Making software sustainable

Through measuring and optimizing energy & emissions

</> GREEN CODING;

Who am I

Arne Tarara - Green Coding Solutions

- CEO & Founder for Green Coding Solutions
- Software Developer 16+ years
- We specialize in making software sustainble through benchmarking and optimization
- All our tools are open source for industry and R&D

</> GREEN CODING;

Software emits CO2

And we as developers can and should contribute to reduce it's climate impact

The concept of Green Coding

And how can it help?

- Workloads / user demands are the reason why hardware is manufactured and used at all
- Virtually every modern digital device runs on software
- Green coding focuses on awareness, resource consumption and reduction

- **Efficiency** - Savings by changing software/hardware
- **Responsibility** - emissions are reduced by those who cause them. Not with the service provider
- **Transparency** - measure and communicate/report on resource consumption
- **Specialized CO2/energy saving techniques**

Our entry question

What is the goal of this workshop?

We want to make software sustainable

Our definition of a sustainable software and our approach

- A sustainable software knows about it's **energy** and **CO2 emission**
- A sustainable software is constantly **monitored** and **optimized** for savings
- A sustainable software uses the **least amount of resources** for a given task (compared with similar applications)
- A sustainable software is lean and does **combat software bloat**

- **=> Empower you to make claims about Software CO2 and use it in hobby, client, university projects. Or submit PR to open source :)**

</> GREEN CODING;

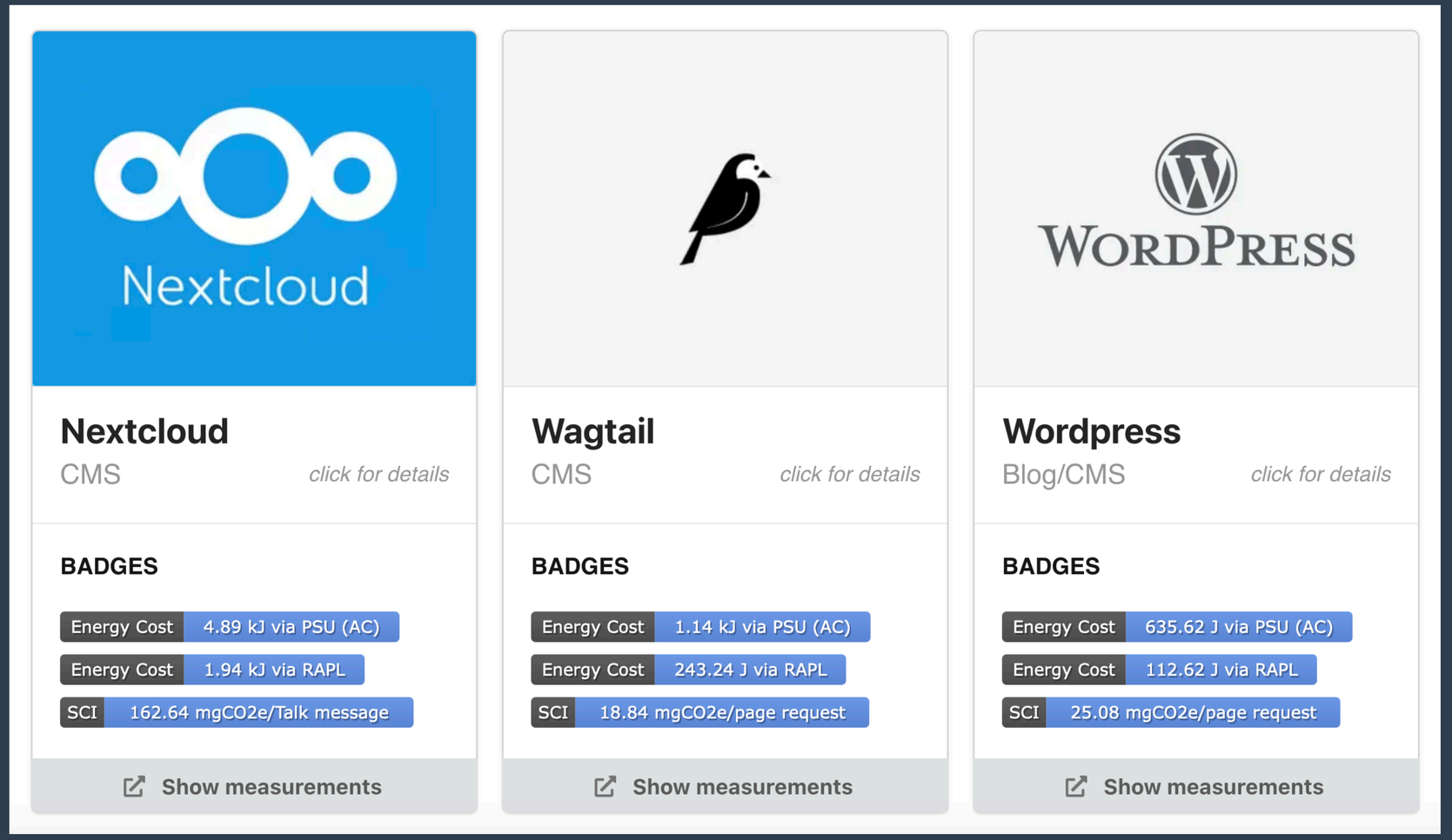
The goal

Making a CO2 claim about your company / product

- "Our operations consume XXX tons CO2 per year"
- "A request to our ChattyGPTx Model costs XXg CO2"
- "One Google search costs XX gCO2"
- "My Gitlab pipelines uses XX gCO2 per run when run in Berlin, but only YY gCO2 per run when done in Paris"

The goal

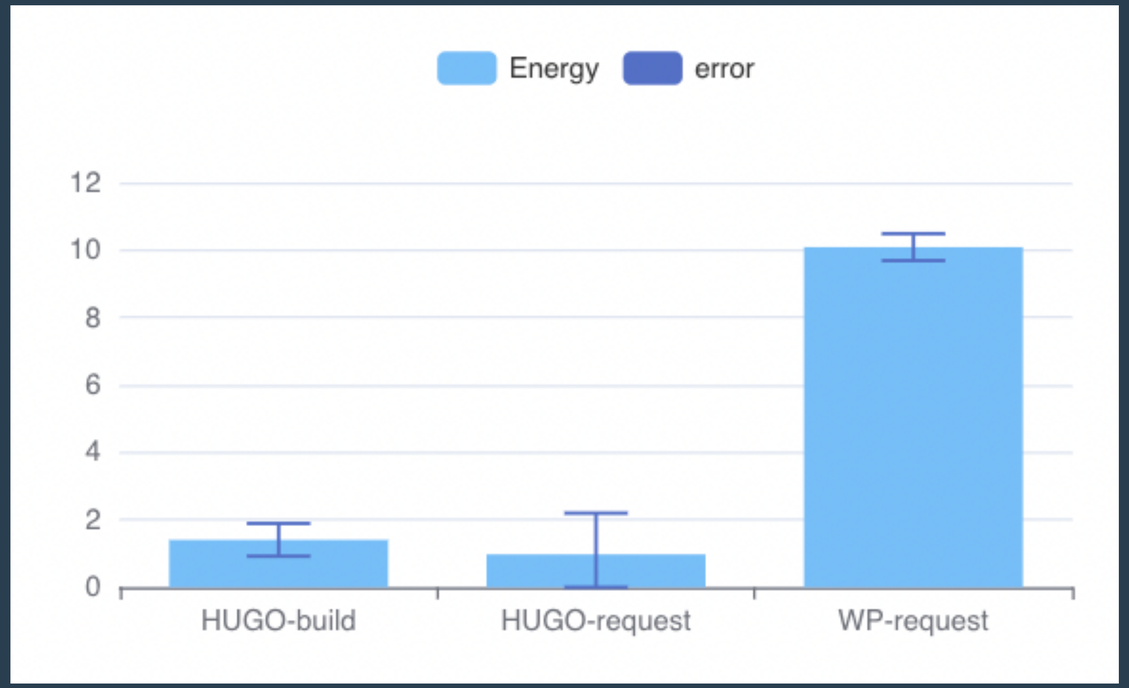
To talk about software in a comparative way



The goal

To have actionable data

- Wordpress vs. Cloudflare [HUGO]
  - 8-fold savings per request
  - Net profit from the very first request
- The complete rendering of the static page costs as much as a dynamic request with Wordpress



https://www.green-coding.io/case-studies/wordpress-vs-hugo-cloudflare/

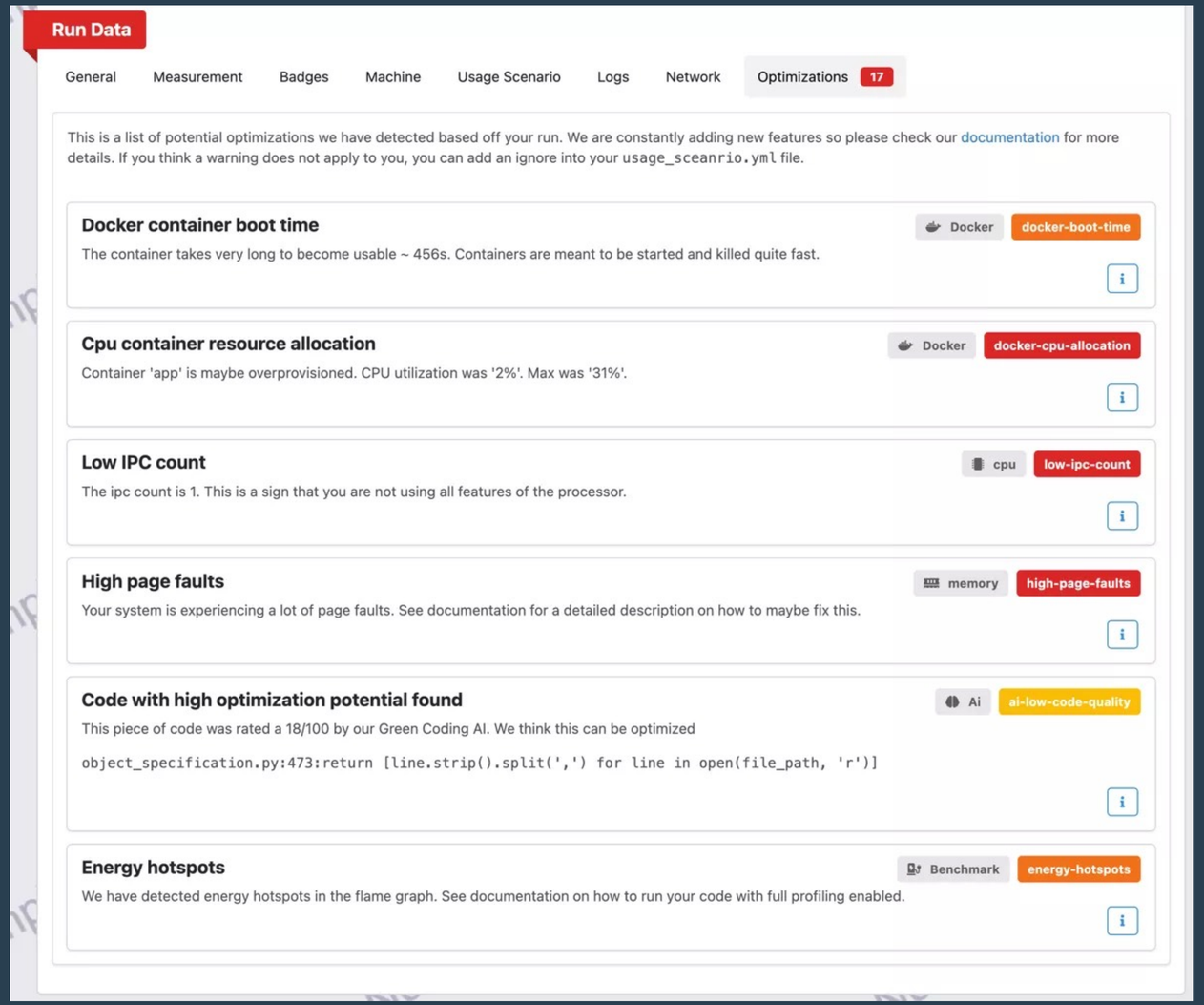
- Serverless hosting allows the server to switch off between requests

GREEN CODING;

The goal

To also get recommendations

- Optimization features
- Based on resource limits and performance metrics
- Includes runtime, build and Idle phases
- Currently in beta in GMT



GREEN CODING;

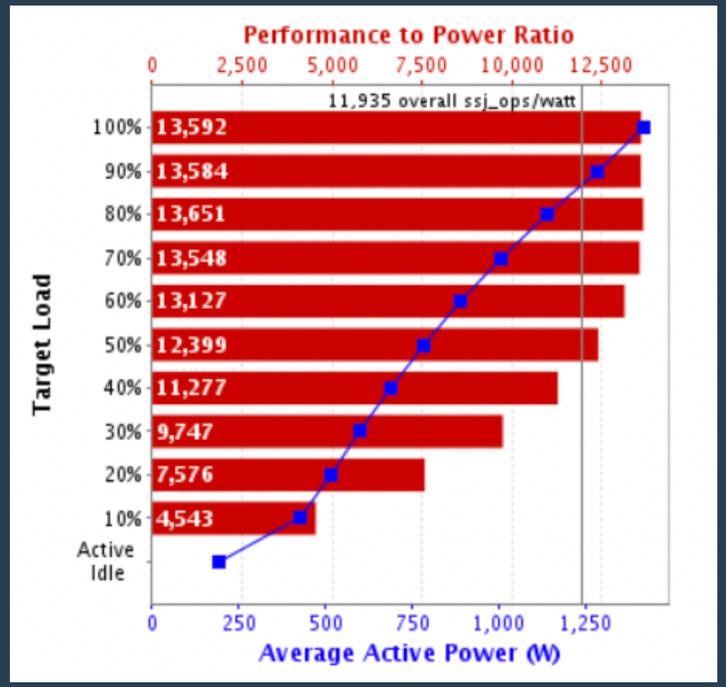
Our follow-up question

What technical parts do we need to get to a sustainable software?

Getting the energy

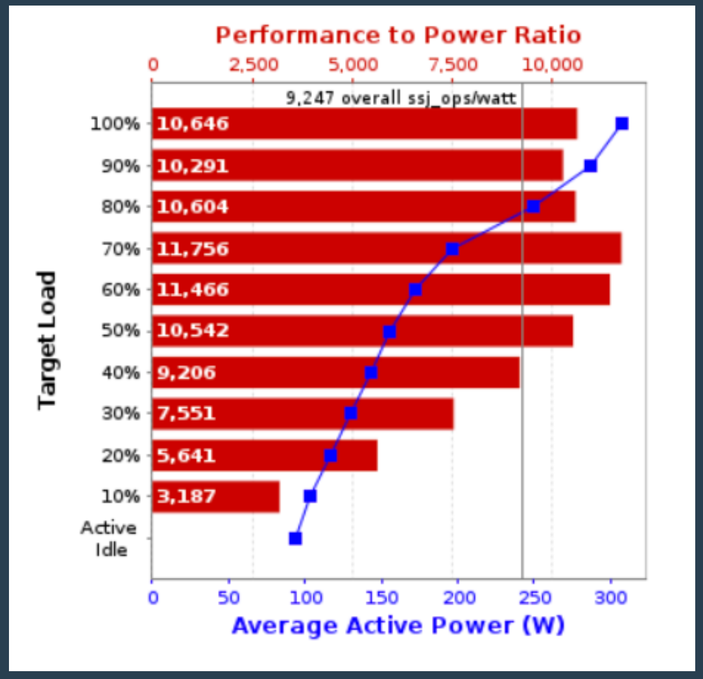
Through measuring directly - Or through estimation via Machine Learning

High Idle, but almost linear



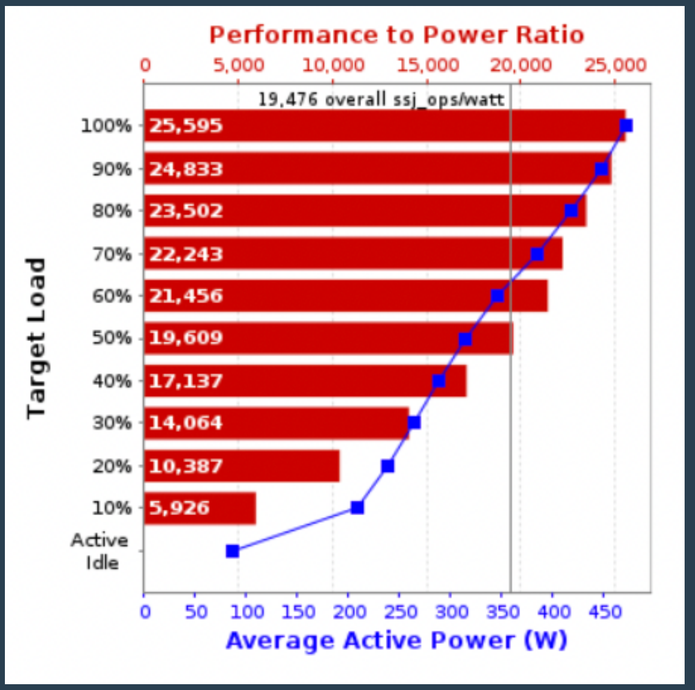
ASUSTeK Computer Inc. RS720Q-E9-RS8 (2019)

50% Power increase at 70% utilization



Hewlett Packard Enterprise ProLiant DL110 Gen10 Plus

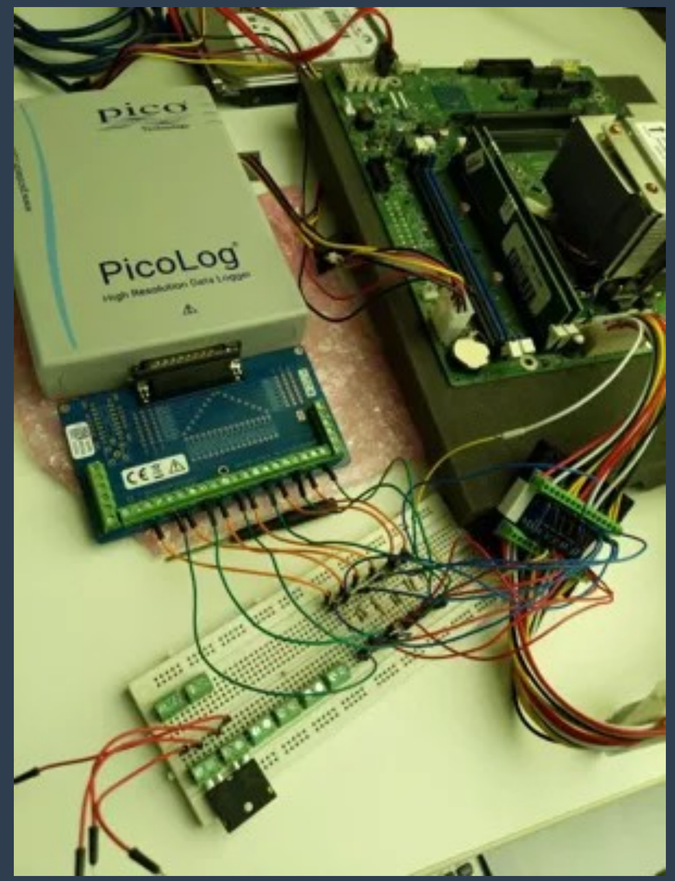
Idle optimized



QuantaGrid D43K-1U (2022)

Measuring total machine energy

ATX / DC vs. AC-Measurements

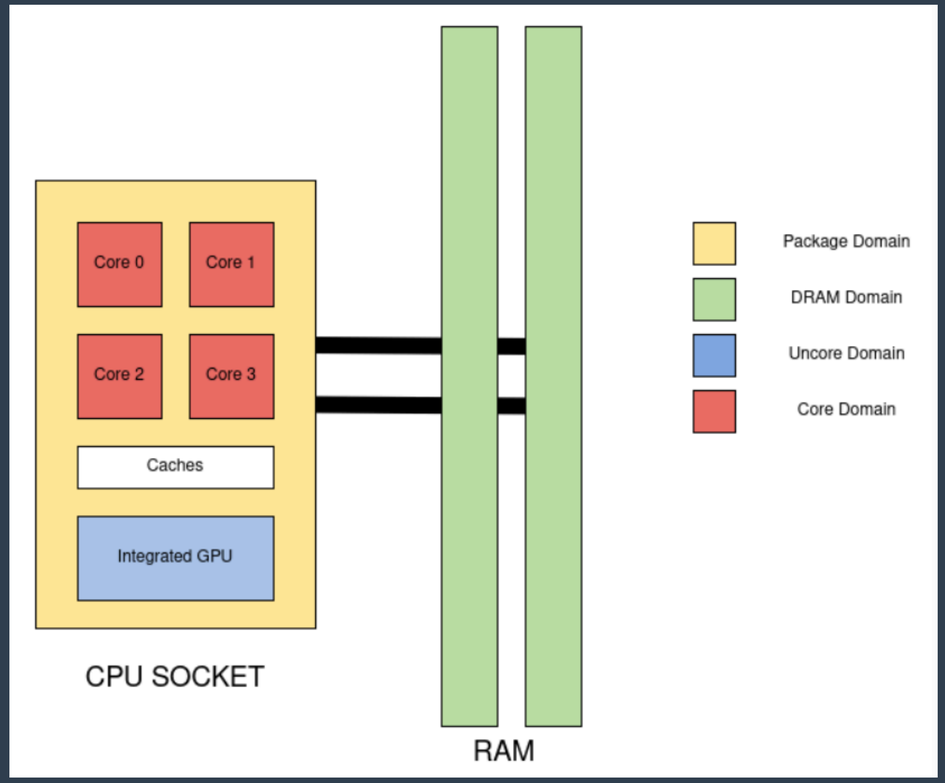


GREEN CODING;

Details on RAPL

The most used technology atm

- Energy measurement capabilities on most modern Intel/AMD processors
- Measure:
  - CPU Energy per Core / Package
  - RAM
  - Integrated GPU
- Software model of capacitor readings on mainboard
  - Resolution 1ms / 15.3 microJoules
- Exposed in Linux kernel through device



Source: https://pyjoules.readthedocs.io/en/stable/devices/intel_cpu.html

GREEN CODING;

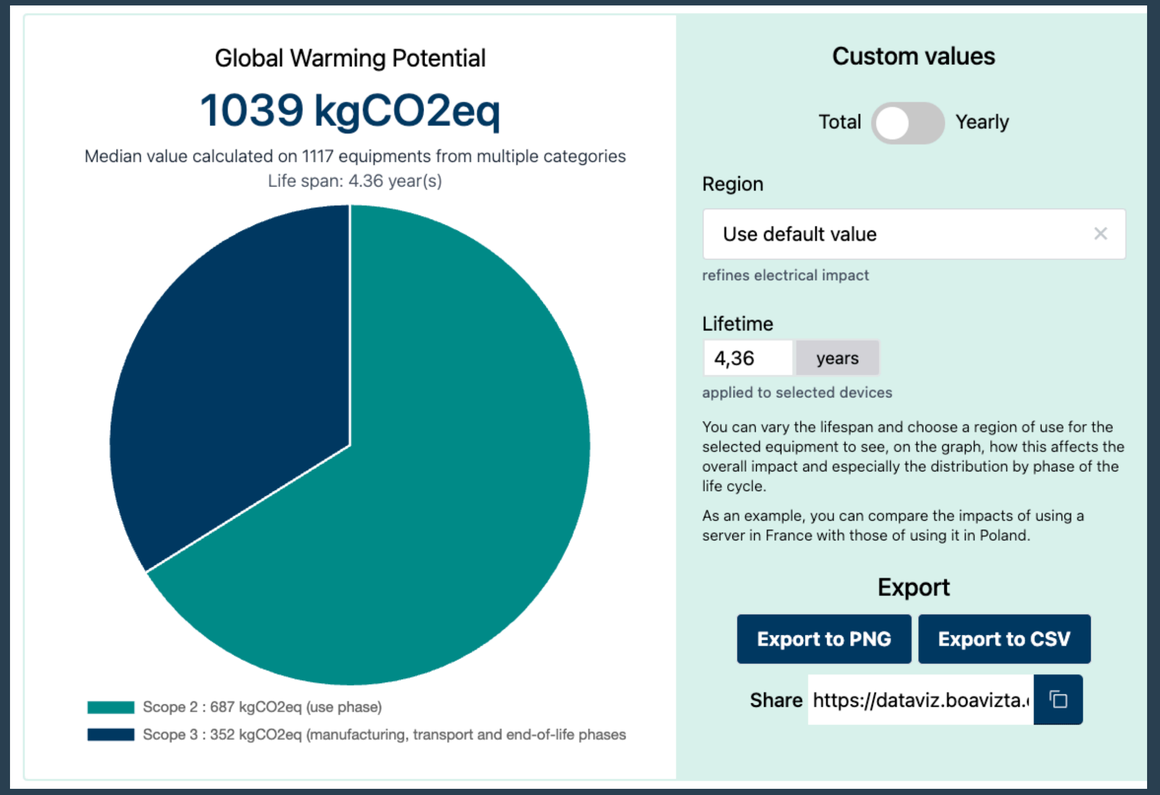
How to get embodied carbon

Using Life-Cycle-Assessment databases

- **Boavizta**
  https://dataviz.boavizta.org/manufacturerdata
- **Microsoft**
  https://tco.exploresurface.com/sustainability/calculator
- **Dell**

  Example: https://www.delltechnologies.com/asset/en-us/products/servers/technical-support/Full_LCA_Dell_R740.pdf

- ... many more

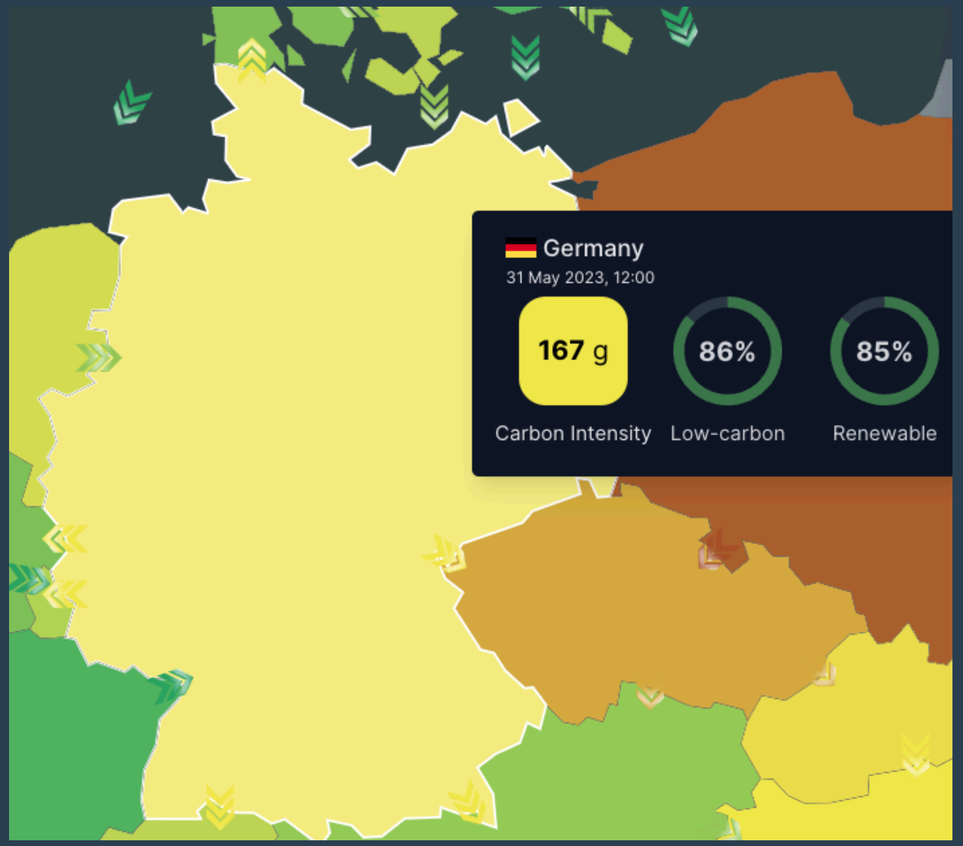


Source: https://dataviz.boavizta.org/manufacturerdata

How to get emission factors

Using open data sources

- **Electricitymaps**
  https://www.electricitymaps.com/
- **Carbon-Aware-SDK**
  https://github.com/Green-Software-Foundation/carbon-aware-sdk
- **Bundesnetzagentur (DE)**
  https://www.smard.de/home
- **Wattime (US)**
  https://www.watttime.org/
- ... many more



Source: https://app.electricitymaps.com/zone/DE

SCI by the Green Software Foundation

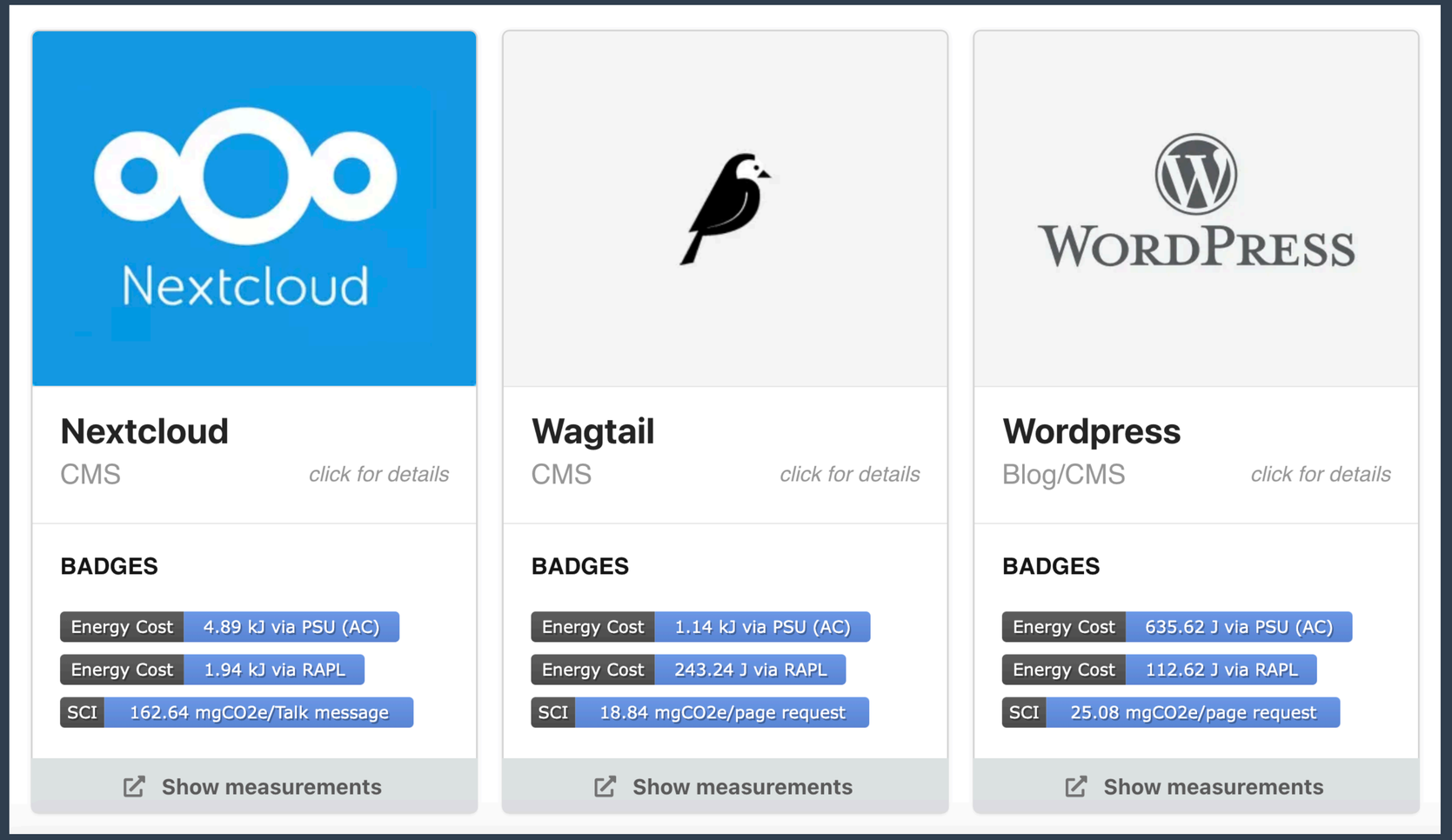
ISO Norm (2024)

SCI = (E * I) + M per R

- (E) - Energy consumption (kilowatt hours) for different components:
  - Ex. CPU/GPUs, Data storage, Memory, Network
- (I) - Emissions factors
- (M) - Embodied emissions
  - Ex. data for servers, mobile devices and laptops
- (R) - Unit of work / use case

The goal

To talk about software in a comparative way



The tools

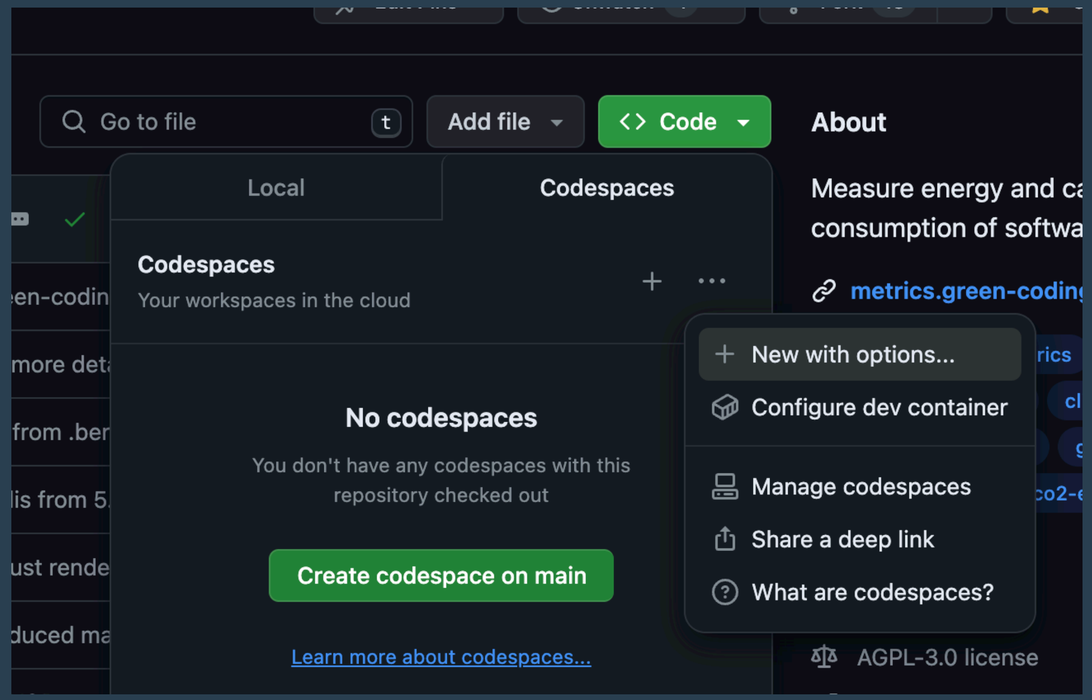
Screenshots!

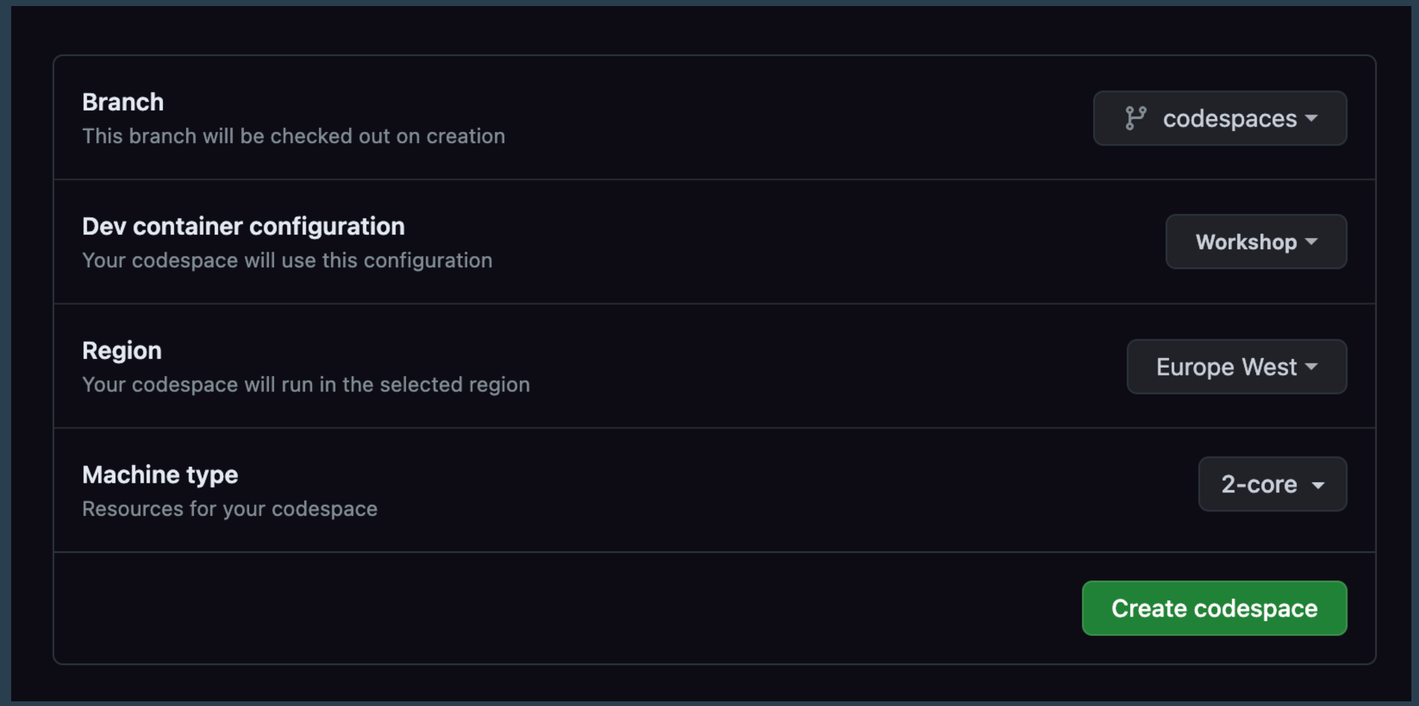
(In case something breaks 🤡)

First, let's fire up a Github Codespace

It takes a while to install ...

- Lets go to https://github.com/green-coding-solutions/green-metrics-tool

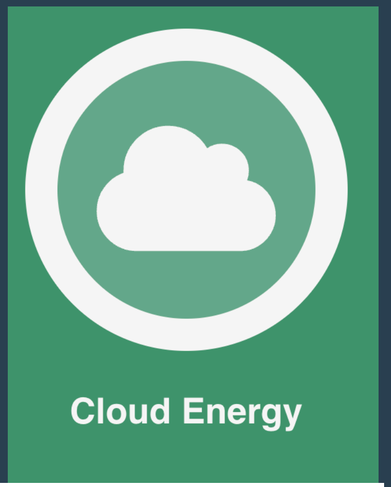




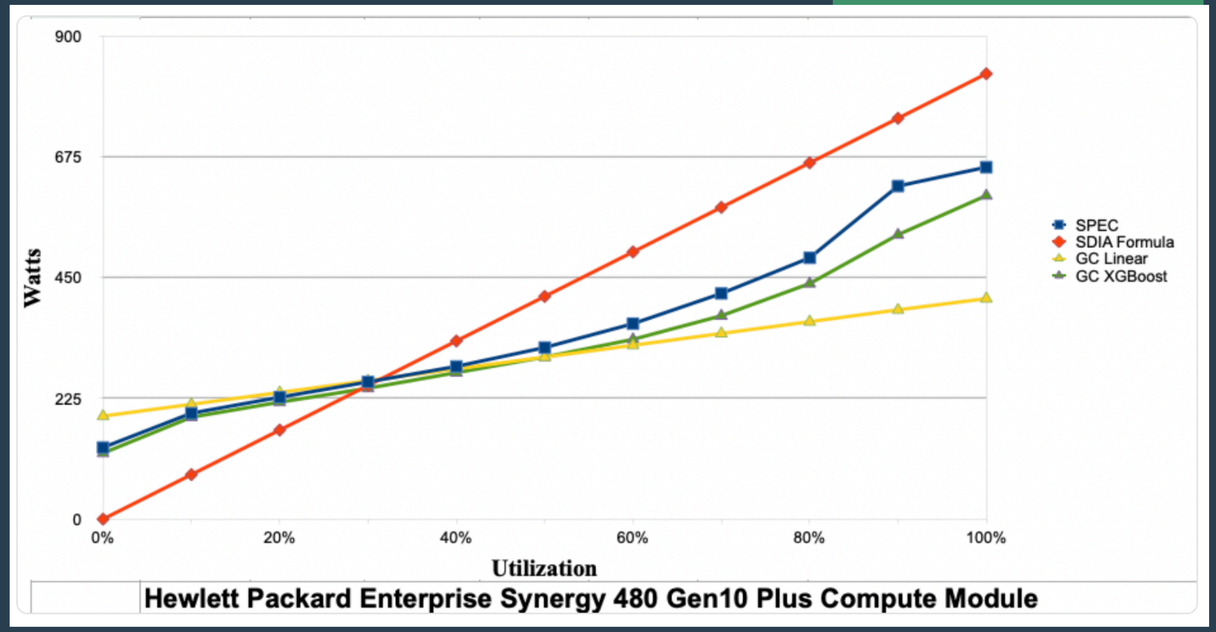
GREEN CODING;

Cloud Energy

Estimation for Cloud and VMs



- The setup of the model is based on a research paper from Interact DC and the University of East London
- 90%+ Accuracy in/out-of sample
- Near 0% overhead
  - XGBoost + POSIX stream implementation



- Lets go to https://github.com/green-coding-solutions/spec-power-model

Cloud environments

XGBoost estimation

- Using ML Models based on power curves of actual machines
  - Non-Linear!
- Caveats:
  - CPU Frequency is needed to be assumed constant
- See our article on this in detail:
  - https://www.green-coding.berlin/case-studies/cpu-utilization-usefulness/
  - https://www.green-coding.berlin/case-studies/hyper-threading-and-energy/
  - etc.

Scaphandre - Hubblo

open-source RAPL based command line tool

- Neat feature: Can split by process

```
Host:   13.1463 W
        package         core            dram            uncore
Socket0 13.1463 W |     10.879847 W     0.748591 W      0.071402 W

Top 5 consumers:
Power           PID     Exe
10.400553 W     16621   "stress"
2.08011 W       16610   "scaphandre"
0.166408 W      2786    "gnome-shell"
0.083204 W      3915    "Xwayland"
0.041602 W      4621    "guake"
----------------------------------------------------------------
```

Let's run Scaphandre!

via https://hubblo-org.github.io/scaphandre-documentation/tutorials/compilation-linux.html

- ## Could not get it working with current version ...
- # from https://www.rust-lang.org/tools/install
- curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
- source "$HOME/.cargo/env"

- git clone https://github.com/hubblo-org/scaphandre.git
- cd scaphandre
- cargo build # binary path is target/debug/scaphandre
- git checkout v0.5.0
- sudo ./target/debug/scaphandre run stdout -t 0

- # But if you have docker in rootless mode:
- sudo docker run -v /sys/class/powercap:/sys/class/powercap -v /proc:/proc -ti hubblo/scaphandre stdout -t 15

- # now we run stress to see changes in separate terminal
- stress-ng -c 1

perf_events

- sudo apt install linux-tools-$(uname -r)
- perf list | grep power # to see what we have available on the system
- perf stat -e power/energy-pkg/ # to read package

- ## Mini Benchmark
- perf stat -e power/energy-pkg/ sleep 10 # to get system baseline over ten seconds
- perf stat -e power/energy-pkg/ stress-ng -c 1 -t 10 # to get system baseline over ten seconds

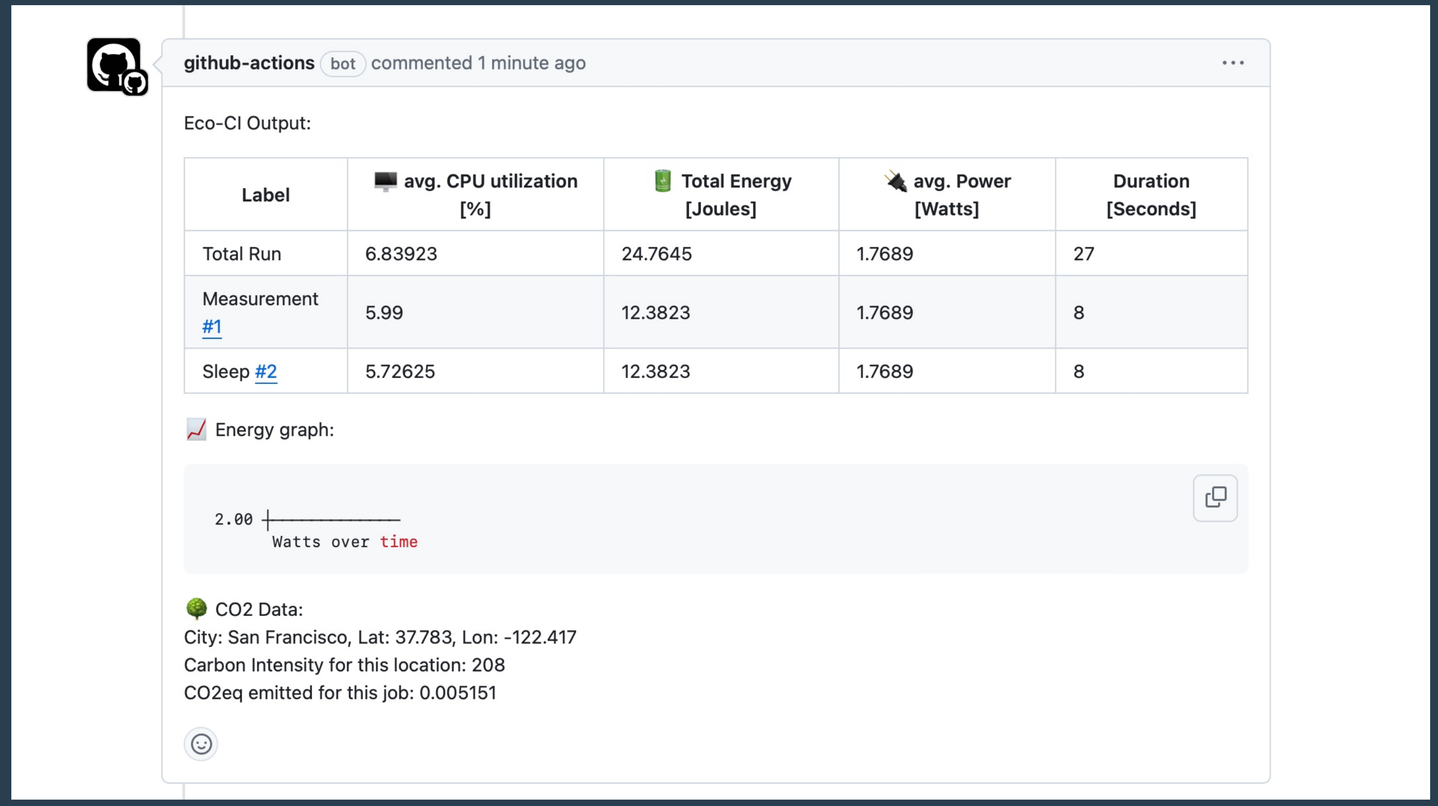
- # Look at IPC
- perf stat -e instructions,cache-misses stress-ng -c 1 -t 1 # to get system baseline over ten seconds

- # Look at "default" defailed view
- perf stat -d stress-ng -c 1 -t 1 # to get system baseline over ten seconds

Eco-CI

Estimation in CI / CD Pipelines

- Integration into Github / Gitlab directly
- Export to central dashboard
- Statistical comparisons over time
- PR-triggered measurements



github-actions bot commented 1 minute ago

Eco-CI Output:

Label	🖥️ avg. CPU utilization [%]	🪫 Total Energy [Joules]	🔌 avg. Power [Watts]	Duration [Seconds]
Total Run	6.83923	24.7645	1.7689	27
Measurement #1	5.99	12.3823	1.7689	8
Sleep #2	5.72625	12.3823	1.7689	8

📈 Energy graph:

```
2.00 ┼──────────
      Watts over time
```

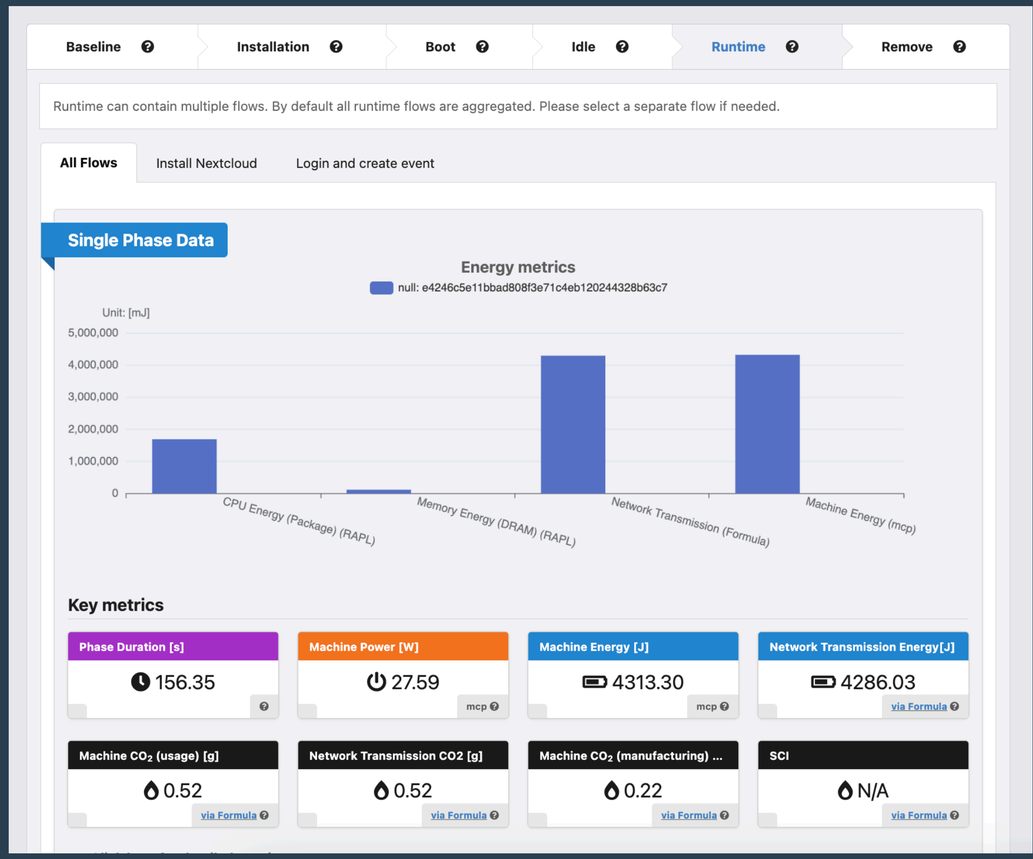
🌳 CO2 Data:
City: San Francisco, Lat: 37.783, Lon: -122.417
Carbon Intensity for this location: 208
CO2eq emitted for this job: 0.005151

Lets got to: https://github.com/green-coding-berlin/green-metrics-tool to see it in action

Quantifying a product

Example: Green Metrics Tool

- Benchmarking via Standard-Usage-Scenarios (academia based via UCB / Öko-Insitut e.V.)
- Look at software in all phases of execution Building, Idle, Runtime etc.

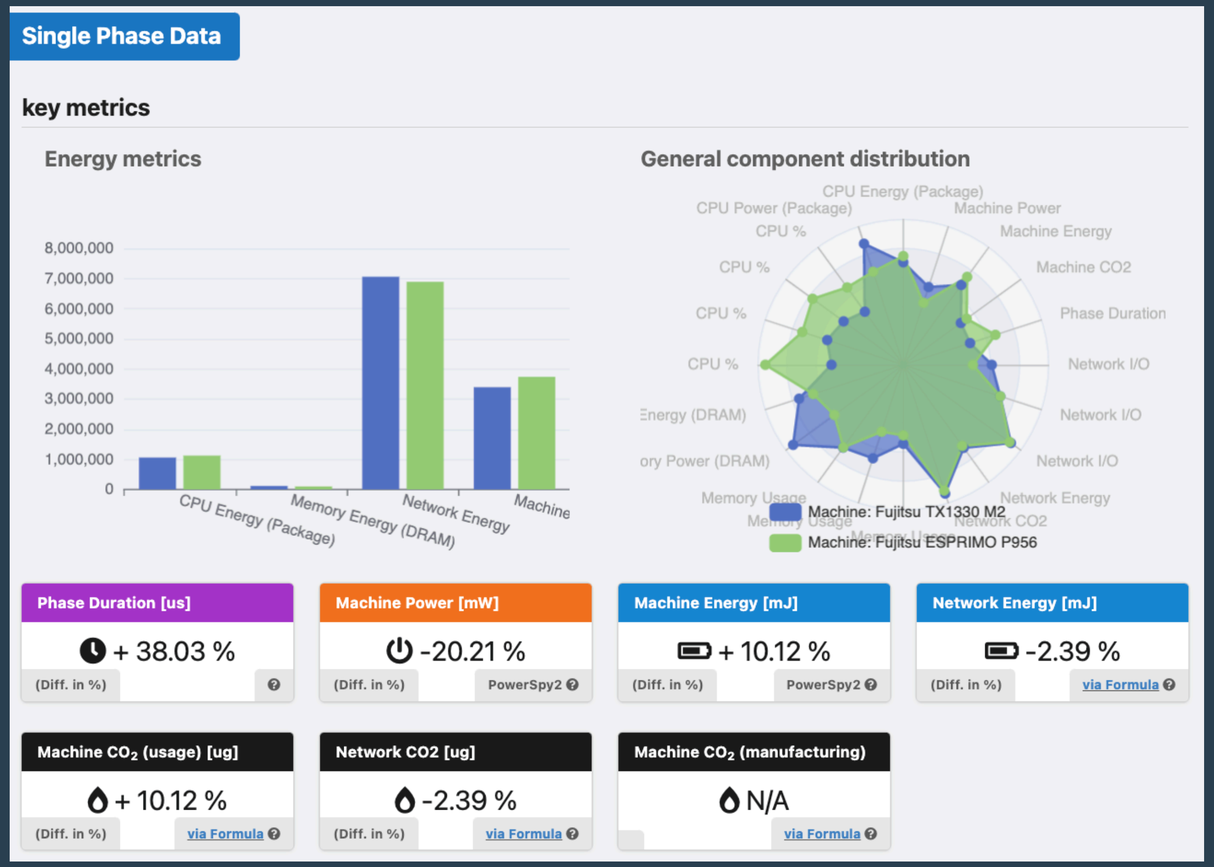


- Include embodied carbon and modular metrics reporters
- SCI (ISO-Norm tbd 2024) / Blauer Engel compatible (German Eco Label)

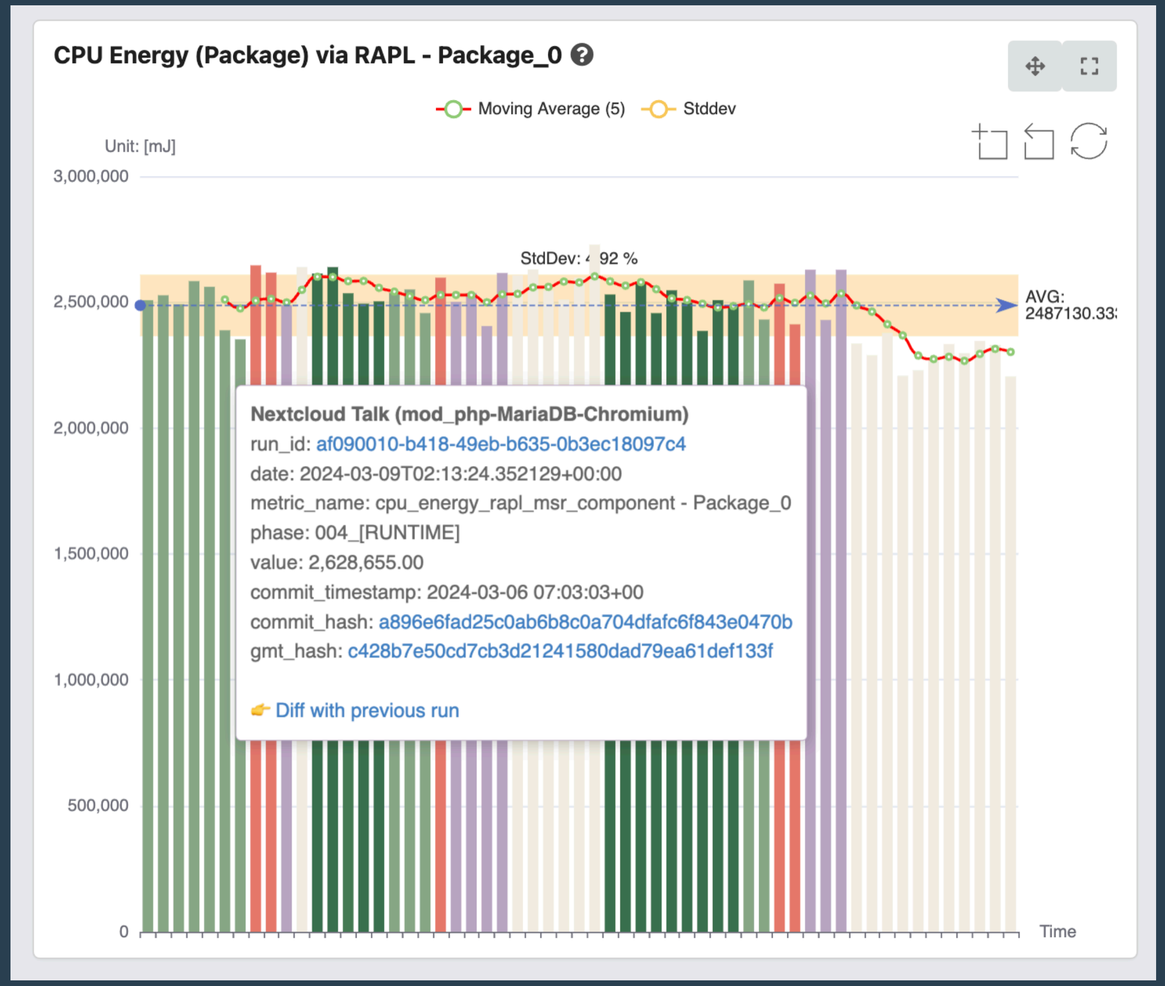
Online Dashboard: https://metrics.green-coding.io/index.html

Green Metrics Tool

More Features



Comparisons between Hardware / Repository / Branch etc.



Measurement per Commit - Timeline View

GREEN CODING;