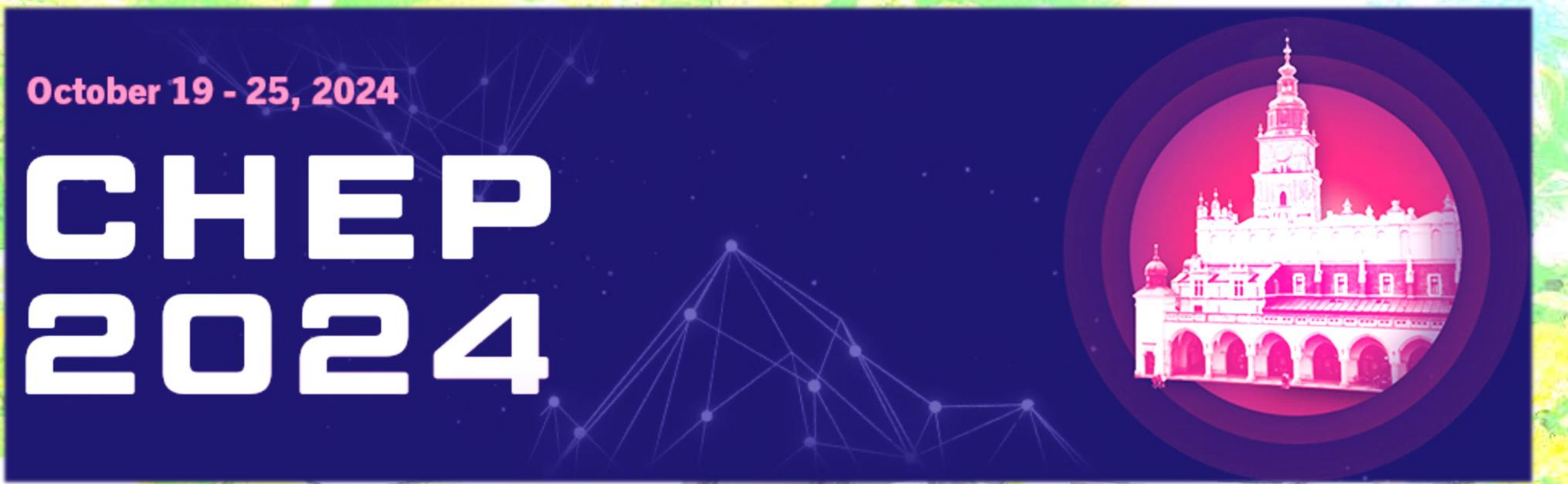
Heterogeneous Computing & Power Efficiency in HEP



Dr. Emanuele Simili

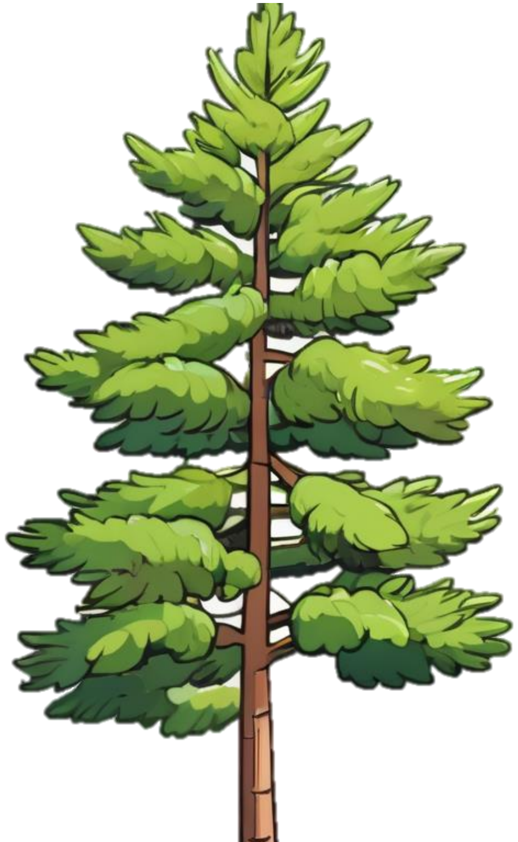
CHEP2024 @ Krakow - 23 October 2024

Outline

- Benchmarking and Power Measurement
  - motivations & methodology
  - new **IPMI** validation campaign against external **PDU**
  - new **Figure of Merit** and updated **HS23/Watt** and **Frequency Scan** results
  - comparison with other benchmarks: (**ROOT**), **DB12**, **Geant4**
  - single vs. dual socket server performance

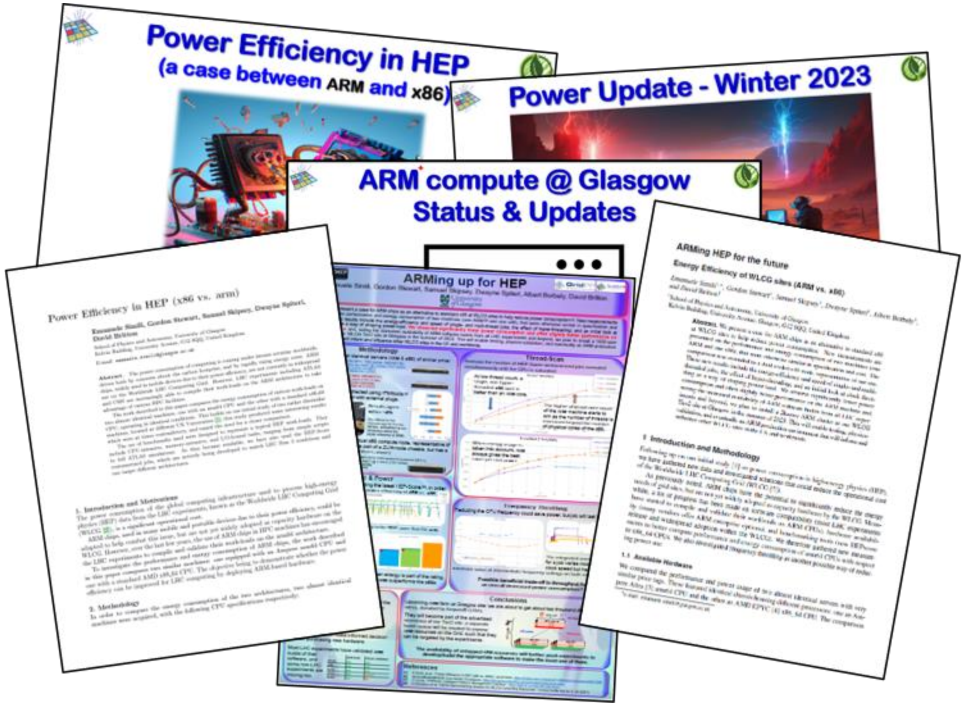
- Heterogeneous Tier2 Cluster @ **ScotGrid Glasgow**
  - configuration and dual queue management
  - physics validation results

- Ongoing efforts & Outlook
  - testing new machines
  - developing an analysis suite within **HEP-Score**
  - first look at emerging architectures (**RISC-V**) … next!



Motivations & Methodology

In 2021 we started investigating alternative architectures for Grid computing, starting with **ARM** chips …


Power Efficiency in HEP
(a case between ARM and x86)
Power Update - Winter 2023
ARM compute @ Glasgow
Status & Updates
Power Efficiency in HEP (x86 vs. arm)
ARMing up for HEP
ARMing HEP for the future

Lot has happened since then:

- most LHC experiments ported their software to ARM,
- physics validations had been performed,
- heterogeneous computing cluster set-up (x86 + ARM),
- HEP-Score collaboration and improved methodology,
- dissemination of results, …

Methodology:

As benchmark, we rely on the HEP-Score & the HEP-Benchmarking Suite:

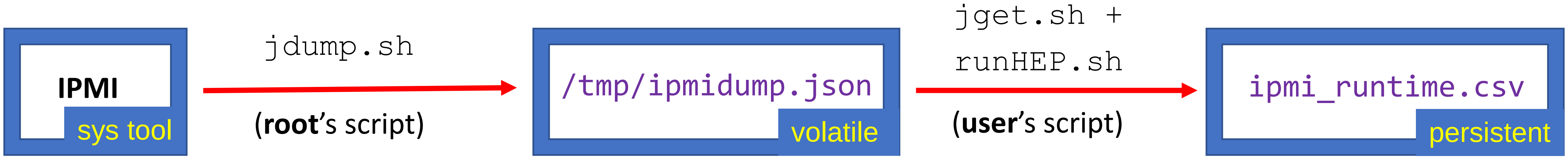
HEP-Suite: https://gitlab.cern.ch/hep-benchmarks/hep-benchmark-suite
HEP-Score: https://gitlab.cern.ch/hep-benchmarks/hep-score

While the benchmark executes, a script collects and exports CPU, RAM, Frequency and Power usage (via **IPMI tools**) into a CSV file.

Results are then processed to generate plots, integrate the energy usage, and do some statistical calculations. Cumulative results are then compared among different machines.

Data Processing

The following diagram outlines the various step from data collection to processing and visualization:



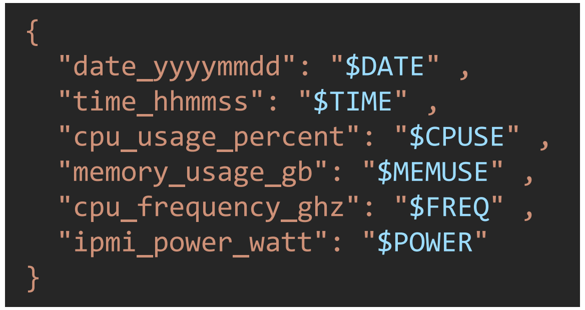
IPMI
sys tool

jdump.sh
(**root**’s script)

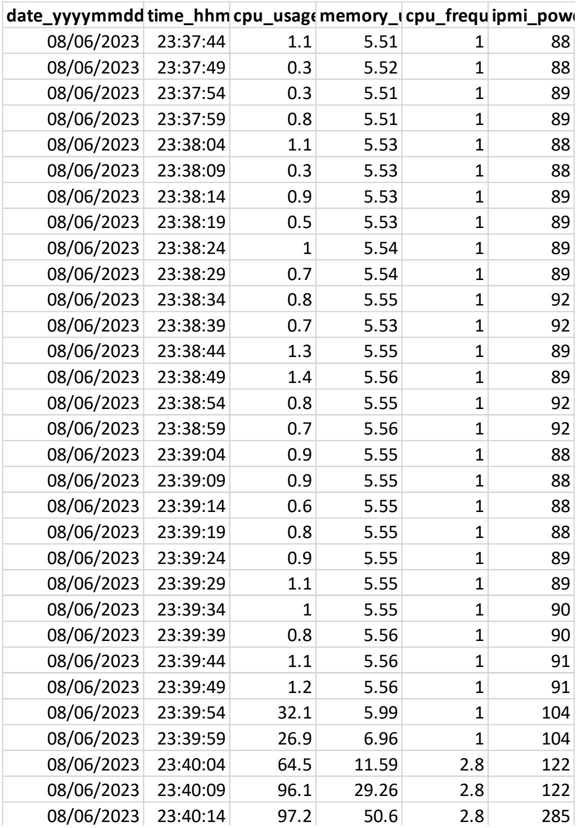
/tmp/ipmidump.json
volatile

jget.sh +
runHEP.sh
(**user**’s script)

ipmi_runtime.csv
persistent

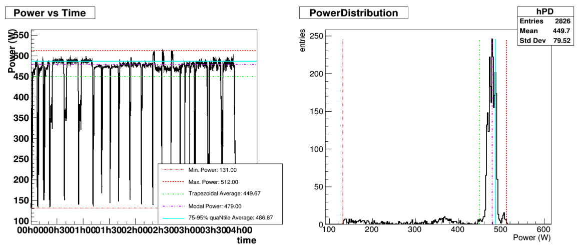


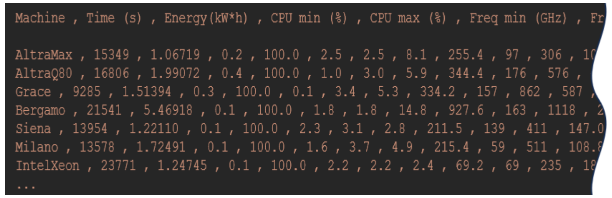
```
{
  "date_yyyymmdd": "$DATE" ,
  "time_hhmmss": "$TIME" ,
  "cpu_usage_percent": "$CPUSE" ,
  "memory_usage_gb": "$MEMUSE" ,
  "cpu_frequency_ghz": "$FREQ" ,
  "ipmi_power_watt": "$POWER"
}
```



date_yyyymmdd	time_hhm	cpu_usage	memory_u	cpu_frequ	ipmi_pow
08/06/2023	23:37:44	1.1	5.51	1	88
08/06/2023	23:37:49	0.3	5.52	1	88
08/06/2023	23:37:54	0.3	5.51	1	89
08/06/2023	23:37:59	0.8	5.51	1	89
08/06/2023	23:38:04	1.1	5.53	1	88
08/06/2023	23:38:09	0.3	5.53	1	88
08/06/2023	23:38:14	0.9	5.53	1	89
08/06/2023	23:38:19	0.5	5.53	1	89
08/06/2023	23:38:24	1	5.54	1	89
08/06/2023	23:38:29	0.7	5.54	1	89
08/06/2023	23:38:34	0.8	5.55	1	92
08/06/2023	23:38:39	0.7	5.53	1	92
08/06/2023	23:38:44	1.3	5.55	1	89
08/06/2023	23:38:49	1.4	5.56	1	89
08/06/2023	23:38:54	0.8	5.55	1	92
08/06/2023	23:38:59	0.7	5.56	1	92
08/06/2023	23:39:04	0.9	5.55	1	88
08/06/2023	23:39:09	0.9	5.55	1	88
08/06/2023	23:39:14	0.6	5.55	1	88
08/06/2023	23:39:19	0.8	5.55	1	88
08/06/2023	23:39:24	0.9	5.55	1	89
08/06/2023	23:39:29	1.1	5.55	1	89
08/06/2023	23:39:34	1	5.55	1	90
08/06/2023	23:39:39	0.8	5.56	1	90
08/06/2023	23:39:44	1.1	5.56	1	91
08/06/2023	23:39:49	1.2	5.56	1	91
08/06/2023	23:39:54	32.1	5.99	1	104
08/06/2023	23:39:59	26.9	6.96	1	104
08/06/2023	23:40:04	64.5	11.59	2.8	122
08/06/2023	23:40:09	96.1	29.26	2.8	122
08/06/2023	23:40:14	97.2	50.6	2.8	285

ipmi2root.C

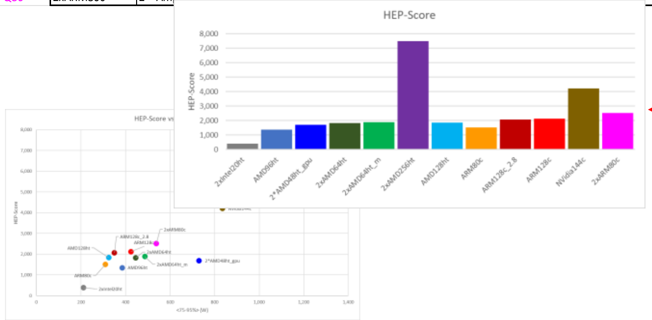


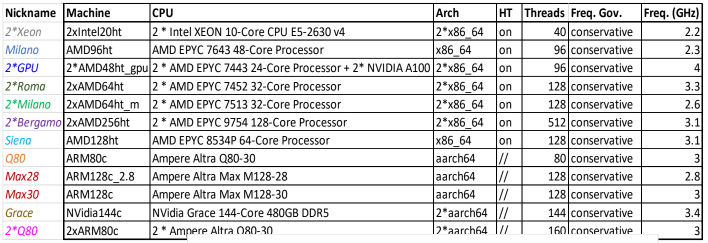


```
Machine , Time (s) , Energy(kW*h) , CPU min (%) , CPU max (%) , Freq min (GHz) , Fr

AltraMax , 15349 , 1.06719 , 0.2 , 100.0 , 2.5 , 2.5 , 8.1 , 255.4 , 97 , 306 , 10
AltraQ80 , 16806 , 1.99072 , 0.4 , 100.0 , 1.0 , 3.0 , 5.9 , 344.4 , 176 , 576 ,
Grace , 9285 , 1.51394 , 0.3 , 100.0 , 0.1 , 3.4 , 5.3 , 334.2 , 157 , 862 , 587 ,
Bergamo , 21541 , 5.46918 , 0.1 , 100.0 , 1.8 , 1.8 , 14.8 , 927.6 , 163 , 1118 , 2
Siena , 13954 , 1.22110 , 0.1 , 100.0 , 2.3 , 3.1 , 2.8 , 211.5 , 139 , 411 , 147.0
Milano , 13578 , 1.72491 , 0.1 , 100.0 , 1.6 , 3.7 , 4.9 , 215.4 , 59 , 511 , 108.8
IntelXeon , 23771 , 1.24745 , 0.1 , 100.0 , 2.2 , 2.2 , 2.4 , 69.2 , 69 , 235 , 18
...
```

Excel

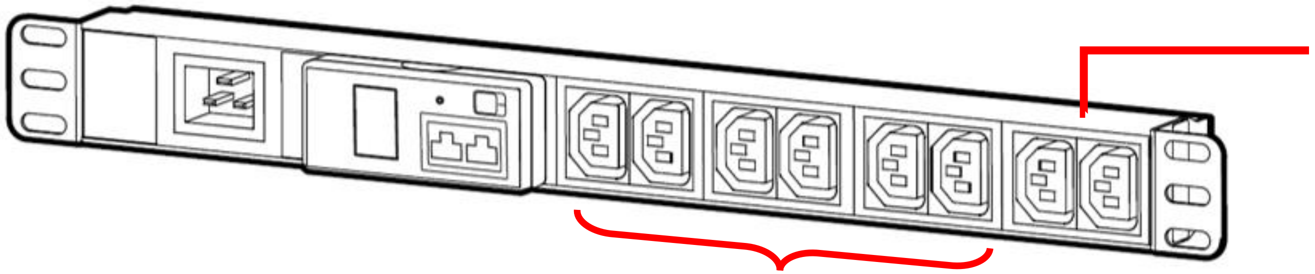




Nickname	Machine	CPU	Arch	HT	Threads	Freq. Gov.	Freq. (GHz)
*2*Xeon*	2xIntel20ht	2 * Intel XEON 10-Core CPU E5-2630 v4	2*x86_64	on	40	conservative	2.2
Milano	AMD96ht	AMD EPYC 7643 48-Core Processor	x86_64	on	96	conservative	2.3
*2*GPU*	2*AMD48ht_gpu	2 * AMD EPYC 7443 24-Core Processor + 2* NVIDIA A100	2*x86_64	on	96	conservative	4
*2*Roma*	2xAMD64ht	2 * AMD EPYC 7452 32-Core Processor	2*x86_64	on	128	conservative	3.3
*2*Milano*	2xAMD64ht_m	2 * AMD EPYC 7513 32-Core Processor	2*x86_64	on	128	conservative	2.6
*2*Bergamo*	2xAMD256ht	2 * AMD EPYC 9754 128-Core Processor	2*x86_64	on	512	conservative	3.1
Siena	AMD128ht	AMD EPYC 8534P 64-Core Processor	x86_64	on	128	conservative	3.1
Q80	ARM80c	Ampere Altra Q80-30	aarch64	//	80	conservative	3
Max28	ARM128c_2.8	Ampere Altra Max M128-28	aarch64	//	128	conservative	2.8
Max30	ARM128c	Ampere Altra Max M128-30	aarch64	//	128	conservative	3
Grace	NVidia144c	NVidia Grace 144-Core 480GB DDR5	2*aarch64	//	144	conservative	3.4
*2*Q80*	2xARM80c	2 * Ampere Altra Q80-30	2*aarch64	//	160	conservative	3

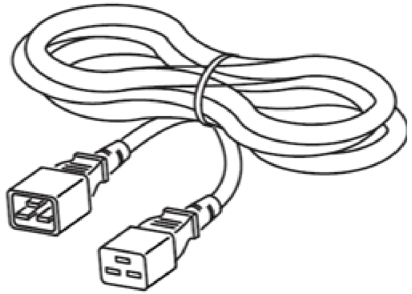
Validation Strategy

We have performed a few tests to validate **IPMI** power readings against a metered **PDU** .
The **PDU** we acquired provides a single reading for all power sockets, so we connected and tested a single machine at each time:



tested machine

empty

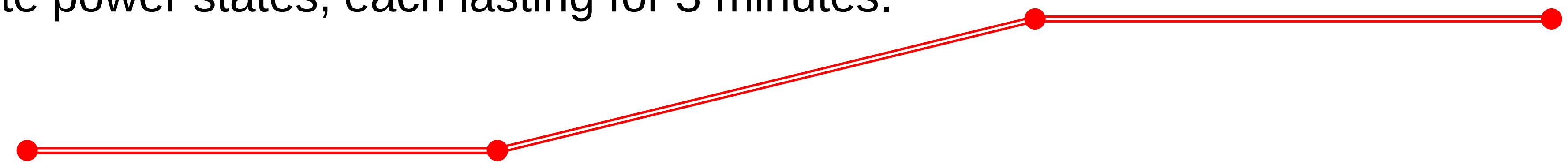


We used two exporter scripts to collected power readings from:

- **IPMI** (via **ipmitool** “Instantaneous Power”) ;
- **PDU** (via **ModBus** “Active power”),

with sampling interval **1 sec** for both.

We ran a three-stage test: idle (**sleep**), busy (**stress**), and a rising job (**stress** loop with increasing threads) to probe intermediate power states, each lasting for 3 minutes.



sleep (3 min.)

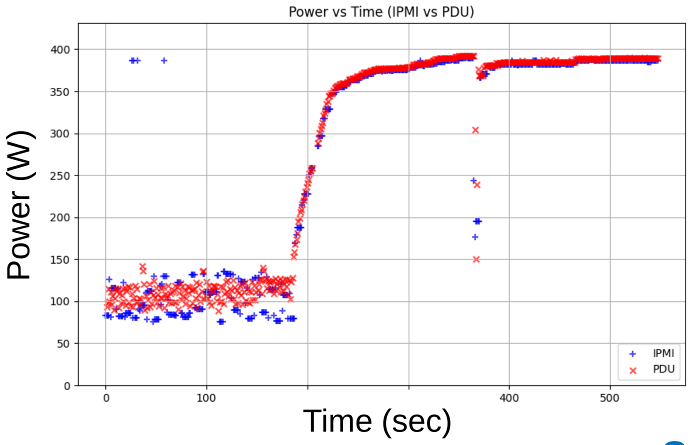
raise (3 min.)

stress (3 min.)

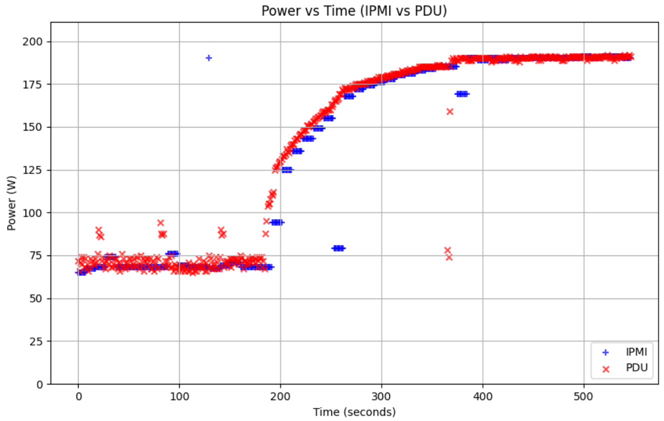
Validation Results

We could test only a limited number of machines (i.e., single chassis with **C13/C14 plug**).

AMD Epyc Milano (GIGABYTE)



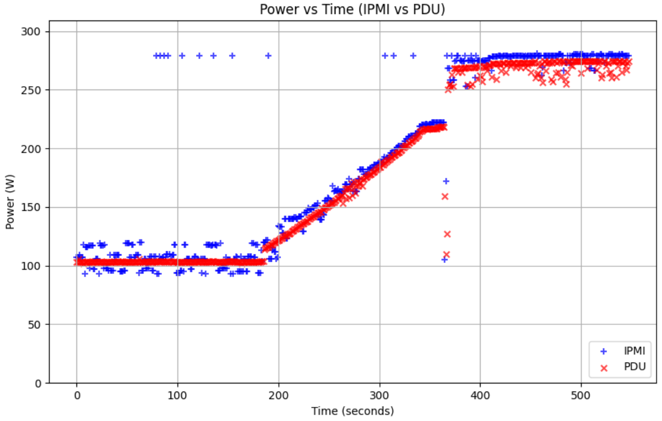
Intel Xeon (HP)



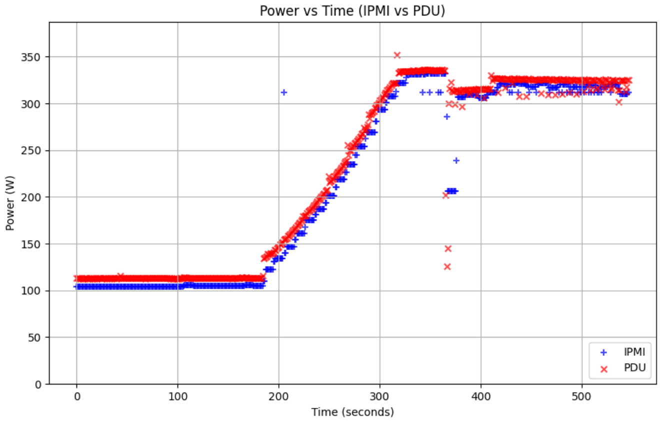
x86

ARM

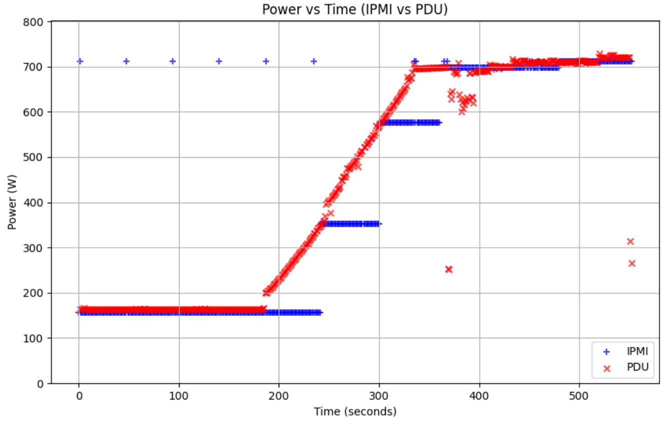
Altra Q80-30 (GIGABYTE)



AltraMax (SuperMicro)



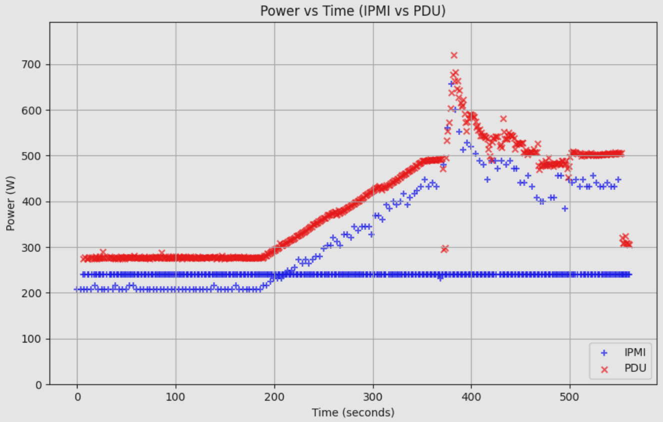
Nvidia Grace (SuperMicro)



+ IPMI

× PDU

Server X



Consistency varies widely across manufacturer, with extreme cases (**Server X**) where various issues with the IPMI implementation on that platform prevented us from completing a definitive assessment.

Validation Conclusions

We'd expect the power difference **Δ<power>$_{\textbf{PDU-IPMI}}$** to have small oscillation around the PDU baseline (~16 Watts) … but it is not really the case!

Known issues:

- there seem to be a lot more oscillations in idle than at full load, probably due to lower efficiency of the **PSU**s at lower output levels, and variation in background load have a relatively larger impact;

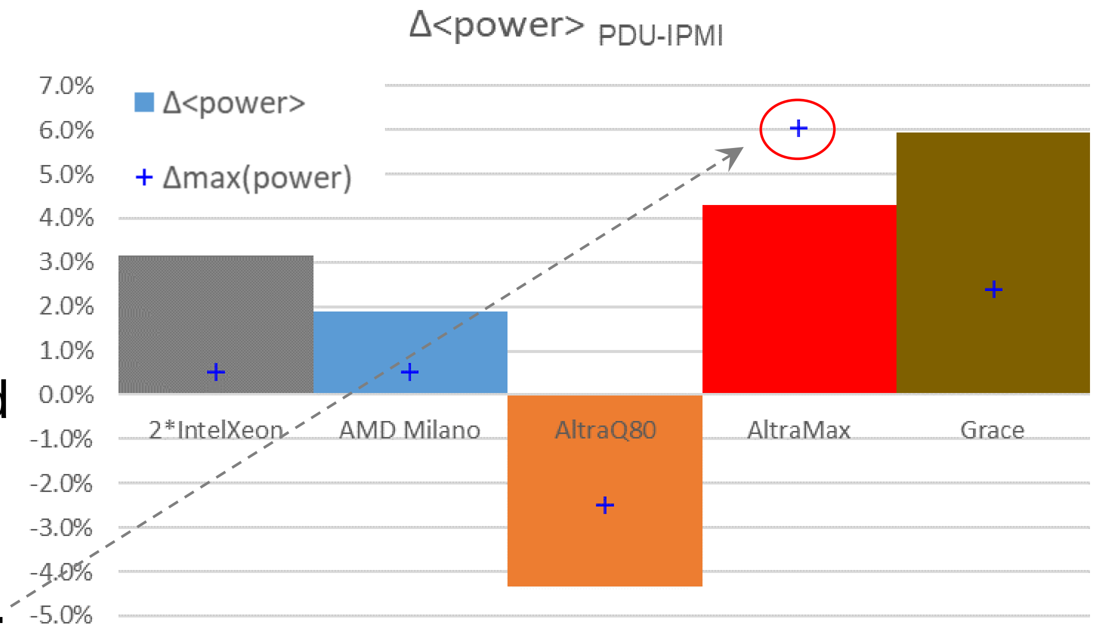
- some machine (e.g., Grace) exhibits up to 50 sec. delay between IPMI and PDU (with IPMI being late), which possibly points to a slower update of the IPMI readings at BIOS level;

- the export scripts is not perfect, especially at high frequency (1 sec.), making the two time-series asynchronous. However, this effect is minimal and can be mitigated by re-synching data and filling gaps.

In conclusion: we observe an agreement on the average power calculated from the two readings within a **|Δ<power>$_{\textbf{PDU-IPMI}}$| ≤ 5%** (on reasonable hardware) or **≤ 6%** in pathological cases (Grace test-box). The Δ is mostly positive (PDU > IPMI), with one exception.

There is a similar agreement over the integrated energy, and we observe an even lower discrepancy in the Maximum power (a few Watts, or below **2.5%**).

The **6%** difference here is just a wobble (see previous slide).



in-House (production)

~ 1.5k cores

2xIntel40ht: Dual Socket Intel XEON E5-2630 v4 (HP)
CPU: 2 * x86 Intel(R) Xeon(R) E5-2630 v4, 10C/20HT @ 2.2GHz (TDP 85W)
RAM 160GB (4 x 32GB + 4 x 8GB) DDR4 2400 MHz → 4 GB/core
HDD: 2TB disk SATA @ 7200 RPM

~ 5k cores

2xAMD64ht: Dual Socket AMD EPYC 7513 (DELL)
CPU: 2 * x86 AMD EPYC 7513 (Milano), 32C/64HT @ 2.6GHz (TDP 200W)
RAM: 512GB (16 x 32GB) DDR4 3200MT/s → 4 GB/core
HDD: 3.84TB SSD SATA Read Intensive

~ 7.5k cores

2xAMD64ht: Dual Socket AMD EPYC 7452 (DELL)
CPU: 2 * x86 AMD EPYC 7452 (Roma), 32C/64HT @ 2.35GHz (TDP 200W)
RAM: 512GB (16 x 32GB) DDR4 3200MT/s → 4 GB/core
HDD: 3.84TB SSD SATA Read Intensive

~ 2k cores

2*ARM80c: Dual Socket Ampere Altra Q80-30 (Ampere)
CPU: 2 * ARM Ampere Q80-30, 80C @ 3GHz (TDP 210W)
RAM: 512GB (32 x 16GB or 16 x 32GB) DDR4 3200MT/s → 3.2 GB/cor
HDD: 2 * 1TB NVMe

ARM128c: Single Socket Ampere Altra Max M128-30 (SuperMicro)
CPU: ARM Ampere M128-30, 128C @ 3GHz (TDP 250W)
RAM: 512GB (8 x 64 GB) DDR4 3200MHz → 4 GB/core
HDD: 8TB NVMe

~ 2k cores

in-House (testing)

AMD96ht: Single AMD EPYC 7003 (GIGABYTE)
CPU: x86 AMD EPYC 7643, 48C/96HT @ 2.3GHz (TDP 225W)
RAM: 256GB (16 x 16GB) DDR4 3200MHz → 2.7 GB/core
HDD: 3.84TB SSD SATA

2xAMD48ht+GPU: Dual Socket AMD EPYC 7443 (DELL)
CPU: 2* AMD EPYC 7443, 24C/48HT @ 2.3GHz (TDP 200W)
GPU: 2* NVIDIA A100 PCIe 80GB (TDP 300W)
RAM: 256GB (16 x 16GB) DDR4 3200MHz → 2.7 GB/core
HDD: 480GB SSD SATA + 5TB SSD SCSI

And, we also have a **RISC-V** test box …

ARM80c: Single socket Ampere Altra Q80-30 (GIGABYTE)
CPU: ARM Ampere Q80-30, 80C @ 3GHz (TDP 210W)
RAM: 256GB (16 x 16GB) DDR4 3200MHz → 3.2 GB/core
HDD: 3.84TB SSD SATA

Grace144c: Dual Socket* NVidia Grace (SuperMicro)
CPU: NVidia Grace 144-Core 480GB DDR5 @ 3.4GHz (TDP 500W)
RAM: 480GB (on chip) DDR5 4237MHz → 3.3 GB/core
HDD: 1TB NVMe + 4TB NVMe

Remote Testing

2*AMD256ht: Dual Socket AMD EPYC 9754 (SuperMicro)
CPU: 2 * x86 AMD EPYC 9754 (Bergamo), 128C/256HT @ 3.1GHz (TDP 360W)
RAM: 1.536TB (24 x 64GB) DDR4 3200MHz → 3 GB/core
HDD: 512GB NVMe + 3.84TB SSD

Super Micro

AMD128ht: Single Socket AMD EPYC 8534P (SuperMicro)
CPU: AMD EPYC 8534P (Siena), 64C/128HT @ 3.1GHz (TDP 200W)
RAM: 576GB (6 x 96GB) DDR5 3200MT/s → 4.5 GB/core
HDD: 1TB NVMe Storage

2xAMD192ht: Dual Socket AMD EPYC 9654 96-Core (…)
CPU: AMD EPYC 9654 (Genoa), 96C/184HT @ 3.7GHz (TDP 340W)
RAM: …
HDD: …

@ RAL

ARM128c: Single Socket Ampere Altra Max M128-28 (XMA)
CPU: ARM Ampere M128-28, 128C @ 2.8GHz (TDP 250W)
RAM: 512GB (8 x 64GB) DDR4 3200MHz → 4 GB/core
HDD: 1TB NVMe Storage

XMA

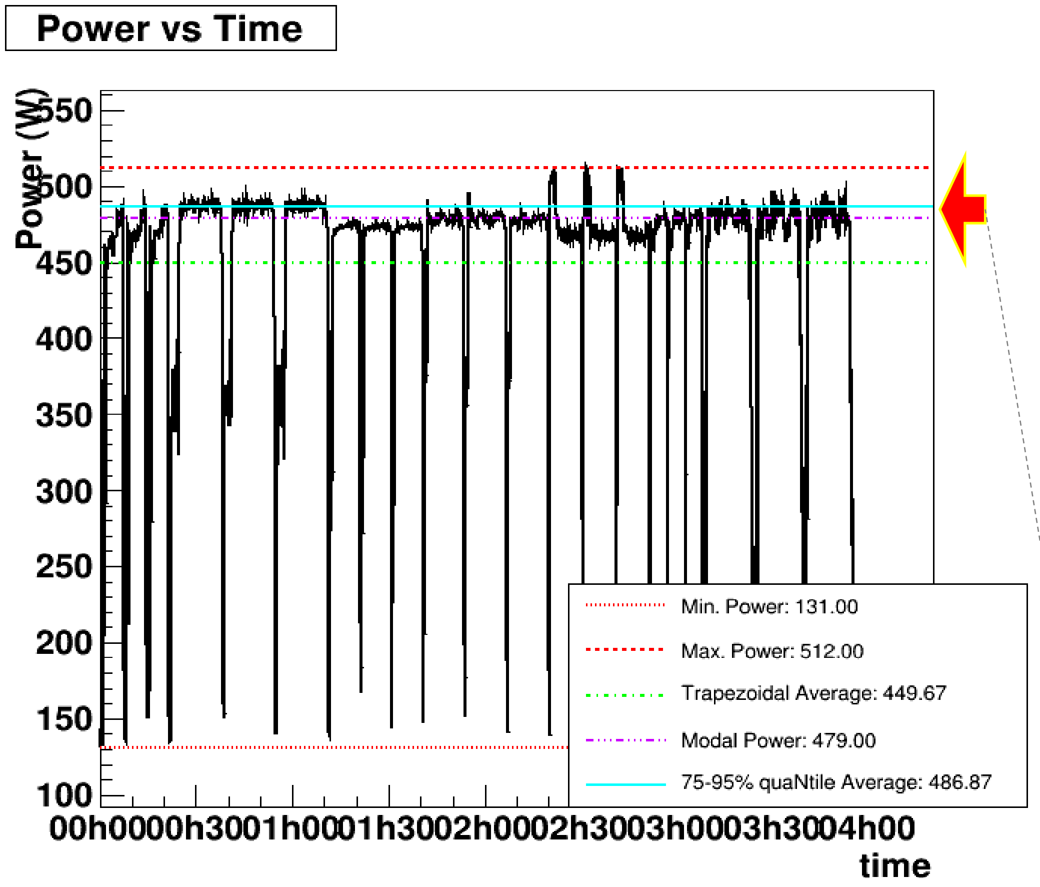
Coming soon : **AmpereOne** (96 - <u>192</u> cores)

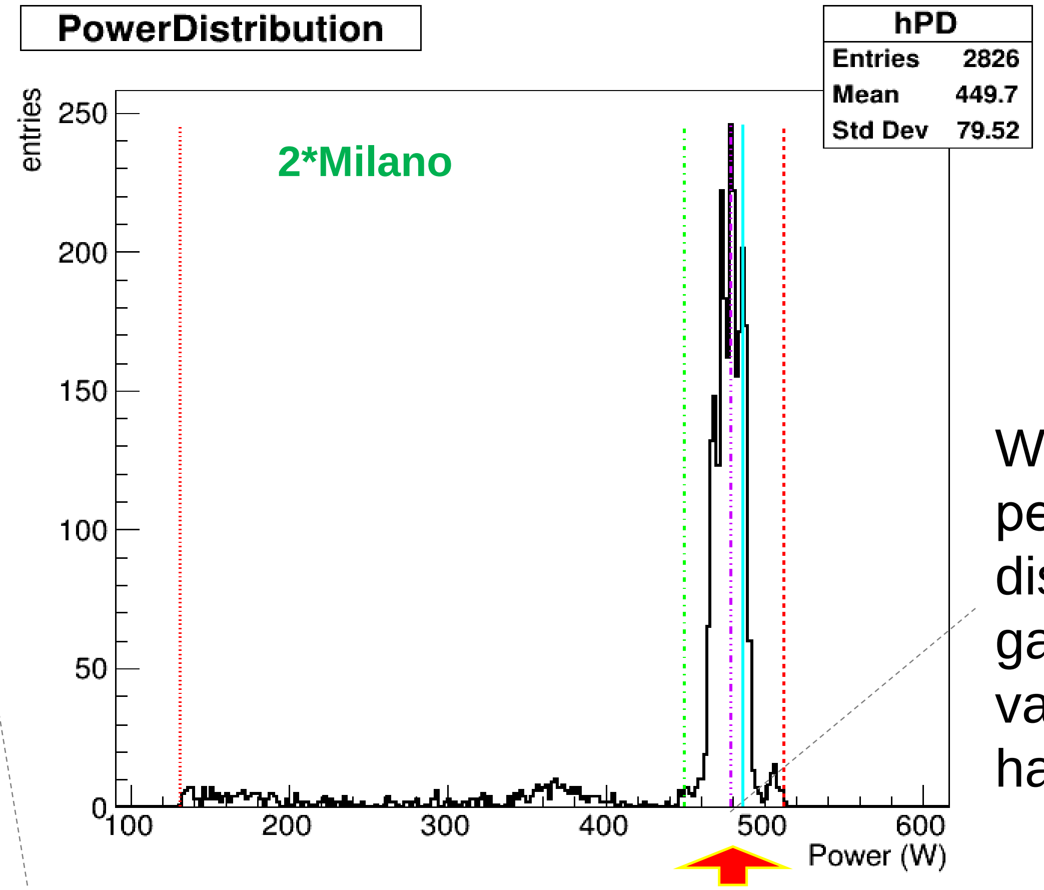
… we expect to get access to a test box next month!

We have expressed our interest in testing new hardware to a few vendors, and from time to time we get remote access to new machines. We have also gathered data from other WLCG sites (**RAL**).

What Watt

We wish to extract an accurate **Figure of Merit** (**FoM**) of power usage for a standard HEP workload from smaller **HEP-Score** containerized jobs, which is easy to implement and consistent across hardware.





We could fit this peak, but … the distribution is not gaussian and varies across hardware.

<75-95%> Blue line sits nicely in the plateau !

By arranging the data in power order we can perform an upper quartile average, but discard the top 5% of data to remove isolated peaks. This we call 75-95% quantile average.

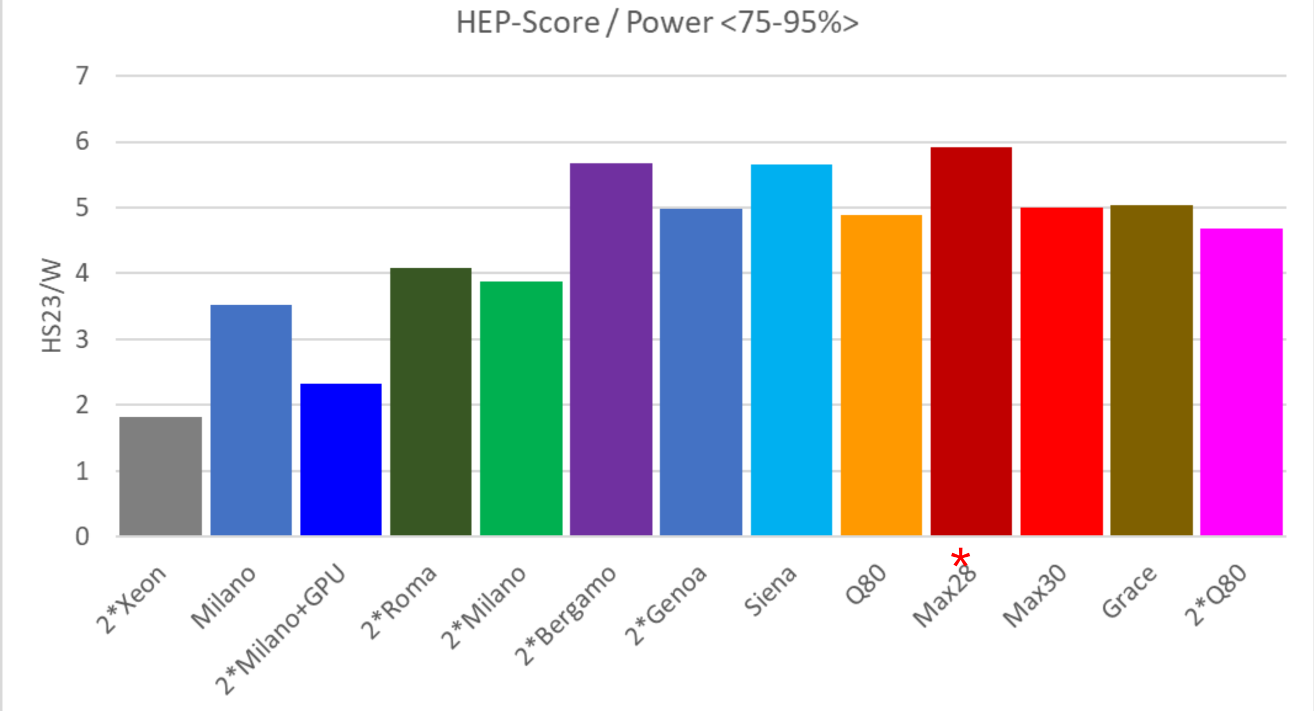
HEP-score/Watt

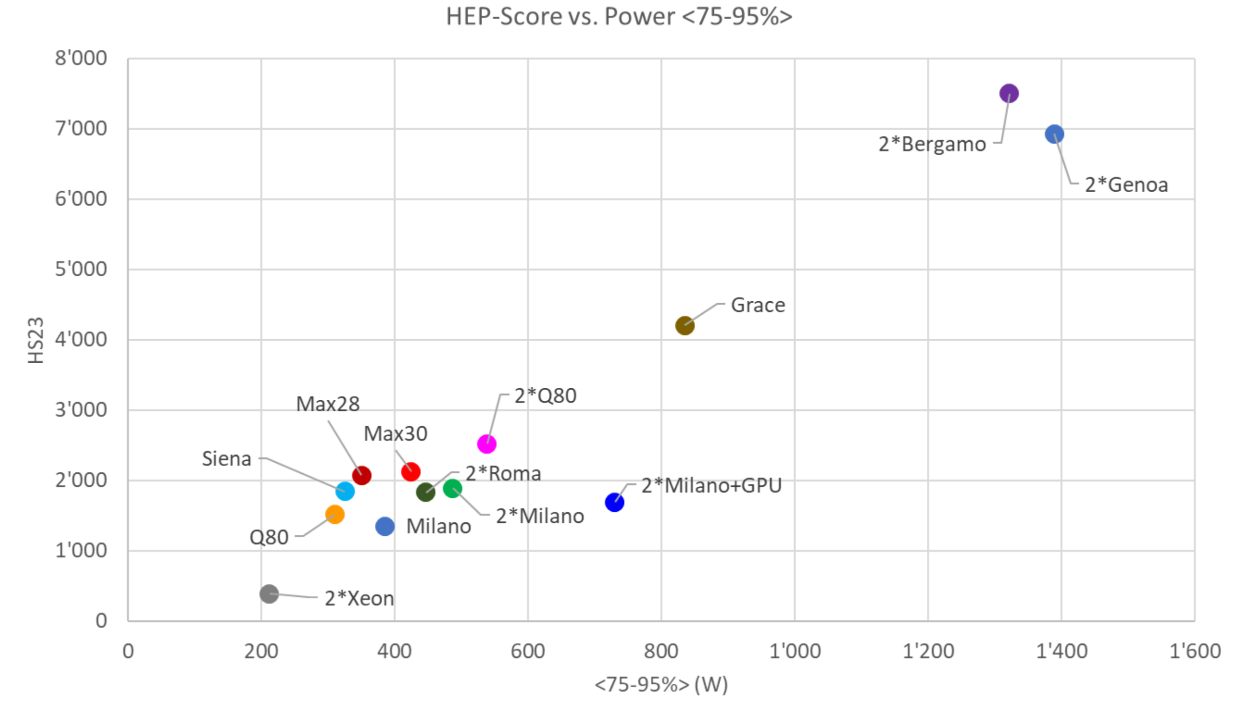
Using this new **FoM**, we measure performance per watt as:

HS23 / Power<75-95%>

GPU not used

Nickname	Machine	CPU	Arch	HT	Threads	Governor	Max Freq. (GHz)
*2*Xeon*	2xIntel20ht	2 * Intel XEON 10-Core CPU E5-2630 v4	2*x86_64	on	40	conservative	2.2
Milano	AMD96ht	AMD EPYC 7643 48-Core Processor	x86_64	on	96	conservative	2.3
*2*Milano+GPU*	2*AMD48ht_gpu	2 * AMD EPYC 7443 24-Core Processor + ~~2* NVIDIA A100 PCIe 80GB~~	2*x86_64	on	96	conservative	4.0
*2*Roma*	2xAMD64ht	2 * AMD EPYC 7452 32-Core Processor	2*x86_64	on	128	conservative	3.3
*2*Milano*	2xAMD64ht_m	2 * AMD EPYC 7513 32-Core Processor	2*x86_64	on	128	conservative	2.6
*2*Bergamo*	2xAMD256ht	2 * AMD EPYC 9754 128-Core Processor	2*x86_64	on	512	conservative	3.1
*2*Genoa*	2xAMD192ht_cor	2 * AMD EPYC 9654 96-Core Processor	2*x86_64	on	384	conservative	3.7
Siena	AMD128ht	AMD EPYC 8534P 64-Core Processor	x86_64	on	128	conservative	3.1
Q80	ARM80c	Ampere Altra Q80-30	aarch64	//	80	conservative	3.0
Max28	ARM128c_2.8	Ampere Altra Max M128-28	aarch64	//	128	conservative	2.8
Max30	ARM128c	Ampere Altra Max M128-30	aarch64	//	128	conservative	3.0
Grace	NVidia144c	NVidia Grace 144-Core 480GB DDR5	2*aarch64	//	144	conservative	3.4
*2*Q80*	2xARM80c	2 * Ampere Altra Q80-30	2*aarch64	//	160	conservative	3.0



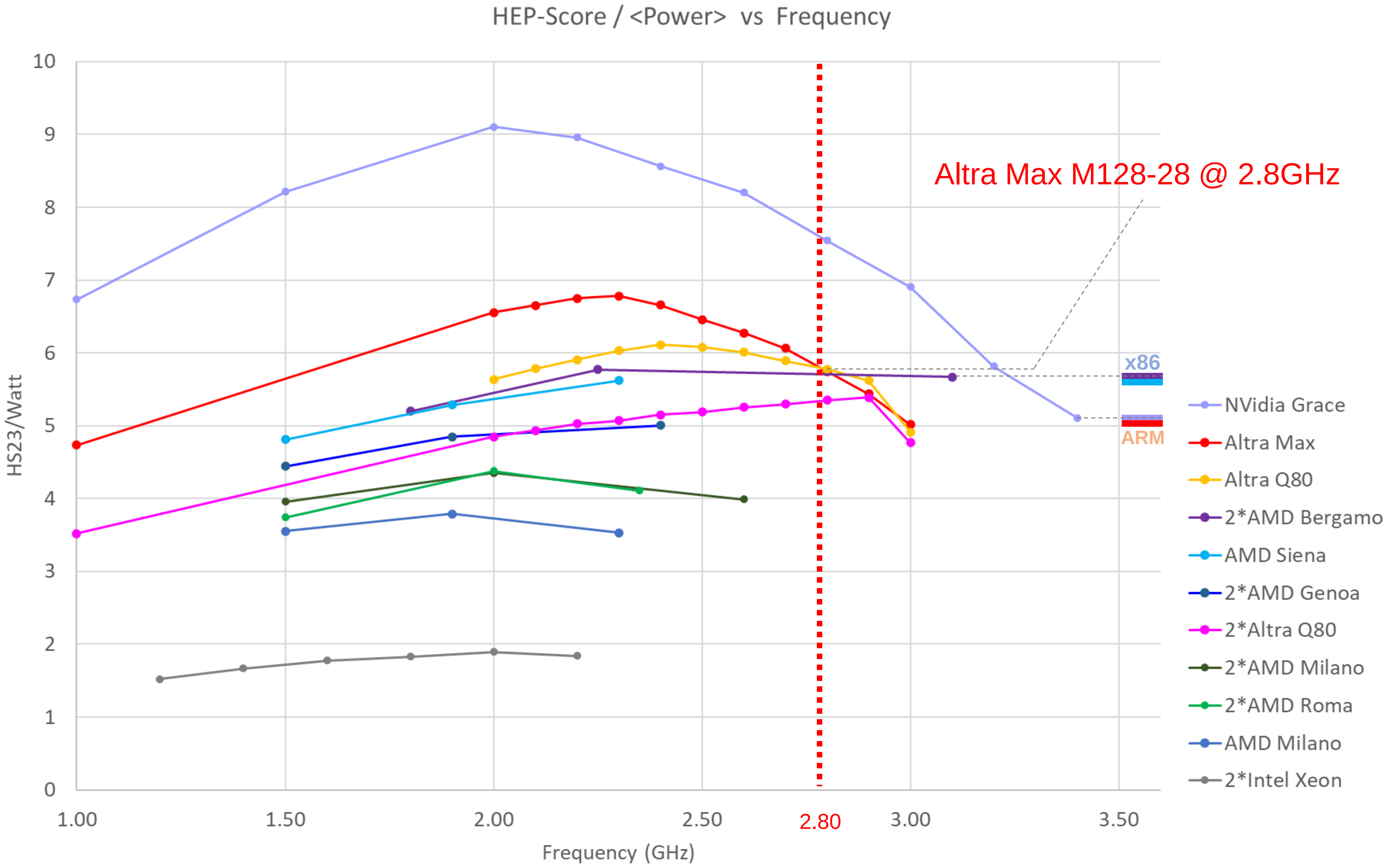


Frequency Scan

HEP-Score/Watt vs. **CPU Frequency** gives a better picture of hardware potentials and shows optimal performance per watt at mid frequency range. At maximum frequency, **x86**s slightly outperform **ARM**s …

… but, **ARM** CPUs allow for a finer tuning of the clock speed, which can be exploited to obtain a better **HS23/Watt** (at the price of a slightly longer execution time).

* The **AltraMax M128-28** has the exact same profile as the **AltraMax M128-30** but the clock maxes out at 2.8 GHz … which happens to achieve the best Score per Watt of all !

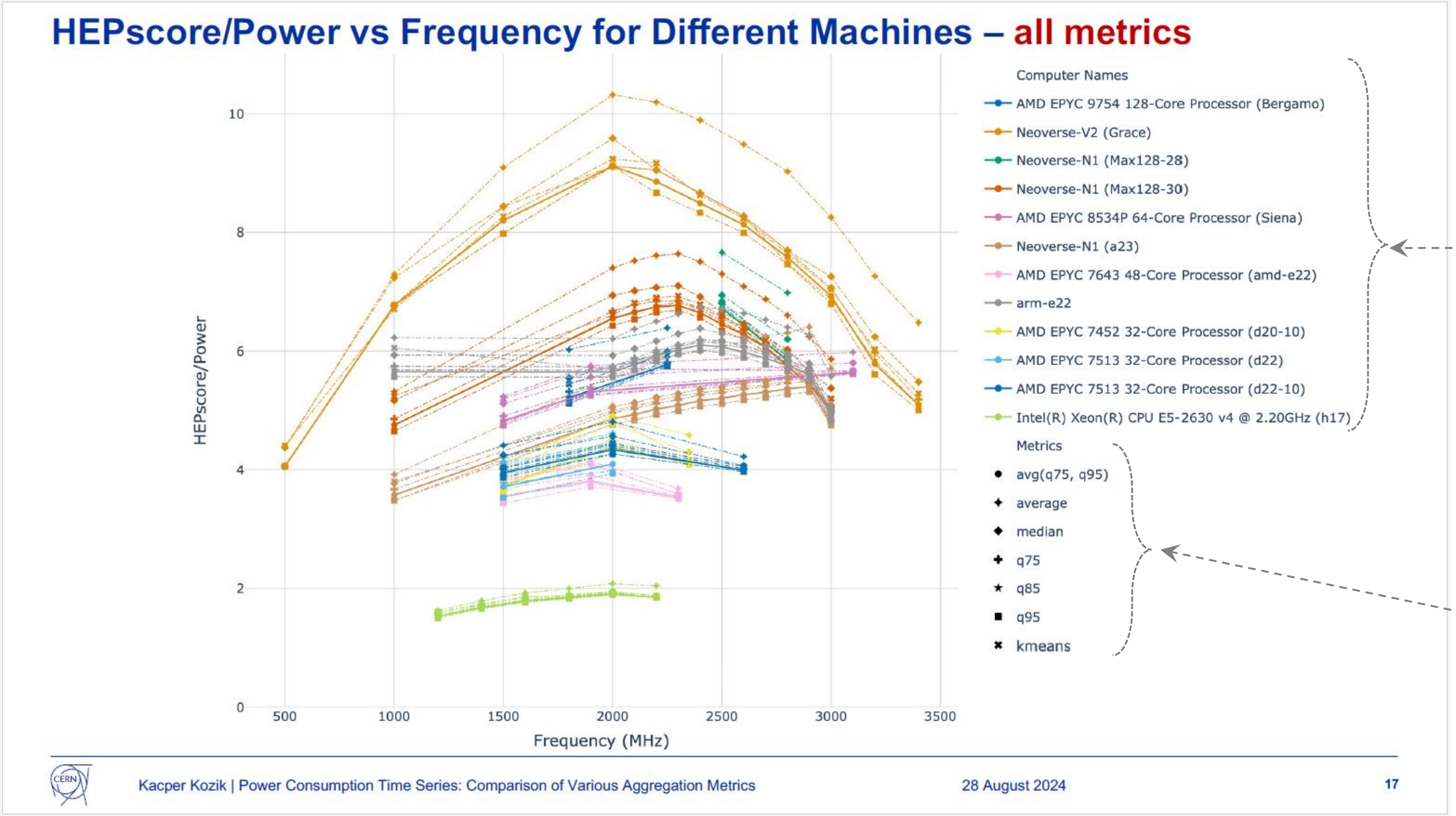


HEP-Score / <Power> vs Frequency

Altra Max M128-28 @ 2.8GHz

What Watt (reprise)

The **HEPiX Benchmark Working Group** has also studied various statistical proxies for power usage. In particular, see the presentation by Kacper Kamil Kozik: https://indico.cern.ch/event/1433496/



HEPscore/Power vs Frequency for Different Machines – all metrics

Kacper Kozik | Power Consumption Time Series: Comparison of Various Aggregation Metrics

28 August 2024

17

The machines are the same from the previous slide, but labels are slightly different.

The “average power” is estimated by using different statistical proxies (metrics), see legend.

Other Benchmarks

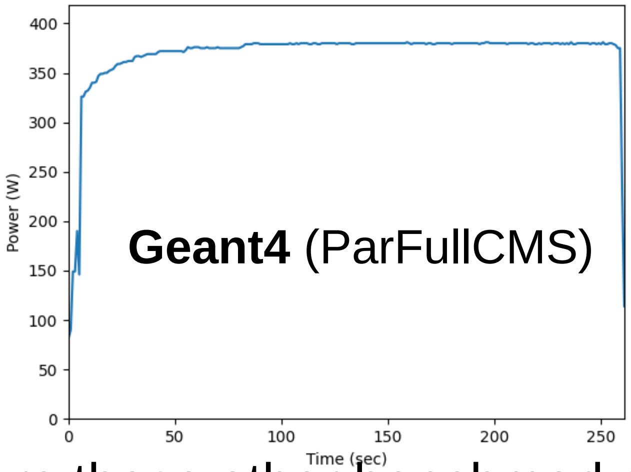
Since **HEP-Score** cannot yet run on every hardware (e.g., **RISC-V**, **GPU**), we may need other benchmarks to assess the performance of new architectures.

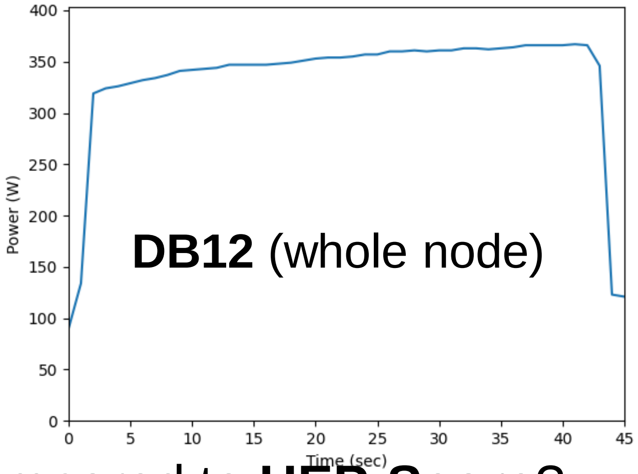
We have tried a few other standard HEP benchmarks: **ROOT bench**, **Geant4** with **CMS** geometry, and **DB12** (single core and whole node).
And … **Celeritas** (see work done by Albert Borbely: https://indico.cern.ch/event/1431542/contributions/6091191/).

ROOT bench:	https://github.com/root-project/rootbench
Geant4:	https://gitlab.cern.ch/geant4/geant4
ParFulCMS:	https://github.com/cms-externals/parfullcms
DB12:	https://github.com/DIRACGrid/DB12

We find this benchmark of little significance for the task at hand. See back-up slides …

The power usage of these benchmarks is calculated as a simple average, as the load shape of whole node benchmarks are almost flat on the high-power plateau (e.g., power timeseries from AMD Milano).



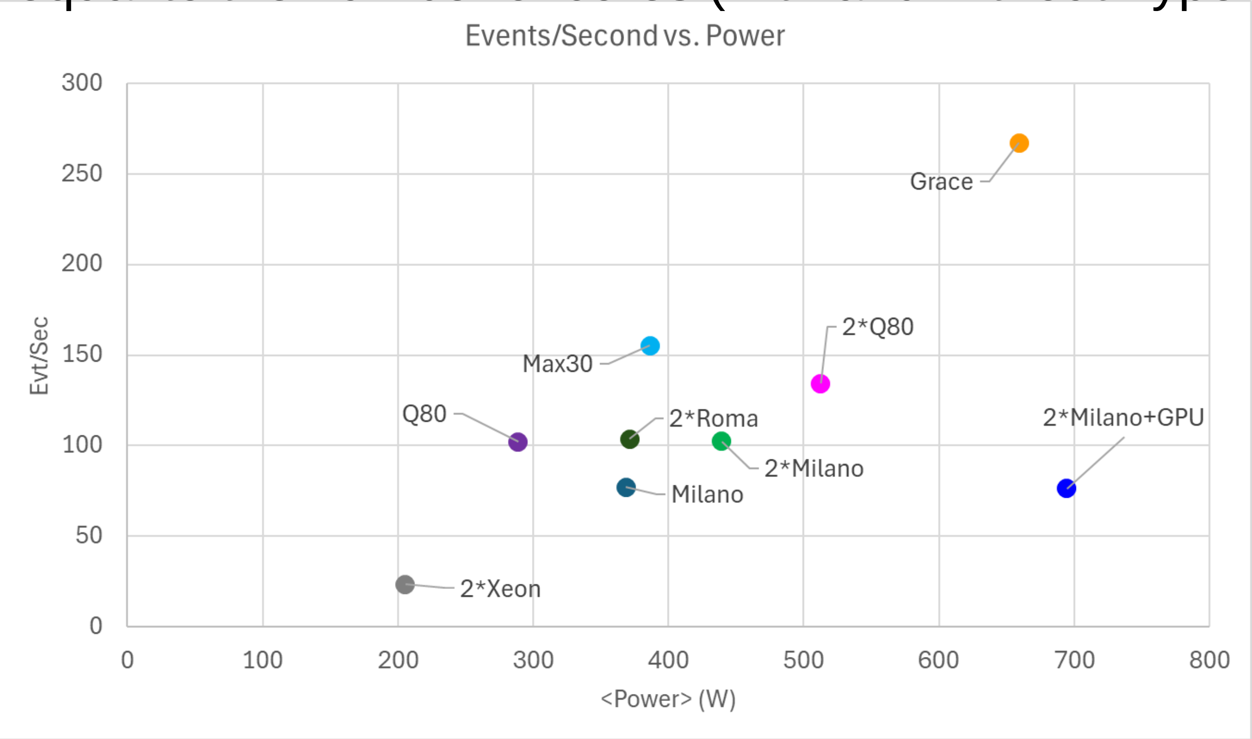


We try to answer these questions: how reliable are these other benchmarks compared to **HEP-Score**? And how can we compare results?

Geant4 Benchmark

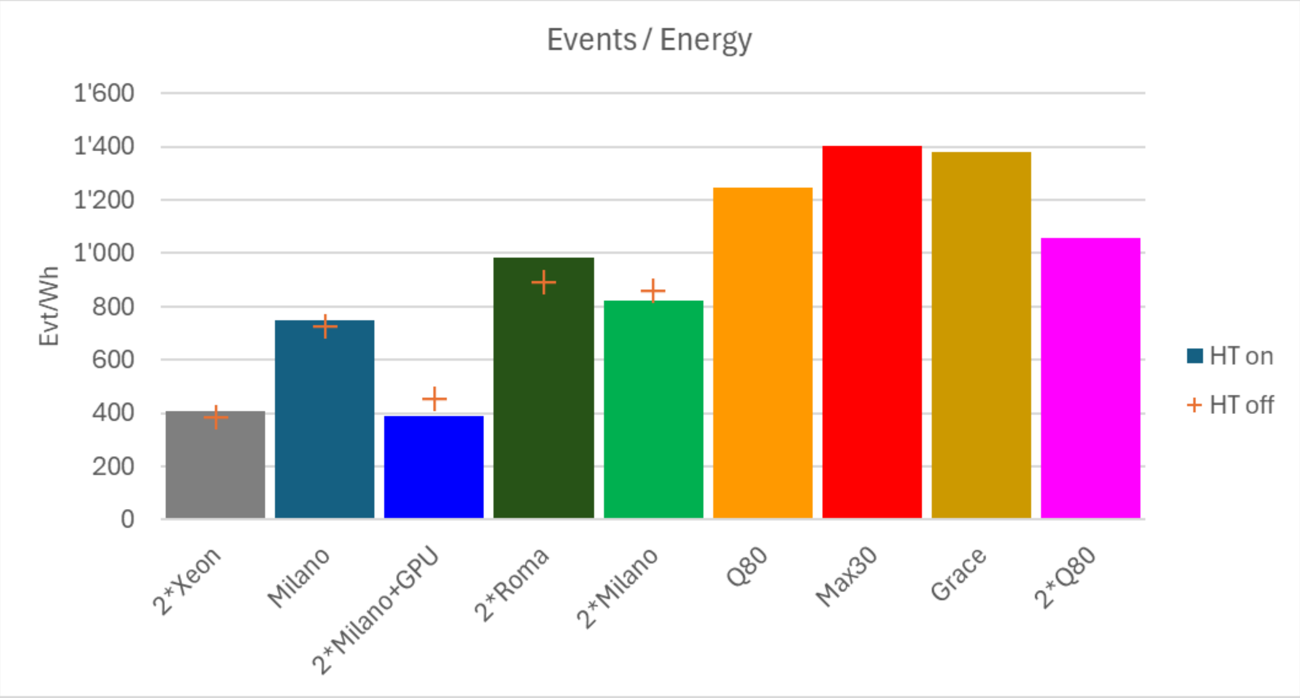
We ran a **Geant4** simulation with **CMS** geometry: https://github.com/cms-externals/parfullcms

With this configuration, we have generated **100k** events as a multithreaded job, with number of threads equal to the number of cores (with and without hyper-threading on **x86** machines).



Geant4 can be forced to use a specific number of threads by setting the environmental variable:

```
export G4FORCENUMBEROFTHREADS=$nproc)
```



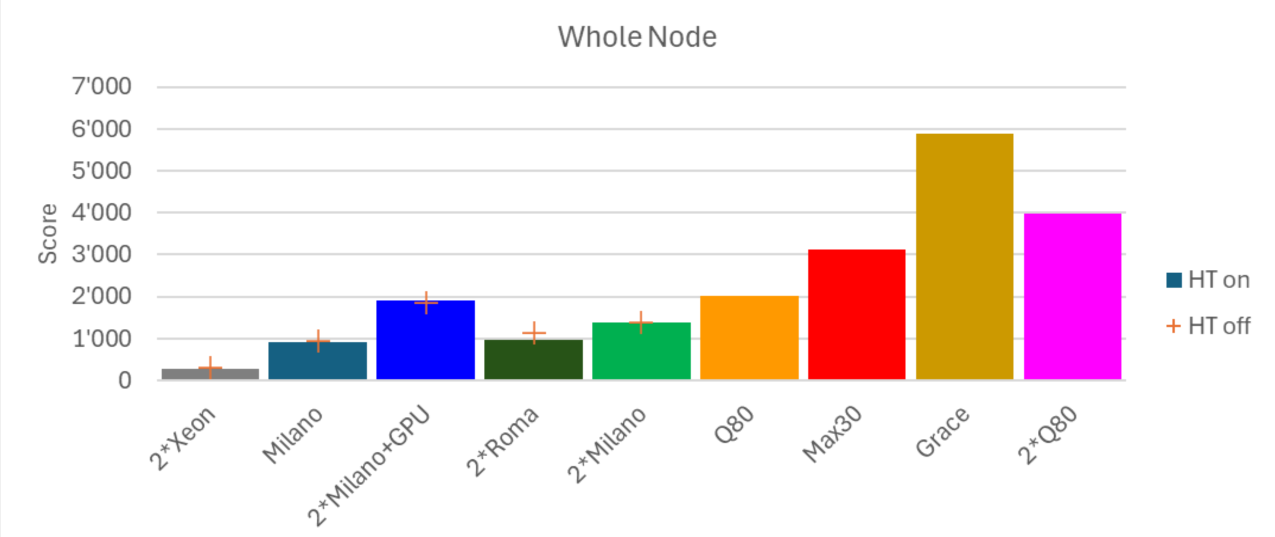
We define our **Geant4 Score** as **Events/Second** (similar to **HEP-Score**, before normalization).

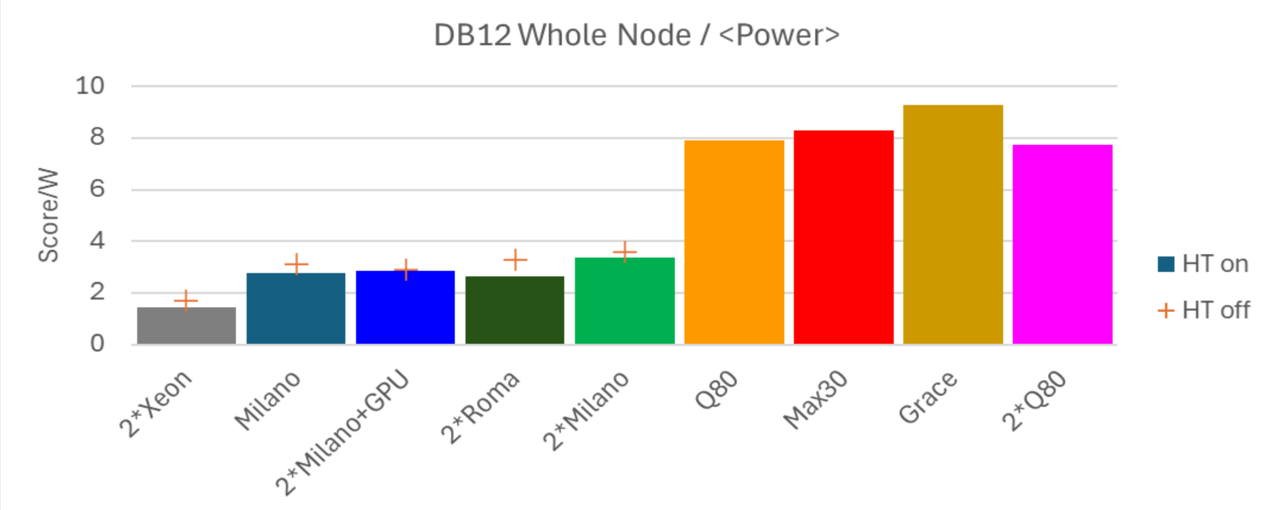
It is interesting to note that:

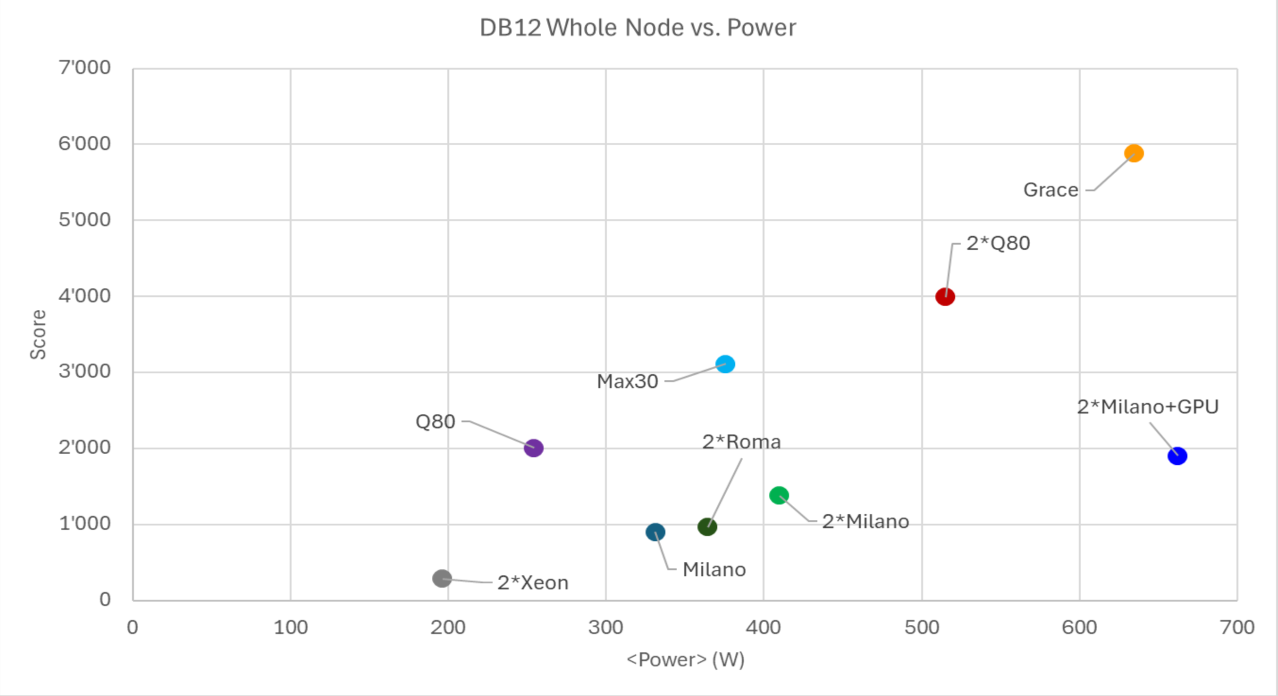
evt/Joule = (evt/sec)/Watt ~ evt/(W*h) ... which should be comparable to **HEP-Score/Watt**.

DB12 Benchmark

DB12 is a benchmark originally developed by the **LHCb** collaboration and written in Python (so it can run almost anywhere). It can run both in **single-core** and **whole-node** mode. We used 10 iterations for longer execution time (and better power measurement) and we focused on the whole node benchmark …



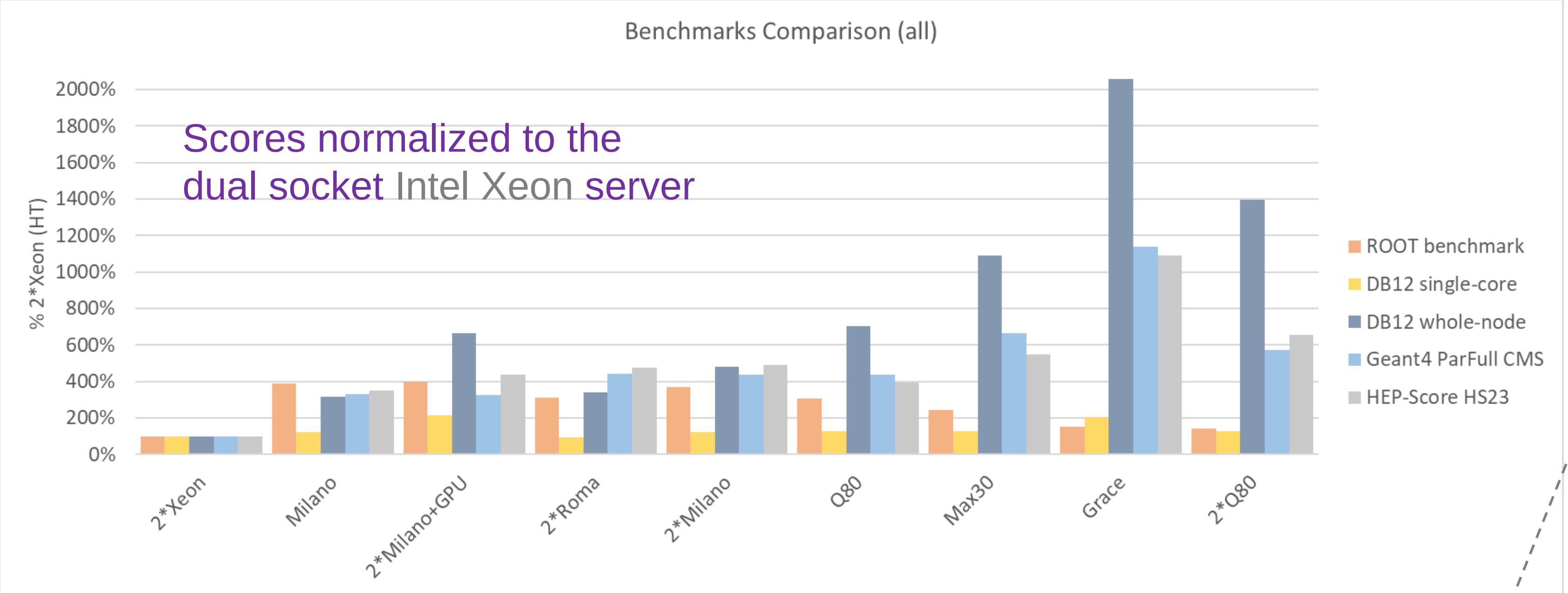




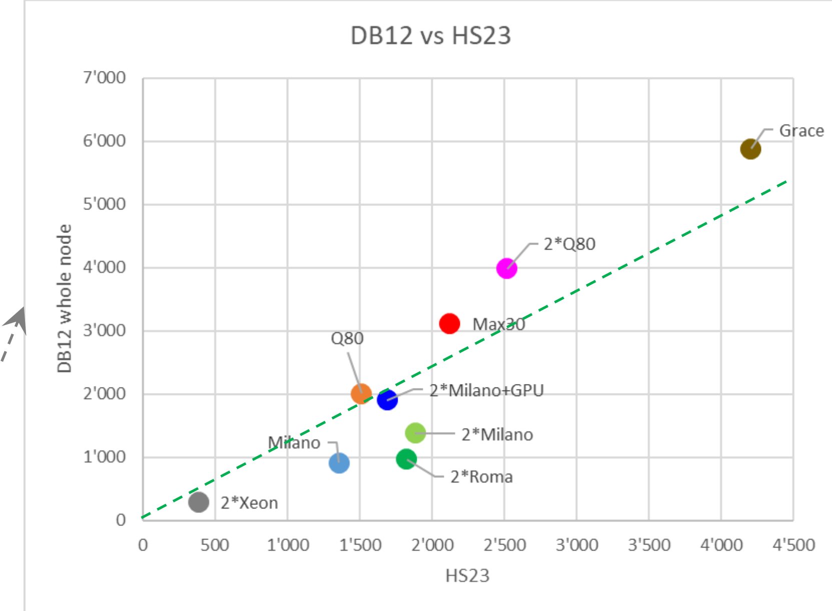
The whole node **Score** (and **Score/Watt**) produce a hardware ranking that looks somehow familiar ...

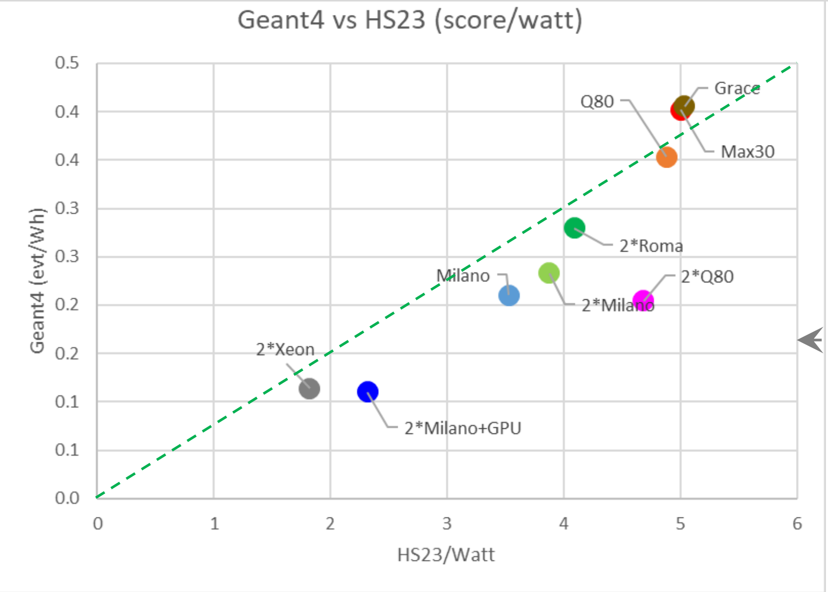
Benchmark Comparison

Finally, we can try to compare the results of these benchmarks over different machines:



Scores normalized to the dual socket Intel Xeon server

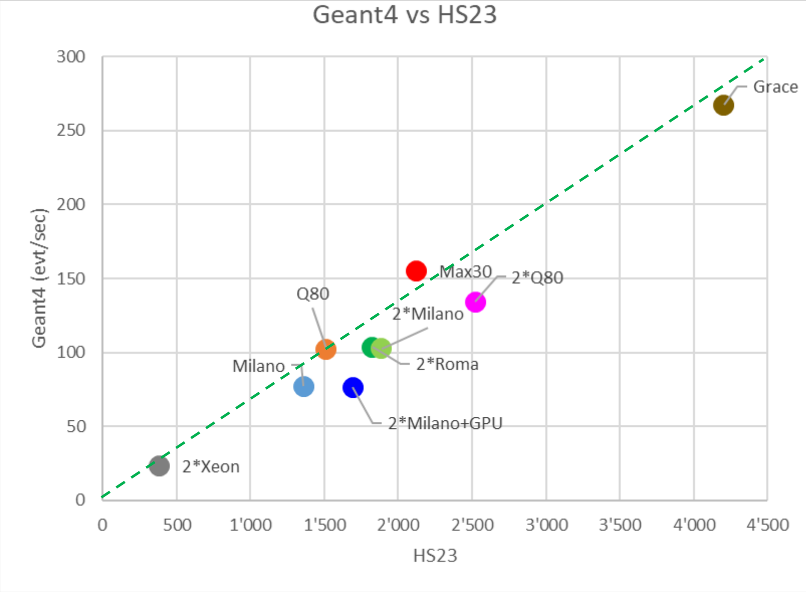




The correlation between **HS23** and **DB12** (whole) is "almost" acceptable. Instead, **Geant4** is a good proxy for HEP-Score.

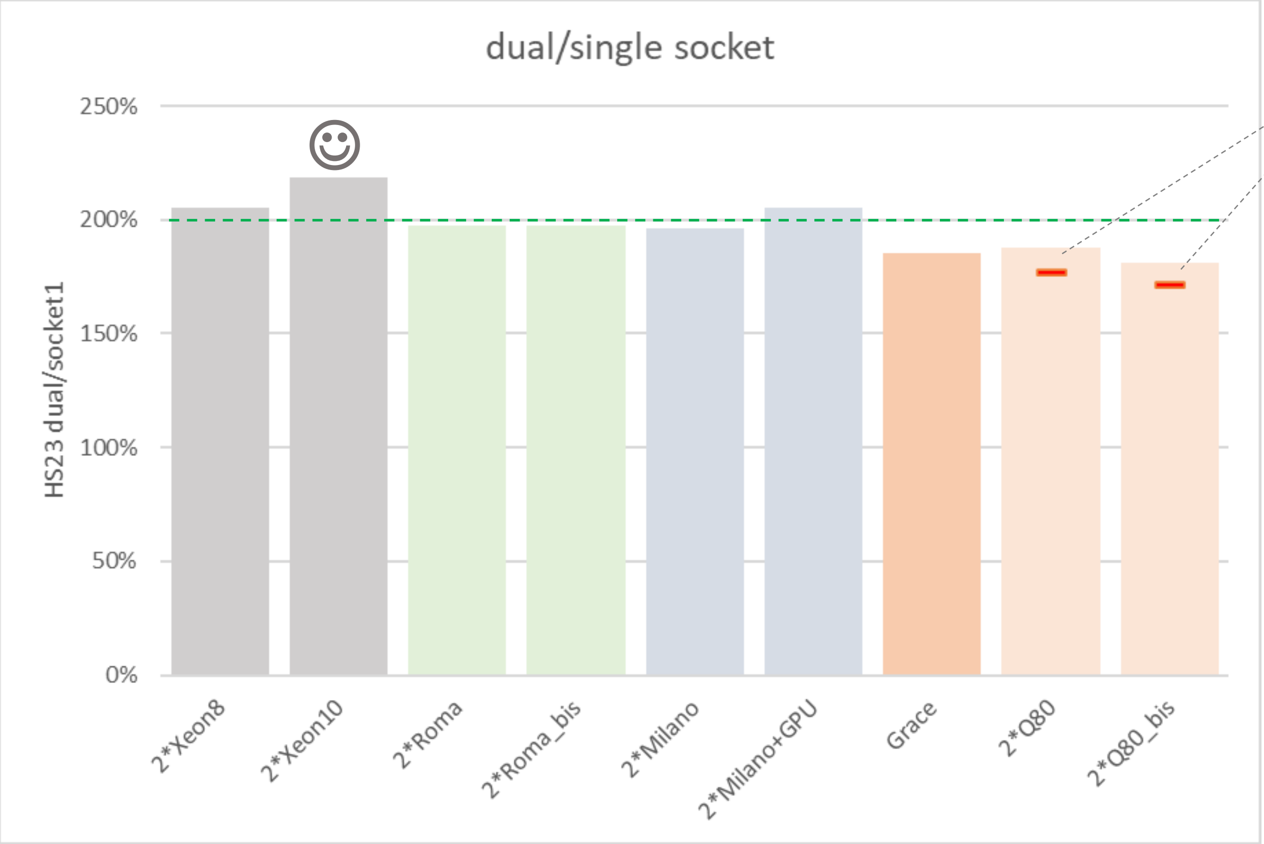
The latter comparison also works decently in term of **HS23/Watt** vs. Geant4 **Evt/Wh**.

Note: green line is not a fit!



Single vs Dual Socket

We have compared the performance of dual socket configuration vs. single socket (on the available dual socket machines: **Ampere Altra Q80**, **AMD Epyc Roma** & **Milano**, and **Intel Xeon**).



Compared to a single socket Q80

Consistent findings show that **ARM** machines in dual-socket configurations exhibit over **10% performance degradation** compared to two single-socket machines, or even compared to the same machine with only one socket enabled.

This effect is a almost absent on **x86** CPUs, where the dual socket configuration is better optimized …

… so much that for our Intel machines 1+1 > 2 ☺

This is a known issue for both **Ampere Altra** and **Altra Max**:

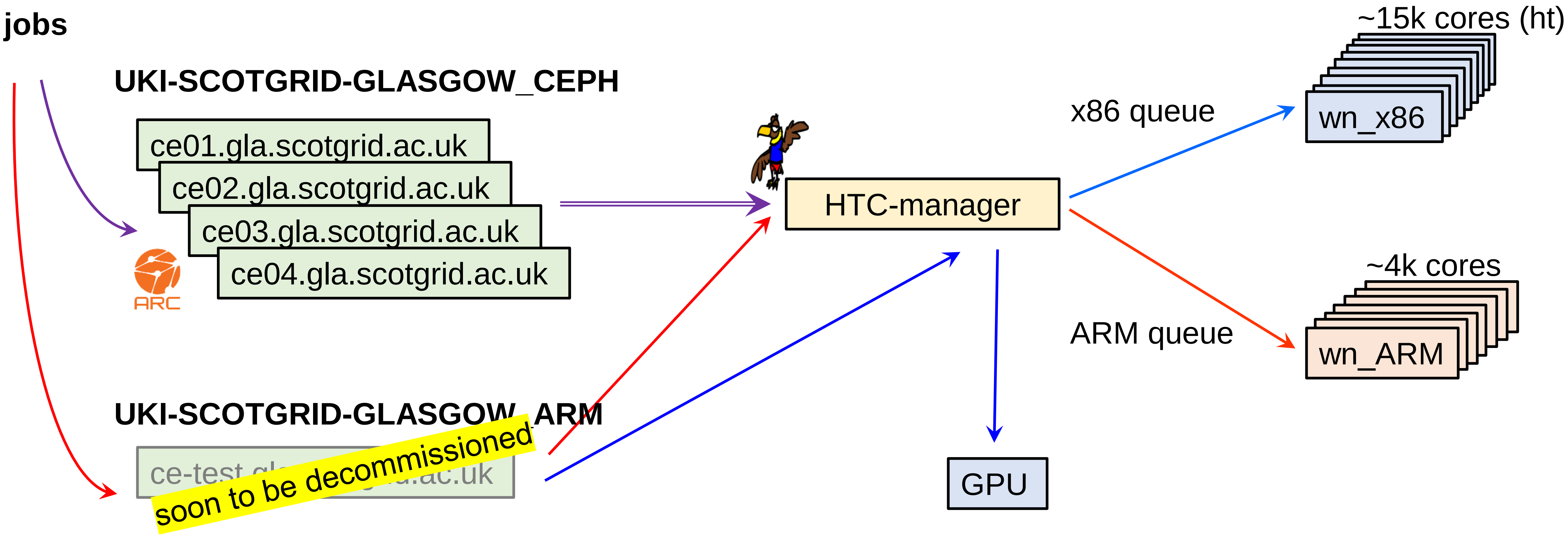
https://www.anandtech.com/show/16315/the-ampere-altra-review/3

https://www.anandtech.com/Show/Index/16979?cPage=2&all=False&sort=0&page=3&slug=the-ampere-altra-max-review-pushing-it-to-128-cores-per-socket

Heterogeneous Compute Cluster

We started providing **ARM** resources at the our WLCG Tier2 cluster by creating a separate queue for ARM (former **ce-test**). After upgrading, we joined both queues within our standard ARC-CE endpoints.

This is a simplified view of our heterogeneous computing cluster (we still keep **ce-test** alive for testing):



jobs

UKI-SCOTGRID-GLASGOW_CEPH

ce01.gla.scotgrid.ac.uk
ce02.gla.scotgrid.ac.uk
ce03.gla.scotgrid.ac.uk
ce04.gla.scotgrid.ac.uk

ARC

UKI-SCOTGRID-GLASGOW_ARM

ce-test.gl d.ac.uk

soon to be decommissioned

HTC-manager

x86 queue

~15k cores (ht)

wn_x86

ARM queue

~4k cores

wn_ARM

GPU

The **`condor_requirements`** setting in the ARC-CE configuration modifies the **ClassAd** for the jobs that **ARC** submits to **HTCondor** by inserting an architecture request (x86, ARM, GPU) …

ARM Physics Validation

Most LHC experiments (**ATLAS**, **CMS**, **ALICE**) have done a first round of extensive Physics Validation campaigns against our ARM cluster @ Glasgow:

- ☺ **ATLAS**: Full simulation and Reconstruction are physics validated.
  <u>ATLAS is ready for pledged ARM resources!</u>
- 😐 **CMS**: Physics validation on ARM mostly successful, but not conclusive.
  <u>CMS is not in a position to use ARM processors in production!</u>
- 😐 **ALICE**: Extensive test of MC simulation jobs, no analysis workflows.
  <u>Recommends ARM segregation or mixed queue with enable/disable!</u>
- ☹ **LHCb**: Groundwork & test samples done, full physics validation not done.
  <u>Production use of ARM unlikely before end of 2024!</u>

Latest reports from **GDB** (June 2024 @ CERN): https://indico.cern.ch/event/1356135/

It’s time for **VO**s to start sending ARM jobs our way … we have over 4k ARM cores !

Conclusions & Outlook

- ❖ Improving on the methodology and developing a complete Analysis framework:
  - energy measurement is now integrated in **HEP-Score**
  - HEP-Score analysis package in development

- ❖ Keep exploring new hardware:
  - benchmark **GPU**+**CPU** with Celeritas (Albert Borbely)
  - test the newly released **AmpereOne** as soon as we get access
  - follow up on **RISC-V** updates and integration … see next presentation:

  “Taking on RISC in HEP for Energy-Efficient Computing”

- ❖ Apply what we have learned so far to make educated hardware choices:
  - share this knowledge with WLCG sites
  - develop a common platform for assessing the carbon cost of WLCG computing

- ❖ See plenary talk by David Britton tomorrow for a more high-level picture:

  “Simulating the Carbon Cost of Grid Sites”

Aknowledments

For the hardware:

- University of Glasgow
- ScotGrid* WLCG Tier2
- GridPP (UK)
- RAL WLCG Tier1 & UKRI
- Ampere Computing (US)
- SuperMicro (UK)
- XMA (UK)

For the code:

- The **ROOT**, **Geant4**, and **LHCb** (**DB12**) teams
- The **CMS** collaboration
- The **HEPiX Benchmarking** Working Group

* **ScotGrid Glasgow**: Emanuele Simili, Gordon Stewart, Samuel Skipsey, Albert Borbely, David Britton

end

References

- E.Simili, G.Stewart, S.Skipsey, D.Spiteri, D.Britton, “Power Efficiency in HEP (x86 vs. ARM)” accepted for publication in IOPscience Journal Of Physics: Conference Series as proceeding for the ACAT2022 conference held in Italy in October 2022.

- E.Simili, G.Stewart, S.Skipsey, D.Spiteri, A.Borbely, D.Britton, "ARMing up for HEP: Energy Efficiency of WLCG sites (ARM vs. x86 and beyond)", published in EPJ Web of Conferences Volume 295 (2024) for the CHEP2023 conference held in the USA in May 2023.

- Ampere® Altra® Multi Core Server Processors, https://amperecomputing.com/processors/ampere-altra

- D.Laurie, “IPMItools: Intelligent Platform Management Interface”, https://github.com/ipmitool/ipmitool

- D.Giordano, et al. “HEPiX Benchmarking Solution for WLCG Computing Resources”, Comput Softw Big Sci 5, 28 (2021)

- …

Abstract

The Glasgow ScotGrid facility is now a truly heterogeneous site, with over 4k ARM cores representing 20% of our compute nodes, which has enabled large-scale testing by the experiments and more detailed investigations of performance in a production environment.

We present here a number of updates and new results related to our efforts to optimise power efficiency for High Energy Physics (HEP) research. We will show updated benchmark results, including a new figure-of-merit designed to characterise the power usage during the execution of the HEP-Score benchmark. Previously, community measurements have used either the average or maximum power, neither of which is a good estimator. We expand our HEP-Score/Watt comparison to include additional machines such as Ampere Altra Q80 and M80, NVidia Grace, and the most recent AMD EPYC chips.

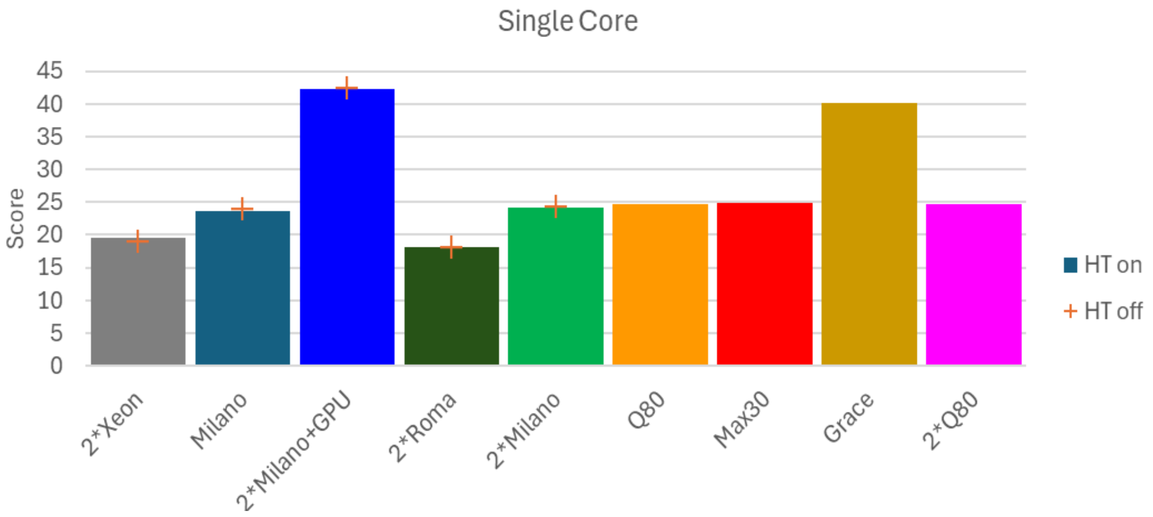
We also introduce a Frequency Scan methodology to better characterize performance/watt trade-offs, potentially informing strategies like frequency scaling during peak hours to optimize power efficiency. In addition, we present a comparison of single-socket versus dual-socket performance, revealing consistent findings that dual-socket configurations exhibit performance degradation compared to two single-socket machines, though of varying magnitudes.

Leveraging HEP-Score jobs and the 'taskset' command to target specific core configurations, we explore performance variations across core groups within the same socket or across dual sockets. Preliminary results show that same-CPU cores have better performance, confirming the importance of workload optimization strategies, such as fine-tuning the job scheduler to prioritize same-socket core utilization.

Our findings contribute to advancing heterogeneous computing strategies and power efficiency optimizations in HEP, paving the way toward more sustainable hardware solutions.

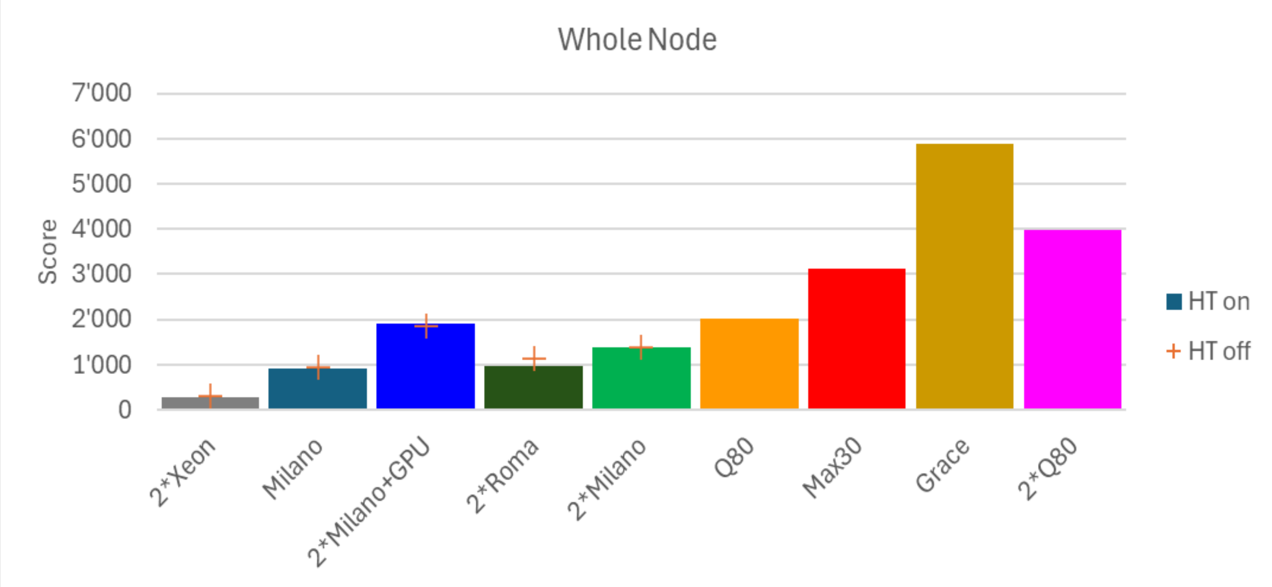
DB12 Benchmark

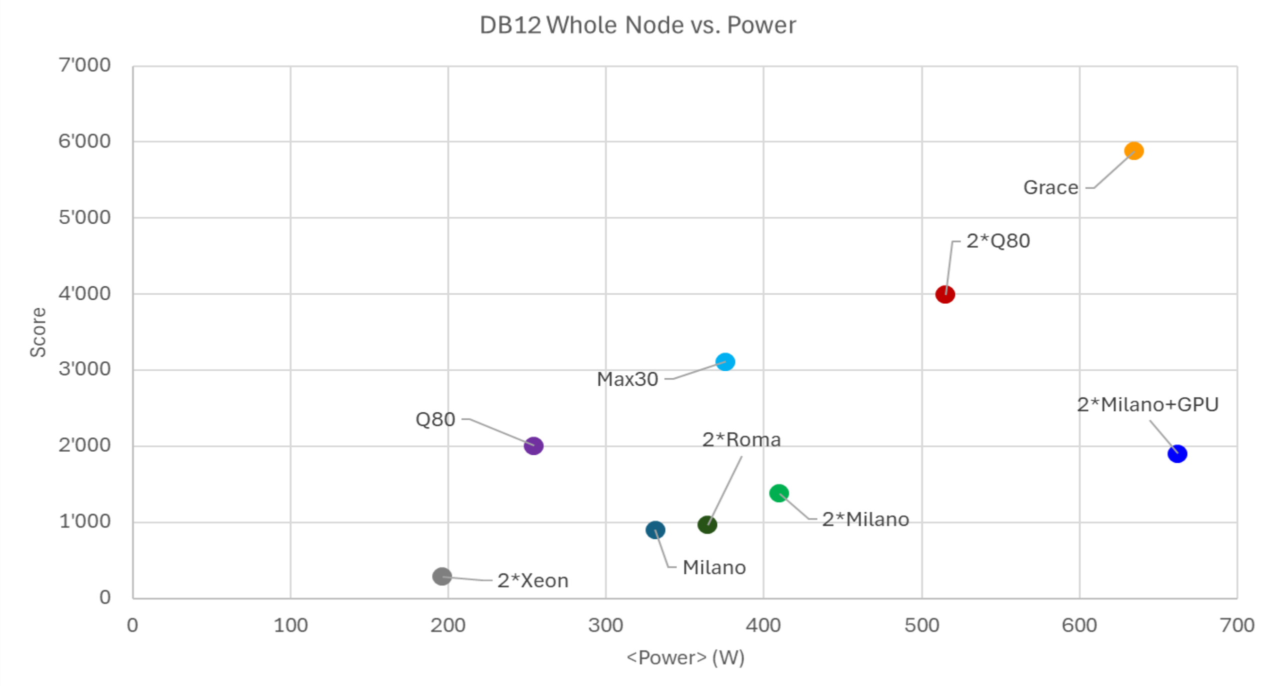
DB12 can run both in **single-core** and **whole-node** mode, it is a benchmark originally developed by the **LHCb** collaboration and written in Python (so it can run almost anywhere).


Single Core

The single core benchmark gives slightly inconsistent results (e.g., why having a passive GPU increase the score?), and cannot be compared to **HEP-Score** (which is a whole-node benchmark).

The whole node **Score** (and **Score/Watt**) is worth trying looking at …


Whole Node

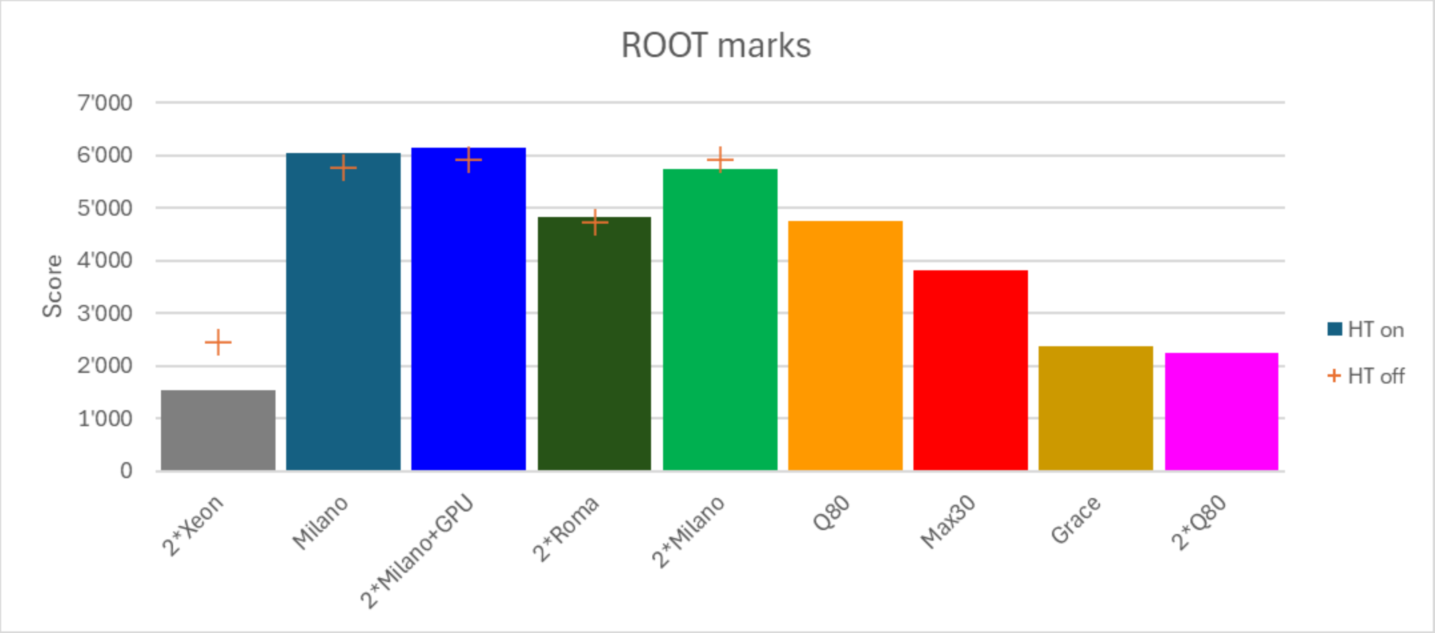

DB12 Whole Node vs. Power

ROOT Benchmark

These are single threaded benchmarks consisting in running typical **ROOT** scripts (fill histogram, fit, …).

Modern **x86** cores perform better than **ARM**. But … results w/out HT are a little inconsistent.

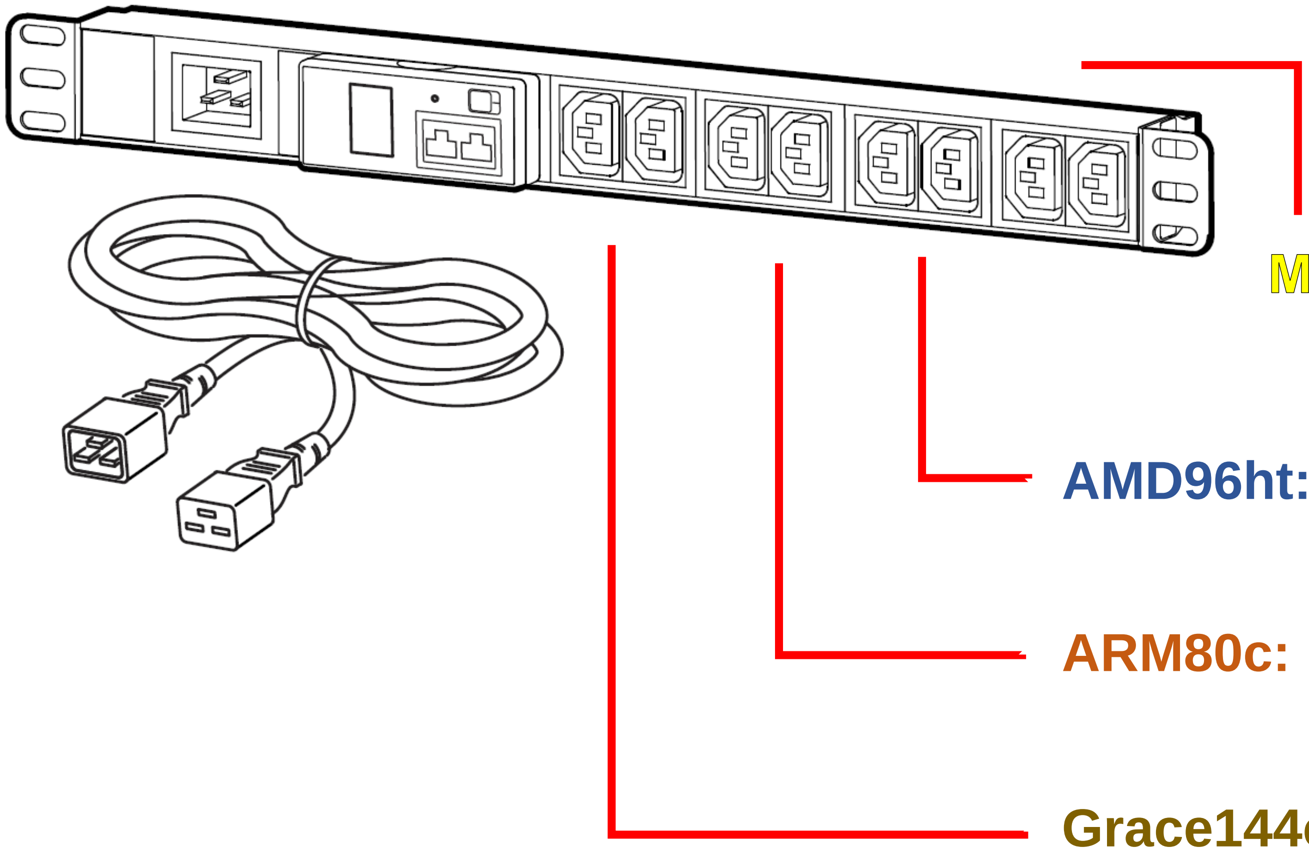
Also, being a single thread benchmark, it does not make sense to calculate Score/Watt, and cannot compare to **HS23** (which is a whole-node benchmark). Also, it executes in a matter of seconds.



In the end, we find this benchmark of little significance for the task at hand. But it may be useful elsewhere.

IPMI Validation

Initial configuration, before we found out that the reading is not per socket ☺



Milk-V Pioneer : Single socket RISC-V 64-Core

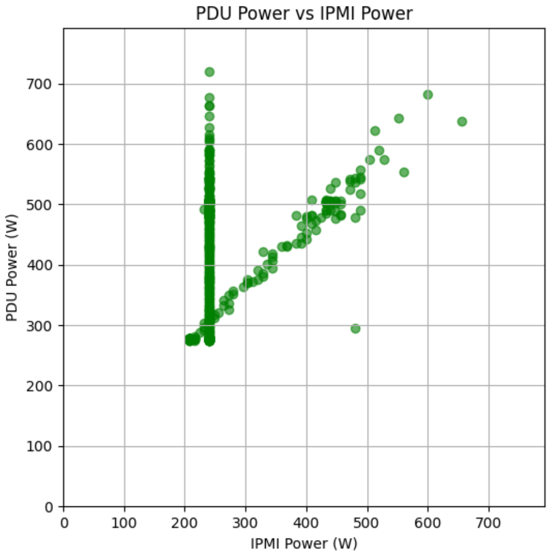
AMD96ht: Single AMD EPYC 7003 48-Core

ARM80c: Single socket Ampere Altra Q80-30 80-Core

Grace144c: Dual Socket NVidia Grace 144-Core

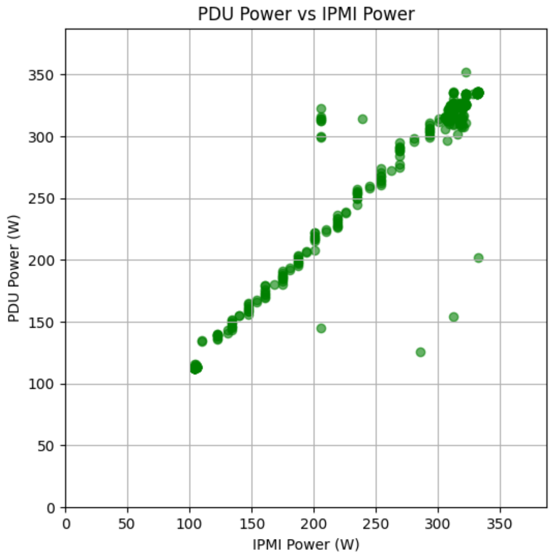
Validation eXtra

Server X (…)

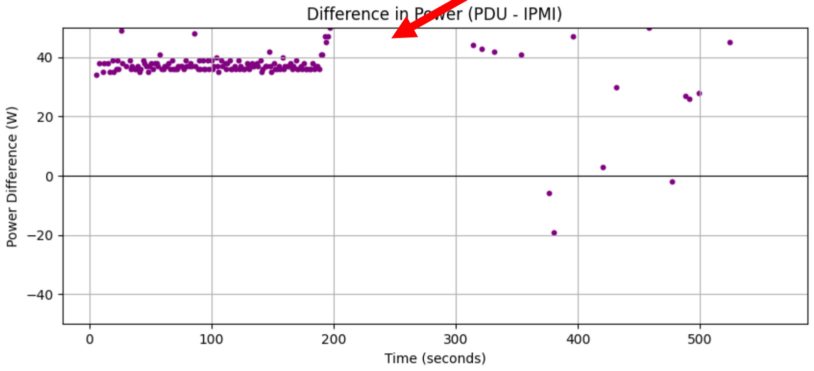


machine	diff_min	diff_max	diff_avg	% max	% average	diff_energy	% energy
*2*IntelXeon*	0.0	1.0	4.3	0.5%	3.1%	0.7	3.2%
AMD Milano	13.0	2.0	5.2	0.5%	1.9%	3.7	5.7%
AltraQ80	9.0	-7.0	-8.1	-2.5%	-4.3%	-1.2	-4.3%
AltraMax	8.0	20.0	9.4	6.0%	4.3%	0.7	1.7%
Grace	6.0	17.0	24.9	2.4%	5.9%	1.4	4.3%
Server X	66.0	64.0	131.9	9.8%	49.8%	19.7	49.0%

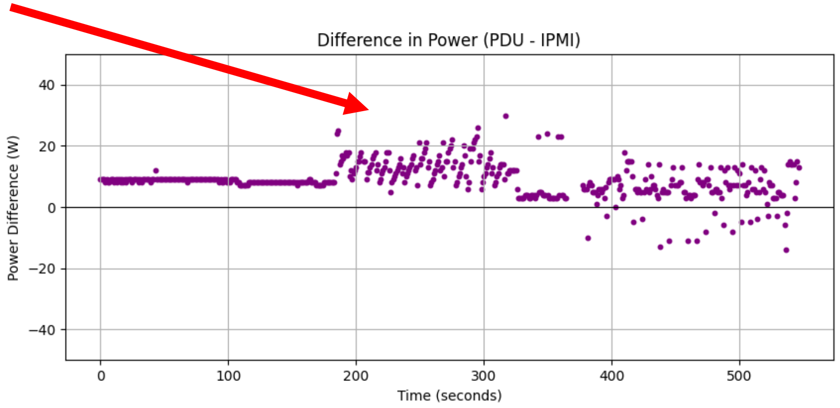
AltraMax (SuperMicro)



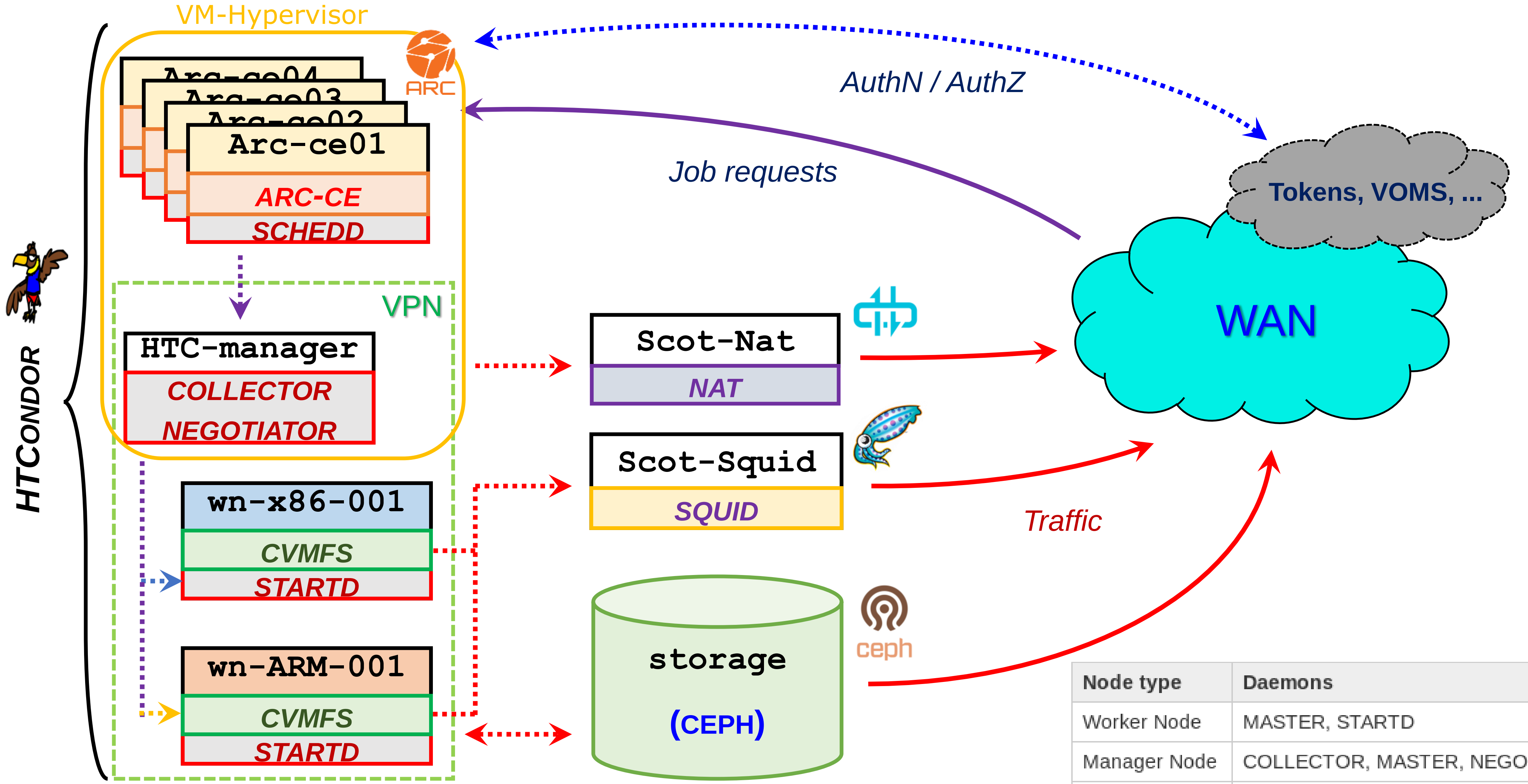
off-scale ! (Δ>50 W)



decent (Δ<10 W)



ScotGrid Tier2 Cluster Overview



Node type	Daemons
Worker Node	MASTER, STARTD
Manager Node	COLLECTOR, MASTER, NEGOTIATOR, SCHEDD
CE Node	MASTER, SCHEDD

Emerging Architectures

We have acquired a RISC-V desktop PC and started experimenting with it:

Milk-V Pioneer : Single socket RISC-V 64-Core Processor (Milk-V)

CPU: SOPHON SG2042 (64 Core C920, RVV 0.71) riscv64 @ 2GHz (TDP 120W)
RAM: 128GB (4 x 32GB) DDR4 3600 MT/s → 2 GB/core
HDD: 1TB PCIe 3.0 NVMe
OS: Fedora 38

Main motivations:

- Open-source and royalty-free architecture,
- Extremely low power usage (**140 Watts** @ full load - 64 cores),
- Growing ecosystem and potential for fast innovation (e.g., EPI will build on RISC-V).

See next presentation …