

# Optimization of ATLAS computing resource usage through a modern HEP Benchmark Suite via HammerCloud and PanDA

---

**Natalia Szczepanek<sup>1</sup>, Domenico Giordano<sup>1</sup>, Alessandro Di Girolamo<sup>1</sup>, Ivan Glushkov<sup>2</sup>, Gonzalo Menendez Borge<sup>1</sup>, Alexander Lory<sup>3</sup>, Ilija Vukotic<sup>4</sup>**

<sup>1</sup>CERN, Geneva

<sup>2</sup>University of Texas at Arlington (US)

<sup>3</sup>Ludwig Maximilians Universitat (DE)

<sup>4</sup>University of Chicago (US)

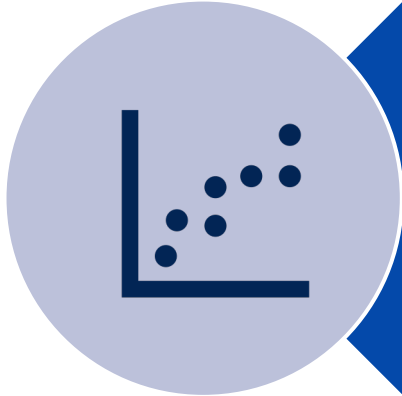
# Role of the accounting in WLCG

- WLCG counts around 1.4 million CPU cores spread over 170 data centres
- Increasing demand for computing resources (CPU, ARM, GPU)
- Need of the reliable accounting:
  - The understanding by experiments and other parties of what computing resources are available, to allow experiments to use them appropriately
  - Reporting the computing resource consumption from the site perspectives to sites, experiments, WLCG and ultimately the funding agencies



# Main objective of this work

---



Understand how a server performs when running HS23 on multi-core job slots



In-depth analysis of the actual versus declared computing site capabilities

# Some Definitions

- **Corepower** value of a server is the HEP-SPEC06 score per core
  - Transition from HS06 to HEPscore23 in April 2023
- Corepower reported by sites (**declared corepower**) is the weighted average of different corepowers of given CPU models available at the site (or queue)
  - Essential metric to understand the computing capabilities based on the specific hardware
- Comparison of corepower declared by sites in [ATLAS-CRIC](#) with **runtime corepower** based on HS23, measured via the job submission infrastructure of the ATLAS experiment
- PanDA Queue (PQ) – PanDA Queue is a concept on the PanDA WFMS of grouping and configuring a set of resources needed for processing workflows

$$\text{corepower\_runtime}^{\text{site}} = \frac{\sum_{x_{on\ cpu}} w_x \cdot \text{corepower\_runtime}_x^{\text{site}}}{\sum_{x_{on\ cpu}} w_x}$$

PanDA Queue	State	type	Cloud	Tier	Final status	core power
<a href="#">pic_MareNostrum4</a>	ACTIVE	production	ES	T1	ONLINE	27.18 27.18
<a href="#">IFIC_MareNostrum4</a>	ACTIVE	production	ES	T2D	ONLINE	27.18 27.18
<a href="#">UAM_MareNostrum4</a>	ACTIVE	production	ES	T2D	ONLINE	27.18 27.18
<a href="#">HPC2N</a>	ACTIVE	unified	ND	T1	online	25.45 25.45
<a href="#">SWT2_GOOGLE_VHMEM</a>	ACTIVE	production	US	T2D	online	24.6 24.6
<a href="#">praguec2_Barbora_MCORE</a>	ACTIVE	production	DE	T2D	online	24.5 24.5
<a href="#">UNIBE-LHEP-UBELIX</a>	ACTIVE	unified	ND	T2	online	21.7 21.7
<a href="#">SLAC</a>	ACTIVE	production	US	T3D	online	20.05 20.05
<a href="#">ANALY_SLAC_GPU</a>	ACTIVE	analysis	US	T3D	ONLINE	20 20
<a href="#">UNIGE-BAOBAB</a>	ACTIVE	unified	ND	T2	online	19.98 19.98
<a href="#">CA-IAAS-T3</a>	ACTIVE	production	CA	T3	online	19 19
<a href="#">SWT2_GOOGLE_ARM</a>	ACTIVE	unified	US	T2D	online	18.77 18.77
<a href="#">DCSC</a>	ACTIVE	unified	ND	T1	online	18.07 18.07
<a href="#">INFN-CNAF_ARM</a>	ACTIVE	unified	IT	T1	online	17.9 17.9
<a href="#">UNI-FREIBURG_NHR</a>	ACTIVE	unified	DE	T2D	online	16 16

# Runtime Corepower

- Runtime corepower per site:
  - For each CPU model on each site calculate the weight as:

$$w_x = \frac{\sum_{i \text{ on jobs}} \text{walltime\_x\_core}_i^x}{\sum_{x \text{ on cpu}} \sum_{i \text{ on jobs}} \text{walltime\_x\_core}_i^x}$$

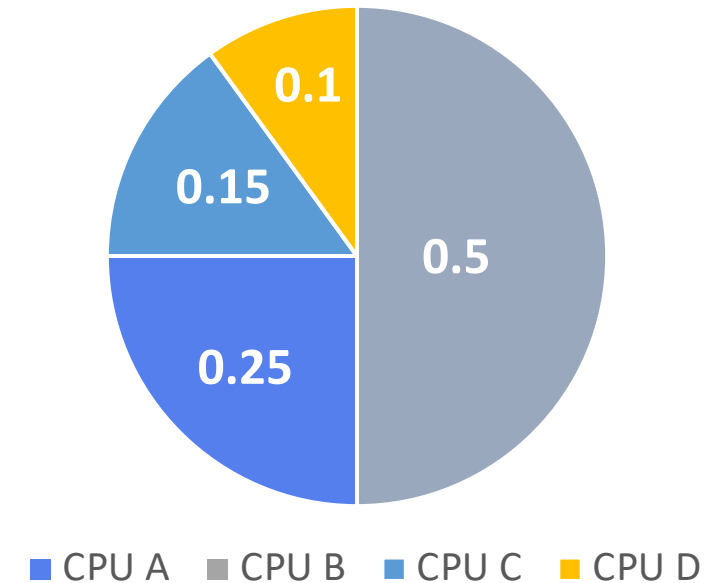
- For each site calculate the weighted average (using available benchmarking CPU Models):

$$\text{corepower\_runtime}^{\text{site}} = \frac{\sum_{x \text{ on cpu}} w_x \cdot \text{corepower\_runtime}_x^{\text{site}}}{\sum_{x \text{ on cpu}} w_x}$$

- Relative change:

$$\text{Relative change} = \frac{\text{corepower\_runtime}_s}{\text{corepower\_declared}_s} - 1$$

**Site X**  
weights derived from all jobs walltime\_x\_core



# Runtime Corepower

- Runtime corepower per site:
  - For each CPU model on each site calculate the weight as:

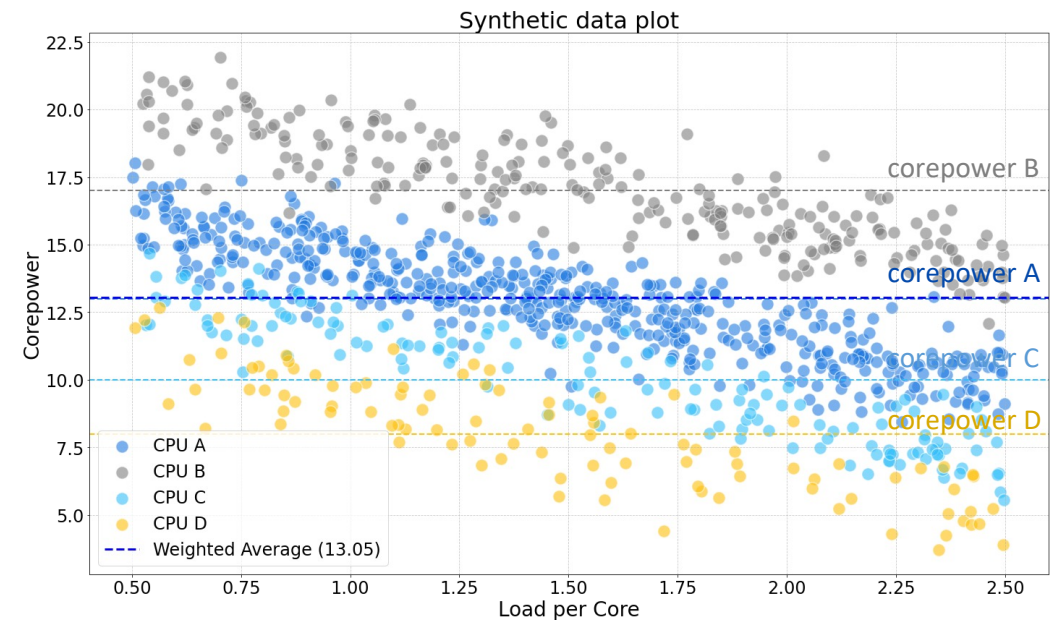
$$w_x = \frac{\sum_{i \text{ on jobs}} \text{walltime\_x\_core}_i^x}{\sum_{x \text{ on cpu}} \sum_{i \text{ on jobs}} \text{walltime\_x\_core}_i^x}$$

- For each site calculate the weighted average (using available benchmarking CPU Models):

$$\text{corepower\_runtime}_{site} = \frac{\sum_{x \text{ on cpu}} w_x \cdot \text{corepower\_runtime}_x^{site}}{\sum_{x \text{ on cpu}} w_x}$$

- Relative change:

$$\text{Relative change} = \frac{\text{corepower\_runtime}_s}{\text{corepower\_declared}_s} - 1$$



# Runtime Corepower

- Runtime corepower per site:
  - For each CPU model on each site calculate the weight as:

$$w_x = \frac{\sum_{i \text{ on jobs}} \text{walltime\_x\_core}_i^x}{\sum_{x \text{ on cpu}} \sum_{i \text{ on jobs}} \text{walltime\_x\_core}_i^x}$$

- For each site calculate the weighted average (using available benchmarking CPU Models):

$$\text{corepower\_runtime}^{\text{site}} = \frac{\sum_{x \text{ on cpu}} w_x \cdot \text{corepower\_runtime}_x^{\text{site}}}{\sum_{x \text{ on cpu}} w_x}$$

- Relative change:

$$\text{Relative change} = \frac{\text{corepower\_runtime}_s}{\text{corepower\_declared}_s} - 1$$

**Site X**  
**Runtime corepower**



$$13 * 0.5 + 17 * 0.25 + 10 * 0.15 + 8 * 0.1$$

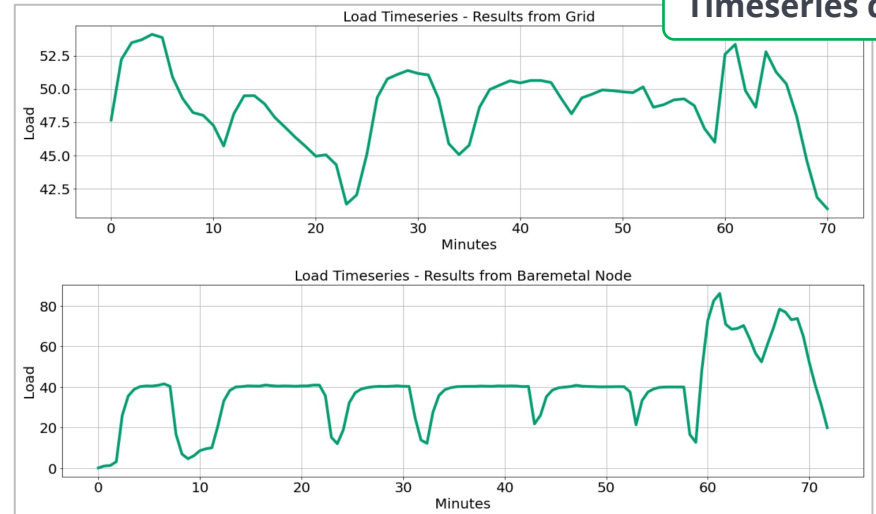
# HEP Benchmark Suite with Plugins

- [The suite](#) is an orchestrator of multiple benchmarks. The suite incorporates metrics such as machine load, memory usage, memory swap, and notably, **power consumption**
- Suite Plugins:
  - Run alongside benchmarks
  - Flexible modification and addition of collected metrics
- [HEPScore23](#) (HS23) - The official HEP Score configuration composed by 7 workloads from 5 experiments

## Suite configuration

```
plugins:  
  CommandExecutor:  
    metrics:  
      cpu-frequency:  
        command: cpupower frequency-info -f  
        regex: 'current CPU frequency: (?P<value>\d+).*'  
        unit: kHz  
        interval_mins: 1  
      power-consumption:  
        command: >  
          sudo ipmitool sensor get 'PS1 Power In' ; sudo ipmitool sensor get  
            'PS2 Power In'  
        regex: 'Sensor Reading\s+:\s*(?P<value>\d+).*'  
        unit: W  
        interval_mins: 1  
      load:  
        command: uptime  
        regex: 'load average: (?P<value>\d+\.\d+),'  
        unit: ''  
        interval_mins: 1
```

## Timeseries data



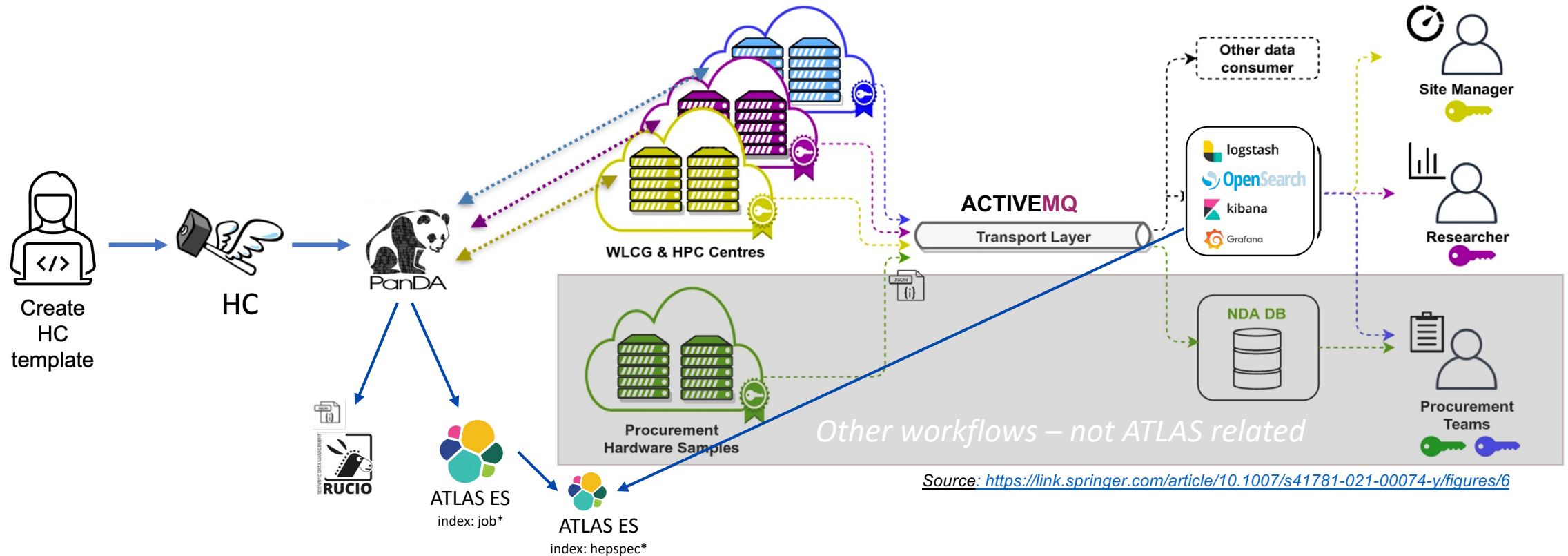


# Submission Infrastructure

Automated submission of HS23 via [HammerCloud](#) and [PanDA](#)

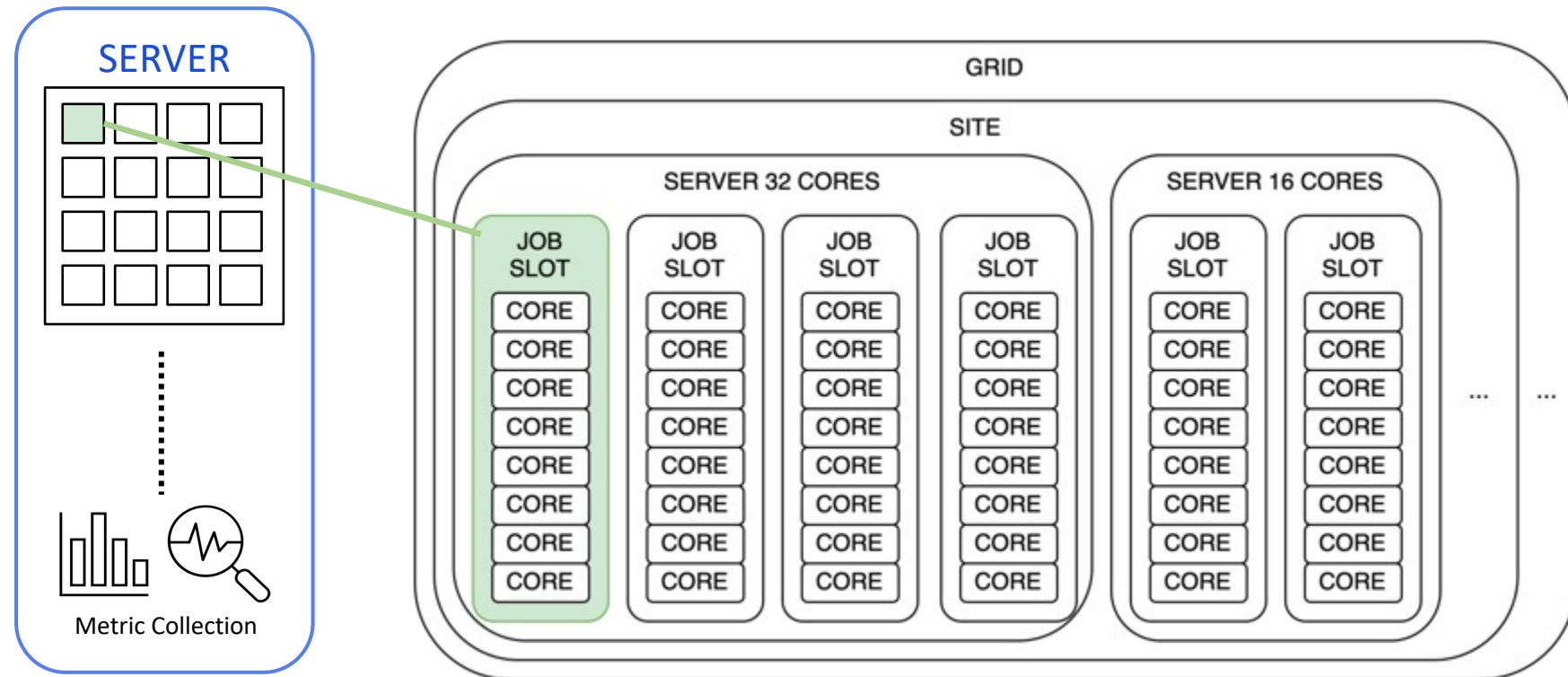
## Infrastructure:

- [PanDA](#), [HammerCloud](#), [Rucio](#), [ActiveMQ](#), [OpenSearch](#), [Elasticsearch](#), [Grafana](#), [Kibana](#)...



# Probing job slots

- Each site has servers with a variety of CPU models and number of cores (256, 128, 64...)
- We are running the benchmark injecting the HEP Suite script as a normal experiment job running inside the PILOT Apptainer
- We probe multi-core job slots (8 cores)



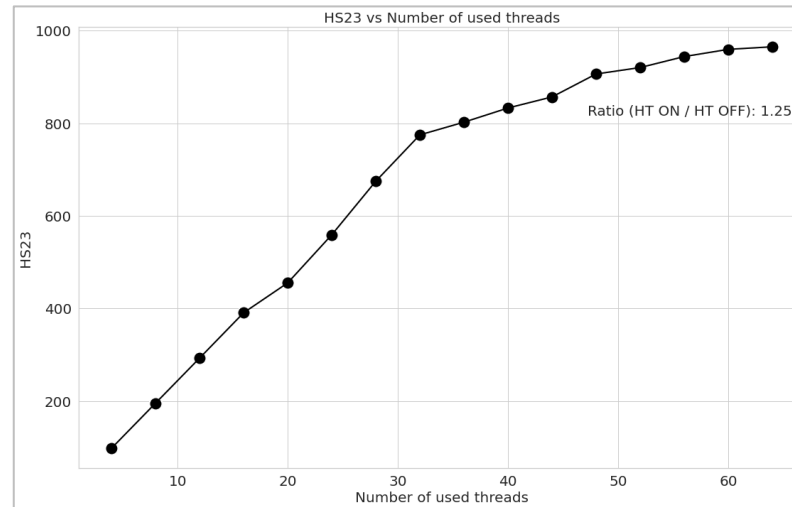
# Analysis

\*Analysis is done per Site  
Site can contain many PanDA Queues, therefore few data points can be visible

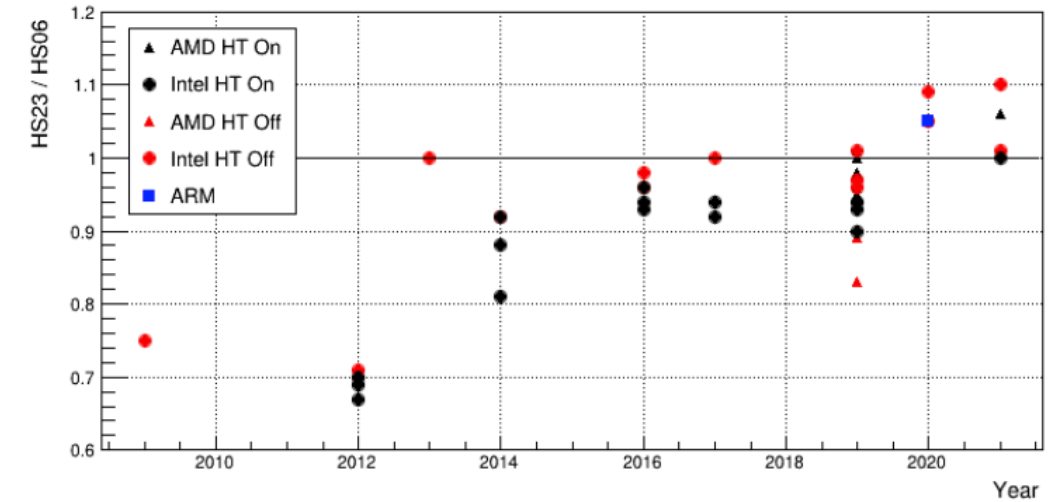
# Systematic Uncertainty

1. HS23/HS06 scaling
2. Calculating average over measured runtime HS23 probes
3. Some CPU can switch between HT ON / HT OFF
4. The performance vs load variation
5. Calculating weights

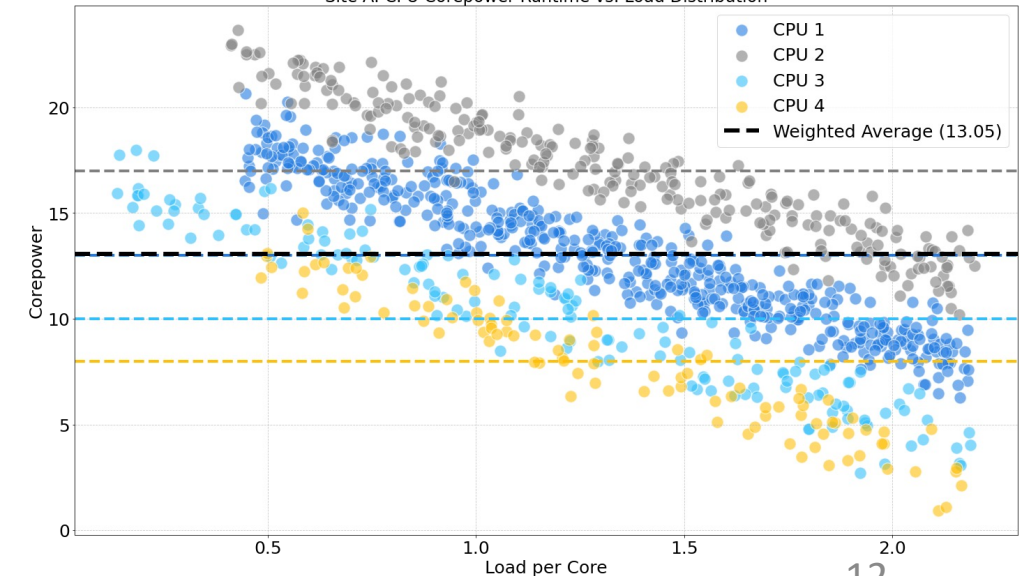
→ **Threshold: considering critical only discrepancies  $> \pm 25\%$**



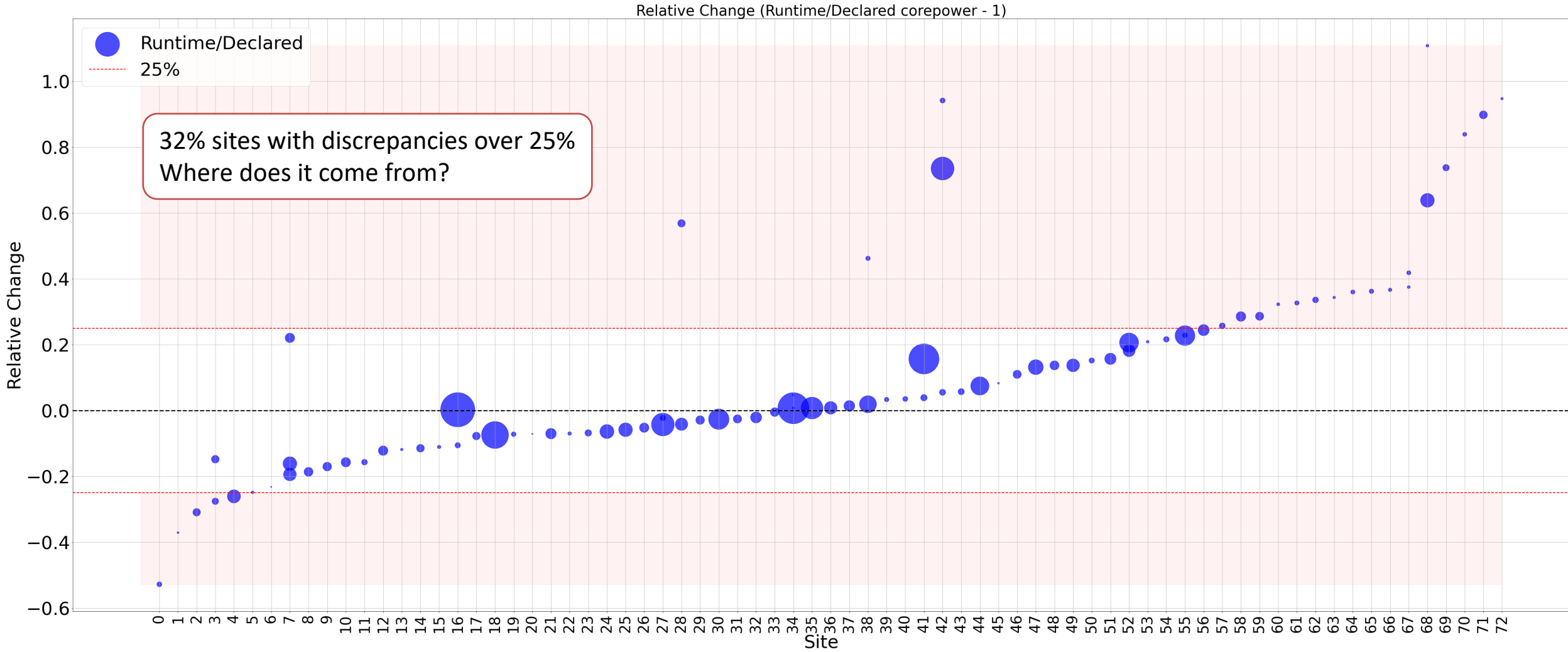
CPU models per year of release



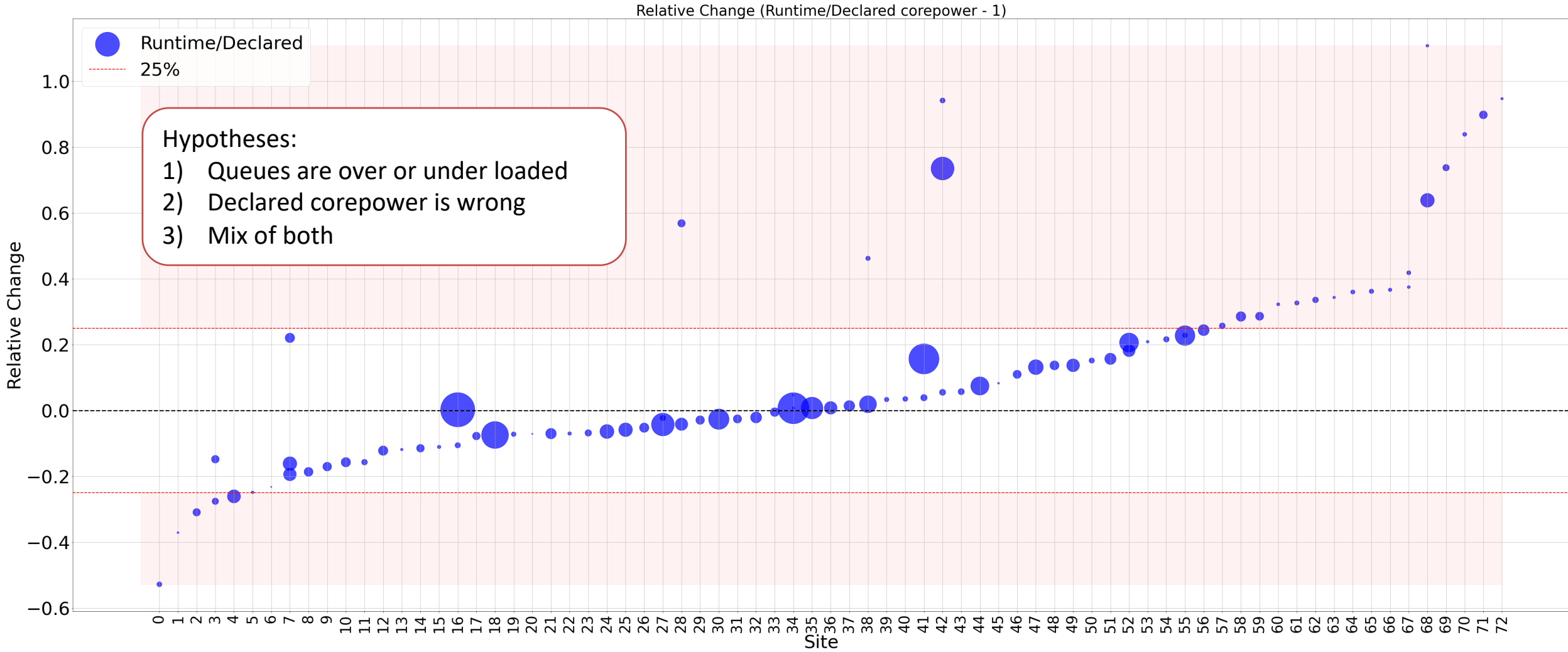
Site A: CPU Corepower Runtime vs. Load Distribution



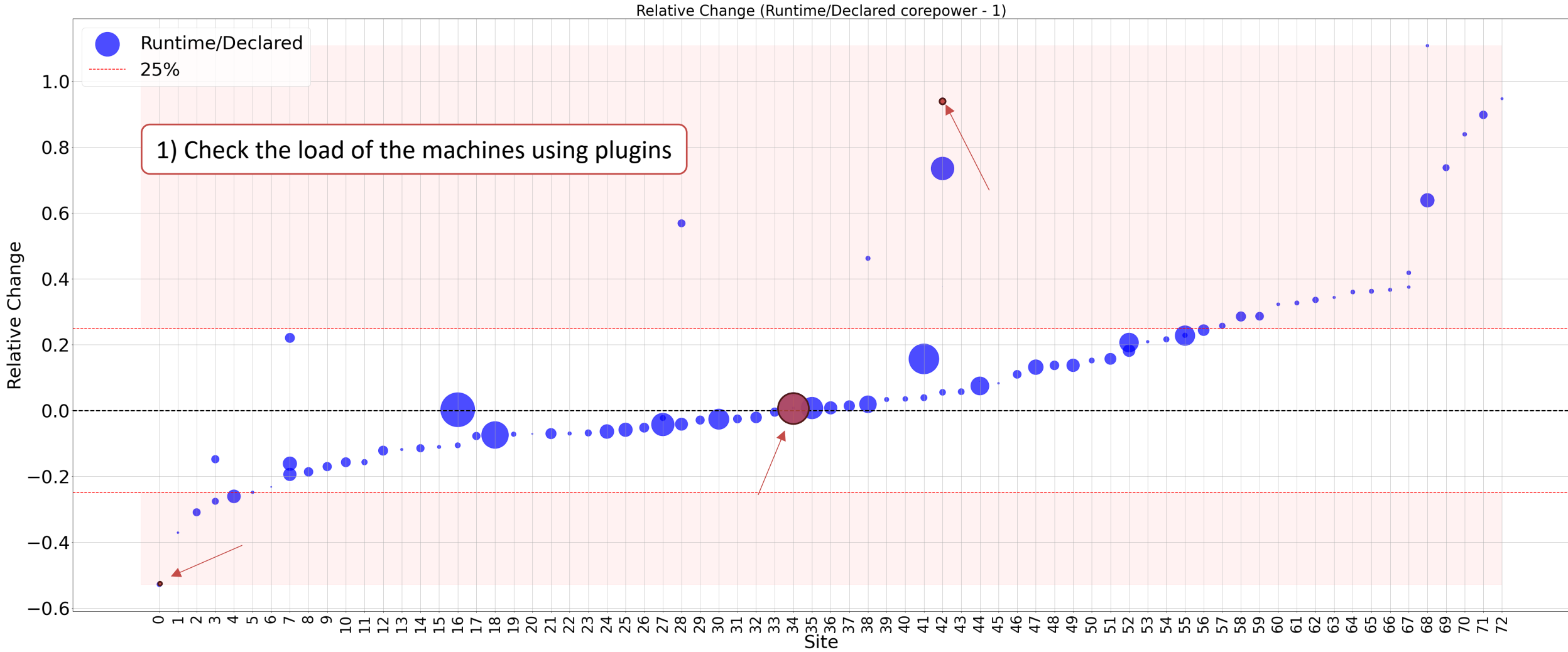
# Relative change for different ATLAS sites



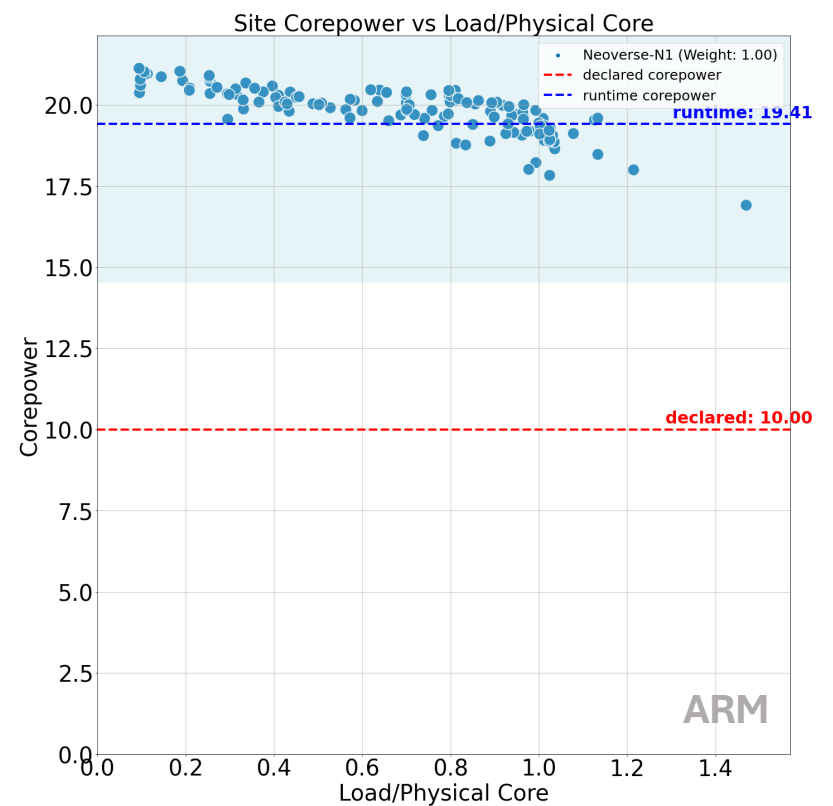
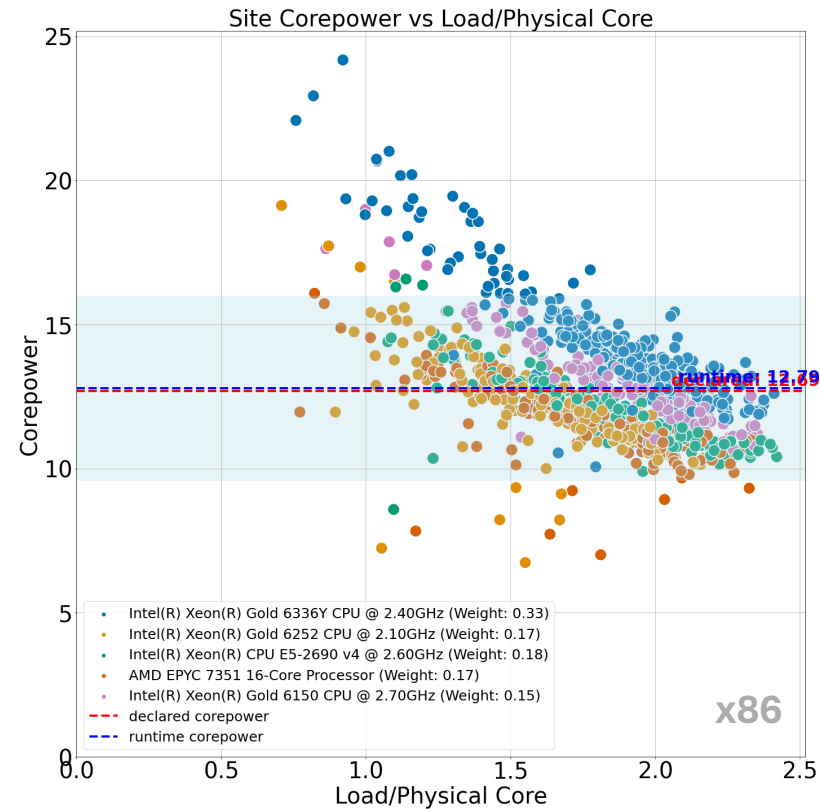
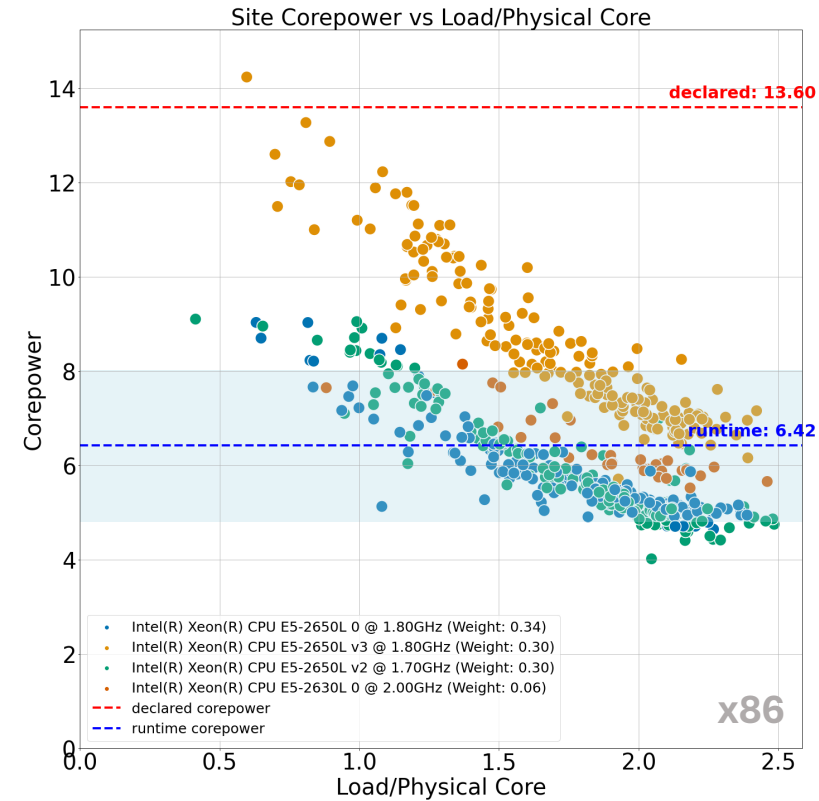
# Relative change for different ATLAS sites



# Relative change for different ATLAS sites



# Corepower vs Load



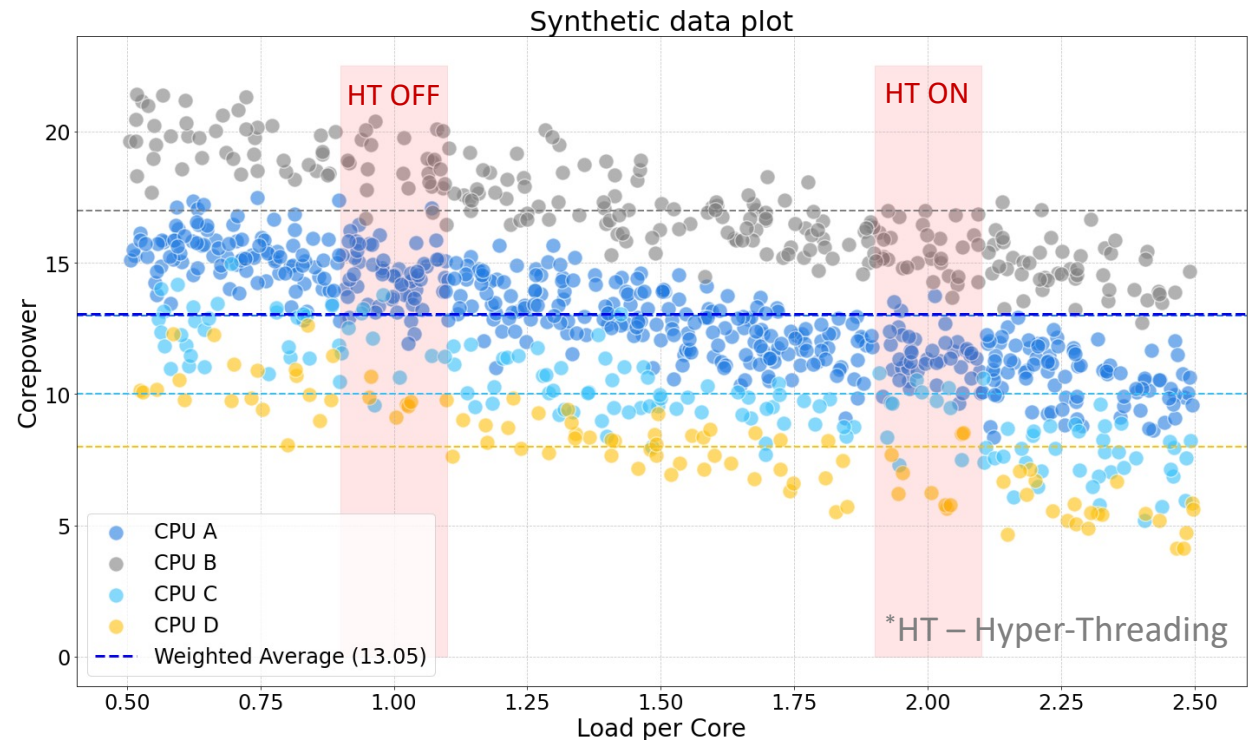
Individual approach to each case:

- Load of the machines has impact on the final performance
- Old CPU Models with high load can cause negative relative change
- ARM is more resistant for a load changes

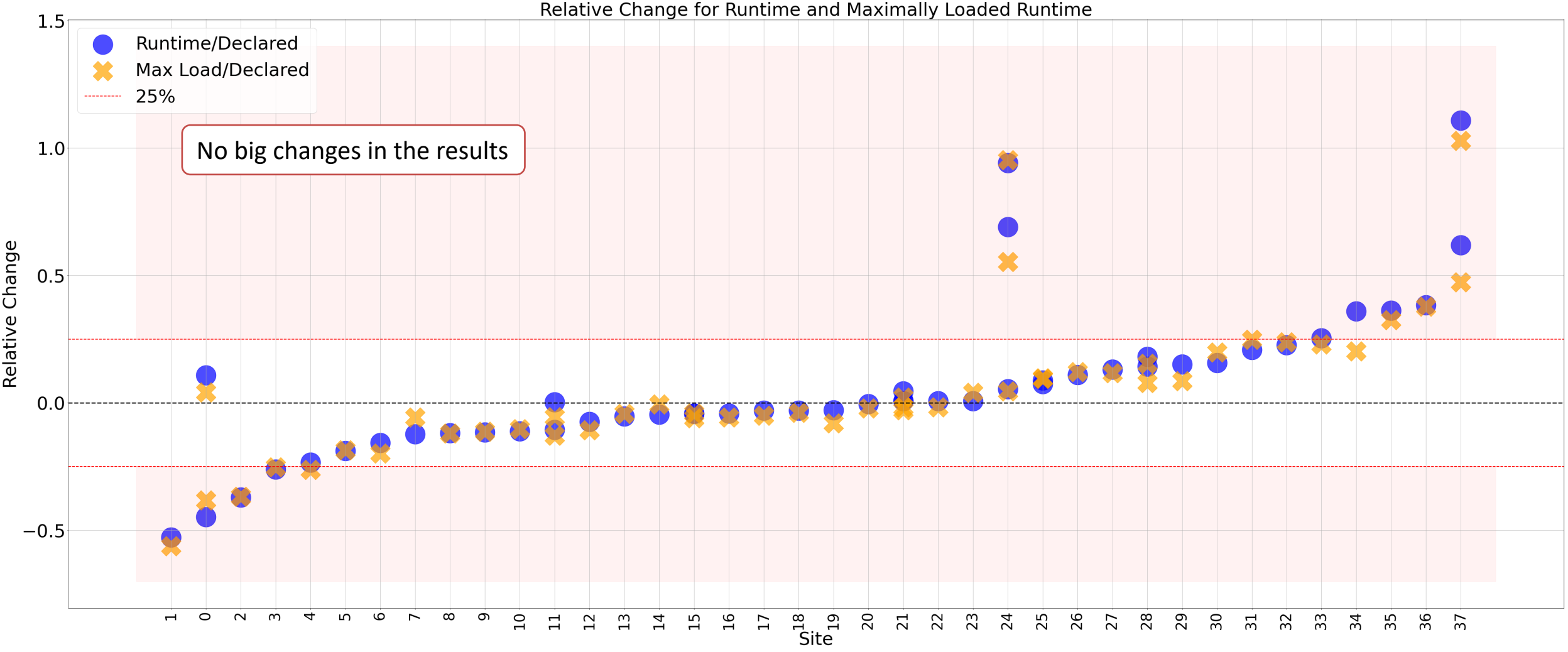


# Loaded Range of Runtime Corepower

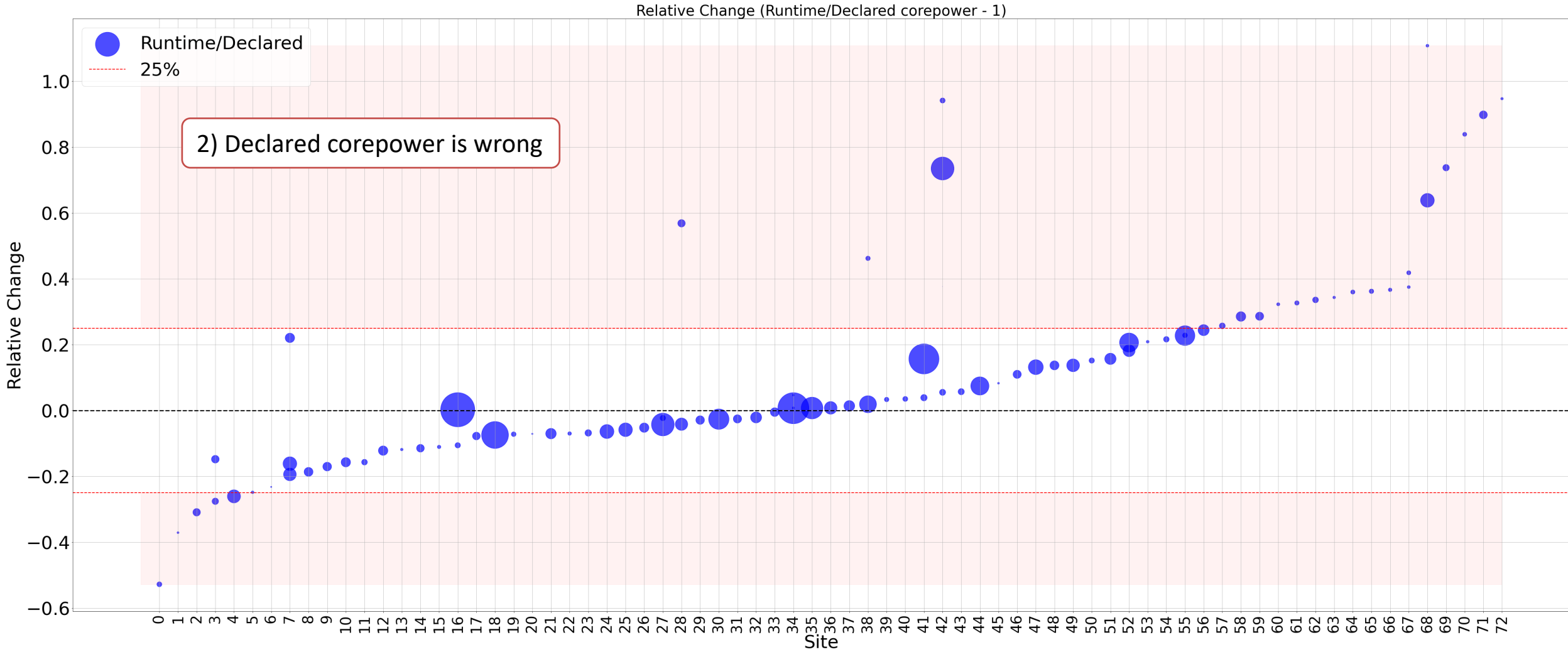
- For sites which runtime corepower is higher than declared, the cause could be that we are computing the average over the whole load range, whereas the declared measurements are done on fully loaded server
- **Validation:** we have evaluated the average score with data only at full load ~1 (HT OFF) or load ~2 (HT ON)
  - Results in the next slide



# Loaded Range of Runtime Corepower

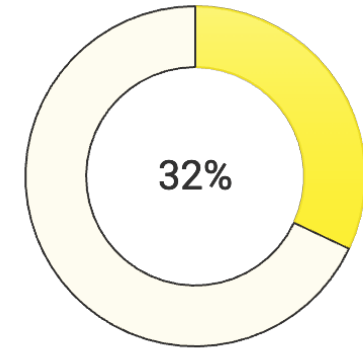


# Relative Change for Different ATLAS sites

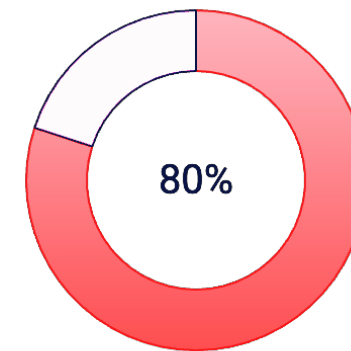


# Analysis of Corepower Values – Data Source

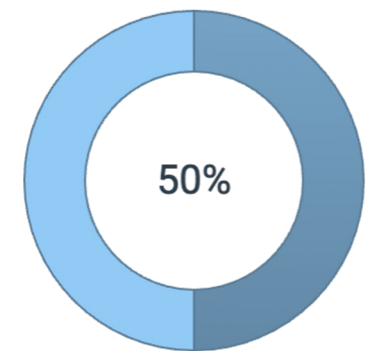
- 32% of the sites with discrepancies over 25%
- Queues with newer architectures suffer more from the wrong declared values (ARM)
- **The declared corepower values from ATLAS-CRIC seem imprecise for some sites**
- Discrepancies visible in and between different data sources
- Considering the site contributions and the differences between declared and runtime values, we found a 6% advantage in favour of the runtime performance



Discrepancies > 25%



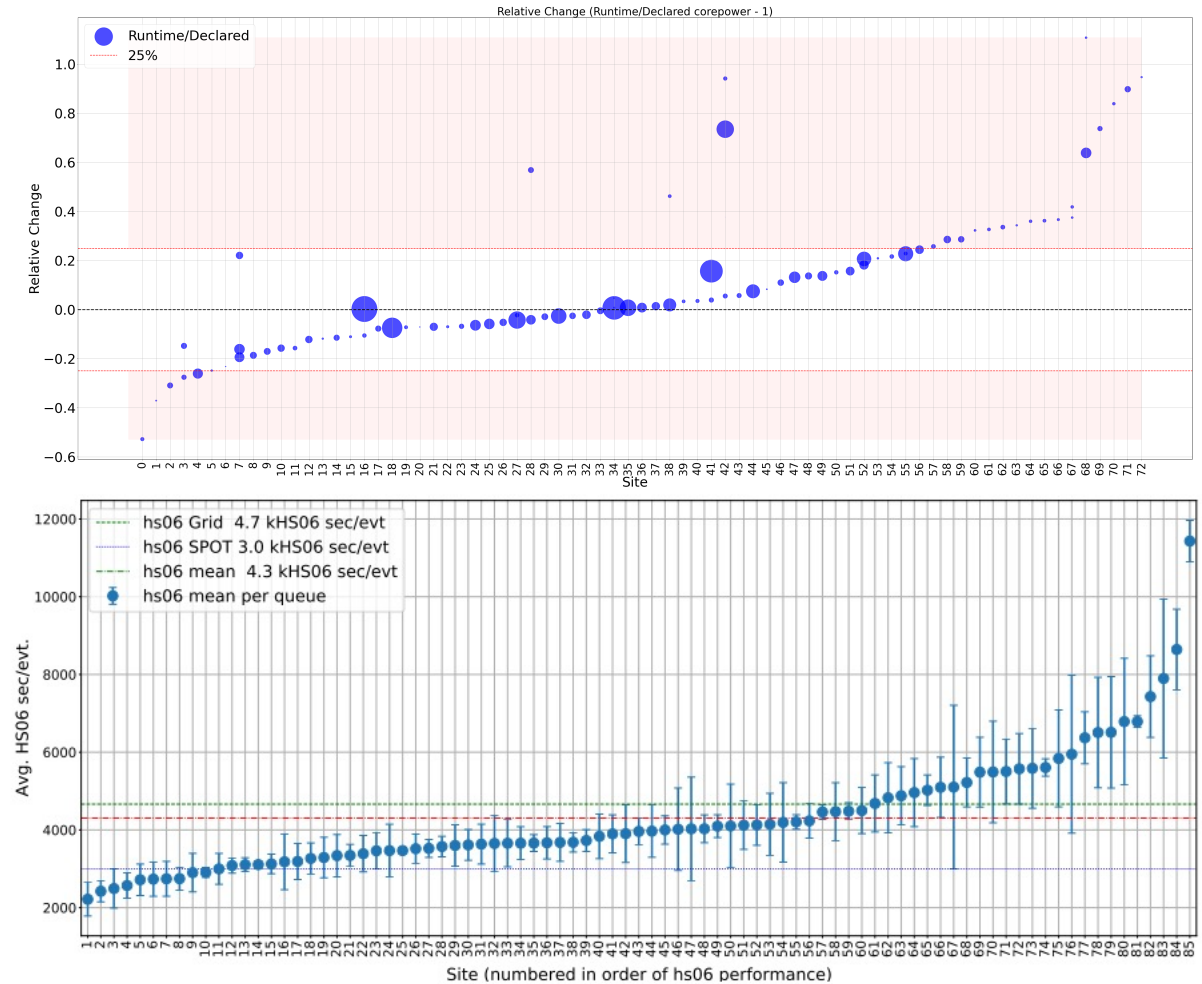
Cloned Panda Queues



Inherited Corepower

# Similar Studies – Similar Results

- Similar studies have been presented in ATLAS and at ACAT
  - “A comparison of HEPSPec benchmark performance on ATLAS Grid-Sites versus ideal conditions” (2022)
- Same type of discrepancies was found



[A comparison of HEPSPec benchmark performance on ATLAS Grid-Sites versus ideal conditions](#)

by Michael Boehler


# Summary

## Infrastructure

- Successful implementation of automated HEPscore23 submission via Big PanDA using HammerCloud
- An approach to validate the declared corepower values, a cross-check with official accounting

## Analysis

- Runtime/Declared corepower relative change suggest that there is a discrepancy between what performance is declared and what is the real, runtime performance
- Considering the site contributions and the differences between declared and runtime values, we found a 6% advantage in favour of the runtime performance
- The approach of measuring the runtime corepower allows to detect sites with large declared corepower discrepancies
- Discrepancies between different data sources has been found and reported



Thank you!  
Q&A



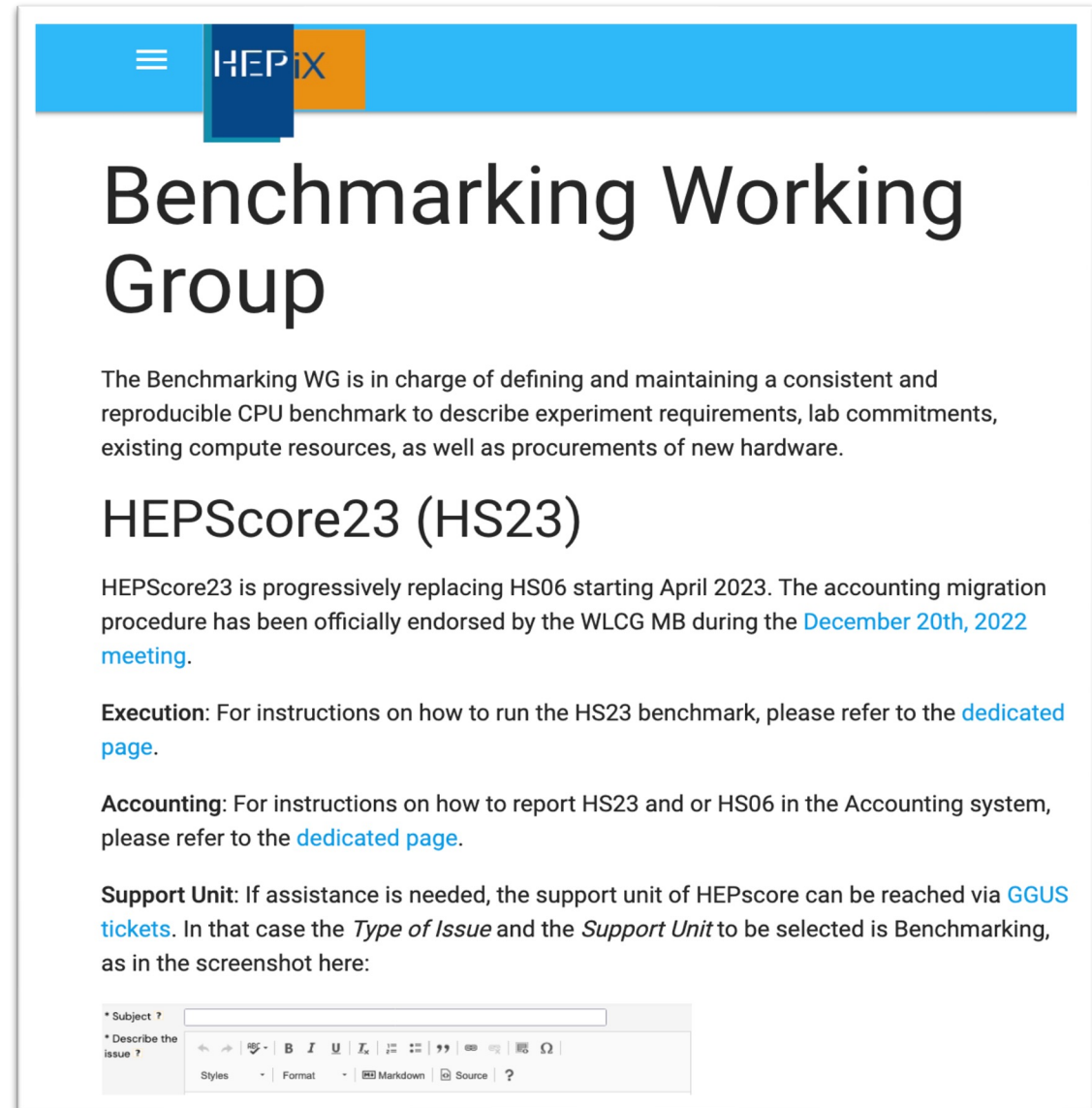
# Appendix



# HEPScore Documentation

- How to run HEPscore
- Table with HS23 scores declared by sites

HS23 table: [http://w3.hepik.org/benchmarking/scores\\_HS23.html](http://w3.hepik.org/benchmarking/scores_HS23.html)



HEPiX

## Benchmarking Working Group

The Benchmarking WG is in charge of defining and maintaining a consistent and reproducible CPU benchmark to describe experiment requirements, lab commitments, existing compute resources, as well as procurements of new hardware.

### HEPScore23 (HS23)

HEPScore23 is progressively replacing HS06 starting April 2023. The accounting migration procedure has been officially endorsed by the WLCG MB during the [December 20th, 2022 meeting](#).

**Execution:** For instructions on how to run the HS23 benchmark, please refer to the [dedicated page](#).

**Accounting:** For instructions on how to report HS23 and or HS06 in the Accounting system, please refer to the [dedicated page](#).

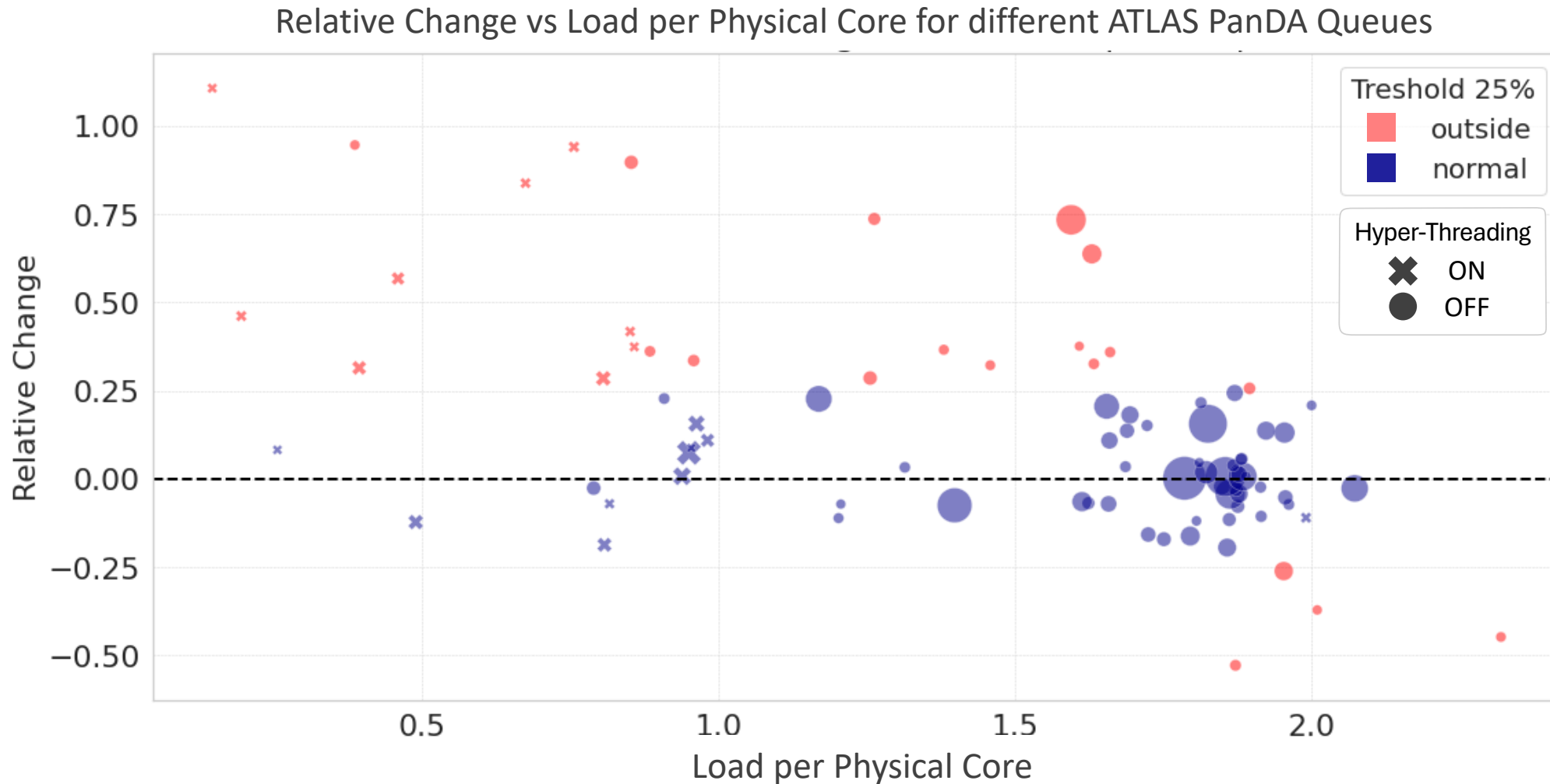
**Support Unit:** If assistance is needed, the support unit of HEPscore can be reached via [GGUS tickets](#). In that case the *Type of Issue* and the *Support Unit* to be selected is Benchmarking, as in the screenshot here:

\* Subject ?  
\* Describe the issue ?

Styles - | Format - | Markdown | Source | ?

Source: <http://w3.hepik.org/benchmarking/>

# Relative Change of Corepower VS Load

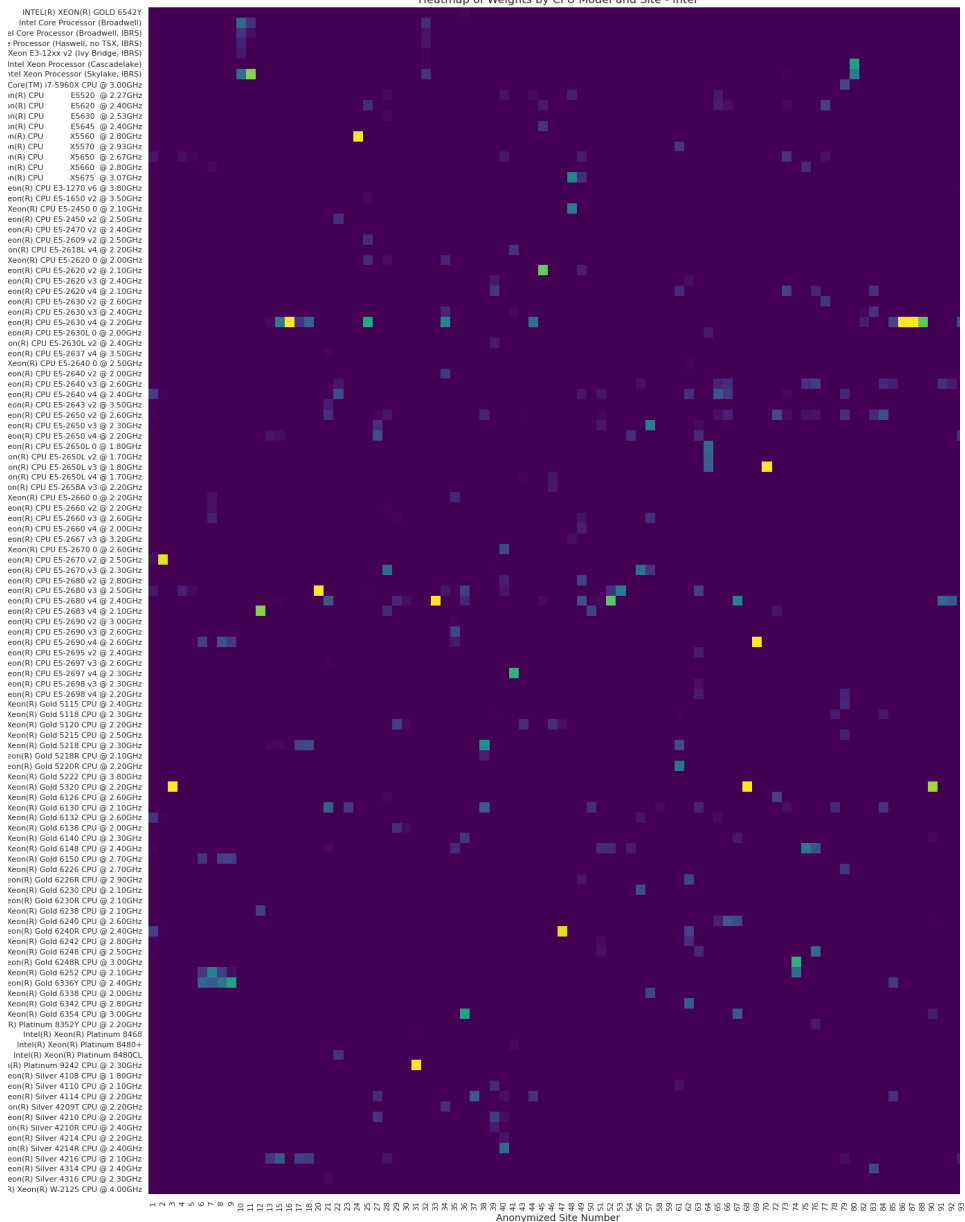


# Infrastructure detailed list

---

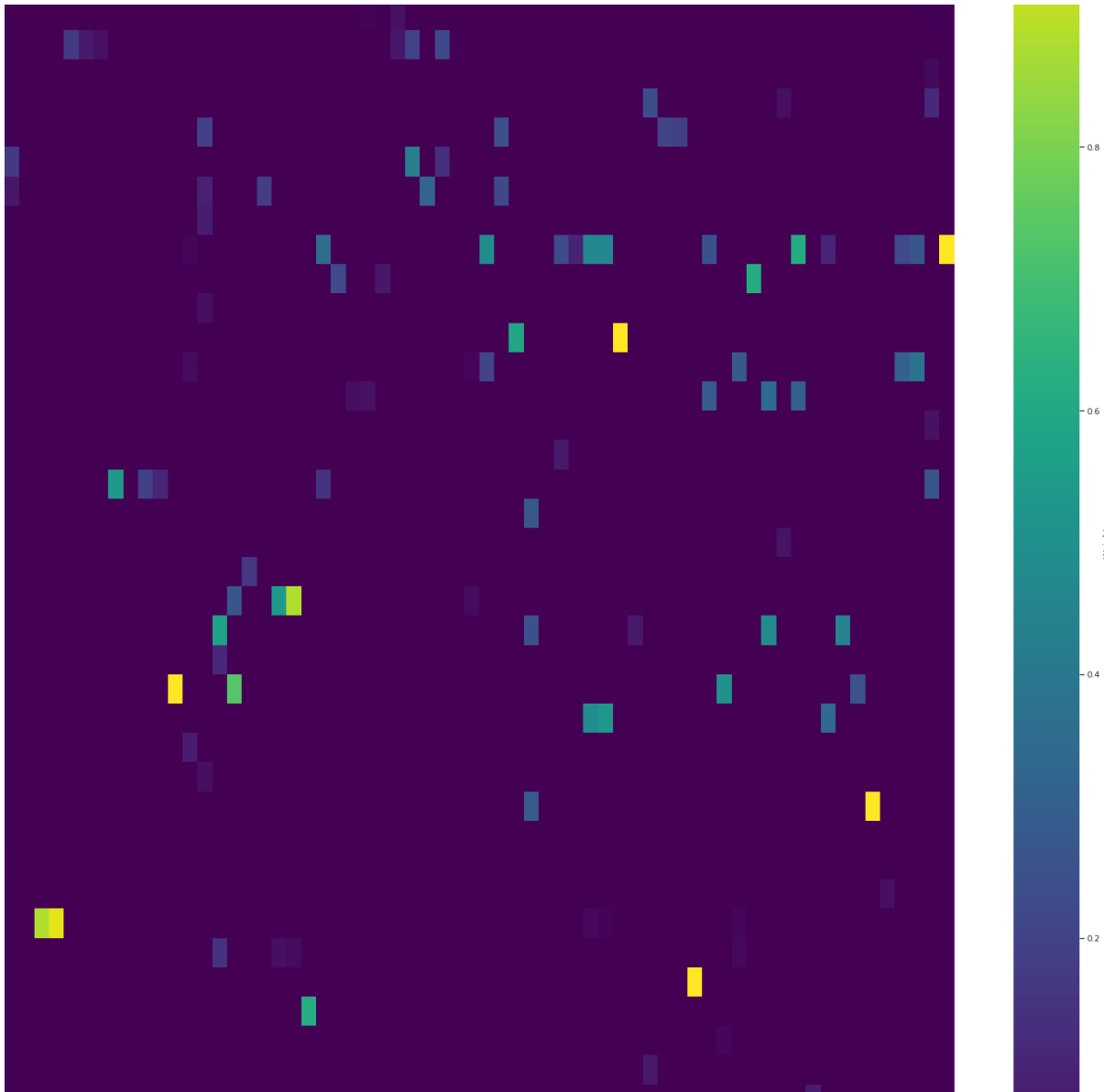
- [HammerCloud](#) - a testing system developed by CERN IT initially for ATLAS and then adapted to CMS and LHCb to run custom tests on Grid sites
- [BigPanDA](#) - the next-generation monitoring system of the ATLAS PanDA Workload Management System (WMS), which was developed in the ATLAS Experiment and brought into production in 2014
- [CRIC](#) - Computing Resource Information Catalogue, a unified information system for WLCG, which aims to facilitate distributed computing operations for the LHC experiments and consolidate WLCG topology information. CRIC also performs data validation and provides coherent view and topology description to the LHC VOs for service discovery and configuration
  - [ATLAS-CRIC](#) – ATLAS CRIC dedicated instance

Heatmap of Weights by CPU Model and Site - Intel



CPU Model (Benchmarking Model)

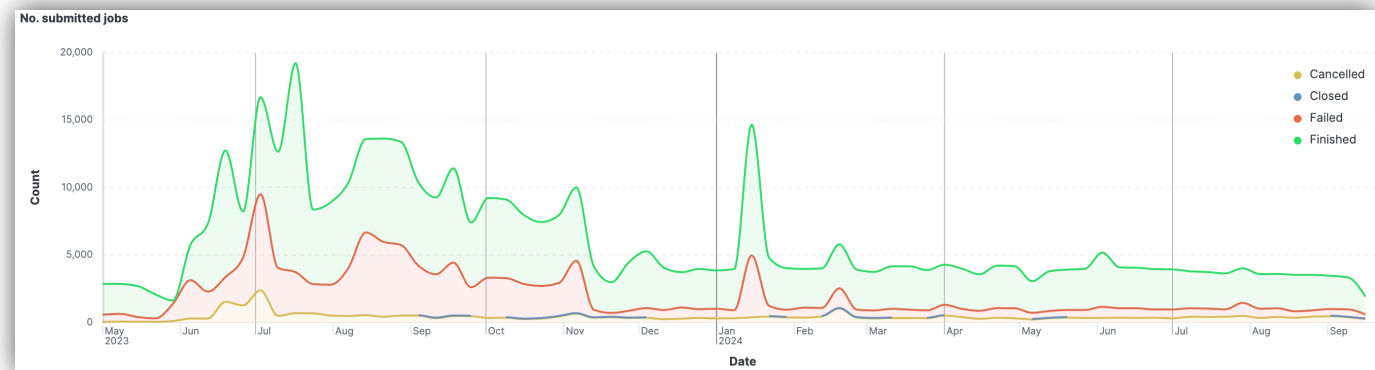
- AMD EPYC 7313 16-Core Processor
- AMD EPYC 7351 16-Core Processor
- AMD EPYC 7351P 16-Core Processor
- AMD EPYC 7352 24-Core Processor
- AMD EPYC 7402 24-Core Processor
- AMD EPYC 7413 24-Core Processor
- AMD EPYC 7443 24-Core Processor
- AMD EPYC 7451 24-Core Processor
- AMD EPYC 7452 32-Core Processor
- AMD EPYC 7453 28-Core Processor
- AMD EPYC 74F3 24-Core Processor
- AMD EPYC 7501 32-Core Processor
- AMD EPYC 7502 32-Core Processor
- AMD EPYC 7513 32-Core Processor
- AMD EPYC 7532 32-Core Processor
- AMD EPYC 7542 32-Core Processor
- AMD EPYC 7543 32-Core Processor
- AMD EPYC 7551P 32-Core Processor
- AMD EPYC 75F3 32-Core Processor
- AMD EPYC 7643 48-Core Processor
- AMD EPYC 7702 64-Core Processor
- AMD EPYC 7702P 64-Core Processor
- AMD EPYC 7713P 64-Core Processor
- AMD EPYC 7742 64-Core Processor
- AMD EPYC 7763 64-Core Processor
- AMD EPYC 7773X 64-Core Processor
- AMD EPYC 7F72 24-Core Processor
- AMD EPYC 7H12 64-Core Processor
- AMD EPYC 9334 32-Core Processor
- AMD EPYC 9354 32-Core Processor
- AMD EPYC 9634 84-Core Processor
- AMD EPYC 9654 96-Core Processor
- AMD EPYC 9754 128-Core Processor
- AMD EPYC Processor
- AMD EPYC Processor (with IBPB)
- AMD EPYC-Rome Processor
- AMD Opteron(TM) Processor 6212



# Test statistics

data from: 07/04/23 – 09/09/24

- Automated job submission every 3 hours on each PanDA resource via HammerCloud
  - 154 PanDA Resources
  - 250 CPU Models
  - 29112 unique hosts
- Over 300k jobs finished
- Each job: 8 core slot
- Median of job's walltime: 81minutes
  - HEP Score23 configuration with 1 repetition
  - 0.05% of total walltime\_x\_core



# HEP Benchmark Project

## HEPScore

- New High-Energy-Physics specific benchmark
- It has replaced HEP-SPEOC6 in 2023
- Open source
- Support for non-x86 architectures
- Uses the workloads from the HEP experiments and combines them in a single benchmark score

## HEPScore23 (HS23)

- The official HEPscore configuration composed by 7 workloads from 5 experiments (3 ST, 4 MT)
- 1:1 normalization with HS06 for the reference CPU Intel® Xeon® Gold 6326 CPU @ 2.90 GHz (HT=On)
- Runtime ~4h

## HEP Benchmark Suite

- Orchestrator of multiple benchmark (HS23, HS06, SPEC CPU2017)
- Central collection of benchmark results

CPU models per year of release

