Assignment 2 part A

Due date: June 12th at 11:59pm Waterloo time

ECE 454 / 750: Distributed Computing

Instructor: Dr. Wojciech Golab wgolab@uwaterloo.ca
Assignment TA: Mohammad Hamdaqa
mhamdaqa@uwaterloo.ca

A few house rules

Collaboration:

- groups of 1, 2 or 3 students
- undergraduate and graduate students may be in the same group

Submission:

- include a cover page containing the names and student numbers of all group members
- electronic submission (one per group) using drop box in LEARN
- only use one of the following formats: pdf, doc, docx, rtf, ppt, or pptx

Evaluation:

- part A of A2 is worth 1/3 of the total weight of A2
- if in doubt, please post questions in Piazza!

Question 1: RPCs

This question concerns the latency of a remote procedure call (RPC), which is the length of time from when the client invokes the RPC to when the client receives the return value of the procedure.

Assume the following: one-way network delay of 100ms between client and server, and 50ms processing delay in the service handler.

Part (a): Estimate the latency of a synchronous RPC, as illustrated in slide 5 of lecture module 04.

Part (b): Estimate the latency of an asynchronous RPC, as illustrated in slide 15 of lecture module 04.

Part (c): Estimate the maximum throughput (RPCs per second) for synchronous calls if the service handler runs inside THsHaServer with a pool of four worker threads on an eight-core processor.

Question 2: NTP

Consider NTP as described in lecture module 06a. Suppose that the actual offset of B relative to A is 1ms, and that the network delays dT_{req} and dT_{res} are independent random variables distributed uniformly over the interval [8ms, 12ms].

Part (a): Determine the probability distribution of θ, the estimate of the offset of host B relative to host A.

Part (b): Determine the probability distribution of δ, the estimate of the network delay between A and B.

Part (c): If NTP collects only one (θ, δ) pair, determine the probability that the estimated θ is in the interval [0.9ms, 1.1ms].

Question 3: logical clocks

Consider slide 17 of lecture module 06a, which shows an example of vector clocks. Observe that each event (send or receive) is tagged with a vector clock indicating the time just after the event has been processed.

Part (a): List all the events that happen before the specific event with vector clock [A:blank, B:3, C:1].

Part (b): List all the events that happen after the specific event with vector clock [A:2 B:2, C:1]. (Y "happens after" X if X happens before Y.)

Part (c): List all the events that are concurrent with the specific event with vector clock [A:3 B:3, C:3].

Question 4: coordination

Consider the Bully Algorithm for leader election, as presented in lecture module 06b. Suppose that one process is permitted to fail by crashing during the execution of the algorithm.

Part (a): Explain how the algorithm breaks under the above assumption. Use illustrations similar to slides 19 and 20 of module 06b. State clearly what correctness property breaks.

Part (b): Of the three correctness properties described in slide 16, are there any that continue to hold in all possible executions even if processes are allowed to crash during the execution of the algorithm? Give a proof sketch.

Note: in this question "crash" refers to a permanent failure

Question 5: consistency

Use the techniques taught in lecture (see module07a - Consistency and Replication - detailed examples) to determine whether the execution shown below is:

a) sequentially consistent

b) causally consistent

Show your work in detail.

P1:	W(x)a			W(x)c		
P2:		R(x)a	W(x)b			
P3:		R(x)a			R(x)c	R(x)b
P4:		R(x)a			R(x)b	R(x)c

Question 6: NoSQL

A small start-up company is setting up a Cassandra cluster distributed across three data centers: one near Seattle Washington, one near Atlanta Georgia, and one near Frankfurt Germany. Each data object will be replicated three ways, with one copy in each data center.

Part (a): Suppose that a client application running in the Seattle data center requires a latency bound of ≤50ms for a Get request. Which of the client-side consistency settings (ONE, QUORUM, ALL) can be used in this particular case assuming no failures?

Answer this question using speed of light calculations, assuming that all messages travel at 2/3 the speed of light in a vacuum and that processing delays are negligible.

Question 6: NoSQL (cont.)

Part (b): Which of the client-side consistency settings ensure that a Get operation can complete even if one data center becomes unavailable?

Part (c): What combinations of client-side consistency settings for Get and Put operations can be used to achieve strong consistency (as defined by Werner Vogels) in the absence of failures?

Part (d): Suppose that the Atlanta data center becomes unavailable due to a natural disaster. How does your answer to part (a) change?