15-440: Distributed Systems
Problem Solving Assignment 2

School of Computer Science
Carnegie Mellon University, Qatar
Fall 2015

Assigned Date: September 14, 2015
Due Date: September 30, 2015

1) Concurrency transparency is a desirable goal for distributed systems. Can we assume that such a property would be automatically provided by *centralized* distributed systems? Discuss. (**Points: 5**)

2) In many communication systems, a call to a *send* operation sets a timer to guard against hanging the client forever in case the server crashes. Suppose that a fault-tolerant system is implemented using many processors for all clients and servers. Consequently, the probability for a client or a server to crash becomes effectively zero. Would it be safe to fully eliminate timeouts in such a system? (**Points: 5**)

3) The original HTTP protocol used TCP/IP as the underlying network protocol. For each page, graphic, or applet, a separate TCP session was constructed, used, and torn down. Because of the overhead of building and destroying TCP/IP connections, performance problems resulted from this implementation method. Would using UDP rather than TCP be a good alternative? What other changes could you make to improve HTTP performance? (**Points: 8**)

4) How do mobile agents compare to static clients making remote invocations? Can web crawlers be considered as mobile agents? Discuss. (**Points: 4**)

5) As a designer of distributed systems, what would you adopt for multimedia data streams, a synchronous or an asynchronous distributed system? Justify your answer. (**Points: 5**)

6) As a designer of distributed systems, what aspects would you select for a system running on a totally reliable network? Explain and justify your answer. (**Points: 5**)

7) Some designers of distributed systems suggest encapsulating messages with timestamps for transmitting them over secure channels. Do you think this is a good idea? Explain your answer. (**Points: 5**)

8) Explain how could a transmitted sequence of packets through a wide area network arrive unordered at their destination (i.e., in an order that differs from that in which they were sent). Why cannot this happen in a local area network? (**Points: 5**)

9) The lower layers of the OSI network model provide datagram service, with no delivery guarantees for messages. A transport-layer protocol such as TCP is used to provide reliability. Discuss the advantages and disadvantages of supporting reliable message delivery at the lowest possible layer. (**Points: 8**)

10) Some designers of distributed systems suggest that a way to handle parameter conversion in RPC systems is to have each sender machine sending parameters in its native representation, with the receiving machine(s) doing the translation (if necessary). Can this work? Discuss. (**Points: 6**)

11) A client makes remote procedure calls to a server. The client takes 5 milliseconds to compute the arguments for each request, and the server takes 10 milliseconds to process each request. The local operating system processing time for each send or receive operation is 0.5 milliseconds, and the network time to transmit each request or reply message is 3 milliseconds. Marshalling or unmarshalling takes 0.5 milliseconds per message.

Calculate the time taken by the client to generate and return from two requests:

(i) If it is single-threaded, and

(ii) If it has two threads that can make requests concurrently on a single processor.

You can ignore context-switching times. Is there a need for asynchronous RPC if client and server processes are multi-threaded? (**Points: 10**)

12) Devise a scenario in which multicasts sent by different clients are delivered in different orders at two group members. Assume that some form of message retransmissions is in use, but that messages that are not dropped arrive ordered. Suggest how recipients might remedy this situation. (**Points: 5**)

13) If a communication paradigm is asynchronous, is it also time-uncoupled? Explain your answer with examples as appropriate. (**Points: 4**)

14) Consider a cluster that has little availability support. Upon a node failure, the following sequence of events takes place:

a. The entire systems is shut down and powered off.
b. The faulty node is replaced if the failure is in hardware.
c. The system is powered on and rebooted.
d. The user application is reloaded and rerun from the start.

Assume one of the cluster nodes fails every 100 hours. Other parts of the cluster never fail. Steps 1 through 3 take two hours. On average, the mean time for step 4 is two hours. What is the availability of the cluster? What is the yearly failure cost if each one-hour downtime costs $82,500? (**Points: 6**)

15) Repeat question 14, but assume that the cluster now has much increased availability support. Upon a node failure, its workload automatically fails over to other nodes. The failover time is only six minutes. Meanwhile, the cluster has hot swap capability: The faulty node is taken off the cluster, repaired, replugged, and rebooted, and it rejoins the cluster, all without impacting the rest of the cluster. What is the availability of this ideal cluster, and what is the yearly failure cost? (**Points: 4**)

16) Read the research paper titled "***An Agent-based Optimization Framework for Mobile-Cloud Computing"*** by Pelin Angin and Bharat Bhargava, and answer the following questions (**Points: 15**):

(i) Summarize the paper in 2 paragraphs
(ii) Discuss at least two strengths and two weaknesses of the paper