Distributed Systems

2018 Exam 2 Review

Paul Krzyzanowski

Rutgers University

Fall 2018

Question 1

Unlike Ricart & Agrawala's algorithm, timestamps are used in Lamport's mutual exclusion algorithm to:

(a) Set a timer for a lease based on the sender's time rather than the receiver's time.

(b) Enable a receiver to discard earlier messages that arrive out of order.

(c) Ensure queues at all participants remain identical.

(d) Allow a participant to see if it can have the resource by comparing its timestamp with that of a received message.

- *(a) Lamport's does not deal with leases.*
- *(b) No. Ordering is not essential here. The assumption is that messages are delivered reliably, though.*
- *(c) Yes. Lamport's algorithm looks at the head of the queue to see which process has access to the resource.*
- *(d) Close but R&A does this. Lamport's simply adds received messages to a priority queue and then checks the queue whenever something is added or removed.*

Question 2

Which mutual exclusion algorithm does not require a client that wants a resource to send any request message?

(a) Centralized

(b) Token ring

(c) Lamport's

(d) Ricart & Agrawala

If a group member receives a token, it has access to the resource.

Question 3

Chang and Roberts improves the ring election algorithm by:
(a) Asking each client to acknowledge a message to avoid lost election messages.
(b) Terminating concurrent elections.
(c) Not requiring messages to be circulated among all live group members.
(d) Using multicasts to reach the entire group at once.

Question 4

The *ring election algorithm* builds a list of processes. At the end, the leader can be chosen as:

(a) The lowest-numbered process.
(b) The first process in the list.
(c) The last process in the list.
(d) Any of the above.

(b) & (c) will produce different results on different processes.

Question 5

Paxos' *abortable consensus* means:

(a) A proposed value may be rejected.

(b) A client can choose to stop the consensus algorithm.

(c) A client may reject the chosen value.

(d) The state of Paxos can be restored to its values before the algorithm was run.

Question 6

At the end of the first phase of the Paxos protocol, a proposer must:
(a) Use the value that was returned by the majority of acceptors.
(b) Propose the value that was provided by the client.
(c) Use the value of the highest accepted proposal ID that was returned, if any were returned.
(d) Use any value but it must be one of the proposed values.

This is how information gets propagated to other acceptors that may have been down or partitioned

At least one acceptor must have been alive from the previous majority that may have accepted a proposal.

Question 7

Unlike Paxos, the *Raft* protocol:
(a) Allows a single value to be chosen from a set of proposed values.
(b) Allows more than one value to be chosen from a set of proposed values.
(c) Includes group membership management.
(d) Includes leader election.

All requests must go through a leader.

A leader is elected via randomized timeouts.

Question 8

In Raft, a value is considered *committed* when:
(a) The leader adds the value to its log.
(b) A majority of followers have acknowledged receipt of the value.
(c) The next term starts.
(d) A candidate receives a majority of votes.

(a) This happens before the value is committed. Once a majority of followers agree to the value, it is considered committed.

(c) That just defines the start of a new post-election period.

(d) That's for running the Raft election algorithm to choose a leader

Question 9

Which of these is not an *ACID property* of transactions?

(a) Atomic: All changes are performed as if they were one operation. If anything fails, they are all undone.

(b) Consistent: Data is in a consistent state before and after the transaction.

(c) Isolated: Intermediate states of the transaction are invisible to other transactions.

(d) Distributed: Transactions be broken into multiple sub-transactions that span multiple systems.

Transactions don't have to be distributed.

The acronym is

Atomic, Consistent, Isolated, Durable

Question 10

The *three-phase commit protocol* was created to:

(a) Enable participants to abort or commit after timeouts rather than wait for all systems to recover.

(b) Ensure that the coordinator receives final acknowledgement of whether a transaction was committed or aborted.

(c) Allow participants to reach commit consensus with each other with no need for a coordinator.

(d) Fix the problem where some transactions commit while others abort under the two-phase commit protocol.

---

b. That happens in both protocols

c. Participants never talk to each other.

d. That wasn't an issue with 2PC. If it was, the protocol would be broken.

Question 11

The *CAP theorem* tells us that:

(a) A distributed transaction requires at least two sets of messages to get agreement to commit.

(b) Distributed transactions must use locks to keep other transactions from accessing their data.

(c) You can only have two out of three: consistency, atomicity, and partition tolerance.

(d) Highly-available distributed systems will have problems with consistency.

(c) CAP = Consistency, **Availability**, Partition tolerance

Question 12

The purpose of *two-phase locking* (2PL) is to:

(a) Keep transactions from accessing intermediate data from other uncommitted transactions.

(b) Keep transactions from accessing any data that was modified by other uncommitted transactions.

(c) Increase concurrency by separating read locks from write locks.

(d) All of the above.

Question 13

Strong strict two-phase locking (SS2PL) improves on two-phase locking (2PL) by:

(a) Keeping other transactions from accessing interim data from other uncommitted transactions.

(b) Keeping other transactions from accessing any data that was modified by other uncommitted transactions.

(c) Increasing concurrency by separating read locks from write locks.

(d) All of the above.

Question 14

A *commit lock*:
(a) Prevents a transaction from committing until all sub-transactions commit.
(b) Keeps transactions from acquiring read or write locks on a resource.
(c) Releases all locks on resources that were used by the transaction.
(d) Makes modifications on the resource permanent.

A commit lock is used during the commit phase

A commit lock is used in two-version based concurrency control to wait for all transactions that might be accessing the non-tentative version of the object to complete.

During the commit lock, no other transaction can access the resource. While the transaction commits, the results are made permanent … but that's not done by the lock.

Question 15

Multiversion concurrency control (MVCC) enables:

(a) A transaction to create multiple versions of an object prior to committing.

(b) Transactions to control the allowable amount of concurrency in the system.

(c) The use of read locks separate from write locks to increase the level of concurrency.

(d) Lock-free reads in transactions by accessing versions of data as of the start of the transaction.

---

a. No. A single version of an object is created when transaction commits if it modified the object

b. No.

d. No. This is read & write locks in general. MVCC does not use locking but checks timestamps of transactions vs. read-write timestamps on objects.

Question 16

Deadlock does not require:
(a) An exclusive lock on a resource.
(b) Waiting on a resource while having a lock on a resource.
(c) The ability for a coordinator to release a process's lock on a resource.
(d) A circular hold and wait dependency.

---

3 of 4 conditions for deadlock:

1. mutual exclusion

2. wait-and-hold

3. Circular wait

The fourth is **non-preemption**.

Question 17

Phantom deadlocks can be avoided by:

(a) Querying each participant for pending release messages.

(b) Using of two-phase locking.

(c) Using strong strict two-phase locking.

(d) Constructing a global wait-for graph.

Question 18

Which technique will abort a process because of deadlock?
(a) Edge chasing.
(b) Wait-die.
(c) Wound-wait.
(d) All of the above.

Edge-chasing tests whether there is a deadlock cycle.

The others force resource allocation in a way that there can't possibly be a cycle.

Question 19

On Linux, remote files can be accessed seamlessly because:

(a) The client module is installed as a file system type under the Virtual File System (VFS) layer.

(b) Applications are compiled with APIs to enable them to access remote files.

(c) The system call interface in the GNU C library (glibc) can tell if programs are trying to access remote files.

(d) Clients always access locally cached files and do not have to interact with servers directly.

Question 20

Which file system uses a download-upload rather than a remote access model?
(a) NFS
(b) Coda
(c) SMB
(d) GFS

Question 21

An AFS server uses *callbacks* to:

(a) Inform clients that a cached copy of their files is invalid.

(b) Ask clients to upload the latest version of their file.

(c) Let a client know that another client is accessing the same file.

(d) Tell a client that another a client has released a lock on a file they want to open.

Question 22

An *SMB oplock*, or *lease*:

(a) Enforces mutual exclusion, so other clients cannot access the remote file.

(b) Allows a client to perform groups of file system operations with no intervening operations from other clients.

(c) Tells a client how it may cache data for a remote file.

(d) Is a discretionary lock that other clients can check to see whether a client locked access to a file.

Question 23

A *client modification log* (CML) in Coda is:

(a) Used by the server to track the changes clients made to files to allow rollback.

(b) A server-managed list of files downloaded by clients so cache invalidations can be sent if those files change.

(c) An optimization on the server to send only the changed parts of files to clients instead of full downloads.

(d) The list of files that were modified on a client while it was disconnected.

Question 24

The design of Google Chubby can best be described as:

(a) One active master with multiple replicas as backups that do not process client requests.

(b) One master with multiple replicas, with client requests load balanced among all live systems.

(c) A set of replicas, each holding a different part of the file system name space.

(d) A peer-to-peer distributed hash table.

Question 25

Chubby does not permit this operation:
(a) Lock a file.
(b) Read a range of bytes from a file.
(c) Subscribe to receive notification of events, such as the modification of a file.
(d) Delete a file.

Question 26

Which statement is not true about the Google File System, GFS?
(a) A file's contents can be bigger than the capacity of any one server.
(b) Information about files is stored on a different server from file contents.
(c) File contents can be replicated by the file system
(d) The metadata of all the files is distributed among multiple servers.

Question 27

The defining characteristic of a parallel file system is:
(a) File data is replicated across multiple servers for high availability.
(b) The file system can process multiple requests concurrently.
(c) File contents are distributed among multiple servers.
(d) Multiple file systems are combined together to appear as one.

---

a. Replication is not part of the definition

b. All file systems are designed for concurrent access

d. Union file systems are not a facet of parallel file systems or NAS

Question 28

Notification servers in Dropbox:

(a) Were used by clients to inform servers of new files.

(b) Provided a group collaboration feature so users could add messages about shared files.

(c) Were an optimization to keep clients from polling.

(d) Allowed a client to be told when its uploads have completed.

Question 29

Consistent hashing is usually implemented by:
(a) Using a hash function where all existing key values remain the same when the number of buckets increases.
(b) Creating a hash function that generates a unique value for each key.
(c) Keeping the number of buckets constant even if the hash function changes.
(d) Having a bucket be responsible for a range of hash values.

(a) Doesn't explain how this helps implement consistent hashing

2 points for a

(b) Every hash function does this

(c) That defeats the point of consistent hashing. Why would you want to change the hash function?

(d) This is the best of the four answers.

Question 30

A node in a three-dimensional Content Addressable Network (CAN) must know of:
(a) 3 neighbors.
(b) 6 neighbors.
(c) 8 neighbors.
(d) 12 neighbors.

A node has 6 neighboring nodes:
up, down, left, right, front, back

Question 31

Amazon Dynamo is called a *zero-hop DHT* because:

(a) A collision-free hash function is used.

(b) Consistent hashing allows the system to grow.

(c) Clients know the complete set of servers and can directly contact the one that holds the data they need.

(d) Each server node maintains a finger table that identifies other nodes.

Question 32

Vector clocks are used in Amazon Dynamo to:

(a) Impose a total ordering of all get and put operations in the system.

(b) Disallow concurrent updates of the same object by choosing the earliest writer.

(c) Determine the sequence of write operations across multiple objects.

(d) Identify conflicting versions of the same object stored on different servers.

The end