Heterogeneous Tier2 Cluster & Power Efficiency Studies @ ScotGrid Glasgow

Abstract

With the latest addition of 4k ARM cores, the ScotGrid Glasgow facility is a pioneering example of a heterogeneous WLCG Tier2 site. The new hardware has enabled large-scale testing by experiments and detailed investigations into ARM performance in a production environment.

I will present an overview of our computing cluster, which uses HTCondor as the batch system combined with ARC-CE as the front-end for job submission, authentication, and user mapping, with particular emphasis on the dual queue management. I will also touch on our monitoring and central logging system, built on Prometheus, Loki, and Grafana, and describe the custom scripts we use to extract job information from HTCondor and pass it to the node_exporter collector.

Moreover, I will highlight our research on power efficiency in HEP computing, showing the benchmarks and tools we use to measure and analyze power data. In particular, I will present a new figure-of-merit designed to characterize power usage during the execution of the HEP-Score benchmark, along with an updated performance-per-watt comparison extended to the latest x86 and ARM CPUs (Ampere Altra Q80 and M128, NVidia Grace, and recent AMD EPYC chips). Within this context, we introduce a Frequency Scan methodology to better characterize performance/watt trade-offs.

Outline

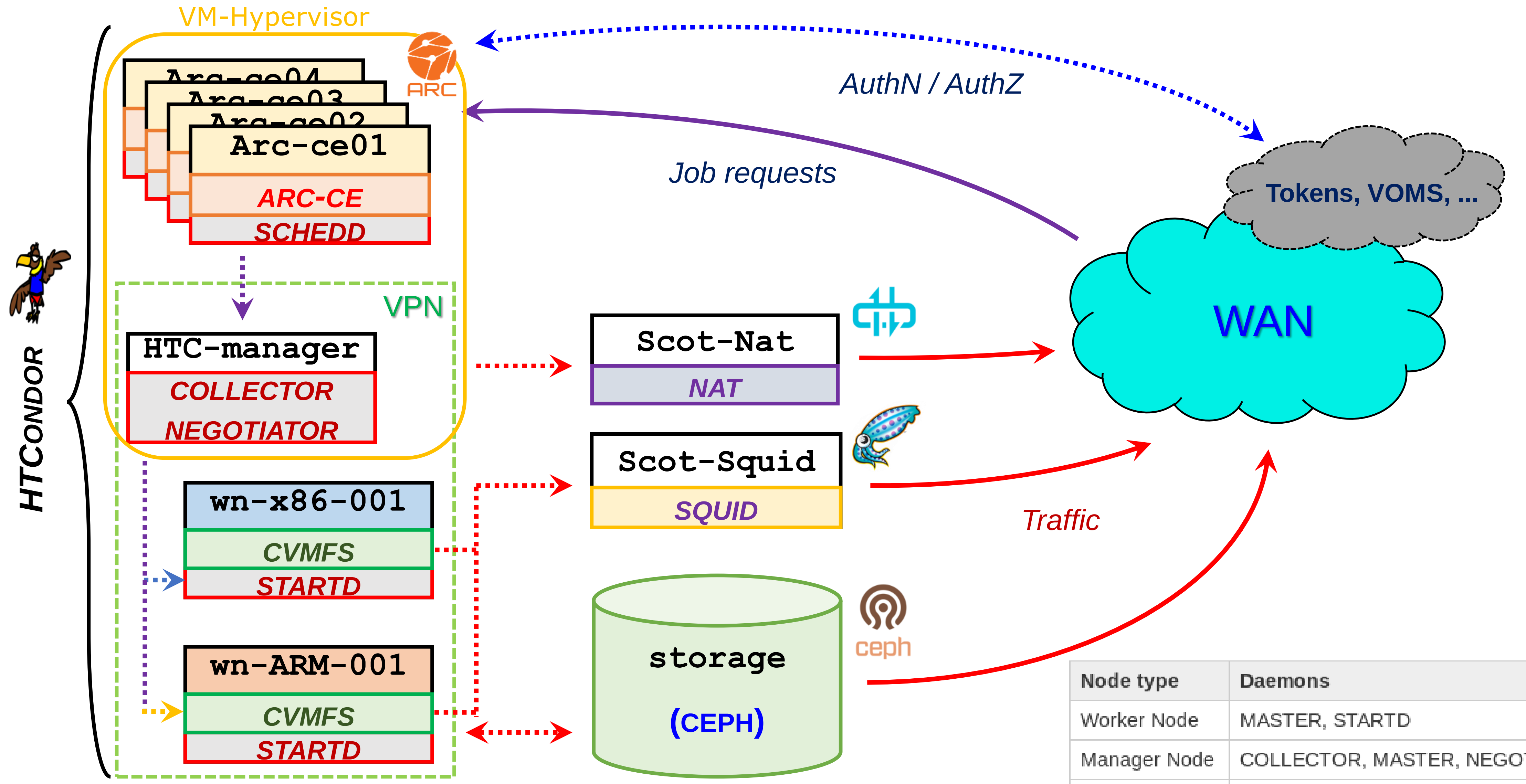
- ➢ Overview of **ScotGrid Glasgow**, a WLCG Tier2 cluster
  - **ARC-CE** + **HTCondor** configuration (**ARGUS**-less auth/)
  - dual queue management (**ARM** & **x86**) … & **GPU**

- ➢ Monitoring tools
  - **Loki** + **Prometheus** + **Grafana** set-up
  - **node_exporter** scripts for HTCondor

- ➢ Benchmarking and Power Measurement
  - motivations and methodology
  - visualization of **HEPscore/Watt** and **Frequency Scan** results

- ➢ Outlook + questions (& advices about cluster management)

ScotGrid Tier2 Cluster Overview



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

ARC-CE configuration

We are using NorduGrid **ARC-CE v6.20** (https://www.nordugrid.org/arc/arc6/), soon to update to **v7**.
The ARC-CE configuration is defined in: `/etc/arc.conf`

VOs AuthN / AuthZ:
Token and/or **VOMS**

pool-users mapping

Local Resource
Management System

ARC services

Coordinates & Settings
(incl. APEL accounting)

Queues definition

```
[common]
hostname = ce0x.gla.scotgrid.ac.uk

[authgroup: bla]
authtokens = xxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx https://iam.grid.hep.ph.ic.ac.uk/ * * *
voms = bla.org * * *

[mapping]
map_to_pool = bla     /etc/grid-security/pool/bla

[lrms]
lrms = condor condor

[arex]
...

[infosys]
...


[queues]
...
```

→ This used to be done by **ARGUS** … which was finally decommissioned

HTCondor configuration

We started using **HTCondor** in 2015. Both the ARC-CE and HTCondor configuration has been evolving throughout the years, following updates and WLCG requirements (i.e., usage of Tokens).

After the latest Cluster Update we use HTCondor **v10.0** (not-too-) soon to be updated to **v23** (/**v24**?)

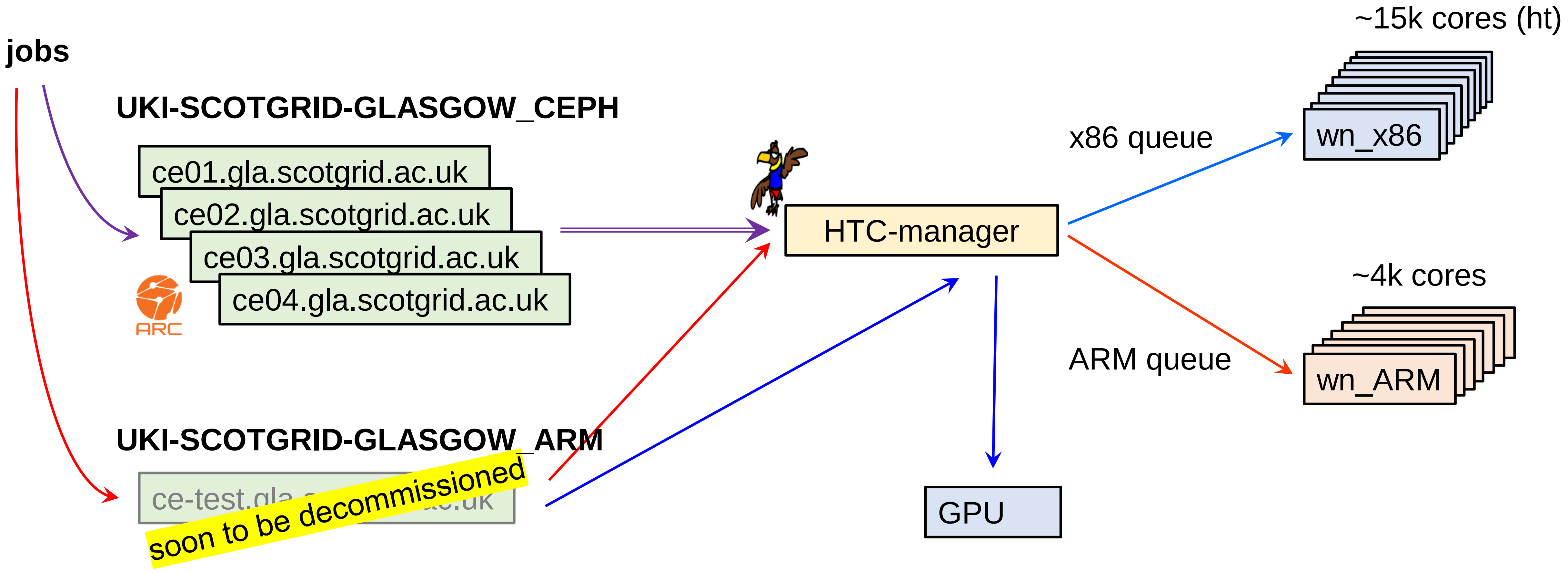
The configuration of HTCondor is organised in a number of files in: `/etc/condor/*.config`

```
- daemon_config:    load the daemons needed on this node (manager, compute, ce)
- security:         security profile and cluster coordinates
                    e.g.  ALLOW_ADMINISTRATOR enables root
                          SEC_DEFAULT_AUTHENTICATION_METHODS = FS, IDTOKENS
- network:           configures network
- mail:              configures notifications email
- credentials:       DELEGATE_JOB_GSI_CREDENTIALS_LIFETIME = 0   (see Ticket-ID: 146596)
- condor-disable:    supports the bespoke condor_enable and condor_disable commands
- wn:                makes a single slot on each machine, then Condor can chop it off
- accounting-groups: maps users to groups for internal accounting (accounting-map)
- status:            CONDOR_Q_ONLY_MY_JOBS = false
- fairshare:         uses the internal accounting to limit VOs resource usage
                     GROUP SURPLUS resources can be assigned to anyone (even above fair share limits)
```

Heterogeneous Compute Cluster

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

This is a simplified 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 Config

(*) … because **HTCondor** doesn't have a concept of queues: it matches jobs to resources based on their **ClassAd**, which includes an architecture entry, which is added by the ARC-CEs.

```
[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 mechanism also works for **GPU** queue (tested already!)

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 …

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

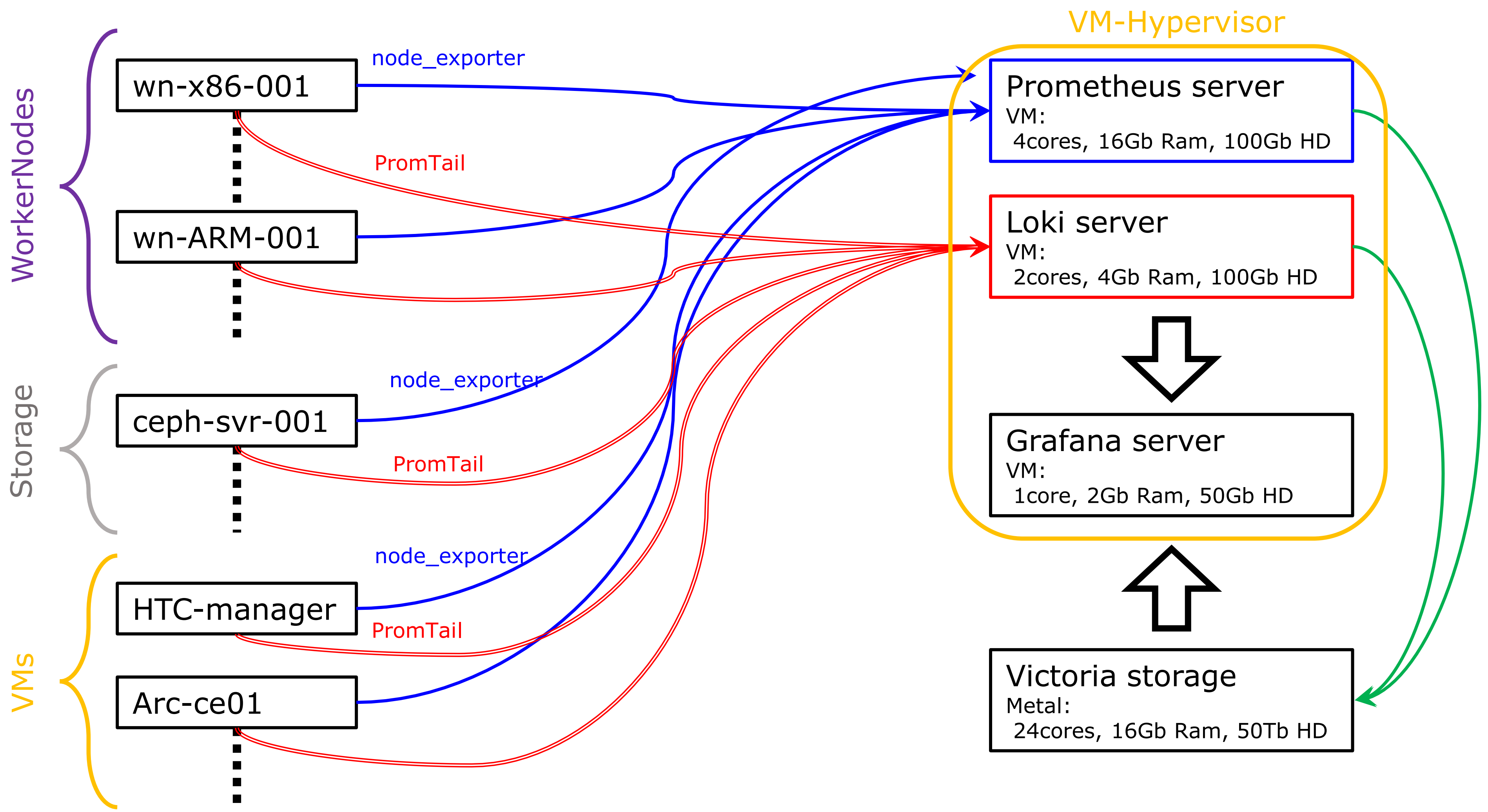
Cluster Monitoring & Logging

Metrics are exported by **node_exporter** and collected by **Prometheus** (VM).

Logs are exported by **PromTail** and collected by **Loki** (VM).

Grafana (VM) pulls data from both servers and provides the tools for querying and building colorful graphs and dashboards.

VictoriaMetrics (metal) archives the collected data into a large storage server. Archived data can be queried by Grafana.



HTCondor Monitoring

We use **node_exporter** to extract metrics for **Prometheus** (and then **Grafana**).
Beside the standard set of metrics, it can export custom ones.

Two **scripts** periodically query HTCondor for job information:

now in python !

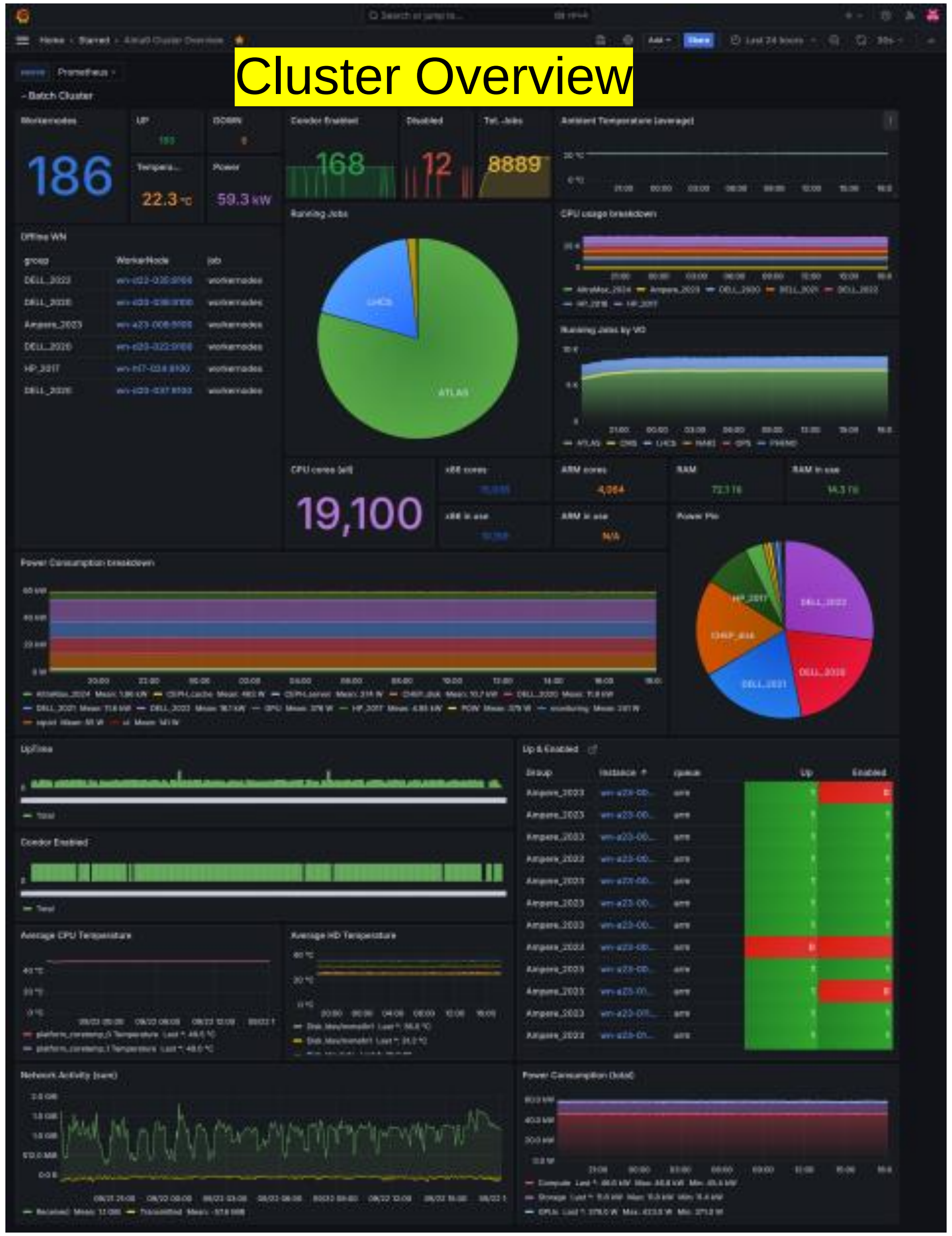
`ce_get_info.py` → runs on the ARC-CEs every 5 min.
`node_get_condorinfo.sh` → runs on all worker-nodes every 2 min.

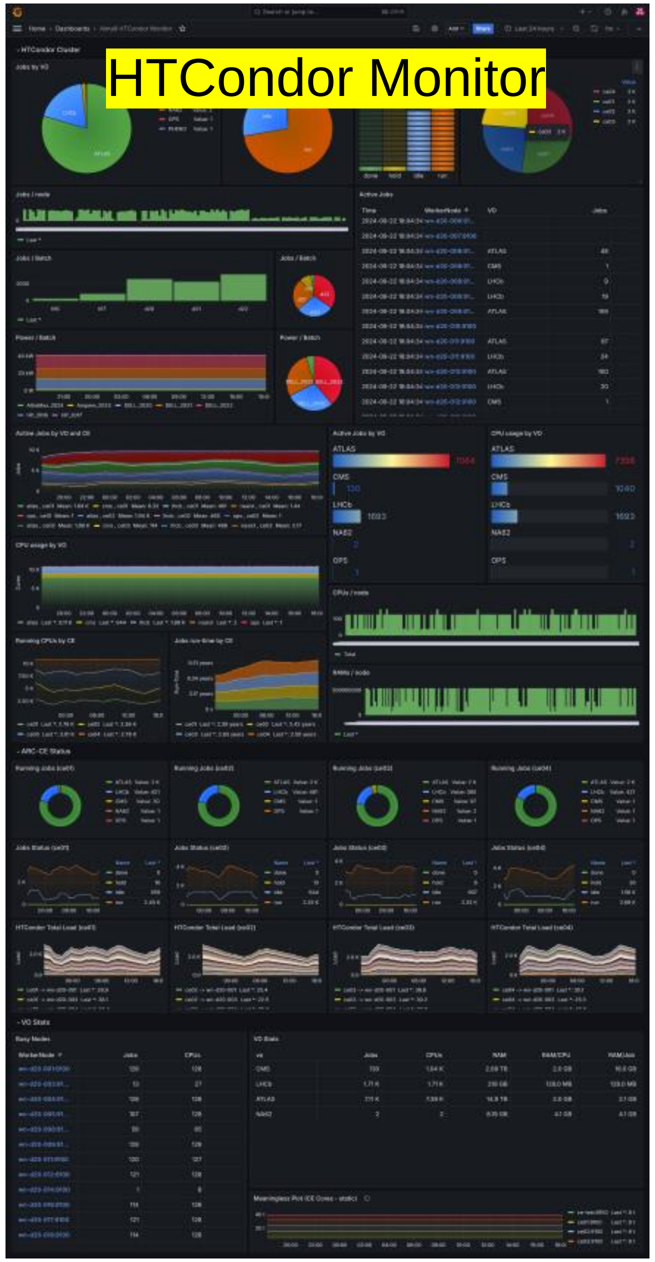
The scripts call **`condor_status -startd`** with **`-autoformat:t`** and parse the output for **node_exporter** to ingest (in `/var/lib/node_exporter/textfile_collector/`):

```
condor_node_info.prom  <
 node_condor_cpu{slot=$SLOTNUMBER,vo=$VO,ce=$CE} ${USECPU}
 node_condor_ram{slot=$SLOTNUMBER,vo=$VO,ce=$CE} ${USEMEM}
 node_condor_load{slot=$SLOTNUMBER,vo=$VO,ce=$CE} ${LOADAV}
 node_condor_runtime{slot=$SLOTNUMBER,vo=$VO,ce=$CE} ${RUNT
 ...
```

```
condor_ce_info.prom  <
 ce_condor_total_jobs {data["Jobs"]}
 ce_condor_queue{{state="done"}} {data["Completed"]}
 ce_condor_queue{{state="run"}} {data["Running"]}
 ce_condor_queue{{state="idle"}} {data["Idle"]}
 ce_condor_queue{{state="hold"}} {data["Held"]}
 ...
```

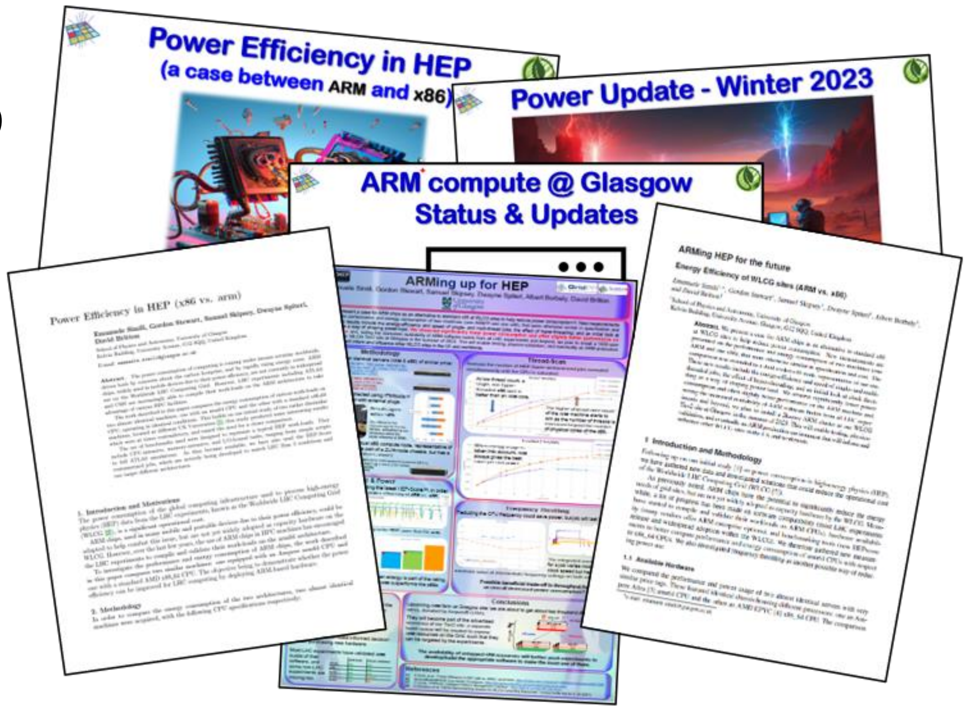
Grafana DashBoard(s)





Benchmarking and Power Measurement

"The power consumption of computing is coming under intense scrutiny worldwide, driven both by concerns about the carbon footprint, and by rapidly rising energy costs. […] " *(abstract to ACAT 2022)*


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

In 2021 we started investigating alternative architectures for Grid computing, starting with ARM chips. Lot has happened since then:

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

Methodology:

Power readings are taken via **IPMI tools** (after some validation against a metered PDU). A script collects and exports CPU, RAM, Frequency and IPMI Power metrics during the execution of a typical job or benchmark.

As benchmark, we mostly used the HEP-Score & the HEP-Benchmarking Suite:

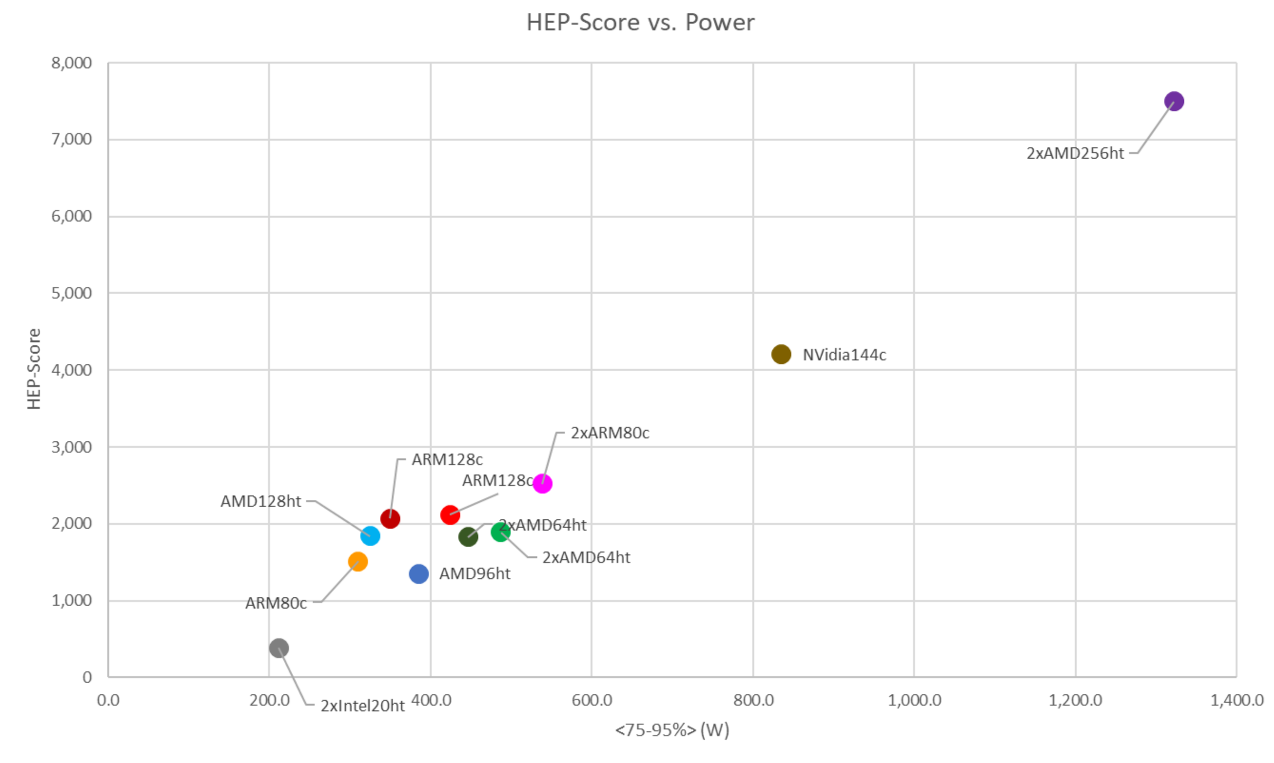
HEP-Suite: https://gitlab.cern.ch/hep-benchmarks/hep-benchmark-suite
HEP-Score: https://gitlab.cern.ch/hep-benchmarks/hep-score

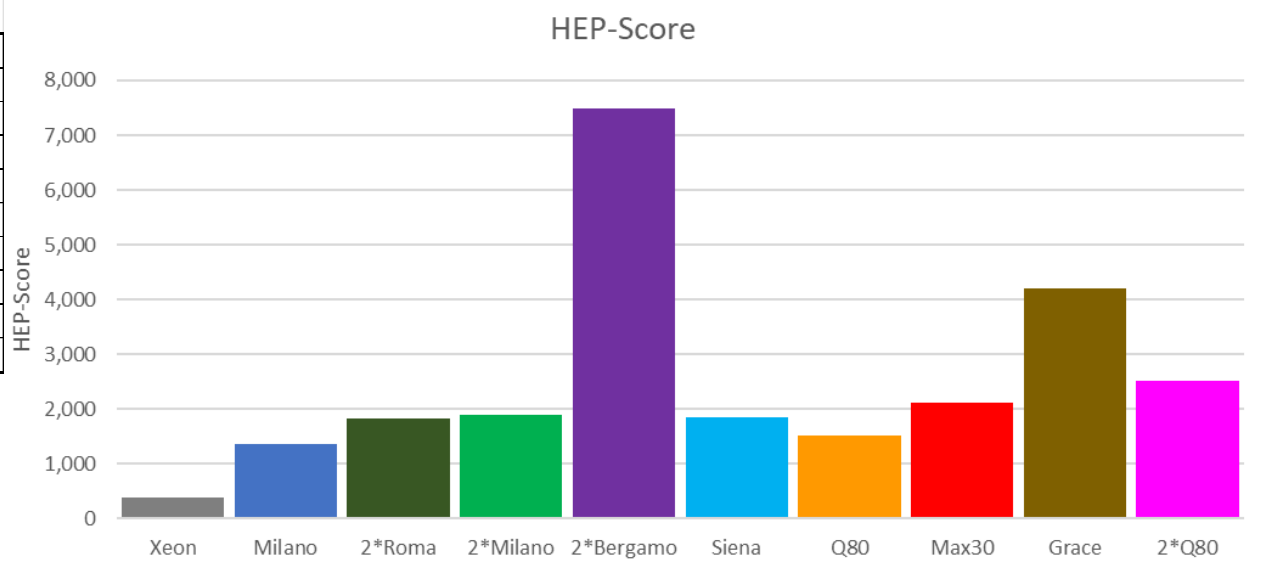
The CSV file containing the data series and the HEP-Score results are processed in ROOT (time profiles plots, power integration, statistical calculations), and cumulative results are visualized … in Excel.

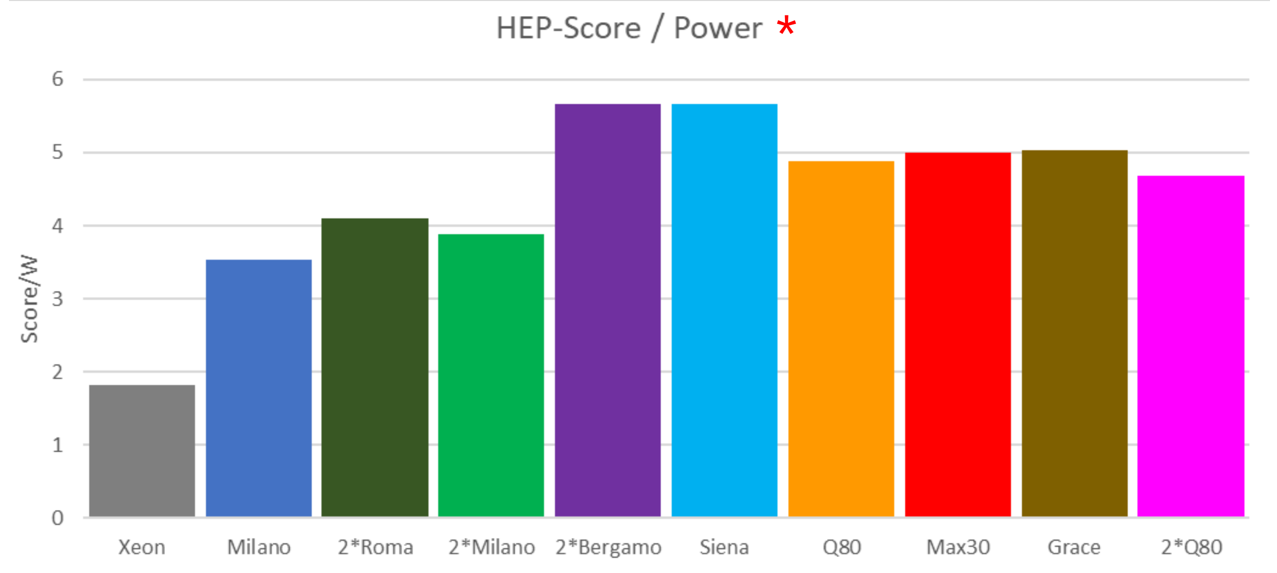
HEPscore/Watt

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*** …

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

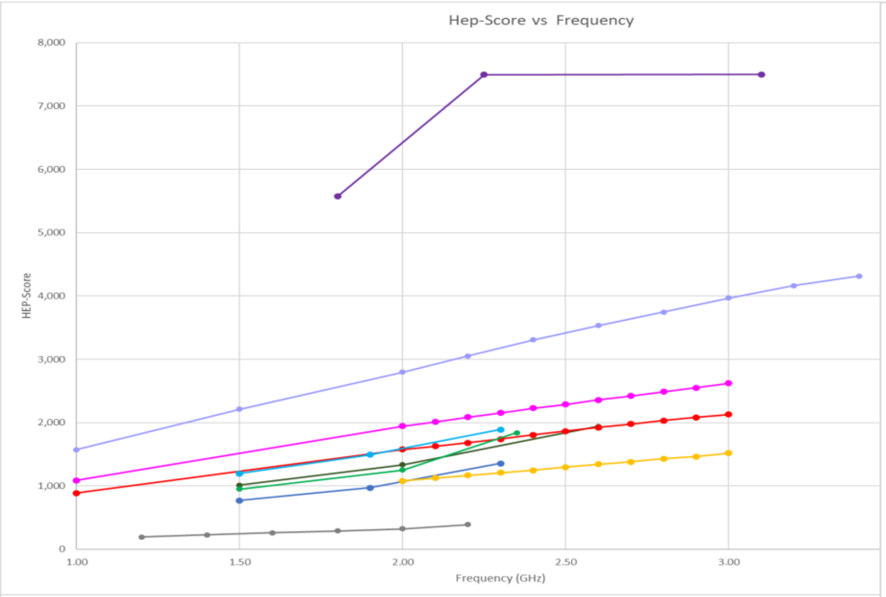


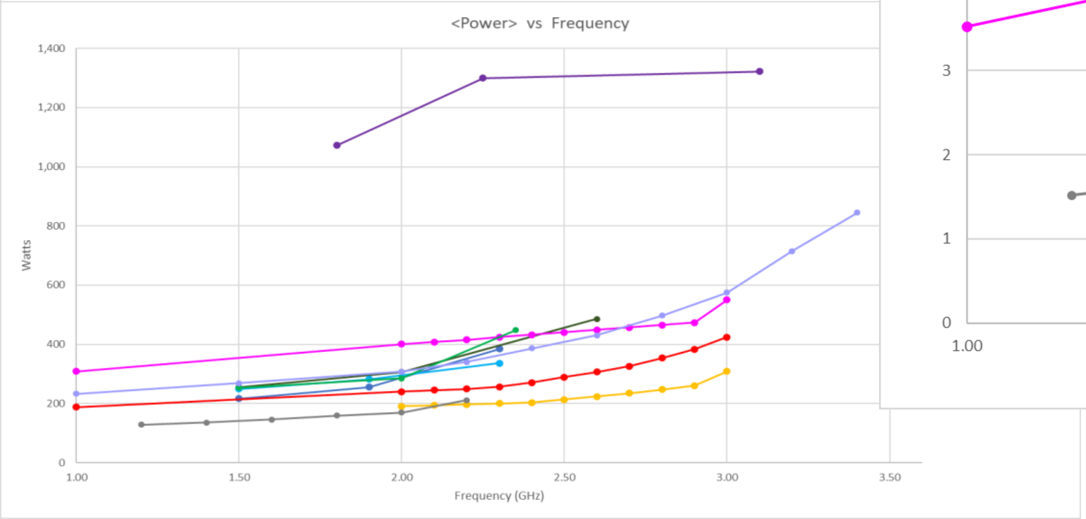


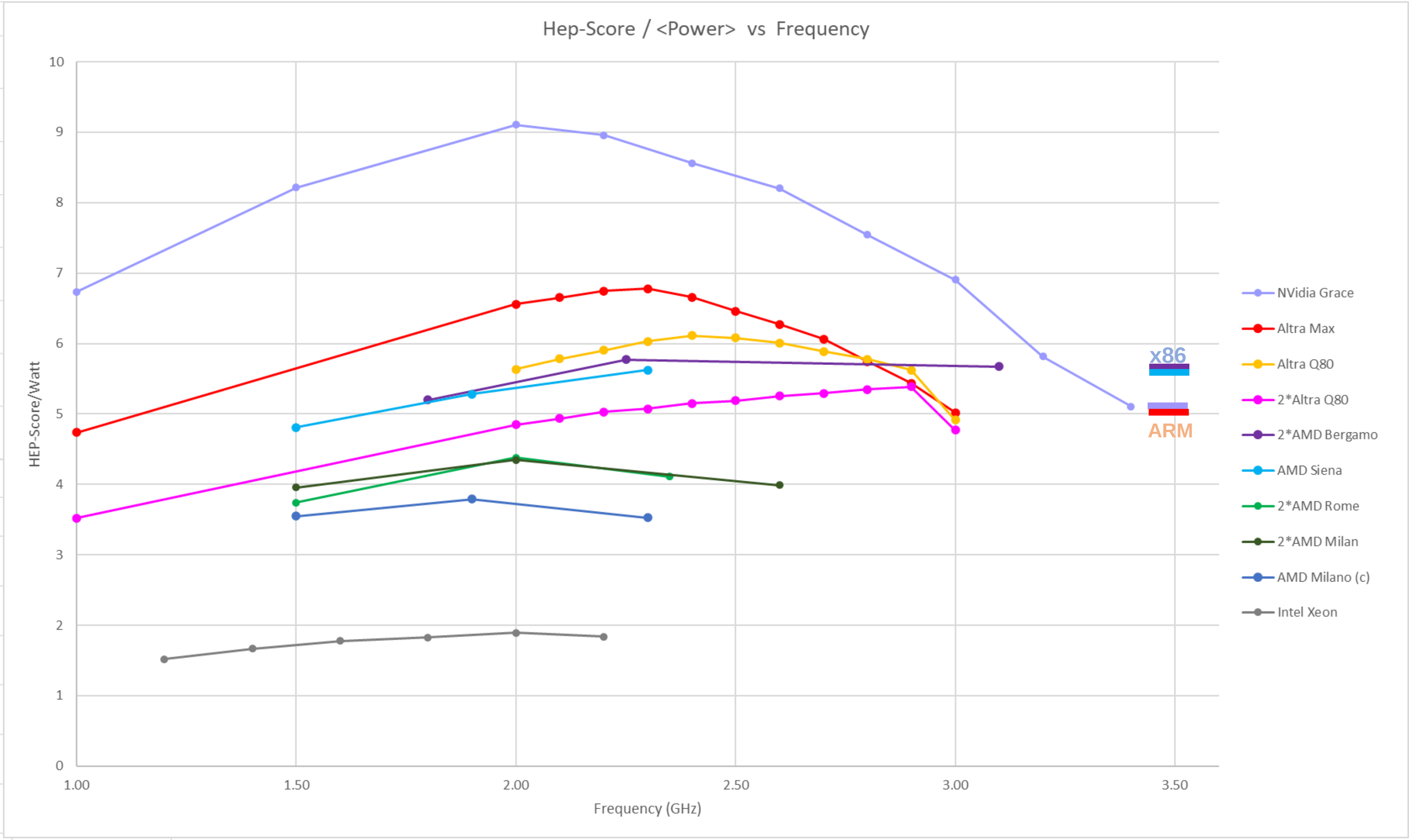


Frequency Scan (update)

HEP-Score/Watt vs. **CPU Frequency** gives a better picture of hardware potentials and shows optimal performances per watt at mid frequency range.







Outlook

❖ We plan to keep using **HTCondor** as Local Resource Management System and **ARC-CE** as Job Submission endpoint. Therefore, we will keep up-to-date on software releases and best practices.

❖ We are progressively expanding our heterogeneous Tier2 cluster beyond **x86** and **ARM**, with ongoing integration of **GPU** resources and possibly **RISC-V** worker-nodes in the next few years.

❖ We will keep pursuing our research in Sustainable HEP Computing by:
- exploring new hardware (**AmpereOne**, **RISC-V**, …),
- improving on the methodology (**HEP-Score** analysis package),
- developing new benchmarks (**GPU+CPU** with **Celeritas**).

end

*"Data spiraling out of control" by **Flux A.I.***

HTCondor management scripts

We use a number of scripts, developed or collected throughout the years: `/usr/local/bin`

```
- condor enable/disable:   sets *StartJobs* variable
- condor healthcheck:      checks health and if broken disables nodes
                           https://indico.cern.ch/event/518392/contributions/2182741/attachments/1296424/1933223/Healcheck_2016_hepsysman.pdf
- condor_ls:               lists current jobs and status
- condor_qexp:             parses condor output to get
- node_get_condorinfo      calls *condor_status* and parses job metrics for *node_exporter* (see next slides)
```

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**, …

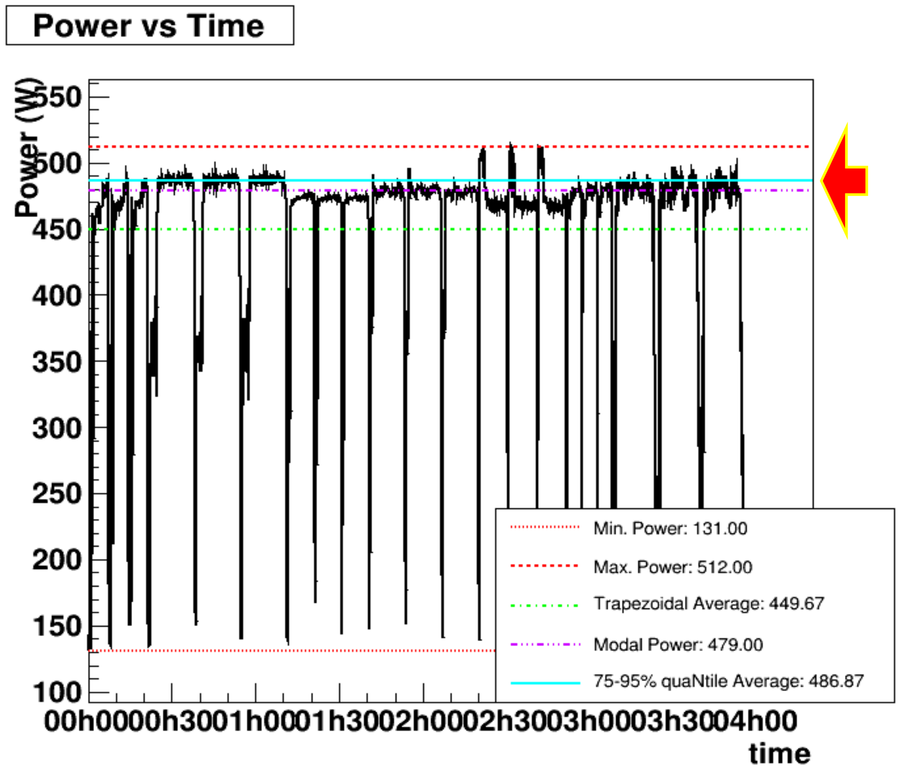
- installed **perfSONAR 5.1 testpoint** (rather than **toolkit**)

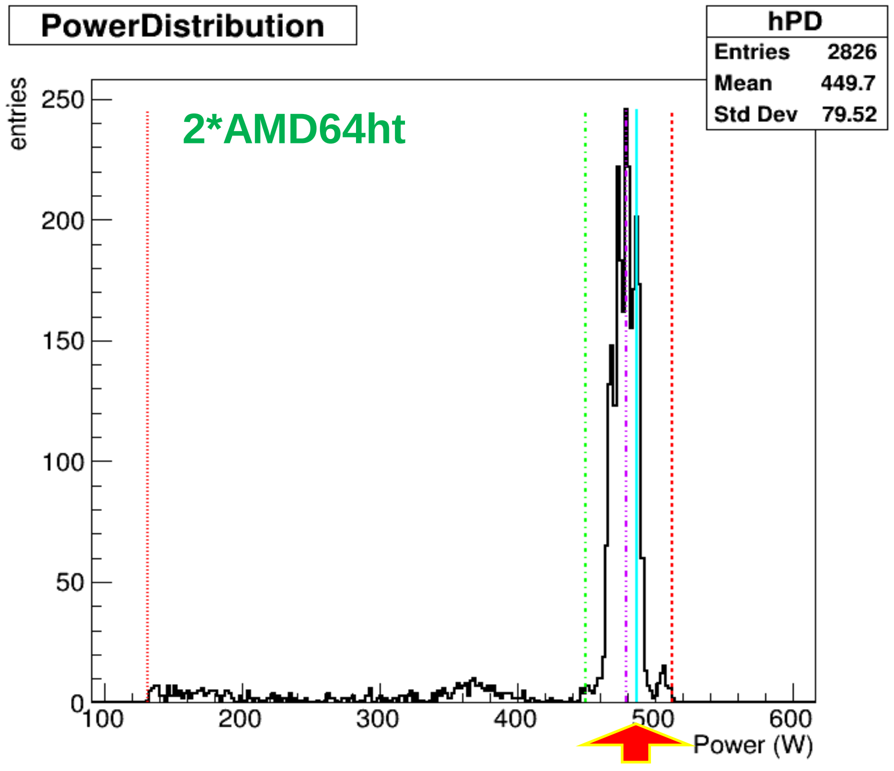
What Watt

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

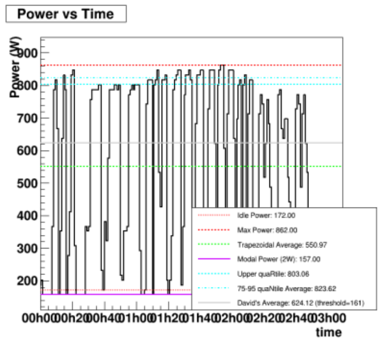


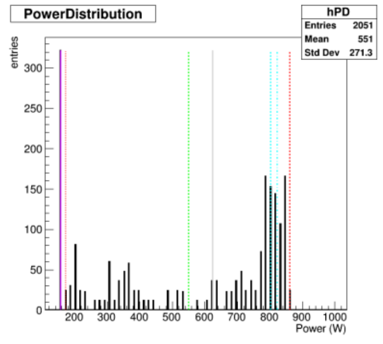


75-95% quaNtile average*

<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)





See also HEPiX presentation on aggregation metrics and k-mean:
https://indico.cern.ch/event/1433496/

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

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

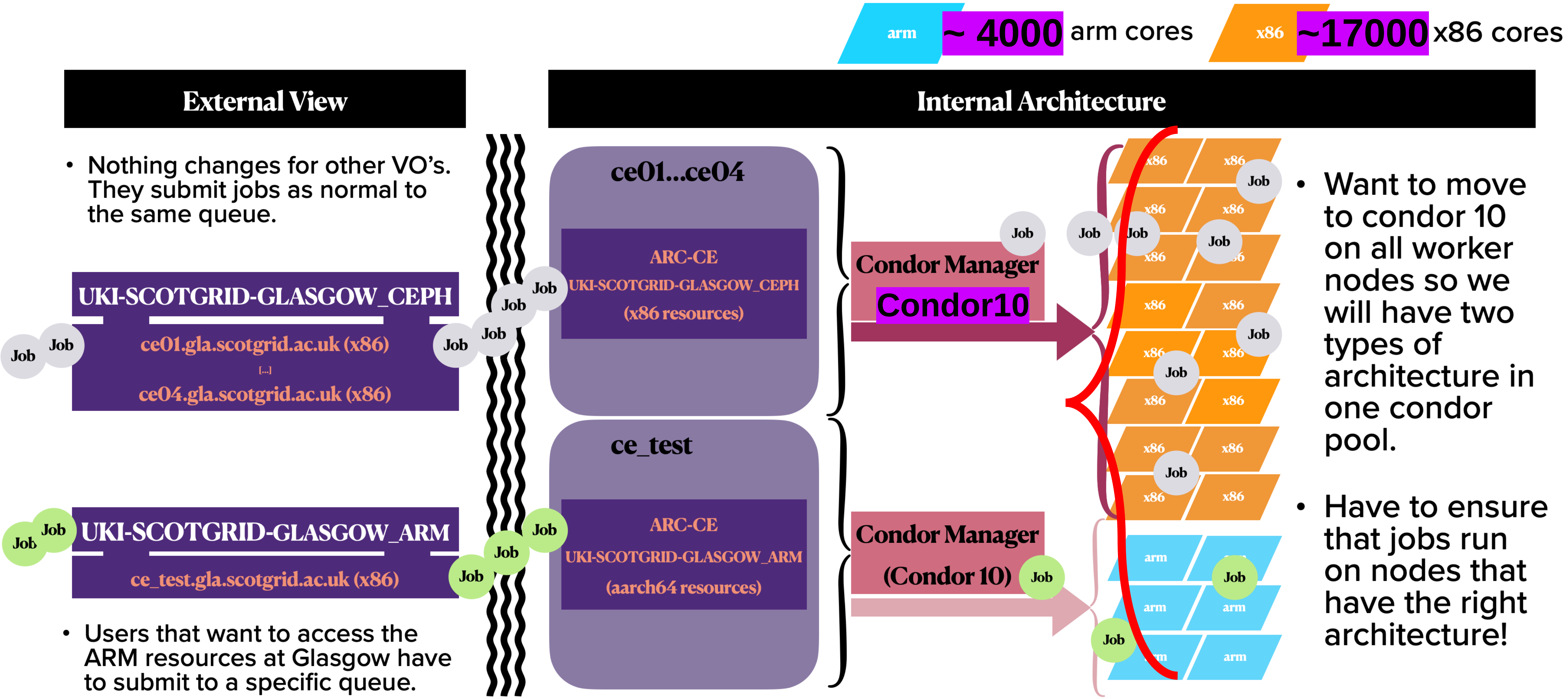
Coming soon : **AmpereOne** (96 - <u>192</u> cores)

SuperMicro

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

ARM + x86 Farm @ Glasgow

(old) Job submission chain @ ScotGrid Glasgow:



- Nothing changes for other VO's. They submit jobs as normal to the same queue.
- Users that want to access the ARM resources at Glasgow have to submit to a specific queue.
- Want to move to condor 10 on all worker nodes so we will have two types of architecture in one condor pool.
- Have to ensure that jobs run on nodes that have the right architecture!

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 !

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:

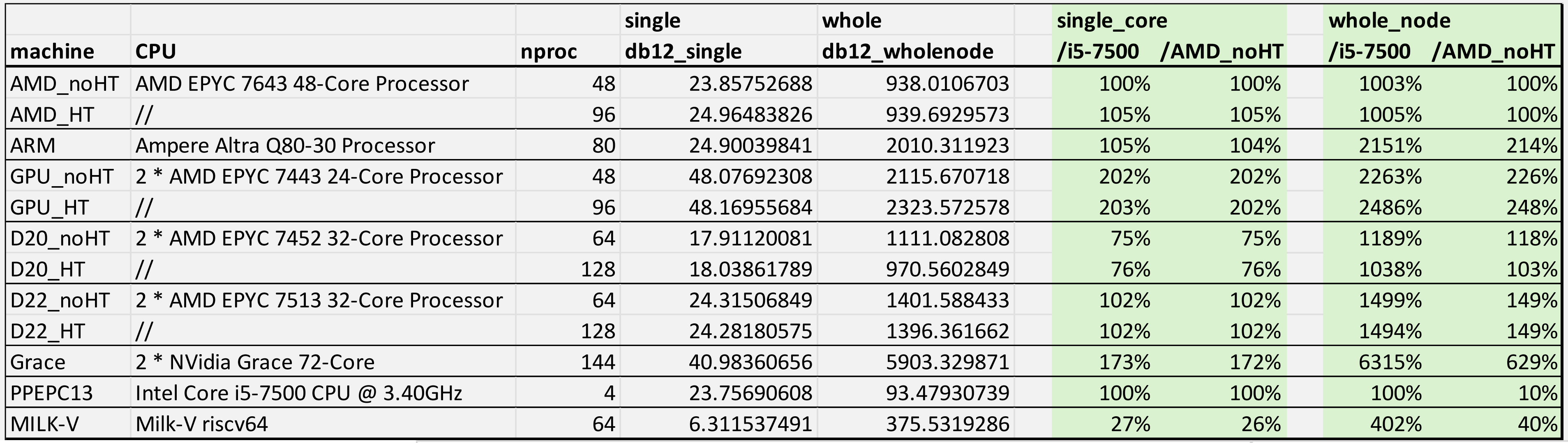
- We managed to compile and install **HEP**, software and **Grid** middleware by building from source:

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)
Geant4:	https://gitlab.cern.ch/geant4/geant4	(original Git)

- People in **CMS** made some progress in porting the **CMSSW** framework to RISC-V:
  most code can be ported, major issues are PyTorch & TensorFlow compatibility
- Accepted talk at CHEP2024: https://indico.cern.ch/event/1338689/ .

RISC Results (preliminary)

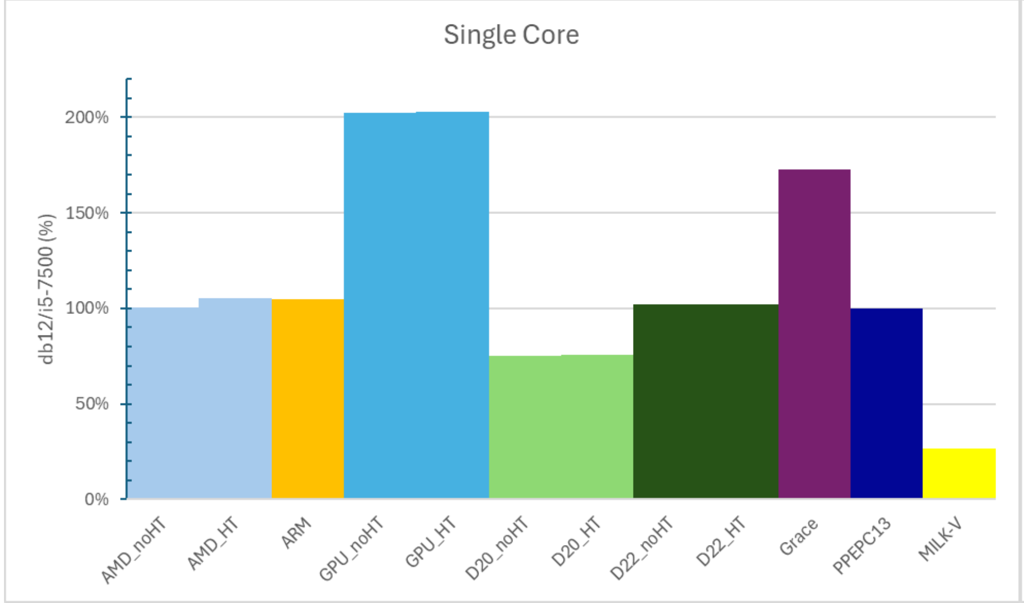
We could run **ROOT tests**, **ROOT benchmarks**, and **DB12** on RISC and few other architectures.
Results are a bit tricky to compare, because not all machines can run all benchmarks …

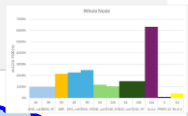


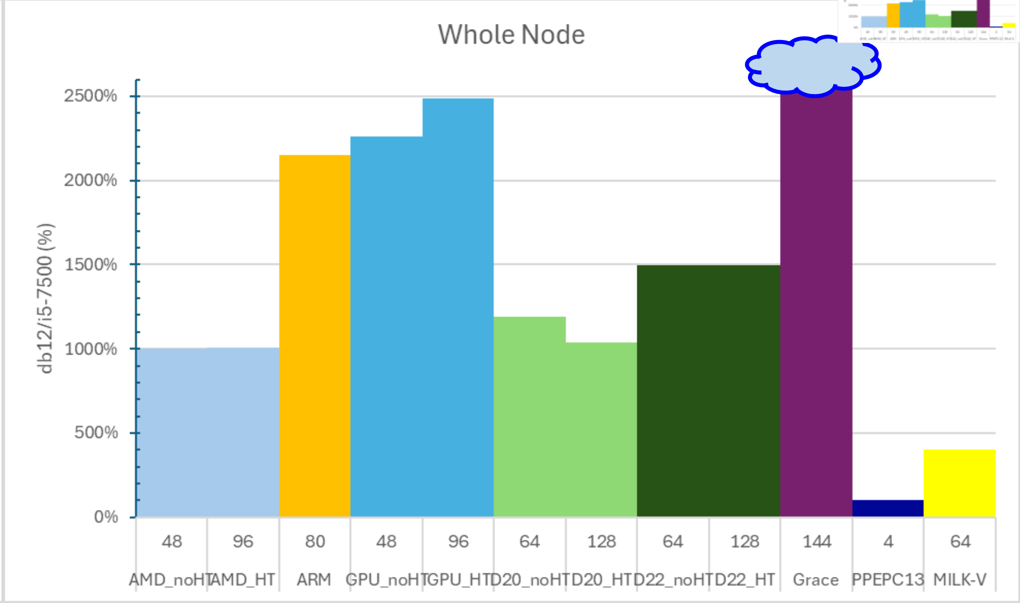
			single	whole	single_core		whole_node	
machine	CPU	nproc	db12_single	db12_wholenode	/i5-7500	/AMD_noHT	/i5-7500	/AMD_noHT
AMD_noHT	AMD EPYC 7643 48-Core Processor	48	23.85752688	938.0106703	100%	100%	1003%	100%
AMD_HT	//	96	24.96483826	939.6929573	105%	105%	1005%	100%
ARM	Ampere Altra Q80-30 Processor	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 !


Single Core
db12/i5-7500 (%)
AMD_noHT
AMD_HT
ARM
GPU_noHT
GPU_HT
D20_noHT
D20_HT
D22_noHT
D22_HT
Grace
PPEPC13
MILK-V




Whole Node
db12/i5-7500 (%)
AMD_noHT
AMD_HT
ARM
GPU_noHT
GPU_HT
D20_noHT
D20_HT
D22_noHT
D22_HT
Grace
PPEPC13
MILK-V