Benchmarking and Energy Studies with HEPScore

Openlab Technical Workshop 2025

David Southwick *on behalf of* HEPiX Benchmarking Working Group

Benchmarking at CERN

Computing at CERN has been dominated by x86 CPU operations for most of the last decades, but this environment is rapidly changing:

- Specialized compute devices (accelerators)
- Specialized architectures (registers, memory, extensions)

Necessitating a paradigm shift in software models to leverage **aggressive offloading to the most efficient device**

To understand this complex environment, new tooling was required to provide the flexibility to account many diverse devices in this increasingly fragmented and rapidly changing field

CERN Prévessin Data Center

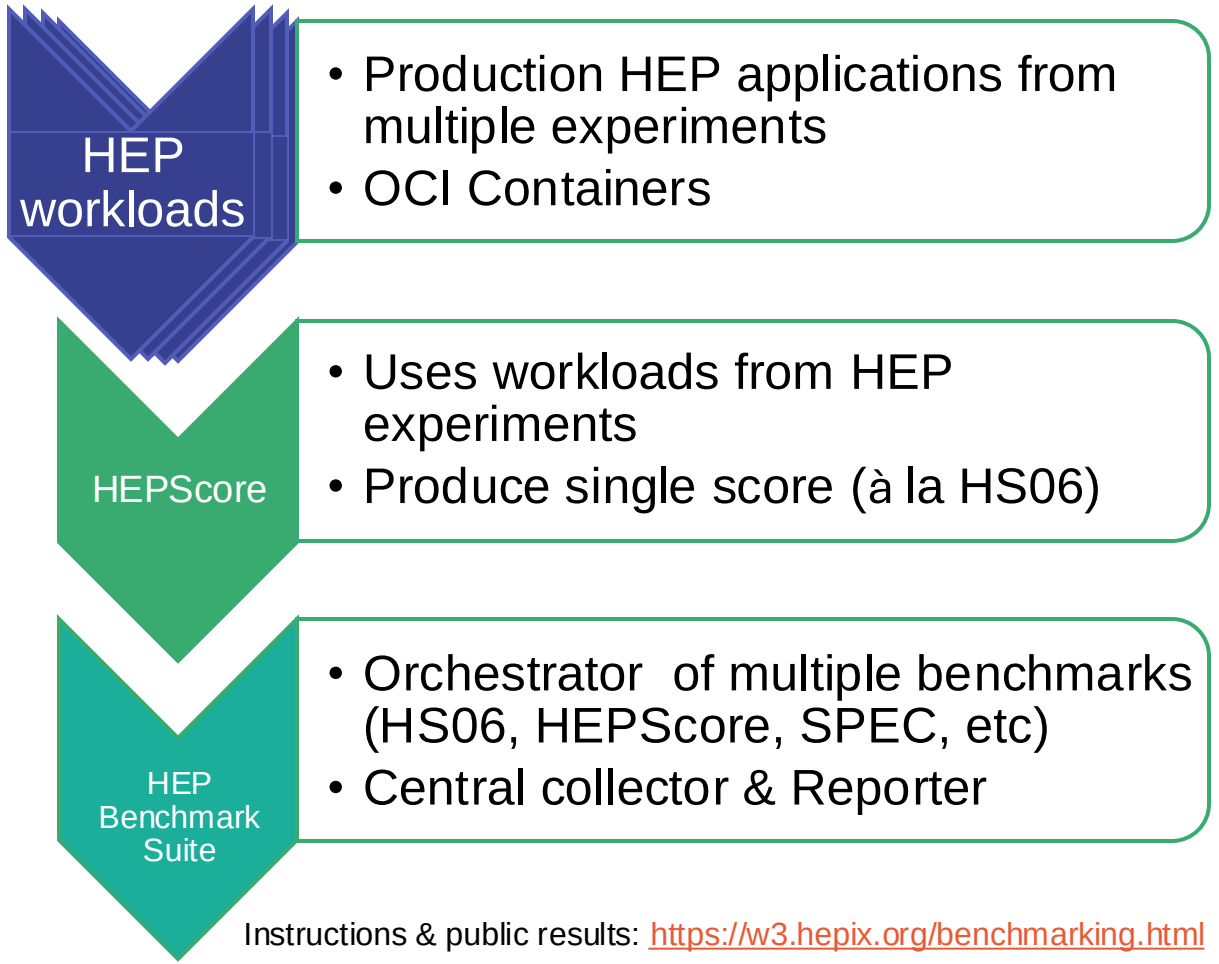
Benchmarking Overview

HEPScore: The next generation of benchmarking for the WLCG

- Intentional “single score” result (à la SPEC)
- Composable/updateable containerized workload array
- Extensible to any hardware that can run Python + containers

HEPScore ratified in April 2023 by the WLCG HEPScore Deployment Task Force as a replacement for HEPspec06

- Now fully replaced HEPspec06



Instructions & public results: https://w3.hepix.org/benchmarking.html

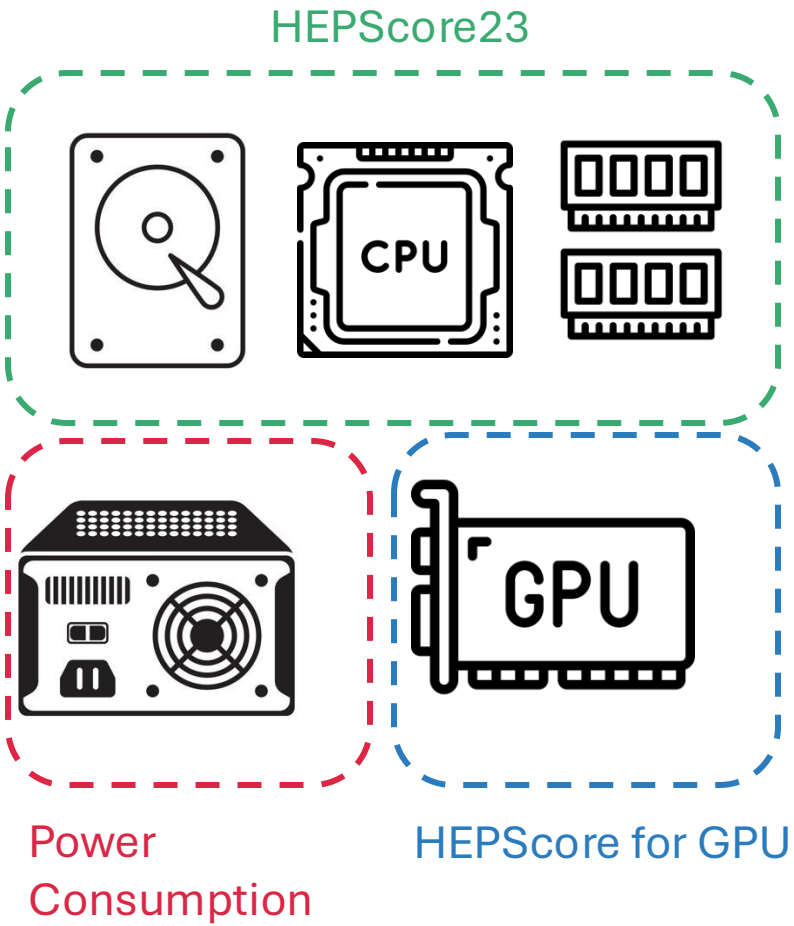
HEPScore developments

Mandate: Define and maintain a consistent, reproducible HEP benchmark

Accurately representing HEP computing needs:
- Based exclusively on production HEP workloads
- Evolve with requirements of HEP community

Many developments of the last two years focusing on power consumption and GPUs

Upcoming major releases with new capabilities:
- HEP Benchmark Suite v3.0
- HEPScore v2.0



Collection and analysis

Expanded capabilities of whole system benchmark via plugins:

- Run parallel to benchmarks
- Same modular YAML configuration
- Flexible adaptation based on needs / setup
- Included in automated reporting

Example of Plugin 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: ''
```

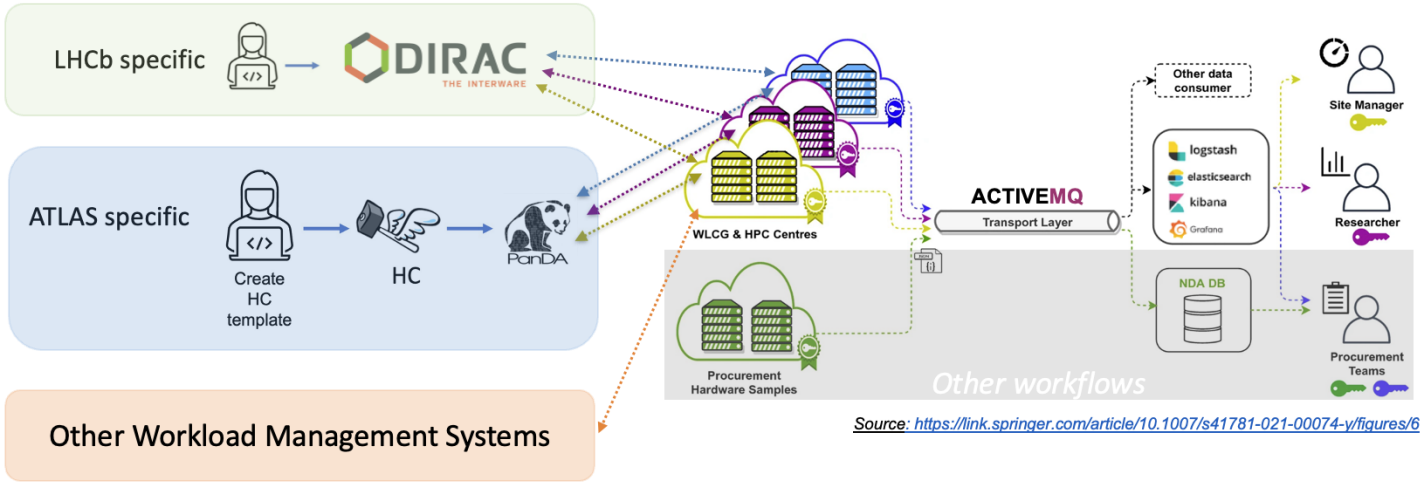
Metric Report


```
"cpu-frequency": {↔},
"used-memory": {↔},
"load": {↔},
"used-swap-memory": {↔},
"power-consumption1": {
    "start_time": "2024-12-03T07:46:29.966361Z",
    "values": [↔],
    "end_time": "2024-12-03T09:03:29.957016Z",
    "config": {
        "regex": "Instantaneous power reading:\\s*(
        "unit": "W",
        "aggregation": "sum",
        "interval_mins": 1,
        "command": "ipmitool dcmi power reading"
    },
    "statistics": {
        "min": 146,
        "median": 432,
        "q25": 425.5,
        "max": 445,
        "mean": 394.7435897435897,
        "q95": 441.15,
        "q75": 437,
        "q85": 439
    }
},
"status": "success"
```

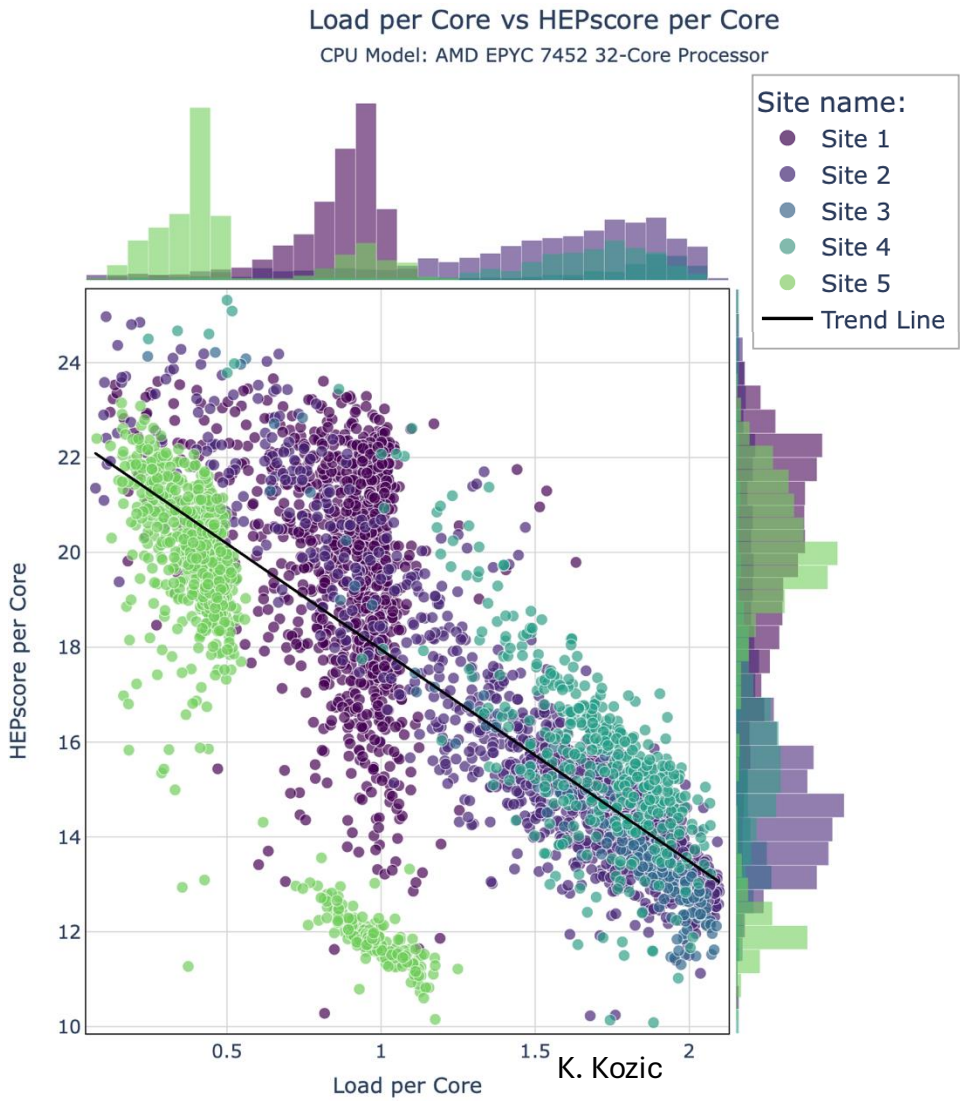
Collection and analysis

HEP Benchmark Suite supports expanded automated reporting:

- Report submission via AMQ and OpenSearch
- Enables scale studies, site performance, monitoring
- Data available in a public table



Source: https://link.springer.com/article/10.1007/s41781-021-00074-y/figures/6



K. Kozic

Heterogeneous Execution

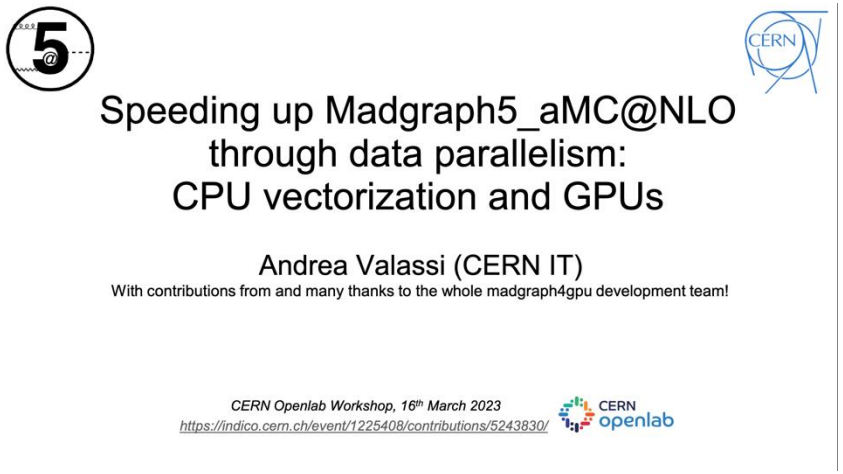
"HEPscore23" comprises a set of 7 production workloads:

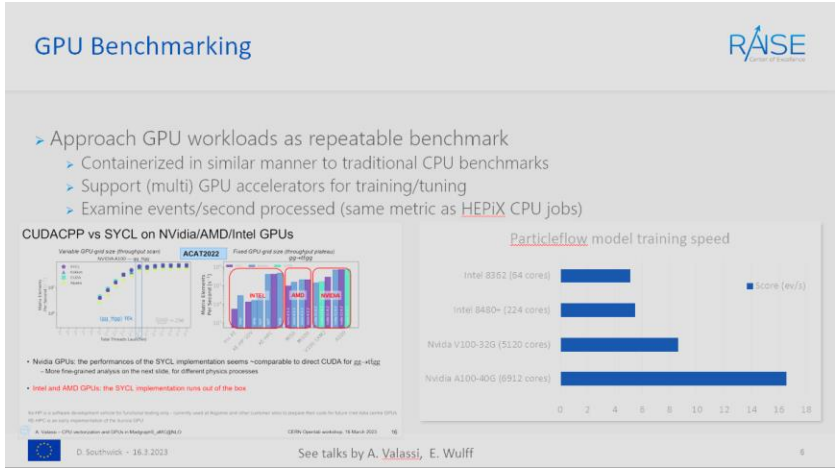
- Available for x86_64 and **aarch64**

HEPscore for GPUs:

- MadGraph4GPU Event Generator (CPU vectorization, GPUs)
- CMS HLT (GPUs)
- MLPF (CPU vectorization, GPUs)
- Flowsim (CPU vectorization, GPUs)

HEP Benchmark Suite includes plugins to measure energy consumption, load, CPU frequency; enabling heterogenous efficiency studies



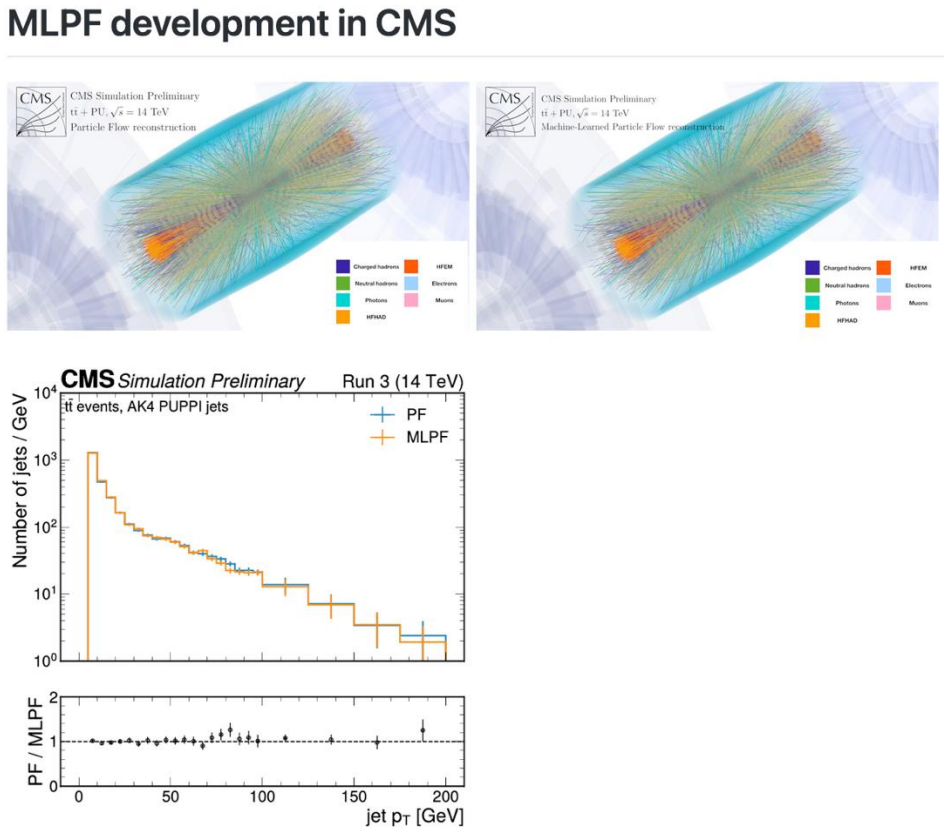


Machine-Learned Particle Flow

An early example of a new generation of ML workloads being developed across all experiments

- Containerized and available as an optional benchmark in HEPscore
- Executes on any hardware supporting ML frameworks
- Aggressively leverages accelerator offloading

ML workloads such as this will take time before they replace their traditional deterministic (x86) counterparts, but this process has already begun in many experiments



See talks by J. Pata, E. Wulff

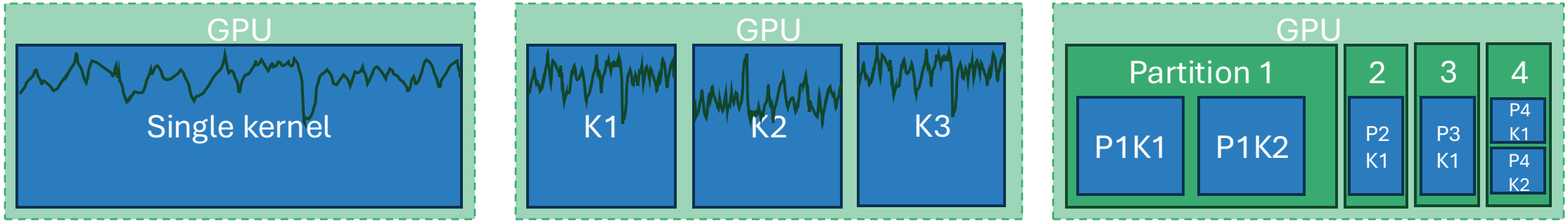
Ongoing work

Accounting to represent node capabilities with a single value becomes much more complicated with accelerators:

- Single score per “compute device” not straightforward either
- Production will have both CPU and GPU loaded in parallel (potentially with different kernels)
- Capturing interaction between these

Investigate how to effectively load GPUs:

- Unclear that same approach used in CPU benchmarks (many copies) still possible / makes sense
- Highly workload dependent – Training vs Inference vs Event Generators vs Reconstruction



Conclusions

Many developments in benchmarking to cover heterogeneous architectures:

- Workloads supporting multiple compile targets
- Leveraging multiple types of compute accelerators and frameworks
- Major infrastructure tooling effort to support HEPscore rollout
- Focus on energy, compute efficiency, gpus across many devices

Future timeline:

- “HEPscore25” planned for Q1 2025, with new release of HEPScore and Suite
- HEP Benchmark Suite v3.0 with enhanced metadata, plugins, statistics
- HEPscore v2.0 with CPU thread scan capability and HEPScore-GPU configurations

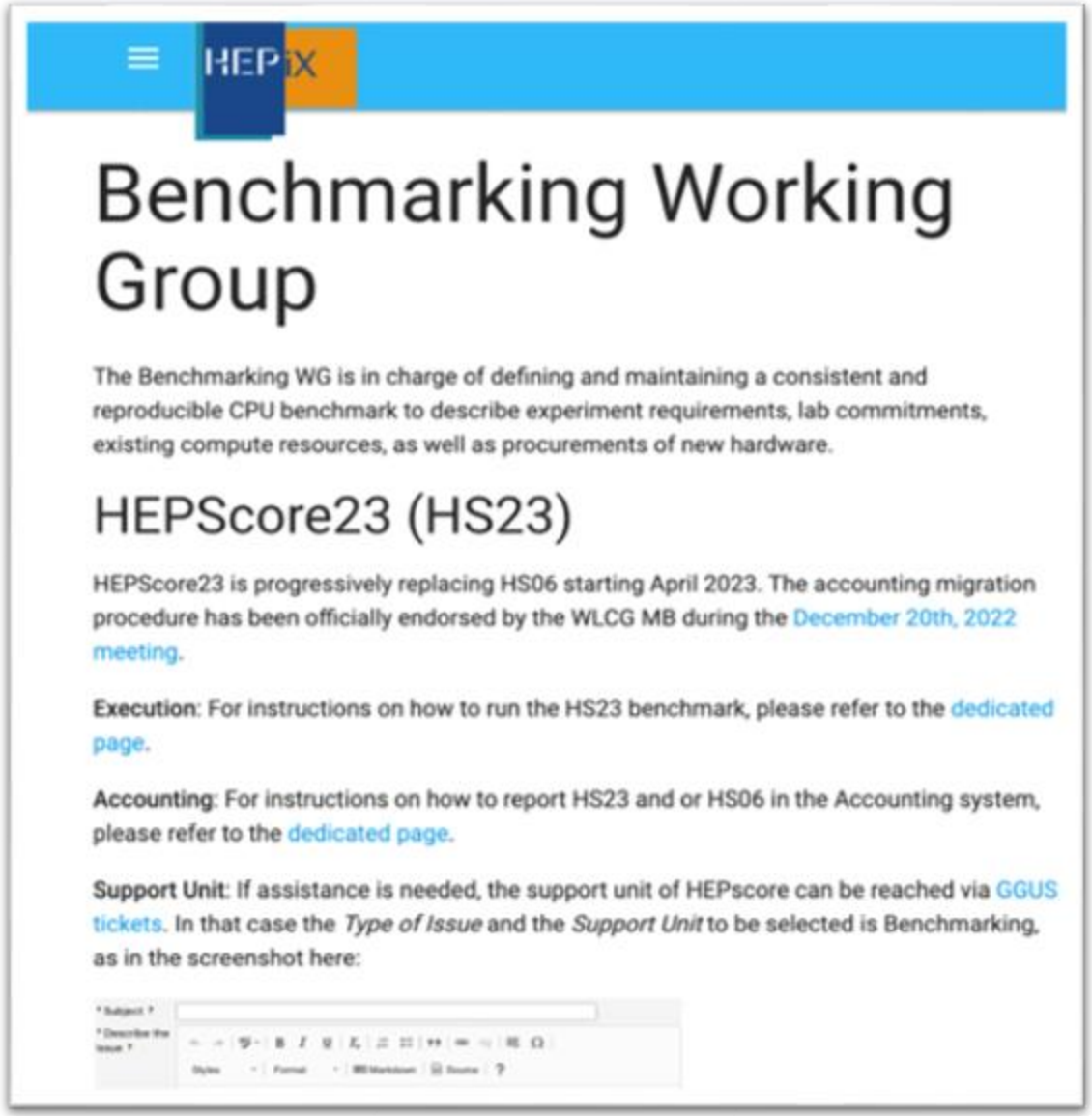
Thank you!

CERN openlab

HEPScore Documentation

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

HS23 table:
http://w3.hepix.org/benchmarking/scores_HS23.html



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.

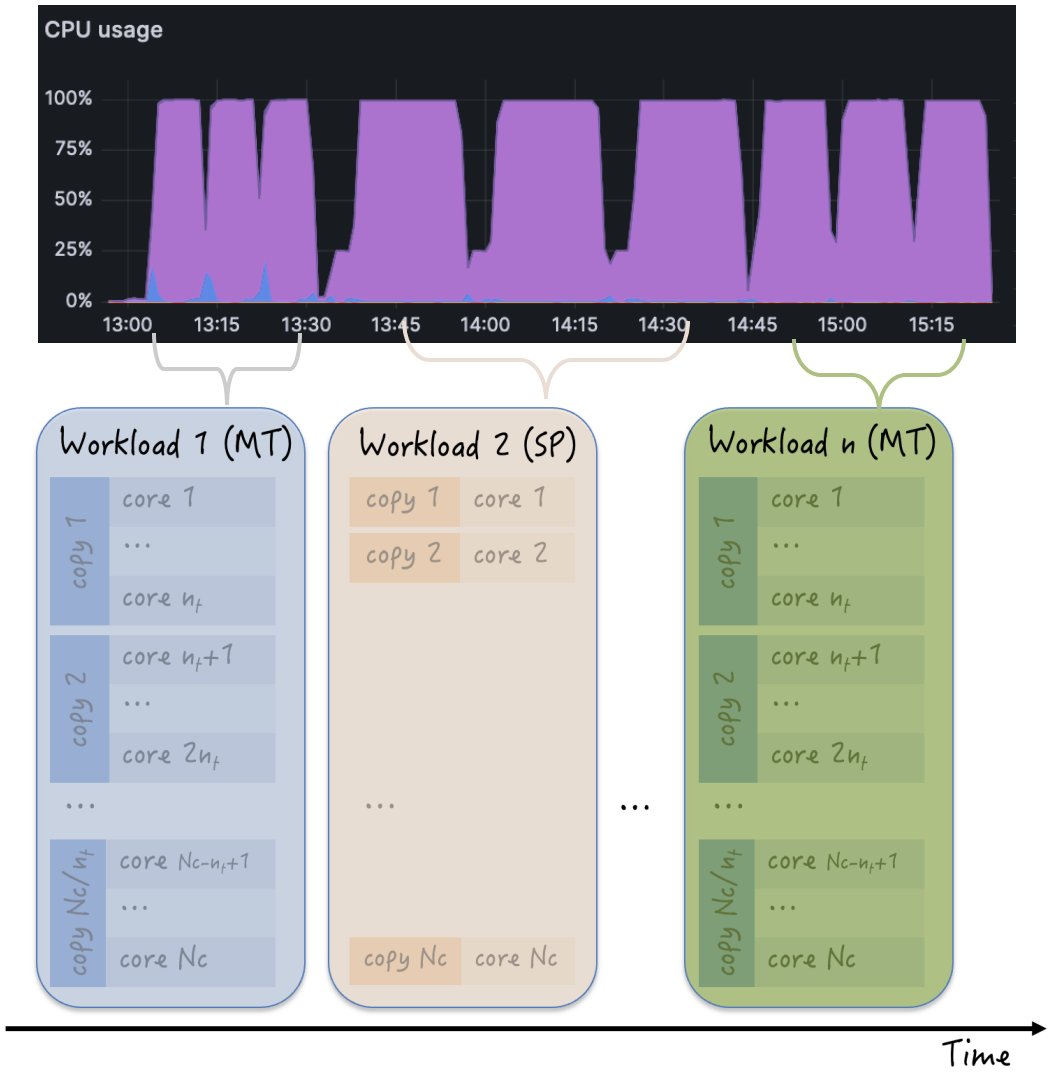
Accounting: For instructions on how to report HS23 and or HS06 in the Accounting system, please refer to the dedicated page.

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:

Source: http://w3.hepix.org/benchmarking/

Workload execution mode

- Each container runs the Experiment executable with a configurable number (n_t) of threads (or processes)
  - Default n_t=4 (or 1 for single threaded applications)
- Fixed the number of threads per executable, the available cores are saturated spawning a ***computed*** number of parallel copies of the executable
- The **score** of each WL is the cumulative event throughput of the running copies
  - When possible, the initialization and finalization phases are excluded from the computation
  - Otherwise, a long enough sequence of events is used
- Typically, run 3 executions of the same workload and select the median score



HEPScore23

- 7 workloads from 5 experiments
  - 3 Single process workloads +
    4 multi thread/process workloads
  - Container images based on **Linux CC7**
- Support for **x86** and **aarch64**
- 1:1 normalization with HS06 for the reference CPU model

  Intel® Xeon® Gold 6326 CPU @ 2.90 GHz (HT=On)

Exp	Workload	Sw version
ALICE	Digi Reco	O2/nightly-20221215-1
ATLAS	Gen sherpa (SP) (*) SP: Single Process	Athena 23.0.3
	Reco	Athena 23.0.3
Belle2	Gen Sim Reco (SP)	release-06-00-08
CMS	Gen Sim	CMSSW_12_5_0
	Reco	CMSSW_12_5_0
LHCb	Sim (SP)	v3r412