Measuring the Embodied Carbon of Continuous Integration Pipelines

Anonymous Author(s)

Abstract

Continuous integration (CI) pipelines run millions of ephemeral build and test jobs every day, yet sustainability assessments of software development almost exclusively consider the operational energy of these jobs, ignoring the embodied carbon of the hardware they run on. This poster presents CICarbon, a measurement methodology and open-source tool that attributes both operational and embodied emissions to individual CI jobs. Embodied carbon is amortised over device lifetime and allocated to jobs in proportion to their exclusive occupation of CPU, memory, and storage resources, following the principles of the Software Carbon Intensity specification but adapted to the bursty, short-lived nature of CI workloads. We apply CICarbon to 14 months of CI history from 27 open-source repositories (1.9 million jobs) executed on a mix of cloud-hosted and self-hosted runners. Our results show that embodied carbon accounts for 43-61% of total CI emissions on typical cloud runners — a share that rises for short jobs, where machine provisioning dominates useful work. We identify three practical reduction levers: test selection cuts total carbon by 29% in our subject repositories, runner right-sizing by 17%, and extending runner hardware lifetime from four to six years by 22%. The poster includes an interactive dashboard demonstration and a call for repository-level carbon budgets in open-source projects.

Keywords: *continuous integration, embodied carbon, software sustainability, DevOps, carbon accounting*

1 Introduction

Every push to a popular repository triggers a cascade of continuous integration jobs: builds, unit tests, integration suites, linters, and artefact packaging. Industry telemetry suggests that CI/CD systems execute hundreds of millions of jobs daily worldwide. Sustainability discussions in software engineering have begun to quantify the operational energy of this activity, but they systematically omit embodied carbon — the emissions from manufacturing, transporting, and disposing of the servers CI runs on.

The omission matters because CI workloads are unusually bursty and short-lived. A median CI job in our dataset runs for under four minutes, and the fixed carbon cost of the hardware slice it occupies is a large fraction of its total footprint. Ignoring embodied carbon therefore not only underestimates totals but also distorts comparisons between optimisation strategies.

This work contributes a job-level carbon attribution methodology for CI, an open-source implementation that ingests standard CI provider telemetry, and an empirical study across 27 open-source repositories that quantifies the embodied share and evaluates practical reduction levers.

2 Method

CICarbon models total job emissions as the sum of an operational term and an embodied term. The operational term follows established practice: measured or estimated energy use multiplied by the location-based grid carbon intensity at execution time. The embodied term amortises the manufacturing footprint of the runner hardware linearly over its service lifetime and allocates the amortised flow to jobs in proportion to their time-weighted share of CPU cores, memory, and storage, mirroring the Software Carbon Intensity specification's treatment of reserved resources.

For cloud-hosted runners, where hardware details are opaque, we construct embodied-carbon priors from public life-cycle assessments of comparable server platforms and propagate their uncertainty through the analysis as intervals rather than point estimates. For self-hosted runners we use vendor product carbon footprint documents directly.

We validate the operational component against wall-plug measurements on three self-hosted runner classes (mean absolute error 7.8%) and conduct a sensitivity analysis over hardware lifetime (3-7 years), utilisation assumptions, and grid-intensity data sources.

3 Results

Across 1.9 million jobs, embodied carbon accounts for 43-61% of total CI emissions on cloud-hosted runners (interval reflects embodied-prior uncertainty) and 35-49% on self-hosted runners, which have longer lifetimes and higher utilisation. The embodied share is strongly anti-correlated with job duration: for jobs under one minute, provisioning and teardown overheads push the embodied share above 70%.

Applying test-selection tooling in our subject repositories reduces total carbon by 29% by skipping tests unaffected by a change. Right-sizing over-provisioned runners (jobs using under 25% of allocated cores) saves a further 17%. Extending assumed hardware lifetime from four to six years reduces the embodied term by one third, cutting totals by 22% — a reminder that procurement policy is a software-sustainability lever.

A per-repository dashboard visualises carbon per job, per workflow, and per merged pull request; maintainers of four subject projects used it in a two-week trial and each identified at least one workflow to prune or reschedule.

4 Conclusion

Embodied carbon is not a rounding error in CI: it constitutes roughly half of pipeline emissions and dominates for short jobs. Job-level attribution is feasible with data CI providers already expose, and it surfaces actionable savings that operational-only accounting misses.

We invite the community to trial CICarbon on their own repositories, and we propose that large open-source projects adopt carbon budgets for CI alongside existing duration budgets.

References

[1] Green Software Foundation. Software Carbon Intensity (SCI) specification, version 1.1. ISO/IEC 21031:2024.

[2] U. Gupta, Y. G. Kim, S. Lee, et al. Chasing carbon: The elusive environmental footprint of computing. In Proc. HPCA, pages 854–867, 2021.

[3] M. Chau, R. Rahman, and I. Ahmed. Are we killing the planet one build at a time? An empirical study of CI energy consumption. In Proc. ICSE, 2024.

[4] T. Durieux, R. Abreu, and A. van Deursen. On the carbon footprint of continuous integration: a large-scale analysis of GitHub Actions. Empirical Software Engineering, 29(4), 2024.

[5] L. Cruz and R. Abreu. Catalog of energy patterns for mobile applications. Empirical Software Engineering, 24(4):2209–2235, 2019.

[6] Dell Technologies. PowerEdge R650 product carbon footprint report. Technical report, 2022.