CS 6210 Spring 2024 Test 2

(120 min Canvas Quiz)

Max Points:104

Distributed Systems [30 points] *2*

Lamport's Logical Clock [8 points] 2

Lamport's ME Lock [4 points] 3

RPC Latency limits [10 points] 3

Active Networks [8 points] 5

Distributed Objects and Middleware [15 points] *6*

SPRING OS [7 points] 6

Enterprise JavaBeans [8 points] 7

Distributed Subsystems [39 points] *8*

DSM [10 points] 8

GMS [15 points] 10

DFS [14 points] 11

Potpourri [20 points] *13*

Distributed Systems [30 points]

Lamport's Logical Clock [8 points]

1. (8 points) A student has implemented a distributed algorithm using Lamport's happened-before relationship to timestamp events. The student is in the middle of debugging the program, and observes the following activities in the program:

P1's activities	P2's activities	P3's activities
E1: msg-send (to P2)	E5: local event	E9: msg-receipt (from P2)
E2: local event	E6: msg-receipt (from P1)	E10: local event
E3: msg-receipt (from P2)	E7: msg-send (to P3)	E11: msg-send (to P1)
E4: msg-receipt (from P3)	E8: msg-send (to P1)	

Please give the causal relationship between the following pairs of events with reasoning. (No credit without reasoning)

(a) (2 points) E1 and E10?

Answer: E1 -> E10. E6 will wait for msg from E1. E7 happens after E6. E9 will wait for msg from E7. E10 happens after E9.

(b) (2 points) E2 and E8?

Answer: E2 || E8. E2 will happen after E1. E8 will happen after E7. Since E2 and E8 are independent, they will be concurrent.

(c) (2 points) E8 and E11?

Answer: E8 || E11. E8 will happen after E7. E11 will happen after E10. Since E8 and E11 are independent, they will be concurrent.

(d) (2 points) E2 and E11?

Answer: E2 || E11. No causal relationship between these two events.

Lamport's ME Lock [4 points]

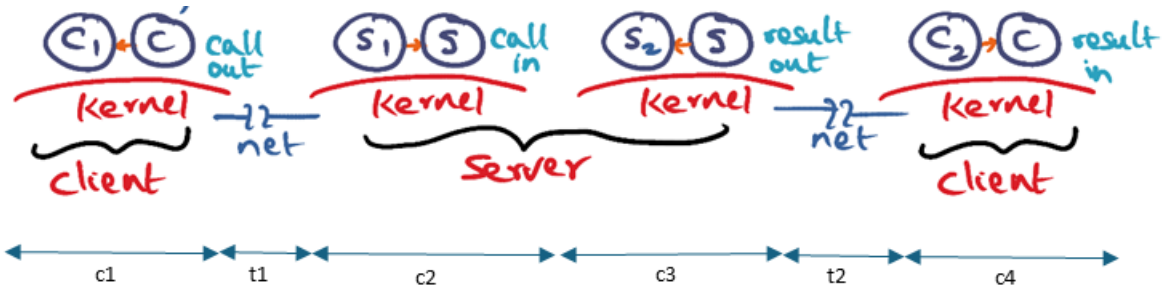
1. Give 4 conditions on which the correctness of the ME lock relies on. [4 points]

Answer:

- Lamport's logical clock
- A way to break a tie when timestamps are the same
- No loss of messages [+1]
- Messages going in order between nodes. [+1]

RPC Latency limits [10 points]

1. [5 points]



In the context of this question and its subparts, consider process context switches and transmission on physical links as the only sources of latency in RPCs (any other source of latency is negligible).

The figure above shows the timeline of a simplified RPC. Any time period demarcated by 'c' followed by a numeral represents the latency introduced by the corresponding context switch for the associated arrow. Similarly, any time period demarcated by 't' followed by a numeral represents the latency introduced by the corresponding transmission over the physical link for the associated arrow.

For this question and its subparts, only use variables provided in the above model of the RPC.

(a) If the context switches not in the critical path of the RPC latency are overlapped with other activities in the critical path of the RPC latency, give a mathematical expression of the expected RPC latency. (2 points)

Ans. Latency = t1 + c2 + t2 + c4 [+2 points]

(b) Can we further optimize the RPC latency? If yes, give the mathematical expression of the optimized RPC latency and the

mathematical condition under which such an optimization would be justified. If no, provide appropriate justification. (3 points)

Ans.

Yes, we can further optimize the RPC latency by keeping the client process spinning on the response from the server rather than context switching the client process out.

Latency = t1 + c2 + t2 [+1.5 points]

Condition to justify spinning rather than context switching at client: c1 + c4 > t1 + c2 + t2 [+1.5 points]

2. (3 points)

(a) How does the server know that an incoming RPC call is not a duplicate?

Ans.

Each RPC call from a given client will have a sequence number. (+1)

(b) What is the purpose of the server-side buffering of the RPC response in a reliable LAN setting?

Ans.

Though message losses and message corruption are rare in a LAN setting, there could still be situations wherein a duplicate RPC call can come to the server. (+1)

(c) How long will this buffered RPC response be kept on the server?

Ans.

Until the next RPC call from the same client (+1).

3. (2 points) Thekkath and Levy suggest having a shared descriptor between the kernel and the RPC client stub for reducing the number of copies in marshalling the arguments of the RPC. The alternative is for the client stub to be downloaded into the kernel. What is the downside of this alternative?

Ans.

Since the client stub is likely to be provided by a third-party service (not written by the OS developer), and will reside within the kernel, it increases the vulnerability of the OS for malicious attacks. [+2 points]

Active Networks [8 points]

You have been given the design of the capsule that will enable the intermediate routers to execute the code for incoming packets for Active Networks. As discussed, if a node receives a capsule whose TYPE has not been seen before, the node will request to load the code from the PREV active node.

1. (2 points)
(a) Why is it possible for the PREV node to not have the code corresponding to the TYPE field?
(b) How often will this occur (justify your answer)?

Answer:
(a) The soft store at an active router is a limited "cache" and stores the "breadcrumbs" of multiple flows that passes through this router. When new flows hit this router, the breadcrumbs of old flows may have to be evicted. (+1)
(b) Not very often. The most likely reason is perhaps that the packet got misrouted through the network and arrives at an active node much later after the flow itself has terminated. (+1)

2. (2 points) Why does it make sense to "drop" the packet if the PREV node does not have the code?

Answer:

The PREV node not having the code is indicative that the associated network flow is quite old and this packet itself may be a result of misrouting of a network flow that has already terminated. (+1)

Even if that's not the case, the higher-level layers (such as transport) in the protocol stack will take the necessary action of retransmitting the "lost" packet. (+1)

3. (4 points) Software Defined Networking (SDN) is often considered as the modern-day evolution of the Active Networks. Give two features of SDN that overcome the drawbacks of Active Networks.

Answer: (any two of the following gets full credit)
- SDN uses a central controller that programs the switches ONE TIME at the beginning of a new flow with rules for packet routing for that flow
- The switches do not execute code but use table look up in hardware to route the packets for each flow.
- The hardware routing in the switches ensures being able to keep up with the line rate of the network flows.

Distributed Objects and Middleware [15 points]

SPRING OS [7 points]

1. [7 points] Assume you are managing a cluster of nodes in a datacenter that is using Spring OS as the network OS. The current deployment looks like this:

   A. Web servers replicated on 3 nodes
   B. SQL Database servers replicated on 2 nodes
   C. File server on a node accessed only by the web servers.

Each of the above servers are hosted on distinct nodes. The clients are expected to make requests to both the web servers and database servers.

i. [3 points] You wish to add a web-server-load-balancer that balances the load of the client requests across the web servers. List the subcontracts needed to effect this architectural change.

Answer:
- Subcontracts on the client machines to direct the webserver requests to the web-server-load-balancer (+1)
- Subcontract on the web-server-load-balancer to receive the client requests (+1)
- Subcontract on the web-server-load-balancer to forward client requests to the webserver (+1)

ii. [2 points] While all clients can read from the database servers, only certain clients are allowed to write to the database servers. How can you accomplish this?

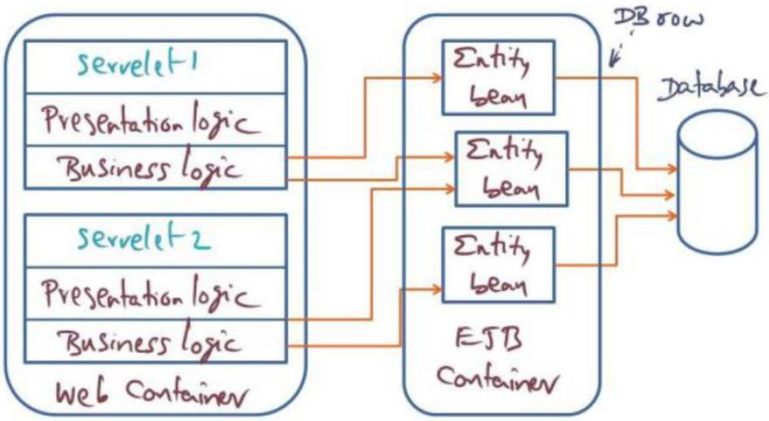
Answer: Maintain an ACL accessible to a front object in the database servers. The requests from the clients are first handled by the front object which performs the access checks before allowing the execution of the request.

iii. [2 points] Your colleague has developed a new custom communication protocol to access the file server and wants to test it. How can he leverage the extensibility of Spring OS to perform his tests?

Answer: Create new proxy objects in the file server and a test client that uses this custom communication protocol to do the testing. No changes are needed in the nucleus or any other part of the OS.

Enterprise JavaBeans [8 points]

1. (4 points) Consider the following design alternative for organizing an N-tier application using EJB.



(a) (2 points) Describe one benefit of structuring an application this way.

Answer: More parallelism/concurrency, since each entity bean can represent a different portion of the data and servlets can make parallel requests to multiple entity beans to get the data they want.

(+2 for parallelism or some notion of increased performance, as long as its supported by some form of explanation)

(b) (2 points) Describe one disadvantage of structuring an application this way.

Answer: It may be a security concern that the business logic is now exposed; it is exposed outside of the corporate network. (+2)

2. (4 points) Imagine you are the designer of an airline booking portal like expedia. You have been tasked with building a multi-tier software architecture for this portal. Enumerate and describe the tiers you will have in your service (concise bulleted list please). [Hint: You have used such services; now you have to turn around and ask yourself how you will design such a portal given all the knowledge you have gained through this course.]

Answer (needs some refinement, in particular web access to the portal):

- Persistence of client sessions to provide search history or recently viewed itinerary
- Provide transaction semantics and atomicity of operations to client for booking a flight ticket
- Ensure security to preserve financial/PII data of the users

- Increase concurrency (across simultaneous client requests and contacting airline servers) while servicing client requests for public data

(+1 for each bullet that conveys some sense of the above; this list isn't necessarily exhaustive)

Distributed Subsystems [39 points]

DSM [10 points]1. [10 points] Consider the following sequence of actions in the following time order happening in a Treadmarks DSM program. Assume a clean copy of X is with the owner at the start of the program, and the program starts execution at time T1. T2 and T3 represent increasing time order.

T1: Process P1:

acq(L1)

modify X

rel(L1)

T1: Process P2:

acq(L2)

modify X

rel(L2)

T1: Process P3:

acq(L3)

modify X

rel(L3)

T2: Process P4:

acq(L1)

modify Z

rel(L1)

T2: Process P5:

acq(L2)

modify X

rel(L2)

T3: Process P6:

acq(L2)

modify X

rel(L2)

a) [5 points] Note that the lock requests from P1, P2, and P3 occur at the same time T1 and all of them modify the same page X in their respective critical sections.

(i) (2 points) How would Treadmarks handle this situation?

Answer: Treadmarks will allow all three critical sections to proceed in parallel since they are for different locks.

(ii) (3 points) What actions will be taken by Treadmarks in executing each of these critical sections (at lock acquisition, during the critical section, and upon lock release)

Answer:

- Lock acquire: If page X exists locally, it will be invalidated. (+1)
- During the critical section: When the process tries to modify X, a page fault will occur. Treadmarks will fetch a pristine copy of page X from the owner; Create a twin for page X (+1)
- Lock release: create diff for page X (+1)

b) [5 points] What actions are taken by Treadmarks when P6 tries to acquire lock L2 at time T3 (at lock acquisition, during the critical section, and at lock release)?

Answer:

- Lock acquire: If P6 has a local copy of X, it will be invalidated. (+1)
- During the critical section: When the process tries to modify X, a page fault will occur. Treadmarks will fetch pristine copy of X from the owner of X; it will then get the diffs from P2 and P5 and then apply the diffs IN ORDER; It will then create a twin for page X (+3)
- Lock release: it will create a diff for page X. (+1)

GMS [15 points]

1. [8 points] Node P faults on page X, which is not present in any of the peer nodes the cluster. Node R houses the globally oldest page Y. Explain how GMS would strive to handle this page fault. Your answer should cover all possible scenarios. Note that this question is NEITHER about the actual implementation of GMS, NOR the Geriatric algorithm.

Answer:

- GMS on node P, brings in the missing page from the disk to its local part of its memory. (+1)
- To make room in its local part GMS on node P has to evict the oldest page on Node P:
  - Case 1: Oldest page Z in Node P is in the local part; Write it to disk if dirty (+1); and send it to Node R (+1)
  - Case 2: Oldest page Z in Node P is in the global part; Send it to Node R (+1)
- Action at Node R:
  - Place page Z in its global part (+1)
  - To make room for page Z, Node R has to evict page Y:
    - Case 1: page Y is in the global part; it is guaranteed clean so simply drop it (+1)
    - Case 2: page Y is in the local part; write it to disk if dirty (+1); simply drop it if it is clean (+1)

2. [4 points]
For this question, assume that GMS implementation has a "master node" that is always up. A new node N joins the GMS cluster. What actions will ensue as a result of this new node joining the GMS cluster?
Answer:

- Node N will notify the master node of the current GMS system of its arrival. (+1)
- The master node will recompute the page-ownership-directory (POD) for each node in the cluster (+1)
- The master node will distribute the newly computer POD to all the nodes including N. (+1)
- Each node will then send to Node N, the GCD entries they currently have, corresponding to the pages assigned to N as the owner. (+1)
-

4. [3 points] Explain how the Geriatrics algorithm illustrates the principle "Think Globally but Act Locally".

Answer:

Think globally [+1.5 points]

- To calculate the minAge and weight vector, GMS uses the age information of all local and global pages in the system

  OR

- GMS also decides the initiator for each epoch, which is based on the memory pressure experienced by each node.

Act Locally [+1.5 points]

- Page eviction decision is taken locally.

OR

- Remote node selection for sending an evicted page (if necessary) is also decided locally

DFS [14 points]

1. [6 points] You and your friend are tasked with implementing a disk-based distributed file system for the College of Computing. You are given a LAN cluster interconnected by 100 Gbps links. Your system should meet the following design objectives:

- Reduce disk accesses as much as possible
- Enable parallel processing of requests as much as possible
- Ensure no file server becomes a hotspot for requests

- Maximize the use of available compute and physical storage in the cluster to enhance performance

(a) [2 points] Your friend suggests storing the metadata for a file at the server that hosts the file on its local disk for simplicity of implementation. What are the downsides to this suggestion?

Ans: Storing files and their metadata on the same server will create hotspots if a server happens to host many hot files

(b) [2 points] You decide to decouple the server location of the metadata of a file from the server that hosts the file on its local disk. How does this reinforce the design objectives?

Ans: • Alleviates hotspot problem. If a server is hosting most of the hot files, it doesn't get overwhelmed with all the metadata requests (+1) • Since the metadata of files is distributed across multiple servers, more CPU and DRAM capacity can be utilized to manage file metadata (+1)

(c) [2 points] You and your friend discuss striping files across distinct and disjoint subsets of node storages. How does this reinforce the design objectives?

Answer: More parallelism in handling requests concurrently for different files striped across different subsets of nodes

2. [8 points] For the purposes of this question assume that the granularity of access is an ENTIRE file. Consider the following scenario in xFS. The contents of a file F are in log segments that are striped across storage servers S1, S2, and S3. Node M1 is the statically assigned metadata manager for the file F. Currently, the file F is in the local cache of Node N1. Node N2 wants to make modifications to F. Assume that S1, S2, S3, M1, N1, and N2 are different nodes of the cluster.

List all the actions and the modifications to the data structures of xFS at the different nodes to satisfy the above modification request of N2.

Answer:

- Node N2 consults the replicated mmap data structure on its node to determine the metadata manager for F. (+1)
- Node N2 contacts M1 (metadata manager for F) to request the file F in write mode. (+1)

- M1 recognizes from its internal data structures that F is now currently in the Unix cache of N1 in read mode, and asks it to forward the file to N2, and delete the file from its local cache. (+2)
- N1 sends the file F to N2. N1 notifies that the file has been sent to N2 and locally deleted. (+1)
- N1 updates its file-directory to note that F is no longer in its Unix cache. (+1)
- M1 updates its internal data structure that N2 has the file F encached in write mode. (+1)
- N2 updates its file-directory to note that F is in its Unix cache. (+1)

Potpourri [20 points]

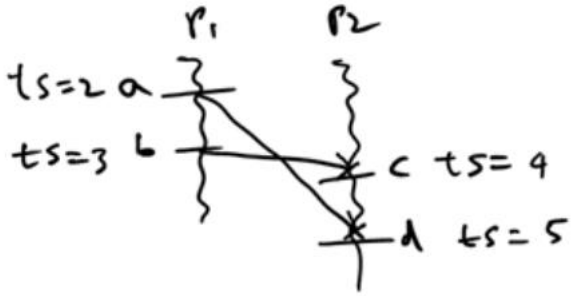
-

1. (2 points)(Answer True/False with justification) Between any two nodes in the distributed system, the arrival of messages at the receiver in the send order is a necessary condition for implementing Lamport's logical clock.

Answer:

False. Here is an example scenario:

Consider two processes P1 and P2, with two msg-send events from P1 to P2 with monotonic counters on P1 and P2 serving out logical timestamps for local events.



Despite messages going out of order the two conditions in "happened before" relation is satisfied for the following events:

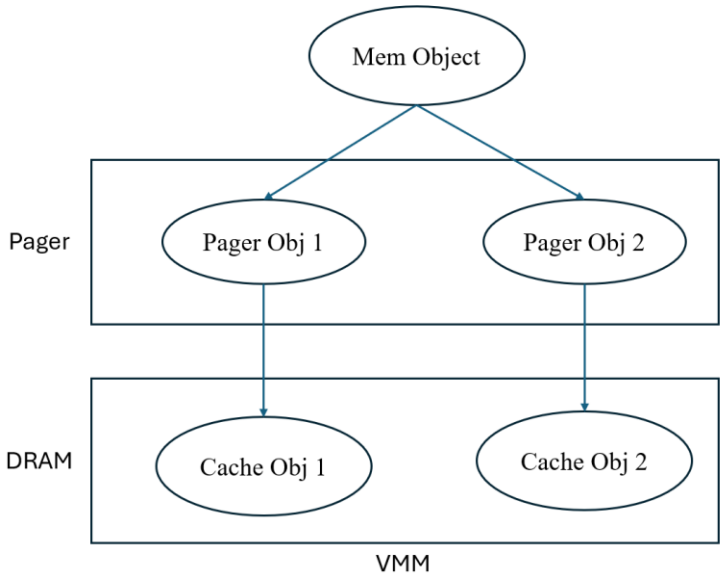
a -> b

c -> d

a -> c

b -> d

2. [2 points] (Answer True/False with justification. No credit without justification) As shown in the Figure below, a memory object is being mapped to two different cache objects in the DRAM through two distinct pager objects. The cache objects are present in two different regions but in the same linear address space as shown. Since the cache objects are in the same linear address space, Spring OS kernel is responsible for maintaining the coherence of the cache objects.



Answer: False. Spring system is not responsible for coherence of different cache objects which are mapped to the same memory object. It is the responsibility of the pager objects to ensure coherence if needed.

3. [4 points](Answer True/False with justification. No credit without justification).

(a) [2 points] In Treadmarks, upon a page fault for a page X in Node N1, the DSM software on N1 broadcasts the virtual page number (VPN) of the faulting page to all the peer nodes.

the peer nodes.

Answer: False.

In Treadmarks, there is a manager node statically assigned for every VPN. The DSM software on N1 contacts the manager node for the faulting page X. (+2 if the above justification is conveyed)

(b) [2 points] In Treadmarks, when acquiring a lock in Node N1, the DSM software on N1 broadcasts the lock acquisition request to all the peer nodes.

Answer: False.

Each lock has a statically assigned manager. N1 will contact the specific manager node for the lock request.

4. [2 points] (Answer True/False with justification) GMS is responsible for handling the coherence of a page that may be present in the "local" part of the memory in multiple nodes.

Answer: False, GMS only handles page faults not page coherence.

5. [2 points]

(Answer True/False with justification) The nodes N1, N2 and N3 in the GMS have received the following weights for the current Epoch: 0.2, 0.3 and 0.5, respectively. In the next Epoch, the initiator node will be N1.

Answer:

False, the initiator node will be the most idle node in the previous epoch, i.e. the node with the highest weight.

6. [8 points] Back in 1985, Sun Microsystems built the first Network File System and dubbed it NFS, and that name has stood the test of time.

Answer True/False with justification for each of the following questions with reference to traditional NFS. No credit without justification.

(a) Multiple network servers can provide file system service.

True, NFS utilizes multiple network servers to store files and serve client requests, with each server being responsible for distinct and disjoint partitions of the entire file system. (+2 or 0)

(b) The network servers for the data (actual file content) and the metadata (information about client nodes that are using the file, etc.) for a given file are not necessarily the same.

False, In NFS, the same server houses the file contents and the metadata related to that file. This is unlike xFS where metadata management is decoupled from the data management.

(+2 or 0)

(c) Individual files are striped across the disks of multiple network servers on the Local Area Network (LAN).

False, Each file is housed on a single server.

(+2 or 0)

(d) A file cached at a client may be used to serve the needs of other network clients for the same file bypassing the network server that hosts the file on its disk.

False, NFS does not keep track of client-side caching, and file caching is a feature of the server that hosts the file.

(+2 or 0)