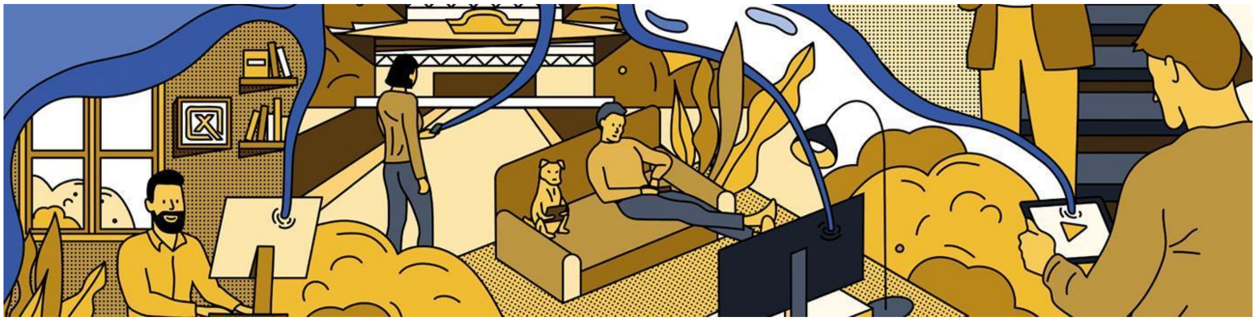


Mission: Emission

Sebastian Manemann, December 2021

“The cheapest and greenest energy is the energy we don’t use”
Jürgen Fischer, Danfoss President of Climate Solutions

Imagine you want to buy a new car. How would you try to reduce your carbon footprint? The obvious answer can only be going electric. It’s the easiest way to do that without letting go off too much comfort.

What sounds like the obvious turns out to be a little more complicated when you really want to calculate the amount of footprint reduction. Where does your electricity come from? Where has your car and more importantly your battery been produced? What physical resources were used and how are they sourced? How much drinkable water has been used ? and, and, and....

When talking about the media streaming industry, reducing our media services footprint is a hot topic. But what is the right approach and how does it really impact the footprint?

Our Ecosystem

The shift from linear to OTT seems to be an unstoppable trend, that got accelerated a lot when covid hit us all. There is research out there that shows an explosion of OTT consumption in 2020 with the same trend in 2021. Since this way of consuming media is here to stay, let's look at the workflow that is applied to produce this content.

To create a single piece of content, many production processes need to be involved. Typically the content gets played out for both, linear and OTT and is then pushed to what we call an “OTT Headend" which transcodes, packages, encrypts and provisions the content towards the CDN.The user will request a manifest to the CDN, which will then deliver the requested manifest and the relevant video/audio chunks, using their infrastructure as well as the local ISP of the user. Although all of those processes could be interesting to look at from an energy perspective, I will focus on the “OTT Headend" part of the Workflow, since this is a part that we carefully analyzed at Quortex.

The traditional OTT headend consists of many processes where only a few are really CPU intensive operations that consume a lot of resources, hence are crucial for possible savings. As we all know, video and audio transcoding is mostly relying on CPU processing. While the processor density and efficiency evolution is slowing down over the last few years, the codec complexity keeps increasing constantly. As good as this is for the User Experience, it’s not helpful when you try to save CPU resources. So since sacrificing Video Quality by choosing less efficient codecs or increasing the used bandwidth per profile do not seem to be valid options, what else is it we can do?

Are we doing Pizza or Video?

Maybe it is worth having second thoughts about the concepts we apply to our workflows. How do other industries deal with rising or varying demands? Imagine you run a pizza place: Would you prepare every pizza and wait for people to order them? No one fancy a cold tuna slice, so that place would not last long I fear.

Translating this Just-in-Time paradigm to our Industry, should make us reconsider the way we work. If we can reverse the current push mode and use a pull mode instead, we could transcode, package and encrypt etc. on demand. Furthermore we would not need to push all of our content to the edge, regardless if consumed or not. That would enable us to use significantly less resources for either of those processes.

Buzzwords for 100 - the Cloud!

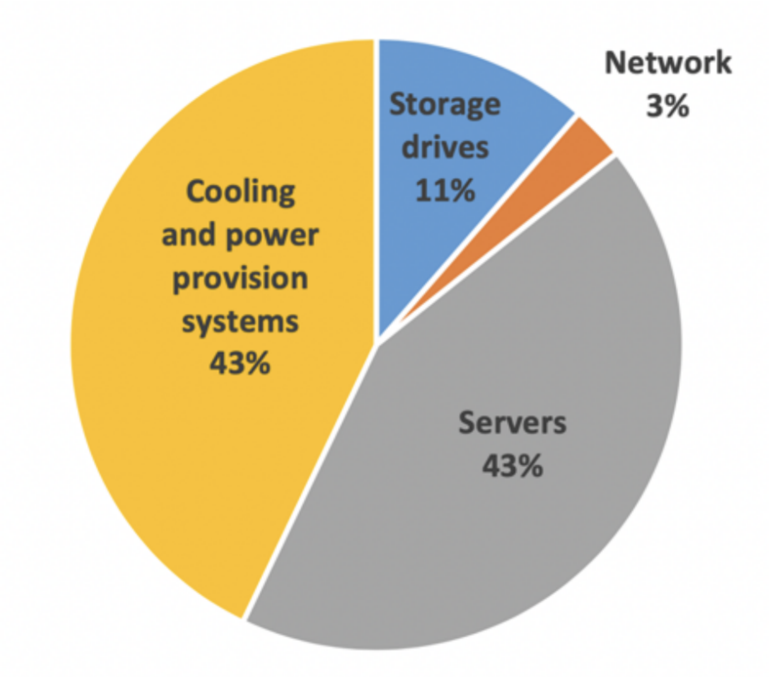
The whole concept of being a cloud provider is based on maximizing the resource utilization. There is a natural interest in having all available resources operating at maximum efficiency 24/7. This might sound as the complete opposite of energy savings, but this pursuit of efficiency has led to cloud providers building in custom cooling systems, automated scaling of physical resources etc. That way the total energy consumption of big data centers has been more or less constant over the last decade, although the demand for computing was and is increasing a lot . This also means that moving your workloads from private datacenters, that are certainly not as efficient, to the public cloud is most likely already decreasing your footprint significantly.

Furthermore cloud providers offer a variety of services like serverless functionality, elastic kubernetes services and even a spot market that evolved around selling unused resources for short time periods at a highly discounted price. While those services require the software layer we want to put on top to deal with specific conditions, they can create a whole new level of elasticity and scalability. It is just about finding the right application, we need to find something really “Cloud Native”.

And since in the cloud consuming less resources is equal to consuming less credit card at the end of the month, we find ourselves in a win-win situation.

What to measure? Watts? Megahertz? Tablespoons?

Considering we have found what we look for and start moving our workloads to the cloud, we want to find out about our carbon footprint in this environment. Since the majority of them are committed to using a significant amount of renewable energy in the next 5-10 Years, typically cloud providers are not providing exact figures on their resource consumption. Here we are back to our car example from the beginning, we just can’t find all the data points to measure. There are a few studies and reports that indicate some figures on how much energy data centers consume and how the energy consumption within a datacenter is distributed. While the total amount of energy consumed might not be as relevant for our calculations, the split is highly interesting.

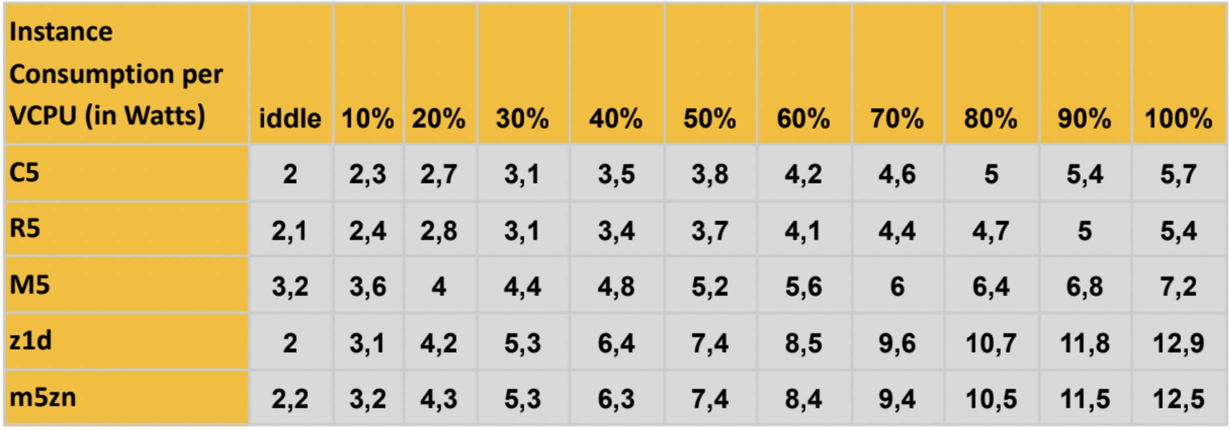


Source: https://energyinnovation.org/2020/03/17/how-much-energy-do-data-centers-really-use/

What we can see here is that servers and cooling systems account for the majority of the energy consumed. What we can do now is calculate the difference between the pull and push workflow. We can draft some values by just comparing the used vCPUs on a given instance type and the power consumption attached to that. Since cooling accounts for a similar percentage, we can just double the result. It will not take into account cooling, water waste and chipset acquisition, but it will identify a clear trend on possible savings.

There is a very good post on medium describing how the power consumption of an EC2 instance on AWS can be calculated, taking into account specific AWS CPU Models and the Power distribution in a physical Server, which you can find here: https://medium.com/teads-engineering/estimating-aws-ec2-instances-power-consumption-c9745e347959

For our calculations, we use the below AWS figures which are similar with Azure, GCP and other cloud providers.

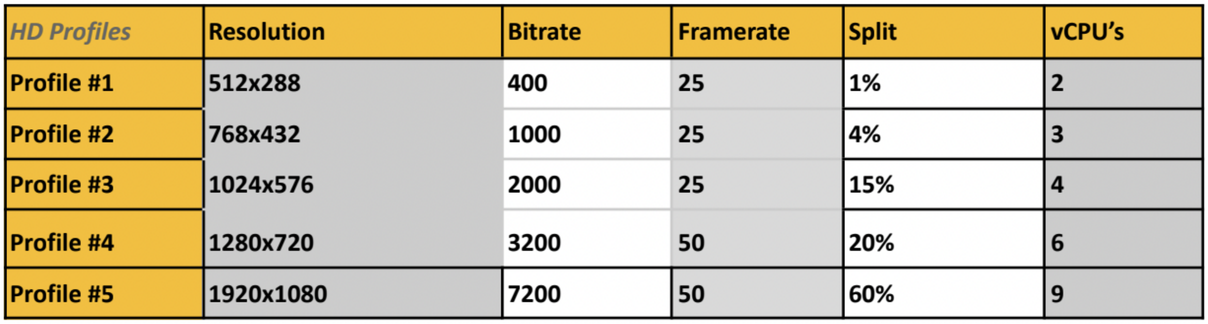


Instance Consumption per VCPU (in Watts)	iddle	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
C5	2	2,3	2,7	3,1	3,5	3,8	4,2	4,6	5	5,4	5,7
R5	2,1	2,4	2,8	3,1	3,4	3,7	4,1	4,4	4,7	5	5,4
M5	3,2	3,6	4	4,4	4,8	5,2	5,6	6	6,4	6,8	7,2
z1d	2	3,1	4,2	5,3	6,4	7,4	8,5	9,6	10,7	11,8	12,9
m5zn	2,2	3,2	4,3	5,3	6,3	7,4	8,4	9,4	10,5	11,5	12,5

Source :Estimating AWS EC2 Instances Power Consumption | by Benjamin DAVY | Teads Engineering

What we can see from the table above is that for a C5 instance the power consumption per vCPU is around 5,7 Watt if fully utilized. For this calculation, I'll assume that video transcoding stresses the CPU to a level of 80%. Let's consider we run 50 24/7 HD channels with 5 profiles each on our OTT platform.

To complete the pull mode scenarios, we need to also invoke a number of users. Let's assume two use cases : a) 1000 and b) 10000 subscribers. Our bitrate ladder looks like this and we have an audience split factor between the profiles.



HD Profiles	Resolution	Bitrate	Framerate	Split	vCPU's
Profile #1	512x288	400	25	1%	2
Profile #2	768x432	1000	25	4%	3
Profile #3	1024x576	2000	25	15%	4
Profile #4	1280x720	3200	50	20%	6
Profile #5	1920x1080	7200	50	60%	9

To assess the total amount of vCPUs needed I used our Open Source cloudbench tool, which is mostly based on ffmpeg for the transcoding part. To process all the profiles, roughly 24 vCPUs are required on a c5 instance.

Hence, our power consumption for the push workflow will look like this: 24vCPU*5W=120W of power for 1 channel/ 120W*50=6000W of power for 50 channels / 6000W*730h=4380 kW.h of energy per month for 50 channels/ 4380*2 (cooling etc)=8760 kW.h with a total of 864000vCPU per month

To put this into perspective: the house I live in consumes roughly 500 kW.h per month, with 5 people on board (including 3 kids that refuse to turn off lights whatsoever). Having that said, our OTT Service will consume energy for 17,28 houses and there are still a lot of items left out of the equation. This consumption will be steady, regardless if there is 0 or 10 000 viewers pulling any profile.

If we calculate the CO2 emission generated by that level of energy, taking the German “energy mix” which includes 40% renewable energy, we get 4095 kg of CO2, per month.

Now, to get some figures for pull mode, we need to identify the audience curve first to figure out what profiles would be pulled at what time. The below curve is based on actual consumption data for an OTT Service that is running for years now.

Once we have the curve, we can put some needed CPU Cycles per timeslot, taking into account the total amount of audience. I’ll spare you 600 rows of excel calculations and just tell you that we end up with a total amount of a) 444vCPUs/Day/1000 Viewers and b) 15075vCPUs/Day/10 0000 Viewers for the 50 HD channels. With that in mind, our totals look like:

Use Case a) 1000 subscribers

444*5= 2219 Watt / 50Channels = 2,219 kW.h / 2,219*30*2 = 133,14 kW.h per month for 50 channels with a total of 13314 vCPUs used.

Use Case b) 10 000 subscribers

15075*5= 75375 Watt / 50 Channels = 75,37 kW.h / 75,37*30*2 = 4521,6 kW.h per month for 50 channels with a total of 452250 vCPUs used.

What we can see is that the pull mode changes the resource utilization drastically. While Use Case A saves a minimum of 8500 kW.h per Month ! (remember, that’s 17 houses...), use case B still saves around 47%, which is significant. This offset is obviously depending on the amount of channels vs the amount of concurrent users. From the projects we do, we can see that normally both are increasing more or less in parallel curves.

All of that savings, financially and in power consumption, are possible by “just” changing your workflow paradigm from push to pull, there are no downsides towards customer experience or service availability. The opposite is the case: by changing to pull mode, there is a whole new level of flexibility that comes with that, like audience aware encoding or rule based delivery. And if we start thinking about utilizing Spot Instances rather than regular ec2, there are even more options to reduce your cloud bill drastically while keeping the power consumption to a minimum.

To make all of that happen, you just need to use the right application - and we at Quortex can help you with that!