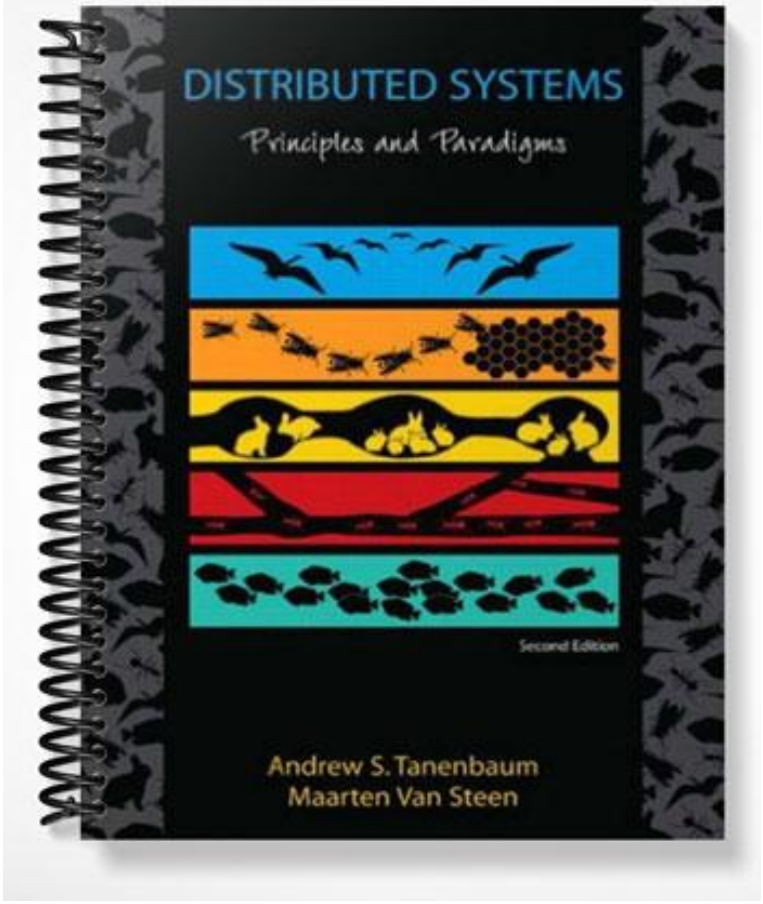
SOLUTIONS MANUAL



DISTRIBUTED SYSTEMS
PRINCIPLES AND PARADIGMS

SECOND EDITION

PROBLEM SOLUTIONS

ANDREW S. TANENBAUM

MAARTEN VAN STEEN

Vrije Universiteit
Amsterdam, The Netherlands

PRENTICE HALL

UPPER SADDLE RIVER, NJ 07458

SOLUTIONS TO CHAPTER 1 PROBLEMS

1. Q: An alternative definition for a distributed system is that of a collection of independent computers providing the view of being a *single system*, that is, it is completely hidden from users that there even multiple computers. Give an example where this view would come in very handy.

A: What immediately comes to mind is parallel computing. If one could design programs that run without any serious modifications on distributed systems that appear to be the same as nondistributed systems, life would be so much easier. Achieving a single-system view is by now considered virtually impossible when performance is in play.

2. Q: What is the role of middleware in a distributed system?

A: To enhance the distribution transparency that is missing in network operating systems. In other words, middleware aims at improving the single-system view that a distributed system should have.

3. Q: Many networked systems are organized in terms of a back office and a front office. How does organizations match with the coherent view we demand for a distributed system?

A: A mistake easily made is to assume that a distributed system as operating in an organization, should be spread across the entire organization. In practice, we see distributed systems being installed along the way that an organization is split up. In this sense, we could have a distributed system supporting back-office procedures and processes, as well as a separate front-office system. Of course, the two may be coupled, but there is no reason for letting this coupling be fully transparent.

4. Q: Explain what is meant by (distribution) transparency, and give examples of different types of transparency.

A: Distribution transparency is the phenomenon by which distribution aspects in a system are hidden from users and applications. Examples include access transparency, location transparency, migration transparency, relocation transparency, replication transparency, concurrency transparency, failure transparency, and persistence transparency.

5. Q: Why is it sometimes so hard to hide the occurrence and recovery from failures in a distributed system?

A: It is generally impossible to detect whether a server is actually down, or that it is simply slow in responding. Consequently, a system may have to report that a service is not available, although, in fact, the server is just slow.

6. Q: Why is it not always a good idea to aim at implementing the highest degree of transparency possible?

A: Aiming at the highest degree of transparency may lead to a considerable loss of performance that users are not willing to accept.

7. Q: What is an open distributed system and what benefits does openness provide?

A: An open distributed system offers services according to clearly defined rules. An open system is capable of easily interoperating with other open systems but also allows applications to be easily ported between different implementations of the same system.

8. Q: Describe precisely what is meant by a scalable system.

A: A system is scalable with respect to either its number of components, geographical size, or number and size of administrative domains, if it can grow in one or more of these dimensions without an unacceptable loss of performance.

9. Q: Scalability can be achieved by applying different techniques. What are these techniques?

A: Scaling can be achieved through distribution, replication, and caching.

10. Q: Explain what is meant by a virtual organization and give a hint on how such organizations could be implemented.

A: A virtual organization (VO) defines a group of users/applications that have access to a specified group of resources, which may be distributed across many different computers, owned by many different organizations. In effect, a VO defines who has access to what. This also suggests that the resources should keep an account of foreign users along with their access rights. This can often be done using standard access control mechanisms (like the rwx bits in UNIX), although foreign users may need to have a special account. The latter complicates matters considerably.

11. Q: When a transaction is aborted, we have said that the world is restored to its previous state, as though the transaction had never happened. We lied. Give an example where resetting the world is impossible.

A: Any situation in which physical I/O has occurred cannot be reset. For example, if the process has printed some output, the ink cannot be removed from the paper. Also, in a system that controls any kind of industrial process, it is usually impossible to undo work that has been done.

12. Q: Executing nested transactions requires some form of coordination. Explain what a coordinator should actually do.

A: A coordinator need simply ensure that if one of the nested transactions aborts, that all other subtransactions abort as well. Likewise, it should

coordinate that all of them commit when each of them can. To this end, a nested transaction should wait to commit until it is told to do so by the coordinator.

13. Q: We argued that distribution transparancy may not be in place for pervasice systems. This statement is not true for all types of transparencies. Give an example.

A: Think of migration transparency. In mnay pervasive systems, components are mobile and will need to re-establish connections when moving from one access point to another. Preferably, such handovers should be completely transparent to the user. Likewise, it can be argued that many other types of transparencies should be supported as well. However, what should not be hidden is a user is possibly accessing resources that are directly coupled to the user's current environment.

14. Q: We already gave some examples of distributed pervasive systems: home systems, electronic health-care systems, and sensor networks. Extend this list with more examples.

A: There are quite a few other examples of pervasive systems. Think of large-scale wireless mesh networks in cities or neighborhoods that provide services such as Internet access, but also form the basis for other services like a news system. There are systems for habitat monitoring (as in wildlife resorts), electronic jails by which offenders are continuously monitored, large-scale integrated sports systems, office systems deploying active badges to know about the whereabouts of their employees, and so on.

15. Q: Sketch a design for a home system consisting of a separate media server that will allow for the attachment of a wireless client. The latter is connected to (analog) audio/video equipment and transforms the digital media streams to analog output. The server runs on a separate machine, possibly connected to the Internet, but has no keyboard and/or monitor connected.

SOLUTIONS TO CHAPTER 2 PROBLEMS

1. Q: If a client and a server are placed far apart, we may see network latency dominating overall performance. How can we tackle this problem?

A: It really depends on how the client is organized. It may be possible to divide the client-side code into smaller parts that can run separately. In that case, when one part is waiting for the server to respond, we can schedule another part. Alternatively, we may be able to rearrange the client so that it can do other work after having sent a request to the server. This last solution effectively replaces the synchronous client-server communication with asynchronous one-way communication.

2. Q: What is a three-tiered client-server architecture?

A: A three-tiered client-server architecture consists of three logical layers, where each layer is, in principle, implemented at a separate machine. The highest layer consists of a client user interface, the middle layer contains the actual application, and the lowest layer implements the data that are being used.

3. Q: What is the difference between a vertical distribution and a horizontal distribution?

A: Vertical distribution refers to the distribution of the different layers in a multitiered architectures across multiple machines. In principle, each layer is implemented on a different machine. Horizontal distribution deals with the distribution of a single layer across multiple machines, such as distributing a single database.

4. Q: Consider a chain of processes P_1, P_2, ..., P_n implementing a multitiered client-server architecture. Process P_i is client of process P_{i+1}, and P_i will return a reply to P_{i-1} only after receiving a reply from P_{i+1}. What are the main problems with this organization when taking a look at the request-reply performance at process P_1?

A: Performance can be expected to be bad for large n. The problem is that each communication between two successive layers is, in principle, between two different machines. Consequently, the performance between P_1 and P_2 may also be determined by $n-2$ request-reply interactions between the other layers. Another problem is that if one machine in the chain performs badly or is even temporarily unreachable, then this will immediately degrade the performance at the highest level.

5. Q: In a structured overlay network, messages are routed according to the topology of the overlay. What is an important disadvantage of this approach?

A: The problem is that we are dealing only with *logical* paths. It may very well be the case that two nodes A and B which are neighbors in the overlay network are physically placed far apart. As a consequence, the logically short path between A and B may require routing a message along a very long path in the underlying physical network.

6. Q: Consider the CAN network from Fig. 2-0. How would you route a message from the node with coordinates (0.2,0.3) to the one with coordinates (0.9,0.6)?

A: There are several possibilities, but if we want to follow the shortest path according to a Euclidean distance, we should follow the route (0.2,0.3) $\rightarrow$ (0.6,0.7) $\rightarrow$ (0.9,0.6), which has a distance of 0.882. The alternative route (0.2,0.3) $\rightarrow$ (0.7,0.2) $\rightarrow$ (0.9,0.6) has a distance of 0.957.