CMPS 4760/6760 Distributed Systems – Spring 21

Homework 4 (15 points)

Due 04/29/21 before class

1 **(6 points)** A server manages the objects $a_1, a_2, ..., a_n$. The server provides two operations for its clients: (1) *read(i)* returns the value of a_i and (2) *write(i, Value)* assigns *Value* to a_i. Consider two transactions T and U defined as follows:

T: x= read(j); y = read(i); write(j, 44*); write(i,* 33*);*

U: x= read(k); write(i, 55*); y = read(j); write(k,* 66*),*

Give serially equivalent interleavings of T and U with the following properties:

(a) that are strict;

(b) that are not strict but could not produce dirty reads;

(c) that could produce dirty reads.

2 **(4 points)** Explain why serial equivalence requires that once a transaction has released a lock on an object, it is not allowed to obtain any more locks. Illustrate your answer with the following transactions T and U:

T: x= read(i); write(j, 44*);*

*U: write(i,*55*); write(j,* 66*).*

Describe an interleaving of T and U in which locks are released early with the effect that the interleaving is not serially equivalent.

3 **(5 points)** A three-phase commit protocol has the following parts:

Phase 1: Is the same as that for two-phase commit.

Phase 2: The coordinator collects the votes and makes a decision; if it is *No*, it aborts and informs participants that voted *Yes*; if the decision is *Yes*, it sends a *preCommit* request to all the participants. Participants that voted *Yes* wait for a *preCommit* or *doAbort* request. They acknowledge *preCommit* requests and carry out *doAbort* requests.

Phase 3: The coordinator collects the acknowledgments. When all are received, it *commits* and sends *doCommit* to the participants. Participants wait for a *doCommit* request. When it arrives they *commit.*

Explain how this protocol avoids delay to participants during their 'uncertain' period due to the failure of the coordinator or other participants. Assume that communication does not fail.