

Principles of Computer Systems

December 16, 2025

This exam has 17 questions totaling 100 points. You have 105 minutes to answer them. If you find yourself spending on any given question more minutes than the number of points attributed to it, you might consider moving on to the next question. The exam has 12 pages in total, including this one.

There is a mix of four kinds of questions:

- *ONE*: One-sentence answers related to a system or concept discussed in class. ONEs allow you to demonstrate your ability to identify, synthesize, and clearly state the essence of what you have learned.
- *OPAR*: One-paragraph answers related to a system or concept discussed. These are similar to ONEs, but with an opportunity to include more ample details and go in more depth.
- *BOE*: Back-of-the-envelope calculations that assess your ability to perform the quantitative reasoning that underlies intelligent design decisions.
- *MCQ*: Multiple-choice questions where you must circle the letter of one or more of the provided answers. No justification is required, and nothing other than the marked answers will be taken into account during grading. The MCQs enable us to evaluate the breadth of your knowledge, covering many topics in a short amount of time.

Your answer to each question must be entirely within the box provided on the exam sheet. You have a few spare sheets at the end of the exam. Anything outside the boxes will be discarded prior to grading.

The exam takes place in room MA A1 10, and you must be physically present to take it. The exam starts at 11:15 and ends at 13:00. The exam is written, and you need to use your own pen. It is your responsibility to write legibly – the POCS staff reserves the right to ignore all or part of your answer if it cannot be easily read.

Rules:

- The exam is closed-book. You are allowed to bring with you one double-sided A4 cheat sheet with whatever information you want on it, in whatever font size you like.
- You are not permitted to interact with or receive or give any assistance for the exam except with/from the course staff.
- If you are noise-sensitive, you may wear ear plugs or ear muffs, but no electronic device is permitted (no headphones, no earbuds, no active noise-canceling device, etc.).

Read each question carefully. You need to provide a correct and complete answer to the *correct* question in order to receive full credit. A correct answer to a *wrong or misinterpreted* question will not earn credit. If you have any doubts, raise your hand, and the course staff will come to help.

Question 1 (3 points): VIPT caches typically achieve lower access latency than PIPT caches by:

- A. Using a fully associative cache organization.
- B. Reducing the critical-path impact of address translation.
- C. Eliminating the need for a separate Translation Lookaside Buffer (TLB).
- D. Allowing the cache index lookup to run in parallel with address translation (TLB lookup).
- E. Performing cache indexing using virtual address bits while tagging with physical address bits.

Question 2 (3 points): The main advantages of organizing system modules using a client/server architecture communicating via messages include the following:

- A. Eliminating memory allocation overheads.
- B. Reducing state sharing between modules.
- C. Executing more securely, in strongly isolated domains.
- D. Eliminating the need for interface documentation.
- E. Guaranteeing exactly-once execution semantics for operations.

Question 3 (3 points): A key difference in motivation between lazy execution and speculative execution is:

- A. Lazy execution primarily seeks to reduce latency, while speculative execution seeks to guarantee correctness.
- B. Speculative execution guarantees termination, while lazy execution does not.
- C. Speculative execution relies heavily on the scalable commutativity rule, whereas lazy execution does not.
- D. Lazy execution aims to reduce total work by avoiding unnecessary computation, while speculative execution uses potentially unnecessary work to reduce latency.
- E. Only lazy execution requires transaction mechanisms to ensure correctness.

Question 4 (2 points): The primary purpose(s) of using abstraction in computer systems design is/are to:

- A. Reduce the overall amount of code needed.
- B. Make emergent behavior more predictable by separating "what" from "how".
- C. Ensure that the system operates in a fail-safe mode.
- D. Enable segmentation-based protection domains implemented in hardware.
- E. Enable components developed by different people to interact through standard interfaces.

Question 5 (2 points): The principle of "if you can't name it, you can't use it" highlights the role of naming and indirection in:

- A. Improving network locality.
- B. Reducing context switching overhead.
- C. Controlling resource visibility.
- D. Ensuring atomicity during concurrent operations.
- E. Enforcing strong consistency.

Question 6 (2 points): Network Address Translation (NAT) is considered a layering violation because the NAT gateway:

- A. Changes the IP address, which should only be handled by the application layer.
- B. Duplicates congestion control logic already present in TCP.
- C. Assumes that it understands the semantics of the TCP header (e.g., port numbers).
- D. Routes packets based on IP prefixes instead of MAC addresses.
- E. Forces packets to follow a path that violates BGP policies.

Question 7 (2 points): How does the Resilient Overlay Network (RON) improve upon traditional Internet routing protocols?

- A. By allowing RON nodes to detect and route around certain network failures within seconds.
- B. By replacing BGP with a more efficient routing protocol.
- C. By using more aggressive route aggregation and update bundling than BGP.
- D. By increasing the default packet size.
- E. None of the above.

Question 8 (3 points): The Akamai paper describes the preferred mitigation strategy against volumetric Distributed Denial of Service (DDoS) attacks to be generally:

- A. Implementing hop-count filtering on application servers.
- B. Immediately withdrawing BGP advertisements to shift all traffic away from the target PoP.
- C. Absorbing the attack traffic at the Point of Presence (PoP) by utilizing overprovisioned bandwidth and filtering irrelevant traffic at the firewall.
- D. Immediately activating loyalty filters to penalize non-allowlisted IP addresses.
- E. Starting input-delayed nameservers with stale data.

Question 9 (3 points): RAID-5 improves upon RAID-4 by mitigating limitations related to parity management and write performance. Specifically, RAID-5 addresses which of the following issues present in RAID-4?

- A. The single parity disk becoming a performance bottleneck during small write operations.
- B. Uneven load caused by concentrating all parity updates on one disk.
- C. Reduced write parallelism due to mandatory parity-disk updates.
- D. The inability to tolerate a single disk failure.
- E. Excessive parity computation overhead compared to other RAID levels.

Question 10 (2 points): According to the CAP theorem, in a distributed system that must tolerate network partitions (P), designers must choose between:

- A. Throughput and Latency.
- B. Durability and Isolation.
- C. Consistency (single up-to-date copy) and Availability (readiness for updates).
- D. Harvest and Yield.
- E. Atomicity and Durability.

Question 11 (4 points): During the execution of a transaction in a system (microservice, database, etc.), the state of the system is modified. What must be true of the system's state if it is to provide Durability (as in "the D in ACID")?

(Answer in 1 sentence)

Question 12 (5 points): How does Akamai's two-tier delegation system help its operation?

(Answer in 1 sentence)

Question 13 (6 points): Define the locality principle in working set analysis.

(Answer in 1 sentence)

Question 14 (15 points): Consider a system for memoizing the results of computing an expensive pure function f . The domain of f (i.e., the set of all possible inputs it can take) or, equivalently, the set of corresponding results that needs to be memoized, has 2^{30} elements (i.e., f takes 30-bit inputs). The system has a single client, which will compute f for at most 128 distinct inputs. The client should then be able to access memoized results for these inputs with low latency.

Design a lookup table to store memoized results of computing f . Use page tables as an inspiration for your design: the lookup should rely only on indexing, not search or complicated arithmetic, and it should not make more than two memory accesses to return memoized results.

Propose a design that minimizes the worst-case amount of memory occupied by the lookup table to memoize all 128 results. Describe your design and compute the worst-case memory consumption. Assume pointers are 64-bit long, and the type of f 's return is 32-bit long.

You can assume that the system does not support virtual memory: there are no page tables, no TLB and no MMU. All instructions reference physical memory directly.

(Write the answer in the small box below, and optionally provide your calculation to justify it.)

Answer:

Calculation (optional):

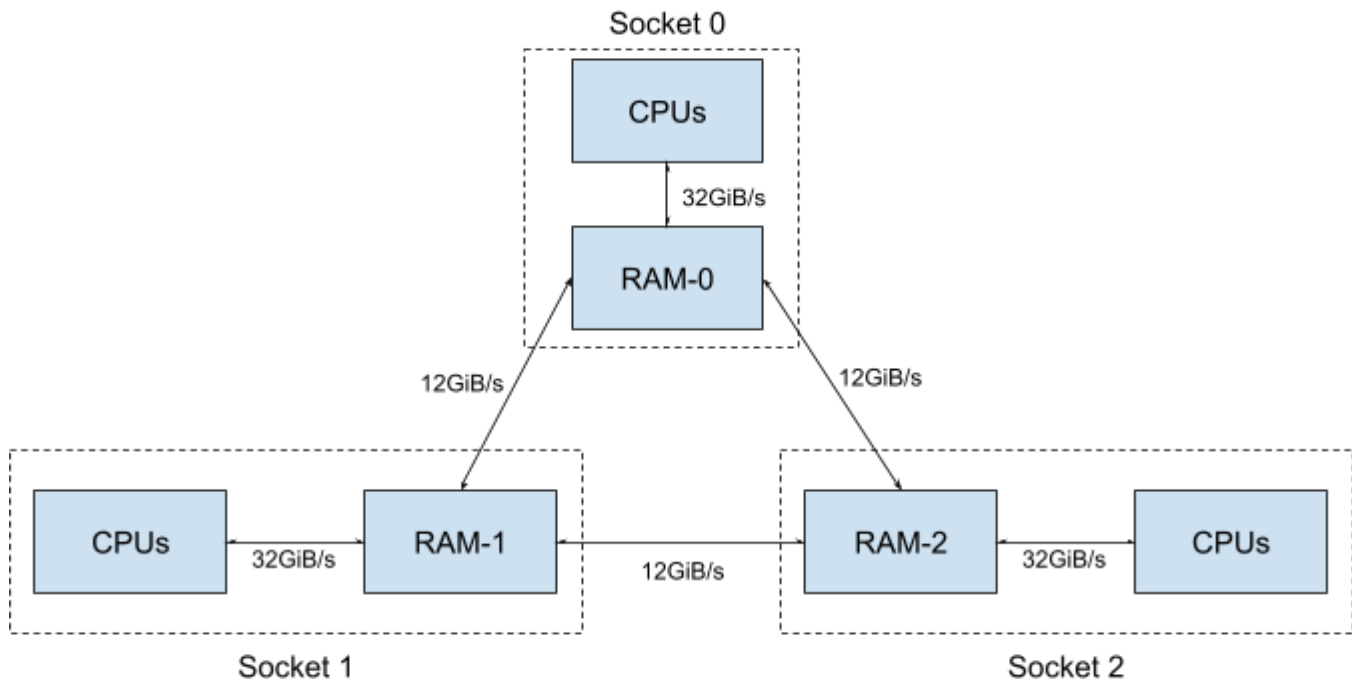
Question 15 (20 points)

You are running an all-reduce sum operation on a machine with 3 sockets. The local RAM on each socket contains an array of 2^{31} integers, each 64-bit. Operationally, the all-reduce sum is an element-wise sum across the three arrays. After the operation, the local RAM of each socket should contain a copy of the array containing the all-reduce sum.

For example, if the three sockets initially store the 2-element arrays [1, 2], [2, 3], [3, 4], respectively, then, after the all-reduce sum, each should store [6, 9] in their local RAM.

Each pair of sockets is connected via a bi-directional link, with 12 GiB/s bandwidth per direction. The CPUs are connected to their local RAM with bi-directional links as well, with 32 GiB/s bandwidth per direction. You can assume that the RAM itself has unlimited size and bandwidth. The CPUs in each socket collectively support 2^{31} 8-byte integer additions per second. All links have negligible latencies.

Your goal is to minimize the time required to complete this all-reduce sum operation across the three arrays. Describe how you would orchestrate this computation, explicitly mentioning where computation happens, and how and when data is moved (between sockets, and between CPUs and their local RAM). Compute the end-to-end latency of the all-reduce sum using your orchestration.



(Write the orchestration description in the box below, then the end-to-end latency answer below that, in the small box, and then optionally provide your calculation to justify your answer.)

Orchestration:

End-to-end latency:

Calculation (optional):

Question 16 (10 points): Remote Procedure Call (RPC) is often described as a "leaky abstraction". Why?

(Answer in maximum 6 sentences)

Question 17 (15 points): Large-scale LLM training workloads exhibit massive parallelism and mostly regular, predictable memory-access patterns. Suppose you design a custom accelerator chip with 1 million simple cores. Would it be appropriate to use a traditional hardware cache-coherence protocol to maintain consistency among the cores' caches? Why or why not? If not, what alternative cache or memory-management approach would be more suitable, and why?

(Answer in maximum 10 sentences)

----- **END OF EXAM** -----

(You can use the following pages for scratch notes. They will be discarded prior to grading.)

