Estimating Energy Consumption and Carbon Costs of GENSIM, DIGI and RECO jobs at LHC

Francesco Minarini, PhD Student @ University of Bologna

Introduction

In the last few years, **the scientific progress driven by computing technologies and innovations was noticeable.**

- AI/HPC implementations in *particle physics* [*]
- AI implementations in industry (*computer vision, LLMs, GenAI*) [**]
- Numerical simulations (for instance in *weather forecasting*) [***]

Electricity consumption from data centres and artificial intelligence (AI) (...) **could double** by 2026 (...)[****]

this clashes with SDGs!

* (Guest et al.) https://doi.org/10.48550/arXiv.1806.11484

** https://l.infn.it/st (PwC report on AI for businesses)

*** https://www.ecmwf.int/en/forecasts

**** https://www.iea.org/reports/electricity-2024

Introduction

The physical limits of Dennard Scaling***** have been essentially reached and transistors cannot get any smaller. This means that from now on efficiency is mainly the result of developers' and computer architects' effort.

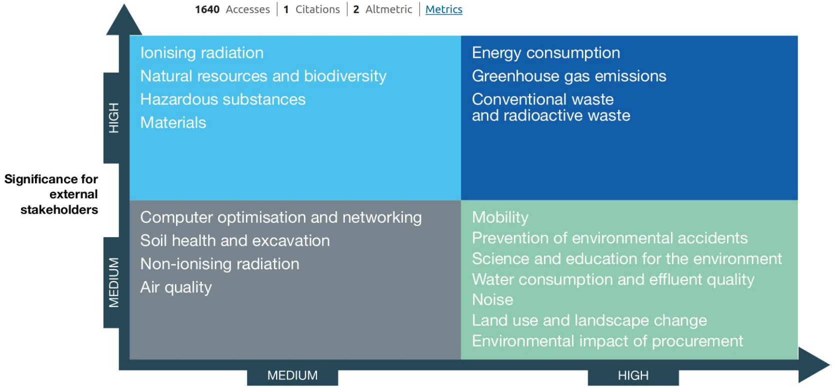
Given that computing is here to stay, we need to increase our "computing footprint awareness".

Editorial | Published: 07 March 2023

Physics should acknowledge its environmental impact and act on it

Nature Reviews Physics **5**, 133 (2023) | Cite this article

1640 Accesses | **1** Citations | **2** Altmetric | Metrics



CERN MATERIALITY MATRIX 2022
The topics identified as being of lower significance to all stakeholders are not comprehensively covered in this report but are subject to monitoring by CERN.

* 10.1109/JSSC.1974.1050511

Theoretical Formula

[*] offers a formula for estimating the **energy consumption of a computing activity A over a generic resource X :**

$$E_{(A \to X)} = T \times (n_c \times P_c \times u_c + n_m \times P_m) \times PUE$$

- T = Elapsed computing time (h)
- n_c = number of used cores
- P_c = power draw normalized to computing cores (kW)
- u_c = CPU usage factor -> [0,1]
- n_m = allocated RAM memory (GB)
- P_m = Power draw of RAM (kW)
- PUE = Power Usage Efficiency of the cluster

* (Lannelongue et al.) https://doi.org/10.1002/advs.202100707

Theoretical Formula

Given the previous formula, it is also possible to estimate the **Carbon Footprint:**

$$F = E_{(A \rightarrow X)} \times CI$$

CI = **Country-wise Carbon Intensity coefficient of electricity production [gCO2/kWh]**

Please note:

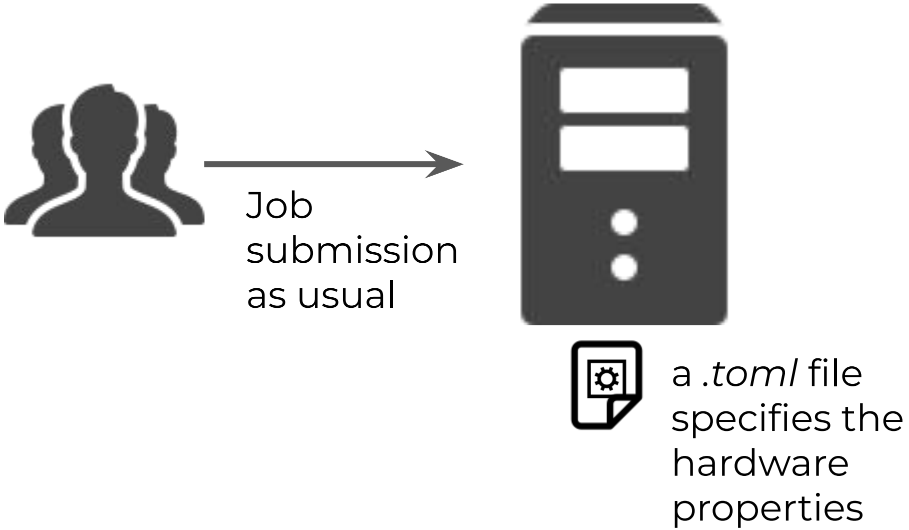
Carbon Intensity strictly depends on how energy is produced *and* exchanged on the energy market. Consequently, carbon footprint should be used carefully.

Implementing the formula - getting the data in RT

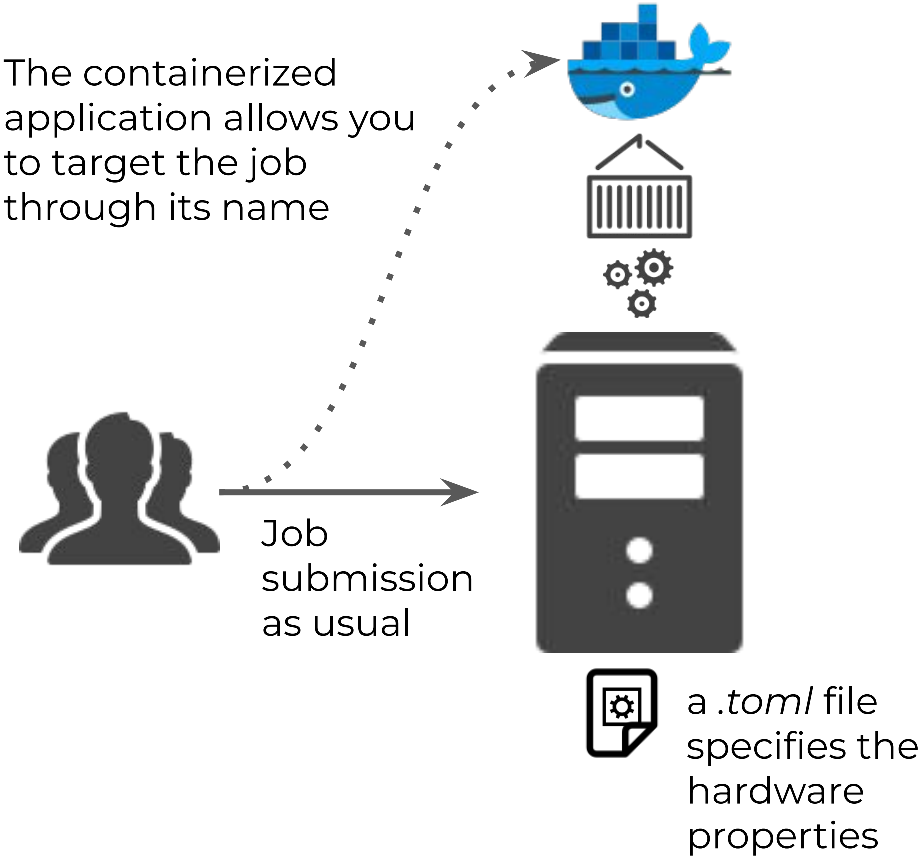
- u_c and n_m can be calculated by leveraging the uniqueness of *PIDs* on Linux and appropriately parsing */proc/<PID>/stat and /status* files. In this way, the monitoring is also "task-agnostic".
- P_c, P_m and PUE are "architectural" data. We gather it in a config file. In this way we can update hardware-related data through a single file.
- T, the elapsed computing time, can be easily measured with simple functions. Many programming languages can do that fairly accurately.

We build a Docker container so that we can get this data without altering privileges.

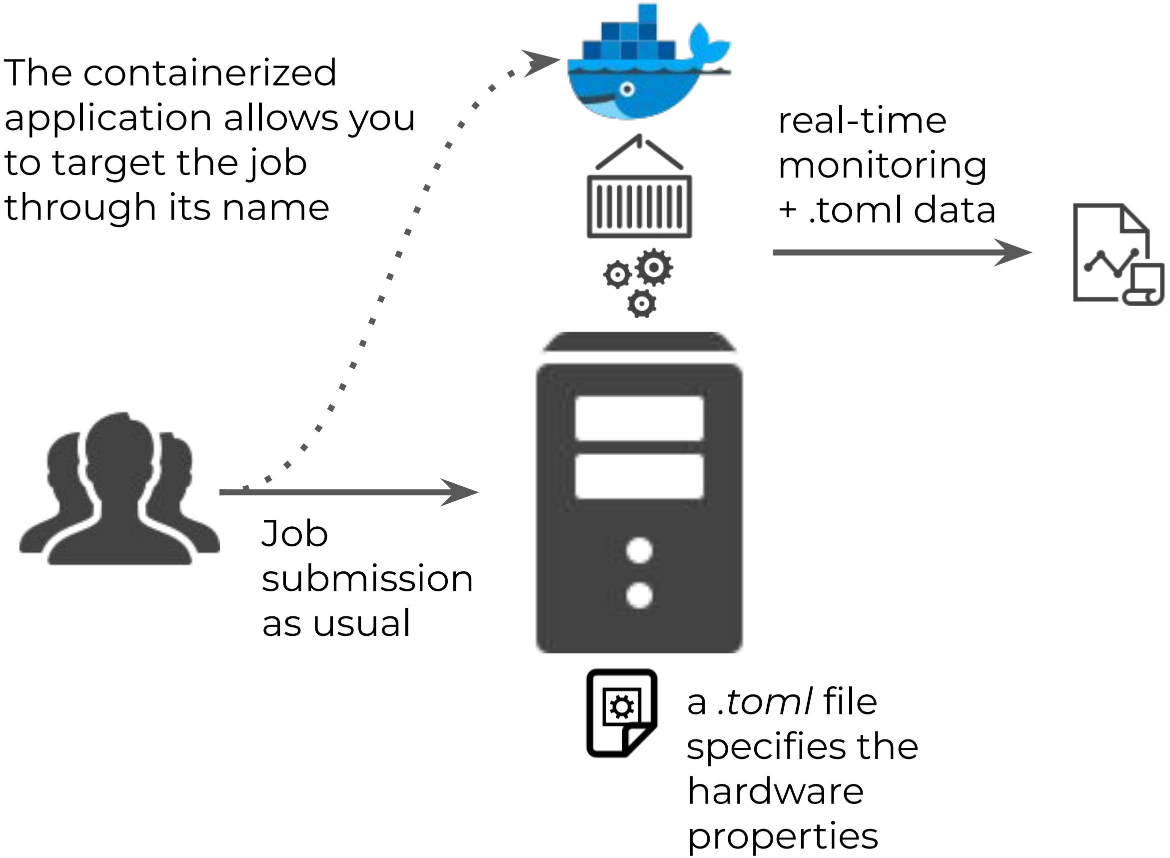
The monitoring at work



The monitoring at work



The monitoring at work



Let's try it!

Experimental Setup

The monitoring was deployed over some containerized reference CMS workloads* in order to test the software on realistic HEP payloads.

I will present only GENSIM data for brevity.

GENSIM: generation and simulation of TTbar events at 14 TeV and 2021 CMS detector for the Run3 era. The workload executes a **CMSSW GENSIM job**, which **embeds the event generation and the Geant4 simulation event by event**. **CMSSW is a multithreaded application**; the default number of threads is 4 and the default number of copies is the number of cores divided by 4.

* https://doi.org/10.1007/s41781-021-00074-y
https://gitlab.cern.ch/hep-benchmarks/hep-workloads

Experimental Setup

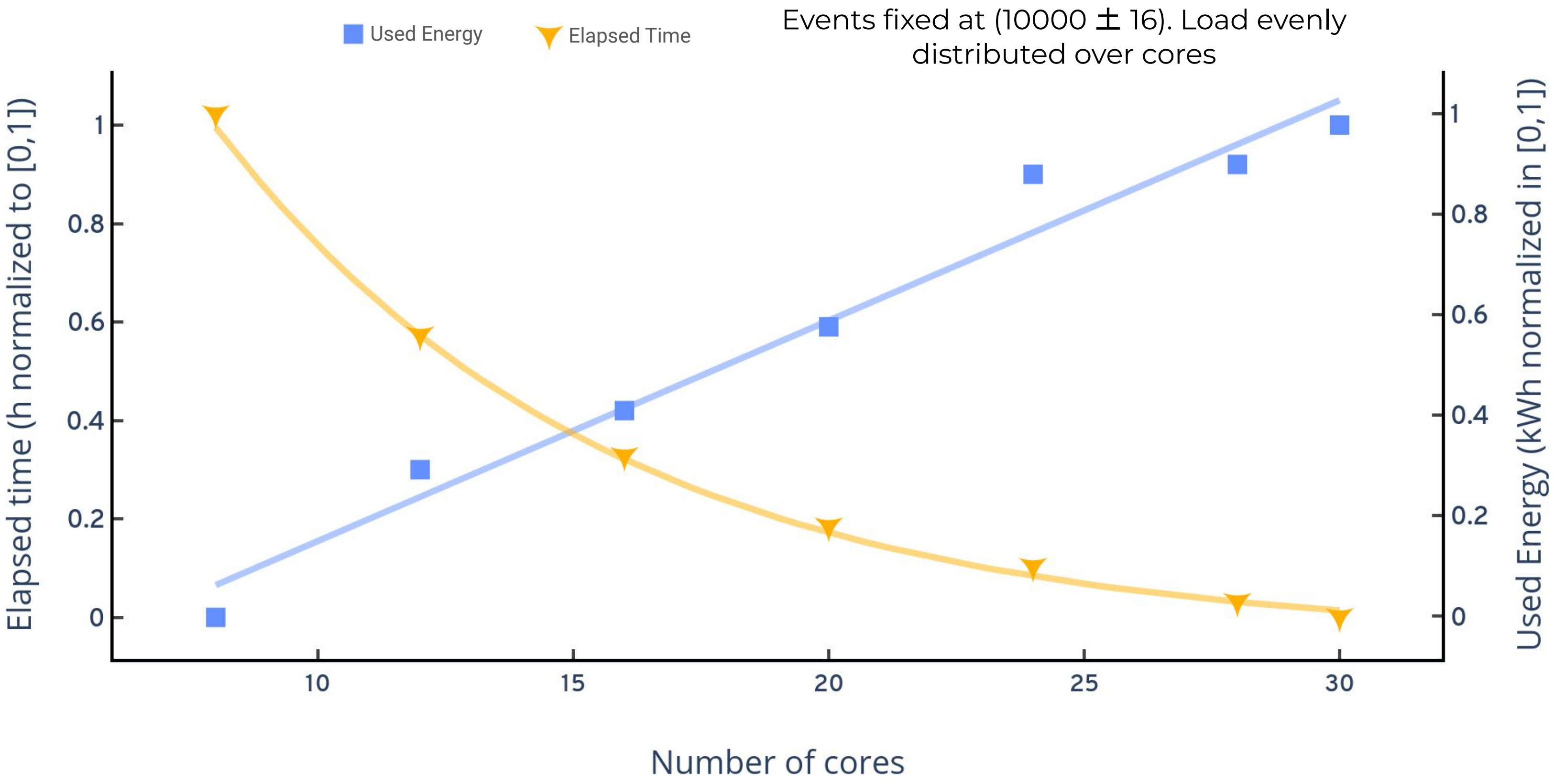
A Slurm node from INFN-CNAF cluster:

CPU Name	Physical Cores	Hyperthreading	RAM(GBs)
Intel Xeon CPU E5-2640 v2	16 (2x sockets 8 cores each)	YES	128

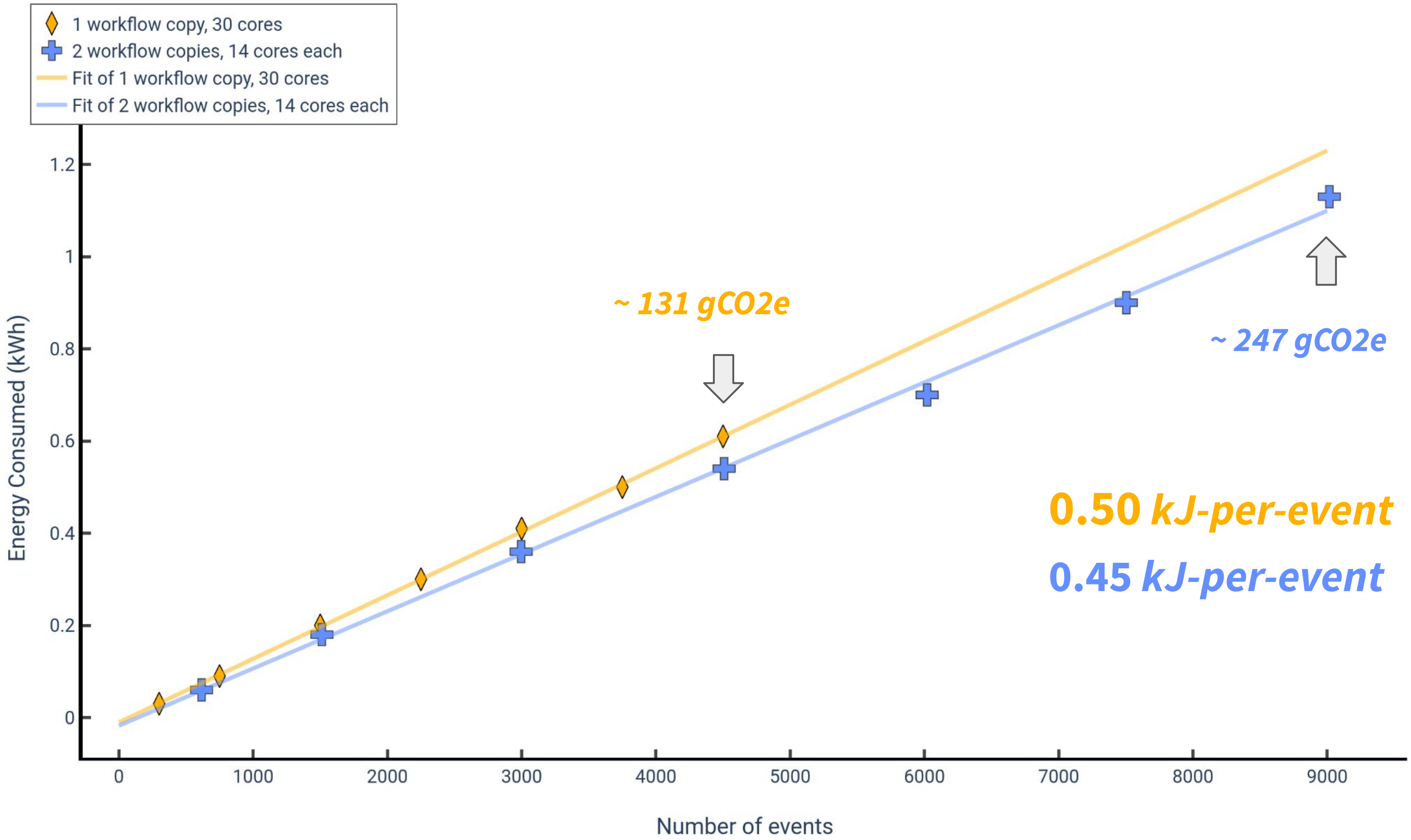
Research question

Can we estimate the energy consumption of HEP workflows/algorithms and find potential margins of improvement for it?

Results - optimal Gensim working point



Results - Submission “tuning”



Conclusions

- **The tool can measure in RT the energy consumption of an arbitrary computing task**.
- The optimal working point plot seems to suggest that **we can “tune” job submissions from a energy + carbon perspective** and potentially “squeeze” more performance from our cores.

Next steps

- Estimation of errors of measurements and potential biases of the method
- *Can storage really be considered “out of this picture”?*
- Apply monitoring over AI4HEP applications artifacts to investigate the impact of AI algorithms.

Thank you!

Contacts:

francesco.minarini3@unibo.it

fminarini

Acknowledgements:

I want to thank INFN-CNAF and its computing farm team for allowing me to use their resources and providing technical support throughout the process.

Example of .toml configuration file

```
# TOML configuration of HPC node under analysis

[owner]
name = "owner"
title = "job"

[infrastructure]
root_folder = "/proc/"
cpu_stat_file = "/stat"
mem_stat_file = "/status"

cpu_family = "IvyBridge"
cpu_tdp = 12
n_cpu = 15
clock_ticks = 100
ram_freq = 1600
ram_size = 1

[energy]
carbon_intensity = 400.0
power_usage_efficiency = 1.65
```

Please note:
these numbers are just for display

Example of submission - target task

```
#!/bin/bash
#
#SBATCH --job-name=<...>
#SBATCH --output=<...>
#SBATCH --nodelist=<node>
#SBATCH -n <N>

singularity exec -C --no-home -B results:/results  --tmpdir /home/HPC/fminarinihpc/temp docker://gitlab-registry.cern.ch/hep-be
nchmarks/hep-workloads/cms-gen-sim-run3-ma-bmk /bmk/./cms-gen-sim-run3-ma/cms-gen-sim-run3-ma-bmk.sh -c 1 -t 15 -e 300
```

Example of submission - monitoring

```
singularity exec -i --no-home -B ~/config.toml:/conf/config.toml docker://francescominarini/kig:egotistic_stendhal ./home/kig_u
ser/KIG_ex $(pgrep -f "cmsRun ./TTbar_14TeV_TuneCP5_cfi_GEN_SIM.py" | awk 'ORS=" "' ; pgrep -f "cmsRun ./TTbar_14TeV_TuneCP5_cf
i_GEN_SIM.py" | xargs -I _ pgrep -f "cmsRun ./TTbar_14TeV_TuneCP5_cfi_GEN_SIM.py" | awk 'ORS=" "')
```

Optimal working point in DIGI