at random to check for idleness by invalidating the TLB cache mappings. GMS flushes the TLB every minute so that when a TLB handler performs a virtual-to-physical translation on a TLB miss, it sets a bit for that physical frame which gets sampled for LRU statistics.

(b) What is the double paging problem and why is it bad?

The double paging problem happens when the VMM somehow knows what page the OS is going to swap out, call it X, and it swaps it out for the OS. If the OS is under memory pressure it will probably try to get rid of X itself so it tries to read X. This causes the VMM to bring X back in for the OS only to have the OS swap it back out (OS reads and then writes it to disk).