Understanding and measuring energy consumption of Artificial Intelligence applications

Danilo Carastan-Santos

Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, LIG

danilo.carastan-dos-santos@univ-grenoble-alpes.fr

Objectives

1. Understand why tracking the energy consumption of AI is important.

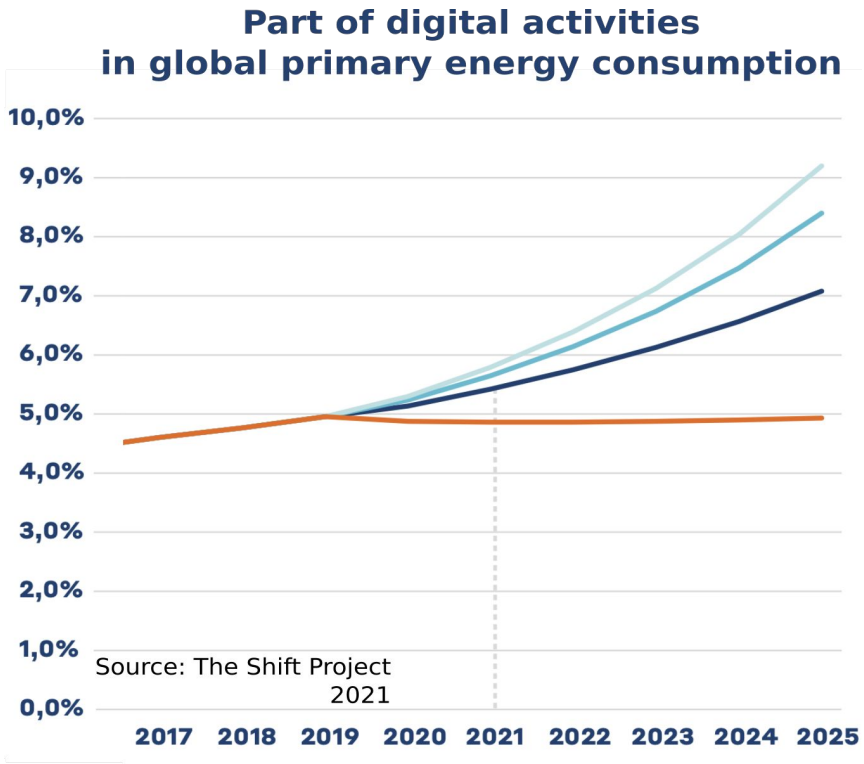
2. Be aware of existing hardware and software tools to measure the energy consumption of AI (and any kind of computing).

3. Develop a first experience on tracking the energy consumption and CO2 emissions of AI model training with software tools.

4. Promote a controlled and frugal development of AI.

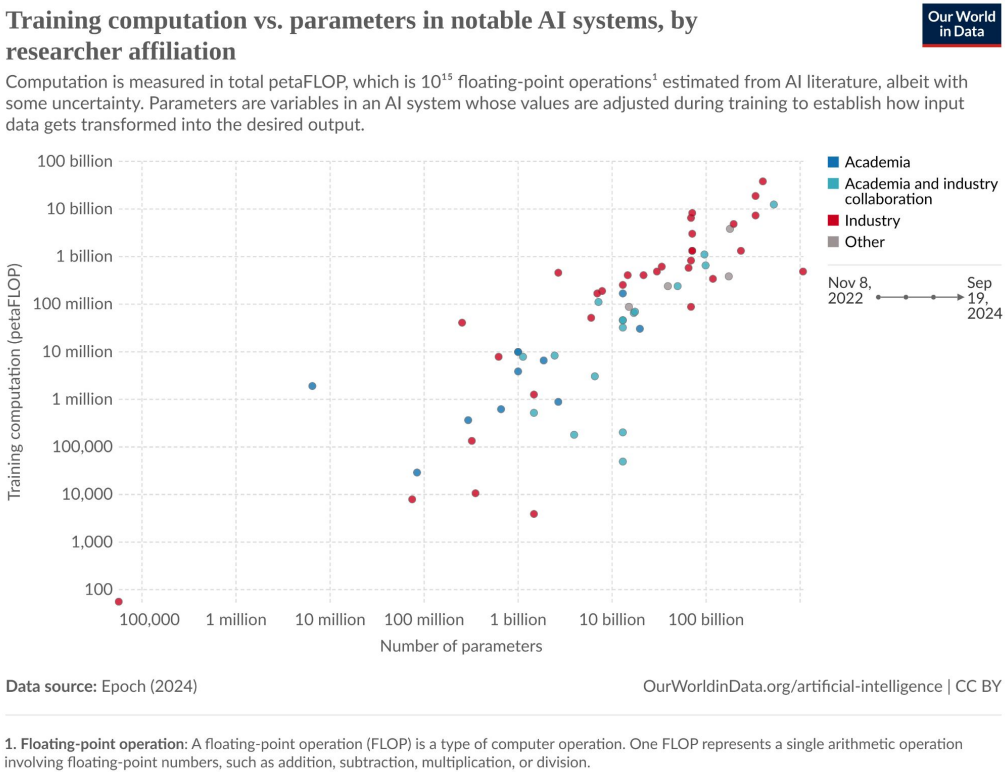
Computing voraciously craves for energy!

- Three top Lines = Mode "business as usual".
- As the curve steepens = More energy demand and more CO2 emissions.



AI craves for energy too!

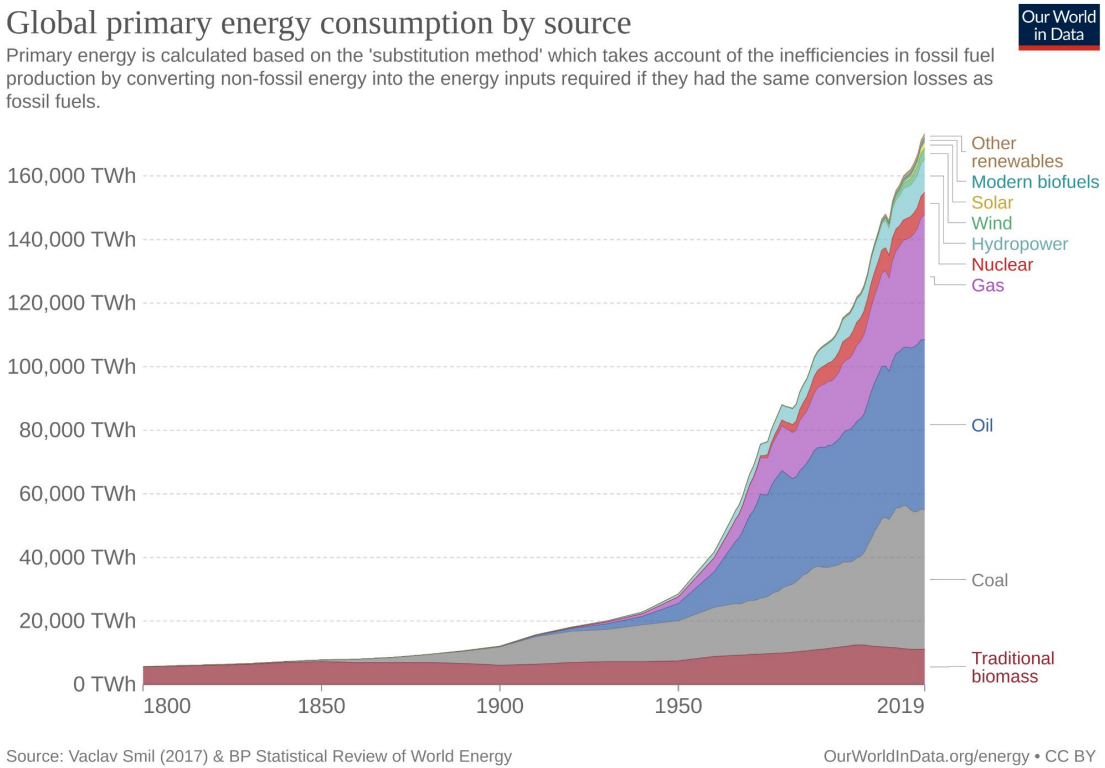
- Y axis = Computing demand = energy demand
- AI contributes to **steepen** the energy curve (previous slide)
- More energy demand = more energy consumption = more CO2 emissions



Source: https://ourworldindata.org/grapher/ai-training-computation-vs-parameters-by-researcher-affiliation?time=2022-11-08..latest

Naive solution: Let's just decarbonize the electricity production :)

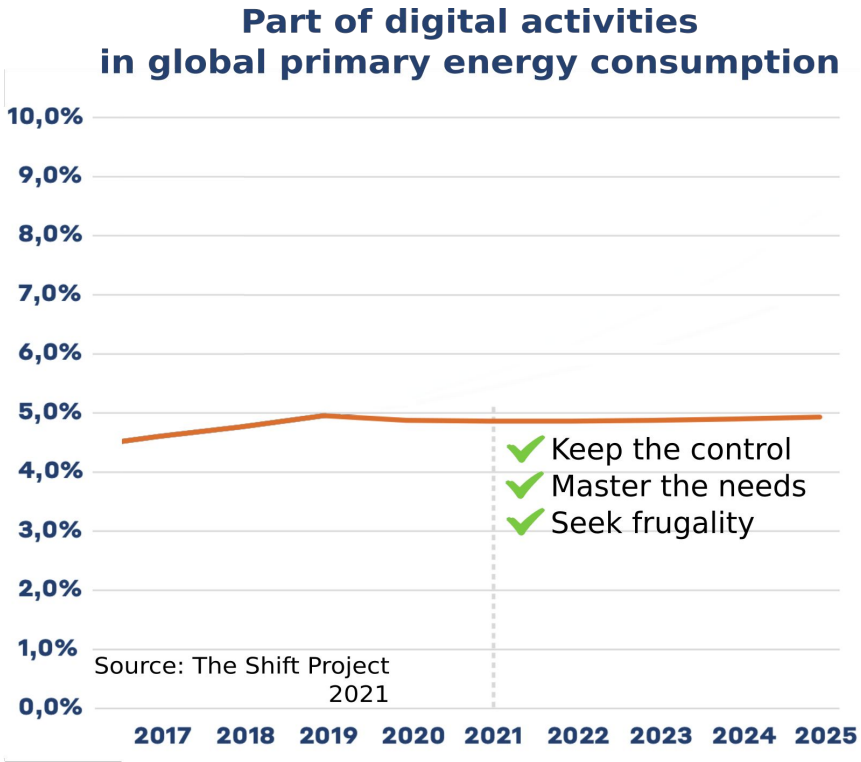
- Only about **16%** of electricity production came from **low-carbon** sources in 2019[1]
- **Low-carbon sources are harder to deploy.** They depend on specific conditions (wind, sunlight, geology) or use rare or dangerous materials (nuclear)
- The increase in energy demand surpasses the creation of low-carbon energy sources
- Can we decarbonize the remaining 84%?



[1]: https://ourworldindata.org/energy-mix

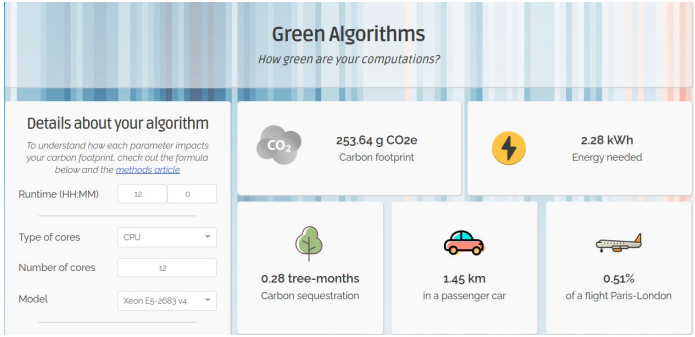
Why is it important to track the energy consumption of AI?

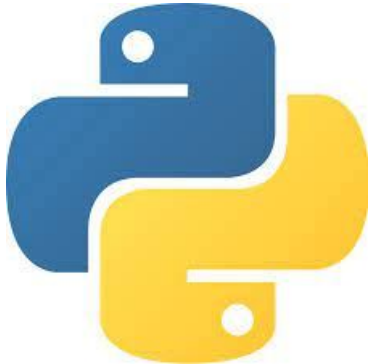
- Master the needs: **Be judicious** to whether we really need to deploy energy hungry AI
- Step to keep the control: know how much energy AI consumes **(this course!)**
- Seek frugality: Seek for **low energy consumption**



How can we measure energy of AI (Or any kind of computing)?

- **Estimations** from **hardware characteristics**
  - Green Algorithms:
  - https://www.green-algorithms.org/)
- **External measurements**
  - Outside the hardware, wattmeters
- **Software power models** (based on hardware counters)
  - Python libraries help to use hardware counters





Estimations from hardware characteristics

TDP Method

Energy = TDP x Utilization x Processing Time

- Utilization ranges from 0 to 1.
- The easiest is to set Utilization = 1 (100% CPU)
- The TDP (Thermal Design Power) is published by the CPU/GPU manufacturer

TDP Method: Pros

- The easiest method to deploy
- Can provide an estimation without running the code
- Enough accurate, if the utilization is constant

TDP Method: Cons

- Only estimates for the CPU/GPU, the rest (e.g., memory) is ignored

Wattmeters and Software tools

Wattmeters

Pros:

- Plugged on the power supply
- Independent of the hardware used

Cons:

- Harder to install and to get data from
- May reflect external factors
- It's yet another gadget

Software Tools

Pros:

- Easier to use and get data from
- Doable via software (Python Packages)

Cons:

- Estimations of the actual energy consumption
- Rely on specific hardware (Intel RAPL, and Nvidia-Smi)

Software tools

- Two main interfaces: Intel RAPL[1] (CPU) and nvidia-smi[2] (GPU)
- Python packages base on these tools

RAPL: Example - accessing hardware counters through the powercap[3] Linux interface

```
(base) dancarastan@dancarastan-Precision-7560:~$ sudo cat /sys/class/powercap/intel-rapl/intel-rapl:0/energy_uj
150019843670
```

nvidia-smi: System management interface

```
(base) dancarastan@dancarastan-Precision-7560:~$ nvidia-smi --query-gpu=index,power.draw --format=csv
index, power.draw [W]
0, 15.69 W
```

1- Khan, Kashif Nizam, et al. "RAPL in Action: Experiences in Using RAPL for Power measurements." ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS) 3.2 (2018): 1-26.
2 - https://developer.nvidia.com/nvidia-system-management-interface
3 - https://www.kernel.org/doc/html/latest/power/powercap/powercap.html

How to access RAPL data?

1. **Model specific registers (MSR)**
   - Accessible by opening `/dev/cpu/N/msr` (Linux), where N is the CPU core number
   - Requires specific CPU expert knowledge: offset and unit to access energy data in `/dev/cpu/N/msr` are machine-dependant
2. **Power capping framework (powercap)**
   - Interface to read RAPL data through files accessible at `/sys/devices/virtual/powercap/intel-rapl`
   - No machine-dependant requirements
   - Counter overflows happen

```
(base) dancarastan@dancarastan-Precision-7560:~$ sudo cat /sys/class/powercap/intel-rapl/intel-rapl:0/energy_uj
150019843670
```

3. **Perf-events subsystem**
   - Get RAPL data with count events
   - Overflows are automatically corrected, but still possible (64-bit integer limit)

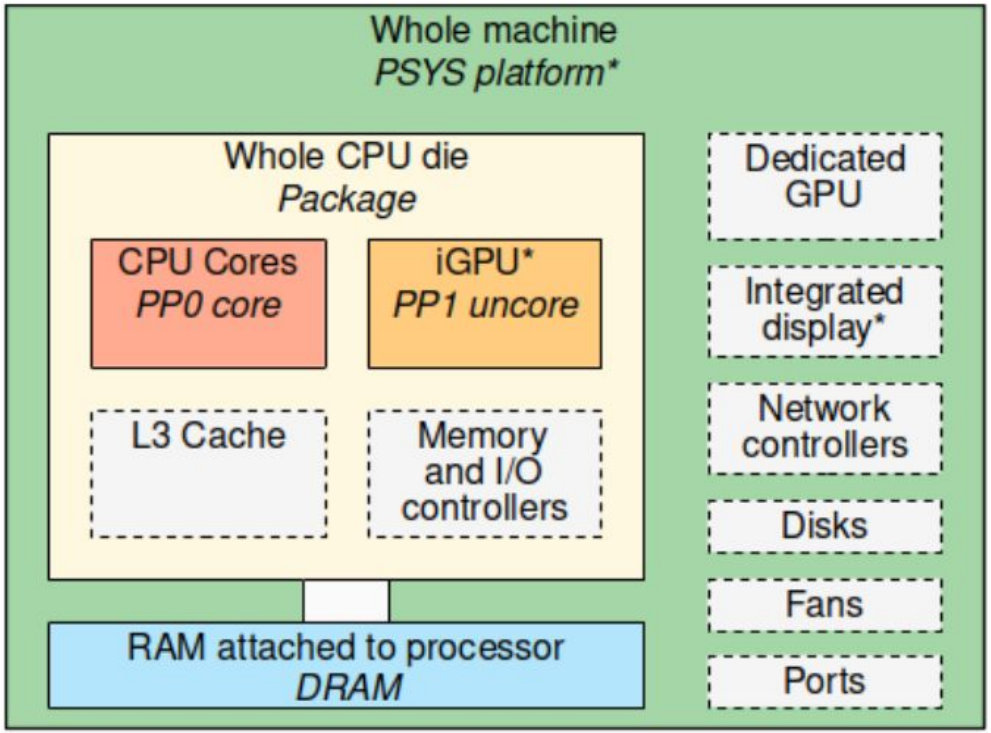
All of these options require some administrator rights

1- More information: Raffin, Guillaume, and Denis Trystram. "Dissecting the software-based measurement of CPU energy consumption: a comparative analysis." *arXiv preprint arXiv:2401.15985* (2024).

RAPL domains

- Hierarchy of RAPL data
- Important to know when using perf-events and powercap

1. Whole machine (PSYS)
2. Whole CPU die (Package)
3. CPU cores (PP0)
4. Integrated graphics (PP1)
5. Memory (DRAM)



Python Packages

- CodeCarbon (https://codecarbon.io/)
- Experiment-impact-tracker (https://github.com/Breakend/experiment-impact-tracker)
- CarbonTracker (https://pypi.org/project/carbontracker/)
- Etc.

- The packages deal with the hardware counters (RAPL, nvidia-smi) for you
- Easy to use “out of the box”
- Easy to integrate in AI code (Tensorflow, PyTorch, etc.)
- Needs some diving in the code if you want more control
  - **That's what we will do :)**

	External and intra-node devices		Power profiling software				Energy measurement software packages			Energy calculators	
	OmegaWatt	*BMC*	*Power API*	*Scaphandre*	*Energy Scope*	*Perf*	*Code Carbon*	*Experiment Impact Tracker*	*Carbon Tracker*	*Green Algorithms*	*ML CO2 Impact*
Developement											
Citation			[40]	[41]	[42]	[46]	[38]	[39]	[37]	[36]	[35]
First (latest) release date			Jul. 2019 (Aug. 2022)	Dec. 2020 (May 2021)	2021	Sept. 2009 (Jan. 2023)	Nov. 2020 (Sept. 2022)	Dec. 2019 (Jan. 2020)	Apr. 2020 (Jul. 2021)	Jul. 2020 (Jun. 2022)	Aug. 2019 (Jul. 2022)
Environment											
Hardware compatibility	Any	Any	Intel RAPL	Intel RAPL	Intel RAPL, Nvidia NVML	Intel RAPL	Any	Intel RAPL, Nvidia NVML	Intel RAPL, Nvidia NVML	Any	Any
Scope	Machine	Machine	CPU, DRAM, process	CPU, DRAM, process	CPU, DRAM, GPU	CPU, DRAM	CPU, DRAM, GPU	CPU, DRAM, GPU, process	CPU, DRAM, GPU	Machine	Machine
Virtualization support			Yes	Yes	No	No	No	No	No		
Job management support			No	No	OAR, SLURM	No	No	No	No		
Functional											
Hardware technology used			RAPL	RAPL	RAPL, NVML	RAPL	RAPL, NVML, TDP	RAPL, NVML	RAPL, NVML	TDP	TDP
Software power model used			Regression based on perf events	CPU usage based				GPU, CPU and RAM usage based			
Default sampling frequency (Hz)	1	0.2	1	0.1	2	10	1/15	1	0.1		
Online reporting	Yes	Yes	Yes	Yes	No	Yes	No	No	No	No	No
Power profiling	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	No	No
User-friendliness											
Availability of source code (License)			Yes (BSD 3-Clause)	Yes (Apache 2.0)	No	Yes (GNU GPL)	Yes (MIT)	Yes (MIT)	Yes (MIT)	Yes (CC-BY-4.0)	Yes (MIT)
Ease of use	Poor	Poor	Poor	Fair	Good	Good	Good	Good	Good	Very good	Very good
Quality of documentation			Good	Good	Good	Good	Fair	Fair	Good	Good	Fair
Configurability	Fair	Poor	Good	Good	Good	Good	Poor	Fair	Poor	Poor	Poor
Resulting data format	HTTP end-point	HTTP end-point	MongoDB, InfluxDB, Prometheus, CSV, Socket, File	Prometheus, Warp10, Riemann, JSON, Stdout	JSON	CSV, Stdout, File	CSV	JSON, Code	File, Code	Web	Web, Latex
Data visualisation possibilities	Grafana (Kwollect)	Grafana (Kwollect)	Grafana (InfluxDB, Prometheus)	Grafana (Prometheus)	Custom Dashboard		Comet			Graphs on the web page	

1- Table source: Jay, Mathilde, et al. "An experimental comparison of software-based power meters: focus on CPU and GPU." *2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*. IEEE, 2023..

A new power profiling software
Alumet (https://alumet.dev/)

⚡ ? Software energy consumption

? Performance

EVIDEN

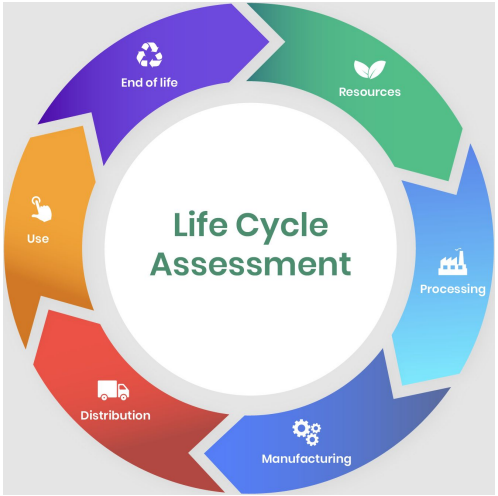
Modular measurement framework and tool

High performance
High frequency

Plugin system
=> bespoke tools

Metrics standard
+ flexible attributes

Written in Rust
(async inside!)



Going beyond: The life-cycle of an AI service

- Power profiling tools can only measure the **use phase** (I.e., Model training and Model deployment)

- Considering more phases require other methods
  - Example: Life-cycle analysis (LCA)
  - Few initiatives for computing technologies
    - https://dataviz.boavizta.org/

1- (right) Figure source: Luccioni, Alexandra Sasha, Sylvain Viguier, and Anne-Laure Ligozat. "Estimating the carbon footprint of bloom, a 176b parameter language model." *Journal of Machine Learning Research* 24.253 (2023): 1-15.

Let's get into practice!

Hands-on material: (https://github.com/danilo-carastan-santos/ai-energy-consumption)

1. **Requirements and installation procedure** (https://github.com/danilo-carastan-santos/ai-energy-consumption/blob/master/requirements.org)

2. **Five sections** (1a, 1b, 1c, 1d, 2a) and **Seven Questions** (Q1, Q2, Q3, Q4, Q5, Q6)

Thank you!

Let's keep in touch!

Email: danilo.carastan-dos-santos@univ-grenoble-alpes.fr

https://www.linkedin.com/in/danilo-carastan-santos/