Q.1 - Explain a scenario where you can use distributed systems.

Ans - Suppose you have created an online education platform named starGrad. In no time, your website becomes popular because of its uniqueness and relevant content. Now, the traffic on your website begins to increase and a number of learners enrol themselves into the courses from different locations. Now, to maintain the performance of your website, you can use a distributed system, which will help in dividing your workload.

Q.2 - What are the characteristics of distributed systems?

Ans- Some of the key characteristics of distributed systems are as follows:

- Resource sharing - This is a feature of distributed systems which can be used to share hardware such as printers and software like files, data, etc., and this helps in reducing costs and also offers convenience.

- Openness - This means the new resource-sharing services are available to everyone.

- Concurrency - The system should be able to perform parallel processing/computation. Here, concurrency in computation allows distributed systems to perform complex tasks in a fast and efficient manner.

- Scalability - The system should be able to easily scale up with an increase in workload. By adding cheap commodity hardware, distributed systems can easily implement horizontal scaling when the workload increases.

- Fault tolerance - This system is built to provide uptime at all times. Fault tolerance can be achieved by incorporating redundancy carefully.

- Transparency - The distributed systems offer transparency in access, location, performance and scalability.

Q.3 - What are the challenges faced while implementing distributed systems?

Ans - Following are the challenges faced:

- Performance - Distributed systems use multiple machines that are connected over a network, which diminishes the performance of the system. You can enhance the system's performance through parallel computation and reduce the network delay by increasing bandwidth and redundancy in the network.

- Scalability - It is the property used to maintain the efficiency of the system constantly by adding the computation machines. A good, scalable system should be capable of smoothly increasing the capacity with the increasing load.

- Fault Tolerance - Fault tolerance is built to provide continuous uptime, and it can be achieved by carefully incorporating redundancy.

- Reliability - It indicates a system that can run continuously without failure. But, the distributed systems are susceptible to faults. Reliability can be ensured by high fault tolerance and the ease of use and maintenance.

- Security - Providing security in the distributed systems is a big challenge as compared to that in the centralised systems, as all the data in the centralised systems is present at a single place. But, data is dispersed in the distributed systems. Security can be achieved by ensuring proper authentication, stateless systems and proper monitoring.

Q.4 - According to you, why would someone prefer distributed systems?

Ans - The use of distributed systems offers the following benefits:

- Distributed systems can be easily scaled: Vertical scaling or Horizontal scaling.
- Resource sharing is easy in distributed systems, and we can share hardware as well as software.
- Distributed systems provide a feature of fault tolerance, which means, if one node fails, the other nodes remain unaffected.
- Distributed systems provide resource sharing, helping in cost reduction.
- In distributed systems, we can perform parallel computation, which results in good performance.

Q.5 - What are the differences between centralised and distributed systems?

Ans -

Centralised Systems	Distributed Systems
Limited storage capacity and computational capability	Theoretically, no upper limit to storage and computational capabilities
Only vertically scalable, which is very expensive after a limit and involves downtime	Both vertically and horizontally scalable; hence, easily scalable
Single point of failure	No single point of failure

Not fault-tolerant	Fault-tolerant

Q.6 - How do you provide security in your website based on distributed systems?

Ans - You can achieve security in the following three ways:

- **Proper authentication** (Transient token-based): This should be done between systems during communication.
- **Stateless systems**: The nodes should not store any unnecessary data related to past communications or computations.
- **Proper vigilance or monitoring**: All the nodes must be continuously monitored, and any signs of vulnerability or failure should be fixed right away before it gets exploited.

Q.7 - Suppose your website has heavy traffic. Which characteristics of the distributed system can you use to handle this?

Ans - Here, you can use scalability. A good, scalable system should be capable of smoothly increasing the capacity with the increasing load. Adding a new node to the cluster requires rearrangement/rebalancing of the cluster.

You can scale a system in two ways:

- Vertical scaling:
  - Vertical scaling means adding more power to the machine by increasing the processor or storage capacity.
  - It is also known as **scaling up**.
- Horizontal scaling:
  - Horizontal scaling means scaling by adding more resources to your machine, which shares the processing and workload across multiple devices.
  - It is also known as **scaling out**.

Q.8 - Differentiate between the types of architecture of distributed systems along with examples.

Ans - The distributed systems mainly have two types of architecture:

- Master-Slave architecture:
    - In this architecture, a Master node controls and coordinates all the other nodes (Slaves) in the network.
    - Typically, the Master node is used for WRITE operations, while the slaves are used for READ operations.
    - The master node ensures consistency among slaves.
    - Typically, there is only one master, making it a single point of failure.
- Master-Master architecture:
    - In this architecture, all the nodes are the same. They keep the information about others by using different communication protocols.
    - Data is equally shared by all nodes. Typically, data is distributed by means of consistent hashing.
    - The client can connect to any of the nodes for READ/WRITE operations irrespective of whether data is present in that node or not.

Q.9 - Explain the concept of CAP Theorem using examples.

Ans - CAP Theorem states that at any time in a system, one can only support two of these three entities:

- Consistency
- Availability
- Partition tolerance

You cannot achieve consistency and availability without partition tolerance. Hence, partition tolerance is a must in the distributed systems. According to the CAP theorem, you can support either consistency or availability.

Therefore, CAP theorem has two systems:

- CP system: In this system, C stands for consistency and P stands for partition tolerance. Thus, the CP systems have a high consistency along with partition tolerance.

  Example - The IRCTC ticket booking needs to be highly consistent to avoid multiple bookings. Here, availability is compromised upon frequently.

- AP system: In the AP systems, A stands for availability and P stands for partition tolerance. Thus, the AP systems have a high consistency along with partition tolerance.

  Example - Search engines such as Bing and Google are also AP systems as availability is a must, whereas consistency can be compromised on.

Q.10- In which scenario can you use Amazon S3 for storage purposes. Explain this using some examples.

Ans - S3 is used when you have to store structured/unstructured data of very high volume/velocity. On the other hand, S3 cannot be used for live applications. Instead, it is used for analytical purposes.

S3 can be used by dumping the log files of different components and applications of your application. It can also be used to store more user data.

Example -

- Amazon tracks all of its users' click data on its e-commerce website and dumps that into S3. From S3, all this data is dumped into the ML/Analytical pipeline for customising the user experience on Amazon.com.

- Another use case of using S3 is that log files of every service being provided by AWS can be stored in S3.

Q.11- In which scenario can you use Apache Spark. Explain with some examples.

Ans - Apache Spark provides a powerful distributed computational system and is widely used in the industry due to its high-speed computational ability.

Spark embodies the fault tolerance of distributed systems and offers lightning-fast speed owing to its in-memory computation model and other optimisation techniques. Also, Spark follows a master-slave architecture, with a master node along with a cluster manager acting as a master and several worker nodes acting as slaves.

Example -

- NASA leverages Spark for log data processing and advanced analytics.

- PySpark, which is a Python API for Spark, is widely used for Data Science and Machine learning when the data sets are simply too large for a single computer.

Q.12- In which scenario can you use Apache Kafka. Use some examples to explain it.

Ans - Apache Kafka, an open-source distributed event streaming platform, is used as a messaging system for the distributed systems.

Kafka can be used in the following systems/scenarios:

- Building non-blocking systems
- Asynchronous systems
- Streaming data ingestion systems where data is collected in real time

Example -

- All the click events on LinkedIn are first moved to Kafka, which acts as the ingestion gateway for the big data ecosystem.

- Netflix uses Kafka for its real-time monitoring and event processing pipelines.

Que.13- In which scenario can you use MongoDB. Explain using some examples.

Ans - MongoDB is a distributed NoSQL database management system. MongoDB can be used as a storage layer for live applications. It meets the transactional online transaction processing requirements.

MongoDB can be used to deal with structured data, but the volume of data that needs to be stored cannot fit in a single machine.

We can use MongoDB when we need to deal with structured data, but the volume of data that needs to be stored does not fit in a single machine and is designed specifically for high-load websites.

Example -

- Adobe relies on MongoDB data management for its cloud data, which is in the scale of petabytes.

Que.14- What do you understand by consistent hashing?

Ans - **Consistent Hashing** is a distributed hashing scheme that operates independently of the number of servers or objects in a distributed hash table by assigning them a position on an abstract circle, or hash ring. This allows servers and objects to scale without affecting the overall system.