HEP Benchmark Suite

The centralized future of WLCG benchmarking

Gonzalo Menéndez Borge[1,*]

[1]CERN, Meyrin, Switzerland

Abstract. The HEPiX Benchmarking Working Group has devised a new HEP-specific benchmark, HEPscore23. This benchmark is an instance of the underlying benchmarking tool that is HEPscore, also created by the Working Group during this endeavor. HEPscore can be set up to run different HEP Workloads, autonomous production applications sourced from various HEP experiments. Through study and analysis, a subset of those workloads was elected, the minimal set that best represented HEP applications overall. To streamline the benchmarking process, this benchmarking framework includes the HEP Benchmark Suite, which facilitates the execution of HEPscore and other benchmarks, such as HEP-SPEC06, SPEC CPU 2006, and DB12. This paper elucidates the rationale behind the benchmark, and the framework created to develop it, delineates the key design considerations, diving into the motivations behind it and the flexibility and advantages that is offers as a replacement of HS06 in the WLCG and HEP communities.

1 Introduction

In the realm of High-Energy Physics (HEP) research, the HEP-SPEC06 (HS06) benchmark [1], based on SPEC CPU 2006 [2], has been the most extensively used benchmark over the last decade. As such, it was the official benchmark of the Worldwide LHC Computing Grid (WLCG) [3] community to estimate the performance of a computing server.

Starting in 2017, performance discrepancies between HS06 and the workloads from different experiments become increasingly apparent [4]. Consequently, there is a growing need for a new benchmark that can more accurately represent today's HEP workloads and is capable of fully benchmarking the cutting-edge servers on which they are executed.

To that end, the HEPiX Benchmarking Working Group [5] began looking into the matter and resolved to create a new benchmark based on the current HEP experiments' workloads; HEPscore. When proposed to the WLCG Management Board, and at their request, a HEPscore Deployment Taskforce was created. Its aim was to develop this new benchmark, as well as discuss with the experiments and sites about HEPscore as a substitute for HS06.

It is in this transition stage that the HEP Benchmark Suite is devised. This tool is a meta-orchestrator of multiple benchmarks, that collects benchmark results in an organized manner, alongside other useful information provided by its plugins. Optionally, the Suite may send the results into a messaging system for its storage in different databases, which greatly eases its analysis, monitoring, and plotting.

[*]e-mail: gonzalo.menendez.borge@cern.ch

2 HEPscore

2.1 Motivation

By the retirement of SPEC 2006, in 2018, benchmarked servers were showing discrepancies between their HS06 score and their performance when executing HEP workloads. The explanation lies not only in the retirement of the underlying benchmark but also in all of the changes that have taken place in the community since its inception.

The rise of multi-threaded and multi-process applications, vectorized instructions (AVX), and new architectures (ARM) alongside containerization, and hardware acceleration (GPUs) all contributed to its accuracy loss for the newer workloads run on the latest hardware. To top it all, licensing was also a showstopper in certain scenarios, given that the SPEC CPU® 2006 copyrighted license is required to run HS06.

HEPscore was therefore devised to address all of these points, always with modularity and flexibility in mind to be able to adapt in this ever-changing environment for as long as possible. Yet there is one more crucial point to be addressed: the benchmark should reflect the behavior of HEP applications. General purpose benchmarks like SPEC 2006, and therefore HS06 based on it, utilize CPU resources differently than HEP applications, as shown by various studies [6, 7]. Figure 1 demonstrates this.

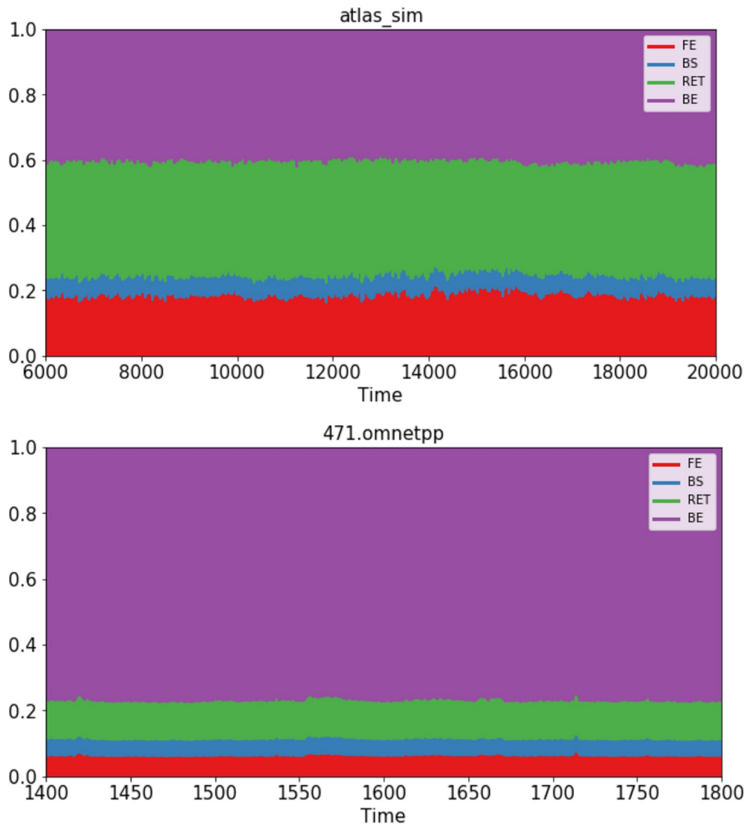

Figure 1. Percentage of CPU cycles per outcome as a function of the application's running time for the ATLAS simulation (top) and the HS06 omnetpp application (bottom). This plot was obtained from Ref. [7]

These plots portray the percentages of the CPU cycles spent in front-end stalls (FE), back-end stalls (BE), in bad speculation (BS), or in completed executions retired from the execution stack (RET), as a function of the application's running time, for the ATLAS simulation (top) and the HS06 omnetpp application (bottom). The initialization and finalization phases of the execution have been excluded.

Thus, even newer versions of generic benchmarks, while they may solve some of the initial issues, will still fail to reflect the behavior of the latest HEP applications. This vital aspect finalizes consolidating the need for a new HEP-specific benchmark [8], HEPscore.

2.2 HEP Workloads

In order to represent in a faithful way the behavior of present HEP applications, the underlying workloads that compose HEPscore must be HEP-based applications. To that end, the Benchmarking Working Group has worked in close relation with software experts from different experiments across the WLCG to create HEP workloads [9].

These workloads are packaged into standalone container images, ensuring they have all the necessary software and data, and eliminating dependencies on remote services. The containers are built in the HEP-Workloads GitLab repository, by harnessing Giltab's continuous integration (CI) capabilities. Containers vary in size from 1 to 4 GB and are hosted on CERN's GitLab container registry.

The experts need to prepare an orchestrator script to manage the environment, application execution, error handling, and result extraction, producing a JSON-format benchmark as the one shown in figure 2.

```
{...,
 "report": {
           "wl-scores": {"gen-sim": 18.8476,
                         "gen": 95.9261,
                         "sim" : 24.2931 },
            "log": "ok"},
...
}
```

Figure 2. Structure of a HEP Workload's JSON report

All HEP-workloads are event-based, with scores representing the processed events per second. Since the orchestrator can run multiple independent copies of an application, available server cores can be utilized efficiently.

Each container comprises a configuration file and input files for event processing. The number of events to process can be adjusted by the users and, therefore, the execution duration. Furthermore, given that the orchestrator and HEP applications run unprivileged, they are suitable for high-performance computing (HPC) sites with security constraints.

2.3 Design

HEPscore is a utility used to manage the execution of multiple HEP-Workloads containers and determine the benchmark score for a specific computer server. Similar to HS06, HEP-Workloads are executed multiple times on a server, and the application score is the median value of successful runs to minimize fluctuations. These scores are normalized relative to a reference server, and a weighted geometric mean is used to combine them into a single benchmark value, allowing for adjustments using weights, unlike HS06.

The HEPscore for a given set of n HEP-Workloads on server m relative to the reference server m_R can be expressed as:

$$HEPscore(m, mR; \boldsymbol{p}) = \alpha \prod_{i=1}^{n} \left(\frac{a_i(m; \boldsymbol{p})}{a_i(m_R; \boldsymbol{p})} \right)^{\frac{wi}{\sum_{j=1}^{n} w_j}} \qquad (1)$$

Here, n represents the number of workloads, p stands for the configuration parameters of both HEPscore and HEP-Workloads, and α serves as a normalization factor, ensuring that the value falls within a practical numerical range.

As the formula considers various configuration parameters, it remains valid across different system architectures and accelerators, provided the HEP-Workloads are built for these systems. HEPscore can execute container images using Docker or Singularity container engines and includes options for Singularity user namespace-based container execution and clearing container image caches to prevent benchmark failures.

As mentioned in the previous section 2.2, HEPscore does not require specific workload details but expects workloads to be containerized, have an orchestrator script as an entry point, accept command line arguments for configuration, and produce a JSON-format report file with workload scores and success/failure flags.

Configuration for HEPscore is done using a YAML file, and it generates JSON reports containing overall scores, individual workload results, runtime parameters, execution times, and system metadata. Different benchmarks can be run by modifying configuration parameters and including different sets of HEP-Workloads, with each parameter set having a unique hash identifier.

2.4 HEPscore23

While HEPscore as such has been devised as a flexible and modular tool, in order to establish a dependable benchmark to supplant HS06, a reliable and representative subset of all the HEP workloads available needed to be identified. It is of utmost importance that the outcomes of the underlying workloads, and thus that of HEPscore, can be consistently reproduced using a specific configuration file and input data.

Achieving this necessitates processing the same event sequence in multiple measurements. This is accomplished by configuring the application to fix certain parameters that impact the sequence, including the type of physics event to simulate, the random number seed to utilize, and the number of events processed per thread.

Table 1. HEPscore 23 workload composition

HEP-Workload	Version	Size [GiB]	Threads	Events/thread
ATLAS GEN	v2.0	10.66	1	200
ATLAS RECO	v2.0	9.75	4	100
CMS GEN-SIM	v1.0	3.07	4	20
CMS RECO	v1.0	3.32	4	50
LHCb SIM	v1.0	1.80	1	10
Belle II GEN-SIM-RECO	v2.0	1.14	1	50
ALICE DIGI-RECO	v2.1	4.26	4	10

The HEPscore Deployment Task Force performed extensive analyses on the matter and, after careful study, converged on a set of 7 benchmarks, available both for x86 and ARM, with its fixed configuration which was called HEPscore23, which can be seen in table 1.

As of April 2023, HEPScore23 has been accepted as the new official benchmark of the WLCG and it is progressively replacing HS06. The accounting migration procedure had been officially endorsed by the WLCG Management Board previously, in December 2022.

3 HEP Benchmark Suite

The significant achievement of HEPscore23 was made possible through the HEP Benchmark Suite, which was developed in conjunction with HEPscore. Just like HEPscore orchestrates workloads, the Suite orchestrates benchmarks. It is capable of executing various benchmarking utilities, including HEPScore, HS06, SPEC2017, and DB12, while maintaining a lightweight and highly portable design. This Python 3-based package minimizes dependencies, ensuring ease of deployment and packaging.

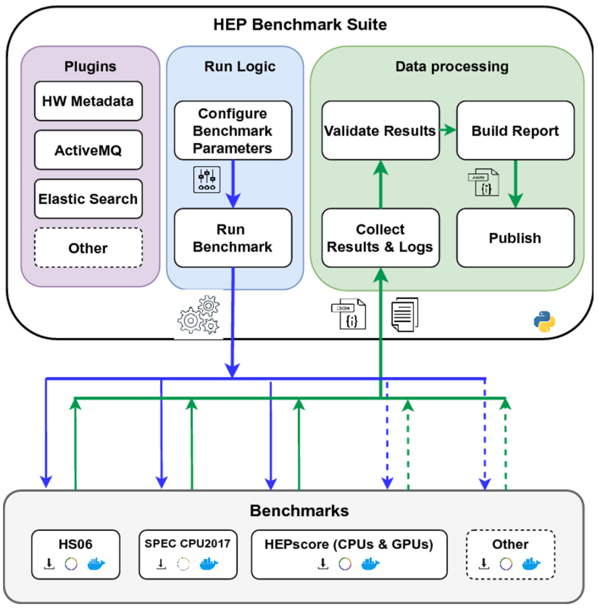

Figure 3. HEP Benchmark Suite component diagram

The Suite's architecture, depicted in Figure 3, comprises three primary components: Plugins, Run Logic, and Data Processing blocks. Plugins encompass additional functionalities such as hardware metadata and interfaces with external services. The Run Logic component executes benchmarks using options specified in a single configuration file, allowing customization of parameters like benchmark selection, core allocation, and user-defined tags. The Data Processing block handles the collection and processing of benchmark data, ultimately generating the final report.

It is worth noting how the benchmarks are decoupled from the Suite, enabling flexibility in modifying the benchmark list or introducing new ones without affecting the Suite's core functionality. For executing HS06 and SPEC CPU 2017 benchmarks, the Benchmarking Working Group provides a container image designed to orchestrate SPEC executions, ensuring standardization in compilation flags and reporting. However, it is essential to highlight that this container image does not include HS06 and SPEC2017 due to licensing constraints, requiring users to provide their own access to SPEC installations and licenses.

All benchmark results, along with their runtime conditions, are stored in a structured JSON format. Host metadata uniquely identifies each host and includes user-defined tags, as well as software and hardware metadata. The list of user-defined tags is the sole JSON object that users can modify to include additional information and enhance metadata. Other JSON objects are automatically populated by the Suite. This modular JSON structure facilitates future schema expansion, with a dedicated report field used for versioning.

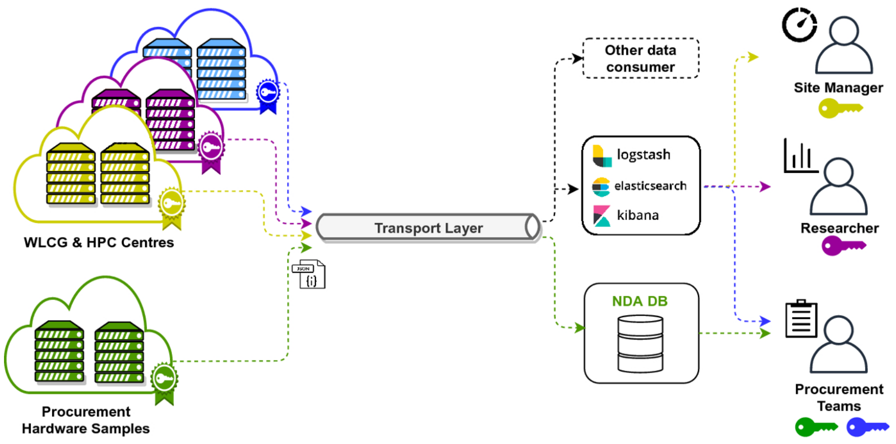

Figure 4. HEP Benchmark Suite optional report flow

Upon completion of benchmarking, the Suite saves the report file locally. Additionally, it offers the option to publish results to a remote ActiveMQ message queue, enabling multiple clients with varying access privileges to access the data (Figure 4). Site managers, for instance, can benchmark their clusters and subsequently review the results using visualization frameworks. This approach also facilitates the integration of benchmark results into researchers' data analytic solutions.

https://doi.org/10.1051/epjconf/202429507023

3.1 Plugins

As shown in figure 3, the suite counts on a number of configurable plugins for various purposes, from collecting additional metadata to sending the results to different endpoints. Regarding the ones focused on retrieving metadata, we could classify them into two kinds based on the information they retrieve.

The first type would be the static plugins, which as their name implies gather static data; i.e. data that do not change over the course of the run. This could include the hostname or CPU model, for instance. Opposed to them, dynamic plugins collect data that changes over the course of the execution. In this category would fit plugins aimed at measuring the CPU usage, memory consumption, et cetera.

As for the latter, a generic dynamic plugin, the CommandExecutor, has been built with adaptability in mind. As is the case with the Suite and HEPscore, it is highly modular and configurable, allowing a great variety of measurements by changing only its configuration.

```
plugins:
  CommandExecutor:
    metrics:
      load:
        command: uptime
        regex: 'load average: (?P<value>\d+.\d+),'
        unit: ''
        interval_mins: 0.5
      used-memory:
        command: free -m
        regex: 'Mem: *(\d+) *(?P<value>\d+).*'
        unit: MiB
        interval_mins: 0.5
      used-swap-memory:
        command: free -m
        regex: 'Swap: *\d+ *(?P<value>\d+).*'
        unit: MiB
        interval_mins: 0.5
```

Figure 5. HEP Benchmark Suite optional report flow

Figure 5 illustrates how to configure the CommandExecutor to measure different metrics. In particular, the configuration in this example collects load, RAM memory, and swap memory data.

The command to be run is configurable and is required alongside the regex to extract the value from the output command and its unit (which can be empty). Optional fields exist to further configure the instances of the plugin, like the interval in which it will be run. This opens the door to an infinity of possible measurements, including energy consumption both of CPUs and GPUs.

4 Summary

This paper has detailed the efforts of the HEPiX Benchmarking Working Group to find a suitable replacement for HS06 after it became outdated. This led to the necessity of a new flexible, modular and HEP-based benchmark, HEPscore, and a set of HEP applications (HEP Workloads).

The HEPscore deployment task force was created to further work on this project, working with experiments all over the WLCG to create the underlying workloads. HEPscore itself was strategically designed to consolidate various profiling metrics into a single benchmark score. Its design demonstrates adaptability for both Grid and HPC facilities and exhibits potential for future expansion into heterogeneous environments.

In parallel, the HEP Benchmark Suite was developed to orchestrate the execution of numerous benchmarks with added functionalities thanks to its plugins. Its power was harnessed to carry out the studies to compare HS06 to different HEP configurations, so as to find the one the would become its replacement.

The fruit of these studies was HEPscore23, the new official benchmark of the WLCG. Now, with the official support from the organization, the HEPiX Benchmarking Working Group is aiding in the migration process, while continuously developing and extending HEPscore, the HEP Workloads and the HEP Benchmark Suite.

References

[1] Michelotto M et al (2010) **A comparison of HEP code with SPEC1 benchmarks on multi-core worker nodes**. J Phys Conf Ser 219:052009. https://doi.org/10.1088/1742-6596/219/5/052009

[2] Henning JL (2016) **SPEC CPU2006 benchmark descriptions**. SIGARCH Comput Archit News 34:1. https://doi.org/10.1145/1186736.1186737

[3] WLCG (2023) **The Worldwide LHC Computing Grid**. http://wlcg.web.cern.ch. Accessed 20 Jul 2023

[4] Charpentier P (2017) **Benchmarking worker nodes using LHCb productions and comparing with HEP-SPEC06**. J Phys Conf Ser 898:082011. https://doi.org/10.1088/1742-6596/898/8/082011

[5] **HEPiX Benchmarking Working Group** (2023). https://w3.hepix.org/benchmarking.html. Accessed 25 Jul 2023

[6] Muralidharan S, Smith D (2019) **Trident: an automated system tool for collecting and analyzing performance counters**. EPJ Web Conf 214:08024. https://doi.org/10.1051/epjconf/201921408024

[7] Giordano D, Santorinaiou E (2020) **Next generation of HEP CPU benchmarks**. J Phys Conf Ser 1525:012073. https://doi.org/10.1088/1742-6596/1525/1/012073

[8] Giordano D (2017) **Benchmark Working Group Update. WLCG Workshop in Manchester**. https://indico.cern.ch/event/609911/contributions/2620190/attachments/1480455/2295576/WLCG_Workshop_2017_benchmarking_giordano.pdf. Accessed 14 Aug 2023

[9] **HEP-Benchmarks Repository** (2023). https://gitlab.cern.ch/hep-benchmarks Accessed 5 Sep 2023