

Sign in with Google

Use your Google Account to sign in to Medium

No more passwords to remember. Signing in is fast, simple and secure.



Continue



Benjamin DAVY

Sep 23, 2021

16 min read

Listen

Building an AWS EC2 Carbon Emissions Dataset

Date	Instance ID	Instance Type	Region	Availability Zone	State	Usage Hours	Usage Bytes	Usage Bytes/Second	Usage Bytes/Second Squared	Usage Bytes/Second Cubed	Usage Bytes/Second to the Power of 4	Usage Bytes/Second to the Power of 5	Usage Bytes/Second to the Power of 6	Usage Bytes/Second to the Power of 7	Usage Bytes/Second to the Power of 8	Usage Bytes/Second to the Power of 9	Usage Bytes/Second to the Power of 10
January 2019	i-0123456789	m5.xlarge	us-east-1	us-east-1a	Running	1000	1000000000000	1000000000000000	1000000000000000000	1000000000000000000000	1000000000000000000000000	1000000000000000000000000000	1000000000000000000000000000000	1000000000000000000000000000000000	1000000000000000000000000000000000000	1000000000000000000000000000000000000000	100

In a previous article we detailed a bottom-up methodology to estimate the power consumption of AWS EC2 instances. This time, we are sharing a dataset containing an estimation of all instances' carbon footprint, related to

Benjamin DAVY
91 Followers

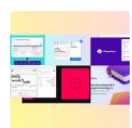
Sharing what's happening @Teads Innovation team

Follow

More from Medium

ibrahim z...

8 Must Have Google Chrome...



Gagan G...

AWS VPC Connectivity Options...



Ravishan...

Building Modern Data Lakes...



Ankit Sar...



both manufacturing and using the servers.


Our goal is to be able to estimate the carbon impact of services relying on EC2 hardware. In the first part of this article we cover two additional steps we took following our initial tests:

- A study on **power measurement variations** observed on similar machines (Chapter 1)
- **On-premise experiments** that were performed to validate our assumptions (Chapter 2)

We then generalize our results to all available EC2 instances, even if we were not able to actually measure them. For this, we had to gather EC2 hardware specifications and define a way to **estimate their power consumption profiles** (Chapter 3).

Finally, we quickly cover how we convert power consumption into

carbon emissions ([Chapter 4](#)) and close the article with a naive proposal for **estimating embodied emissions for hyperscale server hardware** ([Chapter 5](#)).

*If you are curious, you can look directly at the **simple estimator**  we've put together to play with the results. The dataset is also available as a spreadsheet which you can duplicate.*

Foreword on limitations

This is a work in progress initiative and our estimation only covers EC2 server hardware. Here is a simplified overview of a Data Center:



Simplified Data Center Service — Icons from [Freepik](#) and [Icons8](#)

We can see that there are a lot of moving parts around our workloads that should be included in a proper assessment:

- **The Data Center facility as well as Storage and Network equipment** running alongside our EC2 machines. We will want to cover this later.
- **AWS infrastructure running the cloud service:** When we are launching an EC2 instance we are indirectly leveraging other resources that are not reported on our bills (logging and monitoring, etc.).
- **Emissions from unused resources:** We consider our billing reports to reflect our usage. However, the hardware we are using might not always be fully allocated. Our methodology doesn't take into account situations where we would be the only tenant on a physical machine and only "use" a limited portion of its resources. In that case, the rest of the resources would be idle, consuming power but not accounted for in our simple model. An option would be to adapt our tests to run on a portion of a bare metal to observe different allocation factors (between 50% and 70% for

hyperscalers according to unofficial sources).


As we can see, there's room for improvement but we think that this first step can still be useful to better grasp the physical reality of cloud infrastructure. Of course, any feedback is more than welcome.

1 — Power measurement variations observed on similar machines

Since our initial study, we have performed additional tests to make sure our measurements were consistent. The methodology and our assumptions are detailed in our [previous article](#). As a reminder, we have packaged a tool called [turbostress](#) that performs several stress tests to simulate different workloads and report power consumption measures using [Intel RAPL](#).

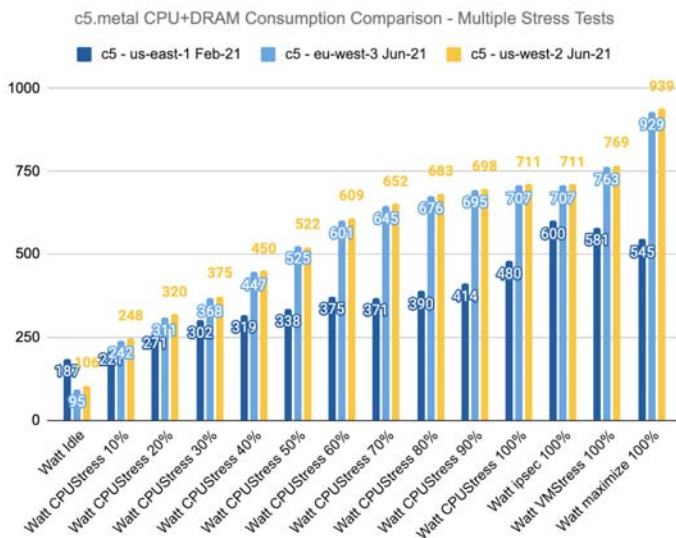
We performed this on available **Intel-based bare metal instances**. Overall,

we have been able to assess the following instances: c5, m5, r5, m5zn, z1d, i3, c5n.

 Raw turbostress exports used in this article can be found in this repository.


We were particularly curious about whether or not we could observe variations from one instance to another. This led us to perform multiple tests and **we found significant differences** with some of our early measurements, namely for the c5, m5, and r5.

Here we compare three different tests on c5.metal instances, performed in three different regions between February (initial experiments) and June 2021.



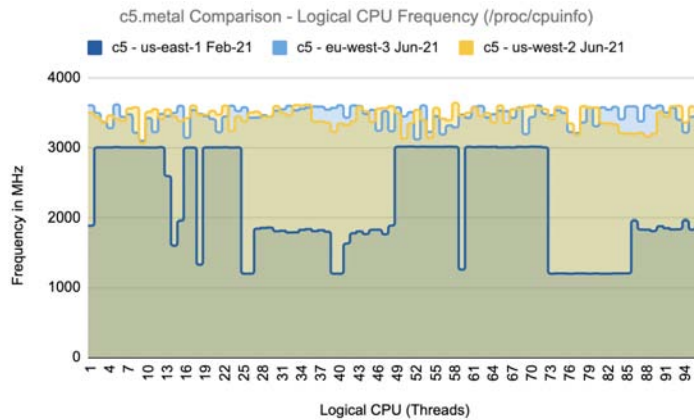
Initial measurement from February '21 in dark blue, new measurements from June '21 show similar figures

Apart from a lower idle consumption, we can see an **important increase in the reported numbers** on our more recent tests. This is concerning, but at least our two new tests are consistent.

Luckily, turbostress outputs the CPU information (/proc/cpuinfo) on top of the power measurements. By looking at this, we were able to identify the most likely culprit for these discrepancies: the **CPU frequency** —  rookie mistake.

Here is a comparison of the reported frequency for each of the 96 logical CPUs (threads). We can see that the

first machine, in dark blue, isn't running at its full capacity. The two others show a comparable "clocking profile," which is reassuring.



We suppose that this is linked to DVFS techniques that can be used to dynamically scale voltage and frequency at the CPU core level for energy-saving purposes.

The following graph shows the average thread frequency compared to the base and max frequency of the CPU model. We used this to **identify measurements performed on underclocked machines** and only keep the others for our study.



We consider our measurements to be valid when the reported frequency sits between base and max values. Fortunately, our measurements on the m5zn, z1d, i3, and c5n were apparently ok.

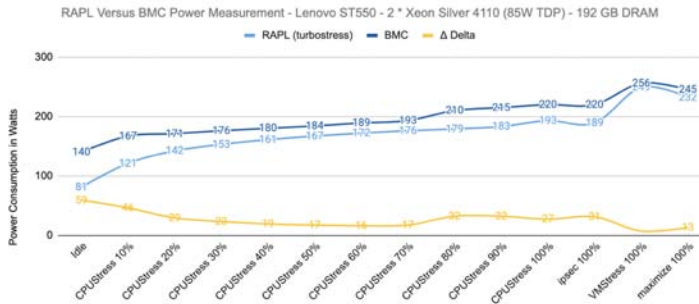
2 — On-premise experiments to validate our assumptions

One of the limitations of RAPL measurements is that we are **only capturing CPU and DRAM power consumption**. Also, this is a software measure and we wanted to know how this would compare with on-premise power readings.

With the help of Workflowers and Hardbricks we were able to run our turbostress protocol on-premise and compare the results with BMC power consumption readings. This should be closer to reality even though it's

still not a perfect measurement using a power analyzer.

Here is a comparison of the two measurements and the difference (Δ in yellow):



This first result seems to confirm that RAPL readings are consistent with BMC readings. The Δ should be a good enough proxy to estimate the consumption related to the rest of the machine, namely:

- fans,
- storage drives,
- network cards,
- etc.

Further tests would ideally be required to definitely confirm this

assumption. Contact us if you want to help 🙋.

3 — Building a dataset for all existing instances

We are only able to perform our test on bare metal instances and these are not available for all instance types. Also, we are not able to perform the same test on ARM-based architectures even though we have access to bare metal options.

In order to build a dataset covering all available instances, we assumed that all instances ultimately rely on a limited number of hardware platforms containing:

- **CPUs:** from 1 to 8 sockets equipped with x86 (Intel and AMD) or ARM chips (AWS Graviton). In most cases, a given instance will rely on one CPU configuration, and when there are two possible versions we choose the one we analyzed.
- **DRAM:** up to 24 TB for SAP HANA

instances

- **Local Storage** (*optional*): SSD (NVMe for recent instances) or HDD type with a varying number of drives
- **GPUs** (*optional*): mainly from Nvidia
- **Other parts** like FPGAs or custom silicon (Nitro and Inferentia cards)

Here is an extract focusing on the CPU platforms available on EC2 as of June 2021. The information comes from AWS' public documentation or was collected via `/proc/cpuinfo` and turbostat:

GPU Name	Platform Number of CPU Socket(s)	Total Number of vCPU per socket	Platform Total Number of vCPU	CPU TDP L1 (in Watts)
Xeon E5-2666 v3	2	20	40	135
Xeon E5-2676 v3	2	24	48	120
Xeon E5-2686 v4	2	36	72	145
Xeon E5-2650	2	16	32	95
Xeon E5-2665	2	16	32	115
Xeon E5-2670	2	16	32	115
Xeon E5-2651 v2	2	24	48	95
Xeon E5-2670 v2	2	20	40	115
Xeon E5-2680 v2	2	20	40	115
Xeon E7-8880 v3	4	32	128	150
Xeon Platinum 8124M	2	36	72	240
Xeon Platinum 8151	2	24	48	240
Xeon Platinum 8175M	2	48	96	240
Xeon Platinum 8176M	8	56	448	165
Xeon Platinum 8252C	2	24	48	240
Xeon Platinum 8259CL	2	48	96	210
Xeon Platinum 8275CL	2	48	96	240
Xeon Platinum 8375C	2	64	128	300
EPYC 7571	2	48	96	200*
EPYC 7R32	1	96	96	280*
Graviton	1	16	16	40*
Graviton2	1	64	64	150*
Core i7-8700B	1	12	12	65

Most of these CPUs are custom-made for AWS, so some of the specifications are guessed (*in italic**). This “CPU platforms” dataset is useful for both power consumption and embodied emission estimations (see chapter 5).

Regarding the power consumption of the instances we have decided to only keep four load levels for practical reasons:

- idle
- 10% load
- 50% load
- 100% load

3.1 — Deriving a power consumption profile from turbostress results

Thanks to our measurements we have enough data to build consumption profiles for the CPU and DRAM domains. Here is how we defined these values :

- For the Idle and 10% load levels we simply keep the CPU and

DRAM values reported by RAPL

- At 50% load, we use the reported RAPL value for the 50% load CPU stress test. For the DRAM we had to cheat and calculate the hypothetical middle point between low and high DRAM workloads as we are not able to specify a DRAM load in our experiment.
- For the 100% load level, we take the average load for the four CPU-intensive stress tests to cover a wider range of workloads. For the DRAM we keep the value from the most power-hungry test.

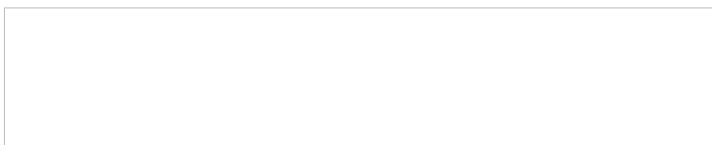
In order to generalize this to all platforms, we tried to find the closest hardware and estimated the consumption based on the TDP and a simple rule of thumb.

For example, we use the measurements performed on the m5.metal equipped with a Xeon Platinum 8175M (240W TDP) to derive the power consumption profile of the Xeon Platinum 8176M (165W TDP)

used in the high memory instances.
This is detailed in the dataset.

For the **AMD and ARM CPUs** we rely on the information we found online and make some assumptions. For example, Graviton 2 ARM processors are based on the Neoverse N1 platform and ARM indicates a 150W TDP for a 64 core CPU for hyperscale data centers.

For these CPUs, we calculated a simple average of our previous measurements based on the watts consumed per watt TDP (L1). Here is the result for each load level as of writing this article:



For a CPU with a 100W TDP , we will consider that the idle consumption will be $100 * 0,12 = 12$ Watts . At full capacity, we are basically considering the TDP value as the

actual power consumption.

3.2 — Estimating the power consumption of the other components

For GPUs, we use the TDP reported by the manufacturers and the same table as for CPUs. We could later revise this by doing some proper measurements using [nvidia-smi](#).

For example, the consumption of a Tesla V100 GPU that has a TDP of 300W will be estimated at $300 * 0,75 = 226$ Watts for an average workload (50% load level) using the above table.

The rest of a commodity server can include other components such as fans, storage drives, network cards, and other parts we couldn't measure or include in the previously listed estimations.

In order to cover for the consumption of all these “other” elements we took a shortcut and

defined a constant value based on the CPU TDP. We defined this according to our on-premise test (from Chapter 2) and other available data.

On the Lenovo ST550 machine see earlier, the average difference between RAPL and BMC power consumption readings corresponds to **~15% of the CPU configuration TDP** ($2 \times 85W$). Using the same approach we tested our simple “model” with the data provided by Dell for the PowerEdge R740.

In a detailed Life Cycle Assessment, the manufacturer communicates the consumption profile of the machine according to the same four load levels. Comparing both values we obtain an average difference corresponding to **~13% of the CPU configuration TDP**.

For now, we have defined a simple heuristic that is equal to **20% of the CPU(s) TDP**. We consider that this value should cover the “other”

components' power consumption.

We know that **this is far from being rigorous**, especially if we include exotic server configurations.

However, we think this is still better than considering this consumption negligible and should be a good starting point for commodity servers.

3.3 — Estimating the power consumption at the instance level

As discussed in our previous article, we consider that **bare-metal resources are cut into instances in a linear fashion**.

In this [great talk](#) from re:Invent 2017, Adam Boeglin describes how c5 instances are sized. In his example, the c5.18.xlarge instance is the equivalent of two c5.9.xlarge instances and the CPU to memory ratio stays the same across all sizes.

In our dataset, we apply a vCPU ratio

$$\frac{\text{Instance number of vCPUs}}{\text{Bare metal number of vCPUs}}$$

to split our bare

metal estimation to the instance level.

4 — Converting power consumption into carbon emissions

This part will be brief!

In order to convert our power consumption into carbon emissions, we simply apply the electricity carbon emission factor for each data center geolocation. The CloudCarbonFootprint team has already done this for AWS.

We also included the Power Usage Effectiveness (PUE) in our estimation. AWS communicates that, according to internal numbers, all their data centers have a PUE under 1.2 .

We decided to stick to that number.

5 — Estimating embodied emissions for EC2 hardware

Most available initiatives are focusing on emissions generated from the use

phase, also called “Scope 2” in carbon accounting (from the cloud provider’s point of view). We are deeply convinced that **we cannot build a proper sustainability strategy by only focussing on Scope 2 emissions.**

Thankfully, this is something that is slowly getting more and more attention.

One of the most recent examples is the study from Udit Gupta et al., [Chasing Carbon: The Elusive Environmental Footprint of Computing](#), where a research team from Harvard & Facebook indicates that:

“most emissions related to modern mobile and data-center equipment come from hardware manufacturing and infrastructure.”

Now, let’s have a quick overview of this whole new kingdom of uncertainties and approximations. First, the **state of the art on**

manufacturing carbon emissions is quite limited for IT equipment.

We can however list some interesting references:

- Life-Cycle Assessment of Semiconductors by Sarah B. Boyd, a spot-on work, indicating that dense electronic parts should account for most of the emissions. Too bad it's dating back from 2011 and hasn't been updated since.
- Dell's collection of PowerEdge carbon product footprint documents, detailing the carbon emissions from the main life cycle phases of their servers.
- The Dell PowerEdge R740 Full Life Cycle Assessment, a detailed and multi-factor impact analysis. To date, the only public document of this kind.

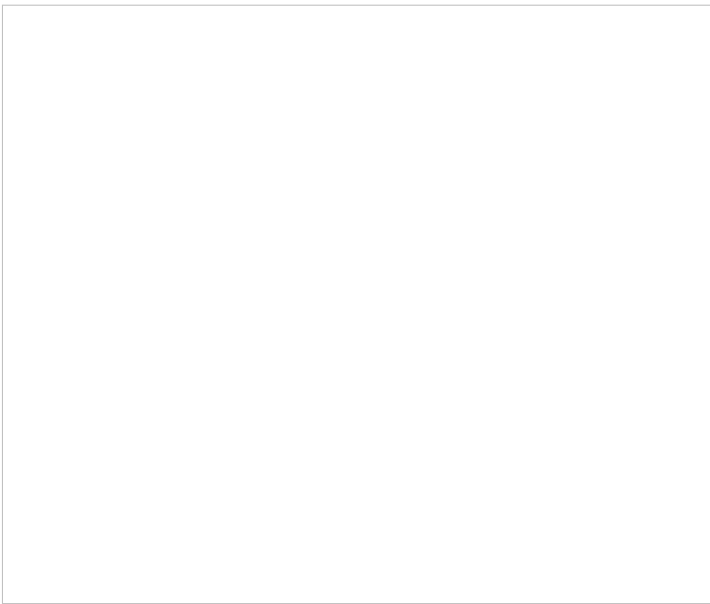
*Here we will only skim the surface. A proper state of the art on IT hardware embodied emissions is ongoing in collaboration with the **Boavizta** initiative. Stay tuned!*

Update January 2022, see this publication from Boavizta — [How to evaluate server manufacturing footprint, beyond greenhouse gas emissions?](#)

5.1 — Carbon product footprint comparison

If we compare Dell’s product carbon footprint data we have a hard time identifying characteristics that could be used as a proxy to estimate the embodied carbon emissions for EC2 hardware.

Here is a table listing the known specifications of the machines and their reported manufacturing carbon footprint:



Data and sources are viewable in the shared documents

We can observe a few interesting things in the first wave of early 2019 reports:

- The carbon footprint **values for the manufacturing phase are relatively close**, from 1141 to 1782 kgCO₂eq .
- The reported manufacturing **emissions cannot be correlated to the weight of the machine**, which could confirm that most of the carbon footprint is linked to semiconductors rather than chassis or other heavy parts.
- Most of the machines are equipped with a **low amount of**

memory and very few of them are equipped with SSDs.

- The number of **CPUs doesn't seem to have a massive impact** on the overall footprint. Again, this is true for this dataset but we shouldn't generalize based on this only.

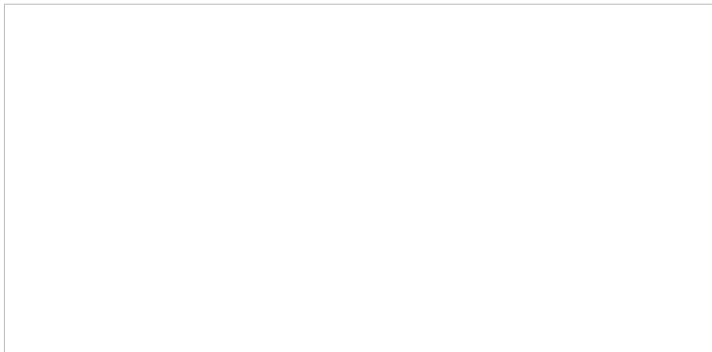
Now, if we have a look at the second wave of reports from early 2021, it's even harder to draw some conclusions:

- This time, all machines are equipped with SSDs but they feature an **even lower amount of memory**.
- Their manufacturing footprint is significantly lower at around ~ 750 kgCO₂eq .

If we assume the methodology to be the same for all these reports it could mean that Dell has greatly improved its supply chain. In any case, **we are missing a detailed analysis by component** so that we could determine whether or not there are

specific parts driving most of the carbon impacts.

One of the only relevant resources available in this area is the aforementioned Life Cycle Assessment performed on the Dell R740 machine (also from 2019). Here is the detailed manufacturing carbon footprint by component for this machine:



Data and sources are viewable in the shared documents

At first glance, we can see that the eight high-volume SSDs have an important impact. However, this configuration seems quite specific and we don't have many equivalents on EC2, except maybe the i3en.

What's also interesting is that **DRAM**

is the second most important driver for manufacturing emissions in this analysis. As mentioned in the study:

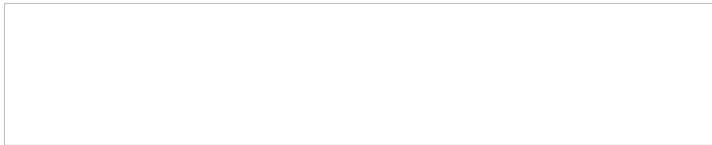
“The twelve 32GB RAM bars used within the configuration account for around 33% of the total mass of the mixed PWB [but] they account for over 90% of the total GWP impact of the PWB Mixed due to their high capacity per RAM bar and the associated complexity and density of the built-in chips and dies.”

Dell also published a product carbon footprint fact sheet for the R740 so we can see if it matches the Life Cycle Assessment (LCA) data. The specifications for the two machines are not similar so we need to adapt a few things on the R740 from the LCA to fall back to a comparable configuration:

- Remove the 8 high-volume SSDs
- Only keep one 32 GB DIMM of DRAM
- Add a couple of storage drives.

The R740 configuration from the fact sheet has three HDDs so we simply multiply the reported manufacturing footprint given for one SSD by three on the other R740, even though SSDs and HDDs must have different manufacturing footprints.

Here is a table comparing the manufacturing impact of the two R740 configurations depending on the source:



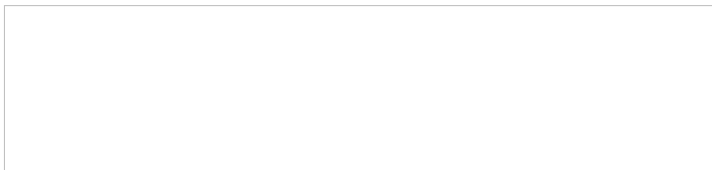
Calculations are available in the shared documents

This result is quite disturbing. While we are not exactly comparing two identical machines **we obtain a drastically different value: 1313 kgCO₂eq versus 550 kgCO₂eq**. It could suggest that these two analyses were not performed using the same methodology and/or using the same emission factors.

5.2 — A naïve proposal, while waiting for a better solution

Failing to find an ideal model to estimate the manufacturing emissions for EC2 hardware, we decided to settle with some arbitrary values:

- We define a **hypothetical minimal rack server** (1 CPU, 16 GB Memory) as a baseline and assume it has a manufacturing carbon footprint of 1000 kgCO₂eq . Here we include AWS' Nitro cards.
- In addition to this baseline, we set **arbitrary carbon footprint values for the main additional parts** we can have in EC2 instances: storage drives, GPUs, additional DRAM (> 16 GB), and CPUs. Here is the summary of these values:



Using these “**Embodied Emission Factors**” we are able to adapt our estimations based on the bare metal

specifications. Once again, we are aware of how limited this approach is:

- If we take the a1 instance family as an example, the 1 tonCO₂eq baseline **could be overestimated** but we have no data for ARM-based servers.
- We use the number of storage drives rather than the storage volume due to the important variations observed in carbon assessments between SSD generations. By doing this, we are **considering any drive from the same generation to have the same embodied emissions** even if they might differ significantly. For example, in 2019 we have instances that were launched with 7.500 GB NVMe SSDs (i3en) and others with 900 GB NVMe SSDs (m5d). Going further would require considering both the volume and release date for more precise estimation. Taking into account the hardware “generation” could be relevant for all parts as well.
- We **did not find any data about**

the manufacturing carbon footprint of modern GPUs. We defined an arbitrary value assuming it would be somewhat comparable to CPU and DRAM manufacturing. This is far from being satisfactory, notably because we are using the same emission factor for all GPUs.

- We did not include exotic parts like FPGAs or specific custom cards (Inferentia) in our estimation.

5.3 — Challenge of distributing embodied emission over time

We now have a value to estimate emissions from the manufacturing phase. However, **applying it to our usage report isn't straightforward.**

The lifespan of a server is set at 4 years in Dell's assessments. We took an easy approach and considered that we can linearly spread embodied emissions. We simply dividing them by the number of hours in a 4 year period to get an hourly rate.

This has **several limitations**. We regularly use instances that were introduced more than 4 years ago and we assume that AWS doesn't install older generation instances when a new one is launched. In that case, should we reward the use of old hardware by lowering their embodied emission factor? For now, we are using the release date information to build a qualitative KPI and observe how "old" our infrastructure is.

Looking at these questions we also identified that the way we distribute embodied emissions can also drive misleading optimization strategies. Let's see.

5.4 — Why carbon footprint is not enough

Here, it's important to remember that **the manufacturing of computing systems has a wider environmental impact** compared to the use phase.

Udit Gupta et al. have pointed out the limitations of only focusing on

carbon emissions:

“environmental impact of computing systems is multifaceted, spanning water consumption as well as use of other natural resources, including aluminum, cobalt, copper, glass, gold, tin, lithium, zinc, and plastic.”

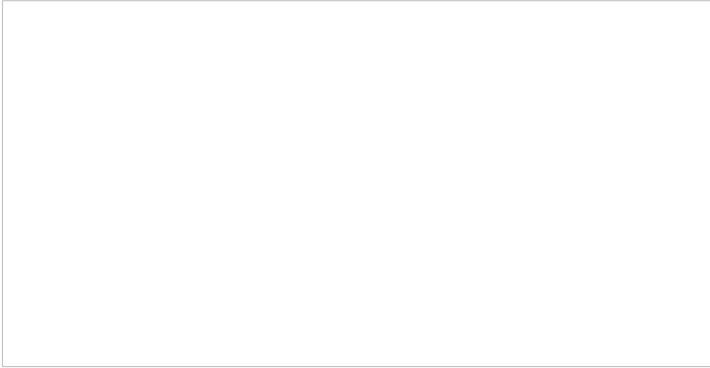
By only looking at a “carbon KPI,” we could be tempted to regularly move our workloads to newer and more energy-efficient instances and somewhat **neglect the other impacts** involved with manufacturing this new hardware.

On that point, we have no preconceived ideas about what would be the less impactful tradeoff. This is where having more multi-factor assessments would be handy.

What now?

All data and sources can be found in a [spreadsheet](#). A simple [estimator](#) page is available as well to play with

the dataset.



EC2 Carbon Footprint Estimator on [Teads Engineering](#)

We hope this work will prove to be useful. We have pushed the bottom-up approach as far as we could, at least on the EC2 part. While this data will serve us to create new KPIs to monitor our cloud platform, we are expecting providers to release more and more data to fuel sustainability initiatives.

Google is now showcasing data centers with the “[Lowest CO2](#)” in their GCP console to guide infrastructure location strategies. This is a good start, although it only covers the impact from the electricity used to run the data center and we’ve seen that it’s much more complex

than that.

There is a great **research opportunity in assessing the tricky cost/performance/impact trade-off** we have to deal with when new hardware is released.

Ideally, we could expect providers to incentivize their clients with impact-aware schemes in the future or, at least, **provide granular reports that include the whole lifecycle.**

 *We would like to thank the community working on this challenge, especially Cloud Carbon Footprint, David Mytton and Boavizta, with a special acknowledgement to Workflowers and Hardbricks for their help in testing our results. Thanks also to Caroline Agase for reviewing the article.*

Resources

- The turbostress tool and raw results used in this study

- Our dataset including calculations in a [spreadsheet](#) (under a [Creative Commons Attribution 4.0 International License](#))
- The [Teads Engineering Sustainability initiative](#) page

Bibliography

- [Life-Cycle Assessment of Semiconductors](#) — Sarah B. Boyd — 2011
- [Chasing Carbon: The Elusive Environmental Footprint of Computing](#) — Udit Gupta et al. — Harvard, Facebook — March 2021



200



3