CMPSCI 677: Operating Systems
Homework 3
Sample Solution
Spring 2002

1. Many distributed algorithms require the use of a coordinating process. To what extent can such algorithms actually be considred distributed? Discuss.

   Answer —

   In a centralized algorithm, there is often one, fixed process that acts as coordinator. Distribution comes from the fact that the other processes run on different machines. In distributed algorithms with a nonfixed coordinator, the coordinator is chosen (in a distributed fashion) among the processes that form part of the algorithm. The fact that there is a coordinator does not make the algorithm less distributed.

2. A distributed system may have multiple, independent critical sections. Imagine that process 0 wants to enter critical section A and process 1 wants to enter critical section B. Can Ricart and Agrawala's algorithm lead to deadlocks? Explain your answer.

   Answer —

   It depends on the ground rules. If processes enter critical regions strictly sequentially, that is, a process in a critical region may not attempt to enter another one, then there is no way that it can block while holding a resource (i.e., a critical section) that some other process wants. The system is then deadlock free. On the other hand, if process 0 may enter critical region A and then try to enter critical region B, a deadlock can occur if some other process tries to acquire them in the reverse order. The Ricart and Agrawala algorithm itself does not contribute to deadlock since each critical region is handled independently of all the others.

3. Linearizability assumes the existence of a global clock. However, with strict consistency we showed that such an assumption is not realistic for most distributed systems. Can linearizability be implemented for physically distri-buted data stores?

   Answer —

   Yes. Linearizability assumes loosely synchronized clocks, that is, it assumes that several events may happen within the same time slot. Those events need to be ranked adhering to sequential consistency.

4. When using a lease, is it necessary that the clocks of a client and the server, respectively, are tightly synchronized?

   Answer —

   No. If the client takes a pessimistic view concerning the level at which its clock is synchronized with that of the server, it will attempt to obtain a new lease far before the current one expires.

5. A file is replicated on 10 servers. List all the combinations of read quorum and write quorum that are permitted by the voting algorithm.

   Answer —

   The read and write quorum values, respectively denoted by N_R and N_W should satisfy the following inequalities —

   $$N_R + N_W > N \quad (1)$$

   $$N_W > N/2 \quad (2)$$

   Here $N = 10$. So the possible values of N_W are 6, 7, 8, 9, 10.
   For $N_W = 6$, the possible values of N_R are 5, 6, 7, 8, 9, 10.
   For $N_W = 7$, the possible values of N_R are 4, 5, 6, 7, 8, 9, 10.
   For $N_W = 8$, the possible values of N_R are 3, 4, 5, 6, 7, 8, 9, 10.
   For $N_W = 9$, the possible values of N_R are 2, 3, 4, 5, 6, 7, 8, 9, 10.
   For $N_W = 10$, the possible values of N_R are 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

6. Can the model of TMR handle Byzantine failures?

   Answer —

   Absolutely. The whole discussion assumed that failing elements put out random results, which are the same as Byzantine failures.

7. In the 2-phase commit protocol, why can blocking never be completely eliminated, even when the participants elect a new coordinator?

   Answer —

   After the election, the new coordinator may crash as well. In this case, the remaining participants can also not reach a final decision, because this requires the vote from the newly elected coordinator, just as before.