TOWARDS SUSTAINABLE CONTINUOUS INTEGRATION

MEASURING ENERGY CONSUMPTION IN BATCH TESTING

An Empirical Study of Static and Dynamic Batching Strategies

Software Engineering • Máté Oszkó • moszko@tudelft.nl

BATCHSTOP4 80.3% Avg. Reduction

LINEAR-4 85.2% Avg. Reduction

PROBLEM STATEMENT

CI/CD ENERGY COSTS

Modern software projects execute thousands of tests daily through continuous integration pipelines, consuming significant computational resources and energy.

~10^8	10.4%	456.9 $MTCO_2$
Builds Ran Every Day [1]	Average Failure Rate [2]	Emissions/Year [3]

RESEARCH GAP

- Limited studies on batch testing energy profiles
- No standard metrics for test energy efficiency
- Lack of tools for energy-aware scheduling

RESEARCH QUESTIONS

MAIN RESEARCH QUESTION

How Do Different Batch Testing Approaches Affect Energy Profiles?

SUBQUESTIONS

To what extent do different batch testing strategies reduce energy consumption compared to baseline?

How do test suite CPU utilisation and baseline failure rate influence the energy savings of batch testing?

DATASET

Projects analyzed:	8
Commits per measurement:	160
Builds ran:	5172
Repetitions per measurement:	3
Total experiment run time:	240h 41m 37s

METHODOLOGY

1. INSTRUMENTATION

EnergiBridge was chosen as a software based tool for measuring energy usage, only the CPU's consumption is in the scope for this project. The measurements were ran on a skeleton Kubuntu installation on a desktop PC.

2. EXPERIMENTAL DESIGN

Two batching algorithms were evaluated across different public github repositories, one static and one dynamic algorithm. Slight adjustments were made to accommodate for the lack of commit independence in public git histories.

BatchStop4 (Static) | Linear-4 (Dynamic)

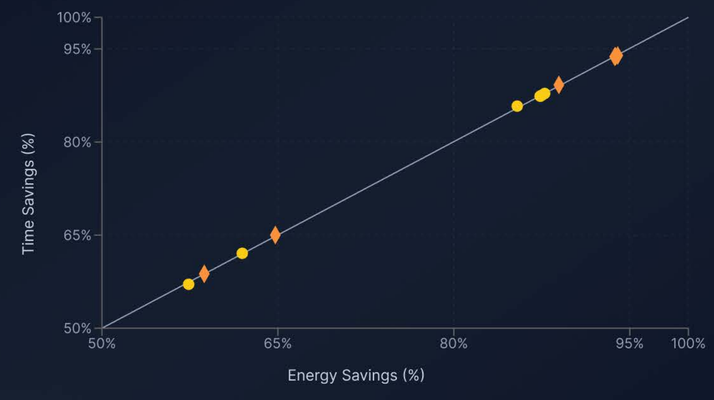
3. DATA COLLECTION

Each project was benchmarked with each algorithm on 160 commits, and each measurement was ran 3 times. Energy readings were sampled at 100ms intervals.

ALGORITHM MEASUREMENT RESULTS

REPOSITORY	Baseline (Wh)	BatchStop4 (Wh)	Linear-4 LWD (Wh)
commons-lang	3029.00	1150.46 ↓ 62.0%	1061.13 ↓ 65.0%
lucene	375.86	45.92 ↓ 87.8%	22.93 ↓ 93.9%
commons-io	233.87	29.28 ↓ 87.5%	14.62 ↓ 93.7%
commons-compress	105.76	13.39 ↓ 87.3%	6.60 ↓ 93.8%
commons-text	58.67	7.42 ↓ 87.4%	3.70 ↓ 93.7%
commons-scxml	29.89	12.84 ↓ 57.0%	12.34 ↓ 58.7%
commons-jxpath	16.82	2.12 ↓ 87.4%	1.06 ↓ 93.7%
commons-logging	9.10	1.30 ↓ 85.7%	0.99 ↓ 89.1%

ENERGY SAVINGS VS TIME SAVINGS



BATCH TESTING ALGORITHMS

STATIC ALGORITHM: BATCHSTOP4

DYNAMIC ALGORITHM: LINEAR-4

Comparison of static bisection and dynamic batch size adaptation strategies

[1] "(PDF) Demystifying Cloud-Native Architectures – Building Scalable, Resilient, and Agile Systems," ResearchGate, Aug. 2025, doi: 10.32996/jcsts.2025.7.4.97.

[2] M. Beller, G. Gousios, and A. Zaidman, "Oops, My Tests Broke the Build: An Explorative Analysis of Travis CI with GitHub," in 2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR), Buenos Aires, Argentina: IEEE, May 2017, pp. 356–367. doi: 10.1109/MSR.2017.62.

[3] N. Saavedra, A. Mendes, and J. F. Ferreira, "Environmental Impact of CI/CD Pipelines," Oct. 31, 2025, arXiv: arXiv:2510.26413. doi: 10.48550/arXiv.2510.26413.