Increasing Awareness for Energy Consumption in Jupyter Notebooks

Marcel Garus
marcel.garus@student.hpi.de
Hasso Plattner Institute
Potsdam, Germany

Sven Köhler
sven.koehler@hpi.de
Hasso Plattner Institute
Potsdam, Germany

Andreas Polze
andreas.polze@hpi.de
Hasso Plattner Institute
Potsdam, Germany

1 Introduction

Jupyter Notebooks [4] offer a development environment that is especially popular in the data science community. Notebooks use a client-server architecture. The frontend of these Python-based notebooks runs in the browser, while the backend can run on a more powerful machine. Free Jupyter Notebook services such as Google Colab [2] make data science and machine learning more accessible to a general audience. These research fields have been getting more compute-intensive over time [9].

The client-server architecture of Jupyter Notebooks hides the operational energy consumption and carbon impact. In a world where resource consumption is a growing concern, this is a problem. For example, in classroom settings, students experiment with expensive computations without being aware of the energy usage caused by their actions. Ostermann et. al [8] suggest that the unconstrained offering of academic cloud infrastructures can make academics develop resource-intensive workflows. Other works [5, 13] showed that giving households frequent feedback about their energy consumption can already lead to voluntary behavior changes.

We want to apply those insights to interactive computing and developed an extension [6] for Jupyter Notebooks that allows users to monitor their energy consumption from within the Jupyter frontend. Previous works [3] have implemented RAM and CPU monitoring similarly. Giving immediate feedback on the energy consumption while the code is developed, may allow users to identify energy-heavy sections early on, so they can be optimized before the full model is trained or a hyperparameter search is performed.

2 Measuring the Energy Consumption

There is no established unified way of getting information about energy consumption from the Linux kernel. We aggregate data from several sources:

Running Average Power Limit (RAPL)
This Intel API tracks the energy consumption of CPU components such as the CPU, RAM, or built-in GPUs on x86_64 architectures.

Microchip MCP39F511N Power Monitor (MCP)
The MCP39F511N is an IC intended for power monitoring. We use it through a demonstration board that is plugged between the wall socket and a device and provides information on the device's power consumption via a serial interface.

Nvidia Management Library (NVML)
This library can query dedicated Nvidia GPUs for their self-reported power draw.

LOCO 2024, Dec 03, 2024, Glasgow, UK
2024.

3 Building an Extension

We developed a Jupyter Notebook extension that shows the energy usage. The extension contains the following components as shown in Figure 1:

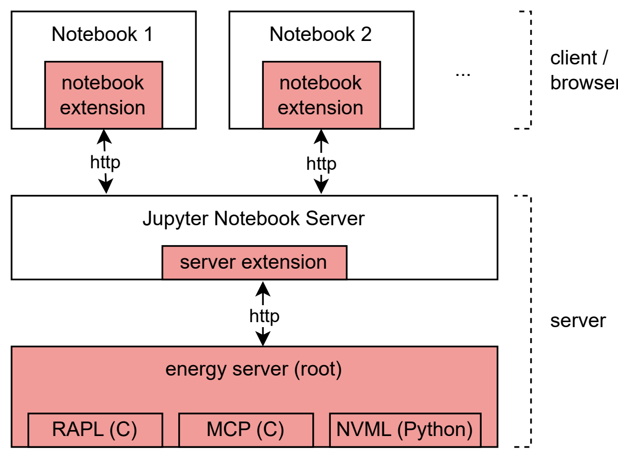

Figure 1: The architecture of Jupyter Notebooks. Our extension adds the red components.

Energy Server The energy server aggregates energy consumption data from RAPL, MCP, and NVML. Some operations (such as accessing the RAPL counters) require elevated privileges. By separating the energy tracking into another process, we can provide those privileges on a granular level. The Jupyter Notebook server, which runs client-provided Python code, is not privileged.

Server Extension This part of the extension runs in the Juypter Notebook server process and communicates with the energy server using HTTP. It also registers an HTTP endpoint for the frontend within the Jupyter Notebook server framework.

Notebook Extensions This part of the extension runs in the client's browser and extends the user interface.

Figure 2 shows a screenshot of the Jupyter Notebook frontend with the extension enabled. The extension adds a button to the notebook's toolbar that shows how much power the computer currently draws and how much energy it consumed since the notebook was opened. It also compares the amount of consumed energy to everyday activities such as baking a pizza. Clicking the button opens a popup with more information.

A short-term graph shows the power draw over the last 100 seconds. The background is tinted green for time spans when the code of the notebook ran, distinguishing it from idle energy usage.

Figure 2: The Jupyter Notebook Energy extension with short- and long-term view of energy usage and carbon intensity.

A long-term graph shows the percentage of energy on the grid that comes from renewable sources, based on the average energy generation in Germany. All data comes from the *European Network of Transmission System Operators for Electricity* (ENTS-OE) [1], a collection of companies that transfer energy through Europe to meet demand. This enables users to adjust their behavior based on whether the available energy is sourced sustainably.

4 Evaluation of the Accuracy and Overhead

We assessed the accuracy and overhead of measuring the energy consumption by comparing the internally reported energy consumption (RAPL) with the one measured externally (MCP). All benchmarks were run on an Asus ZenBook 14 laptop with Ubuntu 20.04.3. We compared an idle system, a memory-bound clustering algorithm [11], a set of compute-bound linear algebra algorithms [10], and a custom benchmark querying n-grams against a rudimentary database index. Figure 3 shows the average energy consumption over 10 measurements.

The externally measured energy consumption is generally higher – about seven times higher in the idle state and 1.2 to 2.0 times higher in the other benchmarks. This is expected: RAPL only measures CPU components; the MCP measures the power consumption of the whole system, including the display, fans, wireless chip, etc. In the idle state, CPU utilization is low. It only accounts for a small part of the energy usage.

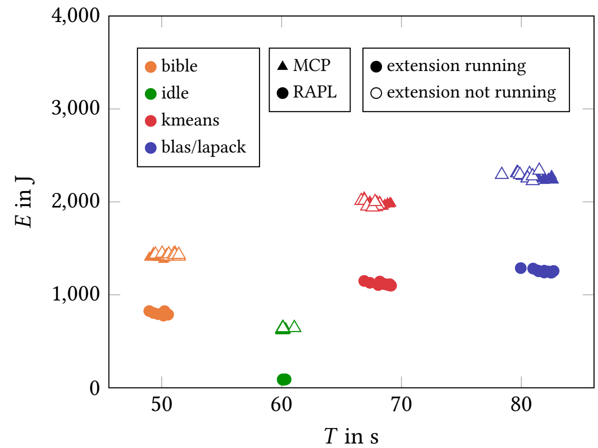

Figure 3: Time and energy usage of various benchmarks, measured internally (RAPL) and externally (MCP) with the extension enabled vs. disabled (10 runs each plus warm-up).

Comparing the energy consumption when the extension is enabled vs. disabled, we can see that the extension introduces no significant overhead regarding energy usage and time. This indicates the extension supports everyday usage.

5 User Study Design

An open question is whether the extension helps people to use less energy, for example by splitting one long editing session into multiple short ones. We plan to conduct a study where participants solve a data science problem in a Jupyter Notebook. Participants are divided into three groups: The first group does not use the extension. The second group also does not use the extension but is asked to reduce their energy usage. The third group uses the extension and is asked to reduce their energy usage. We hope to find a significant difference in how much energy is consumed.

6 Future Work

These are areas where the extension can be improved:

Report the energy generation more precisely. Currently, the extension only shows the CO_2 intensity of the entirety of Germany. It could use a more precise location or even the electricity contract.

Highlight monetary motivation. The extension could allow the user to enter an electricity price to automatically convert the energy use into the added cost on the electricity bill, resulting in an additional incentive to reduce energy consumption [7].

Allow users to add incentives. Organizations may use the measured energy consumption to set either positive incentives, such as university courses publicly recognizing students who trained their models with less energy, or negative incentives, such as research groups requesting baked goods from members with high energy usage. As shown by Sintov et. al [12], competition between peers can be a big motivator to incentivize energy-efficient workflows.

Acknowledgments

To Felix Grzelka and Clemens Tiedt, for proofreading. The authors acknowledge the financial support by the German Research Foundation (DFG) via "Memento" (502228341).

References

[1] 2013. ENTSO-E Transparency Platform. https://transparency.entsoe.eu/
[2] 2017. Google Colab. https://colab.research.google.com
[3] 2024. jupyter-server/jupyter-resource-usage. https://github.com/jupyter-server/jupyter-resource-usage original-date: 2016-04-22T07:01:34Z.
[4] 2024. Project Jupyter. https://jupyter.org/
[5] Wokje Abrahamse, Linda Steg, Charles Vlek, and Talib Rothengatter. 2005. A review of intervention studies aimed at household energy conservation. *Journal of Environmental Psychology* 25, 3 (Sept. 2005), 273–291. https://doi.org/10.1016/j.jenvp.2005.08.002
[6] Marcel Garus. 2022. Jupyter Energy. https://github.com/MarcelGarus/jupyter-energy
[7] Richard D. Katzev and Theodore R Johnson. 1984. Comparing the Effects of Monetary Incentives and Foot-in-the-Door Strategies in Promoting Residential Electricity Conservation. *Journal of Applied Social Psychology* 14, 1 (1984), 12–27. https://doi.org/10.1111/j.1559-1816.1984.tb02217.x _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1559-1816.1984.tb02217.x.
[8] Simon Ostermann, Gabor Kecskemeti, and Radu Prodan. 2014. Fostering energy-awareness in scientific cloud users. In *2014 IEEE 3rd International Conference on Cloud Networking (CloudNet)*. 149–154. https://doi.org/10.1109/CloudNet.2014.6968984
[9] Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2020. Green AI. *Commun. ACM* 63, 12 (Nov. 2020), 54–63. https://doi.org/10.1145/3381831
[10] SciPy. 2021. BLAS/LAPACK benchmark. https://github.com/scipy/scipy/blob/918f0a32bc5a8d8adfb8/benchmarks/benchmarks/blas_lapack.py
[11] SciPy. 2023. Cluster benchmark. https://github.com/scipy/scipy/blob/7a50bbfe241dc30bfc5c/benchmarks/benchmarks/cluster.py
[12] Nicole Sintov, Ellen Dux, Agassi Tran, and Michael Orosz. 2016. What goes on behind closed doors? *International Journal of Sustainability in Higher Education* 17, 4 (Jan. 2016), 451–470. https://doi.org/10.1108/IJSHE-02-2015-0027 Publisher: Emerald Group Publishing Limited.
[13] Markus Weiss and Dominique Guinard. 2010. Increasing energy awareness through web-enabled power outlets. In *Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia (MUM '10)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/1899475.1899495

Received 30 September 2024