You are to work on the following questions *alone*. Typeset your answers and submit as a PDF on Canvas. **Try to keep your answers short.**

1. In this problem set, you will deepen your understanding of distributed transaction protocols by examining the differences between Spanner and FaRM.

   Consider a read-write transaction, where a client or app-server performs a sequence of reads and writes to stored data identified by either keys or memory locations and then tries to commit the series of operations as an atomic and durable transaction. Compare and contrast how Spanner and FaRM handle the different steps involved in executing a transaction.

   (a) **Read operations:** What are the differences between the two systems regarding how they handle reads (e.g., what logic is executed by Spanner/FaRM nodes and clients, what are the communication steps required in performing the read)?

   (b) **Write operations:** What are the differences between the two systems regarding how they handle writes (e.g., what logic is executed by Spanner/FaRM nodes and clients, what are the communication steps required in performing the write)?

   (c) **Prepare and commit operations:** When a client is ready to commit a transaction, it performs a two-phase commit (2PC). What are the differences between the two systems as to how they perform the 2PC? Which of the two systems incurs more "communication" rounds?

2. Now consider a read-only transaction in the context of Spanner.

   (a) How does the handling of reads inside a read-only transaction differ from the handling of reads in read-write transactions?

   (b) What are the performance benefits provided by Spanner for read-only transactions?

3. Let the workload be predominantly read-only transactions. Which of Spanner and FaRM scales better in terms of transaction throughput as we increase the number of nodes in each shard? Justify your answer.

4. Consider the Spanner system. Let the error bound for the clock be ε, i.e., if *TT.now()* returns a time interval $[t-\varepsilon, t+\varepsilon]$ if the local clock is t. Further, let δ be the one-way latency delay of messages inside the datacenter. (Assume the message latency is a constant between any pair of nodes.)

   (a) If there are no other concurrent transactions, what is the latency associated with performing a read-only transaction that involves data resident in a single shard?

   (b) Perform a similar analysis for read-write transactions. Assume that the data is resident in a single shard and ignore the cost of the read phase of the transaction. What is the latency experienced by a client when it is ready to commit a transaction? This latency is the delay experienced by the client, also located inside the datacenter, when it is ready to commit its transaction.