Making software sustainable Through measuring and optimizing energy & emissions



Who am **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





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 :)



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"
- gCO2 per run when done in Paris"

"My Gitlab pipelines uses XX gCO2 per run when run in Berlin, but only YY





The goal To talk about software in a comparative way

ooo Nextcloud	ſ	WordPress
NextcloudCMSclick for details	WagtailCMSclick for details	Wordpress Blog/CMS click for details
BADGES	BADGES	BADGES
Energy Cost4.89 kJ via PSU (AC)Energy Cost1.94 kJ via RAPLSCI162.64 mgCO2e/Talk message	Energy Cost1.14 kJ via PSU (AC)Energy Cost243.24 J via RAPLSCI18.84 mgCO2e/page request	Energy Cost635.62 J via PSU (AC)Energy Cost112.62 J via RAPLSCI25.08 mgCO2e/page request
C Show measurements	C Show measurements	Show measurements



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
- Serverless hosting allows the server to switch off between requests



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

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 <u>beta in GMT</u>

eneral	Measurement	Badges	Machine	Usage Scenario	Logs	Network	Optimizations	17	
⁻ his is a lis letails. If y	t of potential optim ou think a warning	nizations we h does not app	nave detected oly to you, you	based off your run. V can add an ignore in	Ve are cons to your usa	tantly adding ge_sceanric	new features so plea o.yml file.	se check our docum	nentation for r
Docke The cont	r container bo tainer takes very lo	ot time ng to become	e usable ~ 456	s. Containers are me	ant to be st	arted and kille	ed quite fast.	Docker	docker-boo
Cpu co Containe	ontainer resou er 'app' is maybe ov	rce allocat	t ion d. CPU utilizat	ion was '2%'. Max wa	as '31%'.			Jocker d	ocker-cpu-allo
Low IP The ipc o	C count count is 1. This is a	sign that you	are not using	all features of the pr	ocessor.			. c	pu low-ipc
High p Your sys	age faults tem is experiencing	g a lot of page	e faults. See do	ocumentation for a d	etailed deso	cription on how	w to maybe fix this.	memory	high-page
Code v This piec	with high optin	nization po ed a 18/100 by	otential fou our Green Co	I nd oding AI. We think thi	s can be op	timized		4 Ai	ai-low-code-
object_	_specification.	py:473:ret	urn [line.s	strip().split(',	') for li	ne in open((file_path, 'r')]	
Energy	/ hotspots		(In the second sec	Cas desumantation	an haw to r	un vour oodo	with full profiling one	Br Benchmark	energy-ho





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 Performance to Power Ratio 5,000 7,500 10,000 12,500 2,500 11,935 overall ssj_ops/watt 100% - 13,592 90% - 13,584 80% - 13,651 70% - 13,548 Load 60% - 13,127 Target 50% - 12,399 40% - 11,277 30% 9,747 20% - 7,576 10% - 4,543 Active ldle 1,000 250 500 750 1,250 Average Active Power (W)

50% Power increase at 70% utilization



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

Hewlett Packard Enterprise ProLiant DL110 Gen10 Plus

Idle optimized



QuantaGrid D43K-1U (2022)





Measuring total machine energy ATX / DC vs. AC-Measurements







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: <u>https://app.electricitymaps.com/zone/DE</u>



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

ooo Nextcloud	ſ	WordPress
NextcloudCMSclick for details	WagtailCMSclick for details	Wordpress Blog/CMS click for details
BADGES	BADGES	BADGES
Energy Cost4.89 kJ via PSU (AC)Energy Cost1.94 kJ via RAPLSCI162.64 mgCO2e/Talk message	Energy Cost1.14 kJ via PSU (AC)Energy Cost243.24 J via RAPLSCI18.84 mgCO2e/page request	Energy Cost635.62 J via PSU (AC)Energy Cost112.62 J via RAPLSCI25.08 mgCO2e/page request
C Show measurements	C Show measurements	Show measurements



The tools Screenshots! (In case something breaks 🚳)



First, let's fire up a Github Codespace It takes a while to install ...

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



Branch	ළ codespace
This branch will be checked out on creation	
Dev container configuration	Werkehe
Your codespace will use this configuration	worksho
Region	Europe Wes
Your codespace will run in the selected region	
Machine type	2-cor
Resources for your codespace	
	Create codespa





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 <u>https://github.com/green-coding-solutions/spec-power-model</u>









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/
 - <u>https://www.green-coding.berlin/case-studies/hyper-threading-and-energy/</u>
 - etc.



Scaphandre - Hubblo open-source RAPL based command line tool

• Neat feature: Can split by process

Host: 13.1463 W package Socket0 13.1463 W	core 10.879847 W	dram 0.748591 W	unco 0.07	re 1402 W
Top 5 consumers: Power PID 10.400553 W 16621 2.08011 W 16610 0.166408 W 16610 0.083204 W 3915 0.041602 W 4621	<pre>2) plugin for the Docker (Exe "stress" "scaphandre" "gnome-shell" "Xwayland" "auaka"</pre>	CLI. er engine ngine	9,6 MB 20,5 MB 28,5 MB 8,4 MB 27,6 MB	ath is targe a true use cas
Technical description	pr guake g engine multi	media server	4 kB	lease/scaphar



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-C **Estimation in CI / CD Pipelines**

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

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



Label	avg. CPU utilization [%]	Total Energy [Joules]	🔌 avg. Power [Watts]	Duration [Seconds]
otal Run	6.83923	24.7645	1.7689	27
leasurement	5.99	12.3823	1.7689	8
leep <u>#2</u>	5.72625	12.3823	1.7689	8
2.00 + Watts o	ver time			Ĺ
CO2 Data: y: San Francisco	o, Lat: 37.783, Lon: -122.417 or this location: 208			





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 CODING;



Green Metrics Tool More Features

Single Phase Data

key metrics



Comparisons between Hardware / Repository / Branch etc.



Measurement per Commit - Timeline View

