LOAD BALANCING IN SYSTEM DESIGN – A PRACTICAL GUIDE FOR INTERVIEWS & REAL-WORLD ARCHITECTURE

1. INTRODUCTION TO LOAD BALANCING

WHAT IS LOAD BALANCING?

Load balancing is the process of distributing network or application traffic across multiple servers to ensure reliability, scalability, and performance. In a distributed system, it's a crucial component that helps handle traffic spikes, prevent overloads, and improve fault tolerance.

WHY IS IT IMPORTANT?

- Ensures **high availability** and **reliability**.
- Increases **throughput** and reduces **latency**.
- Enables **horizontal scaling**.
- Supports **fault isolation**.

A simple architecture showing a load balancer distributing traffic among multiple servers.

2. TYPES OF LOAD BALANCING

LAYER 4 LOAD BALANCING (TRANSPORT LAYER)

- Operates at the TCP/UDP level.
- Makes routing decisions based on IP address and port.
- Fast and efficient but lacks application-layer insight.

LAYER 7 LOAD BALANCING (APPLICATION LAYER)

- Operates at the HTTP/HTTPS level.
- Makes decisions based on URL path, headers, cookies, etc.

• More flexible and intelligent routing.

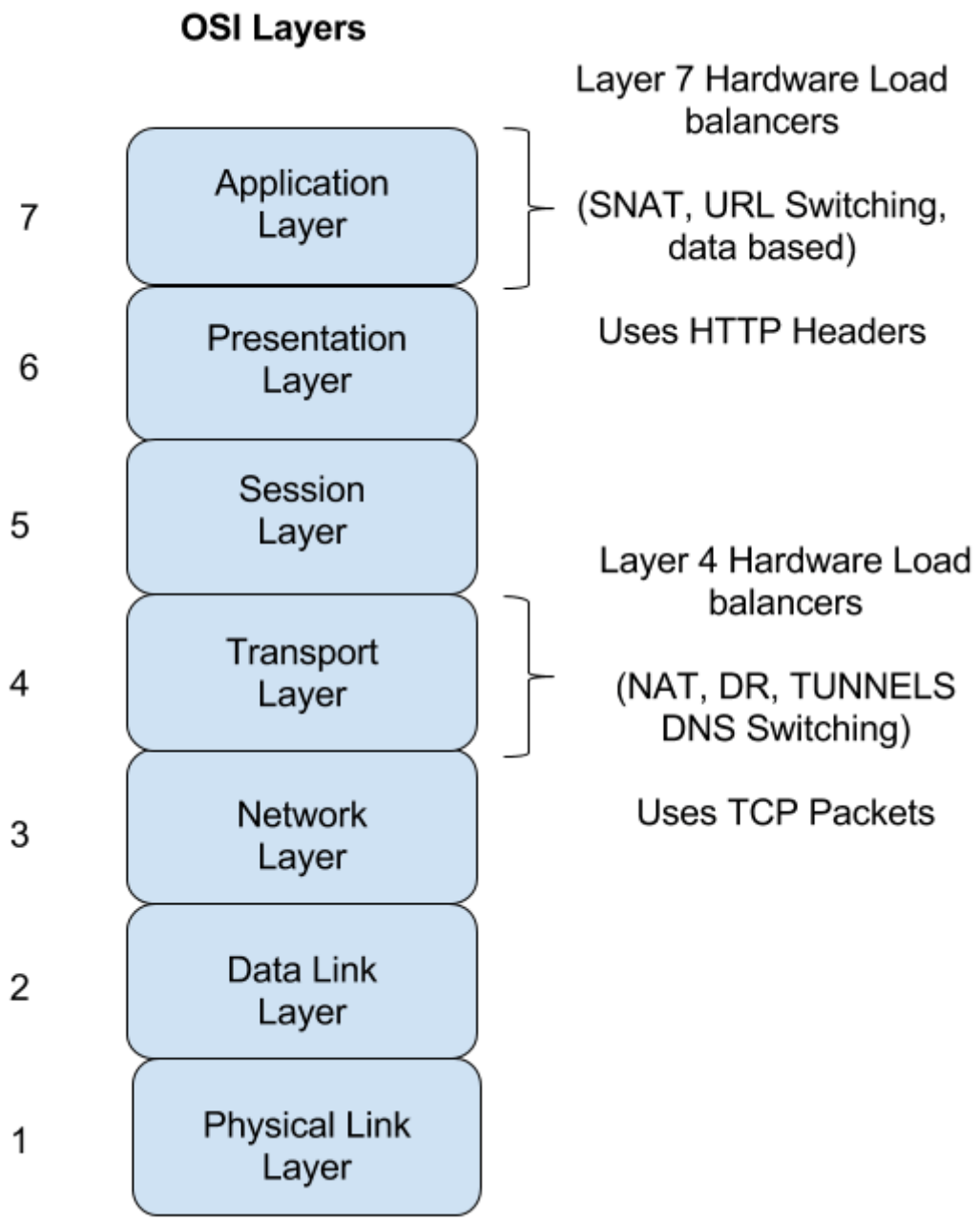


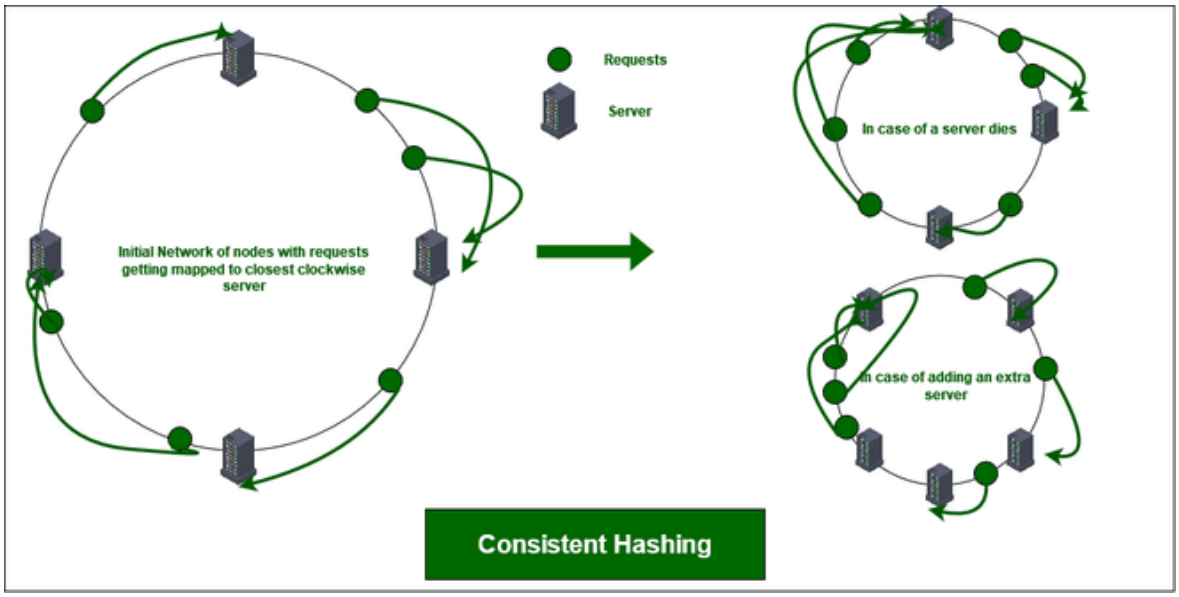
Illustration of Layer 4 and Layer 7 load balancing in the OSI model.

GLOBAL VS LOCAL LOAD BALANCING

• Global: Routes traffic across different geographical regions/data centers.
• Local: Distributes traffic within a single region/data center.

3. LOAD BALANCING ALGORITHMS

1. **Round Robin**
    - Requests are distributed sequentially.
    - Simple, but doesn't consider server load.
2. **Least Connections**
    - Sends traffic to the server with the fewest active connections.
    - Great for long-lived sessions.
3. **IP Hashing**
    - Consistent routing based on user's IP.
    - Helps with session stickiness.
4. **Weighted Round Robin**
    - Assigns weights to servers based on capacity.
    - Distributes traffic accordingly.
5. **Random with Weight**
    - Randomly selects a server, but considers weight.
6. **Consistent Hashing**
    - Used in distributed caches and partitioned data systems.
    - Helps avoid massive reshuffling during scaling.



Visual representation of consistent hashing for distributed data.

4. REAL-WORLD LOAD BALANCERS

- **NGINX**
    - Popular open-source HTTP and reverse proxy server.

    - Supports L7 and L4.
- **HAProxy**
    - High performance TCP/HTTP load balancer.
    - Used in large-scale setups.
- **AWS ELB (Elastic Load Balancer)**
    - Offers Application, Network, and Gateway Load Balancers.
    - Fully managed.
- **Cloudflare / Fastly**
    - CDN providers that also perform L7 load balancing and edge routing.

5. SESSION PERSISTENCE (STICKY SESSIONS)

STICKY SESSIONS

- Ensures a user's requests go to the same server.

USE CASES:

- Shopping carts, temporary user state, non-distributed sessions.

DRAWBACKS:

- Can cause uneven traffic distribution.
- Harder to scale horizontally.

6. HEALTH CHECKS & FAILOVER

ACTIVE HEALTH CHECKS:

- Load balancer periodically pings backend servers.

PASSIVE HEALTH CHECKS:

- Detects failure based on response errors/timeouts.

FAILOVER:

- Automatically routes traffic away from unhealthy servers.

7. SCENARIOS & ARCHITECTURE DIAGRAMS

WEB SERVER LOAD BALANCING

Clients → Load Balancer → Web Servers → Application Servers

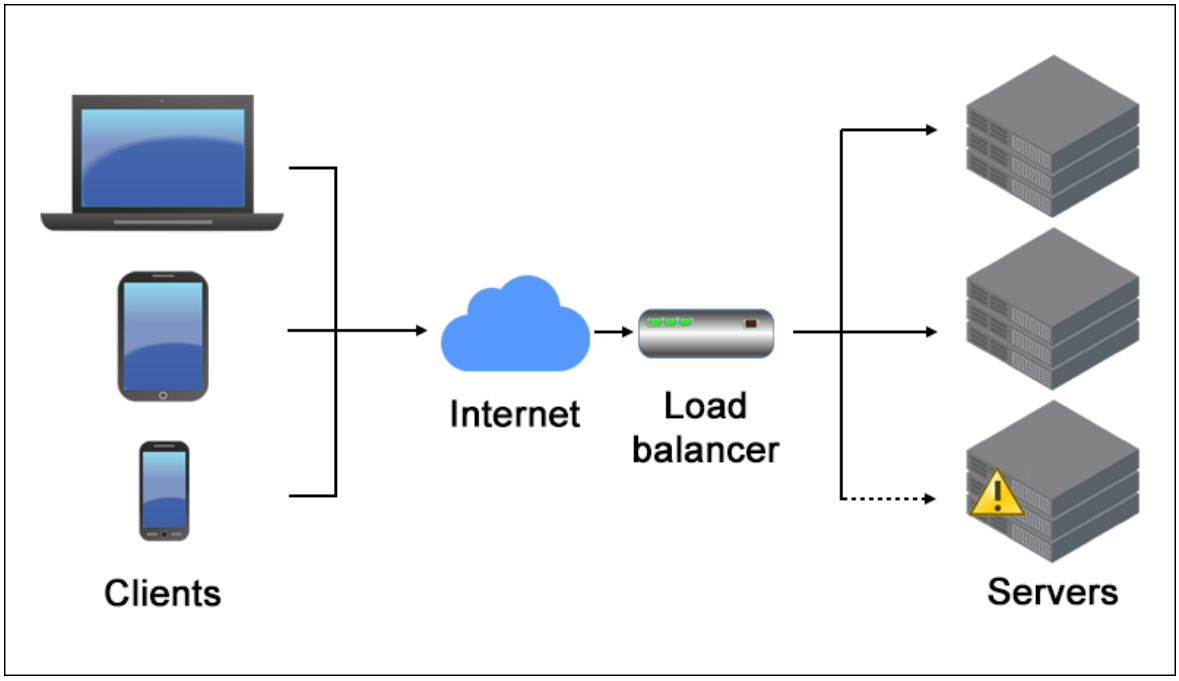


Diagram depicting client requests routed through a load balancer to web servers and a database.

DATABASE LOAD BALANCING

- Writes → Primary
- Reads → Multiple replicas via load balancer

MICROSERVICES ARCHITECTURE

- Service-to-service communication through API Gateway or Service Mesh (e.g., Istio)

8. WHEN TO USE WHICH LOAD BALANCER

Scenario	Best Option
High traffic APIs	NGINX or HAProxy

Scenario	Best Option
Global user base	Cloudflare Global LB
Internal microservices	Envoy or Istio
Cloud-native app	AWS ALB / NLB
Stateful sessions	L7 with sticky sessions

9. TRADE-OFFS & PITFALLS

- **Latency vs Redundancy**
  - Multi-region load balancing increases redundancy but may introduce latency.
- **Cost vs Performance**
  - More complex systems (CDNs, regional LB) increase costs.
- **Single Point of Failure**
  - If the load balancer goes down and isn't redundant, it breaks the whole system.
- **Debugging Complexity**
  - More components = more possible failure points.

10. INTERVIEW QUESTIONS + ANSWERS

1. **What is Load Balancing?**
   Load balancing is a method of distributing network or application traffic across multiple servers to ensure no single server bears too much demand.
2. **Why is Load Balancing important in system design?**
   It ensures availability, fault tolerance, and high scalability while reducing latency and preventing server overload.
3. **What are the main types of Load Balancers?**
   Layer 4 (Transport), Layer 7 (Application), and Global Load Balancers (DNS-based or geo-distributed).
4. **What is a Layer 4 Load Balancer?**
   It uses transport-level info (IP, port, TCP/UDP) to distribute traffic without inspecting application-level content.
5. **What is a Layer 7 Load Balancer?**
   Works at the application layer, routing based on content like URL, headers, or cookies.

6. **What is a Global Load Balancer?**
   Distributes traffic across global data centers using DNS or Anycast for geo-routing.
7. **What is Round Robin Load Balancing?**
   Distributes requests evenly across servers in a circular order regardless of current load.
8. **What is the Least Connections strategy?**
   Directs traffic to the server with the fewest active connections at the time.
9. **What is IP Hashing?**
   Routes requests based on a hash of the client's IP, ensuring session stickiness.
10. **What is Sticky Session?**
    Ensures a user session is always handled by the same server to maintain session context.
11. **What are the drawbacks of Sticky Sessions?**
    Reduces scalability and fault tolerance, as state is tied to a specific server.
12. **What is Health Checking in Load Balancing?**
    Monitors servers and removes unhealthy instances from the traffic rotation.
13. **What's the difference between Active and Passive Health Checks?**
    Active sends periodic probes; Passive relies on actual user traffic failures.
14. **What is Failover in Load Balancing?**
    Automatically redirects traffic from failed instances to healthy ones.
15. **What is a Reverse Proxy Load Balancer?**
    Acts as an intermediary between clients and servers, hiding server details and balancing traffic.
16. **What is Horizontal Scaling?**
    Adding more servers to handle traffic instead of increasing a single server's capacity.
17. **What is a CDN Load Balancer?**
    A CDN uses geo-based load balancing to deliver content from the nearest edge location.
18. **What is SSL Termination in Load Balancing?**
    Offloads the SSL handshake from backend servers to the load balancer.
19. **What is Weighted Round Robin?**
    Assigns weights to servers; servers with higher weight get more requests.

20. **What is Auto-scaling and how does it relate to Load Balancing?**
    Auto-scaling adds/removes servers based on demand; load balancer adapts by redistributing traffic.
21. **How does Load Balancing affect performance?**
    Prevents bottlenecks, ensures high availability, and reduces latency.
22. **How do DNS Load Balancers work?**
    They resolve the same domain to different IPs based on policies like geo-location or round robin.
23. **What is Anycast Load Balancing?**
    Routes traffic to the nearest or lowest-latency server using the same IP globally.
24. **What is Application Gateway in cloud systems?**
    A managed Layer 7 load balancer that provides routing, SSL termination, and WAF integration.
25. **What's the role of Load Balancer in a microservices architecture?**
    It routes requests to appropriate services, maintains resilience, and handles service discovery.
26. **What's the difference between Load Balancer and API Gateway?**
    LB focuses on distributing traffic; API Gateway handles auth, throttling, and protocol translation.
27. **What are common Load Balancer algorithms?**
    Round Robin, Least Connections, IP Hash, Weighted, Random, and Least Response Time.
28. **What is Load Shedding?**
    Proactively rejecting excess traffic to prevent overload and allow graceful degradation.
29. **What is Circuit Breaker pattern?**
    Prevents calls to failing services temporarily, often used alongside load balancers.
30. **Can Load Balancers prevent DDoS attacks?**
    They help distribute traffic but need to be combined with firewalls and rate limiters for true protection.
31. **What is Blue-Green Deployment?**
    Uses two identical environments (blue and green) to switch traffic with zero downtime.
32. **What is Canary Deployment?**
    Gradually rolls out new versions to a subset of users before full deployment.
33. **What's the role of Load Balancer in High Availability systems?**
    Automatically redirects traffic to healthy instances, ensuring minimal downtime.

34. **How does Load Balancing support Disaster Recovery?**
    Can redirect traffic to backup data centers during failures or outages.
35. **What are sticky vs stateless load balancing models?**
    Sticky binds session to server; stateless distributes independently, requiring shared state handling.
36. **What is Server Affinity?**
    Another term for session stickiness, directing a user to the same backend server.
37. **Can Load Balancers perform compression?**
    Yes, some advanced load balancers compress/decompress payloads to improve performance.
38. **What is TCP vs HTTP Load Balancing?**
    TCP (L4) works at lower level and is faster; HTTP (L7) offers smarter routing.
39. **How is health check configured in AWS ELB?**
    Define a protocol, port, and path. If it fails N times consecutively, the target is marked unhealthy.
40. **What's the difference between Classic, ALB, and NLB in AWS?**
    Classic: older, supports both L4 & L7; ALB: application-level; NLB: ultra-high performance at L4.
41. **How does GCP handle load balancing?**
    Via Global HTTP(S), SSL Proxy, TCP Proxy, and Internal Load Balancers, each for different use cases.
42. **What's Kubernetes Ingress vs LoadBalancer type?**
    Ingress routes HTTP; LoadBalancer type provisions an external IP via cloud provider.
43. **What is a Backend Pool in Load Balancing?**
    A group of target servers that receive traffic from the load balancer.
44. **What are sticky sessions in Azure Load Balancer?**
    Azure supports session persistence via client IP or client IP + protocol.
45. **How does Load Balancing work in Service Mesh (e.g. Istio)?**
    Sidecars manage routing, retries, failover, and telemetry for microservices.
46. **What are virtual IPs in Load Balancing?**
    An IP that abstracts backend servers; load balancer routes traffic to the real server behind it.
47. **How do you monitor Load Balancer health?**
    Use logs, metrics, and alerts from cloud providers, Prometheus, or Grafana dashboards.

48. **What is Cross-Zone Load Balancing?**
    Allows requests to be evenly distributed across all backend instances in different zones.
49. **What is a Load Balancer timeout?**
    Defines how long the LB waits for a server response before marking it failed.
50. **What's the biggest tradeoff in Load Balancing?**
    Balancing complexity vs performance — overengineering can increase latency, underengineering can risk downtime.

✅ CONCLUSION:

Load balancing isn't just a backend concern—it's foundational for scalable, reliable systems. Mastering its principles can elevate your system design skills and make a real impact in your next interview or project.