Question 01: Define distributed system. Mention a few real time examples of distributed systems.

Answer:

Distributed System: A distributed system is a collection of entities, each of which is autonomous, programmable, asynchronous and failure-prone, and which communicate through an unreliable communication medium. (Working Definition).

- Entity=a process on a device (PC, PDA)
- Communication Medium=Wired or wireless network

Real time example:

- The Web
- The Internet
- DNS
- Blockchain Technology
- BitTorrent
- Online Banking Systems

Question 02: What design principles considered useful for better performance in a distributed system.

Answer:

Design principles / Common Goals:

- Heterogeneity – can the system handle a large variety of types of PCs and devices?
- Robustness – is the system resilient to host crashes and failures, and to the network dropping messages?
- Availability – are data+services always there for clients?
- Transparency – can the system hide its internal workings from the users?
- Concurrency – can the server handle multiple clients simultaneously?
- Efficiency – is the service fast enough? Does it utilize 100% of all resources?
- Scalability – can it handle 100 million nodes without degrading service? (nodes=clients and/or servers) How about 6 B? More?
- Security – can the system withstand hacker attacks?
- Openness – is the system extensible?

Question 03: Mention the advantages of a distributed computing environment over standalone applications. What are the disadvantages of distributed systems?

Answer:

Advantages:

1) **Performance:** Performance is improved by having each computer in a cluster handle different parts of a task simultaneously.

2) **Scalability:** Whenever there is an increase in workload, users can add more workstations. There is no need to upgrade a single system.

3) **Redundancy and availability:** Multiple instance of data are replicated over the system to prevent downtime or data loss even when one component fail.

4) **Reduced Latency:** If a particular node is located closer to the user, the distributed system makes sure that the system receives traffic from that node.

5) **Fault Tolerance:** Continue operating even when one or more nodes fail.

6) **Reduced Cost:** Because distributed systems can make use of existing resources rather than needing to buy new gear, they can be less expensive than centralized systems.

Disadvantages:

1) **Security:** The data of the user is stored in different workstations. Thus, the user needs to make sure that their data is secured in each of these computers.

2) **Overheads:** When all the workstations try to operate at once, there will be an increase in computing time. This ultimately impacts the system’s response time.

3) **Startup Cost:** Compared to a single system, the implementation cost of a distributed system is significantly higher.

4) **Testing and Debugging**: Because of the complexity of the system or the interactions between many nodes, testing and debugging distributed systems can be difficult.

5) **Network Dependency:** Distributed systems are prone to network errors which result in communication breakdown.

Question 04: Define scalability. What are the challenges involved in designing scalable distributed system?

Answer:

Scalability refers to the ability of a system to handle increasing workload by expanding its resources.

Challenges involved in designing scalable distributed system:

1. **Consistency and Coordination**: As more machines are added to a distributed system, it becomes increasingly difficult to maintain consistency across all machines. Additionally, coordinating the actions of multiple machines can be challenging, especially in situations where data needs to be updated.

2. **Partitioning:** Partitioning is the process of dividing data or workloads into smaller units to be processed in parallel. However, partitioning can be challenging in distributed systems because it requires careful planning to ensure that the data is divided in a way that minimizes communication between machines.

3. **Network Latency and Bandwidth:** As more machines are added to a distributed system, the amount of network traffic between machines increases, which can lead to increased network latency and reduced bandwidth.

4. **Fault Tolerance**: Distributed systems need to be designed to be fault-tolerant, meaning that the system can continue to operate in the event of machine failures or network outages. This can be challenging to achieve because it requires careful planning and redundancy across multiple machines.

5. **Load Balancing:** As more machines are added to a distributed system, load balancing becomes increasingly important to ensure that workloads are evenly distributed across all machines.