

Name:

email:

ID:

Midterm Exam
CS555
10 Mar 2011

You have 1 hr. 20 min. for this exam. The exam has 7 pages. There are 100 possible points. Show all your work for partial credit.

Definitions

Each question is worth 1 point. Answer all questions in this section.

1. Define the following

a) Anonymity

Answer: The ability to operate on system objects without revealing one's identity.

b) Block Cipher

Answer: A kind of encryption where blocks of data are mapped one-to-one to another block of data.

c) Name resolution

Answer: Mapping a name from one namespace into another; resolving a DNS name maps it to a (list of) IP address(es).

d) Hesiod

Answer: The name service in Athena

e) Non-determinism

Answer: Property of a system where output is not completely determined by input. CSP supports non-deterministic loops and branches.

f) Logical Clock

Answer: Mechanism for ordering events in a distributed system based on their causality.

g) 9P

Answer: Plan 9's file system access protocol

h) Hash function

Answer: A function that summarizes data in a small fixed-size number. Small changes in the data can result in large changes in the data.

i) Eventual Consistency

Answer: Property of a replicated system where changes do not immediately propagate to all replicas, but at a later (often bounded) time, the replicas all reflect the change.

j) Stable Property

Answer: Predicate of a distributed system that, once true, remains true in all reachable states. Example: deadlock.

Name:

email:

ID:

Short Answer

Each question gives its value in the question. Answer all questions in this section. This section is worth 50 points.

2. We spent a lot of time in class talking about replication. Define replication (2 pts) and give 2 goals a designer might choose to meet through replication (1 pt each). For each goal, explain how replication can meet that goal. (3 pts each)

Answer:

Replication is providing multiple copies of a system object in such a way that a subset (sometimes one) of the copies can be used and that operations on one copy propagate to the others.

Two goals of replication are to improve *availability* and *performance* of the system. Replication improves availability by allowing a system element to make use of a different set of copies when one or more are unavailable. Performance can be improved because replication gives more choices of how to select the subset of replicas to operate on. If some replicas are closer or on faster hardware, overall performance can be improved by using those replicas preferentially.

3. Consider an e-commerce system that presents a large catalog of items to browse. You are given access to the times each item was viewed – for example time-stamped browsing logs from the previous week. You are asked to design a search system for this site. Describe how you can make use of the logs to provide better search results. Your answer should include the name of a similar system we studied in class. (5 pts)

Answer:

This is analogous to the Connections search system we studied in class. The log of browsing times is analogous to the tracker input. From the browsing times we can extract a relation graph, assuming that items browsed from the catalog near in time are related. From that we can use the Connections algorithms to include and weight those items when returning a search.

I'll give partial credit for doing Google-style analysis of the content pages, but the analog to Connections is much stronger.

Name:

email:

ID:

4. An administrator for a DNS server that is primary for several domains misconfigures the server so that it neither answers queries nor responds to requests from secondary servers. Assuming that the secondary servers remain up and accessible, when does the administrator notice that his names are no longer being resolved and why does it take that long (assume he does not detect the problem from logs, just from DNS behavior)? (3 pts) Describe the actions that the secondary servers take between misconfiguration and the correction of the problem. (2 pts)

Answer:

The names are served until the secondary servers expire the data, which is a parameter set by the primary. Once that expiration happens, the secondary servers stop replying to queries and no one can resolve the names. Until then, when the primary fails to respond, applications will fall back to secondaries and names will resolve correctly.

Until the expiration time, the secondary servers will regularly poll the primary for updates. When the names it will stop resolving them, but continue requesting updates. When the secondary successfully gets an update from the primary, the secondaries will again resolve names.

5. Consider a weighted voting system where the total number of votes is 5, the write quorum size is 4 and the read quorum size is 2. Each replica has 1 vote (there are 5 replicas).

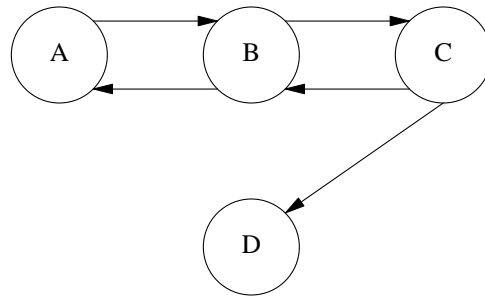
What happens if one of the replicas becomes unavailable to all clients, that is if the server holding it becomes inoperative? (2 pts) What happens if 2 servers go down? (3 pts)

Answer:

If one server goes down, the system can still meet both read and write quorums. It continues to operate, though performance will suffer.

When the second server goes down, there are only 3 votes remaining: enough to read, but not to write. Updates will fail until at least one server comes back on line.

6.



The diagram above represents 4 processes in a distributed system (A, B, C, D). Each arrow represents a directed channel between processes. B starts the Lamport snapshot algorithm. Explain how the algorithm proceeds, specifically, when each process takes a snapshot of its state, when it begins recording each channel's state, and when the process completes recording each channel's state. You can ignore the messages in the computation being snapshotted; that is, you may describe the snapshot algorithm working on an empty network. (10 pts)

Answer:

- B images its state, and sends markers to A and C, and begins recording state on the A → B and C → B channels. (Steps after this one can be reordered, other than the fact a process must send a marker before another can receive it).
 - When A gets the marker, it records the state of the B → A channel, and its state, and sends a marker on the A → B channel. A is now done.
 - When C gets the marker, it records the state of the B → C channel, and its state, and sends a marker on the C → B channel and the C → D channel. C is now done.
 - When B gets a marker from A it finishes recording the state of the A → B channel.
 - When B gets a marker from C it finishes recording the state of the A → C channel. When it has received markers on both the A → B and C → B channels, B is done.
 - When D receives a marker on the C → D channel, it records its state and the state of the C → D channel as empty. D is now done.
7. The Needham/Schroeder protocols make use of nonces. What are these nonces used for (2 pts) and how do they accomplish this (3 pts)? Such protocols are often carried over transport protocols that use sequence numbers for a similar purpose. Why do the Needham/Schroeder protocols use their own nonces instead of relying on the transport sequence numbers? (5 pts)

Answer:

Nonces are used by Needham/Schroeder to prevent repeated requests being treated as new requests. Specifically they are trying to prevent attackers from replaying requests and getting useful information. An endpoint keeps track of the nonces it has seen for some time and denies any repeated nonce.

Even if transport sequence numbers could be used across multiple sessions (and there's an argument that they can), the nonce is properly understood by the application. The application knows what a request is and which requests are required to have nonces, the transport providers do not. The end-to-end principle argues that this function - which is an application function - be provided in the application.

Name:

email:

ID:

8. Several systems we discussed in class have made use of overlay networks. Overlay networks generally forward messages less efficiently than the underlying network. Pick one of the systems we studied that uses overlays, and explain what benefit the overlay has. (1 pt for the system, and 4 for the explanation)

Answer:

The CAN uses an overlay to combine its logical search space and its message routing space. That logical space is both used often: it determines message routing and which nodes share state (neighbors), and dynamic in that nodes can join and leave it, it greatly simplifies the implementation to overlay that structure on the real topology rather than trying to embed it.

This is generally the case: the overlay simplifies embedding a structure into a topology.

Name:

email:

ID:

Long answer

This section is worth 40 points. Each question gives its value. Do all questions in this section.

9. We spent much of this semester so far talking about consistency. These questions each focus on those consistency systems.

- a) Serializable consistency is organizing updates to a set of replicas so that processes agree on the operations that occur and the order in which they happened. If there are no network partitions or failures, which of these implement serializable consistency: LOCUS, FICUS, IVY (Li & Hudak), Linda, Weighted Voting (1 pt each)

Answer:

Only FICUS is non-serializable

- b) How is the DNS like Moira in Athena? (2 pts) How is Grapevine different? (3 pts)

Answer: Each domain in the DNS is centrally configured, much as Moira configures services and pushes the configurations out. Grapevine is distributed in the sense that updates can be made multiple places and a single serialized version distributed.

- c) Consider a disconnected coda node and a node that remains connected to the system. If both change a directory, and the changes are integrated without human intervention, the system has clearly violated serializable consistency. The changes made in the past by the disconnected node are new to the connected one. What can we say about the two sets of changes? (5 pts)

Answer:

The changes do not conflict. That is, no changes that the connected node made would override changes that the disconnected node made. This is true of the actual operations, but there are some semantics that could collide. If the connected node was encoding the current number of files in the directory into the filename, that would be inconsistent after reconnection.

- d) Most of the consistency algorithms we discussed control the operations done by processes on an external object to keep that object's state consistent. What state does the Byzantine Generals algorithm make consistent? (3 pts) What inputs can unsynchronize that state (what is the threat model for the Byzantine Generals algorithm) (2 pts)?

Answer: The Byzantine Generals algorithm synchronizes the internal state of the processes with respect to a cooperative plan. The threat model is that some processes are deliberately or accidentally providing inconsistent and potentially disruptive data.

Name:

email:

ID:

10. Another theme of this semester has been communication. This question addresses those issues.

- a) A monitor protects a shared object that can be accessed using protected access methods, which is used for communication. What semantics does a monitor enforce? (2 pts) What must a developer using a monitor enforce when one of these protected methods is exited? (3 pts)

Answer:

Only one process can be executing one of the protected access methods at any time. In order to keep shared state consistent, a developer must make sure that the monitor invariant (the rules for consistent state in the object) is restored before exiting an access method or waiting on a condition.

- b) The Remote Procedure Call (RPC) system is very practical, and we have seen several implementations of it. Explain each of these functions of the RPC system: Binding (1 pt), RPC run-time (2 pts), stubs (2 pts)

Answer:

Binding is matching a service to a server, done by Grapevine in the Birrell RPC. The RPC run-time is responsible for binding and the communication between client and server, the stubs are responsible for formatting parameters and return values into the shared representation.

- c) Why is implementing a Linda tuple space more difficult than implementing a distributed hash table like the Content Addressable Network? (3 pts) Name a naming system that more closely matches Linda's semantics. (2 pts)

Answer:

The tuple space is more complex than the CAN because the CAN has a fixed search key, and can index data by it. The tuple space can be searched by any member of the tuple space. It is much more like a search than a name resolution.

The Intentional Naming System (INS) searches names in a similar way: matches can be made on any of the attributes in a name. A similar argument can be made for Google, though that is even more free form.

- d) FreeNet is basically a broadcast medium. A user can insert a file for others to view, but it is difficult to change or delete. Explain 2 features of FreeNet that make modification of data inside it difficult. (5 pts)

Answer:

Freenet's anonymity makes it difficult to assign any kinds of rights to a particular piece of data. Deletion and modification are generally access controlled, which depends on some notion of identity or credentials. FreeNet excludes both of these to protect anonymity.

Secondly, FreeNet's caching and distribution policies make it nearly impossible to find all copies of a file, and deleting or changing some copies has strange semantics. They are opportunistically cached and aged out of local caches based on local usage patterns. In order to prevent tampering with files, there is no central location system that does more than find a copy.