1. Dealing with heavy read system – Cache.
2. If we need low latency in the system – Cache-CDN.
3. dealing with heavy write operation - Message queue for async processing.
4. System to ACID compliant – RDBMS
5. If we do not require ACID properties – No SQL Database
6. To handle complex videos and images – Blob / Object Storage.
7. Complex computation like news feed - Message Queue and Cache
8. Requires search data in hig volume of data – Search Index , Tries ,search emngine like elastic search
9. For scaling DB – DB Sharding
10. High availability , performance ,throughput – Load Balancer
11. Faster data delivery globally reliability and high availability - CDN
12. If the system has data nodes ,edge relationship like friend list ,road connection consider using Graph DB
13. Scaling component like server and DB – Horizontal scaling
14. High performance DB queries – DB indexes
15. Batch processing and Message Queue - Bulk Job processing
16. Reduce server load and preventing DOS attack use Rate Limiter
17. If using microservice – API Gateway (Authentication , SSL Terminator , Routing)
18. System has single point of failure – implement redundancy in that component
19. Fault tolerance , durable – implement Data replication
20. Bidirectional fast way communication – use WEbsocketrs
21. To detect failures in distributed system – implement heartbeat
22. To ensure data integrity implement checksum algorithm
23. Transfer data between servers in decentralized way – Gossip protocol
24. Scale Server add /remove of node efficiently , no hotspots – consistent hashing
25. System needs maps ,resources – Geohash, Quad Tree
26. Use generic names lke Object strorage / Message queue instead of using specific technologies like kafka , S3 , EC2
27. Strong consistency is better than high availabitliy
28. Domain name query in broser works – DNS
29. Pagination to limimt the data.
30. LRU cache for evict cahce