

**25 MUST
KNOW
SYSTEM
DESIGN
CONCEPTS**



1. Distributed cache

- These are in-memory data stores that are used to cache data in order to improve the performance of a distributed system.
- Examples include Memcached and Redis.

2. Distributed messaging systems

- These are used to send messages between distributed components of a system.
- Examples include Apache Kafka and RabbitMQ.

3. Distributed file systems

- These are file systems that are designed to store and manage files across a group of servers.
- Examples include HDFS (Hadoop Distributed File System) and Ceph.

4. Distributed databases

- These are databases that are designed to be distributed across multiple servers in order to scale and handle a large amount of data.
- Examples include Google Cloud Spanner and CockroachDB.

5. Load balancers

- These are used to distribute incoming traffic across a group of servers in order to improve performance and availability.
- Examples include HAProxy and NGINX.

6. Distributed coordination systems

- These are used to coordinate the actions of distributed components in a system.
- Examples include Apache ZooKeeper and etcd.

7. Service discovery

- These are used to help components of a distributed system locate and communicate with each other.
- Examples include Consul and Eureka.

8. Monitoring and metrics

- These are used to monitor the health and performance of a distributed system and detect and troubleshoot problems.
- Examples include Prometheus and Datadog.

9. Service mesh

- These are used to manage the communication between the various services in a distributed system.
- Examples include Istio and Linkerd.

10. Distributed tracing

- These are used to trace the flow of a request through a distributed system and help identify performance bottlenecks and errors.
- Examples include Zipkin and Jaeger.

11. Distributed locks

- These are used to coordinate access to shared resources in a distributed system.
- Examples include Redlock and DynamoDB.

12. Distributed schedulers

- These are used to schedule tasks to be executed in a distributed system.
- Examples include Kubernetes and Apache Mesos.

13. Distributed storage

- These are used to store and manage data in a distributed system.
- Examples include Google Cloud Storage and Amazon S3.

14. Remote procedure call (RPC)

- These are used to allow components of a distributed system to communicate with each other by making remote procedure calls.
- Examples include gRPC and Apache Thrift.

15. Stream processing

- These are used to process streams of data in real-time in a distributed system.
- Examples include Apache Flink and Apache Spark.

16. Distributed consensus

- These are used to achieve agreement among distributed components in a system.
- Examples include the Paxos and Raft algorithms.

17. Virtualization

- These are used to abstract the underlying hardware in a distributed system and allow multiple operating systems to run on a single physical machine.
- Examples include VMware and VirtualBox.

18. Job queues

- These are used to asynchronously execute tasks in a distributed system.
- Examples include Celery and Resque.

19. Proxies

Proxies act as intermediaries between clients and servers, forwarding requests and responses between them.

20. Reverse Proxies

Reverse proxies sit in front of a group of servers, forwarding requests to the appropriate server based on the request's URL or other identifying information.

21. Network file system (NFS)

- These are used to allow multiple computers to access and share files over a network.
- Examples include NFS and SMB.

22. Full-text search

Full-text search enables users to search for specific words or phrases within an app or website. To search quickly and efficiently, full-text search relies on an inverted index, which is a data structure that maps words or phrases to the documents in which they appear.

23. Edge Computing

Edge computing refers to processing data at the edge of a network, closer to the source of the data, rather than at a central location. This can be useful in situations where low latency or bandwidth constraints make it impractical to send data to a central location for processing.

24. Notifications and alerts

Distributed systems may include features that allow them to send notifications or alerts to users, such as emails, push notifications, or text messages. These notifications can be used to keep users informed about important events or updates within the system.

25. Content delivery networks (CDN)

CDNs allow users to deliver static content (such as web pages, videos, and images) to users more quickly by storing copies of the content on servers that are physically closer to the users.

➔ Check the following courses from **DesignGurus.io** to learn about these distributed systems and components:

Grokking the Advanced System Design Interview and **Grokking the System Design Interview** at DesignGurus.io

DesignGurus.io