15-440/15-640: Homework 2

Due: October 13, 2017 10:30am (**NO LATE DAY**)

Name:
Andrew ID:

1 Concurrency Control

1. Is 2-phase commit blocking or non-blocking? What about 3-phase commit? Elaborate on your answers by explaining the situations where a transaction blocks or why it does not block.

2. Consider the following program:

```
// DB is a key/value database
acquireWriteLock(key1)
acquireWriteLock(key2)
x = DB[key1]
DB[key2] = x
releaseLock(key2)
acquireReadLock(key3)
y = DB[key3]
DB[key1] = y
releaseLock(key3)
releaseLock(key1)
```

   (a) Which ACID property(s) does this program violate?

   (b) How would strong strict 2-phase locking fix it?

2 Distributed Mutual Exclusion

1. In class we discussed enforcing mutual exclusion, among other ways, via a central server, majority voting, and token ring.

   (a) How many messages are required per request under heavy contention? What are the messages used for? Write your answer for each algorithm.

   (b) List at least one disadvantage of each algorithm.

   (c) Which of these systems is most robust to failure? Why?

2. Consider three processes. The system has totally ordered clocks by breaking ties by process ID. It uses the Ricard & Agrawala algorithm. The timestamp for each process of id i is $T(p) = 10 * L(p) + i$, where $L(p)$ is a regular Lamport clock.

   Each message takes 2 "real-time" steps to get delivered. Critical section takes 2 real-time steps. Fill in the table with the messages that are being broadcast, sent, or received between the processes until all nodes have executed their critical sections. Write "execute critical section" as the action for a node when it enters its critical section. The first three rows have been filled in for you, and the fourth row has been started. Assume that if a process receives messages from the other two processes at the same time, the message that comes from the lower process ID will be received first.

Action Types: Broadcast (B), Receive (R), Send (S), Execute Critical Section (ExCS) Initial timestamps: P1 $\rightarrow$ 111, P2 $\rightarrow$ 212 and P3 $\rightarrow$ 103

Real Time	Process	Lamport Time	Action(to/from)	Contents	Q at P1	Q at P2	Q at P3
1	1	111	B	(request 111)	111		123
	3	123	B	(request 123)			
2	2	222	B	(request 222)	121	222	123
3	1	131	R from 3	(request 123)	111	111	111
	2	232	R from 1	(request 111)	123	123	123
	2	242	R from 3	(request 123)		222	
	3	133	R from 1	(request 111)			
4	1	231	R from 2	(request 222)	111	222	123
	2	252	S to 1	(reply 111)	123		222
	2	262	S to 3	(reply 123)	222		
	3	233	R from 2	(request 222)			
	3	243	S to 1	(reply 111)			

3 Logging and Crash Recovery

1. The ARIES logging and crash recovery design we talked about can also recover from a crash during the recovery phase for a previous crash. How is this achieved?

2. List one advantage and one disadvantage of checkpointing in a log based recovery system.

4 Distributed Replication/Paxos

1. Justify the need for the prepare phase in the Paxos algorithm.

2. You have set up a fault-tolerant banking service for the PNC bank (Paxos National Corporation bank). Based upon an examination of other systems, you've decided that the best way to do so is to use the Paxos algorithm to replicate log entries across three servers, and let one of your employees handle the issue of recovering from a failure using the log.

   The state on the replicas consists of a list of all bank account mutation operations that have been made, each with a unique request ID to prevent retransmitted requests, listed in the order they were committed.

   Assume that the replicas execute Paxos for every operation.[1] Each value that the servers agree on looks like "account action 555 transfers $1,000,000 from Yuvraj A. to Srini S.". When a server receives a request, it looks at its state to find the next unused action number, and uses Paxos to propose that value for the number to use.

   The three servers are S1, S2, and S3.

   At the same time:

   - S1 receives a request to withdraw $500 from Yuvraj.
     - S1 picks proposal number 501 (the n in Paxos)
   - S2 receives a request to transfer $500 from Yuvraj to Srini.
     - S2 picks proposal number 502

   Both servers look at their lists of applied account actions and decide that the next action number is 15. So both start executing Paxos as a leader for action 15.

   The first few messages are given below:

   ```
   S1 -> S1 PREPARE(501)
   S1 -> S1 RESPONSE(nil, nil)

   S1 -> S2 PREPARE(501)
   S2 -> S1 RESPONSE(nil, nil)

   S1 -> S3 PREPARE(501)
   S3 -> S1 RESPONSE(nil, nil)
   ```

   For each of the following scenarios, give a sequence of events that could lead to it.

   (a) The servers agree on the withdrawal as entry 15.

   (b) The servers agree on the transfer as entry 15.

[1] In practice, most systems use Paxos to elect a primary and let it have a lease on the operations for a while, but that adds complexity to the homework problem.

5 Fault tolerance and RAID

A friend of yours bought 25 used hard drives from a sale. Each drive has the following characteristics:

- Capacity: 500 GB
- Sequential speed: 60 MB/s
- Random IOPS: 30 IOPS
- MTTF: 1 year

Your friend wants a lot of speed and a lot of storage, and decides to build a RAID-0 array with the disks.

1. Is this a good idea? Discuss this w.r.t. the effective capacity, sequential speed, random IOPS, the MTTF and data durability.
2. Another friend suggested that a RAID-5 array with 25 disks would be a better idea than RAID-0. Why? Compare all the characteristics of the RAID-5 arrangement with the RAID-0 arrangement.
3. Your friend was a little disappointed on hearing that the system would fail very soon and new disks would be needed to replace the failing ones. Your friend agrees to compromise a little on storage space and speed. What would be a good design so that no additional disks need to be purchased for about a year?