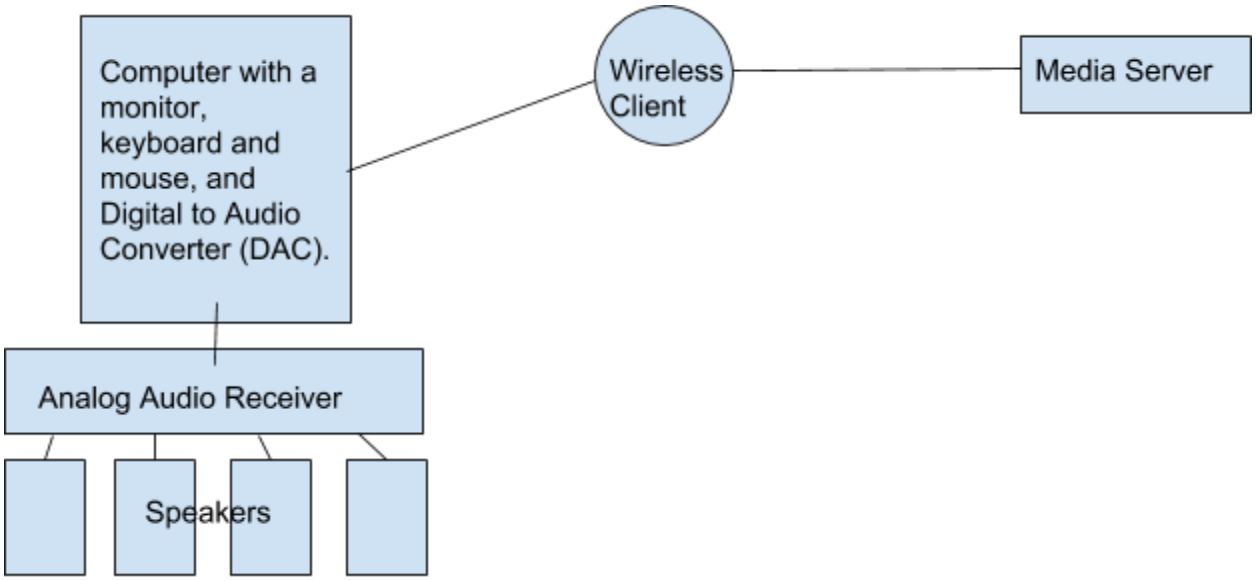
Jason Lawrence
A20381993
1



2. For a system to be scalable it needs to have the ability to increase the user base and increase the amount of resources available to the system. Also a scalable system has to be geographically scalable meaning the users and resources may lie far apart. Lastly, A scalable system has to be easy to manage even if there are many independent administrative organizations.

3. Multiprocessors have fast hardware networking, shared address space between the CPUs, and they run slower than multicomputers because they are just one system.
Multicomputers is made up of several computers with slow hardware networking, they have one physical address space per CPU, and they are faster than multiprocessor systems.

4. To keep network latency from dominating overall performance we can smaller data chunks during request and replies. This allows for the client to work with a the small data that it receives while it is waiting on the rest. Also clients can have multiple sessions open with the server and the data can be split up amongst the sessions and in the end be added up. Lastly, instead of just waiting on a reply the client can work on other request/replies.

5. If there are a lot of processes, n equals a really big number, then p1 will take a really long time because it has to wait for all of the other processes to run before it will get its reply. Also if a process fails or has an error then the process may never get a response back.

6. Assuming that we have an S amount of seeders and N amount of nodes, then initially the download capacity of each node comes to S * bout / N. Since Nodes can help each other out

and the bittorrent system's file sharing nature the download rate relies on the outgoing capacity. Finally, the download capacity max is S*bout/N+bout

7. The layers ideally should be independent of each other. When you have just one header for all of the layers the layers could write to it and all of the other layers could see that write. This is a pretty big security issue since if one layer gets compromised all of the layers are compromised. Where as if each layer has its own header then any writes to it are hidden from the other layers.

8. Calling incr(i, i) using call-by-reference then a pointer to i is passed to the function twice. Since pointer isn't the actual variable just the memory location of the variable it will increment what's stored at that location twice, so 0 will become 2. When copy/restore is used i will be passed to it twice both being 0 and independent of each other. Each i will be incremented making them both 1. Finally they will both be copied back and the second i will overwrite the first i and the final value will be 1.

9. When a computer sends data to another byte 0 sent will be byte 0 received so locating the first byte of the first word is fairly easy. It doesn't matter if the machines are different.

10.
Shared Memory System:
Communication between processors is implicit and transparent. They prefer conventional architectures more since existing processors can be added easily. Protocols for communication are hidden within the system and are closed to the hardware. Communication is controlled by the shared bus system.
Message Passing System:
Processors must explicitly communicate with each other through messages. There are fewer assumptions on the model which leads to a simpler multiprocessing architecture. Although the code is platform specific. Communication protocols are under the users control and are treated as I/O in the system.

Shared memory is a bad match because the computers would to constantly access memory stored somewhere else which would create more traffic on the network increasing network latency.