Keep IT Green

KIG: a tool for Carbon footprint monitoring in physics research

Francesco Minarini, PhD student in Physics @University of Bologna, Italy and INFN

research area: Green Computing for High Energy Physics

International Symposium on Grids & Clouds (ISGC) 2023, Academia Sinica, Taipei, Taiwan.

03/24/2023

Outline

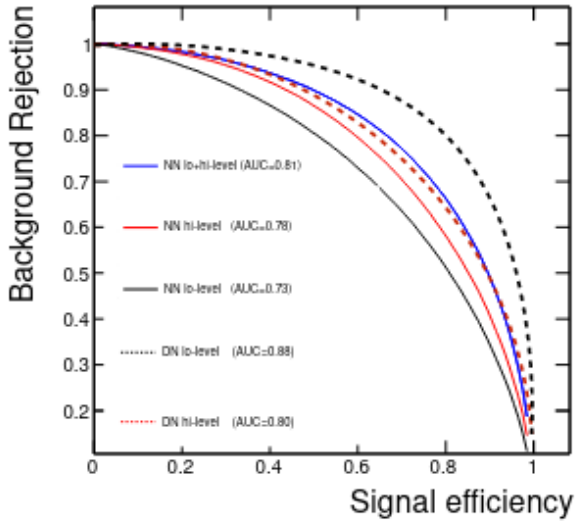
- Motivation
- Energy and Carbon footprint formula
- Design of KIG
- KIG Test-bed
- Results
- Conclusions & Next Steps

Introduction (1of2)

In the last few years, **the scientific progress driven by High-Performance Computing (HPC) technologies was noticeable.** Hardware/software improvements have granted advances throughout a wide variety of disciplines by allowing the implementation algorithms of *higher and higher* computational intensiveness.

- **AI/HPC implementations in *particle physics* [*]** have been successful in a wide number of processes (*i.e. Simulation, Event Selection, Event Reconstruction, Jet classification, BG rejection*).
- **Also other branches of physics have benefitted from HPC, for instance *meteorology*,** where *numerical weather forecasting* has produced simulations of unprecedented detail and reliability.
- **Also businesses are starting to heavily rely on AI and HPC.** Several reports **[**]** show that a majority of companies are adopting AI tools to improve and/or consolidate their position on the market.

https://doi.org/10.1038/ncomms5308



[*] https://doi.org/10.48550/arXiv.1806.11484
[**] https://l.infn.it/st

Introduction (2of2)

So far, the constant evolution of technology (in terms of power and energy-efficiency) has allowed us to keep up with computing requirements and "naturally" curb the energy consumption.

Moore's law: The number of transistors in an integrated circuit *doubles* approx. every 2 years

Dennard Scaling: As transistors get smaller, the power density stays constant = power use stays in proportion with area of chip.

Koomey's law: The number of computations per joule *doubles* every 1.57 years

These laws, Dennard's in particular, **are starting to weaken [4].**

Meanwhile, the **HPC paradigm is gaining momentum** not only in the scientific environment, but also in sectors that historically had moderate interest in high-performance computing.
As a result, the **demand** for HPC resources **will increase** in the near future **=> energy consumption (and Carbon Footprint) will spike,** making computing less "cost-effective" w.r.t. today (*energy is already becoming a problem worldwide*).

The slowdown of hardware progress clashes with the spike of interest in HPC.
Understanding the footprint of modern computing today in order to set up energy curbing strategies in time to ensure future research and their overall accessibility **[1,2].**

Energy and Carbon footprint formula

Lannelongue et al. **[3]** propose a formula to calculate the **energy footprint of some computing activity *A* launched on 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 of computing cores(kW)

u_c = CPU usage factor -physically bound in [0,1]-

n_m = allocated RAM memory(GB)

P_m = Power draw of RAM (kW)

PUE = Power Usage Efficiency of the machine/cluster

The Carbon footprint of the computation is:

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

CI = country-wise Carbon Intensity coefficient (gCO2e* footprint of power-grid electricity).
In the following, Italy's CI will be used for calculations.

*Wright, L.A. et al. ***"Carbon footprinting: towards a universally accepted definition"***, Carbon Management, 2014.

KIG design (1of2)

- **an up-and-running computing node has a non-negligible computational clutter.** The monitoring should not include that.
- Furthermore, **some available tools** for detailed CPU-monitoring **are proprietary** (i.e. *Intel Power Gadget, AMD system monitor)*. This might lead to a **"vendor lock-in"** on the long run.
- Finally, **the monitoring should not require, at any step, to alter the user's set of privileges.** This is coherent with the best-practice of keeping user privilege as limited as contextually possible.

Technical approach:

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

- 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.**
- P_c, P_m and PUE are "bare metal" data and can be gathered in an appropriate external configuration file. This way, if hardware is changed (or requirements change), we just need to update a simple file.
- T, the elapsed computing time, can be easily measured with simple commands. Many programming languages can do that accurately

KIG design (2of2)

we can build a small node-level monitoring library+executable whose duty is to monitor PIDs we are interested in by gathering data involved in the equation *(A)*. To do that, we parse `/proc/PID/stat` and `status` files and integrate this info with bare metal data.

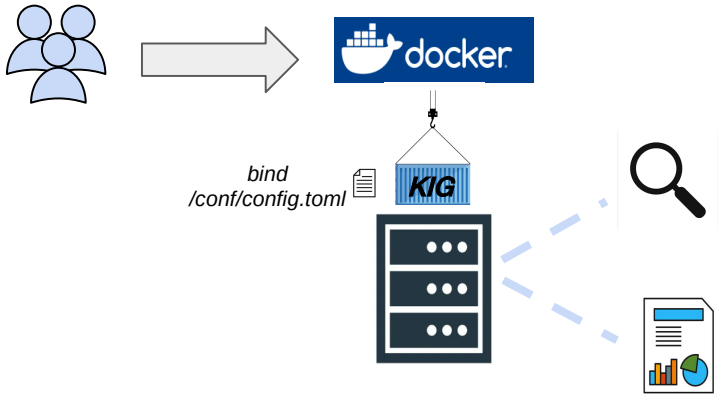
The language chosen for this *PoC* was **C++. The repo of the project will be public and MIT licensed:**

https://github.com/fminarini/KIG

https://fminarini.github.io/KIG/html/index.html

(Recommended) For ease of use and compliance with security standards at computing facilities, **a Docker image (Ubuntu flavoured) was also published on Docker Hub.** *(Docker-Apptainer compatibility was tested)*

https://l.infn.it/sr



Service Mock-up: Users interact with docker to obtain the container. Once the container is up and running with an appropriate configuration file bound to it, users only need to start the monitoring program. At the end of the process, users obtain:

1) A report of consumed W, Elapsed computing time (h) and the carbon footprint of the activity (gCO2e).

2) (if user has write permissions) A text file containing the LaTeX structure of a sum-up table for easy reporting in papers.

KIG Test-bed (1of2)

The monitoring was deployed over some containerized reference CMS workloads (GENSIM, DIGI, RECO)* in order to test the software on realistic payloads.

The choice of CMS was merely opportunistic, the **monitoring** is designed to be **experiment-agnostic.**

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.

DIGI: CMS **reference workload executing digitisation of TTbar events at 14 TeV and 2021 CMS detector with a Run3 setup, using premixed events for the pileup.** The workload executes a CMSSW DIGI job. 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.

RECO: CMS reference workload executing reconstruction of TTbar events at 14 TeV and 2021 CMS detector with a Run3 setup. The workload executes a CMSSW RECO job. 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://link.springer.com/article/10.1007/s41781-021-00074-y

KIG Test-bed (2of2)

2 machines from the INFN-CNAF High-Throughput farm were available as a test-bed.

172 HS06 was available for an extended period of time.
1293 HS06 was available for a restricted period of time.

Measurements were taken at node-level **emulating the behaviour of a user submitting the payload to a single-node.**

	CPU	physical Cores (Total)	Hyperthreading	RAM(GB)
172 HS06	2x AMD opteron 6320	16	NO	128
1293 HS06	2x AMD EPYC 7313	32	YES	128

- **Opteron 6320:** https://www.amd.com/en/products/cpu/6320
- **Epyc 7313:** https://www.amd.com/en/product/10991

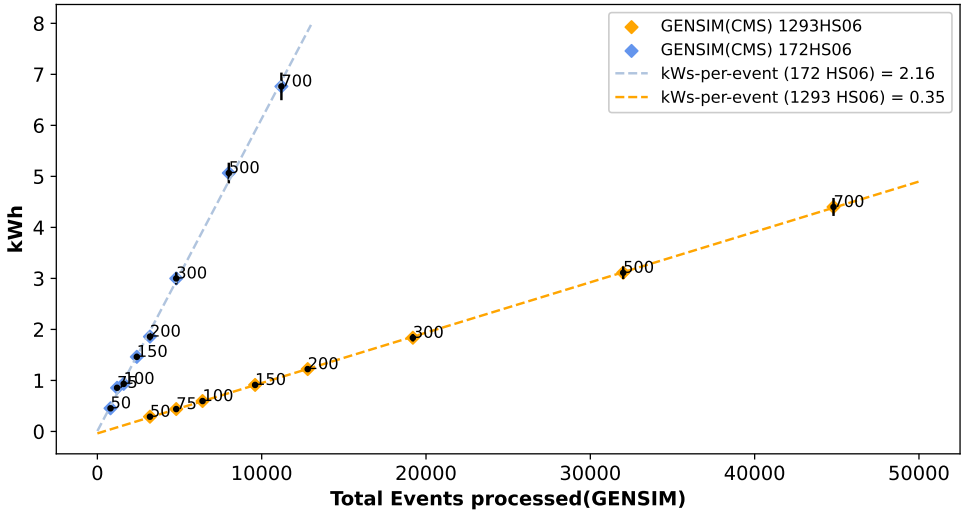
-

Results

- **Comparison of two machines with full usage of cores against the same workload**
  - *Validation of the concept **if** linear trends emerge wrt processed events and difference of machines is highlighted*
- **Footprint characterization of a workload running on a varying the number of requested cores (172 HS06)**

GENSIM-flow runs with different events-per-thread (EPT, tagged over points) configurations

Thread-per-core (TPC) = 4 (fixed), indip.copies (IC) = nCores/4, total events: TPC x IC x EPT



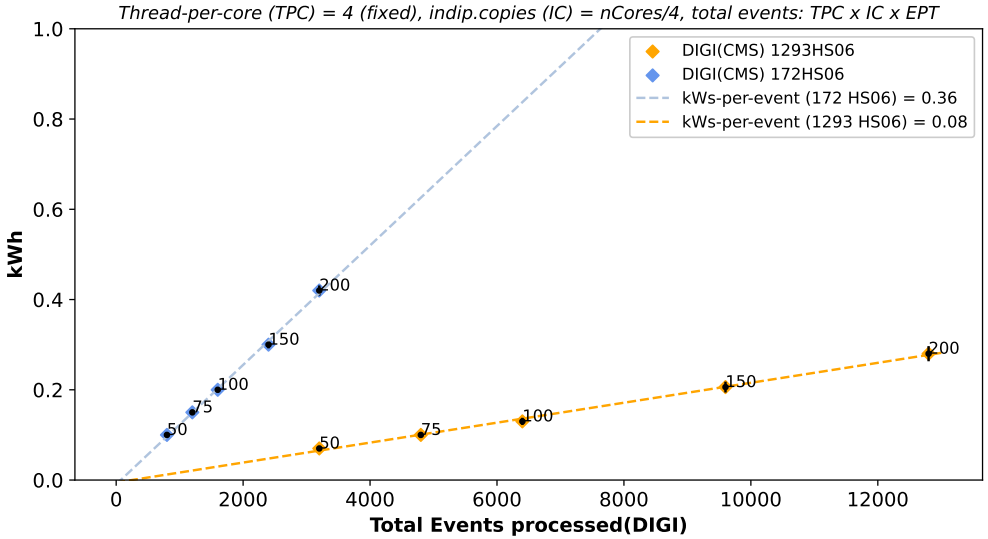
GENSIM	172 HS06			1293 HS06		
	Elapsed Time(h)	*Energy (kWh)*	*Total Events*	*Elapsed Time(h)*	*Energy (kWh)*	*Total Events*
50	0.6	0.42	800	0.27	0.29	3200
75	1.27	1.08	1200	0.39	0.44	4800
100	1.15	1.33	1600	0.52	0.59	6400
150	1.07	1.28	2400	0.77	0.91	9600
200	2.2	2.25	3200	1.01	1.22	12800
300	3.36	3.23	4800	1.5	1.83	19200
500	6.29	5.06	8000	2.5	1.24	32000
700	8.13	7.16	11200	3.52	4.4	44800

kWs-per-event derived from the angular coefficient of best-fit line multiplied by 3600 in order to easily display it in **kJ per event.**

The fluctuation of measured values over repeated measures at same conditions (± 4%) was taken as an early-estimation of errors.

The intercept of fit is fairly compatible with 4% fluctuations around 0.
Intercept(172 HS06) = 0.007
Intercept(1293 HS06) = -0.038

DIGI-flow runs with different events-per-thread (EPT, tagged over points) configurations



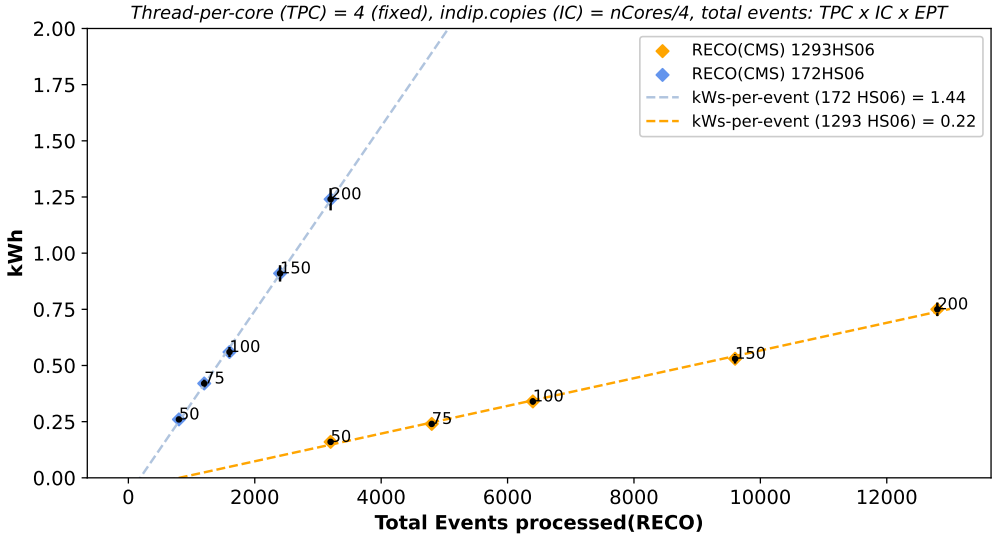
kWs-per-event derived from the angular coefficient of best-fit line multiplied by 3600 in order to easily display it in **kJ per event**

The fluctuation of measured values over repeated measures at same conditions (± 4%) was taken as an early-estimation of errors.

The intercept of fit is fairly compatible with 4% fluctuations around 0.
Intercept(172 HS06) = -0.009
Intercept(1293 HS06) = -0.005

DIGI	172 HS06			1293 HS06		
	Elapsed Time(h)	*Energy (kWh)*	*Total Events*	*Elapsed Time(h)*	*Energy (kWh)*	*Total Events*
50	0.15	0.10	800	0.07	0.07	3200
75	0.21	0.15	1200	0.09	0.1	4800
100	0.27	0.20	1600	0.11	0.13	6400
150	0.41	0.32	2400	0.18	0.20	9600
200	0.52	0.42	3200	0.24	0.28	12800

RECO-flow runs with different events-per-thread (EPT, tagged over points) configurations

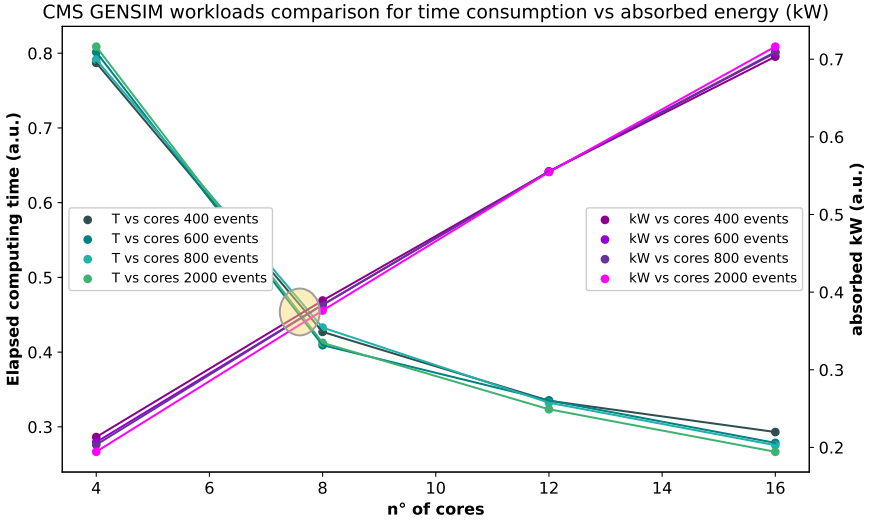


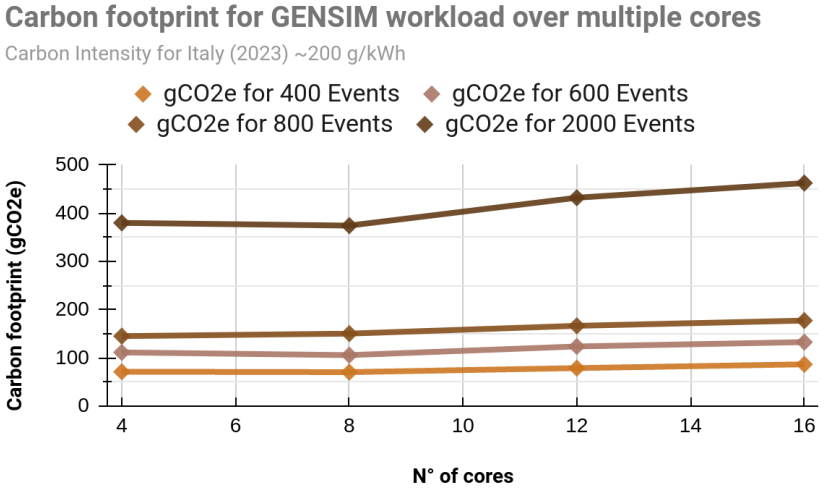
RECO	172 HS06			1293 HS06		
	Elapsed Time(h)	*Energy (kWh)*	*Total Events*	*Elapsed Time(h)*	*Energy (kWh)*	*Total Events*
50	0.40	0.26	800	0.16	0.15	3200
75	0.58	0.42	1200	0.24	0.24	4800
100	0.75	0.57	1600	0.3	0.34	6400
150	1.14	0.91	2400	0.47	0.54	9600
200	1.51	1.24	3200	0.64	0.75	12800

kWs-per-event derived from the angular coefficient of best-fit line multiplied by 3600 in order to easily display it in **kJ per event**

The fluctuation of measured values over repeated measures at same conditions (± 4%) was taken as an early-estimation of errors.

The intercept of fit is slightly out of the fluctuation boundary.
Intercept(172 HS06) = 0.077
Intercept(1293 HS06) = -0.049



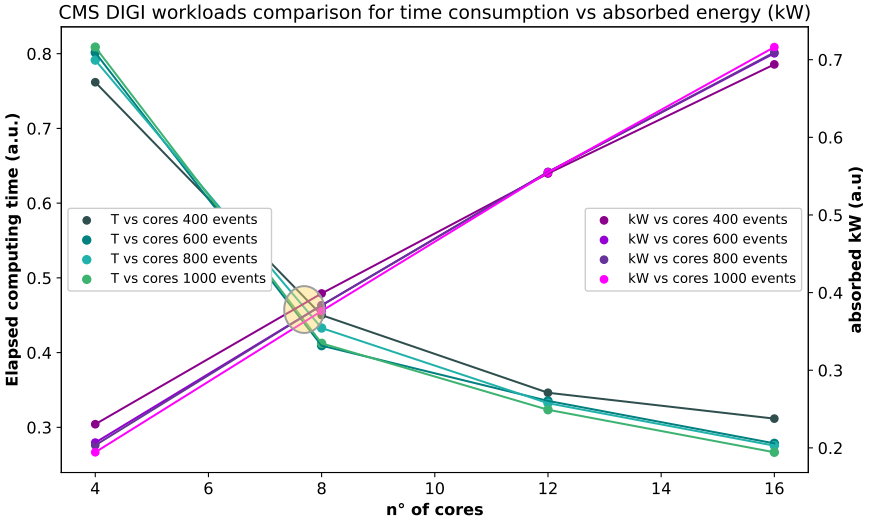


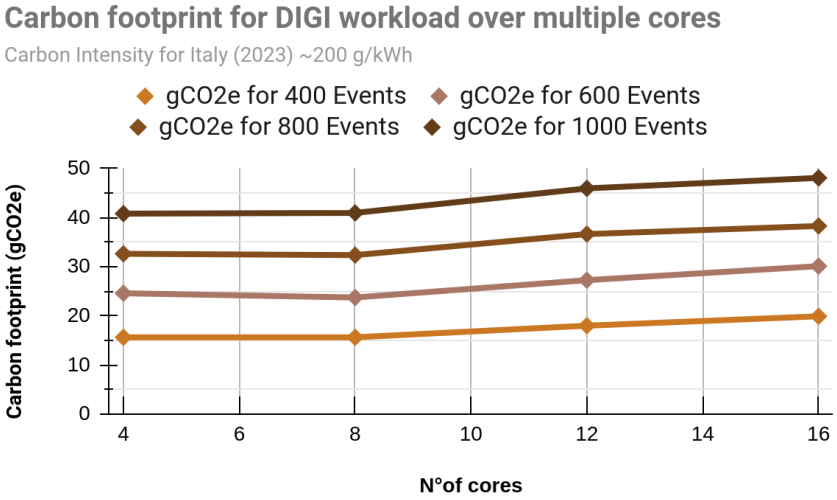
Data was gathered running 4 separate experiments, gradually saturating cores (keeping the total number of processed events constant).

left: Green-gradient lines = **Elapsed computing time** while augmenting the number of working cores *(normalized)*.
Purple-gradient lines = **kW absorption** while requesting more cores *(normalized)*.

right: Report of Carbon footprint for CMS GENSIM workloads. At 8 core we see the lowest footprint

The "8 core tradeoff point" **cannot** be used as a data center recommendation, obviously.
The observation triggers, nonetheless, some further questions (is this working point software-bound or hardware-bound? can we use it to better manage submissions?)



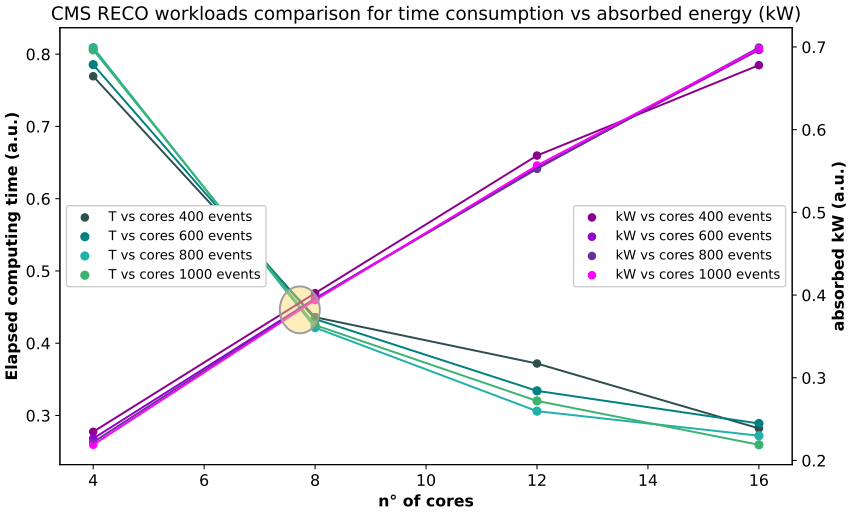


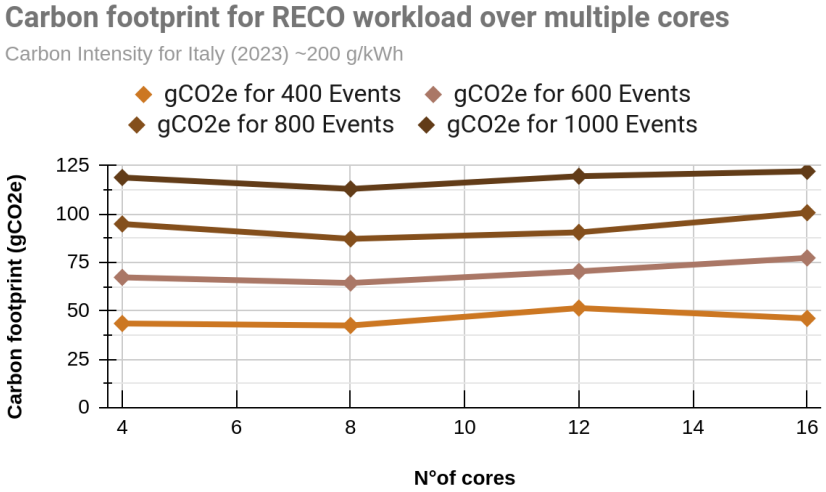
Data was gathered running 4 separate experiments, gradually saturating cores (keeping the total number of processed events constant).

left: Green-gradient lines = **Elapsed computing time** while augmenting the number of working cores *(normalized)*.
Purple-gradient lines = **kW absorption** while requesting more cores *(normalized)*.

right: Report of Carbon footprint for CMS DIGI workloads. At 8 core we see the lowest footprint.

The "8 core tradeoff point" **cannot** be used as a data center recommendation, obviously.
The observation triggers, nonetheless, some further questions (is this working point software-driven or hardware-driven?, can we use it to better engineer submissions?)





Data was gathered running 4 separate experiments, gradually saturating cores (keeping the total number of processed events constant).

left: Green-gradient lines = **Elapsed computing time** while augmenting the number of working cores *(normalized)*.
Purple-gradient lines = **kW absorption** while requesting more cores *(normalized)*.

right: Report of Carbon footprint for CMS RECO workloads. At 8 core we see the lowest footprint.

The "8 core tradeoff point" **cannot** be used as a data center recommendation, obviously.
The observation triggers, nonetheless, some further questions (is this working point software-driven or hardware-driven? can we use it to better engineer submissions?)

Conclusions

- The comparison between two different machines allowed us to have a validation of the monitoring and a **quantitative perspective of the difference between old platforms and newer ones in terms of energy and throughput.** Fitting data gave us back a ***kWs-per-event parameter*** which **might be included in a more "HEP-oriented" characterization of machines.**

- **Focusing** on the impact of core demand in terms of **Time-Energy tradeoff and Carbon footprint** gave us back a **clearer perspective on the footprint of common HEP processes**, which can trigger further analyses to improve the way we manage submissions as well as facilities.

Next Steps

- Test workloads from **different experiments or physics branches** *(Data -> Design -> new Computing)*
- **scale from node level to cluster level.**
- Explore **different computing platforms** (i.e. non x86 architectures) and accelerators (GPUs, for instance).
- Try using the output of KIG (kWs-per-event, Time-Energy working points) to **ponder** on **the way we model and manage submission.**

THANKS FOR THE ATTENTION

francesco.minarini3@unibo.it

https://github.com/fminarini/KIG

Acknowledgements:

- Daniele Bonacorsi (UNIBO), Domenico Giordano (CERN), Andrea Valassi (CERN) for early feedback over results and PoC design.
- INFN-CNAF and INFN-CNAF IT Support for machine provisioning.

References:

[1] Schwartz, R., Dodge, J., Smith, N., Etzioni, O. **"*Green AI*"** https://arxiv.org/abs/1907.10597

[2] Strubell, E., Ganesh, A., mcCallum, A. ***"Energy and policy considerations for Modern Deep Learning Research"*** https://doi.org/10.1609/aaai.v34i09.7123

[3] Lannelongue, L., Grealey, J., Inouye, M. **"*Green algorithms: Quantifying the Carbon Footprint of Computation*"** https://doi.org/10.1002/advs.202100707

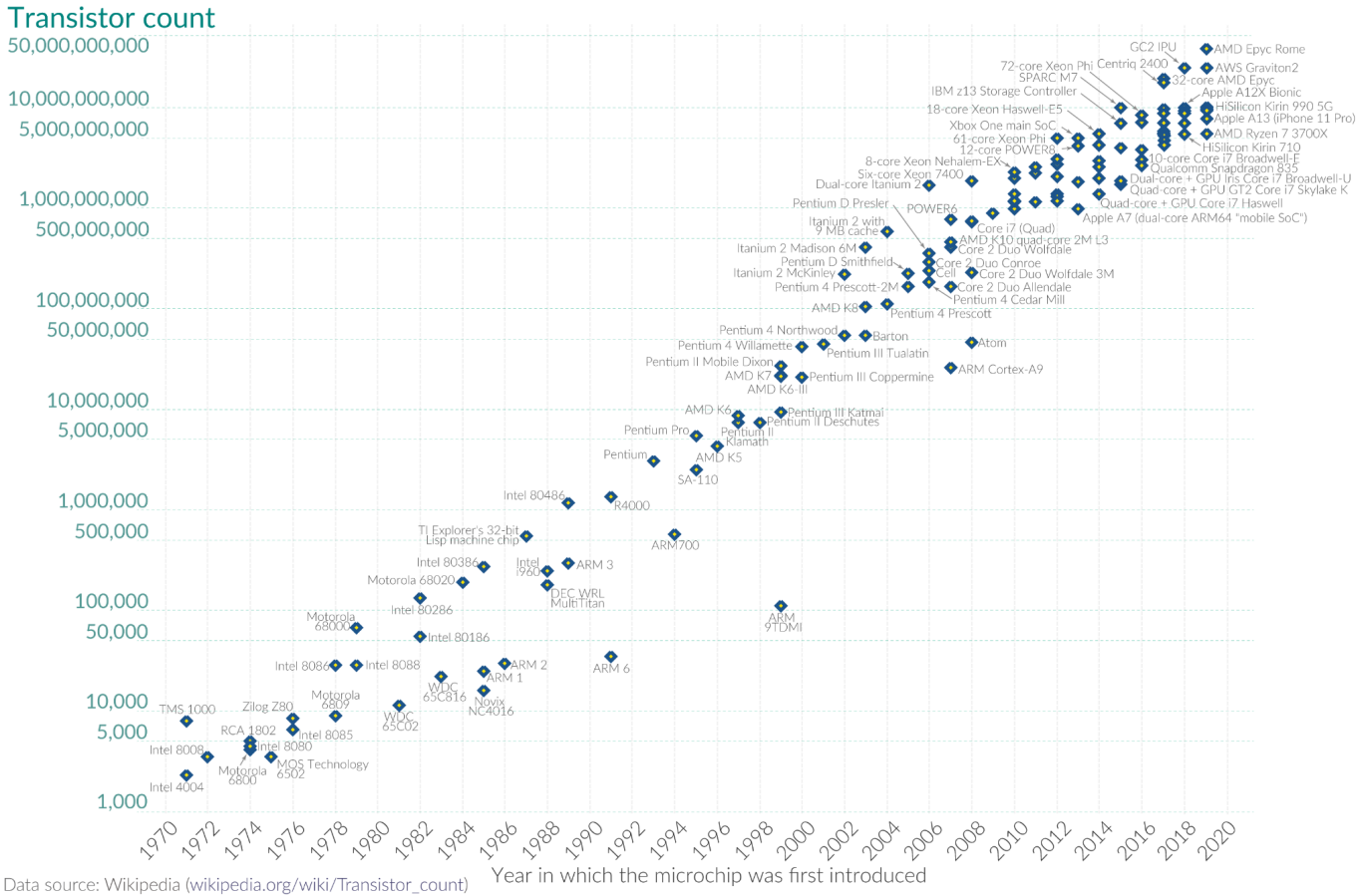
[4] Horowitz, M. ***"Computing's Energy problem (and what we can do about it)"*** https://hal.science/hal-02549565v4

ADDITIONAL MATERIALS

Moore's Law: The number of transistors on microchips doubles every two years

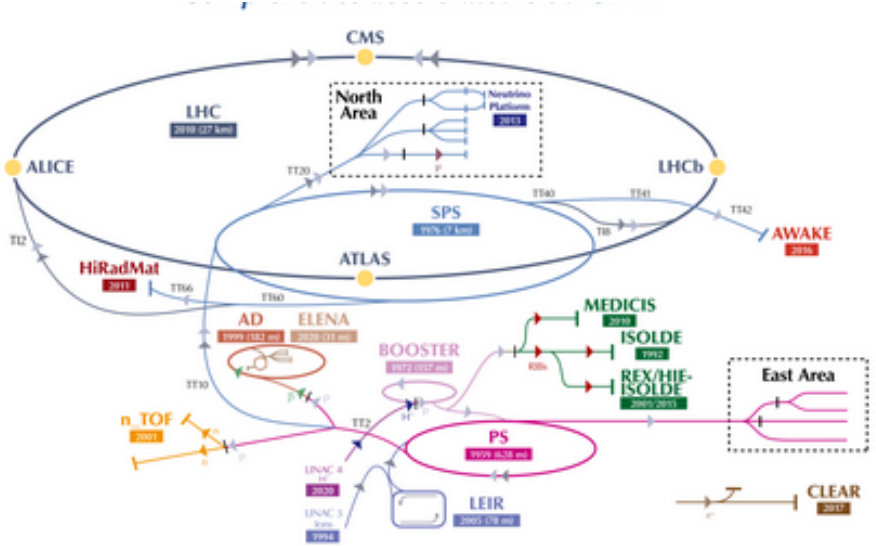
Our World in Data

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years.
This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.



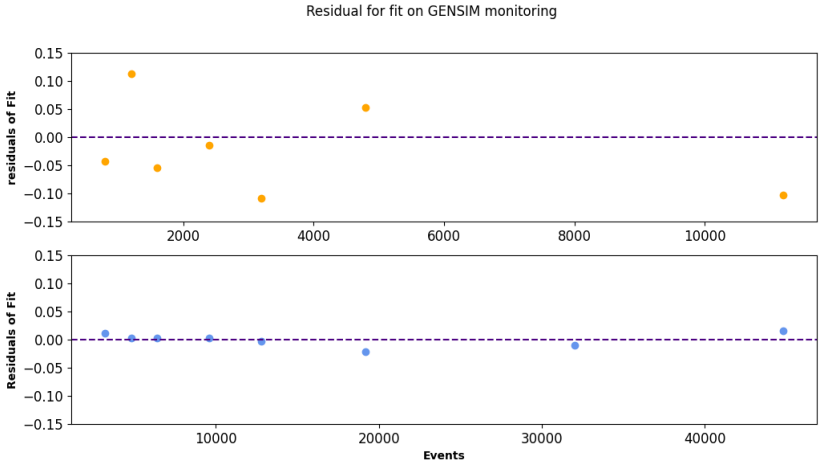
Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)

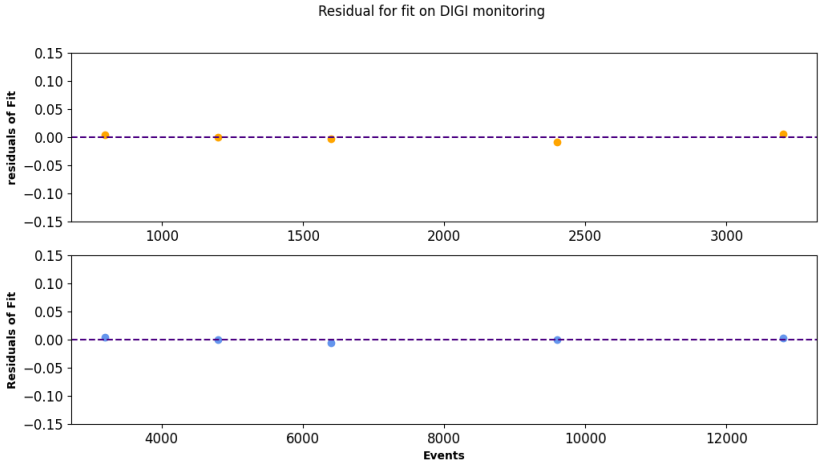
CMS @ LHC (few details)

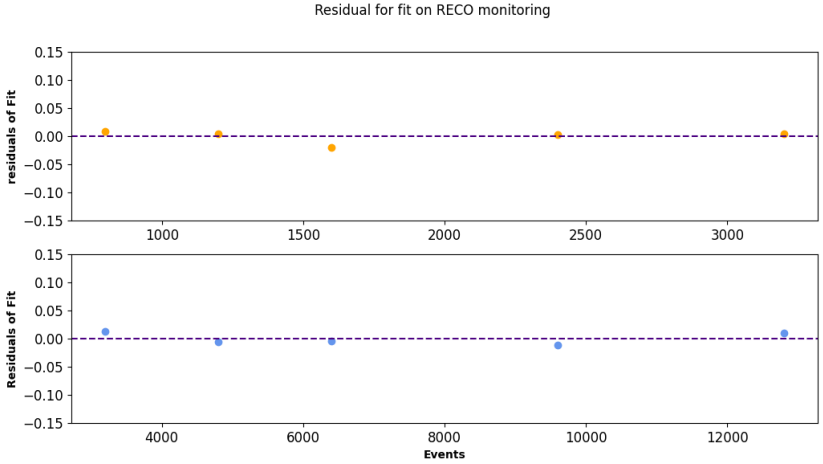


The Compact Muon Solenoid (CMS) is a general-purpose detector at the Large Hadron Collider (LHC). It has a broad physics programme ranging from studying the Standard Model (including the Higgs boson) to searching for extra dimensions and particles that could make up dark matter.

Goodness-of-fit







Using containerized KIG (CLI commands)

A more “in-depth” documentation is available in the Github Repository, in the following a brief representation of a quick-start with KIG.

```
docker pull francescominarini/kig:latest
```

“App-like” usage

```
docker run -dit --name <container_name> --pid host \
--mount type=bind,source="$(pwd)"/<folder_with_toml_file>,target=/conf <IMAGE_NAME> bash

docker exec -it <container_name> ./KIG_ex $(pgrep -f "<monitored_activity>" | awk 'ORS=" "')
```

“interactive” usage

```
docker run -it --name <container_name> --pid host \
--mount type=bind,source="$(pwd)"/<folder_with_toml_file>,target=/conf <IMAGE_NAME> bash
```

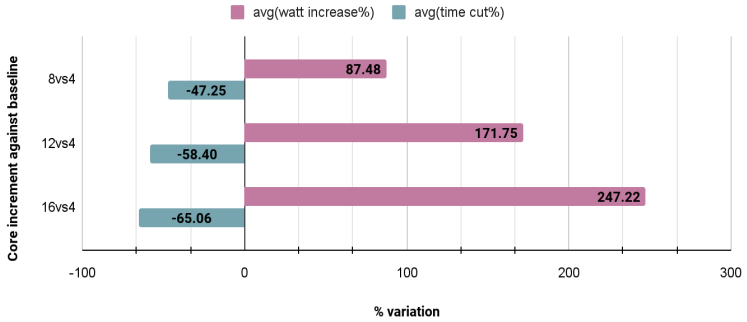
Using containerized KIG (config.toml)

```
# TOML document for my personal PC emulating the conf for a cluster

[owner]
name = "Francesco Minarini"
title = "prova"

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

cpu_family = "Skylake"              #cpu_family (useful for comparisons)
cpu_tdp = 8                        _#value in W per-cpu
n_cpu = 2                           #number of usable physical chips on board.
clock_ticks = 100                   #getconf CLK_TCK, gives equivalence to seconds
ram_family = "DDR4"                 #ram family (useful for comparisons)
ram_freq = 2133                     #ideally frequency tells you the wattage required
ram_slots = 1                       #active ram slots

[energy]
carbon_intensity = 220              #regional impact of electricity prod and consumption
power_usage_efficiency = 1.8        #avg italian PUE
```

Energy-Time cut trade-off

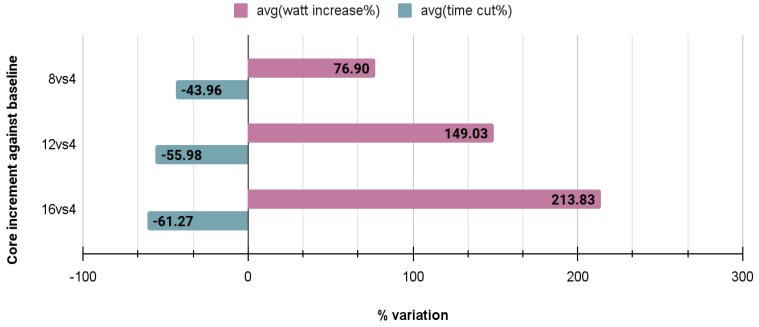
Watt absorption and Time consumption with increasing number of cores on GENSIM flow

Averages taken over 4 experiments with total events [400, 600, 800, 2000] against a 4 core baseline



Watt absorption and Time consumption with increasing number of cores on DIGI flow

Averages taken over 4 experiments with total events [400, 600, 800, 1000] against a 4 core baseline



Watt absorbtion and Time consumption with increasing number of cores on RECO flow

Averages taken over 4 experiments with total events [400, 600, 800, 1000] against a 4 core baseline