Software Life Cycle Assessment (SLCA) in the wild

Geerd-Dietger Hoffmann <didi@green-coding.berlin>
Arne Tarara <arne@green-coding.berlin>
Who am I? / Why listen to me?

- Geerd-Dietger Hoffmann / Didi
- Computer Science M.Sc. from University College London
- Low level Linux/ BSD/ Solaris at IBM and CERN
- Startup CTO: DBook, CigarCities, ClimateFarmers, Ecoworks
- NGO work in West Africa: Ebola response and Polio End Game
- Parental leave
- Part time farmer @ RosselHof 🌱
Who are we? / Why listen to us?

- Green Coding Berlin [https://www.green-coding.berlin/](https://www.green-coding.berlin/)
- Developers of:
  - Green Metrics Tool: precise resource measurements
  - Energy ID: measuring resource usage over time
  - Eco CI: cloud measurements through machine learning model
  - Hog: client side resource usage logging
- "Transparency for software and climate impact" @ Bits & Bäume
Software life cycle assessment

- First Google page only software that does life cycle assessment (next to wikipedia explaining Life-cycle assessment)
- ISO 14040 / ISO 14067
- Most of the values are guesses or "guestimates"
- In this case we are only looking at operational "cost".
Previous work

- Green Cloud Computing (Öko Institute, IZM)
- SDIA (Ecocube)
- Umweltcampus Birkenfeld GREENSOFT model
- etc ...

however no devops tooling exists
Five stages of live cycle assessment

1. Material acquisition and preprocessing
2. Production
3. Distribution and storage
4. Use
5. End of life
The problem with software

Software is never finished
The problem / Just Development

- Local dev environment. Docker, Editors, Linters, Tests
- Code Hosting
- Issues/ Planning
- CI/ CD pipelines
- Dependabot
- Cloud dependencies
- Local compilation of libraries
Solution idea

- Automation !
- Simple. Not a lot of overhead. A tool not a burden.
- See each sprint/ iteration as on cycle of of live cycle assessment
- Understand that software is continuously developed
- Set clear boundaries
  - Is the DNS server part of your software?
  - Encoding library development impact
## Stages

<table>
<thead>
<tr>
<th>Classic</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Material acquisition and preprocessing</td>
<td>Libraries/ Software</td>
</tr>
<tr>
<td>2 Production</td>
<td>Development</td>
</tr>
<tr>
<td>3 Distribution and storage</td>
<td>Deployment</td>
</tr>
<tr>
<td>4 Use</td>
<td>Servers/ Cloud/ etc</td>
</tr>
<tr>
<td>5 End of life</td>
<td>GOTO 1 or Uninstall</td>
</tr>
</tbody>
</table>
**istrue**

1.0.1 • Public • Published 8 years ago

---

### isTrue

A useful function to check if a javascript object evaluates to `true`.

### Getting started

Bower:

```
bower install istrue
```

npm:

```
npm install istrue --save
```

Or clone from github:

```
git clone https://github.com/codecelot/istrue.git.
```

**node:**

```
require("./path/to/repo/main.js");
```

**html embedded js:**

---

Install
```
> npm i istrue
```

Repository

🔗 [github.com/codecelot/istrue.git](https://github.com/codecelot/istrue.git)

Homepage

🔗 [github.com/codecelot/istrue.git#read...](https://github.com/codecelot/istrue.git#read...)

---

**Energy Cost:** 5.16 KJ via PSU (AC)

**SCI:** 0.06 mgCO2e/Unit test

---

**Total Development Energy wH**

2384

---

**License**

BSD

---

---
Development

- Measure the energy while developing
- What about Spotify? *Everything that is happening on the machine counts to the project.*
- Power Hog

**Image:**

- **Xcode:** App with the highest energy usage
- **7.39 Watt Min:** System energy usage
- **All measurement systems are functional**

**Table:**

<table>
<thead>
<tr>
<th>Name</th>
<th>Energy Impact</th>
<th>AVG CPU time %</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.apple.dt.Xcode</td>
<td>157173</td>
<td>41</td>
</tr>
<tr>
<td>com.apple.security.s...</td>
<td>26915</td>
<td>55</td>
</tr>
<tr>
<td>Python</td>
<td>24331</td>
<td>3</td>
</tr>
<tr>
<td>com.apple.Windows...</td>
<td>17783</td>
<td>10</td>
</tr>
<tr>
<td>kernel_coalition</td>
<td>15536</td>
<td>17</td>
</tr>
<tr>
<td>Proton Mail Bridge</td>
<td>13387</td>
<td>64</td>
</tr>
<tr>
<td>hog</td>
<td>8539</td>
<td>26</td>
</tr>
</tbody>
</table>
## Overall CI/CD

<table>
<thead>
<tr>
<th>Label</th>
<th>Energy</th>
<th>Time</th>
<th>Avg. CPU Util.</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average</td>
<td>Std Dev</td>
<td>Std Dev %</td>
<td>Average</td>
</tr>
<tr>
<td>Full Run 🌟</td>
<td>149,235 mJ</td>
<td>36,598 mJ</td>
<td>25%</td>
<td>33s</td>
</tr>
<tr>
<td>javascript tests</td>
<td>149,235 mJ</td>
<td>36,598 mJ</td>
<td>25%</td>
<td>33s</td>
</tr>
</tbody>
</table>

*GREEN CODING*
Development - Open Question

- What about code hosting
- What about tools like Copilot
- What about heating of the office

Currently in the works. Green Web Foundation working on IETF Header.
Green Metrics Tool

Our solution of precise usage measurements of containerized applications.

Phases concept from the Blue Angel:

- Baseline
- Idle
- Usage Scenario
Runtime can contain multiple flows. By default all runtime flows are aggregated. Please select a separate flow if needed.

### Single Phase Data

**Energy metrics**
- CPU Energy (Package) [RAPL]
- Memory Energy (DRAM) [RAPL]
- Network Energy (Formula)
- Machine Energy [mcp]

**Key metrics**

- **Phase Duration [s]**: 152.97
- **Machine Power [W]**: 33.71
- **Machine Energy [J]**: 516.84
- **Network Energy [J]**: 650.72
- **Machine CO2 (usage) [g]**: 0.68
- **Network CO2 [g]**: 0.09
- **Machine CO2 (manufacturing) [g]**: 0.27
- **SCI (pCO2e/Unit test)**: 0.00
Deployment Stage

Total energy consumption

- Machine Energy - a164f3a3e1e33f892fd977f2308f16b5a7fc45
Usage Stage

Use the Green Metrics Tool to get actual values

/save : 0.0502 J
/last_time: 0.024755 J
/badge: 0.050715 J

HTTP Header

x-energy-joule: 0.0502
x-self-energy-accounting: true
GMT Cluster

- In the cloud measurement infrastructure
- Blue Angel Compatible machine, SoftAWERE etc ...
- Can be part of the development workflow
What to do with all this data?

Project Carbon DB

We need a central point where all the data is gathered over the lifetime of the project.

The Green Metrics Tool is the first step in this direction but more work is needed.
Software Lifecycle Assessment

Power Hog

Eco CI

Green Metrics Tool

Energy ID
Conclusion

- Proposed a mapping from Life Cycle Assessment to continuously developed software.
- Introduced initial tooling for all stages.
- We need a central carbon database for software projects in the future!
Questions

Find detailed articles under:
https://www.green-coding.berlin/blog/

Let's work together: didi@green-coding.berlin
References

- [https://kruschecompany.com/agile-software-development/](https://kruschecompany.com/agile-software-development/)
- [https://github.com/marp-team/marpit](https://github.com/marp-team/marpit)