Homework 5

Md Ali
CS 550: Advanced Operating Systems

November 8, 2020

Exercise 1. What is view synchronous? What is atomic message?

Proof. View synchronous is an abstraction that results from the combination of group membership and reliable broadcast, this ensure that the delivery of message is coordinated with installation of views. An atomic message is a broad cast in fault-tolerant distributed computing where every single process in a system of contain multiple processes receive the same set of messages in the same exact order. □

Exercise 2. What is coordinated checkpointing?

Proof. Coordinated checkpointing help tolerate failures with processes must ensure that their checkpoints are consistent. □

Exercise 3. What is a Byzantine failure? How many replicas are needed to survive a k component fail in Byzantine failure? Why?

Proof. Byzantine failures, also know as arbitrary failures, this type of failure is similar to a fail-stop failure mode except the Byzantine has no restriction. This means that the failed node can generate arbitrary data, like for example sending data that makes it look like a functioning node. The amount of replicas that are needed to survive a k component fail in a Byzantine failure is $3k + 1$ replicas for $k > 0$. This is because theoretically speaking, no system that can assume a faulty node of greater than 1/3 of the entire system. For this case we have to have the replicas to be greater than $3k$, hence the smallest amount of replicas that are needed is $3k + 1$. □

Exercise 4. Consider a Web browser that returns an outdated cached page instead of more recent one that had been updated at the server. Is this a failure, and if so, what kind of failure?

Proof. Whether or not it is a failure depends on the consistency that in place for the use for the user. Now, for example, if the web browser gives a client-centric layout then this would simply return a page from the cache without checking its consistency which in turn is a possibility to have a response failure. □

Exercise 5. We have stated that totally ordered multicasting using Lamport's logical clocks does not scale. Explain why.

Proof. Using totally ordered multicasting using Lamport's logical clocks must have all servers up and running which in turn means that if one server is slow or completely unresponsive, this will have to be notified by every single server in the system. This will ultimately hinder performance, so as the number of servers grow, this problem becomes way bigger than typical, so in turn, we can't scale this system easily. □

Exercise 6. In reliable multicasting, is it always necessary that the communication layer keeps a copy of a message for retransmission purposes?

Proof. It isn't always necessary, take for example transferring files. In transferring files, only the data that is still available at the application level, hence there is no need for the communication layer to maintain a copy of the data. □

Exercise 7. In the two-phase commit protocol, why can blocking never be eliminated, even when the participants select a new coordinator?

Proof. Even if there were a new coordinator, this new coordinator may crash as well. This makes the remaining participants not being able to make a final decision from the new coordinator. □