ECE428/CS425/CSE424: Distributed Systems **Spring 2011**

Homework 5 – Solutions **Due: by 2 pm on 05/03/2011**

Total points: 60 points

1. (10 points) Considers *n* nodes numbered 1 through *n*, each holding a *k*-bit value. Node 1 wants to find the parity of the *n*k* bits collectively held by the *n* nodes. One solution for this will be for node 1 to obtain the *k*-bit values from nodes 2 through *n*, and then compute the parity. This will require *(n-1)k* bits of communication. Suggest a more efficient algorithm. What is the worst-case communication complexity (in bits) using your algorithm?

   Nodes 2…*n* can send a 1 if the number of 1s in its *k*-bit value is odd or a bit 0 if it is even. Node 1 can then compute the overall parity. This algorithm requires (n-1) bits of communication.

2. (10 points) Transactions T and U execute on a single server. In the table below, time proceeds from top to bottom, and relative position of operations indicates the relative order in which they were performed. State whether the execution shown is possible in each of the following cases. Assume that a different lock is used for each variable w, x, y and z.

Transaction T	Transaction U
a = read (x);	
	b = read(y)
	c = read (x);
	write (z,1);
	commit
d = read (z);	
write(w,2);	
commit	

   (i) Exclusive locks are used with two phase locking.
   It is not possible. Transaction T acquires a lock for variable x. So transaction U cannot acquire a lock for variable x.

   (ii) Read-write locks are used with ~~strict~~ two phase locking. With read-write locking, the transactions acquire read lock when that is adequate.
   It is possible. Transaction T and U read variable x but they do not write to variable x. So read-lock allows transaction T and U to read variable x.

3. (a) (10 points) The table below shows the interleaving of the operations performed by transactions T, U and V (the tabular notation is similar to that used in question (2). Suppose that optimistic concurrency control with *forward* validation is used. Validation is carried out when a transaction performs closeTransaction operation. When forward validation fails, the transaction performing the validation is immediately aborted. Which of the transactions (if any) must be aborted?

Transaction T	Transaction U	Transaction V
openTransaction		
	openTransaction	
		openTransaction
m = read (w);		
write (y,1);		
	r = read (z);	
		q = read (v);
		write (z,9);
		closeTransaction
	p = read (w);	
	write (v,4);	
	closeTransaction	
n = read(v);		
write (w,4);		
closeTransaction		

Transaction V will be aborted since V writes to variable z that U has read.

4. (10 points) Suppose timestamp ordering is used for transaction concurrency control. Initial value for all timestamps is t_0. Assume that each transaction opens and obtains a timestamp just after the openTransaction operation. In the tables below, time increases from top to bottom.
   Consider the following interleaving of transactions T, U, V and W. Assume that T, U, V and W get timestamps t_1, t_2, t_3 and t_4, respectively, when they perform openTransaction, such that $t_0 < t_1 < t_2 < t_3 < t_4$. There are no other previous or pending transactions.

Transaction T	Transaction U	Transaction V	Transaction W
openTransaction			
	openTransaction		
		openTransaction	
			openTransaction
	write(m,20);		
	write(n,30);		
		write(m,24);	
			y = read(m);
	commit		
x = read(m)			

For each of the operations listed below (from the above table), determine whether it can be executed immediately, or has to be aborted, or has to wait until some other transaction commits (in the latter case, specify which transaction needs to commit first). Explain your answers briefly.

a. write(m,24) operation by transaction V
   Execute immediately. (By applying the rules on page 503 in the textbook.)

b. read(m) operation by transaction W
   Wait until V commits. (By applying the rules on page 504 in the textbook.)

5. (10 points) Consider a quorum consensus method for performing read/write operations to a group of 3 replicas (in the textbook, this method is attributed to Gifford). The following table gives the latency incurred in accessing each of the replicas, the votes assigned to each replica, and the quorum size for read and write operations. Assume that replica 1 is unavailable at the time of a request with probability 0.2, and the other two replicas are each unavailable with probability 0.01.

Latency (ms)			Voting configuration			Quorum sizes	
Replica 1	Replica 2	Replica 3	Replica 1	Replica 2	Replica 3	Read quorum	Write quorum
100	200	400	2	1	1	2	3

a. What is the range of latency incurred in the write operation?
200 ms – 400 ms
(200 ms when replicas 1 and 2 are accessed for the quorum, and 400 ms when replicas 1 and 3 are used for the quorum.)

b. Determine the blocking probability that a quorum cannot be obtained when a write operation is made.

Success probability = Pr{replica 1 is accessed successfully} AND Pr{at least one of replica 2 or 3 is accessed successfully).
= (1-0.2)x(1 – (0.01x0.01)) = 0.8(1- 0.0001)
Blocking probability = 1 – Success probability = 0.20008

6. (10 points) Problem 15.4 from the textbook (4^{th} edition).
p will not be in the next view of q and r.