Optimisation of ATLAS computing resource usage through a modern HEP Benchmark Suite via HammerCloud and Big PanDA

Natalia Diana Szczepanek[1,*], *Domenico* Giordano[1,**], *Ivan* Glushkov[2], *Gonzalo* Menendez Borge[1], *Alessandro* Di Girolamo[1], *Alexander* Lory[3], and *Ilija* Vukotic[4]

[1]CERN, European Laboratory for Particle Physics, Geneva, Switzerland
[2]University of Texas, Arlington, United States
[3]Ludwig-Maximilians-Universität, Munich, Germany
[4]University of Chicago, United States

Abstract. In April 2023, HEPScore23, the new benchmark based on HEP specific applications, was adopted by WLCG, replacing HEP-SPEC06. As part of the transition to the new benchmark, the CPU corepower published by the sites needed to be compared with the effective power observed while running ATLAS workloads. One aim was to verify the conversion rate between the scores of the old and the new benchmark. The other objective was to understand how the HEPScore performs when run on multi-core job slots, so exactly like the computing sites are being used in the production environment. Our study leverages the HammerCloud infrastructure and the PanDA Workload Management System to collect a large benchmark statistic across 136 computing sites using an enhanced HEP Benchmark Suite. It allows us to collect not only performance metrics, but, thanks to plugins, it also collects information such as machine load, memory usage and other user-defined metrics during the execution and stores it in an OpenSearch database. These extensive tests allow for an in-depth analysis of the actual, versus declared computing capabilities of these sites. The results provide valuable insights into the real-world performance of computing resources pledged to ATLAS, identifying areas for improvement while spotlighting sites that underperform or exceed expectations. Moreover, this helps to ensure efficient operational practices across sites. The collected metrics allowed us to detect and fix configuration issues and therefore improve the experienced performance.

ATL-SOFT-PROC-2025-020
28 January 2025

1 Introduction

The Worldwide LHC Computing Grid (WLCG) provides global computing resources for the storage, distribution, and analysis of data generated by the Large Hadron Collider (LHC). Combining around 1.3 million CPU cores across 164 data centers worldwide, the WLCG is

*e-mail: natalia.diana.szczepanek@cern.ch
**e-mail: domenico.giordano@cern.ch

a cornerstone of modern high-energy physics research. As computational demand increases, CERN needs reliable accounting of the computing resources. Accurate accounting enables experiments and stakeholders to understand available computing resources and utilize them effectively.

The High Energy Physics (HEP) community has agreed to use dedicated benchmarks for reporting computational capacities. In April 2023, the previously used HEP-SPEC06 (HS06) [1] benchmark was replaced by HEPScore23 (HS23) [2]. As part of this transition, it became essential to compare the CPU corepower, which is performance score per CPU core, published by ATLAS [3] sites with the runtime corepower observed while running regular jobs in production environment. One of the primary aims was to verify the conversion rate between the old and new benchmark scores and ensure consistency across sites. Traditionally, sites report their performance values annually, providing weighted averages of different corepower values from various CPU models available at this site. However, until now, there was no direct method to validate the numbers provided by these sites. The infrastructure described in this work is designed to monitor and validate the reported values in a production environment by running the benchmark on the grid. This approach not only verifies the annual performance numbers but also enhances transparency and reliability in resource accounting. It also allow for early detection of incorrect values and fix them as soon as it is possible.

2 Submission Infrastructure

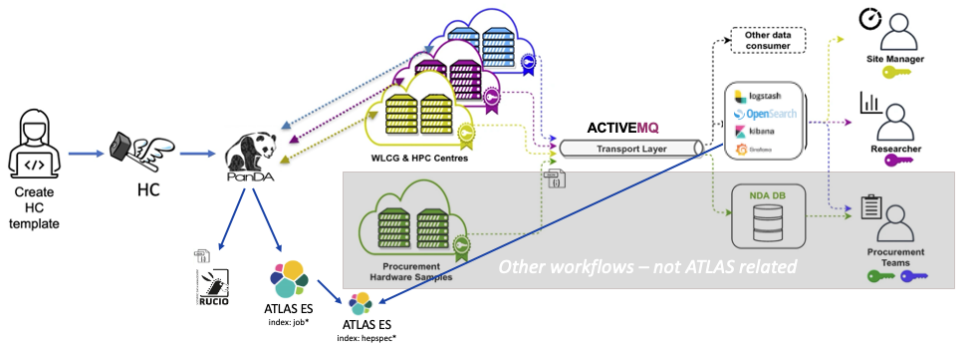


Figure 1: Submission infrastructure includes HammerCloud, PanDA, HEP Benchmark Suite, ActiveMQ, OpenSearch, Grafana, Elasticsearch and Kibana enabling automatic and continuous measurements of runtime corepower of compute resources provided by the WLCG sites to ATLAS.

The submission infrastructure presented in **Figure 1** is an ATLAS adaptation of the workflow described in detail in [4]. HammerCloud (HC) [5], a framework used for testing and benchmarking WLCG resources, is used to dispatch identical jobs at regular intervals via PanDa [6]. To achieve this, a HC template defining the workload, the environment, the submission schedule, and the list of targeted queues is created. *PanDA queue* is a concept on the PanDA Workflow Management System (WMFS) of grouping and configuring a set of resources together needed for processing workflows. Each WLCG Site can include multiple PanDA queues. Every four hours, the job is submitted at each queue. The job runs the enhanced HEP Benchmark Suite [7], executing the HEPScore23 configuration, including

7 workloads from 5 experiments. Upon job completion, the benchmarking results are sent through ActiveMQ [8] to the benchmarking OpenSearch [9] instance and are available for inspection on a dedicated benchmarking Grafana [10] dashboard; simultaneously, PanDA-specific information is sent to the ATLAS Elasticsearch (ES) [11] instance. Subsequently, data from both sources is combined using a Python script and transmitted to the ATLAS ES instance under the `hepspec*` index. The ATLAS ES instance hosts a Kibana [12] dashboard for comprehensive visualization and analysis of the combined data.

3 Data Analysis

Over the past year, a large amount of data has been collected using the infrastructure described in Section 2. **Table 1** provides a statistical summary of the collected data. Each job runs on 8 cores, following the WLCG standards [13]. In each run, performance metrics of each server were collected, along with additional system information, including user-defined metrics such as machine load during script execution, average CPU frequency, memory and swap. These extensive statistics allowed for a detailed analysis of the system performance observed during benchmark runs and enabled comparison with the numbers reported by the sites.

Table 1: Benchmarking jobs statistics

No. #	No. Sites	No. CPU	Walltime [min]	% of total walltime
187045	139	251	81	0.06

3.1 Declared and Runtime Corepower

The corepower of a server has historically been defined as the HS06 per core, serving as a key metric to evaluate computing capabilities based on specific hardware. With the transition from HS06 to HS23 in April 2023, the corepower metric is now represented as HS23 per core. To ensure a smooth transition, it was initially agreed that corepower values would be converted one-to-one for each site. Currently, HS23 is the WLCG standard, and sites are expected to provide updated corepower values.

The corepower value reported by a site, referred to here as *declared corepower*, represents the weighted average of the corepower of different CPU models available at given queue on a site. A single site may consist various hardware resources that are grouped into different queues. The objective of this study was to compare the declared corepower values in ATLAS-CRIC [14] with the performance observed during job execution, hereafter referred to as the *runtime corepower*. To calculate runtime corepower, the weights of different CPU models available at each queue were first derived using all available data from PanDA jobs, leveraging the `walltime_x_core`, defined as the product of job walltime and the number of cores. The weight of each CPU model (x) at a given queue was computed as defined in **Eq. 1**.

$$w_x = \frac{\sum_{i \text{ on jobs}} \text{walltime\_x\_core}_i^x}{\sum_{x \text{ on CPU}} \sum_{i \text{ on jobs}} \text{walltime\_x\_core}_i^x} \quad \text{(Eq. 1)}$$

Subsequently, the weighted average corepower for each queue was calculated using only CPU models for which benchmarking data was available, as presented in **Eq. 2**.

$$\text{corepower\_runtime}^{\text{queue}} = \frac{\sum_{x \text{ on CPU}} w_x \cdot \text{corepower\_runtime}_x^{\text{queue}}}{\sum_{x \text{ on CPU}} w_x} \quad \text{(Eq. 2)}$$

For the analysis, only queues with complete weights data were included. The *relative change* in corepower for a given queue (q) was then defined as the ratio of runtime corepower to declared corepower, minus one, as presented in **Eq. 3**.

$$\text{relative change} = \frac{\text{corepower\_runtime}_q}{\text{corepower\_declared}_q} - 1 \quad \text{(Eq. 3)}$$

It is important to note that systematic uncertainties are inherent in the measurements, and these were carefully considered in this study. Sources of uncertainty include variations in HS23/HS06 scaling [15], fluctuations in runtime HS23 probes, dynamic CPU configurations as switching between hyper-threading enabled and disabled without reporting, performance variations due to system load, and errors in weight calculations critical to the analysis. To address these uncertainties, a discrepancy threshold of ±25% was established. Only relative changes exceeding this threshold were classified as significant discrepancies, so those ones where numbers reported by queues differ substantially from measured runtime corepower.

4 Analysis of Relative Change Corepowers

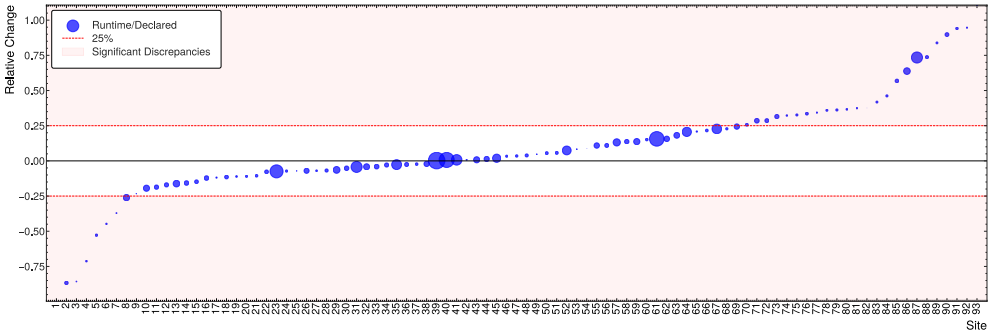
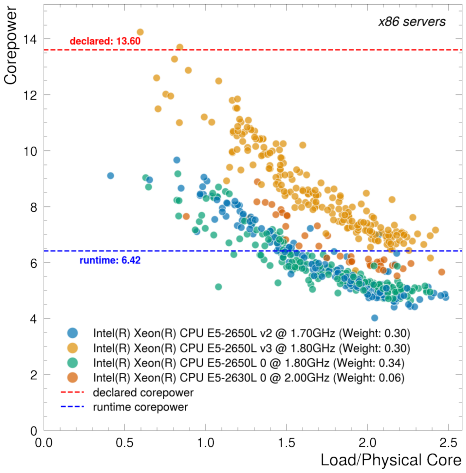


Figure 2: Relative Change following **Eq. 3** for PanDA queue, grouped by site. More than one data point per site indicates that the site provides more than one PanDa queue. The shadowed region highlights critical discrepancies. The size of the marker is proportional to site contribution level calculated based on its walltime_x_core.

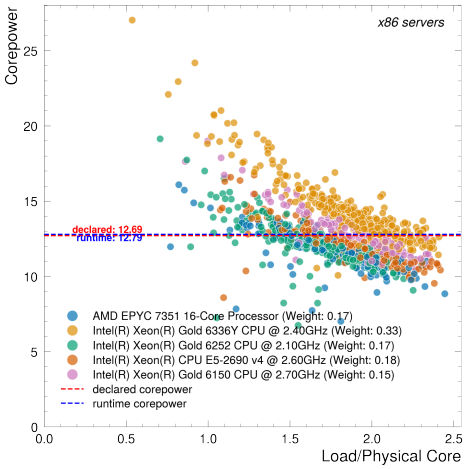
Measurements carried out throughout 139 sites enabled a comprehensive corepower analysis for 72 of them, where complete weight data was available, as shown in **Figure 2**. The different sites were anonymised and shown on the `x-axis`, with the relative change on the `y-axis`. The conservative acceptance threshold of 25% highlights the large discrepancies. A relative change of zero represents a perfect match between the declared and the runtime corepower, it can be seen that most of the points are within the treshold range. Positive relative change indicate underreporting of the declared corepower by the queue on given site, whereas negative values suggest overreporting. From **Figure 2**, it can be observed that 32% of the analyzed sites show critical discrepancies. Two hypotheses were considered to explain these discrepancies: (a) the queues at these sites are either underloaded or overloaded, and/or (b) the declared corepower values are inaccurate.

4.1 Corepower vs Load

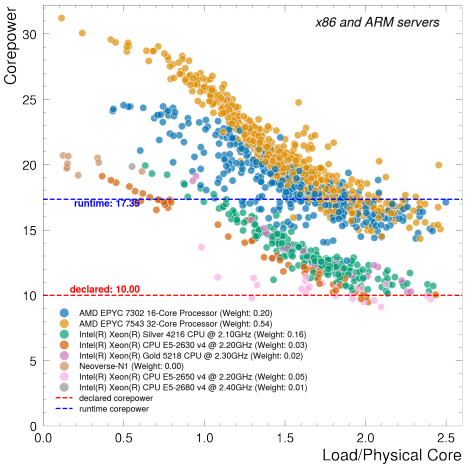
To test the first hypothesis, a corepower versus load analysis was conducted. The average load information during the benchmark execution was retrieved and correlated with the measured performance. The load/physical core metric is defined as the measured load divided by



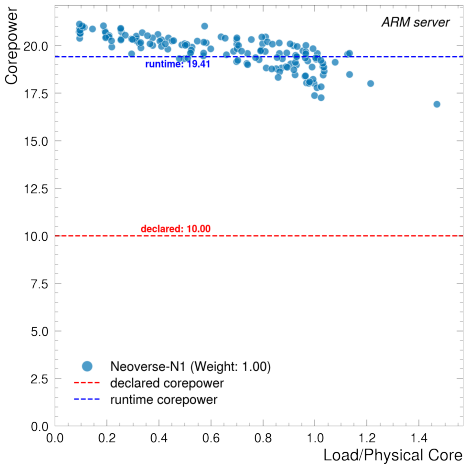
(a) Queue with negative relative change



(b) Queue with neutral relative change



(c) Queue with positive relative change



(d) Queue with positive relative change

Figure 3: Corepower vs load/core correlation of 4 typical cases of queues with: (a) negative relative change, (b) neutral relative change, (c) positive relative change, (d) positive relative change, ARM server. The dashed red and blue lines represent the declared and runtime corepowers respectively. Each color marker identifies a different CPU model on a given queue.

the number of physical cores. As a consequence a fully loaded server with hyper-threading disable or enable will have a load/physical core value of 1 or 2 respectively. In **Figure 3**, the

plot reveals a correlation between machine load/physical core and performance: as the load increases, the performance decreases. This is being especially visible in the range from 1 to 2 of load/physical core.

Four examples were selected to illustrate typical behaviors with respect to the relative change results. **Figure 3a** shows a queue where the declared corepower exceeds the runtime corepower. This queue was fully utilized, but the CPU models in use are from 2012. In this case, the scaling ratio between HS23 and HS06 is 0.7, as taken from Figure 4 in [2], which explains the detected negative relative change. This shows that the values for very old queues should be updated as soon as possible to better plan resource utilization. The second case, shown in **Figure 3b**, represents one of the largest queues, where the relative change is close to zero. Here, the site demonstrates efficient utilization across all available servers, with no discrepancies observed.

The **Figure 3c** provides a third example suggesting that, in many cases, the server is not fully loaded, which could be one of the reasons for the observed higher runtime corepower compared to the declared value. However, this does not appear to be the sole cause. As seen in the figure, there are two servers whose performance match precisely the declared corepower value of 8.74. This observation suggests that the declared corepower might not have been updated to reflect the presence of other, more modern servers in the queue. Instead, it appears to rely on outdated values associated with older hardware, failing to account for the contribution of additional servers with higher performance capabilities.

The fourth example, presented in **Figure 3d**, is a special case involving a single CPU model on a given queue, Neoverse-N1 based on ARM architecture. Unlike the x86 servers, weaker correlation between load and performance is observed. However, a significant discrepancy is evident — a factor of nearly two between the runtime corepower and the declared corepower, favoring the runtime measurement.

The analysis shows that even if there is correlation between the average load of a server and its performance, it is not the only cause of the discrepancies. But it is important to notice that if all servers in a queue would have been consistently underloaded, a positive relative change would be observed, meaning the runtime corepower would appear higher than the declared value. On the other hand, if all servers in a queue would have been consistently overloaded, what is defined as a load above 2 for systems with hyper-threading enabled or above 1 for those with hyper-threading disabled — a negative relative change would be observed, indicating the runtime corepower would appear lower than the declared value, if the declared value would be set correctly.

Another consideration related to the load involves the method of calculating runtime corepower, which is based on the average performance across the entire load range. It was important to verify whether this approach could be responsible for the observed behavior of different queues. To address this, a separate analysis was conducted using only measurements taken when servers were fully loaded. It was assumed that this approach would reduce discrepancies in relative change to within the established threshold.

The results, presented in **Figure 4**, show fewer data points, as fewer servers were fully loaded during the measurements. While this approach did slightly reduce discrepancies in some of the cases, it did not bring them entirely within the expected threshold. This finding confirms that calculating performance across the full load range is valid and that the uncertainty associated with this method is minimal and acceptable for the analysis.

4.2 Analysis of Declared Corepower Data Sources

The declared corepower values used by ATLAS are sourced from the ATLAS-CRIC, where they are directly being reported by site administrators from each site or loaded from other

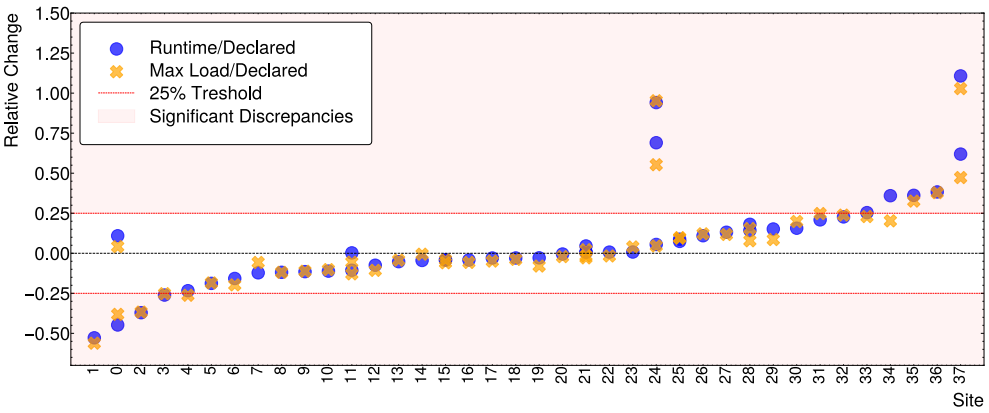


Figure 4: Relative change for different sites calculated when servers were fully loaded

data sources. A detailed analysis of these values revealed that 80% of the PanDA queues were cloned from pre-existing ones, with 50% inheriting their corepower values from the original queues. This practice has significant implications for multiple queues employing newer architectures, such as x86 and ARM-based queues shown in **Figure 3c** and **Figure 3d**, where the discrepancy with the outdated declared corepower is larger.

ATLAS-CRIC serves as the central place, where site administrators can update these values, but it is also an integration point for multiple data sources. During this analysis, discrepancies between the various data sources were also identified and reported accordingly, with corrective actions currently underway.

Considering the combined contributions from all sites and weighting them by their `walltime_x_core`, the discrepancies between the declared corepower and the runtime corepower values result in an overall 6% advantage in favor of the runtime performance. While this result is positive, it highlights the need to update declared corepower values. Failure to do so could increasingly affect decision-making processes that rely on accurate reported data.

5 Conclusions

The automated HEPScore23 submission infrastructure via PanDA and HammerCloud provides a novel approach, previously unavailable for validating the corepower values declared by sites, offering a reliable cross-check for the official accounting system. By leveraging the HEP Benchmark Suite, additional system metrics were collected, enabling a detailed analysis of the runtime versus the declared corepower values. The observed relative changes revealed significant discrepancies for several queues, highlighting differences between declared and actual runtime performance. These discrepancies have the potential to impact decision-making processes, leading to inefficiencies and complications in resource allocation.

A systematic analysis has shown that outdated declared corepower values are a primary source of these discrepancies, underlining the importance of regular updates in the data source. While server load does influence performance, it alone does not fully justify the observed differences. The presented methodology for measuring the runtime corepower proved effective in identifying queues on sites with large discrepancies, enabling quick reporting and resolution. This approach not only ensures better accuracy and transparency in resource accounting but also enhances operational efficiency and fosters better decision-making for the ATLAS experiment and WLCG at large.

References

[1] M. Michelotto, M. Alef, A. Iribarren, H. Meinhard, P. Wegner, M. Bly, G. Benelli, F. Brasolin, H. Degaudenzi, A.D. Salvo et al., *A comparison of HEP code with SPEC1 benchmarks on multi-core worker nodes* (2010), https://dx.doi.org/10.1088/1742-6596/219/5/052009

[2] Giordano, Domenico, Barbet, Jean-Michel, Boccali, Tommaso, Borge, Gonzalo Menéndez, Hollowell, Christopher, Innocente, Vincenzo, Lampl, Walter, Michelotto, Michele, Meinhard, Helge, Ondris, Ladislav et al., *HEPScore: A new CPU benchmark for the WLCG* (2024), https://doi.org/10.1051/epjconf/202429507024

[3] ATLAS Collaboration, *The ATLAS Experiment at the CERN Large Hadron Collider* (2008), https://dx.doi.org/10.1088/1748-0221/3/08/S08003

[4] N. Szczepanek, D. Britton, A.D. Girolamo, E. Ketele, I. Glushkov, D. Giordano, et al., *HEP Benchmark Suite: Enhancing Efficiency and Sustainability in Worldwide LHC Computing Infrastructures* (2024), 2408.12445, https://arxiv.org/abs/2408.12445

[5] CERN, *HammerCloud - Distributed Analysis Testing Framework* (2024), accessed on: 26 November 2024, https://hammercloud.cern.ch/

[6] Maeno, T., Alekseev, A., Barreiro Megino, F.H. et al., *PanDA: Production and Distributed Analysis System* (2024), https://doi.org/10.1007/s41781-024-00114-3

[7] CERN HEP Benchmarks Project, *HEP Benchmark Suite* (2024), accessed on: 26 November 2024, https://gitlab.cern.ch/hep-benchmarks/hep-benchmark-suite

[8] Apache Software Foundation, *Apache ActiveMQ* (2024), accessed on: 26 November 2024, https://activemq.apache.org/

[9] OpenSearch Project, *OpenSearch* (2024), accessed on: 26 November 2024, https://opensearch.org/

[10] G. Labs, *Grafana* (2024), accessed on: 26 November 2024, https://grafana.com/

[11] Elastic, *Elasticsearch* (2024), accessed on: 26 November 2024, https://www.elastic.co/elasticsearch

[12] Elastic, *Kibana* (2024), accessed on: 26 November 2024, https://www.elastic.co/kibana

[13] Worldwide LHC Computing Grid (WLCG), *Worldwide LHC Computing Grid: Technical Standards* (2024), each job runs on 8 cores, following WLCG standards. Specific citation may vary based on context., https://wlcg.web.cern.ch/

[14] CERN ATLAS Collaboration, *Atlas cric* (2024), accessed on: 26 November 2024, https://atlas-cric.cern.ch

[15] D. Giordano, M. Alef, L. Atzori, J.M. Barbet, O. Datskova, M. Girone, C. Hollowell, M. Javurkova, R. Maganza, M.F. Medeiros et al., *HEPiX Benchmarking Solution for WLCG Computing Resources* (2021), https://link.springer.com/misc/10.1007/s41781-021-00074-y