Experience on clock rate adjustment for energy-efficient GPU-accelerated real-world codes

EESP Workshop 2025@ISC
Hamburg 13/6/2025

Giorgio Amati, g.amati@cineca.it
Matteo Turisini, m.turisini@cineca.it
Andrea Monterubbiano, a.monterubbiano@cineca.it
Mattia Paladino, mattia.paladino@e4company.com
Elisabetta Boella, elisabetta.boella@e4company.com
Daniele Gregori, daniele.gregori@e4company.com
Danilo Croce, croce@info.uniroma2.it

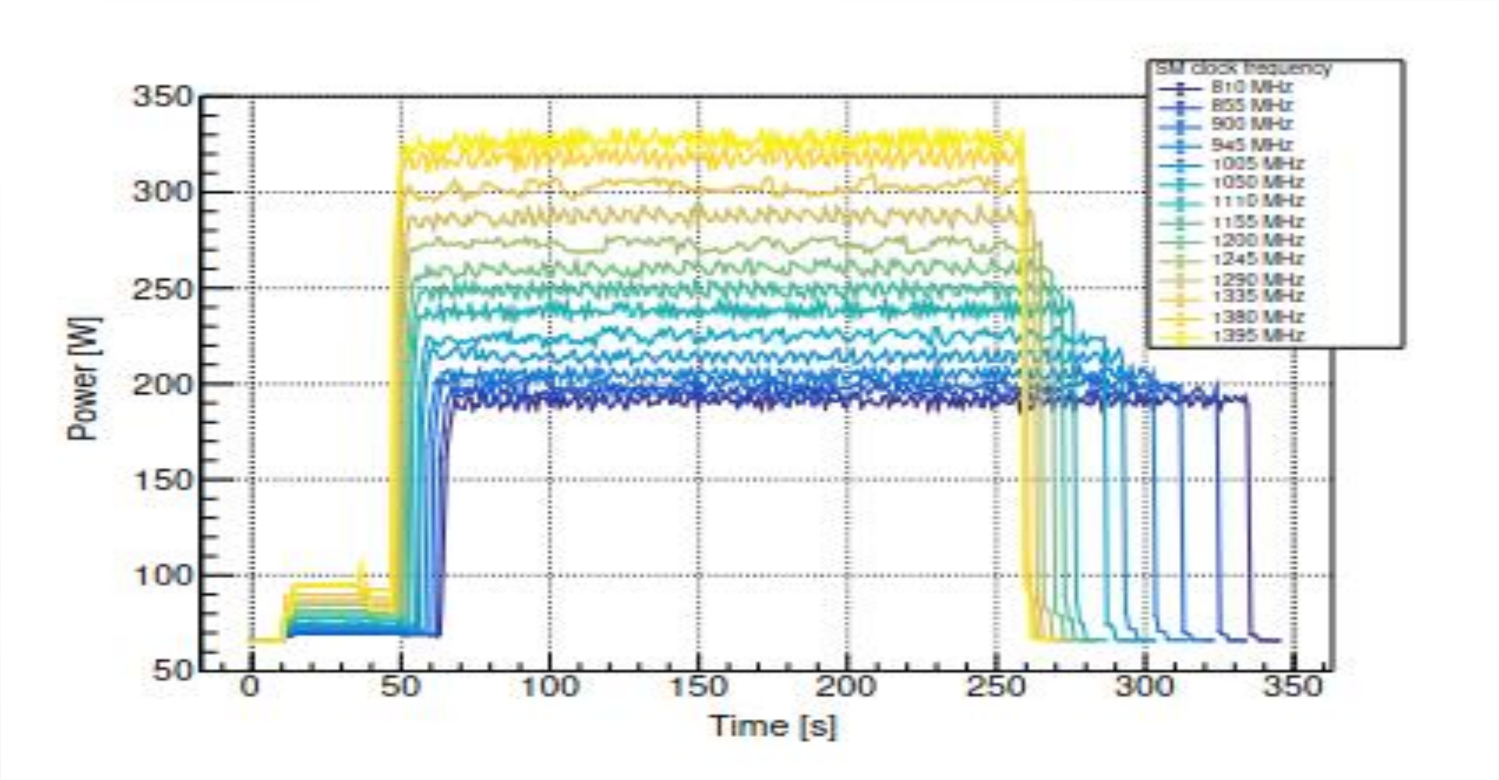
LEONARDO CINECA

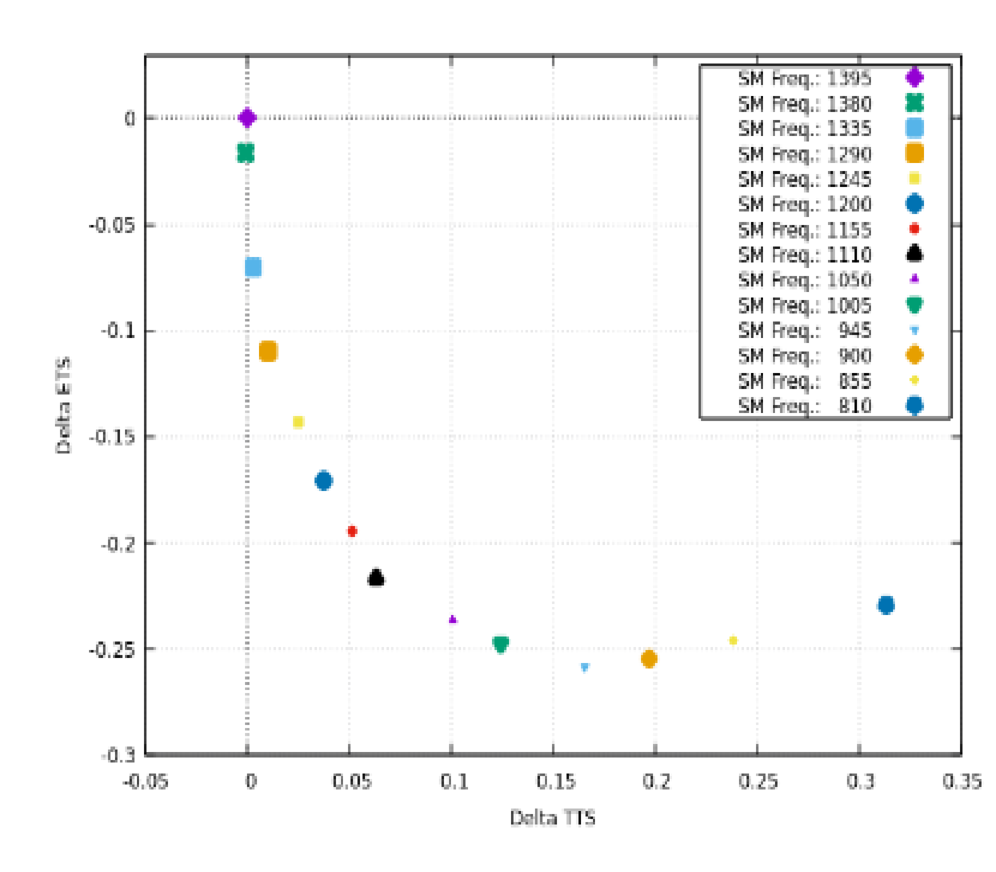
What we want to do?

- Understand better the relation between energy & performance, at user level:
  - **TTS:** time to solution
  - **ETS:** energy to solution
- Tier-0 systems are very energy demanding
  - More than 10 MW
  - Energy Crisis
  - Opening Keynote talk
- Today target is ML/AI stuff
  - Thousands of FPU, with different precision (e.g.: FP16, BF16, ...)
- Can we save some energy?
  - Rule of thumb: 1 MW less in a year is more than 1 million euro saved!!!

Which is the starting point?

From a previous work on LBM optimization, we found that SM frequency reduction produce a gain (using A100@64GB).





- **Energy efficiency: a Lattice Boltzmann study** https://doi.org/10.1145/3659997.366003

Is this behaviour "general"?

LEONARDO CINECA

How?

- Working in user-space
- Based on nvidia-smi
- An energy based-profile
  - Time series of power, frequency, temperature
- Only NVIDIA GPU (for now...)
  - NVIDIA A100@80GB /NVIDIA GH200

```
1, 2025/05/28 20:40:22.666, 256.97, 1395, 1593, 54, 68, 97, 82, P0
2, 2025/05/28 20:40:22.670, 261.30, 1395, 1593, 53, 68, 97, 82, P0
3, 2025/05/28 20:40:22.673, 352.53, 1395, 1593, 54, 67, 97, 82, P0
0, 2025/05/28 20:40:23.677, 263.97, 1395, 1593, 53, 67, 97, 82, P0
1, 2025/05/28 20:40:23.681, 254.83, 1395, 1593, 54, 68, 97, 82, P0
2, 2025/05/28 20:40:23.685, 263.43, 1395, 1593, 53, 66, 97, 82, P0
3, 2025/05/28 20:40:23.689, 350.73, 1395, 1593, 54, 69, 97, 82, P0
0, 2025/05/28 20:40:24.693, 262.11, 1395, 1593, 53, 66, 97, 82, P0
1, 2025/05/28 20:40:24.697, 253.64, 1395, 1593, 54, 69, 97, 82, P0
2, 2025/05/28 20:40:24.700, 262.78, 1395, 1593, 53, 67, 97, 82, P0
3, 2025/05/28 20:40:24.704, 351.57, 1395, 1593, 55, 69, 97, 82, P0
0, 2025/05/28 20:40:25.708, 270.65, 1395, 1593, 54, 67, 97, 82, P0
```

Which code?

Four different codes were selected: they are known to be optimized and can run on O(1000) or more GPUs

- **BGK3D:** CFD Lattice Boltzmann Method based code. Fortran, MPI+OpenACC
- **Pipe:** CFD Finite Fifference code, Fortran, MPI+CudaFortran
- **QISG:** Quantum Spin-Glass Code, C+CUDA
- **Bert:** transformer-based models for natural language processing. It is written in Python and relies on TensorFlow.

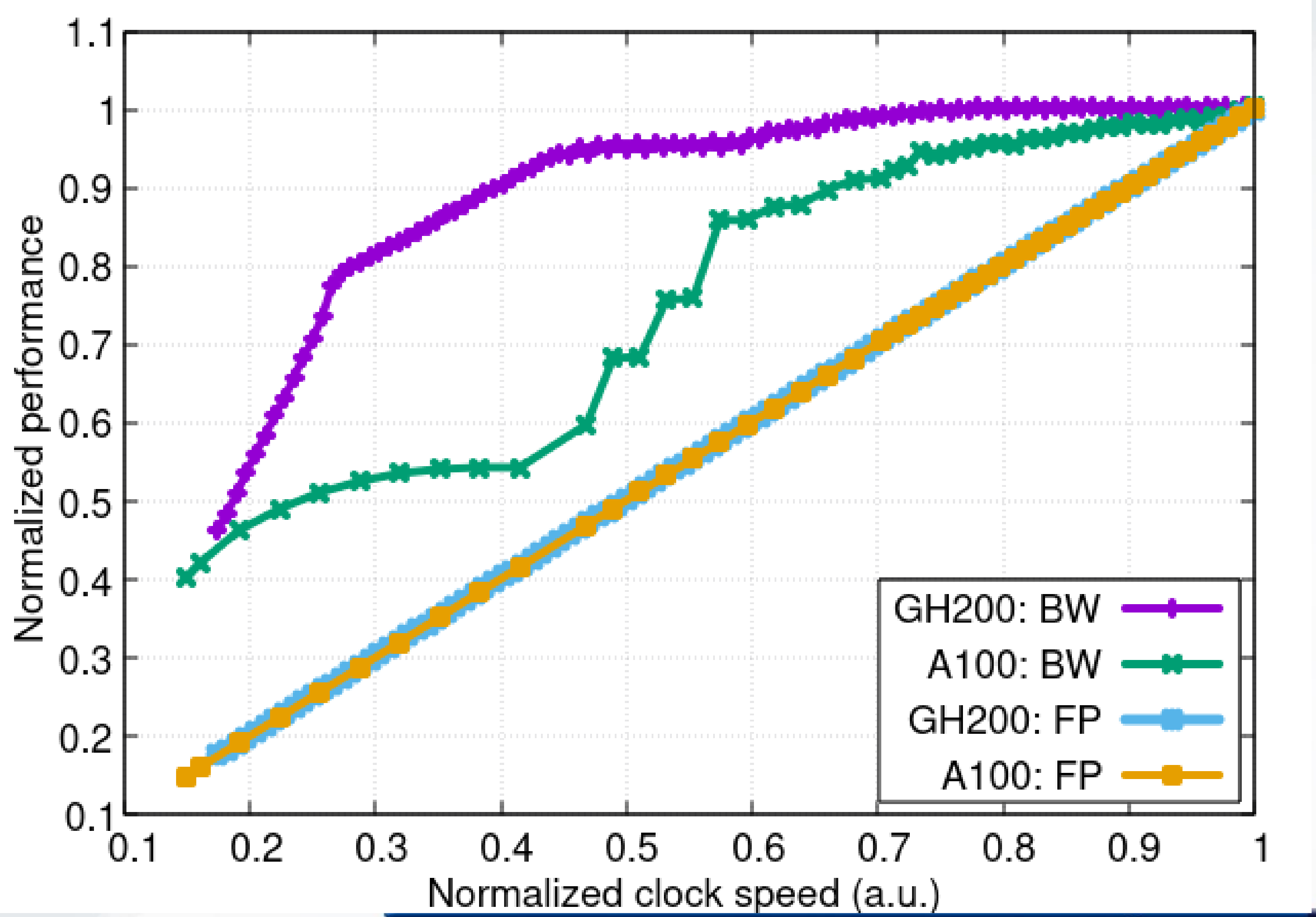
- G. Falcucci et al. “Extreme flow simulations reveal skeletal adaptations of deep-sea sponges”. In: Nature 595.7868 (July 2021)
- M. Bernaschi et al. “The Quantum Transition of the Two-Dimensional Ising Spin Glass”. In: Nature 631 (2024)
- S. Pirozzoli et al. “One-point statistics for turbulent pipe flow up to Re_t=6000”. In: J. of Fluid Mechanics 926 (2021)

What have we found?/1

Which impact of streaming multiprocessor (SM) frequency

BW=Bandwidth

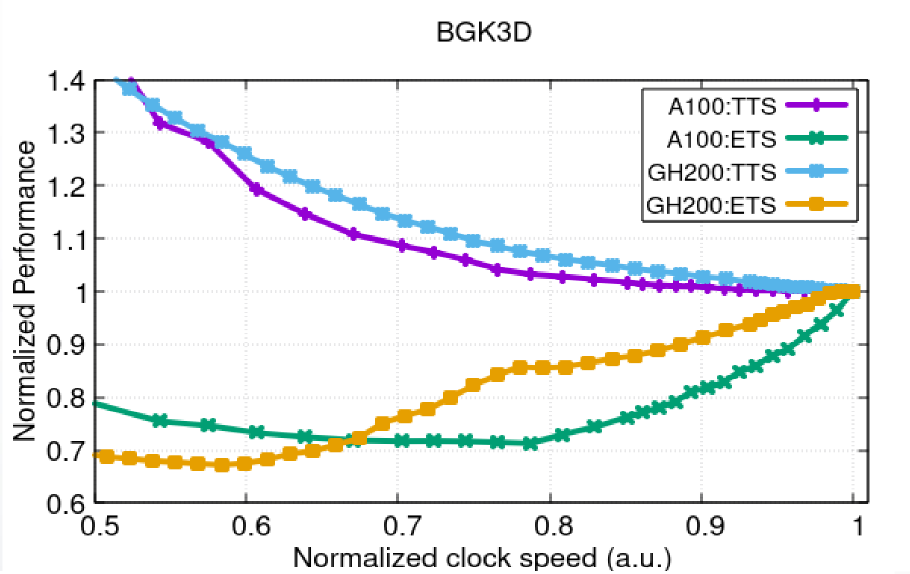
FP=Floating Point

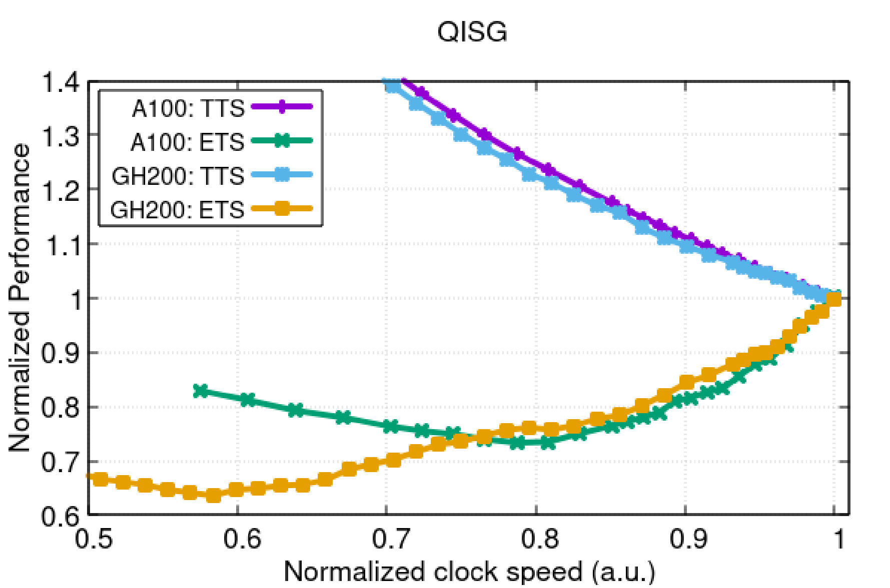


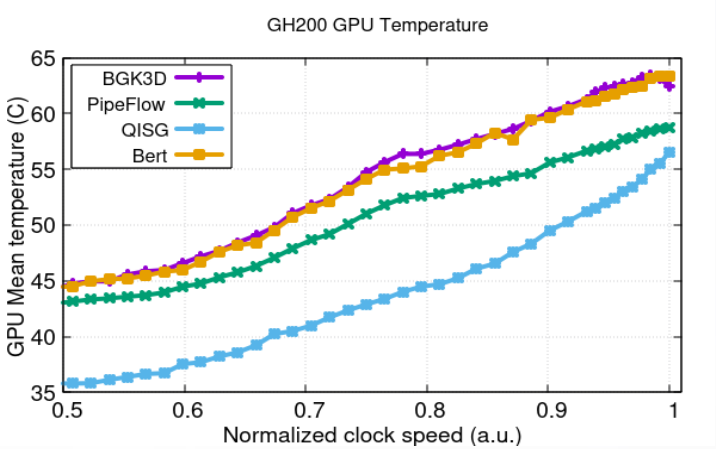
- OpenCL_benchmark

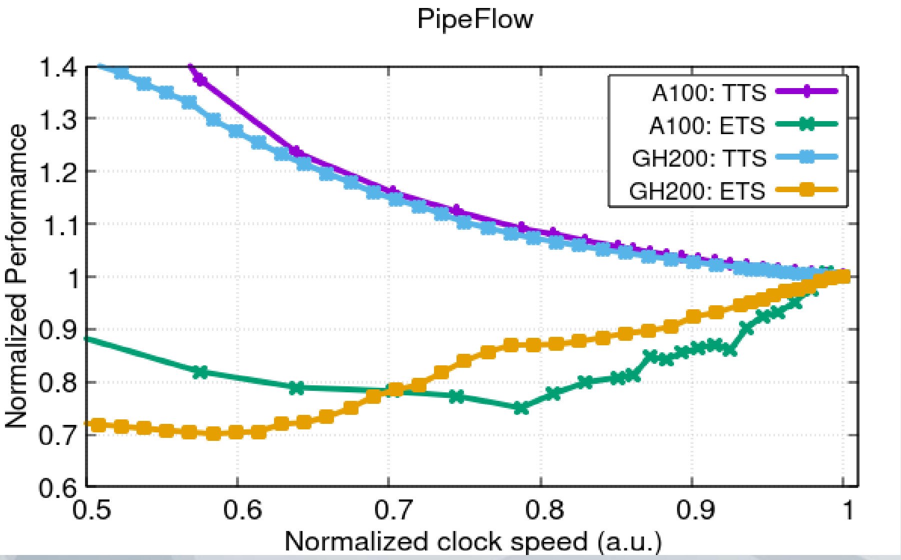
What have we found?/2

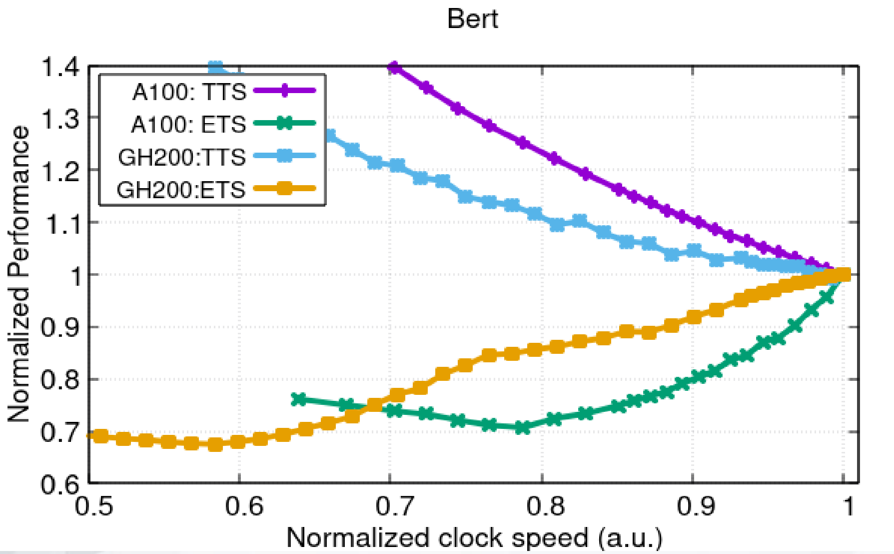
Changing SM frequency has impact ETS, TTS and temperature







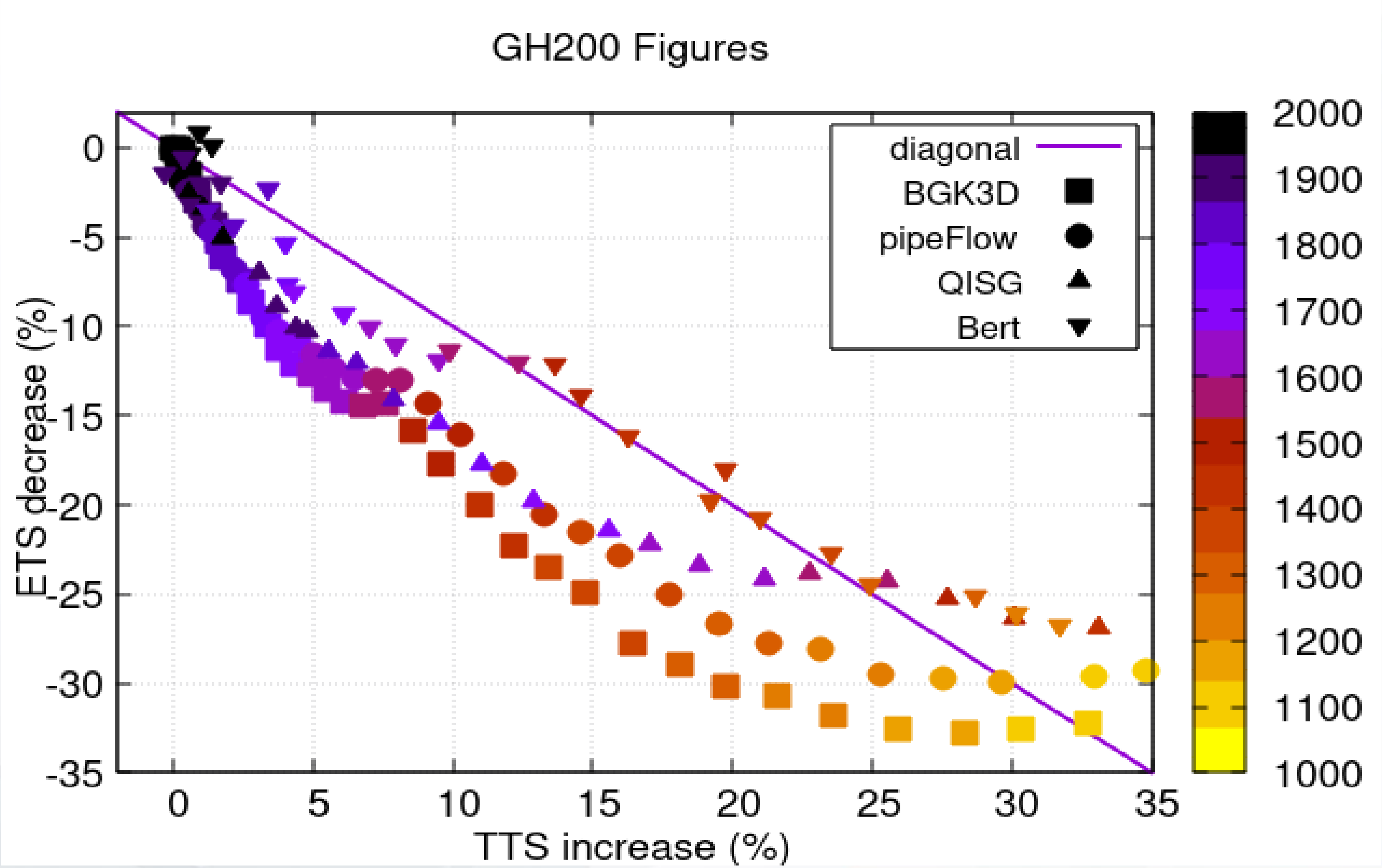




LEONARDO CINECA

What have we found?/3

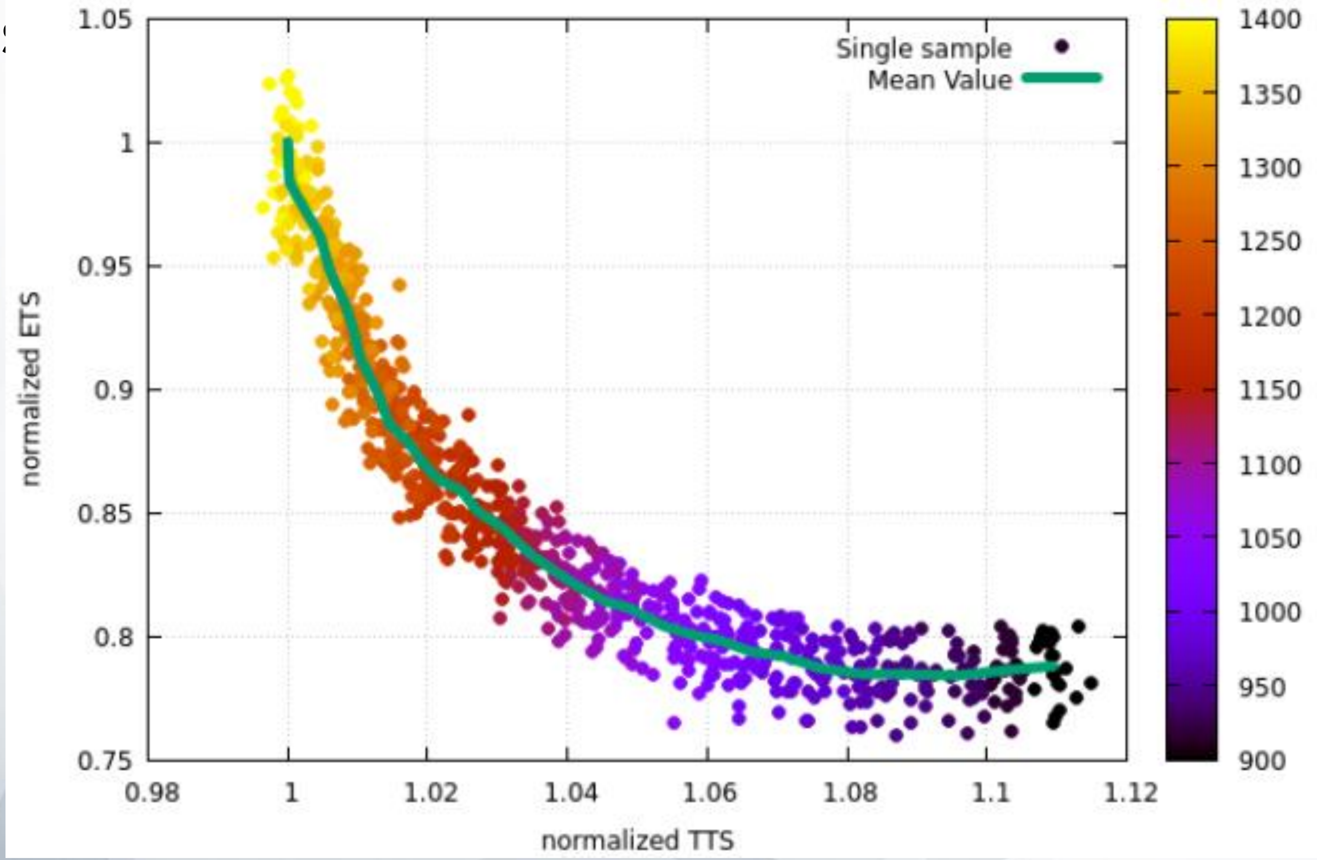
A similar behaviour for all codes (GH200)

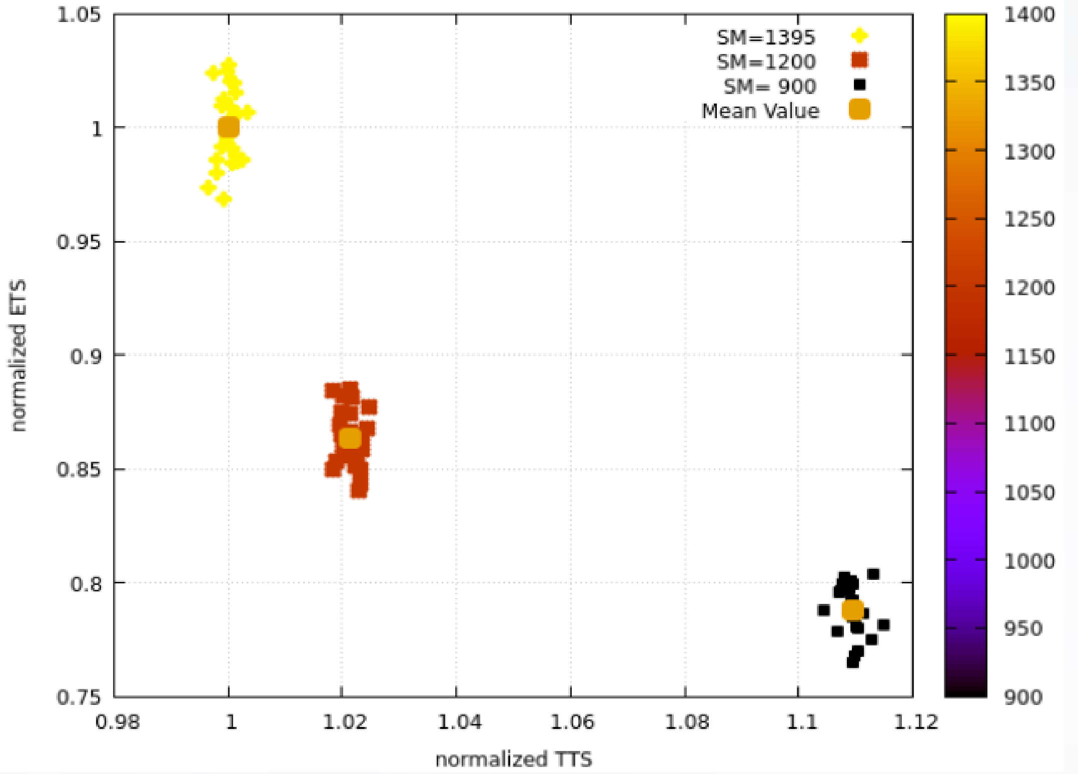


What have we found?/4 (New results)

Leonardo node behaviour (4xA100@64 GB)

- REFMUL3: Finite Difference time domain code, C, MPI+OpenMP target offload
- Changing SM frequency via Slurm
- 24 replica using different nodes for each SM frequency

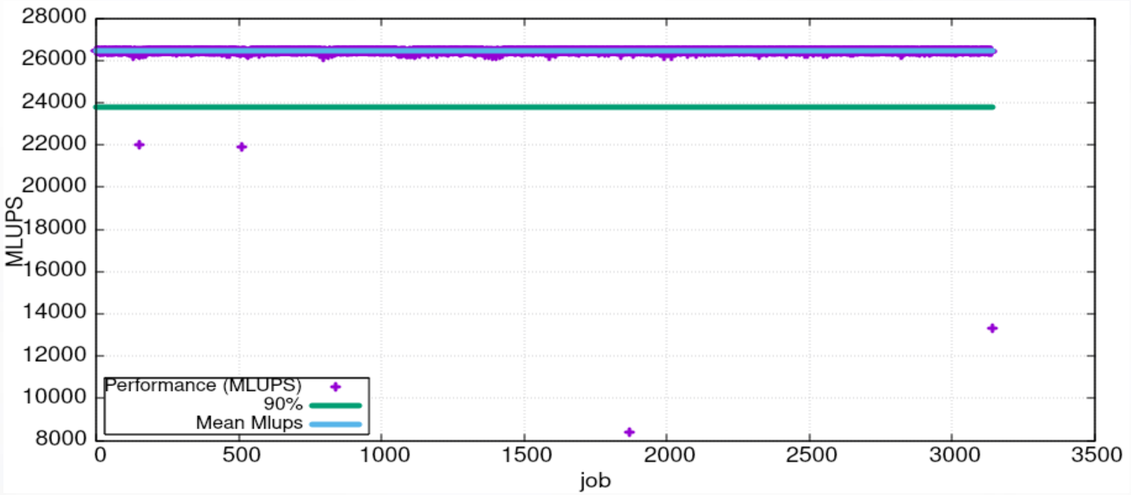


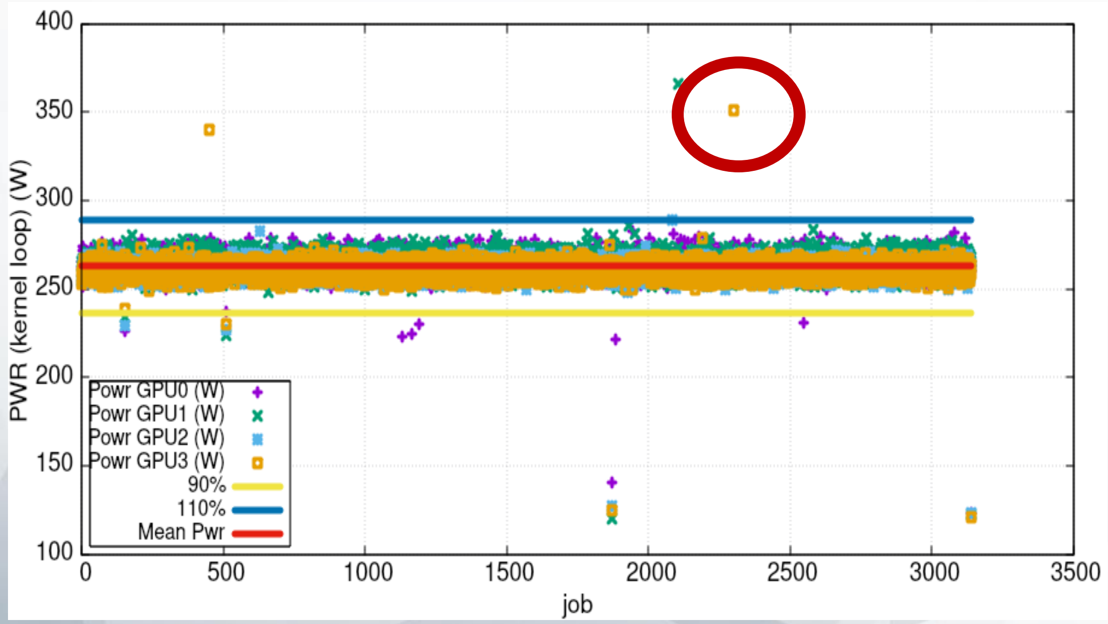


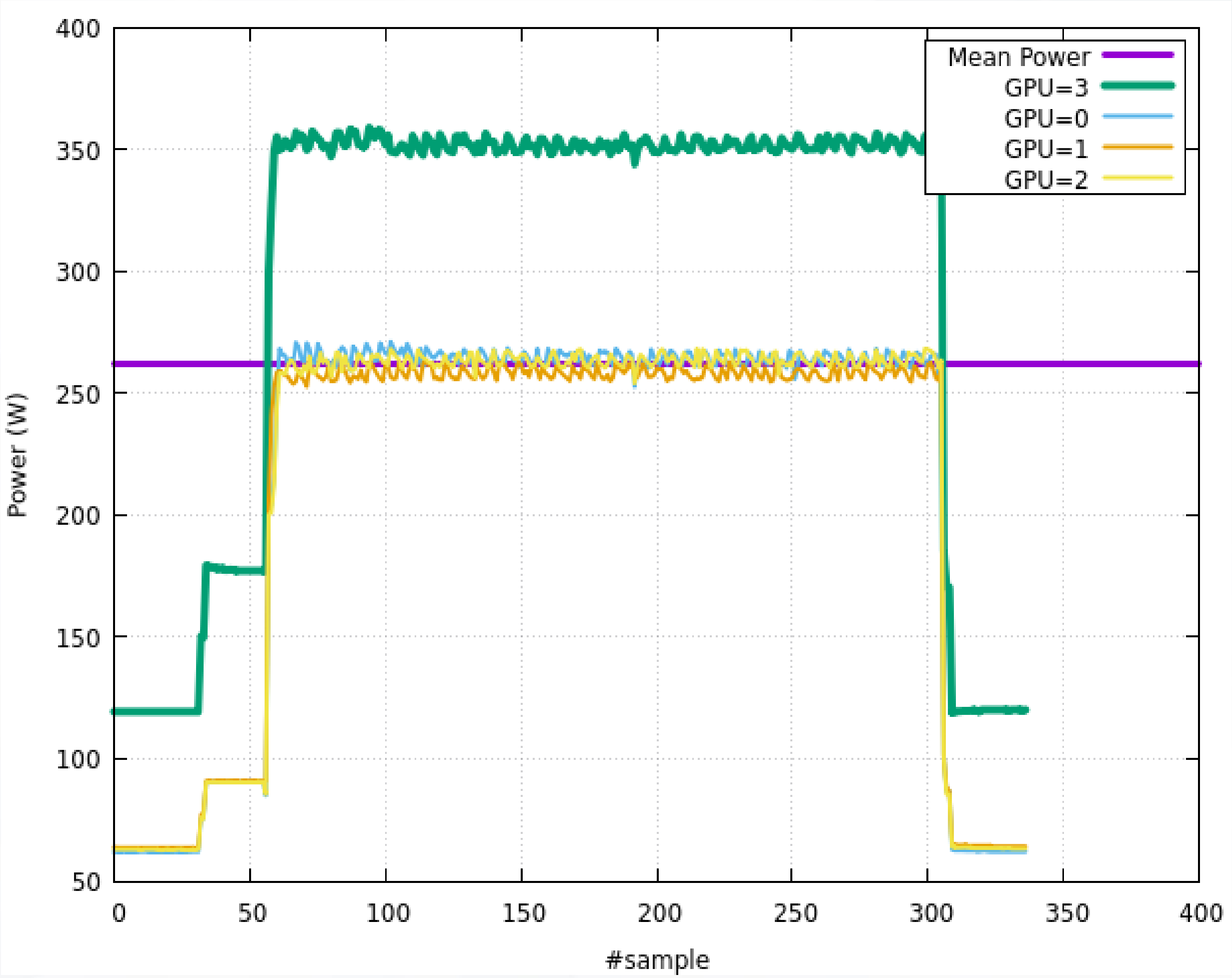
LEONARDO CINECA

Which development?/1 (New results)

Node monitoring: looking for anomalous behaviour (4GPU)



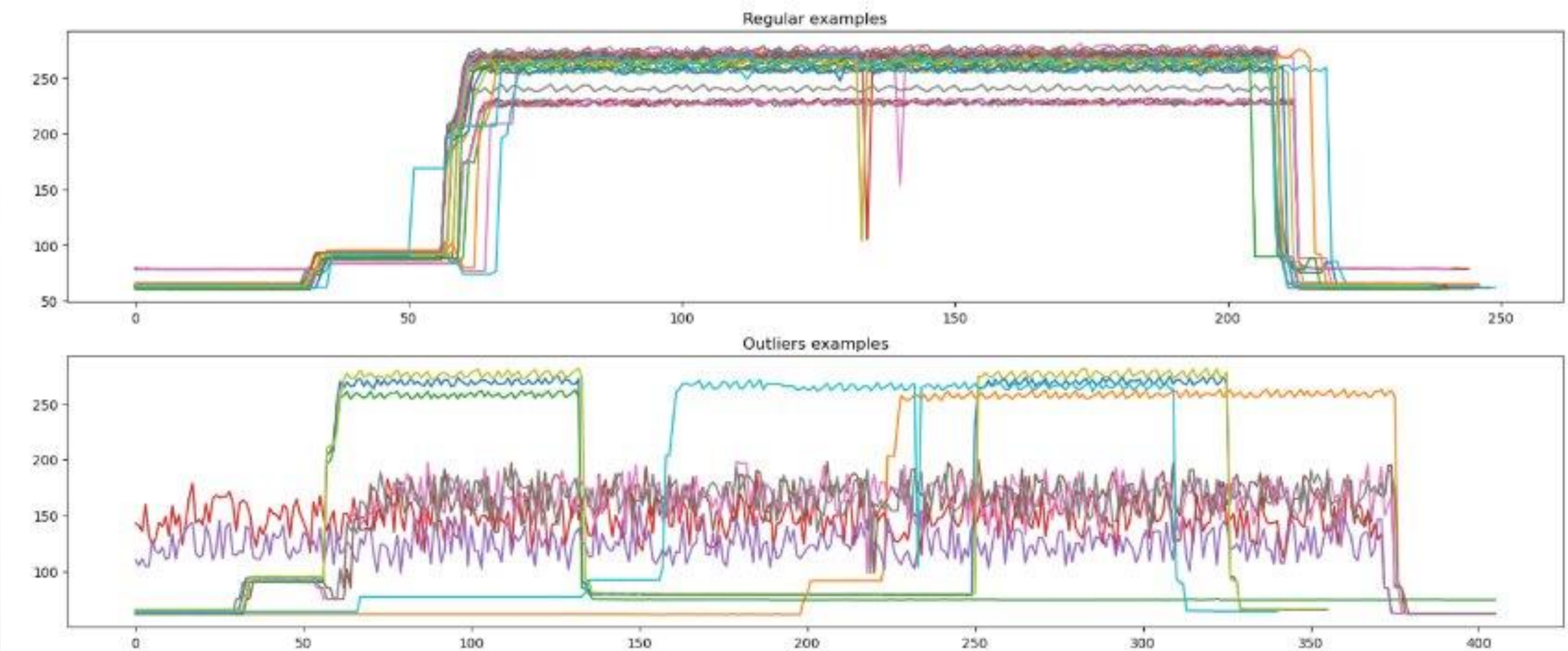




LEONARDO CINECA

Which development?/2 (New results)

Node monitoring: supervised model



What did we learn?

- ETS can be reduced with an acceptable TTS increase
- The ideal SM frequency is code dependent
- Energy (e.g. time series) is also a useful observable
  - Node monitoring
- Frequency reduction produces GPU temperature reduction
  - Impact on system reliability?
- Even multi-GPU present a similar behaviour
  - Next step: multi-node
  - Next step: CPU/RAM contribution

How to face the problem? (personal view)

Energy must be a part of the "accounting" system

- Set Default SM frequency "quite low"
- Allow user to change SM frequency
  - If you increase frequency you pay "more" GPU hour
  - If you decrease frequency you pay "less" GPU hour
- Priority as a function of frequency
  - High frequency --> lower priority
- Force run at high frequency during the night

Acknowledgments

- Giacomo Falcucci (Tor Vergata Univ.)
- Sergio Pirozzoli (La Sapienza Univ.)
- Tiago Ribeiro (ACH-MPG)
- Filipe da Silva (IST-IPFN)
- Massimo Bernaschi (CNR/IAC)
- Daniele Di Bari (CINECA)
- Anna Nikishova (CINECA)
- Matteo Angelinelli (CINECA)
- Donatella Sforzini (CINECA)

CINECA

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

E4 COMPUTER ENGINEERING

ChEESE

Thanks!

CONNECTING THE DOTS

ISC High Performance