DEVOPS

Load Balancing 101: Understanding Global Load Balancing

Published Apr 13, 2016 • 5 min read

By Nic Benders

This post is the second in a two-part series on load balancing. The first post—Load Balancing 101: The Importance of Local Load Balancing—addressed local load balancing, while part two focuses on global load balancing.

In part one of this series we covered the necessity of using *local* load balancing—that is, directing traffic between multiple servers in the same data center—in front of any service you offer. Here, I'll address the need for global load balancing that some organizations eventually face.

Global load balancing is inherently more complex, in that it involves moving traffic between multiple data centers. Many successful organizations eventually reach a scale where they require multiple data centers.

(The big challenge of this isn't actually how to distribute the load; it's how to ensure that traffic that goes to multiple data centers gets synced in the backend.

ensure that traffic that goes to multiple data centers gets synced in the backend, and that users get the right answer no matter where they go. We're not going to address the data synchronicity issue here—that would be a whole other post. Instead, we're going to focus on how to send traffic to multiple data centers.)

There are two very different reasons why businesses move to multiple data centers:

- **High availability:** In this scenario, you've hit the point where your business can no longer tolerate the risk of housing everything in a single building or region. To protect the business, you typically either bring up a completely offline backup data center in the event of a disaster, or you run multiple active, always-on data centers.
- **Data locality:** Some companies need to run multiple data centers for jurisdictional or regulatory reasons. For example, for data privacy reasons you may need to keep European data within Europe, or for performance reasons you may want to make sure your customers in Asia get all their data from your data center in Asia.

Either way, the load balancing logic that you're trying to implement has to do with the origin of the traffic, not with an even distribution of that traffic. Let's look at several solutions:

1. Leveraging a Content Delivery Network (CDN)

CDNs are adept at managing multiple active data centers globally. Our primary CDN is Fastly, which has multiple points of presence around the world and a cutting-edge IP network that relies on Anycast to guarantee that the closest data center can consistently answer queries.

How it works: When a customer sends data to one of our systems that's CDN-brokered, they're talking to an HTTP proxy at a CDN point of presence close to

them. Then that CDN performs basic filtering and routing (see Load Balancing 101: The Importance of Local Load Balancing). The CDN then sends the request through to us. Traffic goes to the CDN based on both geography and performance, and then the CDN can route it to multiple backend data centers based on the desired load level in each data center or other characteristics.

2. Using DNS

This way of balancing traffic between multiple data centers has been around since the early days of the internet. It was much more common 10 years ago; it's not used as much today.

How it works: As a basic example, suppose you have one host name, such as www.example.com, and it returns one of four different IP addresses, one for each of your data centers. While DNS load balancing is relatively inexpensive to implement and maintain, it's slow (turning it on and off takes minutes) and sometimes unpredictable—many DNS clients will hold on to their responses and continue going to the old destination even after it has been updated.

3. Using Anycast and Border Gateway Protocol

This strategy is similar to the approach a CDN takes, just done in-house. It has become more common recently.

How it works: Each of your data centers proposes itself as a possible route for a set of virtual IPs. Then the originating traffic finds the shortest route to those IPs. It picks a path and brings it to one of your data centers, but instead of the data center actually passing that data through in the backend, it serves the traffic and just impersonates those IPs.

The propagation time is much faster than DNS. You have more control over things like how you weight the traffic among various locations. Because it matches to the physical structure of the internet, it encourages a better performance by picking a path that's close to a data center and close to your traffic origin. However, it requires significant internal network experience to set up and manage.

These are the essential, big-picture ways to do global load balancing. They all come with a certain degree of complexity, but more tools for managing that are becoming available all the time. In a sense, the complexity is a sign of your organization's success: It means you've grown to the point where you need to solve these problems.

Be sure to also read ***Load Balancing 101: The Importance of Local Load Balancing*****.**

Background image courtesy of Shutterstock.com.

Share this article

RELATED TOPICS

DEVOPS

By Nic Benders

Nic Benders is General Manager and Vice President of Telemetry Data Platform at New Relic.