Dept. Comp. Sc.	Distributed Systems
Vrije Universiteit	26.01.2004

Part I

This part covers the same material as the midterm exam.

1a Explain what is meant by a *remote-object model*. *5pt*

In this model, a distributed supports a notion of objects in which an object's state is always located at one site. The state is never transparently distributed across multiple sites. Hoever, clients can transparently access an object through proxies that provide implementations of an object's interfaces.

1b What happens when two clients simultaneously invoke a *synchronized* method of a Java remote object? *5pt*

It depends whether the clients reside on the same machine. If they do, one is blocked until the invocation completes. If not, it depends on the implementation of the object at the server. If the server serializes object invocations everything may work fine. However, if the server does not implement any serialization of requests, the two invocations may be carried in an interleaved fashion, leaving the object in an inconsistent state.

1c Consider a Java remote object `mutex` that provides two methods: `lock()` and `unlock()`. This object could be used, for example, to serialze access to shared objects. Sketch an implementation of `mutex`. *5pt*

The crucial issue here, is that access to `mutex` *should itself be serialized. We could do this partly by declaring the two methods* synchronized*. However, the server needs to ensure that any invocation on the object is serialized. A simple trick is to let the server start threads that access a* local*, synchronized object that implements* `mutex`*.*

2a What is the advantage of using a true identifier as a reference to mobile or replicated objects? *5pt*

A true identifier can be considered as just a random bit string which is completely location independent. By maintaining a separate mapping from ID to location(s), and a separate one from human-friendly name to ID, we can allow changes in names to be done independently from changes in location. As a consequence, we can use traditional naming systems that assume that mappings do not change often to resolve human-freindely names (such as DNS), and specialized location services for tracking highly mobile objects.

2b Tracking mobile objects can be done by means of forwarding pointers. What are the drawbacks of this scheme? *5pt*

First, without taking special measures, we can create long chains of pointers which introduce considerable latency. Second, pointers need to be maintained, or otherwise a reference may break. Maintenance kan be expensive in the presence of failing nodes. In practice, chain reduction techniques are needed to solve these problems.

2c Home-based approaches for locating objects have the drawback that, in principle, the reference to the home location may never change. Give a workable DNS-based solution for this problem. *5pt*

Use another level of indirection by making use of persistent and permanent name, which can be resolved to the current *home location. Because that mapping hardly ever changes, it can easily be cached for a long time, effectively giving the same performance as a keeping the home location the same always.*

3a What is the difference between a nested and distributed transaction? *5pt*

Both type of transactions consist of subtransactions. However, the subtransactions of a nested transaction are formulated on logically different databases. Those of a distributed transaction are formulated over differently located parts of the same logical database.

3b Explain how primary two-phase locking works. *5pt*

Consider a transaction T that affects data items $x_1, \ldots, x_n$. In primary 2PL, each data item x_i has a primary copy. The scheduler on the machine on which that copy resides is responsible for granting and releasing locks. Whenever a lock is released, no other lock for any other data item affected by T may be granted. Note that this requires that the schedulers inform each other about the locks they have released.

3c Consider a transaction consisting of invocations on several Java *remote objects*. Each object is protected by its associated object adapter against concurrent access by multiple clients. Do we still need a scheduler for T? Explain your answer. *5pt*

*Yes, of course: without a scheduler you cannot achieve serialization of multiple **transactions**.*

Part II

6a Explain the difference between data-centric and client-centric consistency. *5pt*

With data-centric consistency the objective is to specify systemwide consistency on a set of shared data items in the presence of concuurent read/write operations. For client-centric consistency, consistency is defined only concerning how one specific client experiences the effects of read/write operations when moving between locations.

6b What is read-your-writes consistency, and what would a simple implementation look like? *5pt*

RYW consistency means that the effects of a write operation is always seen by the same client when that client subsequently performs a read operation. A simple implementation is to let the client-side software keep track of where and when it performed its last write operation. When issuing a read request, the latest version of the data that is to be read can be retrieved before passing it on to the client.

6c Give five different examples of replication strategies that implement sequential consistency. *5pt*

(1) all updates first propagate to a (single) master. (a) forward updates, (b) send invalidations, (c) let other replicas check at the master. Variations can be introduced by letting the master serve all data or part of the data. Another variation is to make the master mobile, or to keep it statically located. (2) Use multiple masters in combination with total-ordered multicasting, combined with (a) forwarding updates, (b) forwarding invalidations, (c) forwarding operation.

7a Explain the scalability problem in reliable multicasting. *5pt*

The real problem comes from the fact that you need to have an ACK from every receiver, or in the case of an optimistic approach a NACK when things go wrong. In both cases, there is a considerable chance that the feedback provided by the receivers will simply overload the network and/or the sender.

7b How can epidemic protocols help in achieving scalable reliable multicasting? Which assumptions do we need to make? *5pt*

If we assume that news is allowed to travel slowly, epidemics can provide statistically guarantees that eventually everyone will receive your news. The assumptions that need to be made, besides the tolerance of lazy propagation, is that there are no write-write conflicts and that messages cannot overwrite each other.

7c What is the problem that the 3-phase commit protocol solves? *5pt*

In 2PC, the situation may occur in which all processes need to wait for the coordinator to recover before they can decide to abort of commit the transaction. This may happen when all processes are in a ready state. Note that special measures need to be taken to allow processes to proceed before the coordinator recovers.

8a Explain how mounting works in NFS. *5pt*

You need to mention that a server exports subdirectories, and that a client needs to import them into a local name space. Directories imported by a server cannot be exported; the client needs to import those from the original server.

8b What is the main difference between NFS versions 3 and 4, and why did the designers make this distinction? *5pt*

The main difference lies in the stateful servers available in NFS 4. By making servers stateful, it becomes possible to hand out clients complete copies of files, while at the same time the server can stay in control by demanding that a copy should be returned. This whole-file caching by clients allows for multiple local operations to be carried out locally at the client, which is good when having to deal with wide-area traffic.

8c What are share reservations in NFS? *5pt*

You need to mention that they are used to attach a state to an open file that models the request behavior of a client, and the denial behavior by other clients once a file has been allocated. Details on what can actually be done will get you brownie points.

Final grade: *(1) Add, per part, the total points. (2) Let T denote the total points for the midterm exam ($0 \leq T \leq 45$); $D1$ the total points for part I; $D2$ the total points for part II. The final number of points E is equal to* $\max\{T, D1\} + D2 + 10$.