Qualifier Questions for CS 542 Examination, Spring 2015

Time Allocated 45 minutes after the final examination of CS 542

Question 1: [3 point]

1. How can the protocols and opportunities for transaction processing (including failures) change if communication delay is low and is not a major factor?

1. How can the protocols and opportunities for transaction processing (including failures) change if communication delay is high?

1. How the algorithms for transaction processing are affected in a mobile communication environment where network partitioning is always a possibility?

Question 2: [1 point]

Instead of storing back-up copies of updated database stable storage (hard disk), one can store back-up copies at other sites.

What characteristics of I/O cost and communication cost will allow this scheme to be practical?

Question 3: [1 point]

In a history of concurrent execution of transactions:

What is a dead transaction?

What is a live transaction?

Question 4: [1 point]

What is the difference between a two phase commit and three phase commit protocol?

Question 5:[1 point]

Give a simple termination protocol. Show how it is vulnerable and is blocking due to a failure?

Question 6: [2 point]

How can a Read one copy, Write all copies (ROWA) protocol be changed in a fully replicated distributed database system to deal with failure? Give at least three different schemes.

Prove that 2PL guarantees serializability is simple one

What are the different types of consistency level and give an application for each is tricky one

What are the vulnerabilities in a simple termination protocol and give an example to illustrate this?

What are the various ways transitions in multiple partitions can be merged for keeping mutual consistency among all replicas.

Database management  systems (DBMS)

provide higher level user support than conventional operating systems.

What operating system services are appropriate for support of database management functions?

Several design principles necessary to build high performance and reliable distributed database systems have evolved from conceptual research, prototype implementations, and experimentation.

The relationship between database and operating systems is discussed along with the desirable features in communication software for reliable processing.

What ideas are useful for increasing the degree of concurrency?

Answer: They include flexible transactions, adaptability, prewrites,

multidimensional timestamps, and relaxation of two-phase locking.

What ideas are useful for avoid blocking of transactions due to conflicts?

Answer: Use optimistic, Think of variations of the use 2PL like releasing locks soon after lockpoint.

What are the generic approaches to build CC algorithms?

What is causal ordering among events in the life of a transaction in a  distributed database systems? Specify events and causal relationships.

Give an example of a flexible transaction. How is the decision made to switch an action of the transaction with an alternate action? How rollback is used for serialization.

What are the different ways in which validation can be implemented efficiently in an optimistic concurrency control algorithm?

Give an example of an action that can be undone? How can one implement an Undo operation