98-Comp-B10, Distributed Systems

3 hours duration

NOTES:

1. If doubt exists as to the interpretation of any question, the candidate is urged to submit with the answer paper a clear statement of any assumptions made.

2. This is a CLOSED BOOK examination.
   Only non-programmable calculator is permitted

3. Answer any **five of the seven** questions.
   Only the first five questions as they appear in the answer book will be marked.

4. All questions carry equal weight.

5. Most questions require an answer in essay format. Clarity and organization of the answer are important.

Question # 1. *Characteristics of distributed systems*

a. Explain what is meant by a client program and a server program. Explain and illustrate (in graphical form) the client-server architecture of one major Internet applications (for example the Web, email or ftp).

a. Name two types of software resource and two types of hardware resource that can be shared efficiently in distributed systems. Provide examples of their sharing.

Question # 2. *Fundamental concepts and mechanisms*

a. A client sends a 400 byte request message to a service, which produces a response containing 6000 bytes. Estimate the total time to complete the request in each of the following cases, with the performance assumptions listed below:
   i) Using connectionless (datagram) communication (for example, UDP);
   ii) Using connection-oriented communication (for example, TCP);
   iii) The server process is in the same machine as the client.

   [Latency per packet (local or remote, incurred on both send and receive):5 milliseconds
   Connection setup time (TCP only):5 milliseconds
   Data transfer rate: 10 megabits per second
   MTU: 2000 bytes
   Server request processing time: 2 milliseconds
   Assume that the network is lightly loaded.]

b. Comment on the use of Connectionless (UDP) and connection-oriented (TCP) communication for of each of the following application:
   1. Mail access protocols (for example, IMAP);
   2. Internet radio;
   3. Information browsing (for example, HTTP);
   4. Remote procedure call.

Question # 3. *Client-server systems & inter-process communications*

a. Suppose you were developing a distributed multimedia conferencing application. Which of the following mechanisms would you choose to implement communication between processes in the applications, and why?
   i) Stream sockets
   ii) Datagram sockets
   iii) RPC over streams sockets
   iv) RPC over datagram sockets

a. Give explanation to the design choices that are relevant to decreasing the amount of reply data held at a server. Suppose the RRA protocol is in use. What is the length of time needed for servers to retain unacknowledged reply data? Do the servers have to continually send the reply in an attempt to obtain an acknowledgement?

Question # 4. *Operating systems for distributed architectures*

a. A file server uses caching. The average hit rate achieved is 75%. If the requested block is in the cache, file operations in the server cost 5 ms of CPU time; otherwise, it takes an additional 10 ms of disk I/O time otherwise. Estimate the server's throughput capacity (average requests/sec) if it is: (Explain any assumptions you make)
   1. single-threaded;
   2. two-threaded, running on a single processor;
   3. two-threaded, running on a two-processor computer

b. Compare the thread-per-request architecture with the worker pool multi-threading architecture.

c. What the kernel must provide for a user-level implementation of threads (such as Java on UNIX)? In user-level threads implementations, do page faults present a problem?

Question # 5. *Security*

a. Primary exchanges of public keys are vulnerable to the man-in-the-middle attack. Illustrate two defences against it.

a. Pretty Good Privacy (PGP) is broadly used to secure email communication. Explain the steps needed prior to exchanging email messages with privacy and authenticity guarantees for a pair of users using PGP.

Question # 6. *Distributed file systems*

a. Name and discuss three key design issues for distributed file systems.

b. How does the NFS Automounter help to improve the performance and scalability of NFS?

c. Compare AFS and NFS from stability point view? Are there any limits on AFS scalability, assuming that servers can be added as required? Are there any recent technological developments that would help to offer greater scalability?

Question # 7. *Principles of fault tolerance continual*

a. What is the gossip architecture? Why does a replica manager need to keep both a 'replica' timestamp and a 'value' timestamp?

b. Write pseudocode for a scheme for integrating two replicas of a file system directory that underwent separate updates during disconnected operation.