Department Computer Science **Distributed Systems**
Vrije Universiteit **16.06.2003**

1a Explain what relocation transparency is and give a concrete example of its application. *5pt*

When a system provides relocation transparency it hides the fact that it may move resources between locations. Such moving behind the scenes is typically done for load balancing.

1b Caching and replication are generally applied as scaling techniques, but introduce another scaling problem. What is this problem and how can it be solved? *5pt*

They both lead to global synchronization in the face of updates. The only solution is to tolerate inconsistencies by adopting weak consistency models.

2a What is isochronous transmission mode? *5pt*

A continuous data stream is in isochronous transmission mode when its packets are subject to a maximum and minimum end-to-end delay.

2b Give an example of a complex continuous data stream, and explain how synchronization in a such a stream is generally accomplished. Motivate why synchronization is done that way. *5pt*

A video stream combined with a stereo audio stream. Synchronization is handled by grouping packets from the different streams and sending them together. This grouping ensures that the packets from the different streams stay together and are subject to the same delay and jitter, making it much easier for the receiver to play them back at the same time.

3 Sketch a simple solution for supporting migration transparency for mobile remote objects. *5pt*

A simple solution is to let the client stub keep track of where the object is, in combination with forwarding pointers at the object's side. Alternatively, a mobile object can keep track of its clients, and informs them when it moves to another location.

4a Explain how a remote name space can be mounted. *5pt*

The mount point in the source name space contains information on the access protocol for the remote name space, as well as a global name that refers to the mounting pointof that name space. When resolving a name, the access protocol is executed to obtain access to the remote name space, which, in turn, is instructed to resolve the remainder of the path name.

4b Recursive name resolution is more effective than iterative name resolution. Why? *5pt*

With recursive name resolution, a name server can serve many more clients. Iterative name resolvers are necessarily restricted to a smaller group. As a consequence, clients at different locations that need to have the same name resolved, cannot make use of the same cache.

4c Explain the difference between a name service and a directory service, and why the latter is often more difficult to scale worldwide. *10pt*

The essential difference is that a directory service supports lookups based on (attribute,value) pairs, whereas a name service can resolve only path names. As a consequence, a directory service may require to examine every one of its objects to see whether that object has an attribute with the required value. This exhaustive search imposes inherent scalability problems.

5a Show that Lamport timestamps are insufficient for capturing causality. *5pt*

We need to show that if $C(a) < C(b)$, that we do not necessarily have that $a \rightarrow b$. Consider the sending of message D in Figure 5-7(b). This event will be timestamped 69. However, before that an event timestamped 70 could have taken place in the third process, which actually happened before the sending of D. In fact, it was an independent event.

5b Explain how Lamport timestamps can be used to realize totally ordered multicasting. *5pt*

See book, pages 255–256

5c Consider a distributed system that promises to preserve causality in the delivery of messages. Explain what this promise means, and why the system cannot, in general, keep this promise. *5pt*

The promise states that if a → b, then the system will ensure that a is delivered before b. The problem is that the system must assume that it can record every *event. If two processes communicate* out-of-band, *then that communication will not be caught, potentially violating the preservation of causal relations.*

6a When is an execution of operations by concurrent processes sequentially consistent? *5pt*

When the result is the same as if the operations were executed in some sequential order and all operations of each individual process appear in this sequence in the order specified by its program.

6b What is the essential difference between data-centric and client-centric consistency models? *5pt*

Data-centric consistency models take the effect into account that a write opertion by a process A may have on what is read by process B. Client-centric consistency models consider only the effect of operations local to a single, but possibly mobile process.

7a Reliable multicasting is difficult to scale to very large groups of processes. Why? *5pt*

The problem is that reliability requires that receivers acknowledge the receipt of a message. With large groups, the sender will be swamped with acks. In an optimistic approach, the receivers may decide to send only a negative ackknowledge when they notice a message is missing. But even in that case there is still the risk of a feedback implosion at the sender.

7b Consider the problem of reliably multicasting a message to a large group of processes. If it is required that the message is only *eventually* delivered to each process, what would your solution be? *5pt*

The fact that eventual delivery is required is important, as it allows for an epidemic distribution protocol, such as anti-entropy or gossiping. The problem that remains to be solved is that every process will get the message. This requires a combination of push-pull distribution.

7c Explain what atomic multicasting is. *5pt*

With an atomic multicast, a message is delivered to all processes, or none at all. Moreover, all messages are delivered in the same order to all nonfaulty process.

8 What is a piecewise deterministic execution model, and why is it so useful? *5pt*

In such a model, the execution of a process is assumed to take place as a series of intervals, each interval starting with a nondeterministic event, but for the rest containing only deterministic events. This model allows the replay of processes, provided the nondeterministic events have been logged. The combination of replaying a process and checkpointing is often a cheaper alternative to taking only cheackpoints.

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.*