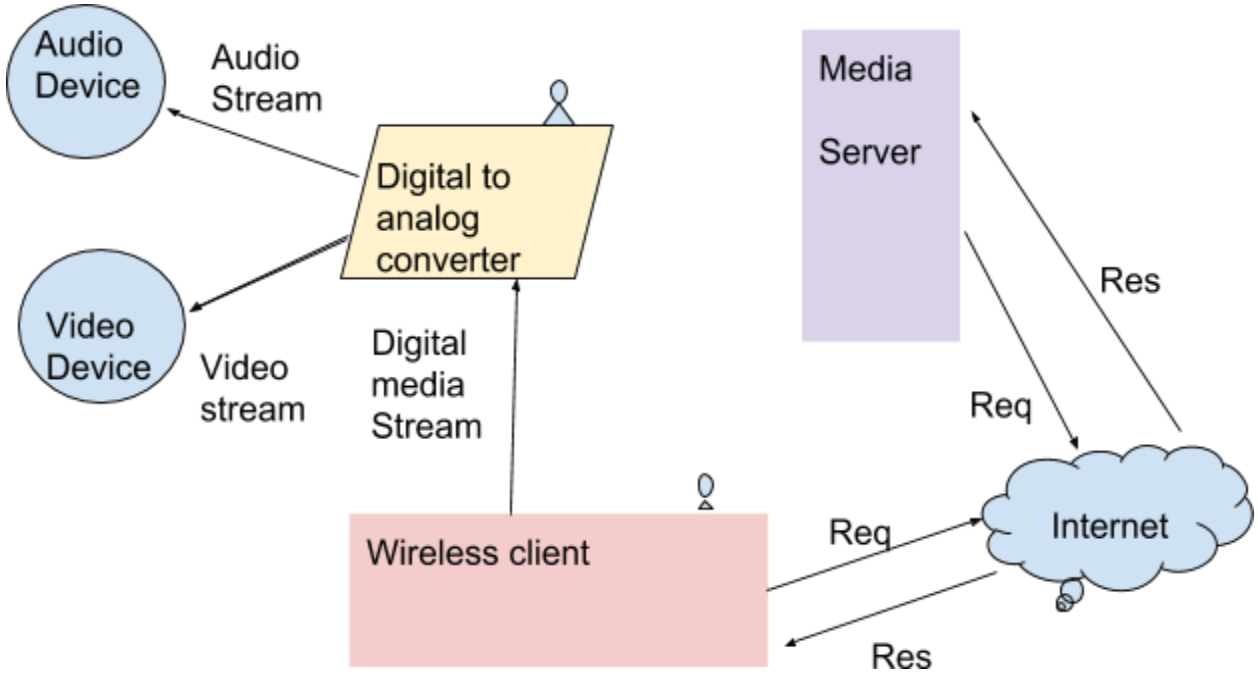
CS550 Written Assignment 1
Saptarshi Chatterjee(A20413922)

Chapter 1

1. 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.



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

If the number of components that comprise the system or geographical size of the system , or number and size of administrative domains of the system , can grow in dimensions without an unacceptable loss of performance, then it's called a scalable system.

3. Q: What is the difference between a multiprocessor and a multicomputer?

Multi Processor	Multicomputer
A multi processor is a single system with multiple CPUs / GPUs.	A multi-computer is a collection of multiple physical or virtual computers.
Tihane-2, Piz Daint, Titan are some high end Multi	Cloud Computing , Cluster Computing , Grid

processor system (Supercomputers)	Computing are some popular Multicomputer system.
Building a High end Multi processor system is costly	It's comparatively cheaper to achieve that computing power using Multi computer (Cloud).
In a multi processor system , a physical address space is shared by all the computing units / CPUs . (Modern GPUs come with dedicated RAM though)	Each machine in a multicomputer system has its dedicated physical memory , which is not shared.
Most powerful Multi processing system have computing power in the range 33.86 PFLOPS	Theoretically , We can exceed exaflop computing power using Multicomputer .

Chapter 2

4. 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?

There is no one solution that fit-all the the cases , for this . A few approach to tackle such problems are

1) Divide and conquer - Client side code can be divided into multiple parts , and when one part waits for the response from server , client can schedule another part
2) Async communication - Instead of waiting for server response , client can perform other tasks and when response is available , process it
3) Caching - We can keep a cache server nearer to client which will store most frequently accessed static data , so that client need not go to server every time it needs them. CDN platforms use this technique.

5. Q: Consider a chain of processes P1,P2,…,Pn implementing a multitiered client-server architecture. Process Pi is client of process P(i+1), and Pi will return a replay to P(i-1) only after receiving a reply from Pi+1. What are the main problems with this organization when taking a look at the request-reply performance at process P1?

A few major issues of such an architecture is -

1) For large N this system will behave poorly .
2) If $P_{2,}$ P_{3} ..... P_{n} are on different machine then for N machine this system has an overhead of n - 2 request reply calls over network which makes the system slow.
3) If any of the layer in this architecture goes down , then it will take the entire system down , making the system too vulnerable to fault.
4)

6. Q: Consider a BitTorrent system in which each node has an outgoing link with a bandwidth capacity Bout and an incoming link with bandwidth capacity Bin. Some of these nodes (called

seeds) voluntarily offer files to be downloaded by others. What is the maximum download capacity of a BitTorrent client if we assume that it can contact at most one seed at a time?

In BitTorrent outgoing bandwidth of seeds is shared between clients. For a system with P seeds and C clients, the combined outgoing bandwidth of the seeders is $P \times B_{out}$, giving each of the clients $P \times B_{out} / C$ download capacity.

If $B_{in} > B_{out}$ and the client decide to share inprogress downloaded content to other clients
Then as BitTorrent clients are dictated by its outgoing capacity the total download capacity will be $\mathbf{P \times B_{out} / C + B_{out}}$.

Chapter 4

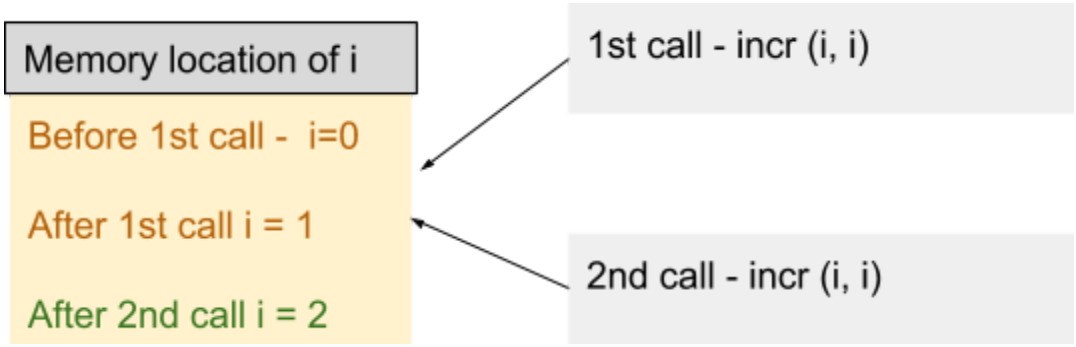
7. Q: In many layered protocols, each layer has its own header. Surely it would be more efficient to have a single header at the front of each message with all the control in it than all these separate headers. Why is this not done?

In many layered protocols and data is passed from one layer to another and each layer has it's own header , making layers independent of each other . If we have a single big Header we will face several problem

1) Each layer will be able to read and write this big header , and it will destroy separation of concern pattern and isolation of layers.
2) We can't modify , Swap in / out a layer without modifying other layers.
3) One layer can accidentally modify headers of other layer , corrupting the communication.
4) It will be hard to enforce security.

8. Q: Consider a procedure incr with two integer parameters. The procedure adds one to each parameter. Now suppose that it is called with the same variable twice, for example, as incr(i, i). If i is initially 0, what value will it have afterward if call-by-reference is used? How about if copy/restore is used?

For call by reference the final value would be 2 . As both call update the same memory location one after another .



For copy restore final value would be 1 . As both call gets initial input as 0 and they update i to one and when they restore the value one's value get overwritten by another .

9. Q: One way to handle parameter conversion in RPC systems is to have each machine send parameters in its native representation, with the other one doing the translation, if need be. The native system could be indicated by a code in the first byte. However, since locating the first byte in the first word is precisely the problem, can this actually work?

Locating first bythe in the first word , shouldn't be a problem here . When a machine sends byte 0 , that byte always arrive at byte 0 . So the destination computer can simply access byte 0 using Byte level Instructions.

Another solution to it is putting the code that identify the native instruction , in every byte of the first word.

10. Q: What trade-off should be made when we decide between a shared memory model and a message passing model?

When two or more processes communicate between each other by sharing memory space then that communication is referred to as Shared Memory Communication.

Distributed shared memory (DSM) is a form of memory architecture where physically separated memories can be addressed as one logically shared address space.

Advantages

- Scales well with a large number of nodes
- Can handle complex and large databases without replication or sending the data to processes
- Generally cheaper than using a multiprocessor system
- Provides large virtual memory space

Disadvantages

- Generally slower to access than non-distributed shared memory
- Must provide additional protection against simultaneous accesses to shared data

Where as message passing system suits the communication between 2 processes which may not reside in same physical system . Client stub converts the procedure call into a message and transfers that message over network , and server stub un-marshalls the message and then calls the local procedure . Passing large values as argument is an issue with this approach , also pass by reference is a problem as one machine doesn't know the physical memory configuration of other machine.

Message passing	Shared memory
Variables have to be marshalled	Variables are shared directly
Processes are protected by having private address space	Processes could cause error by altering data
Processes should execute at the same time	Executing the processes may happen with non-overlapping lifetimes

Source - https://en.wikipedia.org/wiki/Distributed_shared_memory

10.b Q: Why does this make shared memory a bad match for a system distributed across the Internet?

Main memory access ought to be fast . Other wise CPU will waste its computing power waiting for data retrieval from RAM . This problem is called memory wall problem.

Network usually the slowest part is computer systems. So if we have shared memory for distributed system , we need to access RAM over network which makes it very very slow and engenders memory wall problem.