Albert Ludwigs University Freiburg
Department of Computer Science
Prof. Dr. Stefan Leue and Corina Apachiţe
Distributed Systems - WS 2001/2002

Assignment 1 - Solutions

Question 1.1 Give five types of hardware resources and five types of data or software resources that can usefully be shared. Give examples of their sharing as it occurs in distributed systems. (1 point)

Answer 1.1 Examples of hardware resources that can be usefully be shared, and examples of their sharing (0.5p):

- *network capacity* transmission of data (packet transmission) is done using the same circuit, this means, many communication channels share the same underlying circuit
- *screen* network window systems (X-Windows) allow processes in remote computers to update the content of the local windows.
- *disk* file server, virtual disk server
- *memory* cache server
- *CPU* servers do some computation for their clients, hence their cpu is a shared resource

Examples of data/software resources that can be shared in a distributed system (0.5 p):

- *object* there are unlimited possibilities to share objects in distributed systems (for example shared whiteboard, ticket booking, etc)
- *file* file servers enable multiple clients to have read/write access to the same files
- *web-page* web servers enable client programs to have read-only access to the page content
- *database* the content of a database can be usefully shared. There are many techniques that control the concurrent access to a database.
- *exclusive lock* a system-level object provided by a lock server, enabling the coordination of accessing a special resource

Question 1.2 Give an example of a URL.

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

To what extent is an URL location transparent?(1 point)

Answer 1.2 An example of an URL (0.1p):

`http://tele.informatik.uni-freiburg.de/Teaching/ws01/dsys/dsys-tutorial.html`

The main components of an URL (according to RFC 1738) are (0.7p):

1. The protocol to use, te part before the colon (in our case, Hyper Text Transfer Protocol, http). Another examples of protocols that may be used: mailto, ftp, telnet, wais,. . ..
2. The part between // and / is the Domain name of the web server host (in our example, tele.informatik.uni-freiburg.de)
3. The rest refers to information on that host - named within the top level directory used by that Web server (this means, our resourse is located on the web directory of the host tele.informatik.uni-freiburg.de according to the path: Teaching/ws01/dsys/dsys-tutorial.html)

Location transparency (0.2p): we have location transparency in that the address of a particular computer is not included. Inside our research group we can change the web service on another computer.

Question 1.3 Consider a simple server that carries out clients requests without accessing other servers. Explain why it is generally not possible to set a limit on the time taken by such a server to respond to a client request.

What would need be done to make the server able to execute requests within a bounded time?

Is this a practical option?(2 points)

Answer 1.3 We cannot send a limit for the time taken by a server to respond a client request because the arrival of client requests is not predictible. For example we can have limits for executing a computation corresponding to a request, but we can not predict how much a request will have to wait until its execution time arrives. If the server uses threads, there is not possible for the server to allocate sufficient time to a particular request within any given time limit. If the requests are carryed out one at a time, they may wait in the queue for an unlimited amount of time. (0.6p)

We can imagine the following solution, trying to set a time limit in responding to client requests (0.7p):

- we can limit the number of clients to suit the server capacity; to deal with many clients, we should use a server with a higher capacity (many processors...)
- we can replicate the service.

The practicability of our solutions (0.7p):

- setting a limit for the number of clients that are able to suit the service we offer induces the lost of some client requests.
- the introduction of replicas brings the problem of keeping them consistent. This requires also an amount of time that could be used for responding client requests..

Question 1.4 Consider two communication services for use in asynchronous distributed systems. In service A, messages may be lost, duplicated or delayed and checksums apply only to headers. In service B, messages may be lost, delayed or delivered too fast for the recipient to handle them, but those that are delivered arrive ordered and with the correct contents.

Describe the classes of failure exhibited by each service.

Classify their failures according to their effect on properties of validity and integrity.

Can service B be described as a reliable communication service? (2 points)

Answer 1.4 Sevice A can have (0.8p):

- *arbitrary failures:* checksums apply only to headers $\Rightarrow$ message bodies may be corrupted, duplicated messages
- *omission failures*: lost messages

The lost of messages implies that we have no validity property using the service A. The corrupted and duplicated messages deny the integrity property of service A.

Service B can have (0.8p):

- *omission failures:* lost messages, dropped messages.

The integrity property holds but the validity property is denied by the lost of messages. So, the service B can NOT be called reliable. (0.4p)

Question 1.5 Define the integrity property of reliable communication and list all the possible threats to integrity from users and from systems components. What measures can be taken to ensure the integrity property in face of each of these source of threats. (2 points)

Answer 1.5 *The integrity property* is that the message received is identical to the one sent and no messages are delivered twice. (0.4p)

Threats from users (0.6p):

1. injecting bad messages
2. replaying old messages
3. modifying the messages during the transmission (this means, a bad user is altering the messages that travel between two user, A and B)

Threats from the systems components (0.6p):

1. the messages may get corrupted traveling on the underlying network
2. the communication protocols may duplicate the messages

Measures against user threats (0.2p): secure channels (authentication techniques, nonces).

Measures against system components threats (0.2p): checksum to detect corrupted messages, sequence numbers. Remains the problem of dropped messages.

Question 1.6 Compare connectionless (UDP) and connection-orriented (TCP) communication for the implementation of each of the following application-level or presentation-level protocols:

1. virtual terminal access (for example, telnet);
2. file transfer (for example, FTP);
3. user location (for example rwho, finger);
4. information browsing (for example, HTTP);
5. remote procedure call.

(2 points)

Answer 1.6

1. The characteristics of the virtual terminal access: long duration of session, the need of reliable communication, the unstructured character of data transmitted. These could be a reason to choose the TCP communication. (0.4p)
2. File transfer requires transmission of large amount of data. We would use UDP if the error rates are small and the messages can be large. Practical this is not the case, so the TCP is preferred. (0.4p)

3. For user location information, a single message is enough, so the connectionless communication is used. (0.4p)

4. Both connection-oriented and connectionless comunication can be used. Some requests imply the transfer of a big amount of data, so the TCP communication is more indicated (and used). (0.4p)

5. Remote Procedure Calls implies the sending of a small number of packets (invocation, response of the invocation) but reliability is needed (the reliability is not a characteristic of UDP communication). We can use UDP communication, but we have to assure the reliability of communication at a higher level (timeouts and re-trys). (0.4p)