***Questions***

Three computers together provide a replicated service. The manufacturers claim that each computer has a mean time between failures of five days; a failure typically takes four hours to fix. What is the availability of the replicated service?

***Solution:***

The probability that an individual computer is down is 4/(5\*24 + 4) ~ 0.03. Assuming failure-independence of the machines, the availability is therefore 1 – 0.033 = 0.999973.

***Questions***

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.

***Solution:***

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.

***Questions***

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?

***Solution:***

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 coarsegrained 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 typechecking (web services operate by-value type checking without compiler support), (b) there is the inefficiency that we have mentioned.

***Questions***

Give some examples of faults in hardware and software that can/cannot be tolerated by the use of redundancy in a distributed system. To what extent does the use of redundancy in the appropriate cases make a system fault-tolerant?

***Solution:***

**Hardware faults** - processors, disks, network connections can use redundancy e.g. run process on multiple computers, write to two disks, have two separate routes in the network available.

Software bugs, crashes. Redundancy is no good with bugs because they will be replicated. Replicated processes help with crashes which may be due to bugs in unrelated parts of the system. Retransmitted messages help with lost messages.

Redundancy makes faults less likely to occur. e.g. if the probability of failure in a single component is *p* then the probability of a single independent failure in *k* replicas is *p*k.

***Questions***

Describe possible occurrences of each of the main types of security threat (threats to processes, threats to communication channels, denial of service) that might occur in the Internet.

***Solution:***

**Threats to processes**: without authentication of principals and servers, many threats exist. An enemy could access other user’s files or mailboxes, or set up ‘spoof’ servers. E.g. a server could be set up to ‘spoof’ a bank’s service and receive details of user’s financial transactions.

**Threats to communication channels**: IP spoofing - sending requests to servers with a false source address, man in- the-middle attacks.

**Denial of service**: flooding a publicly-available service with irrelevant messages.

***Questions***

In a gossip system, a front end has vector timestamp (3, 5, 7) representing the data it has received from members of a group of three replica managers. The three replica managers have vector timestamps (5, 2, 8), (4, 5, 6) and (4, 5, 8), respectively. Which replica manager(s) could immediately satisfy a query from the front end and what is the resultant time stamp of the front end? Which could incorporate an update from the front end immediately?

***Solution:***

The only replica manager that can satisfy a query from this front end is the third, with (value) timestamp (4,5,8). The others have not yet processed at least one update seen by the front end. The resultant time stamp of the front end will be (4,5,8).

Similarly, only the third replica manager could incorporate an update from the front-end immediately.

***Questions***

A client makes RMIs to a server. The client takes 5 ms to compute the arguments for each request, and the server takes 10ms to process each request. The local OS processing time for each *send* or *receive* operation is 0.5 ms, and the network time to transmit each request or reply message is 3 ms. Marshalling or Unmarshalling takes 0.5 ms per message.

Estimate the time taken by the client to generate and return from 2 requests

1. if it is single-threaded, and
2. (ii) if it has two threads which can make requests concurrently on a single processor. Is there a need for asynchronous RMI if processes are multi-threaded?

***Solution:***

(i) Single-threaded time: 2(5 (prepare) + 4(0.5 (marsh/unmarsh) + 0.5 (local OS)) + 2\*3 (net)) + 10 (serv)) = 50 ms.

(ii) Two-threaded time: (see figure 6.14) because of the overlap, the total is that of the time for the first operation’s request message to reach the server, for the server to perform all processing of both request and reply messages without interruption, and for the second operation’s reply message to reach the client.

This is: 5 + (0.5+0.5+3) + (0.5+0.5+10+0.5+0.5) + (0.5+0.5+10+0.5+0.5) + (3 + 0.5+0.5)

= 37ms.

**Q)** Illustrate the distributed query processing methodology by using diagram?

**Q)** In the context in Distributed Database Management System, what is the difference between **Vertical Fragmentation** and **Horizontal Fragmentation**? Explain by example.

**Q)** What do you understand by transaction transparency and concurrency transparency in distributed database system?

**Q)** How can you differentiate between homogeneous and heterogeneous in distributed database system?

**Q)** Estimate the time required to crack a 56-bit DES key by a brute-force attack using a 500 MIPS (million instruction per second) workstation, assuming that the inner loop for a brute-force attack program involves around 10 instructions per key value, plus the time to encrypt an 8-byte plaintext (see Figure 7.14). Perform the same calculation for a 128-bit IDEA key. Extrapolate your calculations to obtain the time for a 50,000 MIPS parallel processor (or an Internet consortium with similar processing power).

**Q)** Some of the ways in which conventional email is vulnerable to eavesdropping, masquerading, tampering, replay, denial of service. Suggest methods by which email could be protected against each of these forms of attack.