HEPiX Benchmarking Working Group Report

D. Giordano (CERN)
on behalf of
HEPiX Benchmarking WG

HEPiX Spring 2024
16/04/2024

1 year of HEPScore23 (HS23)

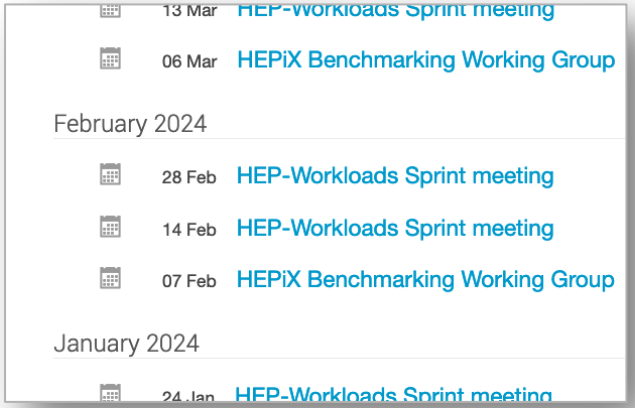
- HS23 has replaced HS06 as of April 1st
- Outline of this report
  - Status
    - Results, procedures
  - Lesson learned
    - Issues, consolidations
  - Improvements
    - Configurable number of cores
    - Metering utilization, power consumption
  - Future

The working group organization

- Active members
  - M. Michelotto, D. Giordano (co-chairs)
  - L. Atzori, *J.M. Barbet*, C. Driemel, *C. Hollowell*, G. Menéndez Borge, A. Sciaba, E. Simili, R. Sobie, D. Southwick, T. Sullivan, N. Szczepanek, A. Valassi, E. Vamvakopoulos
  - Contributors needed. It may be you!

- Meeting frequency
  - Presentations on various topics
    - 1st week of each month
    - Announced to the **hepix-cpu-benchmark** list
  - Jira Sprint meetings
    - 2nd and 4th week of each month
    - Restricted to developers

https://indico.cern.ch/category/1806/



Tasks of the working group

- Software development
  - HEP Workloads, HEPScore, HEP Benchmark suite

- Operation
  - Maintain the infrastructure used to
    - Build workload images using VMs for gitlab runners
    - Validate workloads on bare metal nodes
    - Collect benchmark data into OpenSearch and HDFS DBs
    - Analysis

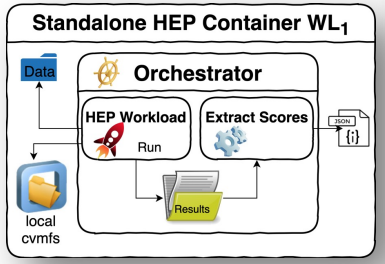
- Organize exchanges on CPU/GPU benchmarking topics

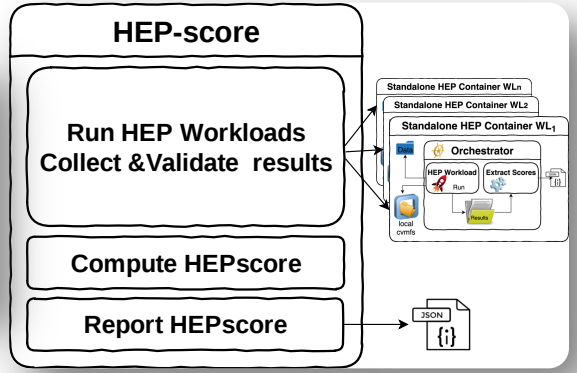
HEPScore23: Intro

https://pixabay.com/photos/belgium-antwerp-shipping-container-1601920/

HEP Benchmarks project

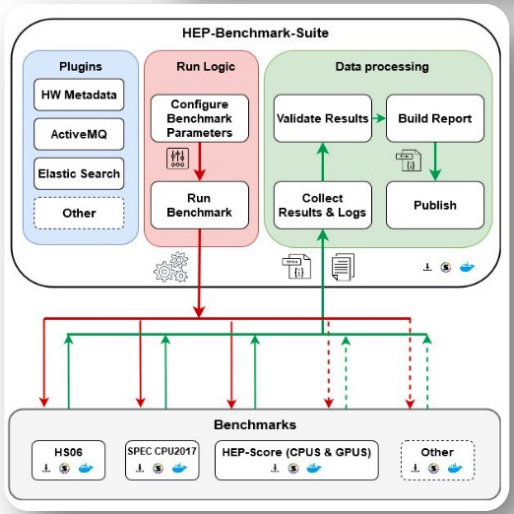
- HEP Workloads (link)
  - Individual reference HEP applications





- HEPScore (link)
  - Uses the workloads of the HEP experiments
  - Combine them in a single benchmark score

- In addition, HEP Benchmark Suite (link)
  - Orchestrator of multiple benchmark (HEPScore, HS06, SPEC CPU2017)
  - Central collection of benchmark results



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

Server utilization metrics

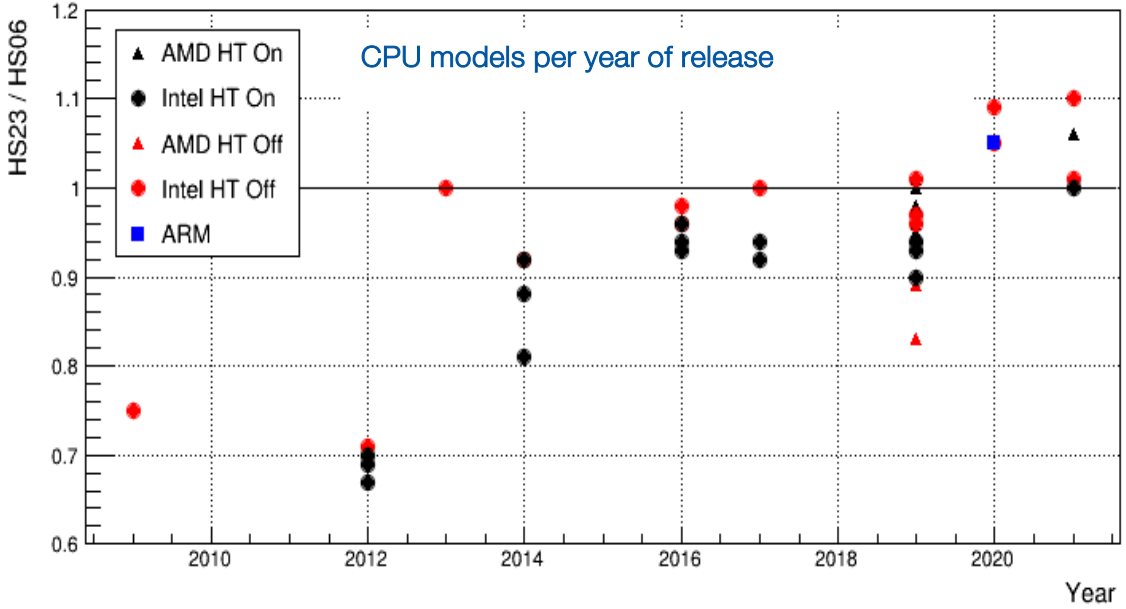
Intel(R) Xeon(R) Gold 6326 CPU @ 2.90GHz (SMT on ⇔ 64 cores)

Performance assessment of an **entire server**

- HS23 saturates the server resources (default)
- Runtime (~4 hours)
- Resolution on repeated measurements: $\sigma/\mu << 1\%$

HS23 vs HS06

Compared to HS06, HS23 provides a more accurate representation of the modernization that has taken place in HEP applications

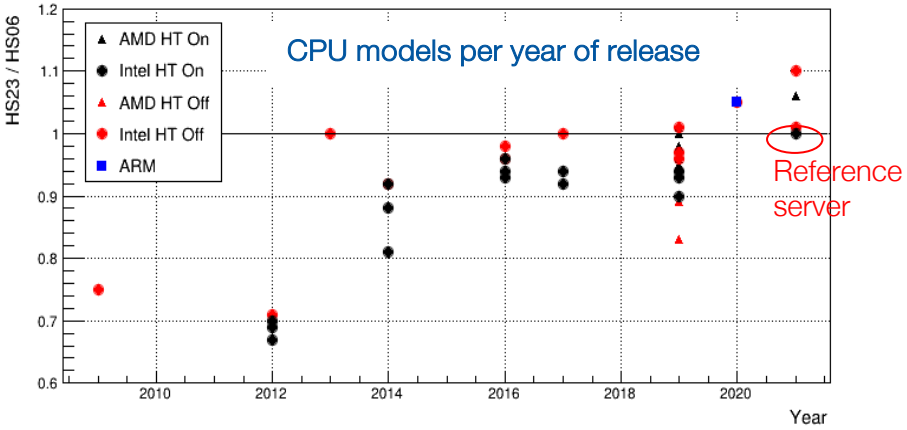


Study performed on multiple CPU models from several WLCG sites

The transition strategy from HS06 to HS23

HS23 is normalized to HS06 on the reference server

– Allows for smooth transition of tables and plots



Sites encouraged to run HS23 only on hardware deployed after April 1st 2023

– Older hardware reported in HS06 score, to avoid changes in pledged vs delivered resources

HEPScore23: Run & Results

Servers “officially” benchmarked so far

Servers benchmarked and data sent to our central benchmark DB

- NB: Many more sites could have executed the benchmark without sending data

- ~20 sites contributed. Kudos!
  We hope to see more in the future!
- ~110 distinct configurations (CPU models, SMT conf., …)
- ~10 ARM-based servers
  - Neoverse-N1 (Altra, Altra Max), Neoverse-V2 (Grace)
- Spread in repeated measurement < 0.5%
- Data available in a public table

HS23 results table

Exposes the benchmark scores of servers profiled at sites

- Reports CPU model, number of online CPUs, number of measurements, **score, spread**, site and hash of the HEPScore configuration

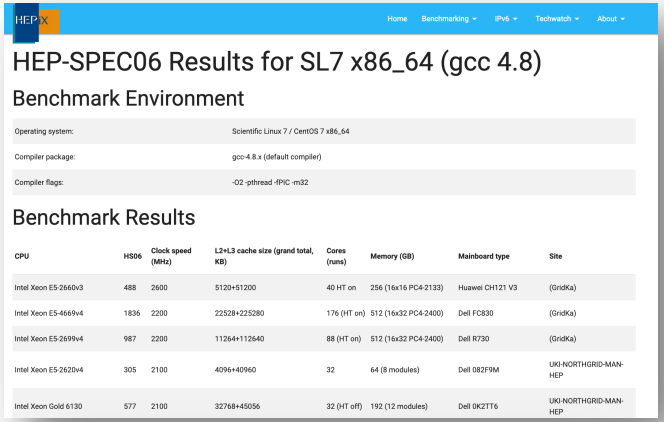
https://w3.hepix.org/benchmarking/scores_HS23.html

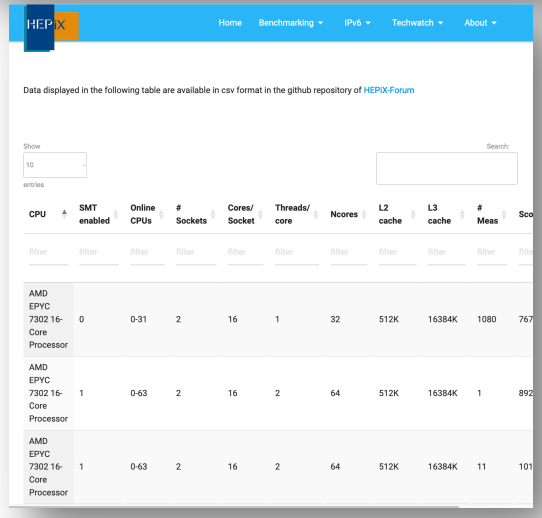
CPU	SMT enabled	Online CPUs	# Sockets	Cores/ Socket	Threads/ core	Ncores	L2 cache	L3 cache	# Meas	Score	Spread	Score/ Ncores	RAM	SWAP	Site
filter	filter	filter	filter	filter	filter	filter	filter	filter	filter	filter	filter	filter	filter	filter	filter
AMD EPYC 9754 128-Core Processor	1	0-511	2	128	2	512	256 MiB (256 instances)	512 MiB (32 instances)	5	7450.248	1.378	14.6	1 TiB	4 GiB	UKI-SCOTGR GLASGO
AMD EPYC 9654 96-Core Processor	1	0-383	2	96	2	384	1024K	32768K	26	6000.578	0.714	15.6	1 TiB	4 GiB	IHEP
AMD															

A familiar place for new data

- HS23 table replaces the HS06 results table available at the same website

- Evolved the update method
  - HS06 table was updated via a manual action, involving email exchange
  - HS23 table updated via
    - Nightly data analysis performed on the OpenSearch DB
    - Injection as CSV in the GitHub repository of the HEPiX website
    - Only results that have been sent to the central OpenSearch DB are exposed
      - Therefore: please consider to send data whenever a server is benchmarked

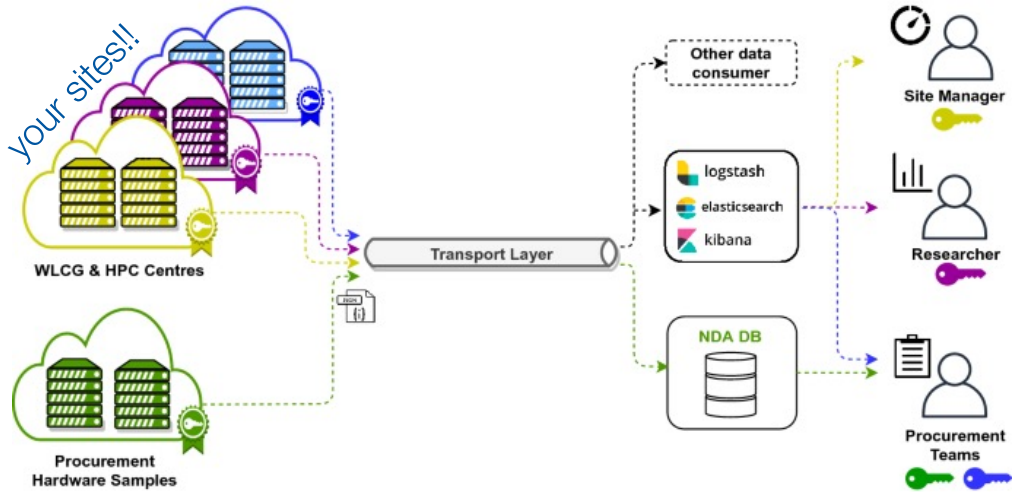




How benchmark results are collected

Benchmark measurements running at sites can be sent and stored in an OpenSearch instance @ CERN managed by the Benchmarking WG

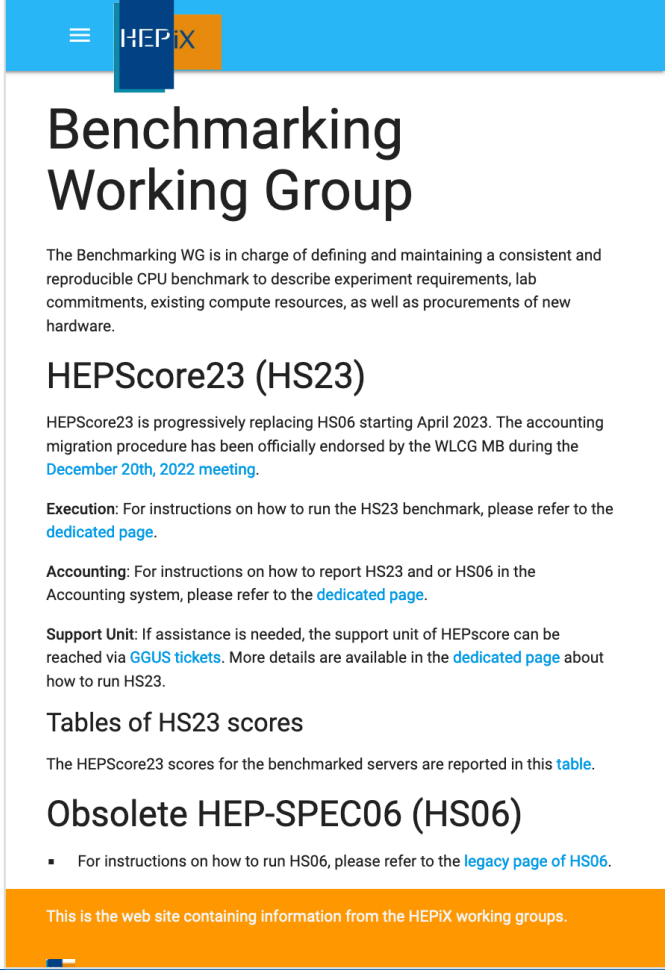
- Benchmark execution via the Benchmark Suite
- Enabled results' publication



Documentation

Available in the official HEPiX working group page
- https://w3.hepix.org/benchmarking.html

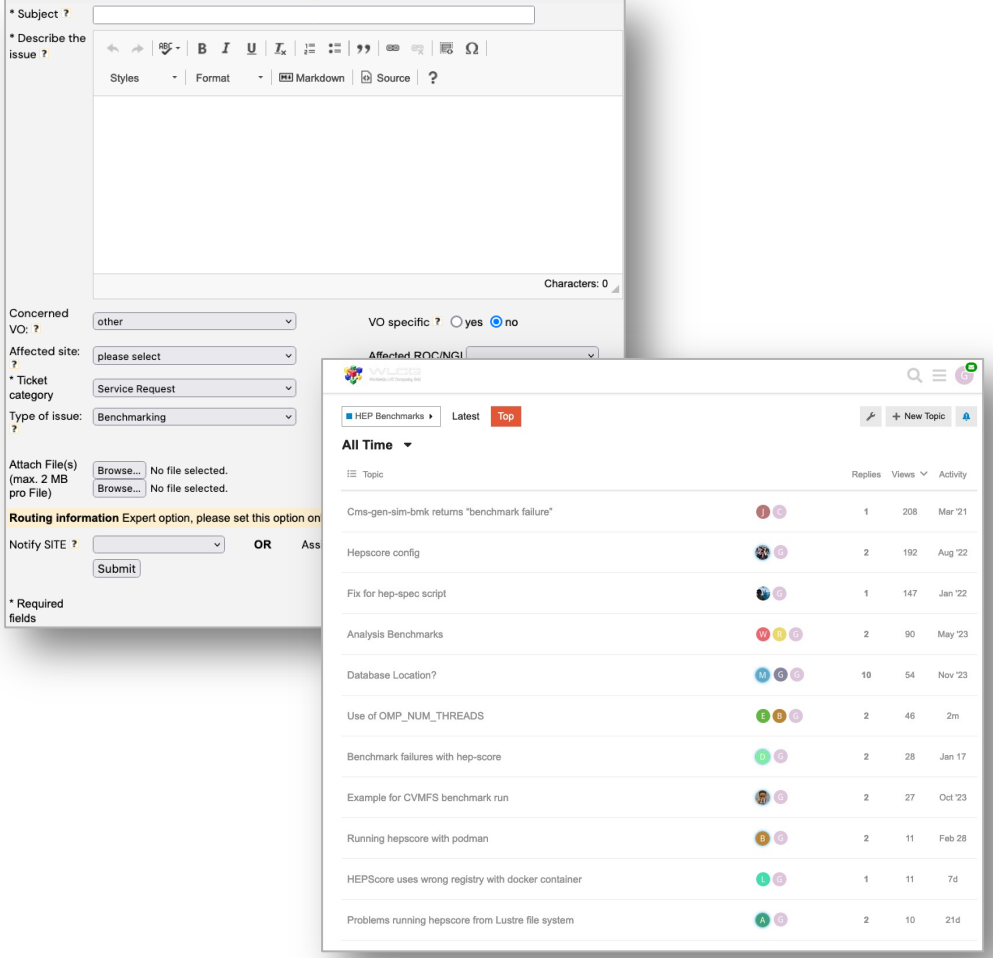
- Describes how to install and run HS23
- GGUS user support
- Legacy pages for HS06
- Instructions for accounting reports
- Table of HS23 scores reported by sites



User Support

- Primarily done via GGUS tickets
  - Alternatively, the HEP Benchmarks Project Discourse Forum
  - Or direct email

- Small amount of requests
  - Certificate DN for publication 10/16
  - Failures 4/16
  - Request of new features 2/16



Issues reported when running HS23

Occurring in a minority of cases:

- 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
- 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
- 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) or CVMFS_CACHE_REFCOUNT = yes in /etc/cvmfs/default.local
- Sporadic failures of Atlas workloads

Documented in the troubleshooting area of the documentation

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

Improvements

Prepared new workload containers

- Fix discovered issues
  - ALICE digi-reco, to reduce the memory footprint and improve the event configuration
  - ATLAS gen, to better account of all the processing steps of the MC application

- Add new features
  - Configurable number of cores to be loaded by the workload
  - Handle multiple container registries in the same configuration file
  - Provide a tarball of all container images

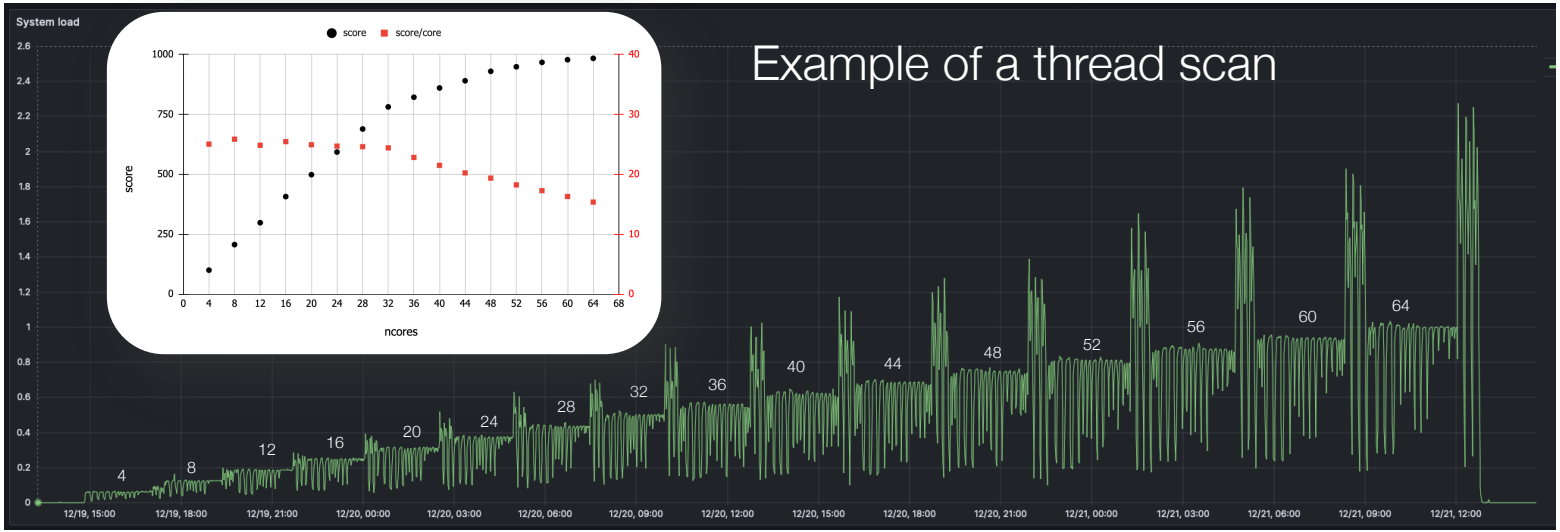
Configurable number of cores in the benchmark

Needed for thread scan

- In the past, done by adapting the config file. A script is available
- Consolidated in a command line argument (--ncores) of HEPScore propagated to each workload

HEPScore Documentation

```
-n [NCORES], --ncores [NCORES]
                      custom number of cores to be loaded. This parameter
                      will change the hash function
```



Multiple registries & Tarball

- A single configuration, multiple container registries
  - Apptainer on docker, sif, unpacked images
  - Docker on docker images

```
-R [REGISTRY], --registry [REGISTRY]
                      override the configured registry.
-i [{docker,shub,dir,oras,https}], --container_uri [{docker,shub,dir,oras,https}]
                    specify container registry type (oras , docker,
                    shub, dir, https).
```

```
66    settings:
67      name: HEPScore23
68      reference_machine: "E423521X1B04810-B Gold 6326 CPU @ 2.90GHz - 64 cores SMT ON"
69      registry:
70        - oras://gitlab-registry.cern.ch/hep-benchmarks/hep-workloads-sif
71        - docker://gitlab-registry.cern.ch/hep-benchmarks/hep-workloads
72        - dir:///cvmfs/unpacked.cern.ch/gitlab-registry.cern.ch/hep-benchmarks/hep-workloads
```

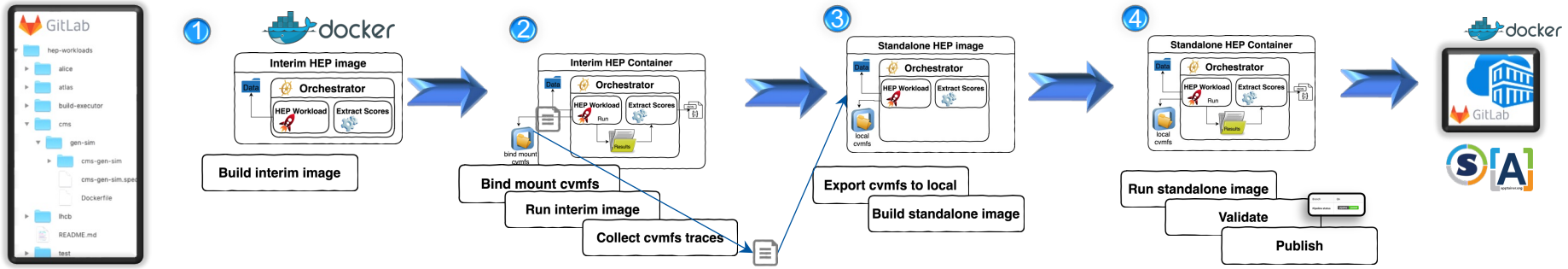
- Pre-download sif images in case of limited network connectivity
  - Typical case for HPC sites
  - Documentation

hep-score --registry dir:///PATH_TO_UNTARRED_WORKLOADS/ <workdir>

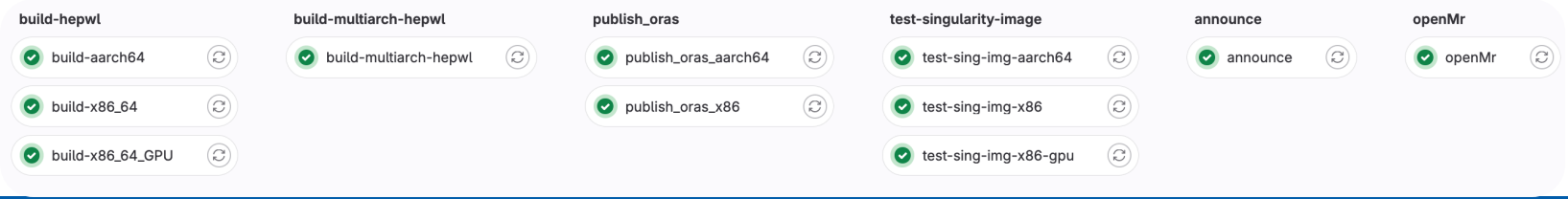
HEP Benchmark Project: Infrastructure

Build infrastructure of workloads' images

- GitLab CI/CD for fully automated build of container images (Docker)
  - Dedicated gitlab runners (VMs) are maintained



- Containers are built for multiple architectures: x86, aarch64, GPUs



Validation infrastructure

- Set of bare-metal nodes @ CERN used for
  - Workload validation
  - Test new benchmark features
  - Dedicated studies

CPU_Model	SMT	CPUs
AMD EPYC 7302 16-Core Processor	1	0-63
Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz	1	0-31
Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz	1	0-47
Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz	1	0-55
Intel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz	1	0-63
Intel(R) Xeon(R) Gold 6130 CPU @ 2.10GHz	1	0-63
Intel(R) Xeon(R) Gold 6326 CPU @ 2.90GHz	1	0-63
Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz	1	0-63

- In addition:
  - Access to new models available on-premise, at vendors' place, other sites
    - Advantage of the results publication in the central OpenSearch DB

Validations

- All new workload containers undergo a validation process
  - Multiple runs on the validation infrastructure
  - Check stability
  - Measure new score in case of algorithmic change

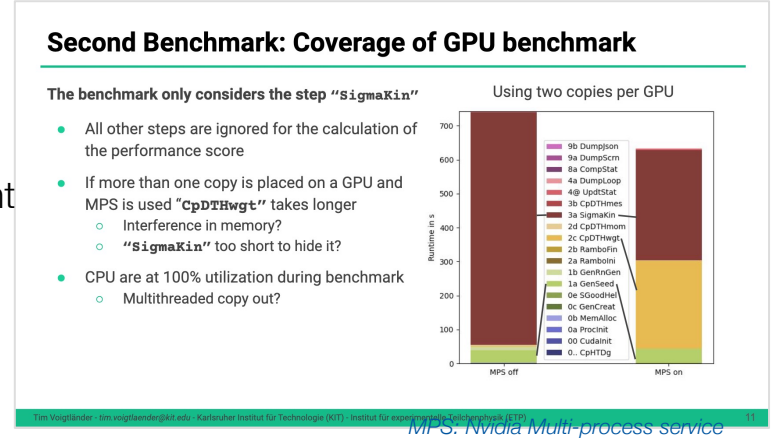
HEPscore Validation (Score)

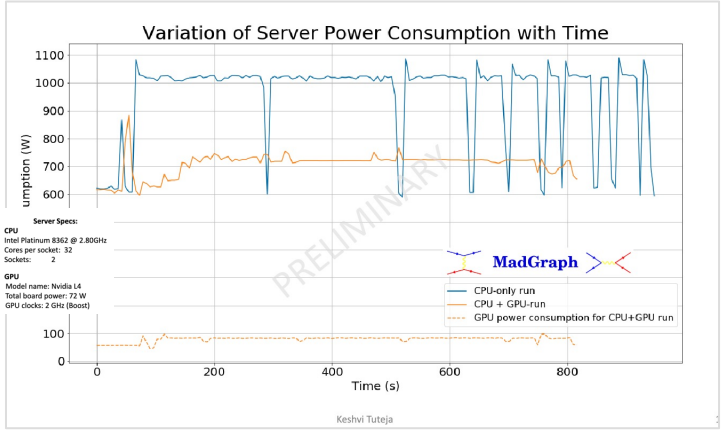
CPU_Model	Ratio ↑	Count HS23	Count contestant
Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz	0.979	115	145
AMD EPYC 7302 16-Core Processor	0.980	171	11
Intel(R) Xeon(R) Gold 6326 CPU @ 2.90GHz	0.983	117	218
Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz	0.996	112	145
Intel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz	0.997	19	151
Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz	0.998	114	151
Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz	0.999	112	147
Intel(R) Xeon(R) Gold 6130 CPU @ 2.10GHz	1.01	44	147

- All these modifications imply a change in the HEPScore configuration and a consequent change of hash
  - Need to prove that this doesn't affect the score, before replacing in production

Benchmarking CPU+GPU

- Introduces a new dimension we are not used to
  - Measuring ev/s processed by a server is just the starting point
- Need to study
  - % of utilization of the GPU vs CPU
  - Offloading on multiple GPUs
  - Energy consumption of the system and its components
- Workloads available (still evolving)
  - CMS HLT, Mardgraph4gpu @ LO, MLPF
- Work in progress, many opportunities for new contributions
  - A prototype version of HEPScore for CPU&GPU is foreseen for 2024





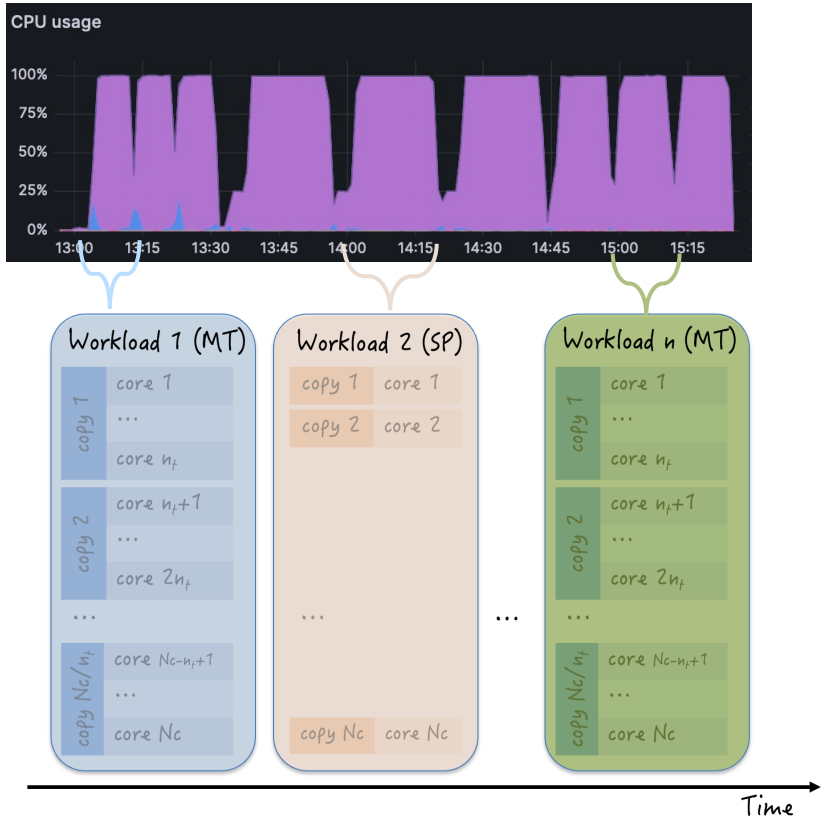
Summary

- After 1 year of increasing adoption, HEPScore23 confirms the expectations
- Improvements and new features will be released before summer in HEPScore v2.0
- GPU workloads exist, but we are still far from having an HEPScore for CPU+GPU
  - Opportunity for new contributors
- Looking forward to seeing more HS23 data in the central benchmark DB

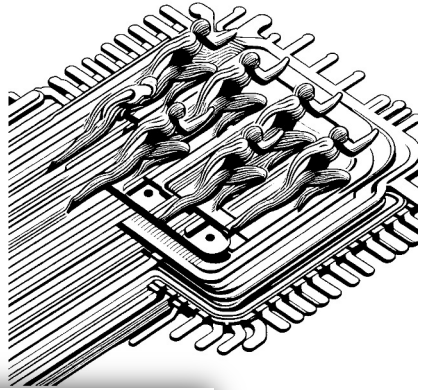

CERN

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



HEPScore definition



Similar functional definition of HS06. Components:

- a set of reference workloads (WLs)
- a measure of performance per WL (m_i): work done in unit of time
- a reference server

$$\bar{x} = \left(\prod_{i=1}^{n} x_i^{w_i}\right)^{1/\sum_{i=1}^{n} w_i}$$

https://en.wikipedia.org/wiki/Weighted_geometric_mean

The score S of a server (**srv**) is defined as the **geometric mean** of the **speed factors** x_i(**srv**,ref) = m_i(**srv**)/m_i(**ref**)

respect to the reference server (**ref**)

	WL_1		WL_2		WL_n		Score $\left(\prod_{i=1}^{n} x_i\right)^{\frac{1}{n}}$	S(A,B)
Ref. Srv	m_1(ref)	1 (by def)	m_2(ref)	1 (by def)	m_n(ref)	1 (by def)	1 (by def)	
Srv A	m_1(A)	x_1(A,ref)	m_2(A)	x_2(A,ref)	m_n(A)	x_n(A,ref)	S(A,ref)	$\frac{S(A,ref)}{S(B,ref)}$
Srv B	m_1(B)	x_1(B,ref)	m_2(B)	x_2(B,ref)	m_n(B)	x_n(B,ref)	S(B,ref)	