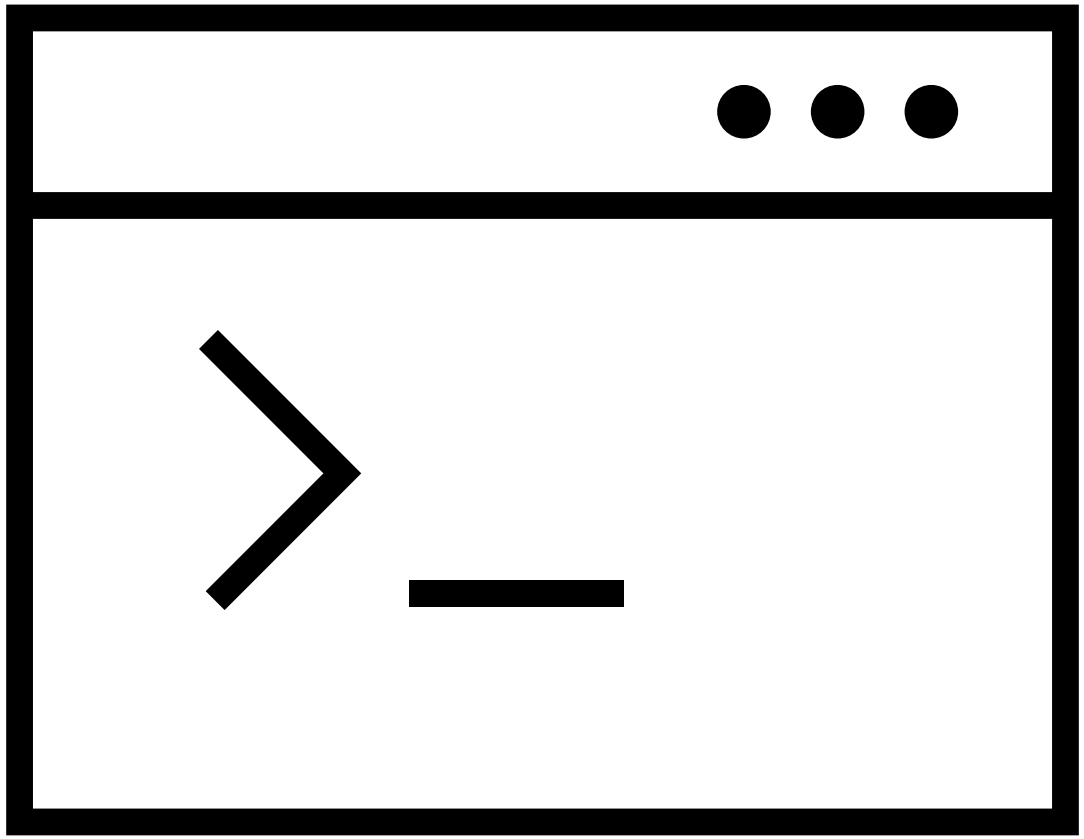
ARM compute @ Glasgow
Status & Updates



Outline

- Update on Glasgow cluster (heterogeneous compute)
  - ScotGrid Glasgow cluster fully updated
  - **ARM** & **x86** resources “transparently” available through CE endpoints
  - Physics Validation updates (June 2024)

- Benchmarking and Power Measurement
  - updates on **HEPscore/Watt** & **Frequency Scan**
  - new **IPMI** validation campaign against external **PDU**
  - first step in testing **RISC-V** architecture for **HEP**

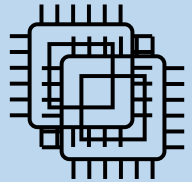
- Upcoming conferences
  - Few contributions will be presented at **HTCondor** workshop (AMS, Sep. 2024)
  - A plenary and a few track talks accepted at **CHEP 2024** (KRK, Oct. 2024)

- Outlook …

in-House (production)

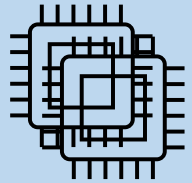
2xIntel40ht: Dual Socket Intel XEON 10-Core CPU E5-2630 v4 (HP) — **~ 1.5k cores**

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



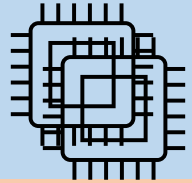
2xAMD64ht: Dual Socket AMD EPYC 7513 32-Core Processor (DELL) — **~ 5k cores**

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



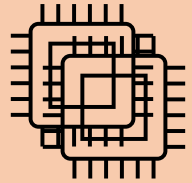
2xAMD64ht: Dual Socket AMD EPYC 7452 32-Core Processor (DELL) — **~ 7.5k cores**

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



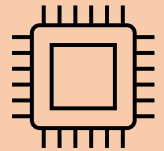
2*ARM80c: Dual Socket Ampere Altra Q80-30 80-Core Processor (Ampere) — **~ 2k cores**

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/core
HDD: 2 * 1TB NVMe



ARM128c: Single Socket Ampere Altra Max M128-30 128-Core Processor (SuperMicro) — **~ 2k cores**

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



Cluster Update

We have completed the update of our Tier2 cluster at Glasgow, due to the long-awaited End Of Life (in June 2024) of **CentOS7,** which we used for the past 4-5 years:

- all compute nodes and services updated to **Alma 9**

- all CEPH nodes updated to **Alma 8** and **CEPH v14** (**Nautilus**), next: **Pacific**

- batch system updated to **HTCondor 10.0**, **ARC-CE v6.20** (waiting for **v7**), no **ARGUS**

- updated monitoring services: **Prometheus 2.5**, **Loki 3**, **Grafana 11**, …

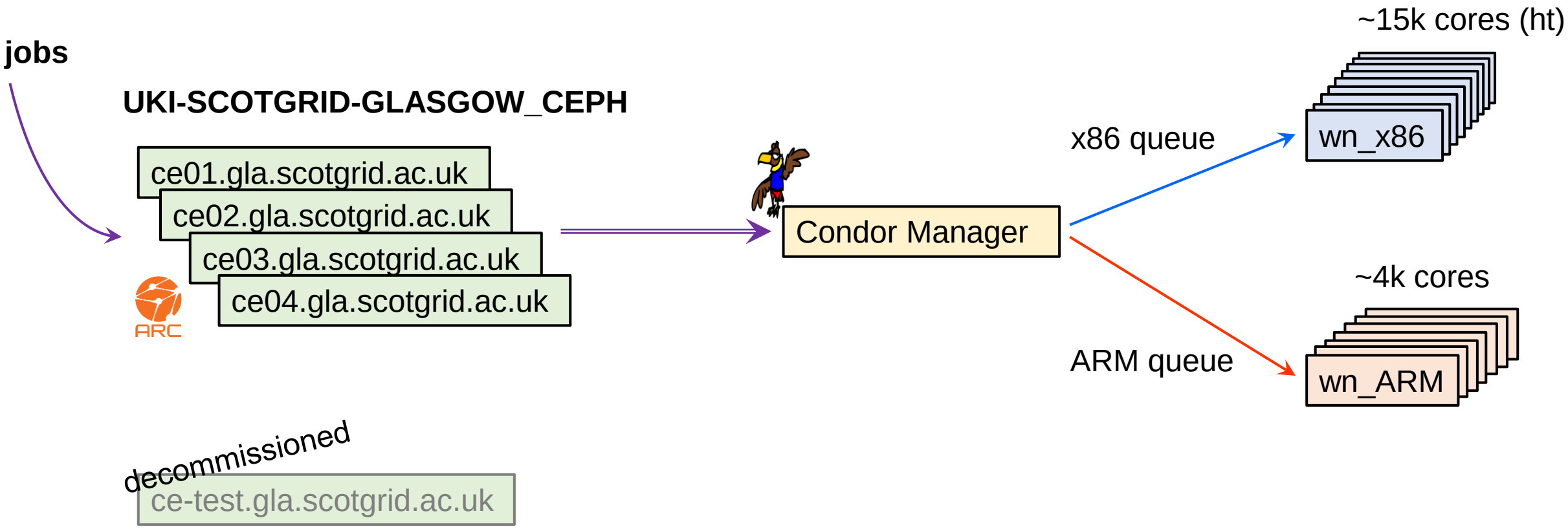
- updated management stuff: **Ansible v2.14**, **GitLab v17.3**, …

- installed **perfSONAR 5.1 testpoint** (rather than **toolkit**)

Heterogeneous Compute Cluster

We started providing **ARM** resources by creating a separate queue for ARM (former **ce-test** endpoint). Then, after upgrading the whole **HTCondor** cluster to **v10**, we tried to join the queues …

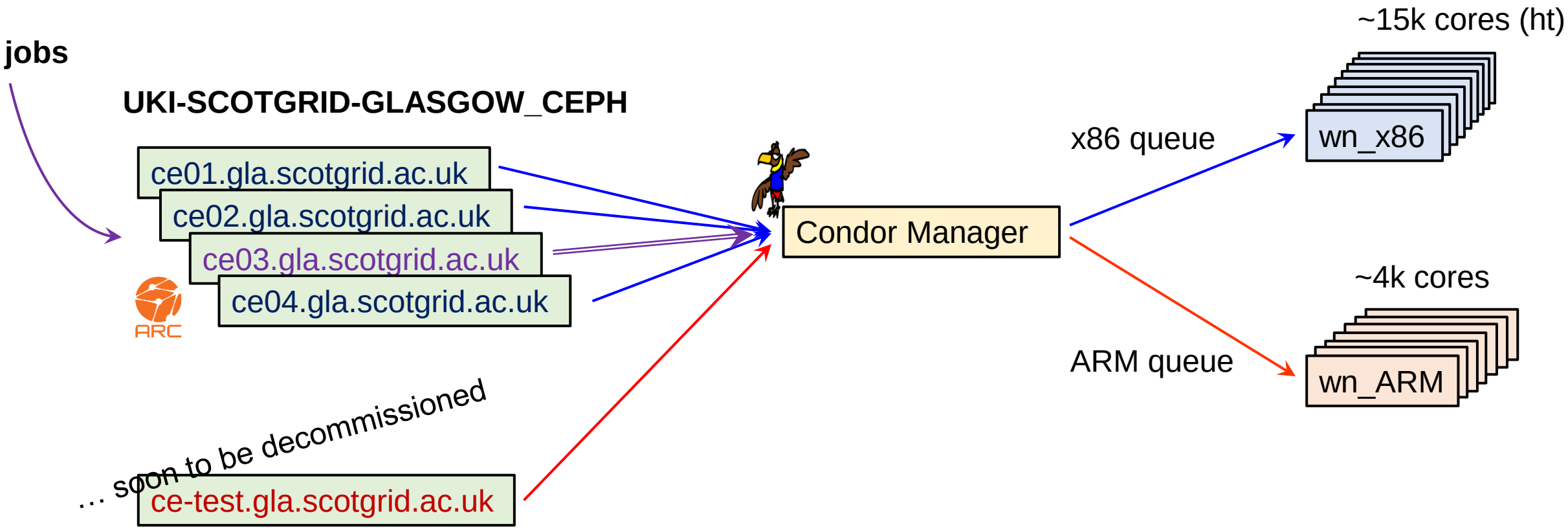
This is the idealized view of our heterogeneous computing cluster:



The **`condor_requirements`** setting in the ARC-CE configuration modifies the **ClassAd** for the jobs that ARC submits to Condor by inserting an architecture request (*)

Heterogeneous Compute Cluster Issues

In reality, we only enabled the dual queue on **ce03**, because not all VOs can deal with the architecture settings (e.g., *BioMed*), and we kept the **ce-test** entry point because it is still in use by some VOs, while the other **ce**s are still single queue (x86).
So, this is the actual view of our heterogeneous compute cluster:



We are still in a transitioning state, but as more **VOs** adhere to the dual queue standard, we hope to make the dual queue submission mechanism more ideal …

Heterogeneous Compute Cluster Config

(*) The **`condor_requirements`** setting in the ARC-CE configuration modifies the **ClassAd** for the jobs that ARC submits to Condor by inserting an architecture request.
Because … **HTCondor** doesn't have a concept of queues: it matches jobs to resources based on their ClassAd, which includes an architecture entry …

```
[queue:condor]
 comment = Condor queue
 condor_requirements = (Arch == "X86_64" && (TARGET.GPUs IS UNDEFINED || TARGET.GPUs == 0))

[queue:condor_arm]
 comment = Condor queue (ARM)
 condor_requirements = (Arch == "aarch64" && (TARGET.GPUs IS UNDEFINED || TARGET.GPUs == 0))

[queue:condor_gpu]
 comment = Condor queue (x86 + GPU)
 condor_requirements = TARGET.GPUs > 0
```

This also works for **GPU** queue (tested already !)

What **VOs** need to do is to ensure that their job submission mechanism specifies the appropriate ARC queue (i.e., what architecture they are targeting).

Note: the ARC default queue selection appeared buggy in earlier version of **ARC v6**, hopefully it will improve in **v7**.

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.
ATLAS is ready for pledged ARM resources!

😐 • **CMS**: Physics validation on ARM mostly successful, but not conclusive.
CMS is not in a position to use ARM processors in production!

😐 • **ALICE**: Extensive test of MC simulation jobs, no analysis workflows.
Recommends ARM segregation or mixed queue with enable/disable!

☹ • **LHCb**: Groundwork & test samples done, full physics validation not done.
Production use of ARM unlikely before end of 2024!

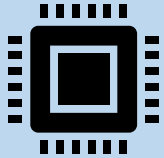
Latest reports from GDB (June 2024 @ CERN): https://indico.cern.ch/event/1356135/

It's time for VOs to start sending ARM jobs our way … we have over 4k ARM cores !

in-House Testing

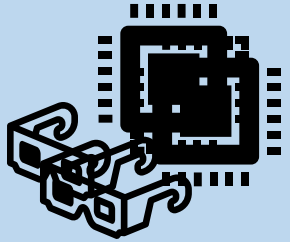
AMD96ht: Single AMD EPYC 7003 48-Core Processor (GIGABYTE)

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



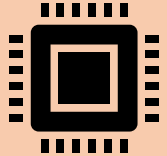
2xAMD48ht+GPU: Dual Socket AMD EPYC 7443 24-Core Processor (DELL)

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



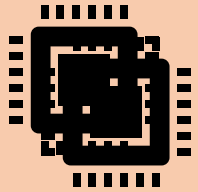
ARM80c: Single socket Ampere Altra Q80-30 80-Core Processor (GIGABYTE)

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 144-Core Processor (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 128-Core Processor (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
OS: Rocky 9.2

SuperMicro

AMD128ht: Single Socket AMD EPYC 8534P 64-Core Processor (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
OS: Rocky 8.8

SuperMicro

~~**ARM128c: Single Socket Ampere Altra Max M128-28 128-Core Processor (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~~
~~OS: Rocky 8.8~~

XMA

Coming soon : **AmpereOne** (96 - 192 cores)

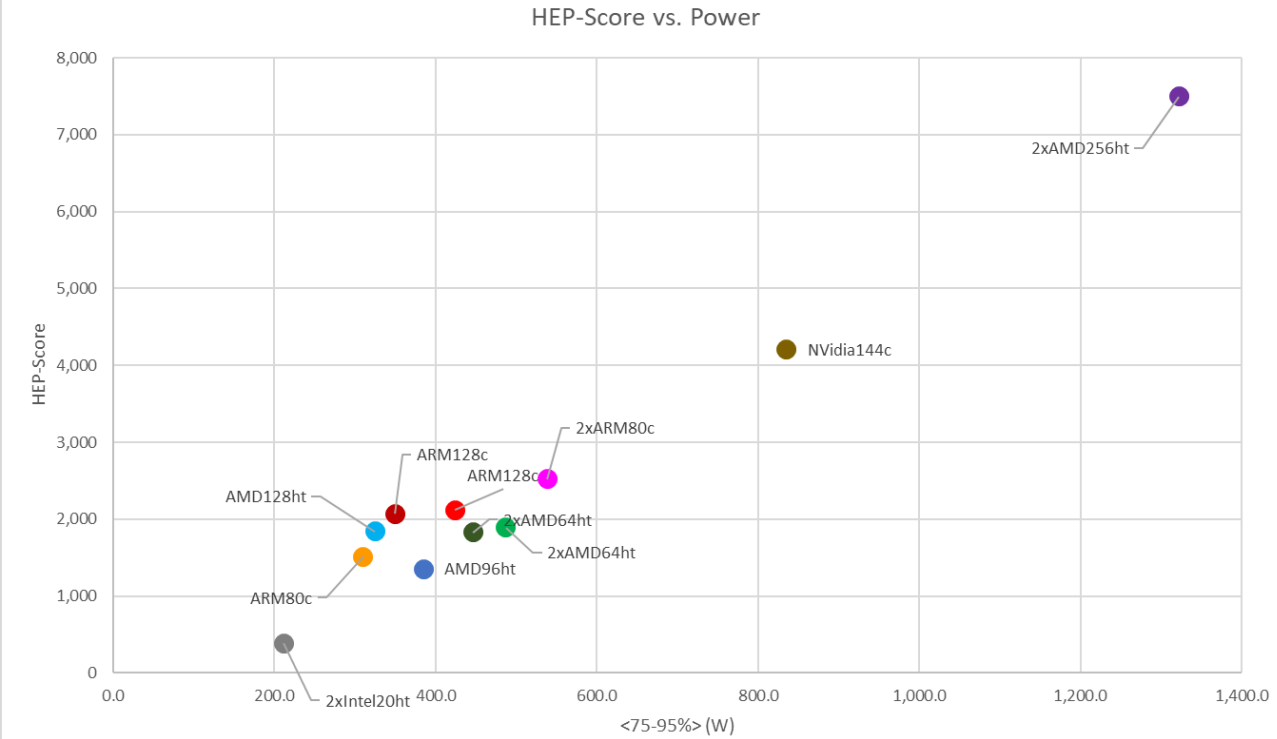
SuperMicro

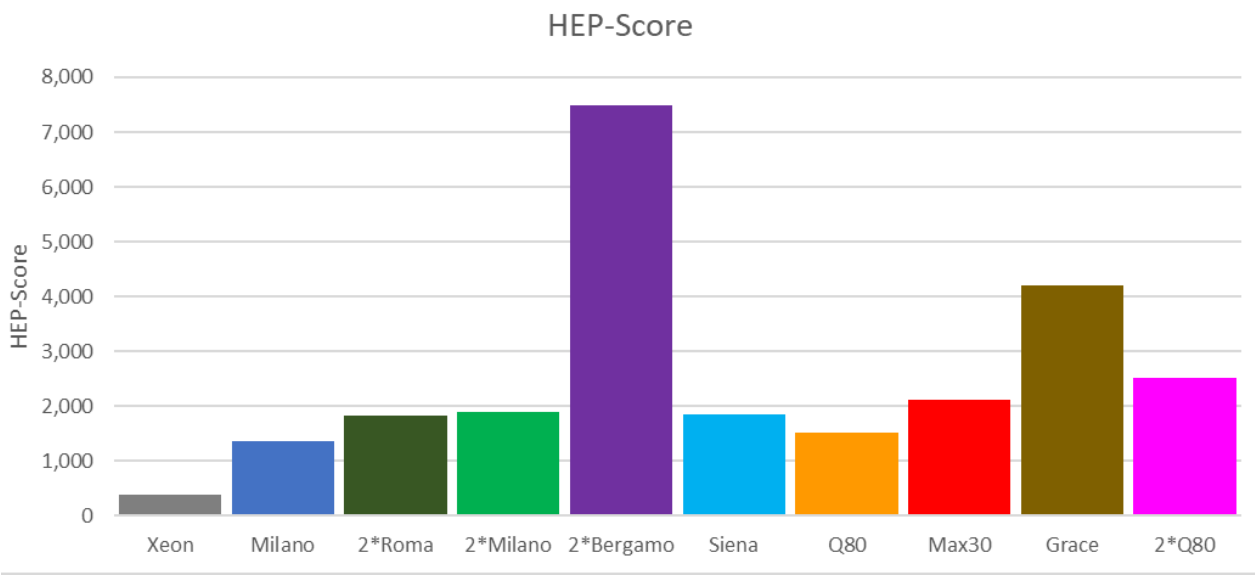
… we expect to get remote access to a test box next month!

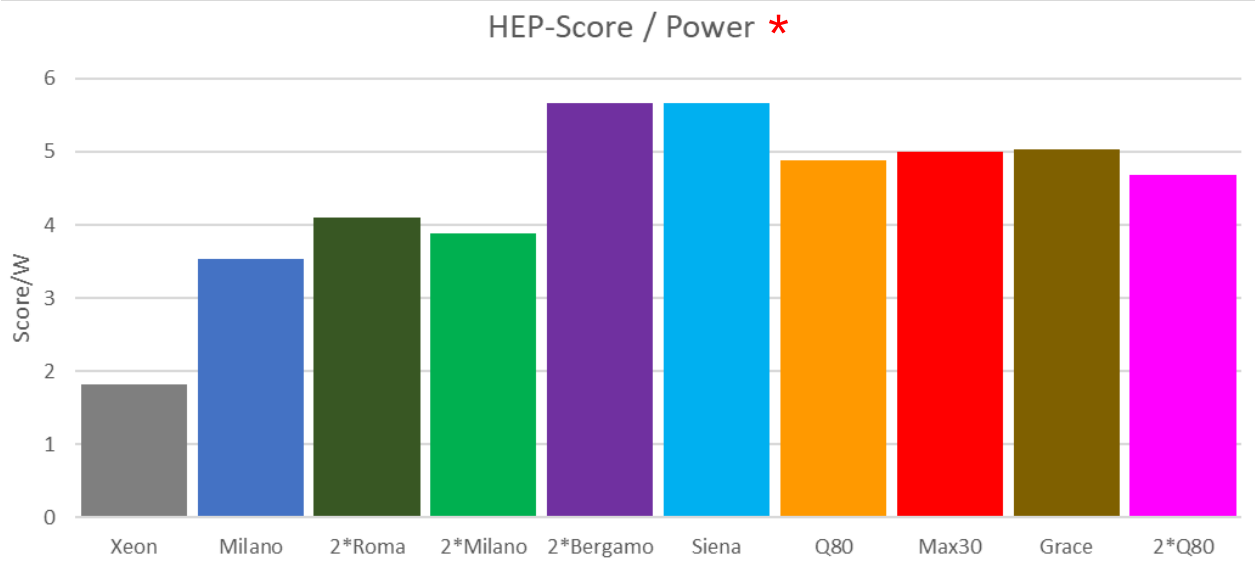
HEPscore/Watt

Using the new **Figure of Merit** (<75-95%> quantile average) as best proxy for Power, we estimate the Performance per Watt as **HEP-Score/Power FoM**.* (Note: replaced Altra Max **M128-28** with **M128-30**).

Nickname	Machine	CPU	HT	Frq Gov	Freq (GHz)	Arch	Threads
Xeon	2xIntel20ht	2 * Intel XEON 10-Core CPU E5-2630 v4	HT	conservative	2.2	2*x86_64	40
Milano	AMD96ht	AMD EPYC 7643 48-Core Processor (HT)	HT	conservative	2.3	x86_64	96
*2*Roma*	2xAMD64ht	2 * AMD EPYC 7452 32-Core Processor	HT	conservative	2.35	2*x86_64	128
*2*Milano*	2xAMD64ht	2 * AMD EPYC 7513 32-Core Processor	HT	conservative	2.6	2*x86_64	128
*2*Bergamo*	2xAMD256ht	2 * AMD EPYC 9754 128-Core Processor	HT	conservative	3.1	2*x86_64	512
Siena	AMD128ht	AMD EPYC 8534P 64-Core Processor (HT)	HT	conservative	3.1	x86_64	128
Q80	ARM80c	Ampere Altra Q80-30	//	conservative	3	aarch64	80
~~*Max28*~~	~~ARM128c~~	~~Ampere Altra Max M128-28~~	~~//~~	~~conservative~~	~~2.8~~	~~aarch64~~	~~128~~
Max30	ARM128c	Ampere Altra Max M128-30	//	conservative	3	aarch64	128
Grace	NVidia144c	NVidia Grace 144-Core 480GB DDR5	//	conservative	3.4	2*aarch64	144
*2*Q80*	2xARM80c	2 * Ampere Altra Q80-30	//	conservative	3	2*aarch64	160

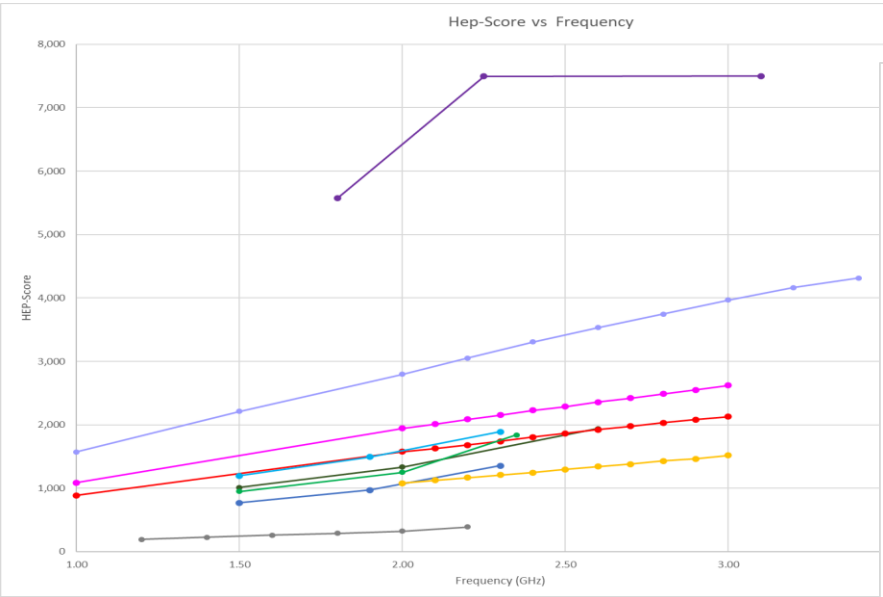


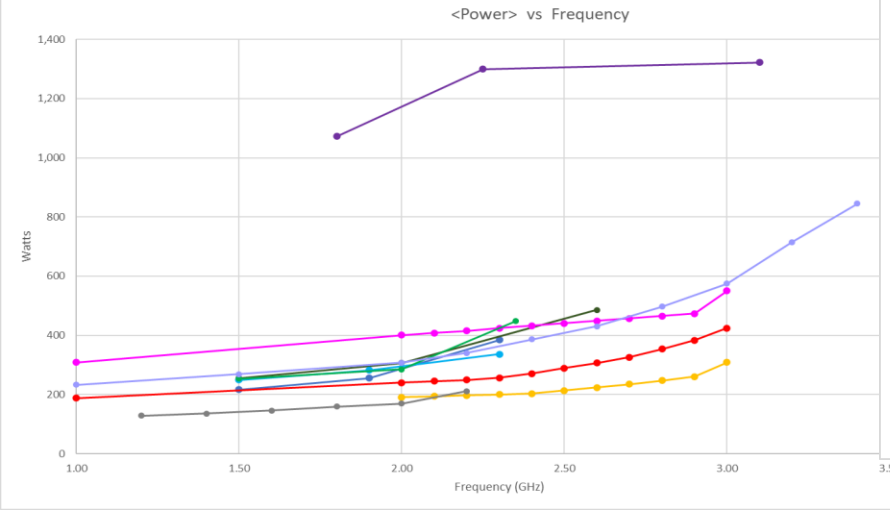


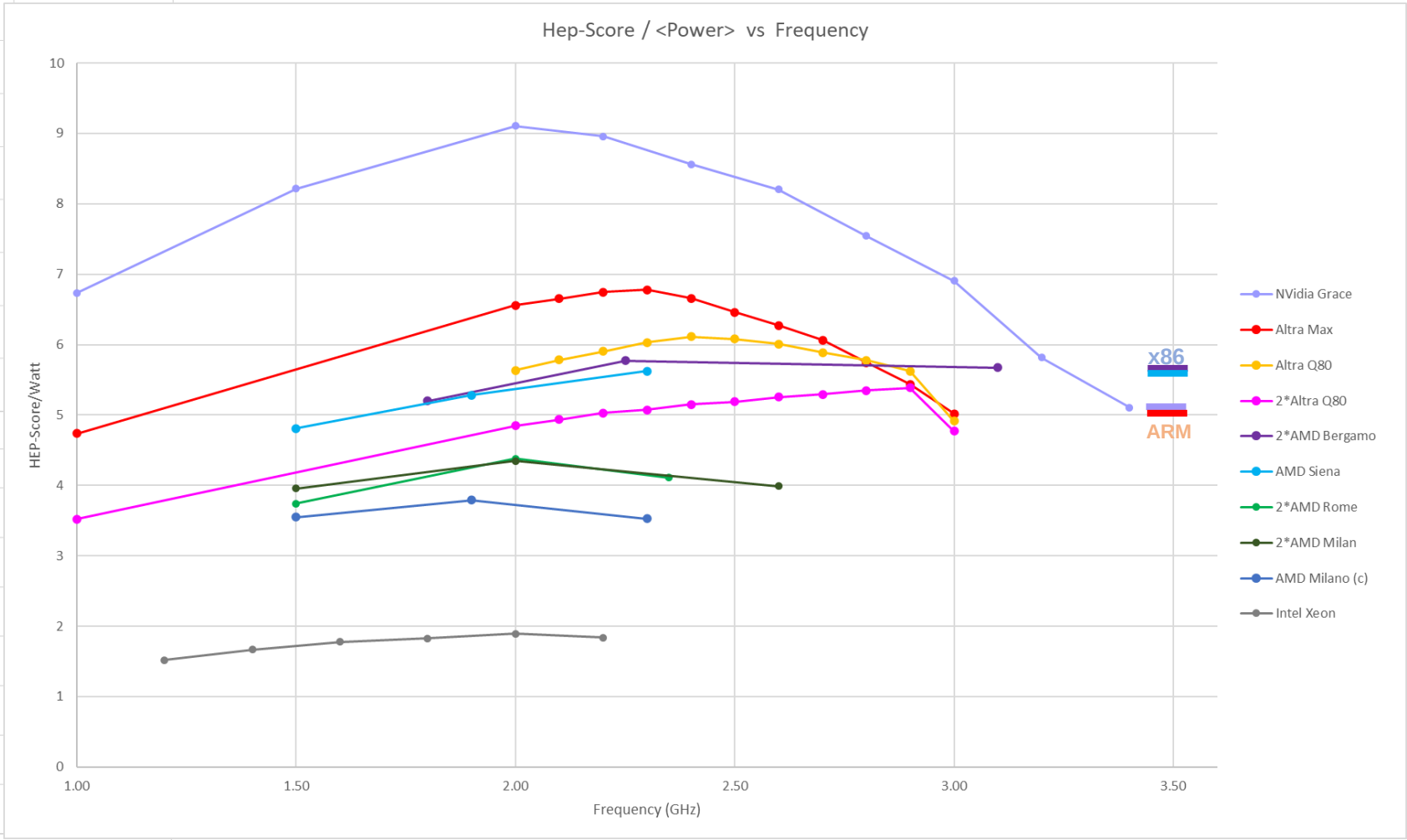


Frequency Scan (update)

HEP-Score/Watt vs. **CPU Frequency** gives a better picture of hardware potentials and shows optimal performances per watt at mid frequency range. Plots have been updated with the most recent Power **F.o.M.** and a scan of the whole Altra Max frequency range, thanks to the newly acquired **M128-30** nodes.

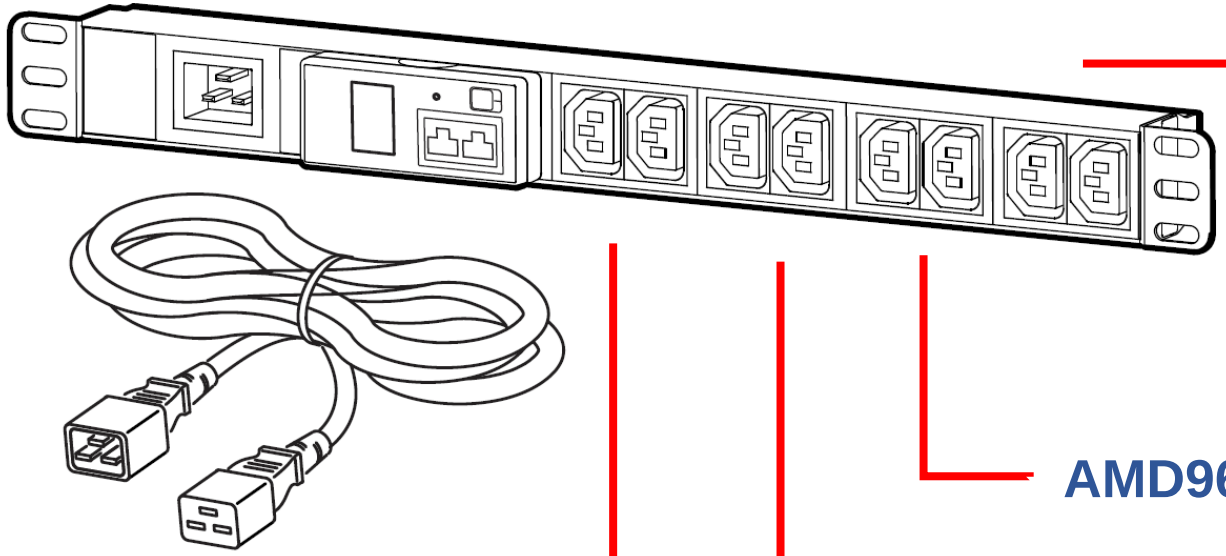






IPMI Validation

We acquired a **metered PDU** with remote readings and we are validating our IPMI power readings against the PDU. This is our current configuration:



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 Strategy

The PDU provides a single reading, so we measured the baseline with all machines off:

date_yyyymmdd	time_hhmmss	pdu_voltage_volt	pdu_current_amp	pdu_power_facto	pdu_energy_kwh	pdu_frequency_h	pdu_power_watts
15/08/2024	22:43:21	229.8	0.9	0.323	82.705	50.077	66
15/08/2024	22:43:26	229.8	0.9	0.323	82.705	50.071	67
15/08/2024	22:43:31	229.8	0.9	0.324	82.705	50.06	67
15/08/2024	22:43:36	229.8	0.9	0.325	82.705	50.055	67
15/08/2024	22:43:41	229.8	0.9	0.325	82.705	50.06	67
…							
15/08/2024	23:20:07	229.7	0.89	0.325	82.749	49.96	67
15/08/2024	23:20:12	229.7	0.9	0.324	82.749	49.965	67
15/08/2024	23:20:17	229.7	0.9	0.324	82.749	49.971	67
15/08/2024	23:20:22	229.7	0.9	0.322	82.749	49.971	67
15/08/2024	23:20:27	229.7	0.9	0.324	82.749	49.982	67

Base consumption is extremely stable at **67 Watts**

When off, the **AMD**, **ARM** and **Grace** test boxes each draw about **16 W** for IPMI/OOB services, **RISC-V** next to nothing, as it's a desktop PC with no BMC.

For testing, we turned on one machine at each time, and collected Power readings from:

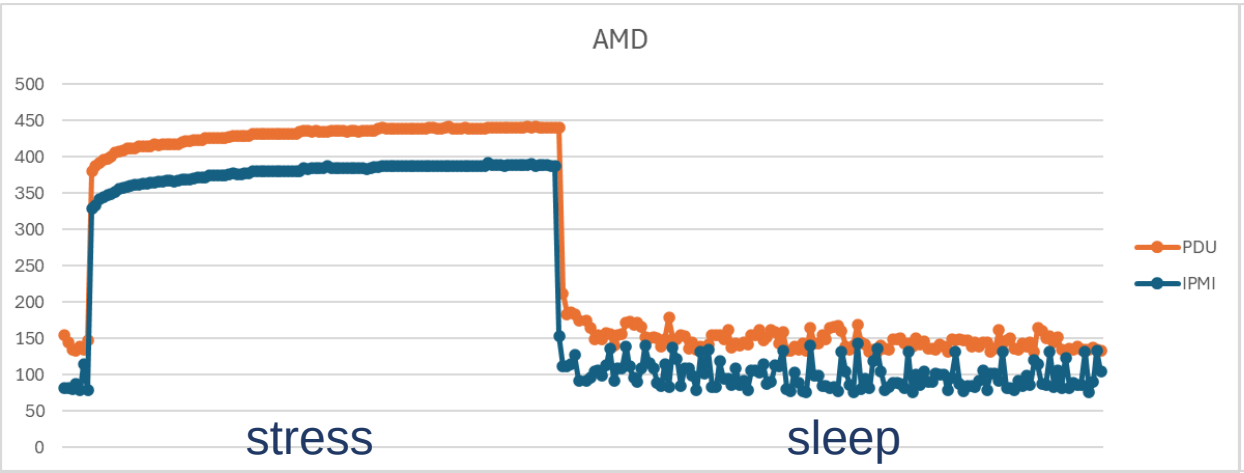
- **IPMI** (via **ipmitool** "Instantaneous Power") ;
- **PDU** (via **ModBus** "Active power"), with sampling interval **5 sec** for both.

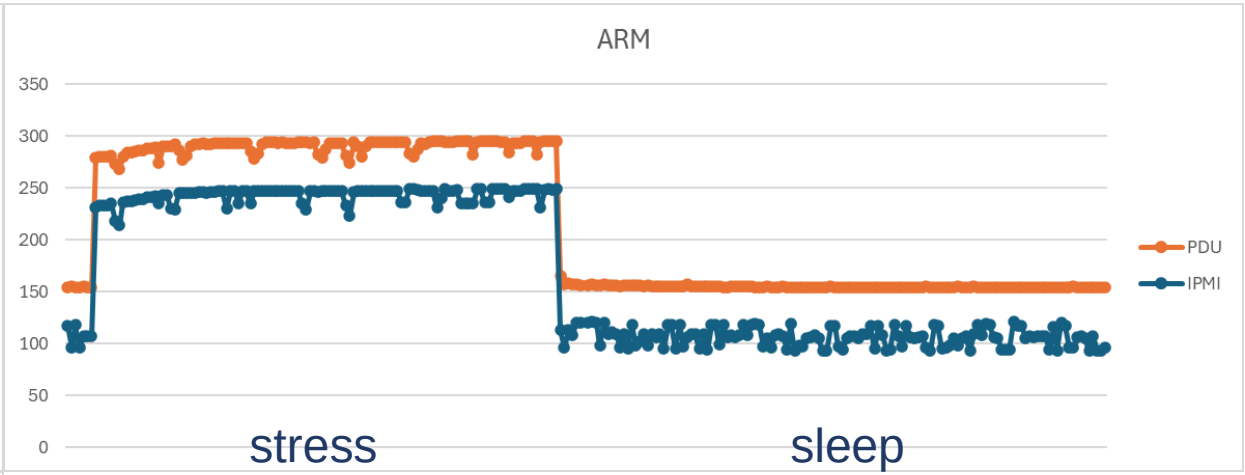
We ran 3 tests: **stress** (10 min.) , **sleep** (10 min.) , **HEP-Score** (3-6h) …

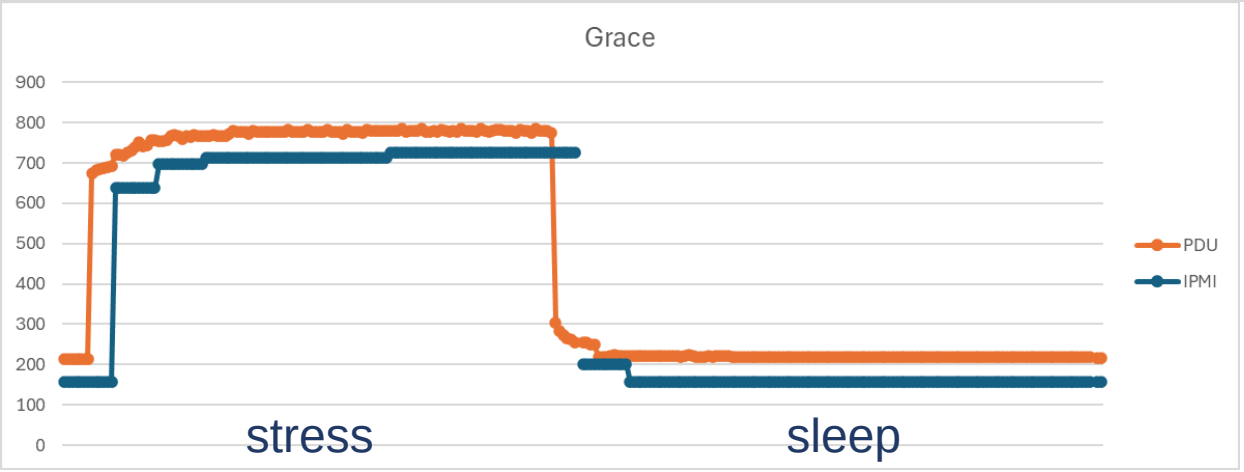
This strategy was motivated by minimizing the presence in the data center … we also tried a different approach, by connecting a single machine per time to the PDU and measuring power usage during **stress** and **sleep**. Results are still being interpreted.

Validation Results (preliminary)

Here a first comparison of 3 machines running **stress** and a **sleep** job, one after the other:







Work in Progress

We'd expect difference (Δ=PDU-IPMI) to have small oscillation around the 67 Watts baseline … but it is not really the case.

Known issues:

- there seem to be a lot more oscillations in idle (**sleep**) than at full load (**stress**), probably due to lower efficiency of the PSUs at lower output levels, and variation in background load having a relatively larger impact;
- the export scripts tend to skip beats or introduces delays, making the two time-series asynchronous;
- PDU readings show a longer tail (?) …

RISC-V testing

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).

Progress:

- We managed to compile and install **ROOT**, **CVMFS** and **XRootD** by building locally from source:
  - **ROOT**: https://github.com/hahnjo/root.git (RISC-V ported version)
  - **CVMFS**: https://github.com/cvmfs/cvmfs.git (original Git)
  - **XRootD**: https://github.com/xrootd/xrootd (original Git)
- Tommaso (**INFN**) managed to compile and run **Geant4**, also by building from source:
  - **Geant4**: https://gitlab.cern.ch/geant4/geant4 (original Git)
- Muzafar (**CMS**) made some progress in porting the **CMSSW** framework to RISC-V:
  most code can be ported, major issues are PyTorch & TensorFlow compatibility.

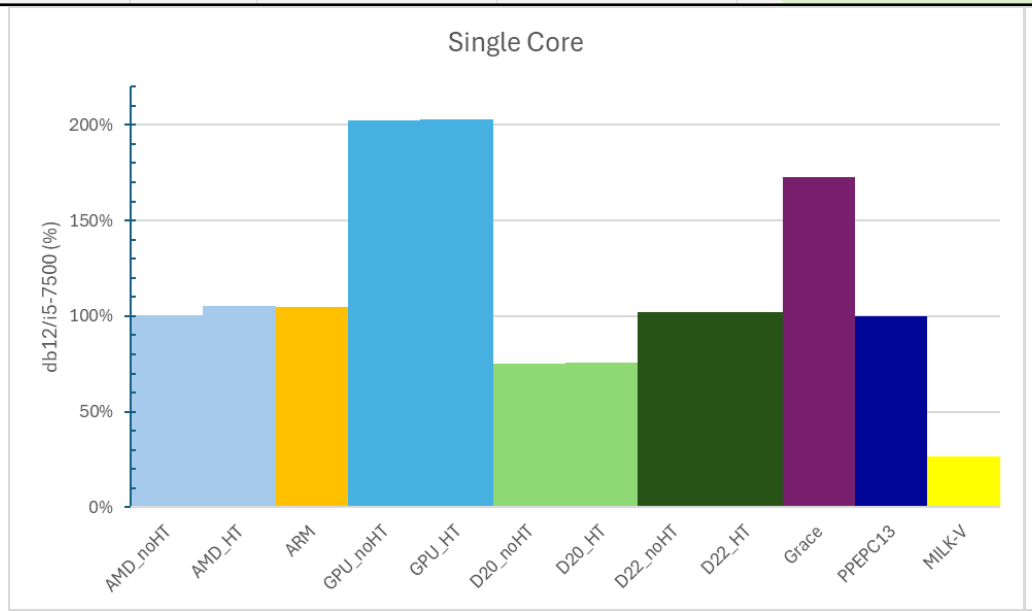
RISC Results (preliminary)

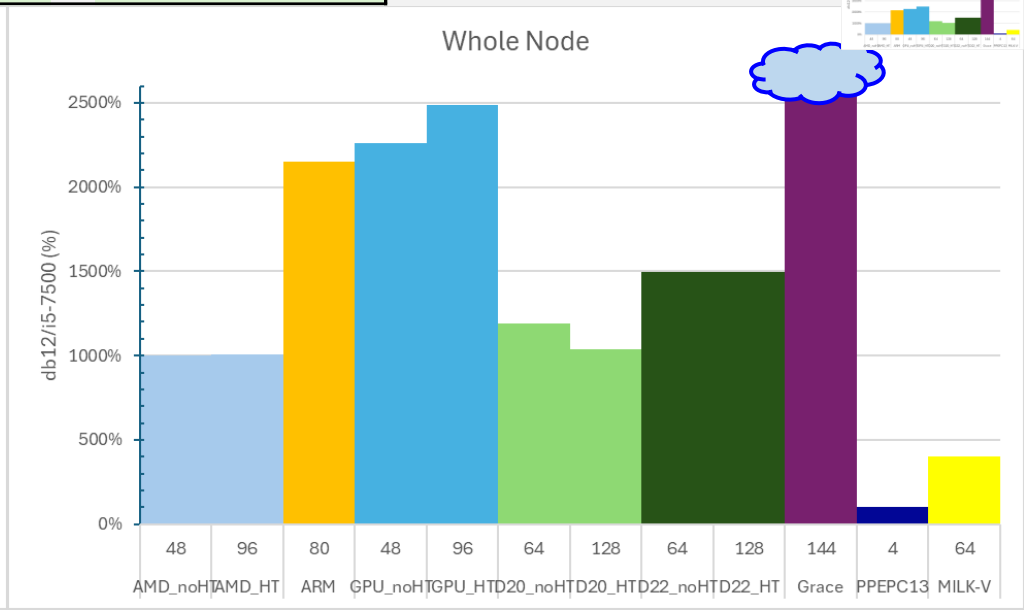
We could run **ROOT tests**, **ROOT benchmarks**, and **DB12** on RISC and few other architectures.
Results are a bit tricky to compare, because not all machines can run all benchmarks …

			single	whole	single_core		whole_node	
machine	CPU	nproc	db12_single	db12_wholenode	/i5-7500	/AMD_noHT	/i5-7500	/AMD_noHT
AMD_noHT	AMD EPYC 7643 48-Core Processor	48	23.85752688	938.0106703	100%	100%	1003%	100%
AMD_HT	//	96	24.96483826	939.6929573	105%	105%	1005%	100%
ARM	Ampere Altra Q80-30 Processor	80	24.90039841	2010.311923	105%	104%	2151%	214%
GPU_noHT	2 * AMD EPYC 7443 24-Core Processor	48	48.07692308	2115.670718	202%	202%	2263%	226%
GPU_HT	//	96	48.16955684	2323.572578	203%	202%	2486%	248%
D20_noHT	2 * AMD EPYC 7452 32-Core Processor	64	17.91120081	1111.082808	75%	75%	1189%	118%
D20_HT	//	128	18.03861789	970.5602849	76%	76%	1038%	103%
D22_noHT	2 * AMD EPYC 7513 32-Core Processor	64	24.31506849	1401.588433	102%	102%	1499%	149%
D22_HT	//	128	24.28180575	1396.361662	102%	102%	1494%	149%
Grace	2 * NVidia Grace 72-Core	144	40.98360656	5903.329871	173%	172%	6315%	629%
PPEPC13	Intel Core i5-7500 CPU @ 3.40GHz	4	23.75690608	93.47930739	100%	100%	100%	10%
MILK-V	Milk-V riscv64	64	6.311537491	375.5319286	27%	26%	402%	40%

Work in Progress

So far, **DB12** is the only benchmark that could run on all hardware !





Upcoming Conferences

❖ Talks accepted at **CHEP** (Krakow, October 2024)

Simulating the Carbon Cost of Grid Sites
Main Authors: David Britton; Steve Lloyd

Comparative efficiency of HEP codes across languages and architectures
Main Author: Samuel Cadellin Skipsey

Heterogeneous Computing and Power Efficiency in HEP
Main Author: Emanuele Simili

Taking on RISC for Energy-Efficient Computing in HEP
Main Authors: Emanuele Simili; Tommaso Boccali; Shazad Muzafar

On-Grid GPU development via interactive HTCondor jobs and Analysis Facility style workflows
Main Author: Albert Gyorgy Borbely

❖ Contributions at the **HTCondor** Workshop (NIKHEF, September 2024)

(titles to be defined)
Main Authors: Emanuele Simili & Albert Borbely

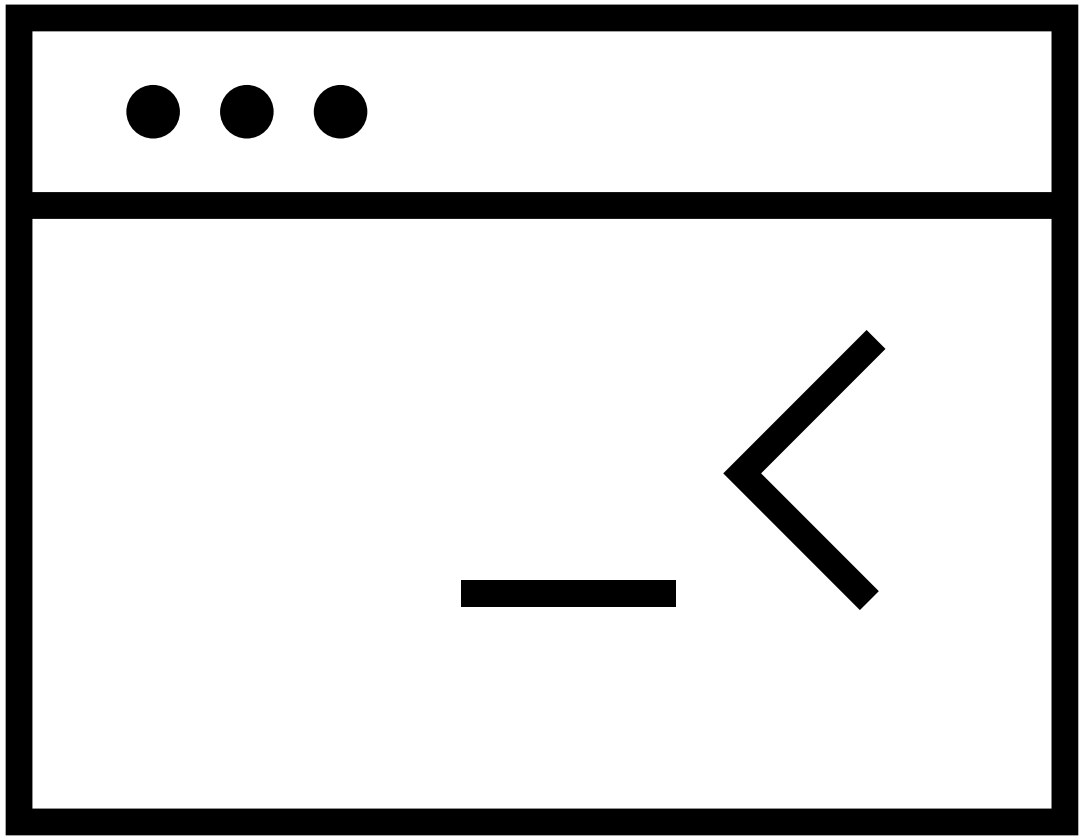
Outlook

- Improve on the methodology and develop an Analysis framework:
  - energy measurement is now integrated in HEP-Score
  - HEP-Score analysis package (HEPiX summer student is studying aggregation metrics, e.g., k-mean: https://indico.cern.ch/event/1433496/)

- Follow up on **RISC-V** updates and integration:
  - Actively participate in the testing and benchmarking cycle
  - Look into new hardware solutions (RISC servers?)

- Keep exploring new hardware:
  - benchmark **GPU**+**CPU** using Celeritas (Bruno)
  - test the newly released **AmpereOne** …
  - what's next?

end



- Major work undertaken at your site since last GridPP meeting (April to July 24)
    - Added 2,304 Ampere Altra Max M128-30 cores
    - Migrated to EL9 and HTCondor 10
    - set-up a heterogeneous endpoint with dual queue (ARM & x86)

2 - Key problems faced and resolved during this time.
    - dealing with a dual queue is tricky, and not all VOs can cope with it

3 - Pending issues and plans for resolving the existing issues.
    - Upgrades to ARC 7 and latest version of HTCondor

4 - Projection of resource availability for next year, will there be any change in the capacity and how much
    - No significant change expected

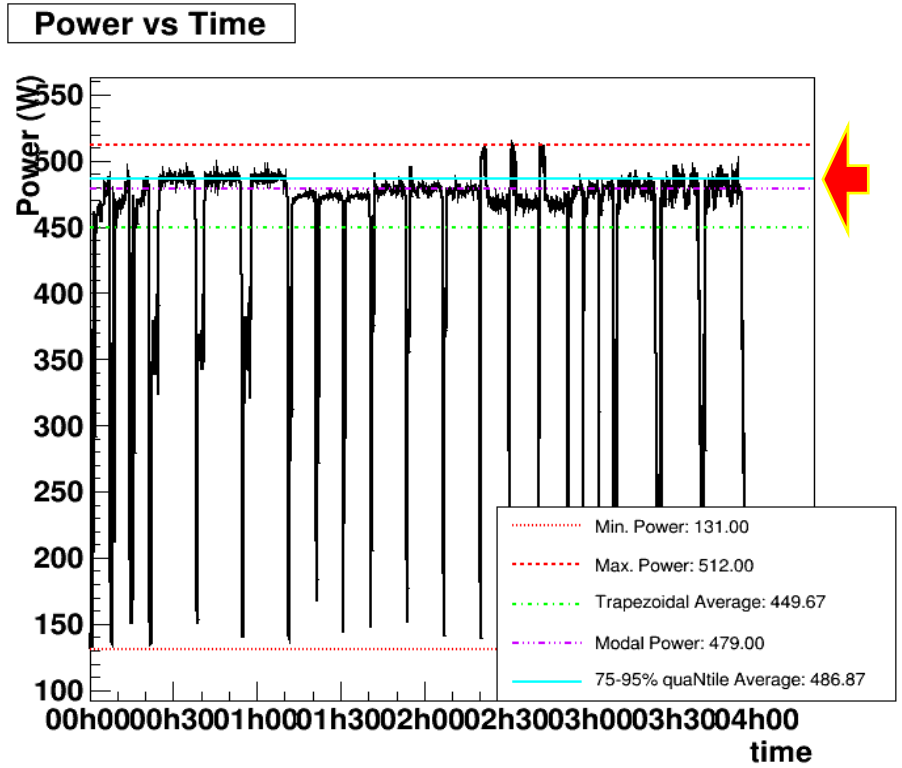
What Watt

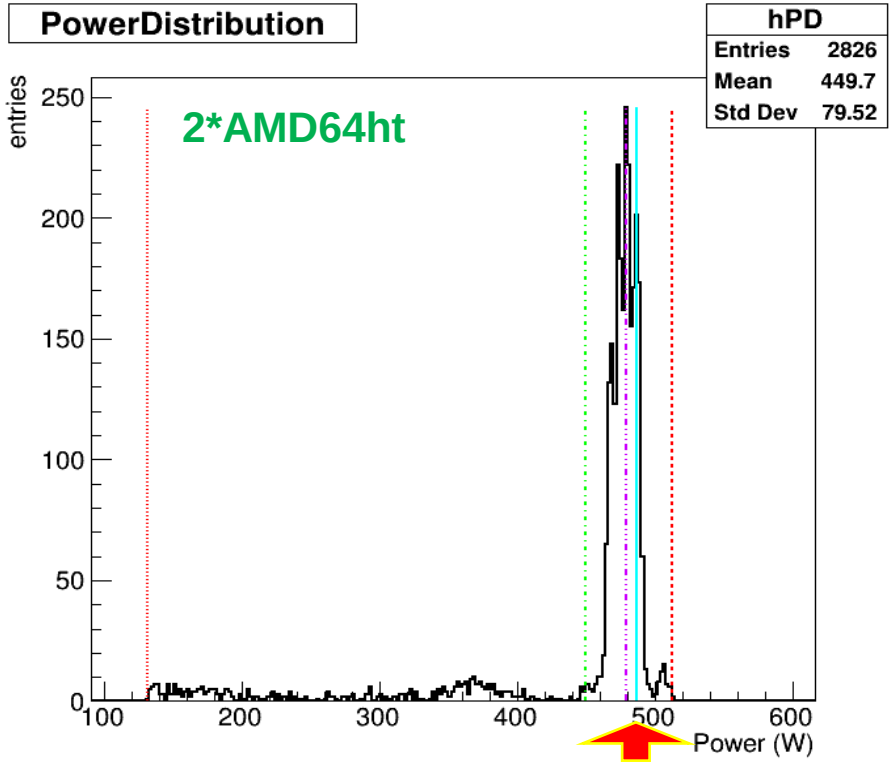
New **Figure of Merit** (**FoM**), i.e., the best proxy of power usage for standard HEP workloads:

FoM should be easy to implement, we could fit this peak, but there are other ways of doing it.

Arrange the data in power order and perform an upper quaRtile average, but removing the top 5% of data

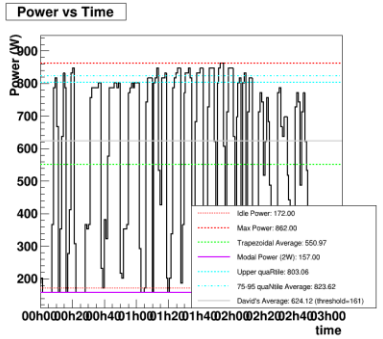
75-95% quaNtile average*

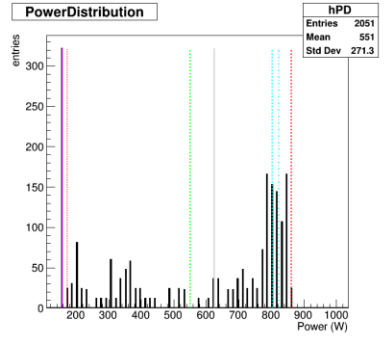




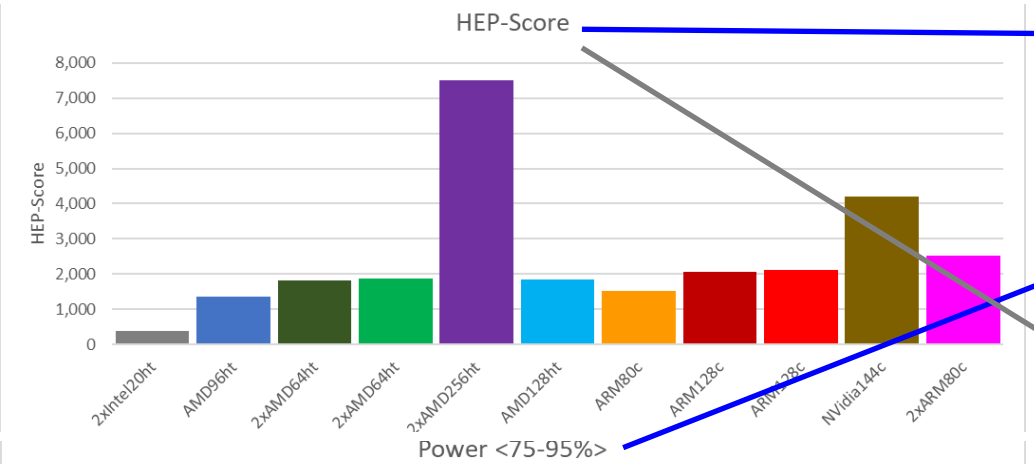
<75-95%> Blue line sits nicely in the plateau

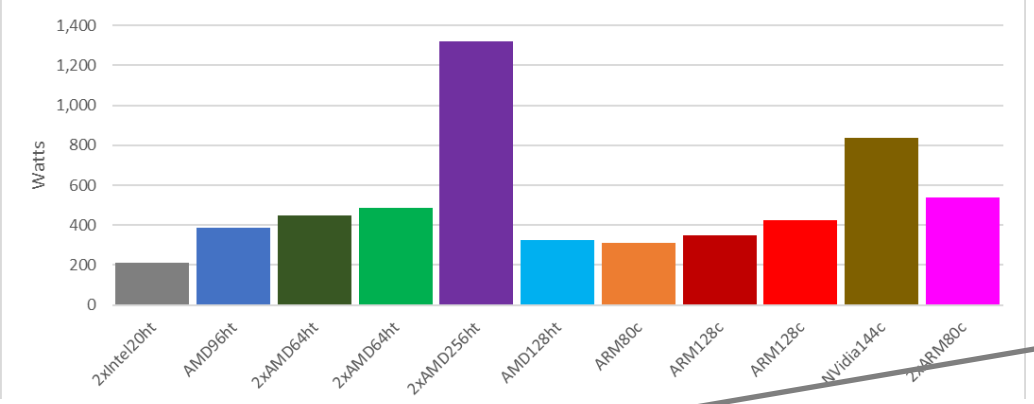
Modal power also sits in the plateau – but we found edge cases where this breaks (e.g., Grace has a very steady idle state)

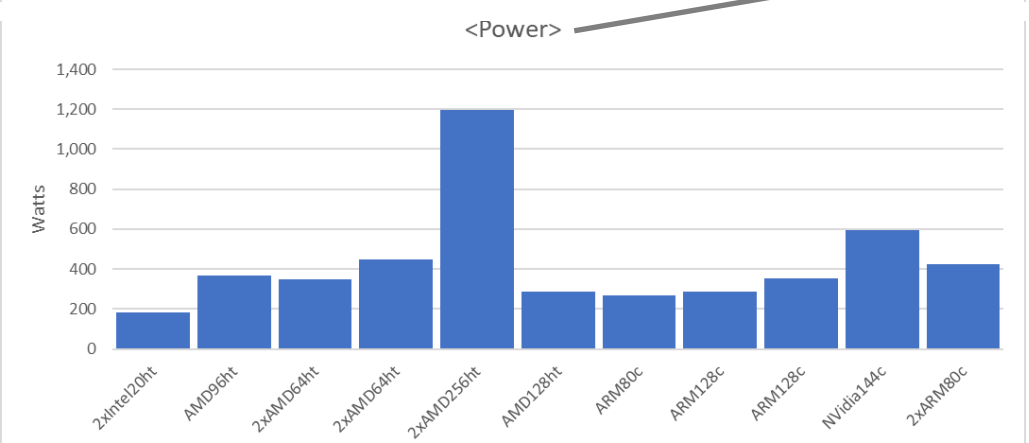


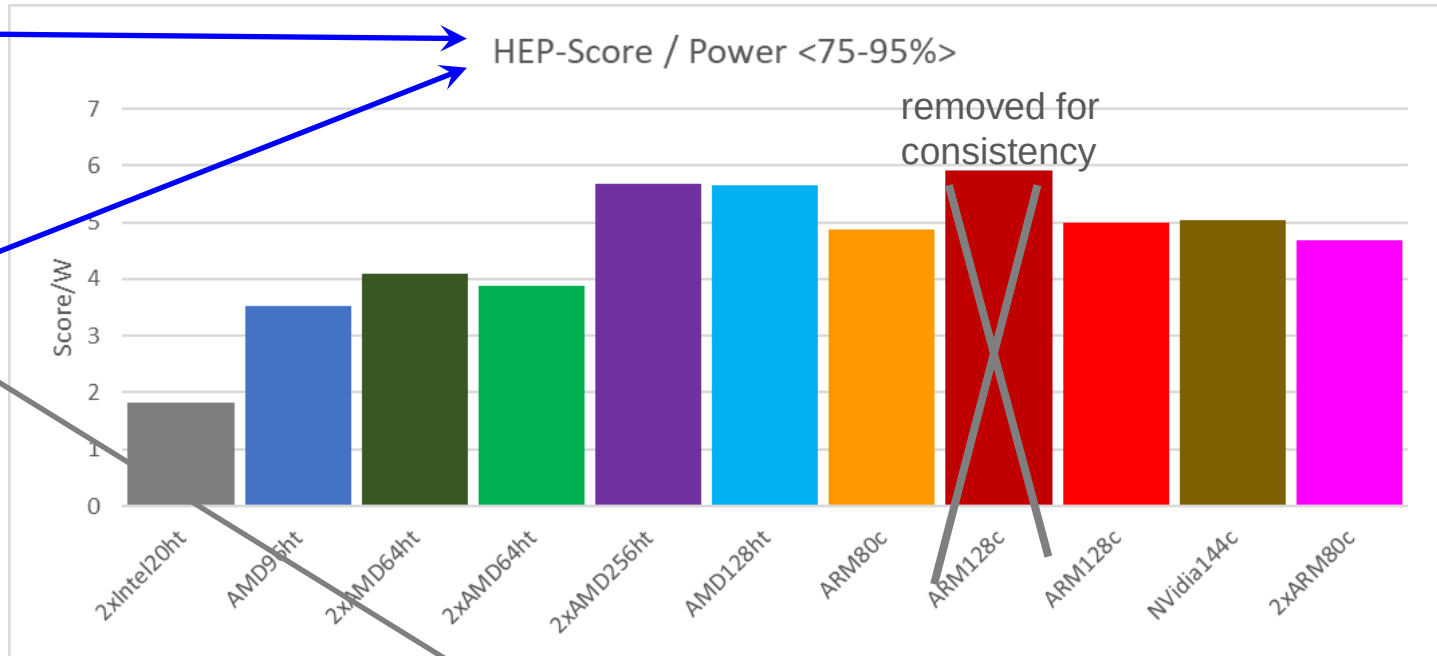


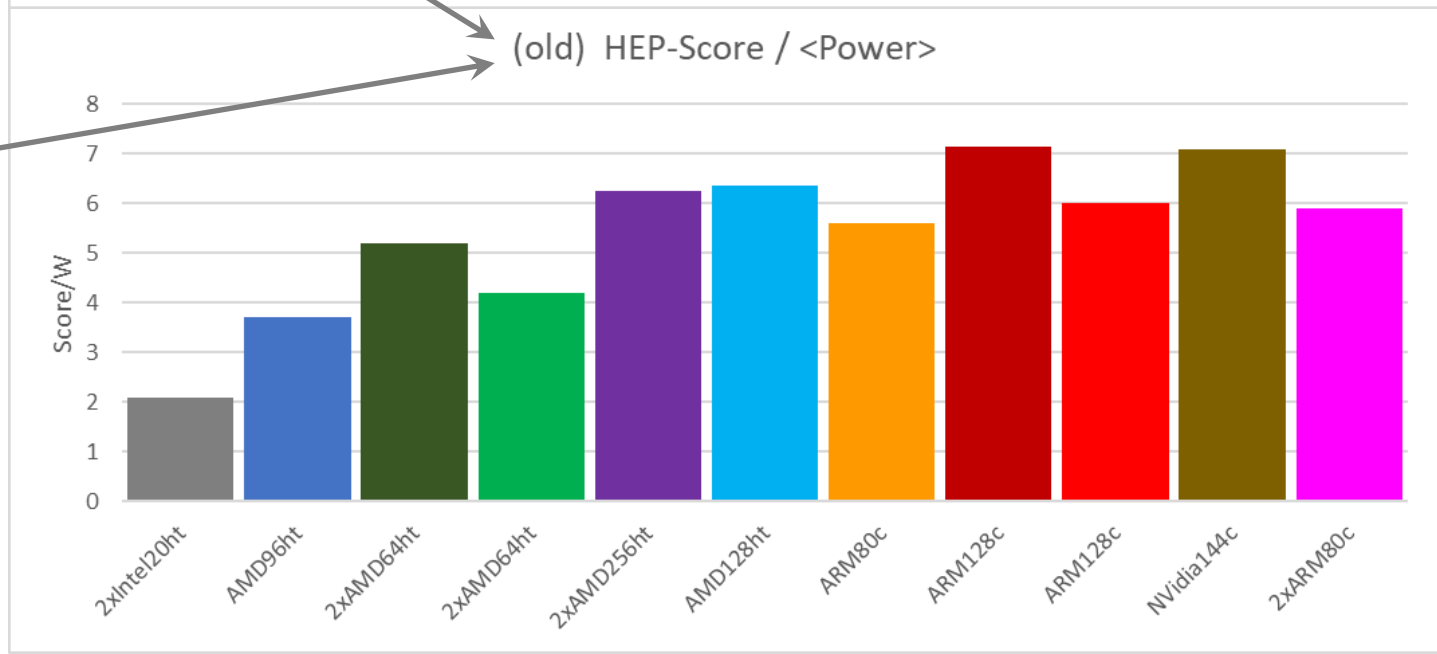
HEPscore/Watt (comparing old & new)







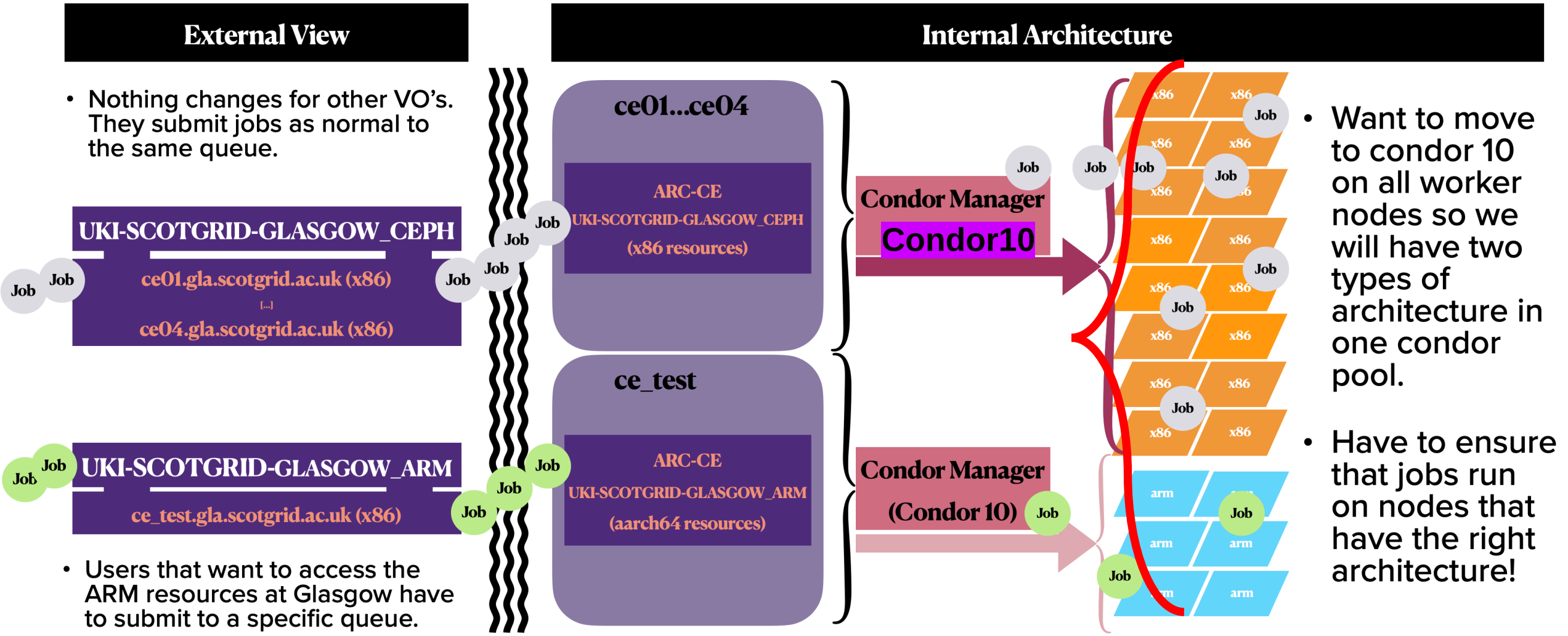




ARM + x86 Farm @ Glasgow

Job submission chain @ ScotGrid Glasgow:

arm ~ 4000 arm cores x86 ~17000 x86 cores



Results^ (preliminary)

We could run **ROOT tests**, **ROOT benchmarks**, and **DB12** … results are a bit tricky to compare, because not all machines can run all benchmarks …

			single	whole	single_core		whole_node		HEPscore		ROOT benchmark	
machine	nproc	CPU	db12_single	db12_wholenode	/i5-7500	/AMD_noHT	/i5-7500	/AMD_noHT		/AMD_noHT		/AMD_noH
AMD_noHT	48	AMD EPYC 7643 48-Core Processor	23.85752688	938.0106703	100%	100%	1003%	100%	1,169.49	100%	5,853.11	100%
AMD_HT	96	//	24.96483826	939.6929573	105%	105%	1005%	100%	1,341.37	115%	5,703.80	97%
ARM	80	Ampere Altra Q80-30 Processor	24.90039841	2010.311923	105%	104%	2151%	214%	1,467.27	125%	4,430.31	76%
GPU_noHT	48	2 * AMD EPYC 7443 24-Core Processor	48.07692308	2115.670718	202%	202%	2263%	226%		0%		0%
GPU_HT	96	//	48.16955684	2323.572578	203%	202%	2486%	248%		0%		0%
D20_noHT	64	2 * AMD EPYC 7452 32-Core Processor	17.91120081	1111.082808	75%	75%	1189%	118%	1,479.85	127%		0%
D20_HT	128	//	18.03861789	970.5602849	76%	76%	1038%	103%	1,826.35	156%		0%
D22_noHT	64	2 * AMD EPYC 7513 32-Core Processor	24.31506849	1401.588433	102%	102%	1499%	149%	1,682.17	144%		0%
D22_HT	128	//	24.28180575	1396.361662	102%	102%	1494%	149%	1,928.05	165%		0%
Grace	144	2 * NVidia Grace 72-Core	40.98360656	5903.329871	173%	172%	6315%	629%	4,205.43	360%		0%
PPEPC13	4	Intel Core i5-7500 CPU @ 3.4GHz	23.75690608	93.47930739	100%	100%	100%	10%		0%	3,885.49	66%
MILK-V	64	Milk-V riscv64	6.311537491	375.5319286	27%	26%	402%	40%		0%	786.24	13%

So far, **DB12** is the only benchmark that can run on all hardware !