Understanding Software Carbon emissions

An overview of methodologies and tools

</> GREEN CODING;

Standard slide for starters

Quick info - Arne Tarara / Green Coding Berlin

- Software-Dev 16+ years
- Founder & CEO **Green Coding Berlin GmbH**
  - We do research and development in open source tools for software resource consumption
  - We help organizations to run a more sustainable digital infrastructure

GSD COMMUNITY

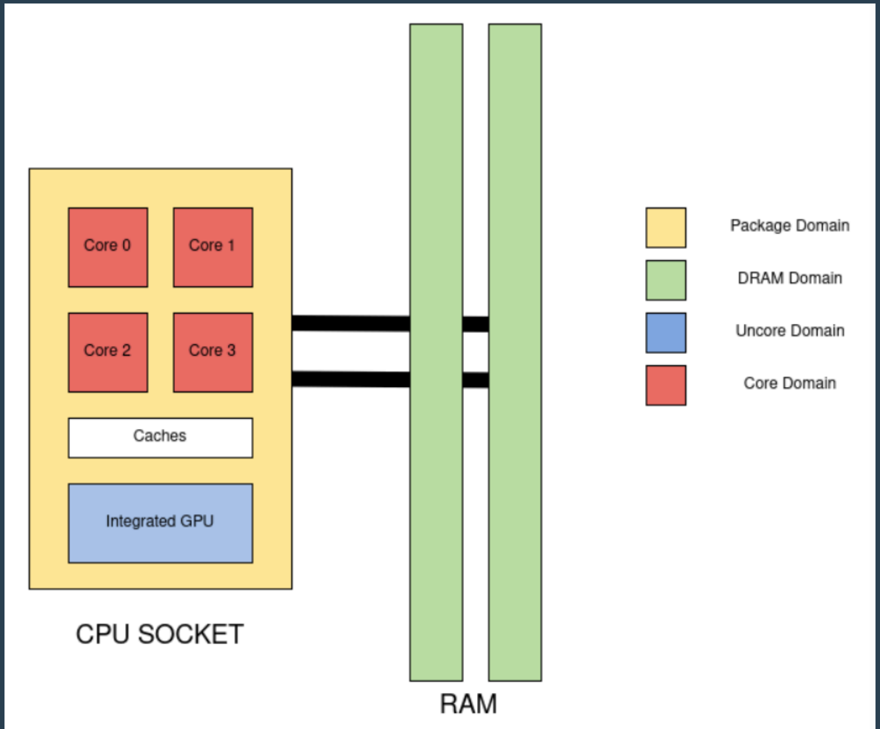
SDIA SUSTAINABLE DIGITAL INFRASTRUCTURE ALLIANCE

Running software uses energy

To make software more sustainable we need to make its consumption visible and use less energy.

How do we measure energy?

Two easy methods: Wall-Plug vs. Hardware/Software-Interfaces



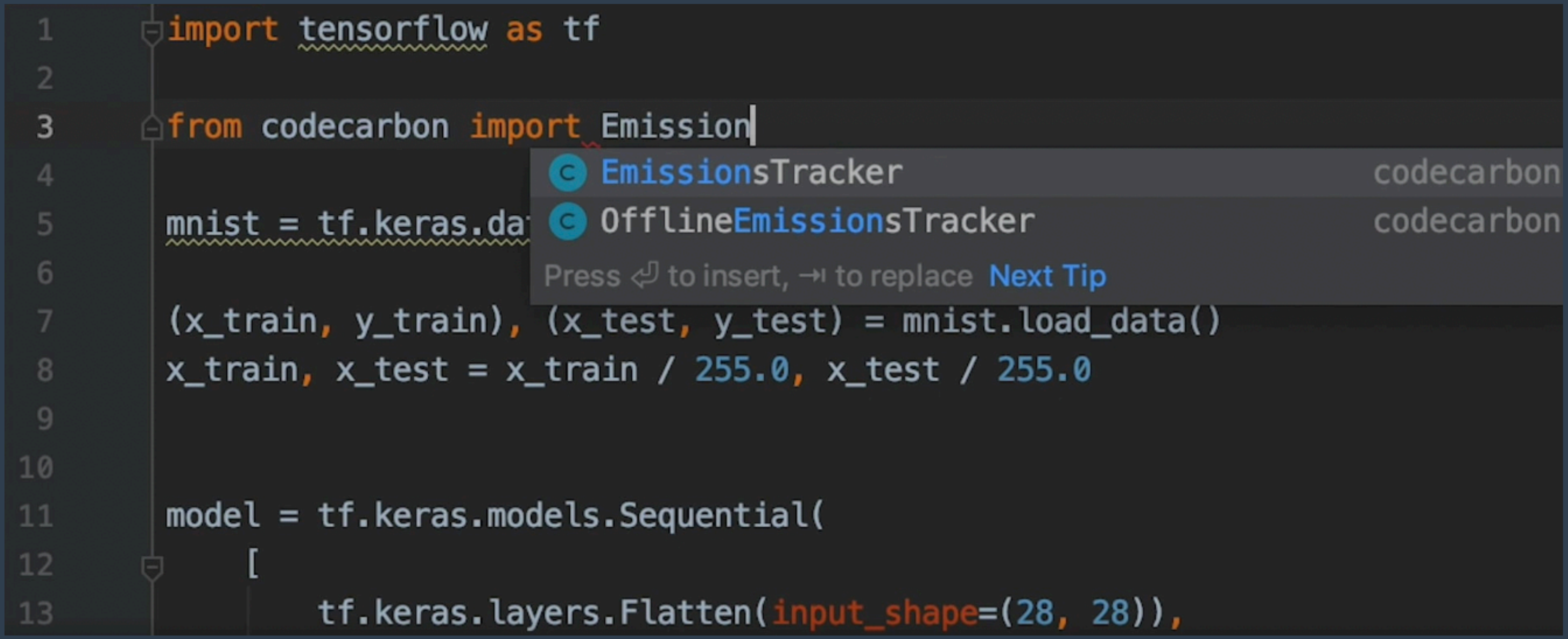
Intel RAPL
Source: https://pyjoules.readthedocs.io/en/stable/devices/intel_cpu.html

Wall-Plug power meter

Code level power meter

codecarbon.io

- Python
- RAPL-based
- NVIDIA GPU support



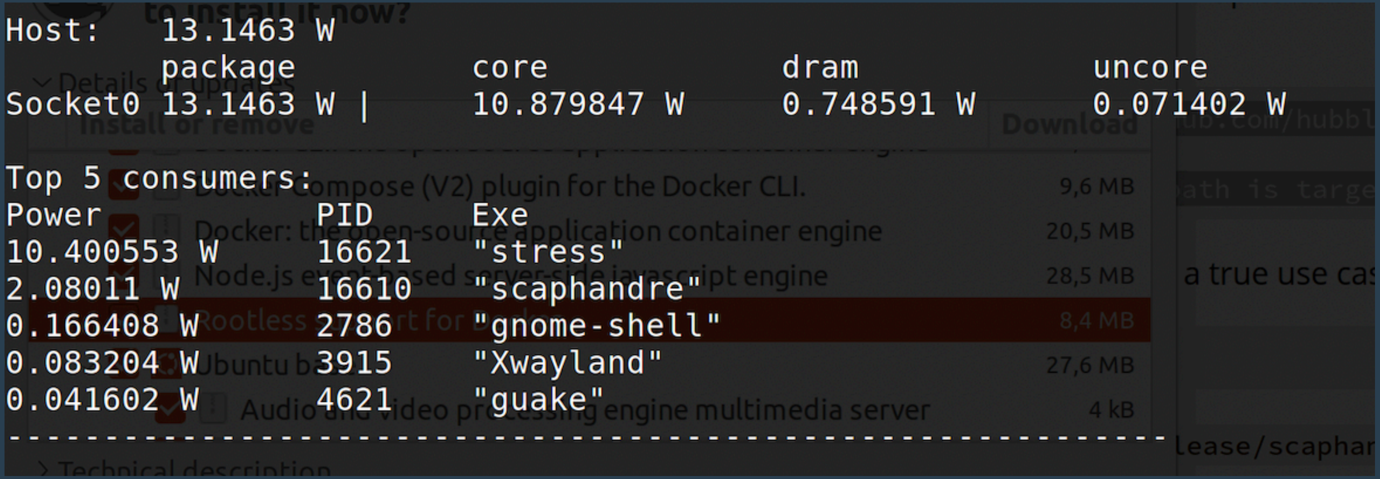
```
import tensorflow as tf

from codecarbon import Emission

mnist = tf.keras.da

(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0


model = tf.keras.models.Sequential(
    [
        tf.keras.layers.Flatten(input_shape=(28, 28)),
```

EmissionsTracker codecarbon
OfflineEmissionsTracker codecarbon
Press ⏎ to insert, ⇥ to replace Next Tip

Scaphandre - Hubblo

open-source RAPL based command line tool

- Neat feature: Can split by process

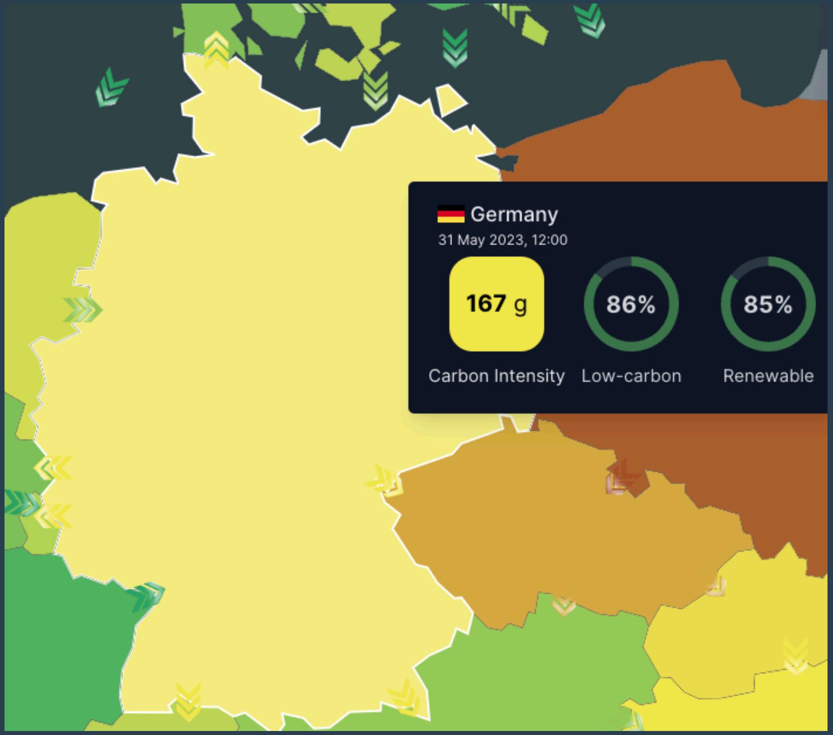


```
Host:   13.1463 W
        package         core            dram            uncore
Socket0 13.1463 W |     10.879847 W     0.748591 W      0.071402 W

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"
------------------------------------------------------------
```

Getting from energy to CO2

Using grid emission factors

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



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

GREEN CODING;

Running software uses energy

To make software more sustainable we need to make its consumption visible and use less energy.

... but is that all that we have to take into account?

Running software uses energy <u>and needs hardware</u>

To make software more sustainable we need to make its consumption visible and use less energy.

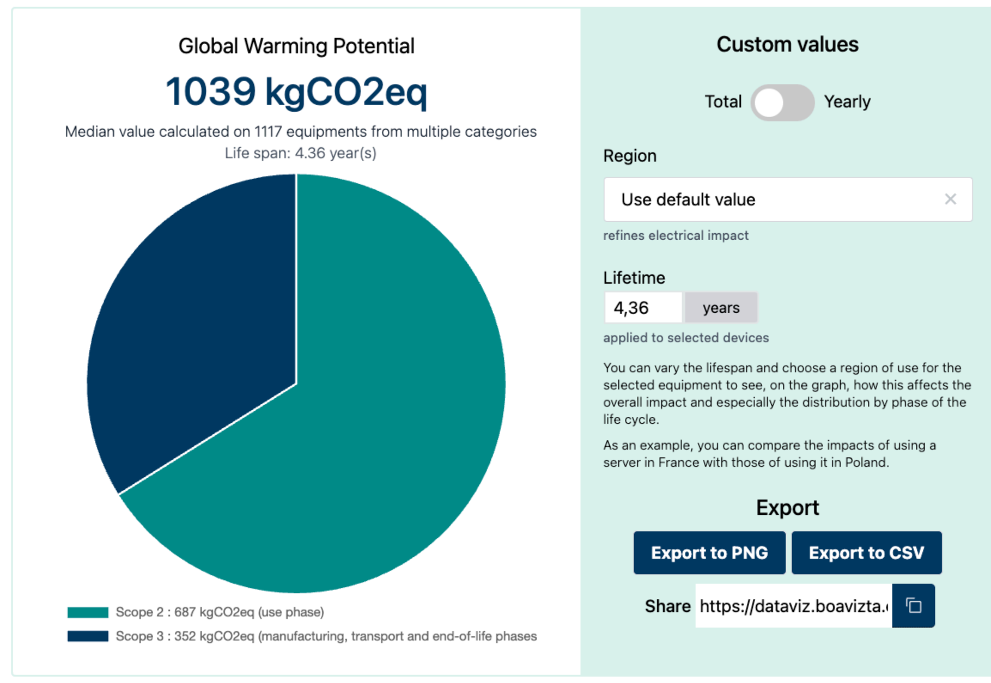
... but is that all that we have to take into account?

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



Global Warming Potential

1039 kgCO2eq

Median value calculated on 1117 equipments from multiple categories
Life span: 4.36 year(s)

🟩 Scope 2 : 687 kgCO2eq (use phase)
🟦 Scope 3 : 352 kgCO2eq (manufacturing, transport and end-of-life phases

Custom values

Total ○ Yearly

Region: Use default value

refines electrical impact

Lifetime: 4,36 years

applied to selected devices

You can vary the lifespan and choose a region of use for the selected equipment to see, on the graph, how this affects the overall impact and especially the distribution by phase of the life cycle.

As an example, you can compare the impacts of using a server in France with those of using it in Poland.

Export

Share https://dataviz.boavizta.

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

GREEN CODING;

Running software uses energy, needs hardware,

To make software more sustainable we need to make its consumption visible and use less energy.

... but is that all that we have to take into account?

... but wait ... is that REALLY all we have to take into account?

Running software uses energy,
needs hardware,
has to be developed, tested and deployed

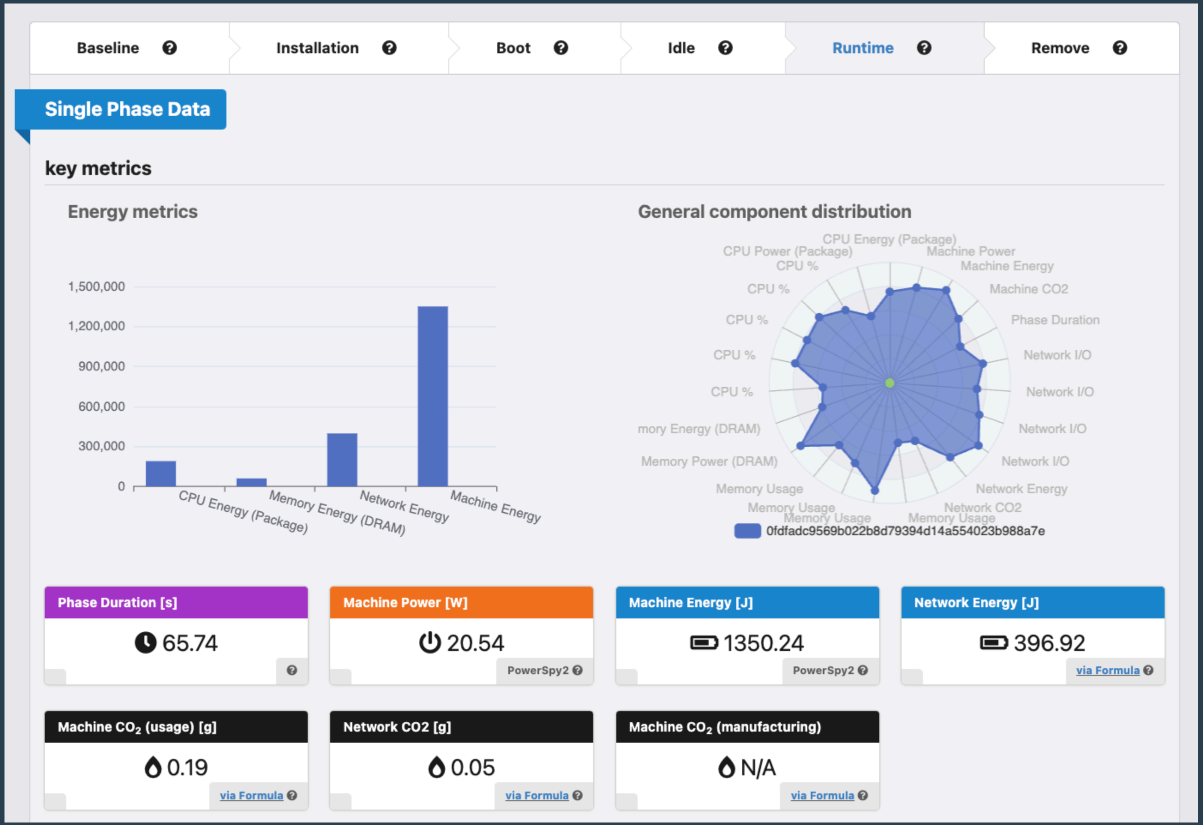
To make software more sustainable we need to make its consumption visible and use less energy.

... but is that all that we have to take into account?

... but wait ... is that REALLY all we have to take into account?

Green Metrics Tool

Understanding energy throughout different software runtime phases



Phase Duration [s]	Machine Power [W]	Machine Energy [J]	Network Energy [J]
65.74	20.54	1350.24	396.92
	PowerSpy2	PowerSpy2	via Formula

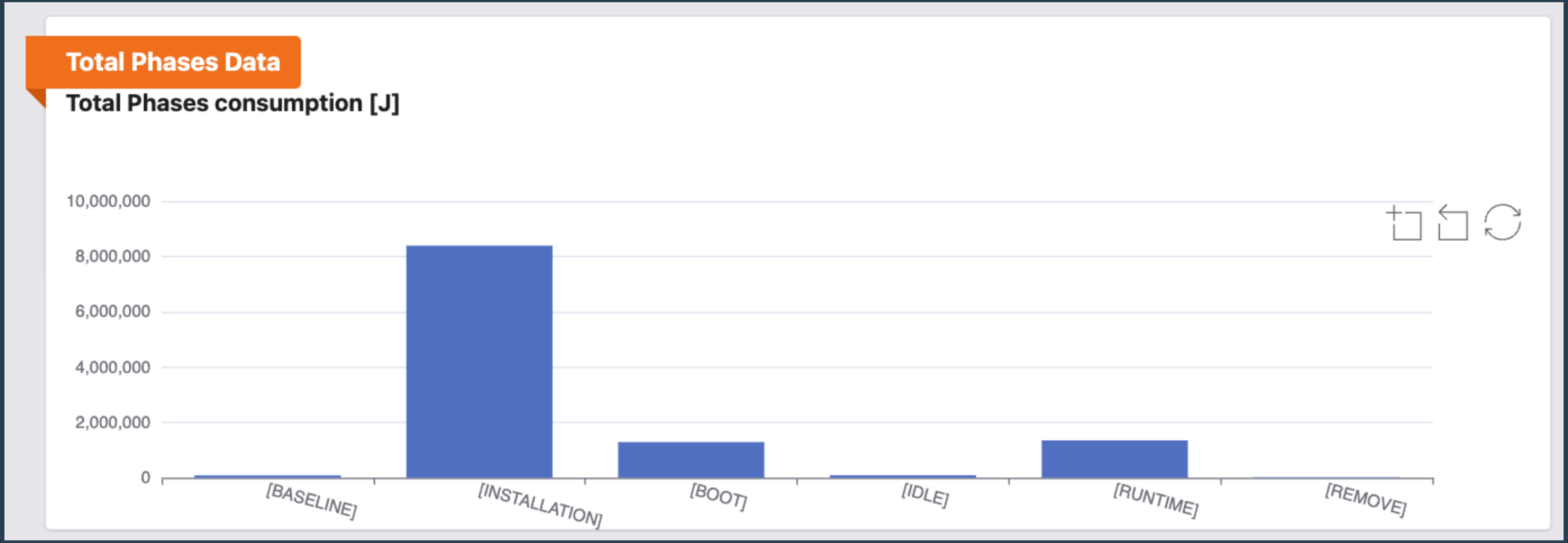
Machine CO_2 (usage) [g]	Network CO2 [g]	Machine CO_2 (manufacturing)
0.19	0.05	N/A
via Formula	via Formula	via Formula

Source: https://metrics.green-coding.berlin/

GREEN CODING;

Green Metrics Tool

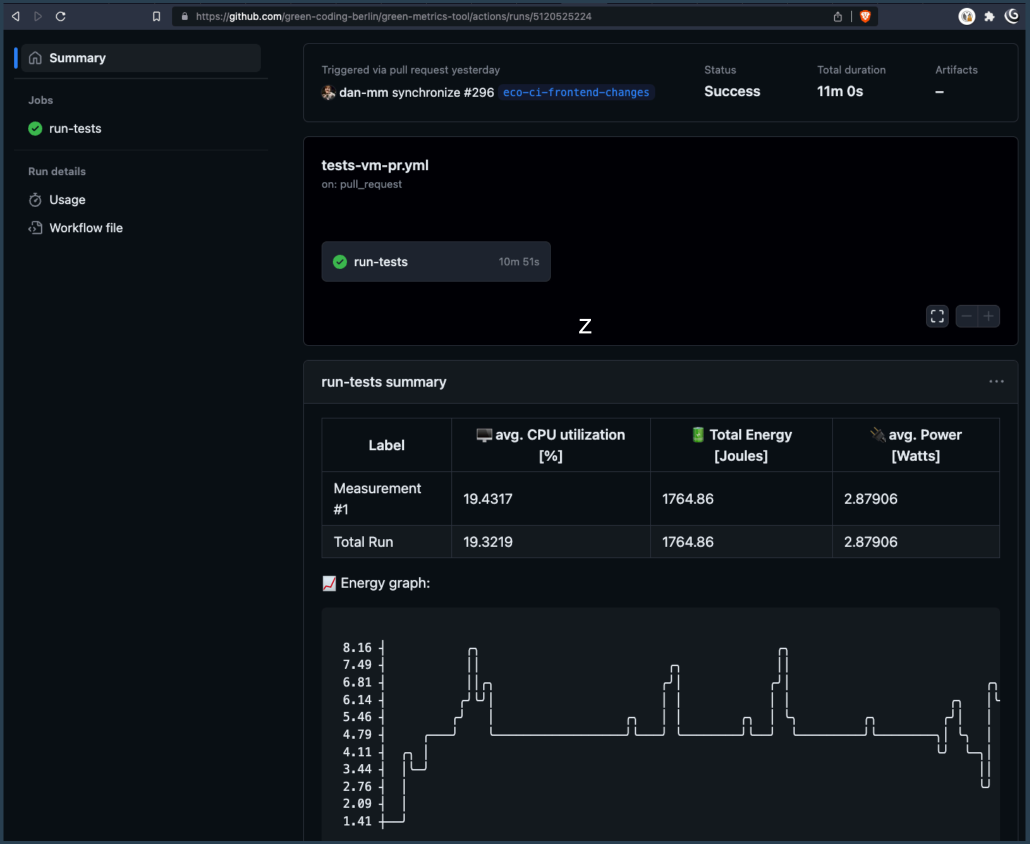
Understanding energy throughout different software runtime phases



Source: https://metrics.green-coding.berlin/

Eco-CI

Estimating CI/CD pipeline energy



Triggered via pull request yesterday
dan-mm synchronize #296 `eco-ci-frontend-changes`

Status: Success

Total duration: 11m 0s

Artifacts: –

tests-vm-pr.yml
on: pull_request

run-tests summary

Label	🖥️avg. CPU utilization [%]	🔋Total Energy [Joules]	🔌avg. Power [Watts]
Measurement #1	19.4317	1764.86	2.87906
Total Run	19.3219	1764.86	2.87906

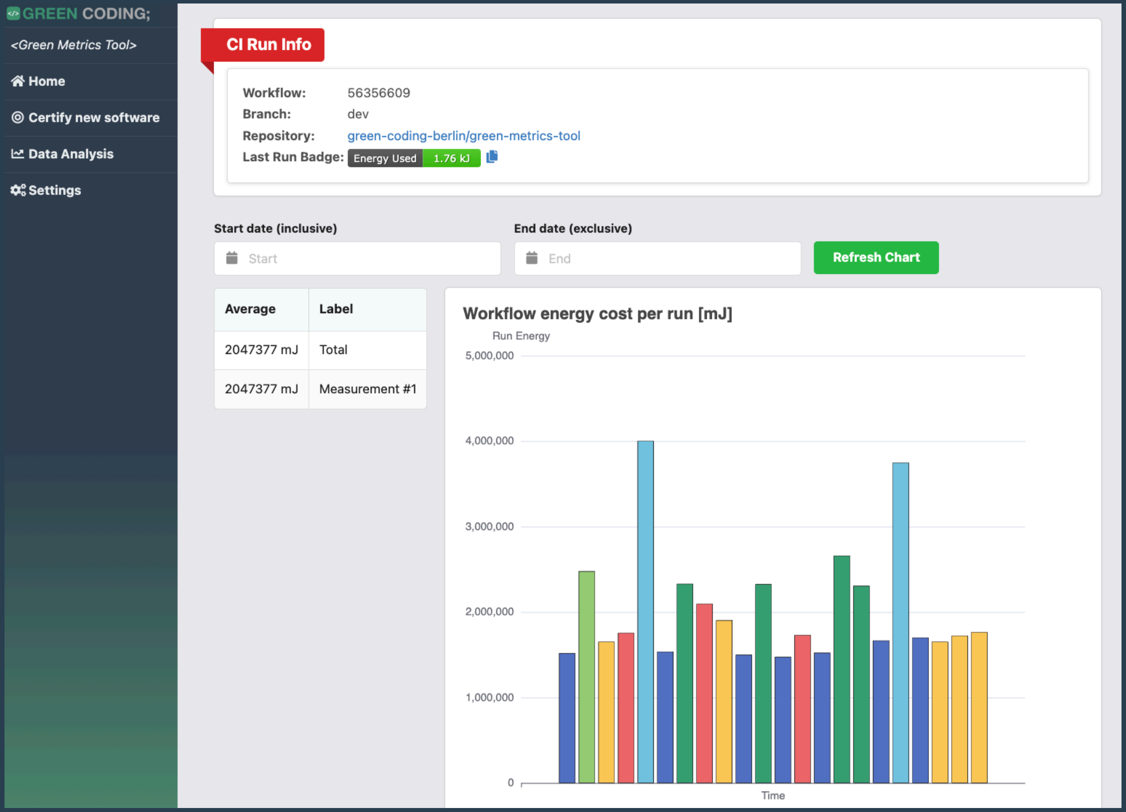
📈 Energy graph:

Source: https://github.com/green-coding-berlin/green-metrics-tool/actions/runs/5120525224

</> GREEN CODING;

Eco-CI

Estimating CI/CD pipeline energy



Workflow: 56356609
Branch: dev
Repository: green-coding-berlin/green-metrics-tool
Last Run Badge: Energy Used 1.76 kJ

Average	Label
2047377 mJ	Total
2047377 mJ	Measurement #1

Source: https://metrics.green-coding.berlin/ci.html?repo=green-coding-berlin%2Fgreen-metrics-tool&branch=dev&workflow=56356609

Summary

- This was a whirlwind tour through the most used concepts of evaluating software carbon emissions.
- There are methodologies out there that also incorporate data-center costs, coolings costs etc.
- What we need the most at the moment is transparency and data.

- To start a paradigm shift: Measure your software and communicate it ❤️

Thank you! Time for Q&A

Follow green-coding.berlin

- Website / Blog / Newsletter: https://www.green-coding.berlin
- Demo Open Data Repository: https://metrics.green-coding.berlin

- Unsere Projekte: https://www.green-coding.berlin/#projects
- Unsere Case-Studies: https://www.green-coding.berlin/case-studies

- Meetup Gruppe (Berlin): https://www.meetup.com/green-coding

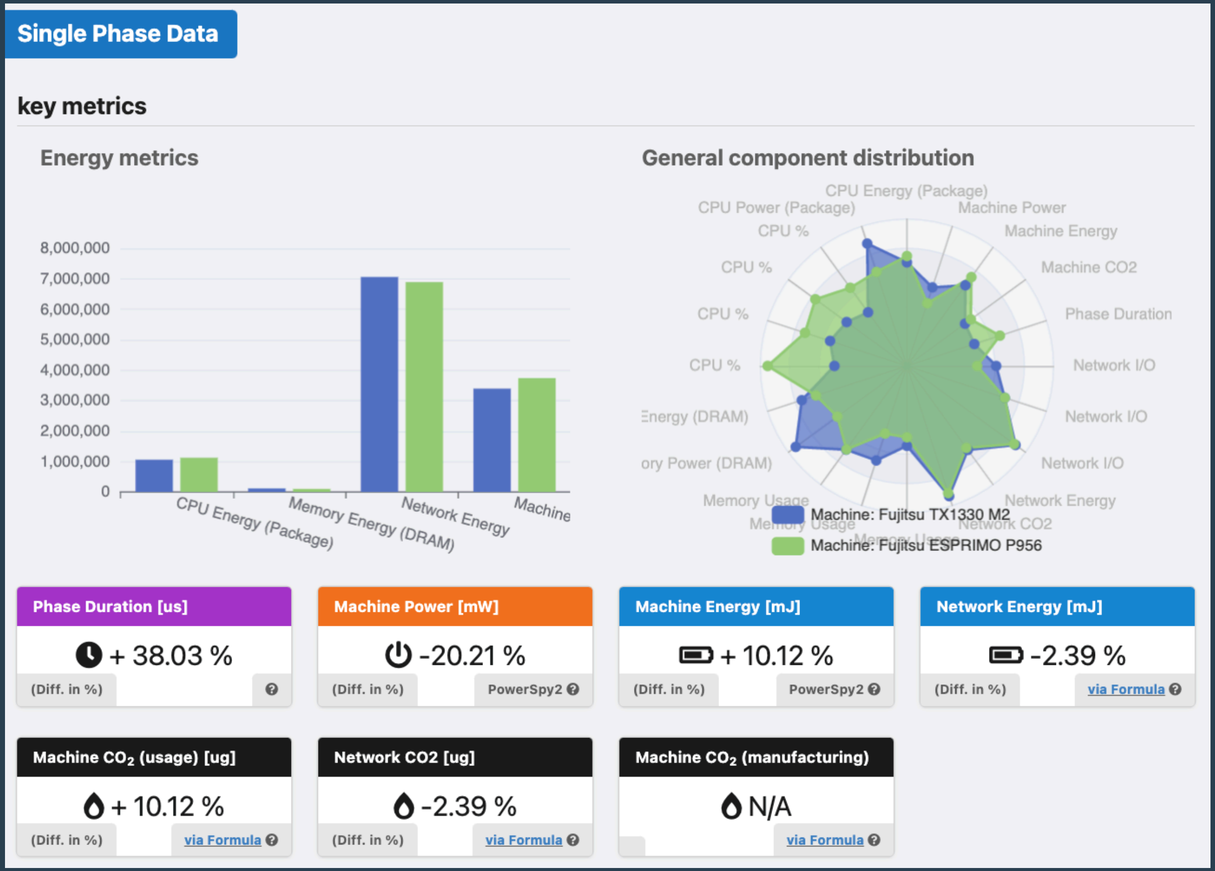
- https://www.linkedin.com/in/arne-tarara / arne@green-coding.berlin

Backup Slides

Here be dragons ...

Green Metrics Tool

Statistical comparisons and API included



Single Phase Data

key metrics

Phase Duration [us]
+ 38.03 %
(Diff. in %)

Machine Power [mW]
-20.21 %
(Diff. in %) PowerSpy2

Machine Energy [mJ]
+ 10.12 %
(Diff. in %) PowerSpy2

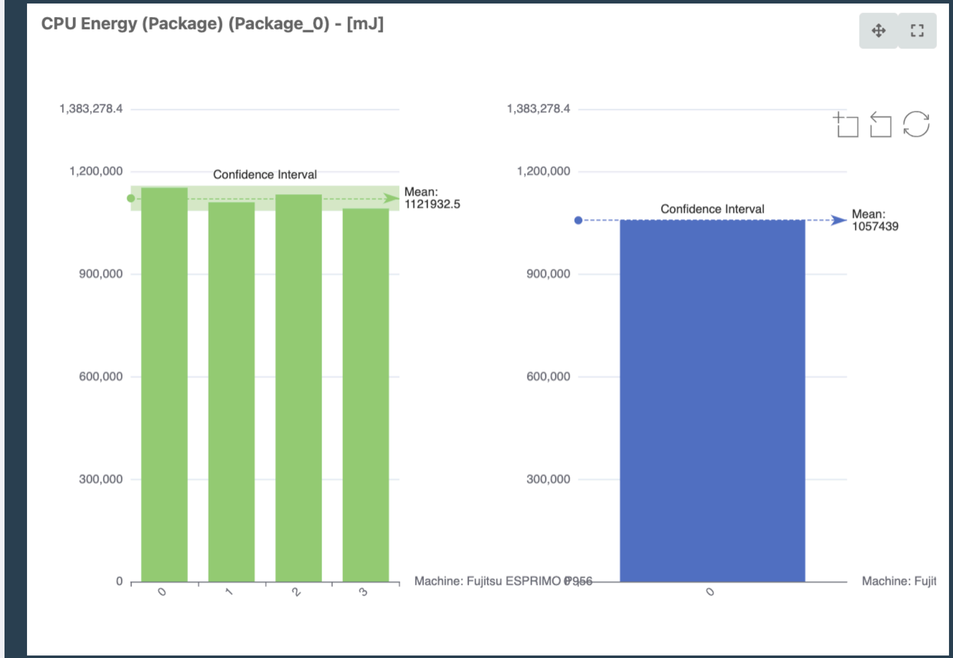
Network Energy [mJ]
-2.39 %
(Diff. in %) via Formula

Machine CO$_2$ (usage) [ug]
+ 10.12 %
(Diff. in %) via Formula

Network CO2 [ug]
-2.39 %
(Diff. in %) via Formula

Machine CO$_2$ (manufacturing)
N/A
via Formula

Source: https://metrics.green-coding.berlin



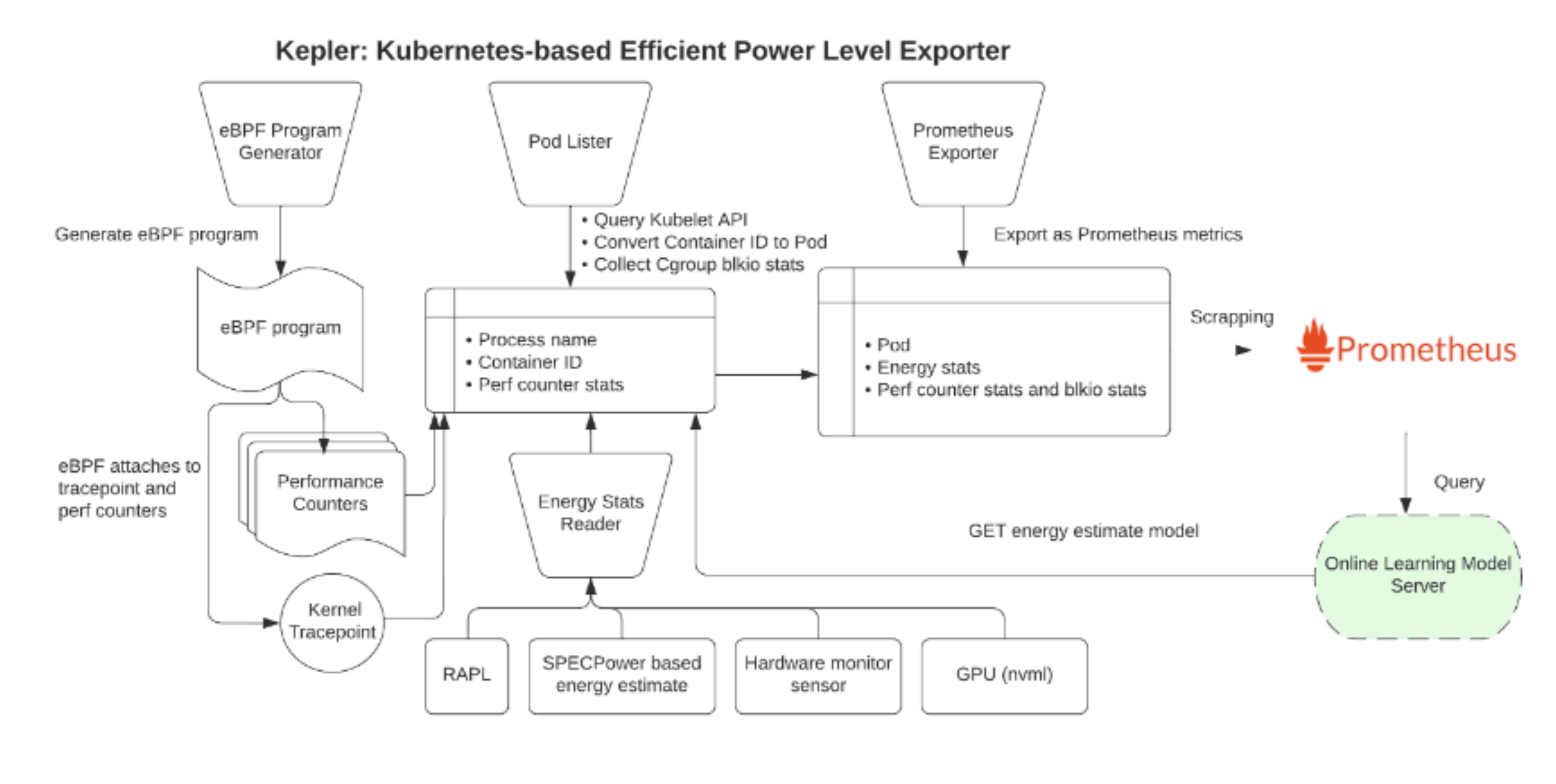
CPU Energy (Package) (Package_0) - [mJ]

Source: https://metrics.green-coding.berlin

</> GREEN CODING;

Distributed Environments / Clusters

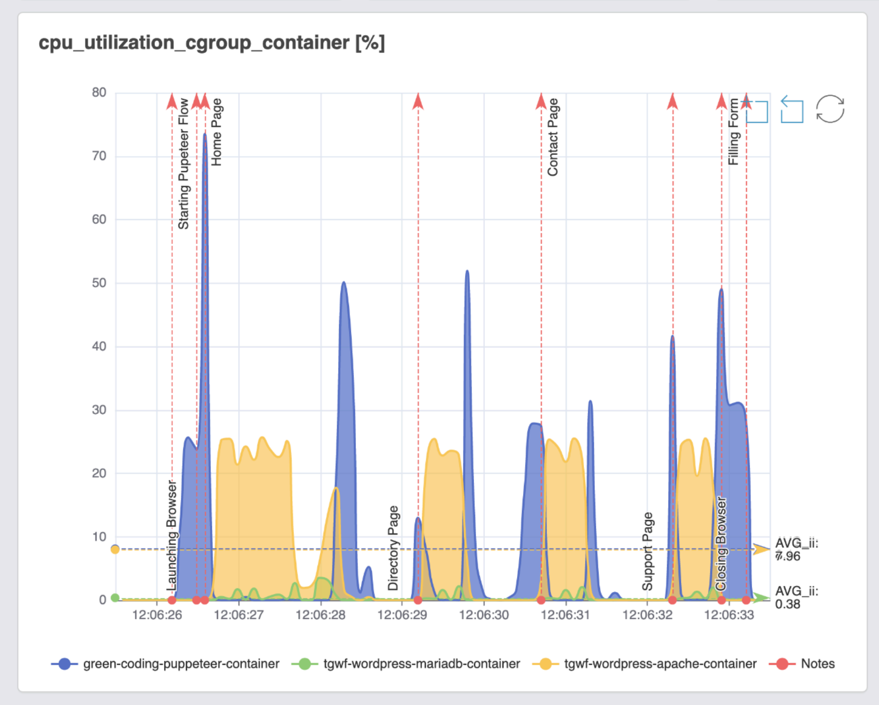
Introducing Kepler


Kepler: Kubernetes-based Efficient Power Level Exporter
eBPF Program Generator
Pod Lister
Prometheus Exporter
Generate eBPF program
• Query Kubelet API
• Convert Container ID to Pod
• Collect Cgroup blkio stats
Export as Prometheus metrics
eBPF program
• Process name
• Container ID
• Perf counter stats
• Pod
• Energy stats
• Perf counter stats and blkio stats
Scrapping
Prometheus
eBPF attaches to tracepoint and perf counters
Performance Counters
Kernel Tracepoint
Energy Stats Reader
Query
GET energy estimate model
Online Learning Model Server
RAPL
SPECPower based energy estimate
Hardware monitor sensor
GPU (nvml)

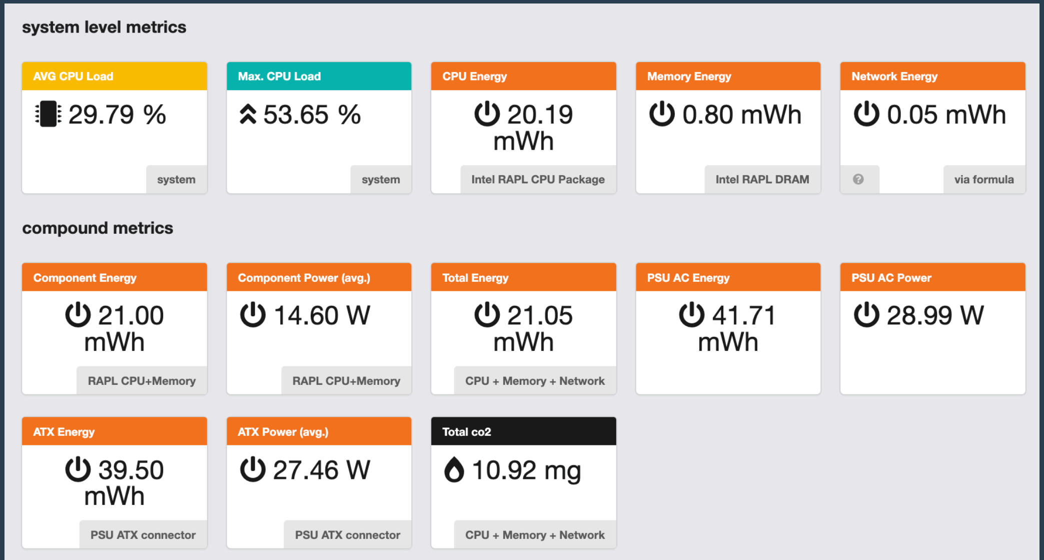
http://sustainable-computing.io

Green Metrics Tool

All-in-One solution for benchmarking, orchestration and transparency



cpu_utilization_cgroup_container [%]



system level metrics

AVG CPU Load
29.79 %
system

Max. CPU Load
53.65 %
system

CPU Energy
20.19 mWh
Intel RAPL CPU Package

Memory Energy
0.80 mWh
Intel RAPL DRAM

Network Energy
0.05 mWh
via formula

compound metrics

Component Energy
21.00 mWh
RAPL CPU+Memory

Component Power (avg.)
14.60 W
RAPL CPU+Memory

Total Energy
21.05 mWh
CPU + Memory + Network

PSU AC Energy
41.71 mWh

PSU AC Power
28.99 W

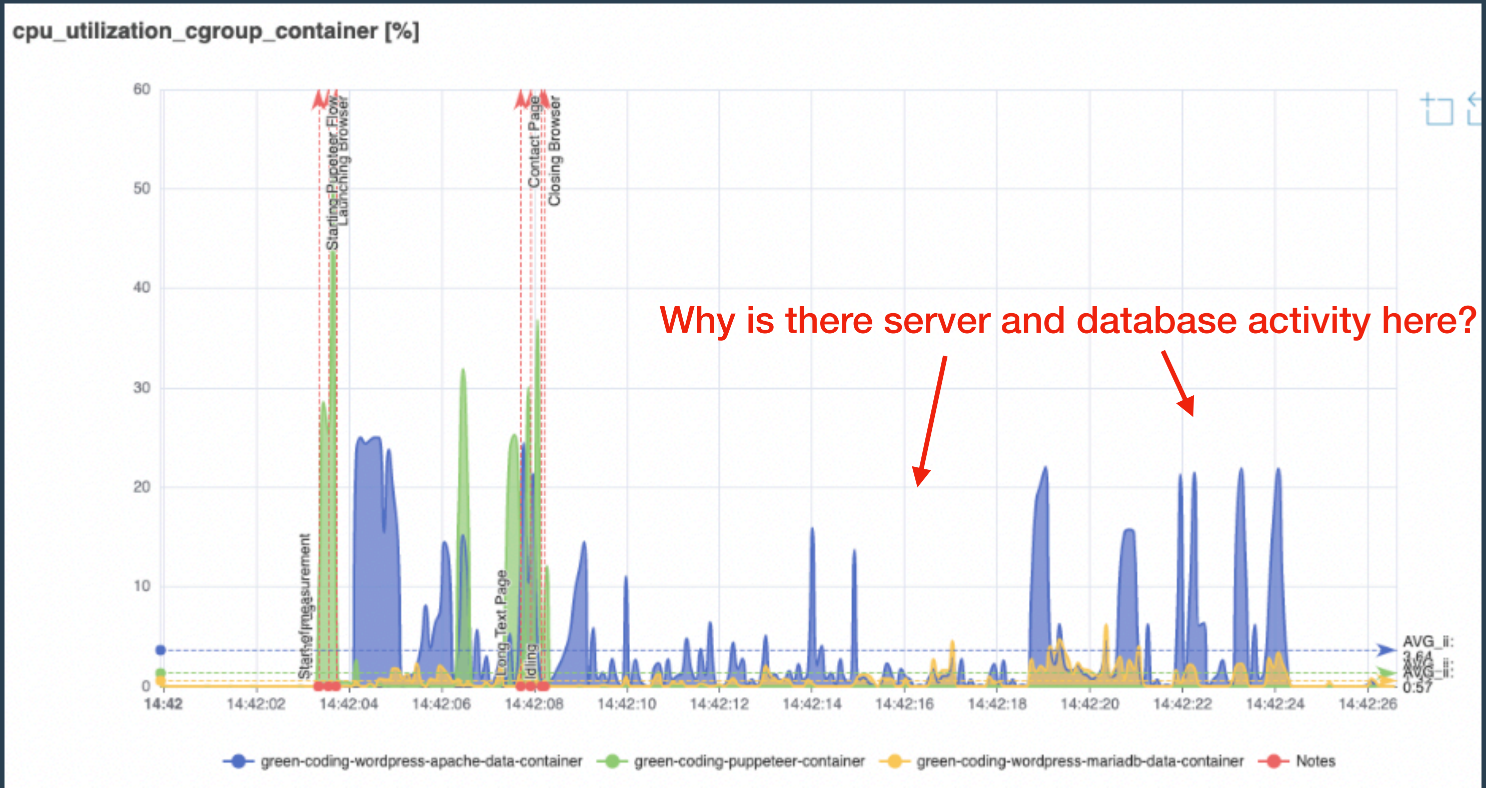
ATX Energy
39.50 mWh
PSU ATX connector

ATX Power (avg.)
27.46 W
PSU ATX connector

Total co2
10.92 mg
CPU + Memory + Network

Green Metrics Tool

Container native; For detailed inspection of architecture

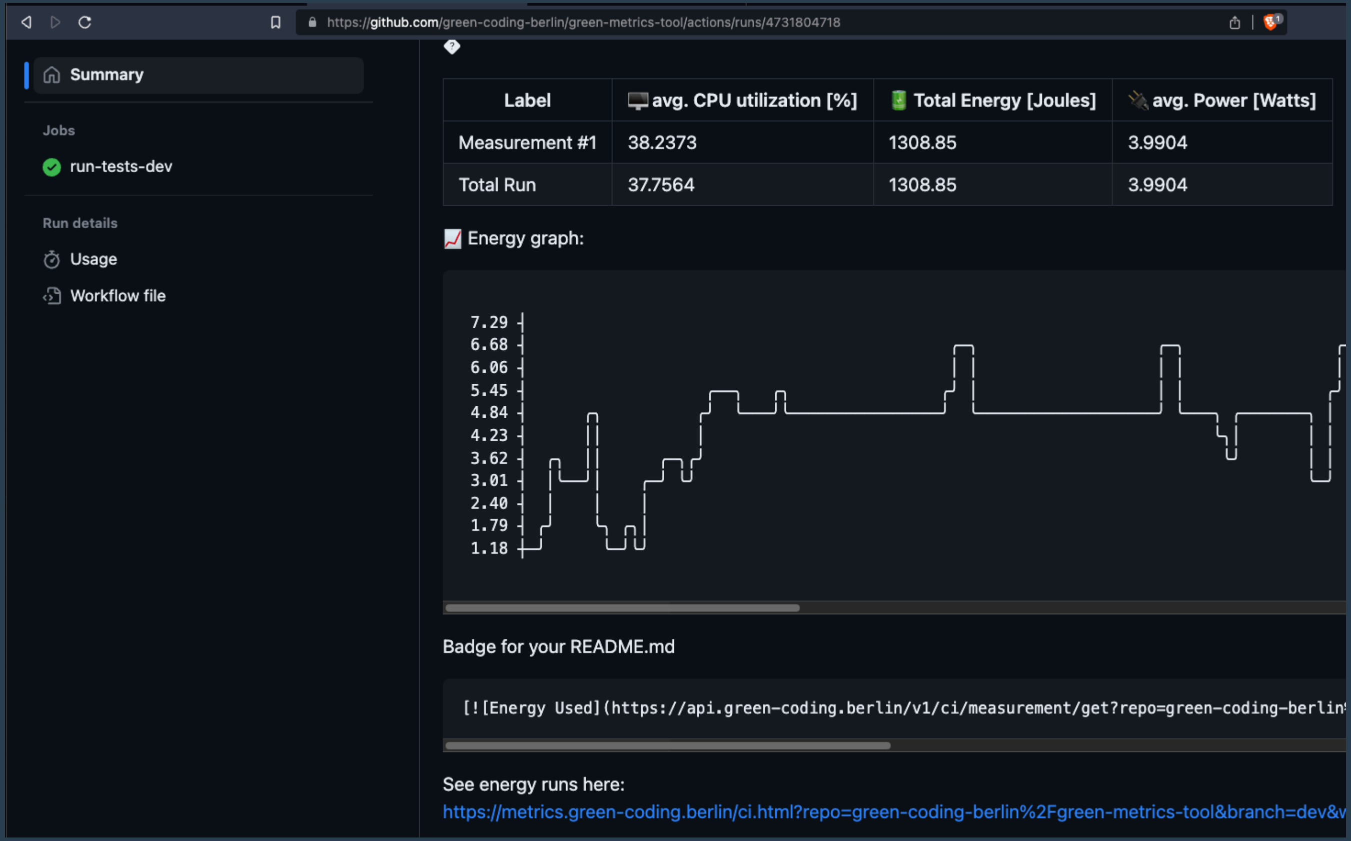


Why is there server and database activity here?

Use-Case mit Wordpress / Django Community

Eco CI Tools - Green Coding Berlin

Energy inside of a CI / CD Pipeline



https://github.com/green-coding-berlin/green-metrics-tool/actions/runs/4731804718
Summary
Jobs
run-tests-dev
Run details
Usage
Workflow file

Label	🖥️avg. CPU utilization [%]	🔋 Total Energy [Joules]	🔌avg. Power [Watts]
Measurement #1	38.2373	1308.85	3.9904
Total Run	37.7564	1308.85	3.9904

📈 Energy graph:

```
7.29 ┤
6.68 ┤
6.06 ┤
5.45 ┤
4.84 ┤
4.23 ┤
3.62 ┤
3.01 ┤
2.40 ┤
1.79 ┤
1.18 ┼
```

Badge for your README.md

```
[![Energy Used](https://api.green-coding.berlin/v1/ci/measurement/get?repo=green-coding-berlin
```

See energy runs here:
https://metrics.green-coding.berlin/ci.html?repo=green-coding-berlin%2Fgreen-metrics-tool&branch=dev&w

Eco-CI Tools on Github - Integrated into Actions directly

</> GREEN CODING;

Weitere Tools ...

for questions regarding a specific tool, please ask in the Q&A!

- powertop
- powermetrics
- Cloud Carbon Footprint
- turbostat
- powerJoularX
- PAPI
- greenframe.io
- ...

Best Practices

According to the Blue Angel for Software

- Measure your system
- Go for reusability
- Go for exportability / interoperability
- Code must also run on older hardware
  - To fight device obsolescence (within reason)

Best Practices

According to the Green Software Foundation - A suprising mix ....

- **Tips that are well known and not really "green software"**
    - Known from other tools: Cache Static data, Minify JS / CSS, HTTP/2, Block Bots, Adblockers etc. ...
    - ... : Use energy efficient AI models 🤡
    - Common knowledge: Right-Size VMs
    - ...
- **Tips that are debatable**
    - Run AI models at the edge
    - Use sustainable regions -> Depends on the network transfer. Hard to wage
    - Stateless design -> On its own no gain
- Source: https://patterns.greensoftware.foundation/catalog/cloud/match-utilization-requirements-of-vm/

Best Practices?

Even simple questions are hard (impossible) to answer

- **Is email more sustainable than paper?**
  - Paper consumes a fixed amount. Email has pot. infinite storage and processing
- **Is Serverless more sustainable than classic VMs?**
  - No solid data on this (Deno / Isolates / Firecracker)
  - Cloudflare / Amazon did decline when asking for sustainability insights
- **Is using AWS Gravitron more sustainable than Intel**
  - What happens to electronic waste?! Life-Cycle ...?
- **Is Python more sustainable than Rust?**
  - Python uses 80-times the Instructions where as Rust uses 1-3. Still people are not changing because of "cost of development" etc. ....

Problems

with current "sustainability" best practices

- **Often they just employ common sense**
  - Like**:** "run less tests"
- **Often generic advice**
  - "Use sustainable libraries"
- **Often can be harmful also**
  - Switching machines brings electronic waste
- **Hard to generalize at the moment**
  - Because we have no data
- **Can contain pitfalls**
  Like: "Less runtime is always better" ... **no**: TurboBoost, HyperThreading etc. may inverse result

- We plead for a simpler solution ....

Best Practices - as backend engineer

An approach for a framework atm.

1. **Quantify** your system!
   How much am I using for what service (Green Metrics Tool, Scaphandre, Kepler etc.)
   ASK your service provider for values you don't know.

2. **Green Energy** (-> Green Web Foundation)
   Constantly / Location-Shifting / Time-Shifting - But not on the moon please!

- TODO: Dark mode -> Yes!
- Turn Video resolution down
- Turn machines off
- Right Size VMs
- Reduce Backups, Logs etc.
- Turbo Boost off!
- Hyper Threading usually on

3. **Idle machines** (Async / Polling / Microservices)
   Only makes sense if you can really turn machines off

4. **Architecture Overhead** (Backups, Logs, Redundancy, Over-Provisioning of Services / Machines)

5. **Language Overhead** (C-Extensions, WebAssembly, Language-Swaps)

6. **Code Changes**
   Even 99% Code runtime reduction is of limited help if the machine then idles

Best Practices - as user TODO

An approach for a framework atm.

1. **Quantify** your system!
   Get a power meter. User RAPL
   ASK your service provider for values you don't know.

2. **Green Energy** (-> Switch energy provider)
   Constantly

3. **Idle machines** (Turn your box and monitor off)

4. **Architecture Overhead** (Backups, Logs, Redundancy, Over-Provisioning of Services / Machines)

5. **Language Overhead** (C-Extensions, WebAssembly, Language-Swaps)

6. **Code Changes**
   Even 99% Code runtime reduction is of limited help if the machine then idles

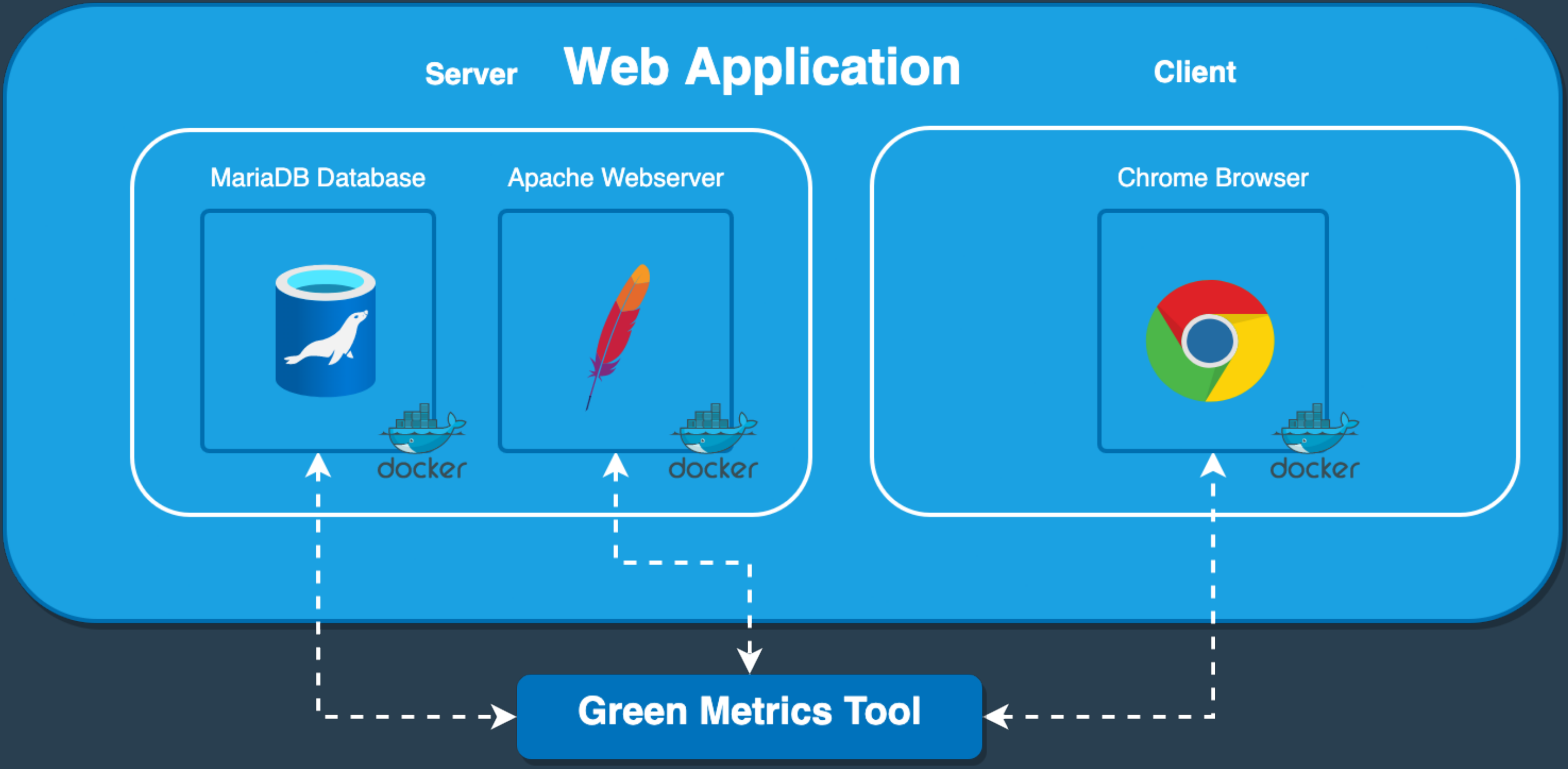
Best Practices

Some general guidance

- **No fancyness until proven helpful**
- **Use less resources -> Yes, common sense!**
    - Services, Machines etc.
- **No off-loading / Backlashes**
    - We are more sustainable, we now use a data provider for that machine!
    - We now use 5 GB less memory (and calculate everything on request :( )
- **General Question: If everybody would do it this way. Would we have enough resources?**
    - Simple chat-bot => ChatGPT
    - Portfolio Website: VPS with Wordpress and database

Green Metrics Tool - Schaubild

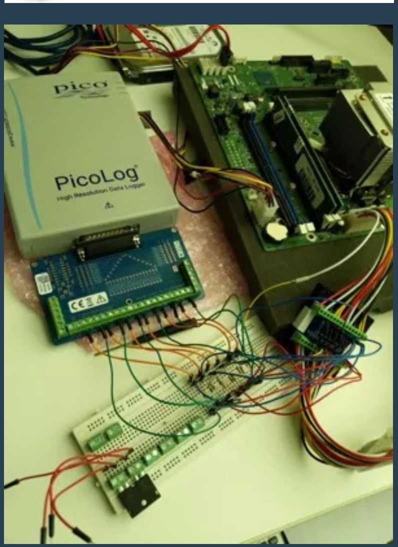
Container Aufbau für Client-Server Apps



Green Metrics Tool - Hardware Setup

Mess-Sensoren im Green Metrics Tool

- AC / DC Energie
  - IPMI
  - GUDE (Blauer Engel)
  - PowerSpy2 - Consumer Grade Oscilloscope
  - Custom Mainboard Connector for Fujitsu ESPRIMO
- CPU % / CPU-Frequenz
- Temperatur
- CPU / DRAM Energie
  - RAPL
- Netzwerk Datenverkehr / Energie
- Container-Metriken
- ... weitere Reporter als Open-Source Plugin-In möglich



GREEN CODING;

Green Metrics Tool - Cluster Setup

Aktuelle Rechner im Green Metrics Tool Cluster

- Fujitsu ESPRIMO P956 - Blue Angel compatible (Ubuntu)
- Fujitsu TX1330 M2 - Single-Tenant Server (Ubuntu)
- Quanta Leopard - Multi-Tenant Server - SoftAWERE compatible (Ubuntu)
- Intel Mac 13" Q3-2015
- M1 Mac 13" Q1-2022

Green Metrics Tool - Supported Software

Software Kategorien und OS

- **Server / Client Anwendungen**
  - Beispiele Nextcloud und Django CMS
- **CLI Anwendungen**
  - Beispiele curl oder ML Anwendungen (sklearn, pytorch)
- **Desktop Apps**
  - Beispiele: Google Chrome, Firefox etc.
- **OS**
  - macOS
  - Linux (Ubuntu & Fedora)
  - Windows (nur Gesamtleistung an der Steckdose)