Ichnos: A Carbon Footprint Estimator for Scientific Workflows

Kathleen West
University of Glasgow
Glasgow, United Kingdom
kathleen.west@glasgow.ac.uk

Yehia Elkhatib
University of Glasgow
Glasgow, United Kingdom
yehia.elkhatib@glasgow.ac.uk

Lauritz Thamsen
University of Glasgow
Glasgow, United Kingdom
lauritz.thamsen@glasgow.ac.uk

1 Introduction

Scientists in many fields, including genomics, materials science, and remote sensing, need to analyse increasing amounts of data [1, 4, 8, 9]. Scientific workflow systems facilitate the automation of such analyses, enabling scientists to compose pipelines out of black-box tasks with data dependencies between them. As these workflows are often used to process large quantities of data, they tend to be resource-intensive and long-running, leading to significant energy consumption and, therefore, carbon emissions. Furthermore, the growing popularity of big data applications has been identified as a driver for the increasing emissions of the ICT sector [5]. As such, it is crucial to quantify and understand the carbon footprint of scientific workflows.

Scientific workflow systems like Nextflow [2], allow for the design, execution and monitoring of workflows on heterogeneous clusters. While these systems usually generate detailed performance traces and logs for executed workflows, they do not produce a record of energy consumed or carbon emitted. Consequently, users must manually monitor power consumption with hardware/software power meters or, otherwise, use a methodology like Cloud Carbon Footprint (CCF)[1] or Green Algorithms (GA) [7], which employ linear power models to translate resource utilisation into energy consumption. In either case, to translate energy consumed into carbon emitted, users need a measure of carbon intensity (CI), such as a yearly average or a more fine-grained metric. Generally, CI measures the amount of carbon (CO_2e) produced per kilowatt-hour (kWh) of electricity consumed, and varies across different locations, seasons, and times, depending on the sources generating electricity and the demand on the grid.

In practice, monitoring power consumption requires the user to attach a physical power meter or to enable a software-based tool like Intel's Running Average Power Limit (RAPL) prior to the execution of a workflow. Without this step, power consumption can only be estimated based on coarse-grained resource utilisation metrics. This is possible using the CCF and GA tooling, but only at reduced accuracy. Both methodologies assume that energy consumption scales linearly, which not necessarily holds in practice [6]. Moreover, to build linear power models, the GA methodology relies on vendor-specified Thermal Design Power (TDP) of assigned compute resources, a proprietary metric that does not reflect key processor settings such as processor frequency and does not indicate idle power consumption. Furthermore, while both methodologies translate power consumption into carbon emissions, they use a static average value to represent the CI of electricity consumed by the compute workload, ignoring that CI is often highly variable.

[1]https://www.cloudcarbonfootprint.org/docs/methodology/

LOCO '24, 3rd December, 2024, Glasgow, Scotland
2024.

To address these limitations, we propose *Ichnos*, a novel and flexible tool to estimate the carbon footprint of Nextflow workflows based on detailed workflow traces, CI time series, and power models. First, Ichnos takes as input the automatically-generated workflow trace produced by Nextflow. Use of these traces is an original contribution, ensuring that users do not need to manually monitor power consumption and enabling analysis of previously executed workflows. Next, Ichnos allows users to provide their own resource power model for utilised compute resources to accurately reflect processor settings, such as the processor frequency, instead of solely relying on a linear function. Finally, Ichnos converts estimated energy consumption to overall carbon emissions using fine-grained time-series CI data for each workflow task and only resorts to coarse-grained yearly averages where high-resolution location-based CI data are not available. Additionally, Ichnos reports estimated energy consumption and carbon emissions per task, providing greater granularity than existing methodologies and allowing users to identify which of their tasks have the largest footprint to address. We provide the implementation of Ichnos as open-source[2]. We demonstrate our tool on traces of two real-world Nextflow workflows, compare the estimated energy consumption against RAPL and the GA methodology, and show the tool's functionality by varying the granularity of provided CI data and varying the processor frequency settings of assigned compute resources.

2 System Design

Ichnos is a tool that produces an estimate of the operational carbon footprint from the execution trace of a Nextflow scientific workflow, using user-configured power and energy data aligning with the execution. Figure 1 provides an overview of the tool's design.

In Phase 1, input data are provided by the user in the form of three items: (1) the workflow trace that includes a task-level summary of resource usage including the runtime, CPU utilisation and allocated memory, (2) the power model used to estimate power consumption for utilised compute resources, and (3) the CI data which should be fine-grained time-series data, if available, or instead a coarse-grained average. To provide an accurate estimate of energy consumption, the user can provide a power function or regression-based model to reflect processor settings.

In Phase 2, resource usage data are extracted from the workflow trace for each task, and the energy consumption is estimated using the provided power model. Subsequently, the energy consumption per task is translated into carbon emissions using the provided CI data. This estimates operational carbon emissions by aligning the tasks of potentially long-running workflow applications with CI data matching the specific execution times. These estimations are

[2]https://github.com/westkath/ichnos

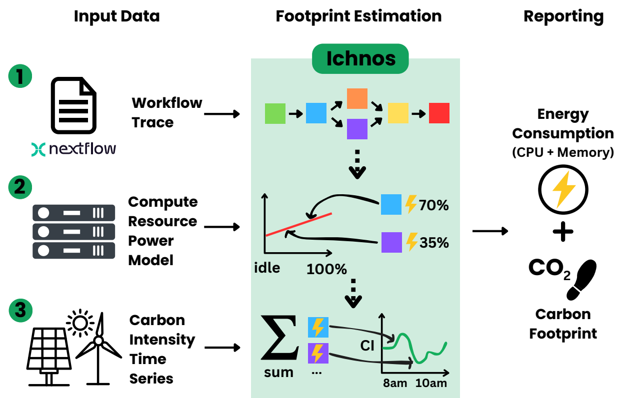

Figure 1: High-level design of the Ichnos Carbon Footprint estimator system with per-task power and emissions estimation, based on provided input data, and detailed reporting.

summed to calculate the power consumption and carbon emissions for the overall workflow execution.

In Phase 3, the energy consumption and carbon emissions estimated for each task are summarised in a carbon footprint trace file, alongside a summary of the overall carbon footprint. Providing a trace file of estimated carbon emissions per task allows users to identify the most power-hungry tasks and review how these align with fluctuating CI to consider how overall carbon emissions could be reduced.

3 Experiments

In this evaluation, we conduct three experiments demonstrating how we use Ichnos: (1) we estimate the carbon footprint for a historical Nextflow workflow trace, (2) we estimate the energy consumption on a compute node where processor frequency was configured before executing a workflow, and (3) we estimate the carbon footprint whilst varying the granularity of CI data.

Use of Historical Traces. We used Ichnos to estimate the carbon footprint of historical executions reported for the FORCE workflow[3], implemented using Nextflow. Specifically, we estimated the carbon footprint of the three workflow executions that occurred on a single node, configuring the yearly average CI in Germany in 2023 as $394gCO_2e/kWh$. Moreover, we used a linear power model ranging between 80–135W to estimate energy consumption. The average energy consumption was $\mathbf{30.51}kWh$, with CPU energy consumption accounting for 99% of overall energy consumption, with memory responsible for the remaining 1%. The translated carbon footprint was $\mathbf{12}kgCO_2e$.

Varying Processor Frequency Settings. We executed the ampliseq[4] workflow, using the full-size real-world dataset provided from nf-core, a community-curated collection of workflows [3], on a server equipped with an 8-core Intel i7-10700T CPU and 32GB of memory. Throughout execution, we monitored energy consumption using perf, a wrapper for RAPL, and we consider these values to be the ground truth for energy consumed. Additionally, we used GA as a baseline for estimating the energy consumption, using the manufacturer-specified TDP of 35W and used the average CPU utilisation over the workflow execution. When using Ichnos, we employed a linear power model and measured the power consumption of the server at idle and at 100% utilisation for each processor frequency (2GHz, 3GHz, 4GHz). The experimental results are outlined in Table 1. Compared to RAPL, Ichnos overestimates energy consumption less than GA, specifically 1.1–2.1x instead of 1.6–5x, and still enables post-hoc estimation.

Frequency (GHz)	Power Consumption (kWh) Ichnos	RAPL	GA
2	0.090	0.044	0.234
3	0.097	0.058	0.141
4	0.085	0.076	0.124

Table 1: Monitored power consumption from RAPL compared with estimates using Ichnos and GA.

Varying the Granularity of CI Data. We took the workflow trace from the ampliseq execution that ran on a single server for 2h 40m in the evening of September 26th 2024 in Glasgow. The CI fluctuation for South Scotland region of the National Grid[5] is depicted in Figure 2. If this information was not available, we would use the average CI for the National Grid in 2023, which was $215gCO2e$[6]. Ichnos estimated that the footprint was $\mathbf{0.33}gCO2e$ using the region-specific time-series CI data, while the footprint estimated using the coarse-grained average would be $\mathbf{18.65}gCO2e$ - an estimate almost 60x larger. This highlights the potential of using a flexible tool such as Ichnos, where the user can provide specific high-resolution data to more accurately estimate the carbon footprint.

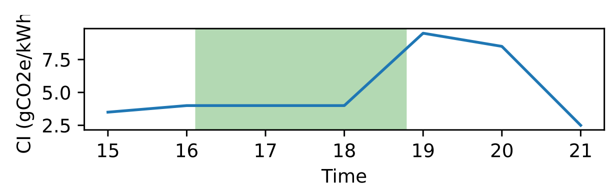

Figure 2: Variable Carbon Intensity in South Scotland on 26/09/2024, with workflow execution highlighted in green.

4 Outlook

In the future, we intend to expand our evaluation with further workflows, more fine-grained CI data, and a greater variety of compute resources. In addition, we plan to automate the process of creating custom power models. Specifically, we will explore integrating a method for resource-efficient profiling of available processor configurations, using microbenchmarks and RAPL on each compute resource, and selecting the power model that best fits the measured energy consumption.

[3]https://github.com/CRC-FONDA/FORCE2NXF-Rangeland
[4]https://nf-co.re/ampliseq/2.11.0
[5]https://carbonintensity.org.uk/
[6]https://app.electricitymaps.com/

References

[1] Bruce Berriman, Ewa Deelman, John Good, Joseph Jacob, Daniel Katz, Carl Kesselman, Anastasia Laity, Thomas Prince, Gurmeet Singh, and Mei-Hui Su. 2004. Montage: A Grid Enabled Engine for Delivering Custom Science-Grade Mosaics on Demand. In *Optimizing Scientific Return for Astronomy through Information Technologies*, Vol. 5493. SPIE.

[2] Paolo Di Tommaso, Maria Chatzou, Evan W. Floden, Pablo Prieto Barja, Emilio Palumbo, and Cedric Notredame. 2017. Nextflow Enables Reproducible Computational Workflows. *Nature Biotechnology* 35, 4 (2017).

[3] Philip A Ewels, Alexander Peltzer, Sven Fillinger, Harshil Patel, Johannes Alneberg, Andreas Wilm, Maxime Ulysse Garcia, Paolo Di Tommaso, and Sven Nahnsen. 2020. The nf-core framework for community-curated bioinformatics pipelines. *Nature biotechnology* 38, 3 (2020).

[4] James A. Fellows Yates, Thiseas C. Lamnidis, Maxime Borry, Aida Andrades Valtueña, Zandra Fagernäs, Stephen Clayton, Maxime U. Garcia, Judith Neukamm, and Alexander Peltzer. 2021. Reproducible, Portable, and Efficient Ancient Genome Reconstruction with Nf-Core/Eager. *PeerJ* 9 (2021).

[5] Charlotte Freitag, Mike Berners-Lee, Kelly Widdicks, Bran Knowles, Gordon S. Blair, and Adrian Friday. 2021. The Real Climate and Transformative Impact of ICT: A Critique of Estimates, Trends, and Regulations. *Patterns* 2, 9 (2021).

[6] Chaoqiang Jin, Xuelian Bai, Chao Yang, Wangxin Mao, and Xin Xu. 2020. A review of power consumption models of servers in data centers. *Applied Energy* 265 (2020).

[7] Loïc Lannelongue, Jason Grealey, and Michael Inouye. 2021. Green Algorithms: Quantifying the Carbon Footprint of Computation. *Advanced Science* 8, 12 (2021).

[8] Paul Muir, Shantao Li, Shaoke Lou, Daifeng Wang, Daniel J. Spakowicz, Leonidas Salichos, Jing Zhang, George M. Weinstock, Farren Isaacs, Joel Rozowsky, and Mark Gerstein. 2016. The Real Cost of Sequencing: Scaling Computation to Keep Pace with Data Generation. *Genome Biology* 17, 1 (2016).

[9] Jörg Schaarschmidt, Jie Yuan, Timo Strunk, Ivan Kondov, Sebastiaan Huber, Giovanni Pizzi, Leonid Kahle, Felix Bölle, Ivano Castelli, Tejs Vegge, Felix Hanke, Tilmann Hickel, Jörg Neugebauer, Celso Rêgo, and Wolfgang Wenzel. 2021. Workflow Engineering in Materials Design within the BATTERY 2030 + Project. *Advanced Energy Materials* 12 (2021).