Final Examination
(Distributed Database Systems, 2011)

Question 1 (20%)

(1) (10%) Which of the following schedules are conflict equivalent? (Ignore the commit "C" and abort "A" commands.)

S1 = W2(x), W1(x), R3(x), R1(x), C1, W2(y), R3(y), R3(z), C3, R2(z), C2
S2 = R3(z), R3(y), W2(y), R2(z), W1(x), R3(x), W2(x), R1(x), C1, C2, C3
S3 = R3(z), W2(x), W2(y), R1(x), R3(x), R2(z), R3(y), C3, W1(x), C2, C1
S4 = R2(z), W2(x), W2(y), C2, W1(x), R1(x), A1, R3(x), R3(z), R3(y), C3

Answer: S1 and S4 are conflict equivalent.

Two schedules are conflict equivalent if their relative orders of execution of the conflicting operations are equivalent.

(2) (10%) Which of the above schedules (S1,S2,S3,S4) are serializable?

Answer: S3 (T2, T3, T1) S4 (serial order: T2, T1, T3)

Question 2 (20%)

Given the algorithms for the transaction managers and the lock managers for the distributed two-phase locking approach.

Answer hint:

Distributed two-phase locking approach - There is a lock manager at every site. Each lock manager handles lock requests for data at that site. Concurrency control is accomplished by the cooperation of lock managers at the sites where data is involved in the transaction.

(P319-322 provide a centralized algorithm, taking for reference.)

Question 3 (25%)

(1) (5%) Compare and analyze the relative merits of centralized and hierarchical deadlock detection approaches in a distributed DBMS.

Answer hint:
A centralized deadlock detection scheme is a reasonable choice if the concurrent control algorithm is also centralized.
It is better for distributed access patterns across sites since deadlocks occurring between any can be immediately identified. However, this benefit comes at the expense of communications

between the central location and every other site.

A hierarchical deadlock detection scheme releases the burden of one single site for deadlock detection, and let more sites get involved.
When access patterns are more localized, perhaps by geographic area, they may likely occur among certain sites with frequent communications. The hierarchical approach is more efficient in that it checks for deadlocks where they most likely happen, the hierarchical scheme splits deadlock detection efforts, thus resulting in greater efficiency.

(2) (10%) Consider the following modification to a local wait-for graph: Add a new node T_{ex}, and for every transaction T_i that is waiting for a lock at a certain external site, add the edge $T_i \rightarrow T_{ex}$. Also add an edge $T_{ex} \rightarrow T_i$ if a transaction executing at a certain external site is waiting for T_i to release a lock at this site.

If there is a cycle in the modified local waits-for graph that does not involve T_{ex}, what can you conclude? If there exists a cycle involving T_{ex}, what can you conclude?

Answer:

A cycle in the modified waits for graph not involving $\mathbf{T_{ex}}$ *clearly indicates that the deadlock is internal to the site with the graph. If there is a cycle involving* $\mathbf{T_{ex}}$*, then there may be a potentially global deadlock.*

(3) (10%) Whenever the local waits-for graph at a certain site suggests that there might be a global deadlock, send the local waits-for graph to the next site. At that site, combine the received graph with the local waits-for graph. If this combined graph does not indicate a deadlock, ship it on to the next, next site, and so on, until either a deadlock is detected or we are back at the site that originated this round of deadlock detection. Is this scheme guaranteed to find a global deadlock if one exists?

Answer:

The scheme is guaranteed to find a global deadlock provided that the dead- lock exists prior to when the first waits-for graph is sent. If this condition is met, then the global deadlock will be uncovered before any node receives a graph containing its own nodes back.

Question 4 (20%)

(1) (5%) Explain the need for a commit protocol for a distributed database management system.

Answer:
Crash recovery in a distributed database is complicated by new kinds of failures such as crash of a remote site or breakdown in communication links. Also, either all participants of a given transaction must commit or none of them must commit. This all-or-none requirement forces us to use some commit protocol, to ensure all sub-transactions either commit or abort.

(2) (5%) In two phase commit (2PC) protocol, the participants send an *ack* message to the coordinator after receiving global-commit/global-abort message. Why is the *ack* messages needed?

Answer:

Ack messages let the coordinator know that its global decision message has been received by the participants and the participant has committed/ aborted, as instructed by the coordinators. It indicates the end of the 2PC protocol, as well as the transaction.
Without ack message, the coordinator has no idea about the status of the participant sites.

(4) (10%) In two phase commit (2PC) protocol, suppose that a site does not get any response from another site for a long time. Can it tell whether the connecting link has failed or the other site has failed? How is such a failure handled?

Answer:

It is impossible for a site to determine whether a particular communication link has failed, or the site at the other end of the link is down.

The failure can be handled based on the log file which indicates the states when the fail happens. (See the slides for reference)

If the site is a coordinator, and if it is in "WAIT" state, …

If the site is a participant, and if it is in the "INITIAL" state, …

Question 5 (15%)

(1) (10%) Is the Web a distributed database system? Please briefly explain your answer.

Answer hint:

Not exactly. No transaction and reliability responsibilities in the Web domain. But the Web provides DBMS-like access to semi-structured data.

(5%) For a web-based E-trading application (like Taobao), please analyze the benefit and loss of enforcing the transaction's ACID properties