Basic load balancer concepts

Every horizontally scaled service uses a load balancer, which may be one of the following:

A hardware load balancer, a specialized physical device that distributes traffic across multiple hosts. Hardware load balancers are known for being expensive and can cost anywhere from a few thousand to a few hundred thousand dollars.

A shared load balancer service, also referred to as LBaaS (load balancing as a service).

A server with load balancing software installed. HAProxy and NGINX are the most common.

This section discusses basic concepts of load balancers that we can use in an interview.

In the system diagrams in this book, I draw rectangles to represent various services or other components and arrows between them to represent requests. It is usually understood that requests to a service go through a load balancer and are routed to a service's hosts. We usually do not illustrate the load balancers themselves.

We can tell the interviewer that we need not include a load balancer component in our system diagrams, as it is implied, and drawing it and discussing it on our system diagrams is a distraction from the other components and services that compose our service.

Level 4 vs. level 7

We should be able to distinguish between level 4 and level 7 load balancers and discuss which one is more suitable for any particular service. A level 4 load balancer operates at the transport layer (TCP). It makes routing decisions based on address information extracted from the first few packets in the TCP stream and does not inspect the contents of other packets; it can only forward the packets. A level 7 load balancer operates at the application layer (HTTP), so it has these capabilities:

Load balancing/routing decisions—Based on a packet's contents.

Authentication—It can return 401 if a specified authentication header is absent.

TLS termination—Security requirements for traffic within a data center may be lower than traffic over the internet, so performing TLS termination (HTTPS → HTTP) means there is no encryption/decryption overhead between data center hosts. If our application requires traffic within our data center to be encrypted (i.e., encryption in transit), we will not do TLS termination.

Sticky sessions

A sticky session refers to a load balancer sending requests from a particular client to a particular host for a duration set by the load balancer or the application. Sticky sessions are used for stateful services. For example, an ecommerce website, social media website, or banking website may use sticky sessions to maintain user session data like login information or profile preferences, so a user doesn’t have to reauthenticate or reenter preferences as they navigate the site. An ecommerce website may use sticky sessions for a user’s shopping cart.

A sticky session can be implemented using duration-based or application-controlled cookies. In a duration-based session, the load balancer issues a cookie to a client that defines a duration. Each time the load balancer receives a request, it checks the cookie. In an application-controlled session, the application generates the cookie. The load balancer still issues its own cookie on top of this application-issued cookie, but the load balancer’s cookie follows the application cookie’s lifetime. This approach ensures clients are not routed to another host after the load balancer’s cookie expires, but it is more complex to implement because it requires additional integration between the application and the load balancer.

Session replication

In session replication, writes to a host are copied to several other hosts in the cluster that are assigned to the same session, so reads can be routed to any host with that session. This improves availability.

These hosts may form a backup ring. For example, if there are three hosts in a session, when host A receives a write, it writes to host B, which in turn writes to host C. Another way is for the load balancer to make write requests to all the hosts assigned to a session.

Load balancing vs. reverse proxy

You may come across the term “reverse proxy” in other system design interview preparation materials. We will briefly compare load balancing and reverse proxy.

Load balancing is for scalability, while reverse proxy is a technique to manage client–server communication. A reverse proxy sits in front of a cluster of servers and acts as a gateway between clients and servers by intercepting and forwarding incoming requests to the appropriate server based on request URI or other criteria. A reverse proxy may also provide performance features, such as caching and compression, and security features, such as SSL termination. Load balancers can also provide SSL termination, but their main purpose is scalability.