

The GNOME™ Conference GUADEC

How can I make my project more environmentally friendly?

Philip Withnall

Endless

philip@tecnocode.co.uk



Motivation

Limiting global warming to 1.5 °C is a global priority

Life cycle analysis and products



Figure: Life cycle analysis (public domain)

Carbon intensity of power generation

Power source	Carbon intensity ($\text{gCO}_2\text{e}/\text{kWh}$)
Hydro	4
Wind	12
Nuclear	16
Solar PV	46
Gas	469
Coal	1001
IT average	300

Figure: Rough carbon intensities of power generation[1]

Embodied carbon in software

GNOME could provide carbon labelling for what we produce

Functional unit and system boundary

Functional unit: one dist tarball of a software release

Functional unit and system boundary

System boundary

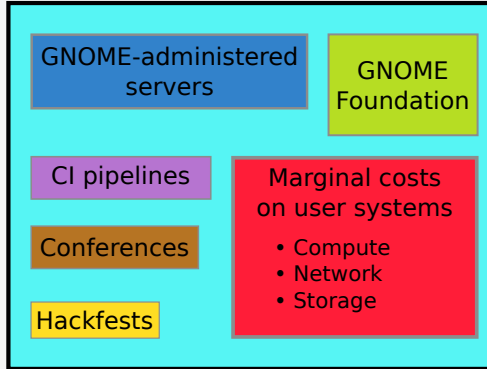


Figure: Proposed system boundary for GNOME

Measuring marginal costs on user systems

- 👉 Use cases
- 👉 sysprof + Builder
- 👉 systemd unit accounting
- 👉 Kernel power state statistics
- 👉 Wattmeter on power supply

Measuring marginal costs on user systems: Use cases

What use cases are you actually solving?

Measuring marginal costs on user systems: sysprof + Builder

```
sysprof-cli -- your-program-here
```

Measuring marginal costs on user systems: sysprof + Builder

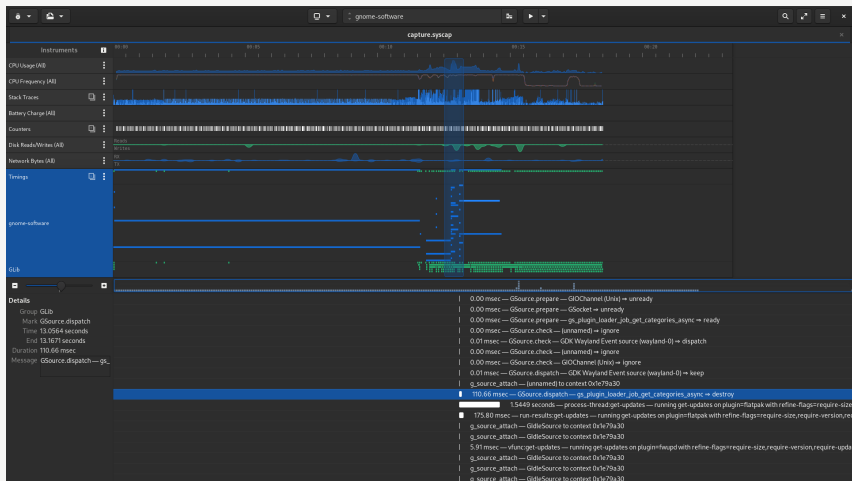


Figure: sysprof results in GNOME Builder

Measuring marginal costs on user systems: systemd unit accounting

```
echo -e "DefaultCPUAccounting=yes\n" \  
        "DefaultIOAccounting=yes\n" \  
        "DefaultIPAccounting=yes" >> \  
        /etc/systemd/system.conf
```

Measuring marginal costs on user systems: systemd unit accounting

```
$ systemctl status geoclue.service
● geoclue.service – Location Lookup Service
   Loaded: loaded (../geoclue.service; ...)
   Active: active (running) since Fri...
 Main PID: 2645 (geoclue)
    IP: 8.1M in, 3.4M out
    IO: 6.0M read, 9.1M written
   Tasks: 4 (limit: 18742)
  Memory: 10.3M
    CPU: 1min 42.217s
  CGroup: /system.slice/geoclue.service
          └─2645 /usr/libexec/geoclue
```

Measuring marginal costs on user systems: kernel power state statistics

```
sudo powertop
```

Measuring CI pipelines

$$N_{\text{pipelines}} \times (\text{pipeline duration} \times 0.114 \text{ kW} \times 300 \text{ gCO}_2\text{e/kW h} + \text{pipeline downloads} \times 17 \text{ gCO}_2\text{e/GB})$$

Measuring the other bits

- 🐾 GNOME-administered servers
- 🐾 GNOME Foundation
- 🐾 Conferences
- 🐾 Hackfests

Measuring the other bits

We're measuring GUADEC (thanks Bartłomiej!)

Improving marginal costs on user systems

- 👉 Where do we want to get to?
- 👉 Be used for less time
- 👉 Do less work; use less network
- 👉 Do work faster; use the network more efficiently
- 👉 Cache better

Improving CI pipelines

- 🐾 Speed up your pipelines (use pre-built Docker images)
- 🐾 Avoid downloads (use pre-built Docker images)
- 🐾 Use shallow clones (see my blog)

Improving the other bits

More measuring to do: Foundation operations, and every time a hackfest is organised

Pulling it all together

- 🐾 GNOME apps should be labelled with their embodied carbon cost: their share of the GNOME project and Foundation overheads, plus their costs for CI and hackfests, in each major release cycle

Pulling it all together

- 🐾 GNOME apps should be labelled with their embodied carbon cost: their share of the GNOME project and Foundation overheads, plus their costs for CI and hackfests, in each major release cycle
- 🐾 We don't have all the data for that yet, but should collect it

Pulling it all together

- 🐾 GNOME apps should be labelled with their embodied carbon cost: their share of the GNOME project and Foundation overheads, plus their costs for CI and hackfests, in each major release cycle
- 🐾 We don't have all the data for that yet, but should collect it
- 🐾 Reduce those embodied costs (optimise CI, make hackfests carbon-neutral)

Pulling it all together

- 🐾 GNOME apps should be labelled with their embodied carbon cost: their share of the GNOME project and Foundation overheads, plus their costs for CI and hackfests, in each major release cycle
- 🐾 We don't have all the data for that yet, but should collect it
- 🐾 Reduce those embodied costs (optimise CI, make hackfests carbon-neutral)
- 🐾 Reduce the marginal costs of your apps (optimise them, and don't waste the user's time)

Open questions

1. What is the power usage of a virtualised server?
2. What is the carbon intensity of our server power supplies?
3. Other life cycle analysis impacts (ozone, eutrophication, water consumption, etc.)
4. How many users do we have??
5. Can we collect better statistics about user systems?

Miscellany

Slide source [https://gitlab.com/pwithnall/
guadec-environmental-presentation-2020](https://gitlab.com/pwithnall/guadec-environmental-presentation-2020)

IPCC SR15 summary <https://www.ipcc.ch/sr15/chapter/spm/>



Creative Commons Attribution-ShareAlike 4.0 International License

Beamer theme: <https://gitlab.gnome.org/GNOME/presentation-templates/tree/master/GUADEC/2020>