Distributed Systems: Concepts and Design

Edition 3

By George Coulouris, Jean Dollimore and Tim Kindberg
Addison-Wesley, ©Pearson Education 2001

Chapter 1   Exercise Solutions

1.1   Give five types of hardware resource and five types of data or software resource that can usefully
be shared. Give examples of their sharing as it occurs in distributed systems.

1.1 Ans.

Hardware:

CPU: compute server (executes processor-intensive applications for clients), remote object server (executes methods on behalf of clients), worm program (shares cpu capacity of desktop machine with the local user). Most other servers, such as file servers, do some computation for their clients, hence their cpu is a shared resource.

memory: cache server (holds recently-accessed web pages in its RAM, for faster access by other local computers)

disk: file server, virtual disk server (see Chapter 8), video on demand server (see Chapter 15).

screen: Network window systems, such as X-11, allow processes in remote computers to update the content of windows.

printer: networked printers accept print jobs from many computers. managing them with a queuing system.

network capacity: packet transmission enables many simultaneous communication channels (streams of data) to be transmitted on the same circuits.

Data/software:

web page: web servers enable multiple clients to share read-only page content (usually stored in a file, but sometimes generated on-the-fly).

file: file servers enable multiple clients to share read-write files. Conflicting updates may result in inconsistent results. Most useful for files that change infrequently, such as software binaries.

object: possibilities for software objects are limitless. E.g. shared whiteboard, shared diary, room booking system, etc.

database: databases are intended to record the definitive state of some related sets of data. They have been shared ever since multi-user computers appeared. They include techniques to manage concurrent updates.

newsgroup content: The *netnews* system makes read-only copies of the recently-posted news items available to clients throughout the Internet. A copy of newsgroup content is maintained at each netnews server that is an approximate replica of those at other servers. Each server makes its data available to multiple clients.

video/audio stream: Servers can store entire videos on disk and deliver them at playback speed to multiple clients simultaneously.

exclusive lock: a system-level object provided by a lock server, enabling several clients to coordinate their use of a resource (such as printer that does not include a queuing scheme).

1.2 How might the clocks in two computers that are linked by a local network be synchronized without reference to an external time source? What factors limit the accuracy of the procedure you have described? How could the clocks in a large number of computers connected by the Internet be synchronized? Discuss the accuracy of that procedure.

1.2 Ans.

Several time synchronization protocols are described in Section 10.3. One of these is Cristian's protocol. Briefly, the round trip time *t* to send a message and a reply between computer A and computer B is measured by repeated tests; then computer A sends its clock setting *T* to computer B. B sets its clock to *T+t/2*. The setting can be refined by repetition. The procedure is subject to inaccuracy because of contention for the use of the local network from other computers and delays in the processing the messages in the operating systems of A and B. For a local network, the accuracy is probably within 1 ms.

For a large number of computers, one computer should be nominated to act as the time server and it should carry out Cristian's protocol with all of them. The protocol can be initiated by each in turn. Additional inaccuracies arise in the Internet because messages are delayed as they pass through switches in wider area networks. For a wide area network the accuracy is probably within 5-10 ms. These answers do not take into account the need for fault-tolerance. See Chapter 10 for further details.

1.3 A user arrives at a railway station that she has never visited before, carrying a PDA that is capable of wireless networking. Suggest how the user could be provided with information about the local services and amenities at that station, without entering the station's name or attributes. What technical challenges must be overcome?

1.3 Ans.

The user must be able to acquire the address of locally relevant information as automatically as possible. One method is for the local wireless network to provide the URL of web pages about the locality over a local wireless network.

For this to work: (1) the user must run a program on her device that listens for these URLs, and which gives the user sufficient control that she is not swamped by unwanted URLs of the places she passes through; and (2) the means of propagating the URL (e.g. infrared or an 802.11 wireless LAN) should have a reach that corresponds to the physical spread of the place itself.

1.4 What are the advantages and disadvantages of HTML, URLs and HTTP as core technologies for information browsing? Are any of these technologies suitable as a basis for client-server computing in general?

1.4 Ans.

HTML is a relatively straightforward language to parse and render but it confuses presentation with the underlying data that is being presented.

URLs are efficient resource locators but they are not sufficiently rich as resource links. For example, they may point at a resource that has been relocated or destroyed; their granularity (a whole resource) is too coarse-grained for many purposes.

HTTP is a simple protocol that can be implemented with a small footprint, and which can be put to use in many types of content transfer and other types of service. Its verbosity (HTML messages tend to contain many strings) makes it inefficient for passing small amounts of data.

HTTP and URLs are acceptable as a basis for client-server computing except that (a) there is no strong type-checking (web services operate by-value type checking without compiler support), (b) there is the inefficiency that we have mentioned.

1.5 Use the World Wide Web as an example to illustrate the concept of resource sharing, client and server.

Resources in the World Wide Web and other services are named by URLs. What do the initials

URL denote? Give examples of three different sorts of web resources that can be named by URLs.

1.5 Ans.

Web Pages are examples of resources that are shared. These resources are managed by Web servers.

Client-server architecture. The Web Browser is a client program (e.g. Netscape) that runs on the user's computer. The Web server accesses local files containing the Web pages and then supplies them to client browser processes.

URL - Uniform Resource Locator

(3 of the following a file or a image, movies, sound, anything that can be rendered, a query to a database or to a search engine.

1.6   Give an example of a URL.

List the three main components of a URL, stating how their boundaries are denoted and illustrating each one from your example.

To what extent is a URL location transparent?

1.6 Ans.

http://www.dcs.qmw.ac.uk/research/distrib/index.html

- The protocol to use. the part before the colon, in the example the protocol to use is http ("HyperText Transport Protocol").

- The part between // and / is the Domain name of the Web server host www.dcs.qmw.ac.uk.

- The remainder refers to information on that host - named within the top level directory used by that Web server   research/distrib/book.html.

The hostname *www* is location independent so we have location transparency in that the address of a particular computer is not included. Therefore the organisation may move the Web service to another computer.

But if the responsibility for providing a WWW-based information service moves to another organisation, the URL would need to be changed.

1.7   A server program written in one language (for example C++) provides the implementation of a BLOB object that is intended to be accessed by clients that may be written in a different language (for example Java). The client and server computers may have different hardware, but all of them are attached to an internet. Describe the problems due to each of the five aspects of heterogeneity that need to be solved to make it possible for a client object to invoke a method on the server object.

1.7 Ans.

As the computers are attached to an internet, we can assume that Internet protocols deal with differences in networks.

But the computers may have different hardware - therefore we have to deal with differences of representation of data items in request and reply messages from clients to objects. A common standard will be defined for each type of data item that must be transmitted between the object and its clients.

The computers may run different operating systems, therefore we need to deal with different operations to send and receive messages or to express invocations. Thus at the Java/C++ level a common operation would be used which will be translated to the particular operation according to the operating system it runs on.

We have two different programming languages C++ and Java, they use different representations for data structures such as strings, arrays, records. A common standard will be defined for each type of data structure that must be transmitted between the object and its clients and a way of translating between that data structure and each of the languages.

We may have different implementors, e.g. one for C++ and the other for Java. They will need to agree on the common standards mentioned above and to document them.

1.8 An open distributed system allows new resource sharing services such as the BLOB object in Exercise 1.7 to be added and accessed by a variety of client programs. Discuss in the context of this example, to what extent the needs of openness differ from those of heterogeneity.

1.8 Ans.

To add the BLOB object to an existing open distributed system, the standards mentioned in the answer to Exercise 1.7 must already have been agreed for the distributed system To list them again:

- the distributed system uses a common set of communication protocols (probably Internet protocols).
- it uses an defined standard for representing data items (to deal with heterogeneity of hardware).
- It uses a common standard for message passing operations (or for invocations).
- It uses a language independent standard for representing data structures.

But for the open distributed system the standards must have been agreed and documented before the BLOB object was implemented. The implementors must conform to those standards. In addition, the interface to the BLOB object must be published so that when it is added to the system, both existing and new clients will be able to access it. The publication of the standards allows parts of the system to be implemented by different vendors and to work together.

1.9 Suppose that the operations of the BLOB object are separated into two categories – public operations that are available to all users and protected operations that are available only to certain named users. State all of the problems involved in ensuring that only the named users can use a protected operation. Supposing that access to a protected operation provides information that should not be revealed to all users, what further problems arise?

1.9 Ans.

Each request to access a protected operation must include the identity of the user making the request. The problems are:

- defining the identities of the users. Using these identities in the list of users who are allowed to access the protected operations at the implementation of the BLOB object. And in the request messages.
- ensuring that the identity supplied comes from the user it purports to be and not some other user pretending to be that user.
- preventing other users from replaying or tampering with the request messages of legitimate users.

Further problems.

- the information returned as the result of a protected operation must be hidden from unauthorised users. This means that the messages containing the information must be encrypted in case they are intercepted by unauthorised users.

1.10 The INFO service manages a potentially very large set of resources, each of which can be accessed by users throughout the Internet by means of a key (a string name). Discuss an approach to the design of the names of the resources that achieves the minimum loss of performance as the number of resources in the service increases. Suggest how the INFO service can be implemented so as to avoid performance bottlenecks when the number of users becomes very large.

1.10 Ans.

Algorithms that use hierarchic structures scale better than those that use linear structures. Therefore the solution should suggest a hierarchic naming scheme. e.g. that each resource has an name of the form 'A.B.C' etc. where the time taken is $O(\log n)$ where there are n resources in the system.

To allow for large numbers of users, the resources are partitioned amongst several servers, e.g. names starting with A at server 1, with B at server 2 and so forth. There could be more than one level of partitioning as in DNS. To avoid performance bottlenecks the algorithm for looking up a name must be decentralised. That is, the same server must not be involved in looking up every name. (A centralised solution would use a single root server that holds a location database that maps parts of the information onto particular servers). Some replication is required to avoid such centralisation. For example: i) the location database might be replicated

at multiple root servers or ii) the location database might be replicated in every server. In both cases, different clients must access different servers (e.g. local ones or randomly).

1.11   List the three main software components that may fail when a client process invokes a method in a server object, giving an example of a failure in each case. To what extent are these failures independent of one another? Suggest how the components can be made to tolerate one another's failures.

1.11 Ans.

The three main software components that may fail are:

- the client process e.g. it may crash
- the server process e.g. the process may crash
- the communication software e.g. a message may fail to arrive

The failures are generally caused independently of one another. Examples of dependent failures:

- if the loss of a message causes the client or server process to crash. (The crashing of a server would cause a client to perceive that a reply message is missing and might indirectly cause it to fail).
- if clients crashing cause servers problems.
- if the crash of a process causes a failures in the communication software.

Both processes should be able to tolerate missing messages. The client must tolerate a missing reply message after it has sent an invocation request message. Instead of making the user wait forever for the reply, a client process could use a timeout and then tell the user it has not been able to contact the server.
   A simple server just waits for request messages, executes invocations and sends replies. It should be absolutely immune to lost messages. But if a server stores information about its clients it might eventually fail if clients crash without informing the server (so that it can remove redundant information). (See stateless servers in chapter 4/5/8).
   The communication software should be designed to tolerate crashes in the communicating processes. For example, the failure of one process should not cause problems in the communication between the surviving processes.

1.12   A server process maintains a shared information object such as the BLOB object of Exercise 1.7. Give arguments for and against allowing the client requests to be executed concurrently by the server. In the case that they are executed concurrently, give an example of possible 'interference' that can occur between the operations of different clients. Suggest how such interference may be prevented.

1.12 Ans.

For concurrent executions - more throughput in the server (particularly if the server has to access a disk or another service)

Against - problems of interference between concurrent operations

Example:

Client A's thread reads value of variable X

Client B's thread reads value of variable X

Client A's thread adds 1 to its value and stores the result in X

Client B's thread subtracts 1 from its value and stores the result in X

Result: X := X-1; imagine that X is the balance of a bank account, and clients A and B are implementing credit and debit transactions, and you can see immediately that the result is incorrect.
   To overcome interference use some form of concurrency control. For example, for a Java server use synchronized operations such as credit and debit.

1.13   A service is implemented by several servers. Explain why resources might be transferred between
        them. Would it be satisfactory for clients to multicast all requests to the group of servers as a way
        of achieving mobility transparency for clients?

1.13 Ans.

Migration of resources (information objects) is performed: to reduce communication delays (place objects in a server that is on the same local network as their most frequent users); to balance the load of processing and or storage utilisation between different servers.

If all servers receive all requests, the communication load on the network is much increased and servers must do unnecessary work filtering out requests for objects that they do not hold.