HEPScore benchmark status

D. Giordano (CERN/IT)

GDB

13 Sept 2023

Outline

- HEPScore23 (HS23) replaces HS06 to benchmark new hardware since April 1st 2023
- Today: project update since the last (April 12) GDB report
  - Recall the HS23 definition
  - Documentation
  - Measurements collected
  - New studies and development
  - Support and noticed issues

HEPScore23

- 7 workloads included
- All workloads have a recent versio of the experiments' SW
  - Support x86_64 and aarch64
- 3 Single process workloads + 4 multi thread/process workloads
- Reference server:

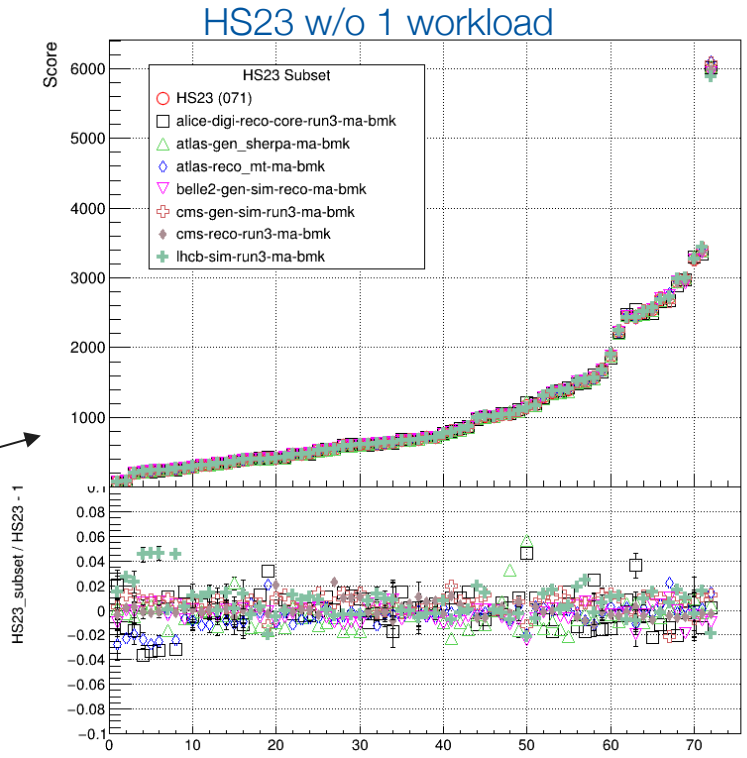
  Intel® Xeon® Gold 6326 CPU @ 2.90 GHz (HT=On)

Exp	Workload	x86_64 / aarch64	Sw version
ALICE	digi-reco	✅	O2/nightly-20221215-1
ATLAS	gen_sherpa (SP) (*) SP: Single Process	✅	Athena 23.0.3
	reco_mt	✅	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

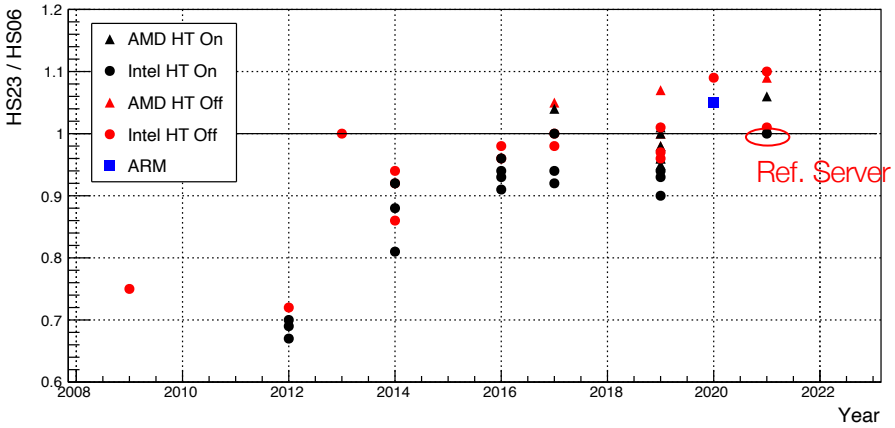
HEPScore23 validation campaign

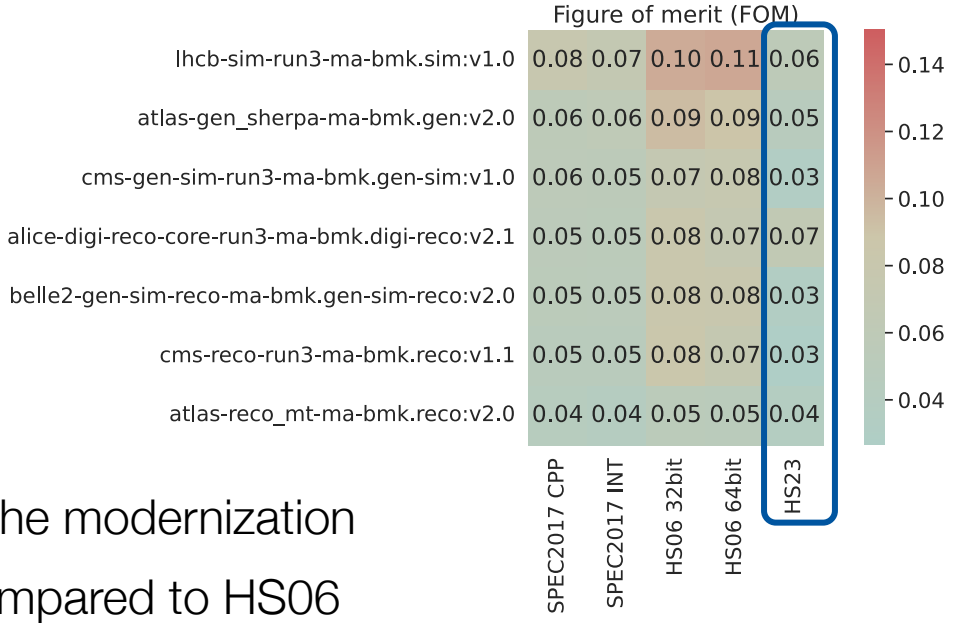
Since March 1st, measurements from a variety of servers and sites

- ~15 sites
- 46 distinct CPU models (Intel, AMD, ARM)
  - Including recent ARM nodes from vendors
- Small spread [‰,%] in repeated measurements
- Study of HS23 robustness:
  score is not significantly affected by the removal of one of the 7 workloads



CPU models by year





HS23 is a more accurate representation of the modernization that has taken place in HEP applications compared to HS06

- Figure of merit (FOM) is a high-level measure of that
  - Def.: Average deviation from linear fit

End of the HEPScore deployment Task Force

At the WLCG MB on May 16 agreed to close the Task Force activity

Objectives accomplished

- Coordinate the collection of new workloads
- Onboard WLCG sites for validation
- Recommend the HEPScore composition
- Strategy for HS06->HEPScore migration

Remaining activities are under the role of other bodies

- Support & development: HEPiX Benchmarking WG
- Accounting & monitor the adoption of HEPScore: Accounting WG

Downfall vulnerability mitigation Vs HS23

- CVE-2022-40982
  - Affected Intel processors: from the 6th (Skylake) to 11th (Tiger Lake) generation
- Mitigation: microcode update from Intel
  - Phoronix measured up to 50% performance penalties in extreme cases (https://www.phoronix.com/review/intel-downfall-benchmarks)
- Which effect for the HEP workloads?

https://downfall.page/

Downfall Attacks

Attack Demo FAQ Advisories Links

Downfall attacks target a critical weakness found in billions of modern processors used in personal and cloud computers. This vulnerability, identified as CVE-2022-40982, enables a user to access and steal data from other users who share the same computer. For instance, a malicious app obtained from an app store could use the Downfall attack to steal sensitive information like passwords, encryption keys, and private data such as banking details, personal emails, and messages. Similarly, in cloud computing environments, a malicious customer could exploit the Downfall vulnerability to steal data and credentials from other customers who share the same cloud computer.

The vulnerability is caused by memory optimization features in Intel processors that unintentionally reveal internal hardware registers to software. This allows untrusted software to access data stored by other

Downfall vulnerability mitigation Vs HS23

- No relevant effect of the mitigation measured for the HEP workloads included in HS23
- Results obtained in the 24 hours after the CVE announcement, thanks to the readiness of the benchmarking infrastructure at CERN

HEPscore score CVE-2022-40982

Microcode change	CPU Model	Microcode 1	Microcode 2	Score before	Score after	Count 1	Count 2	Ratio
0x302	Intel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz	0x5003302	0x5003604	709	709	19	16	1.00
0x302	Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz	0x5003302	0x5003604	715	704	115	16	0.985
0x201	Intel(R) Xeon(R) Gold 6130 CPU @ 2.10GHz	0x2006E05	0x2007006	691	699	44	15	1.01
0x30	Intel(R) Xeon(R) Gold 6326 CPU @ 2.90GHz	0xD000375	0xD0003A5	1018	1011	117	11	0.993
0x0	AMD EPYC 7302 16-Core Processor	0x830104D	0x830104D	981	993	171	23	1.01
0x0	Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz	0x49	0x49	319	319	112	16	1.000
0x0	Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz	0xB00001F	0xB00001F	482	482	112	16	0.999
0x0	Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz	0xB000030	0xB000030	631	631	114	16	0.999

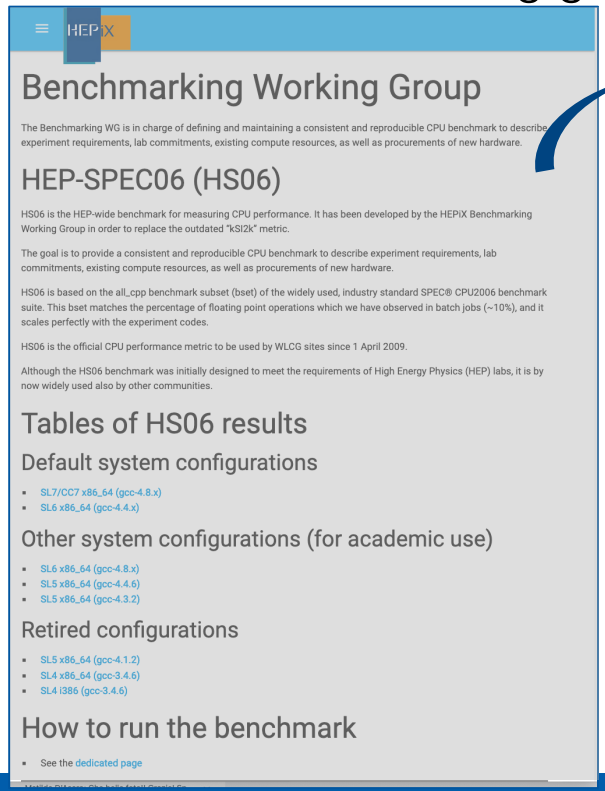
HS06 score CVE-2022-40982

Microcode change	CPU Model	Microcode 1	Microcode 2	Score before	Score after	Count 1	Count 2	Ratio
0x500	Intel(R) Xeon(R) Gold 6130 CPU @ 2.10GHz	0x2006B06	0x2007006	735	751	25	28	1.02
0x302	Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz	0x5003302	0x5003604	770	763	43	32	0.991
0x30	Intel(R) Xeon(R) Gold 6326 CPU @ 2.90GHz	0xD000375	0xD0003A5	1015	1013	63	65	0.998
0x3	Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz	0x46	0x49	365	364	18	32	0.997
0x0	AMD EPYC 7302 16-Core Processor	0x830104D	0x830104D	1036	1029	17	41	0.993
0x0	Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz	0xB00001F	0xB00001F	521	521	17	32	1.000
0x0	Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz	0xB000030	0xB000030	660	662	17	32	1.00

Documentation

https://w3.hepix.org/benchmarking.html

Update the official HEPiX working group page



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.

HEP-SPEC06 (HS06)

HS06 is the HEP-wide benchmark for measuring CPU performance. It has been developed by the HEPiX Benchmarking Working Group in order to replace the outdated "kSI2k" metric.

The goal is to provide a consistent and reproducible CPU benchmark to describe experiment requirements, lab commitments, existing compute resources, as well as procurements of new hardware.

HS06 is based on the all_cpp benchmark subset (bset) of the widely used, industry standard SPEC® CPU2006 benchmark suite. This bset matches the percentage of floating point operations which we have observed in batch jobs (~10%), and it scales perfectly with the experiment codes.

HS06 is the official CPU performance metric to be used by WLCG sites since 1 April 2009.

Although the HS06 benchmark was initially designed to meet the requirements of High Energy Physics (HEP) labs, it is by now widely used also by other communities.

Tables of HS06 results

Default system configurations

- SL7/CC7 x86_64 (gcc-4.8.x)
- SL6 x86_64 (gcc-4.4.x)

Other system configurations (for academic use)

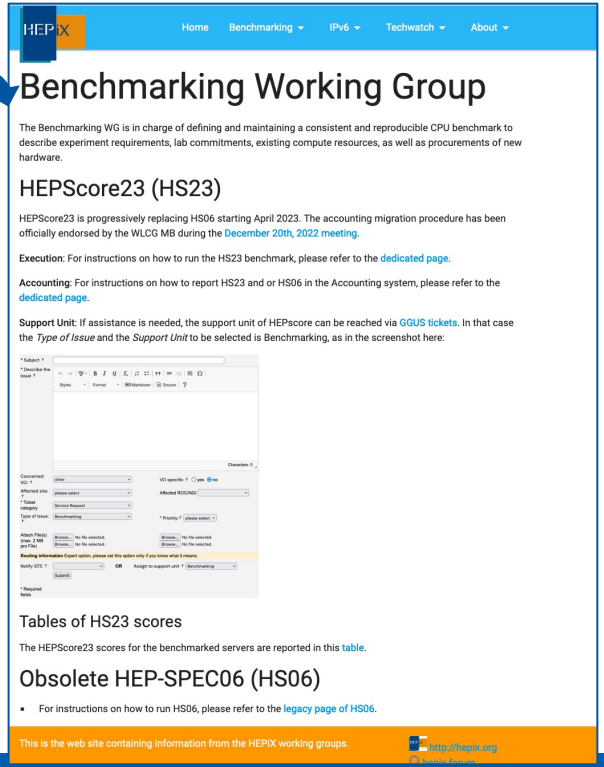
- SL6 x86_64 (gcc-4.8.x)
- SL5 x86_64 (gcc-4.4.6)
- SL5 x86_64 (gcc-4.3.2)

Retired configurations

- SL5 x86_64 (gcc-4.1.2)
- SL4 x86_64 (gcc-3.4.6)
- SL4 i386 (gcc-3.4.6)

How to run the benchmark

- See the dedicated page



Home Benchmarking IPv6 Techwatch About

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:

Tables of HS23 scores

The HEPScore23 scores for the benchmarked servers are reported in this table.

Obsolete HEP-SPEC06 (HS06)

- For instructions on how to run HS06, please refer to the legacy page of HS06.

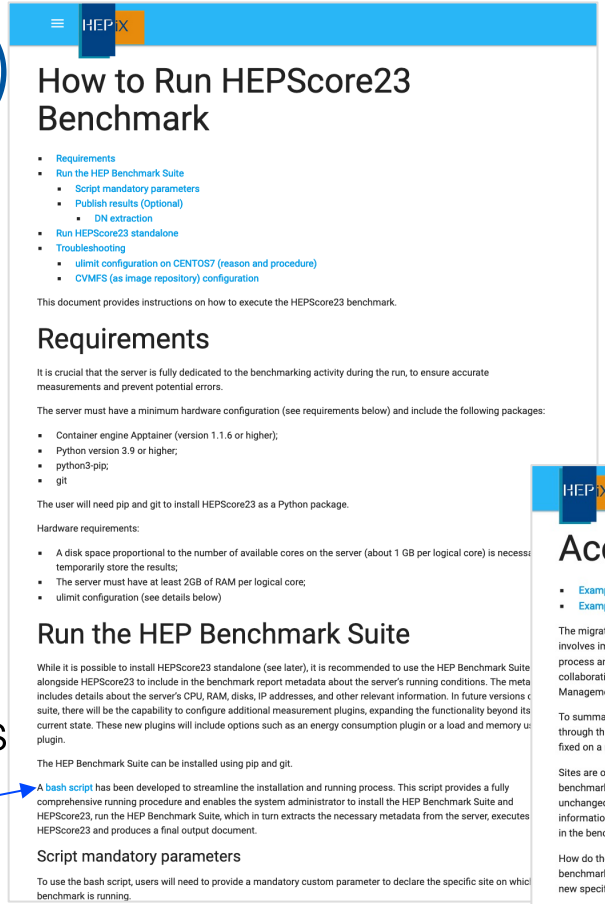
This is the web site containing information from the HEPiX working groups. http://hepix.org

Documentation (II)

Includes:

- Legacy HS06 pages
- HS23 run instructions
- HS23 scores published
- Accounting instructions

⚠ It's highly recommended to follow the provided instructions and use the Suite script



How to Run HEPScore23 Benchmark

- Requirements
- Run the HEP Benchmark Suite
  - Script mandatory parameters
  - Publish results (Optional)
    - DN extraction
- Run HEPScore23 standalone
- Troubleshooting
  - ulimit configuration on CENTOS7 (reason and procedure)
  - CVMFS (as image repository) configuration

This document provides instructions on how to execute the HEPScore23 benchmark.

Requirements

It is crucial that the server is fully dedicated to the benchmarking activity during the run, to ensure accurate measurements and prevent potential errors.

The server must have a minimum hardware configuration (see requirements below) and include the following packages:

- Container engine Apptainer (version 1.1.6 or higher);
- Python version 3.9 or higher;
- python3-pip;
- git

The user will need pip and git to install HEPScore23 as a Python package.

Hardware requirements:

- A disk space proportional to the number of available cores on the server (about 1 GB per logical core) is necessa
  temporarily store the results;
- The server must have at least 2GB of RAM per logical core;
- ulimit configuration (see details below)

Run the HEP Benchmark Suite

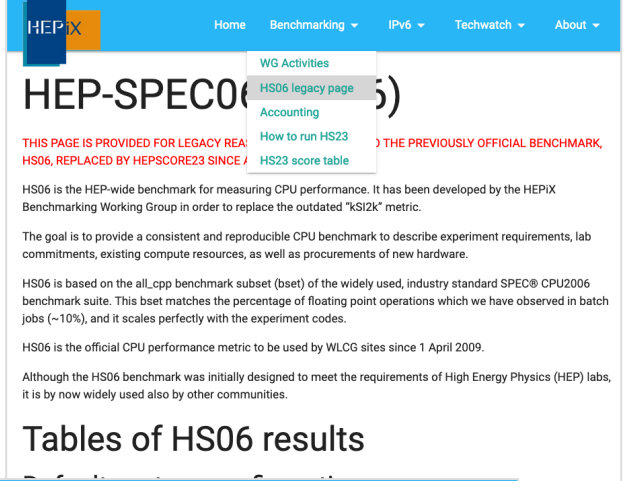
While it is possible to install HEPScore23 standalone (see later), it is recommended to use the HEP Benchmark Suite
alongside HEPScore23 to include in the benchmark report metadata about the server's running conditions. The meta
includes details about the server's CPU, RAM, disks, IP addresses, and other relevant information. In future versions
suite, there will be the capability to configure additional measurement plugins, expanding the functionality beyond its
current state. These new plugins will include options such as an energy consumption plugin or a load and memory u
plugin.

The HEP Benchmark Suite can be installed using pip and git.

A bash script has been developed to streamline the installation and running process. This script provides a fully comprehensive running procedure and enables the system administrator to install the HEP Benchmark Suite and HEPScore23, run the HEP Benchmark Suite, which in turn extracts the necessary metadata from the server, executes HEPScore23 and produces a final output document.

Script mandatory parameters

To use the bash script, users will need to provide a mandatory custom parameter to declare the specific site on whic
benchmark is running.



Home | Benchmarking | IPv6 | Techwatch | About

WG Activities
HS06 legacy page
Accounting
How to run HS23
HS23 score table

HEP-SPEC06)

THIS PAGE IS PROVIDED FOR LEGACY REA) THE PREVIOUSLY OFFICIAL BENCHMARK, HS06, REPLACED BY HEPSCORE23 SINCE A

HS06 is the HEP-wide benchmark for measuring CPU performance. It has been developed by the HEPiX Benchmarking Working Group in order to replace the outdated "kSI2k" metric.

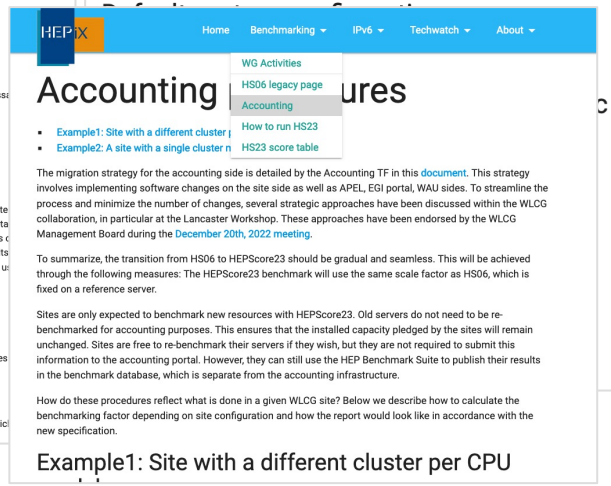
The goal is to provide a consistent and reproducible CPU benchmark to describe experiment requirements, lab commitments, existing compute resources, as well as procurements of new hardware.

HS06 is based on the all_cpp benchmark subset (bset) of the widely used, industry standard SPEC® CPU2006 benchmark suite. This bset matches the percentage of floating point operations which we have observed in batch jobs (~10%), and it scales perfectly with the experiment codes.

HS06 is the official CPU performance metric to be used by WLCG sites since 1 April 2009.

Although the HS06 benchmark was initially designed to meet the requirements of High Energy Physics (HEP) labs, it is by now widely used also by other communities.

Tables of HS06 results



Home | Benchmarking | IPv6 | Techwatch | About

WG Activities
HS06 legacy page
Accounting
How to run HS23
HS23 score table

Accounting ures

- Example1: Site with a different cluster
- Example2: A site with a single cluster n

The migration strategy for the accounting side is detailed by the Accounting TF in this document. This strategy involves implementing software changes on the site side as well as APEL, EGI portal, WAU sides. To streamline the process and minimize the number of changes, several strategic approaches have been discussed within the WLCG collaboration, in particular at the Lancaster Workshop. These approaches have been endorsed by the WLCG Management Board during the December 20th, 2022 meeting.

To summarize, the transition from HS06 to HEPScore23 should be gradual and seamless. This will be achieved through the following measures: The HEPScore23 benchmark will use the same scale factor as HS06, which is fixed on a reference server.

Sites are only expected to benchmark new resources with HEPScore23. Old servers do not need to be re-benchmarked for accounting purposes. This ensures that the installed capacity pledged by the sites will remain unchanged. Sites are free to re-benchmark their servers if they wish, but they are not required to submit this information to the accounting portal. However, they can still use the HEP Benchmark Suite to publish their results in the benchmark database, which is separate from the accounting infrastructure.

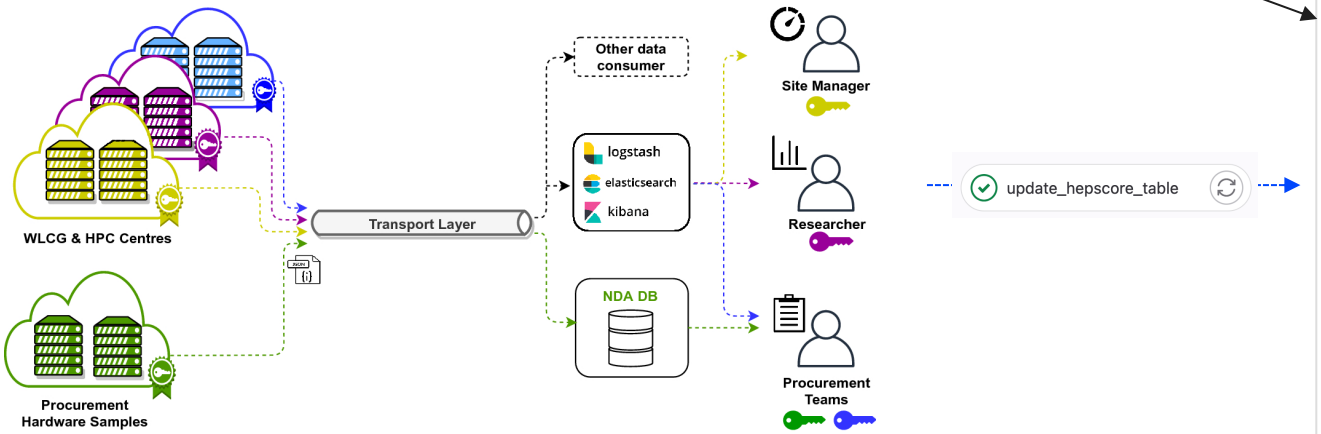
How do these procedures reflect what is done in a given WLCG site? Below we describe how to calculate the benchmarking factor depending on site configuration and how the report would look like in accordance with the new specification.

Example1: Site with a different cluster per CPU

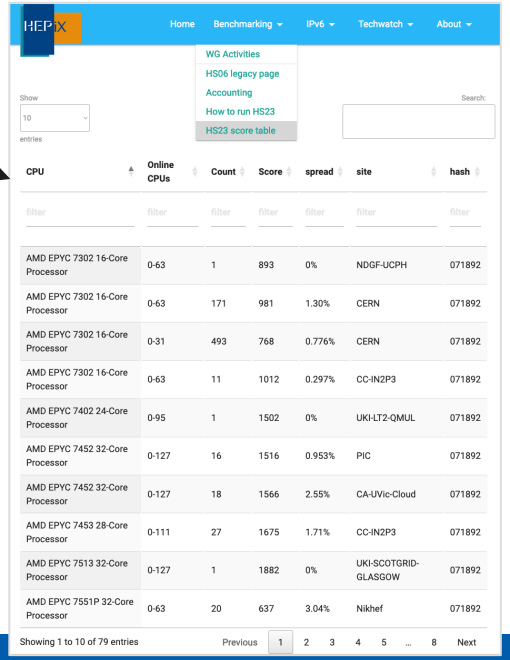
Benefit of running HS23 via the Suite script

Actively participate in building the community repository of results

- Sites send data to the centralized benchmark DB
  - Grid certificate DN to be included in the publisher list (read the doc)
- Automated analysis updates the public HS23 result table



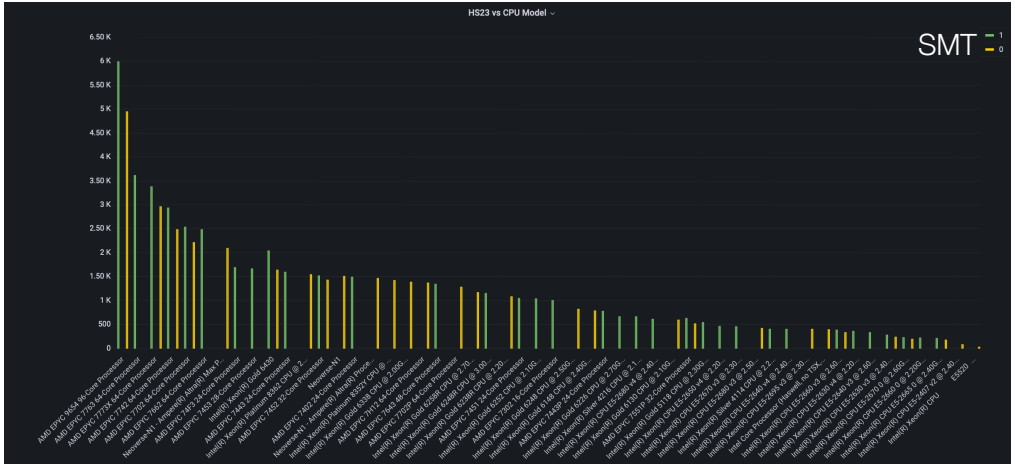
https://w3.hepix.org/benchmarking/scores_HS23.html

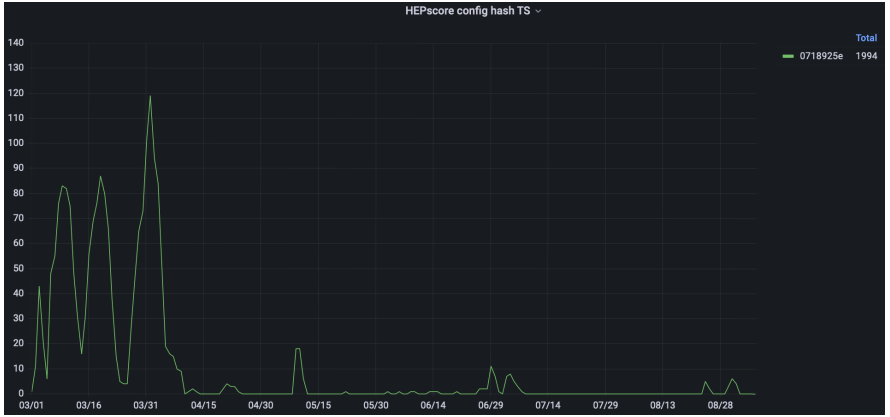


Data collected since March

96 entries: 55 CPU models,

~ 20 sites, multiple configs





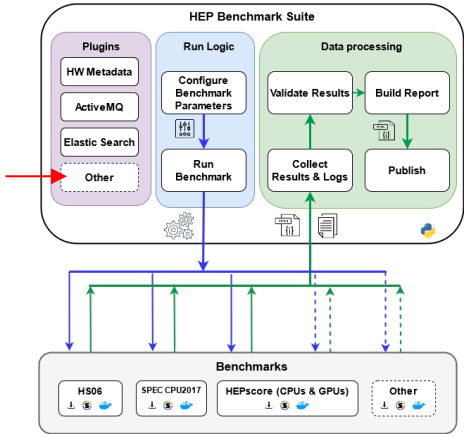
After the validation phase, sites contribute when new HW arrives

The Suite offers even more…

Extended the metadata plugin library to timeseries metrics:

energy consumption, load, CPU frequency

- Can be correlated with the Benchmark score measurements
- Developed to answer to the increasing interest in energy Vs performance studies and the need of a common dataflow



Example: Plugin Configuration

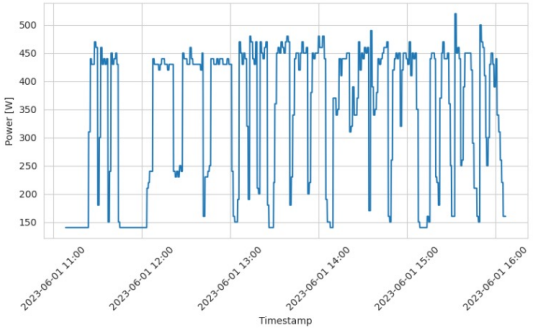
```
{
  "name": "power-consumption",
  "description": "Retrieves power consumption of the system. Requires elevated p
  "command": "ipmitool dcmi power reading",
  "regex": "Instantaneous power reading:\\s*(?P<value>\\d+) Watts",
  "unit": "W",
  "example-output": "\n    Instantaneous power reading:                   124 Wa
  "expected-value": 124
},
```

```
{
  "name": "load",
  "description": "Retrieves the one minute system load average. Note that loa
  "command": "uptime",
  "regex": "load average: (?P<value>\\d+.\\d+),",
  "unit": "",
  "aggregation": "",
  "example-output": " 11:02:47 up  3:03,  1 user,  load average: 0.18, 0.38,
  "expected-value": 0.18
},
```

Plugins' Report


```
"plugins": {
    "CommandExecutor": {
        "hepscore": {…},
        "pre": {
            "load": {
                "start_time": "2023-09-03T11:19:08.772095Z",
                "config": {
                    "command": "uptime",
                    "interval_mins": 1,
                    "aggregation": "sum",
                    "regex": "load average: (?P<value>\\d+.\\d+),",
                    "unit": ""
                },
                "values": [11.05,10.47,10.49,10.22,10.45,10.54],
                "end_time": "2023-09-03T11:24:08.774851Z",
                "statistics": {
                    "min": 10.22,
                    "mean": 10.536,
                    "max": 11.05
                }
            },
            "status": "success",
            "used-memory": {…},
            "used-swap-memory": {…}
        },
        "post": {…}
```


Rebuild time series from stored data



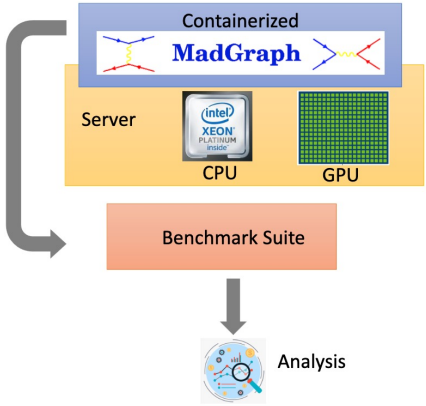
Benchmark GPUs (summer student)

- Summer Student activity at CERN
- Used a containerized Madgraph version ported on CPU+GPU
  - https://indico.cern.ch/event/1225408/contributions/5243830/
- Used the Suite plugins to collect energy consumption

```
plugins:
  CommandExecutor:
    metrics:
      load:
        command: uptime
        regex: 'load average: (?P<value>\d+.\d+),'
        unit: ''
        interval_mins: 0.1
      power-consumption:
        command: sudo ipmitool dcmi power reading
        regex: 'Instantaneous power reading:\s*(?P<value>\d+) Watts'
        unit: W
        interval_mins: 0.1
      gpu-power-consumption:
        command: nvidia-smi --query-gpu=power.draw --format=csv,noheader,nounits
        regex: '(?P<value>\d+(.\d+)?).*'
        unit: W
        interval_mins: 0.1
      gpu-usage:
        command: nvidia-smi --query-gpu=utilization.gpu --format=csv,noheader,nounits
        regex: '(?P<value>\d+(.\d+)?).*'
        unit: W
        interval_mins: 0.1
```

Server Specs:

- **CPU**
  Intel Platinum 8362 @ 2.80GHz
  Cores per socket: 32
  Sockets: 2
- **GPU**
  Model name: Nvidia L4
  Total board power: 72 W
  GPU clocks: 2 GHz (Boost)

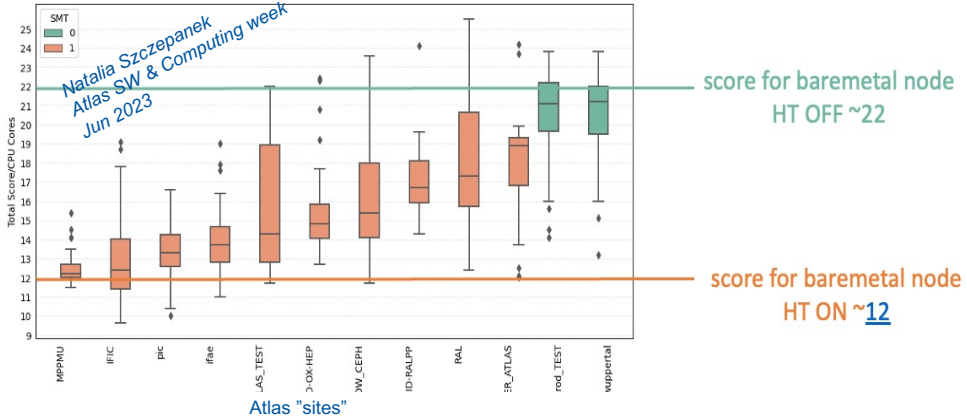


Keshvi Tuteja

Profile performance of grid job slots

- Same CPU models can perform very differently from grid site to grid site (well known fact)
  - Confirmed submitting the Benchmark Suite as job payload to several sites on 8-cores job slots
  - Measured HS23/core

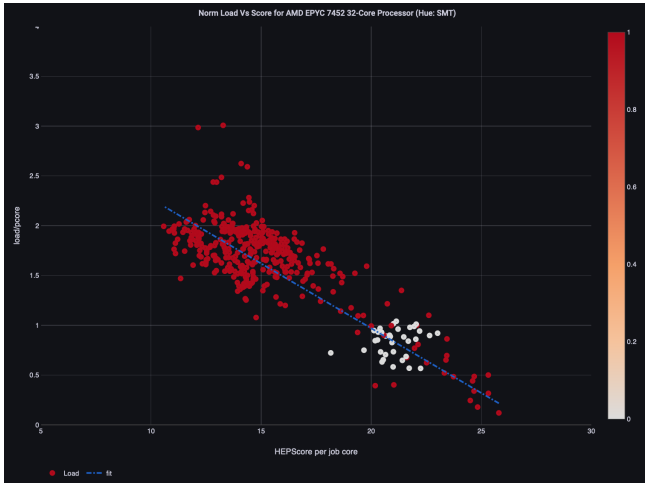
Example: AMD EPYC 7452 32-Core

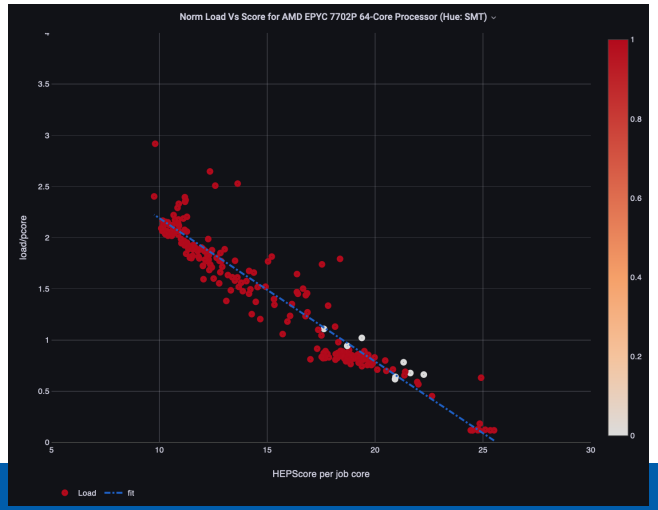


- Instrumented the plugin extension to measure server load and memory utilization during the execution of HS23 (see next slide for the correlation study)

Preliminary Analysis

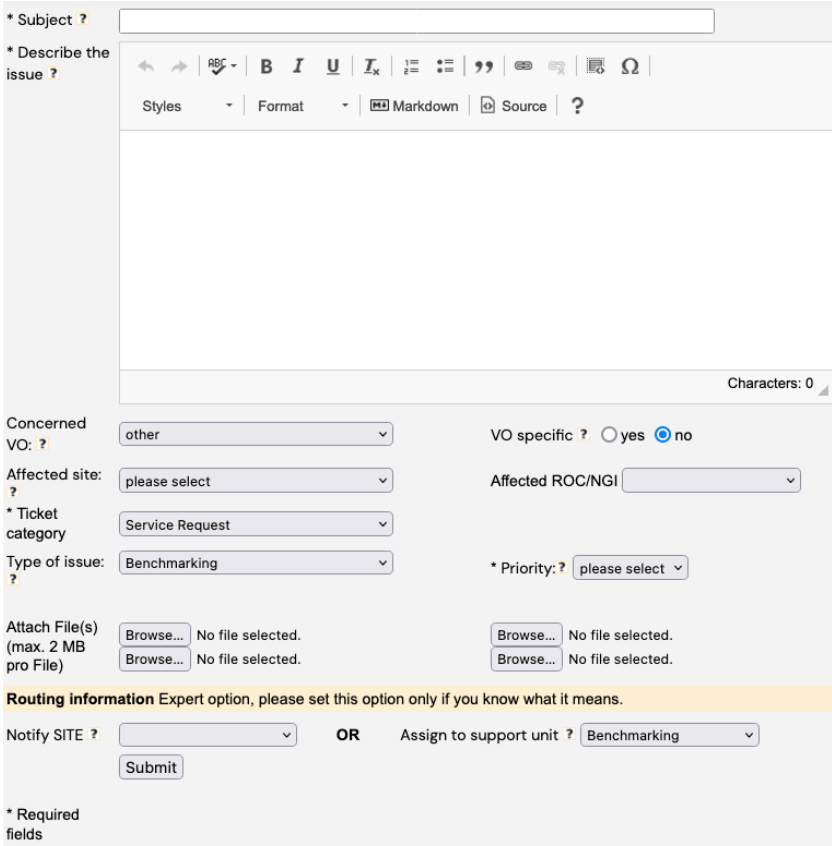
- Data retrieved running the Suite
  - Measured server load and HS23 of an 8-cores job slot
- Derived metrics:
  - X: HS23 / job core
    - Performance of each single core of a job slot of 8 cores
  - Y: load / physical core
    - when HT ON, a server fully loaded will be at 2
- Pretty good linear (anti-)correlation of server load and job slot performance
  - Offers the opportunity to investigate outliers and improve site configurations. Already done for a site





GGUS support

- Available since April
  - 7 GGUS tickets received so far
    - 4: certificate DNs for data publications
    - 3: support
- Please prefer GGUS for traceability reasons
  - Few requests still received via email
    - 7: certificate DNs
- Results (json format) that could not be sent via the standard procedure can be sent via GGUS as attachment



Issues seen so far running HS23

Contribute to extend the troubleshooting doc area

– https://w3.hepix.org/benchmarking/how_to_run_HS23.html

1. Sharp increase of memory utilization for the Alice workload
   - Mainly for CPUs with high number of cores
   - Workaround: add large swap space. Future fix: Consolidate the Alice workload
2. selinux Vs Apptainer
   - Seen in few cases. workaround: disable SELinux.
     FATAL ERROR:write_xattr: failed to write xattr security.selinux for file /image/root/.exec
3. Large cores count CPUs
   - ulimits on CentOS7 may need to be unlimited
   - CVMFS used as registry: max number of open files to be increased (CVMFS_NFILES)
4. Sporadic failures of Atlas workloads (under investigation)

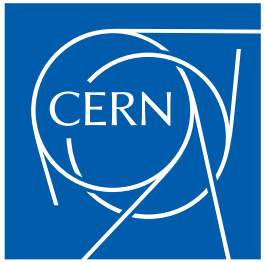
Ongoing work: HEP Workloads

- ❑ Allow to load a configurable number of cores
- ❑ Improve the validation of input params
- ❑ Study score stability Vs number of events/thread
- ❑ Progress on GPU workloads

Summary

- HEPScore23 is passing the test of the user adoption
  - Several studies ongoing
- The Benchmark Suite collects also usage metrics
  - To be released in v3.0 in the coming month
- Some workloads need consolidation to avoid failures
  - New containers will be released
  - Implication: the HS23 hash will change!

    The effect on the score itself will be small
  - The accounting should keep track of these changes


CERN