The journey towards HEPscore, the HEP-specific CPU benchmark for WLCG

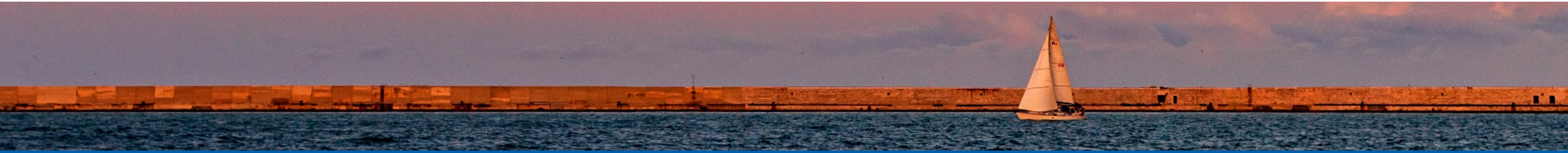
D. Giordano (CERN)

on behalf of

HEPiX Benchmarking WG & WLCG HEPscore Deployment TF

ACAT Bari

24/10/2022



CPU Benchmarking in HEP

Since 2009 the HEP community is using HEP-SPEC2006 (HS06) for pledges & accounting reports, procurement procedures, performance studies of CPUs

- Since 2018 HS06 is not supported anymore by the SPEC org.
  - The move to a new benchmark is desirable

“The first step in performance evaluation is to select the right measures of performance”

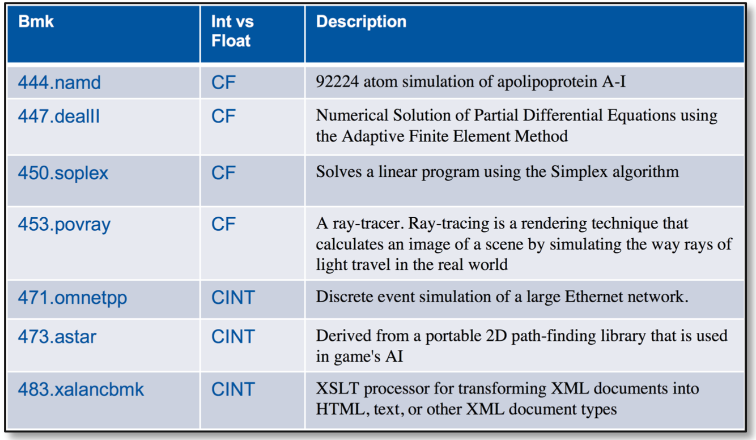
“The types of applications of computers are so numerous that it is not possible to have a standard measure of performance […] for all cases.”

From “*Art of Computer Systems Performance Analysis Techniques For Experimental Design Measurements Simulation And Modeling*” (Raj Jain , Wiley Computer Publishing, John Wiley & Sons, Inc)

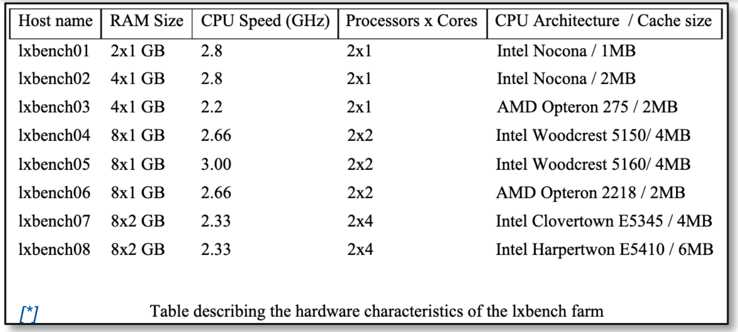
What HS06 is

HS06 is a suite of 7 C++ applications

- Subset of SPEC CPU® 2006 benchmark
  - SPEC's industry-standardized
- CPU-intensive benchmark suite
  - *NB: None of the applications is an event-based detector simulation or reconstruction*
- In **2009,** proven **high correlation** with experiment workloads on a set of servers of that epoch
- Execution time today of the full HS06 suite: O(3h)



Bmk	Int vs Float	Description
444.namd	CF	92224 atom simulation of apolipoprotein A-I
447.dealII	CF	Numerical Solution of Partial Differential Equations using the Adaptive Finite Element Method
450.soplex	CF	Solves a linear program using the Simplex algorithm
453.povray	CF	A ray-tracer. Ray-tracing is a rendering technique that calculates an image of a scene by simulating the way rays of light travel in the real world
471.omnetpp	CINT	Discrete event simulation of a large Ethernet network.
473.astar	CINT	Derived from a portable 2D path-finding library that is used in game's AI
483.xalancbmk	CINT	XSLT processor for transforming XML documents into HTML, text, or other XML document types



Host name	RAM Size	CPU Speed (GHz)	Processors x Cores	CPU Architecture / Cache size
lxbench01	2x1 GB	2.8	2x1	Intel Nocona / 1MB
lxbench02	4x1 GB	2.8	2x1	Intel Nocona / 2MB
lxbench03	4x1 GB	2.2	2x1	AMD Opteron 275 / 2MB
lxbench04	8x1 GB	2.66	2x2	Intel Woodcrest 5150/ 4MB
lxbench05	8x1 GB	3.00	2x2	Intel Woodcrest 5160/ 4MB
lxbench06	8x1 GB	2.66	2x2	AMD Opteron 2218 / 2MB
lxbench07	8x2 GB	2.33	2x4	Intel Clovertown E5345 / 4MB
lxbench08	8x2 GB	2.33	2x4	Intel Harpertwon E5410 / 6MB

[*] Table describing the hardware characteristics of the lxbench farm

[*] "A comparison of HEP code with SPEC benchmarks on multi-core worker nodes"
J. Phys.: Conf. Ser. 219 (2010) 052009
CHEP-09

HEP applications

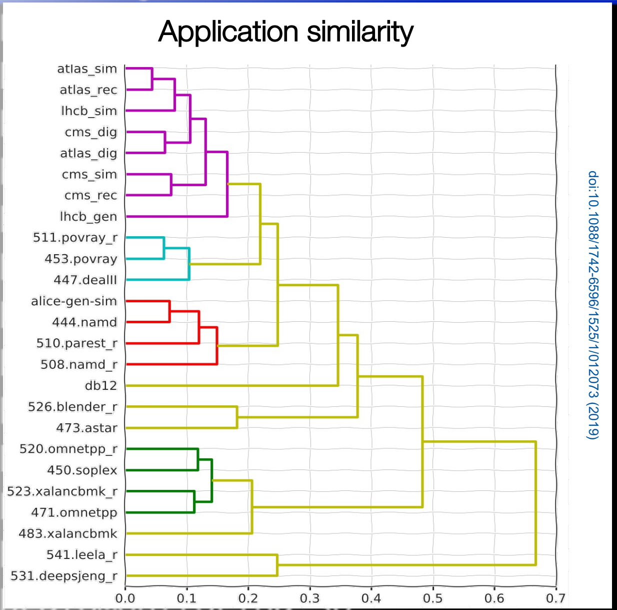
HEP applications consist of

- A cluster of **several hundred** algorithms
- **Complex framework**
- **No hotspots**, linear instruction spread
- Event based

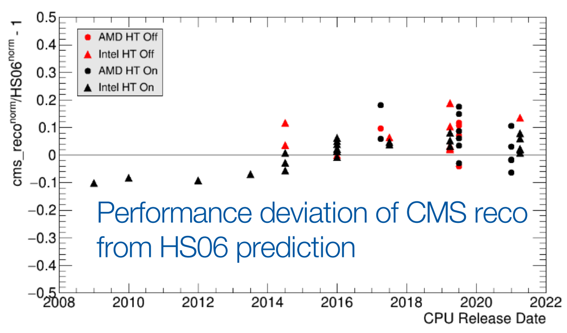
Experiment software is evolved since 2009

- Adoption of new programming approaches (multi-threading and vectorization) and heterogeneous resources (CPUs, GPUs)

In order to be predictive, the new benchmark must scale with the average performance of the job mix running in WLCG



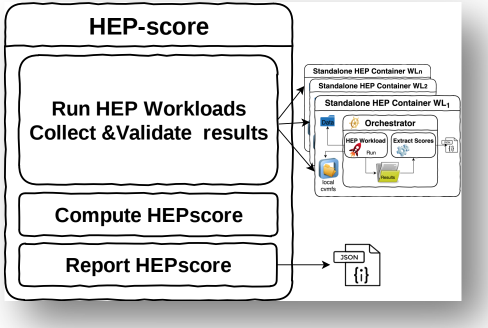
doi:10.1088/1742-6596/1525/1/012073 (2019)



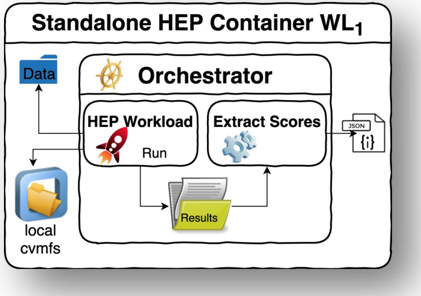
Performance deviation of CMS reco from HS06 prediction

HEP Benchmarks project

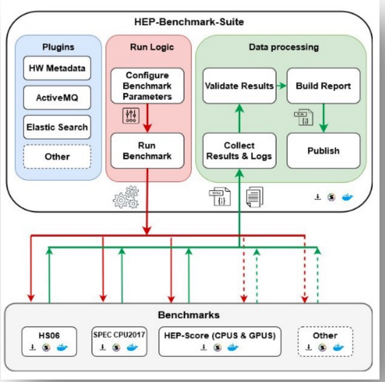
HEPscore has been proposed by the HEPiX Benchmarking WG
- Uses the workloads of the HEP experiments
- Combine them in a single benchmark score



HEPscore relies on HEP Workloads
- Individual **reference** HEP applications



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



All released under GPLv3 license

HEPscore definition

Ingredients:

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

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**)

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

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

	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)	

The challenge

Collect, maintain, extend workloads from several HEP experiments

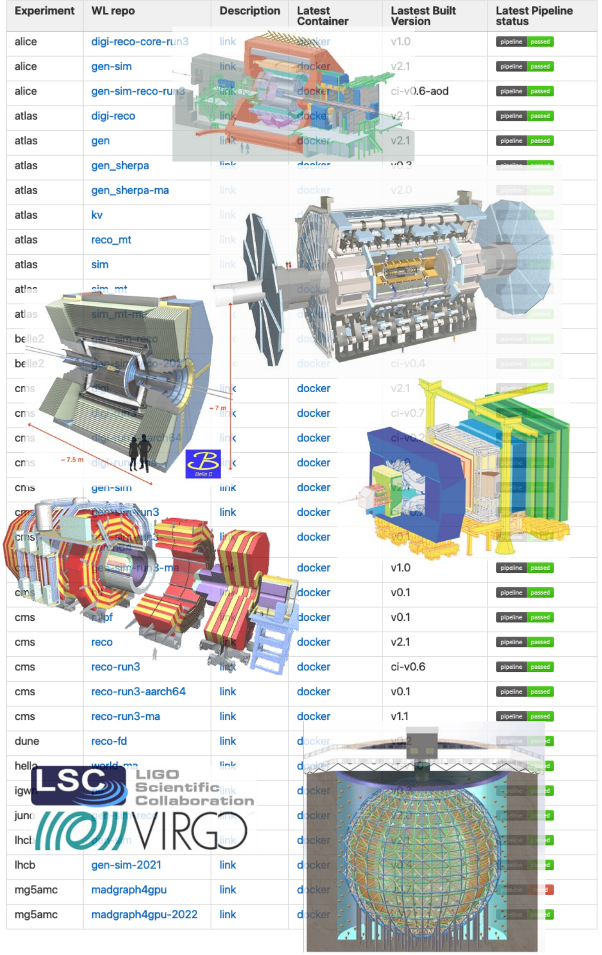
- Not affordable with ad-hoc recipes for each workload

More than **30 workloads from 7 experiments** prepared so far

- Experts from the Experiments focus on providing the workloads: software, data, result parser
- Experts on benchmarking focus on implementing a **unified** approach

Requirements

- Provide consistent CLI, report structure, error logging
- Reproducible results
- Zero burden from accessing remote data, databases, etc
- Not too large package distribution
- Portable
- Long term support

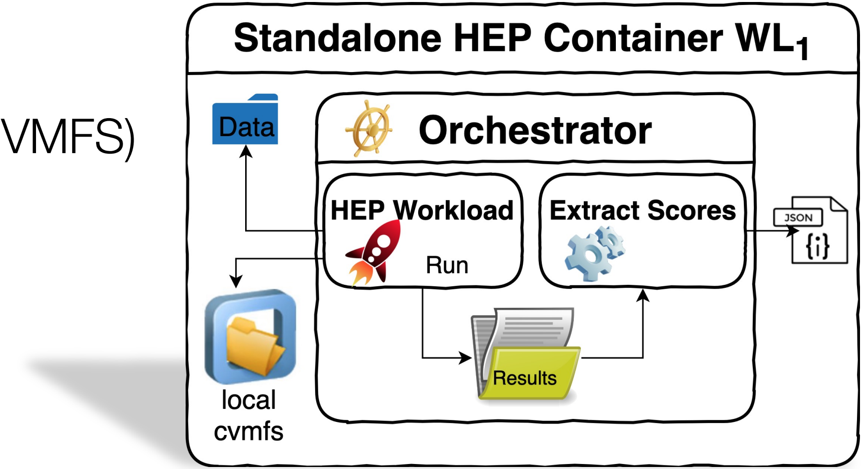


Experiment	WL repo	Description	Latest Container	Lastest Built Version	Latest Pipeline status
alice	digi-reco-core-run3	link	docker	v1.0	pipeline passed
alice	gen-sim	link	docker	v2.1	pipeline passed
alice	gen-sim-reco-run3	link	docker	ci-v0.6-aod	pipeline passed
atlas	digi-reco	link	docker	v2.1	pipeline passed
atlas	gen	link	docker	v2.1	pipeline passed
atlas	gen_sherpa	link	docker	v0.3	pipeline passed
atlas	gen_sherpa-ma	link	docker	v2.0	pipeline passed
atlas	kv	link	docker	[illegible]	pipeline passed
atlas	reco_mt	link	docker	[illegible]	[illegible]
atlas	sim	link	docker	[illegible]	pipeline passed
atlas	sim_mt	link	docker	ci-v1.0	[illegible]
atlas	sim_mt-ma	link	docker	[illegible]	pipeline passed
belle2	gen-sim-reco	link	docker	v2.3	pipeline passed
belle2	gen-sim-reco-2021	link	docker	ci-v0.4	pipeline passed
cms	digi	link	docker	v2.1	pipeline passed
cms	digi-run3	link	docker	ci-v0.7	pipeline passed
cms	digi-run3-aarch64	link	docker	ci-v0.2	pipeline passed
cms	digi-run3-ma	link	docker	v1.0	pipeline passed
cms	gen-sim	link	docker	v2.1	pipeline passed
cms	gen-sim-run3	link	docker	ci-v0.7	pipeline passed
cms	gen-sim-run3-aarch64	link	docker	v0.1	[illegible]
cms	gen-sim-run3-ma	link	docker	v1.0	pipeline passed
cms	[illegible]	link	docker	v0.1	pipeline passed
cms	mlpf	link	docker	v0.1	pipeline passed
cms	reco	link	docker	v2.1	pipeline passed
cms	reco-run3	link	docker	ci-v0.6	pipeline passed
cms	reco-run3-aarch64	link	docker	v0.1	pipeline passed
cms	reco-run3-ma	link	docker	v1.1	pipeline passed
dune	reco-fd	link	docker	v0.2	pipeline passed
hello	world-ma	link	docker	ci-v1.0	pipeline passed
igwn	pe	link	docker	v0.3	pipeline passed
juno	gen-sim-reco	link	docker	v2.0	pipeline passed
lhcb	gen-sim	link	docker	v2.1	pipeline passed
lhcb	gen-sim-2021	link	docker	v0.4	pipeline passed
mg5amc	madgraph4gpu	link	docker	[illegible]	pipeline failed
mg5amc	madgraph4gpu-2022	link	docker	[illegible]	pipeline passed

Standalone containers of HEP workloads

Components of an HEP workload

- SW repository (in general distributed via CVMFS)
- Input data (event and conditions data)
- An orchestrator script per workload
  - Sets the environment
  - Runs the application
  - Parses the output to generate scores

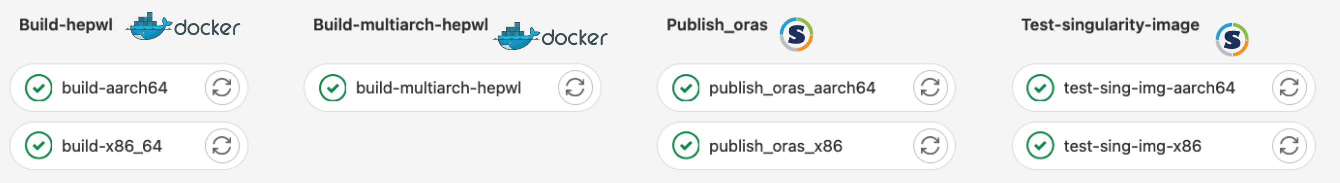


Strategy: build **standalone containers** encapsulating **all and only** the dependencies needed to run the benchmarks

Workloads' repository

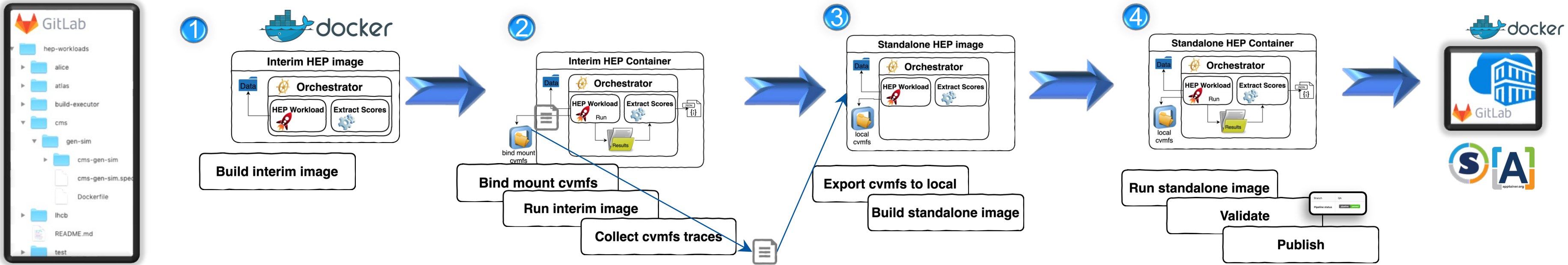
Realized an effective infrastructure to **build and distribute** the HEP workloads

Containers are built for multiple architectures: x86, aarch64, (potentially) Power, GPUs



Build-hepwl	Build-multiarch-hepwl	Publish_oras	Test-singularity-image
✓ build-aarch64	✓ build-multiarch-hepwl	✓ publish_oras_aarch64	✓ test-sing-img-aarch64
✓ build-x86_64		✓ publish_oras_x86	✓ test-sing-img-x86

- **CVMFS Trace** and **Export** [ref] utilities to export the applications' software from cvmfs to local
- GitLab **CI/CD** for fully **automated** continuous integration
- GitLab **Registry** for container distribution (Docker & Singularity/Apptainer)



1 Build interim image

2 Bind mount cvmfs; Run interim image; Collect cvmfs traces

3 Export cvmfs to local; Build standalone image

4 Run standalone image; Validate; Publish

A long process to adopt a new benchmark

2017, WLCG Workshop Manchester

- First proposal of HEP Benchmark with containerized HEP applications (HEPiX Benchmarking WG)

2018/2020

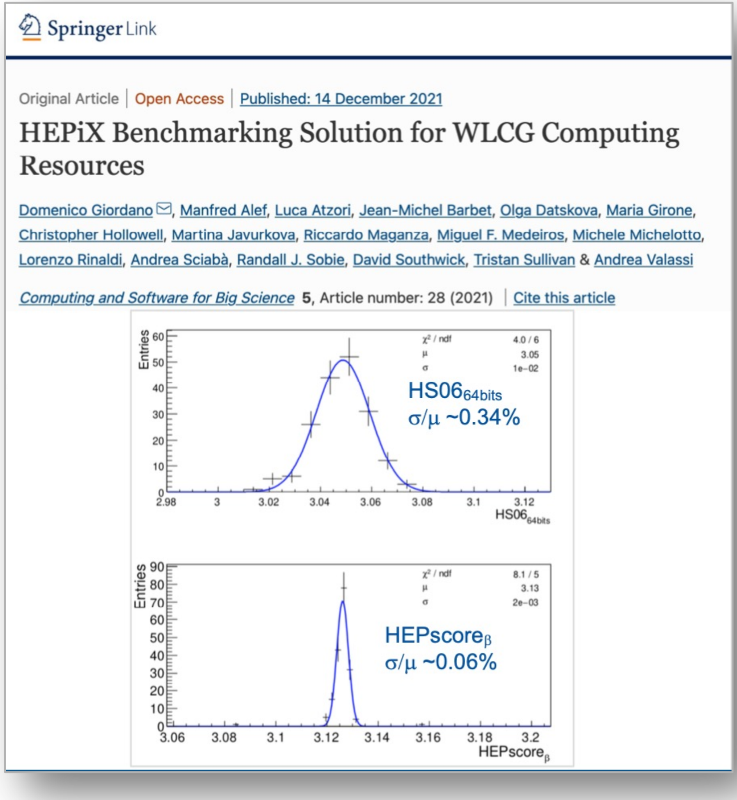
- Design, prototype, validate, deliver

2020/2021

- Proven the technical feasibility of an HEP-Benchmark: HEPscore$_\beta$ using experiments' applications from LHC Run 2 and Belle2
- Include new applications: from LHC Run 3, Juno, IGWN

2022

- Finalize the HEPscore composition
- Discuss a WLCG strategy for the transition from HS06 to HEPscore



SpringerLink

Original Article | Open Access | Published: 14 December 2021

HEPiX Benchmarking Solution for WLCG Computing Resources

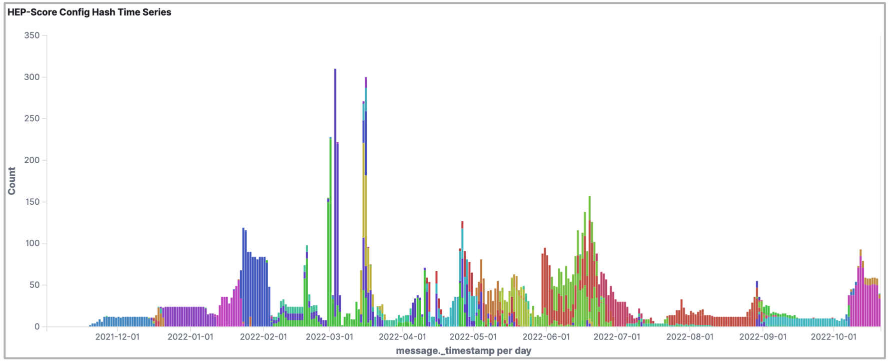
Domenico Giordano, Manfred Alef, Luca Atzori, Jean-Michel Barbet, Olga Datskova, Maria Girone, Christopher Hollowell, Martina Javurkova, Riccardo Maganza, Miguel F. Medeiros, Michele Michelotto, Lorenzo Rinaldi, Andrea Sciabà, Randall J. Sobie, David Southwick, Tristan Sullivan & Andrea Valassi

Computing and Software for Big Science **5**, Article number: 28 (2021) | Cite this article

2022 measurement campaign

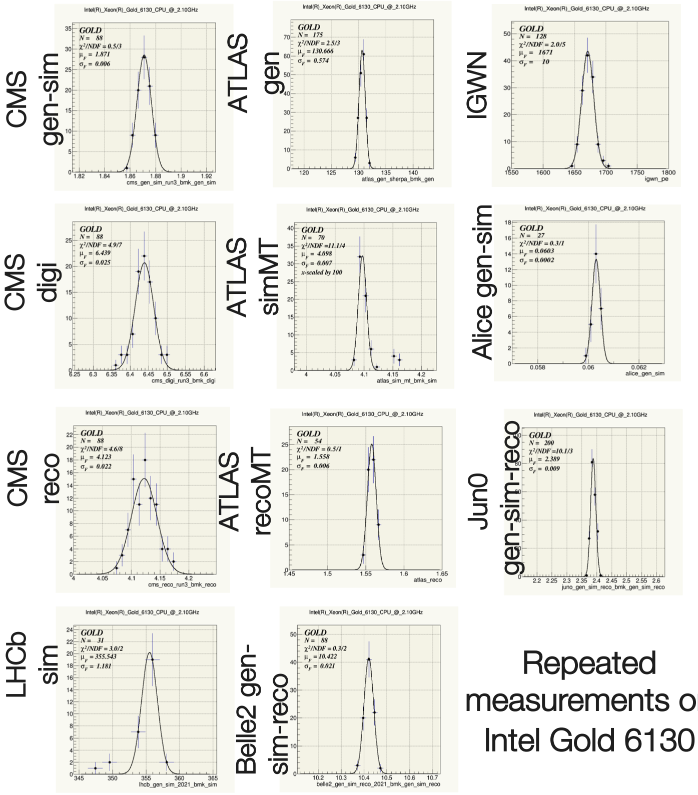
Executed the 11 most recent workloads from LHC experiments, Belle2, Juno, IGWN

- On ~40 different CPU models from ~15 WLCG sites



Measured

- Robustness against failures
- Resolution (σ/μ typically < 1%)
- Performance

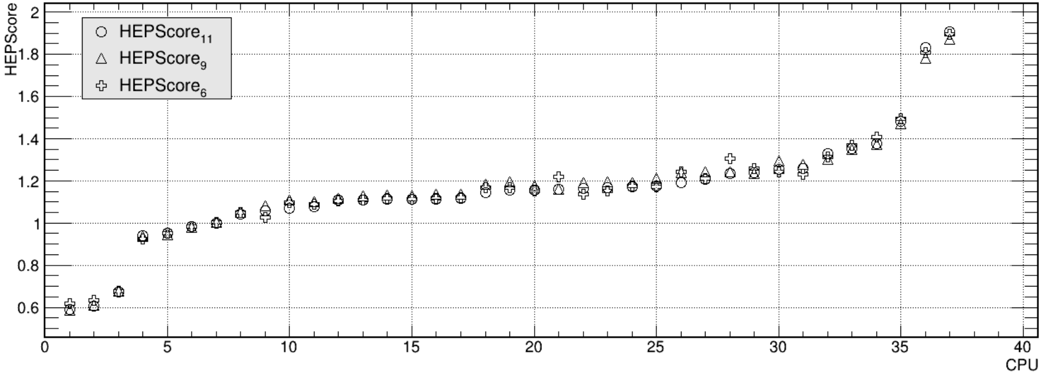


Repeated measurements on Intel Gold 6130

HEPscore candidates

Several combinations investigated:

- All workloads/Exclude the long-running ones/Use a subset/No weights/Weights from Grid fraction of jobs
- Little difference (**few %**) between the candidates



Preference for

- A small workload set for shorter runtime
- The simplest approach of unweighted WLs
- HEPscore composition of **7 workloads**:
  - Alice (digi_reco),
    Atlas (gen_sherpa, reco),
    CMS (gen_sim, reco),
    LHCb (sim), Belle2 (gen_sim_reco)
  - Important to include the Alice Reco workload: reconstruction of Pb-Pb events

Desiderata: **at least some workloads should run on ARM**

Growing awareness and consensus

2 days of HEPscore workshop (19–20 Sept 2022)

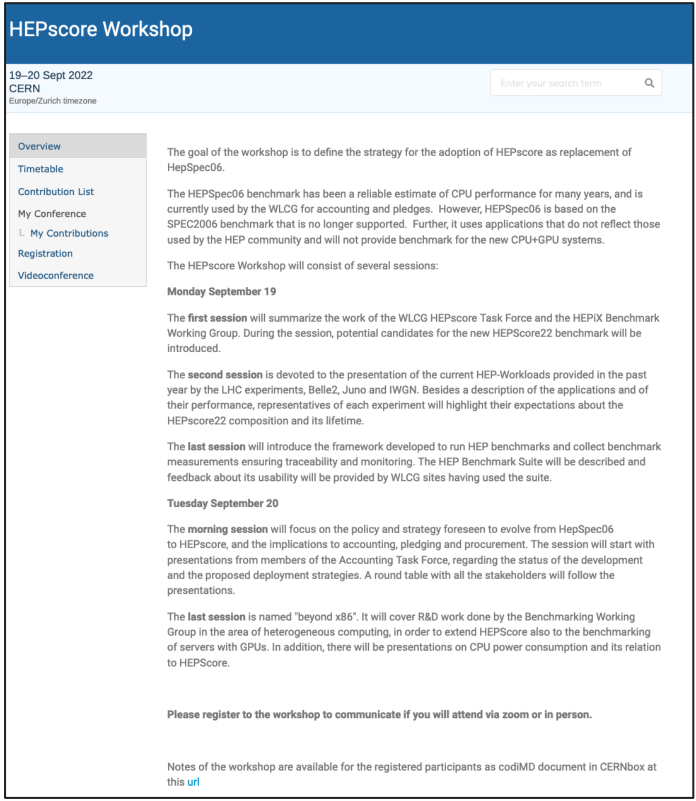
- Good representation of the different parties involved: Experiments, Sites, WLCG board

Valuable feedback from our WLCG community on

- Proposed composition of HEPscore
- Usability of the HEP Benchmark Suite
- Strategy for the adoption of HEPscore as WLCG benchmark

Next steps:

- Collect feedback from ACAT + HEPiX Autumn (next week) + WLCG workshop (in 2 weeks)
- Submit a recommendation to the WLCG MB by the end of 2022



HEPscore Workshop

19–20 Sept 2022
CERN
Europe/Zurich timezone

The goal of the workshop is to define the strategy for the adoption of HEPscore as replacement of HepSpec06.

The HEPSpec06 benchmark has been a reliable estimate of CPU performance for many years, and is currently used by the WLCG for accounting and pledges. However, HEPSpec06 is based on the SPEC2006 benchmark that is no longer supported. Further, it uses applications that do not reflect those used by the HEP community and will not provide benchmark for the new CPU+GPU systems.

The HEPscore Workshop will consist of several sessions:

Monday September 19

The **first session** will summarize the work of the WLCG HEPscore Task Force and the HEPiX Benchmark Working Group. During the session, potential candidates for the new HEPScore22 benchmark will be introduced.

The **second session** is devoted to the presentation of the current HEP-Workloads provided in the past year by the LHC experiments, Belle2, Juno and IWGN. Besides a description of the applications and of their performance, representatives of each experiment will highlight their expectations about the HEPscore22 composition and its lifetime.

The **last session** will introduce the framework developed to run HEP benchmarks and collect benchmark measurements ensuring traceability and monitoring. The HEP Benchmark Suite will be described and feedback about its usability will be provided by WLCG sites having used the suite.

Tuesday September 20

The **morning session** will focus on the policy and strategy foreseen to evolve from HepSpec06 to HEPscore, and the implications to accounting, pledging and procurement. The session will start with presentations from members of the Accounting Task Force, regarding the status of the development and the proposed deployment strategies. A round table with all the stakeholders will follow the presentations.

The **last session** is named "beyond x86". It will cover R&D work done by the Benchmarking Working Group in the area of heterogeneous computing, in order to extend HEPScore also to the benchmarking of servers with GPUs. In addition, there will be presentations on CPU power consumption and its relation to HEPScore.

Please register to the workshop to communicate if you will attend via zoom or in person.

Notes of the workshop are available for the registered participants as codiMD document in CERNbox at this url

https://indico.cern.ch/event/1170924/

Extend HEPscore to heterogeneous resources

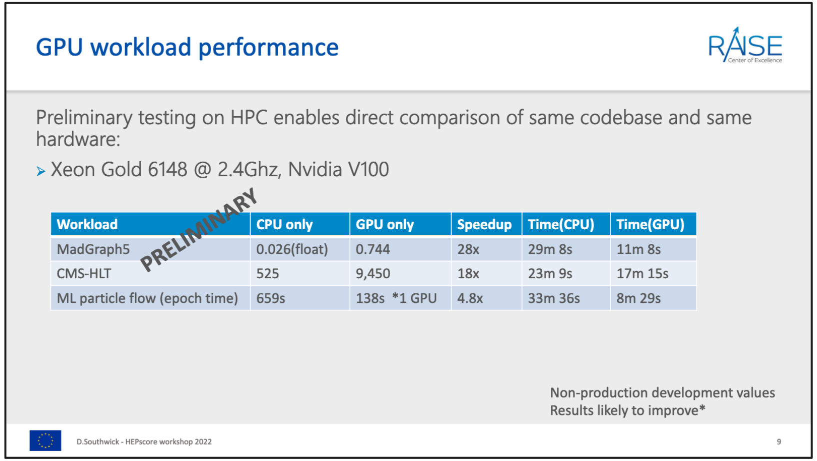
In the future WLCG resources will include GPUs

- This is already true for the online farms
- HEP experiments have/are re-writing their offline applications to use also **GPUs**

HEP Benchmark project:
growing support for heterogeneous workloads

- Madgraph4gpu
- CMS HLT-like
- ML/AI train AI model (e.g. MLPF)

Prototypes of analysis workloads are also available



GPU workload performance

RAISE
Center of Excellence

Preliminary testing on HPC enables direct comparison of same codebase and same hardware:

- Xeon Gold 6148 @ 2.4Ghz, Nvidia V100

PRELIMINARY

Workload	CPU only	GPU only	Speedup	Time(CPU)	Time(GPU)
MadGraph5	0.026(float)	0.744	28x	29m 8s	11m 8s
CMS-HLT	525	9,450	18x	23m 9s	17m 15s
ML particle flow (epoch time)	659s	138s *1 GPU	4.8x	33m 36s	8m 29s

Non-production development values
Results likely to improve*

D.Southwick - HEPscore workshop 2022

9

All this is still too premature to be included in a production HEPscore

Conclusions

The replacement of HS06 with HEPscore for CPUs will very likely happen in 2023

Enabling technologies

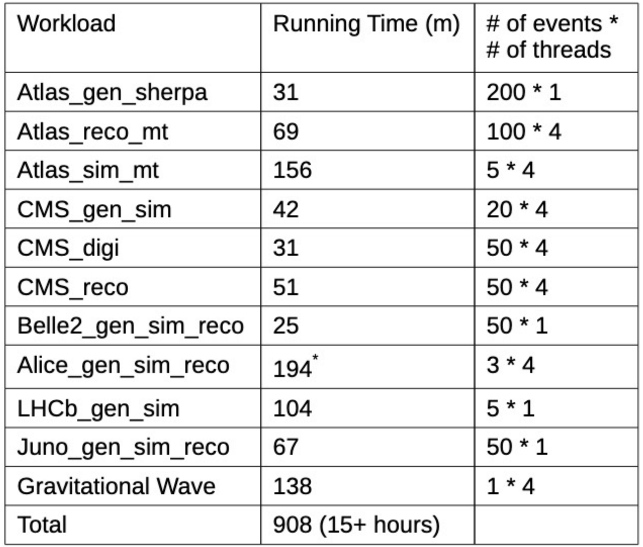
- Implemented framework to snapshot HEP applications in containers
- Created in the last years ~30 standalone containers of HEP workloads
  - Some run on x86 and aarch64
- Deployed a benchmark database and released a software suite to collect benchmark results

Validation

- Performed a large-scale measurement campaign in 2022
- Identified the “golden” HEPscore composition of 7 workloads

CERN

HEP workloads running time



Workload	Running Time (m)	# of events * # of threads
Atlas_gen_sherpa	31	200 * 1
Atlas_reco_mt	69	100 * 4
Atlas_sim_mt	156	5 * 4
CMS_gen_sim	42	20 * 4
CMS_digi	31	50 * 4
CMS_reco	51	50 * 4
Belle2_gen_sim_reco	25	50 * 1
Alice_gen_sim_reco	194[*]	3 * 4
LHCb_gen_sim	104	5 * 1
Juno_gen_sim_reco	67	50 * 1
Gravitational Wave	138	1 * 4
Total	908 (15+ hours)	

Times for three runs on reference machine

[*] - Alice reco currently not included in benchmark score, due to technical problems with reco workload. Reco is ~ 50% of running time. Once issue is resolved, could run only reco to shorten workload length.

Two teams collaborate for this objective

HEPiX Benchmarking WG

Roles

- Evaluation of benchmark alternatives
- Design and development of the **HEP Benchmarks project**
- Validation of the HEP workloads
- Analysis of benchmark measurements

Team of ~13 people

Active (again) since 2018

WLCG HEPscore deployment TF

Roles

- Recommend the HEPscore composition
- Propose migration HS06->HEPscore
- Coordinate the collection of workloads
- Onboard WLCG sites for validation

Team of ~20 people

Started on Nov 4. 2020

CERN