Prepared by Deivyansh

Tools for Carbon Footprinting
Kepler, CodeCarbon and Co2.js

Introduction

- Computing infrastructure and digital applications consume significant energy globally.
- Internet usage alone contributes approximately 3.7% of the world's greenhouse gas emissions.
- This report examines three prominent tools for measuring computing-related carbon footprints: Kepler, CodeCarbon, and CO2.js.
- Each tool provides unique ways to measure carbon emissions in different computing environments: Kubernetes workloads, machine learning models, and web applications.
- The analysis covers key features, architecture, and a comparison to help organizations choose the right tool for sustainability programs.
- Growing computational loads, AI, and ML software demand high energy consumption.
- Measuring carbon footprint accurately is the first step in reducing emissions.
- The three tools discussed target different computing environments:
  - Kepler: Kubernetes and containerized applications.
  - CodeCarbon: Machine learning and Python-based workloads.
  - CO2.js: Web applications and data transfer-related emissions.
- Kubernetes-based Efficient Power Level Exporter (Kepler) tracks energy consumption in cloud-native environments.
- Developed by IBM and Red Hat, part of the CNCF Sandbox project.
- Uses a measurement-based approach rather than estimates from billing data.
- Provides pod-level energy consumption data, helping optimize microservices architectures.
- Uses extended Berkeley Packet Filter (eBPF) for kernel-level energy statistics.
- Employs Running Average Power Limit (RAPL), ACPI, and NVIDIA Management Library (nvml) for hardware energy tracking.

- Deploys daemon sets on Kubernetes nodes to gather process, container, and pod energy metrics.
- Uses machine learning models to estimate power usage in virtualized environments.
- Integrates with Kubernetes API and exports data in Prometheus format for monitoring.
- Lightweight Python package for tracking carbon emissions in machine learning workloads.
- Designed for AI developers to monitor and reduce computational emissions.
- Seamlessly integrates into Python codebases to estimate environmental impact.
- Monitors power consumption of GPUs, CPUs, and RAM.
- Uses regional carbon intensity values to calculate emissions.
- Provides online (real-time data) and offline (pre-installed factors) trackers.
- Emissions are visualized through a built-in dashboard.
- Suggest low-carbon computing regions to optimize workloads.
- JavaScript library by The Green Web Foundation for calculating web-related carbon emissions.
- Aims to support a fossil-free internet by 2030.
- Focuses on data transfer emissions, converting bytes into carbon estimates.
- Works across browsers, Node.js, serverless, and edge environments.
- Two models for estimation:
  - OneByte Model: Simple calculation based on data transfer.
  - Sustainable Web Design Model: Includes device type, network, and CPU usage.
- Supports integration with The Green Web API to check renewable-powered hosting.
- Can be used in performance budgets and integrates with the Impact Framework for in-depth analysis.

Comparison

Target Environments and Use Cases

Tool	Best For	Primary Uses
Kepler	Kubernetes workloads	Cloud-native teams, DevOps
CodeCarbon	Python ML workloads	AI/ML developers, Data Scientists

CO2.js	Web applications	Web developers, Digital agencies

Measurement Methods and Precision

Tool	Hardware	Accuracy
Kepler	Hardware counters, eBPF	High
CodeCarbon	Software-based power estimates	Moderate
CO2.js	Data transfer models	Low-Moderate

Integration Capabilities

Tool	Integration	Ecosystem
Kepler	Kubernetes, Prometheus	Cloud-native observability
CodeCarbon	Python libraries, dashboards	Jupyter Notebooks, ML workflows
CO2.js	JavaScript, APIs	Web development frameworks

Conclusion

- Organizations must adopt tools based on their technology stack:
  - Kepler for Kubernetes-based infrastructure.
  - CodeCarbon for machine learning workloads.
  - CO2.js for web and data transmission emissions.
- No single tool covers all use cases—a multi-tool approach may be necessary.
- These tools mark a significant step toward sustainable computing and climate-conscious technology development.