Two hours

Question ONE is COMPULSORY

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Distributed Computing

Date: Friday 20th May 2016
Time: 09:45 - 11:45

Please answer Question ONE and also TWO other Questions from the remaining THREE Questions provided

This is a CLOSED book examination
The use of electronic calculators is NOT permitted

[PTO]
1. **Compulsory**
   
a) What is a Java servlet?  
(2 marks)

b) What is the main assumption on which Cristian’s clock synchronisation algorithm is based?  
(2 marks)

c) Explain briefly the difference between a name server and a directory server.  
(2 marks)

d) Explain briefly what the relevance of the Byzantine generals problem is for distributed computing systems.  
(2 marks)

e) Why is it practically impossible to achieve strict consistency in a distributed system?  
(2 marks)

f) Explain what copy-restore is and why it is used in RPC.  
(2 marks)

g) What is the worst-case scenario for the number of messages in the Bully algorithm? How many messages do you expect in this case?  
(2 marks)

h) What is the key difference between caching and replication?  
(2 marks)

i) Explain briefly what the use of a checkpoint is in the context of distributed transactions.  
(2 marks)

j) In the context of lab exercise 2, what would you do to launch a denial of service attack against the server?  
(2 marks)
Please answer any two Questions from Questions TWO, THREE and FOUR

2. a) Explain briefly what is wrong with the assumption “latency is zero” in the context of distributed computing. Why is it considered a common fallacy?
   (3 marks)

b) Explain briefly what the four properties commonly denoted by the acronym ACID are when referring to transactions.
   (3 marks)

c) Consider the figure below, which shows 4 processes and a number of communication events taking place over a period of time.

```
P1   P2
    ↘
    ↓
P3   P4
```

Calculate the value of Lamport clocks and vector clocks for each of the 12 events shown above. You can assume that all logical clocks start initially with zeros.
   (6 marks)

d) The following four processes access a shared variable $x$. Each process accesses a different replica of the store used to hold this variable. Before any process starts executing, the value of $x$ is 0 in all the replicas.

```
Process 1: $x=1$;  
Process 2: $x=2$;  
Process 3: $y=0$;  
Process 4: $z=2$;

if($x==1$)  
if($x==2$)  
y=$y+1$;  
y=$y+2$;  
if($x==1$)  
if($x==2$)  
z=$z+2$;  
z=$z+1$;
```

(i) When all four processes have completed executing the statements given, are 3 and 5 possible values of $y$ and $z$ respectively, if the replication uses the sequential consistency model? Justify your answer.
   (4 marks)

(ii) When all four processes have completed executing the statements given, are 3 and 5 possible values of $y$ and $z$ respectively, if the replication uses the causal consistency model? Justify your answer.
   (4 marks)
3. a) Explain briefly what the role of a client stub and a server stub is in RPC. 
(3 marks)

b) Describe in detail how the two phase-commit protocol can implement distributed transactions. 
(4 marks)

c) For the needs of a scientific experiment, twenty data collection instruments are deployed at twenty different locations around the world, one instrument at each location. These instruments collect data on a continuous basis, which can be transmitted to a remote server for storage and processing. To facilitate data storage and processing three servers are going to be placed at three of the twenty different locations. Describe a strategy to decide what locations to use to place the servers. 
(5 marks)

d) In a client-server application, assume that each client request is added to a server queue and three servers can serve requests from the queue. Server A can serve 22 requests per second. Server B can serve 32 requests per second. Server C can serve 46 requests per second.

(i) Assume that at a certain point in time there are 400 requests in the queue. How would you allocate these requests to the servers to achieve load balancing? 
(2 marks)

(ii) If at most 50 clients can operate in parallel, how many requests per second would you advise each client to make so that the server queue is not flooded with requests? 
(2 marks)

(iii) Describe how in the general case of \( n \) servers, each capable of serving \( s_i \) requests per second, you would allocate \( x \) requests from the queue to achieve load balancing. State any assumptions you make. 
(4 marks)
4. This question is based on your understanding of transactions and distributed transactions, in the context of lab exercise 2. The server that processes all requests (reservation, cancellation, availability and bookings retrieval) hosts a database for storing reservations and some message-oriented middleware, which provides an in-queue (to queue requests) and an out-queue (to queue responses to messages).

Suppose there are three messages in the in-queue: two from Tom and one from Jerry. Suppose also that Tom already holds a reservation for slot 544 and Jerry holds a reservation for slots 541 and 543 (assume a total of 1000 slots). The state of the database and queues is shown below.

The pseudo-code that processes a reservation request is shown below. It assumes that all messages are delivered exactly once by the underlying messaging infrastructure. Also, assume that the maximum permitted number of reservations a student can hold at any one time is two, that is, max_res=2.

```
begin_tx;   // this is an ACID transaction to reserve a slot
  dequeue a request from the in-queue
  if requested slot is 'free' and
    number of reservations by student < max_res then
    reserve slot for the student;
    send 'slot reserved' response message;
  else if number of reservations by student >= max_res then
    send 'fail: too many reservations' response message;
  else if slot is not 'free' then
    send 'fail: slot is not free' response message;
  endif
commit;   // transaction commits
```

(Question 4 continues on the following page)
a) Show what the state of the slots table in the database would look like after all in-queue messages are executed (assuming that messages are executed left-to-right, that is, the first message executed is the request from Jerry). Your answer should indicate clearly the response messages generated in each case. (4 marks)

b) Show via an execution sequence (an order in which statements are executed), how Tom can hold more than the maximum permitted number of reservations if two threads are concurrently processing messages from the in-queue and if the code is not enclosed in an ACID transaction (4 marks)

c) After which statements in the pseudo-code for processing a reservation request can the server crash? Describe the state of the database, in-queue, and out-queue after the crash and recovery. (4 marks)

d) Suppose that four threads, running on different CPUs, are concurrently processing messages from the in-queue. Every incoming message requires processing time of 40 milliseconds and, on average, there are about 10000 messages per hour sent to the server. How long do you expect to have messages waiting in the in-queue before their processing starts? (4 marks)

e) Propose a strategy that the server may use to adapt on-the-fly the number of threads so that the length of the in-queue is minimal and threads are kept as busy as possible. State any assumptions you make. (4 marks)