GREEN CODING IN GREEN CLOUD

deep dive into web measurement techniques and rebound effects

GREEN CODING;

Agenda

1. Understanding where CO2 is emitted

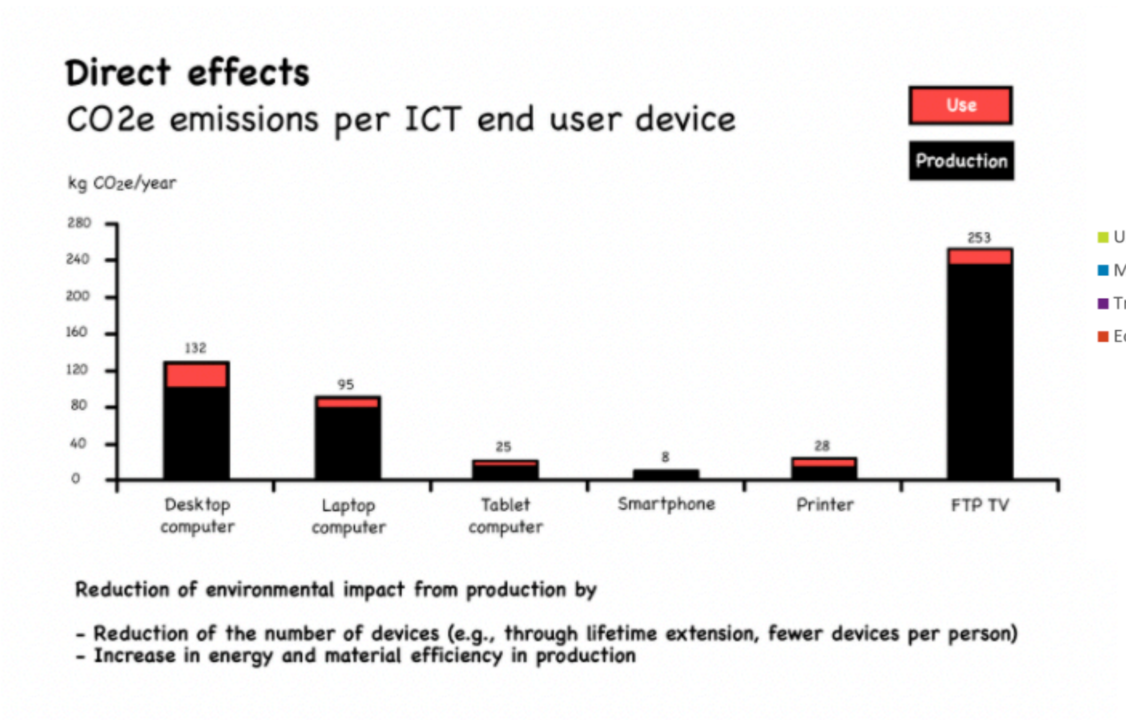
2. Measurement overview & getting to kWh / J / CO2

3. Overview of Open Source Tools

4. Examples & Watch out for rebound effects

Understanding where CO2 is emitted - CO2 comes from producing hardware

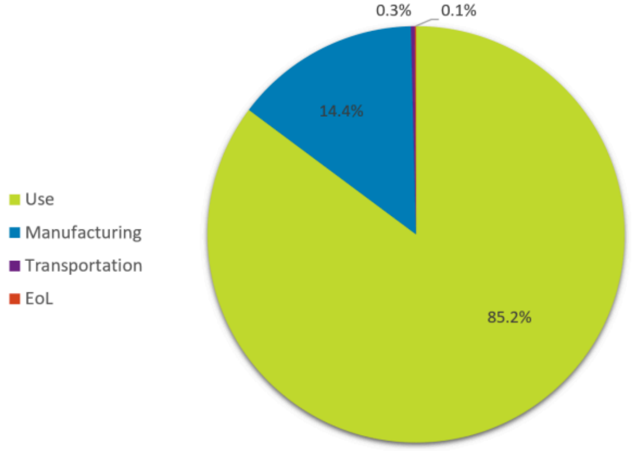
Client Side [1]



Reduction of environmental impact from production by

- Reduction of the number of devices (e.g., through lifetime extension, fewer devices per person)
- Increase in energy and material efficiency in production

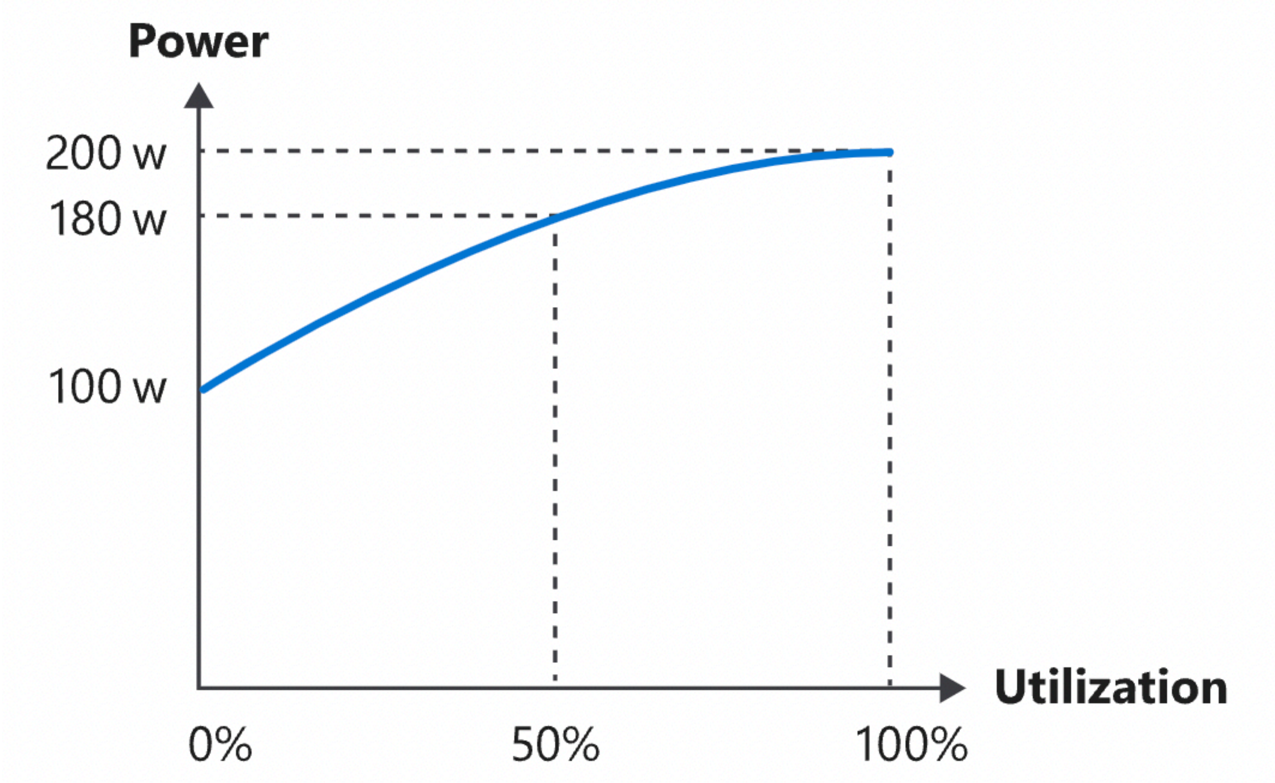
Server Side [2]



Est. product carbon footprint, page 1

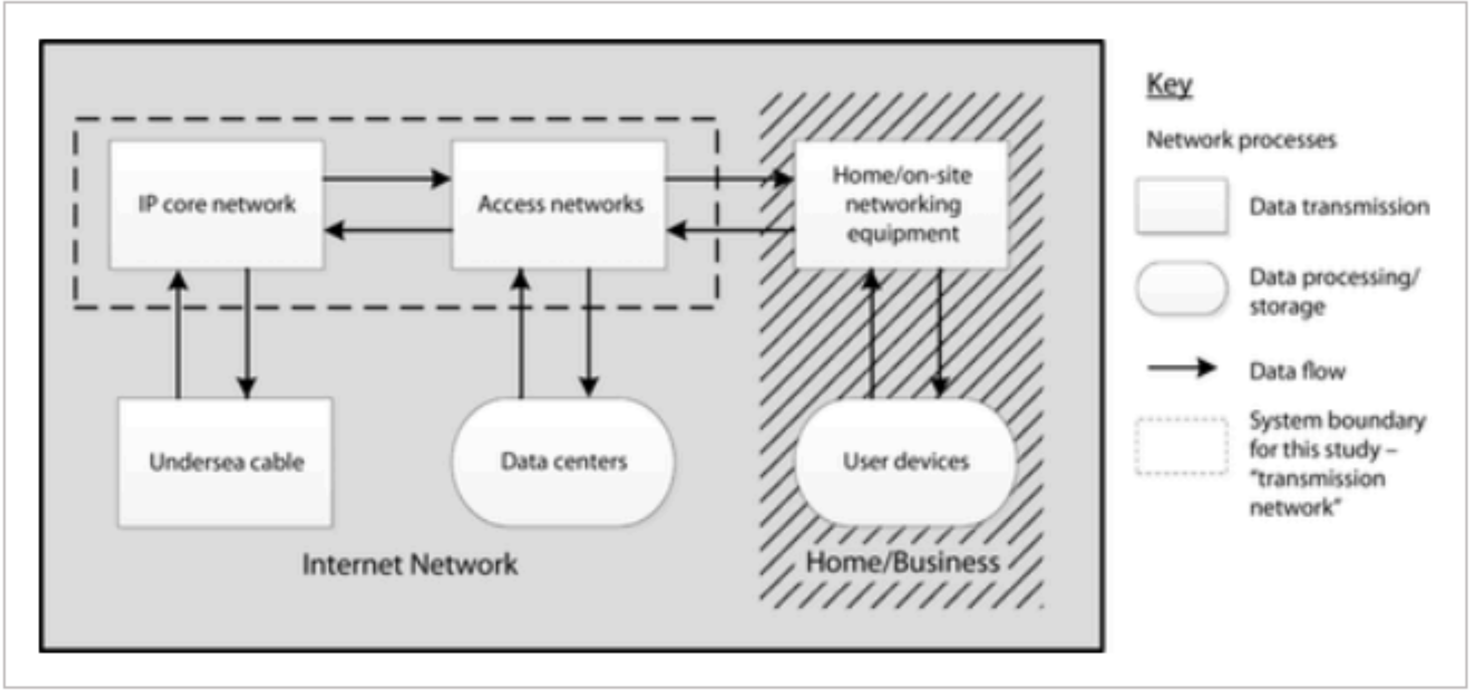
GREEN CODING;

Understanding where CO2 is emitted - CO2 comes from using hardware



Source: Microsoft The Principles of Sustainable Software Engineering [3]

Understanding where CO2 is emitted - CO2 comes from network



Source: Malmodin and Lundén Study 2015

GREEN CODING;

Understanding where CO2 is emitted - Quick Summary

1. CO2 comes from buying hardware

2. CO2 comes from using hardware

    2a. CO2 comes from software not using hardware at all (Idle load)

    2b. CO2 comes from software using hardware inefficiently (bad code)

3. CO2 is emitted from the energy use of the network

In a best case scenario Hyperscalers handle #1 and #2a (if you don't overprovision !)

=> But how to even measure your status quo for #2b?
=> And how to estimate #3

Agenda

1. Understanding where CO2 is emitted
2. Measurement overview & getting to kWh / J / CO2
3. Overview of Open Source Tools
4. Examples & Watch out for rebound effects

How to actually measure software energy use - No accepted approach

Electrical

- AC energy meter in server rack / wall socket (Buffer effects in PSU)
- DC inline current of PSU (hard to integrated afterwards, power lines not exclusive)

Software

- CPU and / or Mem usage (robust transfer model?, idle power?)

Sensors

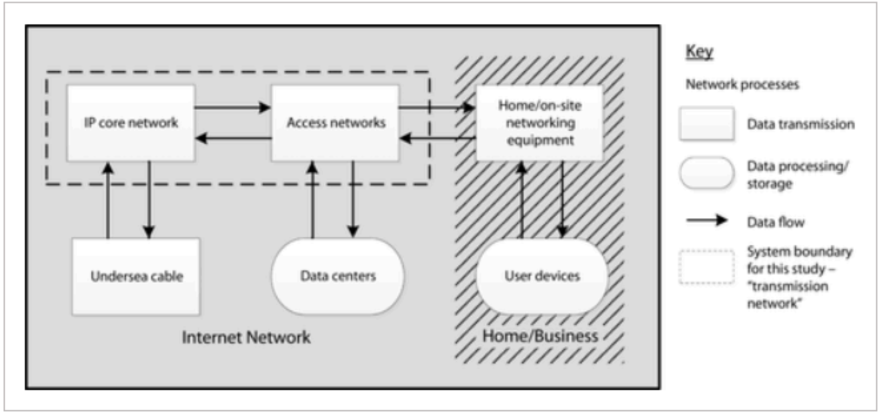
- RAPL (low resolution / averaging / security concerns)

Instruction based

- Decompile and count Instructions (reference values, decompile?)
- perf

How to actually measure software energy use - Measuring network energy

- Data Size
- Distance
- # Network Hops
- Energy efficiency of network equipment
- Carbon Intensity of grid
- Overhead of protocol stack:
    - TLS, HTTP 1/2/3, Multiplexing etc.



Source: Malmodin and Lundén Study 2015

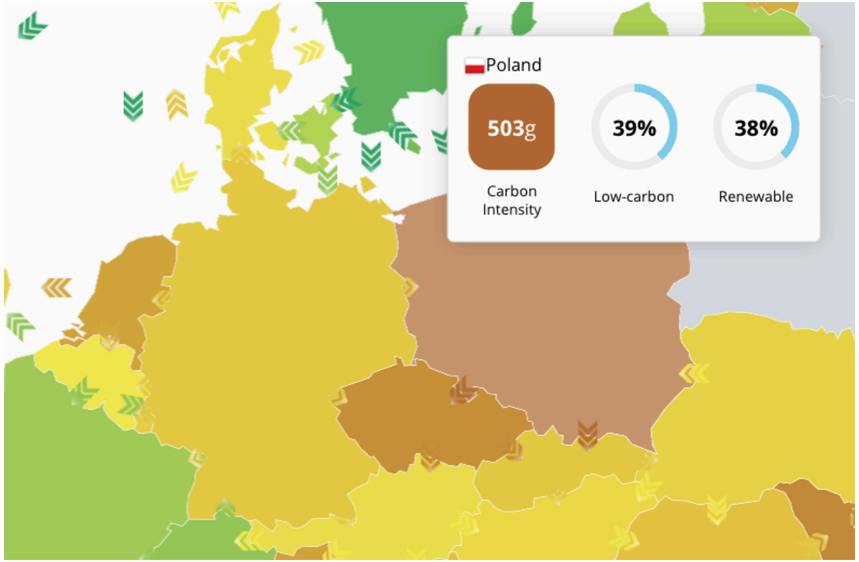
GREEN CODING;

How to actually measure software energy use - 1-Byte Model

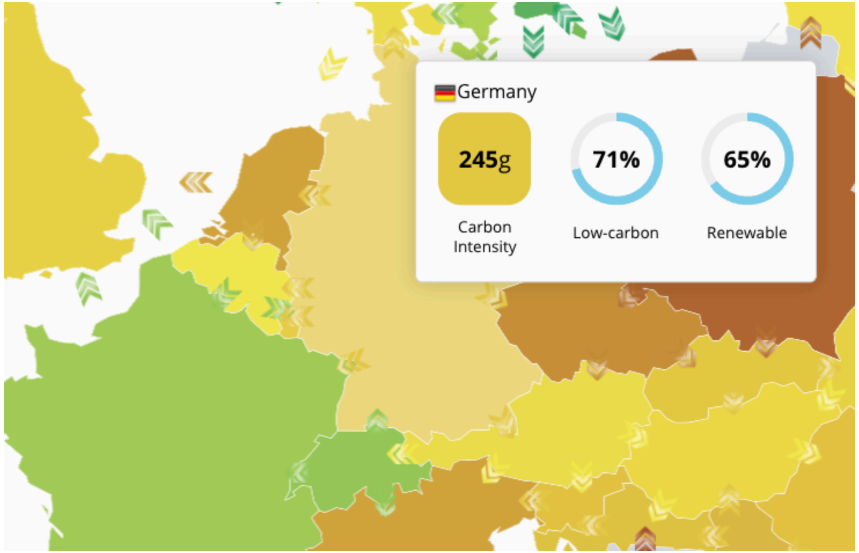
- Based on 2019 study by The Shift Project: **0.023 kWh/GB**

- Global average conversion factor with many downsides:
  - more accurate in western countries than ex. India
  - Critiques say 10-100x too high or too low
  - Timing is based on Data Volume, although often time is more accurate
  - Mixes mobile, fibre, cable and WIFI traffic
  - ...

Calculate Network Energy Use - Carbon intensity of the grid (via API)



https://app.electricitymap.org/map



https://app.electricitymap.org/map

GREEN CODING;

Agenda

1. Understanding where CO2 is emitted

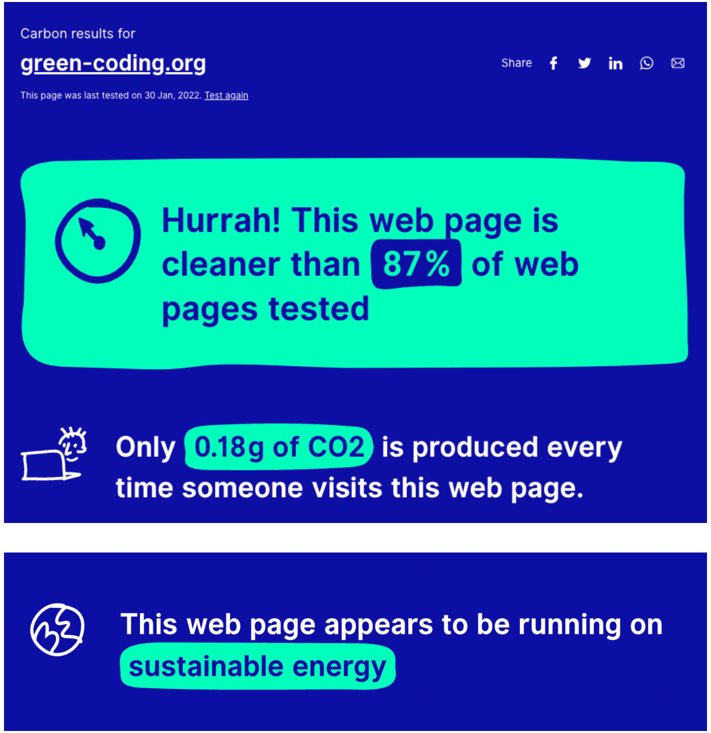
2. Measurement overview & getting to kWh / J / CO2

3. Overview of Open Source Tools

4. Examples & Watch out for rebound effects

Overview of Open Source Tools - Measuring user facing frontends

- Websitecarbon, most famous
- CO2 budget based on transmitted data
- Can compare to a database of average values
- Does not measure client / server compute



Carbon results for

green-coding.org

This page was last tested on 30 Jan, 2022. Test again

Hurrah! This web page is cleaner than 87% of web pages tested

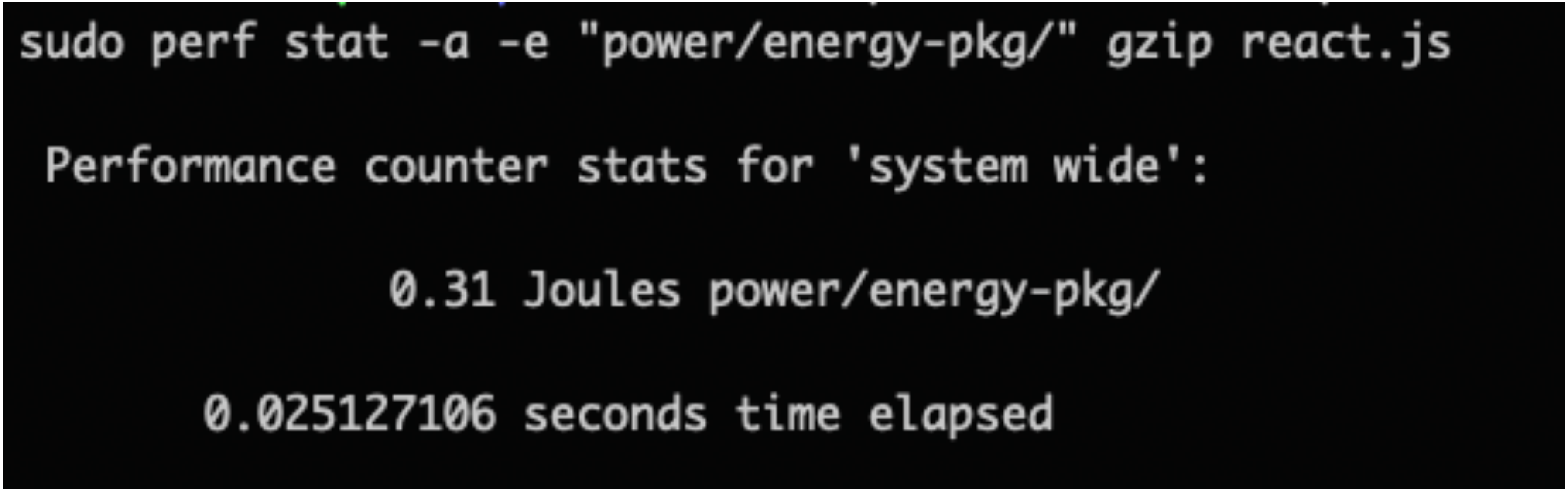
Only 0.18g of CO2 is produced every time someone visits this web page.

This web page appears to be running on sustainable energy

https://www.websitecarbon.com

Overview of Open Source Tools - Measuring PMCs on OS Level

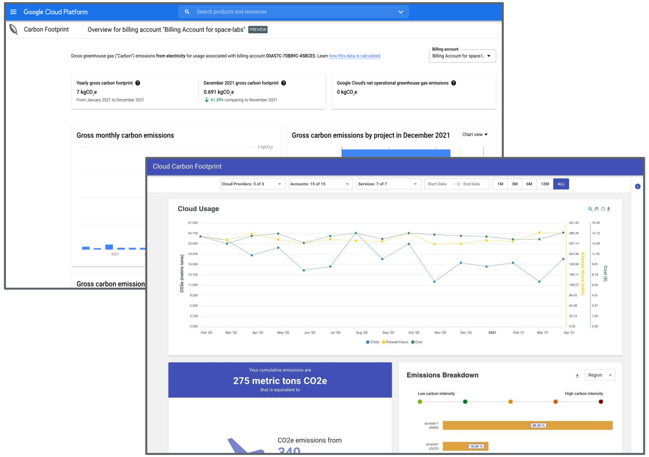
Many open source tools to query Linux subsystem energy reporting:
=> perf, scaphandre etc.



```
sudo perf stat -a -e "power/energy-pkg/" gzip react.js

 Performance counter stats for 'system wide':

              0.31 Joules power/energy-pkg/

       0.025127106 seconds time elapsed
```

Open Source Toolchains - Tools for the Cloud

- Thoughtworks CCF & Google Carbon Footprint industry standard atm
- Both good for company wide carbon accounting (use both! bottom-up vs. top-down)
- Not usable for service- /code-level optimizations => No PMCs in Cloud (security concerns)
- Network traffic not factored in



Google Cloud Platform

Carbon Footprint — Overview for billing account "Billing Account for space-labs" PREVIEW

Gross greenhouse gas ("Carbon") emissions **from electricity** for usage associated with billing account **00A57C-70B89C-45BCE5**. Learn how this data is calculated.

Billing account: Billing Account for space-l...

Yearly gross carbon footprint: 7 kgCO$_2$e — From January 2021 to December 2021

December 2021 gross carbon footprint: 0.691 kgCO$_2$e — 61.39% comparing to November 2021

Google Cloud's net operational greenhouse gas emissions: 0 kgCO$_2$e

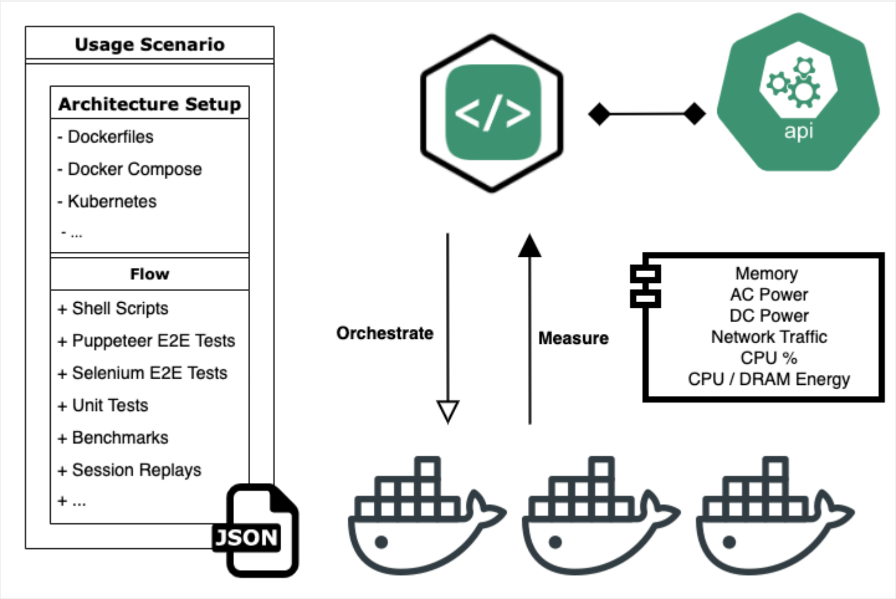
Cloud Carbon Footprint

Cloud Providers: 3 of 3 · Accounts: 15 of 15 · Services: 7 of 7 · Start Date → End Date

Your cumulative emissions are **275 metric tons CO2e** that is equivalent to

CO2e emissions from 340

Open Source Toolchains - Tools that factor in architecture - Ex. Green Metrics Tool



Usage Scenario

Architecture Setup
- Dockerfiles
- Docker Compose
- Kubernetes
- ...

Flow
+ Shell Scripts
+ Puppeteer E2E Tests
+ Selenium E2E Tests
+ Unit Tests
+ Benchmarks
+ Session Replays
+ ...

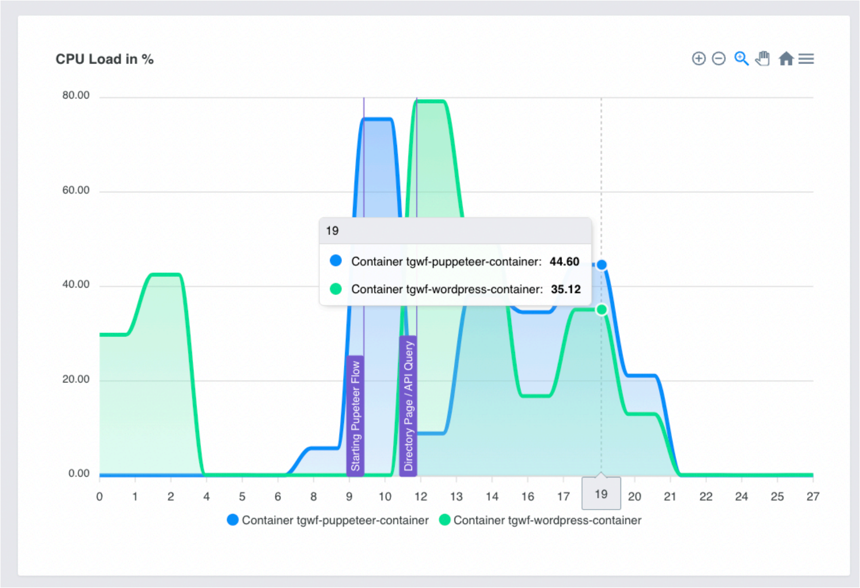
JSON

api

Orchestrate

Measure

Memory
AC Power
DC Power
Network Traffic
CPU %
CPU / DRAM Energy

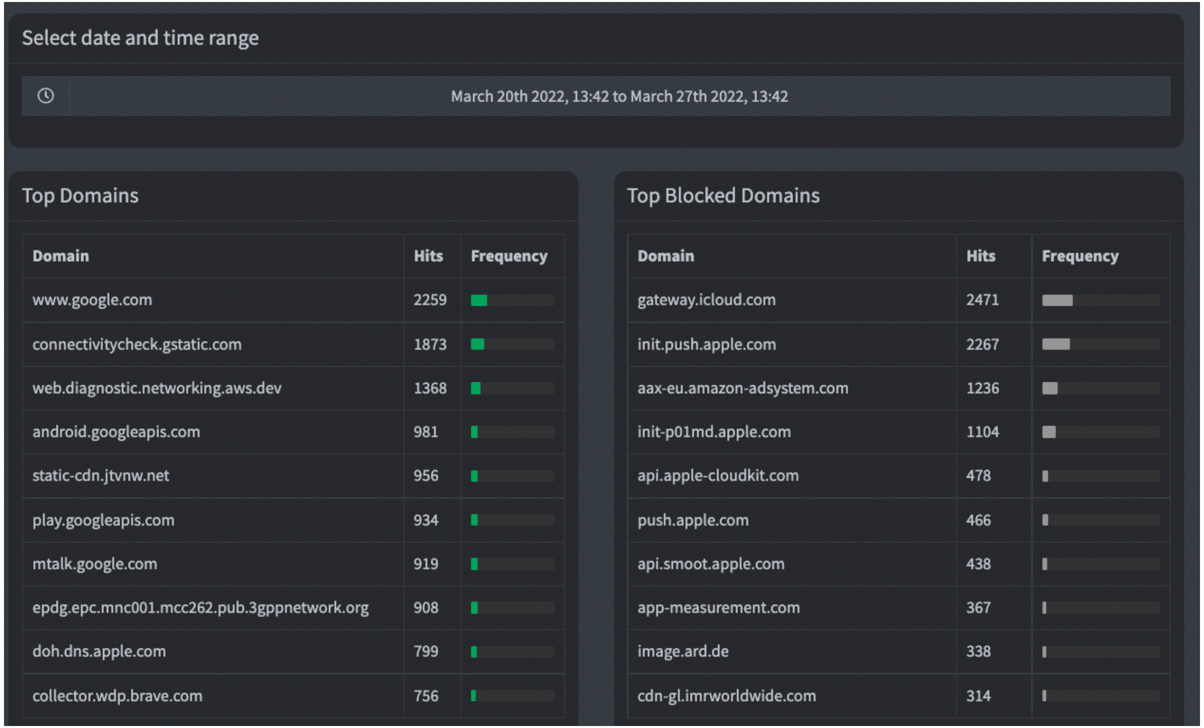


Agenda

1. Understanding where CO2 is emitted
2. Measurement overview & getting to kWh / J / CO2
3. Overview of Open Source Tools
4. Examples & Watch out for rebound effects

How to calculate software energy use - Example: PiHole DNS sink / network filter

- 7 days sample on right => 9479 requests.
- 1 kB assumed
- 9479 kB in total
- **~ 521 MB p.a.**



Select date and time range

March 20th 2022, 13:42 to March 27th 2022, 13:42

Top Domains

Domain	Hits	Frequency
www.google.com	2259	🟩
connectivitycheck.gstatic.com	1873	🟩
web.diagnostic.networking.aws.dev	1368	🟩
android.googleapis.com	981	🟩
static-cdn.jtvnw.net	956	🟩
play.googleapis.com	934	🟩
mtalk.google.com	919	🟩
epdg.epc.mnc001.mcc262.pub.3gppnetwork.org	908	🟩
doh.dns.apple.com	799	🟩
collector.wdp.brave.com	756	🟩

Top Blocked Domains

Domain	Hits	Frequency
gateway.icloud.com	2471	▨
init.push.apple.com	2267	▨
aax-eu.amazon-adsystem.com	1236	▨
init-p01md.apple.com	1104	▨
api.apple-cloudkit.com	478	▨
push.apple.com	466	▨
api.smoot.apple.com	438	▨
app-measurement.com	367	▨
image.ard.de	338	▨
cdn-gl.imrworldwide.com	314	▨

GREEN CODING;

How to calculate software energy use - Example: PiHole DNS sink / network filter

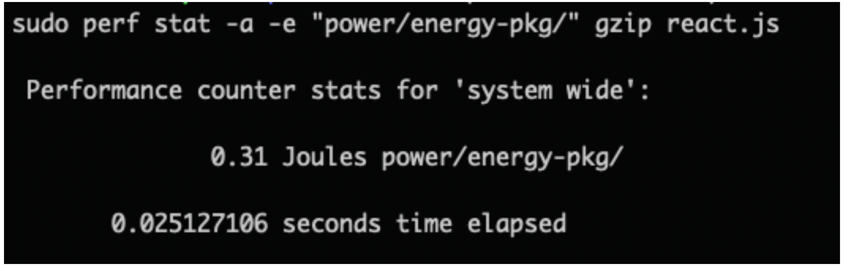
- 521 MB p.a. * .000023 kWh / MB
  => 1.1766 kWh p.a.
- 1.1766 kWh p.a. * 0.519 CO2e/kWh
  => **0.61 kgCO2e p.a.**

Backlash

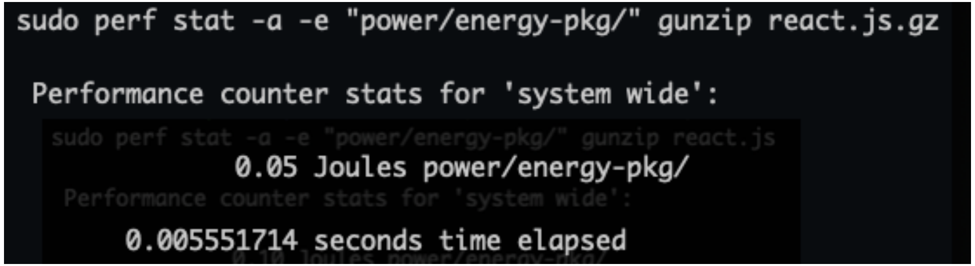
- RaspberryPi v3: 2.2 W
  - **~10 kg CO2e p.a.** (0.165 Trees) 😢
  - not factored in embodied carbon 😢😢
  (~ 45 kg once for a smartphone (substitute) [10])

How to actually measure software energy use - Evaluate compression: gzip

- Transmit without compression: ~626 kB
- Transmit with compression: ~142 kB
- GZIP savings: ~484 kB => **.578 mgCO2e**



```
sudo perf stat -a -e "power/energy-pkg/" gzip react.js

 Performance counter stats for 'system wide':

             0.31 Joules power/energy-pkg/

      0.025127106 seconds time elapsed
```



```
sudo perf stat -a -e "power/energy-pkg/" gunzip react.js.gz

 Performance counter stats for 'system wide':
  sudo perf stat -a -e "power/energy-pkg/" gunzip react.js
             0.05 Joules power/energy-pkg/
   Performance counter stats for 'system wide':
      0.005551714 seconds time elapsed
```

- Energy needed to compress / decompress: 0.41 J => **.00519 mgCO2e**
- Net savings: ~ **.573 gCO2e** (almost 100%)

GREEN CODING;

Rebound effects of technology - Evaluate compression: gzip

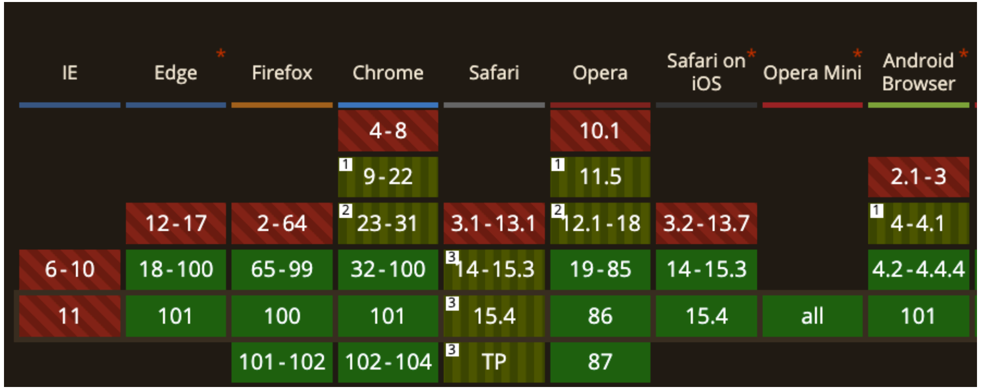
- Ex: Embodied Carbon:
  Digital Tech consumes CO2 when built (45%) & when used (55%)

  => New technology might not run on old device anymore

- Feature (ex. WebP) that requires polyfill with more energy consumption or more data storage (PNG alternative)

- Cutting edge android feature, but manufacturer provides no updates

- ...



IE	Edge *	Firefox	Chrome	Safari	Opera	Safari on iOS *	Opera Mini *	Android Browser *
			4-8		10.1			
			[1] 9-22		[1] 11.5			2.1-3
	12-17	2-64	[2] 23-31	3.1-13.1	[2] 12.1-18	3.2-13.7		[1] 4-4.1
6-10	18-100	65-99	32-100	[3] 14-15.3	19-85	14-15.3		4.2-4.4.4
11	101	100	101	[3] 15.4	86	15.4	all	101
		101-102	102-104	[3] TP	87			

caniuse.com

Carbon impact of an app is a sum of metrics - Summary of lifecycle parts

- Load on Client Side
- Load on Server side
- Network CO2 emissions
- Idle resource consumption
- Resource consumption of distributed storage
- CO2 Procurement costs
- But also: Cost of development, Cost of uninstalling / decomissioning
- => Open-Source can save CO2 through reusing effect

Summary - Take away messages

- Measure, measure, measure. Every metric is better than flying blind
    - Remember: “If you can’t measure it, you can’t improve it”
      (attributed to Peter Drucker, Lord Kelvin and many more :))
- In Cloud use CCF / Google Carbon Footprint
- Measuring possible through many approaches. Each has drawbacks.
  => Choose multiple if possible
- For code-level / service optimizations use synthetic setup with orchestration tools
- Use formulas to approach network & embodied carbon analytically

Want to dive deeper and more details? - Follow Green Coding Berlin

- Website and blog: https://www.green-coding.org

- Check out our newsletter!

- Meetup group: https://www.meetup.com/green-coding

- We do research and talks about the energy use of software, frameworks, programming languages, Browsers etc.

- Github: https://github.com/green-coding-berlin

- Hit me up directly: arne@green-coding.org / https://www.linkedin.com/in/arne-tarara

Sources

- [1] Green Software Foundation Whitepaper: https://greensoftware.foundation/articles/sustainable-systems-user-hardware-and-sustainability (Link)]
- [2] https://i.dell.com/sites/csdocuments/CorpComm_Docs/en/carbon-footprint-poweredge-r740xd.pdf
- [3] https://docs.microsoft.com/en-gb/learn/modules/sustainable-software-engineering-overview/7-energy-proportionality
- Global average CO2eq/kWh: 0.519 (https://docs.microsoft.com/en-us/learn/modules/sustainable-software-engineering-overview/8-network-efficiency)
- https://devblogs.microsoft.com/sustainable-software/measuring-the-carbon-impact-of-web-browsing/
- https://reboxed.co/blogs/outsidethebox/the-carbon-footprint-of-your-phone-and-how-you-can-reduce-it
- Greenlab Study: https://greenlab.di.uminho.pt/wp-content/uploads/2017/10/sleFinal.pdf
- https://devblogs.microsoft.com/sustainable-software/language-impact-on-ui-apps/
- https://tech.oyorooms.com/how-brotli-compression-gave-us-37-latency-improvement-14d41e50fee4
- https://dl.acm.org/doi/10.1145/2896967.2896968
- https://ieeexplore.ieee.org/abstract/document/7809319
- https://onlinelibrary.wiley.com/doi/full/10.1111/jiec.12630