Load balancing

Definition

- Client to gateway:
  - Implementation: DNS resolves to different ip address.
- Gateway to web server:
  - Implementation: Nginx reverse proxy
    - Proxy hides itself
    - Reverse proxy hides the server pool.
- Web server to application server:

    - Implementation: Connection pool
- Application server to database:
    - Implementation: Partition / Sharding

Benefits

- Decoupling
    - Hidden server maintenance. You can take a web server out of the load balancer pool, wait for all active connections to drain, and then safely shutdown the web server without affecting even a single client. You can use this method to perform rolling updates and deploy new software across the cluster without any downtime.
    - Seamlessly increase capacity. You can add more web servers at any time without your client ever realizing it. As soon as you add a new server, it can start receiving connections.
    - Automated scaling. If you are on cloud-based hosting with the ability to configure auto-scaling (like Amazon, Open Stack, or Rackspace), you can add and remove web servers throughout the day to best adapt to the traffic.
- Security
    - SSL termination: By making load balancer the termination point, the load balancers can inspect the contents of the HTTPS packets. This allows enhanced firewalling and means that you can balance requests based on teh contents of the packets.
    - Filter out unwanted requests or limit them to authenticated users only because all requests to back-end servers must first go past the balancer.
    - Protect against SYN floods (DoS attacks) because they pass traffic only on to a back-end server after a full TCP connection has been set up with the client.

Round-robin algorithm

- Def: Cycles through a list of servers and sends each new request to the next server. When it reaches the end of the list, it starts over at the beginning.

- Problems:
    - Not all requests have an equal performance cost on the server. But a request for a static resource will be several orders of magnitude less resource-intensive than a requst for a dynamic resource.
    - Not all servers have identical processing power. Need to query back-end server to discover memory and CPU usage, server load, and perhaps even network latency.
    - How to support sticky sessions: Hashing based on network address might help but is not a reliable option. Or the load balancer could maintain a lookup table mapping session ID to server.

Hardware vs software

Category	Software	Hardware
Def	Run on standard PC hardware, using applications like Nginx and HAProxy	Run on special hardware and contain any software pre-installed and configured by the vendor.
Model	Operate on Application Layer	Operate on network and transport layer and work with TCP/IP packets. Route traffic to backend servers and possibly handling network address translation
Strength/Weakness	More intelligent because can talk HTTP (can perform the compression of resources passing through and routing-based on the presence of cookies) and more flexible for hacking in new features or	Higher throughput and lower latency. High purchase cost. Hardware load balancer prices start from a few thousand dollars and go as high as over 100,000 dollars per device. Specialized training and harder to find people with the work experience necessary to operate them.

	changes	

HAProxy vs Nginx

Category	Nginx	HAProxy
Strengths	Can cache HTTP responses from your servers.	A little faster than Nginx and a wealth of extra features. It can be configured as either a layer 4 or layer 7 load balancer.

- Extra functionalities of HAProxy. It can be configured as either a layer 4 or layer 7 load balancer.
    - When HAProxy is set up to be a layer 4 proxy, it does not inspect higher-level protocols and it depends solely on TCP/IP headers to distribute the traffic. This, in turn, allows HAProxy to be a load balancer for any protocol, not just HTTP/HTTPS. You can use HAProxy to distribute traffic for services like cache servers, message queues, or databases.
    - HAProxy can also be configured as a layer 7 proxy, in which case it supports sticky sessions and SSL termination, but needs more resources to be able to inspect and track HTTP-specific information. The fact that HAProxy is simpler in design makes it perform sligthly better than Nginx, especially when configured as a layer 4 load balancer. Finally, HAProxy has built-in high-availability support.