

Name:

email:

ID:

Midterm Exam
CS555
8 Mar 2007

You have 1 hr. 20 min. for this exam. The exam has 6 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) Kerberos

Answer: The authentication server in Athena

b) Causal broadcast (CBCAST)

Answer: ISIS message that is received by all members of a process group in the same order relative to messages that it's causally related to.

c) RPC runtime

Answer: The part of the RPC system that is concerned with properly transferring the data between server and client. It is also responsible for reporting errors and other miscellaneous interfacing issues.

d) Graft point

Answer: Entry in a FICUS volume where another volume may be attached.

e) Stable property

Answer: A predicate about a distributed system that, if true remains true in all states reachable from the current state.

f) Proxy object

Answer: Java object in JINI that acts as both a marshaller and an RPC runtime to communicate with a remote object to carry out the RPC.

g) Rendez-vous (from CSP)

Answer: Synchronization caused by a message exchange in CSP. Both processes are exactly at the instructions for I/O when the message is exchanged.

h) Weak representative

Answer: In weighted voting a representative without a vote. Such a representative can be included in any quorum, and may be used to improve performance.

i) 9P

Answer: The plan 9 file system access protocol.

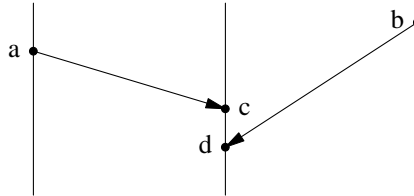
j) Monitor

Answer: A language-based synchronization structure used in Mesa. A set of procedures are collected in the monitor and only one process/thread can be executing in that collection at any time.

Short Answer

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

2. Consider the following process interaction diagram. As in class, time gets later toward the bottom of the page, events in a single process are points along the same vertical line and messages between processes are diagonal lines. Give two valid total orderings of the events below based on Lamport's "happened before" relationship. (2 points)



Explain what the "happens before" relation means for events; this is the same thing as asking what it means for two events to be causally related. (4 points) Why are causal relationships important to a distributed debugger? (4 points)

Answer:

Any ordering where a is before c, b is before d, and c is before d is correct.

If a "happened before" b it means that a can possibly affect b. The information created or delivered at a can have a bearing on the process's action at b. A distributed debugger is interested in this information because if an error is an event in a process, the debugger only needs to keep track of and collect information that is causally related to the error event. Other events, by definition, could not be relevant.

3. A hierarchical name space (or a hierarchical implementation of a flat name space) is usually more scalable than a flat implementation in the sense that it is more efficient to change a name. Explain why. (5 points)

Answer:

By separating the names into independent sections using the hierarchy, it is possible to localize the amount of data that a given change affects. For example if all names are kept in a flat sorted list, any change to a name results in moving all names after that name around. A single name change affects the whole name space. By partitioning the space into independent parts those changes only affect those names in the same division of the hierarchy, e.g., the directory or DNS zone.

In addition to efficiency in supporting changes, a hierarchical name space is often more memory-efficient than a flat name space, especially when there are many names and the usage pattern of the names follows the hierarchy. Again, explain why. (5 points) It may help you frame your answer to think in terms of a traditional file system and block cache.

Answer:

By partitioning the name space into units that approximate the usage pattern the hierarchical system stores names together that are used together. As a result caching those storage units usually results in good cache performance - i.e., good value for each stored name - and therefore fewer fetches of new names.

In a traditional file system directory this means many files in the same directory will be used together, so caching that directory will speed look ups.

Name:

email:

ID:

4. In the Mesa monitor implementation, device drivers signal special condition variables without holding the corresponding monitor lock using a mechanism called a *naked notify*. The condition variables in question are implemented a binary semaphore. Explain the problem this mechanism is solving. (5 points)

Answer: The thread running in the device driver doesn't hold the monitor lock, so another thread may be in the monitor about to sleep on the condition variable. The thread about to sleep may miss the notify sent from the device driver because they're not coordinated, and the possibility of a device driver blocking on a monitor lock is unacceptable. To avoid the race condition caused by unsynchronized access to the condition variable, the naked notify is used.

If you are implementing monitor locks using Linda, which Linda operation will you use to acquire the lock? (1 point) Why must you use that one? (4 points)

Answer: You'll use `in()`. Acquiring a lock requires an operation that both potentially blocks the current thread and atomically changes the state of the shared state used for synchronization. `in()` is the only Linda operation that does both (`read()` does not change the state).

5. Assume that a designer has come up with a new algorithm that changes how the Google system we discussed in class calculates *page rank*. Assuming nothing else about the system changes, what difference will a user see when he or she performs a search? (5 points)

Answer: The same pages will be found, but they will be returned in a different order. Basically, everyone understands how to index the web pages by search terms (though the implementation in the Google paper is time and space efficient), but the breakthrough from Google was in presenting the matches humans (by and large) consider best by using the human-generated structure of the web.

6. If a designer is using a system based on the Byzantine Generals oral messages solution to protect against malfunctioning sensors. Each sensor is a general. How many sensors must the designer deploy to overcome 10 bad sensors? (1 point) How many sensors must the designer deploy if he or she uses the signed messages algorithm (1 point). What additional requirement does the signed message variant impose on the designer? (3 points)

Answer: The oral message solution requires at least 31 sensors to deal with 10 failures. The signed message solution can cope with any number of traitors, but 12 is the smallest interesting set. To use signed messages the sensors must be capable of digitally signing their observations.

Name:

email:

ID:

7. The following is the state of a Time Warp system. Describe completely what actions the process takes upon:

- receiving an anti-message with virtual receive time 10 (3 points)
- receiving a message with virtual receive time 3 (4 points)
- receiving a system message that GVT has advanced to 5 (3 points)

Consider each of these occurrences independently; that is, assume that the state is as diagrammed below when each of the above events occurs.

Current Status

LVT	State
8	x=4; y=10

Saved States

Time	State
2	x=1; y=3
4	x=2; y=3
5	x=2; y=5
7	x=4; y=5

Input Queue

Send Time	1	7	9
Recv Time	2	9	11
Msg	bar	baz	zot
Sign	+	+	+

Output Queue

Send Time	1	4	6
Recv Time	3	5	9
Msg	aaa	bbb	ccc
Sign	-	-	-

Answer:

1. The anti-message is queued in the future (on the input queue) and the process continues.
2. The state rolls back to the saved state at time 2, LVT set to 2, the anti-messages for messages sent after 2 are resent (bbb and ccc) and the process restarted from there (beginning with the replay of the message received at time 2).
3. Saved states for times before 5, input messages received before 5 and output messages sent before 5 are all discarded. Note that you have to keep the VT 5 data.

Name:

email:

ID:

Long answer

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

8. Several of the systems we discussed in this semester have made use of hints or other approximate information to achieve a performance or functionality advantage. For each system below, briefly describe how it uses hints or approximate information and what advantage using the hint has compared to calculating the information exactly. (5 points per system)

- a) Ivy (the distributed virtual memory system)

Answer: Ivy is the classic hint system. Instead of a correct pointer to the owner of a page, each node keeps track of the node that owned a page the last time this node was sure who owned the page. By not keeping track of that explicitly, the overhead of keeping a pointer per page synchronized is avoided. That overhead is mostly in terms of extra messages.

- b) SNS (the cluster-based system for building network services)

Answer: SNS uses approximate load information in its load balancing algorithms. The information is approximate and decentralized to simplify the implementation and to make recovery easy should the centralized load balancer fail.

- c) FreeNet (the anonymous distributed file store)

Answer: When a FreeNet node doesn't have a given file, it uses the node that last returned some data that was near the file being searched for in hash space. While the FreeNet algorithms try to keep data grouped by closeness in hash space, this information is definitely approximate. Using these hints reduces the amount of information a given node has to keep and simplifies the distribution and look up algorithms.

- d) Grapevine

Answer: Grapevine data is approximate while an update is in progress, especially when network load is high or servers have failed without propagating changes. Different registries for the same RName may return different answers to the same query. This trade was made specifically to keep the name and message system highly available.

Name:

email:

ID:

9. Several systems we have discussed this semester rely on replication to meet their goals. This question explores replication and consistency.

- a) There are two primary reasons to replicate data or services in a distributed system. Give the two reasons and explain (briefly) how replication solves those problems. (4 points)

Answer: The two reasons to replicate is faster access to data (more copies means that the chance there's a copy near an accessor improves) and to improve availability (as long as the system can operate with fewer copies than the maximum, the losing copies does not bring the system down.)

- b) Though they are alike in many ways, LOCUS and FICUS differ in the consistency model they present to users because they are intended to operate in different environments. Describe the different environments these systems operate in (3 points) and describe one way in which the consistency model differs between them (3 points)

Answer:

FICUS is intended to operate in the wide area, on a global scale in terms of machines, users, and files while LOCUS operated on a small enough scale that global knowledge of the nodes in the system was feasible.

FICUS implemented non-serializable consistency, meaning that the update pattern of the replicas did not meet traditional models of consistency and that the behavior could change to meet local constraints easily. LOCUS presented a more constrained model that it could enforce using its greater knowledge and control.

- c) What phase of Coda operation is most like LOCUS recovery from partition? (1 point) Describe what happens in this Coda phase and how this is like LOCUS partition recovery. (4 points)

Answer: In re-integration, the changes made by the disconnected coda client (a partition of one) are replayed into the shared file system, inconsistencies are detected, and if they can be mitigated, repaired.

- d) Explain one technique that FICUS, LOCUS, and Coda all use to repair inconsistencies and to avoid human intervention. (4 points)

Answer: All three systems treat some files as having known semantics that can be used to resolve conflicts. Specifically they all recognize the special semantics of directories that only entries to be deleted added or renamed.