Introduction to Load Balancers

Johan von Hacht
johvh@kth.se

April 2020

1 Introduction

Modern websites such as Netflix receive millions of concurrent requests every day [20]. In Netflix's case, it is not feasible for a single server to handle all of these requests. Therefore, in order to scale to the high amount of traffic, multiple servers are needed. However, a major problem is deciding which traffic goes to which server while still ensuring efficient and reliable service.

The job of the load balancers is to make those decisions, i.e. to efficiently distribute traffic across multiple servers [25]. Figure 1 shows the flow of traffic when utilising a load balancer. All of the user's requests are routed through the balancer which forwards the request to one of the back-end servers, determined by a routing algorithm. In later sections, different types of load balancers, as well as routing algorithms, will be described.

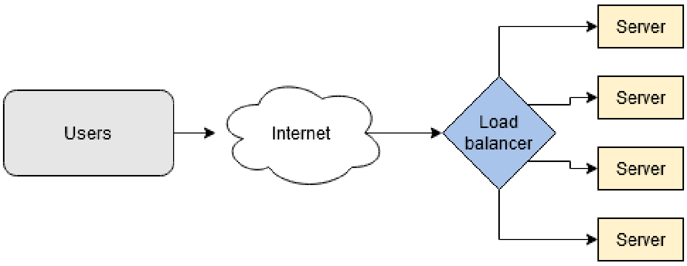

Figure 1: Traffic flow chart.

Beyond the ability to increase the scale of operations, load balancers provide many other benefits as well. Firstly, they can reduce downtime in the case of server outages because they can simply route traffic to other servers not affected by the outage. Additionally, it can make traffic routing more efficient since it can route users to server experiencing low load and response times. Lastly, they can also function as a form of security measure against, for example, DDOS[1] attacks because they disperse the traffic over multiple servers [3].

[1] Distributed denial-of-service (DDoS) attack is an attempt to overwhelm a target server or infrastructure with the intent to disrupt normal traffic [15].

2 Types of load balancers

There are two main types of load balancing techniques, static and dynamic load balancing. A static load balancer relies on predetermined values to decide which server to forward requests to. These values do not change during run-time which means that traffic routing stays the same regardless of system load changes. Dynamic load balancers, on the other hand, decides how to forward traffic based on information gained during run-time, such as current load on the servers. Infrastructure experiencing constant traffic load will perform better with static load balancers while dynamic load balancers will perform better when traffic load is unpredictable. However, dynamic load balancers can be more difficult to implement [19].

Furthermore, all types of load balancing happen in two of the layers in the OSI model[2], the transport layer (layer 4) or the application layer (layer 7). A load balancer in the application layer is called an application load balancer (ALB) and a load balancer in the transport layer is called a network load balancer (NLB) [3].

Network load balancers can only use the information available at the transport layer to make routing decisions. Subsequently, an NLB can only make rudimental decisions since it is limited to information about the requests source and destination IP address and ports. It does not have access to the content of the message it is forwarding [23].

Application load balancers, on the other hand, can make decisions based on the contents of the requests they forward. This allows the ALB to make smarter decisions based on information such as HTTP headers, SSL and HTML form data [3].

One drawback with application-layer load balancers is that they require more resources to operate compared to an NLB. However, according to nginx [24] they seldom cause degraded performance on modern servers.

3 Routing algorithms

When a load balancer receives traffic, it has to decide which destination to forward it to. It does this by following a routing algorithm. A few such algorithms will be described in the following sections.

3.1 Round-Robin

Round-robin (RR) is one of the more widely used routing algorithms. It works by forwarding traffic to available servers in a cyclical manner [2]. As an example, assume that a website has two back-end servers. With round-robin the requests would be alternated between the servers in the following way:

- Request 1 → server A
- Request 2 → server B
- Request 3 → server A
- ...

The algorithm is easy to implement due to its simplistic nature. However, it assumes that the servers have identical capacity. This is not always the case and if one server has a lower capacity than the others, it runs the risk of being overloaded [2].

[2] The OSI model is a standard that allows computers regardless of software systems to communicate with each other [17].

There is a variation of round-robin called weighted round-robin (WRR) where each server is assigned a weight. The weights determine how large portion of the requests each server should handle [26]. WRR solves the aforementioned problem with RR where servers with less capacity ran the risk of overloading. This is because a smaller weight can now be applied to servers with less capacity.

3.2 Least Connections/Outstanding Requests

The least connections algorithm takes the current load of the servers into account when routing traffic. When a request is made, it is forwarded to the server with the least current active connections or outstanding requests [21].

If the capacities of the servers differ greatly, a similar problem as seen in RR can occur. Assume there are two servers, server A with capacity 50 and server B with capacity 100. If server A had 49 connections and server B 50, the least connection algorithm would route the request to server A. However, server A is operating at near capacity, thus it would be better for server B to handle to request. To solve this problem, weights can be implemented in the same way as WRR [28].

A disadvantage with this method is that it can be resource-intensive to keep track of all the open connections and outstanding requests [27].

3.3 Least Time

The least time algorithm considers the time taken for the servers to respond (latency) when making routing decisions. The load balancer gains this information by sending health monitoring requests to the servers and measuring the time it takes for them to respond. When a request is received it is forwarded to the server that took the least amount of time to respond. The theory behind the algorithm is that the speed of the server indicates how loaded the server is [14].

One drawback of this strategy is that workload is often filled up one server at a time. This is because response time is highly dependant on other factors such as physical distance from the load balancer [27].

3.4 IP-hash

The IP-hash algorithm utilises the incoming request's source IP-address to determine which back-end server to forward the request to. Since the source IP address is hashed, a client will always connect to the same back-end server. This characteristic is useful if the client should connect to the same session even if they disconnect and then later reconnect [22].

An issue occurs if a lot of traffic originates from the same source address as it would be routed to the same server, potentially overloading it. However, one can overcome this issue by limiting the number of connections that can be made from the same source to the same back-end server [13].

4 State of the art

The following section highlights the current state of the art of load balancers. First, a method to load balance whole data centres will be described. Thereafter, load balancers options and configurations from two major cloud computing platforms will be presented.

4.1 Global server load balancing

The world is becoming ever more globalised and with that development comes new requirements for applications and services. It is no longer enough to have a single data centre in one location. A single location server setup would, for instance, mean high latency for users on the other side of the world due to the physical limitations of data transfer. Subsequently, a more and more common approach is to have multiple load-balanced server clusters all over the world, often called global server load balancing (GSLB). In other words, instead of load balancing between singular servers, GSLB takes that idea further and balances traffic between whole data centres or regions [18].

Figure 2 shows the traffic flow with GSLB. When a user requests a website utilising GSLB, the DNS[3] response will depend on the geographic location of the client. The global load balancer would identify the server closest to the users' location, defined by some criteria, and adjust the DNS reply. The reply contains information such as the IP address of the server that the client should connect to, to access the website [18].

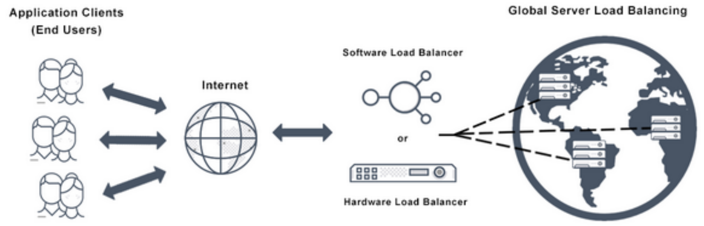

Figure 2: Traffic flow when using GSLB. Image source avinetworks [1].

Using GSLB improves the user experience because the users can be routed to the server closest to them. Furthermore, in the unlikely scenario that a whole data centre was to be disrupted, the application or service would be minimally affected since users would be routed to another data centre. Lastly, it is also easy to scale and maintain the service because whole data centres can be added to the pool of available routes or temporarily turned off and upgraded [18].

4.2 Cloud computing load balancers

It is too expensive for small companies or organisations to have data centres all over the world. An alternative is to utilise cloud computing platforms that already have the infrastructure set up and pay to use their platform. This section will, therefore, present the types of load balancers and routing algorithms found in two major cloud computing platforms, AWS and Azure.

[3] The Domain Name System (DNS) translates hostnames such as www.abc.com into a computer-friendly IP address used to locate the contents of the web page [16].

4.2.1 AWS

On the AWS platform, there are three types of load balancers available: Application, Network and Classic. The application load balancer is best suited for balancing HTTP and HTTPS traffic while the network load balancer is better suited for UDP and TCP traffic. The classic load balancer is the first load balancer released on AWS but seems to be replaced by the aforementioned ALB and NLB's. AWS describes the classic load balancers use-case in the following way: "Classic Load Balancer is intended for applications that were built within the EC2-Classic network." [6].

AWS uses different routing algorithms depending on the load balancer type and the traffic it encounters. The ALB uses round-robin by default but can be changed to least outstanding requests. The NLB utilises a flow hash routing algorithm, similar to IP Hash, and looks at the protocol, source IP/port, destination IP/port and the TCP sequence numbers of the request to route it to a single server. Lastly, the classic load balancer uses round-robin for TCP and least outstanding requests for HTTP and HTTPS [7].

There are also two services called Route 53 and Global Accelerator which facilitates global server load balancing [5]. Route 53 uses DNS to perform global load balancing as described in section 4.1. The Global Accelerator service uses AWS global network instead of DNS to route traffic. The benefit of using the latter is that the request will go through the AWS network instead of over the public internet, which supposedly decreases latency [4]. Route 53 has multiple routing options available, for example, location-based, latency-based and weighted routing [5]. The Global Accelerator routes based on optimal performance but does not specify what this means in detail [4].

4.2.2 Azure

On the Azure platform, there are four load balancing services available: Front Door, Traffic Manager, Application Gateway and Load Balancer [11].

The Load Balancer is a network load balancer, equal to AWS Network Load Balancer. It can be configured with two different routing options, hash-based distribution and source IP-affinity. Hash-based distribution routes traffic from the same session to the same destination server. The source IP-affinity method on the other hand always routes the client to the same endpoint server, regardless of session [8].

The Application Gateway is an application load balancer, equal to AWS Application Load Balancer. It can route traffic based on rules applied by the network administrator. Routes can, for example, be decided based on the URL of the request so that a.abc.com is routed one server and b.abc.com to another. The Application Gateway can also route to server pools, which are clusters of servers, and thereafter use round-robin to decide which particular server to forward to [10].

For global server load balancing, Azure offers Front Door and Traffic Manager which is equivalent to AWS Global Accelerator and Route 53 respectively. Front Door supports four different traffic routing methods: latency, priority, weighted and session affinity [9]. The latency method selects the server with the lowest response time (least time routing). The priority method allows the administrator to specify primary and back-up servers. If the primary servers were to become unavailable, the back-up servers would be used instead. The weighted method divides the traffic based on the provided weights and works similar to the weighted routing algorithms described in section 3. Lastly, session affinity is meant for stateful applications where users are routed to the same back-end server. The Traffic Manager offers the same routing options as Route 53, that is, e.g. latency, weighted and geographic [12].

5 Conclusions

Without load balancers, the internet and many of the applications we come in contact with every day would not be possible. They are a crucial part of software development and DevOps since they facilitate large-scale, redundant applications and user-friendly experiences. Therefore, as a developer, it is important to be familiar with the different load balancing options and technologies that are available to use.

The load balancing options offered by AWS and Azure are very similar, even down to the routing algorithms used. Both providers network balancer use some form of hashing and their application load balancers use round-robin. However, AWS also offers least outstanding requests for their application load balancers. When it comes to global load balancing they offer equivalent services but Azure has more routing configurations with Front Door compared to AWS Global Accelerator.

References

[1] avinetworks. *Global Server Load Balancing Definition*. URL: https://avinetworks.com/glossary/global-server-load-balancing-2/. (accessed: 29-04-2020).

[2] avinetworks. *Round Robin Load Balancing*. URL: https://avinetworks.com/glossary/round-robin-load-balancing/. (accessed: 24-04-2020).

[3] avinetworks. *What Is Load Balancing?* URL: https://avinetworks.com/what-is-load-balancing/. (accessed: 24-04-2020).

[4] Amazon AWS. *AWS Global Accelerator*. URL: https://aws.amazon.com/global-accelerator. (accessed: 30-04-2020).

[5] Amazon AWS. *Choosing a routing policy*. URL: https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-policy.html#routing-policy-simple. (accessed: 29-04-2020).

[6] Amazon AWS. *Elastic Load Balancing*. URL: https://aws.amazon.com/elasticloadbalancing/. (accessed: 28-04-2020).

[7] Amazon AWS. *How Elastic Load Balancing Works*. URL: https://docs.aws.amazon.com/elasticloadbalancing/latest/userguide/how-elastic-load-balancing-works.html#request-routing. (accessed: 28-04-2020).

[8] Microsoft Azure. *Configure the distribution mode for Azure Load Balancer*. URL: https://docs.microsoft.com/en-us/azure/load-balancer/load-balancer-distribution-mode. (accessed: 29-04-2020).

[9] Microsoft Azure. *Front Door routing methods*. URL: https://docs.microsoft.com/en-us/azure/frontdoor/front-door-routing-methods. (accessed: 29-04-2020).

[10] Microsoft Azure. *How an application gateway works*. URL: https://docs.microsoft.com/bs-cyrl-ba/azure/application-gateway/how-application-gateway-works. (accessed: 29-04-2020).

[11] Microsoft Azure. *Overview of load-balancing options in Azure*. URL: https://docs.microsoft.com/en-us/azure/architecture/guide/technology-choices/load-balancing-overview. (accessed: 29-04-2020).

[12] Microsoft Azure. *Traffic Manager routing methods*. URL: https://docs.microsoft.com/en-us/azure/traffic-manager/traffic-manager-routing-methods. (accessed: 29-04-2020).

[13] Zack Busch. *Everything You Need To Know About Load Balancing a Server*. URL: https://learn.g2.com/load-balancer. (accessed: 25-04-2020).

[14] citrix. *What is load balancing?* URL: https://www.citrix.com/glossary/load-balancing.html. (accessed: 24-04-2020).

[15] cloudflare. *What is a DDoS Attack?* URL: https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/. (accessed: 25-04-2020).

[16] cloudflare. *What is DNS?* URL: https://www.cloudflare.com/learning/dns/what-is-dns/. (accessed: 29-04-2020).

[17] cloudflare. *What is the OSI model?* URL: https://www.cloudflare.com/learning/ddos/glossary/open-systems-interconnection-model-osi/. (accessed: 25-04-2020).

[18] efficientip. *What is GSLB?* URL: https://www.efficientip.com/what-is-gslb/. (accessed: 29-04-2020).

[19] Atul Garg. “A comparative study of static and dynamic Load Balancing Algorithms”. In: *IJARCSMS* Volume 2 (Dec. 2014), Page 386–392.

[20] Ansoor Iqbal. *Netflix Revenue and Usage Statistics (2020)*. URL: https://www.businessofapps.com/data/netflix-statistics/#1. (accessed: 24-04-2020).

[21] kemptechnologies. *Load Balancing Algorithms and Techniques*. URL: https://kemptechnologies.com/load-balancer/load-balancing-algorithms-techniques/. (accessed: 24-04-2020).

[22] kemptechnologies. *Source IP Hash load balancing*. URL: https://kemptechnologies.com/glossary/source-ip-hash-load-balancing/. (accessed: 25-04-2020).

[23] nginx. *What Is Layer 4 Load Balancing?* URL: https://www.nginx.com/resources/glossary/layer-4-load-balancing/. (accessed: 25-04-2020).

[24] nginx. *What Is Layer 7 Load Balancing?* URL: https://www.nginx.com/resources/glossary/layer-7-load-balancing/. (accessed: 25-04-2020).

[25] nginx. *What Is Load Balancing?* URL: https://www.nginx.com/resources/glossary/load-balancing/. (accessed: 24-04-2020).

[26] nginx. *What Is Round-Robin Load Balancing?* URL: https://www.nginx.com/resources/glossary/round-robin-load-balancing/. (accessed: 24-04-2020).

[27] J. Potter. *Load Balancing Techniques and Optimizations*. URL: https://www.liquidweb.com/kb/load-balancing-techniques-optimizations/. (accessed: 25-04-2020).

[28] University of Tennessee. *What load balancing methods are available?* URL: https://help.utk.edu/kb/index.php?func=show&e=1699. (accessed: 24-04-2020).