Towards Green HPC?

Camille Coti

École de Technologie Supérieure, Montreal, Canada

Scheduling Workshop 2025

Disclaimer

This talk is very similar to a talk I gave at the GreenNet workshop at ICC 2025

Roadmap

A lot of flops... a lot of watts?
- The Top500 and the Green500
- Current machines: example with Frontier

Modeling power consumption
- Distributed applications

Sharing GPUs
- What is the energy consumption?
- How to use the lessons learned?

Conclusion

Towards the exascale

Exascale = 10^{18} floating point operations per second

- That's a lot of machines!
- ... using a lot of electricity

Is it true?

Towards the exascale

Exascale = 10^{18} floating point operations per second

- That's a lot of machines!
- ... using a lot of electricity

Is it true?

- Department of Energy's Advanced Scientific Computing Advisory Committee (ASCAC), Exascale Computing Initiative Review, August 2015

The post-K system will also be built by Fujitsu and is expected to include advanced technologies such as 3D-stacked memory, an optical interconnect, and **aggressive power saving techniques to meet its 30 MW target.**

Towards the exascale

Exascale = 10^{18} floating point operations per second

- That's a lot of machines!
- ... using a lot of electricity

Is it true?

- Department of Energy's Advanced Scientific Computing Advisory Committee (ASCAC), Exascale Computing Initiative Review, August 2015

The post-K system will also be built by Fujitsu and is expected to include advanced technologies such as 3D-stacked memory, an optical interconnect, and **aggressive power saving techniques to meet its 30 MW target.**

These adaptations will be particularly important in addressing issues such suggested **limits on power (20 MW)** *and total memory capacity (128 PB).*

Where are we nowadays?

Ranking the most energy-efficient machines:

- Green500 rankings: `https://top500.org/lists/green500/2025/06/`
- June 2025

Green500	Top500	System	Power (kW)	Efficiency (Gflops/watts)
1	259	JEDI (GER)	67	72.733
2	148	ROMEO-2025 (FRA)	160	70.912
3	484	Adastra 2 (FRA)	37	69.098
4	183	Isambard-AI phase 1 (UK)	117	68.835
5	255	Otus GPU (GER)		68.177
6	66	Capella (GER)	445	68.053
7	304	SSC-24 Energy Module (SK)	69	67.251
8	85	Helios GPU (POL)	317	66.948
9	399	AMD Ouranos (FRA)	48	66.464
10	412	Henri (USA)	44	65.396

Where are we nowadays?

Awesome! How about the fastest machines?

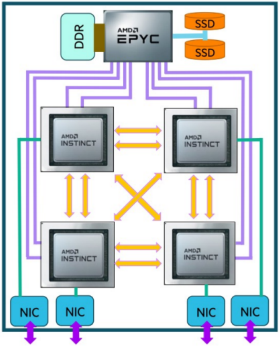
- Top500 rankings: `https://top500.org`
- June 2025

Top500	System	Power (kW)	Efficiency (Gflops/watts)
1	El Capitan (USA)	**29,580.98**	58.89
2	Frontier (USA)	**24,607.00**	54.98
3	Aurora (USA)	**38,698.36**	26.15
4	JUPITER Booster (GER)	13,088.23	60.62
5	Eagle (USA)	??	??
6	HPC6 (ITA)	8,460.90	56.48
7	Fugaku (JAP)	**29,899.23**	15.42
8	Alps (CH)	7,124.00	61.0
9	LUMI (FIN)	7,106.82	53.43
10	Leonardo (ITA)	7,493.74	32.19

Where does the power go?

Example: Frontier

- 9,472 AMD Epyc 7713 "Trento" 64 core 2 GHz, 3.7 GHz boost CPUs
  - 1 on each node
  - Maximum power consumption: 225-250 W
- 37,888 Instinct MI250X GPUs
  - 4 on each node, 2 compute dies on each GPU
  - Maximum power consumption: idle 100 W, at peak 500-540 W
- HPE Slingshot 64-port switches
  - 2 368 in total
  - 481 W +12.5 W per active optical cable



Sources: Frontier Architecture Overview, Subil Abraham, February 28, 2024
https://www.serversupply.com/ HPE Slingshot switches specs

Other hardware

SmartNICs: example **Nvidia Blue Field**

- *The DPU Controller maximum power consumption* **does not exceed 150W** *and is split between the two power sources as follows:*
  - *Up to 66W from the PCIe golden fingers (12V)*
  - *The rest of the consumed power is drawn from the external PCIe power supply connector*
- Source: NVIDIA BlueField-3 DPU Controller User Manual

PEZY-SC (Supercomputer): manycore accelerator, well ranked at Green500 circa 2017

- **PEZY-SC2**
  - 2,048 processing engine cores, 1 GHz
  - Average power consumption: 130 W
- **PEZY-SC3**
  - 4,096 processing engine cores, 1.2 GHz
  - Maxium power consumption: 500 W
- Source: `pezy.co.jp`

Roadmap

A lot of flops... a lot of watts?
- The Top500 and the Green500
- Current machines: example with Frontier

Modeling power consumption
- Distributed applications

Sharing GPUs
- What is the energy consumption?
- How to use the lessons learned?

Conclusion

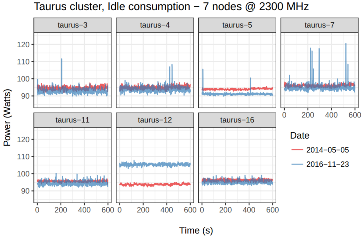
Predicting the energy consumption

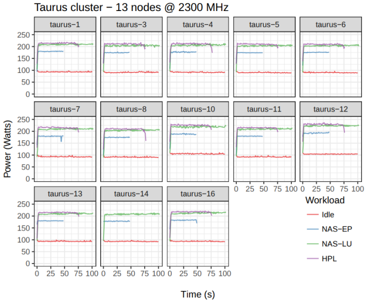
Model the energy with a static part (idle) and a dynamic part (depending on the load):

$$P_{i,f,w}(u) = P_{i,f}^{static} + P_{i,f,w}^{dynamic} \times u$$

i, f, w, u: machine, frequency, workload, usage

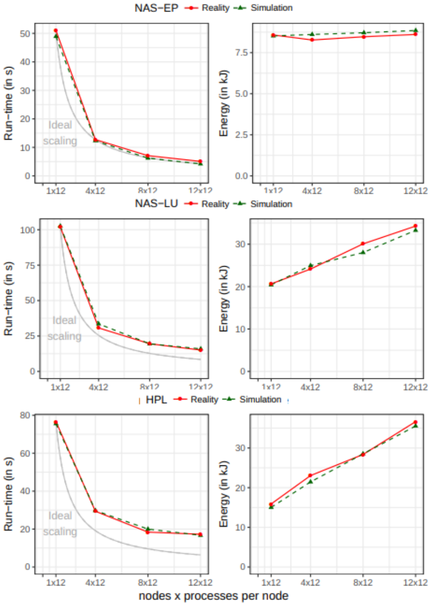
but... it varies a lot between machines and with time





Source: Franz C. Heinrich, Tom Cornebize, Augustin Degomme, Arnaud Legrand, Alexandra Carpen-Amarie, et al.. Predicting the Energy Consumption of MPI Applications at Scale Using a Single Node. Cluster 2017, IEEE, Sep 2017, Hawaii, United States.

Calibration



Aforementioned paper:

- Rigorous calibration of the model
- Unbias the calibration
- Polling the network for communications is not free!

Roadmap

A lot of flops... a lot of watts?
The Top500 and the Green500
Current machines: example with Frontier

Modeling power consumption
Distributed applications

Sharing GPUs
What is the energy consumption?
How to use the lessons learned?

Conclusion

Sharing GPUs

Work done with Pierre Jacquet

- Post doc at ÉTS
- MITACS intern at OVHcloud Montreal

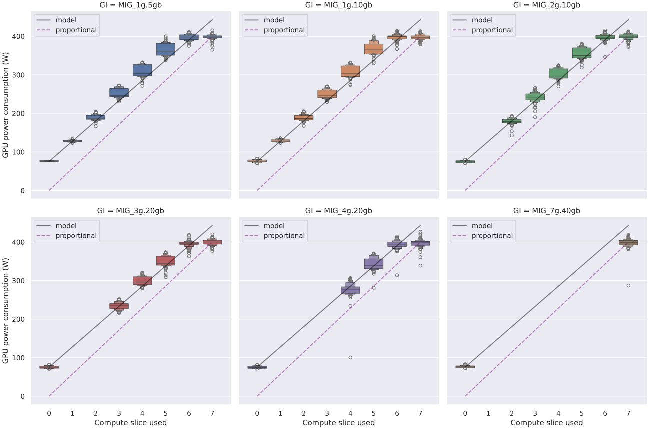
Goal: understand how sharing GPUs impact their energy consumption.

Sharing GPUs

How do we share GPUs?

- Spatially: MIG instances
- Temporally: use the scheduler
- Both!

Is the energy consumption **linear with the number of MIG instances?**



Sharing GPUs

How do we share GPUs?

- Spatially: MIG instances
- Temporally: use the scheduler
- Both!

Is the energy consumption **linear with the number of MIG instances?**
Is the energy consumption **constant when a GPU is shared?**

- Many hardware components on a GPU
- Overlap operations of different programs

→ 2 programs sharing a GPU run faster than twice the time

Consequences

Question: how do we **schedule** applications on GPUs to **minimize their consumption**?

1. Design algorithms
2. Evaluate them

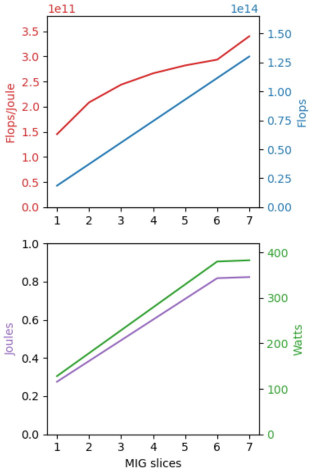
Yes, but HOW?

Consequences

Question: how do we **schedule** applications on GPUs to **minimize their consumption**?

1. Design algorithms
2. Evaluate them

Yes, but HOW?



Simulation in SimGrid[a]

- Energy consumption plugin
- Problem: the GPU's energy consumption is **not linear**
- Here: constant workload *per MIG slice*

[a] https://simgrid.org/

Roadmap

A lot of flops... a lot of watts?
- The Top500 and the Green500
- Current machines: example with Frontier

Modeling power consumption
- Distributed applications

Sharing GPUs
- What is the energy consumption?
- How to use the lessons learned?

Conclusion

What do we know?

HPC draws **too much energy**

- No free beer anymore
- We are beyond the boundaries set in the past
  - Goal: 20MW at exascale
  - Current Top 10 machines: 4 machines beyond that limit
  - 29.6 GW, 24.6 GW, 38.7 GW, 29.9 GW

We are **characterizing energy consumption**

- What is expensive (energy-wise)?
- How can we improve the flops/joule ratio?

What do we do now?

Optimize resource usage

- Time is not the only metrics
- Sometimes time and energy give the same result
  - Sometimes not
  - Sharing: half of the resources, less than twice the time?
  - Flops per joule?
- **More sharing**
  - Better overlap
  - Idling is expensive

Scheduling algorithms need to take it into account

- We need to understand and model energy consumption
  - Simulation tools to assess them
- Idling is expensive

Sharing is caring!