

#STREAMING DEC 12, 2021

# Delivering content the right way, Part III.



PAVEL ŠVEJDA

[CDN](#) [AUTOMATION](#) [VIDEO](#)

Denys Nevozhai

## SERVING CONTENT FROM MULTIPLE CDNS

It's a challenge to provide a top tier video streaming service across Africa. Quality internet networks are few and far between, server locations are often not ideal (we need to connect very distant countries), peering points are limited, and every ISP has different connectivity.

That's where the MultiCDN project shines.

The idea behind the MultiCDN project is to have a choice of CDNs we can connect the customer to, collect data on the quality of experience (QoE) while using each CDN, and, based on that, connect users to CDNs that offer the service with best QoE metrics.

## WHAT IS A CDN?

The textbook **definition**: A content delivery network (CDN) refers to a geographically-distributed group of servers that work together to provide fast delivery of digital content. .

A CDN allows for the quick transfer of assets needed for loading internet content. The popularity of CDN services continues to grow, and today the majority of web traffic is served through CDNs. Text, graphics,

## WHY DO WE USE MULTICDN?

A lot of companies use one CDN that has servers everywhere. The answer is simple — no CDN is ideal for all territories.



imgflip.com

JAKE-CLARK.TUMBLR

On top of that, different datacenters for the same CDN provider can perform quite differently, and sometimes proximity to the end user doesn't actually solve buffering issues. In some countries, we have our own servers that we use as our CDN, and in others we use third-party CDNs. When splitting the traffic among multiple CDNs, you need to compare them somehow to optimize the user experience. A single bad decision can result in a lot of unwanted buffering — the antithesis of a good streaming experience.

later.

We aim to send users to the CDN promising the best possible user experience. To do so, we use our own solution that manages multiple CDNs. It's called MultiCDN, and it directs the traffic based on QoE data from CDNs and variables like ISP or location. Based on user experience on each CDN — and sometimes even a specific datacenter of a CDN — the service picks the right one for each particular user based on the specific ISP and country from which the user is streaming.

## MEASURABLE IMPROVEMENTS

Since the deployment of MultiCDN, we've seen improvements across all of the countries we serve. This is mainly down to having good QoE data, and the automatization of host list generation that reacts to collected data in almost real time.

Here's one example of improved buffering rates, from Nigeria:



## HOW WE MEASURE USER EXPERIENCE

On every platform, we collect metrics from the users' players when they are streaming so we can analyse their experience and make changes based on the data we collect.

Some of the metrics we look at:

- Length of viewing session
- Total buffering time
- Length of each buffering event
- Used CDN and datacenter

These metrics are then aggregated and transformed into what we call QoE (Quality of Experience) metrics.

QoE data is used to calculate the rank, or score, of each CDN. CDN rank is calculated for each ISP-country combination we have data from, and it's really not much of a brain teaser. It is a basic math operation on the metrics we collect to get some session-related numbers to represent CDN performance.

These calculations are pre-generated and performed only once per day to save performance while still monitoring possible changes in user experience and CDN performance.



Everything from collecting experience data, to deciding which CDN is best, is automated, so we can react very quickly to a possible change.



## DEALING WITH SPECIAL CASES

Even before the MultiCDN project was launched, we had already studied the performance of certain CDNs in a few countries. Using the old-but-still-valid data, we can streamline problem-solving even before user experience metrics reflect it. This is rarely used since MultiCDN is learning from the collected metrics, but sometimes it can be useful to be ahead of reality.

That's why we created a set of new fields and options in our configuration. Some of those options are kind of breaking the MultiCDN concept, but no solution is perfect, and this extension makes sense considering the specific conditions we have to handle. Here are some main options we had to add.

## Override decision for given country or ISP

Based on our past experience, we know that some countries have only one CDN that fits our needs. In this case, we skip testing the performance of other CDNs, ignore all the metrics and MultiCDN decisions, and just use the hosts we trust.

Like in the previous option, we already know which CDN performs well. But in this case, there are more of them, so we use the exclude option. To focus on the best-performing CDNs, we simply set which CDNs with non-adequate performance we will ignore for a given country or ISP, and use only the rest of the CDNs.

For example, some of our customers are not based in Africa. It doesn't make sense to serve them content from our data centers in Africa as it would only downgrade the quality of the stream. So, we use third party CDNs and restrict the CDNs in Africa.

As explained above, the MultiCDN extension would notice low performance of certain CDNs in case of a specific country-ISP combination. But there's no reason to make our users experience lower quality of the stream than necessary if we have the information already.

## Rank tweaking coefficients

Sometimes, for testing or to workaround a failure not yet reflected in QoE metrics, we need to increase the final ranking of a particular CDN. We created an option to tweak ranks for certain conditions.

This will also help us in the future when we will need to lower the load on a certain CDN.

### THE NECESSARY EVIL IN MULTICDN

Even if we do send the majority of customers to the current best-performing CDNs, we still have to get the metrics for the other CDNs. How do we do it while not affecting user experience? Definitely not in the peak hours when even a very small percentage of our traffic would mean sending several gbps of traffic into suboptimal places. During peak hours, we really want people to sit comfortably and not be worried about buffering.

So we scheduled times outside of peak hours when we send a minimal amount of users to all CDNs on the list, and test the performance. These scheduled testing periods are very short and help us decide what will be the best CDN during peak times.

This is a fragile concept because we need to have enough data to make data-based automatic decisions while disturbing as few users as possible.

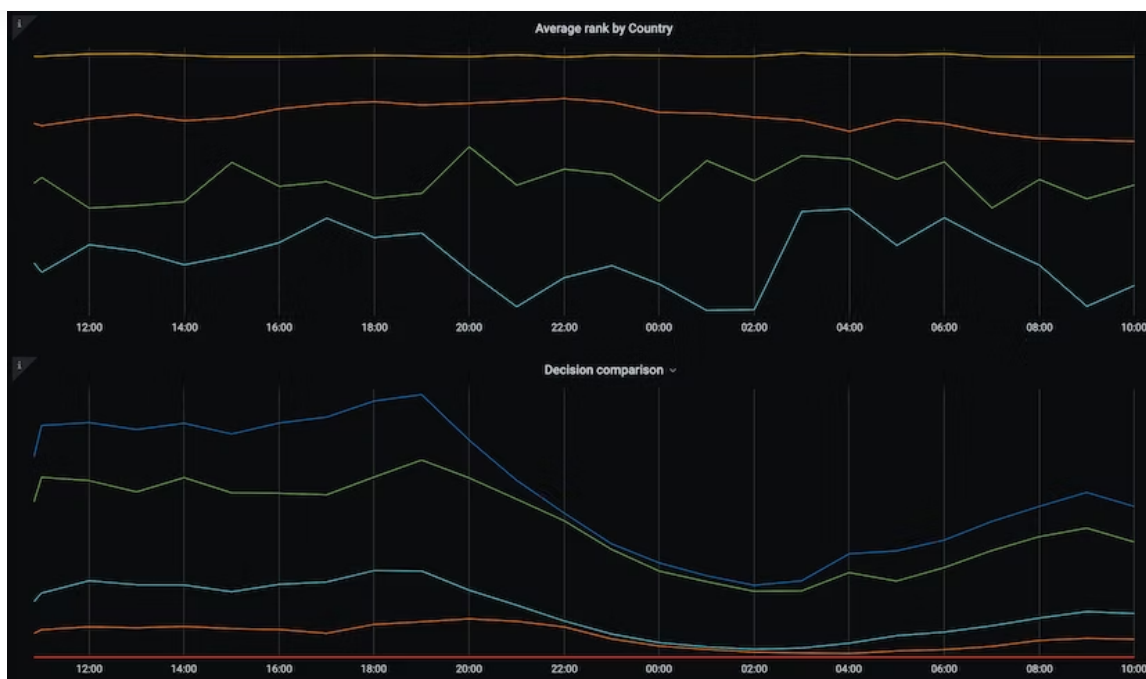
At the end of the day it's just high school statistics.

### MONITORING AND ANALYSING DECISIONS

MultiCDN is a new approach for us. We need to have everything under our thumb and know what is going on in every part of the code. That's why we are keeping all of the decisions, and the conditions the decisions are made under, stored for further tuning of the logic and for overviews of decisions. To make all the raw data in Prometheus more understandable, we created a lot of Grafana dashboards.

On the dashboards, we track the comparison of original decisions we would make at a particular time without MultiCDN to ranks given by MultiCDN. Thanks to this data, we can improve the decisions in the





## HOW YOU DEPLOY THIS GAME CHANGER

The MultiCDN project was huge — we had to touch most of our content-serving components. We collaborated across teams to make informed and effective decisions, foresee the consequences, and streamline the changes.



The most challenging part of the project was, of course, deploying decisions to production. We had to create a backwards compatible solution to avoid putting our current user experience in danger.

We created a lot of switches and toggles to be able to turn the MultiCDN on only for certain conditions and certain content.

At first we deployed MultiCDN to production in passive mode. That means we were only logging the decision that would be made based on the user experience and analysing the data. Thanks to this first step, we fixed some minor bugs and prepared for real deployment. And of course it filled all of our fancy dashboards.



After we saw that everything was working well, we started adding more countries and increasing the percentage. After a while we ended up with 100% decisions made by MultiCDN. This step-by-step process of enabling only a given percentage of assets in selected countries kept the deployment under control.

Now the project is finished successfully, and we have a service that performs data-based automated decisions with only a few overrides and special cases.

## NEXT STEPS

So far, MultiCDN manages video-on-demand streaming. For the next step we want to use MultiCDN to improve user experience of Showmax live streaming, live sporting events, news, and more. It's another great challenge to take on. This time it will be mostly about collecting the QoE data in almost real time, about timely decisions, and prompt response, as the live stream never waits.

Another opportunity is to compare different settings of the same CDN — imagine a user in Nigeria who is streaming from a third party CDN. We have the same data on origins in Falkenstein, Germany, as we do in Johannesburg, South Africa. With MultiCDN, we will always select the setting with an origin that is closer (better) and available.

Stay tuned for the next edition where we will report how it all went!

SHARE ARTICLE VIA:



## You might be interested

### Pytest Appium: iOS/Android Setup BrowserStack and Simulators

#APPS 14 DECEMBER 2023

DAVID ALDORF

### Inspirations from Droidcon Lisbon

#APPS 06 DECEMBER 2023

MICHAL URSINY

### Integration Tests

#APPS 21 NOVEMBER 2023

OLEH PSHENYCHNYI