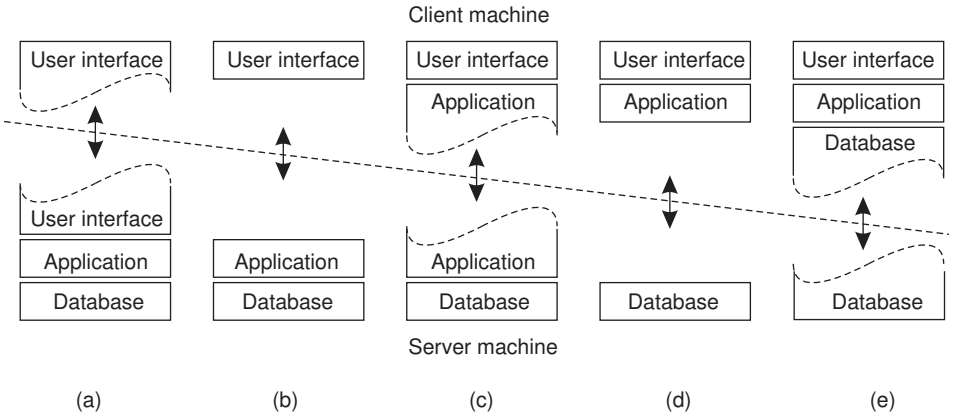
Department Computer Science	**Distributed Systems**
VU University	**19.08.2010**

MAKE SURE THAT YOUR HANDWRITING IS READABLE

1a Give an example organization for each of the following two-tiered client-server architectures. *5pt*



(a) dumb terminal, (b) X terminal, (c) Content management systems or other Web-based applications, (d) many transactional applications in which the client acts as the transaction manager, (e) Web browser with built-in cache.

1b For a long time we could observe a trend from thin clients to fat clients, which then reversed. Why did we tend to move away from fat clients again? *5pt*

The reason is actually simple: fat clients imply that lots of software is to be installed and maintained on client machines, which, in turn, are generaly owned by end users. This requires a lot of remote support and caused management problems.

1c More recently, we are seeing that fat clients are gaining popularity, notably when dealing with Web applications. What happened? *5pt*

Modern Web browsers allow automatic installing, checking and updating of plugins, which are often full-blown applications. Management suddenly became much easier again.

2a Explain the principle of TCP handoff. *5pt*

With TCP handoff, an incoming connection request is forwarded by a switch to a specific server, which then sends the response back directly to the client, using the network address of the switch. It is important that you mention that the server spoofs the switch.

2b Explain how DNS redirection works. What is the underlying assumption for this technique to work? *5pt*

A DNS server receives a request from a client to look up a specific domain name. Based on the client's address, it then returns an address that is best for that client. It assumes that the IP address, which it finds in the request, actually belongs to the real client and is not the address of another client-independent DNS server, as would be the case with recursive lookups.

2c Explain how HTTP redirection works. What do you see as its main drawback? *5pt*

In an HTTP redirect, a URL acts as a soft link to another name, which is subsequently returned to the client for further processing. The main disadvantage is the nontransparency of the name resolution, exemplified by the fact that the client will eventually see the resolved URL of the final server.

3a What is application-level multicasting (ALM)? *5pt*

ALM refers to multicasting over an overlay network, i.e., a network constructed by means of application-level connections between peers. An example is the MBone, but also most P2P systems supporting multicasting.

3b A good ALM scheme establishes a low value for what is known as stretch. What does this mean? *5pt*

Stretch measures the ratio in the delay between two nodes in the overlay associated with ALM, and the delay between those two nodes when traversing an optimal path in the underlying network.

3c How well would you think that peer-to-peer networks such as Chord perform in terms of stretch? *5pt*

Without any further measurements, they will generally perform badly: nodes are placed more or less at random across the Internet which means that we can expect to have high stretch values.

4a Explain Lamport's happened-before relation and how it can be established through fully decentralized logical clocks. *5pt*

The happened-before relation says that if (1) a and b are two successive events in the same process, then $a \rightarrow b$, (2) if a is the sending of a message m and b is the receipt of m, then $a \rightarrow b$, or (3) if $a \rightarrow b$, $b \rightarrow c$, then $a \rightarrow c$. Every process P_i keeps a local counter C_i, which is initially set to 0. When a message m is sent, it receives a timestamp $T(m)$ that is set to C_i, after which C_i is incremented by 1. When process P_j receives a message, it sets its own value C_j to $\max\{C_j + 1, T(m) + 1\}$.

4b Explain how totally-ordered multicasting can be implemented with Lamport's logical clocks. *10pt*

See also book, page 247. The essence is as follows. When process P_i wants to update the replicas, it broadcasts the update m_i to itself and all other processes. Message m_i is timestamped P_i's current value of C_i. When a process P_j receives m_i it puts it in a local queue, ordered by $T(m_i)$, and broadcasts an acknowledgement. Only when m_i is at the head of the queue, and for every process there is a message queued with a higher timestamp, will m_i be passed to the application. Lamport assumes that messages are sent FIFO-wise, and that no messages are lost.

5a Give a simple, yet precise definition of reliable multicasting. *5pt*

The trick in being accurate is freezing the set of senders and receivers. In that case, a simple definition is that if the set of senders and receivers does not change during the multicast, a message will be received by all intended recipients.

5b What is a reliable multicast that is virtually synchronous? *5pt*

In this case, we guarantee that the message sent to the intended group of recipients, is received by all nonfaulty processes. If the sender crashes during the multicast, the recipients will each receive the message, or none will.

5c Explain why virtual synchrony is so convenient for applications. *5pt*

The reason is simplicity: members of a distributed application need not question whether the other group members have received a message: if a message has been delivered to a constituent process, that process can safely assume that all its fellow processes will have seen that message as well.

6a Explain what temporal (de)coupling and referential (de)coupling of processes means, and for each of the four possible combinations, give an example how two processes communicate. *8pt*

See book, chapter 12.

6b Explain what content-based routing is. *7pt*

CBR is established in overlay networks in which a message is routed from sender to receiver(s) based entirely on the content of its payload. CBR establishes referential decoupling of processes and requires that intermediate nodes have installed filters that process incoming messages in order to subsequently decide in which direction that message must be forwarded. See also page 602.

Grading: *The final grade is calculated by accumulating the scores per question (maximum: 90 points), and adding 10 bonus points. The maximum total is therefore 100 points.*