I. From: Algorithms for Scalable Synchronization on Shared-Memory Multiprocessors
1. What is the biggest problem created by most busy-wait techniques for mutual exclusion?
   * Busy-wait creates large amounts of memory and interconnect contention performance bottlenecks that get more pronounced as applications scale.

2. How does a Ticket Spin Lock work?
   * Two counters are used, the first contains the number of requests to acquire the lock, the second is the number of times the lock has been released. The lock is acquired by performing a fetch_and_increment operation on the request counter and waiting until the result (its ticket) is equal to the value of the release counter. The lock is then released by incrementing the release counter.

3. What is the key to a spin lock algorithm that won't induce memory and interconnect contention?
   * Every processor must spin on a separate and "locally-accessible" flag variables and for some other processor to terminate the spin with a single remote write operation.

II. From: Virtual Power: Coordinated Power Management in Virtualized Enterprise Systems
1. What are the two main motivational factors for online management of server systems power?
   * The power provisioned per rack in a datacenter can become a bottleneck with denser and more powerful computing systems.
   * The cost associated with powering and cooling these systems.

2. What is the key difference between "hard" and "soft" power scaling?
   * Hard power scaling is handled by the physical hardware while soft power scaling is handled in software code.

3. What is the key problem with relying on hard power scaling only in a virtualized environment?
   * On any given piece of hardware with multiple virtual machines running on it "hard" scaling can only be activated/engaged if ALL of the virtual environments are not using it.
III. From: Performance and Scalability of EJB Applications

1. What is the most important factor in determining the performance of a J2EE EJB application?
   * Application Implementation Method

2. What are the two different container designs that represent the majority of EJB containers available and what is their key difference?
   * Dynamic proxy approach and pre-compilation approach. Both use reflection to create the necessary classes but as the name suggests pre-compilation happens during application deployment (compilation) and in the case of the dynamic proxy approach it is done during runtime.

3. In what two instances does the session facade pattern improve rather than degrade performance for entity beans?
   * Local communication is very efficient.
   * EJB 2.0 Local interfaces are used.

IV. From: Xen and the Art of Virtualization

1. What is meant by "paravirtualization"?
   * A virtualization technique that presents a software interface to virtual machines that is similar but not identical to the underlying hardware.

2. The paravirtualization technique requires ________ changes to the guest operating system source code.
   * a) no
   * b) few (answer)
   * c) many

3. Why is isolation important in the design of a virtual machine manager?
   * Isolation is important so that one machine does not affect another.

4. Paravirtualization ________ require applications running in the guest OS to be ported?
   * does
   * does not (answer)
V. From: Lessons from Giant-Scale Services

1. What are the three key real world challenges faced by giant-scale services?
   * high availability
   * evolution
   * growth

2. What is the key disadvantage of a "round robin" type load balancer?
   * The requests for service are distributed evenly by count but without regards to the load of the servers the requests are being sent to so some machines may be overloaded.

3. Why are "yield" and "harvest" more important than "uptime" when talking about availability?
   * Uptime doesn't take into account the impact to users.

4. A replicated system maintains 100% ________ under fault.
   * yield
   * harvest (answer)

5. List and briefly explain the three approaches to deploying upgrades in a giant-scale system.
   * Fast reboot = all nodes are taken offline, upgraded, and brought back up at the same time.
   * Rolling upgrade = nodes are upgraded one at a time.
   * Big Flip = half the nodes are upgraded at a time.